Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Multi-Objective Optimization Issues in Short-Term Batch Scheduling

Prakash R. Kotecha*. Mangesh D. Kapadi**
Mani Bhushan*, Ravindra D. Gudi*

**Department of Chemical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai400 076 India (Tel: 91-22-2576 7204; e-mail: ravigudi@iitb.ac.in).*
***Automation Lab, Research & Technology, Honeywel Technology Solutions, Bannerghatta Road, Bangalore 560 076,India*

Abstract: In this article, we propose a multiple objective optimization based approach for the short term scheduling of batch plants to select superior solutions when compared to the single objective problem. Two alternate approaches to optimality, viz. lexicographic and pareto-optimality based formulations are considered here. We demonstrate the suitability of lexicographic optimization for the case when the importance associated with the objectives is known a priori. Next, we also show the practicality of the pareto-optimization based approach when an explicit precedence ordering of the objectives is not known. Case studies involving the objectives of make-span minimization and profit maximization problem are considered here in the discrete state task network representation of (Kondili *et al.*, 1993), to demonstrate the practicality of the above approaches, towards deciding on operating schedules.

1. INTRODUCTION

The short term scheduling of batch plants is important from the viewpoint of chemical industry as it allows the production of a wide range of products even in small amounts within the same production facility. Moreover, such plants can be operated over a wide range of operating conditions for diverse product specifications. These scenarios are frequently found in the manufacture of specialty materials, pharmaceutical, polymers, bio-chemicals and food products. This has led to a considerable attention being given to the short term scheduling of batch plants in literature. The classifications of batch plants have been done on various criteria and are categorized into multi-product or multi-purpose batch plants, sequential or network represented processes. In addition, the batch plants can also have different storage policies such as Zero Wait, Finite Intermediate Storage, No Intermediate Storage, and Unlimited Intermediate Storages. Further, the schedules for batch operation policies are classified based on time representation. Broadly, there have been two approaches on the representation of time in batch scheduling. The discrete-time state task network (STN) representation was presented by (Kondili *et al.*, 1993) while continuous time representations were proposed by (Zhang and Sargent, 1996) and (Mockus and Reklaitis, 1997). The discrete time formulation requires prior knowledge of the time horizon and this horizon is further divided into smaller time intervals such that their length equals to the greatest common factor of the processing times. The processing times of all the tasks are assumed to be constant over the scheduling horizon thereby allowing the determination of the length of time interval. The drawbacks of this representation include the requirement of the processing time to be constant and also the fact that these may prove computationally expensive when the length of the time interval becomes very small. One important advantage of the discrete-time representation is its reference grid against which all the operations competing for shared resources can be positioned and thus facilitating easier modeling of the problem. On the other hand, the continuous time representations are considered to be more accommodative in nature but these have been reported to be computationally expensive for larger problems as their LP relaxation gets poor due to the large number of big-M constraints. Many different objectives have been considered in literature for the short term scheduling problem such as the maximization of profit, the minimization of make-span, the minimization of production cost and the maximization of total sales. An important characteristic of these scheduling problems is that there are multiple optimal solutions which can be exploited to satisfy more than one objective. A detailed discussion on the various aspects of the scheduling of batch plants can be found in the state-of-the-art review article (Mendez *et al.*, 2006). Most of the work in literature has focused on efficiently solving the scheduling problem for longer horizons with the help of either decomposition techniques or hybrid techniques (Jain and Grossmann, 2001), and (Maravelias and Grossmann, 2003).

