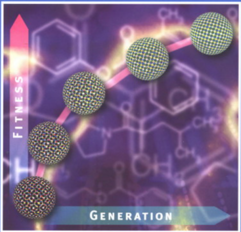
WILEY-VCH

Evolutionary Algorithms in Molecular Design

Edited by David E. Clark



Methods and Principles in Medicinal Chemistry

Volume 8

Edited by
R. Mannhold,
H. Kubinyi,
H. Timmerman

Evolutionary Algorithms in Molecular Design

Edited by David E. Clark

WILEY-VCH

Methods and Principles in Medicinal Chemistry

Edited by
R. Mannhold
H. Kubinyi
H. Timmerman

Evolutionary Algorithms in Molecular Design

Edited by David E. Clark

WILEY-VCH

Weinheim · New York · Chichester · Brisbane · Singapore · Toronto

Series Editors:
Prof. Dr. Raimund Mannhold
Biomedical Research Center
Molecular Drug Research Group
Heinrich-Heine-Universität
Universitätsstraße 1
40225 Düsseldorf
Germany

Prof. Dr. Hugo Kubinyi
ZHF/G, A 30
BASF AG
67056 Ludwigshafen
Germany

Prof. Dr. Hendrik Timmerman
Faculty of Chemistry
Dept. of Pharmacochemistry
Free University of Amsterdam
De Boelelaan 1083
1081 HV Amsterdam
The Netherlands

Volume Editor:
Dr. David E. Clark
Computer-Aided Drug Design
Aventis Pharma Ltd.
Dagenham Research Centre
Rainham Road South
Dagenham
Essex RM10 7XS
United Kingdom

This book was carefully produced. Nevertheless, authors, editors and publisher do not warrant the information contained therein to be free of errors. Readers are advised to keep in mind that statements, data, illustrations, procedural details or other items may inadvertently be inaccurate.

Library of Congress Card No. applied for.

British Library Cataloguing-in-Publication Data: A catalogue record for this book is available from the British Library.

Die Deutsche Bibliothek – CIP Cataloguing-in-Publication-Data
A catalogue record for this publication is available from Die Deutsche Bibliothek

ISBN 3-527-30155-0

Printed on acid-free and chlorine-free paper.

Composition: Datascan GmbH, D-67346 Speyer
Printing: Strauss Offsetdruck. D-69509 Mörlenbach
Bookbinding: I. Schäffer GmbH & Co. KG., D-67269 Grünstadt

Printed in the Federal Republik of Germany.

Contents

Preface

Nature has solved its most complicated problem, the creation, variation, and improvement of living organisms, in a simple and efficient manner. Starting from primitive forms in earth history, mutation and crossover produced variations that had to struggle for their existence and to compete with their ancestors and genetically different variations. Since the pioneering work of Charles Darwin "On the Origin of Species by Means of Natural Selection", published in 1859, we know about these mechanisms and understand, in principle, the development of higher organisms from the very first reproducing organic systems.

If we take a closer look, we see that mankind developed science and technology, following the very same pattern. Discoveries, as well as major and minor technical improvements led to the development of watches, cars, computers, and all kinds of machines. Likewise, in drug research, most new remedies were derived and are still developed from "simple" lead structures, e.g., bioactive plant products, endogenous hormones, neurotransmitters, or screening hits. The bioisosteric replacement of atoms and functional groups (i.e., the mutations of a chemical structure) and the introduction of active parts of other interesting molecules (corresponding to a crossover between two leads) are performed in several iterative cycles, in order to improve the affinity of the compounds to a certain biological target, to improve selectivity, bioavailability and duration of action, and to reduce side effects and toxicity. Thus, without defining their research work as a Darwinian process, medicinal chemists have indeed applied evolutionary principles over the past 100 years! The final drugs always resulted from the "survival of the fittest".

Taking all this into account, it is highly surprising that a systematic investigation of such strategies for mathematical and technical optimization problems started relatively late. The books "Evolutionsstrategie", by Ingo Rechenberg (Germany, 1973), and "Adaptation in Natural and Artificial Systems", by John Holland (USA, 1975), laid the foundation of evolutionary algorithms. Their enormous potential for complex optimization problems, e.g., the travelling salesman or packing problems, has been fully realized in the past two decades. In the meantime, many more problems have been approached by EAs. Everybody who has personal experience in the application of these techniques will agree that the simplicity of programming of these algorithms contrasts significantly with the astonishing speed of the calculations and the high degree of approximation to the "best" solutions. Problems, where no complete simulation is possible due to an intractable number of possible solutions and where even Monte Carlo methods would need years, are solved in a matter of seconds or minutes.

In a series on "Methods and Principles in Medicinal Chemistry" it is now highly appropriate to review the state of the art in the application of evolutionary techniques in different fields of drug research, from conformational analyses of small molecules, docking, *de novo* design, QSAR, and diversity analysis to 3D structure determination and protein folding, to mention only some topics. Thus, the Editors of this series are grateful to David Clark, who has assembled with enthusiasm, ongoing drive and unprecedented success a team of experts in different fields of drug design and stimulated them to produce this informative book within very short time.

March 2000

Raimund Mannhold, Düsseldorf
Hugo Kubinyi, Ludwigshafen
Henk Timmerman, Amsterdam

A Personal Foreword

My first encounter with evolutionary algorithms (EAs) was in the early 1990s during my Ph.D. research at the University of Sheffield where I studied under the guidance of Professor Peter Willett. At that time, the number of published applications of EAs in the field of molecular design could probably have been counted on the fingers of one hand. The publication of this book, less than ten years later, bears witness to the rapidity with which EAs have been assimilated into the discipline, and to the ubiquity of their application.

Since moving into industry, I have maintained a keen interest in EAs, particularly their application to the problems encountered in molecular design situations. One of the fruits of this "hobby" is a regularly updated web-based bibliography (http://panizzi.shef.ac.uk/cisrg/links/ea_bib.html) kindly hosted at the University of Sheffield by Dr. Val Gillet. It was a chance encounter with this site that prompted Hugo Kubinyi to invite me to edit this book – an invitation that I was delighted to accept.

The aim of this book is to provide an up-to-date and comprehensive survey of the applications of EAs in molecular design. It is hoped that this will be of use both to medicinal chemists – who may not be very familiar with EAs – and also to more seasoned molecular design professionals – who will probably have encountered them at one time or another. To aid the former, we have provided an introductory chapter to explain the major terms and concepts and, for the latter, a concluding chapter is included that examines some of the newer techniques entering the field, together with some possible future directions.

Compiling a book such as this can be hard work! An oft-quoted verse from the Bible sums it up: "Of making many books there is no end, and much study wearies the body" (Ecclesiastes 12:12, New International Version). My task as editor would not have been possible without the help of many people. First of all, it has been my pleasure and privilege to work with a group of authors who are experts in the application of EAs to their chosen fields within the broad area of molecular design research. I would like to thank all the authors for their patience, co-operation and sheer hard work in the preparation of this volume. I am also grateful to Dr. Hugo Kubinyi for his invitation to edit the book and for the benefit of his editorial experience along the way. The staff at Wiley-VCH deserve my thanks for their help and expertise at each stage of the production process, particularly Dr. Gudrun Walter, Peter Biel and Elke Lentz. In addition, I am grateful to Dr. Stephen Pickett for granting me permission to work on this project and to my employer, Aventis Pharma, for providing the necessary facilities. Last, but not least, I am indebted to my wife, Barbara, and my young son, Theo, who have encouraged and supported me in this "labor of love" and kept me from falling into the despondency of Eeyore: "This writing business. Pencils and what-not. Over-rated, if you ask me. Silly stuff. Nothing in it" (*Eeyore's Little Book of Gloom*, Egmont Children's Books, 1999).

I hope that, in whatever capacity you read this book, you will find it useful, stimulating and enjoyable.

April 2000 David E. Clark, Dagenham

List of Contributors

Professor Lutgarde M. C. Buydens

Department of Chemometrics
University of Nijmegen
Toernooiveld 1
6525 ED Nijmegen
The Netherlands
Tel.: +31 24 365 3180
Fax: +31 24 365 2653
E-mail: lbuydens@sci.kun.nl

David E. Clark, PhD

Computer-Aided Drug Design
Aventis Pharma Ltd.
Dagenham Research Centre
Rainham Road South
Dagenham
Essex
RM10 7XS
United Kingdom
Tel.: +44 181 919 3353
Fax: +44 181 919 2029
E-mail: david-e.clark@aventis.com

Dr. Valerie J. Gillet

Department of Information Studies
University of Sheffield
Sheffield
S10 2TN
United Kingdom
Tel.: +44 114 222 2652
Fax: +44 114 278 0300
E-mail: v.gillet@sheffield.ac.uk

Dr. David S. Goodsell

Molecular Graphics Laboratory, MB-5
The Scripps Research Institute
10550 North Torrey Pines Road
La Jolla, California 92037-1000
USA
Tel.: +1 858 784 2839
Fax: +1 858 784 2860
E-mail: goodsell@scripps.edu

Professor Kenneth D. M. Harris

School of Chemistry
University of Birmingham
Edgbaston
Birmingham
B15 2TT
United Kingdom
Tel.: +44 121 414 7474
Fax: +44 121 414 7473
E-mail: k.d.m.harris@bham.ac.uk

Dr. Roy L. Johnston

Department of Chemistry
University of Birmingham
Edgbaston
Birmingham
B15 2TT
United Kingdom
Tel.: +44 121 414 7477
Fax: +44 121 414 4403
E-mail: roy@tc.bham.ac.uk

Dr. Benson M. Kariuki

Department of Chemistry
University of Birmingham
Edgbaston
Birmingham
B15 2TT
United Kingdom
Tel.: +44 121 414 7481
Fax: +44 121 414 4403
E-mail: b.m.kariuki@bham.ac.uk

Dr. Garrett M. Morris

Molecular Graphics Laboratory, MB-5
The Scripps Research Institute
10550 North Torrey Pines Road
La Jolla, California 92037-1000
USA
Tel.: +1 858 784 2292
Fax: +1 858 784 2860
E-mail: garrett@scripps.edu

Professor Arthur J. Olson

Molecular Graphics Laboratory
The Scripps Research Institute
10550 North Torrey Pines Road
La Jolla, California 92037-1000
USA
Tel.: +1 858 784 9702
Fax: +1 858 784 2860
E-mail: olson@scripps.edu

Dr. Abby L. Parrill

Department of Chemistry
University of Memphis
Memphis
Tennessee 38152
USA
Tel.: +1 901 678 2638
Fax: +1 901 678 3447
E-mail: aparrill@memphis.edu

Dr. Jan T. Pedersen

Department of Computational Chemistry
H. Lundbeck A/S
Ottiliavej 9
DK-2500
Valby
Copenhagen
Denmark
Tel.: +45 3644 2425
E-mail: jatp@lundbeck.com

Professor Bryan C. Sanctuary

Department of Chemistry
McGill University
801 Sherbrooke Street W.
Montreal
H3A 2K6
Canada
Tel.: +1 514 398 6930
Fax: +1 514 398 3797
E-mail: bryans@chemistry.mcgill.ca

Dr. Sung-Sau So

Hoffmann-La Roche, Inc.
Preclinical R&D
340 Kingsland Street
Nutley
New Jersey 07110-1199
USA
Tel.: +1 973 235 2193
Fax: +1 973 235 2682
E-mail: sung-sau.so@roche.com

Dr. Andrew Tuson

Department of Computing
City University
London
EC1V 0HB
United Kingdom
Tel.: +44 20 7477 8164
Fax: +44 20 7477 8587
E-mail: andrewt@soi.city.ac.uk

Dr. Lutz Weber

Morphochem AG
Gmunder Str. 37-37a
D-81379 Munich
Germany
Tel.: +49 89 78005 0
Fax: +49 89 78005 555
E-mail: lutz.weber@morphochem.de

Dr. Ron Wehrens

Department of Chemometrics
University of Nijmegen
Toernooiveld 1
6525 ED Nijmegen
The Netherlands
Tel.: +31 24 365 2053
Fax: +31 24 365 2653
E-mail: rwehrens@sci.kun.nl

Professor Peter Willett

Department of Information Studies
University of Sheffield
Sheffield
S10 2TN
United Kingdom
Tel.: +44 114 222 2633
Fax: +44 114 278 0300
E-mail: p.willett@sheffield.ac.uk

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

1 Introduction to Evolutionary Algorithms

Abby L. Parrill

Abbreviations

EA	Evolutionary algorithm
EP	Evolutionary programming
ES	Evolution strategy
GA	Genetic algorithm
HPLC	High-performance liquid chromatography
QSAR	Quantitative structure–activity relationship
QSPR	Quantitative structure–property relationship
RMS	Root mean square

1.1 History and Biological Motivation

The past 30 years have seen the independent development of three biologically motivated computational problem-solving methods [1–3] grouped together under the term, *evolutionary algorithms* (EAs). The three parent methods share a biological foundation, the basic principles of Darwinian evolution [4], but differ in their computational implementation of these evolutionary principles [5]. The shared biological foundation of EAs includes treatment of proposed problem solutions as members of a population that vary in adaptation to their environment, or *fitness*. These population members are subjected to *selection pressure*, and survivors breed offspring by the application of genetic operations that may include *mutation*, *crossover* (also called recombination), or both. Optimization proceeds for a number of generations after which fitter population members have evolved from the original population. A general scheme for EAs is demonstrated in Fig. 1. The three parent methods that follow this general scheme are *genetic algorithms* (GAs) [3, 6–8], *evolutionary programming* (EP) [1, 9], and *evolution strategies* (ESs) [2, 10].

The original variants of EAs were initially applied to different types of problems, as well as having differences in implementation. GAs were initially developed as computer models of natural adaptation [3], and have since been extended to many other problem types including pattern recognition and parameter optimization in many different fields [7]. EP was initially used to optimize predictions made by finite state machines through evolution of the state transition tables of the machines [1]. The subsequent application of EP to other problem types has paralleled the expansion of GAs [9, 11]. ESs were first applied to experimental hydrodynamic problems [12]. Subsequent use of ESs has been

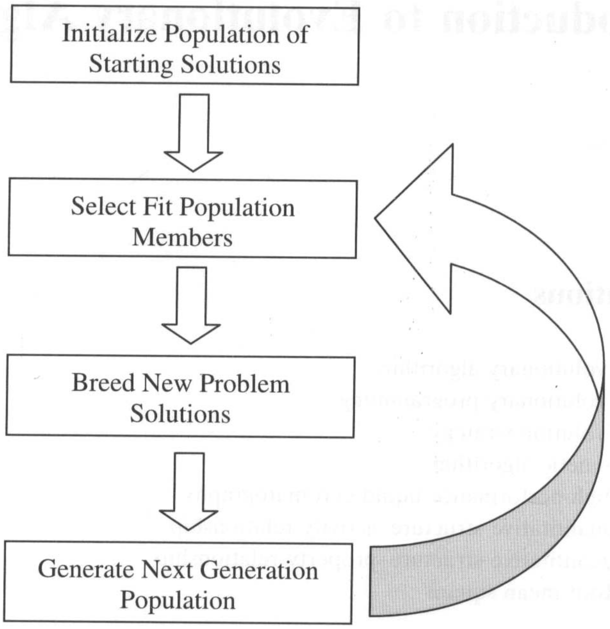

Initialize Population of Starting Solutions
Select Fit Population Members
Breed New Problem Solutions
Generate Next Generation Population

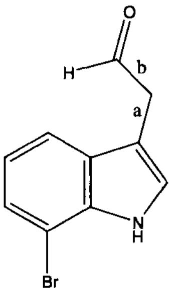
Figure 1. General scheme used by evolutionary algorithms for problem solving.

largely confined to computer science research circles, and has resulted in the strongest theoretical basis of the three methods [2, 5], although fewer practical applications have been published using ESs. Readers interested in a more detailed history of the development of EAs should refer to the recent publication of David Fogel [13]. The implementations of these three methods are outlined and compared throughout the following section in a qualitative manner. Readers interested in a more mathematical overview and comparison of these methods are referred to the work of Bäck and Schwefel [5].

1.2 Descriptive Comparison of Algorithms

1.2.1 Representation

The implementation of any EA begins with the *encoding* of the problem solutions. Historically, GAs have used *binary encodings* whereas ESs and EP have used *real number encodings* [11]. Each encoding method displays distinct advantages. The binary encoding requires translation to and from the natural representation of the problem. Although this requires additional steps, it means that expansion of the GA methodology to a new type of problem requires only the development of code to translate to and from a binary representation. Real number encoding, however, is often a much more natural representation of the problem, especially in scientific problems. Regardless of the encoding type, a coded problem solution is referred to as the *genotype*, and it can be used to determine the decoded *phenotype*. The genotype can also be called a *chromosome*, and consists of a collection of *genes*. Genes are exemplified in the following example and are shown in Fig. 2.



Solution Representation Requires 8 Genes:

1. Dihedral Angle Labeled a
2. Dihedral Angle Labeled b
3. Rotation of entire molecule around x axis relative to standard orientation
4. Rotation of entire molecule around y axis relative to standard orientation
5. Rotation of entire molecule around z axis relative to standard orientation (Binary example uses 3 bits to encode 360 degrees at 45 degree increments)
6. Distance in x direction from standard orientation
7. Distance in y direction from standard orientation
8. Distance in z direction from standard orientation (Binary example uses 3 bits to encode from –2.0 to +1.5 in 0.5 Å increments)

Binary Representation:

Gene: 1			2			3			4			5			6			7			8		
0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	0	0	1	0	1	0	1	1	0

Real-Valued Representation

Gene:	1	2	3	4	5	6	7	8
	120.0	118.2	45.3	121.2	10.1	1.0	1.3	0.8

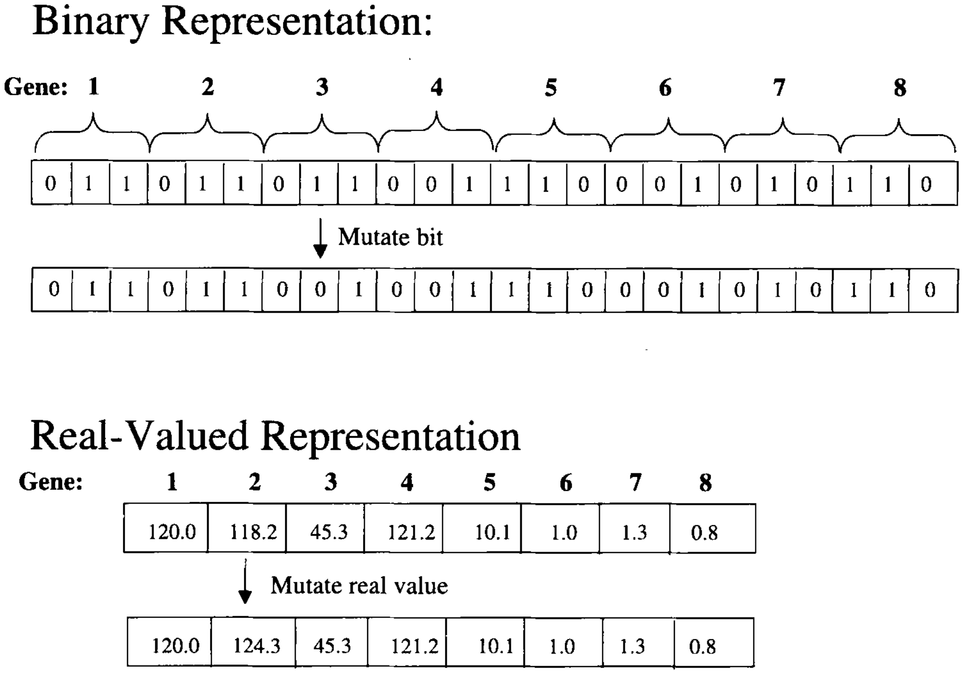
Figure 2. Representation of the conformation of a drug candidate in an enzyme active site using binary and real-valued representations.

An example with relevance to computer-aided drug design is the search for the lowest energy conformation of a drug candidate in the environment of an enzyme active site, the docking problem. The representation must include information describing the conformation of the drug candidate, as well as information about the orientation of the drug candidate relative to the active site. Figure 2 demonstrates in a simplified manner how this information might be encoded in binary and real-valued representations. The conformation of the molecule used in the example requires the representation of only two dihedral angles. In the real-valued representation the actual values of the two dihedral angles are included directly. In the binary representation, however, these values must be converted to a binary form. The example uses a very coarse representation of dihedral angle space, only using three bits to represent the full 360 degree range at 45 degree increments. The orientation of one molecule relative to another can be represented in several ways. The example in Fig. 2 uses three angles to represent the rotation and three distance offsets to represent the translation of the drug candidate relative to a standard orientation.

1.2.2 Evolutionary Operators

1.2.2.1 Mutation

The mutation operator is one of the evolutionary operators that can generate offspring solutions for the next generation. Mutation is the only evolutionary operator in use in EP, it is the primary operator used by ESs, and is a secondary operator in GA applications. The implementation of mutation depends on the representation. Mutation in binary represented problems was originally performed by selecting a bit and setting it randomly to either 0 or 1 (independent of the original value of that bit). It is currently more common to invert the value of bits selected for mutation. Mutation in real-valued representations requires a random change in the value of the real number encoded at a selected position in the genotype. This can be done by selecting one of a limited list of acceptable values, or can be done by adding a small random value centered on zero. Figure 3 demonstrates the effect of the mutation operator on the example introduced in the previous section and shown in Fig. 2.



0	1	1	0	1	1	0	1	1	0	0	1	1	1	0	0	0	1	0	1	0	1	1	0
0	1	1	0	1	1	0	0	1	0	0	1	1	1	0	0	0	1	0	1	0	1	1	0

1	2	3	4	5	6	7	8
120.0	118.2	45.3	121.2	10.1	1.0	1.3	0.8
120.0	124.3	45.3	121.2	10.1	1.0	1.3	0.8

Figure 3. The mutation operator as applied to binary and real-valued solution representations. Binary mutation changes the value of the mutated bit either from 0 to 1 or from 1 to 0. Real-valued mutation changes a value by a random amount in a limited range.

The impact of mutation on a candidate solution can be controlled for real-valued representations. The range of allowed mutations can be modified to suit the problem at hand or dynamically updated during the optimization process. Dynamic control of mutations is assessed in more detail later in this chapter and again in Chapter 12. Binary representations using standard binary coding do not naturally allow this type of control. A solution to this problem is the use of Gray coding. Gray coding is a binary encoding mechanism that encodes adjacent values so that they differ by only one bit. Figure 4 demonstrates an example of both standard encoding and Gray coding. The binary mutation in Fig. 3 changes the gene from 0 1 1 to 0 0 1. If this gene had been coded using the standard binary encoding shown in Fig. 4, the phenotype would change from 135 degrees to 45 degrees. However, if the gene had been coded using the Gray coding system in Fig. 4, the phenotype value would change from 90 to 45 degrees. Thus, Gray coding allows single mutations to the genotype to have a smaller impact on the encoded phenotype of the solutions.

Angle value	Standard binary code value	Gray code value
0	000	000
45	001	001
90	010	011
135	011	010
180	100	110
225	101	111
270	110	101
315	111	100

Figure 4. Three bit codings of angles at 45 degree increments using a standard binary coding system and a Gray coding system.

1.2.2.2 Crossover

The crossover operator is the second evolutionary operator used to generate offspring from selected parents. Crossover is the primary operator in GAs, a secondary operator in ESs, and is not used at all in EP. Crossover combines genetic information from two parents to generate one or two offspring that have features of both parents. Crossover can be implemented in a uniform manner (i.e., uniform crossover), in which each element of the child is selected randomly from one parent or the other. A second type of crossover is to select a point in the gene, and create the child using the information from one parent prior to that point, and data from the second parent after that point. This type of crossover is called one-point crossover. A final type of crossover is to use multiple crossover points. These three types of crossover are shown in Fig. 5. The suitability of crossover to assist in optimization problems relies on the presence of patterns in the genotype which always result in phenotypes having relatively high fitness values. In the previously defined example, such a pattern might be a particular set of distance offsets that move the molecule into a pocket in the receptor. Crossover of a parent having this optimal pattern of distance offsets with another parent can result in a child that retains this beneficial set of distance offsets and has values for other parameters that result in a better overall fitness than either parent.

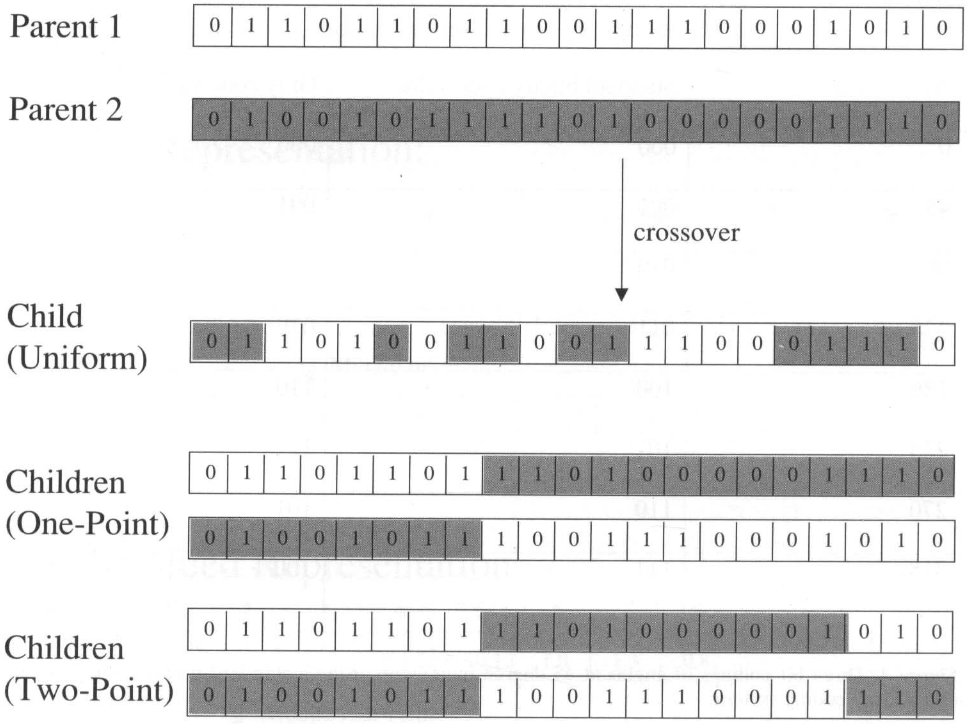


Figure 5. Children resulting from three different crossover mechanisms, uniform crossover, one-point crossover, and two-point crossover. Parts of the gene coming from parent 1 have a white background, those coming from parent 2 have a cross-hatched background.

1.2.3 Selection and the Next Generation

The concepts of biological evolution include a "*survival of the fittest*" mechanism to explain how genetic material is selected to survive into the next generation. The computational implementation of such a mechanism requires a means by which to calculate the fitness of each member of the population, the *fitness function*, as well as an algorithm for selecting members of the population who will pass their genetic material into the next generation.

1.2.3.1 Fitness Functions

Fitness values must reflect the relative quality of the individual solutions for the problem being optimized in order to be useful in directing the evolutionary search toward more promising solutions. Functions for calculating such values must therefore be written specifically for each problem. Some problems naturally suggest their own fitness functions. For example, a conformational search method that seeks the global minimum energy conformation can naturally use the energies of the population members to describe their relative fitnesses. Calculation of fitness values often consumes a majority of the time devoted to solving problems by EAs. Optimization or parallel implementation of fitness functions can therefore provide a significant gain in speed.

There is, of course, no requirement that fitness values be determined computationally. Novel implementations of genetic algorithms in combinatorial chemistry actually used experimentally determined biological assay results as fitness values for population members [14, 15].

1.2.3.2 Selection Algorithms

EAs implement selection mechanisms through algorithms that choose particular parents to be used in generating offspring. GAs are usually implemented with stochastic selection methods. One of these methods is *roulette wheel selection* (also called proportional selection). All members of a population are assigned a segment on a wheel, with each segment having a size in proportion to the fitness of its associated population member. Random positions on the wheel are chosen, and the population member occupying that space becomes a parent. This process is repeated as needed. This selection method can result in premature convergence to a local optimum if there is a candidate solution with a fitness that is greatly superior to that of other population members. This problem can be avoided by using a candidate's rank rather than its actual fitness to determine the size of its space on the wheel. *Tournament selection* relies on competitions between randomly selected population members for selection of parents. Two or more members of the population are compared, and the one having the best fitness value is deemed the winner. The winner of a tournament becomes a parent, and additional tournaments are held to obtain the desired number of parents [16]. Both roulette wheel selection and tournament selection may not select the best member of a population as a parent due to the random nature of the selection process. A common modification is always to include the best solution as a parent, a strategy termed *elitism*.

GAs use two mechanisms to determine the solutions that form the next generation. Two main population models include the *steady-state model* and the *generational model*. A child produced in the steady-state model replaces a single member of the previous generation (often the worst, called the "kill-worst" strategy) and becomes eligible to serve as a parent immediately. In the generational model, all children are produced from parents in a single generation before the next generation is developed. The next generation in a generational model can be selected only from the children or from a combined pool of the children and the members of the previous generation.

ESs use more deterministic selection algorithms. One of these algorithms selects the next generation from the children, called the *(μ, λ) model*. This notation indicates that μ parents are used to generate λ offspring, from which the μ best offspring are selected to form the next generation. An alternate selection algorithm selects the next generation from the union of the parents and children and is termed a *(μ +λ) model*. This notation indicates that μ parents are used to generate λ offspring, and that the next generation is created from the μ best members of μ and λ. This mechanism for developing the next generation is analogous to the generational population model used in some GA applications.

EP uses a probabilistic selection algorithm that has features of both tournament selection and the (μ + λ) model. This selection algorithm competes each member of the combined population of parents and offspring against a set of randomly selected population members. Each individual typically competes with between two and ten other individuals. The population members who are awarded the most wins are used to form the next generation. The formation of the next generation in EP is comparable to the generational model used in GAs.

1.2.4 Self-Adaptation and Learning-Rule Methods

Optimization accomplished by EAs relies on the application of evolutionary operators such as mutation and crossover in order to search the solution space. These evolutionary operators are applied with a particular probability in the development of the new population. This probability and other settings of the operators have an impact on the speed and result of the optimization. ESs were designed to evolve not only the problem solutions, but also the parameters controlling the search strategy [2]. This *co-evolution* or *self-adaptation* process was independently incorporated in EP at a later time [17], and has also made its way into GAs [18].

The implementation of self-adaptation in ESs represents the problem and the standard deviations controlling the search strategy within each member of the population. This results in a search for the optimal solution and the optimal strategy simultaneously. The mutation parameters are modified by a multiplicative function throughout the course of the search. EP varies slightly from this mechanism in that an additive function is used to modify the parameters controlling the search. Comparison of the two methods indicates that the additive mechanism is advantageous when the functions to be optimized are noisy, and that the multiplicative mechanism is more advantageous in other situations [11]. Other work has shown that it is important to first modify the search strategy and then use the modified search parameters to generate the remainder of the gene [19]. This is thought to associate successful strategy parameters with the good problem solutions they generate, rather than having good problem solutions associated with unrelated strategy parameters.

Alternative methods that modify the search strategy in the course of a search are the *learning-rule methods.* These methods track the success of various operator settings for a defined number of generations (or solution evaluations) and then probabilities are reassigned so that more successful operators have higher probabilities [20].

1.3 Implementation Issues and Representative Applications of EAs in Drug Design

The prior sections have outlined the historical background and biological motivation of the three parent types of EAs. This type of description was not meant to imply that all implementations should strictly adhere to any one of these "standards". A recent overview of genetic methods by Luke makes two key points [21]. First, particular choices for the representation, evolutionary operators, selection and fitness function should be adapted to the problem rather than forcing the problem into the mold of a particular EA implementation. The second key point from Luke's overview is that not all problems are suited to solution by EAs. It is also important to note that many reports have now been made on effective combinations of EAs with other search or optimization methods.

1.3.1 Problem-Adapted EA Features

An example of problem-specific genetic operators in computer-aided drug design can be taken from the application of GA methods to the search for the maximum common three-dimensional substructure shared by a pair of molecules [22]. Two knowledge-augmented operators, creep and crunch, were developed to suit this problem. The creep operator adds a pair of atoms to the substructure (a match pair) based on the geometries of the molecules. Atoms from each molecule which have similar distances to two randomly selected matched atom pairs are chosen as an additional match pair. The creep operator has a greater chance to productively add to the size of the matched substructure than the more typical mutation operator, which simply randomly changes a matched pair in this implementation. The crunch operator reduces the number of atom pairs in the substructure match. This operator provides an opportunity to release a suboptimal match that was added without the use of the knowledge-augmented operators. The application of these problem-specific operators was shown to yield substructure matches with lower root mean square (RMS) values than those obtained with the traditional genetic operators alone.

1.3.2 Problem Suitability for EA Implementation

A theoretical comparison by Wolpert and Macready has shown that no algorithm is capable of outperforming any other algorithm over all problems [23]. Any algorithm which clearly outperforms others on a particular problem will suffer in comparison to other algorithms when confronted with a different type of problem. Wolpert and Macready have termed this the "No Free Lunch Theorem". This performance issue has also been demonstrated in an area relevant to computer-aided drug design for the specific problem of selecting maximally dissimilar molecules from a database [24]. The similarity of a molecule to others in the database, in both the problem-specific algorithm and the GA, was measured by comparison of the molecule to the centroid of the database. The GA implementation used the inverse of this similarity measure as the fitness value, and represented sets of molecules as a list of integers, each integer referring to a specific compound in the database. It also included a knowledge-enhanced mutation operator, to replace a molecule in the list with one from the database most dissimilar to the centroid of those already in the list. This mutation operator is equivalent to the selection process used in the problem-specific algorithm for set members after the initial compound was selected. It was determined that the GA could effectively improve the dissimilarity score over random selection, but could not find sets that were as dissimilar as those selected by the problem-specific algorithm even after 100 000 generations.

Two areas in which EA methods have been shown to be highly successful are protein-ligand docking (see Chapter 3) and variable selection for the development of quantitative structure-activity relationships (QSAR, see Chapter 5). A recent comparison of docking methods showed that a GA implementation was more efficient than molecular dynamics, Monte Carlo, or the AutoDock program over five test cases given a relatively small search space (2.5 Å spherical radius) [25]. Of course, consideration of the "No Free Lunch Theorem" [23] should result in a lack of surprise that the GA implementation was not the most efficient when the problem was changed to include a larger search space (11 Å spherical radius). The comparison of docking methods utilized a soft potential during the initial phase of the optimization, an intermediate potential during the intermediate phase, and a hard potential in the final phase. Scaling of the repulsive forces during a docking simulation had previously been shown to allow ligand side chains to better search out suitable pockets without becoming trapped in local minima [26]. The QSAR problem is a challenging one due to the common availability of many more descriptors than data points. EAs have the ability to evolve combinations of these descriptors to develop models without overfitting. Luke recently published a comparison of an EP implementation, a stepwise method, and a comprehensive search method for the generation of a quantitative structure-property relationship (QSPR) model describing high-performance liquid chromatography (HPLC) column capacity factors [27]. He showed that the EP method was able to find five-parameter models that were nearly as good as the best found by the comprehensive method and that the method ran 2000 times more quickly. Allowing the EP method to include six-parameter models enabled it to find the best five-parameter model. The stepwise model generation method was 1000 times faster than the EP implementation, but the best linear model found was worse than all of the top 150 models found by the EP method.

1.3.3 EA Combination Methods

It is also important to note that many reports have now been made on effective combinations of EAs with other search or optimization methods. A very recent example is a two-stage method for the design and evaluation of hydrophobic protein core structures [28]. This two-stage method uses a GA to search conformational space for both side chains and the protein backbone, and subsequently uses Monte Carlo sampling to refine the best results obtained from the GA phase of the search. The Monte Carlo sampling was found to be more efficient in the refinement than continued use of the GA method. A prior example in the field of QSAR modeling used a GA to select sets of molecular descriptors, and subsequently fed those descriptors into a neural network in order to develop the quantitative relationship [29].

Hybrid methods have been applied in many areas, another interesting example being the use of such methods in geometry optimization. A GA implementation was compared with simulated annealing for optimization of small silicon clusters [30]. The GA implementation was found to have favorable convergence early in the search, and then to have difficulty near the optimum. The simulated annealing method showed the reverse of these convergence properties. A combination of GA early in the optimization process and simulated annealing later yielded an order of magnitude improvement in search time over either method alone. More examples of hybrid methods can be found in Chapter 12.

1.4 Conclusions

As subsequent chapters in this book show, EAs have already pervaded all aspects of computer-aided drug design, including geometry optimization, *de novo* design, docking, guidance of combinatorial chemistry, QSAR and the interpretation and analysis of experimental structural information. It is therefore difficult to predict new areas of future EA application that will be fruitful. The application of evolutionary algorithms in drug design, however, is unlikely to taper off in the near future. Clark showed that publications on evolutionary algorithms in computer-aided molecular design have skyrocketed throughout the 1990s [31]. Thus, new implementations of EAs in all of the previously mentioned areas are going to be common in the future.

One EA application area that is just beginning to appear in publications is the use of EAs to assist in the interpretation and analysis of experimental structural information. This is likely to be one area in which EAs will have rising impact in the future. As more research is published on the strengths and weaknesses of EAs, the development of hybrid methods is likely to be a second area to receive increasing attention. Such methods have the capability to utilize the complementary strengths of multiple search and optimization techniques and to more rapidly and thoroughly solve problems of interest to the pharmaceutical industry. The geometry optimization example described in the previous section showed an impressive speed improvement through the use of an evolutionary algorithm in conjunction with simulated annealing. Both the use of combination methods and the number of different methods used in combination with EAs will escalate in the future.

EAs have become a popular method for dealing with challenging search and optimization problems for many reasons. The concept of EAs is relatively straightforward, and the biological foundation is familiar to all. EAs are also straightforward to implement, and once implemented are easily extended to additional problems. While they do suffer in performance comparison against algorithms optimized for particular problems, the flexibility they offer has been a major contributor to their proliferation.

References

[1] L. J. Fogel, A. J. Owens, M. J. Walsh, *Artificial Intelligence Through Simulated Evolution*, Wiley, New York, NY, **1966**.

[2] I. Rechenberg, *Evolutionsstrategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog, Stuttgart, **1973**.

[3] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, **1975**.

[4] C. Darwin, *The Origin of Species*, Dent Gordon, London, **1973**.

[5] T. Bäck, H.-P. Schwefel, An Overview of Evolutionary Algorithms for Parameter Optimization, *Evol. Comput.* **1993**, *1*, 1–23.

[6] K. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Doctoral Dissertation, University of Michigan, MI, **1975**.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, Inc., Reading, MA, **1989**.

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, **1992**.

[9] D. B. Fogel, Applying Evolutionary Programming to Selected Control Problems, *Comput. Math. App.* **1994**, *27*, 89–104.

[10] H.-P. Schwefel, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, *Vol. 26*, Birkhäuser, Basel, **1977**.

[11] T. Bäck, U. Hammel, H.-P. Schwefel, Evolutionary Computation: Comments on the History and the Current State, *IEEE Trans. Evol. Comput.* **1997**, *1*, 3–17.

[12] I. Rechenberg, *Cybernetic Solution Path of an Experimental Problem*, Royal Aircraft Establishment, Farnborough, UK, **1965**.

[13] D. B. Fogel, *Evolutionary Computation: The Fossil Record*, IEEE Press, Piscataway, NJ, **1998**.

[14] L. Weber, S. Wallbaum, C. Broger, K. Gubernator, Optimization of the Biological Activity of Combinatorial Compound Libraries by a Genetic Algorithm, *Angew. Chem. Int. Ed. Engl.* **1995**, *34*, 2280–2282.

[15] J. Singh, M. A. Ator, E. P. Jaeger, M. P. Allen, D. A. Whipple, J. E. Soloweij, S. Chowdhary, A. M. Treasurywala, Application of Genetic Algorithms to Combinatorial Synthesis: A Computational Approach to Lead Identification and Lead Optimization, *J. Am. Chem. Soc.* **1996**, *118*, 1669–1676.

[16] D. E. Goldberg, B. Korb, K. Deb, Messy Genetic Algorithms: Motivation, Analysis and First Results, *Complex Syst.* **1989**, *3*, 493–530.

[17] D. B. Fogel, L. J. Fogel, An Introduction to Evolutionary Programming, in J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, D. Snyers (Eds.) *Artificial Evolution*, Springer-Verlag, Berlin, **1996**, pp. 21–33.

[18] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, NY, **1991**.

[19] D. K. Gehlhaar, D. B. Fogel, Tuning Evolutionary Programming for Conformationally Flexible Molecular Docking, in L. J. Foger, P. J. Angeline, T. Bäck (Eds.), *Evolutionary Programming V*, MIT Press, Cambridge (MA), USA, **1996**, pp. 419–429.

[20] A. Tuson, P. Ross, Adapting Operator Settings in Genetic Algorithms, *Evol. Comput.* **1998**, *6*, 161–184.

[21] B. T. Luke, An Overview of Genetic Methods, in J. Devillers (Ed.): *Genetic Algorithms in Molecular Modeling, Vol. 1*, Academic Press, New York, NY, **1996**, pp. 35–66.

[22] S. Handschuh, M. Wagener, J. Gasteiger, Superposition of Three-Dimensional Chemical Structures Allowing for Conformational Flexibility by a Hybrid Method, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 220–232.

[23] D. H. Wolpert, W. G. Macready, No Free Lunch Theorems for Optimization, *IEEE Trans. Evolut. Comput.* **1997**, *1*, 67–82.

[24] J. D. Holliday, S. S. Ranade, P. Willett, A Fast Algorithm for Selecting Sets of Dissimilar Molecules from Large Chemical Databases, *Quant. Struct.-Act. Relat.* **1995**, *14*, 501–506.

[25] M. Vieth, J. D. Hirst, B. N. Dominy, H. Daigler, C. L. Brooks III, Assessing Search Strategies for Flexible Docking, *J. Comput. Chem.* **1998**, *19*, 1623–1631.

[26] D. K. Gehlhaar, G. M. Verkhivker, P. A. Rejto, C. J. Sherman, D. B. Fogel, L. J. Fogel, S. T. Freer, Molecular Recognition of the Inhibitor AG-1343 by HIV-1 Protease – Conformationally Flexible Docking by Evolutionary Programming, *Chem. Biol.* **1995**, *2*, 317–324.

[27] B. T. Luke, Comparison of Three Different QSAR/QSPR Generation Techniques, *J. Mol. Struct. (THEOCHEM)* **1999**, *468*, 13–20.

[28] J. R. Desjarlais, T. M. Handel, Side-chain and Backbone Flexibility in Protein Core Design, *J. Mol. Biol.* **1999**, *290*, 305–318.

[29] S.-S. So, M. Karplus, Evolutionary Optimization in Quantitative Structure–Activity Relationship: An Application of Genetic Neural Networks, *J. Med. Chem.* **1996**, *39*, 1521–1530.

[30] C. R. Zacharias, M. R. Lemes, A. D. J. Pino, Combining Genetic Algorithm and Simulated Annealing: A Molecular Geometry Optimization Study, *J. Mol. Struct. (THEOCHEM)* **1998**, *430*, 29–39.

[31] D. E. Clark, Evolutionary Algorithms in Computer-Aided Molecular Design: A Review of Current Applications and a Look to the Future, in A. L. Parrill, M. R. Reddy (Eds.): *Rational Drug Design: Novel Methodology and Practical Applications, Vol. 719*, American Chemical Society **1999**, pp. 255–270.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark
Copyright© WILEY-VCH Verlag GmbH, 2000

2 Small-molecule Geometry Optimization and Conformational Search

Ron Wehrens

Abbreviations

EA	Evolutionary algorithm
ES	Evolution strategy
GA	Genetic algorithm
NMR	Nuclear magnetic resonance
NOE(SY)	Nuclear Overhauser effect (spectroscopy)
RMS	Root mean square
SA	Simulated annealing

2.1 Introduction

The optimization of molecular geometry was one of the very first applications of *evolutionary algorithms* (EAs) in chemistry. Once the possibilities of this class of optimization methods were realized, they quickly became very popular in a wide range of chemical application areas (several reviews have included applications of EAs in the optimization of chemical structures; see, e.g., [3–8]. Several reasons can be identified for this popularity. First of all, evolutionary methods have shown good results in problems where other methods have struggled. Second, their principles are easily understood and intuitively appealing. Furthermore, implementation is generally easy, given that several toolboxes are available in the public domain. In the extreme case, one only has to provide an evaluation function (see below) that assesses the quality of a trial solution. In structure optimization, such a function is often provided by molecular mechanics software, but other criteria may be used as well.

Several different forms of EAs are used in practice, but the majority of applications concern *genetic algorithms* (GAs) [9, 10]. These are characterized by a population of trial solutions, each represented in a binary way, a selection operator that emphasizes the best individuals in a population and crossover and mutation operators to generate the next generation. The German school of *evolution strategies* (ESs) [11] differs from GAs in that crossover is not applied, a real-valued representation rather than a binary one is used, and several search parameters such as the mutation rate are encoded in the trial solutions. In this way, the algorithm takes care of its own configuration. Boundaries between the two classes, however, are blurring; GAs often apply real-coded representations and many different, often problem-dependent, evolutionary operators, whereas ESs sometimes apply crossover-like operators or omit the inclusion of search parameters in the strings. To the knowledge of the author, no applications in molecular structure optimization are known of other forms of EA, such as *evolutionary programming* [12], *genetic programming* [13] or *classifier systems* [14], and these will not be treated further in this chapter.

For an algorithm to qualify as an EA, two criteria must be satisfied: (i) a method should manipulate a population of trial solutions; and (ii) some form of selection should be used. In practice, a third element is almost always present. This is that new trial solutions are created by combination of old solutions (the crossover operator, used in many different forms).

This chapter will not treat technical matters in great depth; it is assumed that readers are familiar with the basic concepts of EAs (see Chapter 1). Rather, attention will be focused on the differences between EAs and other structure optimization methods, and strengths and weaknesses of each are identified. In the next section, some fundamental qualities of EAs will be discussed that are important in their application, especially in the field of structure optimization. Next, a short literature review will be given of the application of EAs in conformational search and geometry optimization, concentrating on the evaluation functions used, the representation of molecular structure and the types of molecule under study. Separate sections will be devoted to those applications where comparisons are made between EAs and other optimization methods.

2.2 Evolutionary Algorithms

As stated in the introduction, one of the characteristic features of EAs is that a population of trial solutions is maintained. This is a marked difference from other individual-based optimization methods such as simulated annealing (SA) [15], and has several important consequences. In many cases, population-based search behavior will be markedly different from that of individual-based methods and, as such, the two classes complement each other. Although to date no single optimization method has consistently outperformed all other methods for a range of optimization problems (the "No Free Lunch" theorem [16]), it is possible that for a very specific application there is a clear preference for either the population- or individual-based search. In this section, we highlight those aspects of population-based search that are fundamentally different from individual-based search and that therefore may determine which of these is most suitable for a specific application.

2.2.1 Diversity

Use of a population makes it possible to encourage a better coverage of the search space. Imagine a perfect single-solution optimization method that always finds the globally optimal molecular conformation. In Fig. 1, this would mean that solution A would be found every time. However, the structures corresponding to some of the local optima B-E may also be populated at room temperature, especially if the energy wells are broad. With a single-solution method, only those conformations encountered during the journey from the starting point to the global optimum will be found. A way to find more would be to restart the search from another starting position, but there is no guarantee that another route will be followed. This kind of search behavior is therefore also called "linear".

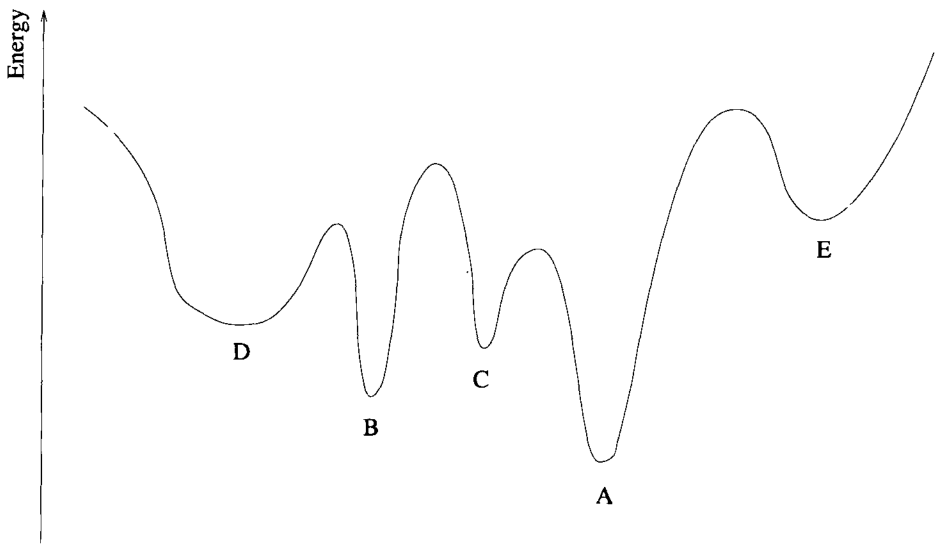


Figure 1. A one-dimensional energy surface. Not only the global optimum A may be important, but B-E may also be populated at room temperature.

With population-based search methods, there are ways around this problem. The ability to determine the similarity between several trial solutions opens up possible ways to promote diversity. Special operators such as forced mutations can be employed to ensure that the population does not converge completely to one single optimum. An even more rigorous approach is to maintain not one population but several subpopulations, which are allowed to evolve individually but for an occasional exchange of genetic material. Reports in the literature indicate that this approach not only leads to several good solutions (B-E in Fig. 1) but also may speed up the route to the global optimum.

2.2.2 Creation of New Solutions

With individual-based search methods, one trial solution is the starting point for the generation of a new solution. The new solution may be generated by a random perturbation of the old solution, or by application of a modifying algorithm. The new solution may or may not replace the old solution as a starting point for the next iteration. This type of search behavior is illustrated in the left plot in Fig. 2; it follows a linear trajectory through search space. The consequence of such a strategy is that two consecutive solutions are probably very much alike, whereas two solutions that are far apart in the solution sequence are less likely to be similar. Unless special attention is paid to this problem, this behavior may make it difficult to overcome large energy barriers (in the case of energy minimization of a molecular structure).

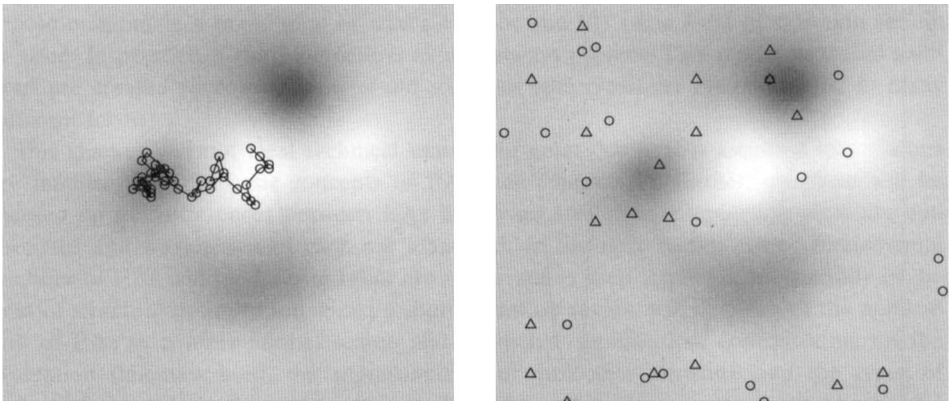


Figure 2. Linear versus non-linear search behavior. The left plot shows a trajectory through search space of an individual-based method; the right plot shows four generations of a GA application in the same search space (in both plots, 40 evaluations were performed). Circles indicate members of the zeroth (random) generation; triangles indicate members of the first generation. The darker the colour in the plot, the more negative the energy.

In the case of EAs, the generation of new structures, especially in the case where crossover is used, does not suffer from this problem. A trial solution in an offspring population may differ quite substantially from both its parents, as indicated in the right plot in Fig. 2. This enables the algorithm to take large steps in the solution space, and the linear behavior of the individual-based search is not observed. The other side of the coin is that this makes it difficult to obtain a precise estimate of the location of the optimum. This is also clearly visible in Fig. 2: whereas the linear method goes straight to one of the local optima, the GA seems to perform an almost random search. This example is of course very simplistic and EAs are much more suited for searching in high-dimensional spaces. To remedy the poor search precision, the final solution(s) of EAs are often further optimized by local optimization methods.

2.2.3 Constraint Satisfaction

In some optimization problems, including structure optimization, the overall quality of a trial solution may be quite good, but because some constraint is violated, the solution is not a valid one. EAs offer several flexible ways of dealing with this situation. One is to modify the solution so that the violation is removed. This so-called "repair" strategy is also applicable in individual-based optimization methods, but has the disadvantage that it is slow, because of the large number of calculations needed to identify and remove the violation. In some cases, the optimization is also forced in the wrong direction because of a repair strategy. An example is depicted in Fig. 3, where a gray area in a Ramachandran-like plot indicates a "no-go" area, and the optimum is located at the other side. Individual-based methods employing repair strategies will not be able to cross the gray area and are forced to go around it.

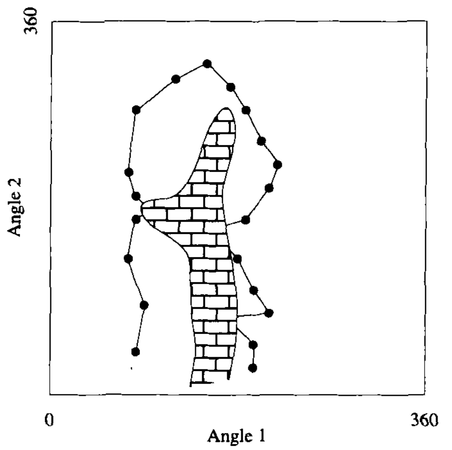


Figure 3. Hypothetical Ramachandran-type plot, indicating forbidden combinations of two torsion angles. Individual-based methods are forced to go around the "no-go" area, whereas population-based methods can more or less ignore it. In the figure, three repair operations are necessary.

Population-based methods employing the repair strategy are also described in the literature, but there are other strategies available. Because of the non-linear search characteristics mentioned in the previous paragraph, there is no need to go around the forbidden area. Solutions violating the constraints can be treated just like other solutions that are bad, for instance by penalizing them or removing them from the population altogether. An advantage of the penalization strategy is that the "genes" of such a solution are still available for reproduction (although with a smaller probability), so that good elements of this solution may still propagate to the next generation. Both penalization and lethalization strategies are much faster than the repair method, and have been shown to give good results [17].

2.3 Technical Aspects of Method Comparisons

The comparison of several global optimization problems for a given application seems to be a simple task. One performs several optimizations with all the methods and finally picks the one that gives the best result. In the context of structure optimization, this may correspond to the structure with the lowest energy, as calculated by a force field method. However, there are several difficulties with this approach. First of all, one should be sure that each optimization method is properly tuned. Because of the large number of adjustable parameters in EA optimization (and in many other optimization methods as well), this is not a trivial task. Often, a one-at-a-time approach is used, starting from standard settings, as published in the scientific literature. There is a very real danger that this will lead to suboptimally configured methods, and so more thorough methods utilizing experimental design theory have been proposed [18, 19]. A full optimization of the search settings may be costly in terms of computing power, and is therefore not often performed. In comparisons in the literature, one should always be aware of the fact that one of the methods employed may have received more attention (e.g., is a newly developed, and hence optimized, method), while others are applied without much tinkering. Comparisons with published results are only possible if exactly the same evaluation function is used in all methods.

Another point is that, especially with stochastic optimization methods such as SA and EAs, results of repeated runs may lead to quite different answers. This variability in the results must be taken into account when comparing optimization methods. One way to do this is to report several values for each optimization (consisting of a number of repeated runs), such as the best solution found, and the quality of the mean and worst final solutions.

These considerations are also important when fine-tuning a search algorithm. With EAs, this can be a difficult task, given the large freedom of the experimenter in choosing operators and search settings. As already stated, most measures of performance focus on the quality of the best solution found by the search method, or a mean value over a number of good solutions. Clearly, this is a one-sided view of the quality of an optimization method, and additional criteria have been suggested. These include measures of diversity during the optimization [20–22], the percentage of cases in which the global minimum was identified [23], the number of evaluations or the CPU time needed to find the optimum [18, 24–26] and the number of different structures found [21]. Exactly what is expected of the optimization method should be reflected in the criteria used to evaluate performance. In the context of structure optimization, a set of four quality criteria has been proposed recently [27]. These concentrate on coverage of the search space, coverage of the solution space and the reproducibility of these quantities in repeated runs. The application of these criteria in the fine-tuning of GA settings has also been described [19]. Because the criteria do not explicitly use values from the evaluation function, it is possible to fine-tune the evaluation function itself. This is important in cases where a weighted sum of several terms is used in evaluating trial solutions (see, e.g., [18, 28]).

2.4 Traditional Methods for Structure Optimization

Traditional methods of structure optimization [29] fall into one of two categories, depending on the data that are available. If experimental distance constraints are available, for instance from NMR spectroscopy, distance geometry methods [30, 31] can be used. The problem is then to convert these incomplete distance constraints to a complete definition of a molecular conformation in Cartesian co-ordinates. The distance constraints are formulated as pairs of upper and lower bounds; a constraint is said to be satisfied if the corresponding distance in a molecular structure is between these bounds.

The most common algorithm for distance geometry uses an eigenvalue decomposition to transform a distance matrix into Cartesian space, and a subsequent optimization step is performed to obtain the final co-ordinates. Often, conjugate gradient minimization [32] is used for this. The distance matrix that is transformed is a random sample of distances between the upper and lower bounds. After this so-called embedding step, a smoothing step may be performed (metrization) after each new distance is picked, to ensure that the triangle inequality holds. Alternatively, an exhaustive search method can be used to manipulate a structure in torsion angle space so that distance constraints are satisfied.

The second class of traditional structure optimization methods aims at finding molecular conformations with a minimal energy. Force fields [33, 34] are used to calculate the energy of a molecular conformation, and an optimization algorithm directs the search into a minimum. Several optimization methods are available: numerical methods [32], Monte Carlo methods and simulated annealing [15], random search and systematic search. To prevent the optimization method from becoming trapped in a local minimum, multiple runs are often performed.

Besides these general optimization methods, some chemistry-specific methods can be used to sample conformational space. In the popular molecular dynamics method [35], each atom in a trial structure is given an initial velocity, and Newtonian equations are used to calculate the structure at a specific instant in time. Although this is an appealing method conceptually, it has several disadvantages: it requires significant computer resources, and is easily trapped in a local optimum. Therefore, its primary use is to study the behavior of conformations in the vicinity of an optimum.

Other examples of structure optimization methods are directed tweak [36] and direct search methods [37]. The latter is an intelligent adaptive grid search that may be very fast because no derivatives need to be calculated. A set of trial conformations is evaluated and the worst of these is replaced by a new conformation. The replacement is done by a series of geometric operations in multi-dimensional optimization space. The method is closely related to the simplex optimization method of Nelder and Mead [38].

2.5 Evolutionary Methods for Structure Optimization

EAs (and in particular GAs) nowadays are very popular methods for finding the optimal geometry of a molecular structure with respect either to agreement with experimental data, such as NMR-derived distance constraints, or internal energy. These two types of problem form the bulk of all EA applications in this area, and they will be treated in more detail in the next paragraphs. Several other applications also deserve a mention. Structure optimization with respect to complementarity to an active site (e.g., [39, 40]) belongs to the domain of molecular docking and will be dealt with in Chapter 3. Another problem is tackled in [41], where a small set of diverse conformers is sought. The fitness of a solution is given by a measure of the difference with all other solutions in the population. The algorithm is initiated from a 3-D structure, and a population is generated by applying random mutations to this structure.

The optimization of clusters of atoms and molecules with evolutionary algorithms is also an active area of research (see, e.g., [23, 42, 43]). In [44], a nested GA is described that can calculate cluster geometries of flexible molecules, where the inner GA loop is used to optimize the structure of each molecule separately. These applications will not be treated further in this chapter, nor will structure optimization as used in the alignment of flexible molecules [45, 46]. These problems are very much related to the applications described in this chapter, however, since only the evaluation function is different. In almost all cases, torsion angles (internal co-ordinates) are used as the molecular representation, assuming standard values for bond angles and bond lengths. This has the advantage that only a few parameters are needed to describe relatively complex structures. However, one important disadvantage of this representation is the so-called "leverage" effect: a small change in one torsion angle may lead to a drastic change in the overall conformation of the molecule. In general, such a representation is found to have difficulties. Note that this effect is independent of whether real or binary coding is used.

2.5.1 Satisfying Constraints from Experiments

One of the first applications of EAs in structure optimization was described by Lucasius et al. [2, 47]. They optimized the structure of a DNA dinucleotide using a standard GA. The method requires an experimental NOESY NMR spectrum, and the chemical shift information for each proton in the structure. Translation of the dihedral angles to a 3-D structure then makes it possible to calculate a theoretical NOE spectrum, which can be compared to the experimental one. The evaluation function in this case is a root mean square (RMS) difference between the intensities of crosspeaks in the theoretical and experimental spectra.

A different approach is taken in [48], where distance constraints derived from NMR spectroscopy are used in the evaluation function. A GA, called DGΩ, was used as a front-end for the distance geometry program DGII [49]. The upper and lower bounds for a set of 58 distance restraints were coded in the strings. Each member of the population was the starting point for a distance geometry calculation, and the fitness of this member was given by the number of distance constraint violations. It was shown that for cyclosporin A, a cyclic undecapeptide notorious for its difficult sampling properties, the GA led to a significant improvement in the sampling behavior of the distance geometry algorithm.

In other applications, the agreement with experimentally obtained distance constraints is used as a fitness function for oligopeptides [50], RNA stem-loop structures [28] and DNA oligonucleotides [18, 47, 51]. Pearlman [52] uses a genetic algorithm to derive an ensemble of structures that best fits NMR data. The resulting weights indicate which conformers are important. In many of these applications, fitness penalties are given for structures with van der Waals overlap between atoms. Other extensions include the use of stereochemical constraints or constraints on the conformation of a substructure. The advantage of such an evaluation function is that it can be calculated quite quickly, much faster than a complete distance geometry optimization or the calculation of a theoretical NMR spectrum.

2.5.1.1 Comparisons with Other Methods

Several comparisons have been made between distance geometry programs and GA approaches. In [53], a GA for the optimization of torsion angles is compared to the DGII distance geometry package [49] for a modified thymine dimer. It was found that DGII was more successful in satisfying the distance constraints in the data. However, this came at the cost of a much larger variability in bond angles, which in some structures assumed rather extreme values. The GA identified a much more tight set of structures because only torsion angles could be modified during the search.

This comparison was extended to include the DGΩ program described in [48], and more or less similar results were obtained [54]. Again, the classical GA yielded the poorest sampling of the conformation space, albeit that all generated structures had a good covalent geometry. DGΩ performed slightly better than DGII, but required much more computing time.

2.5.2 Energy Minimization

An even more popular use of EAs in the optimization of molecular conformation is in the energy minimization of a molecular structure. Here, an appropriate force field calculation is used as the evaluation function. In most cases, only the most important energy terms are taken into account to speed up calculations. Many examples can be found of the structural optimization of small organic molecules [20–22, 24, 55–58], small peptides and peptide analogs [59–62], and proteins [1, 25, 63–65].

Again, the most popular way to represent molecular structure is to use internal coordinates (torsion angles). Tufféry et al. [1, 25] describe an evolutionary algorithm that optimizes the structure of protein side chains by selecting rotamers from a predefined set, where the chance of selecting a specific rotamer is determined by its probability of occurrence, determined beforehand. In most cases, standard genetic operators are used, but the application of specialized operators sometimes leads to improvements. In [56], a graph crossover, where two subgraphs are combined (and repaired if necessary), is used, resulting in an increased robustness of the algorithm. Jin et al. compare three variants of their GAP program, differing in ways to enforce diversity during the optimization [22]. It was concluded that the different crossover operators did not influence the sampling characteristics very much, probably because the crossover operator is only of minor importance at the end of the search, when the diversity in the population has decreased.

Parallellization using island or migration models is reported to improve results, mainly by making it easier for the algorithm to maintain a diverse population [51, 56, 62, 63]. Niching, where the population is divided into subsets of more or less similar individuals, is also used for this purpose. A popular niching technique is sharing, where the fitness of individuals in the same niche is lowered when the niche is overpopulated. Measuring the degree of similarity in the population (also called convergence) is not always straightforward. Torsion angles of –179 and 179 degrees are of course quite close, so simply looking at differences or variances may lead to wrong conclusions. In [21], a simple criterion is defined for the measurement of similarity in a population defined by strings of torsion angles. Each angle is represented by a point on the unit circle, and if the angles are evenly distributed, the mean of all angles will lie at the origin. A larger distance from the mean to the origin indicates a smaller diversity. It should be noted, however, that even a small deviation in torsion angles can lead to large differences in overall 3-D structure, so that the measure is useful for signalling convergence in the population. Cluster analysis is sometimes used to determine the presence of similar conformations [1, 27, 57, 60].

Although in almost all applications the final solutions of an EA are further optimized by a local optimization method (typically steepest descent), it is also possible to perform a local optimization in each evaluation during the EA run [1, 24]. Although in most cases only a few steps are performed, run times will increase significantly, and for small molecules it usually is unnecessary.

2.5.2.1 Comparison with Other Methods

Many papers compare the performances of EAs with other search methods such as simulated annealing, direct search methods and random search (see, e.g., [26, 37, 66, 67]). The general conclusion is that EAs are consistently among the best performing general search algorithms, in many cases performing as well as or only slightly worse than optimization methods specifically designed for the problem at hand. As noted by several other authors, Judson states that GAs are particularly useful for quickly producing a family of low-energy conformers, but are less successful in fine-tuning these towards the exact global optimum [66]. The post-EA steepest descent optimization that is performed in most applications is the result of this realization. Because of the limited accuracy of energy calculations this disadvantage is not too important.

In the context of 3-D database searching, Clark et al. found that GAs were better than distance geometry, systematic search and random search, and both GAs and directed tweak methods performed well enough to be useful in practical applications [67], a conclusion that is also reached in [26].

2.6 Discussion

The many successful applications of EAs, and in particular GAs, have made EAs the *de facto* reference method in the optimization of small- to medium-sized molecules. With larger structures, typically containing hundreds of torsion angles, results have been less encouraging, and it may be necessary to apply EAs to smaller substructures separately before analyzing the overall geometry (see Chapter 11 for more details). Apart from the good results, in that usually very low energies are obtained or many of the experimental constraints are satisfied, EAs have the significant advantage that they provide a family of solutions. In some cases, it is shown that they are not always the most efficient search method in terms of number of evaluations, but with the dramatic increase in computing power of the least few years (and there is little reason to believe that this development is slowing down), efficiency may no longer be the most important aspect. Attention will probably shift to measures of completeness (are all relevant minima identified?), reliability and reproducibility.

As already stated, application of EAs is relatively straightforward, once an evaluation function is available. The most straightforward of these is a force field program, but more often custom-built evaluation functions are written. In most cases, the value that is returned is only required to give a relative quality measure of the trial solution so, for instance, large parts of expensive energy calculations can be omitted. Configuration of EAs may be a problem, and sometimes large performance drops can be observed because of an inadequate choice of operators or search settings. Although standard settings seem to be used widely, small differences in implementation may require expensive meta-optimization. Table 1 summarizes some of the applications mentioned in the text.

Table 1. Summary of applications mentioned in the text.

Compound class	Evaluation	Representation	Reference
Organic molecules	Energy	T	[20, 21, 24, 55, 56, 58]
Proteins/peptides	Energy	P	[1]
	Energy	T	[22, 57, 60, 62, 63, 65]
	Constraints	T	[50, 59, 61]
	Constraints	O	[48]
Nucleic Acids	Constraints	T	[2, 28, 47, 51]
	Constraints	O	[18]

Representation: T: torsion angles, P: predefined set of partial conformations, O: other.

The most critical choice for the success of an EA, however, is probably the representation. The most widely used representation, and also the most efficient one, consists of a series of torsion angles (see also Table 1). This leads to standard values for bond lengths and bond angles; usually, these are optimized in the subsequent local optimization. The biggest disadvantage of a representation by torsion angles is that a small change in one angle may lead to a large change in the value of the evaluation function. This leads to an EA "search landscape" with very sharp peaks, which in test functions is observed to hamper the performance of the search method significantly. On the other hand, the deviation in one torsion angle may be compensated for by another angle, so that two different strings may code for almost the same 3-D structure. At the moment, however, in most applications there is no realistic alternative.

The flexible nature of EAs and many other global search techniques has led to many hybrid methods, where elements of methods from different classes are combined. Examples in other fields include combinations of evolutionary methods with local optimization methods, simulated annealing and tabu search [68] (see Chapter 12 for a further discussion). Two examples of methods having some characteristics of EAs can be found in [69, 70]. In [70], deep local minima are identified by combining torsion angles from a small set of other local minima. In [69], what is essentially a tree-search is performed, where the descendants of each node are generated by crossover-like mechanisms. Again, torsion angles are taken from a pool of previously found local minima.

2.7 Conclusions

Evolutionary algorithms of many flavors have found wide application in the conformational search of small- to medium-sized molecules. Their success has been remarkable, especially since they are general problem solvers, not specifically designed for structure optimization problems. Moreover, the basic algorithm is very simple. Although there are several issues to be addressed, most notably the representation of molecular structure, EAs are likely to continue to play a major role in the future.

References

[1] P. Tufféry, C. Etchebest, S. Hazout, R. Lavery, A New Approach to the Rapid Determination of Protein Side Chain Conformations, *J. Biomol. Struct. Dyn.* **1991**, *8*, 1267–1289.

[2] C. B. Lucasius, M. J. J. Blommers, L. M. C. Buydens, G. Kateman, A Genetic Algorithm for Conformational Analysis of DNA, In L. Davis, (Ed.), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, **1991**, pp. 251–281.

[3] J. Devillers (Ed.), *Genetic Algorithms in Molecular Modeling*, Academic Press, New York, **1996**.

[4] P. Willett, Genetic Algorithms in Molecular Recognition and Design, *Trends Biotech.* **1995**, *13*, 516–521.

[5] G. Bohm, New Approaches in Molecular Structure Prediction, *Biophys. Chem.* **1996**, *59*, 1–32.

[6] D. E. Clark, D. R. Westhead, Evolutionary Algorithms in Computer-Aided Molecular Design, *J. Comput.-Aided Mol. Des.* **1996**, *10*, 337–358.

[7] R. S. Judson, Genetic Algorithms and their Use in Chemistry, In K. B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 10*, VCH, New York, **1997**, pp. 1–73.

[8] G. Jones, Genetic and Evolutionary Algorithms, In P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, P. R. Schreiner (Eds.), *Encyclopedia of Computational Chemistry*, John Wiley & Sons, Chichester, UK, **1998**, Volume 2, pp. 1127–1136.

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, **1992**.

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, **1989**.

[11] H. P. Schwefel, *Numerical Optimization of Computer Models*, Wiley, Chichester, UK, **1981**.

[12] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ, **1995**.

[13] J. R. Koza, *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, **1992**.

[14] L. B. Booker, D. E. Goldberg, J. H. Holland, Classifier Systems and Genetic Algorithms, *Artif. Intell.* **1989**, *40*, 235–282.

[15] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by Simulated Annealing, *Science 1983*, *220*, 671–680.

[16] D. H. Wolpert, W. G. Macready, No Free Lunch Theorems for Optimization, *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82.

[17] A. H. C. van Kampen, C. S. Strom, L. M. C. Buydens, Lethalization, Penalty and Repair Functions for Constraint Handling in Genetic Algorithm Methodology, *Chemom. Intell. Lab. Syst.* **1996**, *34*, 55–68.

[18] A. H. C. van Kampen, L. M. C. Buydens, The Ineffectiveness of Recombination in a Genetic Algorithm for the Structure Elucidation of a Heptapeptide in Torsion Angle Space. A Comparison to Simulated Annealing, *Chemom. Intell. Lab. Syst.* **1997**, *36*, 141–152.

[19] R. Wehrens, E. Pretsch, L. M. C. Buydens, The Quality of Optimization by Genetic Algorithms, *Anal. Chim. Acta* **1999**, *388*, 265–271.

[20] D. B. McGarrah, R. S. Judson, Analysis of the Genetic Algorithm Method of Molecular Conformation Determination, *J. Comput. Chem.* **1993**, *14*, 1385–1395.

[21] N. Nair, J. M. Goodman, Genetic Algorithms in Conformational Analysis, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 317–320.

[22] A. Y. Jin, F. Y. Leung, D. F. Weaver, Three Variations of Genetic Algorithm for Searching Biomolecular Conformation Space: Comparison of GAP 1.0, 2.0, and 3.0, *J. Comput. Chem.* **1999**, *20*, 1329–1342.

[23] J. A. Niesse, H. R. Mayne, Global Optimization of Atomic and Molecular Clusters using the Space-Fixed Modified Genetic Algorithm Method, *J. Comput. Chem.* **1997**, *18*, 1233–1244.

[24] R. S. Judson, E. P. Jaeger, A. M. Treasurywala, M. L. Peterson, Conformational Searching Methods for Small Molecules. II. Genetic Algorithm Approach, *J. Comput. Chem.* **1993**, *14*, 1407–1414.

[25] P. Tufféry, C. Etchebest, S. Hazout, R. Lavery, A Critical Comparison of Search Algorithms Applied to the Optimization of Protein Side-Chain Conformations, *J. Comput. Chem.* **1993**, *14*, 790–798.

[26] J. C. Meza, R. S. Judson, T. R. Faulkner, A. M. Treasurywala, A Comparison of a Direct Search Method and a Genetic Algorithm for Conformational Searching, *J. Comput. Chem.* **1996**, *17*, 1142–1151.

[27] R. Wehrens, E. Pretsch, L. M. C. Buydens, Quality Criteria of Genetic Algorithms for Structure Optimization, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 151–157.

[28] H. Ogata, Y. Akiyama, M. Kanehisa, A Genetic Algorithm-based Molecular Modeling Technique for RNA Stem-Loop Structures, *Nucleic Acids Res.* **1995**, *23*, 419–426.

[29] A. R. Leach, A Survey of Methods for Searching the Conformational Space of Small and Medium-Sized Molecules, In K. B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 2*, VCH, New York, **1991**, pp. 1–55.

[30] G. M. Crippen, *Distance Geometry and Conformational Calculations*, Research Studies Press, Chichester, UK, **1981**.

[31] J. M. Blaney, J. S. Dixon, Distance Geometry in Molecular Modeling, In K. B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 5*, VCH, New York, **1994**, pp. 299–335.

[32] T. Schlick, Geometry Optimization, In P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, P. R. Schreiner (Eds.), *Encyclopedia of Computational Chemistry*, John Wiley and Sons, Chichester, UK, **1998**, pp. 1136–1157.

[33] J. P. Bowen, N. L. Allinger, Molecular Mechanics: the Art and Science of Parameterization, In K. B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 2*, VCH, New York, **1991**, pp. 81–97.

[34] I. Pettersson, T. Liljefors, Molecular Mechanics Calculated Conformational Energies of Organic Molecules: a Comparison of Force Fields, In K. B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 9*, VCH, New York, **1996**, pp. 167–189.

[35] W. F. van Gunsteren, H. J. C. Berendsen, Computer Simulation of Molecular Dynamics: Methodology, Applications, and Perpectives in Chemistry, *Angew. Chem., Int. Ed. Engl.* **1990**, *29*, 992–1023.

[36] T. Hurst, Flexible 3D Searching: the Directed Tweak Technique, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 190–196.

[37] J. C. Meza, M. L. Martinez, Direct Search Methods for the Molecular Conformation Problem, *J. Comput. Chem.* **1994**, *15*, 627–632.

[38] J. A. Nelder, R. Mead, A Simplex Method for Function Optimization, *Computer J.* **1965**, *7*, 308–313.

[39] R. S. Judson, Y. T. Tan, E. Mori, C. Melius, E. P. Jaeger, A. M. Treasurywala, A. Mathiowetz, Docking Flexible Molecules: a Case Study of Three Proteins, *J. Comput. Chem.* **1995**, *16*, 1405–1419.

[40] C. M. Oshiro, I. D. Kuntz, J. S. Dixon, Flexible Ligand Docking using a Genetic Algorithm, *J. Comput.-Aided Mol. Des.* **1995**, *9*, 113–130.

[41] O. Mekenyan, D. Dimitrov, N. Nikolova, S. Karabunarliev, Conformational Coverage by a Genetic Algorithm, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 997–1016.

[42] D. M. Deaven, N. Tit, J. R. Morris, K. M. Ho, Structural Optimization of Lennard-Jones Clusters by a Genetic Algorithm, *Chem. Phys. Lett.* **1996**, *256*, 195–200.

[43] W. J. Pullan, Structure Prediction of Benzene Clusters using a Genetic Algorithm, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 1189–1193.

[44] J. Mestres, G. E. Scuseria, Genetic Algorithms: a Robust Scheme for Geometry Optimizations and Global Minimum Structure Problems, *J. Comput. Chem.* **1995**, *16*, 729–742.

[45] G. Jones, P. Willett, R. C. Glen, A Genetic Algorithm for Flexible Molecular Overlay and Pharmacophore Elucidation, *J. Comput.-Aided Mol. Des.* **1995**, *9*, 532–549.

[46] D. A. Thorner, D. J. Wild, P. Willett, P. M. Wright, Similarity Searching in Files of Three-Dimensional Chemical Structures: Flexible Field-Based Searching of Molecular Electrostatic Potentials, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 900–908.

[47] M. J. J. Blommers, C. B. Lucasius, G. Kateman, R. Kaptein, Conformational Analysis of a Dinucleotide Photodimer with the Aid of the Genetic Algorithm, *Biopolymers* **1992**, *32*, 45–52.

[48] A. H. C. van Kampen, L. M. C. Buydens, C. B. Lucasius, M. J. J. Blommers, Optimization of Metric Matrix Embedding by Genetic Algorithms, *J. Biomol. NMR* **1996**, *7*, 214–224.

[49] T. F. Havel, An Evaluation of Computational Strategies for Use in the Computational Determination of Protein Structure from Distance Constraints Obtained by NMR, *Prog. Biophys. Mol. Biol.* **1991**, *56*, 43–78.

[50] P. N. Sanderson, R. C. Glen, A. W. R. Payne, B. D. Hudson, C. Heide, G. E. Tranter, P. M. Doyle, C. J. Harris, Characterization of the Solution Conformation of a Cyclic RGD Peptide Analogue by NMR Spectroscopy Allied with a Genetic Algorithm Approach and Constrained Molecular Dynamics, *Int. J. Peptide Protein Res.* **1994**, *43*, 588–596.

[51] M. L. M. Beckers, E. P. P. A. Derks, W. J. Melssen, L. M. C. Buydens, Parallel Processing of Chemical Information in a Local Area Network. Part III. Using Genetic Algorithms for Conformational Analysis of Biomacromolecules, *Comput. Chem.* **1996**, *20*, 449–457.

[52] D. A. Pearlman, FINGAR: a New Genetic Algorithm-Based Method for Fitting NMR Data, *J. Biomol. NMR* **1996**, *8*, 49–66.

[53] M. L. M. Beckers, L. Buydens, J. Pikkemaat, C. Altona, Application of a Genetic Algorithm in the Conformational Analysis of Methylene-acetal-linked Thymine Dimers in DNA: Comparison with Distance Geometry Calculations, *J. Biomol. NMR* **1997**, *9*, 25–34.

[54] A. H. C. van Kampen, M. L. M. Beckers, L. M. C. Buydens, A Comparative Study of the DG-OMEGA, DGII and Genetic Algorithm Torsion Angle Optimization Methods for the Structure Elucidation of a Methylene-acetal Linked Thymine Dinucleotide, *Comput. Chem.* **1997**, *21*, 281–297.

[55] T. Brodmeier, E. Pretsch, Application of Genetic Algorithms in Molecular Modelling, *J. Comput. Chem.* **1994**, *15*, 588–595.

[56] S. Beiersdörfer, J. Schmitt, M. Sauer, A. Schulz, S. Siebert, J. Hesser, R. Männer, J. Wolfrum, Finding the Conformation of Organic Molecules with Genetic Algorithms, In H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel (Eds.), *Lect. Notes in Comp. Sci. 1141*, Springer, Heidelberg, **1996**, pp. 972–981.

[57] A. Y. Jin, F. Y. Leung, D. F. Weaver, Development of a Novel Genetic Algorithm Search Method (GAP1.0) for Exploring Peptide Conformational Space, *J. Comput. Chem.* **1997**, *18*, 1971–1984.

[58] M. Keser, S. I. Stupp, A Genetic Algorithm for Conformational Search of Organic Molecules: Implications for Materials Chemistry, *Comput. Chem.* **1998**, *22*, 345–351.

[59] T. Dandekar, P. Argos, Folding the Main Chain of Small Proteins with the Genetic Algorithm, *J. Mol. Biol.* **1994**, *236*, 844–861.

[60] F. Herrmann, S. Suhai, Energy Minimization of Peptide Analogues using Genetic Algorithms, *J. Comput. Chem.* **1995**, *16*, 1434–1444.

[61] T. Dandekar, P. Argos, Identifying the Tertiary Fold of Small Proteins with Different Topologies from Sequence and Secondary Structure using the Genetic Algorithm and Extended Criteria Specific for Strand Regions, *J. Mol. Biol.* **1996**, *256*, 645–660.

[62] B. T. Luke, Applications of Distributed Computing to Conformational Searches, In D. G. Truhlar, W. J. Howe, A. J. Hopfinger, J. Blaney, R. A. Dammkoehler (Eds.), *Rational Drug Design*, Springer, New York, **1999**, pp. 191–206.

[63] C. A. del Carpio, A Parallel Genetic Algorithm for Polypeptide Three Dimensional Structure Prediction. A Transputer Implementation., *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 258–269.

[64] J. T. Pedersen, J. Moult, Genetic Algorithms for Protein Structure Prediction, *Curr. Opin. Struct. Biol.* **1996**, *6*, 227–231.

[65] J. Wang, T. Hou, L. Chen, X. Xu, Conformational Analysis of Peptides using Monte Carlo Simulations Combined with the Genetic Algorithm, *Chemom. Intell. Lab. Syst.* **1999**, *45*, 347–351.

[66] R. S. Judson, M. E. Colvin, J. C. Meza, A. Huffer, D. Gutierrez, Do Intelligent Configuration Search Techniques Outperform Random Search for Large Molecules?, *Int. J. Quantum Chem.* **1992**, *44*, 277–290.

[67] D. E. Clark, G. Jones, P. Willett, Pharmacophoric Pattern Matching in Files of Three-dimensional Chemical Structures: Comparison of Conformational-Searching Algorithms for Flexible Searching, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 197–206.

[68] F. Glover, M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Boston, **1997**.

[69] J. Lee, H. A. Scheraga, S. Rackovsky, New Optimization Method for Conformational Energy Calculations on Polypeptides: Conformational Space Annealing, *J. Comput. Chem.* **1997**, *18*, 1222–1232.

[70] M. S. Head, J. A. Given, M. K. Gilson, "Mining Minima": Direct Computation of Conformational Free Energy, *J. Phys. Chem. A* **1997**, *101*, 1609–1618.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

3 Protein-Ligand Docking

Garrett M. Morris, Arthur J. Olson and David S. Goodsell

Abbreviations

EA	Evolutionary algorithm
EP	Evolutionary programming
ES	Evolution strategy
GA	Genetic algorithm
LGA	Lamarckian genetic algorithm
MC	Monte Carlo
MD	Molecular dynamics
RMSD	Root mean square deviation
SA	Simulated annealing
TS	Tabu search

3.1 Molecular Structure and Medicine

The fundamental tenet that biological function is inextricably linked to molecular structure underpins much of computational molecular biology. Understanding how molecular structures interact, in normal and pathological cases, is one of the most important goals of medicine today. Until relatively recently, new medicines were discovered by accident, and even today, trial and error can be an important part of drug discovery. It could be argued that high-throughput screening is our present-day equivalent, and is used to test large numbers of chemical compounds for potential activity against some pathological organism or process. When a compound shows activity, even if weak, this so-called *lead molecule* may become the basis of a concerted program to improve its potency.

However, when three-dimensional (3-D) structural information of the macromolecular target is available, a more directed approach may be taken. Structure-based drug design methods seek to use knowledge of the structure of the target macromolecule complexed with a given lead molecule. Enzyme functional groups that influence binding are identified and their contribution to substrate binding quantified. This knowledge can then be used to suggest improvements in the lead molecule that would enhance its binding. Putatively improved molecules can be made by organic synthetic chemists, and tested to determine the change in activity. The drug design cycle continues until a molecule with an inhibition constant in the nanomolar range or better is found. This is far from the end of the story, as problems of toxicology, bioavailability and synthetic cost, for example, must also be solved before human trials can begin.

Protein-ligand docking plays a central role in the process of structure-based drug design. In the lexicon of molecular biology, *docking* refers to the process by which one molecule binds noncovalently to another molecule. (Computationally, the term is more loosely used and may also encompass covalent binding.) A small molecule such as a substrate binds, or *docks*, to its target enzyme at the active site. The structure and stereochemistry of the enzyme – in particular at its active site – complement the shape and physico-chemical properties of the substrate so as to catalyze stereospecifically a particular reaction. Of course, some enzymes are more specific than others and can have a wide range of binding affinities for their inhibitors. Drugs usually consist of small molecules that mimic the transition state of the biochemical reaction that the enzyme catalyzes, but bind more tightly than the native substrate. Since experimental data for the structures of the complexes of a target enzyme exist – though not always with the desired ligand – it is immensely valuable to be able to predict the bound structure using the assistance of a computer. This facilitates the testing of many possible inhibitors computationally, before committing the necessary resources for their synthesis.

Protein-ligand docking has found other important uses in biomolecular science. A similar problem arises sometimes in X-ray crystallography in trying to solve the structure of a protein-ligand complex. The positions of the protein atoms can be determined relatively straightforwardly, by analogy with previously determined homologous proteins. In certain cases, however, the electron density of the ligand turns out to be too ambiguous to assign the atomic positions of its binding mode. Computational docking may be used to identify the most favorable conformation of the ligand when bound to the protein molecule, which may then be tested for compatibility with the experimental electron density. Similarly, the binding of substrates, products, and transition states may be predicted computationally, allowing study of enzyme mechanisms and bound states that are impossible to study experimentally.

3.2 Computational Protein-Ligand Docking

Despite improvements in computational power, docking remains a very challenging – and many would argue a "hard" or "NP-complete" – problem. Even with the fastest computers, many docking problems are still intractable. Exhaustive, systematic search methods, often referred to as "brute force" methods, are not always feasible even for the simplest of docking problems, namely docking a rigid molecule to a rigid receptor. Extremely rapid energy function evaluations and coarse sampling of parameter values are necessary to make systematic searches possible. To obtain an idea of the magnitude of the problem, if a 20 Å-sided cube is systematically scanned at translational intervals of 0.5 Å and rotational intervals of 5 degrees, there would be almost 2.4×10^{10} potential docking states. Clearly, we need to be able to search the possible binding configurations very efficiently, if we wish to use systematic search for docking. An example of this is DOT [1], which uses fast Fourier transforms and parallel computation to perform convolutions that rapidly calculate energies for rigid protein-protein docking.

There are various approaches to computational molecular docking. Flexible docking and design methods that use nonevolutionary as well as GA-based methods have been reviewed elsewhere. Rosenfeld et al. [2] focused on two applications, the design of retroviral protease inhibitors and the design of major histocompatability complex (MHC) receptor-specific peptide antigens. They considered methods that ranged from rigid to flexible docking, and that used Monte Carlo/molecular dynamics docking, in-site combinatorial search, ligand build-up methods, and active site mapping and fragment assembly. They also discussed the use of empirical free energy as a target function. Jones and Willett reviewed new algorithmic approaches in 1995 [3], while Lengauer and Rarey reviewed the progress in simulating the flexibility of molecules in docking in 1996 [4]. Finally, for an excellent description of docking in general, written for a readership of computer scientists, see [5].

Broadly speaking, docking is a search or optimization problem, which necessitates a way of ranking potential dockings. The complexity of biological molecules adds an additional wrinkle: fewer levels of conformational flexibility can be modeled to simplify unmanageable searches. The next two sections briefly discuss the types of scoring functions and the level of molecular flexibility that is modeled.

3.2.1 Scoring Functions

The extent of interaction between two molecules is usually expressed quantitatively by an energy value, which is ultimately based on a model of the physical chemistry of atomic interactions. There is often a trade-off between the sophistication of a scoring function and its computational cost. Rapid screening methods use simple geometric or steric criteria to allow very large numbers of ligands to be docked to a given target. However, these often miss potentially important ligands or binding modes, owing to their poor chemical selectivity. In the middle of the spectrum are enthalpic scoring functions, which account for the potential energy of interaction by using pairwise-atomic Lennard-Jones potentials, 12–10 hydrogen bonding terms, and Coulombic electrostatic terms. These resemble the force fields commonly used in molecular mechanics and molecular dynamics codes, like AMBER [6, 7], CHARMm [8] and GROMOS [9, 10], and require moderate computational resources. Perhaps the most complete scoring functions are those which include not only enthalpic terms but also entropic terms, and which estimate free energies of binding. These use a variety of ways of accounting for loss of conformational degrees of freedom and changes in solvation of the ligand upon binding. Examples of programs that use this type of scoring function are LUDI [11–13] and AutoDock 3.0 [14]. These free energy functions are often derived empirically using linear regression analysis, and require careful calibration using a large set of structures of protein-inhibitor complexes of known binding affinity. They have shown excellent results, showing a better prediction of experimental binding constants than purely enthalpic force fields.

The computational cost of energy evaluation may be reduced using precalculated grids [15]. The protein is placed within a grid volume, and a probe atom is sequentially placed at each point. The resultant energy of interaction is calculated. The grid may then be used as a look-up table during docking simulations, greatly speeding the process. Note, however, that this normally requires that the protein be modeled as rigid, although it is also possible to represent an ensemble of protein conformations using the grid formalism.

The EPDOCK method has been used to explore different energy functions in molecular docking [16] by looking to protein folding theory. It was noted that rough energy landscapes contain many kinetic barriers to docking. For proper docking, there is a thermodynamic requirement that the crystal structure has the global minimum energy, and a kinetic requirement that this global minimum is accessible to the search. In a rough landscape, there will be many competing minima separated by steep barriers that will compete with the proper minimum. Citing a paper on protein folding, Verkhivker et al. [16] posited that improved search methods are not the best solution to this "kinetic barrier" problem; instead, they preferred a smoothing of the energy landscape. They noted that standard molecular mechanics force fields give very rough surfaces, because of the large repulsive energies. They sought to find a less "frustrated" energy surface by softening the repulsive potentials, using a simple piece-wise function in which the repulsive energy barrier is easily modified. A common approach is to soften repulsive barriers early in a simulation, thus smoothing the energy landscape, and then to restore the barriers towards the end, theoretically funneling the docked conformation into the global minimum. In this work, however, Verkhivker et al. looked for one soft potential that gave the best results when used throughout the simulation. One might question this approach: molecular mechanics potentials are parameterized against physical data, so the steep repulsive barriers are based in fact, and cannot be lightly thrown away at the end. In any case, the coarse potential yielded poor results, with docked conformations rarely less than 2 Å RMSD from the crystallographic conformation.

