0273-2289/02/102–103/0119/$12.50

Optimization of Media by Evolutionary Algorithms for Production of Polyols

S. V. Patil,[1] V. K. Jayaraman,[2] and B. D. Kulkarni*,[2]

[1]Department of Alcohol Technology, Vasantdada Sugar Institute, Pune 412 307, India; and [2]National Chemical Laboratory, Chemical Engineering Division, Pune 411 008, India, E-mail: bdk@ems.ncl.res.in

Abstract

Biotransformation of sucrose-based medium to polyols has been reported for the first time using osmophilic yeast, *Hansenula anomala*. A new, real coded evolutionary algorithm was developed for optimization of fermentation medium in parallel shake-flask experiments. By iteratively employing the nature-inspired techniques of selection, crossover, and mutation for a fixed number of generations, the algorithm obtains the optimal values of important process variables, namely, inoculum size and sugar, yeast extract, urea, and $MgSO_4$ concentrations. Maximum polyols yield of 76.43% has been achieved. The method is useful for reducing the overall development time to obtain an efficient fermentation process.

Index Entries: Media optimization; polyols yield; genetic algorithm.

Introduction

Production of glycerol and related polyols by the fermentation route, particularly from renewable raw materials, is assuming increasing importance in view of escalating prices of petroleum crude and shortages of fats and oils. Fermentative production of these chemicals from raw materials such as sucrose or sugarcane molasses using osmophilic yeast holds good promise. Osmophilic yeast species of *Saccharomyces*, *Pichia*, *Debaryomyces*, *Candida*, and *Torulopsis* can produce a mixture of polyols. Depending on the species used, it has been shown that they can produce glycerol, erythritol, arabitol, xylitol, mannitol, and so on in relatively good yields. Glycerol is an industrially vital chemical with wide applications. India has continued to import glycerol for the past several years. Mannitol is available from

*Author to whom all correspondence and reprint requests should be addressed.

other sources, but no synthetic route has been reported for erythritol and arabitol. Osmophilic yeast produces glycerol and other polyols essentially under aerobic conditions. These yeasts can convert as much as 60% of sugars to polyols *(1)*. However, their fermentation rates are very slow. The yield of polyols can be influenced by the conditions of growth and production. Of these, the composition of the medium, especially the choice of carbon and nitrogen sources, has the greatest effect on the yields and rate of fermentation. The concentration of sugars affects the yield of polyols *(2)*, and fermentation studies with osmophilic yeasts have been carried out at a concentration of 10–40% *(3,4)*.

Yeast extract and urea have been used as the major nitrogen sources for the production of polyols using osmophilic yeasts *(5)*. Yeast extract also supplies an adequate quantity of growth factors required by the yeast. Hanjay et al. *(6)* found an optimum yield of yeast extract concentration of 0.5%. Above this level, the yield of glycerol reduced drastically. Onishi et al. *(7)* and Spencer and Shu *(8)* have reported similar observations. Most researchers have also used small concentrations of urea and magnesium sulfate as a supplement to yeast extract, obtaining improved yield *(9)*. These studies indicate that the level of sugar and nitrogen in the medium requires careful adjustment in order to obtain optimum results.

Although glucose has been used extensively as a carbon source, sucrose has not been used. In a country such as India, with abundant availability of renewable and cheap raw materials such as sucrose and sugarcane molasses, it is imperative to develop a fermentation process based on utilization of these raw materials. In the present investigation, we have used evolutionary algorithms for optimization of major components of the fermentation media that can affect the yield of polyols using the osmophilic yeast *Hansenula anomala*.

Materials and Methods

Maintenance of Culture

Osmophilic yeast *H. anomala* NCIM-3341 was selected *(10)* and maintained on MGYP agar slants at 4°C and subcultured every month.

Preparation of Inoculum

The composition of the inoculum medium was as follows (% [w/v]): 10% sucrose, 0.25% yeast extract, 0.1% urea at pH 6.0. An Erlenmeyer flask (500 mL) containing 100 mL of medium was sterilized in an autoclave at 15 psi for 20 min. A loopful of culture was transferred from the agar slant to the inoculum medium and incubated at 30°C for 24 h on a rotary shaker at 180 rpm. The grown cell mass was centrifuged at 2225*g* at 4°C for 20 min in sterile, stoppered centrifuge tubes, and the residual wet-cell mass was used to inoculate the experimental flasks.