As mentioned earlier, the solution space of the batch scheduling problem is characterized by computational complexity, multiple objectives as well as existence of multiple solutions. In this paper, we approach the batch scheduling problem from a multiple objective optimization perspective. Specifically, we analyze the optimality in the presence of multiple objectives, in a lexicographic and pareto-optimal sense. When the precedence ordering of the objectives are explicitly known to the designer, lexicographic optimization methods help to suitably reflect these choices and generate optimal solution(s) that represent sequential satisfaction of the objectives. Here we propose to exploit these methods to analyze the quality of the optimal solution under different known precedence ordering of the batch scheduling objectives mentioned above. For the case when the designer has no particular preference for any of these objectives, it is important to analyze the trade-offs that exists between these objectives, in a pareto-optimal sense that looks at all non-inferior solutions. In this paper we consider both the above approaches towards solving the short-term scheduling problem. We assess the suitability of both these methods using the discrete STN based formulation (Kondili *et al.*, 1993).

The following section gives a brief overview of the discrete STN model. The next section is on multi-objective optimization wherein we present the lexicographic based and the ε-constraint based multi-objective formulations for the batch scheduling problem. This is followed by a benchmark case study from the literature on which the proposed strategies are demonstrated.

2. MATHEMATICAL FORMULATION

This section contains the necessary mathematical formulations that are used for the succeeding discussions. In this article, we adopt the following nomenclature for the various parameters and variables.

2.1 Nomenclature

Indices

i	task
j	unit
n	time point
s	state

Sets

F	set of feeds
I	set of processing task
J	set of units
N	total number of time points
P	set of products
S	set of states
IT	set of intermediates

Known Parameters

B_i^{max}	maximum capacity for task i in suitable unit
B_i^{min}	minimum capacity for task i in suitable unit
C_F	cost of the feed F
C_P	cost of the product P
C_s	cost of the state S
H	horizon period
ST_s^0	initial available amount of state s
ST_s^{max}	maximum storage capacity of state s
$U_{i,j}$	task unit suitability matrix i.e., $U_{i,j} = 1$ if the i^{th} task is suitable to be processed in unit j or zero otherwise.
τ_i	processing time for task i in suitable unit in timepoints
$\rho_{i,s}$	proportion of state s produced or consumed

Variables

Binary Variable

$W_{i,n}$	binary variable, equals one to indicate the processing of task i at time point n and zero otherwise.

Continuous Variable

$b_{i,n}^s$	amount of material started processing for task i at time point n
$st_{s,n}$	amount of state s at time point n

2.2 Constraints

A batch scheduling problem involves the determination of the time at which a task is to be started and the amount of material that gets processed in each task at every time-point. The following sets of constraints are similar to that in (Kondili *et al.*, 1993) and helps in the determination of the schedules and batch sizes. This model additionally considers task decoupling (Ierapetritou and Floudas, 1998) wherein if a task i can be performed in two units j and j', then an artificial task (i') is included such that the task i takes place in unit j and the task i' takes place in unit j'. This task decoupling helps in handling the unit dependent processing time for a task in a straight forward manner.

$$\sum_{i \in I} U_{i,j} W_{i,n} \le 1 \qquad \forall\, j,n \tag{1}$$

$$\sum_{i \in I} U_{i,j} W_{i,N} = 0 \qquad \forall\, j \tag{2}$$

$$\sum_{i' \in I} \sum_{n'=n}^{n+\tau_i-1} U_{i',j} W_{i',n'} - 1 \le M\left(1 - U_{i,n} W_{i,n}\right) \forall\, j,n \tag{3}$$

$$st_{s,n} = st_{s,n-1} + \sum_{i:\rho_{i,s}>0,\tau_i<n} b_{i,n-\tau_i}^s \rho_{i,s} + \sum_{i:\rho_{i,s}<0} b_{i,n}^s \rho_{i,s} \qquad \forall s, n > 1 \tag{4}$$

$$st_{s,1} = ST_s^0 + \sum_{i:\rho_{i,s}<0} b_{i,1}^s \rho_{i,s} \qquad \forall s \tag{5}$$

$$st_{s,n} \le ST_s^{max} \qquad \forall s,n \tag{6}$$

$$W_{i,n} B_i^{min} \le b_{i,n}^s \le W_{i,n} B_i^{max} \qquad \forall i,n \tag{7}$$

