Available online at www.sciencedirect.com

ScienceDirect

Progress in Natural Science 18 (2008) 487–493

Progress in Natural Science

www.elsevier.com/locate/pnsc

Special report

Efficient short-term scheduling of multiproduct batch plants under demand uncertainty

Jin Zhu [a], Xingsheng Gu [a,*], Bin Jiao [b]

[a] Research Institute of Automation, East China University of Science and Technology, Shanghai 200237, China
[b] Electrical Engineering Department, Shanghai DianJi University, Shanghai 200240, China

Received 29 October 2007; received in revised form 12 November 2007; accepted 12 November 2007

Abstract

In this paper, a novel formulation for the short-term scheduling of multiproduct batch plants under demand uncertainty is presented. Then it is solved by an improved genetic algorithm. The proposed approach results in an efficient utilization of the plant capability as it allows the optimal selection among all the rescheduling alternatives in a systematic way without the use of any heuristics. Moreover, the objective function can not only maximize the total profit of the plant and minimize the makespan but also allow the flexibility for modeling different weighted instances of the two targets so that a best-possible decision can be determined. According to the discrete characteristic of scheduling of batch plants, through the improvement of the coding method, an effective genetic algorithm is presented. Two examples are given to illustrate the effectiveness of the proposed formulation and algorithm.

Keywords: Scheduling; Batch plants; Demand; Uncertainty

1. Introduction

Batch plants have been widely practiced since long before the development of the modern chemical industry [1,2]. The effective scheduling for batch plants can not only increase the customer services and reduce the need for excess capacities of corporations but also provide the potential of identifying key data and mechanistic understandings of processes. So, it is significant to do further research on scheduling for batch plants.

Most of the work in this area has been limited to deterministic approaches, wherein the problem parameters are assumed to be known with certainty. However, various uncertain effects are inevitable in reality [3], such as processing time, costs and demand. Failure to properly account for product demand fluctuations may lead to unsatisfied customer demands, loss of market share, and excessive inventory costs. Recently, the scheduling of batch plants under demand uncertainty has emerged as an area of active research.

Sand et al. [4] proposed an algorithm to approximate the performance of an ideal online scheduler for a multiproduct batch plant under demand uncertainty. They used a two-level hierarchical framework which consisted of a stochastic linear program for the purpose of long-term planning and a deterministic nonlinear model for short-term scheduling. Gupta et al. [5] used a chance-constrained approach in conjunction with a two-stage stochastic programming model to analyze the trade-offs between demand satisfaction and production costs for a mid-term supply chain planning problem. Vin and Ierapetritou [6] addressed the problem of quantifying schedule robustness under demand uncertainty. They used multiperiod programming to obtain schedules that were feasible

* Corresponding author. Tel.: +86 21 64253121; fax: +86 21 64252576.
E-mail address: xsgu@ecust.edu.com (X. Gu).

1002-0071/$ - see front matter
doi:10.1016/j.pnsc.2007.11.016

over an entire range of parameters, and also proposed robustness metrics based on deviations from the deterministic schedule. Engell et al. [7] reported an application of a scenario-decomposition algorithm, along with some problem-dependent preprocessing procedures, and provided very good computational results for the solution of a two-stage stochastic scheduling model for a polymer plant under demand uncertainty. Although stochastic models optimize the total expected performance measure, they do not provide any control on their variability over different scenarios. Therefore, the application of fuzzy set theory to solve the multistage scheduling is proposed [8].

Scheduling of batch plants is one of the most difficult problems, as it is classified as an NP-complete one. In general, it is difficult to solve the model by some exact mathematical methods such as mixed integer linear programming (MILP) methods or branch and bound (B&B) methods when the size of the problem is too large [9]. So, a good way is to use heuristic optimization algorithm to get the near optimal solution. Genetic algorithm is one of the heuristic optimization algorithms.

It was developed by Holland [10] in 1975, inspired by simulation of natural choice process in biology evolution. Davis [11] was the first one who applied GA to the scheduling problem. Compared to many traditional local search heuristic methods, GA frequently has an advantage when search spaces are multi-dimensional, discontinuous, or highly constrained. A lot of reported work focused on GA has been applied widely in the function optimization, ANN training, pattern recognition, fuzzy control and some other fields [12–15].

