Applied Math. & Information Sciences Letters
An International Journal

Optimum culture conditions for 1,3-Propanediol production from Crude Glycerol using metaheuristic algorithms

Ali Sadollah, Mohsen Naeim Hassan Mohamed, Ardeshir Bahreininejad

Faculty of Engineering, University of Malaya, 50603, Kuala Lumpur, Malaysia

Email: mohsenegypt@um,edu.my; el_khozami@rocketmail.com.

Abstract: 1,3-Propanediol (1,3-PD) is one of the important products used in chemical industry, in particular for polyesters production. Using crude glycerol for producing 1,3-PD is a suitable solution from the economical as well as ecological point of view. In this paper, the optimum conditions of culture for 1,3-PD include of 1,3-PD concentration (g/L) as response was studied using metaheuristic algorithms including genetic algorithms (GAs), simulated annealing (SA), particle swarm optimization (PSO), and imperialist competitive algorithm (ICA). The process parameters studied were: the value of pH (6-8), $(NH_4)_2SO4$ (2-10 g/L), crude glycerol (10-50 g/L), cultivation time (6-14 h), and temperature (31° -43°). The obtained results from metaheuristic algorithms were compared with the results from the response surface methodology (RSM) in literature. The results obtained by metaheuristic algorithms offer higher 1,3-PD concentration compared to the results given by RSM.

Keywords: 1,3-Propanediol, Crude glycerol, Genetic algorithms, Particle swarm optimization, Simulated annealing, Imperialist competitive algorithm, Response surface methodology

1 Introduction

1,3-Propanediol (1,3-PD) has numerous applications for the production of polymers, cosmetics, foods, lubricants, and medicines [1]. 1,3-PD has gained commercial attention as an important monomer to synthesize a new type of polyester, polytrimethylene terephthalate (PTT), which has excellent properties for use by textile and fiber industries [2].

However, traditional chemical methods to produce 1,3-PD are difficult and have low selectivity. Consequently, its high price has hindered the utilization of 1,3-PD in polymer industries. Interest in investigating biochemical processes to produce 1,3-PD originates from the idea of utilizing inexpensive renewable resources .

The microbial conversion of glycerol to 1,3-PD is particularly attractive in that the process is relatively easy and does not generate toxic by-products [3-5]. Increasingly, the bulk production of glycerol as a by-product of the biodiesel and soap industries has made it a low cost, renewable resource .

Considering the environmental benefits, production of 1,3-PD through biochemical processes appears to be an attractive alternative to chemical synthesis [6-8]. The traditional “one-at-a-time” optimization strategy is relatively simple, and the individual effects of culture condition factors can be graphically depicted without the need for statistical analysis.

Unfortunately, it frequently fails to locate the region of optimum response in such procedures, thus, an efficient approach is required for optimization. A combination of factors generating a certain optimum response can be identified through factorial design and the use of response surface methodology (RSM) [9-11].

The RSM is an empirical modeling approach which uses polynomials as local approximations to the true input or output relationship. The objective is to optimize the response (output variables) that is influenced by several independent variables (input variables). The advantage of the RSM is to avoid experiment repetitions for experiments with multiple factors [12,13].

Zheng et al. [14] recently demonstrated statistically optimized culture conditions for 1,3-PD production using pure glycerol by a Klebsiella pneumoniae strain via central composite design (CCD). Accordingly, Baek-Rock et al. [11] applied RSM to optimize culture conditions for 1,3-PD using crude glycerol derived from the biodiesel

industry which the simultaneous effect of five independent variables including crude glycerol, $(NH_4)_2SO4$, pH, cultivation time, and temperature was investigated.

Among optimization methods, metaheuristic optimization algorithms have shown their capabilities for finding the near-optimal solution to the numerical real-valued test problems for which exact and analytical methods may not be able to produce within a reasonable computation time, especially when the global minimum is surrounded by many local minima [15,16].

These algorithms are usually devised by observing phenomena happening in nature, like genetic algorithms (GAs), simulated annealing (SA), particle swarm optimization (PSO), ant colony optimization (ACO), and so forth [17-19].

