Hindawi Publishing Corporation
Journal of Automated Methods and Management in Chemistry
Volume 2009, Article ID 927426, 9 pages
doi:10.1155/2009/927426

Research Article

Optimal Solutions of Multiproduct Batch Chemical Process Using Multiobjective Genetic Algorithm with Expert Decision System

Diab Mokeddem and Abdelhafid Khellaf

Department of Electronics, Faculty of Engineering, University of Setif, 19000 Setif, Algeria

Correspondence should be addressed to Diab Mokeddem, mokeddem_d@yahoo.fr

Received 9 February 2009; Accepted 3 March 2009

Recommended by Peter Stockwell

Optimal design problem are widely known by their multiple performance measures that are often competing with each other. In this paper, an optimal multiproduct batch chemical plant design is presented. The design is firstly formulated as a multiobjective optimization problem, to be solved using the well suited non dominating sorting genetic algorithm (NSGA-II). The NSGA-II have capability to achieve fine tuning of variables in determining a set of non dominating solutions distributed along the Pareto front in a single run of the algorithm. The NSGA-II ability to identify a set of optimal solutions provides the decision-maker DM with a complete picture of the optimal solution space to gain better and appropriate choices. Then an outranking with PROMETHEE II helps the decision-maker to finalize the selection of a best compromise. The effectiveness of NSGA-II method with multiojective optimization problem is illustrated through two carefully referenced examples.

1. Introduction

Batch processes are used in production of many low-volume but high-value-added products (such as speciality chemicals, health care, food, agrochemicals,. . .etc.) because of operation flexibility in today's market-driven environment. Manufactory of these products generally involves multi step synthesis [1]. In addition, if two or more products require similar processing steps, the same set of equipment is considered for at least economical reason. A batch plant producing multiple products is categorized as either a multiproduct plant or a multipurpose plant. Multiproduct plants produce multiple products following a sequential similar recipe. In such a plant, all the products follow the same path through the process and only one product is manufactured at a time. Each step is carried out on single equipment or on several parallel equipment units. Processing of other products is carried out using the same equipment in successive production runs or campaigns. In a multipurpose plant, each product follows one or more distinct processing paths; so more than one product may be produced simultaneously in such plants. The present work is directed toward the optimal design problems of multiproduct batch plants.

In conventional optimal design of a multiproduct plant, production requirements of each product and a total production time for all products are available and specified. The number, the required volume, and size of parallel equipment units in each stage are then determined to minimize the investment. It should be emphasized that batch plantsdesign has been for long identified as a key problem in chemical engineering as reported in literature [2–9]. Formulation of batch plant design generally involves mathematical programming methods, such as linear programming (LP), nonlinear programming (NLP), mixed-integer linear programming (MILP) or mixed-integer nonlinear programming (MINLP). Mathematical programming or different optimization techniques, such as branch and bound, heuristics, genetic algorithm, simulated annealing, are thoroughly used to derive optimal solutions.

However, in reality the multiproduct design problem can be formulated as a multiobjective design optimization problem in which one seeks to minimize investment, operation cost, and total production time, and, simultaneously, to maximize the revenue. Recall that not much work has been reported in the literature on the multiobjective optimal design of a multiproduct batch plant. Huang and Wang

[10] introduced a fuzzy decision-making approach for multiobjective optimal design problem of a multiproduct batch plant. A monotonic increasing or decreasing membership function is used to define the degree of satisfaction for each objective function and the problem is then represented as an augmented minmax problem formulated as MINLP models. To obtain a unique solution, the MINLP problem is solved using a hybrid differential evolution technique. Dedieu et al. [11] presented the development of a two-stage methodology for multiobjective batch plant design and retrofit according to multiple criteria. The authors used a multiobjective genetic algorithm based on the combination of a single-objective genetic algorithm and a Pareto sort procedure for proposing several plant structures and a discrete event simulator for evaluating the technical feasibility of the proposed configurations.

In the case of multiple objectives, an optimum solution with respect to all objectives may not exist. In most cases, the objective functions are in conflict, because in order to decrease any of the objective functions, we need to increase other objective functions. Recently, Solimanpur et al. [12] developed a sophisticated multiobjective integer programming model where the objectives considered were the maximization of total similarity between parts, the minimization of the total processing cost, the minimization of the total processing time and the minimization of the total investment needed for the acquisition of machines [13].

The presence of multiple objectives in a problem usually gives rise to a family of nondominated solutions, largely known as *Pareto-optimal*solutions, where each objective component of any solution along the Pareto front can only be improved by degrading at least one of its other objective components. Since none of the solutions in the nondominated set is absolutely better than any other, any one of them is then an acceptable solution. As it is difficult to choose any particular solution for a multiobjective optimization problem without iterative interaction with the decision maker (DM) [14] one general approach is to establish first the entire set of Pareto-optimal solutions, where an external Decision Maker (DM) direct intervention gives interactive information in the multiobjective optimization loop [15]. So, a satisfactory solution of the problem is found as soon as the knowledge is acquired [16]. Promethee II (Preference Ranking Organisation METHod for Enrichment Evaluations—2nd version) is a popular decision method that has been successfully applied in the selection of the final solution of multiobjective optimization problems. It generates a ranking of available points, according to the DM preferences, and the best ranked one is considered the favourite final solution. It is based on the concept of outranking relation, which is a binary relation defined between every pair (a, b) of alternatives, in such way that, if a is preferred to b (according to the DM interests), then it is said that a outranks b. When these relations are defined between all pairs of alternatives, they are exploited according to some rules in order to rank all solutions from the best to the worst.

