Separation Science and Technology
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713708471

Optimization of a Reverse Osmosis System Using Genetic Algorithm

Z. V. P. Murthy[a]; Jiju Cherian Vengal[a]
[a] Department of Chemical Engineering, Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat, India

To cite this Article Murthy, Z. V. P. and Vengal, Jiju Cherian(2006) 'Optimization of a Reverse Osmosis System Using Genetic Algorithm', Separation Science and Technology, 41: 4, 647 — 663
To link to this Article: DOI: 10.1080/01496390500526854
URL: http://dx.doi.org/10.1080/01496390500526854

Separation Science and Technology, 41: 647–663, 2006

ISSN 0149-6395 print/1520-5754 online
DOI: 10.1080/01496390500526854

Optimization of a Reverse Osmosis System Using Genetic Algorithm

Z. V. P. Murthy and Jiju Cherian Vengal
Department of Chemical Engineering, Sardar Vallabhbhai National Institute of Technology, Surat, Gujarat, India

Abstract: Reverse Osmosis (RO) has found extensive application in industry as a highly efficient separation process. In most cases, it is required to select the optimum set of operating variables such that the performance of the system is maximized. In this work, an attempt has been made to optimize the performance of RO system with a cellulose acetate membrane to separate NaCl-Water system using Genetic Algorithm (GA). The GAs are faster and more efficient than conventional gradient based optimization techniques. The optimization problem was to maximize the observed rejection of the solute by varying the feed flowrate and overall permeate flux across the membrane for a constant feed concentration. To model the system, a well-established transport model for RO system, the Spiegler-Kedem model was used. It was found that the GA converged rapidly to the optimal solution at the 8th generation. The effect of varying GA parameters like size of population, crossover probability, and mutation probability on the result was also studied. The algorithm converged to the optimum solution set at the 8th generation. It was also seen that varying the computational parameters significantly affected the results.

Keywords: Reverse osmosis, genetic algorithm, optimization, Spiegler-Kedem model, membrane transport model

INTRODUCTION

Genetic Algorithms (GAs) are stochastic search methods that mimic the process of natural biological evolution. Genetic algorithms operate on a

Received 2 May 2005, Accepted 5 December 2005
Address correspondence to Z. V. P. Murthy, Department of Chemical Engineering, Sardar Vallabhbhai National Institute of Technology, Surat 395007, Gujarat, India. Fax: 0091 261 2227334; E-mail: zvpm2000@yahoo.com and zvpm@ched.svnit.ac.in

population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. In the field of chemical engineering design, GAs have been applied for different operations (1–11, 24). Reverse Osmosis (RO) is one of the most popular and established membrane separation processes. Osmosis is the flow of solvent through a semipermeable membrane from the less concentrated to the more concentrated region. The osmotic flow is a natural occurrence as the system tends to come to equilibrium and equalize chemical potentials. The osmotic flow can be decreased by applying pressure to the more concentrated solution. The higher the applied pressure, the less is the osmotic flow. When the flow stops, the applied pressure is the osmotic pressure. Reverse osmosis is a process which reverses the normal direction of osmosis by increasing the pressure of the concentrated stream. Industrial use of RO systems has been preferred as they are more capital and energy efficient in comparison to conventional separation techniques such as distillation, evaporation, and electrodialysis (25). Today, RO systems are widely used in desalination and water treatment facilities. The main advantages of RO over other desalination processes are its simple design, lower maintenance costs, easier de-bottlenecking, simultaneous removal of both organic and inorganic impurities, low discharge in the purge stream, and energy savings. RO is a rate-governed pressure-driven process. The solvent flux depends upon the applied pressure difference, trans-membrane osmotic pressure difference, concentration of feed, permeability coefficients of salt and water, and the extent of concentration polarization. The flux increases (at the expense of high concentration polarization) with an increase in the operating pressure difference and permeability coefficients, and decreases with an increase in the salt concentration (24).

Mathematical models and optimization techniques are being used extensively in many areas of chemical engineering process design and operation. Recently, these techniques have been applied to RO systems also (10–13, 26). Attempts have been made to obtain optimal designs of RO units considering cost as the single objective function (27). Sequential quadratic programming (SQP) has been used (26) to find optimal networks of RO modules. These studies involve the optimization of only a single objective function. Like most problems, the design of RO modules is also associated with several non-commensurate, objective functions that need to be optimized simultaneously in the presence of a few constraints. Such problems are best handled using multi-objective optimization (MOO) techniques (1, 24, 27–38).

In the present work, a single objective optimization of a laboratory scale RO system is carried out using a simple genetic algorithm. Observed solute rejection (Ro) of a RO membrane expresses the effectiveness of a membrane to remove salts from the water. As the objective of a RO process is to maximize the solute rejection, Ro is chosen as the optimization parameter. Feed flowrate (Q) and overall flux (Jv) have been reported as important process variables in the operation of a RO plant. Thus these two parameters were chosen as the decision variables (14). The first part of the paper discusses the principles of GA and its working. The next part deals with the description of the optimization problem and the model equations used to represent the system. Further the results obtained after executing the algorithm is discussed in the last part of the paper.