This paper aims at extending the research on production of 1,3-PD (g/L) using metaheuristic algorithms and offer improved results. The remaining of this paper is organized as follows: in Section 2, the 1,3-PD from Crude Glycerol is introduced and the optimizations methods including DOE, GA, SA, PSO, and ICA are described. In Section 3, the results obtained from various optimization methods are compared and discussions are presented in details. Finally, conclusions are given in Section 4.

2 Materials and Methods

2.1. 1,3-Propanediol (1,3-PD) from Crude Glycerol

2.1.1. Crude glycerol

Crude glycerol (purity 42%, w/w) was obtained from a biodiesel-producing company (Kaya Energy, Korea). All other chemicals used were commercially obtained and were of analytical grade [11].

2.1.2. Microorganism and Culture Conditions

K. Pneumoniae ATCC 700721, purchased from the American Type Culture Collection (ATCC, USA), was grown on preculture medium containing: 20 g/L glycerol, 3.4 g/L K_2HPO_4, 1.3 g/L KH_2PO_4, 0.2 g/L $MgSO_4$, 0.002 g/L $CaCl_2$ $2H_2O$, 1 g/L yeast extract, 1 *mL* Fe solution [5 g/L $FeSO_4$ $7H_2O$ and 4 *mL* HCl (37%, w/v)], and 1 *mL* trace element solution [70 mg/L $ZnCl_2$, 100 mg/L $MnCl_2$ $4H_2O$, 60 mg/L H_3BO_3, 200 mg/L $CoCl_2$ $4H_2O$, 20 mg/L $CuCl_2$ $2H_2O$, 25 mg/L $NiCl_2$ $6H_2O$, 35 mg/L Na_2MoO_4 $2H_2O$, 4 *mL* HCl (37%, w/v)]. The seed cells for the fermentation were prepared in a 250 *mL* flask containing 50 *mL* preculture medium. The flasks were incubated at 37°C for 12 hours and were subsequently inoculated into the fermentor at 2% (v/v) concentration. The batch cultivations were conducted in a 5 *L* stirred-vessel fermentor (Kobiotech Co., Ltd., Korea) containing 2 *L* fermentation medium. The pH was controlled by automatic addition of 2 *M* HCl and NH4Cl. The agitation rate was adjusted to 200 rpm [11].

2.2. Design of Experiments

Design of experiments (DOE) is a tool to optimize the planning of experimental research. The role of DOE is to estimate the effect of several variables separately, simultaneously or as combinations [20]. Box and Draper [21] claimed response surface methodology (RSM) is one of statistical methods for modeling and analyzing the relationships between several variables and response variables.

The effects of culture condition parameters in the simulation of production 1,3-PD for obtaining maximum concentration of 1,3-PD have been studied using DOE method [11]. The DOE method selected was RSM coupled with the central composite design (CCD) using the Design-Expert software. The process parameters selected for this study were the value of pH (6-8), $(NH4)_2SO4$ (2-10 g/L), crude glycerol (10-50 g/L), cultivation time (6-14 *h*), and temperature (31° -43°).

Accordingly, the response parameter was 1,3-PD production (concentration) (g/L). Baek-Rock et al. [11] adopted RSM to construct the appropriate objective functions for 1,3-PD concentration (g/L). The quadratic model equation with coded factors (between -2 and 2) represents response for 1,3-PD concentration (g/L) which is given as follows [11]:

$$\begin{aligned} Y = {} & 12.06+0.69X_1+0.57X_2+1.49X_3+0.98X_4-0.08X_5-0.54X_1^2 \\ & +0.03X_2^2-0.91X_3^2-0.36X_4^2-1.87X_5^2+0.18X_1X_2+0.19X_1X_3 \\ & -0.23X_1X_4-0.12X_1X_5+0.30X_2X_3-0.25X_2X_4-0.07X_2X_5+ \\ & 0.05X_3X_4-0.02X_3X_5+0.12X_4X_5 \end{aligned} \quad (1)$$

where X_1, X_2, X_3, X_4, and X_5 are the value of pH, $(NH_4)_2SO_4$, crude glycerol, cultivation time, and temperature, respectively. Y is the response which is concentration of 1,3-PD (g/L).

2.3. Genetic Algorithms

Genetic algorithms (GAs) are members of a collection of methodologies known as evolutionary computation. These techniques are based on the

principals of natural selection and evolution processes that are met in nature. The efficiency of the numerous evolutionary algorithms in comparison to other heuristic techniques has been tested in both generic [22-24] and engineering design [25] problems .