The first GA proposed for multiobjective optimization was VEGA [17]. This is a nonPareto based approach based on the selection of several relevant groups of individuals, each group being associated to a given objective. It is reported that the method tends to crowd results at extremes of the solution space, often yielding to poor convergence of the Pareto front. A more recent algorithm, based on scalarization with a weighted sum function, is proposed in Ishibuchi and Murata [18] where the weights are randomly chosen. Many successful evolutionary multiobjective optimization algorithms were developed based on the two ideas suggested by Goldberg [19]: Pareto dominance and niching. Pareto dominance is used to exploit the search space in the direction of the Pareto front and niching technique explores the search space along the front to keep diversity. The well-known algorithms in this category include Multiobjective Genetic Algorithm: (MOGA) [16], Niched Pareto Genetic Algorithm: (NPGA) [20], Strength Pareto Evolutionary Algorithm: (SPEA) [21], Multiobjective Evolutionary Algorithm: (MOEA) [22], the Nondominated Sorting Genetic Algorithm (NSGA) proposed by Srinivas and Deb [23] was one of the first evolutionary algorithm for solving multiobjective optimization problems. Although NSGA has been successfully applied, the main criticisms of this approach has been its high computational complexity of nondominated sorting, lack of elitism, and need for specifying a tuneable parameter called sharing parameter. Recently, Deb et al. [24] reported an improved version of NSGA, which they called NSGA-II, to address all the above issues.

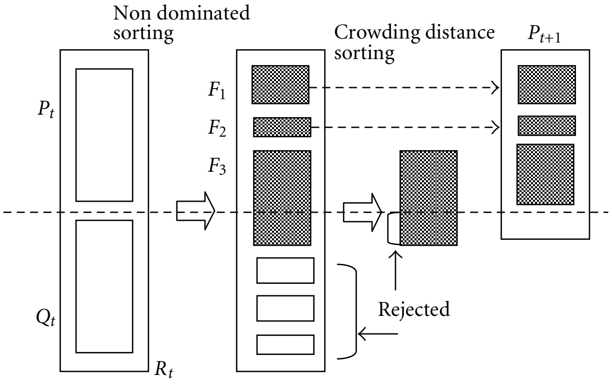


Figure 1: The NSGA II Procedure.

The purpose of this study is to extend this methodology for solution of multiobjective optimal control problems under the framework of NSGA-II. The efficiency of the proposed method is illustrated by solving multiobjective optimization problem.

2. Formulation of the Multiobjective Problem

The problem of multiproduct batch plant covered in this paper can be defined by assuming that the plant consists of a sequence of M batch processing stages that are used to manufacture N different products. At each stage $j$there are N_j identical units in parallel operating out of phase, each with a size V_j .Each product i follows the same general processing sequence.

Batches are transferred from one stage to the next without any delay, that is, we consider a zero-wait operating policy.

In the conventional design of a multiproduct batch plant, one seeks to minimize the investment cost by determining the optimal number, required volume and size of parallel equipment units in each stage for a specified production requirement of each product and the total production time. However, in reality the designer considers not only minimizing the investment but also minimizing the operation cost and total production time while maximizing the revenue, simultaneously

$$\underset{N_j,V_j,B_i,T_{L_i},Q_i,H}{\text{Max}} \quad \text{Revenue} = f_1 = \sum_{i=1}^{N} Cp_i Q_i, \tag{1}$$

$$\underset{N_j,V_j,B_i,T_{L_i},Q_i,H}{\text{Min}} \text{Investment cost} = f_2 = \sum_{j=1}^{M} N_j \alpha_j V_j^{B_j}, \tag{2}$$

$$\underset{N_j,V_j,B_i,T_{L_i},Q_i,H}{\text{Min}} \quad \text{Operation cost} = f_3 = \sum_{i=1}^{N} \sum_{j=1}^{M} C_{E_j} \frac{Q_j}{B_i} + C_{o_i} Q_i, \tag{3}$$

$$\underset{N_j,V_j,B_i,T_{L_i},Q_i,H}{\text{Min}} \quad \text{Total production time} = f_4 = H. \tag{4}$$

So, the multiobjective problem consists of determining the following parameters:

(i) N_j the number of parallel units in stage j,

(ii) V_j the required volume of a unit in stage j,

(iii) B_i size of the batch of product i at the end of the M stages,

(iv) TL_i the cycle time for product i,

(v) Q_i the production requirement of product i and,

(vi) H the total production time,

while satisfying certain constraints such as volume, time, and so forth.

The constraints are expressed as follows:

(1) *Volume constraints.* Volume V_j has to be able to process all the products i:

$$S_{ij} B_i \leq V_j, \qquad \forall i = 1, \ldots, N; \;\; \forall j = 1, \ldots, M. \tag{5}$$

(2) *Time constraint.*The summation of available production time for all products is not more than the net total time for production

$$\sum_{i=1}^{N} \frac{Q_i}{B_i} T_{L_i} \leq H. \tag{6}$$

(3) The limiting cycle time for product i:

$$\frac{\tau_{ij}}{N_j} \leq T_{L_i}, \qquad \forall i = 1, \ldots, N; \;\; \forall j = 1, \ldots, M. \tag{7}$$

(4) *Dimension constraints.* Every unit has restricted allowable range

$$\begin{aligned} V_j^L &\leq V_j \leq V_j^U, \qquad \forall j = 1, \ldots, M, \\ B_j^L &\leq B_j \leq B_j^U, \qquad \forall j = 1, \ldots, N. \end{aligned} \tag{8}$$

3. Elitist Nondominated Sorting Genetic Algorithm (NSGA-II)

The NSGA II Pareto ranking algorithm is an elitist Deb et al. [24] system and maintains an external archive of the Pareto solutions. In contrast to the simple genetic algorithms that look for the unique solution, the multiobjective genetic algorithm tries to find as many elements of the Pareto set as possible. For the case of the NSGA-II, this one is provided with operators who allow it to know the level of nondominance of every solution as well as the grade of closeness with other solutions; which allows it to explore widely inside the feasible region.

In a brief form, the functioning of the multiobjective genetic algorithm NSGA-II can be described through the following steps.