$$\sum_{i \in I} \sum_{n} U_{i,j} W_{i,n} \tau_i \le N - 1 \qquad \forall\, j \tag{8}$$

$$\sum_{n} W_{i,n} \tau_i \le N - 1 \qquad \forall i \tag{9}$$

$$W_{i,n} \le b_{i,n}^s \qquad \forall i,n \tag{10}$$

Constraint (1) ensures that only one task is being processed in a unit at a given time-point. Constraint (2) ensures that no task is being started at the end of the horizon. Constraint (3) ensures that the operation is non-pre-emptive i,e., no other task can start in an unit until the current task has finished. Constraint (4) and (5) conserve the mass balance at any time-point and the first time-point respectively. Constraint (6) bounds the amount of state at any time to be within the maximum storage capacity. Constraint (7) ensures that the amount of material that is being processed in any unit at any given point is bounded by the upper and lower capacities of that unit. Constraint (8) and (9) are tightening constraints and do not affect the optimality of the solution. Constraint (10) makes sure that a task in a unit takes place at any time if only the amount of material processed in that unit is greater than zero. This constraint also does not affect the optimality of the results but avoids redundant results $\left(W_{i,n}=1 \text{ and } b_{i,n}^{s}=0\right)$. However, this constraint can be neglected for those tasks which have $B_i^{min}>0$. It should be noted the above set of constraints do not consider constraints on utilities and demands. These factors can be appropriately incorporated and have been eliminated here for the sake of simplicity. However, the rest of the article will still hold irrespective of the inclusion of such criteria.

2.3 Objective Function

As mentioned earlier, we consider the twin objectives of maximization of profit and the minimization of the make-span. These requirements can be mathematically represented by the following equations.

Minimization of the make-span: We need to include the following constraint to the set of constraints from (1) to (10) so as to accommodate this criterion.

$$W_{i,n}\left(n+\tau_i\right) \le ms \quad \forall i,n \tag{11}$$

The above constraint ensures that the continuous variable ms take the value of the maximum time of all the tasks that get processed. Since, make-span involves the minimization of the maximum time, the objective function can be written as

$$Min \quad ms \tag{12}$$

Maximization of the Profit: Profit is the difference between the revenue produced by sales and the amount spent on the production and can be given by

$$\begin{aligned} Max \quad Profit = & \sum_{s\in(P\cup IT)} C_P st_{s,N} - \sum_{s\in F} C_F\left(ST_s^0 - st_{s,N}\right) \\ & - \sum_{s}\sum_{n=1}^{N} C_s st_{s,n} \end{aligned} \tag{13}$$

The first term on the RHS corresponds to the amount of revenue realized by selling the products while the second term indicates the amount of money spent on the feed with the third term indicating the amount of money spent on the storage. The above expression does not account for the equipment setup costs and the manpower costs. However, these can be included if necessary.

In the following discussions, Formulation I will indicate the maximization of the profit and will include the objective function in (13) and the set of constraint from (1) to (10). Formulation II will indicate the minimization of the make-span and will include the constraint set from (1) to (11) with the objective function as in (12).

3. MULTI-OBJECTIVE OPTIMIZATION

First, we present the lexicographic optimization procedure that helps in the determination of schedules with minimum make-span without reducing the maximum profit that can be derived from the plant. This is followed by the ε-constraint based multi-objective optimization approach to determine the pareto-fronts.