Through these tests, the GAs are identified as robust heuristic tools capable of delivering efficient and robust solutions to diverse design problems. In general, a GA begins its search with a population of random individuals. Each member of the population possesses a chromosome which is comprised of genes. A gene may take on one of two allele values, either a 1 or a 0. All genes within a chromosome are assembled as a binary string of ones and zeros, often with distinct segments .

The first step in creating an offspring population for GAs is to construct a mating pool. The mating pool contains *N* individuals which are copied from the parent population that will be utilized to create the finished offspring population. To create the mating pool, a crowded tournament selection operator may be utilized .

To begin, two solutions from the parent population are selected at random to compete in a tournament in order to construct the mating pool. A copy of the winner of the tournament is kept in the mating pool. This process is repeated until each solution in the parent population has competed twice and the *N* spots in the mating pool have been filled. The resulting mating pool contains more copies of the more desirable solutions in the parent population and fewer copies of the less desirable solutions from the parent population .

With the mating pool complete, the task of constructing the offspring population can commence. To generate two offspring solutions, one begins by selecting two individuals from the mating pool at random. Once selected, the two individuals undergo crossover and mutation operators to create two offspring solutions which are placed in the offspring population [26].

2.4. Particle swarm optimization

Particle swarm optimization (PSO) is an evolutionary computation technique for solving global optimization problems developed by Kennedy and Eberhart [27]. It is a computation technique through individual improvement plus population cooperation and competition which is based on the simulation of simplified social models, such as bird flocking, fish schooling and the swarm theory.

Researchers found that the synchrony of animal's behavior was through maintaining optimal distances between individual members and their neighbors. Thus, velocity plays the important role of adjusting each other for the optimal distances [28]. The PSO algorithm exhibits common evolutionary computation attributes including initialization with a population of random solutions and searching for optima by updating generations.

Potential solutions, called 'birds' or 'particles', are then "flown" through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness) it has achieved so far. This value is called 'pBest'. Another "best" value that is tracked by the global version of the particle swarm optimization is the overall best value and its location obtained so far by any particle in the population. This location is called 'gBest .'

The particle swarm optimization concept consists of, at each step, changing the velocity (i.e. accelerating) of each particle toward its 'pBest' and 'gBest' locations. Acceleration is weighted by a random term with separate random numbers being generated for acceleration toward 'pBest' and 'gBest' locations. The basic swarm parameters position and velocity are updated using the following equations [27]:

$$V_{i+1} = wV_i + c_1 r_1 (pBest_i - X_i) + c_2 r_2 (gBest_i - X_i) \quad (2)$$

$$X_{i+1} = X_i + V_{i+1} \quad (3)$$

where w is the inertia weight for velocities (previously set between 0 and 1), X_i is the current value particle i, V_i is the updated velocity of particle i, $pBest_i$ is the best solution found by particle i, $gBest_i$ is the best solution found by the swarm, r_1 and r_2 are uniform random numbers in the [0,1] range, c_1 means the cognitive component (self confidence of the particle), and c_2 means the social component (swarm confidence) and they are constants that influence how each particle is directed towards good positions taking into account personal best and global best information, respectively .

They usually are set as $c_1 = c_2 = 1.5$. The role of w is crucial for the PSO convergence. It is employed to control the impact of previous velocities on the current particle velocity. A general rule of thumb indicates to set a large value initially to make the algorithm explore the search space and then

gradually reduce it in order to get refined solutions [29,30].

2.5. Simulated annealing

In 1953, Metropolis developed a method for solving optimization problems that mimics the way thermodynamic systems go from one energy level to another [31]. He thought of this after simulating a heat bath on certain chemicals. This method is called simulated annealing (SA). Kirkpatrick et al. [32] originally thought of using SA on a number of problems .

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one .

By analogy with this physical process, each step of the SA algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends both on the difference between the corresponding function values and also on a global parameter T (temperature), that is gradually decreased during the process. The dependency is such that the current solution changes almost randomly when T is large, but increasingly "downhill" as T goes to zero .