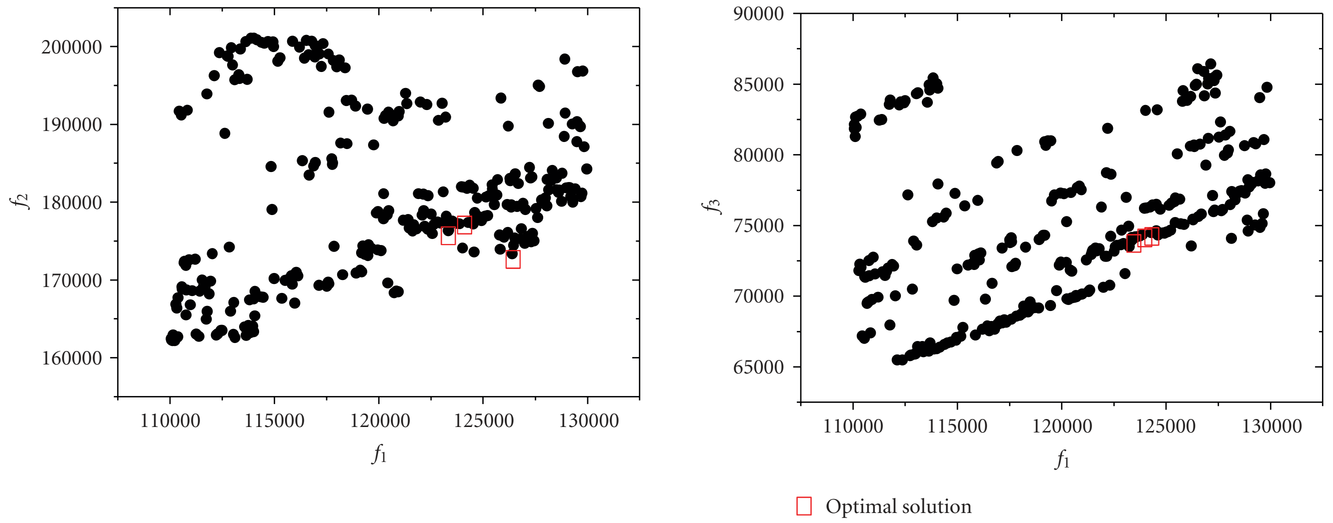
Fast Nondominated Sort. A very efficient procedure, is used to arrange the solutions in fronts (nondominated arranging), in accordance with their aptitude values. This is achieved, creating two entities for each of the solutions. A domination count n_p, the number of solutions which dominates the solution p, and a set (S_p), that contains the solutions that are dominated for p. The solutions of the first front have the higher status of nondominance in the Pareto sense.

Diversity Preservation. This is achieved, by means of the calculation of the crowding degree or closeness for each of the solutions inside the population. This quantity is obtained, by calculating the average distance of two points on either side of a particular solution along each of the objectives. This quantity serves as an estimate of the cuboid perimeter, formed by using the nearest neighbours as the vertices. There is also, an operator called Crowded-Comparison $(\prec_n)$, which guides to the genetic algorithm, towards the Pareto optimal front, in accordance with the following criterion:

$$\begin{aligned} &i \prec_n j \;\text{ if }\; (i_{\text{rank}} < j_{\text{rank}}), \\ &\text{or } (i_{\text{rank}} = j_{\text{rank}}) \text{ and } (i_{\text{didtance}} > j_{\text{distance}}). \end{aligned} \tag{9}$$

In accordance with the previous criterion, between two nondominated solutions, we prefer the solution with the better rank. Otherwise, if both solutions belong to the same front, then, we prefer the solution that is located in a lesser crowded region.

Initial Loop. Initially, a random parent population (P_o) of size N is created. Later this one is ordained, using the procedure of nondominated arranging. Then the usual binary tournament selection, recombination and mutation operators are used to create a new population (Q_0), of size N.



(a)

(b)

Figure 2: Pareto optimal solutions for example 1.

Table 1: Data used in example 1.

Processing times, $\tau_{ij}(h)$				Unit price for the product ($/Kg)		
Product	Mixer	Reactor	Centrifuge	Product	C_p	C_0
A	8	20	4	A	0.35	0.08
B	10	12	3	B	0.37	0.1
Product		Size factors (L/kg)				
A	2	3	4			
B	4	6	3			
	Cost of equipment ($, V in litres)				Minimum size = 250 L	
	250 $V^{0.6}$	500 $V^{0.6}$	340 $V^{0.6}$		Maximum size = 2500 L	
	Operating cost factor (C_E)					
	20	30	15			