31 Lexicographic Optimization

In this section, we address the issue of selecting a promising solution to the short term scheduling problem from the set of multiple optimal solutions (obtained for any of the single objective problem). The determination of such a solution is important because the designer would like to select a solution that is optimal to as many objectives as possible. For example, a designer would like to maximize the profit and as well like to minimize the make-span. The traditional way of solving the maximization of profit (or the minimization of the make-span) does not guarantee satisfaction of both of these criteria. The solution may be optimal with respect to the maximization of profit but the solution need not be the best as there may be another solution which can have the same optimal profit and yet have a smaller make-span. A naïve way to address this issue would be to determine all the optimal solutions for the primary objective (in this case, maximization of the profit) and select the solution which has a better secondary objective (for example the least make-span). However, this would require the determination of all the possible optimal solutions involving the construction of integer cuts and may prove computationally expensive. Another approach would be to solve a subsequent optimization problem with the secondary objective such that the solution to this problem is optimal not only for the secondary problem but is also optimal to the primary problem. In multi-objective optimization literature, this procedure is known as lexicographic optimization. *Lexicographic optimization (or preemptive optimization) is a special form of multi objective optimization that characterizes trade-offs between the various objectives which are given a precedence ordering. The primary objective has the highest priority followed by decreasing priorities on subsequent objectives.* The philosophy in this approach is that even a marginal improvement in a higher precedence objective is considered more valuable than an arbitrarily large improvement in a lower ranked objective. It can be noted that the lexicographic optimization requires the specification of a precedence level between these objectives as primary and secondary objective and hence lexicographic optimization is useful if the designer has an explicit knowledge of the precedence level.

In the following analysis, we assume that the maximization of the profit is a primary criterion and the minimization of the make-span is a secondary criterion. However, this precedence is completely arbitrary and depends on the need of the designer. The need of lexicographic optimization can be understood from the Figure 1. Let f_1 represent the maximization of the profit and f_2 represent the minimization of the make-span. If we were to solve the problem of maximization of profit, we can get either point A or point B. This is because both A and B have the same optimal solution but it can be seen that point A is a better solution to point B as it has a smaller make-span than point B. Thus, it is necessary to have a strategy to handle these issues. The following two-step procedure will address this issue and thereby uncover potentially useful solutions.

Step 1: Solve Formulation I for a given horizon to determine the maximum profit denoted as P^{max} .

Step 2: Solve Formulation II along with the following additional constraint to obtain a better solution (if any) which will make the same amount of profit in a smaller horizon than H.

$$\sum_{s\in(P\cup IT)} C_P st_{s,N} - \sum_{s\in F} C_F \left(ST_s^0 - st_{s,N}\right) - \sum_{s}\sum_{n=1}^{N} C_s st_{s,n} = P^{max} \quad (14)$$

The solution obtained in Step 2 is also an optimal solution to Step 1 and this solution can be uncovered in Step 1 if all the multiple optimal solutions are determined. It can be seen that the solution of Step 2 will be superior (if not the same) to Step 1 for it will have the maximum possible profit in the given horizon and will also schedule the tasks such that they are completed in the earliest possible time. The determination of such points may not have been possible if the problem was solved as in Step 1 alone. This two-step approach has been demonstrated on the benchmark problem (Kondili *et al.*, 1993) in a subsequent section.

Remark: It may be possible that there are multiple-optimal solutions even to the Step 2. In such cases, the designer can add any other criteria in a similar manner such that the optimality of the solution to Step 1 and Step 2 is preserved and is yet superior to other solutions in the third objective.

3.2 Determination of Pareto-optimal front

In lexicographic optimization, we had assumed that an explicit ordering between the various objectives is known a priori. However, it may not be possible for the designer to always suggest an explicit precedence level. For example, a designer may not able to prioritize between the make-span and the profit. In such scenarios, it would be beneficial to determine the trade-offs between these conflicting objectives. Such trade-offs are termed as pareto-optimal fronts in the multi-objective literature. The pareto-optimal front (Deb, 2001) is a collection of the set of non-dominated solutions. A solution is said to be non-dominated if it is feasible and there is no other feasible solution which has better values for all of the objectives. In this section, we use the ε constraint based method (Deb, 2001) to determine the pareto-points to study the tradeoffs between the twin objectives of profit maximization and minimization of the make-span. The ε-constraint method involves posing all the objectives except one into constraints. For example, if there are M objectives, the solution of the following problem will yield a pareto-point.