The allowance for "uphill" moves potentially saves the method from becoming stuck at local optima. Several parameters need to be included in an implementation of SA. These are summarized by Davidson and Harel [33]:

•The set of configurations, or states, of the system, including an initial configuration (which is often chosen at random).

•A generation rule for new configurations, which is usually obtained by defining the neighborhood of each configuration and choosing the next configuration randomly from the neighborhood of the current one.

•The target, or cost, function, to be minimized over the configuration space. (This is the analogue of the energy).

•The cooling schedule of the control parameter, including initial values and rules for when and how to change it. (This is the analogue of the temperature and its decreases).

•The termination condition, which is usually based on the time and the values of the cost function and/or the control parameter.

2.6. Imperialist competitive algorithm

Imperialist competitive algorithm (ICA) is inspired from the social-political process of imperialism and imperialistic competition. Similar to many optimization algorithms, ICA starts with an initial population. Each individual of the population is called a 'country'.

Some of the best countries with the minimum cost are considered as the imperialist states and the rest will be the colonies of those imperialist states. All the colonies are distributed among the imperialist countries based on their power.

To define the algorithm, first of all, initial countries of size $N_{Country}$ are produced. Then, some of the best countries (with the size of N_{imp}) in the population are selected to be the imperialist states. Therefore, the rest with the size N_{col} will form the colonies that belong to imperialists.

Then, the colonies are divided among imperialists according to their power [34]. In such a way that the initial number of each empire's colonies has to be proportional to its power. So, the initial number of colonies of the nth empire will be [35]:

$$NC_n = round\{\left|\frac{Cost_n}{\sum_{i=1}^{N_{imp}} Cost_i}\right| \times N_{col}\} \quad , \quad n = 1,2,...,N_{imp} \tag{4}$$

where NC_n is the initial number of colonies of the nth empire and N_{col} is the total number of initial colonies. To divide the colonies, NC_n of the colonies are randomly chosen and given to the nth imperialist [35].

After dividing all colonies among imperialists and creating the initial empires, these colonies start moving toward their relevant imperialist country. This movement is a simple model of assimilation policy. Also, the total power of an empire is defined by the sum of the cost of the imperialist, and some percentage of the mean cost of its colonies as given [35]:

$$TC_n = \text{Cost}(\text{imperialist}_n) + \xi\{\text{mean}(\text{Cost}(\text{colonies of empire}_n))\} \tag{5}$$

where TC_n is the total power of the nth empire and ξ is a positive small number. After computing the total power of empires, usually the weakest colony

(or colonies) of the weakest empire is chosen by other empires and the competition is started on possessing this colony. Each imperialist participating in this competition, according to its power, has a probable chance of possessing the cited colony.

To start the competition, at first, the weakest empire is chosen and then the possession probability of each empire is estimated. The possession probability P_p is related to the total power of the empire (TC). In order to evaluate the normalized total cost of an empire (NTC), the following equation is used [35]:

$$NTC_n = \max_i\{TC_i\} - TC_n \qquad n, i = 1, 2, 3, ..., N_{imp} \quad (6)$$

During the imperialistic competition, the weak empires will slowly lose their power and getting weak by the time. At the end of process, just one empire will remain that governs the whole colonies [35].

3 Results and discussions

The problem investigated here, is taken from ref. [11] where the results were obtained using the RSM. In the present paper, metaheuristic algorithms including GA, SA, PSO, and ICA were used as optimizers in design optimization of 1,3-PD from Crude Glycerol in terms of concentration of 1,3-PD and the results were compared to the results obtained by the RSM .

The problem discussed here is maximization problem which the objective function is to find the maximum of 1,3-PD concentration (production, g/L). Hence, in order to convert it to minimization problem, -1 is multiplied to the objective function (Eq. (1)).

GA, SA, PSO, and ICA algorithms have shown great potentials for solving optimization problems as they conduct global stochastic search. The GA, PSO, SA, and ICA were implemented in MATLAB programming software and run on Pentium IV 2.53 GHz CPU with 4 GB RAM.

The task of optimizing 1,3-PD production (g/L) was executed in 25 independent runs for all considered optimizers. The initial parameters for GA were; population size of 20 individuals, scattered crossover fraction of 0.8, stochastic uniform as a selection function and rank as a scaling function.

