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 multiobjective optimization problem is illustrated through two carefully referenced examples.

Copyright © 2009 D. Mokeddem and A. Khellaf. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. Introduction

Batch processes are used in production of many low-volume but high-value-added products (such as specialty 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 plants design 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 rise to a family of nondominated solutions, largely to manufacture identical units in parallel operating out of phase,

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.

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 there are N_i 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.

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 while maximizing the revenue, simultaneously requires the investment but also minimizing the operation cost.

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) $B_j$ size of the batch of product $i$ at the end of the $M$ stages,
(iv) $T_L$ 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 $i$:

\[ S_i B_i \leq V_j, \quad \forall i = 1, \ldots, N; \quad \forall j = 1, \ldots, M. \]  

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

\[ \sum_{i=1}^{N} \frac{Q_i}{B_i} T_L \leq H. \]  

(3) The limiting cycle time for product $i$:

\[ \frac{\tau_{ij}}{N_j} \leq T_L, \quad \forall i = 1, \ldots, N; \quad \forall j = 1, \ldots, M. \]

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

\[ V_j^{\text{min}} \leq V_j \leq V_j^{\text{max}}, \quad \forall j = 1, \ldots, M, \]
\[ B_j^{\text{min}} \leq B_j \leq B_j^{\text{max}}, \quad \forall j = 1, \ldots, N. \]

### 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 assign the solutions in fronts (nondominated arrangement), 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 ($\preceq_n$), which guides to the genetic algorithm, towards the Pareto optimal front, in accordance with the following criterion:

\[ i \preceq_n j \text{ if } (i_{\text{rank}} < j_{\text{rank}}), \]
\[ \text{or } (i_{\text{rank}} = j_{\text{rank}}) \text{ and } (i_{\text{distance}} > j_{\text{distance}}). \]  

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 $(P_0)$ 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 $N$. 


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

<table>
<thead>
<tr>
<th>Product</th>
<th>Processing times, ( \tau_{ij} ) (h)</th>
<th>Unit price for the product ($/Kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mixer</td>
<td>Reactor</td>
</tr>
<tr>
<td>A</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>Size factors (L/kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
</tr>
</tbody>
</table>

Cost of equipment ($, V in litres)
- Minimum size = 250 L
- Maximum size = 2500 L

<table>
<thead>
<tr>
<th>Operating cost factor (( C_e ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>15</td>
</tr>
</tbody>
</table>
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

\[ f^L_i \leq f_i \leq f^U_i, \quad i = 1, \ldots, 4. \] (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,
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: $f_1 = 121350$, $f_2 = 171624$, $f_3 = 77299$, $f_4 = 5667$. 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

---

Table 2: Data used in example 2.

<table>
<thead>
<tr>
<th>Product</th>
<th>Processing times, $T_{ij}(h)$</th>
<th>Unit price for the product ($$/Kg)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product</td>
<td>A 1.15 B 5.95 C 3.96</td>
<td>$C_p$ A 0.27 B 0.29 C 0.32</td>
</tr>
<tr>
<td>Extractor</td>
<td>9.86</td>
<td></td>
</tr>
<tr>
<td>Centrifuge</td>
<td>0.4 0.7 0.85</td>
<td></td>
</tr>
<tr>
<td>$y_{ij}$</td>
<td>0.4 0.33 0.3</td>
<td></td>
</tr>
<tr>
<td>Size factors (L/kg)</td>
<td>A 8.28</td>
<td>9.7</td>
</tr>
<tr>
<td>B 5.58</td>
<td>8.09</td>
<td></td>
</tr>
<tr>
<td>C 2.34</td>
<td>10.3</td>
<td></td>
</tr>
<tr>
<td>$\gamma_{ij}$</td>
<td>A 0.4 0.24</td>
<td>0.4</td>
</tr>
<tr>
<td>B 0.15</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>C 0.34</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>Coefficients $C_{ij}$</td>
<td>A 0.2</td>
<td>0.24</td>
</tr>
<tr>
<td>B 0.15</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>C 0.34</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>Cost of equipment ($$$, V in litres)</td>
<td>250 V^{0.6} 250 V^{0.6} 250 V^{0.6} 250 V^{0.6}</td>
<td></td>
</tr>
<tr>
<td>Minimum size = 250 L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximum size = 10000 L</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Operating cost factor ($C_E$)

<table>
<thead>
<tr>
<th>Operating cost factor ($C_E$)</th>
<th>20</th>
<th>30</th>
<th>15</th>
<th>30</th>
</tr>
</thead>
</table>
Table 3: Optimal objective function values of example 1.

<table>
<thead>
<tr>
<th>Case</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>122566.41</td>
<td>175955.39</td>
<td>73499.23</td>
<td>5579.37</td>
</tr>
<tr>
<td>2</td>
<td>124283.15</td>
<td>177389.07</td>
<td>74508.91</td>
<td>5582.87</td>
</tr>
<tr>
<td>3</td>
<td>126393.23</td>
<td>173354.67</td>
<td>84896.90</td>
<td>5638.60</td>
</tr>
</tbody>
</table>

Bounds for objective function: $[f_L^1, f_U^1] = [110000, 130000]$, $[f_L^2, f_U^2] = [150000, 200000]$, $[f_L^3, f_U^3] = [60000, 100000]$, $[f_L^4, f_U^4] = [5500, 6000]$.

Table 4: Optimal objective function values of example 2.

<table>
<thead>
<tr>
<th>Case</th>
<th>$f_1$</th>
<th>$f_2$</th>
<th>$f_3$</th>
<th>$f_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>275766.10</td>
<td>388262.10</td>
<td>156449.20</td>
<td>5505.40</td>
</tr>
<tr>
<td>2</td>
<td>276096.30</td>
<td>369977.30</td>
<td>161392.30</td>
<td>5710.90</td>
</tr>
<tr>
<td>3</td>
<td>281818.70</td>
<td>369843.48</td>
<td>163552.80</td>
<td>5718.00</td>
</tr>
</tbody>
</table>

Bounds for objective function: $[f_L^1, f_U^1] = [250000, 300000]$, $[f_L^2, f_U^2] = [350000, 400000]$, $[f_L^3, f_U^3] = [150000, 200000]$, $[f_L^4, f_U^4] = [5500, 6000]$.
with unit reference membership level for all objectives 

\( f_1 = 274 \, 312, \quad f_2 = 375 \, 688, \quad f_3 = 175 \, 688, \quad f_4 = 5639 \) \n
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 fulfill 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)
- \( C_{Ej} \): Operation cost in stage \( j \) (\$)
- \( C_{Dj} \): Operation cost of product \( i \) to be produced (\$/kg)
- \( C_R \): 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

- \( \alpha_j \): Cost coefficient for unit \( j \)
- \( \beta_j \): Cost exponent for unit \( j \)
- \( \tau_{ij} \): Processing time of product \( i \) in stage \( j \) (h).
References


Submit your manuscripts at http://www.hindawi.com