$$\begin{array}{ll} \text{Min} & f_\mu(\mathbf{x}), \\ \text{Subject to} & f_m(\mathbf{x}) \le \varepsilon_m, m = 1,2,..,M \text{ and } m \ne \mu \\ & g_j(\mathbf{x}) \ge 0, \quad j = 1,2,..,J; \\ & h_k(\mathbf{x}) = 0, \quad k = 1,2,..,K; \\ & x_i^{(L)} \le x_i \le x_i^{(U)}, i = 1,2,..,n; \end{array} \quad (15)$$

In the above formulation, f_μ indicates the μ^{th} objective and ε_m indicates the minimum performance required on the m^{th} objective whereas $g(\mathbf{x})$ and $h(\mathbf{x})$ indicate the inequality and equality constraints respectively. The complete pareto-front can be obtained by solving the above problem with different performance criteria. An important advantage of using this method over other methods is that it helps in the determination of non-convex pareto-fronts. However, the application of the above traditional strategy to problems with multiple solutions suffers from a severe drawback that it can lead to sub-optimal pareto-front. To understand this, consider the scenario in Figure 1 wherein one of the objectives has multiple solutions. Let us assume that the objectives f_1 and f_2 need to be minimized and we minimize f_1 with a performance criteria on f_2 i.e.., $f_2 \le \varepsilon_2$. Under this circumstance, the solution to the problem in (15) may be either A or B. But, it is clear that B is not a pareto-optimal solution when compared to A as the solution A has a better objective function value (f_2) than B.

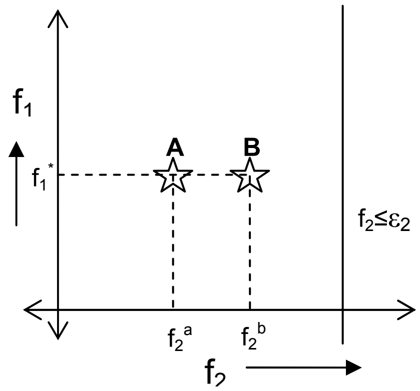

Fig. 1. Issues in ε constraint method.

This can be avoided by solving a secondary problem similar to the Step 2 of the lexicographic formulation. Now, we present the steps needed to determine the optimal pareto-front between the two objectives of maximization of profit and minimization of the make-span.

Step 1: Assume a minimum amount of desired profit (P_m) and solve Formulation II along with the following constraint.

$$\left(\begin{array}{l} \sum_{s\in(P\cup IT)} C_P st_{s,N} - \sum_{s\in F} C_F \left(ST_s^0 - st_{s,N}\right) - \\ \sum_{s}\sum_{n=1}^{N} C_s st_{s,n} \end{array}\right) \ge P_m \quad (16)$$

Let the minimum make-span be ms_m^*.

Step 2: Solve Formulation I with the following constraint to determine the maximum profit that can be obtained.

$$W_{i,n}\left(n+\tau_i\right) \le ms_m^* \quad \forall i,n \tag{17}$$

It can be seen that the constraint in (17) is similar to that in Equation (11) and ensures that the optimality obtained for make-span in Step 1 is maintained in Step 2 also. Let the maximum profit obtained in Step 2 be P_m^*. Now, the point $\left(P_m^*, ms_m^*\right)$ corresponds to one point on the pareto-front while the complete pareto-front can be obtained as shown in Step 3.

Step 3: Solve Steps 1 and 2 in sequence for different values of $\left(P_m\right)$ to obtain the complete pareto-front.