GENETIC ALGORITHMS

Over the past couple of years genetic algorithms have been extensively used (4, 39) to solve optimization problems involving single objective functions. This simple genetic algorithm (SGA) (39) offers advantages (12, 37) over more traditional optimization approaches. Genetic algorithms score over conventional gradient based optimization methods like Newton's method, quadratic programming, conjugate gradient methods etc. in a number of ways. It is a population based technique producing a number of solutions at each iteration, unlike conventional methods which produce a single solution at each stage, thus having a higher probability to converge to local optima. Moreover, Genetic algorithms do not require derivative information, as required by gradient search techniques, or other auxiliary knowledge of the objective function implying that a wide range of functions can be solved using GA. In the early algorithms, binary coding was used for representing the continuous decision variables, i.e., these variables were represented/coded as a series (string) of binary numbers (and then mapped into real numbers for use in model equations). This is an unavoidable compromise and causes problems (12, 37), e.g., it slows down the computing speed and, at times, renders convergence impossible. Modifications (e.g., real coded Gas, the jumping gene adaptation, etc.) are becoming available but each technique has its own limitations (24, 40). Thus, GA can be effectively applied to optimize nonlinear and multi-objective problems. In recent times, a lot of work has been published in the literature on different modifications and applications of GAs in the chemical engineering field (30, 34, 35, 41–58).

Genetic algorithm is a population based optimization technique where the principles of natural evolution are applied to obtain the *fittest* solution of a given problem. Unlike other optimization methods, GA works with a *population* of candidate solutions. Each of these candidate solutions called *chromosome* is given a *fitness* value. The *chromosomes* undergo genetic operations like *selection*, *crossover* (*reproduction*), and *mutation* to yield a new generation of chromosomes. The fitness of the population increases over the generations and finally converges to an optimum value. A flow sheet showing the working of a GA is given (23) in Appendix I. Once the objective function to be optimized and the decision variables of the problem have been defined, the algorithm initializes a random population of chromosomes. Each chromosome represents a solution and a chromosome is constituted of *genes* which represent the value of the decision variable used to arrive at that particular solution. The value of the objective function of each chromosome is taken as the *fitness* of the chromosome and another parameter called fitness function is defined as:

$$\text{Fitness function} = \frac{f(s_i(t))}{\sum_{j=1}^{M} f(s_i(t))} \tag{1}$$

where $f(s_i(t))$ is the fitness of the chromosome s_i and the denominator represents the sum of the fitness of all the members of the population. Depending on the fitness of the chromosomes, they are selected for crossover. The different methods of selection are discussed elsewhere (5). Crossover is the process of producing two offspring solutions by interchanging the genetic properties of two parent chromosomes. Consequently it produces children with opposing mixture of their parents' genes. Another important term is Crossover Probability (P_{cross}) which gives the probability of a chromosome being selected for crossover. By crossover offsprings are generated until the population number is satisfied.

Another important genetic operator is mutation. Mutation leads to the random alteration of the properties of a chromosome. It is possible that the population converges to premature local maxima/minima after few generations, in this context, mutation is important to maintain variety in the population. Like P_{cross} there is mutation probability (P_{mute}) which gives the probability for a chromosome to be mutated. After these genetic operations are completed, a new generation of chromosomes is formed. The above mentioned processes are applied to the candidate solutions of the new generation and the next generation is formed. This evolutionary procedure is continued until the termination criterion is reached. The termination criteria can be the value of the fitness function or a maximum number of generations.

PROBLEM FORMULATION AND OPTIMIZATION

Problem Formulation

The reverse osmosis experiments were performed with a laboratory setup when one of the authors (ZVPM) was completing his Ph. D. at the Indian Institute of Technology, Delhi (14). The membrane-housing cell was made of stainless steel with two halves fastened together with high tensile bolts. The top half of the cell contained the flow distribution chamber and the bottom half was used as the membrane support. Sufficient membrane support arrangement was given to withstand high pressures. Detailed procedure of the experimentation and the specifications of the membrane used are given elsewhere (14, 15, 21). The hydraulic diameter of the cell is 0.9 cm and the average depth is 0.5 cm. The effective membrane surface area is 60 cm^2. The data of observed rejection (Ro) and permeate flux (Jv) was collected for different combinations of concentrations, feed flowrates (Q), and pressures.

The system under study is a cellulose acetate membrane RO set up handling NaCl—water system (14). The feed concentration has been kept constant at 1000 ppm, while varying the feed flowrate from 300 mL/min to 1500 mL/min and the overall flux from 0.0001 cm/s to 0.001412 cm/s. The optimization problem was to maximize the rejection of the solute while varying the feed flowrate and the overall flux across the membrane. To model and obtain the relationship between the variables, common transport models describing RO phenomenon were studied. Of the various models reported (14–22), the Spiegler-Kedem model was found to give satisfactory results. Moreover, the Spiegler-Kedam model was shown to accurately represent the RO system under investigation (14–19). Thus, for the present study, the above model was chosen. The Spiegler-Kedem model is based on irreversible thermodynamics and involves three parameters. The model can be mathematically expressed as:

$$\mathrm{Ro}/(1-\mathrm{Ro}) = [\sigma/(1-\sigma)]\,[1-\exp(-\mathrm{Jv}(1-\sigma)/\mathrm{P_M})]\exp(-\mathrm{Jv}/\mathrm{k}) \quad (2)$$

Here, σ is the reflection coefficient which represents the rejection capability of a membrane, i.e., $\sigma = 0$ means no salt rejection and $\sigma = 1$ means 100% salt rejection, P_M is the local solute permeability per unit membrane thickness, and k is the mass transfer coefficient. The parameters of equation (2), viz. σ, P_M and k were estimated for different RO systems by one of the authors (14–19). Using the values of the estimated parameters, equation (1) was written in terms of the problem variables.