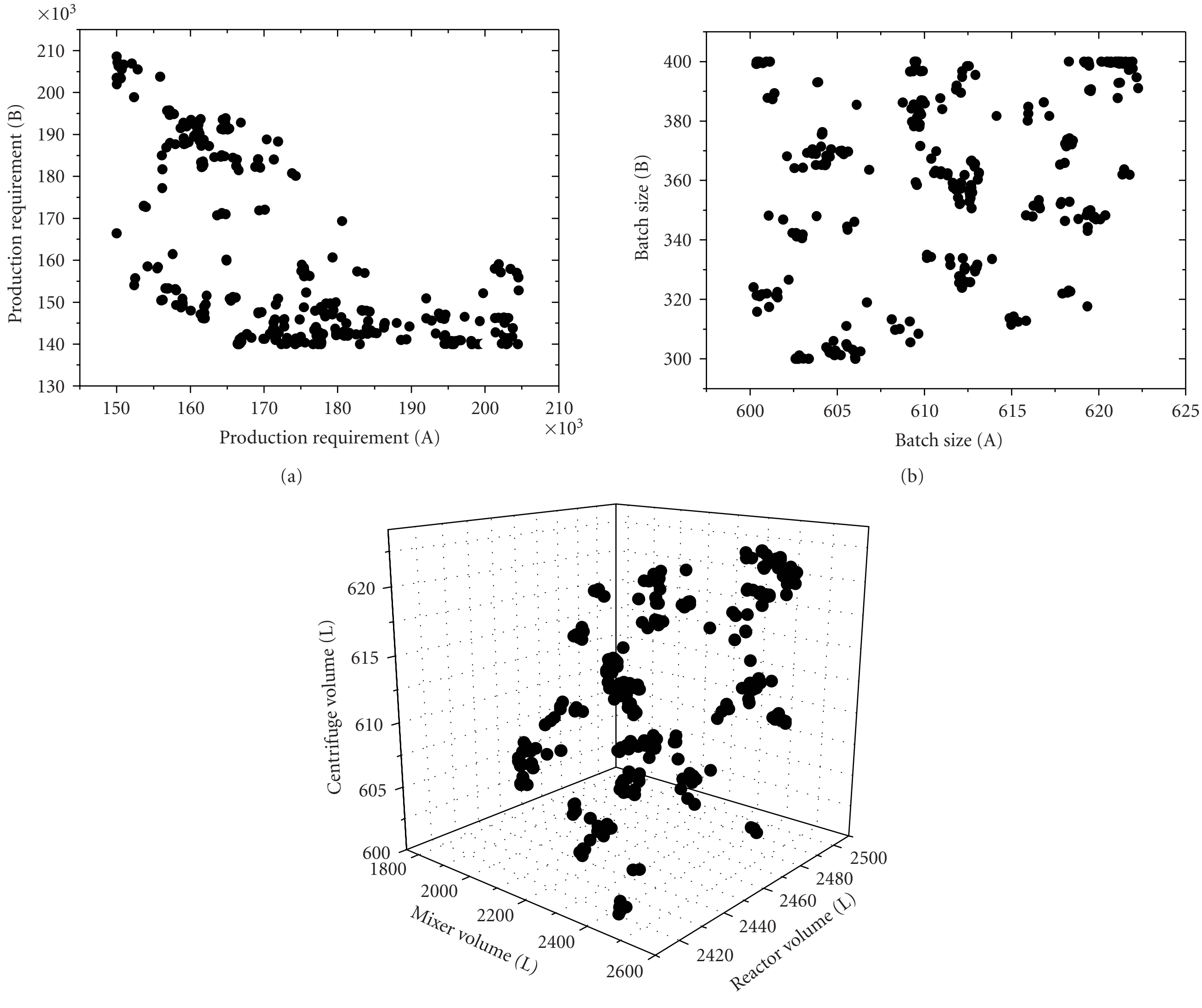
Main Loop. The NSGA-II procedure can be explained, by describing the *th*generation just as it is showed in Figure 1. The procedure begins with the combination of P_t and Q_t forming a new population called R_t, then the population R_t is sorted using the nondomination criterion. Since all previous and current population members are included in R_t, elitism is ensure. The population R_t has a size of $2N$, later, the different fronts of nondominated solutions are created, being F_1 the front that contains the better rank solutions. Figure 4 shows that, during the process of forming the new population P_{t+1}, the algorithm takes all members of the fronts F_1 and F_2, and some elements of the front F_3; this is, because N solutions are needed exactly for the new population P_{t+1} to find them exactly N solutions, the last front is ordained, which for this description is the number 3, arranging the solutions in descending order by means of the crowded comparison ($\prec_n$), and selecting the best solutions needed to fill all population slots. After having the population P_{t+1}, the genetic operators of selection, crossing and mutation, are used to create the new population Q_{t+1} of size N. Finally it is mentioned that the selection process, the crowded comparison operator is used.

4. Description of the Process

To demonstrate the effectiveness of NSGA-II on batch plant processes two examples are given here. The first example is about a batch plant consisting of three processing stages (mixer, reactor, and centrifuge) to manufacture two products, A and B. The second example treats four processing stages (mixer, reactor, extraction and centrifuge) to manufacture three products A, B and C. The data for examples 1 and 2 are illustrated, respectively, in Tables 1 and 2 (the processing times, size factor for the units and cost for each product).

5. Results and Discussion

5.1. Example 1. A four-objective optimization problem is considered and expressed in (1)–(4). The set of decision

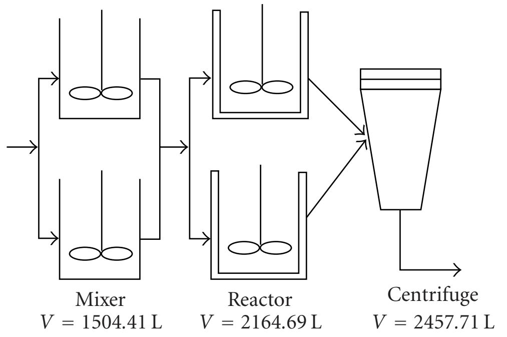


(a)

(b)

(c)

Figure 3: Relationships between some decision variables.



Product	A	B
Batch size (kg)	612.09	359.21
Cycle time (h)	10.13	5.54
Production requirement (kg)	203058.87	143817.7
Total production time (h) = 5582.87		

Figure 4: Optimal design of batch plant for example 1.

variables consists of the batch size, the total production time, the number of parallel units at each stage, the cycle time for each product, and the required volume of a unit in each stage. Since the number of parallel units at each stage is an integer decision variable, we code this variable as a binary variable. All other decision variables are coded as real numbers. Thus, there are 3 integer variables and 10 real variables. In addition to the constraints expressed by (5)–(8), we consider bounds on objective functions as additional constraints to generate feasible nondominated solutions in the range desired by the decision-maker, to have 19 constraints in all

$$f_i^L \le f_i \le f_i^U, \quad i = 1\ldots,4. \tag{10}$$

Then NSGA-II is employed to solve the optimization problem with the following parameters: maximum number of generation up to 200, population size 500, probability of crossover 0.85, probability of mutation 0.05,