In most circumstances, the profit increases with an increase in the make-span. This is because the amount of material (and hence the profit) keeps increasing as the make-span increases. Hence, it would be better to evaluate the trade-offs between the make-span and the profit obtained for every unit of the make-span instead of profit made during the entire make-span. This can be easily obtained from the set of solutions $\left(P_m^*, ms_m^*\right)$ and the pareto-front can be determined from the set $\left(ms_m^*, P_m^* / ms_m^*\right)$ by a straight-forward post-optimality analysis.

4. CASE STUDY

In this section, we show the application of the lexicographic and the multi-objective optimization approach for the discrete STN benchmark example of (Kondili *et al.*, 1993). The STN network with five tasks and nine states is as shown in Figure 2. It also contains the processing times (in hrs) and the proportion of the state produce and consumed.

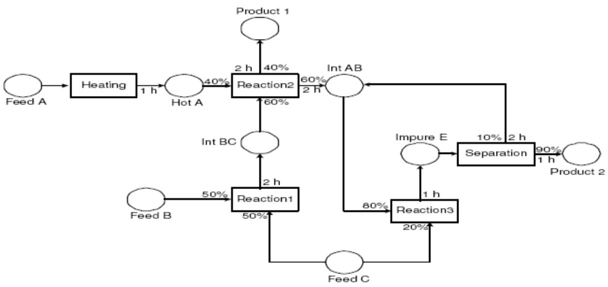

Fig. 2. STN representation of the Case Study.

The equipments available and their processing capacities for a batch are listed in Table 1.

Table 1. Data for unit suitability and capacities

Unit	Tasks Suitable	Capacity (kg)
Heater	Heating	100
Reactor 1	Reactions 1,2 &3	80
Reactor 2	Reactions 1,2 &3	50
Still	Separation	200

It can be seen that the total number of tasks increases to eight due to the task decoupling. The available storage capacities of the intermediate states are given as follows

Hot A	100 kg
Intermediate AB	200 kg
Intermediate BC	150 kg
Impure E	100 kg

The feed and products are considered to be having unlimited storage capacity. The cost of the products P1 and P2 is 10 units/kg and the feed stocks are assumed to have zero cost. Additionally, we penalize the excess production of intermediates at the end of the horizon by -1. These data are similar to that considered (Kondili *et al.*, 1993). We would be referring to this set of data as Case I and Case II would consist of the same data except that the storage of intermediate BC will be zero.

Table 2 shows the results for the maximization of the profit for 3 different horizons.

Table 2. Results for the maximization of profit

Horizon (hr)	Maximization of Profit (unit)
5	442
10	2744.375
15	4723.083

It has to be noted that for the above cases, the complete horizon is scheduled. Next we solve the same problem by the proposed two-step lexicographic approach and the results are as shown in Table 3.

Table 3. Minimization of the makespan in sequence with maximization of the profit

Horizon (hr)	Maximization of Profit (unit)	Minimization of the Make Span (hr)
5	442	4
10	2744.375	10
15	4723.083	15

From Table 3, it can be seen that a better schedule is obtained for the horizon 5. It is to be noted that the profit remains at 442 and yet the make-span is reduced to 4 hours. This is because the Formulation I requires an apriori specification of the horizon and on the specification of the horizon, the formulation only maximizes the profit without considering the make-span. It uses the complete horizon without any consideration towards minimizing the make-span. A similar behavior can also be observed for the Case II. Thus it can be seen that this procedure helps to uncover promising solutions. We now demonstrate the determination of pareto-optimal for Case I. Table 4 shows the various trade-offs that were obtained for different levels of desired profit.

Table 4. Trade-offs between Profit and Make-span for Case I

P_m	ms_m^*	P_m^*

200	4	442
443	6	754
741	7	1682.41
1683	8	1829.75
1830	9	2315
2316	10	2744.375
2745	11	3199.719
3200	12	3602.875
3603	13	3970.833
3971	14	4254.167
4255	15	4723.083
4724	16	5123.208
5124	17	5491.167

Figure 3 shows the above points with respect to the profit per hour and the make-span in hours. It can be seen that the pareto-front comprises of all the points except D and J. This is because the point D is inferior to point C as its make-span is larger than point C and yet its profit per hour is lees than C.