The fitness variable in the present case is Ro, thus equation (2) was written in Ro as,

$$\mathrm{Ro} = \frac{[\sigma/(1-\sigma)][1-\exp(-Jv(1-\sigma)/P_M)]\exp(-Jv/k)}{1+[\sigma/(1-\sigma)][1-\exp(-Jv(1-\sigma)/P_M)]\exp(-Jv/k)} \quad (3)$$

The values of σ, P_M, Jv and k for the constant feed concentration of 1000 ppm were taken (18) from Table 1.

Optimization

The optimization problem was solved using the simple genetic algorithm (SGA) coded in C, obtained from the Kanpur Genetic Algorithm Laboratory, India. The method used to select genes for producing offspring has a

Table 1. Input variables and their values supplied to GA program

Input	Value
Number of Generation	20
Population Size (N_{pop})	20
Crossover Probability (P_{cross})	0.90
Mutation Probability (P_{mute})	0.001
Number of binary coded variables	2
Sub-string length	10
Range of Q	300 mL/min – 1500 mL/min
Range of Jv	0.0001 cm/s–0.001412 cm/s

Initial NaCl concentration	Q, mL/min	$P_M \times 10^5$,cm/s	σ	$k \times 10^4$, cm/s
Membrane parameters P_M, σ, and k used in the present case [18]				
1000	300	4.170	0.9398	81.91
1000	600	4.175	0.9398	140.82
1000	900	4.168	0.9396	197.28
1000	1200	4.167	0.9396	244.14
1000	1500	4.150	0.9393	286.63

Initial NaCl concentration	Q, mL/min	ΔP, atm	Ro	$Jv \times 10^4$, cm/s
Data used to estimate P_M, σ, and k [18]				
1000	300	20	0.7613	1.62
1000	300	30	0.8343	2.81
1000	300	40	0.8752	4.46
1000	300	60	0.9003	6.87
1000	300	80	0.9119	9.44
1000	300	100	0.9183	12.76
1000	600	20	0.7773	1.78
1000	600	30	0.8408	2.94
1000	600	40	0.8831	4.82
1000	600	60	0.9041	6.98
1000	600	80	0.9168	9.81
1000	600	100	0.9233	12.95
1000	900	20	0.7890	1.92
1000	900	30	0.8444	3.02
1000	900	40	0.8863	4.98
1000	900	60	0.9067	7.22
1000	900	80	0.9191	10.11
1000	900	100	0.9256	13.40
1000	1200	20	0.8091	2.22

(*continued*)

Table 1. Continued

Initial NaCl concentration	Q, mL/min	ΔP, atm	Ro	Jv $\times$ 10^4, cm/s
1000	1200	30	0.8478	3.11
1000	1200	40	0.8895	5.19
1000	1200	60	0.9093	7.54
1000	1200	80	0.9216	10.78
1000	1200	100	0.9270	13.76
1000	1500	20	0.8178	2.38
1000	1500	30	0.8539	3.26
1000	1500	40	0.8914	5.32
1000	1500	60	0.9105	7.69
1000	1500	80	0.9224	10.89
1000	1500	100	0.9279	14.12

significant bearing on the performance of the algorithm. In the present problem, Tournament selection was the method used (59, 60). In Tournament Selection, two candidate chromosomes are chosen at random and the chromosome with the higher fitness among the two is selected for mating, thus a population of ‘n’ chromosomes would require ‘2n’ tournaments for each generation. Crossover is the process of combining the characteristics of the two parent chromosomes to produce an offspring. Single point crossover was the technique used in the present problem. In single point crossover, a crossover site along the bit strings of the chromosomes is chosen and the values of the two chromosomes up to that point are swapped to produce a new offspring. A popular method used to maintain the genetic diversity of the population and thus ensuring that the population does not converge to local minima is *mutation*. In binary mutation, a widely used mutation technique, the values of the chromosome selected for mutation is flipped, i.e. from 1 to 0 and vice versa. In the present problem, binary mutation, with a mutation probability of 0.001 was chosen. The program was run for 20 generations which was used as the termination criteria of the problem. The computational time was less than a minute in an Intel Celeron (2.4 GHz) machine. The various inputs given to the program were given in Table 1. Further, the program was used to study the effect of varying the GA computational parameters, viz. $N_{pop,}$ P_{cross}, and P_{mute} on the fitness of the population.

RESULTS AND DISCUSSION

The values of the optimized output variables obtained are given in Table 2. Figure 1 represents the evolution of the chromosomes over the generations.

Table 2. The optimized output variables obtained from GA program

Variable	Value
Maximum fitness (Ro,max)	0.93090
Optimum Q	1495.30 mL/min
Optimum Jv	0.0014 cm/s
Converged in	8th generation

It is observed that the average fitness of the chromosomes increases over the generations and finally converge to an optimum value. In the present study the maximum rejection obtained was 0.93090 at a feed flowrate of 1495.30 mL/min and an overall flux of 0.0014 cm/s. For a better understanding of the comprehensiveness of the algorithm, the complete output set of the 8th generation is given in Appendix II. For each of the 20 chromosomes, the real and binary values of the variables and corresponding fitness are shown.