3.2.2 Level of Allowed Molecular Flexibility

Typical proteins are composed of hundreds to thousands of atoms, most connected by rotatable bonds. Proteins are dynamic, and constantly undergoing small motions. The level at which this motion is modeled will often determine the accuracy and utility of the results. Of course, once again there is a trade-off, and in general docking methods tend to be more thorough and accurate as more conformational flexibility is added to the model. At the same time, however, they become slower and less able to ensure that the search space is adequately explored. Docking methods weight these conflicting needs differently. Common models, listed in order from simple to complex and from very fast to computationally demanding, include: (i) rigid protein and a rigid ligand; (ii) rigid protein and a ligand with rotatable bonds; (iii) rotatable bonds in both ligand and protein (often, the protein backbone is held rigid and selected side chains are flexible); and (iv) fully flexible protein and ligand (including bond stretching and bond-angle bending).

Early versions of DOCK [17, 18] fall into the first class, and are able to dock thousands of ligands to a protein in a single study. These methods ignore the conformational space of the ligand during the docking, simplifying the search space dramatically and accelerating the computation. The quality of the docked solutions can be poor, however, allowing only a gross triage of results. Early versions of AutoDock [15] are in the second class, and allow more accurate docking of several dozen ligands to a protein, with increased confidence in the docked conformation. Class (iii), including programs such as AutoDock 4.0 (currently undergoing testing) and to some extent GOLD [19], are now the best tools for docking and evaluating a set of a few dozen inhibitors. When combined with empirical free energy functions, this level of modeling provides a good prediction of both docked conformation and binding free energy. Molecular mechanics packages such as AMBER [6, 7] and CHARMm [8] are in the final class. Given a random starting point, they cannot sample the docking space broadly enough or simulate a time scale long enough to predict binding reliably. They are often used to "relax" docked conformations produced by the simpler approaches, and to perform free energy perturbation calculations, but these can be much more time-consuming than empirical free energy functions.

Note that the number of rotatable bonds alone is not always indicative of how hard the docking will be. In a study using the program PRO_LEADS [20], dockings of argotroban to thrombin were repeated with an increasing number of rotatable bonds, from one to seven. Surprisingly, the relative performance was quite similar for one through five rotatable bonds, but the performance suffered sharply when the sixth and seventh bonds were allowed to rotate. The explanation for this behavior is found in the branching of molecular structure. The first four rotatable bonds were linearly arranged in a long side chain, the fifth only rotated a carboxylate. In contrast, both the sixth and seventh bonds branched out at the "root" of the long side chain and caused large portions of the ligand to move. Thus, the size, shape and bonding topology of the ligand may have important consequences for the complexity of the problem, in addition to the number of rotatable bonds. It is worth noting here that careful tailoring of the operators of the search method, such as choosing appropriate points for crossover breaks, may yield considerable computational gains.

3.2.3 Testing and Evaluating Docking Methods

The validity of a docking method is typically established by trying to reproduce the crystal structure of one or more protein-ligand complexes. This problem differs from the typical tests of optimization methods, because the crystallographic conformation is rarely the global minimum of the search space. Since the search space is so large, the global minimum is unknown, but good search methods typically find that the best answers all fall near the crystallographic conformation. One must keep in mind, however, when evaluating docking methods by prediction of the crystallographic conformation, that both the energy function and the search method are being tested at once.

In a typical test, the ligand and the protein in a complexed structure are separated, the ligand is randomly translated, re-oriented and conformationally changed, and then docked back into the protein. With stochastic search methods, the ligand is docked several times, giving different results that are dependent on the random number generator and its seed value(s). These docked structures are ranked by score, energy or free energy. Some methods perform conformational clustering, before ranking the clusters by fitness. For each docked structure, the RMSD of the atomic positions in the docked structure from the corresponding atoms in the crystallographic structure is then calculated. Ideally, this should be small, say 1.0 Å or less. The most important criterion is that all the observed protein-ligand interactions be reproduced, including hydrogen bonding, steric and electrostatic interactions.

This is the ideal testing situation, when the answer is known. The best test of a docking method, however, is a blind test where the structure of the complex is not known. The protein or DNA structure is required, either from some other complex, or from the uncomplexed or *apo*-form. In these cases, the docking of the ligand of interest is repeated many times, and then analyzed to see if there are any predominant binding modes. If conformational clustering is performed, there should be one or two well-populated, distinct clusters, preferably with low energies or good scores. This assumes that the conformation of the protein does not change significantly upon ligand binding, and this tends to be a limitation of all docking methods. Biochemical knowledge, from mutagenesis experiments for example, can help to evaluate such docking results, in the absence of any structural experimental data for the complex.

Vieth et al. [21] presented an *ad hoc* "efficiency" measure to evaluate the success of docking. Most methods judge success by conformations that are within a given tolerance, typically 1 to 1.5 Å RMSD away from the crystallographic structure. The authors also included conformations in the range of 2 to 3 Å RMSD, denoting these as "partially docked", and weighting them fractionally within the efficiency measure. The validity of this score is questionable. In molecular docking, "partially docked" conformations can be worse than no solution at all. A 180 degree reversal of rings or peptide planes and other similar large errors will give RMSD values of about 3 Å, providing a misleading underestimation of binding energy and improper prediction of intermolecular interactions involved in recognition.

3.3 Evolutionary Algorithms for Protein-Ligand Docking

Evolutionary algorithms (EA) encompass a range of methods that are based on the principles of natural genetics and *neo-Darwinian* biological evolution. There are two main subgroups of search methods under the heading of EA: (i) genetic algorithm (GA) [22]; and (ii) evolutionary programming (EP) and evolution strategy (ES). Much of the language is similar in all these methods. The most common terms will be covered here, before we go on to discuss the applications of EA methods to protein-ligand docking in section 3.4.

In molecular docking, the *phenotype* can be thought of as the set of Cartesian co-ordinates of the protein-ligand complex, while the *genotype* encodes the information describing how to assemble the separated ligand and protein into a given complex.

In most cases, the particular arrangement of a ligand and a protein can be defined by a set of real values describing the translation, orientation and conformation of the ligand with respect to the protein. Typically, these are composed of a 3-D translation vector, three Eulerian rotation angles, and a collection of torsion angles that describe bond rotations in the ligand and protein. Other formulations based on matching techniques have also been reported (described in Section 3.5). These values are the ligand's *state variables*, and in the genetic and evolutionary algorithms, each state variable corresponds to one *gene*. A particular value of a given gene is called an *allele*. The *genome* corresponds to the set of genes that completely describe one *individual* in the *population*. The state of the ligand corresponds to its *genotype*, while the conversion of the state into atomic coordinates gives its corresponding *phenotype*.

The genetic algorithm is modeled closely on biological evolution, and it mimics many features observed in Nature. The simulation is composed of series of *generations*. Each generation is composed of a *population* of *individuals*, which in molecular docking are various candidate protein-ligand complexes. At each generation, a population of *parents* gives rise to a new population of *offspring*. Typically, a *selection* method is applied, and parents must compete with one another so that only the individuals with the best *fitness* give rise to offspring in the next generation. In molecular docking, the fitness is the total score, enthalpy or free energy of interaction of the ligand with the protein; it is evaluated using a *fitness function*. The iterative process of evolution continues until some termination criterion is met: the maximum number of generations, the maximum number of fitness function evaluations, or convergence of the population. Evolutionary algorithm-based docking methods use different variations of genome representation, parenting, fitness evaluation, and selection.

Two essential methods for creating offspring are borrowed from biological reproduction. Different genes from two parents are recombined and become inherited by their offspring in the process called *crossover* (sometimes this is also referred to as *recombination*). *One-point* crossover occurs when two parents, whose genomes are split at one position, say $A*B$ and $a*b$, mate to give two new children with genomes $A*b$ and $a*B$. In *two-point* crossover, the parents' genomes are split at two locations, say $A*B*C$ and $a*b*c$: the resulting offspring genomes are $A*b*C$ and $a*B*c$. *Uniform* crossover occurs when any number of genes may be crossed, and is typically controlled by a percentage of genes that may be crossed. The attractive global search properties of GA methods are typically attributed to the process of crossover, although crossover alone merely re-shuffles existing traits [23]. Hence, a second operator is used to explore the neighborhood of a particular solution.

The second method by which new offspring are created is *mutation*. Just as in real genomes, small random changes or mutations are occasionally made to individual genes. The amount of mutation is often chosen from a Gaussian distribution, and the breadth (sigma) of this distribution may also be allowed to evolve during the simulation. If the sigma values are allowed to evolve, this may be done in two ways: (i) "sigma-first", where the sigma values are modified and then the offspring generated; and (ii) "sigma-last", where offspring are generated and then the sigma values are modified. The sigma-first method is generally preferred [24]. Mutation might be thought of as performing a local search that complements the global search properties of crossover. A fundamental difference between GAs and EP/ES is that GAs use both crossover and mutation operators, while EP and ESs use only mutation operators.

Selection of the next generation's individuals from the current generation and the new offspring occurs based on the fitness of each individual. Thus, solutions better suited to their environment reproduce, while worse ones die. Several types of selection have been implemented in the context of docking. These include: (i) *step function* or *hard selection*, in which all parents in the top P % of the population have an equal likelihood of being selected; (ii) *roulette wheel selection*, where survivors are chosen probabilistically, with favorable conformations having a greater chance of surviving; and (iii) *tournament selection*, where survivors are chosen in a limited competition between a subset of the population. The *selection pressure* can have an important influence on the extent of the search and on the issue of premature convergence, a condition where all the dockings found early on are similar but not necessarily low in energy. Selection pressure is defined as the relative probability that the best individual will be chosen as a parent compared to the average individual. It can be reduced (and thus diversity maintained) by selecting parents of the next generation using roulette-wheel selection based on their *linear-normalized, rank-based fitness*, rather than fitness. This has been observed to help in improving the global search during a docking (see, e.g., [25]). Another consideration regarding selection is whether parents may survive into the next generation or only their offspring.

Some implementations extend the biological metaphor to include ecology. The population is subdivided by pseudogeographical barriers, giving rise to *niches*. These "islands" may exchange their best individuals occasionally by *migration* from niche to niche. Of course, this adds additional parameters to be optimized, including migration rates and niche sizes. In addition, some implementations include purely artificial concepts, which have common-sense appeal. One is *elitism*, which ensures that one or more of the best individuals always survives on to the next generation. Another is *diversity*, which ensures that a new population is composed of a set of differing individuals. This helps to prevent the search becoming too focused on a small region of the search space.

3.4 Published Methods

While EAs have been used in other fields for a number of years, they are relatively new in their application to molecular docking. The literature clearly documents the difficulty of molecular docking. In most reports, clever modifications are made to the traditional implementation of a method, allowing larger and more complex subjects to be studied.

One of the first reports used the well-tested negative-spheres description from the DOCK program [17] for assigning a fitness, later on adding a GA procedure for docking ligands flexibly [26]. The DOCK method fills the protein active site with spheres, and then roughly docks the ligand by matching atoms with centers of the spheres. This required an alternate formulation of the genome, to allow control of this matching process by the values in each gene. A bit string was used to represent the matching of a set of ligand atoms to receptor spheres, thus defining ligand position and orientation. A traditional bit string was used to assign the torsion angle values of the rotatable bonds in the ligand. The GA optimized the position/orientation matching and torsion angles simultaneously. The GA method was of comparable speed to DOCK when the ligands were treated rigidly, but slower when it allowed for ligand flexibility. Although no positional RMSD values were given for the GA docking results, one figure showed qualitative success in the flexible docking of methotrexate into the active site of dihydrofolate reductase (DHFR) when compared to the crystallographically determined position. Later, GA modifications of DOCK showed success in three systems: DHFR-methotrexate, thymidylate synthase-phenolphthalein, and HIV-1 protease-thioketal haloperidol [27].

The following year, Judson et al. presented a GA-based method for docking flexible ligands to rigid protein binding sites [28], an extension of their work on small molecule conformation searching also using a GA [29]. Their implementation was based on a modified, binary-encoded version of the breeding GA method of Mühlenbein and coworkers [30, 31]. They used a Gray code to represent each torsion angle (see section 3.5 for more information on Gray coding), and included single-point crossover, niches, elitism, and two types of selection: step function selection and roulette wheel selection. The energy evaluation consisted of a two-level filter, which quickly searched for steric overlap, and then calculated a more detailed energy for those that passed the first test. Judson et al. restricted the translational space severely, defining a "pivot" atom in the ligand and restraining it to a small space around the crystallographically known position. They were unable to find the crystallographic conformation of the tetrapeptide Cbz-GlyP-Leu-Leu to thermolysin if they allowed the GA to search the entire set of 20 rotatable bonds at once. They therefore introduced a "growing" algorithm. This began by allowing a small fragment including the pivot atom to explore the active site. The GA was easily able to find good positions for this fragment, and then after a user-defined number of generations the submolecule was "grown" to include the next nearest neighbors in the ligand, until the entire ligand was present. This reduced the number of dimensions of the search space and the method found conformations close in position and energy to the crystal conformation, although requiring a Herculean computational effort. In a later publication, the method performed well on eight large ligands docked to thermolysin, Gly-L-Tyr to carboxypeptidase A, and methotrexate to DHFR [32].

Duncan and Olson described a method called SurfDock that used an evolutionary programming method, combining aspects of genetic algorithms and simulated annealing for the docking of proteins [33]. The molecules were described by parametric representations of their surface geometry and surface physico-chemical properties, such as hydrophobicity and electrostatic potential. These descriptions were built around spherical harmonic functions, allowing varying levels of detail to be chosen by including more or fewer terms in the spherical harmonic series. The level of detail of the molecular representation was increased gradually during the docking. It began with low-resolution surfaces based primarily on shape complementarity, increasing the detail in five steps, finishing with high-resolution surfaces, with a different weight between the shape and energy terms. Their evolutionary method did not use the crossover operator essential to GA methods. The problem consisted of searching the translations and orientations of one rigid molecule with respect to another, so they reasoned that crossover between two good solutions could result in wildly different states, making the search inefficient. Their selection procedure is noteworthy in that it strove to maintain diversity in the population. This helped to compensate for the imprecise evaluation function and also avoided premature convergence. Duncan and Olson presented applications of their procedure with protein-protein complexes of known and unknown structure. A blind test using a protein-protein docking problem published in 1996 showed SurfDock to be very successful [34].

GAME (Genetic Algorithm for Minimization of Energy) was designed to dock one or more rigid ligands to a single, rigid, larger molecule [35]. The major focus was predicting molecular cluster structures, but the method is applicable to protein-ligand docking. Although the GA implemented was typical, the docking procedure was somewhat idiosyncratic. The GA used a single population of individuals, with chromosomes consisting of three translation genes and three rotation genes, each coded as an 8-bit string. If more than one molecule was being docked simultaneously, the individuals carried extra chromosomes to encode the additional states of these molecules. Four stages were used during docking, utilizing a typical enthalpic force field to evaluate the fitness. The first two stages restrained the ligand in its observed binding orientation and allowed the translations to explore only a narrow area around the observed binding position. The third phase began with the ligand at its crystallographically observed translation, but with a random orientation. The final phase began with the ligand placed randomly within a narrow 2 Å window and random orientations. Thus, the first three stages searched only three degrees of freedom, and the last just six. In this way it was possible to dock correctly two deoxyguanosine molecules simultaneously to actinomycin D, but it was noted that had there been conformational change upon binding then the method would have failed. Testing of the method with translational bounds large enough to encompass the whole receptor molecule was not discussed.

DIVALI, reported in 1995, uses an AMBER-type force field as the fitness function for a typical GA [36]. The program was tested on four protein-ligand complexes, using rigid and flexible ligands. Perhaps most noteworthy was the introduction of a masking operator to improve the binary chromosome-based genetic algorithm (see section 3.5). Clark and Ajay also used their genetic algorithm to perform local energy minimization on the crystal structures, allowing the ligand bonds to rotate, by seeding the initial population with the crystal structure and running for 50 generations. The implementation they used had difficulty with the docking of methotrexate, the ligand with the most rotatable bonds, to DHFR. They were forced to restrict the translation and rigid-body orientation of methotrexate in order to find successful dockings.

Jones, Willett and Glen described a GA method for docking a flexible ligand to a partially flexible protein [25]. Protein heavy atoms were kept rigid, but single bonds to hydrogen-bond donating heteroatoms were allowed to rotate. The GA was based on a typical steady-state-with-no-duplicates method, but an unusual representation of the genome was chosen. Two binary strings were used to encode the torsion angles of the ligand and of the protein side chains. In addition to this more conventional representation, a novel description was introduced to describe how the hydrogen bonding sites in the ligand might map to hydrogen bonding sites in the protein active site. Two strings of integers were used; these were expressed to give a phenotype by least-squares fitting the ligand into the active site, such that as many as possible of the hydrogen bonds defined by the mapping were satisfied. The fitness was defined as the number and strength of these hydrogen bonds plus the van der Waals energy of the complex. The method performed well on a number of large protein-ligand complexes, but required several attempts to find the proper conformation of the most difficult problem: folate and DHFR. In later work, in which the program gained the name GOLD (Genetic Optimization for Ligand Docking) [19, 37], the method achieved an excellent 71 % success rate on a data set of 100 "drug-like" complexes selected from the Protein Data Bank. In the new version, the force field for hydrogen bonding was improved and the Gray coding was discarded because of interference with crossover. The original implementation of GOLD and its forerunner relied upon the assumption that the ligand would form hydrogen bonds to the receptor, so it would not have performed as well with purely hydrophobic ligands. A recent report described a modification that additionally defines hydrophobic site points, where appropriate hydrophobic ligand atoms are mapped using the same integer-chromosome type approach.

EPDOCK was one of the first protein-ligand docking programs to use EP to perform a positional and conformational search of a ligand [38, 39], using a flexible ligand and a rigid protein. Each parent was mutated by a random amount to create one or more offspring; as is normal for EP methods, crossover was never used. A tournament selection was performed, and the mutation operator added a different random displacement with a zero-mean Gaussian distribution to each dimension. The variances for each dimension were also allowed to evolve, adding self-adaptive strategy parameters to the search dimensions [40–42]. The best solution at the end of the docking was subjected to conjugate gradient energy minimization, using the same fitness function as that used during the EP docking. EPDOCK performed well in tests with dihydrofolate reductase and HIV-1 protease. In later work, EPDOCK was shown to be successful in the docking of small flexible peptides to streptavidin [43], and two large inhibitors to the protein FKBP-12 [44], which required careful choice of water locations. A detailed test of different EP variations in EPDOCK has been presented in [24].

Meadows and Hajduk [45] reported a GA-based method for protein-ligand docking that exploited NMR-derived constraints [45]. The fitness function was a weighted sum of dispersion/repulsion energies and an error function that sought to match distances to experimental nuclear Overhauser effect (NOE) observations. The method was successful in predicting the complex of biotin with streptavidin. The authors note that the method is useful for defining the minimum set of conformations that fulfil the experimental constraints.

Westhead et al. compared four search methods using the program PRO_LEADS, ultimately finding that GA and EP methods were marginally better for protein-ligand docking [20]. Simple versions of a GA (one-point crossover and mutation) and EP, along with simulated annealing and tabu search (TS), were tested. Simple versions were chosen to reduce the number of parameters that needed to be optimized for each method, attempting to level the playing field. A simple scoring function similar to that of EPDOCK [38, 39] was used along with a grid-based energy look-up table, to speed evaluation. The GA and EP methods were found to be better for local searches, but the ranking of the four methods changed when different protein-ligand systems were tested in full docking simulations. According to a report in 1998 [46], TS was selected as the search method of choice in PRO_LEADS. It gave docked conformations with an RMSD of 1.5 Å or less from the crystal structure in 43 out of 50 protein-ligand test systems.

"Virtual molecular docking" was presented in 1997 in the program STALK, a parallel implementation of a GA-based docking program linked to a virtual reality interface [5]. Levine et al. used the PGAPack parallel genetic algorithm library, a general purpose, data-structure neutral library based on the Message Passing Interface, or MPI, which can take advantage of either dedicated parallel hardware or heterogeneous networks of workstations. The authors linked a GA method to the CAVE immersive environment. The user was able to observe the GA docking in action, or was able to interact with the system, performing the docking by hand. The GA was implemented on a parallel computer to allow the speed necessary for the link to visualization. A standard GA was employed and a typical master/slave model was used to execute function calls in parallel. The method was demonstrated on one system, showing that the GA found conformations of lower energy than those found by manual docking.

Vieth et al. [21] presented an analysis of energy functions and an analysis of search strategies for flexible docking based around the CHARMM energy function [8]. They compared a GA method with two Monte Carlo (MC) methods and molecular dynamics (MD). Their GA was a standard implementation with traditional crossover, but also included niches with migration. MC and MD were performed within CHARMM, and the AutoDock version 2.4 MC simulated annealing method [47] was also used. The authors applied an unusual three-stage simulation in all methods. The first stage took 60 % of the time, and searched widely over the space. In MC and MD, high temperatures were used, and in the GA, a high mutation rate was used. Energy potentials were softened to remove large energetic barriers. In the second stage, local energy basins found in the first stage were searched, taking about 30 % of the time. MC and MD employed lower temperatures, and the GA used a reduced mutation rate, but increased the crossover rate. Harder potentials were used in the last stage, the simulation being quenched using full van der Waals repulsion. Finally, a short MD simulation was performed on the GA and MC results. No comparison of this convoluted schedule with simpler schedules was reported. The authors compared methods by requiring that each used similar amounts of computer time in five standard test systems. The GA, MC, and MD within CHARMM all used a similar number of energy evaluations, but the AutoDock MC performed roughly ten times more (the fast grid-based energy evaluation made this possible within the same computational time limit). The surprising result is that the GA performed significantly better than MC and MD according to their efficiency score (see section 3.2.3), but using a 1 Å RMSD criterion all the methods performed roughly the same, with the GA lagging slightly behind.

Wang et al. [48] presented a two-stage GA to dock peptides to proteins and to predict protein-protein complexes. Both the protein and peptides were treated as rigid bodies. In the first stage, a typical GA was combined with a simple steric energy function and six full degrees of freedom were searched. In the second stage, a more complete AMBER force field was employed and the search was constrained in translation to lie close to the best conformations from the first stage. The method performed very well on eight complexes randomly chosen from the Protein Data Bank.

A hybrid GA method combining local search with a traditional GA was implemented within AutoDock [14, 49]. The local search method used was based on that of Solis and Wets [50]. After each generation, local search was used on a user-defined proportion of the population, essentially allowing these individuals to adapt to their environment. These newly adapted individuals were then allowed to pass on their genes to their offspring. Biologically, this contravenes neo-Darwinian principles, and was proposed by Lamarck in the early twentieth century [51]. Of course, computationally we are not limited to biological rules and can pass genetic information in whatever direction is needed to improve the search. To distinguish this hybrid GA from traditional implementations, it is referred to as the Lamarckian genetic algorithm (LGA). This method performed very well on a variety of test systems, and is described more fully in section 3.6.

3.5 Representation of the Genome

In a traditional GA, the parameters to be optimized are encoded as bit strings, and the mutational and crossover operations are applied directly to these strings. Application of this formalism to docking is straightforward: the translation and rotation angles defining orientation may be defined by six real numbers, and additional angles may be encoded for each torsional degree of freedom. Many of the published GA methods use this simple formalism, but problems may occur.

One problem is that of discontinuity. In the work of Dixon [26], the genome was represented as a traditional string of bits. This can lead to extreme changes when a high-order bit is flipped or a crossover occurs within the bit string of a gene, generating offspring phenotypes very different from their parents, and often well outside the bounds of possibility. The bit string-based GA search is therefore less efficient than those whose representations use a more natural description of the problem (see [52]), such as floating point values ranging from minus π to plus π radians for torsion angles. In effect, the representation is the window through which the search method views the world, and if badly chosen, can severely limit its performance [53–55].

Several approaches have been taken to improve the continuity of the search. The simplest method, employed in AutoDock [14] and many others, is to limit crossover points to breaks between variables, so that the break never occurs, for instance, in the middle of the *x*#co-ordinate but instead between the *x* and the *y*#co-ordinate. Judson et al. [28], for example, also employed Gray coding, which has the property that any two points next to each other in the problem space differ by only one bit (see for example [56]). This alleviates somewhat the problem that purely binary approaches suffer during mutation, namely that two points close in the representation space could be very distant in the solution space.

The bit string nature of the genome may be used to great advantage, however. In the program DIVALI [36], a masking operator was devised for use during modification of the genome. By fixing the most significant bit of each of the translational components, Clark and Ajay effectively divided the grid volume being searched into eight subcubes, and thus tested eight different binding hypotheses simultaneously in each docking. This also enforced diversity in the translational degrees of freedom during the search. Without this additional masking operator, their implementation of the genetic algorithm was unable to find an acceptable solution, for example, for the rigid-body docking of glucose to periplasmic binding protein after hundreds of different runs.

Other formulations of the genome have also been devised to take advantage of different models of protein-ligand interaction. As mentioned earlier, in the program GOLD [19, 25] the chromosome was made up of two binary strings and two integer strings. The binary strings were used in 8-bit bytes to encode the torsional angles in the ligand and in the protein. The torsion angle bytes were Gray-coded in the original implementation [25], but this was dropped in GOLD. The two strings of integers were used to encode the ligand's location and orientation, but not by typical translation and rotation values. Instead, the values encoded a set of possible hydrogen-bond mappings between the ligand and the protein. The position, *P*, of the integer value, *V*, in the first string of integers represented a hydrogen bond donated from the *V*-th hydrogen bond donor in the protein and accepted by the P-th lone pair in the ligand. The second string of integers similarly encoded the hydrogen bonds donated from the ligand to the protein. The conformation of the ligand encoded by the binary part of the chromosome was first constructed and then the ligand was placed in the active site by least-squares fitting according to the H-bond mappings defined in the genome.

3.6 Hybrid Evolutionary Algorithms

Hybrid methods are showing great promise for the creation of faster and more efficient docking tools. In these methods, local searches are performed on some members of the population during the simulation. A hybrid combines the attractive global search properties of evolutionary methods with methods that are better suited to exploring the local energy landscape.

Local search may be added in a biologically consistent way, preserving the biological flow of genetic information. Judson et al. [28] combined a short energy minimization with a traditional GA, and found improved results. During the GA search, a few steps of gradient minimization were applied to individuals whose energy was lower than some pre-defined value. The brief minimization yielded a less misleading energy and improved the ranking of the competing individuals.

Hybrid GA methods may also go beyond the biological model, by allowing genetic information to pass back from the phenotype to the genotype. A nonbiological hybrid GA-local search method was compared to more traditional SA and GA methods using AutoDock Version 3.0 [14]. The hybrid method added local searching to each generation. A certain fraction of the population was selected at random, and an energy minimization was performed on each of these individuals. This novel algorithm was termed an LGA because genetic information was passed from optimized phenotypes back to the genotype. To make their comparison a fair one, the number of energy evaluations was set to similar values in each test. In ten dockings for each of seven diverse test cases, the GA – which used a real-valued genome – obtained lower energies than SA, even though the SA dockings ended up using slightly more energy evaluations than the GA dockings. In fact, there were some cases where the ligand being docked using the SA method became trapped, either partially or wholly, within the protein. The GA was therefore more efficient at searching the docking space than SA. However, the hybrid LGA search method performed even more efficiently than the neo-Darwinian GA in docking tests, consistently finding lower energies than the GA.

The success of the LGA method has shown that the biological metaphor should be taken literally only as far as is useful. It is interesting to note that, whether a global search method is evolutionary or not, hybridizing it with a local search method improves the search efficiency. Trosset and Scheraga [57] demonstrated that the Monte Carlo search method could be improved by incorporating local energy minimization into their ECEPP/3 docking program. Perhaps other hybrid formulations, with or without biological counterparts, await discovery.

3.7 Conclusions

Genetic algorithms, evolutionary programming, and evolution strategies – collectively known as EAs – have proven remarkably successful and have revolutionized the field of protein-ligand docking. Thanks to continuing advances in both software and hardware, it has become feasible to study larger and more complex protein-ligand docking problems. Both the fields of computer science and molecular biology have gained from this interdisciplinary enterprise. Computer scientists have been able to test the limits of their algorithms on "real-world" problems, and molecular biologists have been able to approach systems of increasing biological relevance. Other benefits have emerged: it could be argued that this very active field of research is contributing to our understanding of the physical chemistry underlying molecular recognition (see for example [11]).

While it is impossible to say with certainty, future efforts in EA-based docking will probably be most fruitful in at least three fundamental areas. Improved hybrid search technologies and new and enhanced chromosomal encodings of the docking problem surely await discovery. There is also room for improvement in the fitness functions: ones that not only give good correlation with observed inhibition constants, but also give better selectivity in identifying the experimental binding mode while rejecting false#positives. A broader issue remains to be investigated: how much does the conformation of the *apo*-form of a protein change upon ligand binding? This question is beginning to be addressed in the realm of protein-protein association [58], but the implications for protein-ligand docking could be significant.

With true scientific hubris, we close with this speculative, philosophical note. We have looked to Nature for inspiration in the design of these new algorithmic approaches to docking. The LGA results suggest that the opposite may also be beneficial. If Nature could universally adopt a Lamarckian mechanism, evolution might progress more rapidly than it has thus far.

Acknowledgments

This work is publication 12945-MB from The Scripps Research Institute. It was supported by grant number NIH P01 GM48870.

References

[1] L. F. Ten Eyck, J. Mandell, V. A. Roberts, M. E. Pique, *Proceedings of the 1995 ACM/IEEE Supercomputing Conference*, IEEE Computer Society Press, Los Alamitos, CA, **1995**.

[2] R. Rosenfeld, S. Vajda, C. DeLisi, Flexible Docking and Design, *Annu. Rev. Biophys. Biomol. Struct.* **1995,** *24*, 677–700.

[3] P. Willett, Genetic Algorithms in Molecular Recognition and Design, *Trends Biotechnol.* **1995**, *13*, 516–521.

[4] T. Lengauer, M. Rarey, Computational Methods for Biomolecular Docking, *Curr. Opin. Struct. Biol.* **1996**, *6*, 402–406.

[5] D. Levine, M. Facello, P. Hallstrom, G. Reeder, B. Walenz, F. Stevens, STALK: An Interactive System for Virtual Molecular Docking, *IEEE Comput. Sci. Eng.* **1997**, *4*, 55–65.

[6] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz Jr., D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, P. A. Kollman, A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules, *J. Am. Chem. Soc.* **1995**, *117*, 5179–5197.

[7] S. J. Weiner, P. A. Kollman, D. A. Case, U. C. Singh, C. Ghio, G. Alagona, S. Profeta Jr., P. Weiner, A New Force Field for Molecular Mechanical Simulation of Nucleic Acids and Proteins, *J. Am. Chem. Soc.* **1984**, *106*, 765–784.

[8] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, M. Karplus, CHARMm: a Program for Macromolecular Energy, Minimization and Dynamics Calculations, *J. Comput. Chem.* **1983**, *4*, 187–217.

[9] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, J. R. Haak, Molecular Dynamics With Coupling to an External Bath, *J. Chem. Phys.* **1984**, *81*, 3684–3690.

[10] W. F. van Gunsteren, H. J. C. Berendsen, *GROMOS96*, BIOMOS b.v., Laboratory of Physical Chemistry, University of Groningen, Nijenborgh 4, 9747 AG, Groningen, The Netherlands, **1996.**

[11] H. J. Böhm, Prediction of Binding Constants of Protein Ligands: A Fast Method for the Prioritization of Hits Obtained From De Novo Design or 3D Database Search Programs, *J. Comput.-Aided Mol. Des.* **1998**, *12*, 309–323.

[12] H. J. Böhm, The Development of a Simple Empirical Scoring Function to Estimate the Binding Constant for a Protein-Ligand Complex of Known Three-Dimensional Structure, *J. Comput.-Aided Mol. Des.* **1994**, *8*, 243–256.

[13] H. J. Böhm, LUDI: Rule-Based Automatic Design of New Substituents for Enzyme Inhibitor Leads, *J. Comput.-Aided Mol. Des.* **1992**, *6*, 593–606.

[14] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, A. J. Olson, Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function, *J. Comput. Chem.* **1998**, *19*, 1639–1662.

[15] D. S. Goodsell, A. J. Olson, Automated Docking of Substrates to Proteins by Simulated Annealing, *Proteins: Struct., Funct., Genet.* **1990**, *8*, 195–202.

[16] G. M. Verkhivker, P. A. Rejto, D. K. Gehlhaar, S. T. Freer, Exploring the Energy Landscapes of Molecular Recognition by a Genetic Algorithm: Analysis of the Requirements for Robust Docking of HIV-1 Protease and FKBP-12 Complexes, *Proteins: Struct., Funct., Genet.* **1996**, *25*, 342–353.

[17] I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge, T. E. Ferrin, A Geometric Approach to Macromolecule-Ligand Interactions, *J. Mol. Biol.* **1982**, *161*, 269–288.

[18] B. K. Shoichet, I. D. Kuntz, Matching Chemistry and Shape in Molecular Docking, *Prot. Eng.* **1993**, *6*, 723–732.

[19] G. Jones, P. Willett, R. C. Glen, A. R. Leach, R. Taylor, Development and Validation of a Genetic Algorithm for Flexible Docking, *J. Mol. Biol.* **1997**, *267*, 727–748.

[20] D. R. Westhead, D. E. Clark, C. W. Murray, A Comparison of Heuristic Search Algorithms for Molecular Docking, *J. Comput.-Aided Mol. Des.* **1997**, *11*, 209–228.

[21] M. Vieth, J. D. Hirst, B. N. Dominy, H. Daigler, C. L. Brooks III, Assessing Search Strategies for Flexible Docking, *J. Comput. Chem.* **1998**, *19*, 1623–1631.

[22] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, **1975.**

[23] D. E. Goldberg, The Race, the Hurdle, and the Sweet Spot: Lessons from Genetic Algorithms for the Automation of Design Innovation and Creativity, in P. J. Bentley (Ed.), *Evolutionary Design By Computers*, Morgan-Kaufmann, San Francisco, CA, **1999**, pp. 105–118.

[24] D. K. Gehlhaar, D. B. Fogel, Tuning Evolutionary Programming for Conformationally Flexible Molecular Docking, in L. J. Fogel, P. J. Angeline, T. Bäck (Eds.), *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, MIT Press, Cambridge, MA, **1996**, pp. 419–429.

[25] G. Jones, P. Willett, R. C. Glen, Molecular Recognition of Receptor Sites Using a Genetic Algorithm with a Description of Desolvation, *J. Mol. Biol.* **1995**, *245*, 43–53.

[26] J. S. Dixon, Flexible Docking of Ligands to Receptor Sites Using Genetic Algorithms, C. G. Wermuth (Ed.), *Trends in QSAR and Molecular Modelling 92: Proceedings of the 9th European Symposium on Structure-Activity Relationships: QSAR and Molecular Modelling*, ESCOM Science Publishers, Leiden, The Netherlands, **1993**, pp. 412–413.

[27] C. M. Oshiro, I. D. Kuntz, J. S. Dixon, Flexible Ligand Docking Using a Genetic Algorithm, *J. Comput.-Aided Mol. Des.* **1995**, *9*, 113–130.

[28] R. S. Judson, E. P. Jaeger, A. M. Treasurywala, A Genetic Algorithm Based Method for Docking Flexible Molecules, *J. Mol. Struct. (THEOCHEM)* **1994**, *308*, 191–206.

[29] R. S. Judson, E. P. Jaeger, A. M. Treasurywala, M. L. Peterson, Conformational Searching Methods for Small Molecules. II. Genetic Algorithm Approach, *J. Comput. Chem.* **1993**, *14*, 1407–1414.

[30] H. Mühlenbein, *Foundations of Genetic Algorithms*, Morgan Kaufman, San Mateo, CA, **1991.**

[31] H. Mühlenbein, M. Schomisch, J. Born, *Parallel Comput.* **1991**, *17*, 619.

[32] R. S. Judson, Y. T. Tan, E. Mori, C. Melius, E. P. Jaeger, A. M. Treasurywala, A. Mathiowetz, Docking Flexible Molecules: A Case Study of Three Proteins, *J. Comput. Chem.* **1995**, *16*, 1405–1419.

[33] B. S. Duncan, A. J. Olson, Predicting Protein-Protein Interactions Using Parametric Surfaces, *Chem. Des. Auto. News* **1994**, *July*, 35.

[34] N. C. Strynadka, M. Eisenstein, E. Katchalski-Katzir, B. K. Shoichet, I. D. Kuntz, R. Abagyan, M. Totrov, J. Janin, J. Cherfils, F. Zimmerman, A. J. Olson, B. S. Duncan, M. Rao, R. Jackson, M. Sternberg, M. N. James, Molecular Docking Programs Successfully Predict the Binding of a Beta-Lactamase Inhibitory Protein to TEM-1 Beta-Lactamase, *Nature Struct. Biol.* **1996**, *3*, 233–239.

[35] Y. L. Xiao, D. E. Williams, GAME: Genetic Algorithm for Minimization of Energy, an Interactive Program for Three-Dimensional Intermolecular Interactions, *Comput. Chem.* **1994**, *18*, 199–201.

[36] K. P. Clark, Ajay, Flexible Ligand Docking Without Parameter Adjustment Across Four Ligand-Receptor Complexes, *J. Comput. Chem.* **1995**, *16*, 1210–1226.

[37] C. Sansom, Evolution Goes for GOLD In Silico, *Nature Biotech.* **1997**, *15*, 624–624.

[38] D. K. Gehlhaar, G. M. Verkhivker, P. A. Rejto, C. J. Sherman, D. B. Fogel, L. J. Fogel, S. T. Freer, Molecular Recognition of the Inhibitor AG-1343 by HIV-1 Protease: Conformationally Flexible Docking by Evolutionary Programming, *Chem. Biol.* **1995**, *2*, 317–324.

[39] D. K. Gehlhaar, G. M. Verkhivker, P. A. Rejto, D. B. Fogel, L. J. Fogel, S. T. Freer, Docking Conformationally Flexible Small Molecules into a Protein Binding Site Through Evolutionary Programming, in J. R. McDonnell, R. G. Reynolds, D. B. Fogel (Eds.), *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, MIT Press, Cambridge, MA, **1995**, pp. 615–627.

[40] T. Bäck, H.-P. Schwefel, An Overview of Evolutionary Algorithms for Parameter Optimization, *Evol. Comput.* **1993**, *1*, 1–24.

[41] N. Saravanan, D. B. Fogel, Learning Strategy Parameters in Evolutionary Programming: An Empirical Study, in A. V. Sebald, L. J. Fogel (Eds.), *Proceedings of the Third Annual Conference on Evolutionary Programming*, World Scientific, **1994**, pp. 269–280.

[42] N. Saravanan, D. B. Fogel, K. M. Nelson, A Comparison of Methods for Self-Adaptation in Evolutionary Algorithms, *BioSystems* **1995**, *36*, 157–166.

[43] N. K. Shah, P. A. Rejto, G. M. Verkhivker, Structural Consensus in Ligand-Protein Docking Identifies Recognition Peptide Motifs That Bind Streptavidin, *Proteins: Struct., Funct., Genet.* **1997**, *28*, 421–433.

[44] P. A. Rejto, G. M. Verkhivker, Mean Field Analysis of FKBP12 Complexes With FK506 and Rapamycin: Implications for a Role of Crystallographic Water Molecules in Molecular Recognition and Specificity, *Proteins: Struct., Funct., Genet.* **1997**, *28*, 313–324.

[45] R. P. Meadows, P. J. Hajduk, A Genetic Algorithm-Based Protocol for Docking Ensembles of Small Ligands Using Experimental Restraints, *J. Biomol. NMR* **1995**, *5*, 41–47.

[46] C. A. Baxter, C. W. Murray, D. E. Clark, D. R. Westhead, M. D. Eldridge, Flexible Docking Using Tabu Search and an Empirical Estimate of Binding Affinity, *Proteins: Struct., Funct., Genet.* **1998**, *33*, 367–382.

[47] G. M. Morris, D. S. Goodsell, R. Huey, A. J. Olson, Distributed Automated Docking of Flexible Ligands to Proteins: Parallel Applications of AutoDock 2.4, *J. Comput.-Aided Mol. Des.* **1996**, *10*, 293–304.

[48] J. Wang, T. Hou, L. Chen, X. Xu, Automated Docking of Peptides and Proteins by Genetic Algorithm, *Chemom. Intell. Lab. Syst.* **1999**, *45*, 281–286.

[49] W. E. Hart, C. R. Rosin, R. K. Belew, G. M. Morris, Improved Evolutionary Hybrids for Flexible Ligand Docking in AutoDock, in C. A. Floudas, P. M. Pardalos (Eds.), *Proc. Intl. Conf. on Optimization in Computational Chemistry and Molecular Biology*, Kluwer Academic Publishers, B.V., The Netherlands, **2000**, pp. 209–230.

[50] F. J. Solis, R. J.-B. Wets, Minimization by Random Search Techniques, *Math. Op. Res.* **1981**, *6*, 19–30.

[51] J. B. Lamarck, Of the Influence of the Environment on the Activities and Habits of Animals, and the Influence of the Activities and Habits of These Living Bodies in Modifying Their Organization and Structure, in *Zoological Philosophy*, Macmillan, London, **1914**, pp. 106–127.

[52] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin, **1996.**

[53] K. A. De Jong, Genetic Algorithms: A 10 Year Perspective, in J. J. Grefenstette (Ed.), *Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Hillsdale, NJ, **1985**, pp. 169–177.

[54] K. A. De Jong, On Using Genetic Algorithms to Search Program Spaces, in J. J. Grefenstette (Ed.), *Proceedings of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, Hillsdale, NJ, **1987**, pp. 210–216.

[55] K. A. De Jong, Learning with Genetic Algorithm: An Overview, *Machine Learning* **1988**, *3*, 121–138.

[56] Z. Michalewicz, *Binary or Float?*, in Z. Michalewicz (Ed.), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin, **1996**, pp. 98–99.

[57] J. Y. Trosset, H. A. Scheraga, Reaching the Global Minimum in Docking Simulations: a Monte Carlo Energy Minimization Approach Using Bezier Splines, *Proc. Natl. Acad. Sci. U S A* **1998**, *95*, 8011–8015.

[58] M. J. Betts, M. J. Sternberg, An Analysis Of Conformational Changes On Protein-Protein Association: Implications For Predictive Docking, *Prot. Eng.* **1999**, *12*, 271–283.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

4 *De Novo* Molecular Design

Valerie J. Gillet

Abbreviations

ADME	Absorption, distribution, metabolism, and excretion
CoMFA	Comparative molecular field analysis
EA	Evolutionary algorithm
EP	Evolutionary programming
ES	Evolution strategy
GA	Genetic algorithm
MCSS	Maximum common substructure
QSAR	Quantitative structure-activity relationship
RMS	Root mean square
TS	Tabu search

4.1 Introduction

There are many applications in the chemical industries where novel molecules are required, for example, novel bioactive compounds are required as potential drug candidates in the pharmaceutical industry, novel pesticides are required in agrochemicals, and novel polymers are required in the petrochemical industry. Many computational tools have been developed to assist in the design of novel compounds; however, computer-aided molecular design is still well known as a complex and computationally demanding task. Traditionally, the design of new compounds is approached through trial and error: a compound is synthesized and tested and then a new compound is designed based on the results of the tests. This process typically proceeds through many iterations. *De novo* design, on the other hand, is a direct approach and refers to the process of designing compounds from first principles, that is, without basing the design directly on existing known compounds. *De novo* design is combinatorial in nature with the combinatorics arising from the number of different element types that is available for constructing molecules and the variety of ways in which they can be linked together. For molecules of nontrivial size, this combinatorial explosion results in a search space that is much too large to allow systematic searching of all possibilities. Indeed, it has been estimated that there are of the order of 10^{60} molecules of up to 30 nonhydrogen atoms containing the elements C, N, O, and S [1]. Therefore, sophisticated computational techniques must be employed to allow the search space to be sampled effectively for a given application.

One obvious way to reduce the search space is to restrict the types of molecule that are generated to those that meet the requirements of the application area. This can be achieved by defining appropriate constraints and ensuring that the molecules generated are consistent with the constraints. For example, when designing new compounds as potential drug candidates, the constraints on molecular design may include the receptor site of a protein where the molecule should bind in order to exert a particular biological effect. In other cases, the target protein may not be known; however, it may be possible to specify some constraints on molecular design from series of known active and inactive compounds. In the area of polymer design, the potential candidates are typically constrained to have some desired physical properties such as a particular thermal conductivity. Focusing the design according to constraints can be an effective way of preventing the design of inappropriate molecules; however, even when the constraints are taken into account, the search space is usually still of such an enormous size that a systematic search is not feasible.

Exploring the extremely large search spaces that characterize *de novo* design is a task that is well suited to the application of evolutionary algorithms (EAs) [2]. EAs attempt to solve complex problems using ideas that are based on natural selection and Darwinian evolution. EAs have been developed for the design of molecules in two-dimensional (2-D) and three-dimensional (3-D) forms. Examples of design in 3-D include the design of molecules to bind to a protein receptor site, and the design of molecules to fit a pharmacophore hypothesis or a 3-D quantitative structure-activity relationship (QSAR) such as a CoMFA plot. Examples of molecular design in 2-D include the design of molecules with 2-D structure similar to a known active compound, the design of molecules to fit a traditional 2-D QSAR, and the design of molecules with desired physical properties such as thermal conductivity.

There are three main classes of EAs: genetic algorithms (GAs) [3,4]; evolutionary strategies (ESs) [5], and evolutionary programming (EP) [6]. Most EA applications in *de novo* design have involved GAs, and a brief description of a basic GA is given in the next section. EAs in general are described more fully in Chapter One. The chapter then focuses on the application of EA techniques to the *de novo* design of molecules in different domains. An important part of *de novo* design is the definition of the constraints which are used to guide the design of the molecules. The different types of constraints that have been used in *de novo* design are reviewed. The various EAs that have been developed for *de novo* design under different constraints are then described. In the context of defining the constraints, a number of EAs have been developed for the elucidation of pharmacophores and for the design of receptor models. In the absence of the 3-D structure of a target receptor site, these models can be used as constraints for the *de novo* design of potential drug molecules; these related methods are also reviewed. The applications of EAs in other areas of computational chemistry are covered in the other chapters of the book and have also been reviewed elsewhere [7–13].

4.2 Overview of a Genetic Algorithm

The basic operation of a GA is illustrated in Fig. 1. A GA operates on a population in which each member is usually a fixed length string that represents a potential solution to the given problem. Each population member is assigned a *fitness* according to how well it satisfies the solution requirements. The GA then enters a breeding phase where individuals or parents are chosen to reproduce, generating *offspring* (or children). Evolutionary pressure is applied by basing the selection of parents on the *fitness scores* so that fitter members have a greater chance of being selected than less fit members. The selection method is usually *roulette wheel selection*, a process in which each individual is assigned to a sector of a roulette wheel, with the size of the sector proportional to the fitness of the individual. The size of the sector corresponds to the probability of the individual being selected, thus, fitter individuals have a higher chance of being selected. The standard genetic operators are *crossover* and *mutation*. Crossover involves the exchange of information between two parents, while mutation involves altering one or more randomly chosen bits in the string representation of one parent. Many nonstandard genetic operators have also been devised that are appropriate for particular problem domains. Following genetic modification, the offspring are then inserted into the population to replace existing members, and the new population is scored. The GA iterates through the breeding and scoring phases until some termination condition is reached, for example, a given number of iterations or convergence of the entire population to a maximum fitness.

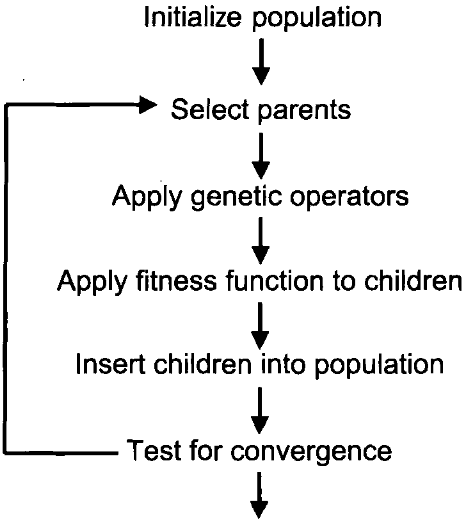


Figure 1. Basic operation of a genetic algorithm.

The basic GA outlined here can be adapted easily to solve problems in different domains with the development of a GA to solve a particular problem involving consideration of the following: the encoding scheme used to represent the problem space within a population of individuals; the fitness function used to evaluate the goodness of an individual; the selection criterion used to choose parents for breeding; the particular genetic operators that are used to produce offspring; and the termination condition for the algorithm.

4.3 Defining the Constraints

The constraints on *de novo* design depend on both the application for which the molecules are being designed and the knowledge that is available. In drug discovery projects, the knowledge about the system can be in the form of a target protein whose structure has been determined by X-ray crystallography, nuclear magnetic resonance or homology modeling. In such cases, the aim is to design novel compounds that are able to bind to the receptor site of the target protein and thus exert some biological effect.

The receptor site of a protein provides constraints in terms of its steric and physico-chemical properties, and potential drug candidates must have properties that are complementary to the receptor site in order to bind. The shape of the receptor site is clearly important in determining the size and general shape characteristics of candidate molecules. The shape constraints are usually described by a volume or boundary into which the molecules must fit. As well as shape, the physico-chemical properties of the receptor site are also important and ligand-receptor binding is generally assumed to be determined by key interactions formed by groups within the surface of the receptor site. From an analysis of the receptor, it is possible to identify regions within the active site where, if a ligand atom is placed, it is likely to interact with the protein via, for example, a hydrogen bond. There are two basic approaches to identifying interaction sites. One approach is based on calculating energies, usually through some variation on the GRID algorithm [14, 15], the other approach is rule-based [16–18]. These localized regions can be used to guide structure generation. Hydrophobic interactions are also important, though they tend to be less localized and so can be more difficult to use to guide structure generation. Nevertheless, they can be incorporated into scoring functions that estimate how likely a molecule is to bind.

The aim of *de novo* design in the context of a receptor site is to generate molecules that satisfy as many of the interaction sites as possible without violating the steric constraints. The receptor provides a set of external constraints on *de novo* design; however, there are also additional constraints on the molecules themselves that are independent of the receptor. For example, the molecules should be synthetically accessible and chemically stable. Ideally, all of this knowledge should be incorporated into the design process.

In the absence of the structure of the target protein, it may be possible to derive constraints in the form of a pharmacophore [19]. A pharmacophore is defined as the spatial arrangement of functional groups required for binding, and it can be determined from series of known active and inactive compounds. In addition, it may also be possible to derive some shape constraints in the form of allowed and excluded volumes. The constraints on *de novo* design, in this case, are to build structures that contain the given pharmacophore. Several programs have been developed for generating pharmacophores, including methods that are themselves based on evolutionary algorithms. The programs based on EAs are reviewed in section 4.5. A number of EA-based methods have also been developed for building pseudo-receptor sites that can then be used as constraints for *de novo* design. These methods are reviewed in section 4.6.

In drug discovery, *de novo* design techniques have also been applied to the generation of molecules that are constrained to be similar to some known target structure, where the similarity may be defined in different ways, for example 2-D and 3-D similarity [20], and also to fit QSARs. A QSAR model is normally a linear combination of functions of molecular descriptors that relate the descriptors to the activities of a series of molecules. EAs have also been used to derive QSARs themselves, and these methods are described in Chapter 5.

Most developments in *de novo* design have been in the design of small molecules as potential drug candidates. However, there have also been applications of evolutionary algorithms in areas such as the *de novo* design of polymers. Here, the aim is to design new molecules with desired physical properties, and hence the constraints are given in terms of required physico-chemical properties such as thermal conductivity. Another example of a property that has been the focus of *de novo* design efforts is biodegradability. In these property-driven examples, the molecules are designed in 2-D; there is no requirement to consider the 3-D conformations of molecules that are relevant for receptor binding.

4.4 Applications of EAs to *De Novo* Design

The early 1990s saw intense research into the development of methods for the *de novo* design of novel compounds to bind to a receptor site of known structure. The many programs developed during this period are reviewed in [21–23]. The constraints on *de novo* design for this type of application have been described in the previous section. The representation of the constraints varies from one program to another, but all are based on the concept of localized interaction sites within a steric cavity. In *de novo* design, molecules are built from smaller units, or building blocks. Exploring the search space fully requires the building of all possible molecules, and hence the building blocks should be the smallest units possible – that is, atoms – and the atoms should be connected together in all possible ways. However, since such a systematic search for all structures satisfying the constraints is not feasible, many different strategies have been developed for sampling the search space. One way of reducing the size of the search space is to use fragments as building blocks, rather than atoms, so that fewer steps are required to build a molecule of a given size (see, e.g., [17,24]). This approach has the advantage that some unlikely combinations of atoms can be avoided, and the disadvantage that certain combinations of atoms can never be built. Another way is to restrict the design domain to particular compound types, for example, peptides [25]. However, even when the search space is restricted by predefined sets of building blocks, it is still too large to be searched systematically. For example, there are 20^4 or 160 000 different tetra-peptides when just their 2-D graphs are considered, and many more possibilities when conformational space is explored, as is required in structure-based drug design. Thus, sophisticated search algorithms are also required for sampling the search space even when the types of molecules that can be generated are restricted. Several different algorithms have been developed including: the use of random numbers to select a subset of possibilities at each building step (see, e.g., [26]); sampling of conformational space (see, e.g., [24]); and the use of tree searching techniques to determine the order in which partially built solutions are processed so that the more promising partial solutions are expanded first [24]. Here, the focus is on programs that have been developed using EAs as a way of exploring the search space. These methods may be thought of as whole molecule approaches in that population members evolve into new and hopefully better molecules as the search progresses, rather than being built sequentially.

Glen and Payne reported one of the earliest evolutionary algorithms for *de novo* design in the Chemical Genesis program [27]. The method is based on a GA that operates on the 3-D structures themselves, rather than on a string representation that is the traditional form of a chromosome in a GA. Operating on the atom co-ordinates and bond connectivities directly has the advantage that realistic conformations of molecules that fit the constraints are selectively bred. Molecules can be evolved to fit different types of constraints, for example, to fit a receptor site, to fit a pharmacophore or to have certain desired molecular properties. The properties or constraints are divided into scalar properties such as molecular weight and surface area, and surface and grid properties that are spatially dependent and are used to describe properties such as electrostatics, molecular shape, and hydrogen bonding ability. A set of optimal properties or target constraints is calculated and the evolving molecules are scored according to their similarity to the target constraints. A penalty is added for molecules with high internal strain energy. When designing molecules to fit a receptor site, the constraints reflect the requirement for complementarity between the molecules and the receptor site.

The algorithm may be initialized in a number of ways: with ethane as the seed molecule; with a series of random fragments extracted from a library; or with a known starting structure, for example, a bound inhibitor within a receptor site. Parent molecules are selected for breeding using roulette wheel selection. The genetic operators are applied directly to the molecules themselves and include specialized implementations of crossover and mutation. Crossover can be either terminal crossover or region crossover. In the former, a terminal part of a molecule is identified (single bond connection) and removed, and the molecule is connected to a similar terminal part of another molecule. In the latter, the internal portions of molecules are exchanged via two single bond connections. In both types of crossover, the exchange is made between parts of the molecules that occupy the same volume of space. Several different types of mutation operator are encoded including: translation and rotation of the molecule; rotation about a bond; modifying an atom or bond type; adding or removing a fragment; creating or breaking a ring. Following crossover and mutation, the geometries of the newly created molecules are optimized using molecular mechanics. The algorithm has been tested for the optimization of a bound inhibitor and also for the *de novo* design of many novel molecules to bind to the enzyme dihydrofolate reductase (DHFR).

Blaney et al. [28] reported a GA-based method for *de novo* design that is based on the representation of molecules using SMILES notation [29], where SMILES is a linear notation that encodes 2-D chemical structure. In the method, the population of candidate molecules is represented by a series of SMILES strings. The GA is initialized with a randomly created population of 10 to 30 molecules each with 5 to 20 atoms. The frequency of occurrence of bond and atom types was determined by an analysis of the Medchem database [30] of 25 000 molecules of known biological activity. A molecule is evaluated by generating 3-D conformations and docking them into the receptor site using a method based on distance geometry [31]. The binding energy is estimated and additional rules are also used to reward or penalize specified substructures. The genetic operators crossover and mutation are performed on the SMILES strings to evolve new molecules. The method was tested on DHFR, where it was able to generate some interesting suggestions for inhibitor design. Subsequently, a more general approach has been described, where molecules can be represented as either SMILES strings or 2-D molecular graphs and they can be evolved to fit a variety of constraints including: similarity to a target compound; similarity to a pharmacophore model; fit to a binding site; fit to a CoMFA model; and actual binding energy [32].

LeapFrog [33] is a program for *de novo* design available within the Tripos suite of programs. Although LeapFrog is not actually an evolutionary algorithm it does use operators, called *moves*, that are similar to genetic operators and that operate directly on the molecules. The method has not been published in detail. It proceeds through iterations where in each pass a ligand is selected and a randomly chosen move performed which modifies the ligand in some way. A number of different moves are available, for example, bridging between ligands, changing the orientation of a ligand, joining and fusing fragments onto the ligand and rotating about bonds. As LeapFrog proceeds, the total number of ligands stored increases, and so a move called Weed is available which allows ligands to be removed from consideration. LeapFrog operates in three different modes: (i) OPTIMIZE, to suggest improvements to existing leads; (ii) DREAM, which proposes new molecules that are expected to have good binding; and (iii) GUIDE, which supports interactive design by evaluating user-defined modifications. Molecules can be designed to fit a receptor site or a CoMFA model [34].

Jansen et al. [35] report the application of LeapFrog to the design of new ligands to fit models of receptor sites. The receptor-site models are built using the program Yak [36] that operates in a number of steps. The first step is to superimpose a series of known active compounds. Vectors are then generated around ligand atoms that are associated with hydrogen bonds and hydrophobic interactions. Finally, residues in the model receptor site are positioned according to an analysis of the clusters of vectors for all superimposed ligands. Receptor-site models were developed for the paclitaxel binding site of tubulin and the serotonin 5-HT$_{1A}$ receptor. In the tubulin case, the receptor site model was built around a series of known ligands; in the serotonin case, the model was based on a combination of homology modeling and the Yak concept. In both cases, LeapFrog was used in DREAM mode to design novel compounds and in OPTIMIZE mode to suggest modifications to previously docked ligands. When LeapFrog was run in DREAM mode, neither case produced interesting ligands. However, the OPTIMIZE mode resulted in some viable solutions with novel interaction modes compared to the known ligands, some of which might represent starting points for further design.

Westhead et al. [37] developed a GA to refine a set of structures that had been generated using the *de novo* design program PRO_LIGAND. The set of structures output from PRO_LIGAND forms the initial population of the GA. As in the Chemical Genesis program described earlier, the GA operates directly on the molecules themselves. The molecules are evaluated using graph theory routines that fit the molecules onto interaction sites within the receptor site and the fitness function involves a weighted sum of the number of interaction sites hit, together with some physical properties of the molecules themselves such as the numbers of rings and rotatable bonds. The interaction sites include hydrogen bond donors, hydrogen bond acceptors, aliphatic lipophilic and aromatic lipophilic sites. The population is evolved by using roulette wheel parent selection and the genetic operators crossover and mutation. Crossover is implemented by cutting the two parents across a single bond and reconnecting the resulting fragments. Two types of mutation are possible: torsional mutation in which one or more torsion angles in a structure are randomized; and a rule-based atom mutation similar to that used by Glen and Payne [27]. The method has been tested by refining structures generated for DHFR and structures generated as mimics for the DNA-binding antibiotic, distamycin.

Two other programs for *de novo* design that are based on evolutionary algorithms have been announced recently; however, details of the methodologies used have not been published. They are mentioned here for completeness. The programs are EAInventor [38] and LigBuilder [39].

The characteristics of programs developed for the *de novo* design of molecules to fit 3-D constraints are summarized in Table 1. The programs have been developed for the design of potential novel drugs and the fitness functions generally involve, either directly or indirectly, some estimate of binding affinity together with penalties for molecules that have undesirable characteristics. In most cases, the chromosomes of the GA are the molecules themselves, in which case the fitness function can be calculated directly. In the program developed by Blaney et al. [29, 32] the chromosomes encode 2-D structures, and 3-D structures must be generated prior to estimating binding affinity.

Table 1. Characteristsics of programs for the *de novo* design of molecules to fit 3-D constraints.

Program	Chromosome encoding	3-D constraints	Fitness function
Chemical Genesis [27]	3-D molecule manipulated directly	Receptor site and Pharmacophore	Comparison of calculated properties with target constraints and internal strain energy penalty
Blaney et al. [28]	Linear SMILES string/2-D graph	Varied constraints including: receptor site	A number of fitness functions available including: estimate of binding energy and similarity to a target compound. Fragment-based penalties also included
LeapFrog [33]	3-D molecule manipulated directly	Receptor site and CoMFA model	No information available
PRO_LIGAND [37]	3-D molecule manipulated directly	Receptor site	Number of interaction sites hit and physical property penalties

A number of programs using EAs have been developed for the design of molecules with desired physical properties. These programs are concerned with properties that depend on 2-D structure rather than the ability to adopt a given 3-D conformation as is required for receptor binding. Venkatasubramanian et al. [40–42] have developed a GA for the design of polymers with desired properties. The properties under consideration include density and thermal conductivity. Molecules are represented as linear strings of symbols where each string may be composed of one or more segments, and each segment represents an elemental, substructural or monomer unit. Nesting is used to differentiate between backbone units and side chain substituents. The initial population of molecules is created at random by choosing units from a predefined set of functional groups. The properties of each polymer in the population are calculated and compared to the desired values. The fitness function is based on the amount by which the calculated properties deviate from the desired properties. Molecules are chosen for reproduction at random but in proportion to their fitness, and are evolved using crossover and mutation as well as a number of new genetic operators that were developed specifically for the problem domain. The new operators include: a blending operator, which involves the end-to-end connection of two parent molecules; insertion and deletion operators, which add or remove groups from either the main chain or a side chain; and the hop operator whereby a randomly chosen group in the main chains "hops" to a randomly chosen position on the main chain.

The first case study was to design semiconductors with required property constraints. The genetic operators were restricted to modifications to the main chain units. Several feasible solutions matching the property constraints were found. The second study involved locating an exact target polymer where the constraints were the properties of the target molecule itself within a specified tolerance. When the initial population was constructed from a random combination of four main chain base groups and three side chain base groups, with manipulations of both main chains and side chains allowed, the GA successfully located the target polymer. The GA was then applied to a much larger problem involving 17 main chain groups and 15 side chain groups with constraints defined by a target structure and including additional restrictions on the bulkiness of the molecules and their chemical stability. In many cases, the algorithm was successful in identifying the target. However, the results varied in terms of success rate; that is, in the number of successful identifications of the target in several runs of the algorithm, as well as in the quality of the final solutions obtained.

One of the difficulties when using GAs is determining optimal parameter settings such as population size and the relative frequencies with which the different genetic operators are applied. Often, parameters are chosen through trial and error. In a more recent study, Sundaram and Venkatasubramanian [43] investigated the sensitivity of their GA with respect to different parameter settings. They found that the optimal settings varied according to the target constraints, and were subsequently able to identify a number of factors that could be used to improve the performance of the GA. These include investigating different sampling schemes, dynamically tuning the parameters during a run based on the performance of the GA, and allowing the user to interact with the GA. (These issues are discussed in more detail in Chapter 12.)

Nachbar [44] uses genetic programming (a subclass of GAs where the chromosomes are trees rather than linear strings) to evolve molecules to fit a QSAR or QSPR (quantitative structure-property relationship) model and a set of desired chemical attributes. The method makes use of the relationship between chemical structures and graph theory. The molecular graph of an acyclic structure is a tree that maps directly into a genetic programming framework where the chromosomes are trees rather than linear strings. The tree is organized as subexpressions where each subexpression is headed by a bond and the leaves are atoms. The root of the whole tree represents the molecule itself. Rings are handled using a labeling system that indicates ring closures and is similar to the encoding used in the SMILES notation [29]. The handling of rings in this way places special restrictions on the genetic operators in that subexpressions can only be excised or replaced if they contain both members of a labeled pair.

Structure generation begins with the creation of a random set of molecules. A molecule is created by firstly selecting a root atom, appending bonds to it to satisfy its valency, and then adding appropriate atoms to the ends of the bonds. The process is repeated with the new atoms becoming the roots of the next level substructures. The structures are grown down each branch until either a terminal atom is added or some depth criterion is reached. Rings can then be created by replacing some number of pairs of hydrogen atoms by ring labels. Some additional rules are used to prevent the creation of very unusual structures. The genetic operators include crossover and mutation as well as some new operators that are specific to the problem domain. Substructure or subexpression crossover and mutation are implemented by exchanging or replacing subexpressions with other subexpressions that have equivalent heads, or bonds, so that valency is maintained. Only subexpressions that either contain no ring labels or that contain matching ring labels are eligible for crossover and mutation. Atom and bond mutations require the appropriate number of hydrogens to be added or removed. Special operators are required to allow ring opening and closing and ring expansions and contractions. For example, ring opening is implemented by locating a pair of ring labels and replacing them by hydrogens; conversely, ring closing is implemented by replacing a pair of hydrogens by ring labels.

The method has been tested by designing compounds to fit a QSAR that relates valence to toxicity. A regression model, for use in the fitness function of the GA, was derived using a training set of 27 alcohols, ethers, aldehydes, ketones, and esters. The GA was initialized with molecules that were constructed from the atoms H, C, and O. The fitness function was a combination of the QSAR, with molecules having predicted toxicity outside a desired range being linearly penalized, and penalties for undesirable chemical features. The method was able to generate a population of diverse structures that included several of those that were used to generate the QSAR. In addition, 30 % of the designed molecules fell in the desired toxicity range.

Globus et al. [45] describe a method for designing molecules based on 2-D similarity to a target compound. Their method is an extension of genetic programming that they call "genetic graphs". Whereas genetic programming evolves tree-structured programs, the method of Globus et al. evolves graphs, that is, cycles are possible within the genetic graph. Thus, the genetic graphs can represent rings within a molecule directly without the use of special labels as required in Nachbar's genetic programming method [44] and in the linear SMILES notation [29]. The graphs represent molecules with the vertices corresponding to atomic elements and the edges corresponding to single, double, or triple bonds. Hydrogen atoms are implicit and valency is enforced. A random population is created by choosing a random number of atoms between half and twice the size of the target molecule. Atomic elements are assigned through a random choice of the elements available within the target. Bonds are added at random to construct a spanning tree, and then a random number of additional bonds is added to form cycles. The number of cycles is between half and twice the number in the target compound. Parents are chosen by tournament selection in a steady-state genetic algorithm. The population is evolved using crossover only. Crossover involves choosing a set of edges that when removed, causes a graph to be split into two disconnected parts. The disconnected parts from two different parents are then exchanged to form offspring. As implemented, crossover allows for cycles to be opened or closed and even allows complex ring systems such as cages to be generated. However, the method cannot generate new cycles when none exist within the population. Unlike Nachbar's method, no special-purpose ring-opening and closing operators are necessary. The fitness function is based on similarity to the target calculated using an atom-pairs based method similar to that described by Carhart et al. [46]. Five test cases have been reported: the design of molecules that are similar to benzene, cubane, purine, diazepam, and morphine.

Devillers and Putavy [47, 48] have designed a hybrid system that uses a back-propagation neural network in conjunction with a GA to design molecules presenting varying degrees of biodegradability. The GA searches for combinations of structural fragments among a set of descriptors, and the candidate molecules are scored using the neural net. The neural net was trained with a set of 38 heterogeneous molecules and then subsequently tested on 49 molecules where it was able to predict the biodegradability correctly for 45 out of the 49 test molecules. The molecules are described by means of 13 binary descriptors that indicate the presence or absence of features that are known to influence the environmental fate of chemicals. Examples of the descriptors used include: an N-containing heterocycle; more than two chlorine atoms; the occurrence of only C, H, O, and N in the molecule. Four molecular weight classes were also defined as an additional descriptor. The GA is initialized by creating a random population of individuals, with a chromosome in the GA consisting of 14 genes corresponding to the 14 molecular descriptors. The fitness of each individual is calculated using the neural net. Note that some of the descriptors are mutually exclusive; for example, the occurrence of more than two chlorine atoms in a molecule and the occurrence of only C, H, O, and N atoms. The fitness function includes a penalty to ensure that chromosomes containing inconsistent descriptor combinations are removed from the population. The genetic operators are crossover and mutation, and candidates are selected for breeding according to their fitness. On termination of the GA, the final population is examined for the percentage occurrences of the various descriptors. The GA regularly generates molecules with codes that match those in both the training set and the predicted set, and also gives results that are in agreement with known characteristics about biodegradability. Further experiments were performed in which structural constraints were introduced in the form of structural descriptors that must be present or absent in the solutions. This involved modifying the initialization process so that it was no longer random, and also the crossover and mutation operators to avoid the loss of the selected descriptor.

The characteristics of the programs developed for the *de novo* design of molecules to fit 2-D constraints are summarized in Table 2. The programs have been developed for the design of molecules in a variety of different domains, for example, from drug design to the design of polymers. Consequently, the fitness functions have been developed to optimize the design of molecules to fit a variety of different constraints. A variety of different encoding schemes are used to represent molecules within the programs.

Table 2. Characteristics of programs for the *de novo* design of molecules to fit 2-D constraints.

Program	Chromosome encoding	2-D constraints	Fitness function
Venkatasubramanian et al. [40–42]	2-D molecule represented as a linear string	Physico-chemical properties	Comparison of calculated physical properties with target values
Nachbar [44]	2-D molecule represented as a tree	QSAR/QPSR	QSAR and penalties for undesirable chemical features
Globus et al. [45]	2-D molecule represented as a graph	2D Similarity	Atom-pairs based similarity measure
Devillers et al. [47, 48]	Molecular fragments represented as a linear string	Biodegradability	Neural net (trained to estimate biodegradability

4.5 Applications of EAs to Pharmacophore Mapping

As already mentioned, in many drug discovery projects it is the case that the 3-D structure of the protein binding site is not known, but activity data are available for a set of compounds that are known to interact with the receptor. In such cases, it may be possible to derive a pharmacophore that can be used to provide constraints for *de novo* design. Pharmacophore elucidation methods attempt to align active molecules in order to identify common structural features that are presumed responsible for the observed activity. To be most effective, programs should take into account conformational flexibility in the molecules and be fully automated. Several approaches to pharmacophore elucidation that are themselves based on EAs have now been reported, and are reviewed here as providing potential starting points for *de novo* design.

Payne and Glen [49] reported the first use of a GA for the alignment of flexible molecules. Constraints are defined as a set of pharmacophore features and distance constraints previously calculated from a template molecule, and the GA attempts to find an optimum superposition of the molecules with respect to the constraints. A chromosome of the GA encodes the conformation of each molecule to be aligned as torsional rotations and parameters that control the conformations of rings. The chromosome is divided into sections, with each section describing the conformation of a different molecule in the set. The GA is initialized with a random population of chromosomes, each representing a different conformation for each molecule. A chromosome is scored by applying the appropriate conformational changes to the original molecules and then least-squares fitting the molecules to the constraints. The fitness function measures the RMS difference between the point properties of the target molecule and the oriented molecules. Other constraints can also be used in the fitness function such as charge distribution and penalties for van der Waals contacts. Individuals are chosen for breeding using roulette wheel selection, and the genetic operators are crossover and mutation.

Subsequently, Jones et al. [50, 51] devised a GA-based program, called GASP, that is able to perform flexible molecular overlay and pharmacophore elucidation without requiring prior knowledge of the pharmacophoric pattern. The chromosomes of the GA encode torsional rotations for each molecule and the intermolecular mappings between putative pharmacophore features in the molecules and a base molecule which is chosen as the one with the smallest number of pharmacophoric features. Examples of pharmacophoric features include hydrogen bond donor protons, hydrogen bond acceptor lone pairs and ring centers. A least-squares fitting process is used to overlay molecules so that as many as possible of the structural equivalences suggested by the mapping are formed. The fitness of a chromosome is given as a combination of the number and the similarity of the overlaid features, the volume of the overlay, and the van der Waals energy of the molecular conformations. The GA uses an island model where the population of the GA is divided into a small number of subpopulations which are kept relatively isolated from one another. Partitioning the population can help to maintain diversity in the population as a whole and allows for the parallel implementation of the algorithm. The genetic operations consist of crossover, mutation and migration, in which a chromosome can move from one subpopulation to another. Roulette wheel selection is used to choose chromosomes for breeding.

Handschuh et al. [52] have also developed a GA for the flexible superposition of 3-D structures. Their method finds maximum common substructures (MCSs) between two molecules. The GA is combined with a gradient-based optimization method known as "directed tweak" [53]. The GA finds an optimal assignment of the atoms between different structures and the geometric fit is optimized through a combination of the GA and the directed tweak technique. Molecules are represented at the atomic level, with a chromosome encoding matching atom pairs between two structures and torsional rotations that represent the conformations of both molecules. The search for the MCS involves two criteria: the size of the substructure, and the fit of the matching atoms. These are conflicting criteria since a larger MCS will by definition have a larger deviation in the co-ordinates of the superimposed atoms. Rather than attempt to combine the different criteria into a single weighted fitness function, as is done in GASP, a set of Pareto solutions is obtained at the end of each run whereby an optimal geometric fit is found for each possible size of MCS. (In Pareto optimization, an optimized state is reached if none of the parameters can be improved further without making another one worse.)

The selection of individuals for breeding is made using restricted tournament selection. In tournament selection, once offspring have been bred they then compete with other individuals for insertion into the new generation. Each individual is compared to a number of randomly chosen *opponents*. If the individual's fitness exceeds that of an opponent, it receives a *win*. At the end of the process, the next generation is chosen from the individuals with the highest number of wins. In restricted tournament selection, tournaments are held between similar individuals, rather than individuals chosen at random, in order to maintain the diversity of the population. The new offspring compete with the most similar individuals found in a subset of the existing population. Two new genetic operators are introduced, in addition to crossover and mutation. They are "creep" and "crunch". The creep operator refines solutions found using crossover and mutation by adding a matching pair of atoms to the match list. The crunch operator reduces the size of the MCS by eliminating a match pair that is responsible for bad geometric distance parameters. The directed tweak optimization method is applied to the superposition represented in a chromosome prior to calculating the fitness function. Handschuh et al's method has been found to produce results that are comparable to those produced by GASP.

Holliday and Willett [54] describe a GA for pharmacophore mapping that identifies the smallest 3-D pattern of pharmacophore points within a set of molecules with the requirement that each molecule contains some user-defined number of the points. Thus, not all molecules are required to contain all the pharmacophore points. The method as described is restricted to finding pharmacophoric patterns in rigid structures, that is, conformational flexibility is not taken into account.

The algorithm consists of two GAs. The first is used to select a population of m points in each molecule that are maximally superimposed. A K-point pharmacophore is then taken as the number of unique points in the superposition. For example, if all the molecules contain the same pharmacophore of size m then K is equal to m. In practice, not all the points will superimpose and K is equal to the number of unique points in the superposition. The second GA refines the output from the first, that is, it tries to find a better fit between the K-point site and the input molecules. In the first GA, the chromosome consists of N sets of m integers where there are N molecules and the minimum number of pharmacophore points required in each molecule is m. The GA is designed to optimize the mN points so that the molecules are maximally superimposed. Breeding is performed using crossover, mutation and roulette wheel parent selection. The fitness function measures how well the m points in each molecule overlap based on calculated interatomic distances. The better the overlap between the points represented in the chromosome, the more fit is the chromosome. Each chromosome in the final population represents a potential K-point site. The population of the second GA is seeded with the output from the first. Each chromosome in the second GA represents the 3-D co-ordinates of the K-point site. The genetic operators include crossover and mutation where different forms of mutation are allowed including: removing a point, adding a randomly selected point, and replacing two points by a single point at their midpoint. Each chromosome is checked using a clique-detection procedure to confirm that an m-point match exists between the K-point site encoded in a chromosome and each of the input molecules. If this is not the case, the chromosome is discarded. The fitness function is the inverse sum of the number of points, K, and the chromosome tolerance which is a measure of how well the molecules fit the pharmacophore. The GA attempts to minimize both the number of points and the tolerance. It has been tested on four sets of compounds having the same activity, including angiotensin-converting enzyme (ACE) inhibitors and antifilarial antimycin analogs. In each case, a three-point pharmacophore was found that was common to all compounds in the class ($m = 3$; $K = 3$) and a five-point pharmacophore was found which had four points in common with each member of the class ($m = 4$; $K = 5$). When tested on a set of 15 heterogeneous ligands selected from the Brookhaven Protein Data Bank (PDB), a six-point pharmacophore was found which had three points in common with each of the 15 ligands ($m = 3$; $K = 6$).

CLEW [55] is a program for the generation of pharmacophore hypotheses through machine learning techniques. Whereas GASP uses information about actives only, the CLEW program also takes account of information generated from the inactive molecules. The analysis begins with the generation of complete sets of conformations using the WIZARD program [56]. The next step involves the perception of pharmacologically relevant features in the molecules such as hydrogen bond donors, hydrogen bond acceptors, hydrophobic regions, π systems, and ionic groups. The molecules are classified into broad structural classes based on their 2-D structure, and machine learning techniques are applied to the actives and inactives within a class to derive rules that relate structure to activity. The common features among all classes are found and a geometrical fitting program is used to find a 3-D fit of the features between minimized conformations of the active structures. Three different techniques are used to derive the rules: a logic-based machine learning method; a GA; and EP. The chromosomes of the GA represent rules in the form of fragments. A fragment is described by a bit string in which the bits indicate the presence or absence of structural features such as donors and acceptors, and also an indication of the environment of the feature. A chromosome is scored according to the relative occurrence of the fragment in the active and inactive molecules. The GA is initialized with randomly assigned rules. The GA uses tournament selection to select rules to include in the breeding population. New rules are evolved using crossover and mutation. The EP method uses the same chromosome representation and the same evaluation function. However, it uses the entire gene pool as the breeding pool. The best members from the gene pool and the breeding pool are used as the next generation's gene pool, and only mutation is implemented; that is, there is no crossover operator. The GA and EP method were found to produce similar results to the logic-based method, although it is reported that numerous runs of the GA were required to duplicate the logic-based results.

4.6 Applications of EAs to Receptor Modeling

Several methods are now available that attempt to build a model of the receptor site from known ligands. Some examples of these are: the widely used CoMFA [34] models that represent the 3-D field properties around a series of superimposed molecules using a probe atom to compute the interaction energies; Hahn's [57] method for building a receptor surface model that is composed of many triangle meshes that attribute properties at points on the surface; and the Yak program [36] that was described briefly in section 4.4. Receptor models can be used as constraints for *de novo* design in the absence of a known receptor site. Programs for pseudoreceptor modeling that are based on EAs are reviewed in this section.

The aim of the GERM (Genetically Evolved Receptor Models) program [58, 59] is to build an atomic-level model of a receptor site based on a small set of known structure-activity relationships. First, a set of compounds with experimentally determined activities are superimposed in low-energy conformations. An initial model of the receptor site is then constructed by placing 40–60 atoms on a grid around the surface of the superimposed active compounds. The atom types consist of 14 typical protein atoms and a null atom type to correspond to no atom at all at a given position. A GA is used to alter and optimize the atom types of the receptor site in order to maximize the correlation between drug–receptor binding calculated using molecular mechanics and measured drug activity. The chromosomes of the GA encode potential receptor models as bit strings; each bit corresponds to a grid point placed around the superimposed ligands together with pseudoreceptor atom assignments. The genetic operators include crossover and mutation. The fitness function involves firstly calculating the energy between each ligand and the receptor model encoded in a chromosome and then calculating the correlation between calculated drug-receptor binding and measured drug activity. The GA is initialized with between 1000 and 5000 models, depending on the number of compounds in the training set. Individuals are chosen for breeding using roulette wheel selection. The method is able to generate several thousand models, all of which have a high correlation between calculated binding energy and measured bioactivity. The models successfully discriminate quantitatively between compounds with varying biological activities.

The PARM (Pseudo Atomic Receptor Model) program [60] is based on the GERM algorithm, but with two significant differences. One difference is that receptor atoms are assigned to the grid according to charges that have been precalculated at every grid point and that are based on the charges of the ligand atoms that are closest to it. The second difference is that the fitness function is based on the cross-validated R^2 of the QSAR equation derived to correlate the interaction energy of each ligand with its bioactivity. The use of cross-validation can prevent overfitting that may occur with the conventional correlation coefficient used in GERM.

Vedani and Zbinden [61] have developed a program called Quasar that is also similar to GERM. Quasar differs from GERM in that it generates a family of receptor surfaces in which the surface is adapted to each ligand used in the study, rather than an averaged surface over all ligands. In addition, Quasar includes "H-bond flip-flop particles" which can simultaneously act as hydrogen bond donors and hydrogen bond acceptors towards different ligand molecules. Initially, an averaged receptor surface is constructed by surrounding the ligands with virtual particles followed by energy minimization (with or without consideration of conformational flexibility in the ligands). Then, each ligand is taken in turn and the averaged receptor surface is optimized to give a family of receptor models. Finally, a GA is used to optimize the family of models by placing atoms on the receptor surfaces in a similar approach to that used in GERM.

4.7 Discussion

Several EAs have been described for *de novo* design, with the majority of the algorithms actually being based on GAs which are a subclass of EAs. The basic GA, illustrated in Fig. 1, has been modified in different ways according to the particular application. One criterion that varies within the methods is the encoding scheme that is used to represent potential solutions within the chromosomes of the GA. Programs that have been developed for designing molecules to fit 3-D constraints can be divided into those that represent 3-D structure within the chromosomes [27, 33, 37] and those that are based on 2-D representations [28]. In the 3-D approaches, the programs usually operate on the molecules directly, that is, the genotype and phenotype of the GA are the same. (In the pharmacophore elucidation programs, 3-D conformations are encoded in linear strings as rotations, translations and torsional rotations [49–52].) The 3-D approaches have the advantage that molecules can be evolved in reasonable conformations directly; however, many different conformations of the same 2-D structure may be represented as different individuals within a single population reducing the diversity of structures that is explored. The 2-D representations such as SMILES strings fit the standard GA model directly and allow more structures to be explored; however, 3-D conformations have to be generated within the GA itself in order to evaluate the goodness of a potential solution. Generating 3-D conformations is computationally expensive and can result in a large number of conformations for each molecule if conformational space is explored fully. Another disadvantage of the 2-D methods is that it can be more difficult to ensure that the molecules represented in the chromosomes are chemically and conformationally sensible. Some newer interesting approaches are based on genetic programming and make direct use of the correspondence between 2-D chemical structures and graphs [44, 45].

The fitness function is another crucial component of an EA, and the particular fitness function that is applied depends on the problem domain. For example, in receptor binding, the fitness function generally attempts to estimate the strength of drug-receptor binding. In other cases, the fitness function measures how well a molecule fits a pharmacophore hypothesis, or how similar it is to some known target compound. Fitness functions have also been developed that measure how well the evolving molecules fit a QSAR or QSPR.

In many cases, specialist genetic operators have been developed, in addition to the standard mutation and crossover operators, to allow chemistry space to be well represented and to ensure that the molecules that are generated are chemically reasonable. This is particularly the case when the GA operates on the molecules themselves.

EAs are typically controlled by a large number of parameters, for example, population size; the relative frequencies of applying the different genetic operators; and so forth. One of the difficulties in implementing EAs is choosing an optimal set of parameters. Often, parameters are chosen through a process of trial and error. One recent study [43] has included a detailed investigation of optimal parameter settings within a GA; however, the conclusions were that the optimal parameters are dependent on the particular constraints to which the GA is applied. Thus, this remains an area of difficulty and suggests that self-adapting GAs in which the parameters are altered during a run of the GA may be beneficial (see Chapter 12).

One of the attractive features of EAs in general is that it is relatively easy to incorporate multiple objectives within the fitness function. For example, in designing potential drug candidates, factors such as synthetic feasibility, chemical stability and properties of absorption, distribution, metabolism, and excretion (ADME) are important, as well as binding affinity. Many approaches encode multiple objectives using a weighted fitness function with a weight assigned to each objective under consideration; however, this may not be the best approach, especially when the objectives are conflicting. An interesting approach to dealing with conflicting objectives is to search for the Pareto surface [52], where solutions at the surface are Pareto optimal; that is, none of the objectives can be improved without causing a degradation in the others.

4.8 Conclusions

Many interesting programs have been developed for *de novo* design, and it is clear that EAs are well suited to tackling the problem. However, there are still significant limitations associated with the various methods. As should be clear, the fitness function is a crucial component of EAs, but this remains one of the major limitations – particularly when designing molecules to fit a receptor site. In this respect, the fitness function for *de novo* design is closely related to the scoring functions used in docking programs and described in Chapter 3. In *de novo* design, the fitness function is applied very frequently, and hence the speed of the calculation is an important factor as well as the accuracy of the result. Much effort has gone into the development of scoring functions; however, this remains as one of the limiting factors in the success of *de novo* design methods. Another potential problem with using EAs to evolve molecules is their tendency to produce molecules that are synthetically inaccessible and are therefore unlikely candidates for drug discovery.

A limitation of all programs developed so far for *de novo* design of ligands to fit a receptor site is the assumption that the receptor remains rigid during binding. This is clearly an invalid assumption. Some docking programs do now account for protein flexibility in a limited way, but there are as yet no reported examples of *de novo* design that allow changes to occur in the conformation of the receptor site.

One of the characteristics of programs for *de novo* design, whether or not they are based on EAs, is the tendency to generate large numbers of solutions all of which satisfy the constraints. Whereas generating a wide range of diverse solutions is usually one of the aims of *de novo* design, it is important that ways of navigating through the large answer sets are provided. This is another area in which there is clearly still much room for improvement.

Despite the intense interest in *de novo* design methods during the early 1990s, the emphasis in the pharmaceutical industry has shifted more recently towards the techniques of combinatorial chemistry and high-throughput screening. However, recent advances in genomics and structure determination techniques will continue to result in more and more receptors becoming available as targets for structure-based drug design. Therefore, *de novo* design techniques are likely to re-emerge as important tools, and an area of increasing importance will be the integration of *de novo* design techniques with combinatorial chemistry.

References

[1] R. S. Bohacek, C. McMartin, W. C. Guida, The Art and Practice of Structure-Based Drug Design: A Molecular Modeling Perspective, *Med. Res. Rev.* **1996**, *16*, 3–50.

[2] J. H. Holland, *Adaption in Natural and Artificial Systems*, MIT Press, Cambridge, MA, **1992**.

[3] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, IEEE Press, Piscataway, NJ, **1995**.

[4] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, Reading, MA, **1989**.

[5] H.-P. Schwefel, *Numerical Optimization of Computer Models*, Wiley, Chichester, **1981**.

[6] L. J. Fogel, A. J. Owens, M. J. Walsh, *Artificial Intelligence through Simulated Evolution*, Wiley, New York, **1996**.

[7] D. E. Clark, Some Current Trends in Evolutionary Algorithm Research Exemplified by Applications in Computer-Aided Molecular Design, *MATCH* **1998**, *38*, 85–98.

[8] D. E. Clark, Evolutionary Algorithms in Rational Drug Design: A Review of Current Applications and a Look to the Future, in A. L. Parrill, M. R. Reddy (Eds.), *Rational Drug Design: Novel Methodology and Practical Applications*, ACS Symposium Series Vol. 719, American Chemical Society, Washington DC, **1999**, pp. 255–270.

[9] J. Devillers (Ed.), *Genetic Algorithms in Molecular Modelling*, Academic Press Limited, New York, **1996**.

[10] G. Jones, Genetic and Evolutionary Algorithms, in P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, P. R. Schreiner (Eds.), *Encyclopedia of Computational Chemistry*, John Wiley & Sons, Chichester, UK, **1998**, Volume 2, pp. 1127–1136.

[11] A. L. Parrill, Evolutionary and Genetic Methods in Drug Design, *Drug Discovery Today* **1996**, *1*, 514–521.

[12] D. E. Clark, D. R. Westhead, Evolutionary Algorithms in Computer-Aided Molecular Design, *J. Comput.-Aided. Mol. Des.* **1996**, *10*, 337–358.

[13] R. Judson, Genetic Algorithms and Their Use in Chemistry, in K. B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry*, Wiley-VCH, New York, **1997**, Volume 10, pp. 1–73.

[14] P. J. Goodford, A Computational-Procedure for Determining Energetically Favorable Binding-Sites on Biologically Important Macromolecules, *J. Med. Chem.* **1985**, *28*, 849–857.

[15] R. C. Wade, P. J. Goodford, Further Development of Hydrogen-Bonding Functions For Use in Determining Energetically Favorable Binding-Sites on Molecules of Known Structure. 2. Ligand Probe Groups with the Ability to Form More Than Two Hydrogen-Bonds, *J. Med. Chem.* **1993**, *36*, 148–156.

[16] D. J. Danziger, P. M. Dean, Automated Site Directed Drug Design – The Prediction and Observation of Ligand Point Positions at Hydrogen-Bonding Regions on Protein Surfaces, *Proc. R. Soc. London* **1989**, *B236*, 115–124.

[17] H.-J. Bohm, Site-Directed Structure Generation by Fragment-Joining, *Perspect. Drug Discovery Des.* **1995**, *3*, 21–33.

[18] V. J. Gillet, G. Myatt, Z. Zsoldos, A. P. Johnson, SPROUT, HIPPO and CAESA: Tools for De Novo Structure Generation and Estimation of Synthetic Accessibility, *Perspect. Drug Discovery Des.* **1995**, *3*, 34–50.

[19] L. B. Kier, *Molecular Orbital Theory in Drug Research*, Academic Press, New York, **1971**.

[20] G. M. Downs, P. Willett, Similarity Searching in Databases of Chemical Structures, in K. B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry*, Wiley-VCH, New York, **1995**, Volume 7, pp. 1–66.

[21] V. J. Gillet, A. P. Johnson, Structure Generation for De Novo Design, in Y. C. Martin, P. Willett (Eds.), *Designing Bioactive Molecules*, American Chemical Society, Washington DC, **1998**, pp. 149–174.

[22] M. A. Murcko, Recent Advances in Ligand Design Methods, in K. B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry*, Wiley-VCH, New York, **1997**, Volume 11, pp. 1–65.

[23] D. E. Clark, C. W. Murray, J. Li, Current Issues in De Novo Molecular Design, in K. B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry*, Wiley-VCH, New York, **1997**, Volume 11, pp. 67–125.

[24] P. Mata, V. J. Gillet, A. P. Johnson, J. Lampreia, G. Myatt, S. Sike, A. L. Stebbings, SPROUT: 3-D Structure Generation Using Templates, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 479–493.

[25] J. B. Moon, W. J. Howe, Computer Design of Bioactive Molecules – A Method for Receptor-Based De Novo Ligand Design, *Proteins: Struct. Funct. Genet.* **1991**, *11*, 314–328.

[26] Y. Nishibata, A. Itai, Confirmation of Usefulness of a Structure Construction Program Based on Three-Dimensional Receptor Structure For Rational Lead Generation, *J. Med. Chem.* **1993**, *36*, 2921–2928.

[27] R. C. Glen, A. W. R. Payne, A Genetic Algorithm for the Automated Generation of Molecules within Constraints, *J. Comput.-Aided. Mol. Des.* **1995**, *9*, 181–202.