The aim of this work is to set up a novel model for the short-term scheduling of multiproduct batch plants under demand uncertainty and solve this model by an improved genetic algorithm. The proposed formulation is based on a continuous-time representation and results in a MILP problem, which leads to a smaller number of binary and continuous variables, thus resulting in reduced computational times. The objective function includes a term for profit maximization and a penalty term that is used to minimize the deviations from the original schedule. According to the discrete characteristic of the scheduling of batch plants, an effective genetic algorithm is improved by a procedure-based coding method.

2. Scheduling model

The short-term scheduling of multiproduct batch plants under demand uncertainty involves the constraints of material balances and inventory constraints, as well as the penalty for production shortfalls. Taking the above constraints into consideration, a novel formulation is proposed. Scheduling model was formulated as a MILP problem using the continuous-time domain representation. Order cancellation can also be considered similar to rush order arrival.

Notations are listed below:

Indices:
i = tasks; j = units; s = states; n = event points
Sets:
I = tasks; J = units; S = states; N = event points within the horizon; I_j = tasks that can be processed in unit j; J_i = units capable of processing task i; I_s = tasks that produce or consume state s;
Parameters:
H = time horizon; H^{upper} = upper bound on time horizon; P_s = price of state s;
$V_j^{\min}, V_j^{\max}$ = minimal and maximal capacity of the specific unit j when processing task i;
SC_s = storage capacity for state s;
T_{ed} = time of arrival of an extra demand;
D_s = demand for state s at the end of the time horizon;
ΔD_s = an additional number units of product at time T_{ed};
ρ_{is}^c, ρ_{is}^p = proportion of state s consumed and produced by task i, respectively;
α_{ij} = constant term of processing time of task i in unit j;
β_{ij} = variable term of processing time of task i in unit j;
Variables:
$yv(j,n)$ = binary variables that assign the utilization of unit j at event point n;
$y(i,n)$ = binary variables that beginning of task i at event point n;
$T^s(i,j,n)$ = time that task i starts in unit j at event point n;
$T^f(i,j,n)$ = time that task i finishes in unit j while it starts at event point n;
$b(i,j,n)$ = amount of material undertaking task i in unit j at event point n;
$qt(s,n)$ = total quantity of state s produced at event point n;
$qs(s,n)$ = quantity of state s at event point n.

On the basis of this notation, the proposed mathematical model for the short-term scheduling of multiproduct batch plants under demand uncertainty involves the following constraints:

Allocation constraints:

$$\sum_{i\in I_j} y(i,n) = yv(j,n) \quad \forall j \in J, n \in N \tag{1}$$

Eq. (1) expresses that at each unit j and at an event point n only one of the tasks that can be performed in this unit should take place.

Capacity constraints:

$$V_j^{\min} \cdot y(i,n) \leqslant b(i,j,n) \leqslant V_j^{\max} \cdot y(i,n) \quad i \in I, j \in J_i, n \in N \tag{2}$$

Eq. (2) expresses the requirement for the minimum amount $V_j^{\min}$ of material in order for a unit j to start operating task i and the maximum capacity of a unit $V_j^{\max}$ when performing task i.

Storage constraints:

$$qs(s,n) \leqslant SC_s \quad \forall s \in S, n \in N \tag{3}$$

Eq. (3) represents the maximum available storage capacity for each state s at each event point n.

Material balances:

$$qs(s,n) = qs(s,n-1) - qt(s,n) + \sum_{i \in I_s} \rho_{is}^p \sum_{j \in J_i} b(i,j,n-1) + \sum_{i \in I_s} \rho_{is}^c \sum_{j \in J_i} b(i,j,n) \quad (4)$$

According to Eq. (4), the amount of material of state s at event point n is equal to that at event point $n-1$ adjusted by any amounts produced or consumed between the event points $n-1$ and n and the amount required by the market at event point n within the time horizon.

Demand constraints:

$$\sum_{n \in N} qt(s,n) \geqslant D_s + \Delta D_s \quad \forall s \in S \quad (5)$$

Eq. (5) represents the requirement to produce at least as much as required by the market.

Duration constraints:

$$T^f(i,j,n) = T^s(i,j,n) + \alpha_{ij} \cdot y(i,n) + \beta_{ij} \cdot b(i,j,n) \quad (6)$$

The duration constraints express the dependence of the time duration of task i at unit j at event point n on the amount of material being processed.