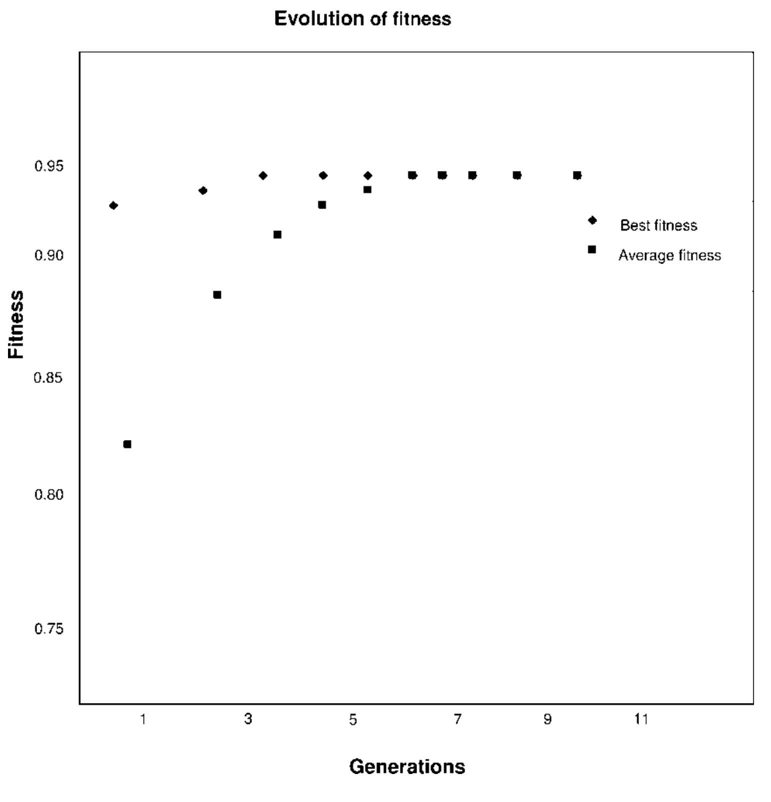


Figure 1. Plot showing the evolution of fitness with generation.

The results show that the rejection increases with both Q and Jv. Unlike conventional optimization methods, GA is a population based method, which means that at the end of each iteration, a population of solutions will be generated and not just a single optimum solution. In the present problem a total of 400 solutions were generated, thus giving the operator a wide variety of variable combinations to choose from for the desired operation of the plant.

Effect of Varying the Computational Parameters

In any genetic algorithm, the values of the computational parameters can be chosen from a wide range. It is important to use the optimum value for the various parameters to obtain the best results. In order to understand the effect of varying the computational parameters and to obtain the optimum values of each, an extensive study was conducted by varying N_{pop}, P_{cross}, and P_{mute}. As reported in literature (10), P_{mute} was found to be the most sensitive parameter. The program was tested for P_{mute} 0.1, 0.05, and 0.001. The effect of P_{mute} on the fitness and the fitness distribution of the population are shown in Fig. 2 and 3, for P_{mute} 0.05 and 0.001, respectively.

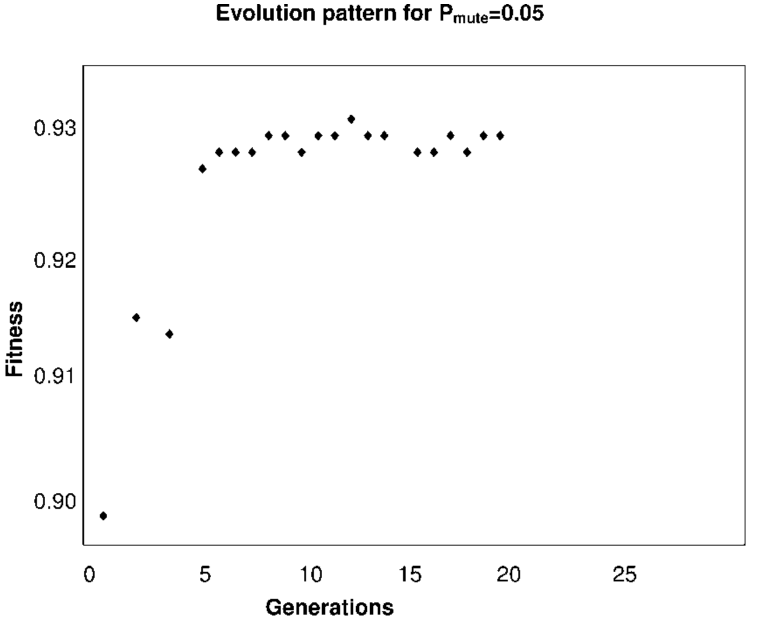


Figure 2. Fitness values over generations for $P_{mute} = 0.05$.

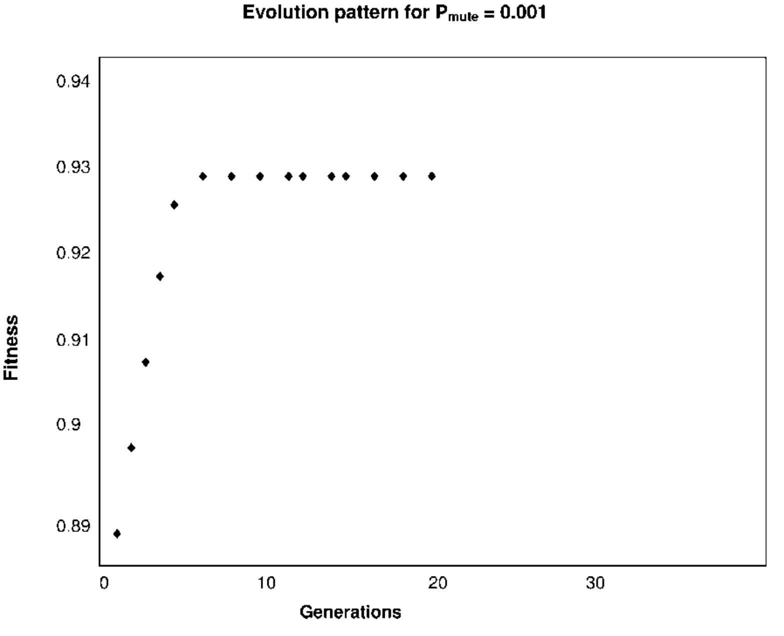


Figure 3. Fitness value over generations for $P_{mute} = 0.001$.