[28] J. M. Blaney, J. S. Dixon, D. J. Weininger, Evolution of Molecules to Fit a Binding Site of Known Structure. Paper presented at the Molecular Graphics Society Meeting on Binding Sites: Characterising and Satisfying Steric and Chemical Restraints, York, U.K., March **1993**. (Abstracts of this and other papers presented at this meeting are available from Prof. R. E. Hubbard, Department of Chemistry, University of York, York, YO1 5DD, United Kingdom. E-mail: rod@yorvic.york.ac.uk).

[29] D. J. Weininger, SMILES a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules, *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.

[30] The Medchem database is maintained by The MedChem Project and BioByte Corp., 201 W. Fourth St. Claremont, CA, USA.

[31] J. M. Blaney, J. S. Dixon, Receptor Modeling By Distance Geometry, *Ann. Rep. Med. Chem.* **1991**, 26, 281–285.

[32] D. Weininger, *Method and Apparatus for Designing Molecules with Desired Properties by Evolving Successive Populations*, US Patent 5434796, **1995**.

[33] LeapFrog is available from TRIPOS Inc., 1699 South Hanley Road, Suite 303, St. Louis, MO 63144.

[34] H. Kubinyi (Ed.), *3-D QSAR in Drug Design*, ESCOM, Leiden, **1993**.

[35] J. M. Jansen, K. F. Koehler, M. H. Hedberg, A. M. Johansson, U. Hacksell, G. Nordvall, J. P. Synder, Molecular Design Using the Minireceptor Concept, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 812–818.

[36] A. Vedani, P. Zbinden, J. P. Snyder, P. A. Greenidge, Pseudoreceptor Modeling: The Construction of Three-Dimensional Receptor Surrogates, *J. Am. Chem. Soc.* **1995**, *117*, 4987–4994.

[37] D. R. Westhead, D. E. Clark, D. Frenkel, J. Li, C. W. Murray, B. Robson, B. Waszkowycz, PRO_LIGAND: An Approach to De Novo Molecular Design. 3. A Genetic Algorithm for Structure Refinement, *J. Comput.-Aided. Mol. Des.* **1995**, *9*, 139–145.

[38] M. Sullivan, Taking Drug Discovery to New Heights, *Today's Chemist at Work* **1999**, January, 44–46.

[39] A. Wang, Personal communication.

[40] V. Venkatasubramanian, A. Sundaram, K. Chan, J. M. Caruthers, Computer-Aided Molecular Design Using Neural Networks and Genetic Algorithms, in J. Devillers (Ed)., *Genetic Algorithms in Molecular Modelling*, Academic Press Limited, New York, **1996**, pp. 271–302.

[41] V. Venkatasubramanian, K. Chan, J. M. Caruthers, Computer-Aided Molecular Design Using Genetic Algorithms, *Computers Chem. Eng.* **1995**, *18*, 833–844.

[42] V. Venkatasubramanian, K. Chan, J. M. Caruthers, Evolutionary Design of Molecules with Desired Properties Using a Genetic Algorithm, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 188–195.

[43] A. Sundaram, V. Venkatasubramanian, Parametric Sensitivity and Search-Space Characterization Studies of Genetic Algorithms for Computer-Aided Polymer Design, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1177–1191.

[44] R. B. Nachbar, Molecular Evolution: a Hierarchical Representation for Chemical Topology and its Automated Manipulation, in *Proceedings of the Third Annual Genetic Programming Conference*, University of Wisconsin, Madison, Wisconsin, 22–25 July, **1998**, pp. 246–253.

[45] A. Globus, J. Lawton, T. Wipke, Automatic Molecular Design Using Evolutionary Techniques, *Nanotechnology* **1999**, *10*, 290–299.

[46] R. Carhart, D. H. Smith, R. Venkataraghavan, Atom Pairs as Molecular Features in Structure-Activity Studies: Definition and Application, *J. Chem. Inf. Comput. Sci.* **1985**, *23*, 64–73.

[47] J. Devillers, C. Putavy, Designing Biodegradable Molecules from the Combined Use of a Backpropagation Neural Network and a Genetic Algorithm, in J. Devillers (Ed.), *Genetic Algorithms in Molecular Modelling*, Academic Press Limited, New York, **1996**, pp. 303–314.

[48] J. Devillers, Designing Molecules with Specific Properties from Intercommunicating Hybrid Systems, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 1061–1066.

[49] A. W. R. Payne, R. C. Glen, Molecular Recognition Using a Binary Genetic Search Algorithm, *J. Mol. Graphics* **1993**, *11*, 74–91.

[50] G. Jones, P. Willett, R. C. Glen, Genetic Algorithms for Chemical Structure Handling and Molecular Recognition, in J. Devillers (Ed.), *Genetic Algorithms in Molecular Modelling*, Academic Press Limited, New York, **1996**, pp. 211–242.

[51] G. Jones, P. Willett, R. C. Glen, A Genetic Algorithm for Flexible Molecular Overlay and Pharmacophore Elucidation, *J. Comput.-Aided Mol. Des.* **1995**, *9*, 532–549.

[52] S. Handschuh, M. Wagener, J. Gasteiger, Superposition of Three-Dimensional Chemical Structures Allowing for Conformational Flexibility by a Hybrid Method, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 220–232.

[53] T. Hurst, Flexible 3-D Searching: The Directed Tweak Technique, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 190–196.

[54] J. D. Holliday, P. Willett, Identification of Common Structural Features in Sets of Ligands Using a Genetic Algorithm, *J. Mol. Graphics Modell.* **1997**, *15*, 221–232.

[55] D. Dolata, A. L. Parrill, W. P. Walters, CLEW: The Generation of Pharmacophore Hypotheses Through Machine Learning, *SAR QSAR Environ. Res.* **1998**, *9*, 53–81.

[56] A. Leach, C. K. Prout, D. P. Dolata, Automated Conformational Analysis: Algorithms for the Efficient Construction of Low-Energy Conformations, *J. Comput.-Aided Mol. Des.* **1990**, *4*, 271–282.

[57] M. Hahn, Receptor Surface Models. 1. Definition and Construction, *J. Med. Chem.* **1995**, *38*, 2080–2090.

[58] H. Chen, J. Zhou, G. Xie, PARM: A Genetic Evolved Algorithm to Predict Bioactivity, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 243–250.

[59] D. E. Walters, R. M. Hinds. Genetically Evolved Receptor Models: A Computational Approach to Construction of Receptor Models, *J. Med. Chem.* **1994**, *37*, 2527–2536.

[60] D. E Walters, T. D. Muhammad, Genetically Evolved Receptor Models (GERM): A Procedure for the Construction of Atomic-Level Receptor Site Models in the Absence of a Receptor Crystal Structure, in *Genetic Algorithms in Molecular Modelling*, J. Devillers (Ed.), Academic Press Limited, New York, **1996**, pp. 193–210.

[61] A. Vedani, P. Zbinden, Quasi-atomistic Receptor Modeling. A Bridge Between 3-D QSAR and Receptor Fitting, *Pharm. Acta Helv.* **1998**, *73*, 11–18.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

5 Quantitative Structure-Activity Relationships

Sung-Sau So

Abbreviations

CAMD	Compter-aided molecular design
COMBINE	Comparative molecular binding energy analysis
CoMFA	Comparative molecular field analysis
EE	Exhaustive enumeration
EP	Evolutionary programming
GA	Genetic algorithm
GARGS	Genetic algorithm-based region selection
GFA	Genetic function approximation
GNN	Genetic neural network
GOLPE	Generating optimal linear partial least squares estimations
LGO	Leave group out
LOO	Leave one out
MAE	Mean absolute error
MLR	Multiple linear regression
MUSEUM	Mutation and selection uncover models
NLM	Nonlinear mapping
NN	Neural network
OLS	Ordinary least squares
QSAR	Quantitative structure-activity relationship
QSPR	Quantitative structure-property relationship
PCA	Principal component analysis
PCR	Principal component regression
PLS	Partial least squares
SA	Simulated annealing
SAR	Structure-activity relationship
SMGNN	Similarity matrix/genetic neural network
SRCC	Spearman's rank correlation coefficient

Symbols

B_1	Verloop steric parameter
L	Verloop steric parameter
r^2	Squared correlation coefficient
π	Hydrophobic substituent constant
q^2	Cross-validated squared correlation coefficient
s	Residual standard deviation
σ_m	Hammett constant

5.1 Introduction

Despite immense efforts in the pharmaceutical industry and academic laboratories, the design of novel drugs remains essentially a trial-and-error process. It is estimated that only one in 10 000 compounds investigated ever emerges as a commercial drug, and the investment required for the overall development may be on the order of $350 million over 15 years – the average time required to bring a therapeutic agent from preclinical evaluation to market launch. Thus, from a scientific point of view and also commercially, it would be desirable to develop a method that is able to reduce this rather large domain of test compounds by filtering out those which are likely to be less effective. The concept of quantitative structure-activity relationships (QSAR) was introduced with this in mind.

QSAR techniques are commonly regarded as the best approaches to computational molecular design when the macromolecular structure of the therapeutic target is not known. The major goal of QSAR is to formulate mathematical relationships between physico-chemical properties of compounds and their biological response in the system of interest. A good predictive model not only enhances our understanding of the specifics of drug action, but also provides a theoretical foundation for future lead optimization. Hansch pioneered this field by demonstrating that the biological activities of drug molecules can be correlated to a few thermodynamic or electronic variables using a simple regression equation [1]. New QSAR methodologies continue to appear in the literature, and two significant developments have been made in recent years. The first is the introduction of a wide range of novel molecular descriptors that may be more specific in the characterization of certain types of interaction than traditional descriptors. The second is the emergence of many sophisticated correlation methods for the determination of QSARs that are a significant improvement over the original linear regression analysis. Many techniques are brought into this field as a result of multidisciplinary efforts from artificial intelligence, multivariate statistics and applied mathematics researchers. In this chapter, we will review the recent advances and, in particular, the role played by genetic algorithms (GAs) in the development of these new tools.

5.2 Key Tasks in QSAR Development

This section outlines the typical steps in the development of a QSAR. The mathematical form of a QSAR is given by the following general expression:

$$\text{Biological activity} = f(x_1, x_2, \ldots, x_n) \quad (1)$$

where f is a mathematical function and x are the molecular descriptors that provide information on the physical and/or chemical attributes of the molecules. Thus, the main challenges in QSAR modeling are to find a suitable set of molecular descriptors and an appropriate function that can accurately explain the biological data.

5.2.1 Descriptor Tabulation

The first step is the tabulation of experimental or computational physico-chemical parameters that provide a molecular description of the chemical entities that is relevant to their underlying biological activities. Traditionally, the descriptors generated are usually a few thermodynamic or electronic variables. In recent years, many novel molecular descriptors, such as fragment fingerprints, molecular connectivity indices [2], vibration-based descriptors [3, 4], electrostatic and steric shape similarity indices [5], autocorrelation vectors [6], electrotopological indices [7], and the description of molecular fields in a three-dimensional (3-D) lattice environment [8] have been introduced. In general, obtaining two-dimensional (2-D) descriptors from molecular structures is a straightforward task. On the other hand, depending on the flexibility of the molecules, the generation of 3-D descriptors can be quite complex because of the uncertainty associated with both the conformation and the alignment of the molecules.

5.2.2 Feature Selection

The next step is to apply a statistical or pattern recognition method to correlate these experimental or computed molecular properties with the biological data. It is ironic that the ease in descriptor generation often creates a problem in subsequent analysis, namely the overfitting of data. This is because the number of descriptors can easily exceed the number of data objects. To circumvent this, a number of algorithms have been proposed to preselect the descriptor set. Feature selection can help to define a model that is easier to interpret. In addition, the reduced model is often more predictive, simply because a better signal-to-noise ratio is obtained when the noninformative inputs are discarded. We must emphasize that descriptor selection is not new: in the past it was performed by a human expert based on experience and scientific intuition as well as a great deal of trial and error. In this section, we will review some common selection strategies that are more objective in nature.

5.2.2.1 Forward Stepping and Backward Elimination

With the increasing availability of fast computers and commercial statistical packages, feature selection based on a stepwise procedure is commonly employed [9]. In the case of a forward stepping strategy, new descriptors are added to the model one at a time until no further improvement is obtained, as judged by a significance measure such as the *F*-statistic. In backward elimination, a full set of descriptors is used as an initial model and the redundant descriptors are pruned in a systematic fashion. While these procedures are computationally fast, they have some shortcomings. First, because of their stepwise nature, the algorithms fail to take into account information that involves any coupled effect among multiple descriptors. Specifically, it is possible that a certain descriptor is eliminated early on because it may appear to be redundant at that stage; but it may later turn out to be the most informative descriptor when other descriptors are eliminated. In other words, an optimal solution is not necessarily obtained. Second, the algorithms are deterministic in nature, and subsequently the final result is represented by a single solution. However, it has been argued that due to both the complexity and the dimensionality of the problem, it is unrealistic to assume the existence of a unique "best" solution. A better approach is to make use of a set of models in which each model may be more suitable for characterizing different regions of parameter space. A number of stochastic algorithms, such as simulated annealing and genetic algorithms, have been introduced to the field of QSAR to address the aforementioned problems.

5.2.2.2 Simulated Annealing

Simulated annealing (SA) is an optimization method based on a physical annealing process [10]. The principle of this method is similar to a Monte Carlo simulation [11], but the algorithm contains a slowly decreasing temperature parameter. At the beginning of the optimization, solutions of lesser quality are accepted more readily according to a Boltzmann-type probability, thereby allowing for the exploration of a large configurational space. As the temperature is slowly reduced, the system converges to a final solution. Because of the stochastic nature of SA, simulations that are initialized with different random seeds can lead to different outcomes. The first chemometric application of simulated annealing in descriptor selection was performed by Sutter and Kalivas [12].

5.2.2.3 Genetic Algorithms

Genetic algorithms (GAs) are stochastic optimization methods that have been inspired by evolutionary principles [13]. The distinctive aspect of a GA is that it investigates many solutions simultaneously, each of which explores different regions of parameter space. The basic design of a genetic algorithm is summarized in the flow diagram shown in Fig. 1. The first step of a GA simulation is the creation of *N* chromosomes, each of which represents a candidate solution of the problem. In the case of feature selection, an appropriate representation of a chromosome can be a numerical string encoding a particular combination of molecular descriptors. The fitness of each chromosome is evaluated based on a numerical function, which reflects the quality of each candidate solution. The next step, reproduction, creates new chromosomes from the existing generation. Through the selection operator, better chromosomes can proliferate preferentially. With crossover, each chromosome has an opportunity to exchange information with the others via a mating procedure. Finally, fitter offspring may appear in the next generation if beneficial mutations take place. As the system gathers more knowledge about the underlying parameter space, the collective search, which may appear to be random at first, begins to gain focus and moves towards more optimal regions. The reproductive cycle is repeated until a predefined number of generations, a specified convergence criterion or a target fitness score is attained.

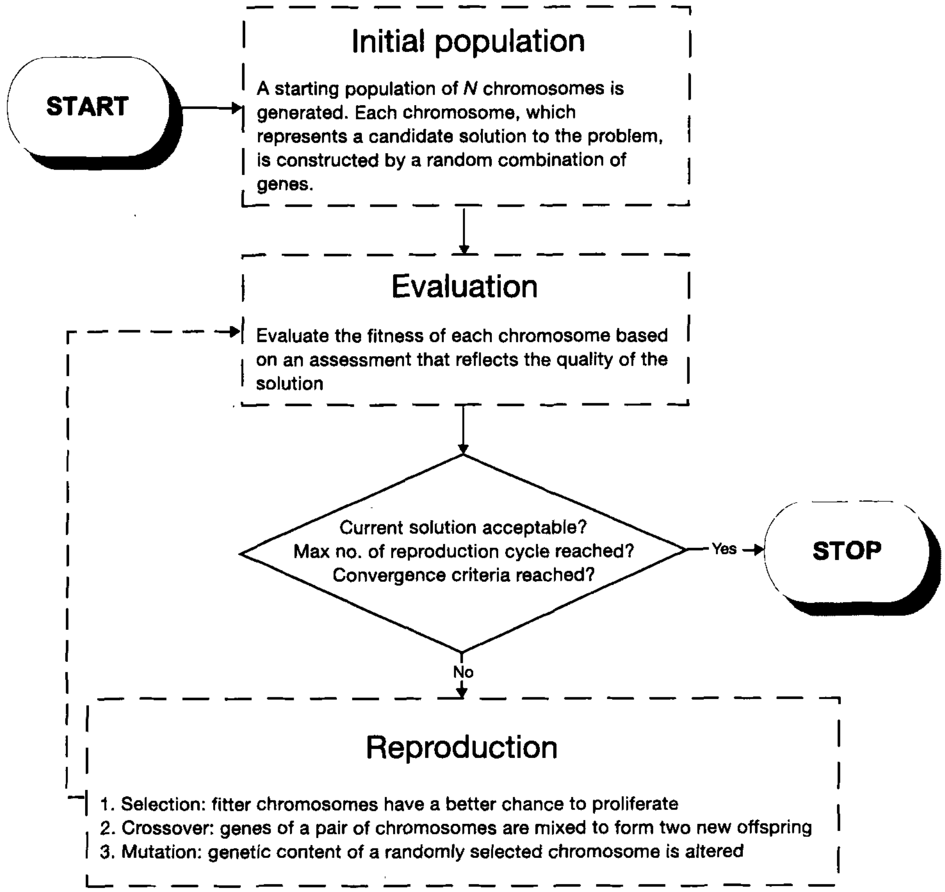

START
Initial population
A starting population of N chromosomes is generated. Each chromosome, which represents a candidate solution to the problem, is constructed by a random combination of genes.
Evaluation
Evaluate the fitness of each chromosome based on an assessment that reflects the quality of the solution
Current solution acceptable?
Max no. of reproduction cycle reached?
Convergence criteria reached?
Yes
STOP
No
Reproduction
1. Selection: fitter chromosomes have a better chance to proliferate
2. Crossover: genes of a pair of chromosomes are mixed to form two new offspring
3. Mutation: genetic content of a randomly selected chromosome is altered

Figure 1. Flow diagram showing a typical genetic algorithm simulation.

5.2.2.4 Exhaustive Enumeration

Although a GA investigates many possible solutions simultaneously, there is no guarantee that the best possible solution will be found in any run of a GA. An exhaustive enumeration (EE) of all possible sets of descriptors is the only method which guarantees to locate the global optimum, although such a brute-force approach is often impractical due to the exponential increase of the number of possible sets that can be formed from a given number of descriptors (i.e., the total number of all possible sets = $2^N - 1$, where N is the total number of descriptors). For a QSAR data set with say, 30 descriptors, this number is already greater than 10^9 and for 100 descriptors, it exceeds 10^{30}.

5.2.2.5 Other Methods

It is beyond the scope of this chapter to provide a comprehensive review of all the feature selection strategies that have appeared in the literature. In what follows, two interesting approaches are discussed briefly. GOLPE (generating optimal linear PLS estimations) is a variable selection procedure that has been developed to obtain the best predictive PLS models. In this approach, combinations of variables derived from a fractional factorial design strategy are used to run several PLS analyses, and only variables that contribute significantly to predictivity are selected [14]. Wikel and Dow have investigated how an artificial neural network (see section 5.2.3.2) can be used for descriptor selection [15]. First, a neural network is trained using all input descriptors. Subsequently, input descriptors are selected based on the weights between the various input and hidden nodes.

5.2.3 Model Construction

5.2.3.1 Linear Methods

Given a set of input descriptors and a set of output responses, one needs to formulate a mathematical expression to relate the two quantities. Multiple linear regression (MLR), or ordinary least squares (OLS), has been the method of choice for QSAR applications in the past, mainly due to its computational simplicity and in part because of the interpretability of the resulting equation. However, this method cannot be applied when the number of variables equals or exceeds the number of objects (for MLR the recommendation is that the ratio between number of objects and variables should be at least greater than five). One way to reduce the number of variables is through the use of principal component analysis (PCA). In this procedure, the input descriptors are transformed to some orthogonal principal components, where a small number of principal components is usually sufficient to capture the essential variance (~90 %) of the original data. These principal properties are then used as the input to a regression analysis. Another powerful method that can be applied to deal with an underdetermined data set is partial least squares (PLS) [16]. There is no restriction in PLS on the ratio between variables and data objects. Unlike MLR, this method can analyze several response variables simultaneously. Briefly, PLS attempts to identify a few latent structures (linear combinations of descriptors) that best correlate with the observables. Cross-validation (see section 5.2.4.1) is employed to avoid overfitting the data. An added advantage of PLS over MLR is that it can deal with strongly collinear input data and can tolerate missing data values. For an in-depth comparison of PLS with MLR, readers are referred to previous publications [14].

5.2.3.2 Nonlinear Methods

Apart from the linear analysis tools mentioned above, there is an increasing interest in the use of methods that are intrinsically nonlinear. Nonlinear mapping (NLM) is a method that attempts to preserve the original Euclidean distance matrix when high-dimensional data are projected to lower (typically two) dimensions. However, NLM does not provide a quantitative relationship between activity and structural descriptors. At the present time, artificial neural networks (NNs) are probably the most commonly used nonlinear method in chemometric applications. NNs are computer-based simulations which contain some elements that appear to exist in living nervous systems. What makes these networks powerful is their potential for performance improvement over time as they acquire knowledge about a problem, and their ability to handle fuzzy "real-world" data. Fig. 2 shows a schematic representation of a neural network. During the training phase, a network is taught data patterns through iterative adjustments of the weight values between the interconnecting nodes. A trained network is able to draw generalizations from these patterns and, more importantly, it can make quantitative predictions for novel patterns. Previous applications [17–19] of neural networks to QSAR have established that the number of adjustable weights plays a crucial role in determining the predictivity of the network. With too few adjustable weights, a network may not be able to extract the relevant relationships from the data; with too many weights, the network may tend to overfit the data because it has the capacity to memorize the entire data set. Finding an optimal network topology to achieve a balance between the two extreme situations is an active area in neural network research [20].

5.2.3.3 Other Methods

In addition to the methods mentioned above, many multivariate analysis methods have been reported in chemometric applications. These have included k-nearest neighbors (KNN), correspondence factor analysis (CFA), linear discriminant analysis (LDA), simple classification analysis (SIMCA), cluster significance analysis (CSA), and canonical correlation analysis (CCA).

5.2.4 Model Validation

Model validation is a critical component of QSAR development. A number of procedures have been established to determine the quality of QSAR models.

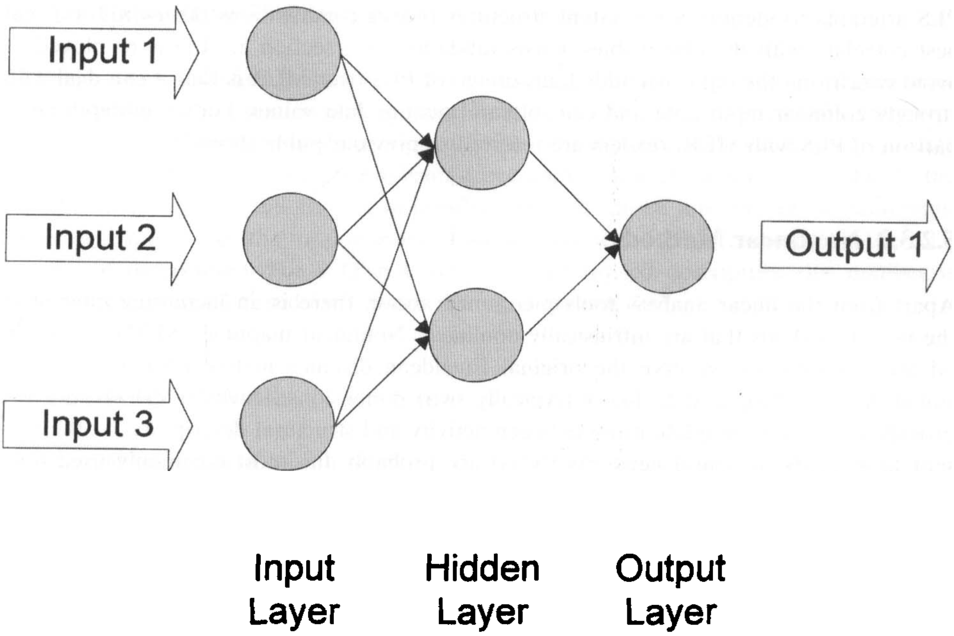


Figure 2. This network is configured with three input, two hidden, and a single output nodes (3-2-1).

5.2.4.1 Cross-Validation

The most popular validation method is cross-validation (CV), also known as jack-knifing or leave-one-out (LOO). This method systematically removes one data point at a time from the training set, and constructs a model with the reduced data set. Subsequently, the model is used to predict the data point that has been left out. By repeating the procedure for the entire data set, a complete set of predicted properties and cross-validated statistics can be obtained. It has been argued that the LOO procedure often overestimates the predictivity of the model and that, subsequently, the QSAR models are overoptimistic [21]. As an alternative to LOO, a leave-group-out (LGO) procedure, which leaves out multiple data objects (typically between 5 % to 20 % of the entire set), can be applied. An added bonus of a LGO procedure is the reduction in computing time relative to a LOO cross-validation, the amount of time saved being inversely proportional to the percentage of removed population.

5.2.4.2 External Test Set

A common use of a QSAR model is to provide activity predictions for new analogs. Thus, in a practical sense, the accuracy of prediction of a test series constitutes the most stringent test of a model. As an external test set, these compounds must not have been used in any way during the construction of the QSAR model.

5.2.4.3 Randomization Test

Another procedure that is easy to perform is a randomization test. In this method, the output values (the biological activities) of the compounds are shuffled randomly, and the resulting data set is examined by the QSAR method against real (unscrambled) input descriptors to determine the correlation and predictivity of the resulting "model". The whole procedure is repeated on many different scrambled data sets. The rationale behind this test is that the significance of the real QSAR model would be suspect if there is a strong correlation between the selected descriptors and the randomized response variables.

5.2.4.4 Measurement of Model Quality

In the field of QSAR, many statistical variables have been reported to indicate the quality of the model. Most commonly, the following measures are included. The first is the r^2 value, which is the Pearson correlation coefficient, and the s value, the residual standard deviation for the predicted values of the training set of compounds.

$$r^2 = \frac{\left\{\sum_{i=1}^{N}\left(y_{calc,i} - \overline{y_{calc}}\right)\left(y_{obs,i} - \overline{y_{obs}}\right)\right\}^2}{\left\{\left(\sum_{i=1}^{N} y_{calc,i}^2\right) - N\overline{y_{calc}}^2\right\}\left\{\left(\sum_{i=1}^{N} y_{obs,i}^2\right) - N\overline{y_{obs}}^2\right\}} \tag{2}$$

$$s = \sqrt{\frac{\sum_{i=1}^{N}\left(y_{calc,i} - y_{obs,i}\right)^2}{N-k-1}} \tag{3}$$

These measures give indications of how well the model fits existing data; that is to say, they measure the explained y variance in the biological data.

For cross-validated statistics, it has been suggested that *PRESS* and q^2 are good estimates of the real prediction error of a model:

$$PRESS = \sum_{i=1}^{N}\left(y_{pred,i} - y_{obs,i}\right)^2 \tag{4}$$

$$q^2 = 1 - \sum_{i=1}^{N}\left(y_{pred,i} - y_{obs,i}\right)^2 \Big/ \sum_{i=1}^{N}\left(y_{obs,i} - \overline{y_{obs}}\right)^2 = 1 - PRESS \Big/ \sum_{i=1}^{N}\left(y_{obs,i} - \overline{y_{obs}}\right)^2 \tag{5}$$

Generally speaking, a q^2 value of 0.5–0.6 is regarded as the minimum acceptance criterion for a reliable QSAR model. For more detailed discussions on model validation and statistical parameters, readers are referred to a previous chapter in this book series [14].

5.3 Availability of GA Programs

A number of commercial and academic programs are available for GA-based QSAR modeling. They include the GFA module that is part of the Cerius^2 molecular modeling package marketed by Molecular Simulations Inc. [22]. A GAPLS program is one of the modules in CHEMISH, an integrated chemometrics system, which is freely available upon request from the research group of Funatsu [23]. Some research groups have utilized the worldwide web as a medium for distributing tools. For example, a GAPLS web page has been installed on the QSAR server at the University of North Carolina to allow the public submission of data sets [24]. Also available is Luke's evolutionary programming (EP) algorithm, which is part of the CHEMSP2 package at the National Cancer Institute's Frederick Biomedical Supercomputer Center [25]. In addition, there are several public domain GA codes and libraries and it is straightforward to link these resources to build customized systems.

5.4 Applications of GAs in QSAR

5.4.1 GA-MLR Approach

The earliest application of GAs in chemometrics was reported by Leardi et al. in 1992 [26]. They employed a GA for variable selection and MLR for quantitative modeling. Their initial test was carried out on an artificial data set with 16 data points, 11 descriptors and a single response variable. Because of the simplicity of this data set, an exhaustive enumeration of all descriptor combinations was possible, thereby allowing a benchmark comparison of both the efficiency and completeness of the GA searches. In their study, they limited the simulation time of the GA to 25 % of the time required for a full exhaustive search. In ten separate runs, the GA run always found the best combination, which was an eight-descriptor model. Another impressive feat was that the GA discovered the ten best combinations in nine out of ten simulations initialized with different random seeds. By comparison, stepwise regression returned a five-descriptor combination that was ranked sixth overall in the list obtained by exhaustive enumeration. After this initial validation, the method was tested on a real data set, which consisted of 41 samples of provola cheese that were described by 69 chemical variables. The objective of the study was to find a meaningful correlation between the age of these samples and their chemical compositions. Using a stepwise approach, Leardi et al. obtained a 12-descriptor regression model that yielded a cross-validated variance of 83 %. The top models from five independent GA runs gave corresponding values of cross-validated variance of 89 %, 85 %, 81 %, 91 %, and 84 %. This particular example clearly demonstrates the stochastic nature of the GA optimization, where different starting conditions can lead to distinct local minima

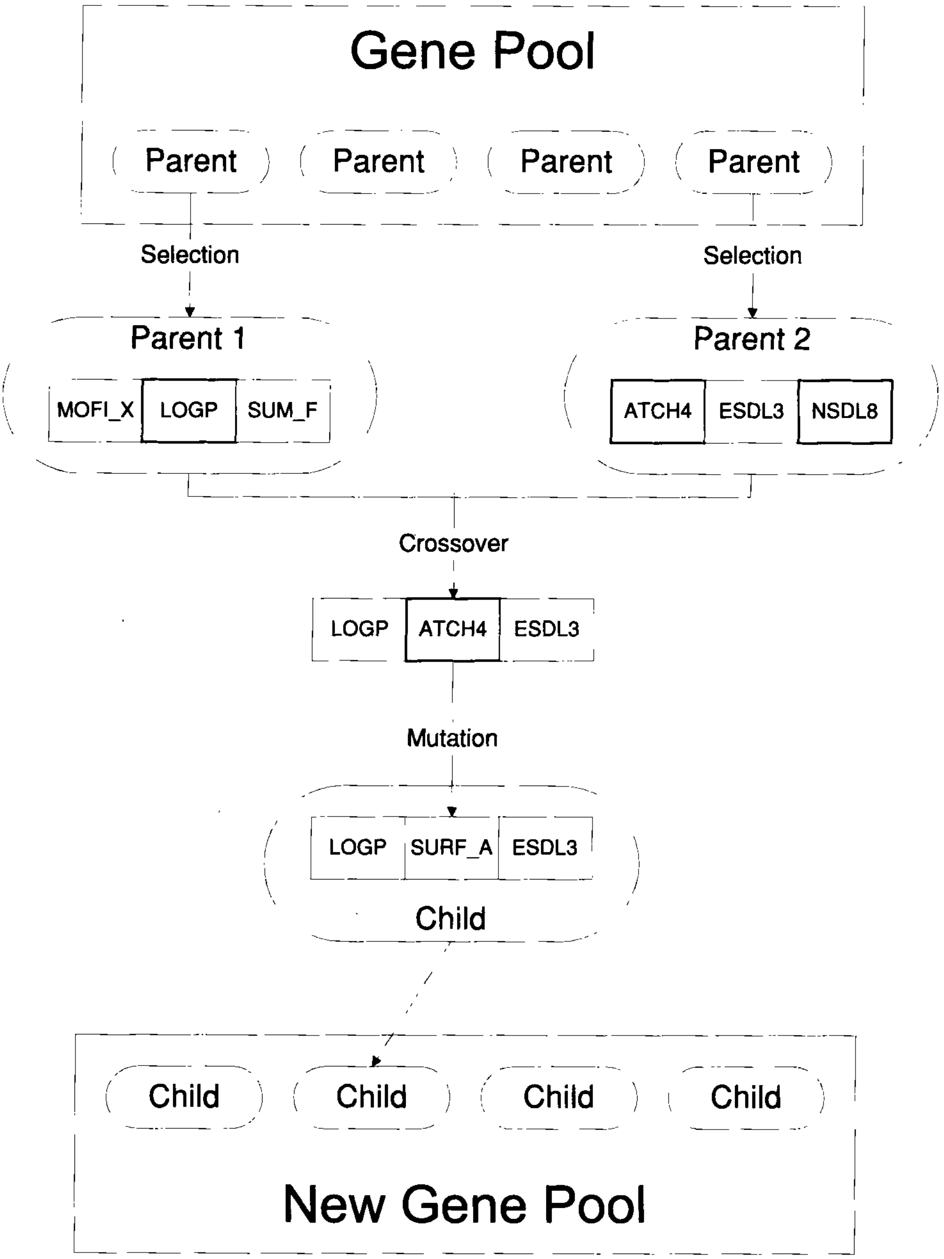


Figure 3. Schematic diagram describing the strategy in GFA.

corresponding to models of quite different qualities. Consequently, one recommendation from the authors is to perform at least two runs on the same data set when a GA is employed as a feature selection tool.

Later, Rogers and Hopfinger published a paper that described a type of GA, termed genetic function approximation (GFA), to select descriptors. The GA was coupled with a standard multiple linear regression (MLR) method to derive QSAR models [27]. GFA is a conventional GA with crossover and mutation operators, and its reproduction strategy is shown in Fig. 3. The calculation began with a population of 300 randomly chosen sets of descriptors, and it took typically 3000–10 000 genetic operations to reach convergence. GFA incorporates the lack of fit (LOF) error measure as its fitness criterion:

$$LOF = \sum_{i=1}^{N} \left(y_{calc,i} - y_{obs,i}\right)^2 \Big/ \left(1 + \frac{c + dp}{M}\right)^2 \tag{6}$$

In this equation, c is the number of variable parameters, d is a user-adjustable smoothing parameter that provides some control over the number of terms in the model, p is the total number of features contained in all basis functions (some basis functions may be combinations of different features or descriptors) and M is the number of samples in the training set. The use of this function tends to prevent the data being overfitted because while adding a new term to the QSAR may reduce the error in the fit, it may not be sufficient to offset the penalty as specified by the denominator. In addition, the d value (the default value of d is 1) allows more flexible user control over the smoothness of the fit. Rogers and Hopfinger also introduced a number of nonlinear basis functions in model construction, including splines, Gaussians, and polynomial functions. Finally, the algorithm permits the number of descriptors to be varied throughout the calculation. The algorithm was tested on the Selwood data set [28], which has been studied extensively in the past and has become a standard test of novel 2-D QSAR methods. The data set consists of a series of 31 antifilarial antimycin analogs and each compound is parameterized by 53 physico-chemical descriptors. Using the GFA method, Rogers and Hopfinger uncovered a number of QSAR models that were significantly better than the models obtained in earlier studies by Selwood et al. [28] and by Wikel and Dow [15]. In particular, there was very little overlap in the exact choice of descriptors among the three studies. LOGP, the sole descriptor encoding hydrophobicity in the set, was the only common choice. The top 20 models have cross-validated r values ranging from 0.849 to 0.812, thus supporting the notion that it is unlikely that a single QSAR model can adequately describe all of the interesting relationships within the data set. The authors also observed the interesting phenomenon that averaging the predictions of a number of top-rated models often led to better predictions than could be obtain from any individual model. In the same paper, Rogers and Hopfinger applied GFA to two additional data sets, again achieving impressive results.

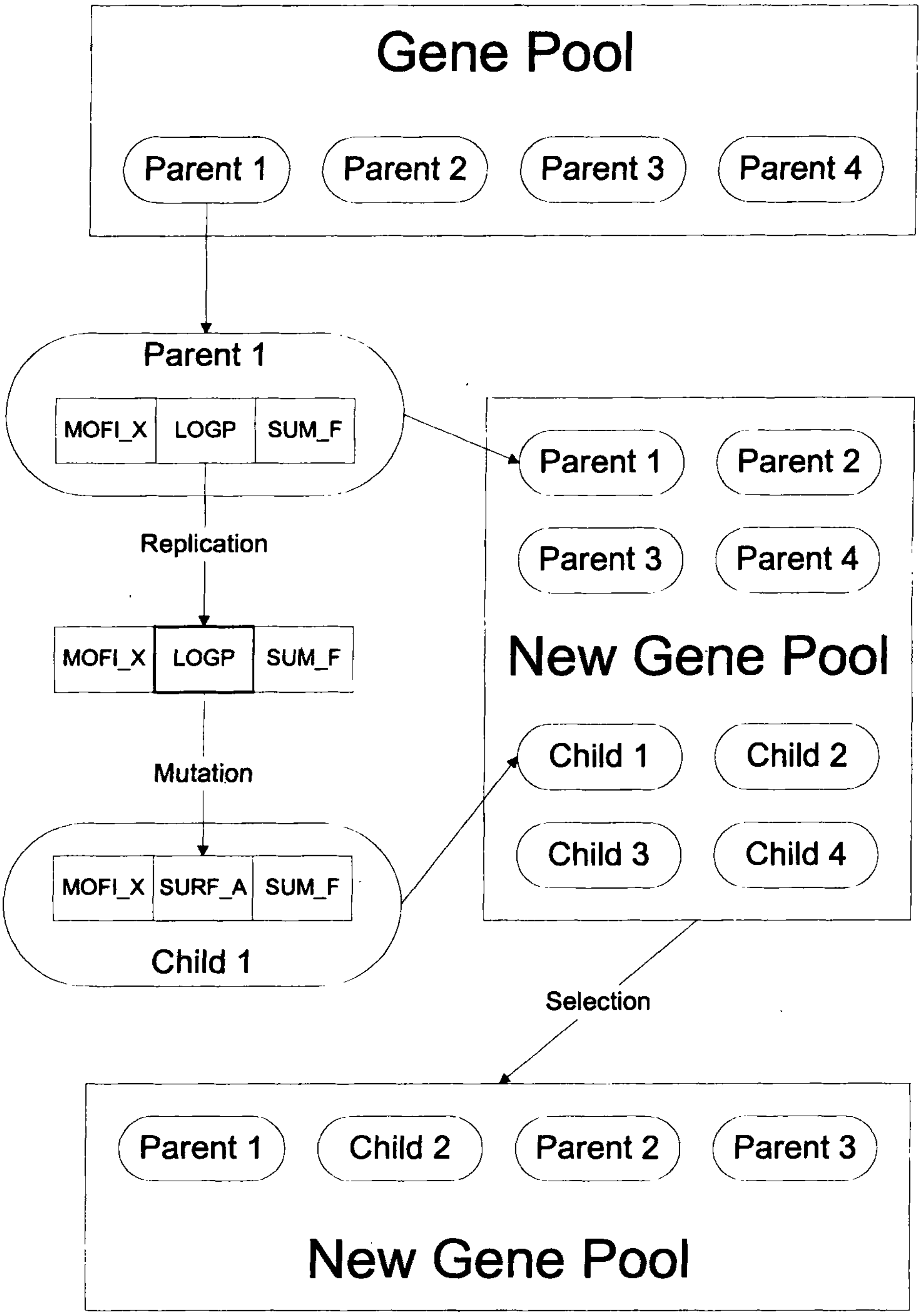


Figure 4. Schematic diagram describing the strategy in EP and MUSEUM.

At about the same time, Kubinyi of BASF and Luke of IBM published two studies, also using the Selwood data set as the benchmark. Both investigators utilized variants of GAs, termed MUSEUM (Mutation and Selection Uncover Models) [29] by Kubinyi and Evolutionary Programming (EP) [30] by Luke. Both algorithms do not contain a crossover operator and they rely solely on a mutation operator to create new offspring (Fig. 4). Independently, both researchers discovered other excellent three-descriptor combinations that were not reported in the GFA study, though it was likely that those combinations might have been visited by GFA but were later destroyed by crossover or mutation operations during evolution. In his paper, Kubinyi also suggested variable selection is an appropriate method for the size of data set that is typical in conventional QSAR studies (i.e., 20–200 descriptors). However, for CoMFA-type analysis, variable selections may carry too great a risk of chance correlation because of the large number of input variables. In a more recent study [31], Luke extended his earlier work and compared the accuracy and efficiency of three different selection strategies, which included stepwise selection, EP, and EE of all possible sets. The basic conclusion from the study is that although EP and EE can reveal the optimal combinations of descriptors, the computational efficiency of the stepwise approach allows for the exploration of different functional forms of descriptors (in his paper, both logarithmic and exponential transformations were examined) that may lead to better results. In this study, Luke introduced an alternative cost function for the fitness criterion:

$$Cost = Base^{|k-n|} \times (Error\,of\,fit) \times \prod Weight(N_i) \qquad (7)$$

where k is the desirable number of descriptors as specified by the user, n is the actual number of descriptors used in the QSAR model, and *Base* is a parameter that is similar to the smoothing parameter (d) in GFA [32]. *Error of fit* is defined by the RMS error, although any other reasonable measure can be used. The third term is a product of the weights associated with each function, where the weights are related to the exponent to which each descriptor is raised. In Luke's implementation, the weight of linear term is one, that of the quadratic term is two, and all other weights are set to 100. The *Base* value turns out to be a very important parameter in the evolution of the models. In this study, Luke demonstrated that by automatically adjusting the value of *Base*, a much more thorough search in parameter space could be achieved.

Leardi extended his earlier GA/MLR work and proposed a general method for model validation and outlier detection [33]. For full validation, Leardi argued that the initial choice of selected variables should be independent of the data points that were used for its validation. He achieved this by partitioning the full data set into k deletion groups, $\{T_1, V_1\}$, $\{T_2, V_2\}$ ... $\{T_k, V_k\}$. For each deletion group, GA-based descriptor selection was applied to the larger data partition, T_i, and several models were formulated based on the statistical parameter (e.g., r^2 or q^2) of this training set only. The remaining data, V_i, served as an external validation set for freshly generated models. Only those combinations exceeding a minimum predictivity threshold would undergo a formal "leave-group-out" validation. Different correlation models, which were re-derived using the same combination of descriptors on different training sets in each of the deletion group, were used to predict the other validation sets. The final predictivity assessment of a given combination would be based on the result of full validation. Fig. 5 shows a schematic representation of this process. In this example, the full data set is split into four deletion groups. Based on the training set T_1, three good combinations of descriptors, $M_{1,1}$, $M_{1,2}$ and $M_{1,3}$ were revealed. The portion of data that was left out in the model building, V_1, was used to validate each combination. In this example, because the predictions of V_1 using combination $M_{1,3}$ are sufficiently predictive, this combination is carried forward to perform a full validation. In this stage, the same descriptor combination $M_{1,3}$ is applied to other deletion groups, T_{2-4}. Since the compositions of these training sets are somewhat different from that of T_1, the scalar parameters (i.e., regression coefficients in case of MLR) describing the model should be slightly different. The re-derived models are used to provide predictions for the other validation sets V_{2-4}, and together with V_1, they form a complete set of prediction results for the combination $M_{1,3}$. The process is repeated for other good combinations from the other deletion groups (e.g., $M_{2,1}$, $M_{2,2}$, etc). The combination that yields the best full-validation result will be considered as the optimal choice of descriptors. An extremely useful application of the above procedure is the detection of outliers. Because an outlier creates significant heterogeneity in the data set, when an outlier is partitioned to a validation subset V_i, the corresponding training set T_i becomes more homogeneous. Thus, the presence of an outlier in V_i is characterized by an anomalously high quality of fit in the corresponding training set T_i and an unusual lack of fit in V_i. Using this method, Leardi successfully identified an outlier in the data set where classical approaches had failed [33].

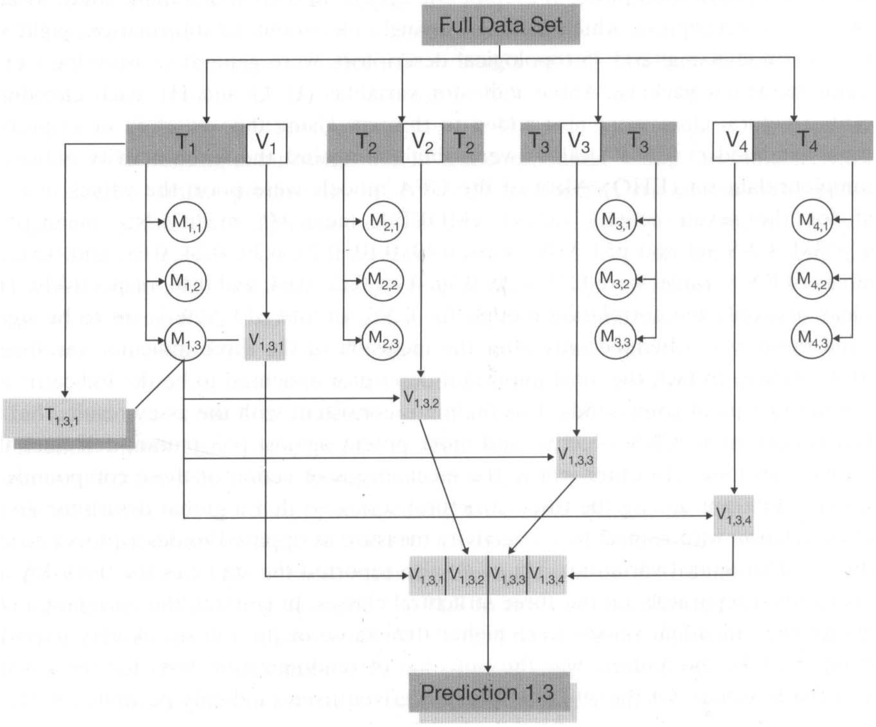


Figure 5. The validation procedure described by Leardi.

Partly due to its commercial availability and its integration into the Cerius^2 modeling environment, Rogers' GFA method has become a popular GA-based algorithm for chemometric applications. In a recent study, Shi et al. of the National Cancer Institute (NCI) applied the GFA program to examine the antitumor activity patterns of a series of ellipticine analogs [34]. These 112 compounds can be classified into three structural subclasses (E, Q, and H types). From the *in vitro* assay results against a panel of 60 human cancer cell lines from different organs of origin, seven representative activity indices were derived:

- MOLT-4, the activity against leukemia cell line MOLT-4;
- mean_60, the mean activity against all 60 cell lines;
- mean_CNS, the mean activity against six CNS cell lines;
- mean_p53W, the mean activity against 19 p53 wild-type cell lines;
- mean_p53M, the mean activity against 41 p53 mutant cell lines;
- CNS_sel (= mean_CNS – mean_60), the CNS selectivity;
- p53_MW (= mean_p53M – mean_p53W), the "p53-inverse" selectivity.

Since the majority of the standard clinical anticancer agents are more potent against p53 wild-type cells than against p53 mutant ones (and the fact that the *p53* gene is mutated in more than half of human tumors), the p53-inverse selectivity is an important parameter for the future optimization of novel therapeutic agents. In their preliminary study, 49 standard molecular descriptors, which included a single electronic, 13 information, eight spatial, two thermodynamic and 25 topological descriptors, were generated using the Cerius^2 molecular modeling package. Three indicator variables (E, Q and H), each encoding a different structural class, were also added to this set. Using this set of 52 descriptors, r^2 and cross-validated r^2 (CV-r^2) values were obtained against the seven activity indices for the complete data set (EHQ). Most of the GFA models were poor: the values of r^2 obtained for the seven activity indices (MOLT-4, mean_60, mean_CNS, mean_p53W, mean_p53M, CNS_sel and p53_MW) were 0.40, 0.10, 0.27, 0.29, 0.24, 0.68, and 0.51, and the values of CV-r^2 range were 0.34, 0.26, 0.30, 0.23, 0.22, 0.64, and 0.46, respectively. Thus, it is clear that only the correlation models for CNS_sel and p53_MW seem to be significant, and these were obtained only after the inclusion of the three indicator variables in the GFA analysis. In fact, the most important descriptor appeared to be the indicator variable for the Q-type of compounds. This finding is consistent with the assay results that this subclass is both more CNS selective and more potent against p53 mutant cell lines than the E or H subclasses. In other words, the mechanisms of action of these compounds are likely to be different among the three structural series, so that a global descriptor gives a higher correlation with respect to a selectivity measure as opposed to descriptors encoding specific, local structural variations. Shi et al. also reported the statistics for the GFA analyses performed separately on the three structural classes. In general, the correlation coefficients for the individual classes were higher than those of the full set. A very important result reported by the authors was the outcome of randomization tests for the CNS_sel index of the H subset. Of the 19 GFA models derived from randomly permuted data, two of them actually yielded higher r^2 values (>0.90) than that of the "real" model, and ten had r^2 values greater than 0.60! So, the apparently high r^2 value of this particular GFA model with real data is probably due to spurious correlation. The result of this study illustrates the importance of model validation in QSAR applications.

5.4.2 GA–PLS

PLS is often regarded as a modern alternative to MLR for chemometric applications, and it has played a critical role in the derivation of QSARs in CoMFA studies [35]. In the past, few researchers paid special attention to feature selection in PLS applications, partly because PLS makes no restriction on the number of variables used and partly because of its high tolerance towards noisy data [36]. It is only recently that this attitude has changed as more people recognize the benefits of feature selection. The use of procedures such as GOLPE [14] and GAs in conjugation with PLS is now increasingly common.

In the past two years, Funatsu and co-workers have published a series of papers [9, 23, 37–39] describing the application of the GAPLS method in QSAR. In their first study on this subject [37], they investigated a series of 35 dihydropyridine derivatives acting as calcium channel antagonists with pIC_{50} values spanning the range between 4.0 and 8.9. In these compounds, there are three variable positions (R_2, R_3, and R_4), and the substitution pattern of each position is described by four descriptors: π, the hydrophobic substituent constant, σ_m, the Hammett σ constant encoding electronic properties, and B_1 and L, the Verloop steric parameters. It was anticipated that with a GA, the redundant parameters would be discarded and only the most important properties would enter the PLS analysis. Their results show that the q^2 value ($q^2 = 0.685$) of the GAPLS model based on six descriptors is superior to the full 12-descriptor PLS model ($q^2 = 0.623$). In addition, they performed an external validation employing the D-optimal criterion to partition the full data set into 21 training compounds and 14 test compounds. Using the same six variables, they obtained a similar predictivity ($q^2 = 0.693$) for the test compounds. The process of descriptor selection also led to the resulting model being more interpretable. The selected features were in accordance with the earlier work of Gaudio et al., who performed extensive QSAR analyses on the same set of compounds [40]. In their second QSAR study [9], Hasegawa et al. applied GAPLS to a data set of 57 benzodiazepines in which each compound was parameterized using 42 physico-chemical descriptors. Two GAPLS models were reported. The first model, which was derived from 13 descriptors, yielded a q^2 value of 0.836; and the second, a 10-descriptor model, yielded a value of 0.835. Both were significantly higher than the q^2 value (0.711) derived from a PLS analysis using all 42 descriptors. Furthermore, when this data set was partitioned into a training set of 42 compounds and a test set of 15 compounds using D-optimal design, the test set r^2 values for the two GAPLS models were 0.702 and 0.737, which again compared favorably to that of the full PLS model (0.593). Most recently, Hasegawa et al. [23] examined a set of 48 HIV-1 protease inhibitors by applying GAPLS to the variables derived from comparative molecular binding energy analysis (COMBINE) [41]. Several improved GAPLS models with significantly better q^2 values than the original study were formulated. In summary, the results of the above studies and a recent study by Leardi and González [36] provide a clear demonstration that a PLS model with an appropriate selection of descriptors can be significantly more predictive than the one using all the variables.

In a recent publication, Kimura et al. described the use of a genetic algorithm for region selection in CoMFA, termed GA-based region selection (GARGS) [39]. This work was an extension to the cross-validated R^2-guided region selection (q^2-GRS) approach advocated by Cho and Tropsha [42]. In the original study, Cho and Tropsha first subdivided the full rectangular lattice into a number of smaller boxes. By performing independent CoMFA calculations on these boxes, they identified the regions of subspace that yielded reasonably predictive "partial" models (as judged by a minimum q^2 threshold). Using only the regions defined by the boxes that were characterized as information rich, a q^2-GRS CoMFA model was generated. Compared to the conventional CoMFA approach, the q^2-GRS procedure produced 3-D QSAR models that were less sensitive to molecular orientation. In addition, these models were apparently more predictive as indicated by a significant increase in the value of q^2. Extending this idea, Kimura et al. performed a GA-based procedure that sought the best combinations of subregions. It was noteworthy that they did not attempt to perform GAPLS analysis on individual CoMFA field descriptors for the following reasons: (i) it is unlikely that an important structural change can be encoded by a single field value, rather, a group of spatially contiguous field variables must be considered; (ii) in a GA, the number of distinct chromosomes (combinations) increases exponentially with the number of genes (descriptors); and (iii) variable selection on a large number of descriptors can inadvertently increase the likelihood of spurious correlation. As an initial example, they studied a set of 20 polychlorinated dibenzofurans with activities against the aromatic hydrocarbon receptor. Using the GARGS procedure, they reduced the number of field variables entering the PLS analysis from 1275 to 43, and increased the internal predictivity of the model from 0.88 to 0.95. In addition, they performed a validation study by splitting the data set into two sets. Based on the coefficients determined from the training set of 15 compounds using the previously determined set of variables, they were able to generate some impressive predictions for the five test set compounds. However, in our opinion this cannot be regarded as a true "external" validation because the test compounds have already introduced a degree of bias into variable (or in this case, region) selection. For a true external validation, we believe that the variable selection process must be performed independently of the test compounds. Nevertheless, the results from this and another application on a set of acetylcholinesterase inhibitors [38] further strengthen the value of region selection in 3-D QSAR analysis.

5.4.3 GA-NN

A major advantage of using neural networks (NNs) rather than linear regression methods (e.g., MLR and PLS) in correlation studies is their ability to handle nonlinear relationships. In a series of four papers [43–47], So and Karplus investigated a new hybrid method (GNN) that combines a genetic algorithm for descriptor selection and an artificial NN for model building. In their first application [43] of GNN on the extensively studied Selwood data set, QSAR models with fitting and predictivity that exceeded all previous published models were obtained. The improvement appears to derive from the selection of nonlinear descriptors because the NN was able to unravel complex relationships using these properties. A major drawback of this approach was the computational expense compared to other methods based on linear regression. However, the intrinsic capability of a NN to handle both linear and nonlinear descriptors partly offset this cost because the need to examine separately each possible nonlinearity can be avoided [48]. So and Karplus also investigated the efficiency of searching using the GFA and EP schemes, and found that the optimal combination was discovered after sampling approximately 3500 combinations. Relative to an exhaustive search, this performance translated to a six-fold speed-up. In their next paper [44], the core GNN simulator was refined by replacing the problematic steepest descent training algorithm with a more efficient training algorithm based on a conjugate gradient method. This led to a significant improvement in the speed of convergence as well as the stability of the solutions. This new simulator was tested on a set of 57 benzodiazepines, which was also examined by Hasegawa et al. in their GAPLS work [9]. So and Karplus obtained a six-descriptor GNN model that yielded a q^2 value of 0.867, which was slightly better than the top-ranking 13-descriptor (q^2 = 0.836) and the 10-descriptor (q^2 = 0.835) GAPLS models reported by Hasegawa et al. and a 10-descriptor NN model derived by backward elimination by Maddalena and Johnston [49].

After enjoying noticeable success with the generation of 2-D QSAR models, So and Karplus investigated how the GNN method could be used to analyze molecular similarity matrices to obtain predictive 3-D QSAR models [45, 46]. Molecular similarity-based descriptors are different from conventional parameters because they provide a numerical measure of resemblance between a pair of molecules based on their spatial or electrostatic attributes, rather than any specific physico-chemical property. A schematic description of the similarity matrix-genetic neural network (SMGNN) approach is shown in Fig. 6. After an initial validation of the methodology on the extensively studied corticosteroid-binding globulin (CBG) steroid data set, SMGNN was applied to eight data sets with a broad range of chemistries. In each case, GNN was able to derive impressive correlation models relating molecular similarity to various biological and physico-chemical measures. Most recently, both the conventional GNN (which operated on standard 2-D and 3-D descriptors) and the SMGNN methods were applied to a set of glycogen phosphorylase inhibitors [47]. Both methods yielded good predictive QSAR models (q^2 = 0.80 and 0.82 for conventional and SM-based GNN). Besides standard structure-activity correlation studies, GNN has also been employed to predict the folding ability of model proteins [50]. Using this approach, the key parameters that determined the kinetic properties of a model system were identified.

Another active research group in the development and application of hybrid systems in QSPR and QSAR is led by Jurs at Pennsylvania State University. Over the years, they have investigated QSPR/QSAR models for a wide range of physical or biological properties based on molecular structure [51–61]. These properties have included aqueous solubility, boiling point, critical temperature, autoignition temperature, toxicity and human intestinal absorption. The last two properties are of special interest to the pharmaceutical industry because a good predictive model can be a cost-saving alternative to *in vivo* animal studies. Using a combined GA–NN approach, Wessel et al. built a correlation model to estimate percent human intestinal absorption (% HIA) from molecular structure [60]. The starting point was a data set of 86 compounds with measured % HIA from the literature. These data were divided into three groups: a training set of 67 compounds; a valida-

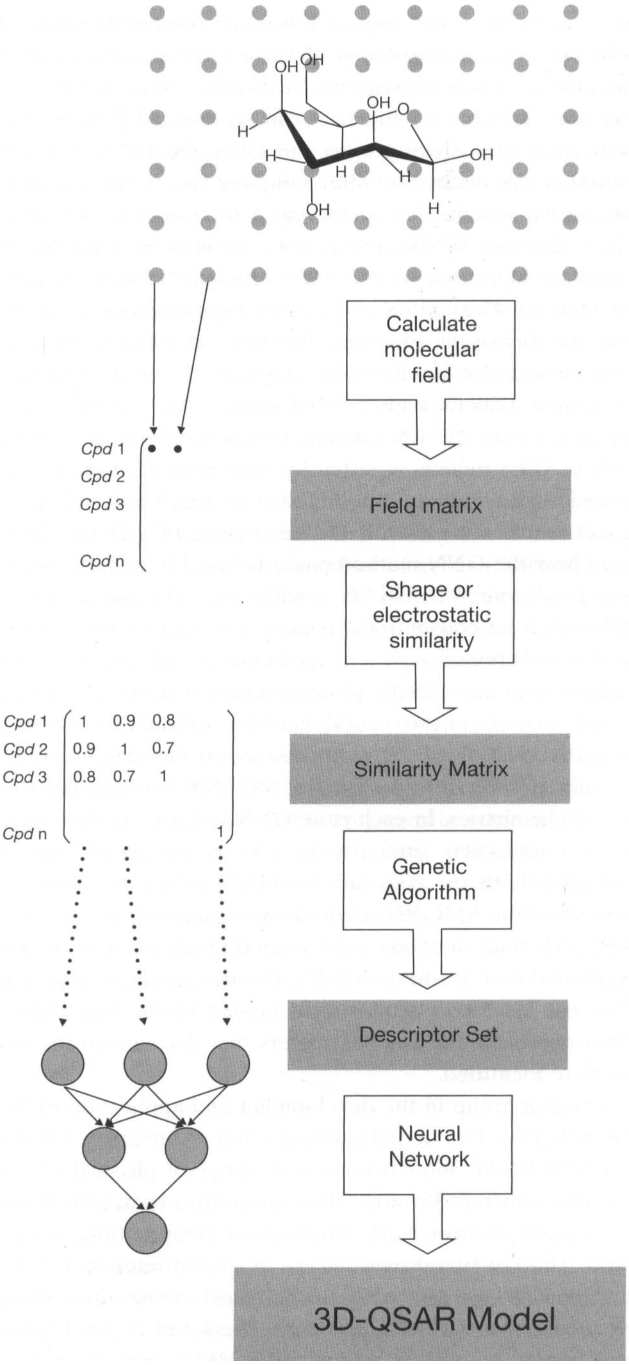


Figure 6. Schematic diagram for the construction of SMGNN QSAR model.

tion set of nine compounds; and an external prediction set of 10 compounds. 3-D structures of these compounds were generated using CORINA [62]. A total of 162 topological, electronic and geometric descriptors were generated for each of the compounds using their in-house ADAPT program. This set of descriptors was augmented by 566 binary descriptors that encoded the presence or absence of certain important substructural fragments. Based on a minimum variance criterion and a correlation analysis, the initial set of 728 descriptors was reduced to a smaller pool of 127. A preliminary analysis was made using regression in conjugation with feature selection based on simulated annealing or a genetic algorithm; however, this procedure did not lead to a satisfactory linear model. It was concluded that the linear methods were unable to take advantage of the data that appeared to be nonlinear in this case. Thus, a GA-NN hybrid system was tested on this data set and a six-descriptor NN model was identified. The mean absolute error (MAE) for the training set was 6.7 % HIA units, that for the validation set was 15.4 % HIA units, and for the external prediction set, 11.0 % HIA units. Three of the six descriptors are related to hydrogen bonding capability, which reflects the lipophilic and lipophobic characteristics of the molecule. The fourth descriptor encodes the number of single bonds, which gives an indication of structural flexibility. The final two descriptors are geometric properties that provide a description of molecular size. One can make reasonable arguments as to why the above set of molecular descriptors can be related to the mechanism of intestinal absorption. As a first attempt to tackle this complex problem, the quality of predictions from validation and external sets are very encouraging. Continuous refinement can be made with addition of reliable experimental data as well as more informative descriptors. We anticipate this kind of tool will play an important role for virtual screening in the future.

5.4.4 Chance Correlation

The examples shown in the previous sections have demonstrated the utility of GAs in descriptor selection, and discussed their role in chemometrics when used in combination with correlation methods such as MLR, PLS, and NNs. The major pitfall of using a GA-based (or any other) variable selection method is that, despite careful planning and extensive validation, chance correlation can still be a concern [63]. Not surprisingly, the critical factor determining the likelihood of chance correlation is again the ratio between the number of descriptors and the number of objects. In a recent study [36], Leardi and González attempted to identify this critical ratio using a series of randomization tests on many data sets. According to their study, a variables/objects ratio of five was suggested to be the critical point above which using a GA selection will produce statistical models that are less reliable. Obviously, this empirical ratio is a good general guideline, although other specific dependencies such as the ratio of signal-to-noise intrinsic to the data (particularly the response variables) should also be considered. In this regard, randomization tests should be performed in combination with any application involving descriptor selections. In addition, Leardi and González have proposed a stop criterion for GA evolution that may prevent overfitting of the data. This criterion is based on the empirical observation that, in general, the fitness of the population increases rapidly in the early phase of evolution, then the improvement becomes much slower in the later stages. It is argued that the early improvement is derived from the modeling of the information component in the data and, at the late stage, a GA begins to "refine" the model by adding the noise component. Obviously, it would be desirable to stop a GA run before any substantial noise is included. A way to determine an optimal stopping point is to divide a series of GA runs into two sets. In the first set, real response data are used, and in the second, data with scrambled response value are examined. Subsequently, one can plot the fitness (perhaps of the top models) as a function of GA generation, and the best moment to stop a GA run would correspond to the time when the difference in fitness between the real and the random lines is at a maximum (Fig. 7). To further reduce the chance of including redundant variables, Leardi and González also introduce a hybridization operator that perturbs the population periodically. This operator makes use of a backward elimination procedure to examine the best chromosomes. This idea originated from a study by Jouan-Rimbaud et al. [63], who have demonstrated that forward selection in the subsets selected by a GA can greatly reduce the number of irrelevant variables.

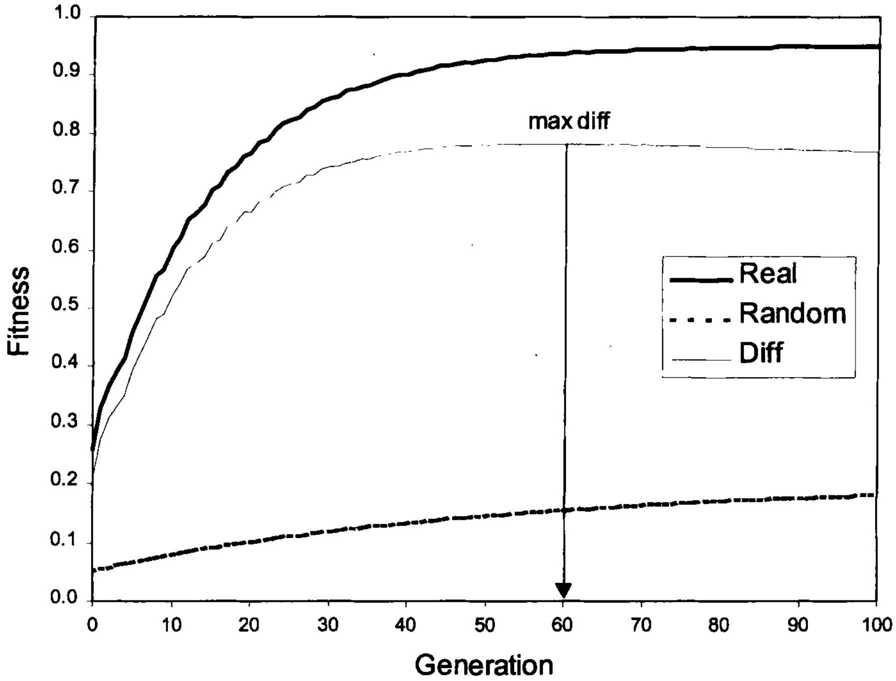


Figure 7. A stop criterion for genetic algorithm evolution.

Another approach that appears to limit chance correlation is the concept of basis models proposed by Rogers [32]. This relies on a GA (in his example, GFA was used) to generate multiples of N statistical models, then PCA is applied to the data matrix containing the prediction errors of these models. Only the most significant components that explain at least $1/N$ of the error variance are retained. The basis models are selected from the original pool of N models whose prediction errors are best correlated to each of the retained components. The predictions from the selected basis models are combined to provide consensus scoring. Using a previously analyzed socio-economic data set [64], Rogers demonstrated that the use of basis models can offer performance superior to that of the single best GA model in the following ways: first, the basis prediction ($\mathrm{SRCC}_{\mathrm{basis}}$ = 0.699) is better than the mean prediction for the best GA models from 20 runs ($\mathrm{SRCC}_{\mathrm{best\,GA}}$ = 0.591); second, the prediction variance of the basis model ($\mathrm{variance}_{\mathrm{basis}}$ = 0.081) is lower than the prediction variance for the best GA models from 20 runs ($\mathrm{variance}_{\mathrm{best\ GA}}$ = 0.123); and finally, there is a positive correlation between the magnitude of prediction error and the prediction variance, so that the latter can be a rudimentary estimate for the accuracy of future predictions.

5.5 Discussion

The recent developments in a number of hybrid QSAR tools involving the combination of a GA with regression or pattern recognition have been reviewed in this chapter. A GA offers an objective and effective means for the selection of important QSAR descriptors. The removal of redundant descriptors can lead to significant increases in both predictivity and interpretability of a model, which are critical factors that determine its utility in molecular design. The major liability of this type of approach is the risk of chance correlation. Thus, when variable selection is employed in QSAR generation, a strong emphasis should be made on model validation through randomization tests or external test set predictions. Another aspect of the work that warrants additional research effort is the use of multiple models derived from a GA simulation. Often, it is not easy to differentiate between individual QSARs because a particular model may be more applicable under certain situations. For this reason, consensus scoring based on multiple QSARs is likely to provide more reliable predictions, though it is not always obvious which models should be picked and how they can be combined. In this regard, the use of basis models proposed by Rogers [32] is a promising direction that should be tested in a more general context in future. Altogether, we believe that variable selection methods using GAs, when correctly applied, can help to define a robust and predictive model that can dramatically reduce the number of chemical synthesis–bioassay cycles during lead optimization.

In the conclusion of a recent review article on neural networks [20], Manallack and Livingstone wrote:

> "We feel that the combination of GAs and neural networks is the future for the [QSAR] method, which may also mean that these methods are not limited to simple structure–activity relationships, but can extend to database searching, pharmacokinetic prediction, toxicity prediction, etc. Neural networks have lived up to the promise of

improving the way we deal with data over conventional methods, however, the need for better molecular descriptors, network interpretability and accessibility to commercial packages encompassing the latest methods still need to be addressed".

It is fair to point out that new applications that utilize GA-based tools are now beginning to appear. In the area of combinatorial library design, Cho et al. reported the use of a GAPLS QSAR model to bias the design of bradykinin-potentiating peptide sequences [65]. A similar idea was explored in database search, where a preconstructed SMGNN model has been suggested to probe 3-D molecular databases [47]. With both strategies, an efficient method for descriptor generation is crucial because a massive virtual library of compounds is being accessed during the search. The commercial availability of GA programs also leads to exciting developments in other areas of research, such as experimental design. Using the GFA module in Cerius^2, Kowar successfully regenerated a number of complex relationships using fewer experiments compared to a more conventional factorial design strategy [66]. In the area of structure-property correlation, one of the most challenging problems remains the prediction of absorption, toxicity, and pharmacokinetic parameters, which has seen only limited success in the past. We anticipate that, with more optimal molecular descriptors, better access to high-quality biological databases and continuous refinement of QSAR methodologies, significant improvements in prediction accuracy will be achieved. In this regard, the work by Wessel et al. [60] has given us the first glimpse of the future. We await, with great eagerness and excitement, the time when *in silico* predictions will play a dominant role in lead development.

References

[1] C. Hansch, T. Fujita, ϱ–σ–π analysis. A Method for the Correlation of Biological Activity and Chemical Structure, *J. Am. Chem. Soc.* **1964**, *86*, 1616–1626.

[2] P. Dang, A. K. Madan, Structure-activity Study on Anticonvulsant (thio) Hydantoins using Molecular Connectivity Indices, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1162–1166.

[3] D. B. Turner, P. Willett, A. M. Ferguson, T. W. Heritage, Evaluation of a Novel Infrared Range Vibration-based Descriptor (EVA) for QSAR Studies. 1. General Application, *J. Comput.-Aided Mol. Des.* **1997**, *11*, 409–422.

[4] D. B. Turner, P. Willett, A. M. Ferguson, T. W. Heritage, Evaluation of a Novel Molecular Vibration-based Descriptor (EVA) for QSAR Studies. 2. Model Validation Using a Benchmark Steroid Dataset. *J. Comput.-Aided Mol. Des.* **1999**, 13, 271–296.

[5] W. G. Richards, Molecular Similarity and Dissimilarity, in A. Pullman, J. Jortner, B. Pullman (Eds.), *Modelling of Biomolecular Structures and Mechanisms*, Kluwer Academic Publishers, Dordrecht, The Netherlands, **1995**, pp. 365–369.

[6] M. Wagener, J. Sadowski, J. Gasteiger, Autocorrelation of Molecular Surface Properties for Modeling Corticosteroid Binding Globulin and Cytosolic *Ah* Receptor Activity by Neural Networks, *J. Am. Chem. Soc* **1995**, *117*, 7769–7775.

[7] L. H. Hall, L. B. Kier, Electrotopological State Indices for Atom Types: a Novel Combination of Electronic, Topological, and Valence Shell Information, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 1039–1045.

[8] R. D. Cramer, III, D. E. Patterson, J. D. Bunce, Comparative Molecular Field Analysis (CoMFA). 1. Effect of Shape on Binding of Steroids to Carrier Proteins, *J. Am. Chem. Soc.* **1988**, *110*, 5959–5967.

[9] K. Hasegawa, K. Funatsu, GA Strategy for Variable Selection in QSAR Studies: GAPLS and D-optimal Designs for Predictive QSAR Model, *J. Mol. Struct.* (*THEOCHEM*) **1998**, *425*, 255–262.

[10] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by Simulated Annealing, *Science* **1983**, *220*, 671–680.

[11] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, Equation of State Calculations by Fast Computing Machines, *J. Chem. Phys.* **1953**, *21*, 1087–1092.

[12] J. M. Sutter, J. H. Kalivas, Comparison of Forward Selection, Backward Elimination, and Generalized Simulated Annealing for Variable Selection, *Microchem. J.* **1993**, *47*, 60–66.

[13] J. H. Holland, *Adaption in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, **1975.**

[14] S. Clementi, S. Wold, How to Choose the Proper Statistical Method, in H. van de Waterbeemd (Ed.), *Chemometric Methods in Molecular Design*, VCH Publishers, Inc., New York, NY, **1995**, pp. 319–338.

[15] J. H. Wikel, E. R. Dow, The Use of Neural Networks for Variable Selection in QSAR, *Bioorg. Med. Chem. Lett.* **1993**, *3*, 645–651.

[16] S. Wold, L. Eriksson, Statistical Validation of QSAR Results, in H. van de Waterbeemd (Ed.), *Chemometric Methods in Molecular Design*, VCH Publishers, Inc., New York, **1995**, pp. 309–318.

[17] T. A. Andrea, H. Kalayeh, Applications of Neural Networks in Quantitative Structure-Activity Relationships of Dihydrofolate Reductase Inhibitors, *J. Med. Chem.* **1991**, *34*, 2824–2836.

[18] S.-S. So, W. G. Richards, Application of Neural Networks: Quantitative Structure-Activity Relationships of the Derivatives of 2,4-diamino-5-(substituted-benzyl)pyrimidines as DHFR Inhibitors, *J. Med. Chem.* **1992**, *35*, 3201–3207.

[19] D. J. Livingstone, D. T. Manallack, Statistics Using Neural Networks: Chance Effects, *J. Med. Chem.* **1993**, *36*, 1295–1297.

[20] D. T. Manallack, D. J. Livingstone, Neural Networks in Drug Discovery: Have They Lived up to their Promise? *Eur. J. Med. Chem.* **1999**, *34*, 195–208.

[21] J. Shao, Linear-model Selection by Cross-validation, *J. Am. Stat. Assoc.* **1993**, *88*, 486–494.

[22] Cerius2, Version 4.0, Molecular Simulations Inc, San Diego, CA.

[23] K. Hasegawa, T. Kimura, K. Funatsu, GA Strategy for Variable Selection in QSAR studies: Enhancement of Comparative Molecular Binding Energy Analysis by GA-based PLS Method, *Quant. Struct.-Act. Relat.* **1999**, *18*, 262–272.

[24] http://mmlin1.pha.unc.edu/~jin/QSAR/

[25] http://fconyx.ncifcrf.gov/lukeb/qsarsp2.html

[26] R. Leardi, R. Boggia, M. Terrile, Genetic Algorithms as a Strategy for Feature Selection, *J. Chemom.* **1992**, *6*, 267–281.

[27] D. R. Rogers, A. J. Hopfinger, Application of Genetic Function Approximation to Quantitative Structure-Activity Relationships and Quantitative Structure-Property Relationships, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 854–866.

[28] D. L. Selwood, D. J. Livingstone, J. C. Comley, A. B. O'Dowd, A. T. Hudson, P. Jackson, K. S. Jandu, V. S. Rose, J. N. Stables, Structure-Activity Relationships of Antifilarial Antimycin Analogues: a Multivariate Pattern Recognition Study, *J. Med. Chem.* **1990**, *33*, 136–142.

[29] H. Kubinyi, Variable Selection in QSAR Studies. I. An Evolutionary Algorithm, *Quant. Struct.-Act. Relat.* **1994**, *13*, 285–294.

[30] B. T. Luke, Evolutionary Programming Applied to the Development of Quantitative Structure-Activity Relationships and Quantitative Structure-Property Relationships, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1279–1287.

[31] B. T. Luke, Comparison of Three Different QSAR/QSPR Generation Techniques, *J. Mol. Struct. (THEOCHEM)* **1999**, *468*, 13–20.

[32] D. Rogers, Evolutionary Statistics: Using a Genetic Algorithm and Model Reduction to Isolate Alternate Statistical Hypotheses of Experimental Data, in T. Bäck (Ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms*, San Francisco, Morgan-Kaufmann, **1997**, pp. 553–561.

[33] R. Leardi, Application of a Genetic Algorithm to Feature Selection Under Full Validation Conditions and to Outlier Detection, *J. Chemom.* **1994**, *8*, 65–79.

[34] L. M. Shi, Y. Fan, T. G. Myers, P. M. O'Conner, K. D. Paull, S. H. Friend, J. N. Weinstein, Mining the NCI Anticancer Drug Discovery Databases: Genetic Function Approximation for the QSAR Study of Anticancer Ellipticine Analogues, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 189–199.

[35] H. van de Waterbeemd, S. Wold, Introduction, in H. van de Waterbeemd (Ed.), *Chemometric Methods in Molecular Design*, VCH Publishers, Inc., New York, NY, **1995**, pp. 1–13.

[36] R. Leardi, A. L. Gonzálex, Genetic Algorithms Applied to Feature Selection in PLS Regression: How and When to Use Them, *Chemom. Intell. Lab. Syst.* **1998**, *41*, 195–207.

[37] K. Hasegawa, Y. Miyashita, K. Funatsu, GA Strategy for Variable Selection in QSAR Studies: GA-based PLS Analysis of Calcium Channel Antagonists, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 306–310.

[38] K. Hasegawa, T. Kimura, K. Funatsu, GA Strategy for Variable Selection in QSAR Studies: Application of GA-based Region Selection to a 3D-QSAR Study of Acetylcholinesterase Inhibitors, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 112–120.

[39] T. Kimura, K. Hasegawa, K. Funatsu, GA Strategy for Variable Selection in QSAR Studies: GA-based Region Selection for CoMFA Modeling, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 276–282.

[40] A. C. Gaudio, A. Korolkovas, Y. Takahata, Quantitative Structure-Activity Relationships for 1,4-dihydropyridine Calcium Channel Antagonists (Nifedipine Analogues): A Quantum Chemical/Classical Approach, *J. Pharm. Sci.* **1994**, *83*, 1110–1115.

[41] A. R. Ortiz, M. T. Pisabarro, F. Gago, R. C. Wade, Prediction of Drug Binding Affinities by Comparative Binding Energy Analysis, *J. Med. Chem.* **1995**, *38*, 2681-2691.

[42] S. J. Cho, A. Tropsha, Cross-validated R^2-guided Region Selection for Comparative Molecular Field Analysis: A Simple Method to Achieve Consistent Results, *J. Med. Chem.* **1995**, *38*, 1060–1066.

[43] S.-S. So, M. Karplus, Evolutionary Optimization in Quantitative Structure-Activity Relationships: An Application of Genetic Neural Networks, *J. Med. Chem.* **1996**, *39*, 1521–1530.

[44] S.-S. So, M. Karplus, Genetic Neural Networks for Quantitative Structure-Activity Relationships: Improvements and Application of Benzodiazepine Affinity for Benzodiazepine/GABA$_A$ Receptors, *J. Med. Chem.* **1996**, *39*, 5246–5256.

[45] S.-S. So, M. Karplus, Three-dimensional Quantitative Structure-Activity Relationships from Molecular Similarity Matrices and Genetic Neural Networks: I. Method and Validations, *J. Med. Chem.* **1997**, *40*, 4347–4359.

[46] S.-S. So, M. Karplus, Three-dimensional Quantitative Structure-Activity Relationships from Molecular Similarity Matrices and Genetic Neural Networks: II. Applications, *J. Med. Chem.* **1997**, *40*, 4360–4371.

[47] S.-S. So, M. Karplus, A Comparative Study of Ligand-Receptor Complex Binding Affinity Prediction Methods Based on Glycogen Phosphorylase Inhibitors, *J. Comput.-Aided Mol. Des.* **1999**, *13*, 243–258.

[48] Ajay, A Unified Framework for Using Neural Networks to Build QSARs, *J. Med. Chem.* **1993**, *36*, 3565-3571.

[49] D. J. Maddalena, G. A. R. Johnston, Prediction of Receptor Properties and Binding Affinity of Ligands to Benzodiazepine/GABA-A Receptors Using Artificial Neural Networks, *J. Med. Chem.* **1995**, *38*, 715–724.

[50] A. R. Dinner, S.-S. So, M. Karplus, Use of Quantitative Structure-Property Relationships to Predict the Folding Ability of Model Proteins, *Proteins: Struct., Funct., Genet.* **1998**, *33*, 177–203.

[51] T. M. Nelson, P. C. Jurs, Prediction of Aqueous Solubility of Organic Compounds, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 601–609.

[52] L. M. Egolf, M. D. Wessel, P. C. Jurs, Prediction of Boiling Points and Critical Temperatures of Industrially Important Organic Compounds from Molecular Structure, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 947–956.

[53] L. Xu, J. W. Ball, S. L. Dixon, P. C. Jurs, Quantitative Structure-Activity Relationships for Toxicity of Phenols using Regression Analysis and Computational Neural Networks, *Environmental Toxicol. Chem.* **1994**, *13*, 841–851.

[54] J. M. Sutter, S. L. Dixon, P. C. Jurs, Automated Descriptor Selection for Quantitative Structure-Activity Relationships Using Generalized Simulated Annealing, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 77–84.

[55] P. C. Jurs, S. L. Dixon, L. M. Egolf, Molecular Concepts: Representations of Molecules, in H. van de Waterbeemd (Ed.), *Chemometric Methods in Molecular Design*, VCH Publishers, Inc., New York, **1995**, pp. 15–38.

[56] M. D. Wessel, P. C. Jurs, Prediction of Normal Boiling Points for a Diverse Set of Industrially Important Organic Compounds from Molecular Structure, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 841–850.

[57] J. M. Sutter, P. C. Jurs, Prediction of Aqueous Solubility for a Diverse Set of Heteroatom-containing Organic Compounds Using a Quantitative Structure-Activity Relationship, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 100–107.

[58] B. E. Mitchell, P. C. Jurs, Prediction of Autoignition Temperature of Organic Compounds from Molecular Structure, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 538–547.

[59] H. L. Engelhardt, P. C. Jurs, Prediction of Supercritical Carbon Dioxide Solubility of Organic Compounds from Molecular Structure, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 478–484.

[60] M. D. Wessel, P. C. Jurs, J. W. Tolan, S. M. Muskal, Prediction of Human Intestinal Absorption of Drug Compounds from Molecular Structure, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 726–735.

[61] B. E. Mitchell, P. C. Jurs, Prediction of Aqueous Solubility of Organic Compounds from Molecular Structure, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 489–496.

[62] J. Sadowski, J. Gasteiger, From Atoms and Bonds to 3-dimensional Atomic Coordinates – Automatic Model Builders, *Chem. Rev.* **1993**, *7*, 2567–2581.

[63] D. Jouan-Rimbaud, D. L. Massart, O. E. de Noord, Random Correlation in Variable Selection for Multivariate Calibration with a Genetic Algorithm, *Chemom. Intell. Lab. Syst.* **1996**, *35*, 213–220.

[64] R. Fair, Econometrics and Presidential Elections, *J. Economic Perspectives* **1996**, *10*, 89–102.

[65] S. J. Cho, W. Zheng, A. Tropsha, Rational Combinatorial Library Design. 2. Rational Design of Targeted Combinatorial Peptide Libraries Using Chemical Similarity Probe and the Inverse QSAR Approaches, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 259–268.

[66] T. R. Kowar, Genetic Function Approximation Experimental Design (GFAXD): A New Method for Experimental Design, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 858–866.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

6 Chemometrics

Ron Wehrens and Lutgarde M. C. Buydens

Abbreviations

ANN	Artificial neural network
CoMFA	Comparative molecular field analysis
EA	Evolutionary algorithm
GA	Genetic algorithm
KNN	*k*-nearest-neighbor classification
LDA	Linear discriminant analysis
MLR	Multiple linear regression
(N)IR	(Near) infrared
NLM	Nonlinear mapping
NMR	Nuclear magnetic resonance
PC(R)	Principal component (regression)
PET	Poly(ethylene terephthalate)
PLS	Partial least squares (or: projection to latent structures)
PRESS	Predictive residual error sum of squares
QSAR	Quantitative structure-activity relationship
SA	Simulated annealing
SDEP	Standard deviation of prediction errors
SIA	Sequential injection analysis
UV	Ultraviolet
WDXRF	Wavelength-dispersive X-ray fluorescence

Symbols

r	product-moment correlation coefficient

6.1 Introduction

The discipline of chemometrics is now some 30 years old, the term being used for the first time in 1972 by Svante Wold. Although the number of pure chemometrics laboratories is quite small, the discipline is firmly established in the chemical community, with chemometrics papers appearing in a broad range of chemical journals. Moreover, two specialized chemometrics journals have been published for quite some time, and several textbooks have appeared [1–4]. A recent definition of the term "chemometrics" reads [3]:

> "Chemometrics is a chemical discipline that uses mathematics, statistics and formal logic (a) to design or select optimal experimental procedures; (b) to provide maximum relevant chemical information by analysing chemical data; and (c) to obtain knowledge about chemical systems."

The links between chemometrics and analytical chemistry have always been very strong, for obvious reasons: quantitative analysis has greatly benefited from multivariate chemometrical regression techniques, but qualitative analysis has also received attention. In recent years, the field has expanded significantly, as is obvious from the number of chemometrical papers, applications appearing in nonchemometrical and nonanalytical journals, and the diversity of applications in chemometrics journals and books.

Optimization problems have always received considerable attention in the chemometrics literature. Initially, the optimization of experimental conditions using experimental designs was stressed where, with a minimal number of experiments, the maximum information must be obtained. With the advent of computer-based chemistry, the number of experiments that could be performed in a reasonable time increased by orders of magnitude, and this led to the development and application of global optimization methods such as evolutionary algorithms, simulated annealing, and simplex optimization.

Overviews of the application of evolutionary algorithms in chemometrics can, for example, be found in [5–10]. In this chapter, the most important application areas for optimization with evolutionary algorithms are highlighted. For clarity, they have been divided into three classes: parameter estimation; subset selection; and miscellaneous applications.

Parameter estimation problems typically arise in cases where there is no analytical solution for a modeling problem, and classical methods rely on iteration from a more or less accurate initial estimate. Examples include nonlinear and robust estimation, curve fitting and neural network applications. The second area considers combinatorial problems where a small set of variables or objects must be selected from a larger set, according to some criterion. Problems of this kind are known to be NP-complete, which means that with increasing complexity an exhaustive search quickly becomes impossible. Some applications which do not quite fit either of these two categories are discussed in the last of the three sections.

6.2 Parameter Estimation

Parameter estimation problems are optimization problems where evolutionary algorithms are used to estimate the parameters in complex nonlinear equations; the quality of a parameter set is judged by comparing the result of the equation to measured data. Examples include curve fitting and solving differential equations (e.g., elucidating rate constants). The representation of a solution in the population is quite simple: each parameter is directly coded in the string, in either a real or binary representation.

6.2.1 Curve Fitting

"Classical" curve fit procedures such as the Newton-Raphson method are iterative procedures requiring a reasonable first guess. In many cases, such a first guess is difficult to obtain, especially when many peaks must be fitted simultaneously. Replacing these classical algorithms with global search methods such as simulated annealing or evolutionary algorithms is a logical step, and many applications have appeared in spectroscopy.

Choy and Sanctuary describe a genetic algorithm (GA) for fitting one-dimensional nuclear magnetic resonance (NMR) signals with a scaled squared error between the experimental and calculated spectrum as the fitness function [11]. The estimated parameters were peak damping factor and frequency; peak amplitude and phase were then obtained by simple regression. A priori knowledge can be introduced by imposing constraints on the parameter values. It was shown using simulated noisy spectra that the GA performed better than an iterative maximum likelihood method proposed by the same authors, although execution times were longer. (For more applications of EAs in NMR, see Chapter 10.)

In the area of X-ray spectroscopy, several groups have used GAs to solve crystal structures from powder diffraction data. The aim in this case is to find parameters describing both the structure of a molecule and its orientation in the unit cell. Shankland et al. use a χ^2 error value as the fitness function, based on the difference between the observed and calculated diffraction pattern [12]. They obtain good results for molecules with up to ten degrees of freedom (three translational, three rotational, and four internal). Essentially the same approach is taken in [13], where it was also concluded that the GA method appeared to be faster than a Monte Carlo procedure. The method was also successfully applied to a structure with 12 torsion angles [14]. More details on applications of EAs in X-ray crystallography appear in Chapter 9.

A combination of a GA and an artificial neural network (ANN) was applied in [15] to fit X-ray equator diffractograms of poly(ethylene naphthalate) (PEN) yarns. Pearson VII lines, employing four parameters for each peak were used to fit the spectra; Gaussian and Lorentzian lines are special cases of the Pearson VII lineshape. The ANN was used as a peak-picking procedure and, as such, yielded rough initial estimates for the search space. The evaluation function F was a combination of the correlation r between the experimental and simulated spectrum and the root mean square error E:

$$F = \frac{E}{(1+r)^2} \tag{1}$$

This function proved to be superior to the separate r and E values. The GA was shown to be much more robust than a steepest descent method that often ended up in a local optimum.

The same program, CFIT, was applied to IR spectra of poly(ethylene terephthalate) (PET) yarns [16]. Again, very good results were obtained, although in this case it was necessary to refine the final GA solution with a local optimization method to obtain the optimal solution.

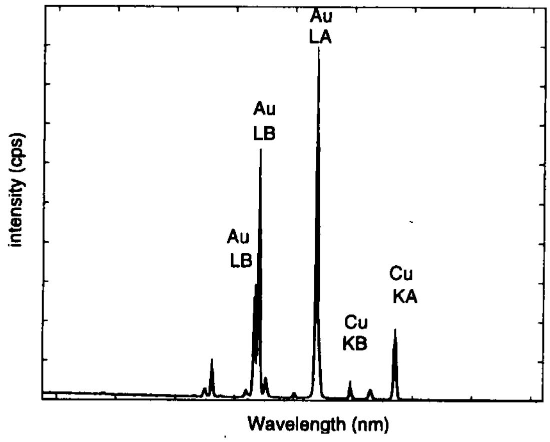


Figure 1. Example of a wavelength-dispersive X-ray fluorescence (WDXRF) spectrum of a gold coin. Due to the huge selectivity for almost all elements in qualitative analysis and the extremely wide dynamic range in quantitative analysis, WDXRF is commonly applied in the determination of the elemental composition of unknown samples.

Dane et al. investigated the use of GAs for the analysis of layered materials with two forms of X-ray fluorescence spectroscopy [17, 18]. An example of such a spectrum is given in Fig. 1. The number of layers, the thickness of each layer and the composition of each of the layers could be elucidated in this way. Theoretical spectra were calculated with the so-called fundamental parameter method [19] and, again, Eq. 1 was used as the evaluation function.

6.2.2 Nonlinear Modeling

The nonlinear calibration of an array of four ion-selective electrodes has been performed using a GA [20]. For each electrode, the cell potential, slope, and selectivity coefficients were optimized. A relative quadratic difference between observed and predicted electrode potentials was used as the evaluation function. The best solutions of six replicated GA runs were used in a subsequent simplex optimization to obtain the final solution. Comparison with a pure simplex approach revealed that the simplex method alone was better in determining the parameters with the largest effect on the error function; the GA-simplex hybrid was better in determining values for relatively small selectivity coefficients only. It was concluded that the electrode array did not constitute a very appropriate model system for multivariate calibration, since unrealistically high species concentrations are needed to obtain any response.