Sequence constraints:

Sequence constraints provide the connections between the starting and final times and the binary variables $y(i, n)$ and $yv(i, n)$. The sequence constraints are classified into,

Same task in same unit

$$T^s(i,j,n+1) \geqslant T^f(i,j,n) - H[2 - y(i,n) - yv(j,n)] \quad (7)$$

$$T^s(i,j,n+1) \geqslant T^s(i,j,n) \quad (8)$$

$$T^f(i,j,n+1) \geqslant T^f(i,j,n) \quad (9)$$

Eqs. (7)–(9) state that task i starting at event point $n+1$ should start after the end of the same task performed at the same unit j, which has already started at event point n.

Different tasks in the same unit

$$T^s(i,j,n+1) \geqslant T^f(i',j,n) - H[2 - y(i',n) - yv(j,n)] \quad (10)$$

Eq. (10) equation establishes the relationship between the starting time of task i at point $n+1$ and the end time of task i' at event point n when these tasks take place at the same unit.

Different tasks in different units

$$T^s(i,j,n+1) \geqslant T^f(i',j',n) - H[2 - y(i',n) - yv(j',n)] \quad (11)$$

If task i' takes place in unit j' at event point n, then we have $T^s(i, j, n+1) \geqslant T^f(i', j', n)$ and hence, task i in unit j has to start after the end of task i' in unit j', otherwise the right hand side becomes negative and the constraint is trivially satisfied.

Completion of previous tasks

$$T^s(i,j,n+1) \geqslant \sum_{n' \in N, n' \leqslant n} \sum_{i' \in I_j} T^f(i',j,n') - T^s(i',j,n') \quad (12)$$

In the above equation represents the requirement of task i to start after the completion of all the tasks performed in past event points at the same unit j.

Time horizon constraints:

$$T^f(i,j,n) \leqslant H \quad i \in I, j \in J_i, n \in N \quad (13)$$

$$T^s(i,j,n) \leqslant H \quad i \in I, j \in J_i, n \in N \quad (14)$$

Eqs. (13) and (14) represent the requirement of every task i to start and end within the time horizon H.

Objective:

In the case of rush order, the additional order may lead to an infeasible problem if the time horizon is kept fixed to the original time horizon because of plant capacity limitations. In this case, the following objectives are considered: (a) Maximize the total profit of the plant and/or maximize the production of rush order product within the original time horizon; (b) Minimize the makespan, i.e., find the shortest time within which the additional order can be delivered.

For case (a), an objective function of the following form is used:

$$\max \sum_s \sum_n p_s \cdot priority(s) \cdot qt(s,n) - penalty \cdot \sum_s priority'(s) \cdot slack(s) \quad (15)$$

The first term in the objective function is used to maximize the profit in the plant. A weighting parameter $priority(s)$ is used to effectively increase the price of a state s and thus maximize the profit preferentially toward the product with the rush order.

The second term is used to maximize the production of the various states s. The factor $priority'(s)$ is a weighting factor that is used to maximize preferentially the production of different states. The penalty term is used to scale the second term in the objective function to be of the same order of magnitude as the first term so that both terms drive the optimization equally. The variable $slack(s)$ is to maintain feasibility under rush order arrival by relaxing the demand constraint as follows:

$$\sum_{n \in N} qt(s,n) \geqslant D_s + \Delta D_s - slack(s) \quad \forall s \in S \quad (16)$$

In the above function, through changes in the values of $priority(s)$ and $priority'(s)$, different objective functions can be obtained that correspond to different weighted instances of product and total plant profit maximization. This provides the flexibility for the decision maker to analyze the effects of the balance between the two objectives before making a decision on which reschedule to implement.

For case (b), the objective function needs to be modified to minimize H. In this case, the total time horizon H is made a variable. Thus, it becomes necessary to incorporate an additional constraint that will maintain the monotonicity in the task starting times. This constraint takes the form

$$T^s(i,j,n) \geqslant T_{\mathrm{ed}}[1 - y(i,n)]y(i',n) \quad (17)$$

and enforces the requirement that, for all tasks that have not yet started at the time of rush order arrival, their starting time T^s will be greater than T_{ed}. Also, in order to avoid the introduction of nonlinear terms, an upper bound on H is used in all sequence constraints, and an additional constraint is added for variable H of the form $H \leqslant H^{upper}$.