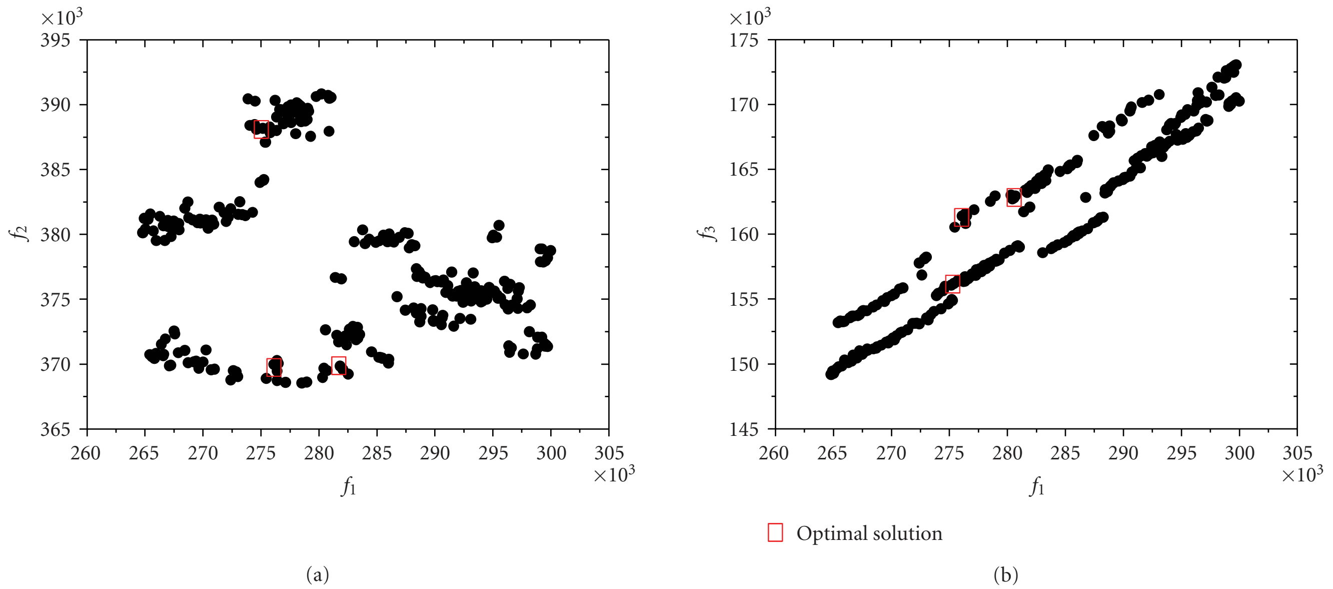


FIGURE 5: Pareto optimal solutions for example 2.

TABLE 2: Data used in example 2.

	Processing times, $\overline{\tau}_{ij}(h)$				Unit price for the product ($/Kg)		
Product	Mixer	Reactor	Extractor	Centrifuge	Product	C_p	C_0
A	1.15	9.86	0.4	0.5	A	0.27	0.08
B	5.95	7.01	0.7	0.42	B	0.29	0.10
C	3.96	6.01	0.85	0.3	C	0.32	0.12
γ_{ij}	0.4	0.33	0.3	0.2			
Product		Size factors (L/kg)					
A	8.28	9.7	6.57	2.95			
B	5.58	8.09	6.17	3.27			
C	2.34	10.3	5.98	5.7			
Product		Coefficients C_{ij}					
A	0.2	0.24	0.4	0.5			
B	0.15	0.35	0.7	0.42			
C	0.34	0.5	0.85	0.3			
	Cost of equipment ($, V in litres)					Minimum size = 250 L	
	250 $V^{0.6}$	250 $V^{0.6}$	250 $V^{0.6}$	250 $V^{0.6}$		Maximum size = 10000 L	
	Operating cost factor (C_E)						
	20	30	15	30			

distribution index for the simulated crossover operation 10 and distribution index for the simulated mutation operation 20.

The Pareto-optimal solutions for example 1 are presented in Figure 2. The revenue (f_1) increases with the increase in operation cost (f_3), while the investment cost (f_2) decreases. When all the four objective functions are considered simultaneously, solutions obtained in the present study show improvement as by Huang and Wang [10] results for the same problem. For example, let us consider the solution presented by Huang and Wang [10] with unit reference membership level for all objectives: $f_1 = 121\,350$, $f_2 = 171\,624$, $f_3 = 77\,299$, $f_4 = 5667$. The solution (1) presented in Table 3 of the present study improves the above solution f_1, f_3, f_4 while f_2 is comparable.

Figure 3 presents the relationships between some chosen decision variables. The large set of multiple optimal solutions