The same authors described the determination of stability constants by a GA using calorimetric and polarographic literature data [21]. An interesting aspect of this implementation is that the chromosomes have a variable length because of the explicit coding of the number of metals and ligands that are involved. Again, a simplex method is used to fine-tune the final GA results.

Global optimization methods such as GAs and simulated annealing (SA) are compared in the determination of rate constants and reaction mechanisms: complex differential equations are solved [22]. The conclusion was that the GA and SA both work well, whereas the simplex algorithm sometimes gets stuck in a local optimum. In general, the search precision of the SA method used is better: the final error is lower than with the GA. This is in agreement with other results from the literature.

Several applications describe the use of GAs in summarizing a data set in terms of complicated basis functions. An example from the field of quantitative structure-activity relationships (QSAR) is described in [23]: the octane number of 293 hydrocarbons, divided into five classes, is predicted. The final solution is again obtained by a hill-climbing method. The occurrence of 15 functional groups was used as representation of the hydrocarbons. The results were much better than those obtained with linear regression, but this is not surprising because of the nonlinear terms taken into account. The advantage of the approach over, for example, neural networks, is that the results can be interpreted chemically. Another example is the description of a data set with wavelets [24], where the optimal coefficients for a set of basis functions containing splines, polynomial terms as well as wavelets are identified with a GA.

A final area in which iterative procedures are often used to obtain the desired results is robust modeling. Here, one seeks to find a model which is not only optimal in terms of a minimal prediction error, but also insensitive to outlying observations. Examples include finding those parameters yielding a minimal *median* squared error (as opposed to the minimal *mean* squared error) [25], finding the optimal discrimination vector in robust linear discriminant analysis (LDA) [26], and finding piecewise linear discriminant functions [27, 28]. In the latter application, it was found that simplex methods consistently outperformed global search methods such as several forms of GA and SA. Several reasons were cited: a good estimate was already available and the number of evaluations was limited; the parameter values depended strongly on each other (*epistasis*), which means that values that are good in one solution may be very bad in another solution, and the response surface suited the simplex algorithm.

6.2.3 Neural Networks

The training of neural networks can be seen as an optimization problem, and the frequently used backpropagation method can be replaced by an evolutionary algorithm (EA). Several papers have appeared proving the feasibility of such an approach [29, 30]. Clear improvements over backpropagation or other training methods have not been reported, however.

Another, potentially more interesting, application is the investigation of the response surface defined by a trained neural network. In this case, the parameters that must be optimized are the *input parameters*, and the evaluation function is simply the output of the network. In [31], a neural network was trained to predict the level of bioactivity, given a set of operating conditions for sequential-injection analysis (SIA). A GA was used to find those conditions leading to maximum bioactivity. The large variability of the values of two of the parameters indicated that they did not have a significant influence on the bioactivity, and they could be set to default values. The optimal predicted bioactivities were larger than the largest value in the training set. Unfortunately, the results were not validated in practice. The dangerous part in this approach is the neural network model; several ANN models may be able to predict the test set to the same degree of accuracy, while having completely different weight sets. It is uncertain that two networks with similar performances on a test set actually describe the same response surface. However, other applications have shown this to be a viable route in practice. For instance, a GA was used to "invert" an ANN that modeled the relationship between physical structure of yarns (input) and ten physical properties such as tenacity, shrinkage, and elongation at break (output). The goal was to find the yarn structure which led to optimal properties [32].

6.3 Subset Selection

Many different forms of subset selection by EAs appear in the chemometric literature. In almost all cases, the EA uses a binary representation, in which a 1 indicates retention of the object or parameter, and a 0 leads to rejection. The reason for the popularity of EAs is their ability to traverse the huge search spaces resulting from the inherently combinatorial nature of the subset selection problem: the number of possibilities of selecting m variables out of a superset of p variables is

$$\frac{p!}{m!(p-m)!} \tag{2}$$

which becomes practically impossible for even moderate values of p and m. Moreover, the correct (minimal) value of m is usually not known. Stepwise procedures usually perform suboptimally.

In quantitative analysis, the two main problem types are feature extraction and object selection. The difference between these is illustrated in Fig. 2. Usually, rows in a data matrix represent objects and columns represent variables. Feature selection then comes down to selecting columns, often with the aim of obtaining more predictive or robust calibration models.

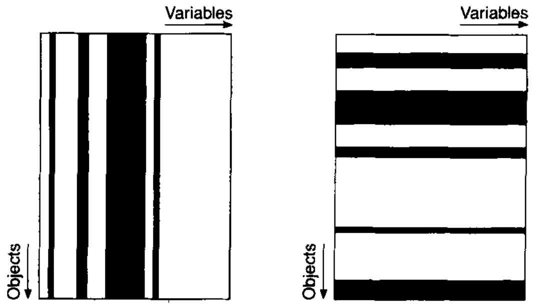


Figure 2. Variable selection (left) versus object selection (right). Rows indicate objects, columns indicate variables.

6.3.1 Feature Selection

Feature selection, or variable selection as it is most often called in the chemometrics literature, is an important application of global optimization methods. It is generally accepted that even so-called full-spectrum methods like partial least-squares regression (PLS) benefit when uninformative variables are removed prior to model building. With large numbers of variables, techniques not applying optimization methods often rely on the weights in the final regression vector or the size or variability of loadings (see, e.g., [33, 34]). Nonevolutionary optimization methods such as simulated annealing have also been applied [35]. However, many papers have appeared where evolutionary methods have been used for variable selection in quantitative applications (e.g., [36–38]); for qualitative applications, such as clustering or classification, it is harder to find examples (an overview of some widely used nonevolutionary methods can be found in [39] and references therein).

Many reports stress the importance of including the complete validation sequence in the process of feature extraction. In practice, this means that the data should be split in several groups and that cross-validation statistics should be used as evaluation criteria. If this is not done, there is a severe chance of *overtraining*: the algorithm just chooses those variables that describe the current dataset, and the generalizing properties of the resulting model are poor. An added benefit is that the cross-validation can also provide information on potentially outlying observations [40].

6.3.1.1 Spectroscopy

Examples of feature selection in spectroscopic calibration can be found in quantitative applications of IR spectroscopy [41–43] as well as qualitative ones [44], mass spectrometry [45] and UV-Visible spectroscopy [46]. Jouan-Rimbaud et al. showed that a GA for wavelength selection sometimes includes uninformative variables if a fitness function is used that is based solely on a predictive sum-of-squares [47]. Therefore, they proposed performing a forward stepwise selection from the final best solution found by the GA. Other options are to use a fitness function which chooses the smallest set of wavelengths that yields a model with a predictive power better than a prespecified threshold [45], or a fitness function which incorporates in some way the number of selected wavelengths [48]. Both have their disadvantages: in the first case, one must specify a threshold value (and, in practice, will have to try several values before a suitable one is found); in the second case, the form of the fitness function is more complicated and will also have to be fine-tuned.

In some cases, a selection is made not from the original variables, but from transformed variables. An example is the selection of principal components (PCs) to obtain an optimal quantitative model. Rather than using the *k* principal components with the largest eigenvalues, a set of PCs is selected according to some other criterion, usually related to the predictive ability of the model. In [48] the following fitness function is used:

$$f = \frac{PC \cdot dw \cdot PRESS}{(2 - dw)} \tag{3}$$

where *f* is the fitness value, *PC* is the number of PCs selected, *dw* is the value of the Durban-Watson criterion which checks whether residuals are normally distributed, and PRESS is the usual predictive residual error sum of squares [3]. The criterion in Eq. 3 minimizes the number of selected PCs, and maximizes predictive ability. In each case, the GA identified the global optimum of this function (as determined by an exhaustive search).

In [49], variable selection is performed on the original variables (wavelengths in the NIR region) as well as Fourier and wavelet coefficients. Contrary to other reports in the literature, (original) variable selection followed by multiple linear regression (MLR) performed worse than PCR and PLS; selection of Fourier or wavelet coefficients by a GA led to models of comparable quality to (full-spectrum) PCR/PLS. This may be an effect of the data sets or the fitness function.

6.3.1.2 3-D QSAR

Variable selection in the context of QSAR is described in a number of applications. Using a smaller set of informative variables can greatly increase the modeling capabilities of Comparative Molecular Field Analysis (CoMFA)-like analyses, and enhances the interpretability. Several algorithmic approaches have been described, most prominently the GOLPE method (Generating Optimal Linear PLS Estimations) [50, 51]. In this approach, experimental designs are used to assess the effect of different combinations of variables on the prediction (as given by the standard deviation of prediction errors, SDEP). A logical extension to this approach is to consider the space of all possible parameter combinations, and to use global search methods such as evolutionary algorithms to find the optimal one(s) [37, 52–54]. Rogers and Hopfinger describe a similar application called Genetic Function Approximation (GFA), where functions of the original variables are also considered [55]. This makes it possible to use quadratic terms or splines. (GFA actually belongs in the Parameter Estimation section, but because some parameters are estimated as zero, it is also a subset selection method.) The fact that several good models are obtained is seen as an advantage by the authors as it facilitates interpretation.

Instead of selecting individual variables from a steric or electrostatic grid, it is also possible to select regions and perform the PLS modeling with the gridpoints from the selected regions only (GA-based region selection, GARGS) [56, 57]. It is shown that GARGS coefficient contour maps are much easier to interpret than the standard CoMFA maps. A comparison with the results of applying GOLPE to the same data set shows that this approach also has a better predictive ability [51, 56].

6.3.1.3 Miscellaneous

Zupan and Novic use the correlation between the biological activity predicted by a neural network and experimental values as the fitness function in a GA for the selection of the 15 most useful variables out of a set of 120 for defining molecular structure [58]. Hou et al. apply a GA to select a subset from 24 features with which to predict the biological activity of two groups of HIV-1 inhibitors [59]. An application in quality control is described in [60], where the subset of product quality variables retaining most of the information contained in the original set is sought.

6.3.2 Object Selection

Subset selection applications where objects rather than variables are selected usually have one of two purposes: either the identification of outlying variables, or exactly the opposite, the construction of representative subsets, usually for clustering or model building purposes.

Outliers are objects which do not conform to the general pattern in the data. One approach, already mentioned, is the application of robust methods which are not influenced very much by the presence of outliers. However, their use is not without drawbacks, such as the high computational demands and the absence of reliable confidence estimates. A more frequently used strategy is to detect, and subsequently remove, outlying observations from the data set. In [61, 62], a GA is used to identify the least informative objects in a data set, and to build a model without them. While they may not be outliers in the statistical sense of the word, the effect of removing them from the data set is very much the same.

The selection of objects to form a representative subset is the subject of several investigations. In [63], a subset of objects is selected in such a way that the variance-covariance matrix of the complete set is approximated. The primary aim of the object selection lies in the field of (statistical) model building: in this way the representativeness of a training set can be assessed. The GA is compared with a kernel-density estimation procedure and it was concluded that, in smooth (simulated) data sets, the latter performed better. For real data, the two techniques had somewhat different characteristics. The GA-based subsets tended to describe the edges of the distribution better than the centers whereas, with the other procedure, stress is placed on the centers. Depending on the application, one of the two procedures should be selected.

In the context of high-throughput biological screening, a different approach can be taken. In such an application, a subset of molecules is selected in which each molecule is representative of a number of similar species. An approach to select an optimal subset for such cases is described in [64], where representative subsets of 100, 300, and 500 molecules were selected from a superset of 5000. The 281 original variables, chemical and physical descriptors, were summarized in 58 principal components prior to subset selection. A fitness function based on the product-moment correlation coefficient appeared to work well. In comparison with other methods, such as clustering, or using a maximum dissimilarity criterion, GAs are suitable for the selection of subsets that preserve local information [65]. Subsets retaining more global information are obtained by other approaches such as the maximum dissimilarity method, while the most representative subsets were obtained by clustering methods. Differences, however, may be caused by the optimization criterion: all methods use slightly different criteria.

In the context of experimental design, the GFA of Rogers and Hopfinger [55] was used to find good functional approximations of a set of experimental data, with a fitness function that incorporated the sum of squared errors, the number of data points, and the number of factors incorporated in the model [66]. Because splines are available to the GFA models as well as linear terms, the GA usually provided a better fit than the classical linear model. The authors then went on to propose an experimental design strategy in which all parameters of interest are varied randomly over the entire experimental range, with GFA being used to model the results. This is somewhat contradictory to the philosophy behind experimental design methods, and probably leads to experimental designs with rather more experiments than actually necessary. An approach which is more orthodox is described in [67], where GAs have been applied to select the most informative experiments for a system with six factors at three to seven levels. The discriminant of the experimental design matrix was maximized, and it was found that 28 experiments sufficed to estimate the 25 parameters of interest.

6.4 Miscellaneous

Several applications do not readily fit into either of the two categories treated above. An example is clustering, where a data set is divided into more or less homogeneous subsets. Although it is possible to formulate a cluster procedure as a subset selection problem (selecting the centers of the clusters and subsequently assigning the remainder of the data to the respective clusters), other formulations are also possible. Rather than treating related problems in different places, they are gathered below in one section.

6.4.1 Clustering and Classification

The division of data sets into separate clusters is described in a number of publications. In all cases, the number of clusters is assumed to be known. For large data sets, Lucasius et al. describe a *k*-medoid clustering method, in which the GA is used to pick the cluster centers [68]. The fitness criterion consists of the minimization of the sum over all clusters of the distances of a mediod to all samples in the same cluster. This is a true object selection problem, and the number of parameters in the GA strings equals the number of cluster centroids. Another application, where the length of the string equals the number of objects, and each object is assigned a cluster number, is less suitable for large data sets [69]. In the latter, the evaluation function is based on a general Gaussian mixture model:

$$J_{GM} = \sum_{k=1}^{K} N_k \ln(|S_k/N_k|) \tag{4}$$

where K is the number of clusters, and N_k and S_k are the number of elements and the scatter matrix of cluster k, respectively. In this kind of model-based maximum-likelihood clustering, the expectation-maximization (EM) algorithm [70] is the usual optimization method.

The k-nearest-neighbor (KNN) classification technique was combined with a GA to predict the conservation of water molecules in proteins upon ligand binding [71]. The task of the GA was to find a weighting scheme for the four variables that would provide optimal KNN performance. After training on data from 13 nonhomologous proteins, the prediction accuracy on seven new proteins was 75 %. In other words, 75 % of the conserved waters were correctly predicted.

The aim of nonlinear mapping (NLM) is to find a low-dimensional representation of high-dimensional data that retains the interobject distances in the low-dimensional representation; that is, the topology of the data is preserved. Subsequently, the data can be inspected visually, or a clustering can be performed in the reduced space. Usually, the starting point is a small (typically two) number of principal components, which is further optimized by steepest descent minimization. In [72], a GA is proposed for NLM; this employs two specialized operators to speed up the search and prevent premature convergence.

Reijmers et al. discuss several versions of phylogenetic clustering with GAs [73, 74]. The problem here is to construct a binary tree that reproduces the observed distances between protein sequences. Several representations of the tree topology are compared. One representation optimizes a working distance matrix from which, by a straightforward algorithm, a tree topology is derived; another representation uses a Prüfer number (see Fig. 3) that directly codes for the topology [75]. The order of individual figures in the Prüfer number is to be optimized. Both representations show markedly different search characteristics [74].

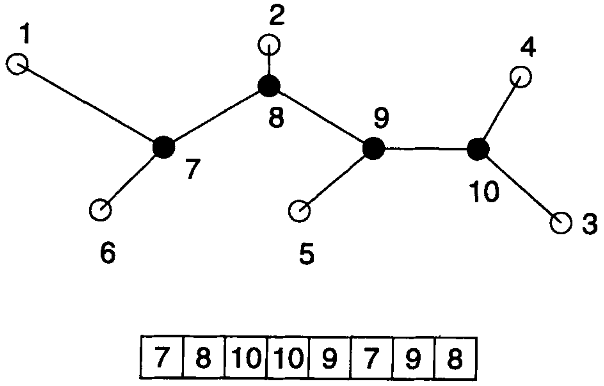


Figure 3. Representation of a bifurcating tree by a Prüfer number. The leaves (open circles) correspond to the objects that are to be clustered. Roughly speaking, the Prüfer number indicates the order in which the nodes (filled circles) are joined to the leaves.

Applications in which elements are to be ordered according to some criterion are sometimes termed *sequencing* problems. In chemistry, this is a relatively rare type of problem. Apart from the phylogenetic clustering using Prüfer numbers, another application is described in the context of the interpretation of 2-D NMR data of proteins [76]. The patterns associated with individual amino acids in the protein are to be ordered in such a way that the patterns match the amino acid types in the (known) sequence, and that there is a maximum of contacts visible in the spectra.

The last example of the use of EAs is the identification of classification rules to determine suitable detectors in ion-chromatography by a *classifier system* [77]. The population now represents different rules, whose individual strength is given by their ability to make classifications in agreement with the input data. New rules are constructed by the usual genetic operators. In a later paper, it was concluded that the crossover operator did not contribute significantly to the generation of effective classification rules [78].

6.5 Discussion

From the wealth of publications in the chemometrics domain describing applications of EAs (and GAs in particular), it is clear that EAs have become part of the standard repertoire of optimization methods. In many cases, writing an evaluation function is easy and application is straightforward; standard search settings suffice to obtain good results. This is especially true for subset selection problems and curve-fitting applications. Surprisingly, the number of detailed comparisons of the search behavior of EAs and other optimization methods is still rather small. As argued in Chapter 2, fair comparisons between stochastic optimization methods are notoriously difficult, if only because one is usually very familiar with one of the methods, and therefore the other methods may receive less attention. In the cases where a comparison is made, the EA does not always appear to be the most efficient method, although it is usually possible to obtain a good estimate of the global optimum. Other methods, most notably SA, are often faster and more precise. The same results have been found outside the field of chemometrics.

Nevertheless, EAs are very popular. The speed of an optimization method is apparently not longer so much of an issue, except for the largest problems. The lack of search precision is countered, if necessary, by the subsequent application of a more local search method such as a gradient method or a simplex optimization. Clearly, these disadvantages are thought to be less important than the ease of implementation and the very good properties in locating global optima. Indeed, in several cases it is reported that EAs could tackle problems which proved beyond other optimization methods.

Since the field of chemometrics is developing and expanding quickly, it is to be expected that EAs will continue to find new applications. As the diversity of the problems increases, it may be expected that standard methods will no longer suffice and that the implementation of optimization methods will have to rise to new challenges. A detailed understanding of the problem characteristics that make an optimization easy for one particular method and difficult for another method is required. Of course, the underlying chemistry should be the main source of inspiration in this respect. Exciting times lie ahead!

References

[1] M. A. Sharaf, D. L. Illman, B. R. Kowalski, *Chemometrics*, John Wiley & Sons, New York, **1986**.

[2] R. G. Brereton, *Chemometrics: Application of Mathematics and Statistics to Laboratory Systems*, Ellis Horwood, Chichester, UK, **1990**.

[3] D. L. Massart, B. G. M. Vandeginste, L. M. C. Buydens, S. de Jong, P. J. Lewi, J. Smeyers-Verbeke (Eds.), *Handbook of Chemometrics and Qualimetrics: Part A*, volume 20A of *Data Handling in Science and Technology*, Elsevier Science Publishers, Amsterdam, **1998**.

[4] B. G. M. Vandeginste, D.L. Massart, L. M. C. Buydens, S. de Jong, P. J. Lewi, J. Smeyers-Verbeke (Eds.), *Handbook of Chemometrics and Qualimetrics: Part B*, volume 20B of *Data Handling in Science and Technology*, Elsevier Science Publishers, Amsterdam, **1998**.

[5] C. B. Lucasius, G. Kateman, Understanding and Using Genetic Algorithms. Part 1: Concepts, Properties and Context, *Chemom. Intell. Lab. Syst.* **1993**, *19*, 1–33.

[6] C. B. Lucasius, G. Kateman, Understanding and Using Genetic Algorithms. Part 2: Representation, Configuration and Hybridization, *Chemom. Intell. Lab. Syst.* **1994**, *25*, 99–145.

[7] D. B. Hibbert, Genetic Algorithms in Chemistry, *Chemom. Intell. Lab. Syst.* **1993**, *19*, 277–293.

[8] R.E. Shaffer, G. W. Small, Learning Optimization from Nature, *Anal. Chem.* **1997**, *69*, 236A – 242A.

[9] R. S. Judson, Genetic Algorithms and their Use in Chemistry, in K. B. Lipkowitz, D. B. Boyd (Eds.), *Reviews in Computational Chemistry, Vol. 10*, VCH, New York, **1997**, pp. 1–73.

[10] B. K. Lavine, A. J. Moores, Genetic Algorithms in Analytical Chemistry, *Anal. Lett.* **1999**, *32*, 433–445.

[11] W. Y. Choy, B. C. Sanctuary, Using Genetic Algorithms with A Priori Knowledge for Quantitative NMR Signal Analysis, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 685–690.

[12] K. Shankland, W. I. F. David, T. Csoka, Crystal Structure Determination from Powder Diffraction Data by the Application of a Genetic Algorithm, *Z. Kristall.* **1997**, *212*, 550–552.

[13] K. D. M. Harris, R. L. Johnston, B. M. Kariuki, The Genetic Algorithm: Foundations and Applications in Structure Solution from Powder Diffraction Data, *Acta Crystallogr.* **1998**, *A54*, 632–645.

[14] B. M. Kariuki, P. Calcagno, K. D. M. Harris, D. Philp, R. L. Johnston, Evolving Opportunities in Structure Solution from Powder Diffraction Data – Crystal Structure Determination of a Mole-

cular System with Twelve Variable Torsion Angles, *Angew. Chem. Int. Ed. Engl.* **1999**, *38*, 835–838.

[15] A. P. de Weijer, C. B. Lucasius, L. Buydens, G. Kateman, Curve Fitting using Natural Computation, *Anal. Chem.* **1994**, *66*, 23–31.

[16] A. P. de Weijer, L. Buydens, G. Kateman, H. M. Heuvel, Spectral Curve Fitting of Infrared Spectra Obtained from Semi-Crystalline Polyester Yarns, *Chemom. Intell. Lab. Syst.* **1995**, *28*, 149–164.

[17] A. D. Dane, P. A. M. Timmermans, H. A. van Sprang, L.M.C. Buydens, A Genetic Algorithm for Model-Free X-Ray Fluorescence Analysis of Thin Films, *Anal. Chem.* **1996**, *68*, 2419–2425.

[18] A. D. Dane, A. Veldhuis, D. K. G. de Boer, L. M. C. Buydens, Application of Genetic Algorithms for Characterization of Thin Layered Materials by Glancing Incidence X-Ray Reflectometry, *Physica B* **1998**, *253*, 254–268.

[19] D. K. G. de Boer, Calculation of X-Ray Fluorescence Intensities from Bulk and Multilayer Samples *X-Ray Spectrom.* **1990**, *19*, 145–154.

[20] M. Hartnett, D. Diamond, Potentiometric Nonlinear Multivariate Calibration with Genetic Algorithm and Simplex Optimization, *Anal. Chem.* **1997**, *69*, 1909–1918.

[21] M. Hartnett, M. Bos, W. E. van der Linden, D. Diamond, Determination of Stability Constants using Genetic Algorithm, *Anal. Chim. Acta* **1995**, *316*, 347–362.

[22] D. B. Terry, M. Messina, Heuristic Search Algorithm for the Determination of Rate Constants and Reaction Mechanisms from Limited Concentration Data, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1232–1238.

[23] R. Meusinger, R. Moros, Determination of Quantitative Structure-Octane Rating Relationships of Hydrocarbons by Genetic Algorithms, *Chemom. Intell. Lab. Syst.* **1999**, *46*, 67–78.

[24] X. Shao, F. Yu, H. Kou, W. Cai, Z. Pan, A Wavelet-Based Genetic Algorithm for Compression and Denoising of Chromatograms, *Anal. Lett.* **1999**, *32*, 1899–1915.

[25] P. Vankeerberghen, J. Smeyers-Verbeke, R. Leardi, C. L. Karr, D. L. Massart, Robust Regression and Outlier Detection for Non-Linear Models using Genetic Algorithms, *Chemom. Intell. Lab. Syst.* **1995**, *28*, 73–87.

[26] T. Li, H. Mei, P. Cong, Combining Nonlinear PLS with the Numeric Genetic Algorithm for QSAR, *Chemom. Intell. Lab. Syst.* **1999**, *45*, 177–184.

[27] R. E. Shaffer, G. W. Small, Genetic Algorithms for the Optimization of Piecewise Linear Discriminant Analysis, *Chemom. Intell. Lab. Syst.* **996**, *35*, 87–104.

[28] R. E. Shaffer, G. W. Small, Comparison of Optimization Algorithms for Piecewise Linear Discriminant Analysis: Application to Fourier Transform Infrared Remote Sensing Measurements, *Anal. Chim. Acta* **1996**, *331*, 157–175.

[29] J. Jiang, J. Wang, X. Song, R. Yu, Network Training and Architecture Optimization by a Recursive Approach and a Modified Genetic Algorithm, *J. Chemom.* **1996**, *10*, 253–267.

[30] M. S. Sanchez, L. A. Sarabia, GINN (Genetic Inside Neural Networks): towards a Non-Parametric Training, *Anal. Chim. Acta* **1997**, *348*, 533–542.

[31] F. R. Burden, B. S. Rosewarne, D. A Winkler, Predicting Maximum Bioactivity by Effective Inversion of Neural Networks using Genetic Algorithms, *Chemom. Intell. Lab. Syst.* **1997**, *38*, 127–137.

[32] A. P. de Weijer, C. B. Lucasius, L. M. C. Buydens, G. Kateman, H. M. Heuvel, Using Genetic Algorithms for an Artificial Neural Network Inversion, *Chemom. Intell. Lab. Syst.* **1993**, *20*, 45–55.

[33] V. Centner, D. L. Massart, O. E. de Noord, S. de Jong, B. M. Vandeginste, C. Sterna, Elimination of Uninformative Variables for Multivariate Calibration, *Anal. Chem.* **1996** *68*, 3851–3858.

[34] B. K. Alsberg, D. B. Kell, R. Goodacre, Variable Selection in Discriminant Partial Least-Squares Analysis, *Anal. Chem.* **1998** *70*, 4126–4133.

[35] J. H. Kalivas (Ed.), *Adaption of Simulated Annealing to Chemical Optimization Problems*, volume 15 of *Data Handling in Science and Technology*, Elsevier Science Publishers, Amsterdam, **1995**.

[36] H. Kubinyi, Evolutionary Variable Selection in Regression and PLS Analyses, *J. Chemom.* **1996**, *10*, 119–133.

[37] R. Leardi, Genetic Algorithms in Feature Selection, in J. Devillers (Ed.), *Genetic Algorithms in Molecular Modelling*, Academic Press, London, **1996**, pp. 67–86.

[38] R. Leardi, A. L. González, Genetic Algorithms Applied to Feature Selection in PLS Regression: How and When to Use Them, *Chemom. Intell. Lab. Syst.* **1998**, *41*, 195–207.

[39] M. Dash, H. Liu, Feature Selection for Classification, *Intell. Data Anal.* **1997**, *1*, 131–156.

[40] R. Leardi, Application of a Genetic Algorithm to Feature Selection under Full Validation Conditions and to Outlier Detection, *J. Chemom.* **1994**, *8*, 65–79.

[41] C. B. Lucasius, M. L. M. Beckers, G. Kateman, Genetic Algorithms in Wavelength Selection: a Comparative Study, *Anal. Chim. Acta* **1994**, *286*, 135–153.

[42] A. S. Bangalore, R. E. Shaffer, G. W. Small, M. A. Arnold, Genetic-Algorithm-Based Method for Selecting Wavelengths and Model Size for Use with PLS Regression: Application to NIR Spectroscopy, *Anal. Chem.* **1996**, *68*, 4200–4212.

[43] R. E. Shaffer, G. W. Small, M. A. Arnold, Genetic Algorithm-Based Protocol for Coupling Digital Filtering and PLS Regression: Application to the NIR Analysis of Glucose in Biological Matrices, *Anal. Chem.* **1996**, *68*, 2663–2675.

[44] W. H. A. M. van den Broek, D. Wienke, W. J. Melssen, L. M. C. Buydens, Optimal Wavelength Range Selection by a Genetic Algorithm for Discrimination Purposes in Spectroscopic Infrared Imaging, *Appl. Spectr.* **1997**, *51*, 1210–1217.

[45] D. Broadhurst, R. Goodacre, A. Jones, J. J. Rowland, D. B. Kell, Genetic Algorithms as a Method for Variable Selection in Multiple Linear Regression and Partial Least Squares Regression, with Applications to Pyrolysis Mass Spectrometry, *Anal. Chim. Acta* **1997**, *348*, 71–86.

[46] M. Arcos, M. Oritz, B. Villahoz, L. A. Sarabia, Genetic-Algorithm-Based Wavelength Selection in Multicomponent Spectrometric Determination by PLS: Application on Indomethacin and Acemethacin Mixture, *Anal. Chim. Acta* **1997**, *339*, 63–77.

[47] D. Jouan-Rimbaud, D. L. Massart, O. E. de Noord, Random Correlation in Variable Selection for Multivariate Calibration with a Genetic Algorithm, *Chemom. Intell. Lab. Syst.* **1996**, *35*, 213–220.

[48] A. S. Barros, D. N. Rutledge, Genetic Algorithm Applied to the Selection of Principal Components, *Chemom. Intell. Lab. Syst.* **1998**, *40*, 65–81.

[49] U. Depczynski, K. Jetter, K. Molt, A. Niemöller, Quantitative Analysis of Near-Infrared Spectra by Wavelet Coefficient Regression using a Genetic Algorithm, *Chemom. Intell. Lab. Syst.* **1999**, *47*, 179–187.

[50] M. Baroni, S. Clementi, G. Cruciani, G. Costantino, D. Riganelli, Predictive Ability of Regression Models. Part II: Selection of the Best Predictive PLS Model, *J. Chemom.* **1992**, *6*, 347–356.

[51] M. Baroni, G. Costantino, G. Cruciani, D. Riganelli, R. Valigi, S. Clementi, Generating Optimal Linear PLS Estimations (GOLPE): an Advanced Chemometrical Tool for Handling 3D-QSAR Problems, *Quant. Struct.-Act. Relat.* **1993**, *12*, 9–12.

[52] W. J. Dunn, D. Rogers, Genetic Partial Least Squares in QSAR, in J. Devillers (Ed.), *Genetic Algorithms in Molecular Modelling*, Academic Press, London, **1996**, pp. 109–130.

[53] K. Hasegawa, Y. Myashita, K. Funatsu, GA Strategy for Variable Selection in QSAR Studies: GA-Based PLS Analysis of Calcium Channel Antagonists, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 306–310.

[54] C. L. Waller, M. P. Bradley, Development and Validation of a Novel Variable Selection Technique with Application to Multidimensional Quantitative Structure-Activity Relationship Studies, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 345–355.

[55] D. Rogers, A. J. Hopfinger, Application of Genetic Function Approximation to QSAR and QSPR, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 854–866.

[56] T. Kimura, K. Hasegawa, K. Funatsu, GA Strategy for Variable Selection in QSAR Studies: GA-Based Region Selection for CoMFA Modelling, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 276–282.

[57] K. Hasegawa, T. Kimura, K. Fumatsu, GA Strategy for Variable Selection in QSAR Studies: Enhancement of Comparative Molecular Binding Energy Analysis by GA-Based PLS Method, *Quant. Struct.-Act. Relat.* **1999**, *18*, 262–272.

[58] J. Zupan, M. Novic, Optimization of Structure Representation for QSAR Studies, *Anal. Chim. Acta* **1999**, *388*, 243–250.

[59] T. J. Hou, J. M. Wang, X. J. Xu, Applications of Genetic Algorithms on the Structure-Activity Correlation Study of a Group of Non-Nucleoside HIV-1 Inhibitors, *Chemom. Intell. Lab. Syst.* **1999**, *45*, 303–310.

[60] M. K. Hartnett, G. Lightbody, G. W. Irwin, Dynamic Inferential Estimation using Principal Components Regression (PCR), *Chemom. Intell. Lab. Syst.* **1998**, *40*, 215–224.

[61] B. Walczak, Outlier Detection in Multivariate Calibration, *Chemom. Intell. Lab. Syst.* **1995**, *28*, 259–272.

[62] B. Walczak, Outlier Detection in Bilinear Calibration, *Chemom. Intell. Lab. Syst.* **1995**, *29*, 63–73.

[63] C. Pizarro Millán, M. Forina, C. Casolino, R. Leardi, Extraction of Representative Subsets by Potential Functions Method and Genetic Algorithms, *Chemom. Intell. Lab. Syst.* **1998**, *40*, 33–52.

[64] Y. Tominaga, Representative Subset Selection using Genetic Algorithms, *Chemom. Intell. Lab. Syst.* **1998**, *43*, 157–163.

[65] Y. Tominaga, Data Structure Comparison using Box Counting Analysis, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 867–875.

[66] T. R. Kowar, Genetic Function Approximation Experimental Design (GFAXD): a New Method for Experimental Design, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 858–866.

[67] A. Broudiscou, R. Leardi, Phan-Tan-Luu, Genetic Algorithm as a Tool for Selection of D-Optimal Design, *Chemom. Intell. Lab. Syst.* **1996**, *35*, 105–116.

[68] C. B. Lucasius, A. D. Dane, G. Kateman, On K-Medoid Clustering of Large Data Sets with the Aid of a Genetic Algorithm: Background, Feasibility and Comparison, *Anal. Chim. Acta* **1993**, *282*, 647–669.

[69] J. H. Jiang, J. H. Wang, X. Chu, R.-Q. Yu, Clustering Data using a Modified Integer Genetic Algorithm (IGA), *Anal. Chim. Acta* **1997**, *354*, 263–274.

[70] G. J. McLachlan, T. Krishnan, *The EM Algorithm and Extensions*, John Wiley & Sons, Chichester, UK, **1997**.

[71] M. L. Raymer, P. C. Sanschagrin, W. F. Punch, S. Venkataraman, E. D. Goodman, L. A. Kuhn, Predicting Conserved Water-Mediated and Polar Ligand Interactions in Proteins Using a K-Nearest-Neighbours Genetic Algorithm, *J. Mol. Biol.* **1997**, *265*, 445–464.

[72] Z.-P. Chen, J.-H. Jiang, Y. Li, R.-Q. Yu, Nonlinear Mapping using Real-Values Genetic Algorithm, *Chemom. Intell. Lab. Syst.* **1999**, *45*, 409–418.

[73] T. H. Reijmers, R. Wehrens, F. D. Daeyaert, P. J. Lewi, L. M. C. Buydens, Using Genetic Algorithms for the Construction of Phylogenetic Trees. Application to G-Protein Coupled Receptor Species, *Biosystems* **1999**, *49*, 31–43.

[74] T. H. Reijmers, R. Wehrens, L. M. C. Buydens, Quality Criteria of Genetic Algorithms for the Construction of Phylogenetic Trees, *J. Comput. Chem.* **1999**, *20*, 867–876.

[75] J. W. Moon, Various Proofs of Cayley's Formula for Counting Trees, in F. Harary (Ed.), *A Seminar on Graph Theory*, Holt, Rinehart and Winston, New York, **1967**, pp. 70–78.

[76] R. Wehrens, C. Lucasius, L. Buydens, G. Kateman, Sequential Assignment of 2D NMR Spectra of Proteins using Genetic Algorithms, *J. Chem. Inf. Comp. Sci.* **1993**, *33*, 245–251.

[77] A. H. C. van Kampen, Z. Ramadan, M. Mulholland, D. B. Hibbert, L. M. C. Buydens, Learning Classification Rules from an Ion Chromatography Database using a Genetic-Based Classifier System, *Anal. Chim. Acta* **1997**, *344*, 1–15.

[78] A. H. C. van Kampen, L. M. C. Buydens, Reinvestigation of a Genetic-Based Classifier System: The Effectiveness of Recombination, *Comput. Chem.* **1997**, *21*, 153–160.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark
Copyright© WILEY-VCH Verlag GmbH, 2000

7 Chemical Structure Handling

Peter Willett

Abbreviations

AA	Atom assignment
CPU	Central processing unit
EINECS	European Inventory of Existing Chemical Substances
GA	Genetic algorithm
MCS	Maximal common substructure
MOS	Maximal overlap set
NMR	Nuclear magnetic resonance
NP	Non-polynomial
WDI	World Drugs Index

7.1 Introduction

Computer-based systems for the storage and retrieval of information pertaining to the two-dimensional (2-D) and three-dimensional (3-D) structures of chemical compounds play an increasingly important role in many areas of chemical research [1]. These systems contain machine-readable representations of large numbers of molecules, and provide a range of retrieval mechanisms by which users can access the stored structural data. In this chapter, we discuss some of the results obtained from a continuing study in Sheffield of the use of genetic algorithms (GAs) for processing the structure representations stored in such systems [2–9]. These studies commenced in 1992, when there was much less familiarity with GAs in the chemical community than is now the case, by looking at the processing of graph-based representations of 2-D chemical structure diagrams. This work was then extended to the processing of 3-D chemical structures, initially using distance-based representations but then encompassing more sophisticated descriptors that take account of the functionally important components of molecules.

The chapter is structured as follows. Section 7.2 provides a brief introduction to current methods for chemical structure handling, and section 7.3 discusses our initial studies into the use of GAs for the processing of 2-D chemical graphs. Section 7.4 describes GAs for the processing of 3-D chemical graphs and the following sections then describe two, very different ways of overlaying pairs of 3-D structures. We conclude by comparing the results obtained here with those obtained from other, nonevolutionary approaches to the processing of chemical structure data.

7.2 Representation and Searching of Chemical Structures

The principal method of representation for a 2-D chemical molecule is a *connection table*, which contains a list of all of the (usually nonhydrogen) atoms within a structure, together with bond information that describes the exact manner in which the individual atoms are linked together. A connection table is an example of a *graph*, a data structure that describes a set of objects, called *nodes* or *vertices*, and the relationships, called *edges* or *arcs*, that exist between pairs of these objects [10]. In the context of a 2-D chemical graph, the nodes denote the atoms and the edges the bonds linking pairs of atoms, thus providing an explicit description of the topology of a molecule. The presence of a query subgraph in another, larger graph can be detected using a *subgraph isomorphism* algorithm [11]: thus, if the graphs denote 2-D chemical moieties, a subgraph isomorphism algorithm provides a basis for *substructure* (or *atom-by-atom*) searching [1], which involves searching a database of molecules to find all those that contain a user-defined query substructure, irrespective of the environment in which the query substructure occurs.

Subgraph isomorphism is known to be NP-complete [12] and will thus be extremely time-consuming when applied in a database searching context (where a subgraph check must be carried out for each structure in the database). This has led to the development of algorithms to minimize the computational costs of substructure searching [13]. The principal heuristic that has been adopted is that of using an initial *screen search*, where a screen is a substructural feature, the presence of which is necessary, but not sufficient, for a molecule to contain the query substructure. The presence of the screens describing a query substructure provides an effective filter for ensuring that only a small fraction of an entire database is passed onto the detailed, atom-by-atom search.

Subgraph matching also forms the basis for substructure searching in 3-D since, as was first noted by Gund [14], it is possible to describe a rigid 3-D molecule by a graph in which the nodes and edges of a graph represent the atoms and interatomic distances, respectively. This is a very appropriate level of description since many of the *pharmacophores*, or *pharmacophoric patterns*, that chemists wish to search for are framed in terms of a set of atoms (or pharmacophore points such as hydrogen-bond donors, hydrogen-bond acceptors, or ring centroids), together with some or all of the associated interatomic distances [15]. Given that a 3-D structure can be represented by a graph, the presence or absence of a pharmacophoric pattern can be confirmed by means of a subgraph isomorphism procedure in which the edges in a database structure and a query substructure are matched if they denote the same interatomic distance (to within any user-specified tolerance, e.g., ±0.5 Å). A subgraph isomorphism search of a 3-D chemical graph is often referred to as *geometric search*, thus differentiating it from the atom-by-atom search that characterizes the processing of 2-D chemical graphs. A screening stage is again invoked to filter out molecules that do not need to undergo the geometric search, with the screens here being based on simple geometric characteristics, most commonly pairs of atoms together with an associated interatomic distance range [16]. Similar approaches can be used for the representation and searching of flexible 3-D structures, as discussed in section 7.4.1.

Thus far we have considered only the use of subgraph isomorphism algorithms for processing chemical graphs, but there have also been many chemical applications of *maximal common subgraph isomorphism* algorithms. Given two graphs, a maximal common subgraph isomorphism algorithm will identify the largest subgraph common to the two graphs that are being compared, where "largest" is defined in terms of the number of matching nodes and/or edges. Chemical applications have focused upon the ability of such algorithms to identify the degree of structural overlap between pairs of chemical molecules, using both 2-D and 3-D structure representations (see, e.g., [17, 18]). The resulting *maximal common substructure* (MCS) can be used as a measure of intermolecular structural similarity. Such measures provide the basis for *similarity searching*, that is, scanning a database to find those molecules, the *nearest neighbors*, that are most similar to a user-defined *target structure*; other types of similarity measure are based on the screens that are used for 2-D substructure searching [19] and on the molecular fields that are discussed in section 7.5.

7.3 Processing of 2-D Chemical Graphs

Our very first study of the use of a GA for chemical structure handling involved the development of a procedure for 2-D substructure searching [3]. This algorithm involved generating mappings of nodes between a query substructure and a database structure so as to maximize the number of matching edges, that is, the number of matching chemical bonds. However, this proved to be an inappropriate choice of application for two reasons. First, while subgraph isomorphism detection is time-consuming, there are efficient algorithms available for the processing of chemical graphs (such as the algorithm due to Ullmann [20] that was used as a comparison for our GA); second, while the GA was able to find matches for a query substructure relatively quickly, its nondeterministic nature meant that it was not possible to conclude unequivocally that a query substructure was absent from a database structure, even if it was run many times with many generations in each run. Taken together, these characteristics suggest that this particular type of graph-matching application is not suitable for processing by a GA; however, we found that the chromosome representation and genetic operators that had been developed could also be used for a much more promising application, viz. the computation of the maximum overlap set (MOS), as described below.

Brown et al. [21] have suggested the use of a *hyperstructure*, as a way of increasing the efficiency of 2-D substructure searching systems. A hyperstructure is a pseudomolecule that is formed by the superimposition of sets of molecules in such a way that the structural overlap between molecules is maximized. A hyperstructure thus encodes a set of chemical structures with minimal redundancy; not only does this minimize the storage requirements but it also has the potential for improving substructure-searching times, since common features will need to be searched only once (rather than once for each structure in which they occur, as in a conventional substructure search). A hyperstructure is formed from a set of structures as follows: the first structure is the base hyperstructure, and subsequent structures have atom nodes mapped to hyperstructure nodes of the same elemental type, in such a way that there is a maximum overlap between structure edges and hyperstructure edges of the same type. A new hyperstructure node is created if there is no hyperstructure node to which a structure node can be mapped, and new hyperstructure edges are created as needed in a similar manner. The MOS problem involves generating the mapping that minimizes structural redundancy and is closely related to the better known MCS problem mentioned previously. GA-based algorithms for MCS identification have been described by Fontain [22], in what was probably the first reported application of a GA to a chemical structure handling problem, and by Wagener and Gasteiger [23].

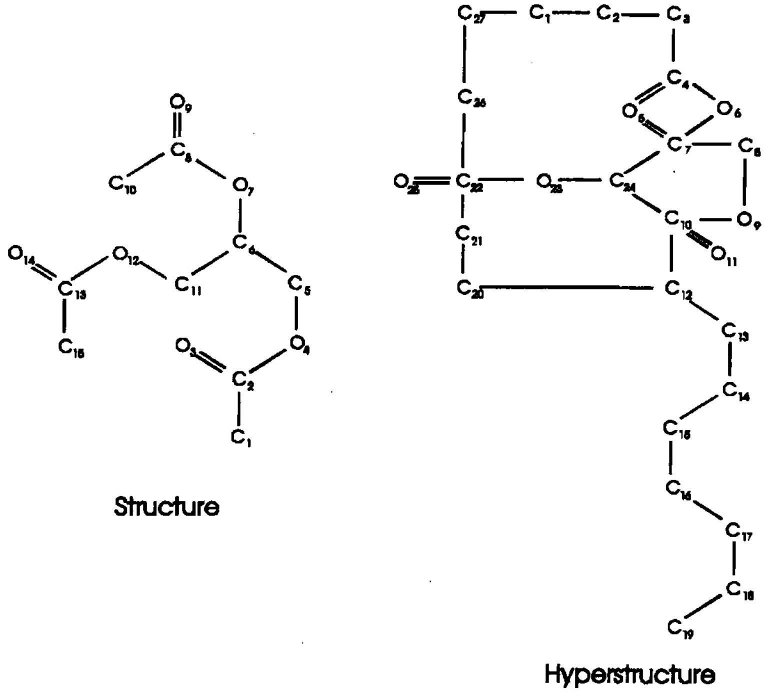


Figure 1. Mapping of a structure to a hyperstructure created by the genetic algorithm, as described by Brown et al. [21].

We have developed a GA to solve (or at least to investigate) the MOS problem. Its workings are illustrated using the example shown in Fig. 1, where the hyperstructure had been created from the superposition of five previous structures. Candidate mappings between the graphs representing a structure and the hyperstructure are encoded in integer-string chromosomes. The position of an integer in a string encodes the value of a query node, while the integer value at that position is the structure node to which that query node maps. For example, the mapping:

{1→27},{2→26},{3→5},{4→6},{5→16},{6→3},{7→11},{8→12},{9→9},
{10→2},{11→24},{12→23},{13→22},{14→25},{15→21}}

encodes as

27 26 5 6 16 3 11 12 9 2 24 23 22 25 21

The fitness of each such chromosome is the number of query edges that map to structure edges (with the same label) under the decoded mapping.

Three genetic operators – mutation, uniform crossover and node-based crossover – were employed, the operands being chosen by fitness-based, roulette wheel parent selection with user-defined weights. The mutation operator takes one parent chromosome and produces one child chromosome, by randomly changing a structure-to-hyperstructure node mapping at a randomly chosen position in the chromosome. Uniform crossover was used, with the reordering operation used in partially matched crossover [24] being applied following crossover. Finally, node-based crossover is a crossover operation, specifically designed for this application, that explicitly combines the most fit parts of the parental chromosomes in the children, as described by Brown et al. [3].

The algorithm was tested on 10 794 structures from the EINECS (European Inventory of Existing Chemical Substances) database by generating a *global hyperstructure*; that is, a hyperstructure that contained all of the individual molecules in the data set. The data set was also divided into clusters of similar structures, using the Jarvis-Patrick clustering method [25], and a hyperstructure was then produced for each cluster. The performance of the GA in generating hyperstructures was compared to the *atom-assignment* (AA) algorithm of Brown et al. [21], which generates hyperstructures by merely assigning structure nodes to the first available hyperstructure node, considering only the node type without any regard for edge matching. The results are shown in Table 1. Times are in CPU seconds on an ESV 3, and there was a total of 203 757 edges in the individual molecules comprising the data set. The results show the GA giving a significant improvement in the quality of the matching. This increased effectiveness is at the expense of CPU time, with the simple AA algorithm being two or three orders of magnitude faster; however, it must be remembered that hyperstructure generation is a "one-off" operation that needs to be carried out only when a database is created. The improvement in compactness is quite marked in the global case, where the resulting hyperstructure has a much smaller number of edges than does the AA-based hyperstructure.

Table 1. Comparison of edge compression and execution times (in CPU seconds) for the creation of hyperstructures using the genetic algorithm (GA) and atom assignment (AA) methods [2, 3].

Hyperstructure	Edges		Run-time	
	GA	AA	GA	AA
Global	1966	3813	332 200	220
Clustered	83 245	96 173	70 770	211

The primary rationale for the generation of hyperstructures is to improve the speed of substructure searching. The AA and GA methods were used to generate global hyperstructures and hyperstructures for each of the clusters that contained more than five individual compounds, and 27 query substructures were then searched against these two sets of hyperstructures. The relative performance of the two approaches is defined by the speed-up, that is, the ratio of the AA search time to the GA search time. The results show a median speed-up of 2.1 for cluster hyperstructure search and of 1.3 for global hyperstructure search, with peak speed-ups of 4.9 and 3.3, respectively. A one-tailed sign test shows that the GA search is significantly faster in both cases ($P < 0.001$ for cluster hyperstructure search and $P < 0.005$ for global hyperstructure search), thus validating the effectiveness of our GA for the MOS problem [2].

7.4 Processing of 3-D Chemical Graphs

As noted in section 7.2, the geometry of a rigid 3-D molecule can be encoded by a graph in which the nodes are the atoms and the edges are the interatomic distances, and such graphs can be processed using subgraph and maximal common subgraph isomorphism procedures analogous to those employed for processing 2-D chemical graphs. For example, searches for pharmacophores in databases of rigid 3-D structures are carried out using screening and subgraph isomorphism techniques derived from those used for 2-D substructure searching [16], and maximal common subgraph isomorphism algorithms can be used for the automatic identification of pharmacophoric patterns in sets of bioactive molecules [18, 26]. In this section we discuss two GAs that we have developed for processing 3-D chemical graphs.

7.4.1 Flexible 3-D Substructure Searching

The screening and subgraph isomorphism procedures that are used for 2-D and for rigid 3-D substructure searching can be further extended to encompass, in part at least, the representation and searching of flexible 3-D structures, where a molecule can adopt some, or many, different conformations by rotating around one or more of the rotatable bonds present in a molecule. One approach is simply to generate a number of low-energy conformations and then to search each of these as if it was a distinct rigid structure [16]. An alternative approach, and the one discussed here (and in more detail by Clark et al. [4, 27]), adopts a graph representation in which the nodes are again the atoms of a molecule; however, rather than representing just a fixed distance, each edge describes a range of distances, specifically the range spanned by the maximum and the minimum interatomic separations that are possible for a given pair of atoms, these separations typically being calculated using techniques derived from distance geometry [28]. It is simple to extend a subgraph isomorphism algorithm so that it can check whether the individual bounded distances in such a distance-range graph are consistent with the distance bounds that have been specified in a query pharmacophore, that is, to carry out a geometric search [27]. However, the distance geometry techniques that are normally employed to calculate such distance ranges cannot take account of all of the strong interdependencies that exist between the sets of distance ranges in a flexible database structure, and thus the hits from the geometric search must be checked in a further, *conformational search.* The GA we have developed for this purpose is described below.

The chromosome for a molecule contains one byte for each of the rotatable bonds in a molecule, with each byte encoding an angle of rotation about one of the flexible bonds as an unsigned Gray-coded integer [24]; thus, given an 8-bit byte, the search resolution is 1.4° (that is, 360/256). When a chromosome is decoded, the rotations are applied to the conformation of the molecule that provides the input to the GA, thus defining a new conformation that is then checked to see if it is a match for the query pattern. Specifically, the GA uses a penalty function to rank individual conformations in order of increasing fitness, with the penalty function comprising two parts as discussed below.

The distance range between each pair of atoms in the pharmacophore was determined for a particular conformation of a database structure. If the distance lay within the allowed bound, then the pharmacophore-match penalty was set to zero; otherwise, a penalty value equal to the modulus of the difference between the distance and the closest bound, in Å, was assigned. The penalty values were then summed over all distances in the structure. The molecular conformations retrieved in a flexible 3-D substructure search must not only match the distance constraints but must also be of reasonable energy. An energy penalty was hence calculated as the scaled difference between the energy of a conformation generated during a search and the energy of the reference conformation stored in the database that is being searched, these energies being calculated using the Lennard-Jones 6–12 potential in the SYBYL forcefield [29]. The overall penalty was then a weighted sum of these two contributions. The GA uses parent selection based on linear normalized fitness values, and the two standard genetic operators (bit mutation and one-point crossover) employ roulette wheel parent selection based on these values. The GA is run until a zero penalty value has been obtained, to (near-)convergence or for a user-defined number of iterations.

The GA was used as part of a flexible substructure-searching system, in which a set of eight pharmacophore queries was searched against a file of 1538 3-D structures. Screen and geometric searches were carried out for each of these queries using the procedures described by Clark et al. [27] and the resulting potential hits then passed on for the final conformational search using several different searching methods: distance geometry, systematic search, the GA, and directed tweak [4]. Of these, distance geometry was found to be far too slow, and attention hence focused on the other three methods, the search results for which are listed in Table 2 (where the results for systematic search were the best of those obtained with a range of different torsional increments for the routine included in the SYBYL package [31]). The GA was run ten times for each pattern, and the two sets of figures listed are for the union of all of these ten runs and the mean number of hits when averaged over the ten runs. It will be seen that the directed tweak method gave the largest number of hits for the three methods, and its performance was confirmed in subsequent searches of a set of 9886 structures, where it was found both to identify more hits and to be noticeably faster than the GA. It was thus concluded that directed tweak was the method of choice for conformational searching, and it has subsequently been adopted for the flexible pharmacophoric pattern matching systems produced by MDL Information Systems [32] and by Tripos Inc. [33]. That said, directed tweak searches torsional space using a pseudoenergy function that involves the sum of the squares of the deviations of the interatomic distances in a database structure from the input distance constraints as specified in the query pharmacophore. This is very efficient for distance-based queries, but substantial modifications to the algorithm are required if other, nondistance constraints (such as valence or torsion angles, or included or excluded volumes) are to be included in a pharmacophore query; conversely, it is easy to encompass such types of constraint in the GA merely by specifying an appropriate penalty function.

Table 2. Percentage of hits from the geometric search outputs that were found to contain the query pharmacophore pattern in a comparison of different conformational searching algorithms for flexible pharmacophoric pattern matching [4].

Pattern	Systematic	Genetic algorithm		Directed
	search	All hits	Hits/run	tweak
1	0.0	0.0	0.0	0.0
2	33.3	43.5	38.8	44.6
3	0.0	0.0	0.0	0.0
4	15.1	32.9	27.1	30.1
5	21.6	34.2	15.2	38.4
6	30.1	37.0	34.6	38.4
7	17.4	25.6	7.7	34.9
8	12.7	15.3	11.8	15.3
Median	16.3	29.3	13.5	32.5
Mean	16.1	23.6	16.9	25.2

7.4.2 Identification of Common Structural Features in Sets of Ligands

We have noted previously that maximal common subgraph isomorphism algorithms can be used for pharmacophore mapping, that is, for the automatic identification of pharmacophoric patterns in sets of bioactive molecules. The results of these tests can then be used to validate, or to modify, the pharmacophore. Programs such as DISCO (DIStance COmparisons) [26] use a clique-detection algorithm to identify the MCS, that is, the largest pattern of pharmacophore features (such as donors or ring-centroids) in 3-D that is common to all of the input molecules. An MCS algorithm provides an obvious basis for pharmacophore identification, but suffers from two limitations. First, the requirement that the same set of features occurs in the same geometric arrangement in *all* of the molecules can lead to MCSs that contain only a pair of features or that involve very large interfeature distance range tolerances. Second, and more generally, the use of the MCS involves the inherent assumption that there is a single, common pharmacophore that is responsible for the observed activity, and this assumption may not be correct. We have sought to overcome these problems by developing a GA called MPHIL (Mapping PHarmacophores In Ligands) to identify a minimal set of features that suffices to cover all of the molecules; an alternative approach to this problem has been described by Barnum et al. [34].

MPHIL identifies a *K-point site*, a pattern consisting of K point-like features and the associated interfeature distances, with which all of the input molecules have at least some minimal number of features in common, subject to the constraint that K should be as small as possible. The input to MPHIL consists of: the 3-D co-ordinates of the features in each of the N bioactive molecules (typically N is in the range 5 to 20) that are to be analyzed, from which the interpoint distances are calculated; the minimal number of points, m, that each molecule must have in common with the K-point site; and the interpoint distance tolerances that must be met if two 3-D patterns are to be regarded as geometrically equivalent. The process involves the identification of a subset of m points from each of the N molecules, giving a total of Nm points. The geometry of each m-point subset may allow the superimposition of points such that one or more points may be common to more than one molecule when the user-defined interpoint distance tolerances are taken into account, this leading to a reduction in the total number of unique points, K. The optimum situation is the case where all molecules contain the same pattern of m points, in which case $K = m$; this corresponds to the identification of an m-point substructure that is common to all of the molecules, as in existing programs for pharmacophore mapping. More generally, $m \leq K \leq Nm$, and MPHIL tries to ensure that K is as small as possible, so that the resulting K-point site has some similarities to the hyperstructure representations that have been discussed in section 7.3 for reducing the computational requirements of 2-D substructure searching.

The identification of the smallest K-point site is an extremely demanding combinatorial problem: if each of the N molecules contains P points, then there are no less than

$$\left[\frac{P!}{m!(P-m)!}\right]^N$$

possible combinations of m-point subsets that may need to be considered. The matching of such subsets is effected in MPHIL by means of an MCS algorithm based on clique detection [18]. This algorithm identifies cliques of size m that are common to a number of molecules and that may thus be contained within the K-point site; the principal novel idea in MPHIL is the use of a GA to generate subsets that can be submitted to these clique-based matching procedures so as to minimize the final value of K. Specifically, our GA uses a chromosome that encodes the 3-D co-ordinates of the K points comprising the current K-point site. For each molecule, a size-m clique must be found in both the molecule and the K-point site represented by the chromosome. This requires N invocations (one for each molecule) of the clique-detection algorithm for each chromosome in each iteration of the GA, and our initial experiments showed this to be far too time-consuming for practical purposes, given the huge number of m-point subsets that would need to be considered. Accordingly, an initial, precursor GA (hereafter referred to as GA-1), has been developed that reduces the computational requirements of the second, clique-detection stage (hereafter referred to as GA-2).

GA-1 is a standard steady-state-with-no-duplicates algorithm that identifies a combination of m points from each molecule, such that the resulting set of points can be maximally superimposed (this corresponding to minimizing the number of distinct points in the final K-point site). Each chromosome in GA-1 contains N sets, one for each of the N input molecules, of m integers, each integer indexing the 3-D co-ordinates of a single point in one of the molecules. An initial population of chromosomes is generated randomly subject to the constraints that the same point does not appear more than once in any set and that no duplicate chromosomes occur. The population of chromosomes is processed using standard one-point crossover, with mutation involving the replacement of one point in a chromosome with a new point from the same molecule. The aim of GA-1 is to identify a maximal overlap of the points of the chromosomes, and the fitness function is thus an inverse measure of how badly the N sets of points overlap, as determined by the interatomic distances calculated from the 3-D co-ordinates of each point. The algorithm is typically run for 100 000 generations, which is quite sufficient to generate a suitable population for GA-2. This second algorithm then assembles these m-point subsets by means of the clique-detection procedure to give a K-point site that is as compact as possible.

Each chromosome for GA-2 contains the 3-D co-ordinates of the K-point site encoded in a high-fitness chromosome output by GA-1, together with the point descriptors and the interpoint distances for that site. These distances are recalculated each time that a new chromosome is generated since mutation may move some of the points (as described below). Each chromosome is also assigned an initial tolerance level which is used to match distances in the clique-detection stage. This level is initially the user-defined tolerance, but may be systematically increased in cases where the K-point site from GA-1 requires a slightly larger tolerance for a fit. The initial fitness function (see below) for each chromosome is also calculated at this stage.

A standard one-point crossover technique is used in which co-ordinates following the crossover point are interchanged between two parent chromosomes to produce children. Mutation occurs in several different forms as follows: a new point is selected randomly from one of the molecules and added to the chromosome; a random point is removed from the chromosome; a random point is moved (or creeps) in a random direction by a user-defined distance; the tolerance for that chromosome is reduced by some user-defined value; the mid-point between two points is calculated and a new point is created at this new position, with the original two points being removed from the chromosome. Clique detection is carried out using the algorithm described by Bron and Kerbosch [35]. For each molecule-chromosome match a check is made to see whether any size-m cliques exist, that is, whether that molecule and the K-point site encoded in that chromosome have a set of m points in common. If no size-m cliques are found, then the new chromosome is discarded. Alternatively, if the new chromosome contains a common clique for each chromosome-molecule comparison, then it replaces a less-fit chromosome in the population. The fitness function is given by the inverse of the sum of the number of chromosome points, K, and the current chromosome tolerance, T; thus GA-2 operates by seeking K-point sites for which both K and T are as small as possible.

The use of the program is illustrated with a data set of 19 angiotensin-converting enzyme and thermolysin inhibitors, possessing between seven and 12 points (specifically the constituent heteroatoms), selected from a set of 58 described by Scott et al. [36]; results with other data sets are reported by Holliday and Willett [6]. The 3-D structures of these molecules were obtained by running the CONCORD structure generation program, followed by minimization using the Tripos force field in SYBYL 6.3 [31]. Sample runs are detailed in Table 3, which describes the user-defined *m* values and tolerances that were employed and in which the run-times are in CPU minutes for C programs running on a DEC Alpha 3000 Unix workstation. With *m* = 3, all 19 compounds contain a common substructure composed of the two oxygens in a carboxylic acid group and an amine. As a result, the same 3-point subset is repeated throughout all of the compounds giving a *K* of 3. This solution, together with the edge tolerances, is shown in Fig. 2. With *m* = 4, a common 4-point substructure is found in 18 of the compounds, this being made up of the carboxylic acid, the amine, and a further nitrogen in close proximity to the 3-point subset. One compound does not contain this second nitrogen, however, and a further point is required to cover this compound. The resulting 5-point solution is shown in Fig. 3a. The absence of a tolerance on one edge (denoted by the broken line) in the solution is due to the fact that this edge is not contained within any of the individual 4-point subsets, one example of which is shown in Fig. 3b.

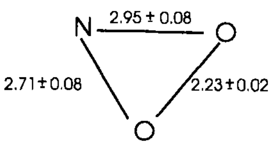


Figure 2. Solution for MPHIL with a data set of 19 angiotensin-converting enzyme and thermolysin inhibitors with *m* = 3.

Table 3. Use of MPHIL on a set of 19 angiotensin-converting enzyme and thermolysin inhibitors: the times (columns 3 and 4) are in CPU minutes on a DEC Alpha 3000 Unix workstation, and the tolerances (columns 2 and 7) in Å [6].

m	Initial tolerance	GA-1 time	GA-2 time	GA-2 iterations	*K*	Final tolerance
3	0.50	3.2	64.7	2044	3	0.10
3	0.50	3.2	66.5	2369	3	0.20
4	0.50	4.4	40.4	1344	5	0.35
4	0.50	4.7	68.2	2382	5	0.30

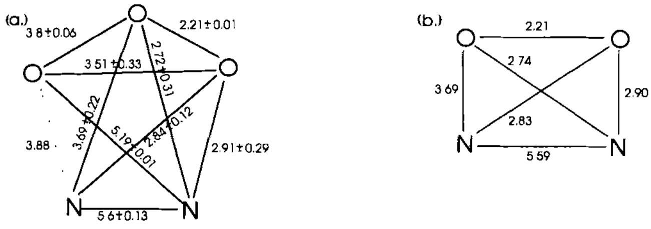


Figure 3. Solution (a) and example 4-point subset (b) for MPHIL on a data set of 19 angiotensin-converting enzyme and thermolysin inhibitors with *m* = 4.

These results (and many others discussed by Holliday and Willett [6]) demonstrate the ability of MPHIL to find the smallest pattern of points in 3-D space that has at least some user-defined number of points in common with each of a set of molecules. The program is also quite efficient: a run typically takes some tens of CPU minutes, and MPHIL is thus sufficiently rapid to enable several different solutions to be obtained without undue effort. That said, there are obvious limitations, in that the most of the experiments thus far have considered only patterns of heteroatoms (rather than more generic pharmacophore-point definitions such as hydrophobic region or hydrogen-bond donor), and in that the program considers only the limiting case in which all of the molecules are considered to be rigid, and represented by a single low-energy conformation.

7.5 Field-Based Similarity Searching

Systems for similarity searching have been introduced in section 7.2. At the heart of any such system is the measure that is used to quantify the degree of structural resemblance between the target structure and each of the database structures. The most common of these measures are based on comparing the fragment screens that are normally used for 2-D substructure searching, so that two molecules are judged as being similar if they have a large number of screens, and hence substructural fragments, in common [19]. Although widely used, fragment-based measures have several limitations [37], most noticeably their focus on topological similarities that take no account of the electrostatic, steric and hydrophobic fields that lie at the heart of modern approaches to the correlation of molecular structure and biological activity [38]. This limitation led to a program of work in Sheffield to develop methods for field-based similarity searching that would facilitate the discovery of nonobvious *bioisosteres*, that is, molecules with different structures but exhibiting the same biological activity. Our initial studies adopted a graph-based approach to the representation and searching of molecular field data [39], but this was supplanted by the GA-based approach that is discussed below.

Field-based similarities may be calculated using methods first developed by Carbó et al. [40]. Given a molecular property P that can be calculated at any point around a molecule, a field may be created around the molecule by integrating P with respect to volume, and the similarity between a pair of molecules may then be calculated based on the overlap of the corresponding fields. This provides an elegant and natural way of quantifying molecular similarities, but it is very time-consuming. Good et al. [41] have described a Gaussian-based procedure that allows the similarities to be calculated much more rapidly and with little loss of accuracy, and we have used this approach in all of our studies. Specifically, we have developed a GA to align the fields representing two molecules so as to maximize the degree of overlap using the field-based measure of similarity normally referred to as the Carbó index [40].

The GA is extremely simple. The target structure for the similarity search is assumed to be held stationary, and a chromosome in the GA encodes the rotations and translations that are to be applied to the current database structure to align it with the target structure. The fitness function is the value of the Gaussian similarity coefficient resulting from the particular alignment encoded within a chromosome, and the GA hence moves the database structure relative to the target structure so as to maximize the calculated similarity. If no account is taken of conformational flexibility then just the rigid-body rotations are encoded in a chromosome; alternatively, if the molecules are allowed to flex, then the chromosome additionally encodes the torsional rotations [8]. However, we have found that the inclusion of flexibility information does little, if anything, to increase the effectiveness of the algorithm unless it is run for a totally unacceptable amount of time, and the experiments reported below hence consider only the rigid version of the GA.

The algorithm was initially developed to align molecules on the basis of the molecular electrostatic potential [9], but it has since been modified so that it can additionally encompass hydrophobic and steric fields [5]; the resulting program, called FBSS (Field-Based Similarity Searching) thus permits the calculation of all three types of similarity by the same basic procedure. Alignments may be made based on a single field-type, or on any combination of the three types of field; here, we report the results when all three types were combined. During the execution of the GA, each alignment of a target structure and a database structure is used to calculate each of the three individual types of field-based similarity and then the fitness for the chromosome encoding that alignment is the mean of the three resulting Gaussian similarity values. No types of weighting or standardization are applied to the individual similarity measures, so that all three types of field are assumed to contribute equally to the overall score of an individual database structure.

The effectiveness of the GA for similarity searching was tested using a 10% sample of the 1995 edition of the *World Drugs Index* (WDI), which contains the 2-D structures and activity classes for about 41 000 compounds for which qualitative biological activity data are available [42]. This subfile was searched by FBSS using the 10 target structures employed in a study of property-based similarity measures by Kearsley et al. [43]. Each of these target structures was used as the basis for a field-based similarity search (which, as noted previously, involved all three types of field) and for a conventional 2-D, fragment-based similarity search (which was implemented using the search routines in the UNITY chemical information management system [31]). In both cases, the search performance was measured by the number of actives present in the top-ranked 300 nearest neighbors; further results are presented and discussed by Drayton et al. [5].

Table 4. Analysis of the active molecules retrieved in the top-300 rank positions by the 2-D and GA-based similarity measures, using ten different target molecules. The left-hand part of the table contains the number of actives retrieved, and the right-hand part of the table contains the diversity of the retrieved active molecules [5].

	Numbers of actives		Calculated diversities	
	2-D	GA	2-D	GA
Apomorphine	60	49	0.32	0.56
Captopril	68	26	0.41	0.70
Cycliramine	94	120	0.59	0.60
Diazepam	46	37	0.51	0.55
Diethylstilbestrol	100	100	0.38	0.35
Fenoterol	58	33	0.42	0.44
Gaboxadol	7	14	0.49	0.52
Morphine	46	30	0.31	0.34
RS86	26	14	0.62	0.70
Serotonin	21	22	0.33	0.52
Mean	52.6	44.5	0.44	0.53

The left-hand portion of Table 4 shows the number of top-ranked actives returned by the two types of similarity measure, where it will be seen that the 2-D measure retrieves more active molecules. This finding might appear counterintuitive, given the importance of molecular fields in determining biological activities [38], but account must also be taken of the types of molecule in the database used here. Many of the molecules with a given activity class in the WDI are topologically very similar, including many close analogs and so-called "me-too" drugs. These molecules are very easy to retrieve using a similarity measure that explicitly encodes topological information; however, they might well not represent the full range of bioactive structural types, whereas such diverse sets of molecules might well be retrieved by measures that do not focus on the specific patterns of atoms and bonds in molecules. An inspection of the various search outputs certainly suggests that the 2-D measure results in less diverse sets of nearest neighbors than do the other measures, and we have sought to quantify this finding by means of a diversity index based on the fragment bit-strings used for the 2-D similarity search [44]. This index was calculated for the sets of active structures retrieved in each of the searches and the results are listed in the right-hand portion of Table 4, where it will be seen that the GA measure results in more diverse sets of compounds (that is, larger values of the diversity index) than does the established 2-D measure; similar results are obtained if all of the nearest neighbors are considered, rather than just the active nearest neighbors as here. This indicates that while the GA measure generally finds fewer actives, those that it does find are better able to suggest novel structural classes that are additional to the close analogs usually retrieved by similarity searches; thus, the main use of the approach may be to act as an "ideas generator", rather than as a high-precision search tool as is the case with 2-D similarity searching. The different types of output suggest that the two types of similarity measure are complementary in nature, as further demonstrated by the fact that each of them retrieved large numbers of active molecules that were not retrieved by the other similarity measure.

The results presented here and elsewhere [5] suggests that our GA provides an effective way of identifying bioisosteres in chemical databases that cannot be retrieved using conventional 2-D similarity measures; current work in Sheffield using the BIOSTER database, which contains pairs of known bioisosteres [45], provides further evidence to support this conclusion [46].

7.6 Generation of Molecular Alignments

The MPHIL and FBSS programs described above involve the alignment of rigid 3-D chemical structures; in both cases, the principal aim of the work has been to identify an efficient matching criterion that permits large numbers of such alignments to be evaluated (either in the repeated invocations of the clique-detection procedure in MPHIL, or in the matching of the target structure with each of the database structures in FBSS). In this section, we describe a further alignment procedure called GASP (Genetic Algorithm Superimposition Program) that has been designed for the detailed processing of sets of flexible 3-D structures, with the aim of providing an effective method for the generation of putative pharmacophoric patterns. The fitness function that has been developed is far more complex than those discussed previously, and we hence focus upon this aspect of the program, rather than upon the results obtained when it is implemented (as with some of the previous GAs in this chapter).

Given a set of N bioactive molecules for which the common pharmacophoric pattern is required, GASP selects one of them as a *base molecule*, to which the other molecules are fitted. A chromosome in GASP encodes two types of information: binary strings that encode angles of rotation about the rotatable bonds in all of the molecules; and integer strings that map pharmacophore *features* (viz. hydrogen-bond donor protons, acceptor lone-pairs, and ring centers) in the base molecule to corresponding features in each of the other molecules. Molecules are then overlaid onto the base molecule in such a way that as many as possible of the structural equivalences suggested by the mapping are formed. The fitness of a decoded chromosome is then a combination of the number and similarity of overlaid features, the volume integral of the overlay and the van der Waals energy of the molecular conformations. The genetic operators are used to drive the algorithm to that molecular superimposition that maximizes the value of this fitness function, corresponding to the best possible structural overlay of a series of active molecules that are presumed to bind to a biological receptor in a similar fashion. Having given a brief introduction to the overall structure of GASP, we now present the principal components of the program; a more detailed account is provided by Jones et al. [7].

Each of the N structures in a data set is analyzed to determine the features that are present, and the base molecule is then that molecule with the smallest number of features. A chromosome in GASP contains a total of $2N$-1 strings, these comprising: N Gray-coded binary strings, each encoding conformational information for one structure with each byte encoding an angle of rotation about a rotatable bond; and N-1 integer strings, each encoding a mapping between features in a molecule (other than the base molecule) to features, of the same type, in the base molecule (e.g., an acceptor lone pair can only be mapped to a comparable lone pair in the base molecule). By associating features in each molecule to base-molecule features, these mappings suggest possible pharmacophoric points: on decoding the chromosome, GASP uses a least-squares fitting routine to attempt to form as many points as possible.

GASP uses two genetic operators: the crossover operator performs two-point crossover on the integer strings, using the PMX crossover operator (including the duplicate removal stage) that is described by Goldberg [24], and traditional one-point crossover on the binary strings; and the mutation operator performs binary-string mutation on binary strings and integer-string mutation on integer strings, using the mutation operators described by Davis [47] and by Brown et al. [3], respectively.

The fitness function lies at the heart of any GA, and the evaluation of the one developed for GASP involves no less than six distinct stages, as summarized in Table 5. Each Gray-coded byte in the binary string is decoded to give an integer value between 0 and 255, and this denotes an angle of rotation for the appropriate rotatable bond. The randomized 3-D co-ordinates for the molecule are used as a starting configuration, and bond rotations are successively applied around the rotatable bonds to generate a new set of co-ordinates for the molecule. The resulting conformations are then passed to the least-squares fitting procedure. Here, a *virtual point* is created for each donor proton and acceptor lone-pair in a molecule at a distance of 2.9 Å from the donor or acceptor, in the direction of the hydrogen or lone pair. Virtual points are also created at the center of each ring. Decoding the chromosome gives a set of pairs of virtual points, one from the current molecule and one from the base molecule. A Procrustes rotation yields a geometric transformation that minimizes the least-squared distance between all of the virtual points in a molecule and the corresponding base-molecule virtual points.

Table 5. Components of the fitness function in GASP [7].

1. A separate conformation is generated for each molecule by applying the bond rotations encoded in the appropriate binary string.

2. Each molecule is superimposed on top of the base molecule using a transformation obtained from a least-squares procedure that fits to the mapping encoded in the appropriate integer string.

3. A van der Waals energy is obtained for the internal steric energy of each molecule.

4. A volume integral is obtained for the common volume between each molecule and the base molecule.

5. A similarity score is generated by determining which features were common to all molecules in the current overlay.

6. A final fitness score is generated by performing a weighted sum on the terms calculated in the three previous stages.

The van der Waals energy is calculated in the third stage of Table 5. The internal steric energy for each molecule is calculated using a Lennard-Jones 6–12 potential with parameters taken from the SYBYL force field [29]. The steric energy of a molecular conformation is expressed as the difference between the 6–12 energy of the conformation and the 6–12 energy of the original input molecular conformation, prior to the randomization of molecular co-ordinates. In order that the van der Waals energy term in the final fitness score be independent of the number of molecules in the overlay, the mean 6-12 energy *per* molecule is determined.

In order to predict which portions of the actives are in contact with the active site, GASP should ideally determine common molecular surface areas as a measure of similarity in the fourth stage of Table 5. However, such a calculation is extremely time-consuming, and pairwise common volumes are thus determined between the base molecule and each of the other molecules. The volume calculation is approximated by treating atoms as spheres and summing the overlay between pairs of spheres in the two different molecules, with the total volume integral between two molecules being determined by summing all the individual terms from atomic common volumes. Once each of the volume integrals has been calculated, the mean volume integral *per* molecule with the base molecule is determined.

The calculation of the similarity score in stage 5 of Table 5 is the most complex part of the fitness function, and is the weighted sum of three terms: a score for the degree of similarity in position, orientation and type between hydrogen-bond donors in the base molecule and hydrogen-bond donors in the other molecules; a score derived from comparing hydrogen-bond acceptors; and a score that results from comparing the positions and the orientations of aromatic rings. To determine how similar two hydrogen-bond donor or acceptor types are, GASP uses methods developed previously in the GOLD program for flexible ligand docking [48]. GASP uses the GOLD procedure for measuring the similarity between hydrogen-bond types *a* and *b*, where *a* and *b* are either both donor types or both acceptor types and where one is in a molecule and the other in the base molecule. The third term in the similarity score requires the use of normals to aromatic rings, with the similarity between a pair of rings, one in a molecule and one in the base molecule, being calculated from the product of the mean normals. The final fitness score is then a weighted sum of the volume integral, similarity and van der Waals energy scores, and the output of the GA is a set of conformations that maximize the goodness of fit among the members of the data set.

GASP has been applied successfully to many different sets of structures, as detailed by Jones et al. [7], and is now distributed commercially by Tripos Inc. [31].

7.7 Conclusions

The previous sections have described the GAs that have been developed in our laboratory for a range of chemical structure handling applications; other applications include the GOLD program mentioned previously [48], the design of combinatorial libraries [49] and the analysis of protein 2-D NMR data [50], *inter alia*. With the obvious exception of the initial, 2-D subgraph isomorphism application, we have found that GA-based approaches provide an effective way of handling the inherently combinatorial nature of many problems in chemical structure handling. Moreover, this degree of success is achieved with programs that are, in many cases, quite simple in concept but that are still able to produce satisfactory solutions even with the extremely large sizes of some of the search spaces considered here; the MPHIL program discussed in section 4.2 provides a good example of this combination of characteristics. Problems can obviously arise from the nondeterministic nature of a GA, but we have not found this generally to be a limitation given the sorts of task that are being considered. For example, two runs using FBSS are most unlikely to give exactly the same ranking of a set of compounds in decreasing order of similarity with the target structure; however, this is not a serious problem if, as suggested in Section 7.5, the principal use of the program is to suggest novel types of structure that are additional to those retrieved by existing types of retrieval mechanism. Again, Jones et al. report extensive experiments which demonstrate the general robustness of the processing in GASP, with similar sets of alignments being generated in different runs of the program [7].

Graph-theoretic approaches have provided the basis for most types of chemical structure handling application for many years, and it is perhaps worth comparing this very different way of processing structural data with the GA-based approaches considered here. If the problem space is sufficiently small, then one should use a graph-based algorithm if at all possible: for example, there are efficient algorithms available for MCS detection on pairs of either 2-D or 3-D structures [17, 18], and thus little reason for developing a nondeterministic approach unless the nature of the graphs that are to be processed precludes the use of a conventional algorithm (as with the hyperstructures discussed in section 7.3, which are too large to be processed efficiently using a conventional graph-matching procedure). In some cases, both approaches may be applicable. For example, our initial attempts at field-based similarity searching sought to represent molecular fields by graphs that could then be aligned by a suitable MCS procedure; specifically, Thorner et al. [39] described the use of graphs in which the nodes were the strongly positive or negative portions of a molecular electrostatic potential and the edges were the distances between the centers of these portions. The approach was found to work quite well in practice; however, the graph-generation procedure required the specification of a large number of parameter values, and the precise natures of the resulting graphs were highly dependent upon the values adopted for these parameters. The decision was hence taken to focus upon the GA-based approach, with the results summarized in section 7.5. In other cases, there may be alternative, nongraph-based methods available, as with the directed tweak algorithm for conformational searching described in section 7.4.1. In general, however, we believe that the results we have obtained suffice to demonstrate that GAs provide a valuable complement to existing techniques for the representation and searching of chemical structure information. Many further such applications remain to be investigated as more powerful types of processing become available, such as the Pareto approach to multicriteria optimization first discussed in a chemical context by Handschuh et al. [51].

Acknowledgments

My thanks are due to the following individuals and organizations for their involvement with the work reported here: Rob Brown, David Clark, Val Gillet, Bobby Glen, John Holliday, Gareth Jones, Ansgar Schuffenhauer, David Wild and Matt Wright for their contributions; the Biotechnology and Biological Sciences Research Council, the James Black Foundation, MDL Information Systems Ltd., ORAC Ltd., the Science and Engineering Research Council, Tripos Inc. and Wellcome Research Laboratories for funding; Derwent Information and Synopsys Scientific Systems for data; and Tripos Inc. for software support. The Krebs Institute for Biomolecular Research is a designated biomolecular sciences center of the Biotechnology and Biological Sciences Research Council.

References

[1] J. E. Ash, W. A. Warr, P. Willett (Eds.), *Chemical Structure Systems*, Ellis Horwood, Chichester, **1991.**

[2] R. D. Brown, G. M. Downs, G. Jones, P. Willett, A Hyperstructure Model for Chemical Structure Handling: Techniques for Substructure Searching, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 47–53.

[3] R. D. Brown, G. Jones, P. Willett, R. C. Glen, Matching Two-Dimensional Chemical Graphs Using Genetic Algorithms, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 63–70.

[4] D. E. Clark, G. Jones, P. Willett, P. W. Kenny, R. C. Glen, Pharmacophoric Pattern Matching in Files of Three-Dimensional Chemical Structures: Comparison of Conformational-Searching Algorithms for Flexible Searching, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 197–206.

[5] S. K. Drayton, K. Edwards, N. E. Jewell, D. B. Turner, D. J. Wild, P. Willett, P. M. Wright, K. Simmons, Similarity Searching in Files of Three-Dimensional Chemical Structures: Identification of Bioactive Molecules, *Internet J. Chem.* at http://www.ijc.com/articles/1998v1/37/.

[6] J. D. Holliday, P. Willett, Identification of Common Structural Features in Sets of Ligands Using a Genetic Algorithm, *J. Mol. Graphics Modell.* **1998**, *15*, 221–232.

[7] G. Jones, P. Willett, R. C. Glen, A Genetic Algorithm for Flexible Molecular Overlay and Pharmacophore Detection, *J. Comput.-Aided Mol. Des.* **1995**, *9*, 532–549.

[8] D. A. Thorner, D. J. Wild, P. Willett, P. M. Wright, Similarity Searching in Files of Three-Dimensional Chemical Structures: Flexible Field-Based Searching of Molecular Electrostatic Potentials, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 900–908.

[9] D. J. Wild, P. Willett, Similarity Searching in Files of Three-Dimensional Chemical Structures: Alignment of Molecular Electrostatic Potentials with a Genetic Algorithm, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 159–167.

[10] R. J. Wilson, *Introduction to Graph Theory*, 4th edition, Longman, Harlow, **1996.**

[11] R. C. Read, D. G. Corneil, The Graph Isomorphism Disease, *J. Graph Theory* **1977**, *1*, 339–363.

[12] M. R. Garey, D. S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, **1979.**

[13] J. M. Barnard, Substructure Searching Methods: Old and New, *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 532–538.

[14] P. Gund, Three-Dimensional Pharmacophore Pattern Searching, *Prog. Mol Subcell. Biol.* **1977**, *5*, 117–143.

[15] L. B. Kier, The Prediction of Molecular Conformation as a Biologically Significant Property, *Pure Appl. Chem.* **1973**, *35*, 509–520.

[16] W. A. Warr, P. Willett, The Principles and Practice of 3D Database Searching, in Y. C. Martin, P. Willett (Eds.), *Designing Bioactive Molecules: Three-Dimensional Techniques and Applications*, American Chemical Society, Washington, **1998**, pp. 73–95.

[17] J. J. McGregor, Backtrack Search Algorithms and the Maximal Common Subgraph Problem, *Software Pract. Exper.* **1982**, *12*, 23–34.

[18] A. T. Brint, P. Willett, Algorithms for the Identification of Three-Dimensional Maximal Common Substructures, *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 152–158.

[19] P. Willett, J. M. Barnard, G. M. Downs, Chemical Similarity Searching, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983–996.

[20] J. R. Ullmann, An Algorithm for Subgraph Isomorphism, *J. Assoc. Comput. Mach.* **1976**, *16*, 31–42.

[21] R. D. Brown, G. M. Downs, P. Willett, A. P. F. Cook, A Hyperstructure Model for Chemical Structure Handling: Generation and Atom-by-Atom Searching of Hyperstructures, *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 522–531.

[22] E. Fontain, Application of Genetic Algorithms in the Field of Constitutional Similarity, *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 748–752.

[23] M. Wagener, J. Gasteiger, The Determination of Maximum Common Substructures by a Genetic Algorithm: Application in Synthesis Design and for the Structural Analysis of Biological Activity, *Angew. Chem. Int. Ed. Engl.* **1994**, *33*, 1189–1192.

[24] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Wokingham, **1989.**

[25] R. A. Jarvis, E. A. Patrick, Clustering Using a Similarity Measure Based on Shared Nearest Neighbours, *IEEE Trans. Comput.* **1973**, *C-22*, 1025–1034.

[26] Y. C. Martin, M. G. Bures, E. A. Danaher, J. DeLazzar, I. Lico, P. A. Pavlik, A Fast New Approach to Pharmacophore Mapping and its Application to Dopaminergic and Benzodiazepine Agonists, *J. Comput.-Aided Mol. Des.* **1993**, 7, 83-102.

[27] D. E. Clark, P. Willett, P. W. Kenny, Pharmacophoric Pattern Matching in Files of Three-Dimensional Chemical Structures: Use of Bounded Distance Matrices for the Representation and Searching of Conformationally-Flexible Molecules, *J. Mol. Graphics* **1992**, *10*, 194–204.

[28] G. M. Crippen, T. H. Havel, *Distance Geometry and Molecular Conformation*, Research Studies Press, Taunton, **1988.**

[29] M. Clark, R. D. Cramer, N. Van Opdenbosch, Validation of the General Purpose Tripos 5.2 Force Field, *J. Comput. Chem.* **1989**, *10*, 982–1012.

[30] R. A. Dammkoehler, S. F. Karasek, E. F. B. Shands, G. R. Marshall, Constrained Search of Conformational Hyperspace, *J. Comput.-Aided Mol. Des.* **1989**, *3*, 3–21.

[31] CONCORD, GASP, SYBYL and UNITY are available from Tripos Inc. at http://www.tripos.com

[32] T. E. Moock, D. R. Henry, A. G. Ozkaback, M. Alamgir, Conformation Searching in ISIS/3D Databases, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 184–189.

[33] T. Hurst, Flexible 3D Searching: the Directed Tweak Technique, *J. Chem. Inform. Comput. Sci.* **1994**, *34*, 190–196.

[34] D. Barnum, J. Greene, A. Smellie, P. Sprague, Identification of Common Functional Configurations Among Molecules, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 563–571.

[35] C. Bron, J. Kerbosch, Algorithm 457. Finding All Cliques of an Undirected Graph. *Comm. Assoc. Comput. Mach.* **1973**, *16*, 575–577.

[36] A. Scott, S. A. DePriest, D. Mayer, C. B. Naylor, G. A. Marshall, 3D-QSAR of Angiotensin-Converting Enzyme and Thermolysin Inhibitors: A Comparison of CoMFA Models Based on Deduced and Experimentally Determined Active Site Geometries, *J. Am. Chem. Soc.* **1993**, *115*, 5372–5384.

[37] D. Flower, On the Properties of Bit String-Based Measures of Chemical Similarity, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 379–386.

[38] H. Kubinyi, G. Folkers, Y. C. Martin (Eds.) *3D QSAR in Drug Design*, Kluwer/ESCOM, Leiden, **1998.**

[39] D. A. Thorner, P. Willett, P. M. Wright, R. Taylor, Similarity Searching in Files of Three-Dimensional Chemical Structures: Representation and Searching of Molecular Electrostatic Potentials Using Field-Graphs, *J. Comput.-Aided Mol. Des.* **1997**, *11*, 163–174.

[40] R. Carbó, L. Leyda, M. Arnau, How Similar is a Molecule to Another? An Electron Density Measure of Similarity Between Two Molecular Structures, *Int. J. Quantum Chem.* **1980**, *17*, 1185–1189.

[41] A. C. Good, E. E. Hodgkin, W. G. Richards, The Utilisation of Gaussian Functions for the Rapid Evaluation of Molecular Similarity, *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 188–191.

[42] The *World Drugs Index* database is available from Derwent Information at http://www.derwent.co.uk/.

[43] S. K. Kearsley, S. Sallamack, E. M. Fluder, J. D. Andose, R. T. Mosley, R. P. Sheridan, Chemical Similarity Using Physicochemical Property Descriptors, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 118–127.

[44] D. B. Turner, S. M. Tyrrell, P. Willett, Rapid Quantification of Molecular Diversity for Selective Database Acquisition, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 18–22.

[45] The BIOSTER database is available from Synopsys Scientific Systems at http://www.synopsys.-co.uk/

[46] V. J. Gillet, A. Schuffenhauer, P. Willett, Similarity Searching in Files of 3D Chemical Structures: Analysis of the BIOSTER Database Using 2D Fingerprints and Molecular Field Descriptors, *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 295–307.

[47] L. Davis (Ed.), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, **1991.**

[48] G. Jones, P. Willett, R. C. Glen, Molecular Recognition of Receptor Sites Using a Genetic Algorithm with a Description of Desolvation, *J. Mol. Biol.* **1995**, *245*, 43–53.

[49] M. J. Bayley, G. Jones, P. Willett, M. P. Williamson, GENFOLD: a Genetic Algorithm for Folding Protein Structures Using NMR Restraints, *Protein Sci.* **1998**, *7*, 491–499.

[50] V. J. Gillet, P. Willett, J. Bradshaw, D. V. S. Green, Selecting Combinatorial Libraries to Optimise Diversity and Physical Properties, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 169–177.

[51] S. Handschuh, M. Wagener, J. Gasteiger, Superposition of Three-Dimensional Chemical Structures Allowing for Conformational Flexibility by a Hybrid Method, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 220–232.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

8 Molecular Diversity Analysis and Combinatorial Library Design

Lutz Weber

Abbreviations

CAMD	Computer-assisted molecular design
EA	Evolutionary algorithm
EM	Ensemble of molecules
GA	Genetic algorithm
QSAR	Quantitative structure-activity relationship
VB	Valence bond

8.1 Introduction

The simultaneous synthesis of many different compounds in an extremely efficient way emerged as a new discipline in chemistry during the last decades of the twentieth century. Combinatorial chemistry has now become an established tool, especially in pharmaceutical research, to aid in the discovery of new molecules and materials with desired properties from collections of different compounds – called combinatorial libraries. New developments in chemical synthesis strategies and instrumentation, as well as new high-throughput testing methods, have enabled the synthesis and screening of many hundreds to thousands of compounds in a shorter time than previously possible.

As a consequence of these new developments, we are now facing questions that had not received much attention before the inception of combinatorial chemistry. How large is accessible chemical space? How diverse are different molecules? How many and which molecules need to be made to find an optimal one? What is chemical similarity and does it relate to similarity in property space? Can one optimize the information content of a compound library? How general are chemical reactions?

Clearly, to answer these questions, a tight integration of experimental design with novel computational and mathematical methods is required. When this is achieved, combinatorial chemistry may well make a significant contribution towards moving chemistry from a purely empirical science of singular observations to a science with more abstract and logical reasoning based on mathematics.

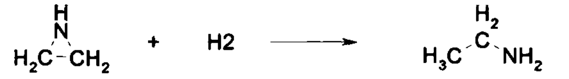
This chapter is intended to provide an overview of some recent approaches towards the description of molecular diversity and, as a consequence, the design of combinatorial compound libraries. The problem of describing molecular diversity and its relation to the corresponding properties of individual compounds or compound collections can be subdivided into several areas: the definition of a chemical and a property space; suitable similarity measures for both spaces; and the relation between these measures. While the definition of a property space and suitable measures therein are somewhat easy to obtain and obviously context-dependent, the definition of a chemical space and its relation to the property space of interest is far from obvious.