Fermentation

For optimization studies, we selected inoculum size, initial sugar, yeast extract, urea, and $MgSO_4$ as variables. The range of inoculum considered was 2–8 mL. The sucrose concentrations were kept between 20 and 40% (w/v); yeast extract between 0.125 and 0.5% (w/v), urea between 0.1 and 0.4% (w/v), and $MgSO_4 \cdot 7H_2O$ between 0.01 and 0.04% (w/v). To improve the oxygen-transfer capacity of the shake flask, only 40 mL of fermentation medium in a 500-mL Erlenmeyer flask was used. The fermentation medium was sterilized in the same manner as inoculum medium. After inoculation, the flasks were incubated on a rotary shaker at 30°C and 180 rpm. At a specified time, depending on the initial sugar concentration, 3-mL samples were removed aseptically and centrifuged at 2225*g* for 20 min. The residual cell mass was used for estimation of dry cell weight, and the clear supernatant was used for estimation of residual sugars and total polyols.

Analytical Methods

Cell mass was estimated gravimetrically as dry cell weight. Total reducing sugars after inversion were estimated spectrophotometrically with the alkaline copper reagent of Somogyi *(11)* and arsenomolybdate reagent of Nelson *(12)*. Total polyols was determined by the spectrophotometric method of Lambert and Neish *(13)*.

Optimization of Media

Optimization of media can be carried out in several ways. The simplest method is the one-variable-at-a-time approach. This approach is, however, extremely inefficient in locating a true optimum when interaction effects are present. Multivariable methods for optimization of media are used to overcome problems with interaction effects. Two categories of methods, the sequential design approach and simultaneous statistical design approach, are used for this purpose. In the sequential method, experiments are successfully performed in the direction of improving the performance index until the optimum is reached. Although several methodologies exist for sequential design, the simplex approach has been the most widely used method. It can handle many variables and starts with a design consisting of a simplex with $n + 1$ dimensional space. Subsequent improvements are obtained by reflection. The other popular method is the simultaneous statistical design methodology. These methods allow simultaneous and efficient variation of all media components *(14–17)*. Both single- and two-level factorial experiments have been designed for optimization of media.

The common goal in the bioprocess industry is to reduce the overall development time. A critical part of this development is to fix an equation for the manufacturing process. Because of the competitiveness in the market this has to be done in a constrained time frame. It is therefore imperative to optimize the process rapidly and maximize the information from fewer

numbers of experiments. Recently, evolutionary methods such as genetic algorithms, simulated annealing, tabu search, and ant colony optimization methods are becoming increasingly popular. Artificial neural networks have also been widely employed in various bioprocess engineering applications including optimization of media *(18–20)*. In the present study, we used the genetic algorithm procedure for optimization of culture media with a view to maximize the polyol yield.

Genetic Algorithms: An Overview

Genetic algorithms are stochastic search techniques that mimic the process of natural selection *(21,22)*. They have the inherent ability of simple representation to encode highly complicated structures and use simple transformations to improve such structures to reach optimal solution. In the natural evolution, each species searches for beneficial adaptation in a complicated and dynamically differing environment. The knowledge gained in the search is embodied in the chromosome of its members. Random mutation, numerical creeping, and crossover are the genetic operators, which alter the chromosomal makeup. Conventional genetic algorithm as an optimization problem basically involves five components: a chromosomal representation of solutions, an evaluation function mimicking the role of the environment, rating solutions in terms of their current fitness, genetic operators that alter the compositions of children during reproduction, and values of the parameters that the algorithm uses (such as population size, probabilities of applying genetic operators).