Accordingly, for PSO, the initial parameters were; population size of 20 individual, the inertia weight for velocities of 0.8 and cognitive and social components (c_1 and c_2) of 1.5. Similarly for SA, the initial parameters considered as initial temperature of 100, cooling ratio of 0.98, fast annealing process and re-annealing interval of 100 iterations.

Also, the initial parameters for ICA were chosen as number of country of 20, number of imperialist country of 3, revolution rate of 0.3. The number of function evaluations (NFEs) determines the speed (computational effort) and the robustness of the algorithm. Less NFEs, means less time to reach the global optimum. This feature returns back to the structure of the algorithm.

Best solution represents the accuracy of the method. The NFEs and best solution are dependent on each other. The ideal situation is the less NFEs and more accurate solution.

For comparison with other algorithms, a maximum NFEs of 2000 (NFEs = Number of iterations × Number of population) was imposed for all reported optimizers. The maximum number of iterations was 100 for GA, PSO, and ICA. Accordingly, for SA the maximum number of iterations of 2000 was chosen.

The statistical results obtained from various optimization algorithms include of worst, mean, best solutions, and standard deviation (SD) are shown in Table 1. By observing Table 1, in terms of statistical optimization results, ICA is superior to other methods in this paper except the best solution with SD equal to zero. However, PSO detected its best solution better than ICA, GA, and SA.

Table1. Statistical results obtained from four metaheuristic algorithms for 1,3-PD concentration. “SD” stands for standard deviation.

Methods	Worst solution	Mean solution	Best solution	SD
GA	15.3468	15.3469	15.3470	6.45E-05
PSO	14.6010	16.2161	15.3471	0.22
SA	15.1238	15.2660	15.3440	0.06
ICA	15.34709	15.34709	15.34709	0

Table 2 represents the comparison of optimization results using RSM, GA, PSO, SA, and ICA in terms of coded and real values, and response. As shown in Table 2, all considered metaheuristic

algorithms surpassed the RSM in terms of maximum of production (concentration, g/L).

Table 2. Comparison of results using RSM, GA, PSO, SA, and ICA in terms of coded and actual values for 1,3-PD. "*Y*" is the 1,3-PD concentration (g/L). "*C*" and "*A*" stand for coded and actual values, respectively

Factors	RSM		GA		PSO		SA		ICA	
	C	A	C	A	C	A	C	A	C	A
X_1	0.74	7.37	1.128	7.57	1.125	7.56	1.089	7.54	1.125	7.56
X_2	1.00	8	2	10	2	10	2	10	2	10
X_3	0.52	35.2	1.277	42.77	1.277	42.77	1.275	42.75	1.277	42.77
X_4	0.80	10.8	0.381	10.76	0.380	10.76	0.435	10.87	0.380	10.76
X_5	-0.04	36.88	-0.086	36.74	-0.089	36.73	-0.117	36.64	-0.089	36.73
Y	13.74		15.3470		15.3471		15.3440		15.34709	

Based on the optimization results, the actual optimum culture conditions for 1,3-PD production using Crude Glycerol are pH = 7.56, $(NH_4)_2SO4$ = 10 g/L, crude glycerol = 42.75 g/L, cultivation time = 10.76 h, and temperature = 36.73° C. The model predicted that a maximum of 15.3471 g/L 1,3-PD can be obtained using the above optimum conditions.

Figures 1 to 4 demonstrate production (g/L) history versus the number of iterations (generations) for 1,3-PD using GA, PSO, SA, and ICA, respectively. As can be seen from Figures 1 to 4, ICA and PSO found their best solution faster than other reported algorithms at almost 24 iterations (480 function evaluations) in terms of convergence rate .

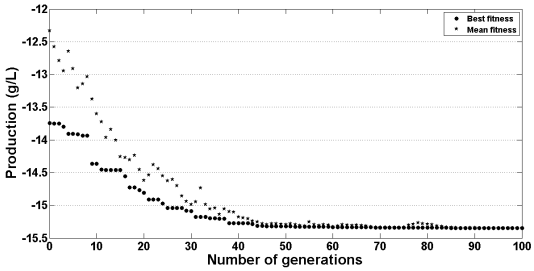

Figure 1: Production (concentration) (g/L) history with respect to number of generations using GA