By analogy to genetics, one may regard such a chemical space or the chemical description of a molecule as the "genotype", whereas its property would be the "phenotype". This analogy matches the idea of chemistry that uses discrete "building blocks" (atoms, reagents, starting materials, and reactions) to yield whole molecules. While the chemistry space is by its nature discontinuous, the related property space is mostly continuous, for example, the lipophilicity of molecules.

Although the introduction of the concept of chemical genotypes is arbitrary, it allows the application of evolutionary algorithms (EAs) – computational methods that are well suited to discover the relationship between genotypes and phenotypes. Furthermore, combinatorics in a mathematical sense is nothing other than statistical analysis in a discontinuous space, thereby matching the requirements of evolutionary algorithms that are based on discrete, discontinuous genes or descriptions. Combinatorial optimization methods [1] have proven to be useful in solving multidimensional problems, and are now being used with success in various areas of combinatorial chemistry. Thus, evolutionary computation as a specific branch of combinatorial optimization may aid in the selection of information-rich subsets of available compound libraries or in designing screening libraries and new compounds to be synthesized, thereby adding a new quality to combinatorial chemistry.

8.2 The Diversity of Genotypes: The Space of Chemistry

Various conventions have been developed to describe chemical structures. One of the most important is valence bond (VB) theory that uses certain rules for connecting atoms with bonds. This emerged from Sir Arthur Cayley's representation of molecules by graph theory in the nineteenth century. A general VB-based algebraic representation of the chemical similarity of molecules or "ensembles of molecules" (EM) has been developed by Ugi et al. [2] using *be*- (valence *b*onds and *e*lectrons) and *r*- (*r*eaction) matrices. Within this representation, all atoms of a molecule, their connectivities and shared electrons are used to compute a metric "chemical distance" between isomeric molecules [3] or EM. This description of molecules lends itself to a formal and generally applicable structure-based diversity measure based on counting the changes in these matrices that are needed to transform one molecule to another. For example, the representation of aziridine and ethylamine by *be*-matrices is given in Fig. 1. The four steps needed for their interconversion by a hydrogenation reaction are given by the *r*-matrix. This reflects the chemical distance between the molecules in a straightforward way and provides a metric for the respective chemical space. In terms of "small molecule genetics", four mutations are needed to convert from the genome of aziridine to the genome of ethylamine. As this is a rather small distance, both molecules are similar with respect to their chemical structures.



	H	H	N	H	C	C	H	H	H	H
H	0	1	0	0	0	0	0	0	0	0
H	1	0	0	0	0	0	0	0	0	0
N	0	0	2	1	1	1	0	0	0	0
H	0	0	1	0	0	0	0	0	0	0
C	0	0	1	0	0	1	1	1	0	0
C	0	0	1	0	1	0	0	0	1	1
H	0	0	0	0	1	0	0	0	0	0
H	0	0	0	0	1	0	0	0	0	0
H	0	0	0	0	0	1	0	0	0	0
H	0	0	0	0	0	1	0	0	0	0

***be*-matrix for aziridine**

	H	H	N	H	C	C	H	H	H	H
H	0	-1	1	0	0	0	0	0	0	0
H	-1	0	0	0	1	0	0	0	0	0
N	1	0	0	0	-1	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0
C	0	1	-1	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0

***r*-matrix**

	H	H	N	H	C	C	H	H	H	H
H	0	0	1	0	0	0	0	0	0	0
H	0	0	0	0	1	0	0	0	0	0
N	1	0	2	1	0	1	0	0	0	0
H	0	0	1	0	0	0	0	0	0	0
C	0	1	0	0	0	1	1	1	0	0
C	0	0	1	0	1	0	0	0	1	1
H	0	0	0	0	1	0	0	0	0	0
H	0	0	0	0	1	0	0	0	0	0
H	0	0	0	0	0	1	0	0	0	0
H	0	0	0	0	0	1	0	0	0	0

***be*-matrix for ethylamine**

Figure 1. *be*- and *r*-matrices for the hydrogenation reaction of aziridine.

It can be seen that chemistry has well-defined metric properties that can be used to compare two molecules or even reactions. For convenience, the detailed VB-based model described above can be simplified by introducing derived descriptors or encoding schemes. Examples of such schemes are the SMILES notation introduced by Weininger [4] or the letter codes for peptides, carbohydrates, and nucleic acids. The reconstruction of a VB-structure from the respective encoding is accomplished according to a defined set of rules. These rules can be encoded themselves and used to build whole molecules [5, 6] in a more comprehensive way using substructural representations, for example, >C<, -H, -O-, >C=O, $-C_6H_5$, as building blocks.

Contrary to general encoding schemes, combinatorial libraries allow an efficient encoding using arbitrary encoding schemes. Binary strings [7, 8], alphanumeric values or real-valued strings [9] have been used to encode the starting materials of a combinatorial library. This encoding strategy relies on a context convention, similar to the idea of the peptide letter code where a letter, for example, A standing for alanine, represents a whole molecule. For such molecular genomes, an operator is needed that translates each encoding into a general VB-type representation of real molecules assuming a given chemical assembly reaction. The explicit description of each different molecule in such compound libraries is often called *enumeration* (Fig. 2). The respective operator may also be understood as a synthesis scheme or a routine that drives an automated synthesizer generating a particular molecule.

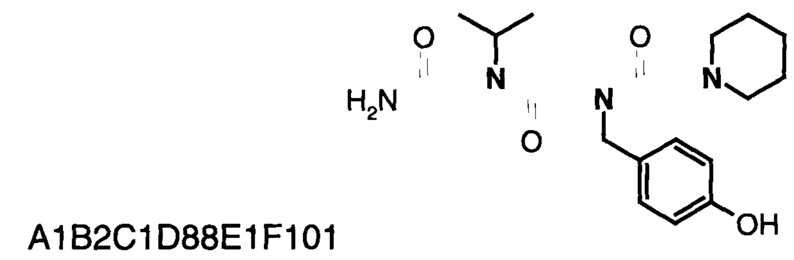


Figure 2. Encoding a tripeptoid structure with an arbitrary alphanumeric genome [9] using building blocks B2, D88, F101 (in bold) and reaction schemes A1, C1, and E1.

The number of possible different molecules constitutes an enumerable search space of practically unlimited size. For example, the number of all different proteins comprising only 200 amino acids is 20^{200}, a number that is much larger than the number of particles in Space (estimated to be about 10^{88}). Similarly, the number of "drug-like" molecules with a molecular weight below 500 Da that could be synthesized in principle is far in excess of our synthetic and even computational capabilities, and is not yet possible to calculate. The program MOLGEN [10] can calculate all possible isomers of a given constitution and while a hydrocarbon of the general formula C_6H_{14} has only five isomers, $C_{18}H_{38}$ has some 60 523!

Finally, it is important to note that any genotype, for instance the triplet UUC in Nature's encoding scheme for the amino acid phenylalanine, does not reflect any biological or physico-chemical property of the encoded phenotype directly. Rather, the direct consequence of the encoding scheme is the metric of the chemistry space of interest.

8.3 The Diversity of Phenotypes: The Property Space

A broad range of interesting molecular properties can be calculated or estimated from a compound's chemical structure. Broadly, one can distinguish between properties that can be derived from:

- the two-dimensional (2-D) representation of the chemical structure;
- the three-dimensional (3-D) structure;
- physical property calculations; and
- experimental data such as biological activities.

Quantities such as molecular weight, the number of heteroatoms, substructural fragments, and the number of hydrogen-bond donors and acceptors can be determined exactly. A range of several hundred topological indices is accessible via programs such as MOLCONN-Z [11]. Other parameters may be dependent on the 3-D conformation of the molecules (3-D parameters) – a property that is unknown *a priori* and can only be approximated. In the case of conformationally flexible molecules, multiple conformations are possible, each of which contributes according to its Boltzmann weight to the overall phenotype of the molecule.

Many properties of interest cannot be calculated or estimated from the chemical structure. These properties are often those that reflect the interaction of molecules with other complex systems such as living organisms. Molecules that exhibit the same property may be chemically very distant or dissimilar, for example, lithium is an important antidepressant, as is amitryptiline. Obviously, it would be very difficult to conclude from the chemical structure of amitryptiline that the atom lithium will show the same pharmacological property. Conversely, very similar molecules may have very different properties, for example the glycine derivative glyphosphate (Fig. 3) is well known for its herbicidal activity; however, any small variation of the chemical structure abolishes this activity.

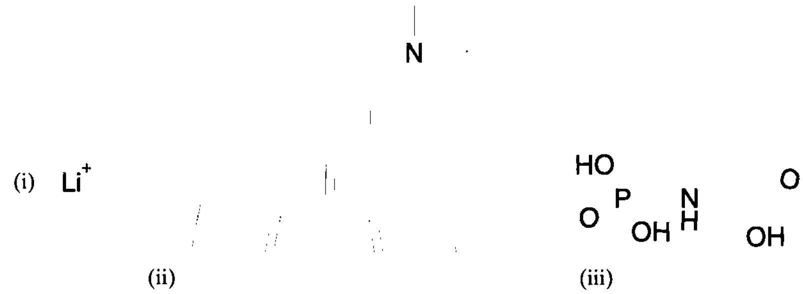


Figure 3. Antidepressants (i) lithium, (ii) amitryptiline and (iii) the herbicide glyphosphate from Monsanto.

These examples show that the metric of the chemistry space and the metric of the property space are not necessarily related or, in other words, the similar-property principle is not inviolable. Any method for diversity-based predictive selection will fail in these cases. Nevertheless, there are well-known examples where the metric of the chemistry space and the property space seem to be closely correlated. Thus, on the level of DNA, a single mutation of UUC encoding phenylalanine yields UUA encoding leucine, an amino acid very similar to phenylalanine in terms of its hydrophobicity. Conversely, three mutations are required to obtain CAA which encodes glutamate, a very different amino acid with respect to both leucine and phenylalanine. An example involving small organic molecules is given in Fig. 4, which depicts similar molecules that are all available commercially as local anesthetics.

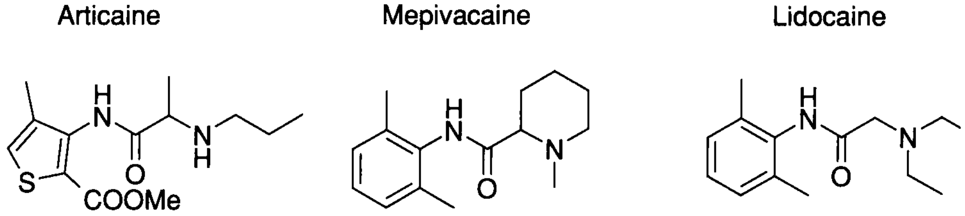


Figure 4. Glycine derivatives that are used as local anesthetics.

The above examples have been given to illustrate the nature of the correlation between the genotype (chemistry) and phenotype (property space): there are areas of both predictable and nonpredictable behavior. Within a certain area of predictable behavior one may find explicit, rather simple models that describe this relationship – these are the classical quantitative structure-activity relationships (QSARs). QSAR methods are based on the somewhat naive assumption that similar compounds exhibit similar biological activity. For example, in recent work, a compilation of descriptors is given for the task of developing models that allow the differentiation of "active" and "inactive" molecules [12].

Models that allow the description of structure-property relationships over predictable as well as nonpredictable areas are the focus of current research using evolutionary computational methods. Adopting the model of areas of nonpredictable and predictable relationships between the genotype and phenotype spaces, Fontana and Schuster [13] investigated the shape change of short RNA sequences depending on the changes in the sequence of those RNA aptamers. They give a definition of *continuity* between genotypes and phenotypes: neighbors in sequence space (i.e., genotypes) that are also neighbors in the shape space (i.e., phenotypes) are said to exhibit continuity. A change in sequence space that gives rise to a different shape is called a *transition*. Two types of transitions are observed: *continuous*, as defined above, and *discontinuous* (or in our context, a different genotype-phenotype area, the properties of which cannot be predicted or extrapolated from the previous one).

Selecting compounds in such a way that each continuous structure-property relationship area is represented by one compound would give a library of maximally diverse compounds of *discontinuous transitions*. Selecting compounds from only one continuous area of the structure-property relationship would give similar compounds that could still be diverse if each *continuous transition* were represented by one compound.

8.4 Diversity and Distance Calculation

The selection of diverse molecules or sublibraries out of a larger library is a combinatorial problem and thus, a computationally intensive task. Choosing n maximally diverse compounds out of a library of N compounds requires the evaluation of

$$\frac{N!}{n!(N-n)!} \tag{1}$$

subsets [14]. The diversity comparison of all molecules has a computational complexity of order $O(N^2)$. Thus, in the case of large libraries, the full computational evaluation of all relationships becomes nonfeasible.

The distance between two molecules in chemistry or property space can be calculated by various methods based on the coordinates x_k and y_k of molecules x and y in the k-dimensional space [15]. For example, the Euclidean distance:

$$D(x,y) = \sqrt{\sum_1^k (x_k - y_k)^2} \tag{2}$$

the squared Euclidean distance that gives more weight to molecules that are more distant:

$$D(x,y) = \sum_1^k (x_k - y_k)^2 \tag{3}$$

the City-Block or Manhattan distance:

$$D(x,y) = \sum_1^k \left| x_k - y_k \right| \tag{4}$$

or the Cosine coefficient calculated from the cosine of the angle formed by the two molecular vectors in the property space. The latter has been recommended [16, 17] because it lends itself to rapid calculation. However, it is dependent on the definition of the origin. Each of the above distance metrics leads to different results with respect to compound selection – a rather unsatisfactory result which is discussed by Agrafiotis and Lobanov [18]. Outliers may also have undesired effects. Thus, a highly dissimilar compound within a given compound set will tend to make all the other compounds appear more similar than they really are.

Cell-based methods have been developed that partition the space of interest into boxes or cells. The compounds are then assigned to these cells [15, 18, 19], which provides a computationally faster evaluation of distances. The optimal number of bits (b) per property descriptor has been suggested as $b = (N/2)^{1/d}$ where N is the number of molecules and d is the number of descriptors used. This procedure leads to approximately two molecules per cell [15].

Using the metric measures described above, the selection of maximally diverse compounds can be evaluated. The MaxMin selection method has been proposed as effective and efficient for compound selection. MaxMin maximizes the minimal pairwise dissimilarities between the compounds to be selected for a maximally diverse compound set [20].

Fractal theory has been used to judge the quality of selected subsets that were obtained by various methods [21]. The underlying idea is that the fractal dimension of an optimally selected subset should be the same as for the original library. The fractal dimension is easy to calculate with a box counting method, and therefore provides a facile measure for comparing different diversity selection methods.

8.5 Connecting the Structure and the Property Space: Evolutionary Algorithms

Evolutionary algorithms, such as genetic algorithms (see below), are inspired by the principles of Darwinian evolution and do not require an explicit knowledge of structure-activity relationships. Several properties of EAs make them attractive for dealing with problems of molecular diversity and compound selection [22]:

- They are ideal for optimization problems where the search space is very large and the solution consists of a subset of a large number of parameters, each of which can be independently varied. In the case of the genotype of combinatorial or other compound libraries, these independent parameters are, for example, the starting materials for synthesis or the substructures found in the complete molecules.
- Finding solutions in the discrete, discontinuous and nonsteady space of structure-property relationships is very similar to evolutionary processes in Nature.
- Very large compound libraries that cannot be enumerated for practical reasons can only be handled with heuristic methods.

One possible disadvantage of evolutionary algorithms is that they are stochastic; therefore, each run may yield a different result – providing a set of good solutions instead of the best solution. However, this can be an advantage in situations where multiple solutions are of interest, as is often the case in computer-assisted molecular design (CAMD).

A series of other heuristic optimization algorithms that will not be covered in this chapter includes fuzzy logic, neural networks, cellular automata, fractals and chaos theory and hybrid variants. These have been collectively termed "soft computing" methods [23].

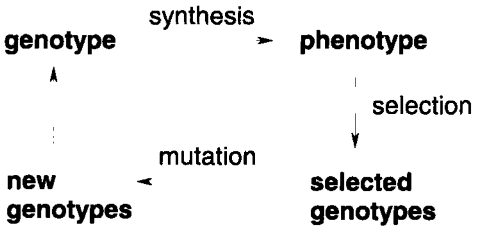


Figure 5. Selecting compounds with desired properties with evolutionary algorithms.

The principles of evolution can be generalized in a simple scheme that displays the basic elements necessary for the implementation of artificial evolutionary systems (Fig. 5) for the selection of compounds with desired properties. Generally, new generations of compounds with better fitness are evolved based on the fitness of the properties of previous generations. The genotype is the description of the chemical space and the phenotype of the particular compound is either calculated or measured.

Many evolutionary algorithms have been developed to select optimal combinations from a pool of combinatorial possibilities. One of the first was developed by Rechenberg [24] at the Technical University of Berlin. These algorithms were applied to the calculation of optimal wing profiles for airplanes. In his basic work, a wide variety of different selection and mutation operators were defined, giving rise to a series of possible optimization strategies. One special subcase of such evolutionary optimization algorithms has been named and made popular by Holland. This is the "genetic algorithm" (GA) [25], so-called because of its similarity to the principles of DNA-based evolution in living organisms. The application of evolutionary, and especially genetic, algorithms towards problems in chemistry is still in its infancy but has increased rapidly, as shown in recent reviews [26, 27].

A GA usually starts with a randomly chosen set of different entities encoded by genomes – the "population". The evaluation and ranking of the fitness of these genomes is performed based on the properties of their phenotypes during the selection step. Various methods are possible to select parents that are to be considered for creating the new generation of offspring – the "children". These new genomes are then generated from the ranked list of parents by GA functions that are inspired by those of DNA-like genetics: *death*, *replication*, *mutation*, *deletion*, *insertion*, and *crossover* (Fig. 6).

Replication regenerates an equivalent individual (child = parent). *Mutation* sets one or more bits in the parent genome to a different value, thereby providing a new child. *Crossover* combines two or more parent genomes to build child genomes by mixing elements of the parents according to various rules. *Deletion* deletes a bit or bit strings from the parent genome, *insertion* introduces new bits or bit strings. According to the chosen encoding scheme, these functions may have different meanings when applied to the genome. Thus, crossover applied to a genome where the variable parts of a compound structure are encoded by a binary representation may result in a different outcome to crossover applied to a real-valued representation. In a real-valued encoding scheme, a crossover operation may intersect the genome only in between these real-values (i.e., building blocks or substituents). Therefore, the building blocks of the respective parents are exchanged among the children, but both their number and nature will remain the same. In the binary string representation however, the crossover may also take place within the binary string that represents a given building block (i.e., between the bits that make up the gene). As a result, the children may contain different building blocks from their parents. This is also the case for Nature's encoding of proteins, for example, a crossover within the triplet UUC (phenylalanine), for instance UU/C, may result in the novel building block UUA (leucine) which was not present in the DNA encoding of the parent protein. This crossover strategy has the effect that the crossover operator is effectively a mixture of crossover and mutation, with the mutation probability increasing with the length of the encoding string for a given building block. For example, there are four possibilities to cut UUC: at the beginning and the end (/UUC or UUC/) as well as two possibilities of intra-building block cuts (U/UC or UU/C).

Death:	1001 1110 1111	=>	
Replication:	1001 1110 1111	=>	1001 1110 1111
Mutation:	1001 1110 1111	=>	1**1**01 1110 1111
Deletion:	1001 1110 1111	=>	1001 110 1111
Insertion:	1001 1110 1111	=>	1001 111**01**0 1111
Crossover:			
parent 1	**1001** 1110 1111	=>	**1001 1001 1001**
parent 2	1001 **0001 1001**	=>	1001 0110 1111

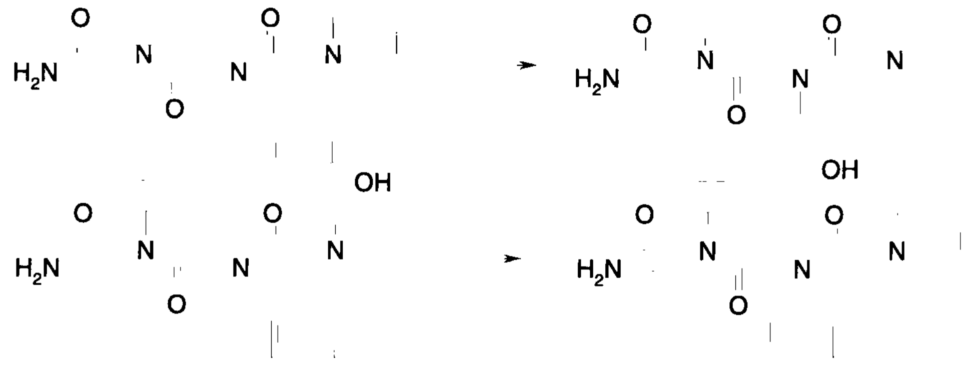


Figure 6. Operators of genetic algorithms as they may be applied to the encoding of molecules from combinatorial libraries. The tripeptoid example from Fig. 2 has been chosen to illustrate the DNA-like crossover with binary bit strings.

While the first strategy of replacing only whole building blocks appeals more to the intuition of chemists, the latter (see Fig. 6) is more similar to natural crossover in DNA. Contrary to evolution in Nature, we are completely free to define how these GA operators are applied in the computer; for example, a new child may have more than just one or two parents.

After evaluating the fitness of the molecules, the GA includes a ranking and selection step that determines which genome is subject to which genetic operator. This step is an implementation of the idea of "survival of the fittest" where fit genomes are allowed to survive and weak genomes are eliminated. The methods that are found in the literature differ significantly. In one example, a ranked list of genomes is generated and then the genomes that are to be killed off are determined, for instance, those that are older than a specified number of cycles. The remaining list of a predetermined population size is then treated by combinations of replication, mutation or crossover chosen either with equal probability, stochastically or with a distribution according to the rank of the genome in this list. Thus, in an example method called "best third", the worst third of all genomes is eliminated, the best third of the genomes is simply replicated to the new generation (giving rise to an "elitist" selection), and the middle third is subject to mutation and crossover to generate new children. However, while many EA variants are possible, it has not been shown that a specific version is superior. Many other parameters can be set and varied during the course of a GA experiment: for example, size of populations, number of surviving genes, mutation rate, number of parents per child and finally the ranking or fitness function. This parameter set constitutes the "breeding" recipe for molecules of higher fitness. However, finding optimal parameters for a given problem is an optimization problem in itself (see Chapter 12).

The "structure" of the search space has a large influence also on whether or not a GA will be successful [24, 25]. The properties of the compounds that are the basis for selection are either measured or estimated by calculation, and may be erroneous. This error may be regarded as "noise" in the property space. Levitan and Kauffman have investigated the influence of noise on the performance of GAs [28]. Interestingly, the ability to find more optimal solutions in a shorter time with a GA increases when the fitness measurement is slightly noisy! Apparently, this is because a population leaves suboptimal areas of the search space more easily in the presence of noise.

The good searching power of GAs is believed to originate from the building-block hypothesis [29–32]. This model assumes that the combination of "fit" building blocks (or contiguous bit strings or schemes of genes) in the genomes may yield higher-order schemes of even better fitness. This optimization behavior matches perfectly the discontinuous structure space of chemistry that is formed by building blocks and which can then be statistically analyzed by the GA. In chemical terms, such building blocks may be atoms, substituents, reagents, starting materials or even reactions. Combinatorial libraries are especially suitable for GA-based evaluation since they are generated from systematic arrays of building blocks. The length of the encoding string for a given building block also has an influence on the optimization behavior, as shown in [32]. Other criteria that may influence search space coverage and selection pressure are given in [33]. For a recently published book on the theory of GAs, see [29].

Other prominent heuristic algorithms differ from GAs, but are conceptually similar. Simulated annealing (SA), inspired by the natural process of "annealing", starts with a random individual (classically SA works on individuals not populations) [34] and evaluating its fitness (D_{curr}). A new individual is generated by making a random change to the initial state; this individual is accepted if its fitness (D_{new}) is below a value that is calculated by a formula similar to

$$rnd < e^{-[(Dcurr - Dnew)/T]} \qquad (5)$$

where *rnd* is a random number between 0 and 1, and T is a parameter analogous to the temperature in the Boltzmann distribution equation [35].

8.6 Evolutionary Methods for Compound Selection

Genetic algorithms have been used to select molecules exhibiting structural similarities to a given target molecule, such as a known drug, from large virtual libraries. In this case, the fitness function is a description of molecular similarity to the target. Using GAs for the generation of diversity, one may also choose a molecular property and select molecules that are different with respect to that property. For example, selecting molecules to have dissimilar and unique molecular weights would facilitate, for example, the deconvolution of combinatorial library mixtures by mass spectroscopy.

The first published example of similarity selection by a GA was reported by Sheridan and Kearsley who used tripeptoid molecules [9] that were synthesized *in silico*. The molecules were built by selecting from a set of 2507 × 2507 × 3312 building blocks, giving a virtual library size of about 20 billion. From this library, a specific tripeptoid (see Fig. 2) was chosen as the target molecule. A topological descriptor using atom pairs separated by a specific number of bonds was used as a similarity measure. Several GA strategies were studied including the stochastic + best-third selection and random + neighbors mutation. In the stochastic selection procedure, parents are chosen randomly from the previous population to generate new children, whereas in the best third method, the top-scoring best third of all parents are transferred unchanged, the worst third is eliminated, and the middle third is used to generate new children via crossover. Random mutation permits each gene to be mutated with equal probability, whereas neighbors mutation follows a given rule that a mutation may lead only to a similar building block. With a population size of 300 molecules, elitist best-third selection and neighbors mutation, the right solution was found after only 12 generations! This result is astonishing since only 3600 peptoids were examined out of the 20 billion possible. Known cholecystokinin (CCK) and angiotensin-converting enzyme (ACE) antagonists were also chosen as molecular targets to search for similar tripeptoids. A striking structural similarity between the proposed peptoids and the target molecules was achieved after only 20 generations.

The analysis of the diversity of combinatorial libraries appears to be more amenable to computational methods than that of general molecule collections due to the well-defined "closed" chemical space of the former. The optimal design of combinatorial compound libraries was the aim of a recent application reported by Brown and Martin [36]. A GA was designed that is useful in selecting optimal subsets out of a pool of possible starting materials for combinatorial library synthesis. In one example, 360 and 259 precursors were available for a reaction product with two sites of diversity, giving rise to 93 240 possible compounds. The library design required the selection of a pool of 100 × 100 starting materials for a compound library that was required to be optimal with respect to diversity as well as having a minimum product and substituent molecular weight redundancy of its compound members. In this example, the chromosome encodes an entire sublibrary rather than individual compounds. The number of possible chromosomes is 2.5×10^{164}. To solve the problem, a complex fitness function was assembled by using the weighted mean of individual fitness criteria, with the result that several properties of the library are optimized simultaneously.

Selecting optimally diverse compounds out of a large combinatorial library based simply on the product structures is synthetically inefficient, as such a subset does not represent a combinatorial library. Several methods have therefore investigated the design of diverse combinatorial compound libraries by selecting optimal sets of building blocks for synthesis [37, 38]. It has been shown [38] that selecting reactant pools based on the diversity of their products as opposed to the reactants results in libraries of higher diversity while still maintaining synthetic efficiency. In this example, the sum of all molecular pairwise dissimilarities of the compounds of a sublibrary was maximized using a GA to select reactant pools. The optimized combinatorial library was then found to be very close in diversity to optimally dissimilar noncombinatorial subsets.

Using molecular volume, lipophilicity, charge, and H-bond donor or acceptor descriptors [39, 40], it has been shown that peptoid-type combinatorial libraries may be designed to exhibit the same density of descriptor distributions counted per atom [41] as commercial drugs.

This method has been developed further by Liu et al. [42] for molecules that are defined by a core backbone structure with various attachment points for substituents. After generating the new compounds by a GA, the conformation of each compound was minimized using the MM2 force field. The 3-D similarity of the molecules was then calculated by comparing 28 principal components extracted from their Comparative Molecular Field Analysis (CoMFA) matrix. As an example, a maximally diverse benzodiazepine library with three points of diversity was constructed showing a fast convergence of the diversity criterion after four generations. The same methodology was also used to obtain a library of molecules similar to (-)-huperzine A, a natural acetylcholinesterase (AChE) inhibitor. One of the proposed compounds showed better inhibitory activity than the parent molecule – thus providing an experimental proof-of-concept. A GA has also been used to improve the parameters of QSAR models in a study using AChE inhibitors [43]. Here, the various 3-D properties of existing molecules are encoded as genomes and a GA selects those that have the most predictive value. The GA-driven selection of descriptors has been found to be so useful that several commercial and proprietary diversity assessment programs include this feature.

A GA has been used in a similar way to propose new polymer molecules that mimic a given target polymer [44]. The molecules were built in the computer by linking predefined genes (main chain or side chain atoms or atom groups) subject to several chemical rules designed to ensure stability. Some novel and interesting GA operators were introduced such as insertion and deletion of genes into chromosomes, shifting main chain atom groups into other positions on the chromosome, or blending parent chromosomes into one large chromosome. Even more chemical rules are needed when generating general, nonpolymeric molecules of all structural classes with a GA [6]. Target molecules with a given molecular weight and 3-D shape were chosen as an example. The method was stated to be of use for the design of any class of molecules, such as enzyme inhibitors, polymers, or new materials. Similarly, a stochastic diversity search method has been used to select a small library of polymers with a diverse range of glass transition temperatures and hydrophobicities [45]. From a library of 112 synthesized polymers, 17 were chosen as a training set. After developing a quantitative structure-property relationship (QSPR), the model was used to select both 17 maximally diverse polymers and a set of similar polymers out of the 112. The comparison with their experimental properties enables a judgment to be made on their diversity. Although the number of polymers considered is not very high, this work is the first example of diversity and similarity selection that can be verified based on experimental data in the field of materials discovery.

An interesting example of selecting active compounds from a large database of general molecules with a GA was presented by Gobbi and Poppinger [46]. Molecules were encoded with a bit string of length 16 384 in which individual bits are set according to the occurrence of predefined substructural elements. After crossover, the molecule that is most similar to the new offspring was retrieved from the database and added to the population. Once a parent was used more than 10 times it was eliminated from the parent set. The GA was tested in a simulation using a data set from the National Cancer Institute comprising 19 596 molecules with biological activities. Using a population size of 20 compounds, most or all highly active compounds were found after examining only 1 % to 10 % of the complete database. This method may offer a means to replace conventional, "blind" high-throughput screening in the future enabling a significant reduction of screening costs.

A similar method was used to develop biological activity profiles based on H-bond donor and acceptor properties and other structural features that may differentiate "drug-like" molecules from others using the World Drug Index (WDI) as a "drug-like" set and the SPRESI database [47] as a "nondrug-like" set. The resulting set of descriptors was used to classify compounds as drug-like or nondrug-like. Using a randomly assembled set of drugs and compounds from the chemical supplier's catalogs, the ratio of drugs versus nondrugs was evaluated for the selected set of predicted drug-like molecules. A GA-driven selection using the descriptors as a fitness function versus a random selection obtained an initial enhancement of up to 6.4 in favor of predicted drugs. The enhancement ratio was found to be higher when using molecules selected from specific biological activity classes (hormones 8.3, anticancers 6.8 and for antimicrobials 6.2) but lower for other, more general, classes as anesthetics, blood and central nervous system (CNS) drugs. This behavior is understandable when considering the structural diversity of the known drugs for these areas, and indicates the general limitation of this method. However, the GA-based selection of such activity profiles performed better than a purely statistical analysis of substructural fragments. Subsequently, such profiles were also used to select optimally diverse, but drug-like, combinatorial libraries [14] based on product properties.

A program that combines several of the ideas of evolutionary compound selection described above was developed by Tropsha and coworkers [48, 49]. Simulated annealing or GAs were used to select building blocks for combinatorial libraries that comprise products that show similarity with a given target molecule. Building blocks are then associated with a probability that they give rise to products that fulfil the design criteria. Building blocks with a frequency above random are then chosen as suitable for a sublibrary for synthesis. The Kier-Hall topological 2-D descriptors were used. While this program can be seen as the continuation of the early work of Sheridan and Kearsley [9], the generality and experimental value has still to be shown. A similar method for selecting diverse compounds was implemented by the same authors. Simulated annealing guided evaluation (SAGE) of molecular diversity of virtual combinatorial libraries [50] was performed using a diversity function that was stated to give better sampling than the MaxMin procedure.

8.7 Computer-Aided Evolutionary Chemistry

The evolutionary principles of Nature have been used to develop experimental evolutionary methods such as phage display libraries, combinatorial biochemistry, or even artificial evolution of enzymes [51]. The idea of the experimental application of evolutionary chemistry for small, nonoligomeric molecules is based on the idea of replacing Nature's "informatics system" (DNA) by molecules encoded in the computer and applying GAs to these artificial genomes. Some early examples have been reported of the successful integration of GAs, organic synthesis and biological testing in evolutionary feedback cycles that yield molecules with desired properties.

In one example, a population of 24 hexapeptides was randomly chosen from the 64 million possible hexapeptides and optimized for trypsin inhibition using a GA [52]. Biological testing was performed with a chromogenic trypsin assay. According to the "best-third" method described earlier, the best eight peptides out of 24 were duplicated, the worst eight were eliminated, and the remainder were kept to arrive at a new population of 24 peptides. This new population of peptides was then changed using a crossover rate of 100 %, with children being generated from two randomly chosen parents. Thereafter, mutation was applied with a probability of 3 %. The average enzymatic inhibition of the synthesized peptides (at 200 μM concentration) was improved from 16 % in the first randomly chosen population to 50 % in the sixth generation. Moreover, 13 out of the 25 most active peptides had a consensus sequence of Ac-XXXXKI-NH_2 and eight of these possessed an Ac-XXKIKI-NH_2 sequence. The most potent peptide identified was Ac-TTKIFT-NH_2 with an inhibition of 89 %, being identical with a previously found trypsin inhibitor from a phage display library.

In a further example, substrates for stromelysin were searched for in the pool of possible hexapeptides [53]. Only 300 peptides were synthesized in five generations to obtain useful substrates. The peptides were synthesized on solid support and fluorescence-marked at the N-terminus. After proteolytic cleavage, the nonfluorescent beads were analyzed. The starting sequences were biased by using 60 peptides with the sequence $X_1PX_3X_4X_5X_6$, but the bias was removed in all subsequent generations. From each population of 60 peptides, the best was copied to the new generation, the others were changed using a crossover rate of 60 %. The new peptides were then subjected to mutation with a rate of 0.1 % applied to each bit of the 30-bit gene, giving a 3 % overall mutation rate. The GA was terminated when 95 % of the peptides in the population were identical. The hexapeptide MPQYLK was identified as the best substrate for stromelysin in the final generation, being cleaved between the amino acids tyrosine (Y) and lysine (L). The selectivity of the new substrates versus collagenase was also determined, and selective substrate sequences were identified for both enzymes. Therefore, this method may not only help to find new substrates but to also obtain structure-activity and selectivity ideas for new inhibitors.

A combination of GAs and neural networks was used for the design of signal peptidase I cleavage sites [54, 55]. Here, simulated molecular evolution exploits the rules of peptide sequence space. A neural network was trained with a set of protein sequences that are known cleavage sites for the peptidase. The trained network was used afterwards as a fitness function to predict new leader peptidase substrate sequences such as FFFFGWYGWA-RE. These proposed artificial cleavage sites were then cloned in *Escherichia coli* within a fusion protein that turned out to be processed by the secretory proteases as predicted.

The first example of GA-driven selection and synthesis of nonpeptidic molecules was applied to thrombin inhibitors [7]. Using 10 isonitriles, 40 aldehydes, 10 amines and 40 carboxylic acids, 160 000 Ugi four-component reaction combinations are possible. Whereas in the initial population, the best reaction product exhibited an IC_{50} of about 300 μM, a thrombin inhibitor with a submicromolar IC_{50} was discovered after 20 generations comprising 20 single Ugi products per population. To our surprise, this *N*-aryl-phenylglycine amide derivative **A** (Fig. 7) is the three-component side product of the four-component reaction **B**. However, the encoding was based on the process of combining the four starting materials and not the final, expected products. The applied GA is obviously not product structure-based and the fitness function depends on the process, including possibly varying yields of the reaction, mistakes in biological screening, and so on. The GA therefore appears to be quite tolerant of experimental mistakes and may still yield good results even if the starting hypothesis is wrong, as false-negative results are simply eliminated and not remembered. The elimination of false-positive results takes somewhat longer – depending on how often a good genome is allowed to replicate. This "fuzzy" and robust optimization property makes GAs especially attractive for real-time experimental optimizations as described above.

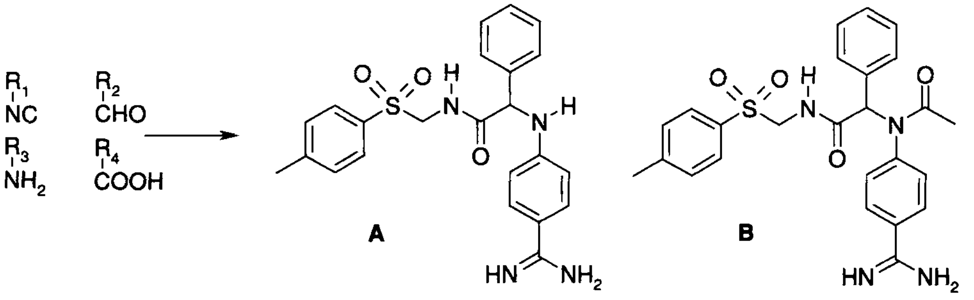


Figure 7. *N*-aryl-phenylglycine amide-type thrombin inhibitor selected from 160 000 possible reaction products with a GA and a thrombin inhibitor assay as the feedback function.

We have recently synthesized a full combinatorial library with a Ugi three-component reaction comprising 15 360 parallel reaction products. These compounds may be viewed as a follow-up library to the above example's product **A** [57]. All products were measured for their IC_{50} inhibitory activity against the serine protease thrombin, a popular test case.

The chemistry space had the dimension 12 × 8 × 60, while the IC_{50}s of the property space were measured on a continuous scale between 100 μM and 90 nM. The presence of the desired products was verified by mass spectroscopy. This structure-property landscape was then used to investigate the optimization behavior of various GAs and parameter sets. Several rough conclusions could be drawn that may guide the application of GAs for evolutionary chemistry. For example, low mutation rates and high crossover rates were found to be generally useful. In this case, the differences between GAs using binary or real-valued encoding of building blocks were small. To assess the GA's improved selection rate versus random selection a simple parameter is useful

$$Lr(x+1) = \sum_{1}^{n} \frac{IC50x}{n} - \frac{IC50(x+1)}{n} \tag{6}$$

where, assuming a population size n, the difference (Lr) at generation (x + *1*) of the average IC_{50}s of the parents population x and the child population (x + *1*) gives a measure for the "learning rate" per individual. In random selection, this value is close to 0 as shown in Fig. 8 for a population size of 80 and random selection.

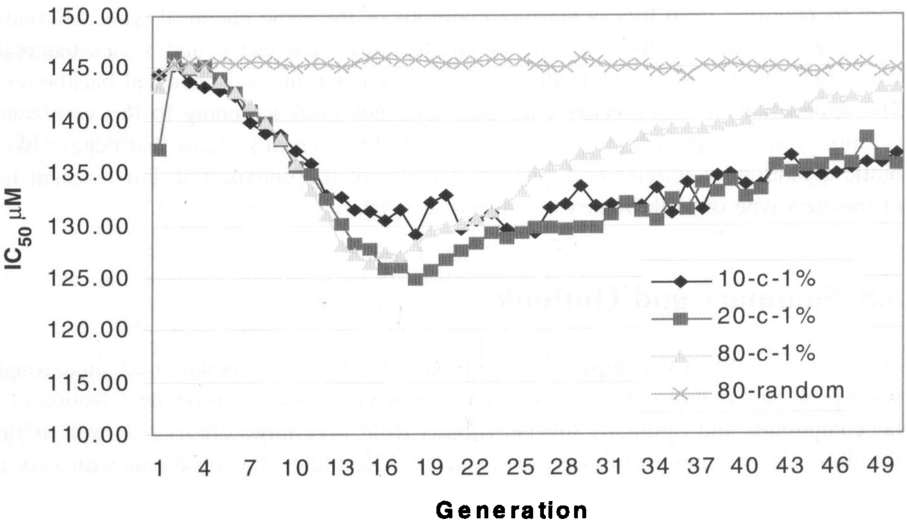


Figure 8. The average activity of the new children population at each generation where only new compounds are considered that have not yet been made. A total of 100 GAs for each parameter set was run and averaged to give one curve using the complete library of 15 360 reaction products. The mutation rate was 1 %, crossover rate 100 %, and population sizes were 10, 20, and 80, respectively. For comparison, random selection is shown with a population size of 80, a mutation rate of 100 % and 100 % crossover rate.

An interesting feature of the GAs shown in Fig. 8 is that, by around generation 16, the most active compounds have already been discovered. Since their resynthesis is not allowed, the search space of 15 360 becomes depleted of active compounds and the "learning" decreases. Also, the speed of learning is not very different when different population sizes are considered! Thus, smaller populations learn more per individual and, as a consequence, it is not very useful to increase the population sizes considerably.

The yield of a certain chemical reaction is a parameter that influences the biological result when crude reaction products are tested. Thus, it should be also possible to use evolutionary feedback functions to optimize reaction conditions, resulting in higher yields of the respective biologically active reaction product. We applied a GA to optimize the yield of a multicomponent reaction by varying the solvents, time points for reagent addition, and concentrations of the starting materials [57]. After several cycles of optimization, a different, higher yielding reaction protocol for the Ugi four-component reaction was found.

In the context of evolutionary chemistry, the world of multicomponent reactions is especially interesting for constructing biologically relevant molecules. Some prebiotic reactions that are believed to have yielded the building blocks of life (e.g., amino acids and nucleotides) were likely to have been multicomponent reactions. Classical combinatorial chemistry is understood as the variation of substituents around a common chemical core by choosing from lists of starting materials of the same chemical type. Alternatively, we have proposed a different type of combinatorial chemistry that varies the reactant types and may be useful in the discovery of new reactions and chemical backbones [58]. The application of evolutionary chemistry and biological screening to this combinatorial reaction-finding approach [57] may yield chemical-biological systems that behave like prebiotic evolutionary systems and give rise to tales of the unexpected. For a recent review of this new type of evolutionary chemistry, see [59].

8.8 Summary and Outlook

Heuristic evolutionary computation methods like GAs are being used increasingly to solve problems in molecular diversity. Examples have been given of the selection of similar compounds and optimally diverse subsets from very large libraries as well as finding solutions for parameterization problems in QSAR studies. For problems with very large, multidimensional search spaces that are NP-complete, evolutionary computation seems to be the method of choice, particularly because the implementation of EAs is relatively straightforward. As several authors have shown, problems of combinatorial chemistry are especially amenable to GA-driven compound selection since the systematic discrete discontinuous search space fits well with the strengths of combinatorial optimization. In addition, the optimization of a lead molecule to a drug and the development of materials are understood by many scientists as evolutionary processes. These advantages may be the reason why GAs seem to be favored over other alternative computational techniques such as cell-based methods, fuzzy logic, cellular automata, or neural networks. Evolutionary algorithms are combinatorial optimization procedures lacking a clear theory to guide the design and parameterization [29]. Therefore, optimal GAs have to be developed using real experimental data with trial and error. The central task in the implementation of EAs is the identification of suitable molecular descriptors. The disadvantage of GAs for general applications is their intrinsic sequential nature, as learning takes place only over a rather unpredictable number of optimization cycles. The speed of a GA-driven optimization therefore depends strongly on the cycle time of synthesis and screening, which prevents the use of long synthesis procedures for compound generation.

Evolution in Nature generates diversity and similarity by changing, for example, an essential amino acid required for biological activity by a single mutation in the DNA. By analogy, GAs are used to generate diversity and similarity by mutation and crossover with respect to the applied diversity selection function for small molecules. To improve computational methods, it may be useful to study evolutionary processes in Nature in more detail. By understanding the role and underlying rules of mutations and crossover, continuous and discontinuous transitions, one may develop evolutionary computation methods that are more efficient and reliable. For example, it is already known that mutations often do not occur with simple statistical probability in DNA, but are more likely in positions where they may produce more "useful" proteins. Other new methods may include the application of information theory and stochastic algorithms to develop GAs to aid in the task of selecting optimally diverse compounds [60] for biological screening.

References

[1] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver, *Combinatorial Optimization*, Wiley, London, **1997**.

[2] I. Ugi, J. Bauer, K. Bley, A. Dengler, A. Dietz, E. Fontain, B. Gruber, R. Herges, M. Knauer, K. Reitsam, N. Stein, Computer-assisted Solution of Chemical Problems – the Historical Development and the Present State of the Art of a New Discipline of Chemistry, *Angew. Chem. Int. Ed. Engl.* **1993**, *32*, 201–227.

[3] I. Ugi, M. Wochner, E. Fontain, J. Bauer, B. Gruber, R. Karl, Chemical Similarity, Chemical Distance and Computer Assisted Formalized Reasoning by Analogy, in M. A. Johnson, G. M. Maggiora (Eds.), *Concepts and Applications of Molecular Similarity*, John Wiley & Sons Inc., New York, **1990**, pp. 239–288.

[4] D. Weininger, SMILES: a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules, *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.

[5] C. Le Bret, Rebuilding Connectivity Matrices from Two-Atom Fragments Using the Genetic Algorithm. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 678–683.

[6] R. C. Glen, A. W. R. Payne, A Genetic Algorithm for the Automated Generation of Molecules within Constraints, *J. Comput.-Aided Mol. Des.* **1995**, *9*, 181–202.

[7] L. Weber, S. Wallbaum, C. Broger, K. Gubernator, A Genetic Algorithm Optimizing Biological Activity of Combinatorial Compound Libraries, *Angew. Chem. Int. Ed. Engl.* **1995**, *107*, 2453–2454.

[8] D. J. Wild, P. Willett, Similarity Searching in Files of Three-Dimensional Chemical Structures. Alignment of Molecular Electrostatic Potential Fields with a Genetic Algorithm, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 159–167.

[9] R. P. Sheridan, S. K. Kearsley, Using a Genetic Algorithm to suggest Combinatorial Libraries, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 310–320.

[10] MOLGEN, Chemical Concepts, Boschstr. 12, D-69469 Weinheim, Germany.

[11] MOLCONN-Z, eduSoft, P. O. Box 1811, Ashland VA 23005

[12] D. M. Bayada, H. Hamersma, V. J. van Geerestein, Molecular Diversity and Representativity in Chemical Databases, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1–10.

[13] W. Fontana, P. Schuster, Continuity in Evolution: On the Nature of Transitions, *Science* **1998**, *280*, 1451–1455.

[14] V. J. Gillet, P. Willett, J. Bradshaw, D. V. S. Green, Selecting Combinatorial Libraries to Optimize Diversity and Physical Properties, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 169–177.

[15] D. Gorse, A. Rees, M. Kaczorek, R. Lahana, Molecular Diversity and its Analysis, *Drug Discovery Today* **1999**, *4*, 257–264.

[16] J. D. Holland, S. S. Ranade, P. Willett, A Fast Algorithm for Selecting Sets of Dissimilar Molecules from Large Chemical Databases, *Quant. Struct.-Act. Relat.* **1995**, *14*, 501–506.

[17] D. B. Turner, S. M. Tyrrell, P. Willett, Rapid Quantification of Molecular Diversity for Selective Database Acquisition, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 18–22.

[18] D. K. Agrafiotis, V. S. Lobanov, An Efficient Implementation of Distance-Based Diversity Measure Based on k-d Trees, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 51–58.

[19] D. Schnur, Design and Diversity Analysis of Large Combinatorial Libraries Using Cell-Based Methods, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 36–45.

[20] M. Snarey, N. K. Terrett, P. Willett, D. J. Wilton, Comparison of Algorithms for Dissimilarity-based Compound Selection, *J. Mol. Graph. Modell.* **1997**, *15*, 372–385.

[21] Y. Tominaga, Data Structure Comparison Using Box Counting Analysis, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 867–875.

[22] R. D. Brown, D. E. Clark, Genetic Diversity: Applications of Evolutionary Algorithms to Combinatorial Library Design, *Exp. Opin. Ther. Patents* **1998**, *8*, 1447–1460.

[23] D. J. Maddalena, Applications of Soft Computing in Drug Design, *Exp. Opin. Ther. Patents* **1998**, *8*, 249–258.

[24] I. Rechenberg, *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution*, Frommann-Holzboog, Stuttgart, **1973**.

[25] J. H. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, MI, **1975**.

[26] D. E. Clark, Some Current Trends in Evolutionary Algorithm in Research Exemplified by Applications in Computer-Aided Molecular Design, *MATCH* **1998**, *38*, 85–98.

[27] L. Weber, Evolutionary Combinatorial Chemistry: Application of Genetic Algorithms, *Drug Discovery Today* **1998**, *3*, 379–385.

[28] B. Levitan, S. Kauffman, Adaptive Walks with Noisy Fitness Measurements, *Mol. Diversity* **1995**, *1*, 53–68.

[29] T. Baeck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, **1997**.

[30] J. H. Holland, *Hidden Order – How Adaptation Builds Complexity*, Addison-Wesley, Reading, MA, **1996**.

[31] M. Forrest, M. Mitchell, Relative Building-block Fitness and the Building-block Hypothesis, in D. Whitley (Ed.), *Foundations of Genetic Algorithms 2*, Morgan Kaufmann, San Mateo, CA, **1993**, pp. 109–126.

[32] C. Stephens, H. Waelbroeck, Schemata Evolution and Building Blocks, *Evol. Comput.* **1999**, *7*, 109–124.

[33] R. Wehrens, E. Pretsch, L. M. C. Buydens, Quality Criteria of Genetic Algorithms for Structure Optimization, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 151–157.

[34] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, Equation of State Calculations by Fast Computing Machines, *J. Chem. Phys.* **1953**, 21, 1087–1092.

[35] L.-X. Sun, Y.-L. Xie, X.-H. Song, J.-H. Wang, R.-Q. Yu, Cluster Analysis by Simulated Annealing, *Comput. Chem.* **1994**, *18*, 103–108.

[36] R. D. Brown, Y. C. Martin, Designing Combinatorial Library Mixtures using a Genetic Algorithm, *J. Med. Chem.* **1997**, *40*, 2304–2313.

[37] A. C. Good, R. A. Lewis, New Methodology for Profiling Combinatorial Libraries and Screening Sets: Cleaning Up the Design Process with HARPick, *J. Med. Chem.* **1997**, *40*, 3926–3936.

[38] V. J. Gillet, P. Willett, J. Bradshaw, The Effectiveness of Reactant Pools for Generating Structurally-Diverse Combinatorial Libraries, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 731–740.

[39] T. J. Hou, J. M. Wang, N. Liao, X. J. Xu, Applications of Genetic Algorithms on the Structure-Activity Relationship Analysis of Some Cinnamamides, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 775–781.

[40] N. A. Shemetulskis, J. B. Dunbar, B. W. Dunbar, D. W. Moreland, C. Humblet, Enhancing the Diversity of a Corporate Database using Chemical Clustering and Analysis, *J. Comput.-Aided Mol. Des.* **1995**, 9, 407–416.

[41] E. J. Martin, J. M. Blaney, M. A. Siani, D. C. Spellmeyer, A. K. Wong, W. H. Moos, Measuring Diversity: Experimental Design of Combinatorial Libraries for Drug Discovery, *J. Med. Chem.* **1995**, *38*, 1431–1436.

[42] D. Liu, H. Jiang, K. Chen, R. Ji, A New Approach to Design Virtual Combinatorial Library with Genetic Algorithm Based on 3D Grid Property, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 233–242.

[43] K. Hasegawa, GA Strategy for Variable Selection in QSAR Studies: Application of GA-based Region Selection to a 3D-QSAR Study of Acetylcholinesterase Inhibitors, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 112–120.

[44] V. Venkatasubramanian, K. Chan, J. Caruthers, Evolutionary Design of Molecules with Desired Properties Using the Genetic Algorithm, *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 188–195.

[45] C. H. Reynolds, Designing Diverse and Focused Combinatorial Libraries of Synthetic Polymers, *J. Comb. Chem.* **1999**, *1*, 297–306.

[46] A. Gobbi, D. Poppinger, Genetic Optimization of Combinatorial Libraries. *Biotechnol. Bioeng.* **1998**, *61*, 47–54.

[47] V. J. Gillet, P. Willett, J. Bradshaw, Identification of Biological Activities Using Substructural Analysis and Genetic Algorithms, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 165–179.

[48] W. Zheng, S. J. Cho, A. Tropsha, Rational Combinatorial Library Design. 1. Focus-2D. A New Approach to the Design of Targeted Combinatorial Chemical Libraries, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 251–258.

[49] S. J. Cho, W. Zheng, A. Tropsha, Rational Combinatorial Library Design. 2. Rational Design of Targeted Combinatorial Peptide Libraries Using Chemical Similarity Probe and the Inverse QSAR Approaches, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 259–268.

[50] W. Zheng, S. J. Cho, C. L. Waller, A. Tropsha, Rational Combinatorial Library Design. 3. Simulated Annealing Guided Evaluation (SAGE) of Molecular Diversity: A Novel Computational Tool for Universal Library Design and Database Mining, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 738–746.

[51] M. T. Reetz, A. Zonta, K. Schimossek, K. Liebeton, K.-E. Jaeger, Creation of Enantioselective Biocatalysts for Organic Chemistry by In Vitro Evolution, *Angew. Chem. Int. Ed. Engl.* **1997**, *36*, 2830–2832.

[52] Y. Yokobayashi, K. Ikebukuro, S. McNiven, I. Karube, Directed Evolution of Trypsin Inhibiting Peptides using a Genetic Algorithm, *J. Chem. Soc. Perkin Trans. 1* **1996**, 2435–2437.

[53] J. Singh, M. A. Ator, E. P. Jaeger, M. P. Allen, D. A. Whipple, J. E. Soloweij, S. Chowdhary, A. M. Treasurywala, Application of Genetic Algorithms to Combinatorial Synthesis: a Computational Approach to Lead Identification and Lead Optimization, *J. Am. Chem. Soc.* **1996**, *118*, 1669–1676.

[54] G. Schneider, U. Hahn, A. Fatemi, G. Müller, P. Wrede, Peptide Design in Machina: Artificial Signal Peptidase I Cleavage-sites are Processed In Vivo. *Minimal Invasive Medizin* **1995**, *6*, 72–77.

[55] P. Wrede, O. Landt, S. Klages, A. Fatimi, U. Hahn, G. Schneider, Peptide Design Aided by Neural Networks: Biological Activity of Artificial Signal Peptidase I Cleavage Sites, *Biochemistry* **1998**, *37*, 3588–3593.

[56] K. Illgen, T. Enderle, C. Broger, L. Weber, Simulated Molecular Evolution in a Full Combinatorial Library, submitted.

[57] L. Weber, K. Illgen, M. Almstetter, Discovery of New Multi-Component Reactions with Combinatorial Methods, *SYNLETT* **1999**, *3*, 366–374.

[58] O. Lack, L. Weber, New Reactions for Combinatorial Chemistry, *Chimia* **1996**, *50*, 445–447.

[59] A. V. Eliseev, J. M. Lehn, Dynamic Combinatorial Chemistry: Evolutionary Formation and Screening of Molecular Libraries, *Curr. Topics Microbiol. Immunol.* **1999**, *243*, 159–172.

[60] D. K. Agrafiotis, Stochastic Algorithms for Maximizing Molecular Diversity, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 841–851.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

9 Evolutionary Algorithms in Crystallographic Applications

Kenneth D. M. Harris, Roy L. Johnston and Benson M. Kariuki

Abbreviations

EA	Evolutionary algorithm
GA	Genetic algorithm

Symbols

R	R-factor
R_{wp}	Weighted profile R-factor
ρ	Scaled R-factor
$I(hkl)$	Integrated intensity of the reflection (hkl)
E	Energy
F	Fitness
$\boldsymbol{S}$	Space group
Γ	The set of variables defining a trial structure
(x, y, z)	Fractional co-ordinates (of an atom or the center-of-mass of a molecule, etc.)
(θ, ϕ, ψ)	Rotation angles defining the orientation of a molecular fragment
τ	Torsion angle
ξ	A general variable in the set Γ
$\Re$	Random number
$\mathbf{P}_j$	The population in generation j of a GA calculation
$\mathbf{I}_{j+1}$	The intermediate population involved in forming generation j+1 of a GA calculation
N_p	Number of structures in a population
N_m	Number of mating operations
N_x	Number of mutation operations
N_g	Number of generations in a GA calculation
R_{min}	Lowest value of R-factor in a population
R_{max}	Highest value of R-factor in a population
R_{ave}	Average value of R-factor in a population
$(a, b, c, \alpha, \beta, \gamma)$	Unit cell parameters
$R_{wp}^{(i)}$	Weighted profile R-factor for region i of a powder diffraction pattern
R_{wp}^{total}	Overall weighted profile R-factor (summed over all regions)

9.1 Introduction

Several aspects of crystallography, which we interpret in its widest sense to mean the study of properties of crystalline solids, are directly amenable to investigation using evolutionary algorithms. Crystalline solids are characterized by well-defined structures with long-range, three-dimensional (3-D) periodic ordering of atoms, ions or molecules (see [1] and [2] for general overviews of crystallography). Thus, a crystal structure may be represented by specifying the periodicity of the structure (the concept of the "lattice") and the positions of the atoms, ions or molecules within the repeating unit. The repeating unit within the structure is termed the "unit cell". Furthermore, on taking account of the symmetry of the crystal structure, the unique part of the unit cell contents (the "asymmetric unit") represents only a fraction of the complete unit cell (except in the case of space group P1). Thus, the periodic structural properties of a macroscopic crystalline solid (comprising billions and billions of atoms, ions or molecules) may actually be defined in terms of an amazingly small number of parameters – the unit cell dimensions $\{a, b, c, \alpha, \beta, \gamma\}$, the space group symmetry $\boldsymbol{S}$, and the fractional co-ordinates $\{(x, y, z)_j\}$ of each atom (labeled j) in the asymmetric unit. Here we ignore structural defects (i.e., local deviations from the perfect periodic structure discussed above) and we ignore the structure at the surfaces of the crystal (by definition, the surface structure is nonperiodic in 3-dimensions). In the diffraction pattern from a crystalline solid, the positions of the diffraction maxima depend on the periodicity of the structure (i.e., the dimensions of the unit cell), whereas the relative intensities of the diffraction maxima depend on the distribution of scattering matter (i.e., the atoms, ions or molecules) within the repeating unit.

In general, the properties of a crystalline solid (except those properties that are influenced by defects or by the crystal surfaces) depend directly on the bulk crystal structure, and may therefore be expressed, at least in principle, as a direct function of the structural parameters $\{a, b, c, \alpha, \beta, \gamma\}$, $\boldsymbol{S}$ and $\{(x, y, z)_j\}$. As such, there are considerable opportunities to exploit evolutionary algorithms (EAs) in crystallographic applications by making use of the relationships between these structural parameters and properties of crystals. For example, if the structure and a particular property are both known experimentally, then evolutionary algorithms could be applied to establish the relationship between the structure and the property in question (for example, by finding optimum values for the parameters in relationships with known functional forms). Alternatively, if a particular property is determined either experimentally or computationally for a crystalline solid of unknown structure, and if the dependence of this property on the crystal structure is already known and quantified by a well-defined relationship, then evolutionary algorithms may be applied to establish the best set of structural parameters $\{(a, b, c, \alpha, \beta, \gamma); \boldsymbol{S}; (x, y, z)_j\}$ to fit the experimentally or computationally determined property.

In this chapter, we focus on the implementation and application of evolutionary algorithms in a variety of crystallographic contexts. All examples discussed here concern procedures for establishing the best set of structural parameters to fit properties determined experimentally or computationally. In this regard, the main experimentally measured property is the diffraction pattern of the material (we focus primarily on powder diffraction data), and the procedure to establish the best set of structural parameters by fitting experimental diffraction data is termed "structure determination" (the term "structure solution" is defined in section 9.2.1). The main computationally derived property is the potential energy of the crystal, and the procedure to establish the set of structural parameters corresponding to minimum energy is termed "structure prediction". We also mention a hybrid approach incorporating both diffraction data and energy for structure determination. In these applications, the "population" in the evolutionary algorithm comprises a set of trial crystal structures, with the "genetic code" for each structure represented by the structural parameters $\{(a, b, c, \alpha, \beta, \gamma); \boldsymbol{S}; (x, y, z)_j\}$, or an appropriate subset of these parameters. The quality ("fitness") of a given structure is assessed from some property that depends on (and is determined directly from) the values of the structural parameters. Such properties include crystallographic R-factor (which quantifies the level of agreement between an experimental diffraction pattern and the diffraction pattern calculated from a given set of structural parameters) and energy (which may be computed from a given set of structural parameters provided an appropriate potential energy function is available for the system of interest).

The main focus of this chapter is the application of genetic algorithm techniques to solve crystal structures of molecular materials directly from powder diffraction data (section 9.2). In this application, the aim is to find the set of atomic positions $\{(x, y, z)_j\}$ [knowing the unit cell $\{a, b, c, \alpha, \beta, \gamma\}$ and space group $\boldsymbol{S}$] that give the optimal representation of an experimental powder diffraction pattern. We then describe the application of a genetic algorithm, based on the use of structure-based cost functions and computed lattice energies, for the prediction of inorganic crystal structures (section 9.3). We also discuss the application of a genetic algorithm for indexing powder diffraction patterns – that is, to find the unit cell parameters $\{a, b, c, \alpha, \beta, \gamma\}$ directly from the experimental powder diffraction data (section 9.4). Finally, some applications of genetic algorithms in other crystallographic contexts, including protein crystallography, are discussed in section 9.5. General introductions to genetic algorithms may be found in references [3–5] and other chapters of this book.

9.2 Crystal Structure Solution from Powder Diffraction Data using Genetic Algorithms

9.2.1 Background

Crystal structure determination from diffraction data (either for single crystal or powder samples) can be divided into three stages: (i) unit cell determination ("indexing") and space group assignment; (ii) structure solution; and (iii) structure refinement. In *structure solution*, the aim is to derive an approximate description of the crystal structure by direct consideration of the experimental diffraction data, but starting from no knowledge of the actual arrangement of atoms, ions or molecules in the unit cell. If the approximate structure solution is a sufficiently good representation of the true structure, a good quality crystal structure may then be obtained subsequently (in the *structure refinement* stage) by refinement of this structural model against the experimental diffraction data. For powder diffraction data, refinement of crystal structures can be carried out fairly routinely using the Rietveld profile refinement technique [6, 7]. In general, structure solution from powder diffraction data is a significantly greater challenge than structure refinement. A schematic overview of the different stages of structure determination from powder diffraction data is shown in Fig. 1.

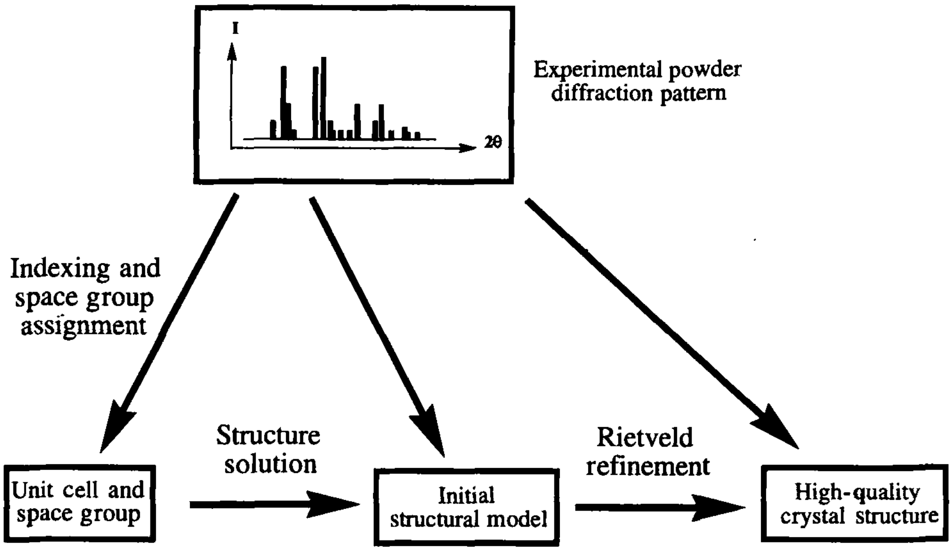

I
2θ
Experimental powder diffraction pattern
Indexing and space group assignment
Structure solution
Rietveld refinement
Unit cell and space group
Initial structural model
High-quality crystal structure

Figure 1. Diagram illustrating the different stages involved in determination of a crystal structure from powder diffraction data.

As many important materials cannot be prepared as single crystals appropriate for conventional single crystal diffraction studies (nor for synchrotron-based microcrystal diffraction techniques), the ability to solve crystal structures directly from powder diffraction data promises to open up many new avenues of structural science. Although single crystal and powder diffraction patterns contain essentially the same information, in the former case this information is distributed in 3-D space whereas in the latter case the 3-D diffraction data are "compressed" into one dimension, which generally leads to considerable overlap of peaks in the powder diffraction pattern. As discussed below, such peak overlap gives rise to significant difficulties in solving crystal structures from powder diffraction data. The techniques currently available for structure solution from powder diffraction data can be subdivided into two categories – "traditional" approaches and "direct-space" approaches.

The *traditional approach* [8–15] for solving crystal structures from powder diffraction data is to use the intensities I(*hkl*) of individual reflections *extracted* directly from the powder diffraction pattern. Structure solution may then be attempted by using these I(*hkl*) data in the types of structure solution calculation that have been developed for single crystal diffraction data, such as direct methods or Patterson methods. However, as there is usually extensive overlap of peaks in the powder diffraction pattern (particularly for low-symmetry structures), it is often difficult to extract unambiguous values of the intensities I(*hkl*) of the individual diffraction maxima. Unreliable values of the intensities I(*hkl*) can lead to severe difficulties in subsequent attempts to solve the structure using such "single-crystal-like" approaches. In order to overcome this problem, we either require improved techniques for extracting and utilizing peak intensities (there have been several important developments in this area [16–24]), or we require alternative structure solution strategies (see below) that allow the experimental powder diffraction profile to be used directly in its "raw" digitized form, without any requirement to extract the intensities I(*hkl*) of individual diffraction maxima from the experimental powder diffraction pattern.

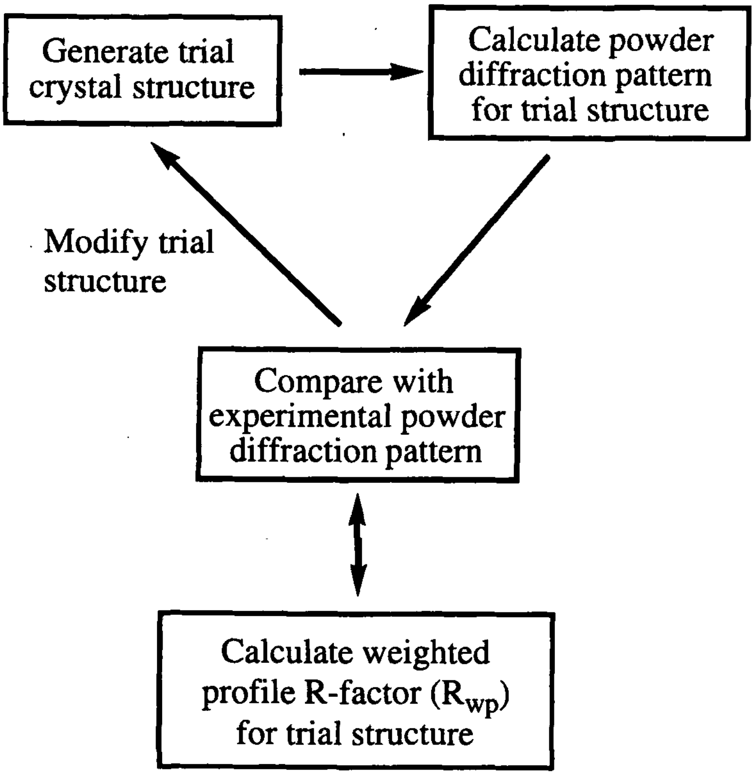

Generate trial crystal structure
Calculate powder diffraction pattern for trial structure
Modify trial structure
Compare with experimental powder diffraction pattern
Calculate weighted profile R-factor (R_{wp}) for trial structure

Figure 2. Schematic illustration of the basis of the "direct-space" approach for crystal structure solution from powder diffraction data.

In the *direct-space approach* [13, 25] for solving crystal structures from powder diffraction data, trial crystal structures are generated in direct space, independent of the experimental powder diffraction data, and the suitability of each trial structure is assessed by direct comparison between the powder diffraction pattern calculated for the trial structure and the experimental powder diffraction pattern (Fig. 2). The comparison between the experimental and calculated powder diffraction patterns is quantified using an appropriate R-factor. To date, almost all reported direct-space approaches have used the weighted profile R-factor R_{wp}, which is the R-factor normally used in Rietveld refinement [6]. The definition of R_{wp} is:

$$R_{wp} = 100\mathrm{x}\left(\frac{\sum_i w_i[y_i(obs) - y_i(calc)]^2}{\sum_i w_i[y_i(obs)]^2}\right)^{1/2} \tag{1}$$

where y_i(obs) is the intensity of the ith data point in the experimental powder diffraction profile, y_i(calc) is the intensity of the ith data point in the calculated powder diffraction profile, and w_i is a weighting factor for the ith data point. Importantly, R_{wp} considers the entire digitized intensity profile, rather than the integrated intensities of individual diffraction maxima. Thus, R_{wp} takes peak overlap implicitly into account and uses the digitized powder diffraction data directly as measured. The use of R_{wp} to assess the correctness of a structural model clearly requires that the peak shape and peak width parameters used to construct the calculated powder diffraction pattern are consistent with those that define the experimental powder diffraction pattern. In practice, this can be readily established by prior analysis of the peak shapes and peak widths in the experimental powder diffraction pattern. Alternatively, other definitions of the R-factor based on extracted peak intensities may be used to assess the agreement between calculated and experimental powder diffraction data within direct-space structure solution strategies. Our research in this field has focused on the use of the profile R-factor R_{wp}, thus using the measured experimental diffraction data directly without further manipulation.

The direct-space strategy for structure solution aims to find the trial crystal structure that has the lowest possible R-factor, and the approach is equivalent to exploring a hypersurface $R(\Gamma)$ to find the global minimum on the hypersurface. Here Γ represents the set of variables that defines the structure. In principle, any technique for global optimization may be used to locate the lowest point on the $R(\Gamma)$ hypersurface, and success has been achieved using Monte Carlo [25–32] and simulated annealing [33–40] search algorithms as the basis of direct-space techniques for powder structure solution. In addition, the use of grid search techniques has also been reported [41–44]. Recently, genetic algorithms have also been applied in this field [45–53]. In this section, we focus on fundamental and applied aspects of our implementations of genetic algorithm techniques to achieve global optimization with respect to the $R_{wp}(\Gamma)$ hypersurface.

In all the applications of structure solution discussed here, we assume that the unit cell parameters {a, b, c, α, β, γ} and space group $\boldsymbol{S}$ are already known from prior analysis of the experimental powder diffraction pattern. We also assume that the contents of the unit cell (for example, the types and number of atoms, ions or molecules) are known, at least to a sufficiently good approximation, but that the positions and structural arrangement of these constituents within the unit cell are not known. The structure is defined in terms of a "structural fragment", which represents an appropriate collection of atoms within the asymmetric unit, and is "coded" using a set (denoted Γ) of variables that represents the positions of the atoms and/or molecules in the unit cell. For a collection of independent atoms, the set Γ would comprise the fractional co-ordinates $\{(x, y, z)_j\}$ for each of these atoms. However, when the structural fragment comprises a molecule of known constitution, it is greatly advantageous to specify the structural fragment in terms of the position and orientation of the molecule as a whole, together with any variables describing unknown aspects of the intramolecular geometry (such as torsion angles), rather than in terms of the fractional co-ordinates $\{(x, y, z)_j\}$ of each individual atom. Thus, for a molecular fragment, the position may be defined by the fractional co-ordinates $\{x, y, z\}$ of the center of mass or a predefined pivot atom, the orientation may be defined by rotation angles $\{\theta, \phi, \psi\}$ around a set of orthogonal axes, and the intramolecular geometry may be specified by a set of n variable torsion angles $\{\tau_1, \tau_2, \ldots, \tau_n\}$. These concepts may be extended to the case of two or more (identical or nonidentical) molecular fragments within the asymmetric unit.

In general, the bond lengths, bond angles and any known torsion angles (i.e., if the molecular conformation, or aspects of it, are known *a priori*) are generally fixed, and may be taken either from standard values for the type of molecule under study or from the known geometry of a similar molecule. Ideally, the structural fragment should include all atoms with significant scattering power (i.e., all nonhydrogen atoms in the case of powder X-ray diffraction) within the asymmetric unit, but it may sometimes be advantageous to omit certain atoms to restrict the number of variables to be optimized (the omitted atoms may be found later by difference Fourier techniques). Clearly the choice of structural fragment in any given structure solution problem is not unique, although certain choices of structural fragment may be significantly advantageous over others.

The possibility of using genetic algorithm (GA) techniques in structure solution from powder diffraction data was realized independently by two research groups. Our approach [45–47, 51–54] and the approach of Shankland, David and coworkers [48–50] differ in the definition and handling of the fitness function and other aspects of the implementation of the GA. Details may be found in the papers cited. In our approach, the fitness of each member of the population is determined using an appropriate function of R_{wp}, emphasizing the philosophy of using the digitized powder diffraction data directly as measured. Importantly, the use of appropriate functions of R_{wp} (rather than R_{wp} itself) provides considerable scope for optimization of the GA strategy and incorporates the advantages of dynamic scaling. In the approach of Shankland, David and coworkers, the fitness of members of the population is given by the figure-of-merit χ^2, which is based on the intensities of individual reflections $I(hkl)$ extracted from the powder diffraction pattern using the Pawley refinement procedure [55,56]. The definition of χ^2 incorporates the co-variance matrix (as derived from the Pawley refinement), and the use of the co-variance matrix in this way may serve to overcome problems that may otherwise arise when considering the intensities of individual reflections extracted from the powder diffraction pattern.

In this chapter we focus on fundamental aspects of our implementation of the GA method for structure solution from powder diffraction data (section 9.2.2), highlighting some examples of the application of this method (section 9.2.3).

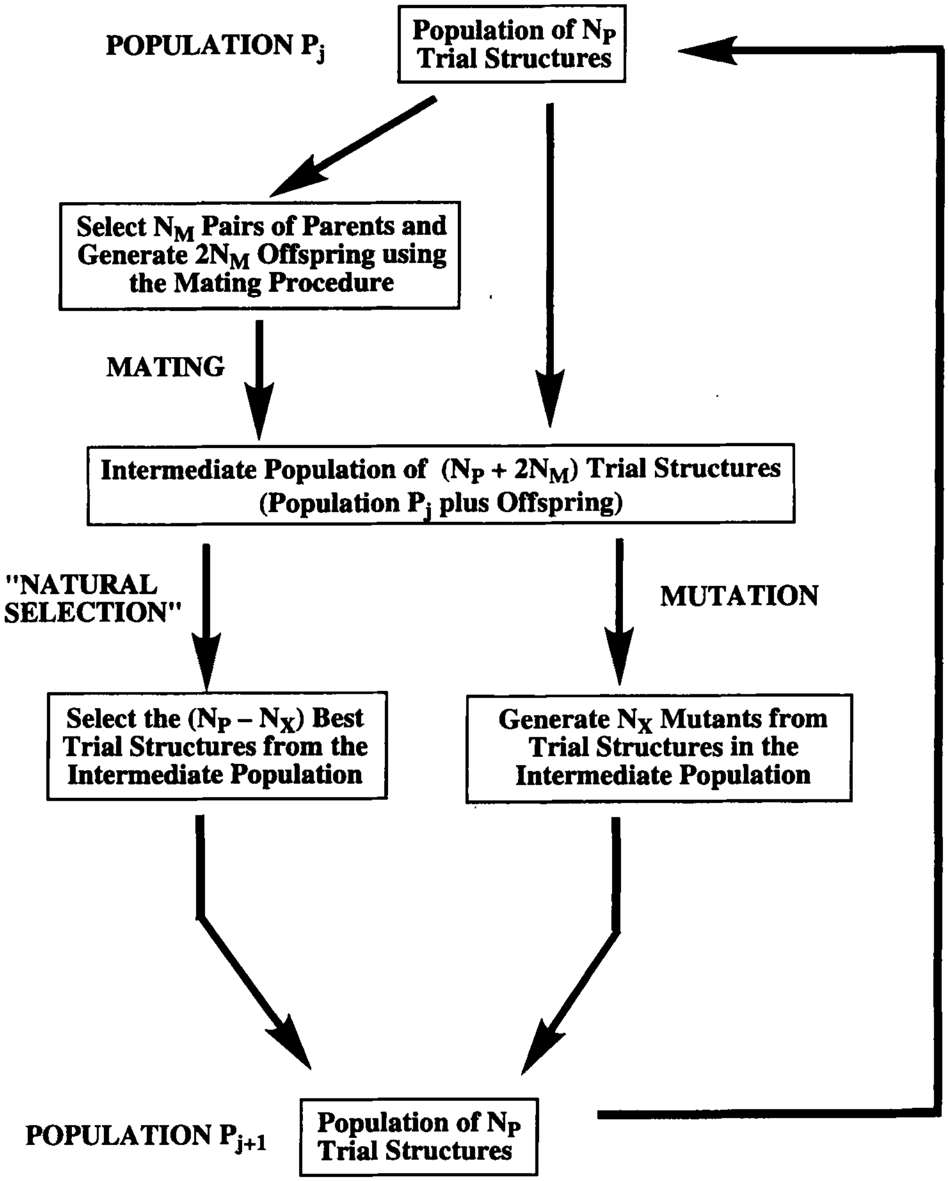

POPULATION P_j
Population of N_P Trial Structures
Select N_M Pairs of Parents and Generate $2N_M$ Offspring using the Mating Procedure
MATING
Intermediate Population of $(N_P + 2N_M)$ Trial Structures (Population P_j plus Offspring)
"NATURAL SELECTION"
MUTATION
Select the $(N_P - N_X)$ Best Trial Structures from the Intermediate Population
Generate N_X Mutants from Trial Structures in the Intermediate Population
POPULATION P_{j+1}
Population of N_P Trial Structures

Figure 3. Flow chart representing the evolution of the population from one generation (population P_j) to the next generation (population P_{j+1}) in the program GAPSS [54].

9.2.2 Methodology

9.2.2.1 Overview

Our GA approach for structure solution from powder diffraction data is implemented in the program GAPSS [45–47, 51–54], and a flow chart describing the operation of this program is shown in Fig. 3. Before running the GA calculation, it is necessary to know the lattice parameters {a, b, c, α, β, γ} and space group $\boldsymbol{S}$ (determined from prior analysis of the powder diffraction pattern) and to make an appropriate choice of the structural fragment. The population comprises a set of trial crystal structures, with each member of the population defined by a set Γ of variables. We note that the values of these variables are real numbers. Each member of the population is characterized uniquely by the values of the variables in Γ, which define its "genetic code".

The initial population $\mathbf{P}_o$ comprises N_p randomly generated structures. During the GA calculation, the population evolves through a sequence of generations, with a given population $\mathbf{P}_{j+1}$ (generation j+1) generated from the previous population $\mathbf{P}_j$ (generation j) by the operations of mating, mutation and natural selection. It is important to note (see below) that mutations create new genetic information within the population, whereas mating serves to redistribute the existing genetic information. The overall scheme for generating population $\mathbf{P}_{j+1}$ from population $\mathbf{P}_j$ in our GA method is summarized in Fig. 3. The number (N_p) of structures in the population remains constant for all generations, and N_m mating operations and N_x mutation operations are involved on passing from population $\mathbf{P}_j$ to population $\mathbf{P}_{j+1}$.

In implementing GA strategies for structure solution, there is considerable scope for diversity and flexibility in the methods and rules that may be used to carry out particular evolutionary operations and in the definition of the fitness function. Furthermore, details of the flow-chart shown in Fig. 3 may differ from one implementation of the GA to another.

9.2.2.2 The Structural Fragment

As discussed in section 9.2.1, each member of the population is a trial crystal structure defined by a set of variables Γ, representing the position, orientation and intramolecular geometry of the structural fragment. The choice of structural fragment for any particular problem is not necessarily unique. For the general case of a rigid molecule, six variables are required: $\Gamma = \{x, y, z, \theta, \phi, \psi\}$. For the general case of a structural fragment with a number (n) of unknown torsion angles τ_i, each member of the population is defined by (6+n) variables: $\Gamma = \{x, y, z, \theta, \phi, \psi, \tau_1, \tau_2, \ldots, \tau_n\}$.

Clearly, when molecules occupy special positions in the crystal structure, the number of variable degrees of freedom may be reduced from (6+n). For example, when a molecular inversion center is coincident with a crystallographic inversion center, the molecule has no translational degrees of freedom, and each member of the population would be defined by (3+m) variables: $\Gamma = \{\theta, \phi, \psi, \tau_1, \tau_2, \ldots, \tau_m\}$ (where m refers to the number of variable torsion angles in half the molecule).

9.2.2.3 The Fitness Function

The quality of a given trial structure is defined by its fitness (F). The value of fitness dictates whether the trial structure survives into subsequent generations (through natural selection) and determines the probability with which it takes part in mating. In our GA, fitness is defined as a function of the weighted profile R-factor R_{wp}. The advantages of using R_{wp} to assess the level of agreement between calculated and experimental powder diffraction patterns were discussed in section 9.2.1. To calculate the powder diffraction profile corresponding to any given trial structure requires: (i) the lattice parameters (to determine peak positions); (ii) the atomic positions, as defined by (and obtained directly from) the parameters in the set Γ, and atomic displacement parameters (to determine peak intensities); (iii) analytical functions to describe the peak shapes and peak widths (as a function of diffraction angle 2θ); and (iv) a description of the background intensity. The shape of a peak in a powder diffraction pattern depends on features of both the instrument and the sample, and different types of peak shape function are appropriate under different circumstances. The most widely used peak shape for powder X-ray diffraction data is the pseudo-Voigt function, which allows flexible variation between Gaussian and Lorentzian character [6,7]. After constructing the calculated powder diffraction pattern for a given trial structure Γ, the value of R_{wp} is obtained by fitting the calculated and experimental powder diffraction patterns through variation of the overall scale factor (which serves to put the calculated and experimental powder diffraction patterns on the same absolute intensity scale).

To determine the fitness of a given member of the population from its value of R_{wp}, it is advantageous to consider the following scaled R-factor:

$$\rho = \frac{R_{wp} - R_{min}}{R_{max} - R_{min}} \tag{2}$$

where R_{min} and R_{max} are the lowest and highest values of R_{wp} in the population, respectively. The value of ρ lies in the range $0 \leq \rho \leq 1$. The fitness is then expressed as an appropriate function of ρ, and the following fitness functions have been used in our work:

exponential $F(\rho) = \exp(-S\rho)$ (3)
tanh $F(\rho) = 1/2\,[1 - \tanh\{2\pi(2\rho - 1)\}]$ (4)
power $F(\rho) = 1 - \rho^n$ (5)
cosine $F(\rho) = 1/2[1 + \cos(\pi\rho/2)]$ (6)

In each case, $F(\rho)$ takes its highest value (i.e., $F(\rho) = 1$) when $\rho = 0$ (i.e., $R_{wp} = R_{min}$) and takes its lowest value when $\rho = 1$ (i.e., $R_{wp} = R_{max}$). The values of R_{min} and R_{max} are continually updated as the population evolves during the GA calculation, representing "dynamic scaling" of the fitness function. For a given fitness function, the ability to discriminate between different structures in the population depends on the value of $(R_{max}-R_{min})$. In general, as $(R_{max}-R_{min})$ becomes smaller, there is greater discrimination in fitness between a given pair of structures.

All fitness functions defined above have maximum fitness F(0) = 1 when $R_{wp} = R_{min}$ (i.e., for the best member of the population). For the *power*, *tanh* and *cosine* functions, the value of minimum fitness (i.e., for $R_{wp} = R_{max}$) is F(1) = 0. For the *exponential* function, on the other hand, the minimum fitness is F(1) = exp(-S) [note that for $S \geq 5$, $F(1) \leq 0.01$]. The major difference between the fitness functions defined above concerns their behavior for values of R_{wp} between R_{min} and R_{max}, as shown in Fig. 4.

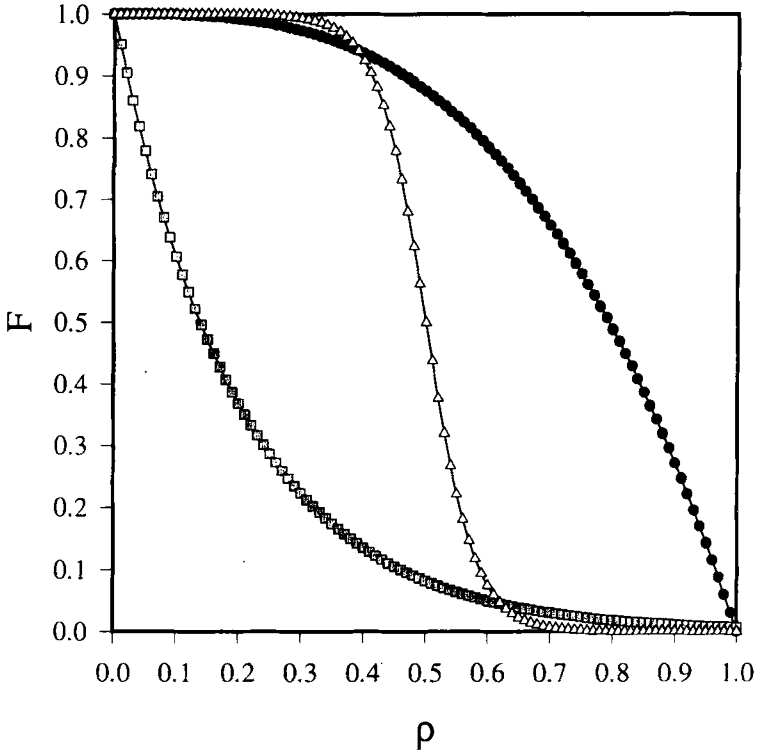


Figure 4. Graphs showing some of the fitness functions discussed in the text: *exponential* function with S = 5 (squares); *tanh* function (triangles); *power* function with n = 3 (circles).

The *exponential* function discriminates well between different good structures, as a wide range of values of F(ρ) is covered by good structures (with low values of ρ). On the other hand, the curve is shallow around $\rho = 1$, and a wide range of structures with high values of ρ all have very low fitness. The *power* function is convex for n > 1 and gives good discrimination among poor structures (for which the curve is steepest) but little discrimination among good structures. For example, for n = 3 (Fig. 4), all structures with $\rho \leq 0.5$ have $F(\rho) \geq 0.9$. In this regard, the behaviour of the *power* function with n > 1 is directly opposite to the behavior of the *exponential* function. The *tanh* function does not discriminate significantly among good structures and does not discriminate significantly among poor structures. However, the *tanh* function does discriminate well among structures within the intermediate region $0.3 \leq \rho \leq 0.7$ (note the step-like character of the *tanh* function). The *cosine* function has similar properties to the *tanh* function, but is less steep in the intermediate region. In practice, the *linear* fitness function (i.e., the *power* function with n = 1) has been employed in some of our work.

In many cases, it may be advantageous to change the definition of fitness function systematically during the GA calculation. For example, the *tanh* function (which discriminates mainly between good structures and poor structures, but provides little discrimination among different good structures) can be advantageous in the early stages of the GA calculation, whereas the *exponential* function (which provides better discrimination between different good structures) can be advantageous in the later stages of the GA calculation.

9.2.2.4 The Mating Procedure

The probability of selecting a given structure to take part as a "parent" in mating is related to its fitness, with structures of high fitness more likely to be selected. In our procedure for selecting parents, a structure (with fitness F) is chosen from the population at random and a random number $\mathfrak{R}$ (with $0 \leq \mathfrak{R} \leq 1$) is generated. The selected structure is then allowed to take part in mating if $F > \mathfrak{R}$. This selection procedure is continued to find a second structure that is allowed to mate with the first. Pairs of structures selected consecutively in this way are allowed to mate with each other, until the required number (N_m) of mating operations has been carried out. Note that a given structure could be selected several times for mating in a given generation.

We now consider some specific methods that may be used to generate offspring by combining the parameters in the sets Γ for the two selected parents. For a rigid structural fragment defined by six parameters $\{x, y, z, \theta, \phi, \psi\}$, one approach for mating is single-point crossover in which the sets of variables defining the two selected parents are cut and spliced between the positional and orientational parameters to produce two offspring. Thus, the parents $\{x_a, y_a, z_a \mid \theta_a, \phi_a, \psi_a\}$ and $\{x_b, y_b, z_b \mid \theta_b, \phi_b, \psi_b\}$ would lead to the two offspring $\{x_a, y_a, z_a \mid \theta_b, \phi_b, \psi_b\}$ and $\{x_b, y_b, z_b \mid \theta_a, \phi_a, \psi_a\}$. Such single-point crossover at a fixed point has the potential disadvantage that, for a given pair of parents, it will always lead to the same pair of offspring, and may therefore contribute to loss of diversity within the population. Although the use of single-point crossover between the positional and orientational parameters is attractive in view of the physical significance associated with separating the positional and orientational information, there is no guarantee that this represents the most efficient approach for finding the optimal structure solution, and single-point crossover at randomly selected positions within the set of variables or multiple-point crossover can have several advantages. In these methods, a given pair of parents could produce several different pairs of offspring. An alternative procedure is to take the six variables from each parent and distribute them (on a random basis) between the two offspring, with no restriction on which combination of groups may come from each parent.

For a structural fragment with two variable torsion angles, one method for mating divides the eight variables that define each parent into four groups $\{x, y, z \mid \theta, \phi, \psi \mid \tau_1 \mid \tau_2\}$.