3. Scheduling algorithm

GA is adopted for the scheduling of batch plants, and is used as follows.

3.1. Coding method

Coding is an improved procedure-based method. Here, scheduling results are coded in accordance with the sequence of procedures, where the same symbol is assigned to a particular procedure. Results are described by the sequence in a special chromosome. For example, one chromosome for 3 tasks, 3 units the scheduling problem is [1 3 1 2 2 1 3 3 2], where "1" denotes part J_1, "2" denotes part J_2, and "3" denotes part J_3. Three "2"s in the chromosome sequence denote three procedures of part J_2; the first "2" refers to the first procedure of part J_2, and others can be inferred in the same way. Each individual part's procedure sequence constraint has been considered for this coding method, so there is no dead loop for decoding [16].

3.2. Fit function

When the objective function is to maximize the total profit, a scheduler decoding the information carried in the composite chromosome is used to evaluate the fitness of each chromosome. The fitness value is set to the inverse of the maximum profit from the generated schedule. The actual schedule is deduced through a simulation, which analyzes the state of the waiting queues in front of the machine and the preference lists, gene 1 to gene n from each chromosome segment [17].

When the scheduling target is to minimize the makespan for the work process, the fit function is designed as follows:

$$F(X) = \begin{cases} C_{\max} - f(X), & \text{if } (C_{\max} \geqslant f(X)) \\ 0, & \text{if } (C_{\max} < f(X)) \end{cases}$$
$$F' = aF + b \quad a = 1/(F_{\text{avg}} - F_{\min}) b = -F_{\min}/(F_{\text{avg}} - F_{\min}) \tag{18}$$

In early stages, fitness value differences among individuals are large, and few individuals possess a high proportion during the choosing process. This decreases the diversity of a group. In the final stage, fitness values among individuals may be very close, so it is easy to enact a random choosing process. Here, "a" and "b" are used to adjust fitness values to avoid premature selection in early stages, and to ensure a random selection process in the final stage.

3.3. Choosing algorithm

A proportional model-based strategy is used,

$$P_{is} = F_i \Bigg/ \sum_{i=1}^{M} F_i \tag{19}$$

where P_{is} is the probability that an individual will be chosen for the proportional model. The choosing error for the proportional method is large because sometimes chromosomes with a high fitness value will not be selected. Thus, the following steps should be followed to ensure that the best chromosomes are selected [18]: (i) Determine the individuals with the highest and lowest fitness values; (ii) If the fitness of the best individual in the last group is much higher than that of the best individual, then this individual will be considered the best individual; (iii) If the fitness of the best individual in the last group is much lower than that of the best individual, then the best individual will be used to replace the worst individual in the last group.

3.4. Crossover operator

A linear order crossover procedure is used [19]. First, one couple of parent chromosomes is selected randomly; then, one part is selected randomly, which has m operations corresponding to each parent chromosome, and "H" is used to replace operations for unselected parts in the first parent chromosome, so a new chromosome is generated. Now, we turn k positions to the left or right, where k is random, but the distance among the operations of the selected parts must be kept stable. The operations of the unselected parts in the second parent chromosome are used to replace "H" in the first parent as the original relative sequence. Thus, a new chromosome is generated. Similarly, m operations corresponding to the selected parts in the second parent chromosome and the operations corresponding to the unselected parts in the first parent chromosome can be used to generate another new chromosome.

3.5. Mutation algorithm

Mutation algorithm is for selecting one part randomly, and a corresponding set of m operations are conducted randomly, at the same time, the distance among them and the relative sequence of other operations is kept stable.

4. Experimental results and analyses

In this section, we present some examples to illustrate the performance of the proposed model and the improved GA. All of the following scheduling problems are solved on a 1.8 GHz, 512 MB AMD PC.

Example 1. Two different products are produced through five processing stages: heating; reactions 1, 2, and 3; and separation of product 2. The data for this example are presented in Ref. [20].

Example 2. Four products are produced through eight tasks from three feeds. There are nine intermediates in the system. In all, six different units are required for the whole process. The STN representation and data are shown in Ref. [21].