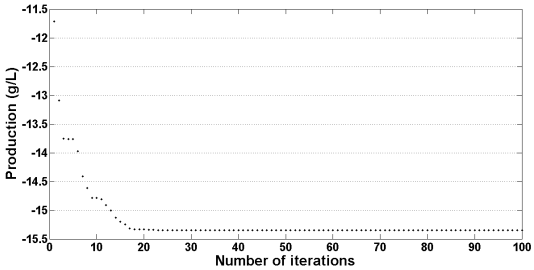

Figure 2: Production (g/L) history with respect to number of iterations using PSO

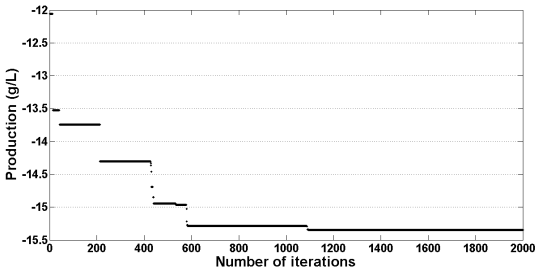

Figure 3: Production (g/L) history with respect to number of iterations using SA

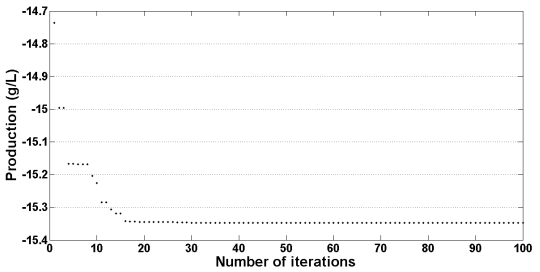

Figure 4: Production (g/L) history with respect to number of iterations using ICA

Figure 5 demonstrates the 1,3-PD production (g/L) versus the time using RSM [11] and PSO. In terms of optimum time, when the time passes the concentration of 1,3-PD is increased conditionally as shown in Figure 5.

However, this raise is not continued after almost 11 hours. The value of optimum time for PSO is obtained 10.76 hours which gives us more concentration (15.3471 g/L) and also less elapsed time compared to the RSM (10.8 h).

As can be seen from Figure 5a, after almost 10.8 hours the value of concentration of 1,3-PD is reduced. This behavior is similar for PSO after 10.76 hours. The optimization of 1,3-PD for culture conditions offers less culture time and more production of 1,3-PD using metaheuristic algorithms.

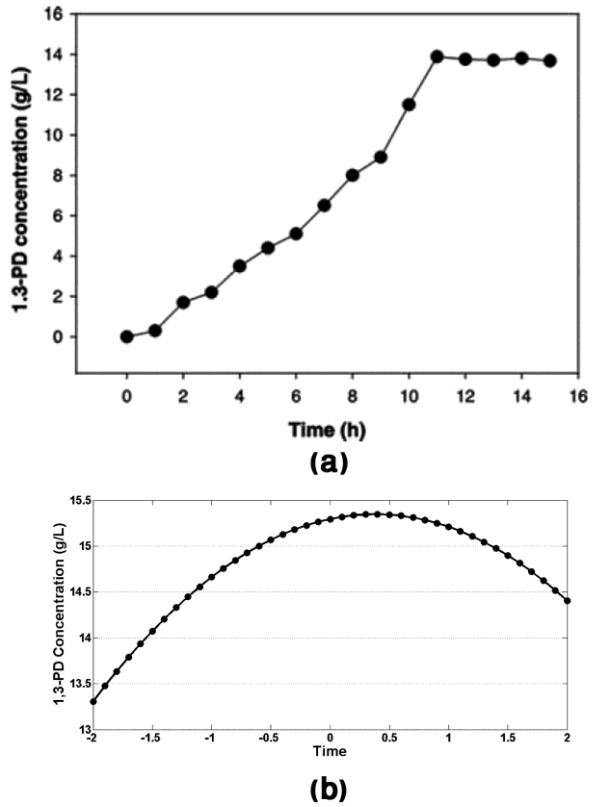

Figure 5: 1,3-PD production under the statistically optimized culture conditions with respect to the time using: (a) RSM with actual values [11], (b) PSO with coded values

4 Conclusions

This paper extended some findings for 1,3-Propanediol (1,3-PD) design in optimization of culture conditions with five independent factors including the value of pH (6-8), $(NH_4)_2SO4$ (2-10 g/L), crude glycerol (10-50 g/L), cultivation time (6-14 h), and temperature (31° -43°). The optimum conditions of 1,3-PD is expected to maximize the production of 1,3-PD.