For mating between two selected parents, the four groups are divided into two sets of two groups, which can be done in three different ways:

(i) $\{x, y, z \mid \theta, \phi, \psi\}$ and $\{\tau_1 \mid \tau_2\}$
(ii) $\{x, y, z \mid \tau_1\}$ and $\{\theta, \phi, \psi \mid \tau_2\}$
(iii) $\{x, y, z \mid \tau_2\}$ and $\{\theta, \phi, \psi \mid \tau_1\}$

In a given mating operation, one of these ways of dividing the four groups is chosen (with equal probability), and two offspring are generated by taking the first set of two groups from one parent and the second set of two groups from the other parent, and vice versa. Thus, mating the parents $\{x_a, y_a, z_a \mid \theta_a, \phi_a, \psi_a \mid \tau_{1a} \mid \tau_{2a}\}$ and $\{x_b, y_b, z_b \mid \theta_b, \phi_b, \psi_b \mid \tau_{1b} \mid \tau_{2b}\}$ will lead with equal probability to one of the following pairs of offspring:

(i) $\{x_a, y_a, z_a \mid \theta_a, \phi_a, \psi_a \mid \tau_{1b} \mid \tau_{2b}\}$ and $\{x_b, y_b, z_b \mid \theta_b, \phi_b, \psi_b \mid \tau_{1a} \mid \tau_{2a}\}$
(ii) $\{x_a, y_a, z_a \mid \theta_b, \phi_b, \psi_b \mid \tau_{1a} \mid \tau_{2b}\}$ and $\{x_b, y_b, z_b \mid \theta_a, \phi_a, \psi_a \mid \tau_{1b} \mid \tau_{2a}\}$
(iii) $\{x_a, y_a, z_a \mid \theta_b, \phi_b, \psi_b \mid \tau_{1b} \mid \tau_{2a}\}$ and $\{x_b, y_b, z_b \mid \theta_a, \phi_a, \psi_a \mid \tau_{1a} \mid \tau_{2b}\}$

An alternative mating procedure takes a weighted average (interpolation) of corresponding parameters from the two parents, leading from two parents $\{x_a, y_a, z_a \mid \theta_a, \phi_a, \psi_a \mid \tau_{1a} \mid \tau_{2a}\}$ and $\{x_b, y_b, z_b \mid \theta_b, \phi_b, \psi_b \mid \tau_{1b} \mid \tau_{2b}\}$ to one offspring $\{x_o, y_o, z_o \mid \theta_o, \phi_o, \psi_o \mid \tau_{1o} \mid \tau_{2o}\}$, with $\xi_o = (1 - \lambda)\, \xi_a + \lambda\, \xi_b$ (where ξ represents each of the variables). The value of λ is in the range $0 \leq \lambda \leq 1$, and may be weighted according to the fitness values of the two parents.

Clearly, many different options exist for mating, each of which may be more or less advantageous in different circumstances. For systems involving greater numbers of parameters than the examples given above, more complex rules governing the mating procedure may be adopted (see also section 9.2.3).

9.2.2.5 The Intermediate Population

As the number of mating operations in each generation is N_m and each mating operation leads to two offspring, the total number of offspring produced in each generation is $2N_m$. An intermediate population ($\mathbf{I}_{j+1}$) containing N_p+$2N_m$ structures is then constructed, taking the N_p structures from the previous generation ($\mathbf{P}_j$) and the $2N_m$ offspring generated by the mating procedure. At this stage the values of R_{min} and R_{max} for the intermediate population are determined, and the values of fitness for all members of the intermediate population are re-calculated. Although each structure carried through to the intermediate population from the previous generation has the same value of R_{wp} as in the previous generation, the value of fitness may change (as the values of R_{min} and R_{max} may differ between populations $\mathbf{P}_j$ and $\mathbf{I}_{j+1}$). If two or more structures are identical within predefined tolerance limits, all but one of these structures is eliminated from the intermediate population. The structures in the intermediate population are then ranked according to their fitness, in preparation for natural selection.

9.2.2.6 The Mutation Procedure

In each generation, a number (N_x) of mutant structures are generated in order to introduce new genetic information within the population, and thus to help to maintain diversity. In our mutation procedure, N_x "parent" structures are selected at random from the intermediate population, and a new mutant structure is generated from each selected "parent" by introducing random changes to the values of one or more variables in its genetic code (Γ). It is important to note that the "parent" structures used to create the mutants are not replaced by the mutants, but remain within the intermediate population.

As an example, for a structural fragment with two torsional degrees of freedom, one method for carrying out mutation is to randomly select two of the four groups of parameters $\{x, y, z \mid \theta, \phi, \psi \mid \tau_1 \mid \tau_2\}$ and to assign a new random value to one parameter within each of the selected groups. For systems involving greater numbers of parameters, more complex rules governing the mutation procedure may be adopted.

In principle, the mutation procedure could be introduced in several different ways within the overall scheme for converting population $\mathbf{P}_j$ to population $\mathbf{P}_{j+1}$. However, it is important (as in the scheme shown in Fig. 3) that the mutant structures are allowed the opportunity to take part in mating operations *before* the process of natural selection is carried out. Thus, while many of the mutants will themselves represent poor quality structures (and will be rejected subsequently from the population at the natural selection stage), they may nevertheless contain useful genetic information which may be passed into the population through the mating procedure.

An alternative technique is dynamic mutation, in which selected parameters are subjected to random *displacements* from their values in the "parent" structure. Thus, for a particular variable ξ in the set Γ, the new (mutated) value ξ_m is given by

$$\xi_m = \xi_p + (\mathcal{R} \cdot \Delta\xi_{max}) \qquad (7)$$

where ξ_p is the value of ξ in the "parent" structure, $\mathcal{R}$ is a random number between −1 and +1, and $\Delta\xi_{max}$ is a maximum allowed displacement. Dynamic mutation is particularly useful for fine-tuning the population in the later stages of the GA calculation, when static mutation may in general cause perturbations that are too large (leading to structures with very low fitness which would probably be rejected in the next generation). There is considerable scope for optimizing the strategy of using static mutation in the initial stages and then introducing dynamic mutation in the later stages of the GA calculation.

9.2.2.7 Natural Selection

The (j+1)th generation ($\mathbf{P}_{j+1}$) is produced by taking the N_p–N_x best (highest fitness) members of the intermediate population $\mathbf{I}_{j+1}$ together with the N_x mutant structures generated from population $\mathbf{P}_j$. The values of R_{wp} for the mutants are calculated and the new values of R_{min} and R_{max} for population $\mathbf{P}_{j+1}$ are evaluated, and the fitness of each structure in the new population $\mathbf{P}_{j+1}$ is then determined. The complete cycle shown in Fig. 3 is then repeated for a specified number (N_g) of generations, or until some predetermined termination criterion is satisfied (such as reaching a sufficiently low value of R_{min}). We note that the value of R_{wp} for the best structure in population $\mathbf{P}_{j+1}$ must be less than or equal to the value of R_{wp} for the best structure in population $\mathbf{P}_j$, and thus R_{min} cannot increase from one generation to the next. The population size (N_p) remains constant from one generation to the next, and the best structures in a given generation are almost certain to be carried forward into the next generation (i.e., it is unlikely for all offspring generated by mating to have higher fitness than the fittest members of the population in the previous generation). The overall quality of the population, assessed from the average value of R_{wp} (denoted R_{ave}), generally improves from one generation to the next. However, as mutants are included in the calculation of R_{ave}, the value of R_{ave} may sometimes increase slightly on passing from one generation to the next.

9.2.2.8 The Overall Procedure

Typically, our GA structure solution calculations have involved a population size (N_p) of a hundred or a few hundred structures. In each generation, typically 50 to 100 mating operations (N_m) are carried out and 10 to 20 mutant structures (N_x) are generated. In general, a few hundred generations (N_g) are sufficient to obtain the correct structure solution. Actual examples of the choices of these parameters are given in section 9.2.3.

The complete evolutionary cycle involving mating, mutation and natural selection is repeated for a specified number (N_g) of generations or until convergence is reached. The structure solution calculation is started from a randomly generated initial population (with random values for the degrees of freedom in each structure in the population). As the GA is, in part, a stochastic procedure, there is no absolute guarantee that the global minimum will actually be located in a given calculation with a finite number of generations. Thus, a good strategy is to repeat the calculation several times from different random initial populations – finding the same structure or very similar structures repeatedly is a strong indication that these structures represent the global minimum.

9.2.2.9 Schemata

For the GA method to be an efficient approach for global optimization, the evolutionary procedure must be able to recognize the existence of certain combinations of parameters that are associated with high fitness. These groups of parameters are known as "schemata", and their existence is crucial for the success of the GA as an optimization technique. Thus, for a given member of a population, if a subset of the parameters is close to optimal but the other parameters are far from optimal, it is important that the GA calculation can identify the subset that is close to optimal and that the calculation retains and propagates this subset of parameters in the evolutionary process. We are currently carrying out systematic studies to understand more fully the nature of schemata within GA methods for solving crystal structures from powder diffraction data. The parameters that define the structure in the GA calculation could, in principle, be chosen in a number of different ways (in the GA method described here, the parameters are $\{x, y, z, \theta, \phi, \psi, \tau_1, \tau_2, \ldots, \tau_n\}$), and the question of how strongly schemata are expressed may depend on the particular choice of parameters. It is also important that the strategy for mating should ensure that schemata are preserved and propagated in the evolution of the population, rather than destroyed by an inappropriate method for crossover. Clearly the existence and proper handling of schemata is conducive for facile convergence of the GA calculation towards the global minimum.

9.2.3 Examples of Applications

Several structures of varying degrees of complexity have been solved from powder diffraction data using our GA method. As illustrative examples, three case studies are highlighted. First, we describe structure solution calculations for two previously known crystal structures (polymorphs) of a completely flexible molecule – the α and β phases of L-glutamic acid. Second, we describe structure solution of another completely flexible molecule – heptamethylene-bis(diphenylphosphine oxide) $[Ph_2P(O)\cdot(CH_2)_7\cdot P(O)Ph_2]$ (the structure of this material was previously unknown). Third, we describe the structure solution of a steroid molecule with flexible side-groups attached to the steroid ring system – form 2 of fluticasone propionate (the structure of this material was previously unknown).

We emphasize that studies of highly flexible molecules represent the greatest challenges in direct-space structure solution. In such cases, several torsion angles defining the molecular conformation are required as variables in the structure solution calculation. In general, the complexity of the structure solution calculation increases as the dimensionality of parameter space increases. The hypersurfaces are defined by 10 degrees of freedom for L-glutamic acid, 18 degrees of freedom for $Ph_2P(O)\cdot(CH_2)_7\cdot P(O)Ph_2$ and 12 degrees of freedom for fluticasone propionate.

9.2.3.1 Case Study 1: The α and β Phases of L-glutamic acid

L-Glutamic acid $HO_2C(CH_2)_2CH(NH_2)CO_2H$ is known to exist in two different polymorphic forms, denoted the α and β phases. In both crystal structures [57, 58], the L-glutamic acid molecules are in the zwitterionic form (Fig. 5a). Both structures have the orthorhombic space group $P2_12_12_1$. The unit cell parameters for the α phase are: a = 10.28 Å, b = 8.78 Å, c = 7.07 Å. The unit cell parameters for the β phase are: a = 5.16 Å, b = 17.30 Å, c = 6.95 Å. Both structures have one molecule in the asymmetric unit. The powder X-ray diffraction patterns for the α and β phases of L-glutamic acid were recorded at ambient temperature in transmission mode on a Siemens D5000 diffractometer, using Ge-monochromated $CuK_{\alpha 1}$ radiation and a linear position-sensitive detector covering 8° in 2θ. In each case, the total range of 2θ was 3° to 50°, measured in 0.02° steps and collected over 2 hours.

In our GA structure solution calculations for both polymorphs, the structural fragment comprised all nonhydrogen atoms of the L-glutamic acid molecule (Fig. 5b). Standard geometries (bond lengths and angles) were used, with all C–O bond lengths taken to be equal (the C–O single and C=O double bonds are readily assigned during Rietveld refinement). The four torsion angles $\{\tau_1, \ldots, \tau_4\}$ defining the conformation of the L-glutamic

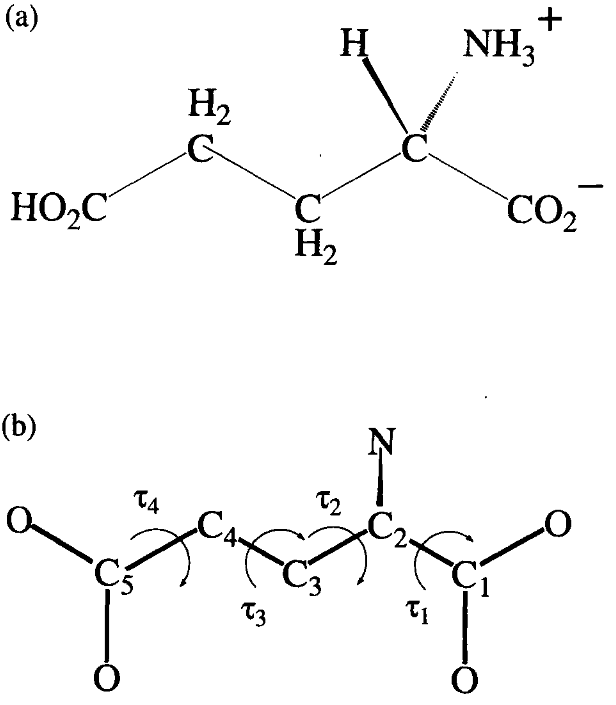


Figure 5. (a) Molecular structure of the L-glutamic acid zwitterion. (b) Structural fragment used in the GA structure solution calculations for L-glutamic acid, showing the variable torsion angles.

acid molecule are indicated in Fig. 5b. The position of the structural fragment was defined by the {x, y, z} co-ordinates of the central carbon atom (C_3) of the molecule. For these calculations, the positional, orientational and torsional variables were all discretized, with grid sizes of 0.01 for all fractional co-ordinates and 10° for all angles. However, in most of our other applications of the GA method, we have chosen not to discretize the variables in this way.

The GA calculation involved the evolution of 100 generations of a population of 100 structures. In each generation, 200 offspring (100 pairs of parents) and 10 mutations were considered. In this case, the *tanh* fitness function was used. For mating and mutation, the 10 variables were considered in terms of six groups $\{x, y, z \mid \theta, \phi, \psi \mid \tau_1 \mid \tau_2 \mid \tau_3 \mid \tau_4\}$. To carry out a mating operation between two parents, the six groups from each parent were distributed between the two offspring, with no restriction on which combination of groups may come from each parent (in each mating operation, the combination of groups coming from each parent was determined on a random basis). In carrying out the mutation procedure on a selected structure, two groups were selected at random, and a random change was made to one variable within each of the selected groups.

The progress of the GA structure solution calculation is assessed from the Evolutionary Progress Plot, which shows the best (R_{min}) and average (R_{ave}) values of R_{wp} for the population as a function of the generation number. The Evolutionary Progress Plots for the α and β phases are shown in Fig. 6. For the β phase, there is a rapid initial drop in R_{min}, whereas the progress for the α phase is more step-like. It is clear that the GA structure solution calculation converges rapidly in both cases.

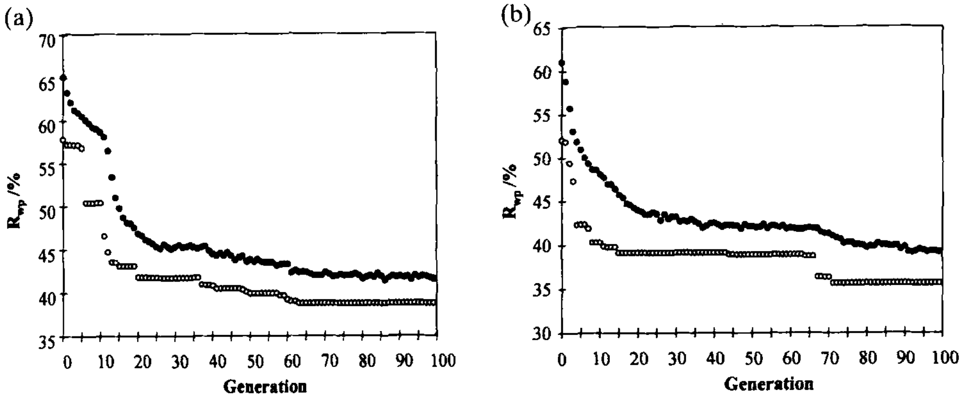


Figure 6. Evolutionary Progress Plots showing the evolution of R_{ave} (filled circles) and R_{min} (open circles), as a function of generation number, in the GA structure solution calculations for (a) the α phase and (b) the β phase of L-glutamic acid.

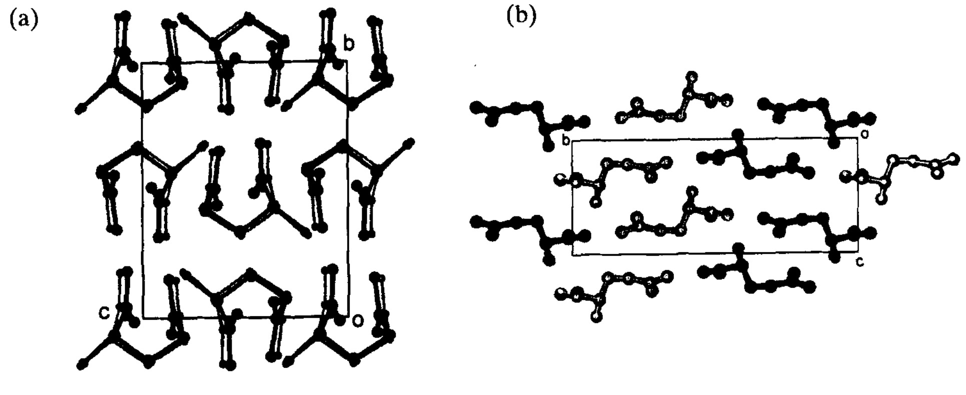


Figure 7. Comparison between the position of the structural fragment in the best structure solution obtained in the GA structure solution calculation (light shading) and the positions of the corresponding atoms in the known crystal structure (dark shading) for (a) the α phase and (b) the β phase of L-glutamic acid.

The best structure solution (i.e., the structure with lowest R_{wp} in the final generation) for the α phase is shown in Fig. 7a, and the best structure solution for the β phase is shown in Fig. 7b. For comparison, the known crystal structures [57, 58] of the α and β phases are also shown. For each phase, the structure solution generated by the GA calculation is in excellent agreement with the known structure, and in each case the structure solution will refine readily (using the Rietveld method) to the known crystal structure. For both phases, the maximum distance between an atom in the structure solution and the corresponding atom in the known crystal structure is less than 0.5 Å. We emphasize that the L-glutamic acid molecule has a significantly different conformation in the α and β phases, and that the GA structure solution calculations successfully found the correct conformation for each phase.

9.2.3.2 Case Study 2: $Ph_2P(O)\cdot(CH_2)_7\cdot P(O)Ph_2$

The structure determination of $Ph_2P(O)\cdot(CH_2)_7\cdot P(O)Ph_2$ [51] represents one of the most complex molecular crystal structures that has been solved directly from powder diffraction data. Structure solution of this previously unknown structure was carried out using our GA method, involving 18 degrees of freedom with 12 variable torsion angles.

The powder X-ray diffraction pattern of $Ph_2P(O)\cdot(CH_2)_7\cdot P(O)Ph_2$ was recorded at ambient temperature using the instrument described in section 9.2.3.1. The total range of 2θ was 5° to 60°, measured in 0.02° steps and collected over 6 hours. The powder pattern was indexed by the program ITO [59], giving the unit cell: a = 12.59 Å, b = 10.20 Å, c = 22.89 Å, β = 105.5°. From systematic absences, the space group was assigned as $P2_1/n$, and density considerations suggested that there is one molecule in the asymmetric unit.

In the GA structure solution calculation, the structural fragment comprised all nonhydrogen atoms of the $Ph_2P(O)\cdot(CH_2)_7\cdot P(O)Ph_2$ molecule. Standard bond lengths and bond angles were used, and the atoms of each benzene ring and all atoms directly bonded to it were constrained to be co-planar. The molecule was subjected to translation and reorientation within the unit cell, together with variation of all 12 torsion angles (Fig. 8) that define the molecular conformation. Thus, each structure in the GA calculation was defined by 18 variables $\{x, y, z, \theta, \phi, \psi, \tau_1, \tau_2, \tau_3, \ldots, \tau_{12}\}$.

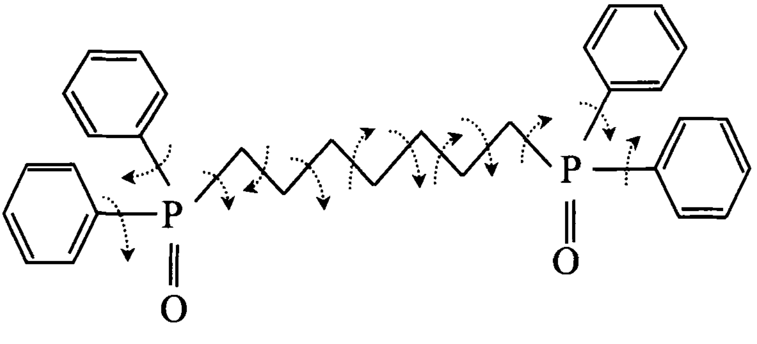


Figure 8. The molecular structure of $Ph_2P(O)\cdot(CH_2)_7\cdot P(O)Ph_2$, showing the variable torsion angles considered in the GA structure solution calculation.

The GA structure solution calculation involved the evolution of 500 generations of a population of 100 structures. In each generation, 100 offspring (50 pairs of parents) and 20 mutations were considered. In this case, the *linear* fitness function $F(\rho) = 1-\rho$ was used. For mating and mutation, the 18 variables were subdivided into 14 groups $\{x, y, z \mid \theta, \phi, \psi \mid \tau_1 \mid \tau_2 \mid \tau_3 \mid \ldots \mid \tau_{12}\}$. In the mating operation, the 14 groups from each parent were distributed between the two offspring, with no restriction on which combination of groups may come from each parent (in each mating operation, this was determined on a random basis). In carrying out the mutation procedure on a selected structure, seven groups were selected at random, and a random change was made to one variable within each of the selected groups. The evolution of R_{wp} during the GA calculation is shown in Fig. 9, and demonstrates that the overall quality of the population improves as the population evolves. The lowest value of R_{wp} in the population decreases significantly, with evidence for a particularly significant evolutionary event at generation 163.

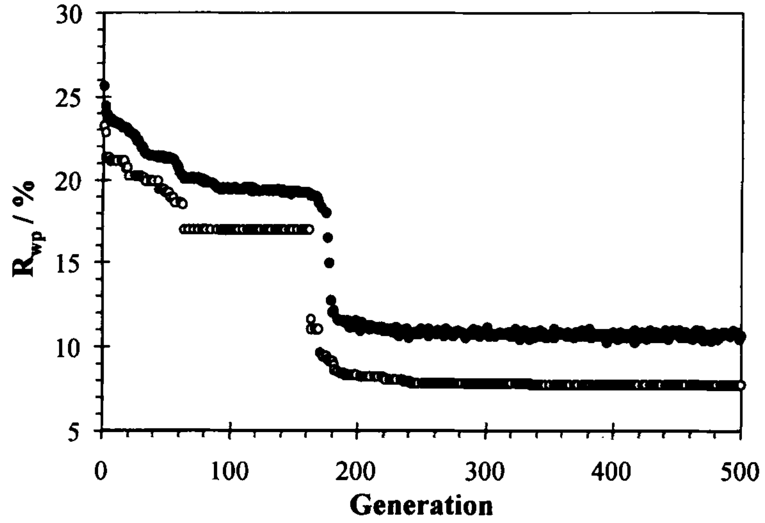


Figure 9. Evolutionary Progress Plot showing the evolution of R_{wp} for the best structure in the population (open circles) and the average R_{wp} for the structures in the population (filled circles) as a function of generation number in the GA structure solution calculation for $Ph_2P(O)\cdot(CH_2)_7\cdot P(O)Ph_2$.

The structure with lowest R_{wp} in the final generation was taken as the starting model for Rietveld refinement. The positions of all nonhydrogen atoms were refined, with standard geometric restraints applied to bond lengths and bond angles. The final Rietveld refinement (Fig. 10) gave $R_{wp} = 5.0\,\%$ and $R_p = 3.8\,\%$. In the crystal structure (Fig. 11), the molecule adopts an unexpected (but completely plausible) conformation, with one *gauche* bond in the $(CH_2)_7$ chain and the other parts of the chain close to all-*trans* conformations. It could not have been predicted in advance that this specific conformation would be adopted in the crystal structure, emphasizing the importance of allowing complete conformational flexibility of the structural fragment in the GA structure solution calculation.

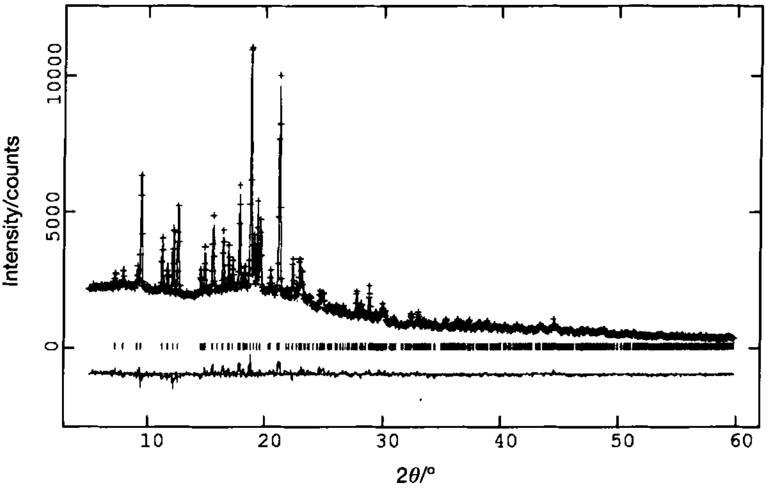


Figure 10. Experimental (+ marks), calculated (solid line) and difference (lower line) powder X-ray diffraction profiles for the Rietveld refinement of $Ph_2P(O) \cdot (CH_2)_7 \cdot P(O)Ph_2$. Reflection positions are marked. The calculated powder diffraction profile is for the final refined crystal structure.

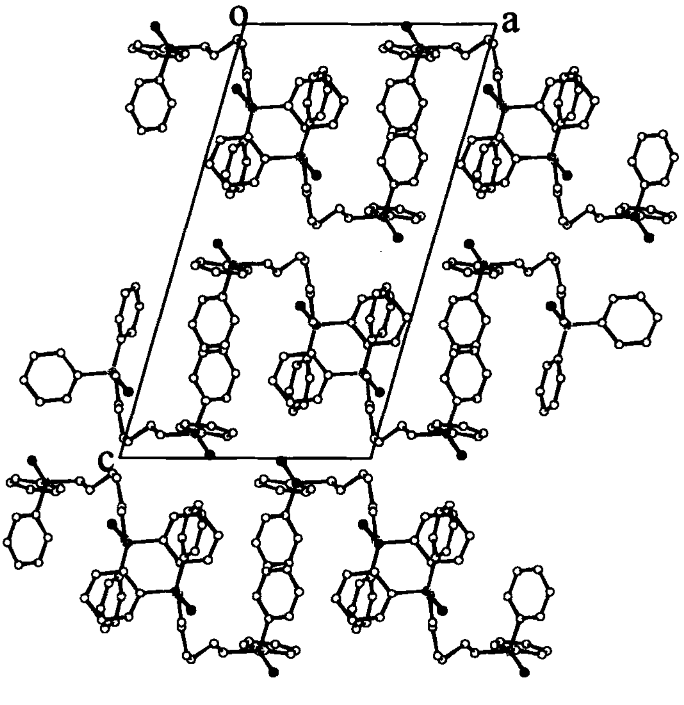


Figure 11. Final refined crystal structure of $Ph_2P(O) \cdot (CH_2)_7 \cdot P(O)Ph_2$ (hydrogen atoms not shown) viewed along the *b*-axis.

9.2.3.3 Case Study 3: Fluticasone Propionate

As the final example, we consider the case of a steroid molecule, fluticasone propionate (FP; $C_{25}H_{31}F_3O_5S$; Fig. 12), which has pharmaceutical importance as a potent anti-inflammatory agent which suppresses inflammation of the bronchial passages in the lungs. FP has been found to exist in two different polymorphic forms. Form 1 can be obtained by crystallization from a variety of solvents (typically acetone), and the crystal structure of form 1 is known. In attempts to produce crystals of FP of controlled size and morphology for pharmaceutical applications, crystallization in a supercritical fluid medium (with ethanol or acetone as solvent) was carried out, but was found to yield a new polymorph (form 2). As form 2 was obtained only by the supercritical crystallization method, yielding polycrystalline powder samples, structural characterization of form 2 could not be carried out by single crystal X-ray diffraction. Instead, the crystal structure of form 2 of FP was determined directly from powder X-ray diffraction data using the GA method [52].

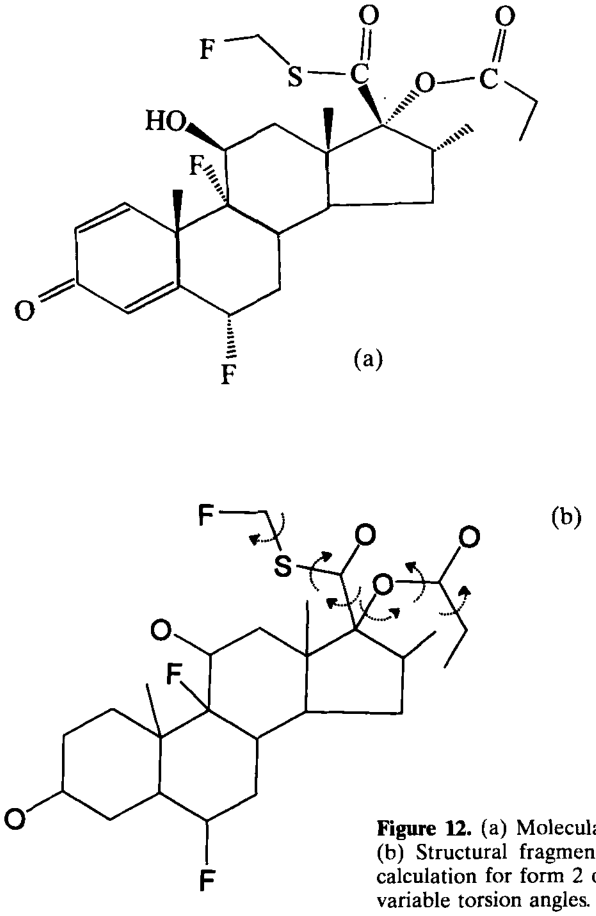


Figure 12. (a) Molecular structure of fluticasone propionate. (b) Structural fragment used in the GA structure solution calculation for form 2 of fluticasone propionate, showing the variable torsion angles.

The powder X-ray diffraction pattern of form 2 of FP was recorded at ambient temperature using the instrument described in section 9.2.3.1. The total 2θ range was 5° to 60°, measured over 12 hours in steps of 0.02°. The powder X-ray diffraction pattern was indexed by the program ITO [59] [the final refined unit cell is a = 23.2434(9) Å, b = 13.9783(5) Å, c = 7.6510(3) Å]. Systematic absences are consistent with space group $P2_12_12_1$, and density considerations suggest that there is one molecule in the asymmetric unit.

In the GA structure solution calculation, all nonhydrogen atoms of the FP molecule were used to define the structural fragment. The tetracyclic ring system was considered as a rigid unit, comprising one planar six-membered ring (designated A), two six-membered rings in the chair conformation (B and C), and a five-membered ring in an envelope conformation (D). The side-groups attached to the D ring were considered as flexible units, with their conformations defined by six variable torsion angles. Thus, the GA calculation involved 12 degrees of freedom $\{x, y, z, \theta, \phi, \psi, \tau_1, \tau_2, \ldots, \tau_6\}$. Bond lengths and angles were taken from known structures of other steroid molecules and from other information on standard molecular geometries [60]. The GA calculation involved the evolution of 60 generations of a population of 100 structures. In each generation, 200 offspring (involving 100 pairs of parents) and 10 mutations were considered. For mating and mutation, each of the 12 variables was considered as an independent gene. In mating two parents to generate two offspring, the 12 variables from each parent were combined and distributed between the two offspring, with no restriction on the combination of variables allowed to pass from a given parent to a given offspring. In the mutation procedure, six variables of the selected structure were chosen at random, and a new random value was assigned to each of these variables. The structure solution with lowest R_{wp} in the final generation was taken as the starting structural model for Rietveld refinement, giving $R_{wp} = 4.8\ \%$ and $R_p = 3.3\ \%$ for the final refined structure.

In the crystal structure of form 2 of FP (Fig. 13), the molecules form stacks along the *c*-axis with adjacent molecules related by translation. Zig-zag chains of molecules related by the 2_1 screw operation along the *b*-axis are linked by C–O–H...O=C hydrogen bonds involving the hydroxyl group (C ring) and carbonyl group (A ring) of adjacent molecules (O...O, 2.8 Å; C–O...O, 110°). This structure provides interesting similarities and contrasts with the structure of form 1 of FP. Both structures contain similar hydrogen-bonded chains (described above along the *b*-axis in form 2), but differ in the structural relationship between adjacent chains of this type. In form 2, adjacent chains are anti-parallel (related by a 2_1 axis), whereas in form 1, adjacent chains are parallel to each other (related by translation).

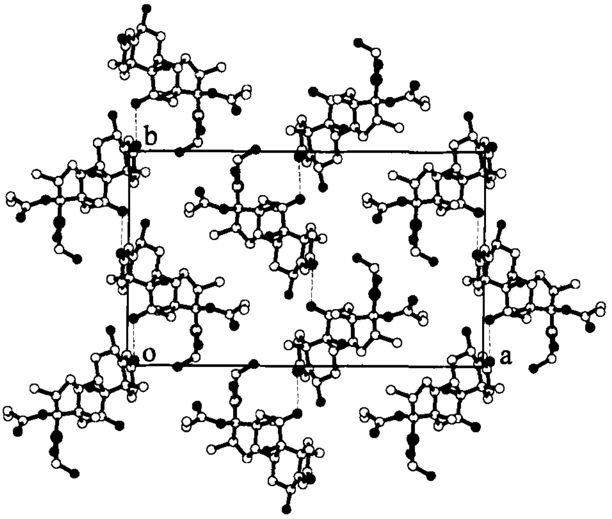


Figure 13. Final refined crystal structure of form 2 of fluticasone propionate (hydrogen atoms not shown) viewed along the *c*-axis. Dashed lines indicate hydrogen bonding interactions.

9.2.4 Discussion

Although the successful application of GA techniques for powder structure solution has been demonstrated, there is considerable scope for further development and optimization of the strategies for implementing GA techniques in this field. In this regard, we are currently exploring several fundamental aspects of the GA technique (with the aim of optimizing the procedures for searching $R(\Gamma)$ hypersurfaces) and developing new ways of defining the hypersurface such that global optimization may be achieved more efficiently.

The main factor limiting the scope of direct-space approaches for structure solution is the dimensionality of the hypersurface to be explored. The complexity of the search procedure (and the time required to achieve successful structure solution) depends more directly on the number of degrees of freedom in the optimization (the number of variables in Γ) rather than the number of atoms in the asymmetric unit. Thus, direct-space structure solution is generally more straightforward for a molecule that is essentially rigid than a molecule that is completely flexible (i.e., for which the conformation is not known in advance), irrespective of the number of atoms in the molecule. In contrast, the complexity of structure solution using traditional approaches depends more directly on the number of atoms in the asymmetric unit.

A feature of considering $R_{wp}(\Gamma)$ (or alternative definitions of R-factor) in direct-space structure solution is that the figure-of-merit is based purely on experimental data, and is not biased by the introduction of any arbitrary parameters or assumptions. However, it may be advantageous under certain circumstances to combine the powder diffraction data with other "direct-space" information, such as the computed potential energy $E(\Gamma)$. Thus, an alternative strategy for direct-space structure solution is to consider a new hypersurface $G(\Gamma)$, defined as an appropriate function of $E(\Gamma)$ and $R_{wp}(\Gamma)$: i.e., $G(\Gamma) = \mathcal{F}(E(\Gamma), R_{wp}(\Gamma))$. Provided a reliable definition of energy is available for use in this type of application, this hybrid approach may have significant advantages over the consideration of $R_{wp}(\Gamma)$ alone. The key to this approach lies in the definition of the function $\mathcal{F}$, such that the contributions of $E(\Gamma)$ and $R_{wp}(\Gamma)$ are appropriately weighted in a manner that reflects the differing characteristics of the $E(\Gamma)$ and $R_{wp}(\Gamma)$ hypersurfaces. In this regard, we are currently exploring the optimization of the function $\mathcal{F}$ in direct-space structure solution using our GA method [61].

9.3 Crystal Structure Prediction using GAs

Section 9.2 concentrated on the determination of crystal structures, based on the comparison between calculated and experimental powder diffraction patterns. Another area of considerable interest concerns the prediction of crystal structure based on minimization of the energy of a solid as a function of the unit cell parameters and/or the positions of the atoms, ions or molecules in the unit cell. Clearly this type of approach is susceptible to the quality of the parameterized energy function used to compute the energy of the solid, and any assumptions and approximations inherent within the parameterization, and for this reason it can be advantageous to validate the results of such structure predictions against experimental data (for example, diffraction data). As discussed above for the R_{wp} hypersurface, the energy hypersurface for a polyatomic solid is multidimensional and contains many local minima in addition to the global minimum (which corresponds to the lowest energy structure). Clearly an appropriate algorithm is required to carry out rapid, reliable searching of the energy hypersurface in order to find the global minimum.

In 1995, Catlow and coworkers [62] developed a GA for use in crystal structure prediction of inorganic crystalline solids. The method was used to predict the previously unknown crystal structure of the ternary oxide Li_3RuO_4. In a recent publication [63], the same group reported the development of a modified version of their earlier GA approach for crystal structure prediction. Here we describe in detail the most recent implementation of their GA method for structure prediction. The method comprises the following stages:

Stage 1: A GA is used to produce plausible candidate structures on the grounds of ionic co-ordination and approximate energy considerations (discussed below).

Stage 2: The co-ordinates of the plausible structures generated in Stage 1 are locally optimized by minimization of the calculated lattice energy (which provides a better measure of the quality of a candidate structure than the criteria used in Stage 1). In this implementation, the lattice energy is calculated using empirical potentials based on the Born model of a crystal, but including three-body terms in those cases (such as α-quartz) for which covalency may be important. The calculation of lattice energy and conjugate-gradient minimization are performed within the GULP (General Utility Lattice Program) package of Gale [64].

Stage 3: The powder diffraction patterns calculated for the best candidate structures generated in Stage 2 are compared to the experimental powder diffraction pattern. The co-ordinates are then refined (for example, by Rietveld refinement) to minimize the differences between the calculated and experimental powder diffraction patterns.

The method requires knowledge of the unit cell (determined, for example, by indexing the powder diffraction pattern of the material of interest) and the unit cell contents, which may be deduced from density considerations. The space group P1 is assumed. The unit cell is divided up into a three-dimensional Cartesian grid with 2^n divisions along each unit cell edge (for a cubic unit cell), yielding a total of 2^{3n} smaller cells. Thus, this approach involves discretization of parameter space. The ions that constitute the unit cell contents are then placed randomly into these subcells and the position $(x, y, z)_j$ of each ion is represented by a concatenated string of three binary numbers (each of length n and ranging in value from 1 to 2^n). Thus, each candidate structure corresponds to a string of $3nN_a$ binary digits, where N_a is the number of ions in the unit cell.

The initial population comprises 2M candidate structures ("adults") and is generated at random. The quality of each candidate structure is then assessed using the cost function (fitness function) described below. Pairs of parent structures are then chosen (either by "tournament selection" or based on their fitness) to participate in the "crossing" (mating) operation, which involves the exchange of random pieces of the strings representing each of the parent structures. The mutation operation is performed by randomly changing some digits in the binary code from 0 to 1 or from 1 to 0 for certain candidate structures. The GA approach is elitist as the best N members of any given generation are allowed to pass into the succeeding generation. In addition, randomly generated "foreign" structures are admitted at each cycle to ensure population diversity.

The cost function (generalized from the implementation in [62]) is a weighted sum of the following contributions: (i) the discrepancy between the ionic charge and the sum of bond valences between each ion and its nearest neighbors; (ii) the Coulombic repulsion between ions with charges of the same sign; (iii) the discrepancy between the expected and calculated first-shell co-ordination numbers of all the ions; (iv) the Coulombic attraction between ions with charges of opposite sign; (v) a bond valence term between ions with charges of the same sign; and (vi) the discrepancy between the expected and calculated second-shell co-ordination numbers of all the ions. Note that in the reported work [63], only terms (i) – (iii) were actually included in the cost function. It is important to emphasize that this cost function provides a robust measure of deviations from standard geometric features, and can be calculated rapidly (more rapidly than calculating the lattice energy and/or the R-factor for every trial structure in the population).

With this method, the correct (previously known) structures of 38 binary oxides (of the types M_2O, MO, M_2O_3 and MO_2), including α-quartz and some polymorphs of TiO_2, and a number of ternary oxide structures (such as $MgAlO_4$, $Tl_2Mn_2O_7$ and several perovskite structures) have been generated successfully [63]. In a number of cases, it was found that incorrect polymorphs were predicted and some disordered structures were generated, although such structures can be readily eliminated on the basis that they give a poor fit to the experimental powder diffraction pattern. The fact that the polymorphs of SiO_2 were difficult to generate may suggest that an alternative fitness function could be advantageous in the case of structures based on distinct polyhedral networks.

9.4 Indexing Powder Diffraction Data using GAs

9.4.1 Background

As discussed in section 9.2.1, the first stage of crystal structure determination from powder diffraction data involves determination of the unit cell by "indexing" the powder diffraction pattern. Clearly it is not possible to proceed with structure solution unless the correct unit cell has been found at this initial stage. In contrast to the recent advances in techniques for structure solution, there has been relatively little fundamental development of indexing methods since the pioneering work over 20 years ago (see [14] for a recent review), and reliable indexing of powder diffraction data using existing techniques is often the limiting step in the structure determination process. Recognizing these issues, a new indexing technique, based on whole-profile fitting and global optimization using a GA, has been developed recently [65].

The positions of the peaks in a powder diffraction pattern depend only on the unit cell dimensions (lattice parameters) {a, b, c, α, β, γ}, and the aim of indexing is to determine the correct lattice parameters from knowledge of the peak positions in the experimental powder diffraction pattern. The most widely used indexing programs (ITO [59], TREOR [66] and DICVOL [67]) all consider the measured positions of peak maxima for a number of selected peaks (general reviews are given in [14, 68, 69]). However, as discussed above, experimental powder diffraction patterns typically have considerable peak overlap and sometimes peak displacements, which can lead to several problems in indexing (elaborated in [65]). For reliable indexing by existing methods, at least 20 measured peak positions are usually required, especially for low-symmetry structures. For such techniques, the selected peaks tend to be confined to the low-angle region as peak overlap is more severe at high diffraction angles. Peak overlap is particularly severe for materials with low symmetry and large unit cells, such that extraction of peak positions may be difficult even at low diffraction angle. In many cases, certain peaks which may be crucial for correct indexing may be obscured or completely unresolved due to peak overlap. Another issue concerns the presence of impurity phases in powder samples, which often constitutes an intractable problem for indexing by existing methods (unless the presence of the impurity phases is known *a priori*) – thus, if an impurity peak is inadvertently selected for the indexing procedure, it is generally difficult or impossible to find the correct unit cell.

In view of these issues, there is a clear requirement for new indexing strategies that are not based on using a few selected peak positions but instead use all regions of the powder diffraction pattern, including the overlapped peaks. In this section, we give an overview of a new indexing strategy in which the correctness of trial sets of lattice parameters {a, b, c, α, β, γ} is assessed using a whole-profile fitting procedure and the hypersurface defined by these parameters is explored using a GA. This approach is fundamentally different from any previous indexing method, and is intrinsically more robust for dealing with the problems discussed above.

We note that a method for indexing powder diffraction data by using a GA within the framework of the conventional indexing strategy (i.e., requiring a set of extracted peak positions) has also been proposed [70].

9.4.2 Methodology

First, we describe our method [65] for constructing a powder diffraction profile for a set of trial lattice parameters $\{a, b, c, \alpha, \beta, \gamma\}$, allowing R_{wp} to be determined. The lattice parameters determine the peak positions, and parameters describing the peak shape and peak width are used in the Le Bail profile-fitting procedure [56, 71] to partition the observed intensities, producing a calculated powder diffraction pattern from which R_{wp} is calculated. It should be noted that the Le Bail technique is not based on minimization of R_{wp}, although R_{wp} will be lowest when all the parameters of the model are correct.

The aim of our indexing strategy is to search parameter space $\{a, b, c, \alpha, \beta, \gamma\}$ in order to find the parameter set that represents the best agreement (lowest R_{wp}) with the experimental powder diffraction pattern. In effect, we require to search the hypersurface $R_{wp}(a, b, c, \alpha, \beta, \gamma)$ to find the global minimum. We note that some of these parameters may be fixed by the metric symmetry, and it is only in the case of triclinic systems that all six parameters are treated as variables. We have chosen to use a GA in this field based on the recognition that certain features of $R_{wp}(a, b, c, \alpha, \beta, \gamma)$ hypersurfaces are particularly suited for exploration using GAs. Thus, certain parts of the experimental data depend specifically on only one parameter or a well-defined group of parameters, and therefore schemata should be expressed strongly. For example, the positions of the $\{h00\}$ peaks depend only on the lattice parameter $\{a\}$, the positions of the $\{hk0\}$ peaks depend only on the subset of lattice parameters $\{a, b, \gamma\}$, and so on.

In the GA approach for indexing powder diffraction data, each member of the population is a set of trial lattice parameters $\{a, b, c, \alpha, \beta, \gamma\}$ (or a subset of these parameters, depending on the crystal system), and the population evolves through mating, mutation and natural selection. The fitness of each set of trial lattice parameters depends on its value of R_{wp}, and the *linear* fitness function $[F(R_{wp}) = 1 - (R_{wp} - R_{min})/(R_{max} - R_{min})]$ has been used in our work so far. As before, R_{min} and R_{max} denote the lowest and highest values of R_{wp} in the current population, and are updated for each new generation of the population. Our strategy for carrying out the GA calculation is similar to that implemented in our GA technique for structure solution (see section 9.2.2).

The presence of peaks at low diffraction angles can be crucial for correct indexing, but these peaks are often weak and the use of a conventional R_{wp} may fail to recognize the significance of these peaks. To overcome this limitation, we use a modified definition of R_{wp} in which the powder diffraction pattern is divided into different regions and R_{wp} is calculated separately for each region. These individual values of R_{wp} (denoted $R_{wp}^{(i)}$ for each region i) are then summed to obtain the overall R_{wp} (subsequently denoted R_{wp}^{total}). With this approach, the residuals for a given region are scaled according to the total intensity of the region (the lower the total intensity, the higher the scaling factor). Thus, a region containing only weak peaks can make as much contribution to R_{wp}^{total} as a region containing peaks of much greater absolute intensity. There is also scope for overcoming the problems discussed above by using new definitions of R_{wp} in which weighting schemes (w_i) based on d-spacing, rather than intensity, are used.

An important advantage of whole-profile fitting is that the presence of minor impurities does not limit the success of the procedure. Peaks due to impurity phases simply contribute a constant amount to the R-factor, and the global minimum in R_{wp}^{total} will arise when the majority phase is indexed correctly.

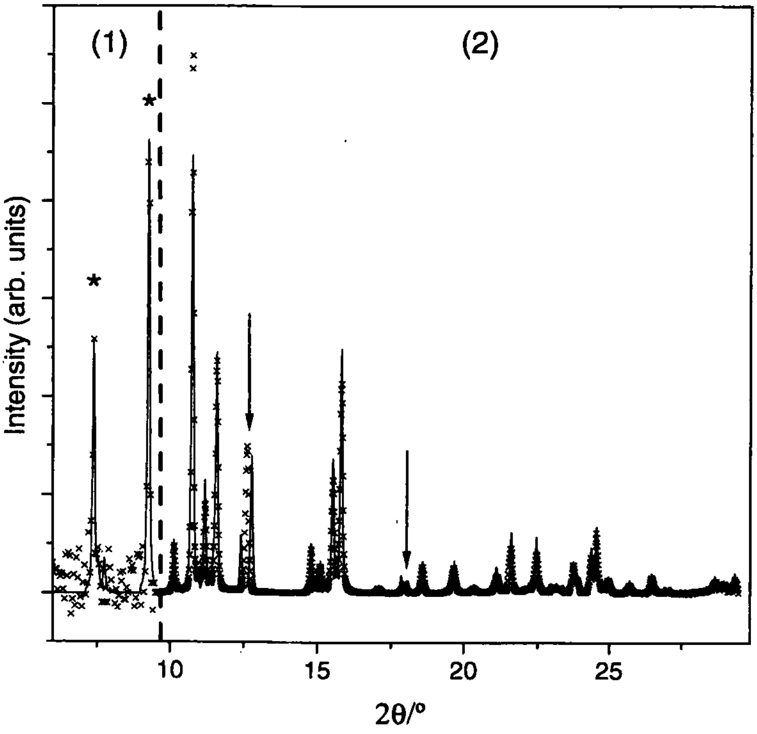


Figure 14. The fit of the calculated powder diffraction data (solid line) to the experimental powder diffraction data (crosses) for the best (correct) unit cell obtained in the GA indexing calculation for phase-II of gallium arsenide. The arrows highlight the impurity peaks and the asterisks mark the positions of the very weak reflections that are crucial for correct indexing. Regions (1) and (2) are discussed in the text. In this plot, the total intensity of region (1) is amplified relative to the total intensity of region (2).

9.4.3 Example of Application

The GA indexing method described above is illustrated here for unit cell determination of the orthorhombic (space group Cmcm) phase-II of gallium arsenide (GaAs). The unit cell determined previously [72] from powder X-ray diffraction data is: a = 4.9703(1) Å, b = 4.7801(3) Å, c = 5.2717(4) Å. Importantly, the experimental data (see Fig. 14) contain an impurity phase, representing a particularly challenging problem for indexing the powder diffraction pattern. Furthermore, indexing the powder diffraction pattern successfully relies on correctly fitting the positions of the two very weak peaks (marked by asterisks in Fig. 14) at low diffraction angle, but these two peaks would have negligible effect on a conventional R_{wp} defined across the whole powder diffraction pattern. However, by dividing the data into two regions [(1) $6° \leq 2\theta < 9.5°$ and (2) $9.5° \leq 2\theta < 30°$] with our modified definition of R_{wp}^{total}, both regions influence the fit significantly.

The GA calculation [65] involved 500 generations of a population comprising 100 sets of trial lattice parameters {a, b, c} for the orthorhombic system. In each generation, 25 mating operations (leading to 50 offspring) and 20 mutation operations were carried out. Each individual lattice parameter was constrained to be less than 8 Å during the calculation, and the unit cell volume was constrained to be less than 200 Å^3. Fig. 15 shows the evolution of the lowest value of R_{wp}^{total} for the population and the average value of R_{wp}^{total} for the population as a function of generation number. The lattice parameters corresponding to lowest R_{wp}^{total} at the end of the GA calculation were a = 4.968 Å, b = 4.773 Å, c = 5.285 Å, in close agreement with those determined previously [72] (see above). Fig. 14 shows the fit of the calculated powder diffraction pattern to the experimental powder diffraction pattern for the best unit cell obtained from our GA indexing calculation. In our experience, even if only the strongest impurity peak is included, all of the commonly available indexing programs fail, whereas our whole-profile fitting GA method successfully indexed the majority orthorhombic phase in spite of the presence of the impurity phase.

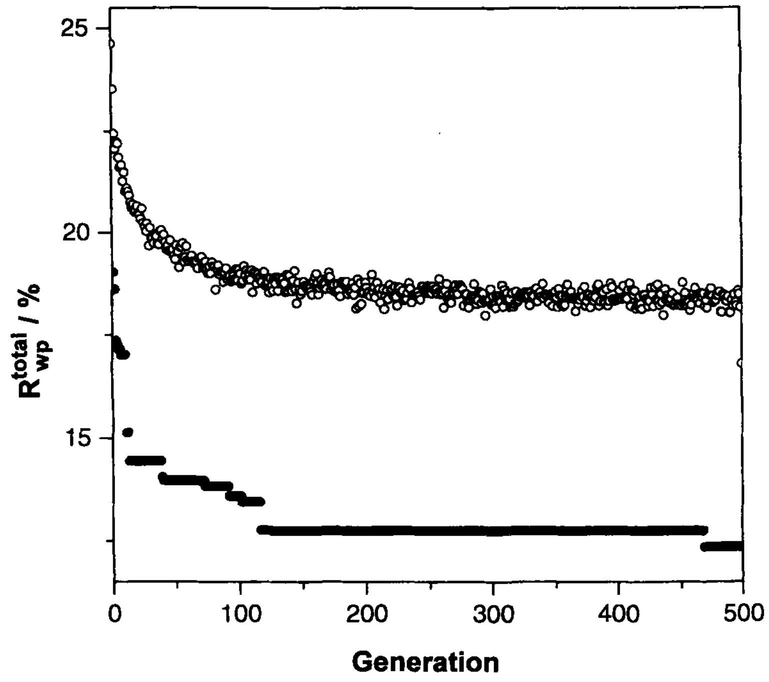


Figure 15. Evolutionary Progress Plot showing the evolution of R_{wp}^{total} for the best unit cell in the population (filled circles) and the average R_{wp}^{total} for the population (open circles) as a function of generation number in the GA indexing calculation for phase-II of gallium arsenide.

9.5 Other Applications of EAs in Crystallographic Areas

9.5.1 Biological Crystallography

Although several of the other chapters in this book are concerned with the application of evolutionary algorithms in the study of molecules of biological or biochemical importance, it is relevant here to mention some reported applications of such techniques in crystal structure analysis of proteins and other biological macromolecular systems. We note the significant challenges [73] faced in the measurement and analysis of the diffraction data from single crystals of such materials, arising in part from the fact that the diffraction data have significant intensity only at low diffraction angles (low resolution).

In this field, a GA method has been developed [74] for *ab initio* phasing of low-resolution X-ray diffraction data from highly symmetric structures. The direct-space parameterization used incorporates information on structural symmetry, and has been particularly applied to the case of icosahedral viruses. A GA has also been introduced [75] to speed up molecular replacement searches by allowing simultaneous searching of the rotational and translational parameters of a test model, while maximizing the correlation coefficient between the observed and calculated diffraction data. An evolutionary programming approach for six-dimensional molecular replacement searches has also been described [76]. GA methods have also been used [77] to search for heavy atom sites in difference Patterson functions.

A related area concerning protein structure determination has involved the development [78] of a GA method for determining the low-resolution structures of proteins in solution from small angle X-ray scattering. This method has been applied to determine the low-resolution structure of the active site cavity of lysozyme, the bilobed structure of gamma-crystallin, and the horseshoe shape of pancreatic ribonuclease inhibitor. The method shows promise for applications to other proteins and large macromolecular assemblies.

9.5.2 Miscellaneous Applications

In the area of surface science, Landree et al. [79] have introduced a GA method for direct phasing of surface diffraction data as part of a strategy to determine surface structure using direct methods. The method has been applied to both centrosymmetric and noncentrosymmetric two-dimensional structures.

Tam and Compton [80] have reported a GA approach for determining the Miller indices of crystal faces, using interplanar angles measured experimentally. A two-step "divide-and-conquer" strategy was adopted, and was found to speed up the convergence to the global minimum. The method was tested on crystals of triphenylmethyl chloride, leading to excellent agreement with the results obtained from other indexing methods.

Knorr and Mädler have recently introduced an evolution strategy for refining structural fragments in orientationally disordered crystals [81]. The fragments are embedded into the electron densities which are derived from the experimental data using maximum entropy techniques [82]. The method has also been used for refinement against structure factors derived from these electron densities. The evolution strategy consists only of fitness-based selection and mutation stages (i.e., there is no mating/crossover). Applications include determination of reorientation pathways for the $[PF_6]^-$ anion in KPF_6 and determination of the conformation of the flexible organic molecule 1,3-dioxolane in a sodalite host [81].

9.6 Concluding Remarks

As highlighted above, it is clear that evolutionary algorithms have already made an impact in optimization problems within a number of crystallographic areas, and we may forecast with confidence that evolutionary algorithms will find increasing applications in other areas of crystallographic sciences in the years to come.

Although several other approaches may also be used for global optimization (including, for example, Monte Carlo and simulated annealing methods), it is not advantageous at this stage to enter a detailed general comparison between these approaches and GAs, as the relative merits depend critically on the particular problem at hand and details of the particular implementation of each approach. Nevertheless, it is worthwhile to highlight the fact that GAs operate in an intrinsically parallel manner, with many different regions of parameter space (corresponding to different members of the population) investigated simultaneously. Furthermore, information on these different regions of parameter space is passed actively between individual members of the population by the mating procedure, promoting efficient convergence towards the global minimum. The implicit parallel nature of GAs makes them efficient and robust vehicles for optimization, and particularly advantageous for optimization problems defined by a large number of variables. However, in assessing the suitability of applying GAs in any particular optimization problem, it is important to recognize the role of schemata, and to ensure (for example, by appropriate definition of the parameter space and/or the fitness function and/or the mechanism for mating) that schemata are strongly expressed and properly utilized in the GA.

Importantly, several different aspects of GAs (such as the procedures for mating, mutation and natural selection, and the definition of the fitness function and parameter space) are open to re-definition and adaptation such that they can be optimally suited to a particular type of optimization problem. In this regard, there is much scope to derive highly optimized procedures for the implementation of GAs in crystallographic applications, as well as in other fields of science.

Acknowledgments

For collaborations and many useful discussions in connection with our research on structure solution using genetic algorithms, we are grateful to Dr. E. Tedesco, Mr. G. W. Turner, Mr. S. Habershon, Mr. O. J. Lanning, Dr. R. W. Lancaster, Dr. D. Philp, Miss P. Calcagno, Dr. S. Ahn and Miss K. Psallidas, and similarly in connection with our research on indexing powder diffraction data we are grateful to Professor R. J. Nelmes, Dr. S. A. Belmonte and Dr. M. I. McMahon. We also wish to thank the following organizations for funding our research in these areas: EPSRC, Nuffield Foundation, MSI, Glaxo Wellcome, and Ciba Specialty Chemicals.

References

[1] J. D. Dunitz, *X-ray Analysis and the Structures of Organic Molecules*, Verlag Helvetica Chimica Acta, Basel, **1995**.

[2] J. P. Glusker, K. N. Trueblood, *Crystal Structure Analysis – A Primer*, Oxford University Press, Oxford, **1985**.

[3] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, Michigan, **1975**.

[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Massachusetts, **1989**.

[5] H. M. Cartwright, *Applications of Artificial Intelligence in Chemistry*, Oxford University Press, Oxford, **1993**.

[6] H. M. Rietveld, A Profile Refinement Method for Nuclear and Magnetic Structures, *J. Appl. Crystallogr.* **1969**, *2*, 65–71.

[7] R. A. Young (Ed.), *The Rietveld Method*, International Union of Crystallography and Oxford University Press, Oxford, **1993**.

[8] A. N. Christensen, M. S. Lehmann, M. Nielsen, Solving Crystal Structures from Powder Diffraction Data, *Aust. J. Phys.* **1985**, *38*, 497–505.

[9] A. K. Cheetham, A. P. Wilkinson, Structure Determination and Refinement with Synchrotron X-ray Powder Diffraction Data, *J. Phys. Chem. Solids* **1991**, *52*, 1199–1208.

[10] L. B. McCusker, Zeolite Crystallography – Structure Determination in the Absence of Conventional Single-crystal Data, *Acta Crystallogr.* **1991**, *A47*, 297–313.

[11] A. K. Cheetham, A. P. Wilkinson, Synchrotron X-ray and Neutron Diffraction Studies in Solid State Chemistry, *Angew. Chemie, Int. Ed. Engl.* **1992**, *31*, 1557–1570.

[12] P. R. Rudolf, Techniques for Ab Initio Structure Determination from X-ray Powder Diffraction Data, *Mater. Chem. Phys.* **1993**, *35*, 267–272.

[13] K. D. M. Harris, M. Tremayne, Crystal Structure Determination from Powder Diffraction Data, *Chem. Mater.* **1996**, *8*, 2554–2570.

[14] J. I. Langford, D. Louër, Powder Diffraction, *Rep. Progr. Phys.* **1996**, *59*, 131–234.

[15] D. M. Poojary, A. Clearfield, Application of X-ray Powder Diffraction Techniques to the Solution of Unknown Crystal Structures, *Acc. Chem. Res.* **1997**, *30*, 414–422.

[16] W. I. F. David, The Probabilistic Determination of Intensities of Completely Overlapping Reflections in Powder Diffraction Patterns, *J. Appl. Crystallogr.* **1987**, *20*, 316–319.

[17] W. I. F. David, Extending the Power of Powder Diffraction for Structure Determination, *Nature* **1990**, *346*, 731–734.

[18] J. Jansen, R. Peschar, H. Schenk, On the Determination of Accurate Intensities from Powder Diffraction Data – Estimation of Intensities of Overlapping Reflections, *J. Appl. Crystallogr.* **1992**, *25*, 237–243.

[19] M. A. Estermann, L. B. McCusker, C. Baerlocher, Ab Initio Structure Determination from Severely Overlapping Powder Diffraction Data, *J. Appl. Crystallogr.* **1992**, *25*, 539–543.

[20] M. A. Estermann, V. Gramlich, Improved Treatment of Severely or Exactly Overlapping Bragg Reflections for the Application of Direct-methods to Powder Data, *J. Appl. Crystallogr.* **1993**, *26*, 396–404.

[21] C. J. Gilmore, K. Shankland, G. Bricogne, Applications of the Maximum Entropy Method to Powder Diffraction and Electron Crystallography, *Proc. Royal Soc. (London)* **1993**, *A442*, 97–111.

[22] D. S. Sivia, W. I. F. David, A Bayesian Approach to Extracting Structure Factor Amplitudes from Powder Diffraction Data, *Acta Crystallogr.* **1994**, *A50*, 703–714.

[23] C. J. Gilmore, Maximum Entropy and Bayesian Statistics in Crystallography: A Review of Practical Applications, *Acta Crystallogr.* **1996**, *A52*, 561–589.

[24] W. I. F. David, On the Number of Independent Reflections in a Powder Diffraction Pattern, *J. Appl. Crystallogr.* **1999**, *32*, 654–663.

[25] K. D. M. Harris, M. Tremayne, P. Lightfoot, P. G. Bruce, Crystal Structure Determination from Powder Diffraction Data by Monte Carlo Methods, *J. Amer. Chem. Soc.* **1994**, *116*, 3543–3547.

[26] B. M. Kariuki, D. M. S. Zin, M. Tremayne, K. D. M. Harris, Crystal Structure Solution from Powder X-ray Diffraction Data: The Development of Monte Carlo Methods to Solve the Crystal Structure of the γ Phase of 3-Chloro-*trans*-cinnamic acid, *Chem. Mater.* **1996**, *8*, 565–569.

[27] M. Tremayne, B. M. Kariuki, K. D. M. Harris, The Development of Monte Carlo Methods for Crystal Structure Solution from Powder Diffraction Data: Simultaneous Translation and Rotation of a Structural Fragment within the Unit Cell, *J. Appl. Crystallogr.* **1996**, *29*, 211–214.

[28] M. Tremayne, B. M. Kariuki, K. D. M. Harris, Solution of an Organic Crystal Structure from X-ray Powder Diffraction Data by a Generalized Monte Carlo Method: Crystal Structure Determination of 1-Methylfluorene, *J. Mat. Chem.* **1996**, *6*, 1601–1604.

[29] M. Tremayne, B. M. Kariuki, K. D. M. Harris, Structure Determination of a Complex Organic Solid from X-ray Powder Diffraction Data by a Generalized Monte Carlo Method: The Crystal Structure of Red Fluorescein, *Angew. Chemie, Int. Ed. Engl.* **1997**, *36*, 770–772.

[30] L. Elizabé, B. M. Kariuki, K. D. M. Harris, M. Tremayne, M. Epple, J. M. Thomas, Topochemical Rationalization of the Solid State Polymerization Reaction of Sodium Chloroacetate: Crystal Structure Determination from Powder Diffraction Data by the Monte Carlo Method, *J. Phys. Chem.* **1997**, *B101*, 8827–8831.

[31] M. Tremayne, B. M. Kariuki, K. D. M. Harris, K. Shankland, K. S. Knight, Crystal Structure Solution from Neutron Powder Diffraction Data by a New Monte Carlo Approach Incorporating Restrained Relaxation of the Molecular Geometry, *J. Appl. Crystallogr.* **1997**, *30*, 968–974.

[32] K. D. M. Harris, B. M. Kariuki, M. Tremayne, Crystal Structure Solution from Powder Diffraction Data by the Monte Carlo Method, *Mat. Sci. Forum* **1998**, *278-291*, 32–37.

[33] J. M. Newsam, M. W. Deem, C. M. Freeman, *Accuracy in Powder Diffraction II: NIST Special Publ. No. 846*, pp. 80–91, **1992**.

[34] D. Ramprasad, G. B. Pez, B. H. Toby, T. J. Markley, R. M. Pearlstein, Solid-state Lithium Cyanocobaltates with a High-capacity for Reversible Dioxygen Binding – Synthesis, Reactivity and Structures, *J. Amer. Chem. Soc.* **1995**, *117*, 10694–10701.

[35] Y. G. Andreev, P. Lightfoot, P. G. Bruce, Structure of the Polymer Electrolyte Poly(ethylene oxide)$_2$:LiN(SO_2CF_3)$_2$ Determined by Powder Diffraction using a Powerful Monte Carlo Approach, *Chem. Commun.* **1996**, 2169–2170.

[36] Y. G. Andreev, G. S. MacGlashan, P. G. Bruce, Ab Initio Solution of a Complex Crystal Structure from Powder Diffraction Data using Simulated Annealing Method and a High Degree of Molecular Flexibility, *Phys. Rev. B* **1997**, *55*, 12011-12017.

[37] C. M. Freeman, A. M. Gorman, J. M. Newsam, Simulated Annealing and Structure Solution, in C. R. A. Catlow (Ed.), *Computer Modelling in Inorganic Crystallography*, Academic Press, San Diego, pp. 117–150, **1997**.

[38] W. I. F. David, K. Shankland, N. Shankland, Routine Determination of Molecular Crystal Structures from Powder Diffraction Data, *Chem. Commun.* **1998**, 931–932.

[39] G. E. Engel, S. Wilke, O. König, K. D. M. Harris, F. J. J. Leusen, Powder Solve – A Complete Package for Crystal Structure Solution from Powder Diffraction Patterns, *J. Appl. Crystallogr.* **1999**, *32*, 1169–1179.

[40] A. M. T. Bell, J. N. B. Smith, J. P. Attfield, J. M. Rawson, K. Shankland, W. I. F. David, A Synchrotron X-ray Powder Diffraction Study of 4-(2,3,4-Trifluorophenyl)-1,2,3,5-dithiadiazolyl – Crystal Structure Determination using a Global Optimisation Method, *New J. Chem.* **1999**, *23*, 565–567.

[41] G. Reck, R.-G. Kretschmer, L. Kutschabsky, W. Pritzkow, POSIT: A Method for Structure Determination of Small Partially Known Molecules from Powder Diffraction Data – Structure

of 6-Methyl-1,2,3,4-tetrahydropyrimidine-2,4-dione (6-methyluracil), *Acta Crystallogr.* **1988**, *A44*, 417–421.

[42] J. Cirujeda, L. E. Ochando, J. M. Amigó, C. Rovira, J. Rius, J. Veciana, Structure Determination from Powder X-ray Diffraction Data of a Hydrogen-bonded Molecular Solid with Competing Ferromagnetic and Antiferromagnetic Interactions – The 2-(3,4-Dihydroxyphenyl)-α-nitronyl Nitroxide Radical, *Angew. Chemie, Int. Ed. Engl.* **1995**, *34*, 55–57.

[43] R E. Dinnebier, P. W. Stephens, J. K. Carter, A. N. Lommen, P. A. Heiney, A. R. McGhie, L. Brard, A. B. Smith III, X-ray Powder Diffraction Structure of Triclinic $C_{60}Br_{24}(Br_2)_2$, *J. Appl. Crystallogr.* **1995**, *28*, 327–334.

[44] R. B. Hammond, K. J. Roberts, R. Docherty, M. Edmondson, Computationally Assisted Structure Determination for Molecular Materials from X-ray Powder Diffraction Data, *J. Phys. Chem.* **1997**, *B101*, 6532–6356.

[45] B. M. Kariuki, H. Serrano-González, R. L. Johnston, K. D. M. Harris, The Application of a Genetic Algorithm for Solving Crystal Structures from Powder Diffraction Data, *Chem. Phys. Lett.* **1997**, *280*, 189–195.

[46] K. D. M. Harris, R. L. Johnston, B. M. Kariuki, M. Tremayne, A Genetic Algorithm for Crystal Structure Solution from Powder Diffraction Data, *J. Chem. Res. (S)* **1998**, 390–391.

[47] K. D. M. Harris, R. L. Johnston, B. M. Kariuki, The Genetic Algorithm: Foundations and Applications in Structure Solution from Powder Diffraction Data, *Acta Crystallogr.* **1998**, *A54*, 632–645.

[48] K. Shankland, W. I. F. David, T. Csoka, Crystal Structure Determination from Powder Diffraction Data by the Application of a Genetic Algorithm, *Z. Kristallogr.* **1997**, *212*, 550–552.

[49] K. Shankland, W. I. F. David, T. Csoka, L. McBride, Structure Solution of Ibuprofen from Powder Diffraction Data by the Application of a Genetic Algorithm Combined with Prior Conformational Analysis, *Int. J. Pharm.* **1998**, *165*, 117–126.

[50] T. Csoka, W. I. F. David, K. Shankland, Crystal Structure Determination from Powder Diffraction Data by the Application of a Genetic Algorithm, *Mat. Sci. Forum* **1998**, *278-281*, 294–299.

[51] B. M. Kariuki, P. Calcagno, K. D. M. Harris, D. Philp, R. L. Johnston, Evolving Opportunities in Structure Solution from Powder Diffraction Data – Crystal Structure Determination of a Molecular System with 12 Variable Torsion Angles, *Angew. Chemie, Int. Ed. Engl.* **1999**, *38*, 831–835.

[52] B. M. Kariuki, K. Psallidas, K. D. M. Harris, R. L. Johnston, R. W. Lancaster, S. E. Staniforth, S. M. Cooper, Structure Determination of a Steroid Directly from Powder Diffraction Data, *Chem. Commun.* **1999**, 1677–1678.

[53] K. D. M. Harris, R. L. Johnston, B. M. Kariuki, An Evolving Technique for Powder Structure Solution – Fundamentals and Applications of the Genetic Algorithm, *Anales de Química, Int. Ed.* **1998**, *94*, 410–416.

[54] R. L. Johnston, B. M. Kariuki, K. D. M. Harris, GAPSS: *Genetic Algorithm for Powder Structure Solution*, University of Birmingham, **1997**.

[55] G. S. Pawley, Unit-cell Refinement from Powder Diffraction Scans, *J. Appl. Crystallogr.* **1981**, *14*, 357–361.

[56] H. Toraya, Position Constrained and Unconstrained Powder Pattern Decomposition Methods, in R. A. Young (Ed.), *The Rietveld Method*, International Union of Crystallography and Oxford University Press, Oxford, pp. 254–275, **1993**.

[57] M. S. Lehman, T. F. Koetzle, W. C. Hamilton, Precision Neutron Diffraction Structure Determination of Protein and Nucleic Acid Components VIII: The Crystal and Molecular Structure of the β Form of the Amino Acid L-Glutamic Acid, *J. Cryst. Mol. Struct.* **1972**, *2*, 225–233.

[58] M. S. Lehman, A. C. Nunes, A Short Hydrogen Bond Between Near Identical Carboxyl Groups in the α Modification of L-Glutamic Acid, *Acta Crystallogr.* **1980**, *B36*, 1621–1625.

[59] J. W. Visser, A Fully Automatic Program for Finding the Unit Cell from Powder Data, *J. Appl. Crystallogr.* **1969**, *2*, 89–95.

[60] F. H. Allen, O. Kennard, D. G. Watson, L. Brammer, A. G. Orpen, R. Taylor, Tables of Bond Lengths Determined by X-ray and Neutron Diffraction – Bond Lengths in Organic Compounds, *J. Chem. Soc., Perkin Trans. 2* **1987**, S1–S19.

[61] O. J. Lanning, S. Habershon, K. D. M. Harris, R. L. Johnston, B. M. Kariuki, E. Tedesco, G. W. Turner, Definition of a "Guiding Function" in Global Optimization: a Hybrid Approach Combining Energy and R-factor in Structure Solution from Powder Diffraction Data, *Chem. Phys. Lett.*, **2000**, *317*, 296–303.

[62] T. S. Bush, C. R. A. Catlow, P. D. Battle, Evolutionary Programming Techniques for Predicting Inorganic Crystal Structures, *J. Mater. Chem.* **1995**, *5*, 1269–1272.

[63] S. M. Woodley, P. D. Battle, J. D. Gale, C. R. A. Catlow, The Prediction of Inorganic Crystal Structures using a Genetic Algorithm and Energy Minimisation, *Phys. Chem. Chem. Phys.* **1999**, *1*, 2535–2542.

[64] J. D. Gale, GULP: A Computer Program for the Symmetry Adapted Simulation of Solids, *J. Chem. Soc., Faraday Trans.* **1997**, *93*, 629–637.

[65] B. M. Kariuki, S. A. Belmonte, M. I. McMahon, R. L. Johnston, K. D. M. Harris, R. J. Nelmes, A New Approach for Indexing Powder Diffraction Data Based on Whole-profile Fitting and Global Optimization using a Genetic Algorithm, *J. Synchrotron Radiation* **1999**, *6*, 87–92.

[66] P.-E. Werner, L. Eriksson, M. Westdahl, TREOR: A Semi-exhaustive Trial-and-error Powder Indexing Program for All Symmetries, *J. Appl. Crystallogr.* **1985**, *18*, 367–370.

[67] A. Boultif, D. Louër, Indexing of Powder Diffraction Patterns for Low-symmetry Lattices by the Successive Dichotomy Method, *J. Appl. Crystallogr.* **1991**, *24*, 987–993.

[68] R. Shirley, *Data Accuracy for Powder Indexing*, Natl. Bur. Stand. (US) Spec. Publ. No. 567361, **1980**.

[69] D. Louër, *Accuracy in Powder Diffraction II: NIST Special Publ. No. 846* (E. Prince, J. K. Stalick, Eds.), pp. 92–104, **1992**.

[70] W. Paszkowicz, Application of the Smooth Genetic Algorithm for Indexing Powder Patterns – Tests for the Orthorhombic System, *Mat. Sci. Forum* **1996**, *228–231*, 19–24.

[71] A. Le Bail, H. Duroy, J. L. Fourquet, Ab Initio Structure Determination of $LiSbWO_6$ by X-ray Powder Diffraction, *Mater. Res. Bull.* **1988**, *23*, 447–452.

[72] M. I. McMahon, R. J. Nelmes, D. R. Allan, S. A. Belmonte, T. Bovornratanaraks, Observation of a Simple Cubic Phase of GaAs with a 16-Atom Basis (SC16), *Phys. Rev. Lett.* **1998**, *80*, 5564–5567.

[73] J. P. Glusker, M. Lewis, M. Rossi, *Crystal Structure Analysis for Chemists and Biologists*, Wiley-VCH, New York, **1994**.

[74] S. T. Miller, J. M. Hogle, D. J. Filman, A Genetic Algorithm for the Ab Initio Phasing of Icosahedral Viruses, *Acta Crystallogr.* **1996**, *D52*, 235–251.

[75] G. Chang, M. Lewis, Molecular Replacement using Genetic Algorithms, *Acta Crystallogr.* **1997**, *D53*, 279–289.

[76] C. R. Kissinger, D. K. Gehlhaar, D. B. Fogel, Rapid Automated Molecular Replacement by Evolutionary Search, *Acta Crystallogr.* **1999**, *D55*, 484–491.

[77] G. Chang, M. Lewis, Using Genetic Algorithms for Solving Heavy Atom Sites, *Acta Crystallogr.* **1994**, *D50*, 667–674.

[78] P. Chacon, F. Moran, J. F. Diaz, E. Pantos, J. M. Andreu, Low-resolution Structures of Proteins in Solution Retrieved from X-ray Scattering with a Genetic Algorithm, *Biophys. J.* **1998**, *74*, 2760–2775.

[79] E. Landree, C. Collazo-Davila, L. D. Marks, Multi-solution Genetic Algorithm Approach to Surface Structure Determination using Direct Methods, *Acta Crystallogr.* **1997**, *B53*, 916–922.

[80] K. Y. Tam, R. G. Compton, GAMATCH – A Genetic Algorithm Based Program for Indexing Crystal Faces, *J. Appl. Crystallogr.* **1995**, *28*, 640–645.

[81] K. Knorr, F. Mädler, The Application of Evolution Strategies to Disordered Structures, *J. Appl. Crystallogr.* **1999**, *32*, 902–910.

[82] K. Knorr, F. Mädler, R. J. Papoular, Model-free Density Reconstruction of Host/guest Compounds from High-resolution Powder Diffraction Data, *Micropor. Mesopor. Mater.* **1998**, *21*, 353–363.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

10 Structure Determination by NMR Spectroscopy

Bryan C. Sanctuary

Abbreviations

2-D	Two-dimensional
3-D	Three-dimensional
4-D	Four-dimensional
COFFEE	Consistency based objective function for alignment evaluation
COSY	Correlation NMR spectroscopy
DFT	Discrete Fourier transform
DG	Distance geometry
DGΩ	Distance geometry-optimized metric matrix embedding by genetic algorithms)
FFT	Fast Fourier transform
FID	Free induction decay
FINGAR	FIt NMR using a genetic algorithm
FT-NMR	Fourier-transformed NMR
GA	Genetic algorithm
GARANT	General algorithm for resonance assignment
GAT	Genetic algorithm for structure determination in torsion angle space
GENFIT	Generic non-linear alignment algorithm using GA
GENFOLD	A genetic algorithm for folding protein structures using NMR
HMQC	Heteronuclear multi-quantum correlations NMR spectroscopy
IQML	Iterative quadratic maximum likelihood method
IR	Infrared
LPSVD	Linear prediction singular value decomposition method
MCD	Main-Chain-Directed resonance assignment strategy
MD	Molecular dynamics
MP	Matrix pencil method
MRI	Magnetic resonance imaging
MRS	Magnetic resonance spectroscopy
NMR	Nuclear magnetic resonance spectroscopy
NOESY (NOE)	Nuclear Overhauser enhancement spectroscopy
QSAR	Quantitative structure–activity relationship
SA	Simulated annealing
SAGA	Sequence alignment by genetic algorithm
SNR	Signal-to-noise ratio

TLS	Total least squares method
TOCSY	Total correlation NMR spectroscopy

Symbols

$d_{\alpha N}(i,i+1)$	Denotes inter and intra distances between atoms in amino acids
^3J	Nuclear spin-spin coupling constant
ϕ,ψ	Two dihedral angles associated with peptide bonds

10.1 Introduction

From its inception, as a curiosity of fundamental research, in the 1940s, nuclear magnetic resonance (NMR) has evolved and found application in almost every branch of science. It moved from physics to chemistry in the 1950s, with the discovery of chemical shifts; to heteronuclear spins and low natural abundant spins, with the advent of fast Fourier transform (FFT) techniques in the late 1960s and early 1970s; to medical imaging in the late 1970s and 1980s; and into biochemistry in the late 1980s and 1990s with the discovery of multidimensional NMR [1].

It is a branch of spectroscopy that has benefited from the weakness of spin couplings, leading to simple spectra, and slowness of relaxation, leading to narrow lines in most applications. NMR also is blessed by being based, for the most part, on nuclear spins with magnitudes predominantly of 1/2 and 1. The most common nuclei in the biosciences, hydrogen, carbon, nitrogen, phosphorus and oxygen, by having a nuclear spin, lend themselves easily to NMR and provide many different avenues for exploitation. ^{12}C has no nuclear spin, and so there is no spin coupling to hydrogen ^1H, with its spin 1/2. When needed, the isotope, ^{13}C, can be studied, and this also has a spin of 1/2. Nitrogen, ^{14}N, has a spin of 1 and oxygen has no nuclear spin. Labile hydrogens can easily be replaced by deuterium ^2H, with spin 1, and isotopic substitution is a useful technique that enables the study of specific parts of biomolecules.

The advantage of having spins of 1/2 and 1 clearly lies in the fact that they are the simplest quantum systems to deal with. Although dipole-dipole coupling still presents problems for exact calculations, the majority of spin dynamics, and the response of spins to pulses, is well established [1]. This area – call it "spin gymnastics" – makes it possible to follow the spin polarization from one pulse to another. Considerable effort has been put into spin gymnastics [2] and spins co-operate well both theoretically and experimentally.

NMR spectroscopy, as a field, has moved from research to technology, much like X-ray crystallography evolved into an experimental technique in the late 1960s. A review by Wagner et al. in 1992 compares and contrasts NMR and X-ray techniques in the structure determination of proteins [3]. This is not to say that new fundamental areas in NMR will not be discovered. For example, very high magnetic fields may still hold some surprises, but for the most part NMR has settled down. This also does not mean that there are not exciting and fundamental applications yet to be developed, especially in the area of spin relaxation, and these will rest upon established statistical mechanical methods.

It is also not an exaggeration to say that from the time the sample tube is placed in the magnet, running spectra and analyzing the data is almost totally automated. This is a credit to the companies that design and develop the instrumentation. The areas that are not fully automated are those which are reviewed here. NMR spectra provide four categories of data: frequencies (chemical shifts); coupling constants (J values); intensities; and line widths (relaxation times) [1]. The first three categories contain the information that allows a structure to be determined, and the last allows molecular dynamics to be studied.

Although structure determination in most chemical applications – especially when coupled with IR spectroscopy and X-ray crystallography – offers few challenges, the situation is less satisfactory for biomolecules. These large molecules contain repeating units of nucleic acids or amino acids. As such, the spectra are crowded, with the spins resonating at similar positions, making assignments difficult. All NMR studies of biomolecules are done with two-dimensional (2-D) and three-dimensional (3-D) NMR which separate and simplify 1-D experiments. Due to slow tumbling of large proteins, present 2-D NMR studies are limited to molecular weights of about 10 kDa, whereas larger proteins (up to about 25 kDa) can be treated by routine 3-D NMR, although this limit is continually being challenged. NMR and X-ray crystallography are complementary techniques, one being applicable to small proteins or polypeptides that may not crystallize, and the other to larger proteins that must crystallize.

Within the domain of protein NMR, there are three problems. One is spectral parameter estimation [4] from the free induction decay (FID). This analysis is usually handled by fast Fourier or discrete Fourier transform techniques, but alternative algorithms have been proposed [5]. Another, alluded to above, is the sequence-specific resonance assignment of these spectra [6, 7]. Manually this is a difficult and time-consuming task, but attempts to automate this step have been encouraging. The third problem is protein structure determination using distance restraints and dihedral angles.

It is in these three areas of the NMR of biomolecules that genetic algorithms (GAs) have had the greatest impact and made contributions. GAs have also been applied to the generation of NMR pulse shapes. Thus, it is to these four areas that this review is limited.

NMR of biomolecules has been reviewed exhaustively in excellent monographs. Up to 1995, a complete and exhaustive treatment of biological NMR has been edited by Evans [8], and up to 1997 by Markley and Opella [9]. Freeman's book on spin choreography [2] (spin gymnastics) is beautifully written, nonmathematical, and highly intuitive.

The sequence-specific assignment problem is discussed in Wüthrich's book [6] which, although written in 1986, remains a valuable resource and introduction to this area. Wüthrich also reviewed the area in 1996 [10], while another review – which includes a summary of optimization techniques, including GAs – is by Zimmerman and Montelione [11]. Data analysis, and a comparison of heuristic optimization methods, GAs and simulated annealing (SA), are reviewed and compared by Weber et al. [12].

10.2 Protein Structure Determination from NMR Data

During the past 15 years, remarkable progress has been made in applying NMR spectroscopy to the study of proteins [6, 13]. For example, in NMR, the solution conditions can be varied, the internal dynamics and chemical exchange phenomena can be characterized, and the effects of temperature change can be studied.

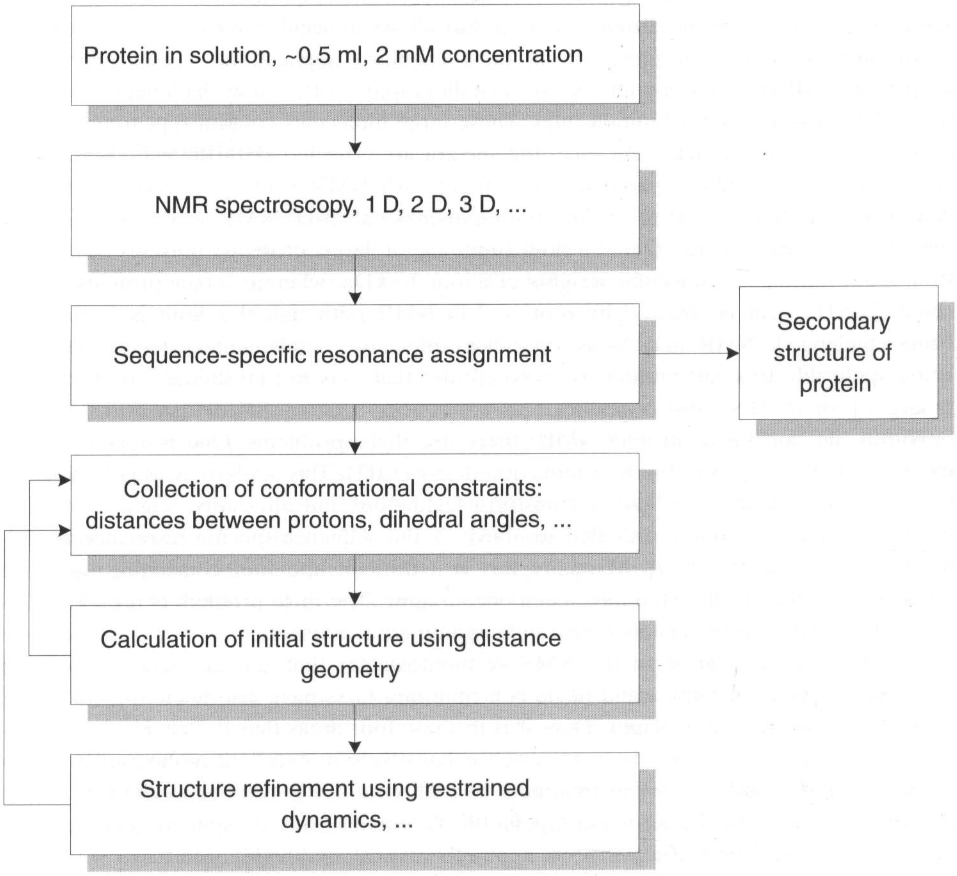

Protein in solution, ~0.5 ml, 2 mM concentration
NMR spectroscopy, 1 D, 2 D, 3 D, ...
Sequence-specific resonance assignment
Secondary structure of protein
Collection of conformational constraints: distances between protons, dihedral angles, ...
Calculation of initial structure using distance geometry
Structure refinement using restrained dynamics, ...

Figure 1. The flowchart of the protein structure determination process.

10.2.1 Basic Approach

The steps for determining solution structures from NMR data are depicted in Fig. 1. Multidimensional NMR data are acquired as a series of one-dimensional (1-D) spectra. Once the NMR spectra are acquired, individual peaks in the spectra must be assigned to sequence-specific locations in the primary structure of the protein before the distance information from NOESY (Nuclear Overhauser Enhancement SpectroscopY) can be fully interpreted. Sequence-specific NMR resonance assignment plays a pivotal role in the structure determination process.

Full analysis of the NOESY spectrum provides many distance restraints between the hydrogen atoms of a protein. The interproton distance can be calculated from the intensity of the NOE cross peaks, provided that a fixed distance can be found for calibration. Generally speaking, an NOE peak with strong intensity may indicate that two protons are within 2.5 Å of each other, while a weak NOE peak corresponds to an upper limit of 5 Å.

Many other geometrical restraints can be inferred using various methods. One of the restraints available from NMR data comes in the form of information concerning dihedral angles. Two dihedral angles are associated with each peptide bond. The angle ϕ is the torsion angle between bonds N–H and C_α–αH while the angle ψ is another torsion angle between bonds C_α–αH and C'–O'(Fig. 2).

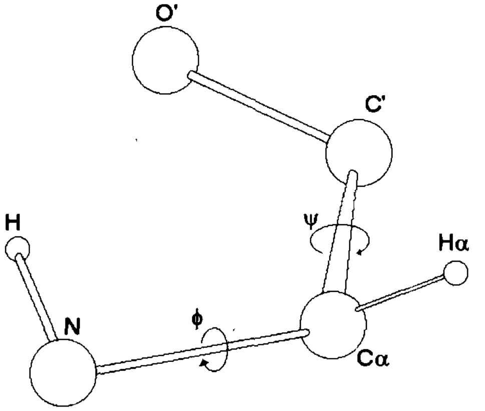


Figure 2. The torsion angles of an amino acid residue.

The dihedral angle ϕ can be calculated from the vicinal spin-spin coupling constant $^3J_{\alpha H\text{-}NH}$ using the Karplus equation [14, 15]:

$$^3J_{\alpha H\text{-}NH} = 6.4\cos^2\theta - 1.4\cos\theta + 1.9 \qquad (1)$$

where $\theta = |\phi - 60|$ and ^3J is given in Hertz. With the use of Eq. (1), measurement of $^3J_{\alpha H\text{-}NH}$ gives complementary information to NOE distance restraints for calculating the initial structure of a protein.

The next step is to determine an initial protein structure which is consistent with the hundreds of NOE restraints and, frequently, with some other conformational restraints. Distance geometry is the most commonly used mathematical procedure by which distance restraints are converted into 3-D structures [16]. The distance geometry procedure essentially is a projection from a high-dimensional space (in which thousands of distance relations can be accommodated) into ordinary 3-D space. The initial structures calculated from distance geometry almost always violate many of the experimental restraints. Subsequently, structure refinement is required to obtain a high-resolution protein structure.

10.2.2 The Important Role of Sequence-Specific Resonance Assignment

As described above, NMR spectra contain information from which biomolecular structures in solution can be determined. However, none of the embedded information can be used without having the resonances of the biomolecules assigned. In other words, it must first be determined which resonances come from which nuclear spins. This is a common problem or process in all types of spectroscopy. The process of associating specific spins in the molecule with specific resonances is called *sequence-specific resonance assignment.*

Sequence-specific resonance assignment is essential in three areas of biomolecular NMR application: (i) biomolecular structural analysis; (ii) intermolecular interaction with biopolymers; and (iii) studies of molecular dynamics. The importance of resonance assignment in these three areas is discussed below.

As a first discussion, consider the determination of protein structures from NMR data. The structural information comes mainly from NMR cross peaks. An NOE peak between two hydrogen atoms (or groups of hydrogen atoms) is observed if these hydrogens are located at shorter distances than approximately 5.0 Å from each other. Without sequence-specific resonance assignment, it is impossible to determine to which of the two hydrogen atoms a specific distance restraint refers. On the other hand, when combined with resonance assignment, these distance restraints can be attributed to specific sites along the protein chain and therefore the 3-D structure can be formed.

The second application where resonance assignment is pivotal is in studies of intermolecular interaction. For example, in the study of the protein-DNA binding interaction, it is important to know the binding sites. The intermolecular NOE peaks can be manifested at short distances between nuclear spins located in different interacting molecules. Without sequence-specific assignment, such NOE data merely indicate that the intermolecular interaction has occurred. When combined with assigned resonances, the NOE data identify the binding sites of the intermolecular contacts.

The study of protein dynamics has made significant progress during the past few years. These studies rely on the observation of certain spectral properties in distinct NMR lines (peaks) that can be correlated with intramolecular motion. Once the NMR lines responsible for the study region (such as a methyl group) have been assigned, it is then possible to investigate the desired spectral properties in the corresponding spectra. For a recent and more detailed review of NMR of macromolecules, see Driscoll and Kristensen [17].