Recently, several improvements in genetic algorithms over the basic algorithm have been proposed. These include problem-specific crossover operators, elitist selection strategy, and diversity measures. Problem-specific crossover operators improve algorithm efficiency in the speed and quality of obtaining global solutions. Elitism ensures that the best solution from the previous generation is preserved in the current generation by replacing the weakest new individual. Incest prevention is used to maintain genetic diversity. This is usually implemented by allowing mating to occur only if the Hamming distance is above an acceptable threshold. Genetic algorithms have the following features, which make them different from the conventional (deterministic) algorithms:

1. They work on a population of points instead of a single point.
2. They work on the representation rather than on the variables.
3. They facilitate a global optimum.
4. They are inherently parallel.
5. They do not require derivatives for fitness function.

Genetic algorithms have been used in a wide variety of fields for optimization purposes. One of the earliest applications of genetic algorithms in bioprocess engineering was for obtaining optimal media composition *(23)*. Genetic algorithms were found to be able to bring out highly optimized media experimentally even in noisy systems. Matsuura et al.

(24) calculated the optimal trajectories for fermentation processes using genetic algorithms. A genetic algorithm was used to estimate parameters in fermentation dynamic models, and its effectiveness was tested with three typical models: an equation for biomass determination, an equation for substrate consumption, and an equation for products. The estimation results obtained were better than the Marquardt optimization procedure. Weuster-Botz et al. *(26)* developed a fed-batch process for the production of L-lysine with *Corynebacterium glutamicum* in a stirred-tank reactor. Optimizations of medium composition in parallel shake flasks were done using genetic algorithms. The other relevant applications of genetic algorithms, which are useful in bioprocess optimization, can be found in refs. *26–29*.

The conventional binary coded genetic algorithm having discrete search space has been very successful in obtaining solutions for the combinatorial optimization problems. However, in solving problems with continuous search space, coding of the real valued variables in finite length strings causes a number of difficulties: inability to obtain arbitrary precision in the obtained solution, fixed mapping of the problem variable, inherent hamming difficulty associated with binary coding, and processing of Holland's schemata in continuous space *(30)*. Several real coded genetic algorithms were used in the recent past to overcome these difficulties *(31)*. We have employed one such strategy for optimizing the media variables in the production of polyols. We have prepared our own software, and the details of our algorithm for optimizing the media components to maximize polyol yield are discussed next.

The Algorithm

The algorithm's steps are initialization, selection, crossover, and mutation, which are discussed next.

Step 1: Initialization

The algorithm is initiated by creating a population having a certain size. This is done by randomly generating solutions having the values of the variables $x1$ to $x5$ (representing inoculum size, sugar concentration, yeast extract concentration, urea concentration, and magnesium sulfate concentration, respectively) within the allowable ranges specified. After generating the random populations, we have to assign a fitness for all the members of the population. The fitness for our problem is the yield of polyols.

Step 2: Selection

For the sake of convenience, the population members are sorted in the order of decreasing fitness (objective function value) with a view to identifying the strong and weak members of the population. The strong members are called parents and are selected to take part in crossover operations. In these operations, they are suitably combined to produce children. These children then replace the existing weak members. The population of the succeeding generation thus consists of the strong parents and the children produced by genetic combination of the strong parents. In our work, we have termed the top 50% of the population as parents.

Step 3: Crossover

In the crossover operation, a certain number of the weaker populations are removed. These are replaced by new solutions, which are created by genetic-like combinations of fitter solutions. Crossover operations are done as follows: A fitter solution, i.e., a parent from the top half of the population, is selected randomly (weaker solutions are not selected for crossover operation), and the value of the first variable ($x1$) of the selected parent is set as the value of the first variable of the newly created solution. The value of the second variable ($x2$) of the new solution is set to the value of the corresponding variable of the randomly chosen second parent with a probability equal to the crossover probability, *CP*. Subsequent values of the variables of the new solution are chosen by the same procedure. Thus, for $CP = 1$, each variable of the new solution has a different parent, and for $CP = 0$, the new solution is identical to a randomly selected single parent. After crossover operation is performed, the new solution replaces the weakest solution in the population. Similarly, other weaker solutions are replaced by repeating the process.