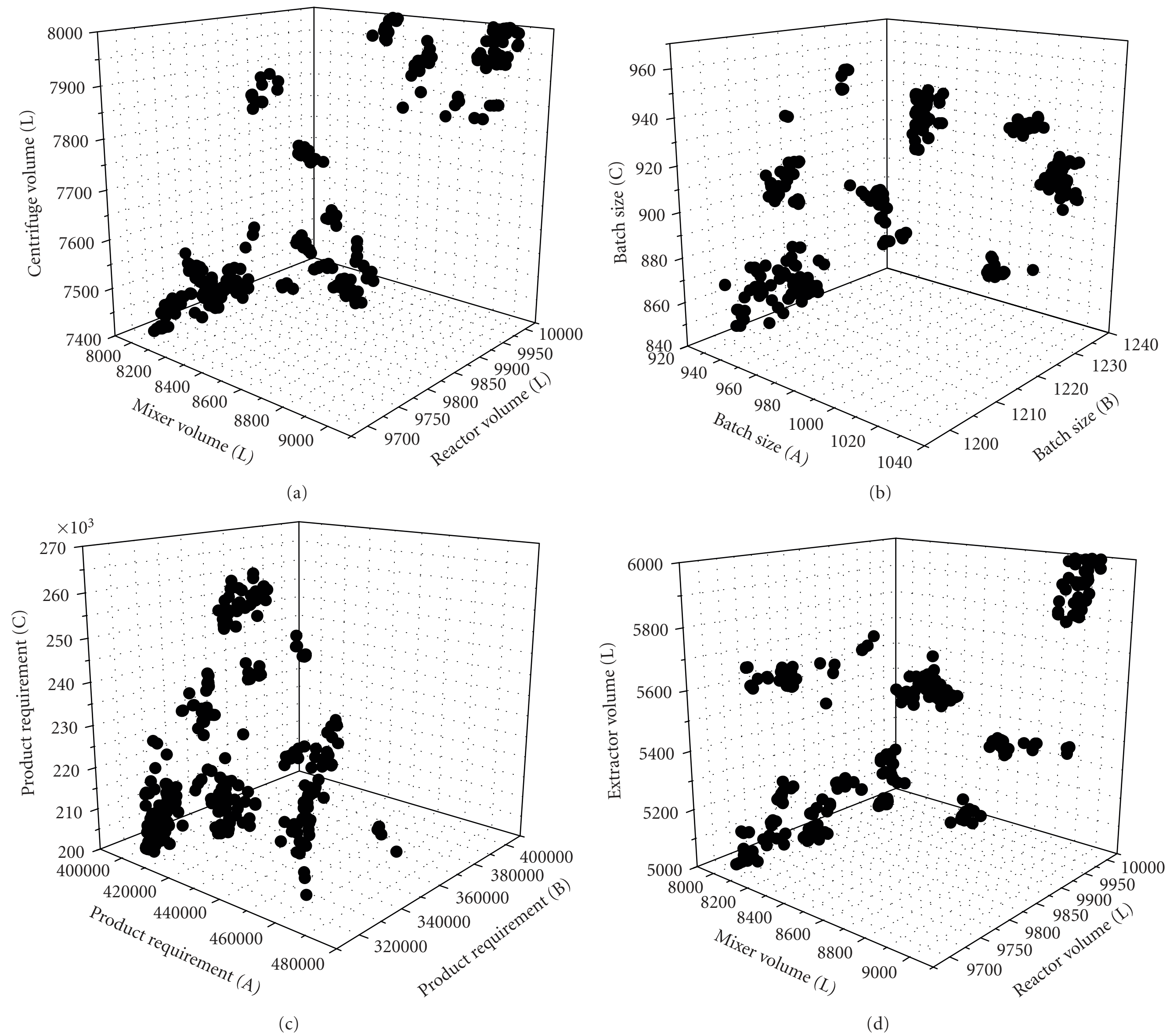


Figure 6: 3 Dimension plot of the relationship between some decision variables.

Table 3: Optimal objectif function values of example 1.

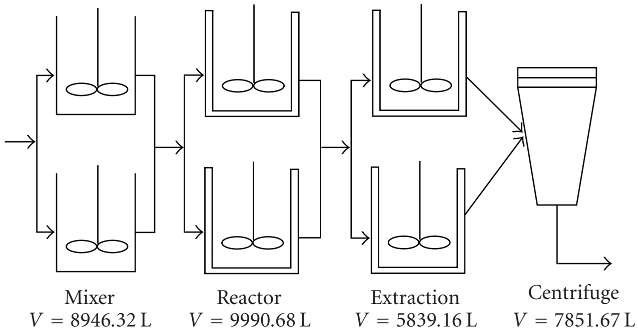
	Optimal objectif function values			
Case	f_1	f_2	f_3	f_4
1	122566.41	175955.39	73499.23	5579.37
2	124283.15	177389.07	74508.91	5582.87
3	126393.23	173354.67	84896.90	5638.60

Bounds for objective function: $[f_1^L, f_1^U] = [110000, 130000]$, $[f_2^L, f_2^U] = [150000, 200000]$, $[f_3^L, f_3^U] = [60000, 100000]$, $[f_4^L, f_4^U] = [5500, 6000]$.

Table 4: Optimal objectif function values of example 2.

	Optimal objectif function values			
Case	f_1	f_2	f_3	f_4
1	275766.10	388262.10	156449.20	5505.40
2	276096.30	369977.30	161392.30	5710.90
3	281818.70	369843.48	163552.80	5718.00

Bounds for objective function: $[f_1^L, f_1^U] = [250000, 300000]$, $[f_2^L, f_2^U] = [350000, 400000]$,$[f_3^L, f_3^U] = [150000, 200000]$, $[f_4^L, f_4^U] = [5500, 6000]$.



Product	A	B	C
Batch size (kg)	1024.18	1233.61	903.29
Cycle time (h)	6.3	6.12	5.48
Production requirement (kg)	41067.64	347391.86	200866.3
Total production time (h) = 5505.4			

Figure 7: Batch plant optimal design for example 2.

provides the decision maker with immediate information about the relationship among the several objective criteria and a set of feasible solutions. Thus, it helps the decision-maker to select a highly confident choice of solution. The fixed optimal plant structure as 221 corresponds to a two mixers, two reactors, and one centrifuge design. The optimal solution is shown in Figure 4.

5.2. Example 2. The batch plant in this case consists of four processing stages to manufacture three products A, B, and C, with four-objective optimization problem as expressed in (1)–(4). The set of decision variables remains the same as that in example 1. But we deal with 4 integer variables, 14 real variables, and 31 constraints which includes bounds on objective functions.

The same model equations of example 1 are used here except the processing time, τ_{ij} in (7). The time required to process one batch of product $i$in stage $j$is expressed as:

$$\tau_{ij} = \overline{\tau}_{ij} + c_{ij}B_i^{\gamma_j}, \quad \forall i, \quad \forall j, \tag{11}$$

where $\tau_{ij} \geq 0$, $c_{ij} \geq 0$ and γ_j are constants and B_i is the batch size for product i.