10.2.3 The Difference Between Resonance Assignment and NOE Assignment

Before describing the strategy of protein resonance assignment, the sometimes confusing term "NOE assignment" should first be clarified.

The sequence-specific assignment of protein resonances is a process of associating specific nuclear spins in the protein with specific resonances, that is, chemical shifts. The process may or may not involve NOE data. In the traditional resonance assignment strategy using homonuclear 2-D NMR, the inter-residue connectivities are established from NOESY data. Recently, heteronuclear 3-D NMR has provided inter-residue connectivities through a series of triple resonance experiments; thus, there is no need to use NOE data.

NOE assignment is the analysis of the NOESY peak set to locate as many proton-proton distance restraints as possible. The sequence-specific resonance assignment usually assigns only a few backbone NOE correlations, such as $d_{\alpha N}(i,i+1)$, $d_{NN}(i,i+1)$, $d_{\alpha N}(i,i+3)$, and so forth. The backbone NOE correlations provide the required sequential connectivities for placing amino acid residues at their corresponding locations along the primary sequence. The majority of the NOE peaks, however, remain unassigned in the resonance assignment stage. The NOE assignment process is responsible for determining all the short- and long-range inter-residue NOE correlations.

Chemical shift degeneracy sometimes makes complete NOE assignment difficult in the protein side-chain region. For example, consider 10 protons resonating at 1.88 p.p.m., and there is an NOE peak with frequencies (1.88, 2.43) to be assigned. It is difficult to determine which one of the 10 protons is responsible for that NOE peak.

10.3 Summary of Manual Assignment Strategy

Resonance assignment has been one of the major hurdles for protein structural analysis from NMR data. Significant progress has been made through the introduction of 2-D, 3-D and even four-dimensional (4-D) NMR experiments. When combined with systematic approaches for spectral analysis, although it is still tedious and time-consuming work, the resonance assignment of protein spectra is no longer an unmanageable task.

Except for resonance assignment, most other aspects of protein structure determination rely heavily on computers. Therefore it is natural to seek software that permits fully automated resonance assignment. Before reviewing the main aspects of automated resonance assignment, the traditional but effective manual assignment strategy is described.

10.3.1 Manual Assignment from Homonuclear 2-D NMR Spectra

After the 2-D COSY (Correlation SpectroscopY) and NOESY experiments were first applied to proteins, it was realized that the intra- and inter-residue covalent linkage could be readily achieved, provided that the NMR data were of high quality. The idea for systematic assignment of proton resonances in proteins was first proposed by Wüthrich et al. [18] in 1982. Another approach, proposed by Englander and Wand [19], uses the same COSY and NOESY information, but in a different order. This approach is referred to as Main-Chain-Directed (MCD) assignment.

Wüthrich's assignment strategy includes the following steps:

1. The spin systems of the protons in individual amino acid residues are identified using as many as possible of the through-bond ^{1}H–^{1}H connectivities, which are mainly provided by 2-D COSY experiments.
2. Sequentially neighboring amino acid ^{1}H spin systems are identified from observation of the sequential NOE connectivities $d_{\alpha N}(i,i+1)$, $d_{NN}(i,i+1)$, or possibly $d_{\beta}N(i,i+1)$
3. Combining the information in the above, it is possible to establish chains of amino acid spin systems corresponding to peptide segments that are sufficiently long to be unique when compared to the primary sequence of protein. Sequence-specific assignment can then be obtained by matching the identified chains of spin systems with the corresponding segment in the independently determined protein primary sequence.

10.3.2 Identification of Amino Acid Proton-Proton Spin Systems

Identification of proton-proton spin connectivities (spin graphs) of individual amino acid residues is usually achieved by analysis of ^{1}H COSY spectra in D_2O solution after replacement of all labile protons with deuterium. An attempt is made to collect all J-coupled resonances arising from the same amino acid residue. The 20 common amino acid residues produce 10 different COSY connectivity patterns for the aliphatic protons and four patterns for the aromatic rings. Fig. 3 shows all of the 14 patterns from COSY spectra.

In a crowded COSY spectrum, spectral overlap and chemical shift degeneracy make the identification of unique patterns difficult. A RELAYED-COSY or (Total Correlation SpectroscopY) TOCSY spectrum [20], which provides redundant information about the amino acid patterns, often allows the ambiguous assignments to be resolved.

10.3.3 Sequential Assignment via Proton-Proton NOE

Using 2-D COSY and possibly TOCSY spectra, the ^{1}H amino acid spin systems can be identified. As shown in Fig. 3, certain amino acids have unique connectivity patterns, such as Val, Ile, Ala, Gly, Leu, and Thr. It is possible to assign the deduced spin systems to those unique amino acids directly. However, for AMX-type spin systems (one αH and two βHs), unique assignments are generally unachievable. Wüthrich [6] proposed four different methods to classify the amino acid types; these are summarized in Table 1.

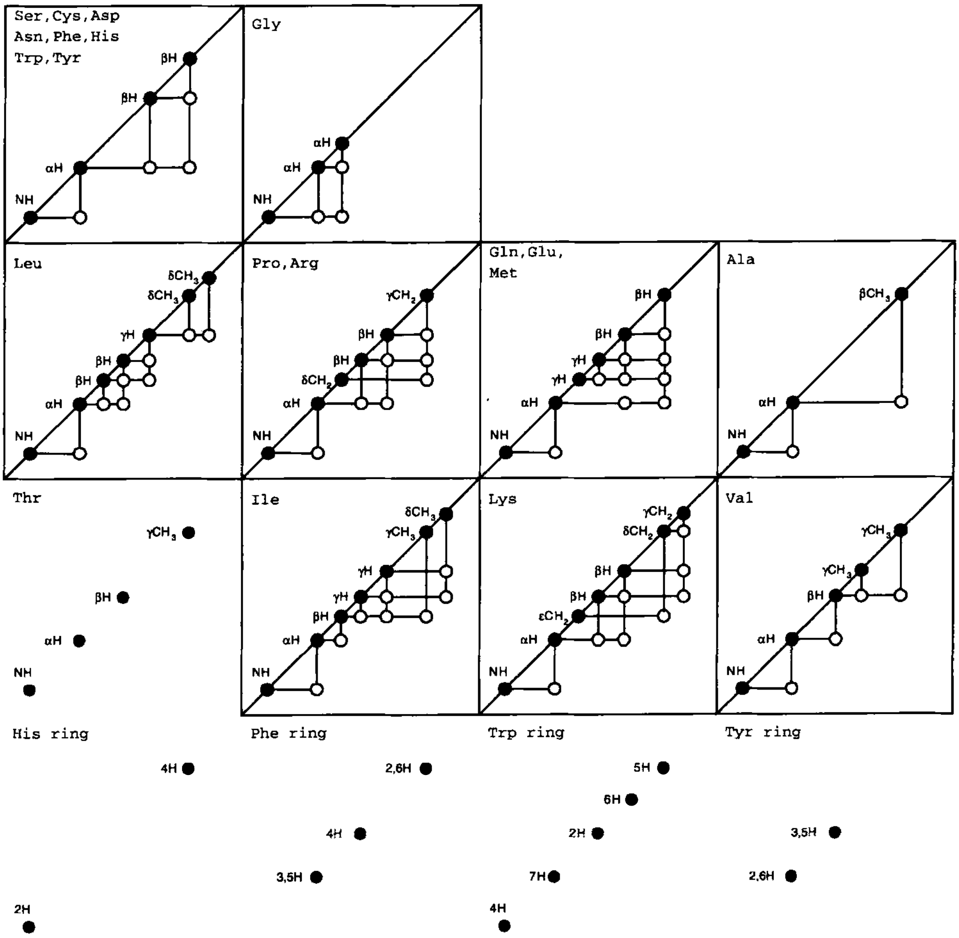


Figure 3. The 20 common amino acids and their spin coupling graphs.

Table 1. Four different classifications of the 20 amino acids.

Category	Number of amino acid types in this category	Descriptions
1	8	Gly, Ala, Val, Leu, Ile, Thr, (all αCH-βCH$_2$), (all others)
2	13	Gly, Ala, Val, Leu, Ile, Thr, Phe, Tyr, Trp, His Ser, (Cys, Asp, Asn), (all others)
3	15	Gly, Ala, Val, Leu, Ile, Thr, Phe, Tyr, Trp, His Ser, (Cys, Asp, Asn), Pro, (Lys, Arg), (Met, Glu, Gln)
4	18	Gly, Ala, Val, Leu, Ile, Thr, Phe, Tyr, Trp, His Ser, Cys, (Asp, Asn), Pro, Lys, Arg, Met, (Glu, Gln)

Before the NOE information can be used to create sequential connectivities, the deduced spin systems must be classified according to one of the above amino acid types. This task is achieved by inspecting the chemical shifts and the spin coupling patterns.

Wüthrich [6] also found that there is a high possibility that at least one proton among the NH, αH, or βH from one residue will be near (less than 3.5 Å, that is within the allowed NOE range) to the NH of the following residue. Thus by searching appropriate $d_{\alpha N}$, d_{NN} or $d_{\beta N}(i, i{+}1)$ NOE correlations in the NOESY spectrum, it should be straightforward to step from one residue to the next along the primary sequence. Once the connections between spin systems are established, the connected spin systems must be matched with the known protein primary sequence.

10.3.4 Manual Assignment from Heteronuclear 3-D NMR

Heteronuclear 3-D NMR experiments make use of the one-bond couplings, $^1J_{H-X}$ where X is ^{13}C or ^{15}N, to overcome the spectral line broadening problem. Several triple resonance NMR experiments have been designed to conduct the sequence-specific resonance assignments without using crowded NOESY spectra.

The inter-residue correlations are traditionally provided by NOE-type experiments where through-space dipolar couplings contribute to the observed cross peaks. Certain triple resonance NMR experiments, such as 3-D HNCA, HNCO, HCA(CO)N, also provide inter-residue correlations where one-bond scalar couplings contribute to the observed cross peaks. The names of these experiments are derived from the pathway by which polarization is transferred within or between amino acids. By properly combining several triple resonance NMR experiments, it is possible to establish a sequential walk from one residue to the next without using NOE information. Fig. 4 is an example where assignment is carried out by merging previously assigned frequencies with each subsequent spectrum.

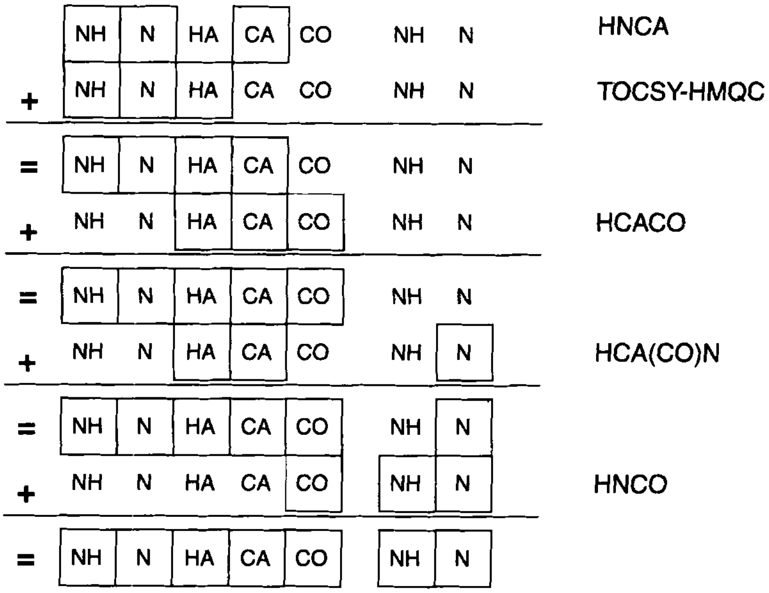


Figure 4. Correlations observed in the five triple resonance NMR experiments.

In the first two steps [HNCA and TOCSY-HMQC (Heteronuclear MultiQuantum Correlations)], the NH and ^{15}N frequencies of residue (*i*) are used to obtain the assignment of the C_α and αH of the same residue. The C_α and αH frequencies are then used to obtain assignments for the CO of residue (*i*) and ^{15}N of residue (*i+1*) with the HCACO and HCA(CO)N experiments. Finally, the CO and ^{15}N frequencies are used to find the NH proton frequency of residue (*i*) with the HNCO spectrum, thus completing one cycle of the assignment.

10.4 Automated Sequential Assignment in NMR Spectroscopy

As discussed in section 10.2, valuable information such as the distance restraints cannot be extracted until the spectra are properly assigned. Due to complexities and imperfections such as missing peaks, spectral overlapping and artifacts in the spectra, the assignment process is time-consuming and often incomplete. These difficulties also provide challenges for automated assignment procedures. Considerable effort has been devoted to the development of computer programs designed to permit automatic assignment, or at least to aid in the interpretation process.

The strategy of automated resonance assignment essentially parallels the manual assignment strategy. Although the integration of resonance assignment and structure calculation [21] has been proposed, almost all of the published attempts are designed for spin system identification, sequential assignment or both. In other words, structure calculations are usually separated from resonance assignments. Following this division, the automated assignment procedures are discussed first in this section. Following this, structure calculations are reviewed in section 10.5.

In general, automatic resonance assignment can be divided into three stages. The first stage is to extract all possible spin systems from the spectra, such as COSY and TOCSY in the 2-D case. In the second stage, attempts are made to map these spin systems to the 20 amino acid types. Chemical shift and spin coupling topologies are used in the pattern recognition process. Due to the clustering of the amino acid chemical shifts in narrow regions and the possibilities of missing peaks in the spin systems, however, each spin system is usually mapped to more than one amino acid type. That is, rather than mapping a spin system to a specific amino acid, a candidate list is generated in which all the possible mappings are listed and a ranking system is devised. The final stage is sequentially to assign these recognized spin systems to the primary sequence of the protein. This step is difficult to automate, as an exhaustive search of all possible sequential assignments based on the spin systems' mapping lists often leads to a combinatorial explosion.

In this section, the characteristics of automated resonance assignment tools are discussed, along with some problems and the limitations of automated assignment procedures.

10.4.1 Basic Requirements of Automated Methods

A complete automated assignment program should be capable of extracting spin systems from available spectral data. Furthermore, a tool should be provided for identification of the amino acid types. As for the sequence-specific assignment, both common approaches, that is use of inter-residue NOE and use of triple resonance heteronuclear 3-D NMR, should be taken into consideration. The design should allow the sequential connectivities to be created in a reasonable amount of time. A variety of algorithms has been applied to implement the above requirements, including those using systematic approaches [21–28], as well as artificial intelligence, such as expert systems [29], neural networks [30, 31], restraint propagation [32] and GAs [33].

An important characteristic of a good automated assignment program is that it should have the flexibility to accept many different types of NMR data from various experiments. NMR spectroscopists are continuously creating novel experiments. The continuing advance of NMR hardware and biotechnology generates specific experiments for a specific protein sample. Algorithms designed for specific types of experiments, however, sometimes outperform general-purpose algorithms because the latter might be unable to take full advantage of all the information embedded in a spectrum.

Although the ultimate goal of resonance assignment is complete automation, human intervention is inevitable in today's automated assignment tools due simply to the complexity of the spectral data which makes complete automation difficult to achieve. Automated assignment software should not become a "black box" which prevents users from understanding the internal actions and processes. It is better to allow the software to have the capability of interacting with users at various stages during the assignment process while simultaneously preventing it from becoming merely a book-keeping tool. In this regard, degenerate chemical shifts can result in strange spin systems. An example of this is a spin system with one αH and four βHs that can be generated due to degenerate αH chemical shifts. Although it is easy for computer algorithms to determine which spin systems are incompatible with the 20 common amino acids using the spin coupling patterns, human inspection is useful to separate such degenerate chemical shifts.

In order to obtain an accurate assignment, a program should ideally be able to use as much information as is available. Knowledge about the structural environment, such as a helix, β-strand, and coil may make it possible to predict the chemical shift range of certain protons.

In the actual assignment, the known chemical shift ranges can be treated as additional evidence to confirm or reject an assignment. The experimental conditions under which the spectra are acquired may help users predict which peaks are present in the spectra, and which are not. A mutant or homologous protein may be assigned rapidly as long as the original protein has been sequentially assigned [34, 35]. A 2-D ^{13}C HMQC spectrum may help to unfold the ^{13}C chemical shifts of a 3-D spectrum. Such miscellaneous information is sometimes essential for a successful resonance assignment.

In terms of the quality of the NMR data, a good automated assignment tool should be able to overcome problems caused by false and missing peaks. The software should tolerate missing peaks to a considerable extent, just as it should also be able to reject false data. To accomplish this, automated assignment algorithms should inspect all logical relationships that exist between a suspicious peak and its surroundings. A genuine peak must have several coupled neighboring peaks as evidence, whereas a false peak may have one connected neighbor and is less likely to have two or three neighbors.

Data processing prior to assignment also plays a significant role in the design of automated assignment software. Spectral artifacts which might be confusing to automated assignment procedures should be removed prior to the start of the actual assignment process. Before performing a Fourier transform on the time-domain data, zero filling, linear prediction [36] and Karhunen-Loeve transformation [37] are useful. After Fourier transform, ridges of t_1 noise can be removed manually [38, 39].

The most critical preassignment processing is the peak picking procedure. The simplest approach is either to pick all points above a given threshold, or to use a maxima detecting procedure to find local maxima. These simple approaches seem incapable, to date, of providing reliable peak lists. A great number of peaks – many of them noise – can be generated. A more advanced approach is to implement user-defined peak shapes (for example, ellipsoids) and search for peaks having those shapes in the spectrum. Garrett et al. [40] have designed a software package called CAPP (Computer-Aided Peak Picking) based on this approach. Artificial neural networks [41], after training with examples, also have the capability to distinguish real from false peaks.

Spectral alignment is another preassignment problem. Almost all assignment strategies use several different types of spectra, and the same hydrogen atom may appear at slightly different positions in those spectra. This chemical shift inconsistency can cause problems when comparing chemical shifts or peaks from two or more different spectra. If the inconsistency is systematic – that is, all nuclear spins shift in the same direction by roughly the same distance – the correction is straightforward. Otherwise, a common approach is to introduce tolerance values in the actual assignment stage. Every comparison between two chemical shifts from different spectra must pass the tolerance, though of course, some incorrect matches are inevitable.

10.4.2 Human versus Machine

Some people argue that automated assignment tools do not have much use, simply because computers can do no more than humans can do. Although the argument is true, this does not imply that the computer-assisted assignments are valueless. Due to the complexity of the task, complete automation of resonance assignment remains a worthwhile goal. Properly designed automated assignment software, however, does reduce the effort and the time required to assign a spectrum.

Another common argument is that automated assignment tools should be able to obtain the results with fewer data than humans need. Many of the present automated assignment programs simply emulate manual assignment strategies. It is apparent that to achieve the goal of "using fewer NMR experiments", one must exclusively implement different assignment strategies for computers. We would like to emphasize, however, that computer programs cannot achieve what people cannot. If a person cannot complete the assignment using a limited data set in an unlimited amount of time, there is no reason to expect that computers will succeed.

Manual assignment is not 100 % deterministic; that is, independently obtained assignments from two persons might differ because of differences in human bias and intuition. In contrast, every step is deterministic in most computer assignment strategies, as intuition and bias are not involved. If a person is able to assign protein NMR data without using any personal bias or intuition – that is, every step has a clear logical basis – computer-assisted assignment tools should be able to produce identical assignment in much shorter time. This is probably the main advantage of using automated resonance assignment tools.

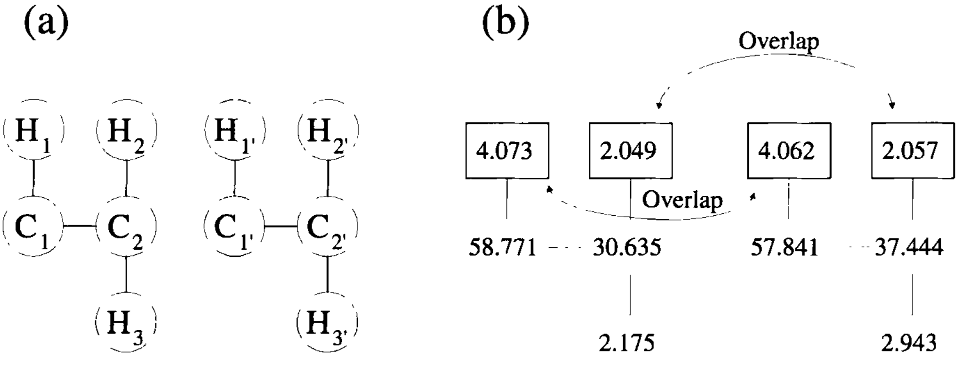


Figure 5. Schematic illustration showing how overlapped resonances are resolved. (a) Fragments from two molecules are shown. (b) The chemical shifts of the protons and carbons are displayed. Resonances in boxes are those having significantly overlapped chemical shifts.

Even so, cases exist that cannot be resolved. Putting missing peaks and low resolution aside for the moment, consider two glutamines at positions E16 and E21 in the 90-residue protein N-domain of chicken skeletal troponin-C. As seen in Fig. 5, the data for these two residues, obtained from 3-D HCCH-COSY/TOCSY, are heavily overlapped, and neither human nor machine can disentangle this. Such cases of severe overlap underscore the major challenge in devising automated methods.

10.4.3 Automated Procedures which Incorporate GAs

Zimmerman and Montelione [11] reviewed and compared progress in the development of semi- and fully automated approaches to protein resonance assignments up to 1995. Wehrens et al. [33] treated the sequential assignment as a subset selection problem. In reference to the difficulties relating to the combinatorial explosion, consider that there are N spin systems identified and these systems are to be mapped to M positions in the protein sequence (assume $N > M$ in most cases). The size of the search space is therefore $N!/(N-M)!$. This number can become too large for systematic and exhaustive searches. To solve this problem, a GA with a specially developed subset encoding along with specifically designed crossover and mutation operators was used. The candidate solutions were represented as a permutation of the N possible elements and only the first M elements are evaluated in the fitness function. The fitness function used in this work was basically the count of how many valid pattern combinations were presented in the solution. The crossover operator was designed to preserve position as much as possible and two special mutation operators, *reorder mutation* and *trade mutation*, were used. Reorder mutation swaps two elements in the first M elements of a solution, while trade mutation swaps an element from the first M elements with an element of the last N-M elements.

The method was tested on a number of real and simulated data sets, and the authors [33] assert that the GA is a promising technique to be used in automatic sequential assignment of NMR spectra, especially in cases where the errors present prohibit other techniques from producing meaningful results. Our own work in this area has tackled the problem of missing peaks and overlap for 2-D NMR [22–24] and 3-D NMR [26–28] using graph theory and fuzzy mathematics. Essentially, peak sets from various experiments are merged using a number of restraints. The resulting spin graphs are then mapped to amino acids using a knowledge base of chemical shifts. The knowledge base contains not only the chemical shifts, but also the standard deviations of each resonance. Pattern recognition of the graphs makes use of fuzzy mathematics. The main difference between the 2-D and 3-D strategy is that, in the former case, the algorithms attempt to identify the complete residue, while in the latter case, 3-D data were first used to generate the protein backbone. Following this, the aliphatic side chains are mapped to the backbone to give the final assignment.

Good results were obtained with good data sets, although the generation of too many amino acid candidates for a given spin graph often led to a combinatorial explosion. Although refinements improved this, a way to reduce the number of candidates is to improve the knowledge base. Employing neural networks [42], the conformational dependencies of the chemical shifts of ^{1}H, ^{13}C, and ^{15}N were studied. Depending on the conformation, (helix, sheet, and coil), small but systematic shifts of the chemical shift data were found. Other approaches to improve pattern recognition based upon neural networks, GAs and SA have had success in this area and are discussed below.

In another application of neural networks, Huang et al. [43] used 456 spin systems from several proteins containing various types of secondary structure to train the network. This was tested on human ubiquitin, which contained no homologies with the training set. In 60 % of the spin systems, the correct amino acid class was among the top two choices given by the network, while for 96 % of the spin systems, the secondary structure was correctly identified, thus showing the potential for neural networks for resonance assignments.

GARANT is General Algorithm for Resonance AssignmeNT [44, 45] for the automatic resonance assignment of NMR spectra of proteins. Available free from Wüthrich's laboratory (http://www.mol.biol.ethz.ch/wuthrich/software/garant/), GARANT uses data from various NMR spectra, different line shapes, peak intensities or 3-D structures and chemical shifts of homologous proteins. This greatly aids in circumventing the problems of incomplete data. Moreover, it combines the matching of spin graphs with a scoring technique and the use of a GA with a local optimization routine. The task of finding the optimal homomorphism between the graphs and observed peaks grows exponentially with the size of the problem, and use of a GA avoids these excessive calculations. Tests show that combining GA and local optimization routines yields results that are clearly superior to those obtained when using either of the two techniques separately. In the systems studied [44], nearly complete assignment of the polypeptide backbone resonances and assignment of over 80 % of the amino acid side-chain resonances was obtained without manual intervention. The authors point out that further improvement of automated resonance assignments may primarily depend on the development of peak-picking procedures with improved handling of spectral overlap and noise bands.

A pattern recognition algorithm, developed by Croft et al. [46] claims success in reducing the search time in pattern recognition. Using 3-D data, assignment strategies are applied directly to the NMR data to identify both side-chain and backbone spins. The program goes through a series of filters, the last of which employs various heuristic arguments to remove redundant results. Applicable to a wide variety of spectrum types, this approach reduces computational time and shows good results on the systems studied.

Filtering and reducing the number of candidates from pattern recognition involves assigning a score to each. Lukin et al. [47] report success in the use of SA to maximize the score. First, they use statistical methods to identify the spin systems from 3-D NMR, and then SA generates the final sequential assignment. In the cases studied, although these authors report excellent assignment results, they also point out that a number of problems arise due to poor dispersion of chemical shifts and low SNR (signal-to-noise ratio). They also indicate that chemical exchange and conformational heterogeneity can lead to different sets of chemical shifts. These problems should be overcome by the SA optimization leading to one assignment. The algorithms are designed especially to use 3-D NMR data and are applicable to larger proteins.

10.4.4 Conclusion

There is no doubt that during the past five years significant progress has been made in the automation of the sequence-specific assignment of protein resonances. Earlier bookkeeping techniques have given way to serious attempts to carry out complete assignments. The programs have moved from 2-D to 3-D NMR; allowed for a large variety of experiments; encouraged manual input and involved schemes which more or less followed manual protocols. The difficulty that these attempts face is a combinatorial explosion of possible assignments as the protein size increases. The use of neural networks, SA, and GAs – and combinations of these – have shown various degrees of success in tackling this problem. A significant problem, mentioned in most applications, is the inability of peak picking programs to pick accurate peak lists, and this is an area where improvements are needed. It seems likely that these approaches will eventually lead to methods that will give reliable and reproducible results which the experimentalist can trust, and which will allow for the routine and rapid structure determination of many interesting proteins.

10.5 Protein Structure Optimization

As discussed in section 10.2, once the sequence-specific resonance assignment has been carried out, it is then known from NOESY experiments which amino acid residues are in close proximity. Although a complete assignment is not necessary, the NOESY data available provide distance restraints that are used in distance geometry optimization. A multidimensional space is reduced to a 3-D structure space, and in the process a global minimum is sought which corresponds to the best structure.

Several studies using GAs for protein structure prediction have been made during the past five years, and a detailed review about this topic has been prepared by Pedersen and Moult [48]. Besides these, GAs have also been applied to molecular modeling, QSAR (quantitative structure-activity relationships) and drug design [49], as described in the remainder of this book.

DGII [50] is a well-established method that uses distance geometry (DG) to determine structure. The individual structures generated by distance geometry (DGII) calculation are usually of poor quality when the experimental data are incomplete and imprecise. In most cases, they are only partially matched with the structural properties of the true structure. Another method called DGΩ (Distance Geometry-Optimized Metric matrix Embedding by Genetic Algorithms) was proposed [51], which uses a GA to combine well-defined parts of individual structures generated by a distance geometry program such as DGII. Sets of new upper and lower bounds are then identified within the original experimental restraints that restrict the sampling of the metrization algorithm to the most promising regions of conformational space. DGΩ appears to improve the convergence behavior as well as sampling properties compared to standard distance geometry techniques.

In the DGΩ approach, a complete set of modified restraints is encoded in each string (chromosome). These string-represented structures are then used as input for the DGII algorithm. After calculation and fitness evaluation, these strings are recombined by the crossover operator and a special mutation operator is then applied to adjust the restraints.

By using this evolutionary optimization strategy, the process limits the sampling of the metrization algorithm to specific ranges located within the original bounds, and a better structure can often be generated. Details about the implementation of the algorithm and further discussions are available [51].

Van Kampen et al. [52] compared DGΩ with the earlier DGII and GAT [52], a GA for direct optimization in torsion angle space. On comparing the three methods, it was found that DGΩ was slightly better than DGII and GAT in sampling and convergence properties, but required more computational effort. GAT-generated structures, although inferior to those from DGII and DGΩ, have a better defined covalent geometry. In spite of combining DGII with a GA, the DGΩ method required the same refinement time as when SA was used.

Van Kampen and Buydens [53] also compared a GA with SA for structure determination of a heptapeptide in torsion angle space. The study focused on the crossover operation in GA and concluded that, in this case, the GA was outperformed by SA.

May and Johnson [54, 55] used a GA, dynamic programming and least-squares minimization to compare protein structures. Their first paper describes how a GA is used to search for optimal structure comparisons, and the second discusses refinements of their methods. Pointing out that there are a number of reasons for wanting to compare the structures of two proteins, they describe how a GA is used, with the only required input being the sets of co-ordinates. Their method defines at least the same number of topological equivalences as the other procedures for this purpose, but always with a lower root mean square distance between them.

In a continuation of this work [54, 55], May and Johnson describe an extension to their methods of finding local structural similarities among families of unrelated protein structures. Their program [56], called GENFIT, evolved from the above work and can locate and superimpose regions of local structural homology regardless of their position in a pair of structures, the fold topology or the chain direction. Again based upon a GA, the computations use parallel processing and converge rapidly. A number of examples are given which convincingly demonstrate the viability and usefulness of their methods. In a similar, and earlier approach, protein surfaces are compared using a GA, again reporting good success [57].

In a paper by Li et al. [58], a GA is used to identify the calcium ion positions in the crystal structure of a bovine prothrombin fragment. After NMR is used to determine the polypeptide structure, a GA is used to find the most frequently occurring low-energy structure for the metal placement, after which molecular dynamics (MD) simulations were undertaken to minimize the energy. The initial placement of the metal ion was determined by obtaining search parameters that converged to the lowest energy structure. The majority of the structures obtained were identical, and this was chosen as the starting point for refinement, while the others were diverse and discarded. This paper [58] describes the method.

Ca^{2+} conantokin-T complexes [59, 60] were studied with NMR. The peptide structure was determined by conventional 2-D NMR, using TOCSY and DQF-COSY, from which the resonances were assigned. Following this, the structure was determined by use of distance geometry based upon NOEs. The genetic algorithm of Li et al. [58] was used to study the docking of the metal ions. Using the techniques described, the initial positions of the metal atoms were determined using the GA, after which molecular simulations were used to refine the docking locations. This application of a GA to elucidate protein function lends support to the methods and its utility for further similar studies. The structure and function of conantokin G has also been studied by Rigby et al. [61] using NMR. A GA was used to locate the initial calcium position, which was then refined by MD.

GENFOLD [62] is a GA that calculates protein structures using restraints obtained from NMR, such as distances derived from NOEs and dihedral angles. Three proteins were studied, the largest being 108 residues. The program calculated structures that were close to the target structures and could then be refined by SA to give structures indistinguishable from the targets. Applicable to both helical and sheet proteins, the size of the protein does not increase the optimization time significantly. The advantages offered by GENFOLD over other structure determination methods are discussed in detail.

Beckers et al. [63] use a GA to interpret NOE distance restraints on the 3-D structure of dimers present in a DNA complex. The GA was used to minimize violations of the distance restraints. The torsion angles were allowed to vary over a wide range, and by keeping the bond angle geometries fixed, while applying the DG calculations, it was shown that bond angle geometry has a significant effect on the resulting conformation. The algorithm optimized to one family of structures and this differed from distance geometry calculations. The distance geometry calculations produced unreliable bond angles, while the GA yielded more reliable results, thereby lending credibility to the GA approach.

The distance restraints from NMR are not consistent with a single conformation, but rather an ensemble of conformers. With a program called FINGAR [64] (FIt NMR using Genetic AlgoRithm), Pearlman studies a weighted set of structures that best fits measured NMR-derived data. Noting that DG calculations can generate structures that fulfil the NOE restraints, he points out that DG does not take into account the potential energy effectively. He shows appreciable advantages over commonly used refinement methods using FINGAR and suggests that it is a viable alternative to DG and MD, especially for small molecule structure refinement. Recently [65], FINGAR has been extended to improve the scoring function by removing problem restraints. Problem restraints are inaccuracies in data and incorrect assignments of NOE peaks. These errors can cause the convergence to fail or to produce a distorted structure. The improved method locates the problem restraints and removes them, leading to a disproportionate improvement in the value of the scoring function.

SAGA [66] (Sequence Alignment by Genetic Algorithm) is an optimization approach which avoids the problem of converging to a local minimum. Emphasis is placed on constructing the objective function (OF) and its optimization by a GA. When applied to systems with known tertiary structure, SAGA performed well. Notredame and Higgins [66] compare SAGA to molecular simulation algorithms and other stochastic optimization approaches, and find that SAGA does as well or better in the cases studied. Although not as fast as other optimization procedures, they point out that they have not attempted to improve the efficiency of SAGA but rather have focused on the method. Their most recent work [67] has improved the efficiency and developed a new method called COFFEE (Consistency based Objective Function For alignmEnt Evaluation). They show that multiple sequence alignments can be optimized for their COFFEE score with the GA package SAGA.

10.6 NMR Spectral Parameter Estimation from Time Domain Data

10.6.1 Shortcomings of Discrete Fourier Transform

Discrete Fourier Transformation (DFT) is the most important technique for time series estimation. The first "FT-NMR" experiment was carried out by Ernst and Anderson in 1965 [68]. In the same year, Cooley and Tukey published an algorithm for computing the DFT in a much faster way [69]. Computation order improved from $O(N^2)$ to $O(N \log N)$. There is no doubt that DFT has played a crucial role in the evolution of the NMR technique, and revolutionized both instrumentation and application.

Despite the usefulness of DFT, it has some well-known shortcomings for NMR data analysis under certain conditions, such as low digital resolution and spectral side bands in truncated data and short time series. All these hamper the quantification of spectroscopic data. One of the examples is the quantification of *in vivo* magnetic resonance spectra. *In vivo* magnetic resonance spectroscopy (MRS) allows the noninvasive investigation of metabolic states in selected organs of human body. To use this technique as a medical diagnostic tool, reliable quantification of the metabolite concentrations is vital. Severe problems, however, are encountered because *in vivo* MRS data usually suffer from a poor SNR, a low spectral resolution, and strong overlapping peaks. Owing to these reasons, various spectral processing algorithms have been proposed to overcome the intrinsic limitations of DFT. Among them is a class of methods which employ the specific mathematical properties of exponential decay behind the NMR data model to quantify the spectral parameters directly from the time domain data [70]. The advantage of such time domain analysis is that any data preprocessing can be avoided before the data are fitted to models.

10.6.2 Alternative Data Processing Methods

NMR data processing is treated in depth by Hoch and Stern [71], whose book covers many topics in NMR data processing such as fast Fourier transform (FFT), parameter estimation, maximum entropy, and many others. Of the various spectral processing algorithms that have been proposed here to overcome limitations of DFT, only those methods based on time domain data analysis are discussed. One advantage of time domain analysis is that any data preprocessing can be avoided before the data are fitted to models. Spectral parameters, such as frequencies, damping factors, phases and amplitudes of the signal components can be extracted directly from the time domain FID data. Moreover, time domain analysis is more desirable for analysis from the point of view of automatic data processing. In spite of these advantages, the commonly used time domain data analysis methods usually fail when the SNR becomes low. It is in this area that GAs have advantages.

Common methods of spectral parameter estimation are Iterative Quadratic Maximum Likelihood (IQML) [72]; Linear Prediction Singular Value Decomposition (LPSVD) [73]; Total Least Square (TLS) [74] and Matrix Pencil (MP) [75]. Although all these methods have found to be advantageous in application, the focus here is on comparing the performances of these methods to a GA approach when applied to data with low SNR.

IQML appears to outperform the other three methods when the SNR is low and when the signal components of the FIDs are not overlapped. In situations where there is serious peak overlap and the signal components have a wide ranges of damping factors (decaying rates) and amplitudes, IQML performs similarly to the TLS method and it outperforms the most commonly used LPSVD method and the recently proposed MP method.

Usually, some *a priori* knowledge is available such as the frequencies and damping factors of some peaks. Optimization methods can incorporate these as restraints when searching parameter space. Using *a priori* knowledge, GA optimization can outperform IQML when the SNR of the FID is low, although IQML does not use *a priori* knowledge. Fig. 6 shows the results from eight damped sinusoidal signals with their frequencies, damping factors and amplitudes selected randomly. Ten trials were performed and in each, different random noise was added. The ranges of the frequency restraints do not need to be narrow. Although the computational time for GA optimization is longer than conventional methods, a GA can be used as a complementary method when the SNR is low and can be incorporated into different models when necessary. Fig. 6 shows that the GA outperforms the IQML methods, especially with respect to the resolution of overlapped lines.

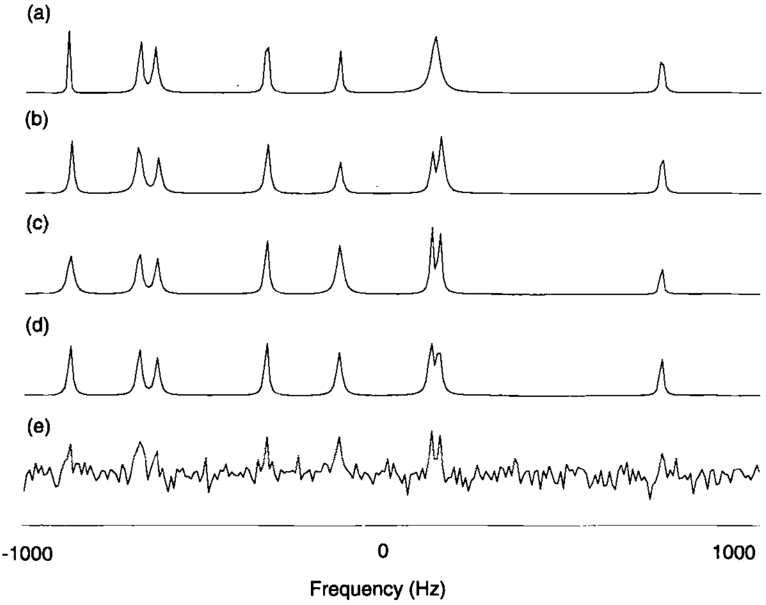


Figure 6. (a) FFT of the FID (average of 10 trials) reconstructed by IQML methods (window size used is equal to 48). (b) FFT of the FID (average of 10 trials) reconstructed by GA method with frequency restraints [42]. (c) Same as (b) with different frequency restraints [42] (d) FFT of the noiseless FID. (e) FFT of one of the FIDs in the 10 trials showing the noise.

Metzger et al. [76], in a similar study to the above, applied a GA to spectral quantification. They note that most fitting methods developed for high-resolution NMR are based on a Lorentzian line shape and do not work well for *in vivo* data where the peaks have more complicated line shapes. Moreover, linear prediction methods do not allow for *a priori* knowledge. Nonlinear optimization procedures, although allowing for *a priori* knowledge, are time consuming and have difficulty reaching a global minimum. Metzger et al. [76] demonstrate that GA optimization offers a robust alternative for analyzing spectroscopic data, especially those obtained with spatial localization techniques. They show GA to be a general technique which can accommodate a variety of line shapes, and allows for the incorporation of *a priori* knowledge.

Weber et al. [12] also consider heuristic optimization algorithms (GAs and SA) as viable alternatives to spectral quantification for MRS. They point out that poor SNR, low spectral resolution, and strongly overlapping peaks limit the clinical application of MRS using conventional methods, and that heuristic optimization methods can help resolve these problems. For maximum likelihood methods (IQML), the minimization often terminates at a local minimum. Weber et al. [12] find that GAs and SA are valuable alternatives, and their report outlines the details of how the GA is implemented, and discusses the criteria for fitness and mutations. Several cases are found where conventional optimization techniques became trapped in local minima, but a GA found better values. It was also found that the starting values have less influence on the results than in conventional techniques. Although more reproducible results were found with SA than with the GA, extremely deep minima were sometimes found with the GA and not with SA. With different advantages and short-comings of both SA and GAs, it was not possible for Weber and coworkers to recommend one method over another.

10.7 NMR Pulse Shapes

In analyzing complex spin systems by NMR, much effort has been devoted to solving the problem of exciting a spin system nonuniformly across a frequency band under investigation, for example, tailored excitation and solvent suppression sequences. The rationale for designing such sequences is usually based on the approximate Fourier transform relation between the time-domain pulse envelope and the frequency-domain excitation pattern. This fails, however, for pulses with large flip angles. In these cases, the frequency-domain excitation profile of such a pulse train is calculated by solving the Bloch equations numerically, and the shape of this excitation spectrum is monitored.

Freeman and coworkers have reported that a GA approach leads quickly to acceptable pulse shapes [77, 78]. Simple examples were used to demonstrate the idea. Pulse shaping functions are used for gene encoding, and instead of defining an objective function to evaluate the fitness of the candidate solutions, the operator intervenes and chooses the most "promising" chromosomes as parents for the next generation. After some generations, pulse trains that are suitable for specific purposes can be generated. It was concluded that the genetic evolution approach could be successfully used for NMR pulse shaping. Moreover, insight can be obtained by noting how the various shaping functions influence the excitation spectrum.

In similar studies, Lunati et al. [79, 80] have applied evolutionary strategies to optimize adiabatic pulses for use in MRI, as well as high-resolution NMR. Adiabatic pulses [81] have now supplanted earlier attempts to tailor and create selective excitation pulses using composite pulses. An important reason for this is that composite pulses have a high power requirement, while adiabatic pulses do not. Lunati et al. have approached the problem by optimizing not only the adiabaticity factor, but also simultaneously minimizing the power. They find that their algorithm for selective excitation pulses converges in a few minutes on a PC.

10.8 Summary

The ultimate goal of protein research lies in understanding their function, how to intervene clinically, and the origins and mechanisms of life. NMR is a valuable tool along the way, and GAs have made significant contributions to this area. One of the advantages of GAs is that they have wide applicability and are not limited to specific areas. For example, Reijmers et al. [82, 83] use a GA to construct phylogenetic trees [84] which show the divergence of protein structures over their evolutionary history. There seems no better application of GAs than to track the natural history of protein evolution.

The application of GAs, however, has made significant contributions to the very areas of protein structure determination by NMR that have caused the most difficulties. These are: data processing; the resonance assignment problem; and the structure refinement problem. In all cases where optimization is required, GAs have done as well as, if not better than, alternative methods. Protein structure determination from NMR is still not completely automated. It seems likely that human intervention will always be necessary in order to improve the speed and accuracy of subsequent optimizations. Certainly, in the light of the successes to date, this progress will continue and GAs will play a significant, if not the major, role in overcoming the existing difficulties.

Acknowledgments

I have been fortunate to have had two recently graduated students [Dr. Kuobin Li (1996) and Dr. Wing-Yiu Choy (1998)] who each wrote excellent theses [5, 7] on computer-assisted assignment methods and numerical methods for the NMR data processing. I have relied extensively on their work and am grateful to them for critical reading of the manuscript. This work is supported by a grant from the Natural Sciences and Engineering Council of Canada (NSERC).

References

[1] R. R. Ernst, G. Bodenhauser, A. Wokaun, *Principles of Nuclear Magnetic Resonance in One and Two Dimensions*, Oxford Science Publications, Oxford, **1987**.

[2] R. Freeman, *Spin Choreography – Basic steps in High Resolution NMR Spectroscopy,* Oxford University Press, Oxford, **1997**.

[3] G. Wagner, S. G. Hyberts, T. F. Havel, NMR Structure Determination in Solution: A Critique and a Comparison with X-ray Crystallography, *Annu. Rev. Biophys. Biomol. Struct.* **1992**, *21*, 167–198.

[4] J. C. Hoch, A. S. Stern, *NMR Data Processing*, Wiley-Liss, New York, **1996**.

[5] W. Y. Choy, *Using Numerical Methods and Artificial Intelligence in NMR Data Processing and Analysis*, PhD Thesis, Department of Chemistry, McGill University, **1998**.

[6] K. Wüthrich, *NMR of Proteins and Nucleic Acids*, Wiley, New York, NY, **1986**.

[7] K. B. Li, *Development of Computer-Assisted Methods for the Resonance Assignment of Heteronuclear 3D NMR Spectra of Proteins*, PhD Thesis, Department of Chemistry, McGill University, **1996**.

[8] J. N. S. Evans, *Biomolecular NMR Spectroscopy*, Oxford University Press, Oxford, **1995**.

[9] J. L. Markley, S. J. Opella, *Biological NMR Spectroscopy*, Oxford University Press, NY, **1997**.

[10] K. Wüthrich, Biological Macromolecules: Structure Determination in Solution, in D. M. Grand, R. K. Harris, (Eds.), *Encyclopedia of Nuclear Magnetic Resonance*, John Wiley & Sons, New York, **1996**.

[11] D. E. Zimmerman, G. T. Montelione, Automated Assignment of Nuclear Magnetic Resonance Assignments for Proteins, *Curr. Opin. Struct. Biol.* **1995**, *5*, 664–673.

[12] O. M. Weber, C. O. Duc, D. Meier, P. Boesiger, Heuristic Optimization Algorithms Applied to the Quantification of Spectroscopic Data, *Magn. Reson. Med.* **1998**, *39*, 723–730.

[13] T. L. James, V. J. Basus, Generation of High-resolution Protein Structures in Solution from Multidimensional NMR., *Annu. Rev. Phys. Chem.* **1991**, *42*, 510–542.

[14] M. Karplus, Contact Electron-spin Coupling of Nuclear Magnetic Moments, *J. Chem. Phys.* **1959**, *30*, 11-15.

[15] A. Pardi, M. Billeter, K. Wüthrich, Calibration of the Angular Dependence of the Amide Proton-C Proton Coupling Constants, $^3J_{HN}$, in a Globular Protein: Use of $^3J_{HN}$ for Identification of Helical Secondary Structure, *J. Mol. Biol.* **1984**, *180*, 741–751.

[16] T. F. Havel, I. D. Kuntz, G. M. Crippen, Theory and Practice of Distance Geometry, *Bull. Math. Biol.*, **1983**, *45*, 665–720.

[17] P. C. Driscoll, S. M. Kristensen, NMR of Natural Macromolecules, G. A. Webb (Ed.) in Specialist Periodical Reports, *Nuclear Magnetic Resonance*, The Royal Society of Chemistry, pp. 292–336, **1998**.

[18] K. Wüthrich, G. Wider, G. Wagner, W. Braun, Sequential Resonance Assignments as a Basis for Determination of Spatial Protein Structures by High Resolution Proton Nuclear Magnetic Resonance, *J. Mol. Biol.* **1982**, *155*, 311–319.

[19] S. W. Englander, A. J. Wand, Main-chain-directed Strategy for the Assignment of ^{1}H NMR Spectra of Proteins, *Biochemistry* **1987**, *26*, 5953–5958.

[20] L. Braunschweiler, R. R. Ernst, Coherence Transfer by Isotropic Mixing: Application to Proton Correlation Spectroscopy. *J. Magn. Reson.* **1983**, *53*, 521–528.

[21] H. Oschkinat, D. Croft, Automated Assignment of Multidimensional Nuclear Magnetic Resonance Spectra, *Methods Enzymol.* **1994**, *239*, 308–318.

[22] J. Xu, B. C. Sanctuary, CPA: Constrained Partitioning Algorithm for Initial Assignment of Protein 1-H Resonances from MQF-COSY, *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 490–500.

[23] J. Xu, B. C. Sanctuary, B. N. Gray, Automated Extraction of Spin Coupling Topologies from 2D NMR Correlation Spectra for Protein ^{1}H Resonance Assignment, *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 475–489.

[24] J. Xu, S. K. Straus, B. C. Sanctuary, L. Trimble, Use of Fuzzy Mathematics for Complete Automated Assignment of Peptide ^{1}H 2D NMR Spectra, *J. Magn. Reson. B* **1994**, *103*, 53–58.

[25] J. Xu, P. L. Weber, P. N. Borer, Computer-assisted Assignment of Peptides with Non-Standard Amino Acids, *J. Biomol. NMR* **1995**, *5*, 183–192.

[26] K. B. Li, B. C. Sanctuary, Automated Extracting of Amino Acid Spin Systems in Protein using 3D HCCH-COSY/TOCSY Spectroscopy and Constrained Partitioning Algorithm (CPA), *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 585–593.

[27] K. B. Li, B. C. Sanctuary, Automated Assignment of Proteins using 3D Heteronuclear NMR. Part I: Backbone Spin Systems Extraction and Creation of Polypeptides, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 359–366.

[28] K. B. Li, B. C. Sanctuary, Automated Assignment of Proteins using 3D Heteronuclear NMR. Part II: Side Chain and Sequence-specific Assignment, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 467–477.

[29] C. Yu, J.-F. Hwang, T.-B. Chen, V.-W. Soo, RUBIDIUM, a Program for Computer-Aided Assignment of Two-dimensional NMR Spectra of Polypeptides, *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 183–187.

[30] J. U. Thomsen, B. Meyer, Pattern Recognition of the ^{1}H NMR Spectra of Sugar Alditols using a Neural Network, *J. Magn. Reson.* **1989**, *84*, 212–217.

[31] B. J. Hare, J. H. Prestegard, Application of Neural Networks to Automated Assignment of NMR Spectra of Proteins, *J. Biomol. NMR* **1994**, *4*, 35–46.

[32] D. Zimmerman, C. Kulikowski, L. Wang, B. Lyons, G. T. Montelione, Automated Sequencing of Amino Acid Spin Systems in Proteins using Multidimensional HCC(CO)NH-TOCSY Spectroscopy and Restraint Propagation Methods from Artificial Intelligence, *J. Biomol. NMR* **1994**, *4*, 241–256.

[33] R. Wehrens, C. Lucasius, L. Buydens, G. Kateman, Sequential Assignment of 2D-NMR Spectra of Proteins using Genetic Algorithms, *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 245–251.

[34] C. Redfield, C. M. Dobson, Sequential 1-H NMR Assignment and Secondary Structure of Hen Egg White Lysozyme in Solution, *Biochemistry* **1988**, *27*, 122–136.

[35] C. Redfield, C. M. Dobson, ^{1}H NMR Studies of Human Lysozyme: Spectral Assignment and Comparison with Hen Lysozyme, *Biochemistry* **1990**, *29*, 7201–7214.

[36] H. Barkhuijsen, R. de Beer, W. M. M. J. Bovee, D. van Ormondt, Retrieval of Frequencies, Amplitudes, Damping Factors, and Phases from Time-domain Signals using a Linear Least-square Procedure, *J. Magn. Reson.* **1985**, *61*, 465–481.

[37] L. Mitschang, C. Cieslar, T. A. Holak, H. Oschkinat, Application of the Karhunen-Loeve Transformation to the Suppression of Undesired Resonances in Three-dimensional NMR, *J. Magn. Reson.* **1991**, *92*, 208–217.

[38] S. Glaser, H. R. Kalbitzer, Improvement of Two-dimensional NMR Spectra by Weighted Mean t1-ridge Subtraction and Antidiagonal Reduction, *J. Magn. Reson.* **1986**, *68*, 350–354.

[39] K. Neidig, H. R. Kalbitzer, Improved Representation of Two-dimensional NMR Spectra by Local Rescaling, *J. Magn. Reson.* **1990**, *88*, 155–160.

[40] D. S. Garrett, R. Powers, A. M. Gronenborn, G. M. Clore, A Common Sense Approach to Peak Picking in Two-, Three-, and Four-dimensional Spectra using Automatic Computer Analysis of Contour Diagrams, *J. Magn. Reson.* **1991**, *95*, 214–220.

[41] S. A. Corne, P. Johnson, J. Fisher, An Artificial Neural Network for Classifying Cross Peaks in Two-dimensional NMR Spectra, *J. Magn. Reson.* **1992**, *100*, 256–266.

[42] W. Y. Choy, B. C. Sanctuary, G. Zhu, Using Neural Network Predicted Secondary Structure Information in Automatic Protein NMR Assignment, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 1086–1094.

[43] K. Huang, M. Andrec, A. Heald, P. Blake, J. H. Prestegard, Performance of a Neural-network-based Determination of Amino Acid Class and Secondary Structure from ^{1}H-^{15}N NMR Data, *J. Biomol. NMR* **1997**, *10*, 45–52.

[44] C. Bartels, M. Billeter, P. Guentert, K. Wüthrich, Automated Sequence-Specific NMR Assignment of Homologous Proteins using the Program GARANT, *J. Biomol. NMR* **1996**, *7*, 207–213.

[45] C. Bartels, P. Guentert, M. Billeter, K. Wüthrich, GARANT: A General Algorithm for Resonance Assignment of Multidimensional Nuclear Magnetic Resonance Spectra. *J. Comput. Chem.* **1997**, *18*, 139–149.

[46] D. Croft, J. Kemmink, K. Neidig, H. Oschkinat, Tools for the Automated Assignment of High-resolution Three Dimensional Protein NMR Spectra based on Pattern Recognition Techniques, *J. Biomol. NMR* **1997**, *10*, 207–219.

[47] J. A. Lukin, A. P. Gove, S. N. Talukdar, C. Ho, Automated Probabilistic Method for Assigning Backbone Resonances of (^{13}C, ^{15}N)-labeled Proteins, *J. Biomol. NMR* **1997**, *9*, 151–166.

[48] J. T. Pedersen, J. Moult, Genetic Algorithms for Protein Structure Prediction, *Curr. Opin. Struct. Biol.* **1996**, *6*, 227–231.

[49] J. Devillers (Ed.) *Principles of QSAR and Drug Design: Genetic Algorithms in Molecular Modeling*, Academic Press, New York, NY, **1996**.

[50] T. F. Havel, An Evaluation of Computational Strategies for use in the Determination of Protein Structure from Distance Restraints Obtained by Nuclear Magnetic Resonance, *Prog. Biophys. Mol. Biol.* **1991**, *56*, 43–78.

[51] A. H. C. van Kampen, L. M. C. Buydens, C. B. Lucasius, M. J. J. Blommers, Optimisation of Metric Matrix Embedding by Genetic Algorithms, *J. Biomol. NMR* **1996**, *7*, 214–224.

[52] A. H. C. van Kampen., M. L. M. Beckers, L. M. C. Buydens, A Comparative Study of the DG-OMEGA, DGII, and GAT Method for the Structure Elucidation of a Methylene-Acetal Linked Thymine Dinucleotide, *Comput. and Chem.* **1997**, *21*, 281–297.

[53] A. H. C. van Kampen, L. M. C. Buydens, The Ineffectiveness of Recombination in a Genetic Algorithm for the Structure Elucidation of a Heptapeptide in Torsion Angle Space. A Comparison to SA, *Chemom. Intell. Lab. Syst.* **1997**, *36*, 141–152.

[54] A. C. W. May, M. S. Johnson, Improved Genetic Algorithm-Based Protein Structure Comparisons: Pairwise and Multiple Superpositions, *Protein Eng.* **1995**, *8*, 873–882.

[55] A. C. W. May, M. S. Johnson, Protein Structure Comparisons Using a Combination of a Genetic Algorithm, Dynamic Programming and Least-Squares Minimisation, *Protein Eng.* **1994**, *7*, 475–485.

[56] J. V. Lehtonen, K. Denessiouk, A. C. W. May, M. S. Johnson, Finding Local Structural Similarities Among Families of Unrelated Protein Structures: A Generic Non-linear Alignment Algorithm, *Proteins: Struct. Funct. Genet.* **1999**, *34*, 341–355.

[57] A. R. Poirrette, P. J. Artymiuk, D. W. Rice, P. Willett, Comparison of Protein Surfaces Using a Genetic Algorithm, *J. Comput.-Aided Mol. Des.* **1997**, *11*, 557–569.

[58] L. Li, T. A. Darden, S. J. Freedman, B. C. Furie, B. Furie, J. D. Baleja, H. Smith, R. G. Hiskey, L. G. Pedersen, Refinement of the NMR Solution Structure of the Gamma-Carboxyglutamic Acid Domain of Coagulation Factor IX Using Molecular Dynamics Simulation with Initial Ca^{2+} Positions Determined by a Genetic Algorithm, *Biochemistry* **1997**, *36*, 2132–2138.

[59] Z. Chen, T. Blandl, M. Prorok, S. E. Warder, L. Li, Y. Zhu, L. G. Pedersen, F. Ni, F. J. Castellino, Conformational Changes in Conantokin-G Induced upon Binding of Calcium and Magnesium as Revealed by NMR Structural Analysis, *J. Biol. Chem.* **1998**, *273*, 16248–16258.

[60] S. E. Warder, M. Prorok, Z. Chen, L. Li, Y. Zhu, L. G. Pedersen, F. Ni, F. J. Castellino, The Roles of Individual Gamma-Carboxyglutamate Residues in the Solution Structure and Cation-dependent Properties of Conantokin-T, *J. Biol. Chem.* **1998**, *273*, 7512–7522.

[61] A. C. Rigby, J. D. Baleja, L. Li, L. G. Pedersen, B. C. Furie, B. Furie, Role of Gamma-Carboxyglutamic Acid in the Calcium-induced Structural Transition of Conantokin G, a Conotoxin from the Marine Snail Conus Geographus, *Biochemistry* **1997**, *36*, 15677–15684.

[62] M. J. Baylay, G. Jones, P. Willett, M. P. Williamson, GENFOLD: A Genetic Algorithm for Folding Protein Structures Using NMR Restraints, *Protein Sci.* **1998**, *7*, 491–499.

[63] M. L. M. Beckers, L. M. C. Buydens, J. A. Pikkemaat, C. Altona, Application of a Genetic Algorithm in the Conformational Analysis of Methylene-Acetal-Linked Thymine Dimers in DNA: Comparison with Distance Geometry Calculations, *J. Biomol. NMR* **1997**, *9*, 25–34.

[64] D. A. Pearlman, FINGAR: A New Genetic Algorithm-Based Method for Fitting NMR Data, *J. Biomol. NMR* **1996**, *8*, 49–66.

[65] D. A Pearlman, Automated Detection of Problem Restraints in NMR Data Sets Using the FINGAR Genetic Algorithm Method, *J. Biomol. NMR* **1999**, *13*, 325–335.

[66] C. Notredame, D. G. Higgins, SAGA: Sequence Alignment by Genetic Algorithm, *Nucleic Acids Res.* **1996**, *24*, 1515–1524.

[67] C. Notredame, L. Holm, D. G. Higgins, COFFEE: An Objective Function for Multiple Sequence Alignments, *Bioinformatics* **1998**, *14*, 407–422.

[68] R. R. Ernst, W. A. Anderson, Application of Fourier Transform Spectroscopy to Magnetic Resonance, *Rev. Sci. Instr.* **1966**, *37*, 93–102

[69] J. W. Cooley, J. W. Tukey, An Algorithm for the Machine Calculation of Complex Fourier Series, *Math. Comput.* **1965**, *19*, 297–301.

[70] R. de Beer, D. van Ormondt, Spectrum Analysis in *NMR: In-vivo Magnetic Resonance Spectroscopy 1: Probeheads and Radiofrequency Pules*, P. Diehl, E. Puck, H Günter, R. Kosfeld, J. Seeleg (Eds.), Springer-Verlag, Berlin **1992**, 26, pp. 201–258.

[71] J. C. Hoch, A. S. Stern, NMR Data Processing, Wiley-Liss, New York, **1996**.

[72] Y. Bresler, A. Macovski, Exact Maximum Likelihood Parameter Estimation of Superimposed Exponential Signals in Noise, *IEEE Trans. ASSP* **1986**, *34*, 1081–1089.

[73] R. Kumaresan, D. W. Tufts, Estimating the Parameters of Exponentially Damped Sinusoids and Pole-Zero Modeling in Noise, *IEEE Trans. ASSP* **1982**, *30*, 833–840.

[74] A. Rahman, K. B. Yu, Total Least Squares Approach for Frequency Estimation Using Linear Prediction, *IEEE Trans. ASSP* **1987**, *35*, 1440–1454.

[75] Y. B. Hua, T. K. Sarkar, Matrix Pencil Method for Estimating Parameters of Exponentially Damped/Undamped Sinusoids in Noise, *IEEE Trans. ASSP* **1990**, *38*, 814–824.

[76] G. J. Metzger, M. Patel, X. Hu, Application of Genetic Algorithms to Spectral Quantification, *J. Magn. Reson. B* **1996**, *110*, 316–320.

[77] R. Freeman, X. L. Wu, Design of Magnetic Resonance Experiments by Genetic Evolution, *J. Magn. Reson.* **1987**, *75*, 184–189.

[78] X. L. Wu, R. Freeman, Darwin's Ideas Applied to Magnetic Resonance. The Marriage Broker, *J. Magn. Reson.* **1989**, *85*, 414–420.

[79] E. Lunati, P. Cofrancesco, M. Villa, P. Marzola, A. Sbarbati, Evolution Strategy Optimization for Adiabatic Pulses in MRI, *J. Magn. Reson.* **1999**, *138*, 48–53.

[80] E. Lunati, P. Cofrancesco, M. Villa, P. Marzola, F. Osculati, Evolution Strategy Optimization for Selective Pulses in NMR, *J. Magn. Reson.* **1998**, *134*, 223–235.

[81] J. Baum, R. Tyco, A. Pines, Broadband and Adiabatic Inversion of a Two level System by Phase-modulated Pulses, *Phys. Rev.* **1985**, *A 32*, 3435–3447

[82] T. H. Reijmers, R. Wehrens, L. M. C. Buydens, Quality Criteria of Genetic Algorithms for Construction of Phylogenetic Trees, *J. Comput. Chem.* **1999**, *20*, 867–876.

[83] T. H. Reijmers, R. Wehrens, F. D. Daeyaert, P. J. Lewi, L. M. C. Buydens, Using Genetic Algorithms for the Construction of Phylogenetic Trees: Application to G-Protein Coupled Receptor Sequences, *BioSystems* **1999**, *49*, 31–43.

[84] D. L. Swofford, G. J. Olsen, Phylogenetic Inference, in D. M Hillis, C. Moritz (Eds.), *Molecular Systematics*, Sinauer Associates, Sunderland, CA, **1990**, pp. 411-501.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

11 Protein Folding

Jan T. Pedersen

Abbreviations

AM1	Austin model 1
BPTI	Bovine pancreatic trypsin inhibitor
BW	Black/White
CG	Conjugate gradient
EA	Evolutionary algorithm
GA	Genetic algorithm
GHRH	Growth hormone releasing hormone
HZ	Hydrophobic zipper
MD	Molecular dynamics
MC	Monte Carlo
MM	Molecular mechanics
NN	Neural network
OA	Orthogonal arrays
RS	Random search
SA	Simulated annealing

11.1 Introduction

Establishing the functional conformation of a protein or peptide from its primary amino acid sequence remains a central problem in biology. The most recent progress towards this goal has been in the areas of comparative modeling [1, 2] and fold recognition [3, 4]. This progress has been a consequence of the increasingly large number of experimental structures that has become available. The database of known structures is used to assist directly in the determination of the conformation of protein and peptide sequences for which there exist no experimental structure information. The classical *ab initio* protein structure prediction problem – in which only the amino acid sequence and an energy model of atomic or residual interactions are available – has been more resistant to attack by theoretical methods [5, 6].

Two predominant schools of thought exist within the field of theoretical protein folding. The first school believes that computational models of protein folding can be used to understand the basic physics and kinetics driving the natural folding process. The other is driven solely by the development of algorithms that can directly predict the experimental structure from the amino acid sequence alone. This chapter will not focus on either of these two schools of thought, but will rather focus on the use of evolutionary algorithms (EAs) to solve and understand protein folding.

In the following text, the aim of understanding protein folding is discussed in section 11.2. The peptide and energy models that have been used, together with the simulation protocols and a short overview of the spaces in which evolutionary algorithms can be applied are outlined in section 11.3. In section 11.4, the current understanding of protein folding from the perspective of evolutionary algorithms is summarized. Finally, in section 11.5, we explore how EA simulations of protein folding can be used in molecular design and drug discovery.

11.2 Searching for Functional Conformations of Peptides

The protein folding problem is traditionally divided into two subproblems: (i) the conformational search problem; and (ii) the identification of suitable discriminating functions. Arguably, there exists a solution to both of these problems. Established molecular dynamics (MD) methods, together with an all-atom force field and an explicit solvent description, are believed to reproduce reliably the motion of a polypeptide chain as a function of time [7]. If this is true, it would be possible to generate the functional conformation of a protein by starting a simulation with a random conformation and running for a sufficient length of time. The problem with MD simulations is that the time needed to search the conformational space adequately is intractable. Current MD simulation methods and state-of-the-art hardware can generate MD trajectories that represent a time window of 10^{-8} seconds. *In vitro* protein folding typically occurs over a period of 1 second to 10 minutes. Simulations can be speeded up in a number of ways: simplified representations of polypeptide chains can reduce the number of atoms that need to be handled in the simulation; larger steps in the simulation allow more conformational space to be searched with the same computational power; and novel, simplified energy models that require fewer mathematical operations allow faster evaluation of the inter- and intramolecular forces acting on each atom during the simulation. Larger steps have often been made in Monte Carlo (MC) type simulations performed in dihedral angle space rather than Cartesian space. Large trial angles are applied to individual or multiple torsion angles and the resulting energy evaluated; if the energy decreases, the new conformation is accepted. If the energy increases, there is usually some probability for accepting the new conformation based on the Metropolis test [8].

Both MD and MC simulation methods are regarded as attempts to reproduce the true physical sequence of events during folding or at equilibrium. The intractability of MD and MC simulations for larger molecular systems has led to the development of a large number of algorithms that attempt to explore the functional conformations of the polypeptide chain under the assumption that the functional conformation lies at a global energy minimum (thermodynamic hypothesis) [9, 10] or that the functional conformation is the most frequently visited conformation under native conditions (kinetic hypothesis) [11–13].

11.3 Evolutionary Algorithms

Evolutionary algorithms make use of the optimization procedures of natural gene-based evolution, namely, mutations, crossovers, and replication [14, 15], and evolve a population of species under the selection pressure of an objective evaluation function. EAs are a member of the class of co-operative search methods [16, 17] which lend themselves to distributed computing on parallel computing machines (see Chapter 12).

11.3.1 Algorithm Types in Protein Folding

Genetic algorithms (GAs), and variants thereof, are by far the most frequently used EAs in protein folding. Many types of atomic representation and simulation protocol have been developed, and these will be discussed in the following sections.

In a GA, a number of searches are performed in parallel and information is exchanged between them. It is expected that this information exchange can increase the search efficiency by a larger factor than the number of parallel processes, and this has been demonstrated in some instances [16]. GAs may be described in the following way (Fig. 1). An initial population of trial solutions is established, usually represented by strings. Mutations are introduced independently into each string; these are operations within a single search trajectory, analogous to those of a traditional Monte Carlo procedure [8]. In the canonical GA, a mutation constitutes changing a single bit in the string describing a solution. Theoretically, there is no reason why more general operators could not be used. After some number of mutations has been performed, new strings are created by crossover operations: two members of the population are selected, a break point in each of the strings is chosen, and two new population members are created by joining the left portion of one string to the right portion of the other, and vice versa. The operation of creating new strings is repeated until a new population of accepted strings is established, and then another phase of mutations is begun. This sequence of steps is repeated until the population converges to essentially a single string. A fitness function is used to assess the quality of single mutations and of new strings formed in the crossovers.

Many details have to be decided in implementing an EA. The ratio of mutations to crossovers must be optimized. The fitness function may be used to assess all mutations or just crossovers. Only changes that increase fitness can be accepted, or some fitness-decreasing changes may be allowed in a manner similar to the Metropolis test used in MC methods [8]. The selection of positions for mutations may be random or based on some measure of local fitness. Members of the population may be chosen randomly for crossover trials or selected based on their fitness. Similarly, crossover points in the string may be randomly chosen or selected based on some measure of the likelihood of success. Some members of the previous generation may be transferred directly to the next generation without crossover ("elitism"). Problems of premature convergence may be reduced by using subpopulations of strings that crossover only among themselves for an extended period of the simulation ("island models"). For protein structure prediction and simulation in particular, the nature of the strings describing the molecular conformations must also be decided. The only theories of EA behavior deal with a very simple framework

[14, 15] and in practice the optimum protocol is very problem dependent. In spite of this, EAs have proved very popular and several reviews have been written on their use in chemistry and protein folding [18–20].

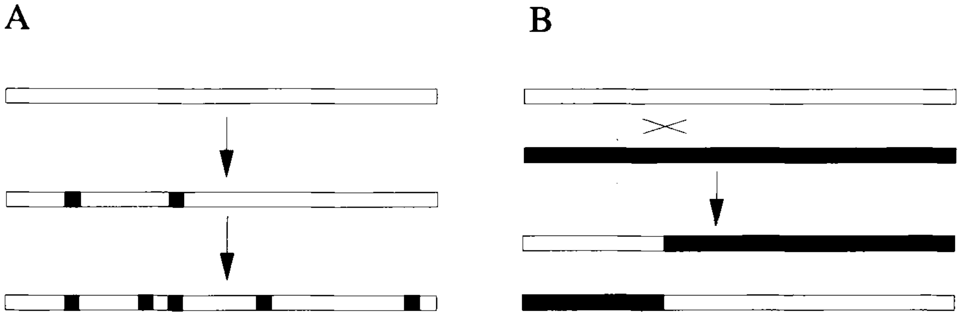


Figure 1. General outline of an evolutionary algorithm. (A) In a traditional Monte Carlo algorithm, individual local changes are made to a string of information. A fitness function is used to force changes to be made that will improve the fitness of the string. (B) In an EA, whole chunks of information are allowed to flow between members of a population. In the most trivial form one cut is made in the sequence of information, but in principle any number of chunks could be cut from one string and inserted into the other.

11.3.2 Atomic Representation

The detail of representation employed depends on the question one is interested in answering and how exhaustive a search has to be performed. The main types of representation that have been used for protein folding simulations are outlined in Figs. 2 and 3: lattice models, united-atom models, and all-atom representations. In the following sections, each of these methods will be discussed in detail.

11.3.2.1 Lattice Models

Lattice models are the simplest "toy" systems for studying protein folding, and are interesting because they lend themselves to exhaustive searching as a control. They also allow one to address some basic thermodynamic and kinetic questions [21]. Lattice models provide a useful demonstration of the potential advantages of EAs for structure simulations; however, the model is so simple that it leaves open the question of how much insight can be gained into real protein folding.

Unger and Moult [22, 23] compared the effectiveness of MC and GA searches for finding the global minimum energy on a simple two-dimensional (2-D) lattice protein model of the sort developed by Lau and Dill [24]. In these models, two types of residues (hydrophobic and hydrophilic) are scored, with an energy function that scores -1 for each pair of nonbonded hydrophobic interactions. In this study, chain lengths of 20 and 64 residues

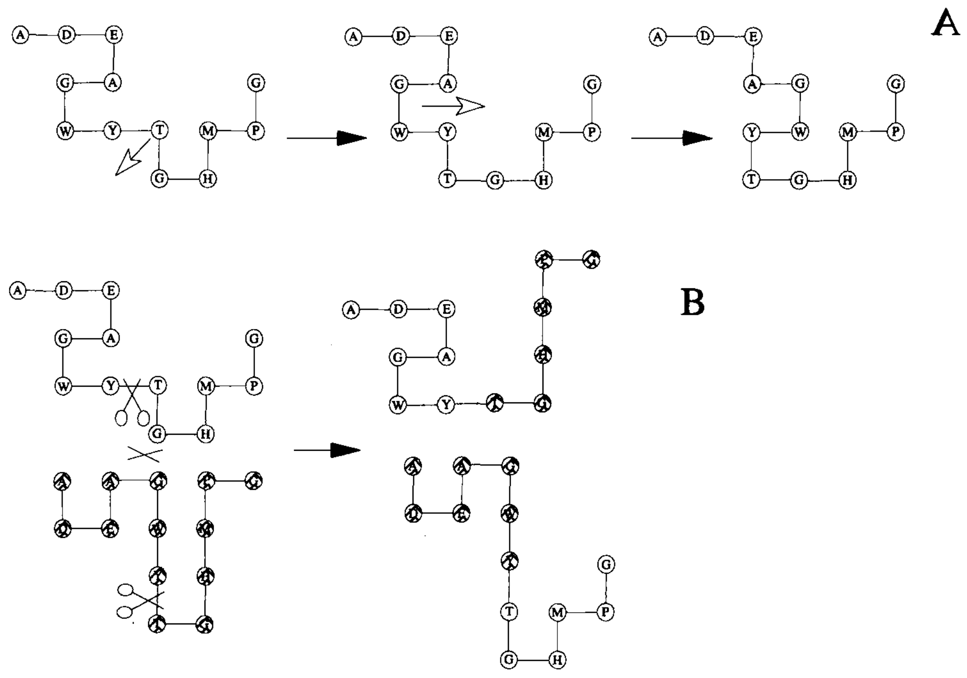


Figure 2. Two- and three-dimensional "bead" or "lattice" models of polypeptide chains. (A) A typical Monte Carlo sequence for simulation on a lattice. Many types of move have been developed. Here, a corner-flip is shown first followed by a camshaft move. (B) The evolutionary simulation of a population of conformations. Two conformations are selected at random from a population and each of the two population member's are cut at the same point in the chain. The two halves are then swapped with the other two halves and the two chains are glued together to generate two new members of the next generation. In this example, the chains are joined in the same orientation as existed in the parent structures. In more advanced lattice EAs, an orientational search is performed before the two halves are joined.

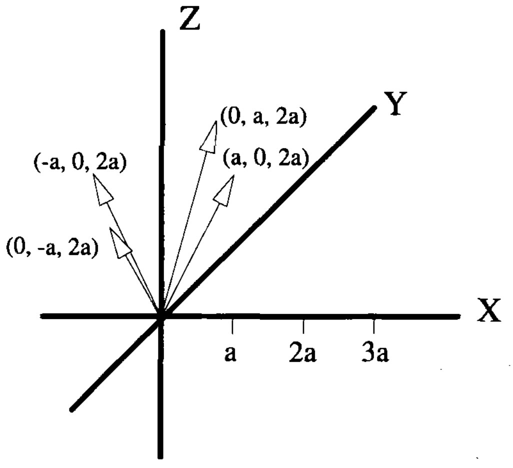


Figure 3. A 210 lattice. One residue at the origin (0,0,0) is connected with a consecutive *neighbor in the peptide chain by selecting* one of 24 neighbors (23 when in the middle of the chain). The four possible lattice points in the +z direction are shown, equivalent positions in +x, -x, +y, -y, -z direction exist.

were used. Three types of MC methods were included in an attempt to provide a fair basis for comparison. A population size of 200 was used in the GA. A string of bond angles along the chain was used to describe the trial conformations (Fig. 2). MC steps and GA crossovers were implemented as randomly chosen changes to a randomly selected bond angle (Fig. 2). Crossover sites were also selected randomly. Under these conditions, the GA is significantly more effective than the traditional MC search. For the shortest sequences, both methods find the global minimum, but the GA requires one or two orders of magnitude fewer energy evaluations. For the longer sequences, the MC algorithm did not find the global minimum in the available CPU time. In all but one intentionally difficult case, the GA was successful. The probable reason for the better performance of the GA is its ability to find accepted moves once the chain has adopted a compact conformation whereas most MC moves for a compact chain will be rejected.

Lattice models are also well suited to explore the basic properties of search algorithms. Judson et al. [25] have explored GA simulation methods together with other traditional search methods such as simulated annealing (SA), conjugate gradient (CG) minimization and random search (RS). Simulations were performed on 19-, 37- and 61-atom linear polymers on a 2-D triangular lattice using a simple Lennard-Jones potential to evaluate atomic interactions. This work confirmed the earlier findings of Unger and Moult that GAs are able to find lower energy conformations than traditional linear search methods. For these simple model systems, an SA protocol coupled with a conjugate gradient search performs almost as well as the GA. The work by Judson et al. also provides an excellent introduction to the basic properties and parameters of GAs.

König and Dandekar [26] have used a simple lattice model described by Lau and Dill [24] to explore how to widen the basic EA search. They found that the use of systematic crossovers between individuals which differ in more than four bits increases the convergence rate for shorter chains. In this search, the probability for generating new, previously unvisited conformations is increased by only making crossovers between "diverse" parents.

GAs have been applied to a lattice model which attempts to reproduce real peptide geometries by Sun *et al.* [27]. The 210-type lattice model (Fig. 3) [28] represents each amino acid by a single point. Neighboring residues in the sequence are restricted to positions in space which are a combination of $(0, \pm a, \pm 2a)$ along the three principal directions in space. In this case, a is the lattice unit dimension. The lattice model is then described by the virtual $C\alpha$-$C\alpha$-$C\alpha$ bond angles (ϕ) and virtual dihedral angles (ψ) $C\alpha$-$C\alpha$-$C\alpha$-$C\alpha$ which form the information string (chromosome) that is used for GA simulations. Sun et al. [27] use this projected lattice model to explore a parameter optimization technique based on orthogonal arrays (OA). This method allows the identification of optimal simulation parameters through iterative sampling of finer and finer sets of limited combinations of parameters. A set of parameters derived from a simulation on crambin was applied to a simulation of cytochrome B562. Although the study is quite limited and leaves open the question of how problem-dependent parameters may be, it warrants further investigation.

11.3.2.2 United-Atom Models

United-atom models move away from a lattice and into dihedral angle space. In these representations, the side chain of the polypeptide chain is typically replaced by one or two atoms at the center of the all-atom side chain. The dihedral angles of the side chain are then selected from a small library of preferred conformations in order to reduce the computational resources required.

Sun et al. [29] used a description of a protein molecule that consisted of a full backbone together with one virtual atom per side chain. A potential of mean force derived from known protein structures was used to assess the fitness of trial conformations. An additional constraint was the experimental radius of gyration. A library of peptide fragment conformations two to five residues long, constructed from known protein structures, was used to construct initial conformations and to perform mutational changes. A population size of 90 was used. Low final root mean square deviations from the experimental structure were reported. The significance of these results is difficult to assess, as the fragments were selected from the library on the basis of sequence similarity and the library contained the two larger structures that were successfully reproduced. This would presumably introduce a strong bias towards the expected structure. Nonetheless, the method presented is interesting and worthy of further study.

Another united-atom model is the Cα structure representation used by Bowie et al. [30]. Small fragments of proteins were constructed using a fold recognition algorithm. Nine-residue segments were selected from a library of fragment conformations on the basis of their environment codes [30]. A similar method was used for some larger fragments (15–25 residues). Homologous structures were carefully removed from the database. The method of selecting initial conformations did enhance the local structure accuracy to a value higher than that expected by chance alone. Structures were then improved by a GA procedure in which each gene represented the set of dihedral angles of a structure and mutations made changes to one angle. For recombination, segments of one gene were replaced with segments of another. Mutations had a high probability of occurring at the fragment junctions. The fitness was evaluated with a function containing contributions from the fold recognition profile fit, hydrophobicity, accessible surface area, atomic overlap and "sphericalness" of the structure. The weighting of the terms was strongly biased by the experimental structure. Under these conditions, some native-like structures were efficiently generated, along with competing low-energy, but incorrect, structures.

Dandekar and Argos [31] used Cα backbone models of proteins in real space. A simplified bit-string encoding of ϕ/ψ space, allowing each ϕ/ψ pair to adopt one of seven possible angle combinations was used [32]. Each side chain was represented by a sphere of 1.9 Å and the global energy or fitness function consisted of hydrogen bonding, secondary structure preference and a hydrophobic scatter term, each of which were scaled according to heuristic constants. The method generally relied on the correct preassignment of secondary structure for success.

An interesting multilevel simulation method has been presented by Standley et al. [33]. In this method, the peptide folding process is first modeled by a Monte Carlo simulation using a coarse representation of the amino acid chain consisting of only one Cα-Cβ atom pair per residue. The sidechain is not included explicitly but is represented by a effective radius in the potential at the position of the Cβ atom, which is dependent on the specific nonbonded residue residue interactions. A reduced angle set of 18 selected ϕ/ψ pairs from the allowed regions of the Ramachandran map is used. The combined MC/GA simulation is performed on a cylinder sphere representation that is defined by local hydrophobicity calculated using the fold recognition potential of Casari and Sippl [34]. In the MC simulation whole chunks (loops) of structure are swapped between a precalculated library of loop conformations and the current structure. This reduction of chain complexity dramatically reduces the degrees of freedom for the polypeptide chain and increases simulation speed. All energy evaluations in this first step are performed on the Cα-Cβ representation.

Coarse models from the above simulations are then converted into a more detailed model in which each residue is described by six atoms: N, H, Cα, Cβ, C and O. The variables are ϕ/ψ angles which are allowed to assume 532 states. The loop library from above is replaced by a presorted library of three-residue fragments. The fitness function is changed from a fold-recognition potential to a combination of the fold-recognition potential and three additional physical terms: a torsional term, a hydrogen bonding term, and a disulphide bonding constraint. This simulation protocol has been tested on two small α/β-proteins BPTI and 1CTF. The method appears to be able to reproduce the secondary structure and is also able to generate topologies which are comparable to the native conformation. The generation of three-dimensional topologies is likely to be driven mainly by the disulphide constraints and the many local secondary structure constraints. There are three disulphide bridges in the native conformation of BPTI and none in 1CTF.

11.3.2.3 All-Atom Models

Only a few full-atom conformational search algorithms for polypeptides have been described in the literature, the main limitation being the calculation of n-squared (N^2) interactions for larger systems.

Pedersen and Moult [35, 36] have explored a GA designed to predict the structure of small fragments (12–22 residues long) of proteins. The simulation method is an extension of an earlier torsion space MC method [37]. In the GA method (Fig. 4), only fragments that are expected to have their conformation determined independently from the rest of the structure were selected [38]. A full heavy atom and polar hydrogen representation of the polypeptide backbone and side chains was used. Conformations with excessive steric overlap were rejected. Fitness was evaluated using a potential based on point-charge electrostatics and accessible surface area. Terms in the force field were parameterized using a potential of mean force analysis of experimental structures. A gene comprised a string of ϕ/ψ and χ angles representing a conformation. No mutation steps were used. Crossover points were weighted towards positions in the polypeptide chain that varied the most in the current population. An extensive annealing of side chain conformations was performed at crossover points before evaluating the fitness of the new gene. The population size was 200–300, and the GA was run for 40–50 generations. The parameters of the algorithm were optimized systematically using a set of fragments of known structure. Searches were performed in parallel on clusters of workstations or on parallel-architecture computing servers. Experience with this procedure shows it to be substantially more effective than the MC procedure at generating low-energy structures [36].

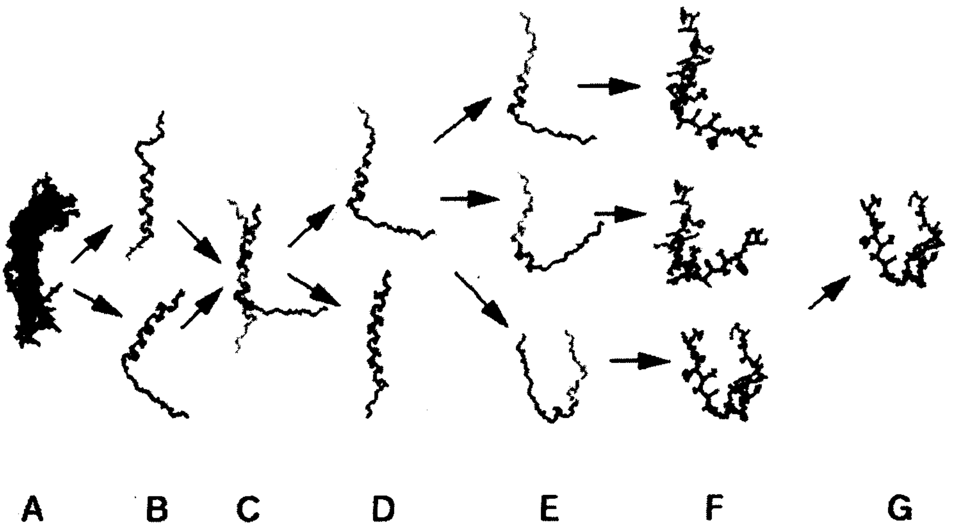


Figure 4. General scheme for an all-atom model EA simulation. (A) A population of structures is generated. (B) A pair of random or biased conformations are selected from the population. (C) One or more crossover points are selected in the backbone. (D) The two or more halves are swapped. (E) A joint search is performed to generate a pool of children. (F) Side chains are generated on all members of the crossover pool and structures are energy minimized. (G) Lowest energy conformation(s) are selected as crossover products for the next generation.

Cui et al. [39] have presented a GA using a model which is based on a traditional molecular mechanics (MM) united-atom model. This model includes all dihedral angles and only uses the united-atom model for energy evaluation. The simulation method is virtually identical to that presented by Pedersen and Moult [40]. The simulation is biased by a reliable neural network secondary structure prediction method which works from the primary sequence of a given protein. The EA simulation chromosome consists of ϕ, ψ, χ_1, χ_2,... values for each residue. Initial and mutation ϕ/ψ backbone angle pairs are selected from areas of the Ramachandran map that are restricted by the secondary structure prediction, that is, if helix is predicted for a particular residue position ϕ/ψ is chosen randomly such that: $-75 < \phi <-55$, $-50 < \psi < -30$. This technique dramatically reduces the search space for the polypeptide chain. Side chain conformations are selected from the Ponder and Richards side chain library [41, 42].

The forcefield of Cui et al. [39] consists of only two terms: a hydrophobic interaction term that is calculated as the nonpolar solvent accessible surface area and a truncated Lennard-Jones term that penalizes overlapping atoms. The truncated Lennard-Jones term allows a considerable atom-atom overlap and thereby allows atoms to pass through each other during the simulation. This simplistic potential is able to generate some quite impressive topologies of five small model proteins between 46 and 120 residues in length. The fact that this simulation protocol is able to identify lower energy structures, compared to the "native" conformation for two of the five structures indicates that the potential may not fully reflect the forces that drive natural protein folding.

Herrmann and Suhai [19, 43] have performed extensive searches on small di- and tripeptides using the AM1 forcefield. For these small model systems, an exhaustive search can be performed, identifying all available low-energy conformations by enumeration. In this algorithm, changes are made directly to the dihedral angles stored in the Z-matrix, which contains all internal parameters of the molecular system that is being simulated. The global minimum energy conformations were identified for a number of peptides up to four residues in length. The method was not tested on larger systems.

Jin et al. performed a more detailed study on the small penta-peptide circulatory hormone [Met]-enkephalin [44]. In this study, random ϕ/ψ and χ angles were assigned to an initial population of structures. An all-atom representation together with the ECEPP/2 forcefield was used. This study tested a GA on a realistic peptide system and gave similar results to those reported by Herrmann and Suhai [19, 43].

Tuffery et al. have applied GA simulations to the problem of finding the correct set of side chain rotamers for a protein, given the experimental backbone conformation [45, 46]. In these simulations, the gene is a string of side chain rotamer angles (χ-angles) for a complete protein. The half of the current gene pool with the lowest energy is carried forward to the next generation. The other half of the population is replaced by mutation and crossover operations. Mutations cause changes in the angle of a single rotamer and the mutation rate is decreased exponentially during the run. Crossovers either transfer a randomly chosen contiguous segment of rotamer values between genes, or a selected subset from the whole protein. In practice, this method was found to be effective, but not as efficient as an alternative EA procedure. Several methods have been reported in the literature that can successfully identify side chain rotamer solutions given the exact backbone conformation. If side chains are built on an approximate backbone conformation, much less success is observed [1, 2, 47, 48].

Ring and Cohen have used a GA to construct initial conformations for loop regions in proteins, using four possible conformations for each residue [49]. In an eight-residue loop, a population of 30 initial genes consisting of bit strings describing trial conformations was used. Mutations consisted of single bit changes. This simulation converged after only eight generations and had to be coupled with extensive minimization in order to fit the chain ends to the protein scaffold.

11.3.3 Simulation Protocols

The choice of optimal simulation parameters for an EA is very problem dependent. Nearly all publications on EA methods for protein folding simulations use a different algorithm and molecular representation, making it difficult to extract any general information about optimal simulation parameters such as population size, crossover frequency, mutation rate, etcetera. For example, it is not necessary to consider side chain conformations in 2-D or 3-D lattice simulations, whereas all-atom representations may need extensive local mutations or annealing in order to accommodate chain-ends and side chain packing after crossovers. The number of generations that has to be simulated is highly dependent on how recombination is performed. If elitism is used, where the best conformations are always carried over to the next generation, structural convergence is likely to happen more quickly than otherwise. Some of the most commonly used parameter values for a selection of the GA simulation protocols reported in the literature are listed in Table 1.

Table 1. Some basic parameters for protein folding EAs reported in the literature. A typical population size is 100–1000 and the number of generations is on the order of 50–1000. (*) indicates that this number has been estimated from other data in the listed reference.

Reference	Model	Population	Generations	Chain length
[25]	Triangular 2-D lattice	50	–	19–61
[22]	2-D-lattice	200	300	20–64
[26]	2-D-lattice	100–800	–	20–85
[50]	(2,1,0)-lattice	50	20–50	46
[27]	(2,1,0)-lattice	1000–3000	500–2000(*)	46
[31]	N, $C\alpha$, C, O	632	632	46–128
[29]	N, $C\alpha$, $C\beta$ C, O	200	–	30–144
[39]	United-atom/All-atom	500	> 100	46–70
[43]	All-atom	100–500	200	2–4
[36]	All-atom	200	40–60	10–14

These parameters should only be used as a crude guideline for the optimization of new methods. The most important parameter to be monitored is the frequency of successful crossovers (also known as the survival rate) which indicates if the simulation has converged or not. This parameter is only seldomly reported.

11.3.4 Recombination Spaces

Previously, EAs in protein folding have only been applied at the structural level; that is the information string that has been used has encoded the conformation of a peptide chain. In principle, an EA is a general optimization procedure that could work in any parameter space. This section will discuss some of the less conventional applications of EAs in the protein folding field.

Typically, recombination in protein folding simulations is performed by simple cutting and pasting of structural fragments. This frequently leads to poor geometries. Rabow and Sheraga have addressed this problem elegantly by using Cartesian combination operators as a method for blending two conformations together [50]. This is done simply by applying a linear combination to the Cartesian co-ordinates of the two parents: q^{child} $(\gamma, q^A, q^B) = \gamma\ (q^B - q^A) + q^A$. Linear combinations are applied to a 210-type lattice model of the polypeptide. The linear combination can only be applied to fairly similar conformations and will tend to generate children containing distorted bond angles and bond lengths. Therefore, it has to be combined with a re-fitting of the children onto the lattice using a short minimization before the next round of crossovers. The advantage of the linear combination method is that it preserves many of the long-distance interactions observed in the parent structures.