Step 4: Mutation

The crossover operator is mainly responsible for the search aspect of genetic algorithms, but a mechanism is needed to keep and maintain the algorithm's diversity; otherwise, the algorithm very quickly converges to the local optimum and never reaches the global optimum. The mutation operator is used for this purpose. There are several ways in which mutation can be performed in the real coded algorithm. We used the following method for mutation: Select a variable randomly and add a perturbation randomly (positive or negative direction is chosen with equal probability) with the specified mutation probability (*MP*). If a population member at a given generation number t is $\bar{x}$ $(x_1, x_2, \ldots\ldots, x_m)$ and if x_i is selected for mutation, it is replaced with

$$x_i^{new} = \begin{cases} x_i + \Delta(t, X_i^u - x_i) \\ \text{or} \\ x_i - \Delta(t, x_i - X_i^l) \end{cases} \qquad \text{Both with equal probabilities}$$

in which X_i^l and X_i^u are, respectively, the lower and upper bounds for the variable x_i.

The function $\Delta(t, y)$ returns a value in the range $[0, y]$ in which y is any real number. The function is designed such that the probability of $\Delta(t, y)$ being close to zero increases as t increases and is defined as

$$\Delta(t, y) = y \cdot (1 - r^{[1 - (t/T)]^B})$$

in which r is a random number from [0, 1], T is the maximum number of generations, and B is the nonlinearity parameter. Each variable x_i of each population member is mutated as shown with a probability equal to the mutation probability, *MP*.

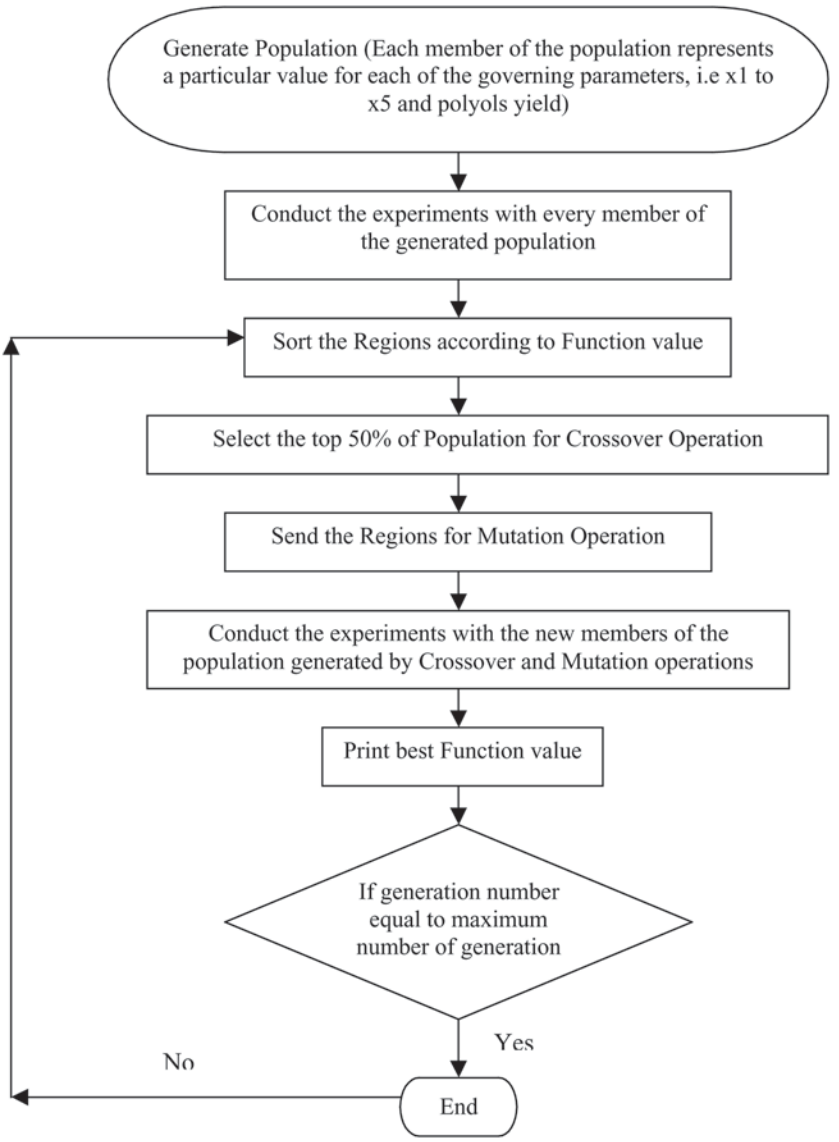

Fig. 1. Flow chart of real coded genetic algorithm.