It is observed that with an increase in the mutation probability, the variety in the population increases and the chromosomes do not converge to an optimum. While with a decrease in P_{mute} it was observed that the values quickly converged to an optimum. As a large variation in the fitness in each

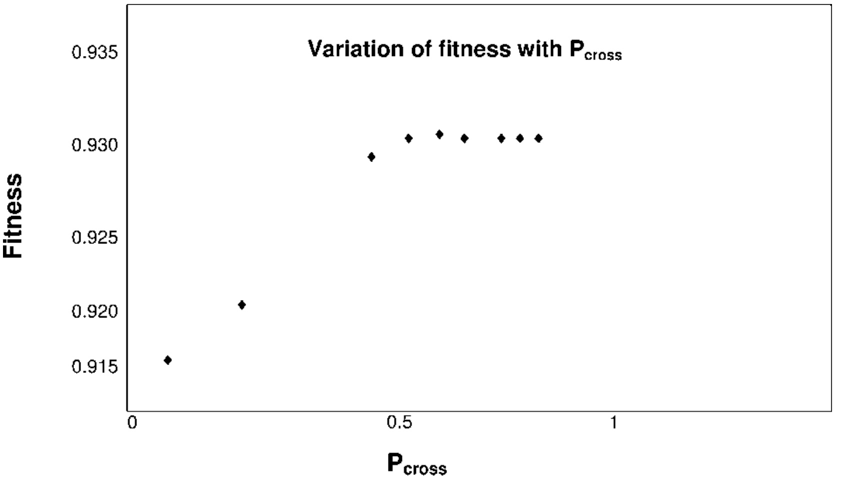


Figure 4. Plot showing the variation of Fitness with different P_{cross} values.

generation is not preferred, it is advisable to keep P_{mute} to a low value. A P_{mute} of 0.001 was found to be satisfactory in the present study. Similarly, P_{cross} was changed over a wide range keeping P_{mute} constant.

Figure 4 shows the variation of fitness with different P_{cross}. It is observed that the fitness of the chromosomes increase with increasing P_{cross}. It is noted that the function gives optimum values at a P_{cross} of $0.85 - 0.9$ and a further increase in P_{cross} does not increase the fitness, which is a similar trend reported elsewhere (10). A similar trend was observed with the variation of N_{pop}. A low initial population produced low fitness chromosomes. It was seen that for the present study, the average fitness value of the population converged to the optimum value at the 12th generation and a further increase in the number of generations is a waste of computational time.

CONCLUSION

A simple genetic algorithm was used to solve a single objective optimization problem of a RO system. It was found that a flowrate of 1495.30 mL/min and an overall flux of 0.0014 cm/s gave the optimum rejection of 0.93090. A study on the effect of varying the computational parameters on the solution was also studied. It was found that the algorithm used converged quickly and efficiently to an optimum solution. Moreover, a large number of solutions were generated which allows one to choose the required input variable set for the desired rejection. It was also observed that varying the computational parameters had a significant effect on the results generated. Mutation probability is the most sensitive parameter and its values should be kept low. The fitness was seen to increase with an increase in the crossover probability and the population number. These results match with those reported in the literature (9). GAs are seen to be extremely powerful in solving multi-objective problems (11) and a similar methodology as explained in the paper can be used to optimize a multi-objective RO system.

