Chinese Journal of Chemical Engineering xxx (2017) xxx–xxx

Contents lists available at ScienceDirect

Chinese Journal of Chemical Engineering

journal homepage: www.elsevier.com/locate/CJChE

Rule-based scheduling of multi-stage multi-product batch plants with parallel units☆

Bin Shi *, Xinrui Qian, Shanshan Sun, Liexiang Yan

Department of Chemical Engineering, Wuhan University of Technology, Wuhan 430070, China

ARTICLE INFO

Article history:
Received 28 September 2016
Received in revised form 25 December 2016
Accepted 21 March 2017
Available online xxxx

Keywords:
Line-up competition algorithm
Order assignment rule
Multi-stage multi-product
Parallel unit
Scheduling optimization

ABSTRACT

A novel rule-based model for multi-stage multi-product scheduling problem (MMSP) in batch plants with parallel units is proposed. The scheduling problem is decomposed into two sub-problems of order assignment and order sequencing. Firstly, hierarchical scheduling strategy is presented for solving the former sub-problem, where the multi-stage multi-product batch process is divided into multiple sequentially connected single process stages, and then the production of orders are arranged in each single stage by using forward order assignment strategy and backward order assignment strategy respectively according to the feature of scheduling objective. Line-up competition algorithm (LCA) is presented to find out optimal order sequence and order assignment rule, which can minimize total flow time or maximize total weighted process time. Computational results show that the proposed approach can obtain better solutions than those of the literature for all scheduling problems with more than 10 orders. Moreover, with the problem size increasing, the solutions obtained by the proposed approach are improved remarkably. The proposed approach has the potential to solve large size MMSP.

1. Introduction

In batch chemical industry many products are processed through several sequentially connected stages. In order to improve the efficiency of production, non-identical parallel processing units are used in each stage. In such plants, the scheduling of plant operations is a routine activity. However, because of the many alternate ways in which orders can be assigned to various units and produced in different sequences, the task of optimal scheduling is formidable. Hence, the multi-stage multi-product scheduling problem (MMSP) in batch plant has received great attention from both academia and industry. In the past decades, many studies on MMSP have been reported in the literature [1–7]. The main solution methods is to establish mixed-integer linear programming (MILP) model first, including objective function and a set of equality and inequality constraints, then solve the model using a modeling software.

Pinto and Grossmann [8] presented a continuous-time mixed-integer linear programming (MILP) model with continuous time representation for MMSP, which relied on the use of parallel time axes for units and tasks. Later, they proposed an alternative model in which the pre-ordering of orders was imposed explicitly by applying a representation of the time slots for the units. This resulted in a significant reduction in the computational time [9].

Hui and Gupta [10,11] established a MILP model for MMSP using three-dimensional 0–1 variables instead of tetra-index 0–1 variables to represent the sequence-dependent relationship between order and equipment. The main advantage of this model compared to the other previously proposed formulations was the significant reduction in model size and consequently shortening the solution time.

Gupta and Karimi [12] presented a continuous-time MILP formulation without using time slots for short-term scheduling of such plants. The formulation allowed both sequence-dependent and unit-dependent setup times and common operational considerations such as order/unit release times. They developed novel constraints for assignment of consecutive orders on a single unit and evaluated them thoroughly to identify the best constraints. Case study results showed the superiority of the proposed formulations.

Castro and Grossmann [13] presented a multiple-time-grid, continuous-time MILP model for MMSP. The formulation could handle both release time and due dates with different time-based objective functions efficiently, such as the minimization of total cost, total earliness, or makespan. Afterwards, Castro, Grossmann and Novais [14] presented another two kinds of continuous time MILP model based on multiple-time-grid to solve MMSP with sequence-dependent changeover time constraints.

Xue and Yuan [15] put forward MILP model with pre-ordering approach to solve MMSP. The pre-ordering approach could identify the infeasible assignments through which the number of binary variables was significantly reduced. Illustrative examples showed that the size of the proposed model was small, thus significantly shortening the solution time in comparison with the existing models in literature.