The last three steps—selection, crossover, and mutation—are repeated for a fixed number of generations. These two operations are carried out repeatedly until the required number of generations is completed. The algorithm is given in Fig. 1.

Results and Discussion

As already mentioned, optimization of medium was carried out in parallel shake-flask experiments, using an evolutionary algorithm. To begin

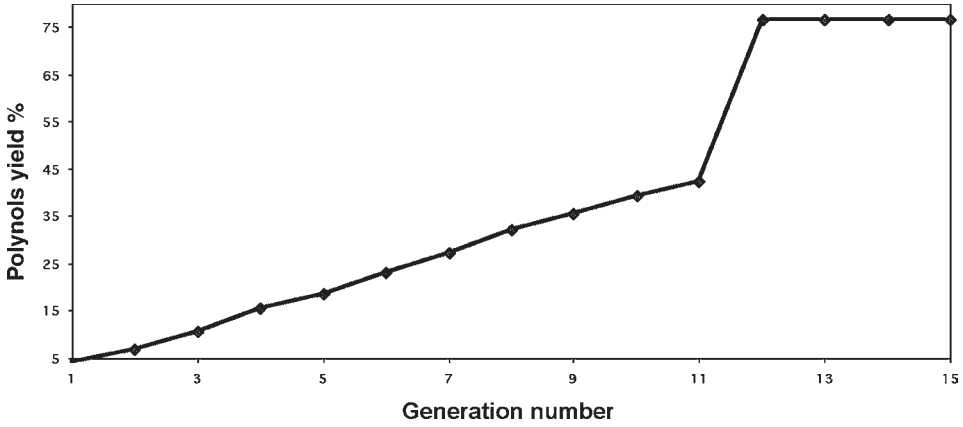

Fig. 2. Optimization of medium for polyols production with real coded genetic algorithm.

Table 1
Optimized Medium Composition and Maximum Polyols Yield

Inoculum size (mL)	Sugar (%)	Yeast extract (%)	Urea (%)	$MgSO_4 \cdot 7H_2O$	Polyol yield (%)
2.05	30.05	0.125	0.11	0.0256	76.43

Table 2
Genetic Algorithm Parameters

Population	Weaker regions removed per generation	Crossover possibility	Mutation possibility	Maximum no. of generations
20	6	0.8	0.1	15

with, the genetic algorithm randomly generates 20 sets of governing parameters. With these 20 sets of values experiments are performed. Each experiment was performed in duplicate to check the reproducibility. It was found that the results in terms of polyol yield (based on sugars utilized) varied with a maximum of 7.5%. The average yield values of the duplicate experiments are reported here. The 20 different experimental values of polyol yield were transferred to the algorithm. The algorithm used this information to combine the parameter values of the best 10 experiments and suggested six new sets of parameter values (with the help of crossover and mutation operations) for performing experiments in the next generation. This procedure was repeated for 15 generations. The progress of the algorithm in obtaining the maximum in the polyol yield is shown in Fig. 2. It can be seen that the algorithm moves steadily toward the maximum with

progression of the generations. After the twelfth generation, we observed that the maximum yield of the polyols did not change. Maximum polyol yield obtained was 76.43%. The best process variable concentrations to achieve this yield are given in Table 1. This optimum combination was obtained by doing a total of 104 experiments. The algorithm's parameters are given in Table 2.

Conclusion

Experimental optimization of medium composition in parallel shake-flask experiments using a real coded genetic algorithm was successfully carried out. To our knowledge, this is the first report of polyol production from sucrose-based medium. The optimum yield of polyols obtained by this procedure was 76.43%. The number of experiments required for obtaining the optimum performance was much less than for the conventional approach. The method is very useful for reducing overall development time and for maximizing the information from a small number of experiments.