The proposed mathematical model, which was offered by response surface methodology (RSM), was optimized using metaheuristic algorithms including genetic algorithms (GAs), simulated annealing (SA), particle swarm optimization (PSO), and imperialist competitive algorithm (ICA). The optimal culture conditions for maximum 1,3-PD production using metaheuristic algorithms were as follows: pH = 7.56, $(NH_4)_2SO4$ = 10 g/L, crude glycerol = 42.75 g/L, cultivation time = 10.76 h, and temperature = 36.73 °C.

The metaheuristic algorithms optimized the RSM model that a maximum of 15.3471 g/L 1,3-PD can be obtained using the above optimal conditions, while the RSM detected the maximum of 13.74 g/L 1,3-PD. In terms of statistical results, ICA, except the PSO for the best solution, was superior to other considered optimizers.

Increasing the production of 1,3-PD is a promising research which is a valuable chemical intermediate potentially used in the manufacture of polymers.

Acknowledgment

The authors would like to acknowledge for the Ministry of Higher Education of Malaysia and the University of Malaya, Kuala Lumpur, Malaysia for the financial support under UM.TNC2/IPPP/UPGP /628/6/ER013/2011A

.

References

[1] Huang, H., C. S. Gong, and G. T. Tsao (2002) Production of 1,3-propanediol by Klebsiella pneumoniae. Appl. Biochem. Biotechnol. 98-100: 687-698.

[2] Zeng, A. P. and H. Biebl (2002) Bulk chemicals from biotechnology: the case of 1,3-propanediol production and the new trends. Adv. Biochem. Eng. Biotechnol. 74: 239-259.

[3] El-Ziney, M. G., N. Arneborg, M. Uyttendaele, J. Debevere, and M. Jakobsen (1998) Characterization of growth and metabolite production of Lactobacillus reuteri during glucose/glycerol cofermentation in batch and continuous cultures. Biotechnol. Lett. 20: 913-916.

[4] Houng, J. Y., K. C. Chen, and W. H. Hsu (1989) Optimization of cultivation medium composition for isoamylase production. Appl. Microbiol. Biotechnol. 31: 61-64.

[5] Reimann, A., H. Biebl, and W. D. Deckwer (1998) Production of 1,3-propanediol by Clostridium butyricum in continuous culture with cell recycling. Appl. Microbiol. Biotechnol. 49: 359-363.

[6] Biebl, H. (1991) Glycerol fermentation of 1,3-propanediol by Clostridium butyricum. Measurement of product inhibition by use of a pH-auxostat. Appl. Microbiol. Biotechnol. 35: 701-705.

[7] Forsberg, C. W. (1987) Production of 1,3-propanediol from glycerol by Clostridium acetobutylicum and other Clostridium species. Appl. Environ. Microbiol. 53: 639-643.

[8] Mu, Y., H. Teng, D. J. Zhang, W. Wang, and Z. L. Xiu (2006) Microbial production of 1,3-propandiol by Klebsiella pneumoniae using crude glycerol from biodiesel preparations. Biotechnol. Lett. 28: 1755-1759.

[9] Biebl, H., A. P. Zeng, K. Menzel, and W. D. Deckwer (1998) Fermentation of glycerol to 1,3-propanediol and 2,3-butanediol by Klebsiella

pneumoniae. Appl. Microbiol. Biotechnol. 50: 24-29.

[10] Box, G. E. P., W. G. Hunter, and J. S. Hunter (1978) Statistic for Experimenters: An Introduction to Design, Data Analysis, and Model Building. John Wiley and Sons Inc., New York, NY, USA.

[11] Baek-Rock Oh, Jeong-Woo Seo, Min Ho Choi, and Chul Ho Kim, Optimization of Culture Conditions for 1,3-Propanediol Production from Crude Glycerol by Klebsiella pneumoniae Using Response Surface Methodology, Biotechnology and Bioprocess Engineering, 2008, 13: 666-670.