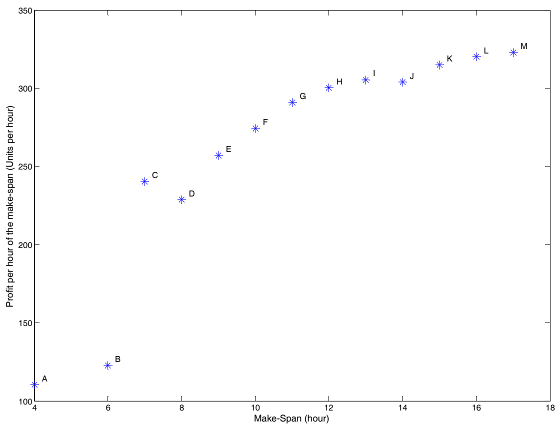

Fig. 3. Trade-offs between make-span and profit per hour for Case I.

Similar arguments also hold for the Case II in Figure 4. The points a, c, d, g, h, j, and k form the pareto-front whereas the remaining points are dominated solutions.

6. CONCLUSIONS

In this work, we have demonstrated the benefits of multi-objective optimization over the single-objective optimization framework for the short-term scheduling of batch plants. We have presented techniques to solve the multi-objective optimization both with known and unknown precedence level in the objectives. Future extensions of this work can include the solution of a single optimization problem with the lexicographic pattern instead of solving two separate problems. However, this could have computational issues as the make-span minimization problem has been reported to have a weak LP relaxation due to its min-max nature. Hence it would be interesting to see the behavior of problems that combine both the maximization of profit and minimization of the make-span in a single objective.

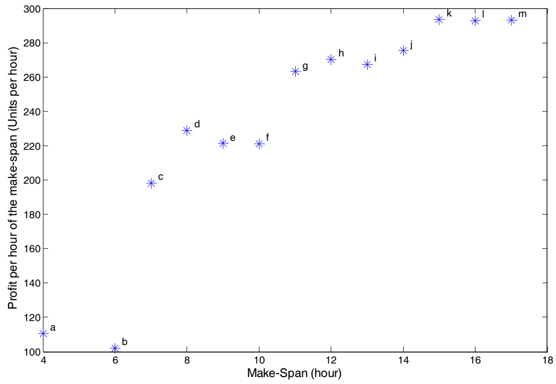

Fig. 4. Trade-offs between make-span and profit per hour for Case II.

REFERENCES

Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA, John Wiley & Sons, Inc.

Ierapetritou, M. G. and C. A. Floudas (1998). Effective Continuous-Time Formulation for Short-Term Scheduling. 1. Multipurpose Batch Processes. *Ind. Eng. Chem. Res.* **37**(11): 4341-4359.

Jain, V. and I. E. Grossmann (2001). Algorithms for Hybrid MILP/CP Models for a Class of Optimization Problems. *Informs Journal On Computing* **13**(4): 258-276.

Kondili, E., C. C. Pantelides and R. W. H. Sargent (1993). A general algorithm for short-term scheduling of batch operations--I. MILP formulation. *Computers & Chemical Engineering* **17**(2): 211-227.

Mendez, C. A., J. Cerda, I. E. Grossmann, I. Harjunkoski and M. Fahl (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering* **30**(6-7): 913-946.

Mockus, L. and G. V. Reklaitis (1997). Mathematical programming formulation for scheduling of batch operations based on nonuniform time discretization. *Computers & Chemical Engineering* **21**(10): 1147-1156.

Zhang, X. and R. W. H. Sargent (1996). The optimal operation of mixed production facilities--a general formulation and some approaches for the solution. *Computers & Chemical Engineering* **20**(6-7): 897-904.