For Example 1, the deterministic schedule was solved to maximize the total profit within a fixed time horizon of 8 h. The schedule is obtained for an order of 55 units of P1 and 70 units of P2. Similarly, for Example 2, the deterministic schedule was determined to give the minimum demand of 520 units of P1, 1215 units of P2, 290 units of P3, and 1350 units of P4. The corresponding schedule is shown in Fig. 1.

A rush order of 60 additional units of P1 that arrives at time $T_{\text{rush}} = 2$ h is considered for Example 1. An objective function of maximizing the total profit is considered. Values of $priority(s) = 0$ for all states, $priority'(s) = 1$ for P1 and 0 for all other states, and $penalty = 1$ are considered. In this case, the objective function corresponds to the maximization of the production of P1 only. The results for optimal scheduling are reported in Fig. 2.

It is found that, within the fixed time horizon of 8 h, the maximum production of P1 is only 84 units, which corresponds to only part of the total order of 115 units,

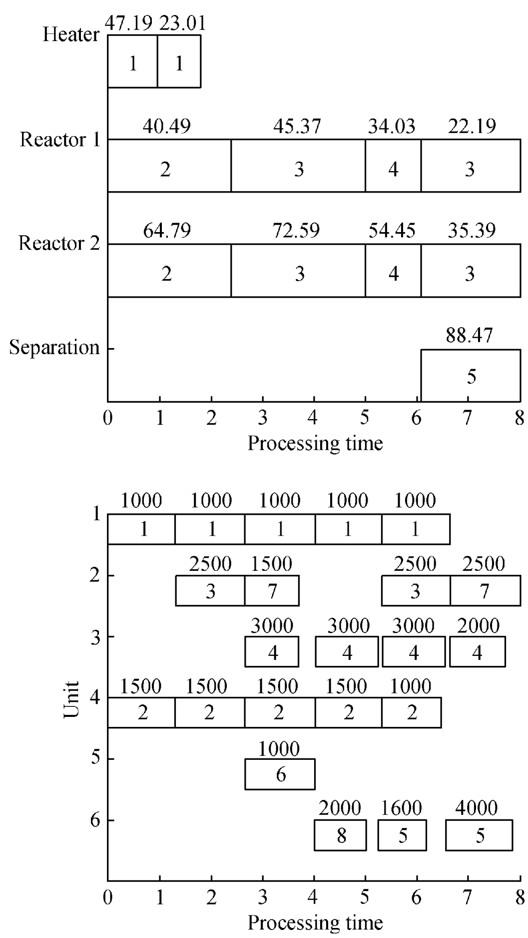

Fig. 1. Gantt charts of deterministic scheduling for Examples 1 and 2.

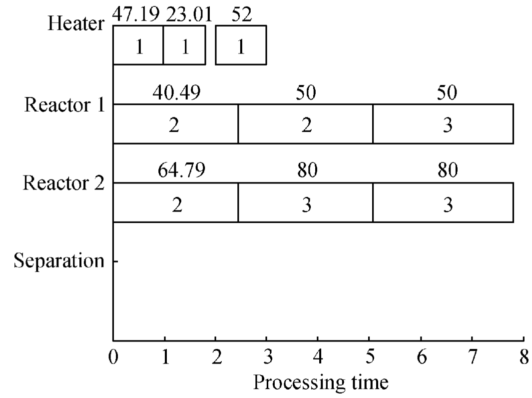

Fig. 2. Gantt chart of scheduling under demand uncertainty for Example 1.

$slack(\text{P1}) = 31$ units. It means that the problem is infeasible. It means it is impossible to fulfill the rush order on P1 within the fixed time horizon of 8 h. Thus, consider an alternative objective function to determine the minimum makespan by which the additional order can be delivered.

Investigation on the effects of rush order arrival time on the makespan was carried out. For Example 1, the makespan for the deterministic schedule is 7.53 h. Scheduling under demand uncertainty is carried out to support an additional order of 30 units of P1 and an additional order of 30 units of P2. A similar study is performed for Example 2. The makespan for the deterministic schedule is 4.48 h. Uncertain scheduling is performed to handle a rush order of 200 additional units of P3. Results in Fig. 3 shows that the proposed model predicts a nonuniform increase in the minimum makespan depending on the time the rush order arrives. When the rush order arrives before 4 h, the makespan does not change significantly. This is a result of the inherent flexibility of the plant in terms of the capability of a task to be performed on more than one unit. When the rush order arrives between 4 and 6 h, then it is possible that one additional batch of some intermediate would add constant processing time or set up time to the makespan, thus causing the jump. P1, P2, and P3 are produced in the same campaign, which is the reason that similar trends