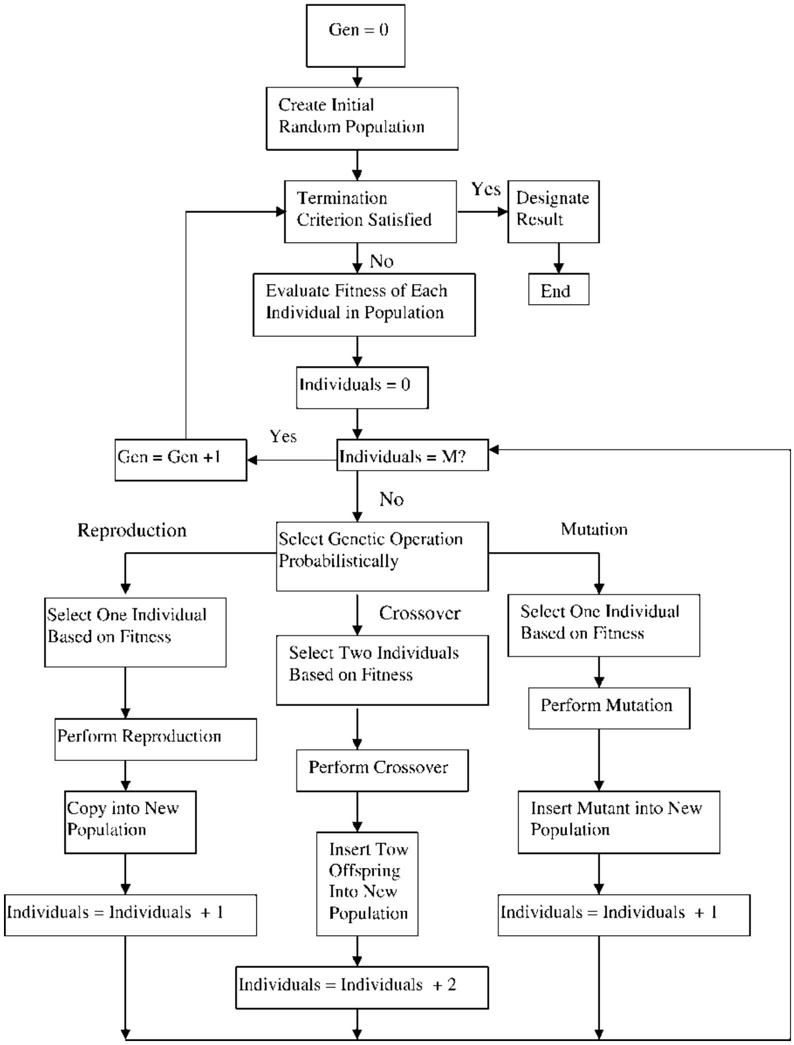
NOMENCLATURE

GA	Genetic Algorithm
$f(s_i(t))$	Fitness of the chromosome s_i
Jv	Volumetric flux across the membrane
k	Mass transfer coefficient
N_{pop}	Number of chromosomes in a population
ΔP	Pressure difference across the membrane
P_{cross}	Crossover probability
P_M	Local solute permeability per unit membrane thickness
P_{mute}	Mutation probability
Q	Feed flow rate

RO	Reverse osmosis
Ro	Observed rejection of solute
s_i	Chromosome i
σ	Reflection coefficient; 0 for no rejection; 1 for total rejection

APPENDIX I

Working of a Genetic Algorithm Program



APPENDIX II

No.	Jv(cm/s)	Q (mL/min)	Fitness
1.	0.00141	1347.50733	0.93084
String =1101111111-1011111011			
2.	0.00141	1295.89443	0.93081
String = 1101111111-1000101011			
3.	0.00141	850.14663	0.93046
String = 1101111111-1010101110			
4.	0.00141	1448.38710	0.93088
String = 1101111111-1100101111			
5.	0.00141	1263.04985	0.93079
String = 1101111111-1010110011			
6.	0.00141	894.72141	0.93051
String = 1101111111-1101111110			
7.	0.00141	1343.98827	0.93083
String = 1101111111-0101111011			
8.	0.00141	882.99120	0.93050
String = 1101111111-1000111110			
9.	0.00141	1343.98827	0.93082
String = 0101111111-0101111011			
10.	0.00141	1448.38710	0.93086
String = 0101111111-1100101111			
11.	0.00141	743.40176	0.93030
String = 0101111111-0101111010			
12.	**0.00141**	**1495.30792**	**0.93090**
String = 1101111111-1101111111			
13.	0.00141	893.54839	0.93051
String = 1101111111-0101111110			
14.	0.00141	1342.81525	0.93083
String = 1101111111-1001111011			
15.	0.00141	1448.38710	0.93086
String = 0101111111-1100101111			
16.	0.00141	1270.08798	0.93078
String = 0101111111-1101110011			
17.	0.00141	1263.04985	0.93079
String = 1101111111-1010110011			
18.	0.00141	1448.38710	0.93088
String = 1101111111-1100101111			
19.	0.00141	1333.43109	0.93083
String = 1101111111-1000111011			
20.	0.00141	1225.51320	0.93076
String = 0101111111-1010100011			

The first 10 bits of a string represent the value of Jv and the next 10 represent the value of Q.

REFERENCES

1. Androulakis, I.P. and Venkatasubramanian, V. (1991) A genetic algorithmic framework for process design and optimization. *Computers and Chemical Engineering*, 15: 217–228.