References

1. Spencer, J. F. T. and Sallans, H. R. (1956), *Can. J. Microbiol.* **2,** 72–79.
2. Onishi, H. (1960), *Bull. Agric. Chem. Soc. Japan* **24,** 131–140.
3. Spencer, J. F. T., Roxburgh, J. M., and Sallans, H. R. (1957), *J. Agric. Food. Chem.* **5,** 64–67.
4. Groleau, D. (1995), *Biotech. Lett.* **17(3),** 315–320.
5. Button, D. K., Garver, J. C., and Hanjay, G. J. (1966), *Appl. Microbiol.* **14,** 292–294.
6. Hanjay, G. J., Hendershot, W. F., and Peterson, W. H. (1960), *Appl. Microbiol.* **8,** 5–11.
7. Onishi, H., Saito, N., and Koshiyama, I. (1961), *Agri. Biol. Chem. (Toyko)* **25,** 124–130.
8. Spencer, J. F. T. and Shu, P. (1957), *Can. J. Microbiol.* **3,** 559–567.
9. Ramachandran, N. and Sulebele, G. (1979), *Ind. J. Microbiol.* **9,** 136–141.
10. Patil, S. V. and Borawake, S. D. (1989), in *Proceedings of Thirty-Ninth Annual Convention of DSTA*, Ravetkar, D. M., ed., The Deccan Sugar Technologists' Association, Pune, India, pp. B-1–B-6.
11. Somogyi, M. (1944), *J. Biol. Chem.* **160,** 61–66.
12. Nelson, N. (1945), *J. Biol. Chem.* **163,** 375–380.
13. Lambert, N. and Neish, A. C. (1950), *Can. J. Res.* **28(sect. B)** 83–90.
14. Achary, A., Hariharan, K. A., Bandhyopadhyaya, S., et al. (1997), *Biotechnol. Bioeng.* **55,** 148–154.
15. Hounsa, C. G. (1996), *Appl. Microbiol. Biotechnol.* **45,** 764–770.
16. Lee, S. L. and Chen, W. C. (1997), *Enzyme Microb. Technol.* **21,** 436–440.
17. Zhu, Y., Knol, W., Smits, J. P., et al. (1996), *Enzyme Microb. Technol.* **18,** 108–112.
18. Glassey, J., Ignova, M., Ward, A. C., et al. (1997), *J. Biotech.* **52,** 201–205.
19. Montague, G. and Morris, A. (1994), *TIBTECH* **12,** 312–323.
20. Glassey, J., Montague, G. A., Ward, A. C., et al. (1994), *Biotechnol. Bioeng.* **44,** 397–405.
21. Holland, J. H. (1986), in *Induction: Process of Inference,* Learning and Discovery, MIT Press.
22. Goldberg, D. E. (1989), in *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley.
23. Freyer, S., Weuster-Botz, D., and Wandrey, C. (1992), *Bioengineering (Graefelfing, Fed. Repub. Ger.)* **8(5+6),** 16–25.
24. Matsuura, K., Shiba, H., Nunokawa, Y., et al. (1993), *Seibutsu Kogaku Kaishi.* **71(3),** 171–178.
25. Weuster-Botz, D., Kelle, R., Frantzen, M., et al. (1997), *Biotechnol. Prog.* **13(4),** 387–393.

26. Bohling, H. and Voss, H. (1997), *Bioforum Int.* **1(2),** 86, 88–90.
27. Beste, H., Fackeldey, M., Willems, M., et al. (1997), *Chem. Eng. Technol.* **20(6),** 403–413.
28. Zuzek, M., Friedrich, J., Cestnik, B., et al. (1996), *Biotechnol. Tech.* **10(12),** 991–996.
29. Weuster-Botz, D. and Wandrey, C. (1995), *Process Biochem.* **30(6),** 563–571.
30. Deb, K. and Agrawal, R. B. (1955), *Complex Systems* **9,** 115–148.
31. Michalewicz, Z. (1994), in *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, New York, Chapter 5, p. 95.