Sun et al. show that one can efficiently explore the thermodynamic minima of a model function using a GA or a GA/SA hybrid method [51]. In these simulations, the information exchanged consists of thermodynamic parameters or simple function parameters $x1$, $x2$ and $f\ (x1, x2)$. Although this method is not applied directly to peptide systems, it suggests alternative simulation methods [51]. For example, it is possible to envisage methods where the forcefield or scoring function is self adjusting, changing the parameters so as to optimally fit the state of the folding simulation.

11.4 Understanding Protein Folding through EA

As well as being an efficient method for solving a global minimum problem, EAs can be used as a tool to provide insight into the protein folding problem. This section will review some of these applications.

11.4.1 Lattice Models

Current thermodynamic models of protein folding are driven by simple 2-D and 3-D lattice model simulations of BW (Black/White)-type polymers [52, 53]. BW polymers are simple bead models where a scoring function favors black-black or white-white interactions. The observation from these types of simulation is that the native folded state is well separated from all other conformations by a significant energy gap. Folding in these models happens by a hydrophobic zipper (HZ) mechanism in which the local structure is formed first, generating hydrophobic contacts between local residues [54]. This formation of local structure is followed by an aggregation of local structure into domains and finalized by a condensation of loops and termini on the surface.

The factors that govern the speed of folding in lattice models have recently been investigated by Dinner et al. [55] using a GA coupled with a neural network (NN). This study concluded that there is probably not any unifying feature that characterizes fast folding sequences, but it did identify an additional force in the HZ model which is important for fast folding. This was that early contacts are not only made between residues that are close in sequence but also between residues that are "topologically close".

As mentioned earlier, the validity of translating results obtained by lattice simulations to the folding of real proteins is unclear. However, the existence of nucleation sites that consistently form early in the simulations, together with stable cooperative structure is in concordance with experimental findings and studies of more detailed models (see [55] for a discussion of the relationship of lattice simulations to experiments).

11.4.2 Detailed Models

Rose [11, 12, 56] has proposed that the natural process of protein folding is hierarchical and predominantly kinetically driven. In other words, the native conformation of a protein is not necessarily a global free energy minimum, but rather the most easily accessible energy minimum, or the most frequently visited energy minimum. If the folding process is hierarchical, this implies that, early in the folding process, local interactions play a dominant role. Initially, the chain folds up locally to form supersecondary structures that act as early folding units [38] around which the rest of the structure condenses. Rose and Srinivasan [57] have evolved this concept to produce a simulation algorithm which reflects the theory of hierarchical folding. Local structure is evolved first by only calculating interactions between residues that are close in sequence. Once the local structure has formed, it is frozen and not changed for the rest of the simulation. The linear interaction distance is gradually increased during the simulation. This algorithm is able to generate near-native local super-secondary structure, to some extent confirming the validity of the hierarchical hypothesis.

Recent experimental studies on protein folding have partly confirmed that, in at least some cases, the folded state may be characterized by a kinetic trap [13, 58], where a significant energy barrier locks the native state in its folded conformation. Based on experimental evidence and a theory of protein folding, Baker and coworkers [59–61] have developed a simulation strategy which is essentially an evolutionary method. Local structure is evolved independently of the rest of the structure by the identification of local sequence-structure similarities to a library of local structure fragments. A complete predicted structure is then built by identifying overlapping structural fragments that contain local sequence homology to the target. This method is able to generate near-native super-secondary structure fragments equivalent to the method of Rose discussed previously [59, 60].

The use of detailed models for the understanding of protein folding is evolving rapidly and the studies of Baker and Rose show much promise for the elucidation of protein folding mechanisms.

11.5 The Application of Evolutionary Protein Folding Algorithms in Molecular Design

Protein-folding simulations are currently only generating crude models of peptide fragments and protein topologies, but some applications of these in molecular design have been presented in the literature. These indicate some of the possible uses EA folding simulations may have in the future:

- Mapping of conformational space for small peptides and the use of generated structural ensembles to design peptidomimetics.
- Protein design or the re-design of sequences with a known fold.
- Engineering of protein physico-chemical properties to design mutants that have increased thermostability, increased rate of folding, etc.

Often in structure-based drug design [62], the structure of a rigid, experimentally determined binding site is used for the design of specific ligands. This design process is frequently aided by the knowledge of the structure of a natural ligand. Although this method has been proven in some cases, it only represents a simplistic scenario of ligand-receptor interactions. While the receptor binding site can in some cases be approximated by a static model, the ligand is most frequently a small highly flexible molecule and understanding the correct binding thermodynamics and kinetics requires a detailed knowledge of the free-energy distribution of ligand conformations [63]. EAs may be able to solve this problem by providing equilibrium distributions of ligand conformations in the bound and unbound state.

Pedersen and coworkers [64, 65] have performed extensive GA simulations of small 20- to 40- residue peptides corresponding to the sequence of various incretine peptide hormones (GRF-like family). Simulations on a 29-residue peptide hormone corresponding to the sequence of growth hormone releasing hormone (GHRH) were used to design peptidomimetics of the GHRH molecule [64]. A structurally conserved four- to five-residue region of the GHRH peptide was observed in the simulated population of structures. This conserved region was used to construct small organic molecules which retained GHRH activity.

Jones [66] has used a GA simulation method to optimize the sequences of a number of designed protein structures using a fold recognition pairwise potential as the fitness function [67]. In this work the structure of the peptide chain is kept fixed and the sequence is altered through crossovers and mutations within a population of 500 random sequences. It is possible for this algorithm to find sequences which are considerably more compatible with the structure than the native sequence. None of these sequences has been synthesized to confirm the designs; however, the method has been applied to the Paracelsus challenge [67, 68] which asks the question "Is it possible for an all-α protein to have more than 50 % sequence homology to an all-β protein?".

This type of design algorithm may prove valuable for the design of mutants that aim at altering the physico-chemical properties (such as thermostability) of a given protein.

11.6 Conclusions

Evolutionary algorithms have turned out to be well suited for protein folding simulations and should continue to provide useful insights into the protein folding problem. EAs efficiently search a large, complex, conformational space. In addition, EAs are trivially parallelizable and, for that reason, a prime choice for implementation on emerging parallel computer hardware platforms.

Simple models allow for the exhaustive exploration of simple polymer systems and may potentially answer general questions about the kinetic and thermodynamic behavior of polymer molecules. More detailed models may be used to answer questions about the conformational behavior of small peptides.

From a practical perspective, the examples above outline some of the uses of protein folding EAs for molecular design. For example, the exhaustive conformational search of small peptides allowing the exploration of the thermodynamic distribution function and the design of protein structures and protein mutants.

As our ability to model protein folding accurately increases, it is to be expected that such simulations, including those based on EAs, will play an increasing role in "real world" molecular design situations.

References

[1] S. Mosimann, R. Meleshko, M. N. G. James, A Critical Assessment of Comparative Molecular Modeling of Tertiary Structures of Proteins, *Proteins: Struct., Funct., Genet.* **1995**, *23*, 301–317.

[2] A. C. R. Martin, M. W. MacArthur, J. M. Thornton, Assessment of Comparative Modeling in CASP2, *Proteins: Struct., Funct., Genet.* **1997**, *Suppl. 1*, 14–28.

[3] C. M. R. Lemer, M. J. Rooman, S. J. Wodak, Protein Structure Prediction by Threading Methods: Evaluation of Current Techniques, *Proteins: Struct., Funct., Genet.* **1995**, *23*, 337–355.

[4] A. Marchler-Bauer, M. Levitt, S. H. Bryant, A Retrospective Analysis of CASP2 Threading Predictions, *Proteins: Struct., Funct., Genet.* **1997**, *Suppl. 1*, 83–91.

[5] T. Defay, F. E. Cohen, Evaluation of Current Techniques for *ab initio* Protein Structure Prediction, *Proteins: Struct., Funct., Genet.* **1995**, *23*, 431–445.

[6] A. M. Lesk, CASP2: Report on *ab initio* Predictions, *Proteins: Struct., Funct., Genet.* **1997**, *Suppl. 1*, 151-166.

[7] E. M. Storch, V. Daggett, Molecular Dynamics Simulation of Cytochrome B5: Implications for Protein-Protein Recognition, *Biochemistry* **1995**, *34*, 9682-9693.

[8] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, Equation of State Calculations by Fast Computing Machines, *J. Chem. Phys.* **1953**, *21*, 1087–1091.

[9] P. L. Privalov, Stability of Proteins: Small Globular Proteins, *Adv. Protein Chem.* **1979**, *33*, 167–236.

[10] C. B. Anfinsen, Principles that Govern the Folding of Protein Chains, *Science* **1973**, *181*, 223–230.

[11] R. L. Baldwin, G. D. Rose, Is Protein Hierarchic? I. Local Structure and Peptide Folding, *Trends Biochem. Sci.* **1999**, *24*, 26–33.

[12] R. L. Baldwin, G. D. Rose, Is Protein Hierarchic? II. Folding Intermediates and Transition States, *Trends Biochem. Sci.* **1999**, *24*, 77–83.

[13] D. Baker, Metastable States and Folding Free Energy Barriers, *Nature Struct. Biol.* **1998**, *5*, 1021–1024.

[14] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, San Mateo, CA, **1989.**

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, **1975.**

[16] S. H. Clearwater, B. A. Huberman, T. Hogg, Cooperative Solution of Constraint Satisfaction Problems, *Science* **1991**, *254*. 1181–1183.

[17] B. A. Huberman, The Performance of Cooperative Processes, *Physica* **1990**, *D 42*, 38–47.

[18] R. S. Judson, Genetic Algorithms and Their Use in Chemistry, *Rev. Comput. Chem.* **1997**, *10*, 1–73.

[19] F. Herrmann, S. Suhai, Genetic Algorithms in Protein Structure Prediction, in S. Suhai, (Ed.), *Computational Methods in Genome Research*, Plenum Press, New York, **1994**, pp. 173–190.

[20] S. M. Le-Grand, K. M. Merz, The Genetic Algorithm and Protein Tertiary Structure Prediction in K. M. Merz (Ed.), *The Protein Folding Problem and Tertiary Structure Prediction*, Birkhauser, Boston, **1994**, pp. 109-124.

[21] E. I. Shakhnovich, Theoretical Studies of Protein-folding Thermodynamics and Kinetics, *Curr. Opin. Sruct. Biol.* **1997**, *7*, 29–40.

[22] R. Unger, J. Moult, Genetic Algorithms for Protein Folding Simulations, *J. Mol. Biol.* **1993**, *231*, 75–81.

[23] R. Unger, J. Moult, Effect of Mutations on the Performance of Genetic Algorithms Suitable for Protein Folding Simulations, *Computer Aided Innovation of New Materials*, Elsevier Science Publishers B. V., North Holland, **1993**, *II*, 1283–1286.

[24] K. F. Lau, K. A. Dill, Theory for Protein Mutability and Biogenesis, *Proc. Natl. Acad. Sci. USA* **1990**, *87*, 638–642.

[25] R. S. Judson, M. E. Colvin, J. C. Meza, A. Huffer, D. Gutierrez, Do Intelligent Configuration Search Techniques Outperform Random Search for Large Molecules, *Int. J. Quantum Chem.* **1992**, *44*, 277–290.

[26] R. König, T. Dandekar, Improving Genetic Algorithms for Protein Folding Simulations by Systematic Crossover, *BioSystems* **1999**, *50*, 17–25.

[27] Z. Sun, X. Xia, Q. Guo, D. Xu, Protein Structure Prediction in a 210-type Lattice Model: Parameter Optimization in the Genetic Algorithm using Orthogonal Arrays, *J. Protein Chem.* **1998**, *18*, 39–46.

[28] J. Skolnick, A. Kolinski, Dynamic Monte Carlo Simulations of a New Lattice Model of Globular Protein Folding, Structure and Dynamics, *J. Mol. Biol.* **1991**, *221*, 499–531.

[29] S. Sun, P. D. Thomas, K. A. Dill, Simple Protein Folding Algorithm using a Binary Code and Secondary Structure Constraints, *Protein Eng.* **1995**, *8*, 769–778.

[30] J. U. Bowie, R. Luthy, D. Eisenberg, A Method to Identify Protein Sequences that Fold into a Known Three-dimensional Structure, *Science* **1991**, *253*, 164–170.

[31] T. Dandekar, P. Argos, Folding the Main-chain of Small Proteins with the Genetic Algorithm, *J. Mol. Biol.* **1994**, *236*, 844–861.

[32] M. J. Rooman, J. P. A. Kocher, S. J. Wodak, Prediction of Protein Backbone Conformation based on Seven Structural Assignments, *J. Mol. Biol.* **1991**, *221*, 961–979.

[33] D. M. Stanley, J. R. Gunn, R. A. Friesner, A. E. McDermott, Tertiary Structure Prediction of Mixed α/β Proteins via Energy Minimisation, *Proteins: Struct., Funct., Genet.* **1998**, *33*, 240–252.

[34] G. Casari, M. J. Sippl, Structure-derived Hydrophobic Potential: Hydrophobic Potential Derived from X-ray Structures of Globular Proteins is able to Identify Native Folds, *J. Mol. Biol.* **1992**, *224*, 725–732.

[35] J. T. Pedersen, J. Moult, *Ab initio* Structure Prediction for Small Polypeptides and Protein Fragments using Genetic Algorithms, *Proteins: Struct,, Funct., Genet.* **1995**, *23*, 454–460.

[36] J. T. Pedersen, J. Moult, Protein Folding Simulations with Genetic Algorithms and a Detailed Molecular Description, *J. Mol. Biol.* **1997**, *269*, 249–269.

[37] F. Avbelj, J. Moult, Determination of the Conformation of Folding Initiation Sites in Proteins by Computer Simulation, *Proteins: Struct., Funct., Genet.* **1995**, *23*, 129–141.

[38] J. Moult, R. Unger, An Analysis of Protein Folding Pathways, *Biochemistry* **1991**, *30*, 3816–3824.

[39] Y. Cui, R. S. Chen, W. H. Wong, Protein Folding Simulation with Genetic Algorithm and Supersecondary Structure Constraints, *Proteins: Struct., Funct., Genet.* **1998**, *31*, 247–257.

[40] J. T. Pedersen, J. Moult, *Ab initio* Protein Folding Simulations with Genetic Algorithms: Simulations on the Complete Sequence of Small Proteins, *Proteins: Struct., Funct., Genet.* **1997**, *supp. 1*, 179–184.

[41] J. W. Ponder, F. M. Richards, Internal Packing and Protein Structural Classes, *Cold Spring Harbor Symp. Quant. Biochem.* **1987**, *52*, 421–428.

[42] J. W. Ponder, F. M. Richards, Tertiary Templates for Proteins. Use of Packing Criteria in the Enumeration of Allowed Sequences for Different Structural Classes, *J. Mol. Biol.* **1987**, *193*, 775–791.

[43] F. Herrmann, S. Suhai, Minimization of Peptide Analogues using Genetic Algorithms, *J. Comput. Chem.* **1995**, *16*, 1434–1444.

[44] A. Y. Jin, F. Y. Leung, D. F. Weaver, Development of a Novel Genetic Algorithm Search Method (GAP1.0) for Exploring Peptide Conformational Space, *J. Comput. Chem.* **1997**, *18*, 1971–1984.

[45] P. Tuffery, C. Etchebest, S. Hazout, R. Lavery, A New Approach to the Rapid Determination of Protein Side-chain Conformations, *J. Biomol. Struct. Dyn.* **1991**, *8*, 1267–1289.

[46] P. Tuffery, C. Etchebest, S. Hazout, R. Lavery, A Critical Comparison of Search Algorithms Applied to the Optimization of Protein Sidechain Conformations, *J. Comput. Chem.* **1993**, *14*, 790–798.

[47] S. Y. Chung, S. Subbiah, How Similar Must a Template Structure be for Homology Modeling by Sidechain Packing Methods? *Pac. Symp. Biocomput.* **1996**, pp. 126–141.

[48] R. Samudrala, J. Moult, Determinants of Side-chain Conformational Preferences in Protein Structures, *Protein Eng.* **1998**, *11*, 991–997.

[49] C. S. Ring, F. E. Cohen, Conformational Sampling of Loop Structures using Genetic Algorithms, *Isr. J. Chem.* **1994**, *34*, 245–252.

[50] A. A. Rabow, H. A. Sheraga, Improved Genetic Algorithm for the Protein Folding Problem by Cartesian Combination Operator, *Protein Sci.* **1996**, *5*, 1800–1815.

[51] R. L. Sun, J. E. Dayhoff, W. A. Weigand, *A Population-based Search from Genetic Algorithms through Thermodynamic Operations*, Techniccal Research Report, ISR (Institute for Systems Research), University of Maryland, **1994.**

[52] A. Sali, E. Shakhnovich, M. Karplus, How does a Protein Fold? *Nature* **1994**, *369*, 248–251.

[53] A. Sali, E. Shakhnovich, M. Karplus, Kinetics of Protein Folding. A Lattice Model Study of the Requirements for Folding to the Native State, *J. Mol. Biol.* **1994**, *235*, 1614–1636.

[54] K. A. Dill, K. M. Fiebig, H. S. Chan, Cooperativity in Protein Folding Kinetics, *Proc. Natl. Acad. Sci. USA* **1993**, *90*, 1942–1946.

[55] A. R. Dinner, S. S. So, M. Karplus, Use of Quantitative Structure-Property Relationships to Predict the Folding Ability of Model Proteins, *Proteins: Struct., Funct., Genet.* **1998**, *33*, 177–203.

[56] G. D. Rose, Hierarchic Organization of Domains in Globular Proteins, *J. Mol. Biol.* **1979**, *134*, 447–470.

[57] R. Srinivasan, G. D. Rose, LINUS: A Hierarchic Procedure to Predict the Fold of a Protein, *Proteins: Struct., Funct., Genet.* **1995**, *22*, 81–99.

[58] E. Alm, D. Baker, Matching Theory and Experiment in Protein Folding, *Curr. Opin. Struct. Biol.* **1999**, *9*, 189–196.

[59] C. Bystroff, D. Baker, Local Structure Prediction using a Library of Sequence-Structure Motifs, *J. Mol. Biol.* **1998**, *281*, 565–577

[60] K. T. Simons, I. Ruczinski, C. Kooperberg, B. Fox, C. Bystroff, D. Baker, Improved Recognition of Native-like Protein Structures using a Combination of Sequence-dependent and Sequence-independent Features of Proteins, *Proteins: Struct., Funct., Genet.* **1999**, *34*, 82–95.

[61] K. T. Simons, R. Bonneau, I. Ruczinski, D. Baker, *Ab initio* Protein Structure Prediction of CASP III Targets using ROSETTA, *Proteins: Struct., Funct., Genet.* **1999**, *Suppl. 3*, 171–176.

[62] L. M Amzel, Structure-based Drug Design, *Curr. Opin. Biotechnol.* **1998**, *9*, 366–409.

[63] M. K. Gilson, J. A. Given, B. L. Bush, J. A. McCammon, The Statistical-thermodynamic Basis for Computation of Binding Affinities: A Critical Review, *Biophys. J.* **1997**, *72*, 1047–1069.

[64] P. H. Andersen, L. O. Gerlach, B. S. Hansen, L. Helmgaard, T. Andreasen, A. Hansen, O. Olsen, P. Gaudreau, J. T. Pedersen, Pharmacology, Functionality Dynamics and Structure Activity relations of a G Protein-coupled Receptor Illustrated with the Growth Hormone Rreleasing Hormone (GHRH) Receptor and GHRH(1-29)-NH2 in S. Frøkjær, L. Christrup, P. Krogsgaard-Larsen (Eds), *Peptide and Protein Drug Delivery*, Alfred Benzon Symposium 43, Munksgaard, Copenhagen, Denmark, **1998**, pp. 50–60.

[65] J. T. Pedersen, *Ab initio* Structure Prediction for Small Peptides of the Glucagon Hormone Family, using Torsion Space Monte Carlo and Genetic Algorithms, *ISMB* **1996**, Presented at ISMB 1995, Cambridge, UK, September 12–16, text can be requested from author.

[66] D. T. Jones, De Novo Protein Design using Pairwise Potentials and a Genetic Algorithm, *Protein Sci.* **1994**, *3*, 567–574.

[67] D. T. Jones, C. M. Moody, J. Uppenbrink, J. H. Viles, P. M. Doyle, C. J. Harris, L. H. Pearl, P. J. Sadler, J. M. Thornton, Towards Meeting the Paracelsus Challenge: The Design, Synthesis, and Characterisation of Paracelin-43, an alpha-Helical Protein with over 50 % Sequence Identity to an All-beta Protein, *Proteins: Struct., Funct., Genet.* **1996**, *24*, 502–513.

[68] G. D. Rose, Protein Folding and the Paracelsus Challenge, *Nature Struct. Biol.* **1997**, *4*, 512–514.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

12 New Techniques and Future Directions

Andrew Tuson and David E. Clark

Abbreviations

AI	Artificial intelligence
CAMD	Computer-aided molecular design
EA	Evolutionary algorithm
EP	Evolutionary programming
ES	Evolution strategy
GA	Genetic algorithm
GP	Genetic programming
HIV	Human immunodeficiency virus
HTS	High-throughput screening
KBS	Knowledge-based systems
KNN	K-nearest neighbors
LGA	Lamarckian genetic algorithm
MCS	Maximal common substructure
NFL	No free lunch
NMR	Nuclear magnetic resonance
PC	Personal computer
RMS	Root mean square
RNA	Ribose nucleic acid
SA	Simulated annealing
TS	Tabu search
VR	Virtual reality

12.1 Introduction

It should be clear from the preceding chapters in this book that techniques based upon evolutionary algorithms (EAs) have been applied to many, if not the majority, of the tasks that comprise the discipline of computer-aided molecular design (CAMD). The burgeoning interest in EAs has been remarkable: Milne estimated that between 1989 and 1992 there were only five chemically oriented research articles that employed EAs; between 1993 and 1996, however, that figure mushroomed to 210 [1]. The number has continued to increase at a steady rate, prompting several review articles that survey this growing area of research [2, 3]. The early implementations of EAs, generally genetic algorithms

(GAs), in CAMD tended to be fairly simple, canonical forms, often inspired by pseudocodes from popular textbooks (e.g., [4]). However, in recent years, there has been a trend towards increasing sophistication and the adoption of what might be considered more advanced techniques.

It is the purpose of this chapter to highlight some of these newer EA techniques that are finding their way into CAMD applications and also to introduce some of the wider aspects of EA research that could be of value to the CAMD community. We shall also provide an overview of some possible alternative methods that could be put at the disposal of computational chemists and discuss how these methods can be applied to CAMD, what opportunities are available, and what barriers to their wider adoption need to be overcome. In addition, we seek to look forward to see what other new approaches may be forthcoming from the EA community.

12.2 Basic EA Theory

One of the active research areas within the evolutionary computation community continues to be the search for a more secure theoretical framework from which to derive guidance for those applying and developing EAs and upon which future research could be based. The problem is succinctly stated by Bäck et al.: 'We know that [evolutionary algorithms] work, but we do not know why' [5]. In practical terms, the use of current EA theory remains at the level of justifying useful metaphors that assist in the design of EAs. An example is the "building block hypothesis" [4] that suggests the EA designer should identify high-quality components (building blocks) that can be usefully combined, then design crossover operators that allow this to happen. At present, given that a *quantitative and predictive* theory for EA search dynamics is sought, the direct impact of current EA theory on CAMD practice may well be limited. This is for three reasons. First, the search space would need to be characterized, and this may require more search points to be sampled than just running the EA would involve. Second, the mathematical effort involved may be too great to be justified for complex real problems. Finally, an end-user would be more likely to adopt a system that can be justified in terms of the problem being solved (which they understand) rather than an abstract mathematical theory (which they might not).

It would seem for the present time that the application and tuning of EAs is likely to guided more by empiricism than theory. As Fogel notes, "the hope of improving evolutionary algorithms in general function optimization on the basis of mathematical theory appears dim" [6].

12.3 Domain-Specific Representation and Operators

For any evolutionary algorithm, the representation (or encoding) of the individuals comprising the evolving solution population and the set of operators used to generate offspring are perhaps the two most important components of the system, in many cases being crucial to the success of the algorithm. Clearly, there is a strong link between the representation chosen and the operators that can be employed (or vice versa) and so considerations about one will inevitably affect the choices made about the other [7]. Traditionally, EAs have represented the population members by binary strings (GAs) or vectors of real values [evolutionary programming (EP), evolution strategies (ESs)], and the archetypal variational operators have been developed with these in mind. However, in more recent times, there has been much enthusiasm for the employment of more "natural", or problem-specific, representations (and thus, operators). In general, the guiding principle today seems to be "mold the algorithm to the problem, not the problem to the algorithm" [8–10].

There are at least two reasons for this. First, as EAs have been applied to a wider range of problems, it has been increasingly found that many do not map well into a binary string- or vector-based representation. Thus, workers have been driven to experiment with more natural representations and operators and have often found success by doing so. Second, and more fundamentally, the "no free lunch" (NFL) theorems of Wolpert and Macready [11] state broadly that, for any algorithm, any elevated performance on one class of problems is exactly countered by poorer performance on another class. Thus, an EA tailored for a specific application by the incorporation of problem-specific knowledge in the encoding and operators is likely to outperform a canonical, "black-box" implementation.

There are a number of examples from the molecular design literature that illustrate the points made above. In developing a GA for protein folding simulations, Moult and Unger opted not to use any form of encoding of the population members but rather to allow genetic operators to act directly upon the conformations of the protein model [12]. Several workers in the field of *de novo* molecular design have adopted a similar philosophy, having the evolving molecular structures as the population members and developing crossover and mutation operators to suit the problem representation [13–16]. For instance, in the work of Glen and Payne [14], two different types of crossover and no fewer than 12 different mutation operators were employed for the manipulation and variation of molecular structures.

12.3.1 How to do it? Principled EA Design

Given the above, and the success of EAs for many CAMD applications, the question of how to design an effective EA is nontrivial. Thus, design remains an *ad hoc* process partly, but not wholly, due the lack of an effective theory. In this context, a *principled* design approach for EAs is required if they are to reach their full potential. One recent approach [17] is based on the idea of viewing EAs as *knowledge-based systems* (KBS) – "a computer system that represents and uses knowledge to carry out a task" [18]. Furthermore, given that the artificial intelligence (AI) community has, for many years, been addressing the problems of how to design KBS, if EAs could be placed in a KBS framework, then it may be possible to exploit the extensive research already carried out into KBS design.

One such useful concept from the KBS literature is the distinction made between the *knowledge level* (the description of *what* domain knowledge is present in the system), and the *symbol level* (the description of *how* the knowledge is represented in the system – the data structures used and the operations on them) [19]. The design of EAs is, at present, conducted primarily at the symbol level, and therefore the knowledge/domain assumptions made by the EA designer are implicit and may well not be properly utilized. A knowledge level analysis would make these explicit and ensure that they are fully considered in the EA design. To this end, Tuson [17] proposes that the domain knowledge used by an EA can play three roles:

- *Problem-solving knowledge* – this assists the search by providing *problem-specific* knowledge that structures and guides the search.
- *Problem specification (goal) knowledge* – specifying the characteristics of desirable solution(s) (i.e., the evaluation function).
- *Search control knowledge* – given a defined search space, how do we go about searching it? Any knowledge of the search process is represented here.

Given that we would be interested in exploiting chemical knowledge, it would appear that the problem-solving knowledge role is most relevant. However, exactly what is meant by this? A working definition is that it is all of the knowledge that can be directly related to the problem itself, and which is not involved with specifying the quality of a solution. This allows for a clean separation from the technique-specific aspects of the search algorithm. Once these roles have been defined, then there are various types of *knowledge sources* available that need to be identified. Some example EA problem-solving knowledge sources are outlined below:

- Problem features that correlate with solution quality (i.e., the decision variables);
- How these features interact (because strongly interacting decision variables could be usefully considered as a single unit);
- Areas of the search space which could be excluded from the search (to reduce the size of the search space);
- Areas of the search space where good solutions lie (so that an initialization strategy could be used to start the search in a potentially productive region).

Some reflection should suggest to the reader that the above sources can be framed as questions that a chemist would be able to answer in his/her role as a domain expert without necessarily having any expertise in optimization. In addition, Tuson [17] shows that the above sources can be formalized in such a way that suitable EA crossover and mutation operators can be derived. Therefore, it appears possible that future EA design will center on the modeling and acquisition of expert knowledge rather than algorithm design, in much the same way that the KBS community has been advocating in recent years [20]. Furthermore, KBS methods such as those used for knowledge acquisition should allow the EA designer to extract the required knowledge more effectively. Given the need for more effective solutions to CAMD problems, such principled design approaches should help ensure the scalability of EA methods in the future.

12.4 Self-Adaptation

One of the major difficulties in using an EA is in deciding on the settings for the (many) adjustable parameters that are associated with this class of algorithm. Thus, another key area for evolutionary computation research identified in [5, 6] is that of *self-adaptation*. An excellent review of the issue of parameter control in EAs has recently been published [21].

Most research on self-adaptation has emerged from the ES and EP communities, and has thus been primarily concerned with the mutation operator, which is the main variational operator for these two classes of EA. The motivation for developing self-adaptive techniques was the realization that, given a population of real-valued vectors, the optimization performance of the algorithm can be improved by applying perturbations of different magnitude to each of the variables in the vector, in other words, each dimension of the search. This is particularly so when the variables have different units of dimension, for example, pressure and temperature [22]. To try to determine *a priori* optimal values for these mutational step sizes is virtually impossible, especially as the optimal values may alter during the course of a search as the algorithm traverses the fitness landscape. The two main approaches towards self-adaptation seek to counter this problem by allowing the mutational step sizes to adapt themselves as the evolutionary search progresses.

12.4.1 Co-evolutionary Approaches

The most commonly used method of self-adaptation is simply to encode the EA parameters of interest (termed *strategy parameters*), such as the size of the mutation step made, into the encoded solution. These can then be operated on and allowed to *co-evolve* (hopefully) effective values alongside the solution's decision variables. This approach was proposed independently by the ES [23] and EP [24] communities. The driving force for developing these techniques was the observation that, given a population of competing solutions, the solutions with strategy parameters that produce children with increased fitness will have a better chance of having their material (including the choice of strategy parameters) passed on to later generations. This is a consequence of the fitter children having a "head start" on the children of the other solutions with less suitable strategy parameters. This results in an implicit selection pressure upon the strategy parameters that should drive them towards appropriate values during the EA run.

The ES variant of co-evolutionary self-adaptation is the more commonly used because it has demonstrated a generally superior optimization performance across a number of test functions [25]. The method of Schwefel [23] is as follows: each of the real-valued vectors of variables x comprising the population is given an accompanying vector of strategy parameters, σ, where $\sigma(\mathrm{i})$ denotes the standard deviation to use when applying a zero-mean Gaussian mutation to component $x(\mathrm{i})$ of the parent vector. Then:

$$\sigma'(\mathrm{i}) = \sigma(\mathrm{i}) \exp(\tau_0 \mathrm{N}(0,1) + \tau \mathrm{N}(\mathrm{i})(0,1)) \qquad (1)$$

and

$$x'(\mathrm{i}) = x(\mathrm{i}) + \mathrm{N}(0,\sigma'(\mathrm{i})) \qquad (2)$$

where $\tau = 1/[2(n^{1/2})]^{1/2}$, $\tau_0 = 1/(2n)^{1/2}$, N(0,1) is a standard Gaussian random variable sampled once for all n dimensions, and N(i)(0,1) is a standard Gaussian random variable sampled anew for each of the n dimensions. Thus, under the Schwefel scheme, at each generation, the strategy parameters for the individual are mutated and the new values are used to generate the offspring (the "sigma-first" method [26]).

As a final note it should be noted that this approach does not only apply to real coded problems. For instance, one could exchange the standard deviation of a Gaussian distribution for the bit-wise mutation probability if a binary-encoding was to be used. However, the most successful applications of this approach have been for real-coded problems.

12.4.2 Learning Rules

The other method for on-line adaptation of EA parameters is to utilize explicit *learning rules* that change EA parameters according to some measure of operator quality. For instance, given an EA with two operators (i.e., crossover and mutation), the performance of each operator could be measured in terms of the improvements in solution quality that it produces when applied (termed *operator productivity*). A simple example of a learning rule that exploits this is COBRA (Cost Operator-Based Rate Adaptation) which was developed by Corne and coworkers [27]. This assigns different initial probabilities to each operator and periodically reassigns the "bag" of operator probability values by ranking the operators in order of their recent operator productivity, giving the highest probability to the operator with the highest productivity. This straightforward method has been shown to be effective for real-world timetabling problems.

More sophisticated learning-rule adaptation methods exist which allow finer adjustments of operator probabilities and incorporate performance metrics that augment operator productivity to reward operators that increase the effectiveness of other operators that are applied later (e.g., when mutation maintains diversity so that crossover can be effective). An early example of this approach was the "1/5 rule" that emerged from the evolution strategies community [28]. Other instances have been devised by Davis [29] and Julstrom [30].

12.4.3 CAMD Applications

Two groups of CAMD researchers have experimented with self-adaptive mutation in the context of using EP algorithms for the conformationally flexible docking of ligands to proteins. In the development of their AGDOCK program [26, 31–33], Gehlhaar and coworkers have used both the sigma-first method outlined above and a variation they term "sigma-last" in which the parent σ values are first used to create offspring and then mutated. They found that the sigma-first method was superior, probably because the offspring positions in the search space are determined by the offspring strategy parameters. This avoids the situation that can occur with the sigma-last method whereby offspring may be generated that have useful position vectors but poor strategy parameters [26]. Westhead et al. [34] also employed a sigma-first method but in their work found that better results were obtained using Cauchy, rather than Gaussian, mutation to perturb the x vectors in Eq. (2) – a tactic suggested by Yao and coworkers [35, 36]. More recently, the Agouron group have also applied their self-adaptive EP methods to the problem of solving X-ray crystal structures by molecular replacement in a program called EPMR (Evolutionary Programming for Molecular Replacement) [37].

Other workers have experimented with using an annealing mutation operator to control the mutation rate during RNA folding simulations [38]. The annealing mutation operator permits a large number of mutations early in the search, but the mutation rate is decreased hyperbolically as the search progresses towards convergence. While this does not constitute self-adaptation as such, it is similar in spirit.

12.4.4 Meta-EAs

It is also worth briefly mentioning an alternative approach to the problem of setting parameter values – that of *meta-evolution.* In meta-evolution, one EA controls a population of other EAs, each of which is initiated with different parameter settings. As this population of algorithms operates on the problem in hand, the ones showing better performance have a higher probability of surviving and spawning new "child" EAs. In this way, good problem-solving strategies can be spread to the whole population of EAs. This type of method has been pioneered in programs such as DAGA-2 [39]. Such an approach also lends itself naturally to parallel architectures, of which more will be said in a later section.

12.4.5 No Magic Bullet!

Since the NFL theorem notes that no optimizer is more effective than another when considered over all problems, this implies that self-adaptive EAs cannot be a general solution to the tuning problem. A corollary of this is that the designer is in fact making assumptions that, if valid, lead to improved performance. Work in [40] investigated self-adaptation methods and found that the assumptions made were as follows:

- The performance metric used must suggest the correct adaptation decisions, for example, operator productivity arguments may not suggest the right action if the search is being hampered by a lack of diversity.
- EA performance must be related in a clear fashion to the EA strategy parameters being adapted (otherwise how would the mechanism be able to suggest a useful adaptation?).
- Given that the above are satisfied, the gains achievable by a good adaptation must outweigh the costs involved in performing the adaptation and acquiring the required information.

These assumptions apply to all self-adaptation methods, including the co-evolutionary approaches (as they implicitly process operator productivity information). Furthermore, Tuson and Ross [40] identified a number of problems for which adaptation mechanisms fail, due to one or more of the above assumptions being invalid. However, on a more positive note, self-adaptation has proved to be a useful addition for a number of applications and should definitely be considered. Even failure of the mechanism should, with consideration of the above assumptions, allow the designer to understand better the nature of the problem at hand.

12.5 What *is* a Good Solution?

As noted earlier (section 12.3.1), the fitness function is itself a source of domain knowledge as it can be regarded as a specification of what is a desirable solution. However, for real-world problems it is not necessarily straightforward to set out *a priori* what constitutes a desirable solution! Instances of how this can occur will now be discussed, and interactivity will be shown to be a possible solution to this issue, as well as to the more general problem of how to exploit the CAMD practitioner's knowledge to the full.

12.5.1 Constrained and Multi-Objective Problems

One issue that has conveniently been ignored so far is that many CAMD problems do not have just a single objective. In reality, there are often constraints upon what makes a feasible solution, as well as a number of objective functions to optimize. These objectives are not always mutually optimizable because a gain in one objective may lead to a reduction in another. Unfortunately, the trade-off can be difficult to represent *a priori*, as the designer often does not know what this will be at the outset (or at best will have a very vague idea).

Therefore, EA methods that can handle both constraints and multiple objectives are needed in CAMD applications. The handling of constraints can be dealt with in a number of ways such as penalizing infeasible solutions, or designing operators that only generate feasible solutions – these options are discussed in some length in [41]. The handling of multiple objectives has been dealt with by considering the idea of *Pareto optimality*. A solution is termed *Pareto optimal* if it is *nondominating* with respect to the other solutions in the search space; that is, no solutions exist that have better values for one or more objectives and at least equivalent (if not better) values for the remainder (Fig. 1).

The domination condition can of course be expressed as a comparison function for tournament/ranking selection in an EA; that is, solution A is better than B if A dominates B. Owing to its population-based nature, an EA can be used to find a *number* of high-quality nondominating solutions that represent different trade-off decisions. The user can then be given these solutions and left to decide which offers the most appropriate trade-off, thus avoiding the problem of how to specify the trade-off between competing objectives. This approach can be implemented in a number of ways, and the reader is directed to [42, 43] for overviews on the subject.

In the field of CAMD, a good example of a multi-objective optimization problem is the search for the 3-D maximum common substructure (MCS) among two or more molecular structures [44]. In this type of search, two generally contradictory parameters contribute to the fitness function: the size of the MCS, and the closeness of the geometric fit. In other words, it is generally observed that as the size of the proposed MCS (expressed as

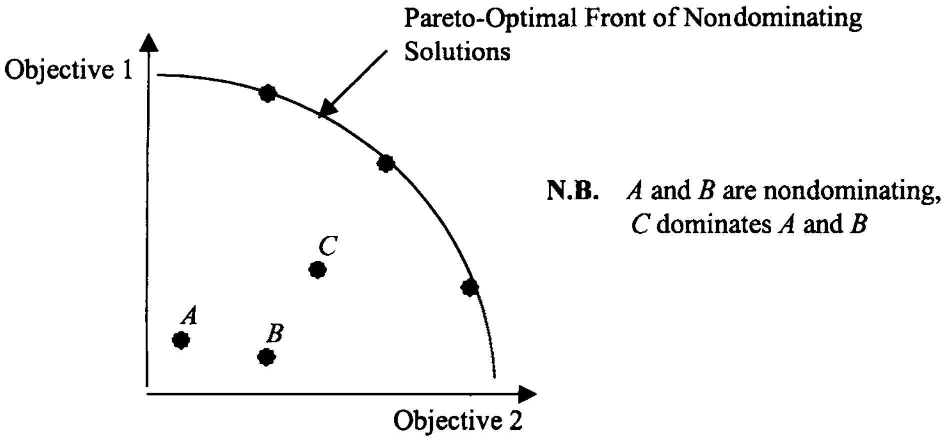

Pareto-Optimal Front of Nondominating Solutions
Objective 1
Objective 2
A
B
C
N.B. A and B are nondominating, C dominates A and B

Figure 1. Illustration of Pareto optimality.

the number of atoms it comprises) increases, the quality of the geometric fit between the two structures (expressed as RMS distance in Å) decreases. The challenge is to find an optimum that accounts for both criteria. Consequently, Handschuh et al. [44] employed Pareto optimality in their GA fitness function for molecular MCS determination and superposition. In this way, they were able to select good results from the resulting Pareto sets of solutions.

12.5.2 Interactive EAs

One approach to multi-objective problems is to introduce some interactive component into the algorithm. This approach arises from the fact that the domain expert's knowledge can often be difficult to formulate exactly; a situation that can also occur elsewhere in the EA design. For example, constraints are rarely set in stone. More often than not, they can be relaxed under certain conditions. Again, the circumstances under which this is possible are often very difficult to define, or of such a nature that a computer system could not be allowed to make the decision autonomously. In addition, the CAMD practitioner may also wish to impose additional constraints on the problem in order to make the search more tractable – effectively making additional hypotheses during the EA run about where good solutions may lie which can be dynamically revised during the optimization process.

Therefore, interaction represents a potentially useful mechanism for fully exploiting the CAMD practitioner's rich, although possibly implicit, knowledge of the problem domain to improve the EA's effectiveness. Interaction also brings an additional advantage: human users are very reluctant to adopt decisions that they have not had a role in making. In many cases, this can lead to solutions being modified by hand or being ignored, both of which defeat the point of using an EA optimizer. In this context, *interactive EAs* have become the center of some interest of late. One instance of this is the work of Biles [45].

This approach takes a "user-as-fitness-function" approach that substitutes the fitness function for a user-assigned score in an admittedly *ad hoc* manner. Though this is in the spirit of what we are trying to achieve, Tuson et al. [46] raise a number of objections. First of all, most realistic optimization problems will require on the order of thousands of function evaluations and this is clearly infeasible for a human expert to perform. For instance, Gehlhaar et al. typically use 70 000 fitness function evaluations for each docking run with AGDOCK [31]. In addition to this scalability problem, Biles' approach ignores the fact that it is usually possible to construct a reasonably close model of what a good solution is, thereby reducing the human effort required. Finally, even a partial attempt at formalizing the goal knowledge can be a useful exercise in promoting a better understanding of the problem being solved. Therefore, a future issue in using interactive EAs for real-world problems such as CAMD will be to address scalability problems such as those above. In fact, preliminary work on this has begun which attempts to integrate, in a principled manner, the benefits of interactivity with those of an automated fitness function [46].

Interestingly, attempts have been made to implement interactive EAs in a CAMD setting. The LeapFrog *de novo* molecular design system has an interactive mode that allows the user to interact with the evolutionary algorithm-based molecular design process [47]. Another interactive program is STALK [48], a virtual reality (VR)-based system for protein-ligand docking. Here, the user can influence the docking process interactively using VR technology.

12.6 Parallel Algorithms

In the execution of an EA, it is invariably the fitness evaluation step that is the most computationally expensive. However, the fact that this step is decoupled from the rest of the algorithm makes EAs highly amenable to parallelization, creating a *parallel* EA. Some of the various kinds of parallel EA have been classified by Cantu-Paz [49]. For the purposes of this chapter, two types of parallel EA are of particular interest.

The first type is what Cantu-Paz terms "coarse-grained" parallelism. In this formalism, the population is divided into a small number of subpopulations that are kept relatively isolated from one another, evolving on separate processors or machines. The optional introduction of a *migration* operator permits the exchange of individuals between subpopulations at specified intervals. At least two models of coarse-grained parallelism are possible: the first is termed the "island model" and the second is known as the "stepping stone" or "ring" model (Fig. 2). Both partition the population into subpopulations in the same way, the difference between them being that, while the island model allows migration between any two subpopulations, the ring model allows migration only between neighboring subpopulations [49]. When using coarse-grained parallelism, there is an additional benefit on top of the speed-up gained from distributing the fitness calculations over a number of processors. The partitioning of the population into subpopulations may also help to maintain genetic diversity in the population as a whole. This can help search efficiency by ensuring good sampling of the search space and may also help prevent premature convergence to suboptimal solutions. This benefit of coarse-grained parallelism can be taken advantage of even on serial processors, as will be demonstrated below. A thorough discussion of island models can be found in [50]. The second type of parallelism is "fine-grained" parallelism. In this, the population is subdivided into a large number of very small subpopulations. In the limiting (and ideal) case, each individual is assigned to its own processing element in a massively parallel computer [49].

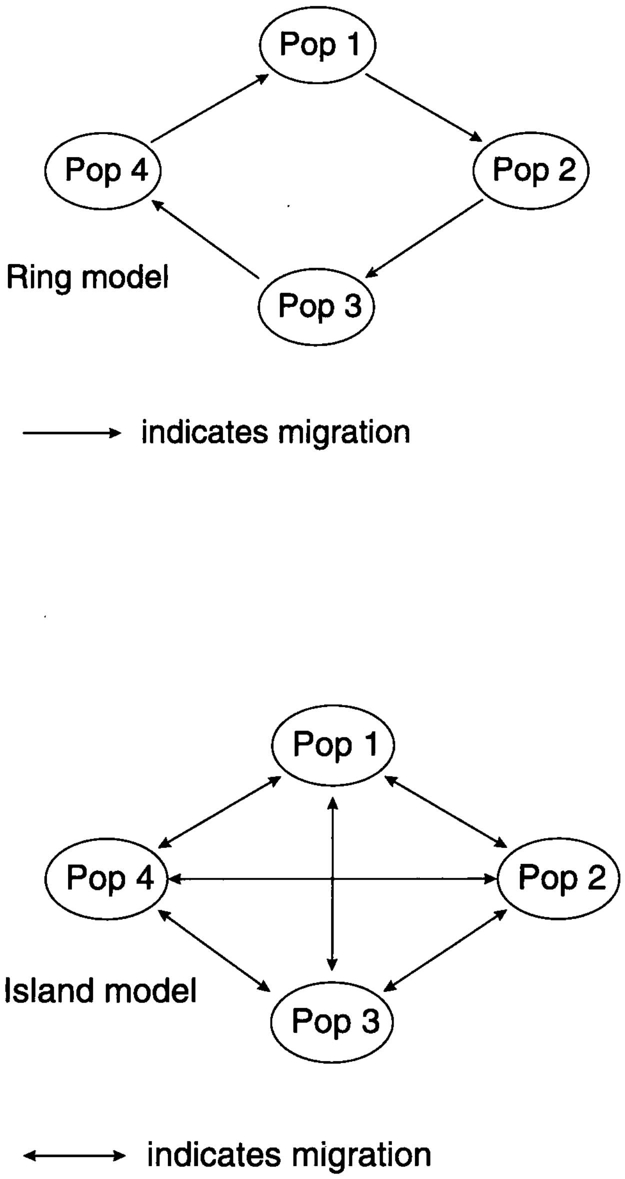


Figure 2. "Ring" (or "stepping-stone") and "island" models for parallel evolutionary algorithms (EAs).

Both types of parallelism have been experimented with in CAMD applications. The island model has been successfully employed by Jones and coworkers in developing genetic algorithms for protein-ligand docking [51] and molecular superposition and pharmacophore identification [52]. These algorithms were run on a serial machine with the subpopulations residing on the same processor. Nevertheless, in experiments comparing the use of a single population of 500 individuals to the use of five subpopulations of 100 individuals, it was found that the island model gave equivalent results in slightly shorter run times [51]. This would seem to indicate that the maintenance of genetic diversity through distributed populations can aid search efficiency as well as the quality of the final solution. Beckers et al. obtained a similar result using a stepping stone model in their GA for structure determination from NMR spectra [53]. Their parallel GA was run over a local area network of workstations and significant speed-ups compared to a sequential implementation were observed arising from both the parallelization of the fitness evaluation and the fact that the parallel runs required fewer fitness evaluations to reach convergence (indicating a more efficient search). A stepping stone model was also used by Del Carpio in work on protein folding in which a network of five transputers was employed [54]. A variant of an island model has been developed by Notredame et al. in their PRAGA (Parallel RNA Alignment Genetic Algorithm) program [55]. PRAGA's parallelization is based upon a three-branched tree with three levels. Using this arrangement, it was possible to achieve 80 % of the maximum theoretical speed-up while obtaining a final solution as good as that obtained from a single population of equal size to the sum of the 13 subpopulations used in the parallel implementation. Other parallel GAs for docking [48], identifying regions of local structural similarity in proteins [56], and for conformational searching [57] have also been reported.

There have been fewer applications of massively parallel evolutionary algorithms in CAMD, presumably because of a lack of suitable machines. However, Shapiro and Wu have developed a GA for RNA folding predictions that runs on a MasPar MP-2 16384-processor machine [58]. This GA begins by initializing a population of simple RNA structures, one for each processor. At each generation, for each processor, the GA selects two parents from the structures stored on the processor in question and its eight neighbors. This step takes advantage of the eight-way interconnected mesh structure of the MasPar machine. Mutation and crossover operations are then performed to generate two child structures, and the best of these replaces the current structure for the particular processor. All these operations take place in parallel, generating 16384 new structures at each generation. Finally, Wild and Willett discuss a number of different models of parallelism in the context of a GA for similarity searching in databases of 3-D chemical structures [59]. In their work, a Kendall Square Research KSR-1 machine with 64 processors was used and a simple strategy of assigning single molecules to single processors was found to be the most successful.

Parallel computation is rapidly becoming more popular, as it is increasingly recognized that many organizations possess large numbers of quite powerful PCs or workstations, many of which are idle for long periods. By employing distributed computing methods, it is possible to utilize this resource. Another driving factor is that of cost: it is nowadays often cheaper to buy several high-end PCs than one workstation. By using the PCs in a parallel fashion, excellent performance can be obtained [60]. For these reasons, it is likely that there will continue to be much interest in parallel EAs and many more applications of them in CAMD.

12.7 Hybrid Algorithms

While EAs can often yield excellent results when applied alone to a problem, additional benefits can often be derived from their combination with other computational methods. There is often considerable synergy to be exploited between EAs and optimization methods from mathematical programming, greedy or local search algorithms or other heuristic search algorithms [such as tabu search (TS) and simulated annealing (SA)] [61]. EAs have also been successfully hybridized with neural and fuzzy computing methods [62, 63].

Given that EAs are often configured as global optimization methods, one of the most common hybridizations is with a local optimization method aimed at refining the solutions from the EA. This has been frequently employed in CAMD applications. For instance, the Powell minimization algorithm [64] has been employed to refine the final solutions generated by evolutionary docking algorithms [32, 34]. Hibbert has also reported the beneficial combination of a GA with a steepest descent minimizer for the estimation of kinetic parameters [65]. In studies of the conformational search of large molecules, McGarrah and Judson found that frequent gradient optimization of the conformational energy during the course of a GA search gave a marked advantage over methods in which the energy was optimized by the GA alone [66]. Similar results have been obtained in protein folding experiments [54]. By contrast, in studies on molecules having smaller conformational search spaces, Judson et al. found that a GA was capable of locating good solutions without the computationally expensive optimization step [67]. GAs in which the population members generated by the genetic operators are replaced by the results of the local optimizer before undergoing selection and reproduction have been termed "Lamarckian" GAs (LGAs). This is a reference to Jean Batiste de Lamarck's discredited assertion that phenotypic characteristics acquired during an individual's lifetime can become heritable traits. Morris et al. have described such a GA for protein-ligand docking [68]. In their experiments, they found that the LGA outperformed both simulated annealing and a traditional GA. The conformational search procedure reported by Frey would also appear to be Lamarckian in nature [69].

As detailed in Chapter 5, neural networks have been usefully combined with evolutionary algorithms for the generation of quantitative structure-activity relationships (QSARs). The most advanced example of this is the Genetic Neural Networks method of So and Karplus [70–73]. Other work in this area is reported in [74]. Also in the QSAR field, GAs have been hybridized with quadratic partial least squares (QPLS) to provide a method with significant improvements over the conventional QPLS approach [75].

A K-nearest-neighbors (KNN) classification algorithm has been hybridized with a GA to form the CONSOLV program developed by Raymer et al. [76]. This program derives and applies rules to decide whether a given water molecule in a protein active site is likely to be involved in or displaced by ligand binding. The same group of workers have also experimented with a KNN-genetic programming hybrid and found it to give superior results to the GA variant [77].

There have been a number of accounts in CAMD of the hybridization of an EA with another heuristic search algorithm. Gunn has described the combination of a GA with Monte Carlo SA for protein folding simulations and suggested that the efficiency of the resulting hierarchical algorithm exceeded what would be expected from either of the components used independently [78]. Several groups of workers have reported that, in the context of ligand-protein docking, such hybrid algorithms (GA/TS [34, 79], EP/SA [80] and GA/SA [81]) can yield superior performance to either of the algorithms used in isolation. Similarly, Zacharias et al. [82] have developed a combined SA/GA method for optimizing the geometry of silicon clusters which is reported to outperform SA or GA alone by an order of magnitude, in terms of the CPU time required to attain convergence. An SA/GA hybrid (termed an "annealing evolutionary algorithm") has also been reported by Cai et al. for spectral fitting [83]. Finally, Waller and Bradley have implemented a novel variable selection procedure that combines elements of evolutionary algorithms with Monte Carlo and tabu search methods [84].

As with self-adaptation (section 12.4), despite their successful applications, hybrid methods cannot be considered a universal solution to the design of effective EA-based CAMD solutions. Consider two techniques (an EA and another method) each with their own repertoire of behaviors. Combining these behaviors (which is in effect what hybridization does) allows the EA designer access to more behaviors, thus making it more likely that the desired one is present. Unfortunately, this has the drawback of making the task of selecting (i.e., tuning) the correct behavior more difficult, as there are now more choices to consider. Therefore, although hybridization is undoubtedly useful, it is advised that EA designers structure their experimentation by formulating some rationale for why it would be useful for the problem *before* the EA-hybrid is built and run.

12.8 New Algorithms

2.8.1 Genetic Programming

One subarea of evolutionary computation that is worth mentioning separately is *genetic programming* (GP) [85]. Although the original idea behind GP was to evolve computer programs, for our purposes it can best be thought of as the use of an EA to optimize mathematical functions represented as trees. The leaf/terminal nodes indicate variables and constants that are successively operated on by function nodes on their way to the root node, giving the final result. The choice of the terminal and function nodes is user (and also problem) dependent.

GP can potentially be used for the same applications as a neural network, for example, regression and classification. However, GP does have one distinct advantage. If the function nodes are chosen well, the trees evolved by GP can be readily interpretable. One notable example was the use of GP, with a set of function nodes corresponding to Monod kinetics, in the identification of the kinetics of a fermentation process [86]. The results obtained were readily interpreted and provided valuable insights into the process for the chemists and biologists working on it.

In addition, the tree structure of GP allows the tree to be of whatever complexity is needed to solve the problem, whereas the complexity of a neural network is constrained by the user's chosen network configuration. However, as this could increase the chances of GP "overfitting" the data, it is unclear whether it is an advantage or not. In fact, users of GP are advised to read a good neural networks text (e.g., [87]) because, despite claims to the contrary, issues such as overfitting, data preparation, and validation are equally applicable in GP.

In terms of CAMD applications, GP has yet to make a real impact. There are, however, some applications that show early promise. As mentioned earlier, the group at Michigan State University have experimented with both GAs and GP in combination with a KNN classifier with the intent of predicting water sites in X-ray crystal structures. In their experiments, they found GP to give superior results to the GA [77]. For instance, GP could give better predictions using fewer features and it was also capable of deriving nonlinear relationships, something not permitted by the GA's encoding. Handley [88] has described using GP to detect alpha-helical regions in proteins while Koza has investigated GP for classifying transmembrane segments [89]. The prediction of RNA structure has also been attempted [90]. In terms of data analysis and QSAR, nonlinear principal components analysis using GP has also been reported [91].

From a more analytical chemical point of view, Kell and coworkers have applied GP to a number of problems including the analysis of pyrolysis mass spectra [92–94] and nonlinear dielectric spectroscopy data [95]. Goodacre and Gilbert reported that, when applied to the analysis of pyrolysis mass spectra of caffeine-containing drinks, GP was much more useful than neural networks because the interpretable nature of the evolved function trees enabled the spectra to be deconvoluted, identifying the mass ions of particular significance for the classification [94].

Based on these few examples, it would seem that GP should have an exciting future in CAMD where the derivation of interpretable classifications is becoming of great interest, particularly in the analysis of high-throughput screening data. Some possible future directions for GP research and applications are given by Koza [96].

12.8.2 Neighborhood Search

Any discussion of EA methods would be incomplete with noting that they can be placed in the framework of a class of optimization algorithms known as *neighborhood (local) search* (see Fig. 3 for a taxonomy of neighborhood search algorithms). Consider an EA with a population of size one. Since crossover would not be possible, this would constitute a search by trial-and-error (*hillclimbing*), where a solution is repeatedly perturbed at random by a neighborhood (or mutation) operator until an improved solution is found. Of course, local optima are a problem for hillclimbing and numerous approaches have been devised to overcome this. Many chemists would, for instance, be familiar with simulated annealing [97], which probabilistically accepts nonimproving moves by a process analogous to that of annealing in metals and spin-glasses. A full overview is impossible here and the reader is directed to the references [98].

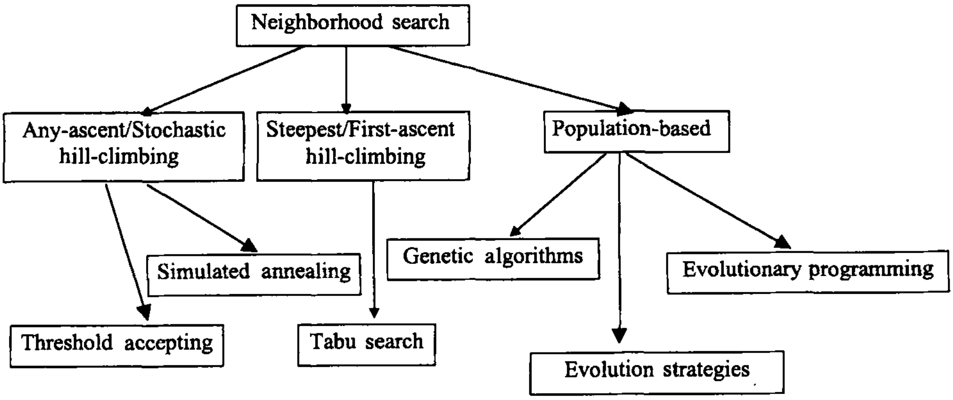


Figure 3. A taxonomy of neighborhood search methods.

One variant of neighborhood search that is worthy of attention here is *tabu search* [99, 100]. The approach here is to exploit the *memory* of the search history by using *explicit* rules (*logic*) to guide the search. Certain applications of operators can be made forbidden (*tabu active*) by application of the logic. For example, a common form of (short-term) memory used is *recency*. When no allowed improving move is available, the least nonimproving move is taken instead, and the inverse of this move is made tabu-active for a short time (the *tabu tenure*). This prevents the search cycling back into the local optimum from which it came.

Like GP, tabu search has yet to be fully exploited in CAMD, although there have been a few pioneering applications. An early example was that of Kvasnicka and Pospichal who used tabu search for the evaluation of chemical distance (molecular similarity) between chemical graphs [101]. Hong and Jhon applied a tabu search-based method to the optimization of argon cluster geometries [102]. In their work, they used the concept of a "tabu region", rather than a classical tabu list, to encourage sampling away from previous configurations. Pardalos et al. have reported that a tabu search procedure for lattice-based protein folding simulations was able to outperform SA in many cases [103]. Most recently, tabu search has been very successfully employed as a search algorithm for protein-ligand docking by Westhead et al. [34, 104], who consider it the method of choice in their application, and also Hou and coworkers [79].

The fact that tabu search is controlled by explicit rules may well be of interest to CAMD practitioners. Given that expert performance is partly due to an efficient search strategy, might it be possible to extract the human expert's underlying strategy and express it in a tabu search framework? The authors are unaware of work in this area, but its utility to CAMD practitioners should be clear.

12.8.3 Extending the Evolutionary Metaphor

Given the successes of EAs in a wide range of applications, much effort has been expended in the EA community with the aim of exploiting the evolutionary metaphor further, in other words, "putting more genetics into genetic algorithms" [105]. At first sight, this view does seem to have a great deal of validity, as two examples can confirm. First, the metaphor of co-evolution has been used both in self-adaptation (see section 12.4.1) and in a predator-prey sense to improve EA performance [106]. Second, the metaphor of "sexual selection" [107], where solutions are chosen for crossover on the basis of their similarity, so as to promote an effective balance between inbreeding and out-breeding, has been shown to improve EA performance for a number of problems.

There is little doubt that there is some scope for the future development of EAs based upon a closer observation of natural processes. Burke et al. list five key attributes of biological genetic representations not normally found in their computational analogs [105]:

- Biological genomes vary in length during evolution.
- Biological genes are independent of position.
- Biological genomes may contain noncoding regions.
- Biological genomes may contain duplicative or competing genes.
- Biological genomes have overlapping reading frames.

Burke et al. have experimented with these features in their VIV program with some interesting initial results.

In terms of applications of new biologically motivated algorithms in the CAMD domain, a co-evolutionary approach has been reported by Rosin and coworkers for the analysis of resistancė-evading peptidomimetic inhibitors of HIV-1 protease [108, 109]. In this work, a set of inhibitors and a set of mutant proteases compete with each other. The fitness function models the viability of a particular mutant virus when challenged by a given inhibitor. At each generation, new inhibitors are selected to block optimally the current set of proteases, and new protease mutants are selected that retain their ability to cleave their viral substrates in the presence of these inhibitors. In this scenario, the goal of the drug designer is to find an inhibitor that maximally inhibits the entire range of possible protease mutants [108]. Based on their simulations, Rosin et al. made a number of recommendations for the design of HIV-1 protease inhibitors with a view to maximizing their resistance-evading capacities. Manby et al. have developed a predatory GA designed to locate not merely the global optimum of a system but other low-lying minima – often an important consideration in CAMD [110].

At the close of this subsection, a word of caution would again seem apt. Unfortunately, in some instances, it seems that the drive to replicate more closely the underlying evolutionary metaphor is driven by a belief that such natural processes inherently possess some additional computational power – examples of this view can be readily found in the evolutionary computation literature. However, the NFL results discussed in section 12.3 earlier and the effectiveness of other approaches such as tabu search would seem to warn against adhering to this view too strongly. In fact, Glover and Laguna suggest that this view is possibly due to the wave of "neo-Romanticism" in the late twentieth century, characterized by a benign view of Nature [100]. These authors further argue that though such metaphors do have a place – to help suggest ideas from which to launch an investigation – the problem begins when the metaphor is taken too far and is allowed to define the actual research agenda.

12.9 New Application Domains

The preceding sections, combined with the previous chapters of this book, show clearly just how widely EAs have been applied in CAMD. In fact, it is hard to think of any part of the field that has not seen the experimental, if not successful, application of EAs. One area that is currently of great interest is that of data mining: the analysis of the vast volume of data that is generated by modern high-throughput biological and pharmacokinetic assays with a view to extracting explanatory and predictive models. There is some evidence that evolutionary computation is being used for data mining in the life sciences [77, 111] and medicine [112, 113] but applications in the domain of high-throughput screening (HTS) data are still awaited.

12.10 Collaborations and Commercial Applications

In addition to the new methods and application domains discussed above, there are two other trends that are worth highlighting in relation to EAs and CAMD. These are the increase in collaborative research between EA experts and CAMD scientists and the growing number of commercial software products incorporating EA methods.

In terms of the former, there are at least three examples of productive collaborations between EA and CAMD researchers. Natural Selection Inc. and Agouron Pharmaceuticals Inc. have worked together in the development of the EPDOCK and EPMR programs [32, 37], while researchers from the Computer Science and Biochemistry Departments of Michigan State University have collaborated in the creation of CONSOLV [76]. Similarly, the Department of Computer Science and Engineering and the Department of Molecular Biology at the Scripps Research Institute have combined to study co-evolutionary analysis applied to HIV-1 protease resistance [108, 109] and to develop new versions of the AUTODOCK docking program [68]. In all these cases, the former groups supply EA expertise while the latter bring in-depth knowledge of the problem in hand. As EA methodologies continue to develop, and as CAMD researchers seek to gain maximum value from their EA applications, such collaborations are likely to become more common and bring benefit to both communities.

New EA applications continue to appear in commercial CAMD software. Commercial programs using EAs are now available for *de novo* molecular design [47], conformational analysis [47], ligand-protein docking [47, 114, 115], molecular superposition and pharmacophore identification [47], QSAR [114] and combinatorial library design [114]. A related and interesting new development is the formation of a company specializing in applying EA-based CAMD methods to drug discovery [116].

12.11 Conclusions

A decade ago, the application of EAs in CAMD was almost unheard of. Today, it is commonplace, almost routine. Nonetheless, as we have tried to illustrate in this chapter, there are still many avenues to be explored in the search for more powerful and robust applications. Collaboration in this exciting quest promises much for both the EA researcher and the CAMD practitioner.

References

[1] G. W. A. Milne, Mathematics as a Basis for Chemistry, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 639–644.

[2] D. E. Clark, Evolutionary Algorithms in Rational Drug Design: A Review of Current Applications and a Look to the Future, in A. L. Parrill, M. R. Reddy (Eds.), *Rational Drug Design: Novel Methodology and Practical Applications*, ACS Symposium Series Vol. 719, American Chemical Society, Washington DC, USA, **1999**, pp. 255–270.

[3] G. Jones, Genetic and Evolutionary Algorithms, in P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, P. R. Schreiner (Eds.), *Encyclopedia of Computational Chemistry, Volume 2*, John Wiley & Sons, Chichester, UK, **1998**, pp. 1127–1136.

[4] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, USA, **1989**.

[5] T. Bäck, U. Hammel, H.-P. Schwefel, Evolutionary Computation: Comments on the History and Current State, *IEEE Trans. Evol. Comput.* **1997**, *1*, 3–17.

[6] L. J. Fogel, Future Research in Evolutionary Computation, in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, UK, **1997**, Section H1.2.

[7] Z. Michalewicz, Introduction to Search Operators, in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, UK, **1997**, Section C3.1.

[8] B. T. Luke, An Overview of Genetic Methods, in J. Devillers (Ed.), *Genetic Algorithms in Molecular Modelling*, Academic Press, USA, **1996**, pp. 35–66.

[9] D. B. Fogel, P. J. Angeline, Guidelines for a Suitable Encoding, in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, UK, **1997**, Section C1.7.

[10] Z. Michalewicz, S. Esquivel, R. Gallard, M. Michalewicz, G. Tao, K. Trojanowski, The Spirit of Evolutionary Algorithms, *J. Comput. Inf. Technol.* **1999**, *7*, 1–18.

[11] E. D. H. Wolpert, W. G. Macready, No Free Lunch Theorems for Optimization, *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82.

[12] R. Unger, J. Moult, Genetic Algorithms for Protein Folding Simulations. *J. Mol. Biol.* **1993**, *231*, 75–81.

[13] D. R. Westhead, D. E. Clark, D. Frenkel, J. Li, C. W. Murray, B. Robson, B. Waszkowycz, PRO_LIGAND: An Approach to De Novo Molecular Design. 3. A Genetic Algorithm for Structure Refinement. *J. Comput.-Aided Mol. Des.* **1995**, *9*, 139–148.

[14] R. C. Glen, A. W. R. Payne, A Genetic Algorithm for the Automated Generation of Molecules within Constraints. *J. Comput.-Aided Mol. Des.* **1995**, *9*, 181–202.

[15] M. Sullivan, Taking Drug Discovery to New Heights, *Today's Chemist at Work* **1999**, *8*, 44–46 (January issue).

[16] A. Globus, J. Lawton, T. Wipke, Automatic Molecular Design Using Evolutionary Techniques, *Nanotechnology* **1999**, *10*, 290–299.

[17] A. L. Tuson, *No Optimization Without Representation: A Knowledge Based Systems View of Evolutionary/Neighbourhood Search Optimization*, Ph.D. Thesis, University of Edinburgh, UK, **1999**.

[18] M. Stefik, *Introduction to Knowledge Systems*, Morgan Kaufmann, San Mateo, CA, USA, **1995**.

[19] A. Newell, The Knowledge Level, *Artif. Intell.* **1982**, *18*, 87–127.

[20] E. Motta, *Reusable Components in Knowledge Modelling*, Ph.D. Thesis, Open University, UK, **1997**.

[21] A. E. Eiben, R. Hinterding, Z. Michalewicz, Parameter Control in Evolutionary Algorithms, *IEEE Trans. Evol. Comput.* **1999**, *3*, 124–141.

[22] D. B. Fogel, Mutation: Real-valued Vectors, in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, UK, **1997**, Section C3.2.2.

[23] H.-P. Schwefel, *Numerical Optimization of Computer Models*, Wiley, Chichester, UK, **1981**.

[24] D. B. Fogel, L. J. Fogel, J. W. Atmar, Meta-evolutionary Programming, in R. R. Chen (Ed.), *Proceedings of the 25th Asilomar Conference on Signals, Systems and Computers*, Maple Press, San Jose, CA, USA, **1991**, pp. 540–545.

[25] N. Saravanan, D. B. Fogel, K. M. Nelson, A Comparison of Methods for Self-Adaptation in Evolutionary Algorithms, *BioSystems* **1995**, *36*, 157–166.

[26] D. K. Gehlhaar, D. B. Fogel, Tuning Evolutionary Programming for Conformationally Flexible Molecular Docking, in L. J. Fogel, P. J. Angeline, T. Bäck (Eds.), *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, MIT Press, Cambridge, MA, USA, **1996**, pp. 419–429.

[27] D. W. Corne, P. M. Ross, H.-L. Fang, *GA Research Note 7: Fast Practical.Evolutionary Timetabling*, AI Technical Report, University of Edinburgh, UK, **1994**.

[28] T. Bäck, F. Hoffmeister, H.-P. Schwefel, A Survey of Evolution Strategies, in D. Whitley (Ed.), *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, USA, **1991**, pp. 2–9.

[29] L. Davis, Adapting Operator Probabilities in Genetic Algorithms, in J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms and Their Applications*, Morgan Kaufmann, San Mateo, CA, USA, **1989**, pp. 61–69.

[30] B. A. Julstrom, What Have You Done for Me Lately? Adapting Operator Probabilities in a Steady-state Genetic Algorithm, in L. J. Eshelman (Ed.), *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, USA, **1995**, pp. 81–87.

[31] D. K. Gehlhaar, D. Bouzida, P. A. Rejto, Reduced Dimensionality in Ligand-Protein Structure Prediction: Covalent Inhibitors of Serine Proteases and Design of Site-Directed Combinatorial Libraries, in A. L. Parrill, M. R. Reddy (Eds.), *Rational Drug Design: Novel Methodology and Practical Applications*, ACS Symposium Series Vol. 719, American Chemical Society, Washington DC, USA, **1999**, pp. 292–311.

[32] D. K. Gehlhaar, G. M. Verkhivker, P. A. Rejto, C. J. Sherman, D. B. Fogel, L. J. Fogel, S. T. Freer, Molecular Recognition of the Inhibitor AG-1343 by HIV-1 Protease: Conformationally Flexible Docking by Evolutionary Programming, *Chem. Biol.* **1995**, *2*, 317–324.

[33] D. K. Gehlhaar, G. M. Verkhivker, P. A. Rejto, C. J. Sherman, D. B. Fogel, L. J. Fogel, S. T. Freer, Docking Conformationally Flexible Small Molecules into a Protein Binding Site Through Evolutionary Programming, in J. R. McDonnell, R. G. Reynolds, D. B. Fogel (Eds.), *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, MIT Press, Cambridge, MA, USA, **1995**, pp. 615–627.

[34] D. R. Westhead, D. E. Clark, C. W. Murray, A Comparison of Heuristic Search Algorithms for Molecular Docking, *J. Comput.-Aided Mol. Des.* **1997**, *11*, 209–228.

[35] X. Yao, Y. Liu, Fast Evolutionary Programming, in L. J. Fogel, P. J. Angeline, T. Bäck (Eds.), *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, MIT Press, Cambridge, MA, USA, **1996**, pp. 257–266.

[36] X. Yao, Y. Liu, G. Lin, Evolutionary Programming Made Faster, *IEEE Trans. Evol. Comput.* **1999**, *3*, 82–102.

[37] C. R. Kissinger, D. K. Gehlhaar, D. B. Fogel, Rapid Automated Molecular Replacement by Evolutionary Search, *Acta Crystallogr.* **1999**, *D55*, 484–491.

[38] B. A. Shapiro, J. C. Wu, An Annealing Mutation Operator in the Genetic Algorithms for RNA Folding, *CABIOS* **1996**, *12*, 171–180.

[39] G. Wang, T. W. Dexter, W. F. Punch, Optimization of a GA and Within a GA for a 2-Dimensional Layout Problem, in E. Goodman, W. F. Punch, V. Uskov, (Eds.), *Proceedings of the First International Conference on Evolutionary Computation and Its Applications*, Russian Academy of Sciences, Russia, **1996**, pp. 18–29.

[40] A. Tuson, P. W. Ross, Adapting Operator Settings in Genetic Algorithms, *Evol. Comput.* **1998**, *6*, 161–184.

[41] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, Berlin, Germany, **1996**.

[42] C. M. Fonseca, P. J. Fleming, An Overview of Evolutionary Algorithms in Multiobjective Optimization, *Evol. Comput.* **1995**, *3*, 1–16.

[43] C. A. C. Coello, A Comprehensive Survey of Evolutionary-based Multiobjective Optimization Techniques, *Knowledge Inf. Syst.* **1999**, *1*, 269–308. See also http://www.lania.mx/~ccoello/EMOO/EMOObib.html.

[44] S. Handschuh, M. Wagener, J. Gasteiger, Superposition of Three-Dimensional Chemical Structures Allowing for Conformational Flexibility by a Hybrid Method, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 220–232.

[45] J. A. Biles, GenJam: A Genetic Algorithm for Generating Jazz Solos, in *Proceedings of the 1994 International Computer Music Conference*, ICMA, San Francisco, USA, **1994**.

[46] A. Tuson, P. Ross, T. Duncan, On Interactive Neighbourhood Search Schedulers, in *Proceedings of the 16th Workshop of the UK Planning and Scheduling SIG*, **1997**.

[47] LeapFrog, Sybyl, FlexiDock and GASP. Available from Tripos Inc., 1699 S. Hanley Road, St. Louis, MO, USA.

[48] D. Levine, M. Facello, P. Hallstrom, G. Reeder, B. Walenz, F. Stevens, STALK: An Interactive System for Virtual Molecular Docking, *IEEE Comput. Sci. Eng.* **1997**, *4*, 55–65.

[49] E. Cantu-Paz, A Summary of Research on Parallel Genetic Algorithms, *IlliGAL Report No. 95007*, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA, **1997**.

[50] W. N. Martin, J. Lienig, J. P. Cohoon, Island (Migration) Models: Evolutionary Algorithms Based on Punctuated Equilibria, in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, UK, **1997**, Section C6.3.

[51] G. Jones, P. Willett, R. C. Glen, A. R. Leach, R. Taylor, Development and Validation of a Genetic Algorithm for Flexible Docking, *J. Mol. Biol.* **1997**, *267*, 727–748.

[52] G. Jones, P. Willett, R. C. Glen, A Genetic Algorithm for Flexible Molecular Overlay and Pharmacophore Elucidation, *J. Comput.-Aided Mol. Des.* **1995**, *9*, 532–549.

[53] M. L. M. Beckers, E. P. P. A. Derks, W. J. Melssen, L. M. C. Buydens, Parallel Processing of Chemical Information in a Local Area Network. III. Using Genetic Algorithms for Conformational Analysis of Biomacromolecules, *Comput. Chem.* **1996**, *20*, 449–457.

[54] C. A. Del Carpio, A Parallel Genetic Algorithm for Polypeptide Three-Dimensional Structure Prediction: A Transputer Implementation, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 258–269.

[55] C. Notredame, E. A. O'Brien, D. G. Higgins, RAGA: RNA Sequence Alignment by Genetic Algorithm, *Nucleic Acids Res.* **1997**, *25*, 4570–4580.

[56] J. V. Lehtonen, K. Denessiouk, A. C. W. May, M. S. Johnson, Finding Local Structural Similarities Among Families of Unrelated Protein Structures: A Generic Non-linear Alignment Algorithm, *Proteins: Struct., Funct., Genet.* **1999**, *34*, 341–355.

[57] B. T. Luke, Applications of Distributed Computing to Conformational Searches, in D. G. Truhlar, W. J. Howe, A. J. Hopfinger, J. Blaney, R. A Dammkoehler (Eds.), *Rational Drug Design*, Springer, New York, USA, **1999**, pp. 191–206.

[58] B. A. Shapiro, J. C. Wu, Predicting RNA H-Type Pseudoknots with the Massively Parallel Genetic Algorithm, *CABIOS* **1997**, *13*, 459–471.

[59] D. J. Wild, P. Willett, Similarity Searching in Files of Three-Dimensional Chemical Structures: Alignment of Molecular Electrostatic Potential Fields with a Genetic Algorithm, *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 159–167.

[60] M. C. Nicklaus, R. W. Williams, B. Bienfait, E. S. Billings, M. Hodoscek, Computational Chemistry on Commodity-type Computers, *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 893–905.

[61] T. Ibaraki, Combinations with Other Optimization Methods, in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, UK, **1997**, Section D.3.

[62] W. V. Porto, Neural-Evolutionary Systems, in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, UK, **1997**, Section D.1.

[63] C. L. Karr, Fuzzy-Evolutionary Systems, in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, UK, **1997**, Section D.2.

[64] M. J. D. Powell, A Restart Procedure for the Conjugate Gradient Method, *Math. Programming* **1977**, *12*, 241–254.

[65] D. B. Hibbert, A Hybrid Genetic Algorithm for the Estimation of Kinetic Parameters, *Chemom. Intell. Lab. Syst.* **1993**, *19*, 319–329.

[66] D. B. McGarrah, R. S. Judson, Analysis of the Genetic Algorithm Method of Molecular Conformation Determination, *J. Comput. Chem.* **1993**, *14*, 1385–1395.

[67] R. S. Judson, E. P. Jaeger, A. M. Treasurywala, M. L. Peterson, Conformational Searching Methods for Small Molecules. II. Genetic Algorithm Approach, *J. Comput. Chem.* **1993**, *14*, 1407–1414.

[68] G. M. Morris, D. S. Goodsell, R. S. Halliday, R. Huey, W. E. Hart, R. K. Belew, A. J. Olson, Automated Docking using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function, *J. Comput. Chem.* **1998**, *19*, 1639–1662.

[69] C. Frey, An Evolutionary Algorithm with Local Search and Classification for Conformational Searching, *MATCH* **1998**, *38*, 137–159.

[70] S.-S. So, M. Karplus, Evolutionary Optimization in Quantitative Structure-Activity Relationship: An Application of Genetic Neural Networks, *J. Med. Chem.* **1996**, *39*, 1521–1530.

[71] S.-S. So, M. Karplus, M. Genetic Neural Networks for Quantitative Structure-Activity Relationships: Improvements and Application of Benzodiazepine Affinity for Benzodiazepine/GABA(A) Receptors, *J. Med. Chem.* **1996**, *39*, 5246–5256.

[72] S.-S. So, M. Karplus, Three-Dimensional Quantitative Structure-Activity Relationships from Molecular Similarity Matrices and Genetic Neural Networks. 1. Method and Validations, *J. Med. Chem.* **1997**, *40*, 4347–4359.

[73] S.-S. So, M. Karplus, Three-Dimensional Quantitative Structure-Activity Relationships from Molecular Similarity Matrices and Genetic Neural Networks. 2. Applications, *J. Med. Chem.* **1997**, *40*, 4360–4371.

[74] J. Kyngas, J. Valjakka, Evolutionary Neural Networks in Quantitative Structure-Activity Relationships of Dihydrofolate Reductase Inhibitors, *Quant. Struct.-Act. Relat.* **1996**, *15*, 296–301.

[75] H. Yoshida, K. Funatsu, Optimization of the Inner Relation Function of QPLS Using Genetic Algorithm, *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 1115–1121.

[76] M. L. Raymer, P. C. Sanschagrin, W. F. Punch, S. Venkataraman, E. D. Goodman, L. A. Kuhn, Predicting Conserved Water-Mediated and Polar Ligand Interactions in Proteins Using a K-nearest-neighbors Genetic Algorithm, *J. Mol. Biol.* **1997**, *265*, 445–464.

[77] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, Genetic Programming for Improved Data Mining: Application to the Biochemistry of Protein Interactions, in J. R. Koza, D. E. Goldberg, D. B. Fogel, R. L. Riolo (Eds.), *Genetic Programming 1996: Proceedings of the First Annual Conference*, MIT Press, Cambridge, MA, USA, **1996**, pp. 375–380.

[78] J. R. Gunn, Sampling Protein Conformations Using Segment Libraries and a Genetic Algorithm, *J. Chem. Phys.* **1997**, *106*, 4270–4281.

[79] T. Hou, J. Wang, L. Chen, X. Xu, Automated Docking of Peptides and Proteins by Using a Genetic Algorithm Combined with a Tabu Search, *Protein Eng.* **1999**, *12*, 639–648.

[80] D. Bouzida, D. K. Gehlhaar, P. A. Rejto, G. M. Verkhivker, S. T. Freer, Efficient Configurational Search Methods for Flexible Ligand Docking, *Abstracts of the 11th European Symposium on Quantitative Structure-Activity Relationships: Computer-Assisted Lead Finding and Optimization*, Lausanne, Switzerland, **1996**, P-37D.

[81] J. Chen, H. Chi, Fast Docking of Drug Molecules to their Receptor, *Chin. Sci. Bull.* **1999**, *44*, 904–908.

[82] C. R. Zacharias, M. R. Lemes, A. Dal Pino, Combining Genetic Algorithm and Simulated Annealing: A Molecular Geometry Optimization Study, *J. Mol. Struct. (THEOCHEM)* **1998**, *430*, 29–39.

[83] W. Cai, L. Wang, Z. Pan, X. Shao, Analysis of Extended X-ray Absorption Fine Structure Spectra Using Annealing Evolutionary Algorithms, *Anal. Commun.* **1999**, *36*, 313–315.

[84] C. Waller, M. Bradley, Development and Validation of a Novel Variable Selection Algorithm with Application to Multidimensional QSAR Studies, *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 345–355.

[85] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, **1992**.

[86] H. Pohlheim, P. Marenbach, Generation of Structured Process Models Using Genetic Programming, in T. C. Fogarty (Ed.), *Evolutionary Computing: AISB Workshop (LNCS 1143)*, Springer-Verlag, Berlin, Germany, **1996**.

[87] T. Masters, *Practical Neural Network Recipes in C++*, Academic Press, **1993**.

[88] S. Handley, Automated Learning of a Detector for Alpha-Helices in Proteins via Genetic Programming, in S. Forrest (Ed.), *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, USA, **1993**, pp. 271–278.

[89] J. R. Koza, Classifying Protein Segments as Transmembrane Domains Using Genetic Programming and Architecture-Altering Operations, in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, UK, **1997**; Section G6.1.

[90] K. Yamaguchi, C. A. Del Carpio, A Genetic Programming Based System for the Prediction of Secondary and Tertiary Structures of RNA, *Genome Inf. Ser.* **1998**, *9*, 382–383.

[91] H. G. Hiden, M. J. Willis, M. T. Tharn, G. A. Montague, Nonlinear Principal Components Analysis Using Genetic Programming, *Comput. Chem. Eng.* **1999**, *23*, 413–425.

[92] J. Taylor, R. Goodacre, W. G. Wade, J. J. Rowland, D. B. Kell, The Deconvolution of Pyrolysis Mass Spectra Using Genetic Programming: Application to the Identification of Some Eubacterium Species, *FEMS Microbiol. Lett.* **1998**, *160*, 237–246.

[93] R. J. Gilbert, R. Goodacre, A. M. Woodward, D. B. Kell, Genetic Programming: A Novel Method for the Quantitative Analysis of Pyrolysis Mass Spectral Data, *Anal. Chem.* **1997**, *69*, 4381–4389.

[94] R. Goodacre, R. J. Gilbert, The Detection of Caffeine in a Variety of Beverages using Curie-point Pyrolysis Mass Spectrometry and Genetic Programming, *Analyst* **1999**, *124*, 1069–1074.

[95] A. M. Woodward, R. J. Gilbert, D. B. Kell, Genetic Programming as an Analytical Tool for Non-linear Dielectric Spectroscopy, *Bioelectrochem. Bioenerg.* **1999**, *48*, 389–396.

[96] J. R. Koza, Future Work and Practical Applications of Genetic Programming, in T. Bäck, D. B. Fogel, Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing, Bristol, UK, **1997**, Section H1.1.

[97] S. Kirkpatrick, C. D. Gelatt Jr., M. P. Vecchi, Optimization by Simulated Annealing, *Science* **1983**, *220*, 671–680.

[98] C. R. Reeves, *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, Oxford, UK, **1993**.

[99] D. Cvijovic, J. Klinowski, Taboo Search: An Approach to the Multiple Minimum Problem, *Science* **1995**, *267*, 664–666.

[100] F. W. Glover, M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Boston, MA, USA, **1997**.

[101] V. Kvasnicka, J. Pospichal, Fast Evaluation of Chemical Distance by Tabu Search Algorithm, *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1109–1112.

[102] S. D. Hong, M. S. Jhon, Restricted Random Search Method based on Taboo Search in the Multiple Minima Problem, *Chem. Phys. Lett.* **1995**, *267*, 422–426.

[103] P. M. Pardalos, X. Liu, G. L. Xue, Protein Conformation of a Lattice Model Using Tabu Search, *J. Global Optimization* **1997**, *11*, 55–68.

[104] C. A. Baxter, C. W. Murray, D. E. Clark, D. R. Westhead, M. D. Eldridge, Flexible Docking Using Tabu Search and an Empirical Estimate of Binding Affinity, *Proteins: Struct., Funct., Genet.* **1998**, *33*, 367–382.

[105] D. S. Burke, K. A. De Jong, J. J. Grefenstette, C. S. Ramsey, A. S. Wu, Putting More Genetics into Genetic Algorithms, *Evol. Comput.* **1998**, *6*, 387–410.

[106] W. D. Hillis, Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure, *Artificial Life II*, Addison Wesley, **1991**.

[107] M. Ratford, A. L. Tuson, H. Thompson, The Single Chromosome's Guide to Dating, in G. D. Smith et al. (Eds.), *Artificial Neural Nets and Genetic Algorithms: Proceedings of the Third International Conference On Artificial Neural Networks And Genetic Algorithms (ICANNGA 97)*, Springer-Verlag, Berlin, Germany, **1997**.

[108] C. D. Rosin, R. K. Belew, G. M. Morris, A. J. Olson, D. S. Goodsell, Coevolutionary Analysis of Resistance-Evading Peptidomimetic Inhibitors of HIV-1 Protease, *Proc. Natl. Acad. Sci. (USA)* **1999**, *96*, 1369–1374.

[109] C. D. Rosin, R. K. Belew, W. L. Walker, G. M. Morris, A. J. Olson, D. S. Goodsell, Coevolution and Subsite Decomposition for the Design of Resistance-Evading HIV-1 Protease Inhibitors, *J. Mol. Biol.* **1999**, *287*, 77–92.

[110] F. R. Manby, R. L. Johnston, C. Roberts, Predatory Genetic Algorithms, *MATCH* **1998**, *38*, 111–122.

[111] M. L. Raymer, W. F. Punch, E. D. Goodman, P. C. Sanschagrin, L. A. Kuhn, Simultaneous Feature Scaling and Selection Using a Genetic Algorithm, in T. Bäck (Ed.), Proceedings of the Seventh International Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco, CA, USA, **1997**, pp. 561–567.

[112] P. S. Ngan, M. L. Wong, W. Lam, K. S. Leung, J. C. Cheng, Medical Data Mining using Evolutionary Computation, *Artif. Intell. Med.* **1999**, *16*, 73-96.

[113] J. Laurikkala, M. Juhola, S. Lammi, K. Viikki, Comparison of Genetic Algorithms and Other Classification Methods in the Diagnosis of Female Urinary Incontinence, *Methods Inf. Med.* **1999**, *38*, 125–131.

[114] CHARMm, C^2.GA and C^2.LibSelect. Available from Molecular Simulations Inc., 9685 Scranton Road, San Diego, CA, USA.

[115] GOLD (Genetic Optimization for Ligand Docking). Available from Cambridge Crystallographic Data Centre, 12 Union Road, Cambridge, CB2 1EZ, UK.

[116] Nanodesign Inc., Suite 300, Research Park Centre, 150 Research Lane, Guelph, Ontario, N1G 4T2, Canada. See http://www.nanodesign.com.

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

APPENDIX: Internet Resources for Evolutionary Algorithms

There are many web sites and mailing lists that provide useful information concerning evolutionary algorithms. Below is a selection of these resources.

General Resources for Evolutionary Algorithms

- http://www.aic.nrl.navy.mil/galist/ – The Genetic Algorithms Archive is a repository for information related to research in genetic algorithms and other forms of evolutionary computation. This site provides source code for many GA implementations, an archive of messages from the GA list discussion group and announcements about GA-related conferences. Also, links are given to many interesting sites around the world with material related to evolutionary computation.
- http://www.cs.clemson.edu/GA/clife/Welcome.html – ENCORE (the EvolutioNary Computation Repository network) is the electronic appendix to the Hitch-Hiker's Guide to Evolutionary Computation, a compendium of files on the art of evolutionary computation.
- http://www.dcs.napier.ac.uk/evonet/ – EvoNet (the network of excellence in Evolutionary Computing). EvoNet has been set up to encourage co-operation between evolutionary computing researchers in Europe.

Resources for Applications in Computer-Aided Molecular Design

- http://panizzi.shef.ac.uk/cisrg/links/ea_bib.html – a regulary updated bibliography of EA applications in CAMD maintained by the editor of this book and hosted at the Department of Information Studies, University of Sheffield.
- http://members.aol.com/btluke/gmovr1.htm – Brian T. Luke's overview of genetic methods.
- http://www.scripps.edu/pub/olson-web/doc/autodock – the AutoDock homepage (see chapter 3).
- http://panizzi.shef.ac.uk/cisrg/chem.html – the Computational Chemistry Research Group at the Department of Information Studies, University of Sheffield – home of GOLD, GASP, FBSS and SELECT programs (see chapters 3, 4 and 7).
- http://mmlin1.pha.unc.edu/~jin/QSAR/ – the QSAR server at the Laboratory for Molecular Modelling, University of North Carolina, Chapel Hill, including the GA-PLS method (see chapter 5).
- http://www.mol.biol.ethz.ch/wuthrich/software/garant/ – GARANT program homepage. GARANT is a program for automatic resonance assignment (see chapter 10).

Evolutionary Algorithms in Molecular Design
Edited by David E. Clark

Index

WILEY-VCH

Evolutionary Algorithms in Molecular Design

Edited by David E. Clark

Methods and Principles in Medicinal Chemistry

Volume 8

When trying to find new methods and problem-solving strategies for their research, scientists often turn to nature for inspiration. An excellent example of this is the application of Darwin's Theory of Evolution, particularly the notion of the 'survival of the fittest', in computer programs designed to search for optimal solutions to many kinds of problems. These 'evolutionary algorithms' start from a population of possible solutions to a given problem and, by applying evolutionary principles, evolve successive generations with improved characteristics until an optimal, or near-optimal, solution is obtained.

This book highlights the versatility of evolutionary algorithms in areas of relevance to molecular design with a particular focus on drug design. The authors, all of whom are experts in their field, discuss the application of these computational methods to a wide range of research problems including conformational analysis, chemometrics and quantitative structure-activity relationships, de novo molecular design, chemical structure handling, combinatorial library design, and the study of protein folding. In addition, the use of evolutionary algorithms in the determination of structures by X-ray crystallography and NMR spectroscopy is also covered.

These state-of-the-art reviews, together with a discussion of new techniques and future developments in the field, make this book a truly valuable and highly up-to-date resource for anyone engaged in the application or development of computer-assisted methods in scientific research.

ISBN 3-527-30155-0

9 783527 301553