2. Fraga, E.S. and Senos Matias, T.R. (1996) Synthesis and optimization of a nonideal distillation system using a parallel genetic algorithm. *Computers and Chemical Engineering*, 20: S79–S84.
3. Garrard, A. and Fraga, E.S. (1998) Mass exchange network synthesis using genetic algorithms. *Computers and Chemical Engineering*, 22: 1837–1850.
4. Deb, K. (1995) *Optimization for Engineering Design: Algorithms and Examples*. Prentice-Hall, New Delhi.
5. Srinivas, N. and Deb, K. (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2: 221–248.
6. Gupta, R.R. and Gupta, S.K. (1999) Multiobjective optimization of an industrial nylon-6 semibatch reactor system using genetic algorithm. *Journal of Applied Polymer Science*, 73: 729–739.
7. Bhaskar, V., Gupta, S.K., and Ray, A.K. (2000) Multiobjective optimization of an industrial wiped film poly(ethylene terephthalate) reactor. *AIChE Journal*, 46: 1046–1058.
8. Rajesh, J.K., Gupta, S.K., Rangaiah, G.P., and Ray, A.K. (2000) Multiobjective optimization of steam reformer performance using genetic algorithm. *Industrial and Engineering Chemistry Research*, 39: 706–717.
9. Inamdar, S.V., Gupta, S.K., and Saraf, D.N. (2004) Multiobjective optimization of an industrial crude distillation unit using the elitist non-dominated sorting genetic algorithm. *Chemical Engineering Research and Design*, 82 (A5): 611–623.
10. Yuen, C.C., Aatmeeyata, Gupta, S.K., and Ray, A.K. (2000) Multiobjective optimization of membrane separation modules using genetic algorithm. *Journal of Membrane Science*, 176: 177–196.
11. Marriott, J. and Sorensen, Eva (2003) A general approach to modelling membrane modules. *Chemical Engineering Science*, 58: 4975–4990.
12. Marriott, J. and Sorensen, Eva (2003) The optimal design of membrane systems. *Chemical Engineering Science*, 58: 4991–5004.
13. Jain, S. and Gupta, S.K. (2004) Analysis of modified surface pore flow model with concentration polarization and comparison with Spiegler-Kedem model in reverse osmosis systems. *Journal of Membrane Science*, 232: 45–61.
14. Murthy, Z.V.P. (1996) *Studies on Reverse Osmosis Membrane Transport Models*, Chemical Engineering Department, Ph. D. Thesis, Indian Institute of Technology, New Delhi, India.
15. Murthy, Z.V.P. and Gupta, S.K. (1997) Estimation of mass transfer coefficient using a combined nonlinear membrane transport and film theory model. *Desalination*, 109: 39–49.
16. Murthy, Z.V.P. and Gupta, S.K. (1998) Thinfilm composite polyamide membrane parameters estimation for the phenol-water system by reverse osmosis. *Separation Science and Technology*, 33: 2541–2557.
17. Murthy, Z.V.P. and Gupta, S.K. (1999) Sodium cyanide separation and parameters estimation for reverse osmosis thinfilm composite polyamide membrane. *Journal of Membrane Science*, 154: 89–103.
18. Vaidya, S.Y., Simaria, A.V., and Murthy, Z.V.P. (2001) Reverse osmosis transport models evaluation: A new approach. *Indian Journal of Chemical Technology*, 8: 335–343.
19. Murthy, Z.V.P. and Vora, M.M. (2004) Prediction of reverse osmosis performance using artificial neural network. *Indian Journal of Chemical Technology*, 11: 108–115.
20. Soltanieh, M. and Gill, W.N. (1981) Review of reverse osmosis membrane and transport models. *Chemical Engineering Communications*, 12: 279–363.

21. Murthy, Z.V.P. and Gupta, S.K. (1996) Simple graphical method to estimate membrane transport parameters and mass transfer coefficient in a membrane cell. *Separation Science and Technology*, 31: 77–94.
22. Lamba, N., Murthy, Z.V.P., and Kumar, R. (2002) Membrane processing of an aqueous waste stream from catalyst manufacturing plant. *Separation Science and Technology*, 37: 191–202.
23. Koza, J.R. (1994) Genetic programming as a means for programming computers by natural selection. *Statistics and Computing (Historical Archive)*, 4: 87–112.
24. Guria, C., Bhattacharya, P.K., and Gupta, S.K. (2005) Multi-objective optimization of reverse osmosis desalination units using different adaptations of the non-dominated sorting genetic algorithm (NSGA). *Computers and Chemical Engineering*, 29: 1977–1995.
25. Ho, W.S.W. and Sirkar, K.K. (Eds) (1992) *Membrane Handbook*, Van Nostrand Reinhold, New York.
26. Maskan, F., Wiley, D.E., Johnston, L.P.M., and Clements, D.J. (2000) Optimal design of reverse osmosis module networks. *AIChE Journal*, 46: 946–954.
27. Wiley, D.E., Fell, C.J.D., and Fane, A.G. (1985) Optimization of membrane module design for brackish water desalination. *Desalination*, 52: 249–265.
28. Costa, C.B.B. and Filho, R.M. (2005) Evaluation of optimisation techniques and control variable formulations for a batch cooling crystallization process. *Chemical Engineering Science*, 60: 5312–5322.
29. Babu, B.V., Chakole, P.G., and Syed Mubeen, J.H. (2005) Multiobjective differential evolution (MODE) for optimization of adiabatic styrene reactor. *Chemical Engineering Science*, 60: 4822–4837.
30. Mitra, K. and Gopinath, R. (2004) Multiobjective optimization of an industrial grinding operation using elitist nondominated sorting genetic algorithm. *Chemical Engineering Science*, 59: 385–396.
31. Rajesh, J.K., Gupta, S.K., Rangaiah, G.P., and Ray, A.K. (2001) Multi-objective optimization of industrial hydrogen plants. *Chemical Engineering Science*, 56: 999–1010.
32. Tarafder, A., Rangaiah, G.P., and Ray, A.K. (2005) Multiobjective optimization of an industrial styrene monomer manufacturing process. *Chemical Engineering Science*, 60: 347–363.
33. Silva, C.M. and Biscaia, E.C., Jr. (2003) Genetic algorithm development for multi-objective optimization of batch free-radical polymerization reactors. *Computers and Chemical Engineering*, 27: 1329–1344.
34. Tarafder, A., Lee, B.C.S., Ray, A.K., and Rangaiah, G.P. (2005) Multiobjective optimization of an industrial ethylene reactor using a nondominated sorting genetic algorithm. *Industrial and Engineering Chemistry Research*, 44: 124–141.
35. Mitra, K., Majumdar, S., and Raha, S. (2004) Multiobjective optimization of a semibatch epoxy polymerization process using the elitist genetic algorithm. *Industrial and Engineering Chemistry Research*, 43: 6055–6063.
36. Bhaskar, V., Ray, A.K., and Gupta, S.K. (2000) Applications of multiobjective optimization in chemical engineering. *Reviews in Chemical Engineering*, 16: 1–54.
37. Deb, K. (2001) *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley: Chichester, UK.
38. Coello Coello, C.A., Van Veldhuizen, D.A., and Lamont, G.B. (2002) *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer: New York.
39. Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimisation, and Machine Learning*. Addison-Wesley: Reading, MA.