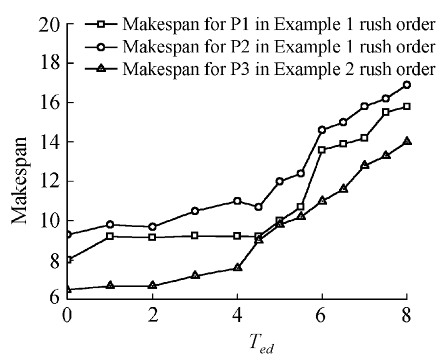

Fig. 3. Effect of rush order arrival time on makespan for Examples 1 and 2.

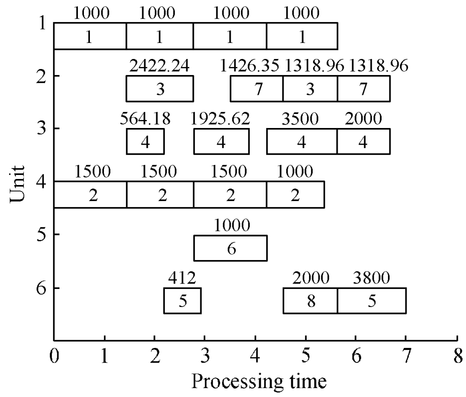

Fig. 4. Gantt chart of scheduling under demand uncertainty for Example 2.

are observed. Fig. 4 shows the Gantt chart of optimizing the makespan for Example 2.

An important issue in the determination of the optimal schedule after a rush order arrives is the balancing of the objectives of delivering the rush order and maximizing overall plant profit. The proposed objective function allows the flexibility for modeling different weighted instances of the two targets so that a best-possible decision can be determined. Take an additional order of 60 units of P1 in Example 1 as an example. Two cases that correspond to different forms of the objective function of maximizing the total profit are then considered. (1) Case 1: Values of $priority(s) = 1$ for all states, $priority'(s) = 0$ for P1 and 1 for all other states, and $penalty = 100$. For this case, this objective function balances both the maximization of the plant profit and the attempted maximization of production of P1. (2) Case 2: Values of $priority(s) = 0$ for all states, $priority'(s) = 1$ for P1 and 0 for all other states, and $penalty = 100$. This means that the first term in the objective function is zero and the objective reduces to maximizing the production of P1 only.

The total P1 production and scaled values of the total plant profit versus time of rush order arrival for both case 1 and case 2 are shown in Fig. 5. The Results show that (1) both cases result in almost the same production for P1. (2) The plant profit determined in case 1 is significantly higher than that obtained by case 2. This is due to the fact that case 2 maximizes only the production of P1 without considering any possible profit increase due to the production of P2. Thus, we can conclude that the use of the objective function as in case 1 that has priorities balanced equally toward profit maximization and maximum P1 production results in better total profit and can be directed toward complete satisfaction of the rush order by increasing the priority of the corresponding product.

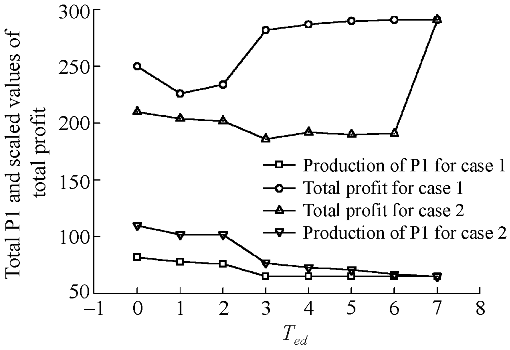

Fig. 5. Effect of objective function on production versus total plant profit.

5. Conclusions

In this paper, a novel formulation for the short-term scheduling of multiproduct batch plants under demand uncertainty is presented, then is solved by an improved genetic algorithm. The proposed approach results in an efficient utilization of the plant capability as it allows the optimal selection among all rescheduling alternatives in a systematic way without the use of any heuristics. Moreover, the objective function can not only maximize the total profit of the plant and minimize the makespan but also allow the flexibility for modeling different weighted instances of the two targets so that the best-possible decision can be determined.