[12] Maghsoud, A., Amir, A.A.N. and Komeil, G. (2008). Response surface methodology and genetic algorithm in optimization of cement clinkering process, J. Appl. Sci., 8(15): 2732-2739.

[13] Low, K.L., Tan, S.H., Zein, S.H.S., McPhail, D.S. and Boccaccini, A.R. (2011). Optimization of the mechanical properties of calcium phosphate/multi-walled carbon nanotubes/bovine serum albumin composites using response surface methodology, Mater. Des., 32: 3312-3319.

[14] Zheng, Z. M., Q. L. Hu, J. Hao, F. Xu, N. N. Guo, Y. Sun, and D. H. Liu (2008) Statistical optimization of culture conditions for 1,3-propanediol by Klebsiella pneumoniae AC 15 via central composite design. Bioresour. Technol. 99: 1052-1056.

[15] Osman, I.H. and Laporte, G. (1996). Metaheuristics: a bibliography, Ann. Oper. Res., 63: 513-623.

[16] Blum, C. and Andrea, R. (2003). Metaheuristics in combinatorial optimization: overview and conceptual comparison, ACM Comput. Surv., 35(3): 268-308.

[17] Yang, X.S. (2010). Nature-Inspired metaheuristic algorithms, 2nd Edition, Luniver Press.

[18] Yang, X.S. (2010). Engineering applications, John Wiley & Sons.

[19] Glover, F. and Kochenberger, G.A. (2003). Handbook of metaheuristics, Kluwer Academic Publishers.

[20] Sanchez, N., Martinez, M. and Aracil, J. (1997). Selective esterification of glycerin to 1-glycerol monooleate: optimization studies, Ind. Eng. Chem. Res., 36(5): 1529-1534.

[21] Box, G.E.P. and Draper, N.R. (1987). Emprical model-building and response surface, New York, John Wiley & Sons.

[22] Areibi, S., Moussa, M. and Abdullah, H. (2001). A comparison of genetic/ memetic algorithms and other heuristic search techniques, In: ICAI 2001, Las VeGAs, Nevada.

[23] Youssef, H., Sait, S.M. and Adiche, H. (2001). Evolutionary algorithms, simulated annealing and tabu search: a comparative study, Eng. Appl. Artif. Intell., 14: 167-181.

[24] Elbeltagi, E., Hegazy, T. and Grierson, D. (2005). Comparison among five evolutionary-based optimization algorithms, Adv. Eng. Inf., 19: 43-53.

[25] Giraud-Moreau, L. and Lafon, P. (2002). Comparison of evolutionary algorithms for mechanical design components, Eng. Optim., 34: 307-322.

[26] Goldberg, D. (1989). Genetic algorithms in search, optimization and machine learning, Addison-Wesley, Reading, MA.

[27] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization, In: IEEE IJCNN, Perth, Australia, 4: 1942-1948.

[28] Kennedy, J. and Eberhart, R. (1997). A discrete binary version of the particle swarm algorithm, In:IEEE Sys. Man. Cybern., Computational Cybernetics and Simulation, Orlando, FL, 5: 4104-4108.

[29] Dong, Y., Tang, J., Xu, B. and Wang, D. (2005). An application of swarm optimization to nonlinear programming, Comput. Math. Appl., 49: 1655-1668.

[30] Bergh, F. and Engelbrecht, A.P. (2006). A study of particle swarm optimization particle trajectories, Inform. Sciences, 176: 937-971.

[31] Fleischer, M. (1995). Simulated annealing: past, present, and future, In: Arlington, VA, USA, IEEE, 155-161.

[32] Kirkpatrick, S., Gellat, J.R.C.D. and Vecchi, M.P. (1983). Optimization by simulated annealing, Science, 220: 671-680.

[33] Davidson, R. and Harel, D. (1996). Drawing graphs nicely using simulated annealing, ACM T. Graphics, 15: 301-331.

[34] E. Atashpaz-Gargari and C. Lucas, Imperialist Competitive Algorithm: An Algorithm for Optimization Inspires by Imperialistic Competition, IEEE Congress on Evolutionary Computation, Singapore, 2007.

[35] A. Khabbazi, E. Atashpaz-Gargari, and C. Lucas, Imperialist competitive algorithm for minimum bit error rate beamforming, Int. J. Bio-Inspired Computation. (2009), pp.125–133.