40. Guria, C., Verma, M., Mehrotra, S.P., and Gupta, S.K. (2005) Multi-objective optimal synthesis and design of froth flotation circuits for mineral processing using the jumping gene adaptation of genetic algorithm. *Industrial and Engineering Chemistry Research*, 44: 2621–2633.
41. Sarkar, D. and Modak, J.M. (2005) Pareto-optimal solutions for multi-objective optimization of fed-batch bioreactors using nondominated sorting genetic algorithm. *Chemical Engineering Science*, 60: 481–492.
42. Singh, M.K., Banerjee, T., and Khanna, A. (2005) Genetic algorithm to estimate interaction parameters of multicomponent systems for liquid-liquid equilibria. *Computers and Chemical Engineering*, 29: 1712–1719.
43. Wongrat, W., Srinophakun, T., and Srinophakun, P. (2005) Modified genetic algorithm for nonlinear data reconciliation. *Computers and Chemical Engineering*, 29: 1059–1067.
44. Styczynski, M.P. and Stephanopoulos, G. (2005) Overview of computational methods for the inference of gene regulatory networks. *Computers and Chemical Engineering*, 29: 519–534.
45. Katare, S., Bhan, A., Caruthers, J.M., Delgass, W.N., and Venkatasubramanian, V. (2004) A hybrid genetic algorithm for efficient parameter estimation of large kinetic models. *Computers and Chemical Engineering*, 28: 2569–2581.
46. Grosman, B. and Lewin, D.R. (2004) Adaptive genetic programming for steady-state process modeling. *Computers and Chemical Engineering*, 28: 2779–2790.
47. Ahmad, A.L., Azid, I.A., Yusof, A.R., and Seetharamu, K.N. (2004) Emission control in palm oil mills using artificial neural network and genetic algorithm. *Computers and Chemical Engineering*, 28: 2709–2715.
48. Tarca, L.A., Grandjean, B.P.A., and Larachi, F. (2004) Embedding monotonicity and concavity in the training of neural networks by means of genetic algorithms: Application to multiphase flow. *Computers and Chemical Engineering*, 28: 1701–1713.
49. Leboreiro, J. and Acevedo, J. (2004) Processes synthesis and design of distillation sequences using modular simulators: A genetic algorithm framework. *Computers and Chemical Engineering*, 28: 1223–1236.
50. Sarkar, D. and Modak, J.M. (2004) Optimization of fed-batch bioreactors using genetic algorithm: Multiple control variables. *Computers and Chemical Engineering*, 28: 789–798.
51. Elliott, L., Ingham, D.B., Kyne, A.G., Mera, N.S., Pourkashanian, M., and Wilson, C.W. (2005) Reaction mechanism reduction and optimization using genetic algorithms. *Industrial and Engineering Chemistry Research*, 44: 658–667.
52. Balasubramanian, P., Bettina, S.J., Pushpavanam, S., and Balaraman, K.S. (2003) Kinetic parameter estimation in hydrocracking using a combination of genetic algorithm and sequential quadratic programming. *Industrial and Engineering Chemistry Research*, 42: 4723–4731.
53. Suggala, S.V. and Bhattacharya, P.K. (2003) Real coded genetic algorithm for optimization of pervaporation process parameters for removal of volatile organics from water. *Industrial and Engineering Chemistry Research*, 42: 3118–3128.
54. Elliott, L., Ingham, D.B., Kyne, A.G., Mera, N.S., Pourkashanian, M., and Wilson, C.W. (2003) Multiobjective genetic algorithm optimization for calculating the reaction rate coefficients for hydrogen combustion. *Industrial and Engineering Chemistry Research*, 42: 1215–1224.
55. Sarkar, D. and Modak, J.M. (2003) Optimisation of fed-batch bioreactors using genetic algorithms. *Chemical Engineering Science*, 58: 2283–2296.

56. Lavric, V., Iancu, P., and Plesu, V. (2005) Genetic algorithm optimisation of water consumption and wastewater network topology. *Journal of Cleaner Production*, 13: 1395–1405.
57. Kasat, R.B. and Gupta, S.K. (2003) Multi-objective optimization of an industrial fluidized-bed catalytic cracking unit (FCCU) using genetic algorithm (GA) with the jumping genes operator. *Computers and Chemical Engineering*, 27: 1785–1800.
58. Hinchliffe, M.P. and Willis, M.J. (2003) Dynamic systems modelling using genetic programming. *Computers and Chemical Engineering*, 27: 1841–1854.
59. Alvarenga, G.B. and Mateus, G.R. (2004) Hierarchical tournament selection genetic algorithm for the vehicle routing problem with time windows. *Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*, : 410–415.
60. Emereev, A.V. (1999) Modeling and Analysis of Genetic Algorithm with Tournament Selection. In *Lecture Notes In Computer Science: Selected Papers from the 4th European Conference on Artificial Evolution*. Fonlupt, C., Hao, J-K., Lutton, E., Ronald, E.M.A. and Schoenauer, M. Eds., Springer-Verlag: London; Vol. 1829, 84–94.