Thus, the processing time is not a constant, but depends on the decision parameters of the batch size. Table 2 presents the necessary data for the problem.

The constrained multiobjective MINLP problem is solved by NSGA-II with the same set of NSGA-II parameters as used in example 1.

As in example 1, the revenue (f_1) increases as operation cost (f_3) increases, while the investment cost (f_2) decreases following operation cost (f_3). The Pareto-optimal solutions for example 2 are presented in Figure 5. The relationships of the various decision variables are shown in Figure 6. Let mention that when all four objective functions are considered simultaneously, the solutions obtained in the present study improve significantly the results presented by Huang and Wang [10] for the same problem. For example, the solution presented by Huang and Wang [10] with unit reference membership level for all objectives (f_1 = 274 312, f_2 = 375 688, f_3 = 175 688, f_4 = 5639) the solution (1) presented in Table 4 of the present study improves the above solution f_1, f_3, f_4 while f_2 is comparable.

In this example, the plant structure evolved as optimal is: two mixers, two reactors, two extractors, and one centrifuge as presented in Figure 7.

Implementation of a trade-off analysis is dependent upon the availability of the decision-maker's preferences.

6. Conclusion

A multiobjective decision in a batch plant process design is considered and a non dominating sorting genetic algorithm (NSGA-II) is developed to get an optimal zone containing solutions under the concept of Pareto set. NSGA-II capability has been proved in evolving the entire set of nondominating solutions along the Pareto front in a single run of the algorithm. Thus, the Decision Maker (DM) is provided with the best trade-off operating zone. Furthermore, a better confident choice of design among several compromises of the decision maker can be achieved if the decision variables effects on the objective functions are known.

Finally, the large set of solutions presents a useful base for further alternative approaches to fulfil the DM targets.

The inherent dynamic nature of batch processes allows for their ability to handle variations in feedstock and product specifications and provides the flexibility required for multiproduct or multipurpose facilities. They are thus best suited for the manufacture of low-volume, high-value products, such as specialty chemicals, pharmaceuticals, agricultural, food, and consumer products, and most recently the constantly growing spectrum of biotechnology-enabled products. Reduced time to market, lower production costs, and improved flexibility are all critical success factors for batch processes.

Nomenclature

B_i	Size of the batch of product i at the end of the M stages (kg)
C_{E_j}	Operation cost in stage j (\$)
C_{O_i}	Operation cost of product i to be produced (\$/kg)
C_{P_i}	Price of product i (\$/kg)
H	Total production time (h)
M	Number of stages in the batch process
N	Number of products to be produced
N_j	Number of parallel units in stage j
Q_i	Production requirement of product i (kg)
S_{ij}	Size factor of product i in stage j (L/kg)
TL_i	Cycle time for product i (h)
V_j	Required volume of a unit in stage j (L)

Greek Symbols

α_j	Cost coefficient for unit j
β_j	Cost exponent for unit j
τ_{ij}	Processing time of product i in stage j (h).

References

[1] A. Lamghabbar, S. Yacout, and M. S. Ouali, "Concurrent optimization of the design and manufacturing stages of product development," *International Journal of Production Research*, vol. 42, no. 21, pp. 4495–4512, 2004.

[2] D.-M. Cao and X.-G. Yuan, "Optimal design of batch plants with uncertain demands considering switch over of operating modes of parallel units," *Industrial and Engineering Chemistry Research*, vol. 41, no. 18, pp. 4616–4625, 2002.

[3] W. Chunfeng and Z. Xin, "Ants foraging mechanism in the design of multiproduct batch chemical process," *Industrial and Engineering Chemistry Research*, vol. 41, no. 26, pp. 6678–6686, 2002.

[4] H. D. Goel, M. P. C. Weijnen, and J. Grievink, "Optimal reliable retrofit design of multiproduct batch plants," *Industrial and Engineering Chemistry Research*, vol. 43, no. 14, pp. 3799–3811, 2004.

[5] S.-K. Heo, K.-H. Lee, H.-K. Lee, I.-B. Lee, and J. H. Park, "A new algorithm for cyclic scheduling and design of multipurpose batch plants," *Industrial and Engineering Chemistry Research*, vol. 42, no. 4, pp. 836–846, 2003.

[6] J. M. Montagna, A. R. Vecchietti, O. A. Iribarren, J. M. Pinto, and J. A. Asenjo, "Optimal design of protein production plants with time and size factor process models," *Biotechnology Progress*, vol. 16, no. 2, pp. 228–237, 2000.

[7] A. Chakraborty, A. Malcolm, R. D. Colberg, and A. A. Linninger, "Optimal waste reduction and investment planning under uncertainty," *Computers & Chemical Engineering*, vol. 28, no. 6-7, pp. 1145–1156, 2004.

[8] L. Cavin, U. Fischer, F. Glover, and K. Hungerbühler, "Multiobjective process design in multi-purpose batch plants using a Tabu Search optimization algorithm," *Computers & Chemical Engineering*, vol. 28, no. 4, pp. 459–478, 2004.

[9] T. Pinto, A. P. F. D. Barbosa-Póvoa, and A. Q. Novais, "Optimal design and retrofit of batch plants with a periodic mode of operation," *Computers & Chemical Engineering*, vol. 29, no. 6, pp. 1293–1303, 2005.

[10] H.-J. Huang and F.-S. Wang, "Fuzzy decision-making design of chemical plant using mixed-integer hybrid differential evolution," *Computers & Chemical Engineering*, vol. 26, no. 12, pp. 1649–1660, 2002.

[11] S. Dedieu, L. Pibouleau, C. Azzaro-Pantel, and S. Domenech, "Design and retrofit of multiobjective batch plants via a multicriteria genetic algorithm," *Computers & Chemical Engineering*, vol. 27, no. 12, pp. 1723–1740, 2003.