The Results obtained from the two examples have been used to illustrate the applicability and efficiency of the proposed formulation and algorithm. The Results show that (1) The proposed model predicts a nonuniform increase in the minimum makespan depending on the time the rush order arrives. When the rush order arrives before 4 h, the makespan does not change significantly. When the rush order arrives between 4 and 6 h, the makespan increases sharply. (2) The objective function as in case 1 results in better total profit and can be directed toward complete satisfaction of the rush order by increasing the priority of the corresponding product.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 60674075 and No. 60774078) and the Key Technologies Program of Shanghai Educational Committee (Grant No. 05ZZ73).

References

[1] Rippin DWT. Batch process systems engineering: a retrospective and prospective review. Comput Chem Eng 1993;17(Suppl):1–13.
[2] Pinto JM, Grossmann IE. Assignment and sequencing model for the scheduling of process systems. Ann Oper Res 1998;81:433–41.
[3] Ierapetritou MG, Pistikopoulos EN. Batch plant design and operations under uncertainty. Ind Eng Chem Res 1996;35:772–87.
[4] Sand G, Engell A, Markert A, et al. Approximation of an ideal online scheduler for a multiproduct batch plant. Ind Eng Chem Res 2000;24:361–74.
[5] Gupta A, Maranas CD, McDonald CM. Midterm supply chain planning under demand uncertainty: customer demand satisfaction and inventory management. Comput Chem Eng 2000;24:2613–25.
[6] Vin JP, Ierapetritou MG. Robust short-term scheduling of multiproduct batch plants under demand uncertainty. Ind Eng Chem Res 2001;40:4543–52.

[7] Engell S, Markert A, Sand G, et al. Scheduling of a multiproduct batch plant by two-stage stochastic integer programming. Technical Report 59, Preprint 520. Institute fur Mathematik, Gerhard Mercator Universitat: Duisburg, Germany; 2002. p. 520–34.
[8] Balasubramanian J, Grossmann IE. Scheduling optimization under uncertainty–an alternative approach. Comput Chem Eng 2003;5:469–90.
[9] Zhang F, Chen F, Tang GC. Convex quadratic programming relaxations for parallel machine scheduling with controllable processing times subject to release times. Prog Nat Sci 2004;14(9):758–64.
[10] Holland JH. Adaptation in natural and artificial system. Michigan: Michigan University Press; 1975, p. 253–66.
[11] Davis L. Job shop scheduling with genetic algorithms. In: Proceedings of the first international conference on genetics and their applications, July 1–3, 1985. p. 136–40.
[12] Du HF, Gong MG, Jiao LC. A novel algorithm of artificial immune system for high-dimensional function numerical optimization. Prog Nat Sci 2005;15(5):463–71.
[13] Al-Rawi M, Karajeh H. Genetic algorithm matched filter optimization for automated detection of blood vessels from digital retinal images. Comput Methods Programs Biomed 2007;87(3):248–53.
[14] Yang XH, Yang ZF, Shen ZY, et al. Node ordinal encoded genetic algorithm for the optimal allocation of water resources. Prog Nat Sci 2005;15(5):448–52.
[15] Lee F, Choon RG, Pandu R, et al. Multi-objective optimization of an industrial penicillin V bioreactor train using non-dominated sorting genetic algorithm. Biotechnol Bioeng 2007;98(3):586–98.
[16] Hemant KNS, Srinivasan G. A genetic algorithm for job shop scheduling–a case study. Comput Chem Eng 1996;31:155–60.
[17] Kirley MLX, Green DG. Investigation of a cellular genetic algorithm that mimics landscape ecology. In: Proceedings of the second asian pacific conference on simulated evolution and learning, Canberra, Australia, November 24–27, 1998. p. 218–27.
[18] Chen G, Wang X. Genetic algorithm and its application. Beijing: People's Post Telecom Publishing House; 1996, p. 268–96.
[19] Zou ZM, Li CX. Integrated and events-oriented job shop scheduling. Int J Adv Manuf Technol 2006;29(5):551–6.
[20] Ierapetritou MG, Floudas CA. Effective continuous-time formulation for short-term scheduling: I. Multipurpose batch processes. Ind Eng Chem Res 1998;37:4341–9.
[21] Kondili E. Optimal scheduling of batch processes. Ph.D. Thesis, Imperial College London, London, UK, 1987.