[12] M. Solimanpur, P. Vrat, and R. Shankar, "A multi-objective genetic algorithm approach to the design of cellular manufacturing systems," *International Journal of Production Research*, vol. 42, no. 7, pp. 1419–1441, 2004.

[13] C. Dimopoulos, "Multi-objective optimization of manufacturing cell design," *International Journal of Production Research*, vol. 44, no. 22, pp. 4855–4875, 2006.

[14] J.-P. Brans and B. Mareschal, "The PROMCALC and GAIA decision support system for multicriteria decision aid," *Decision Support Systems*, vol. 12, no. 4-5, pp. 297–310, 1994.

[15] V. Bhaskar, S. K. Gupta, and A. K. Ray, "Applications of multiobjective optimization in chemical engineering," *Reviews in Chemical Engineering*, vol. 16, no. 1, pp. 1–54, 2000.

[16] C. Fonseca and P. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1993.

[17] J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications (ICGA '85)*, pp. 93–100, Pittsburgh, Pa, USA, July 1985.

[18] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man and Cybernetics, Part C*, vol. 28, no. 3, pp. 392–403, 1998.

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

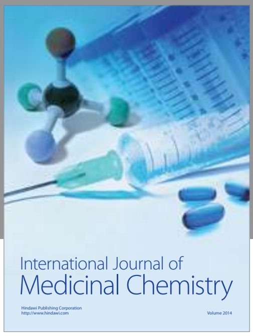
[20] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proceedings of the 1st IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence (ICEC '94)*, vol. 1, pp. 82–87, Orlando, Fla, USA, June 1994.

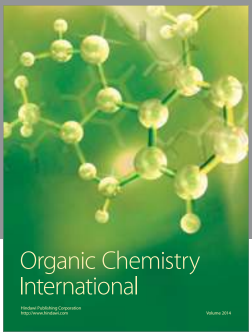
[21] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

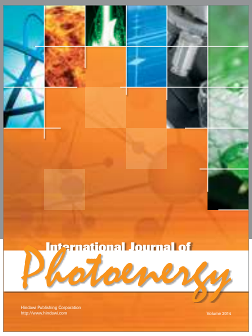
[22] K. C. Tan, T. H. Lee, and E. F. Khor, "Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 6, pp. 565–588, 2001.

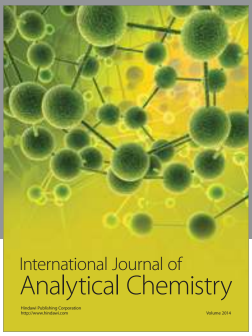
[23] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

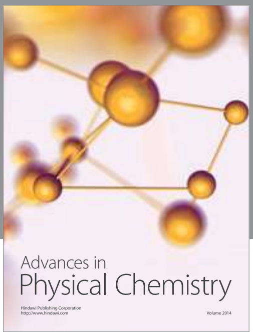
[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

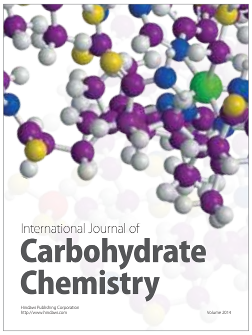

International Journal of
Medicinal Chemistry

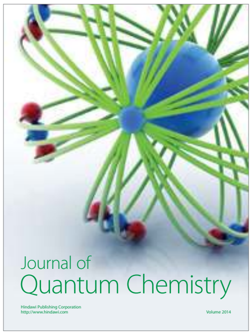

Organic Chemistry
International


International Journal of
Photoenergy


International Journal of
Analytical Chemistry


Advances in
Physical Chemistry

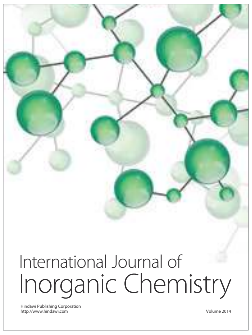

International Journal of
Carbohydrate
Chemistry


Journal of
Quantum Chemistry

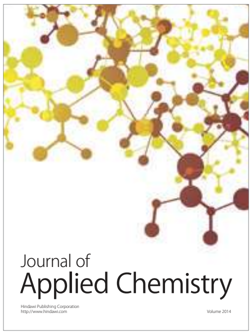
The Scientific
World Journal

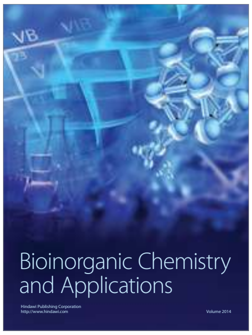
Journal of
Analytical Methods
in Chemistry

Journal of
Spectroscopy

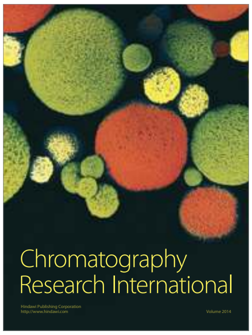

International Journal of
Inorganic Chemistry

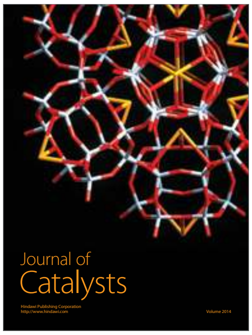
International Journal of
Electrochemistry

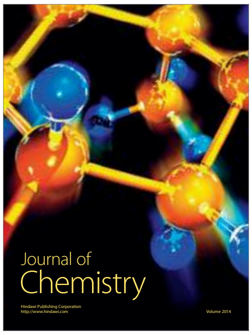

Journal of
Applied Chemistry


Bioinorganic Chemistry
and Applications

Journal of
Theoretical Chemistry


Chromatography
Research International


Journal of
Catalysts


Journal of
Chemistry

International Journal of
Spectroscopy