The above mentioned methods took the same way for solving MMSP. First, a group of binary variables were introduced to indicate the order allocation and the unit selection. Then, the MILP models for MMSP were built using discrete or continuous time representation

☆ Supported by the National Natural Science Foundation of China (No. 21376185)
* Corresponding author.
E-mail address: sbin125@163.com (B. Shi).

http://dx.doi.org/10.1016/j.cjche.2017.03.014
1004-9541/

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

approach. Finally, some traditional deterministic optimization algorithms or commercial computing software were employed to solve to the models [16]. Because of the complexity of MMSP, MILP models established by traditional methods were large-scale and their generality were poor, which made it difficult to deal with complex constraints and only feasible for small size problem. When the problem size increases linearly, the computational time of MILP will increase exponentially. It is difficult for MILP to solve large-size MMSPs. Although some researchers claimed that their MILP models were suitable for large-size problems, the solutions to large-size problems were far from the optimum, even if the algorithm ran for very long time [16]. Therefore, it is urgent to explore reasonable scheduling model and seek more effective approach for solving large-size MMSP.

Different from the traditional MILP models of MMSP, He and Hui [16] presented a heuristic approach based genetic algorithm (GA) for MMSP. In their method, an order sequence was encoded into the chromosome of the GA. Through the evolutionary mechanism of GA, a group of random chromosomes was evolved to achieve the optimal or near-optimal solution.

In this paper, a novel heuristic rule based model for MMSP with parallel unit is presented. The multi-stage multi-product batch production process is then divided into multiple sequentially connected single process stages by using hierarchical scheduling strategies combined with the concepts of “virtual release time” and “virtual due date”. According to the features of the proposed model, line-up competition algorithm (LCA) is introduced to solve the MMSP model. The performance of the proposed method is illustrated by solving two typical complex illustrative MMSP examples.

The rest of this paper is organized as follows. Section 2 details mathematical model of MMSP with parallel units, and presents hierarchical scheduling method; Section 3 introduces computational procedures of LCA for MMSP. In Section 4, problems from the literature are solved to illustrate the performance of the proposed approach. Finally, the conclusions are given in Section 5.

2. Model Development

2.1. Problem definition

In this paper, the scheduling problem of multi-stage multi-product batch plants with parallel units is considered. In MMSP, a set of production orders is to be processed. Each order, involving a single product, requires a certain number of processing stages (as shown in Fig. 1). The features of this process are as follows:

(1) Each order has a predetermined release time and due date;
(2) Each order involves a single product requiring a single processing batch;
(3) Each order requires multiple successive processing stages, and can only be processed once in each stage.
(4) Each order can only be processed in a subset of the given units in each stage. It means that there may be forbidden units for each order. The process time of order is unit-dependent.
(5) Some units have finite unit release times. Hence, they are not available from the beginning of the time horizon of interest.
(6) A unit processes only one order at a time. When one order changes over to another order, time for unit cleaning and preparing is required. The changeover times are sequence-dependent or both sequence- and unit-dependent;
(7) Because of flavor and/or color incompatibilities, some order sequences are forbidden at any equipment item.
(8) Each unit belongs to one production stage, at the same time it can only handle one order.

Given all the above features, the scheduling problem to be tackled involves determining (1) the allocation of orders to units in each stage, (2) sequencing orders in each unit, and (3) timing each order so that all scheduling constraints are taken into account and, at the same time, some measures of the customer satisfaction and/or the plant performance are optimized.

The following assumptions are used to derive the scheduling formulation for MMSP:

(1) All the model parameters are deterministic.
(2) Materials of each order are constant throughout the process, no material loss.
(3) Once a batch of materials was produced, it cannot be interrupted during the process.
(4) No resource constraints except unit are considered.
(5) There is unlimited intermediate storage between two stages.

2.2. Scheduling objectives

Two typical time-based scheduling objectives, total flow time and total weighted process time, are chosen to be optimized [8,16].

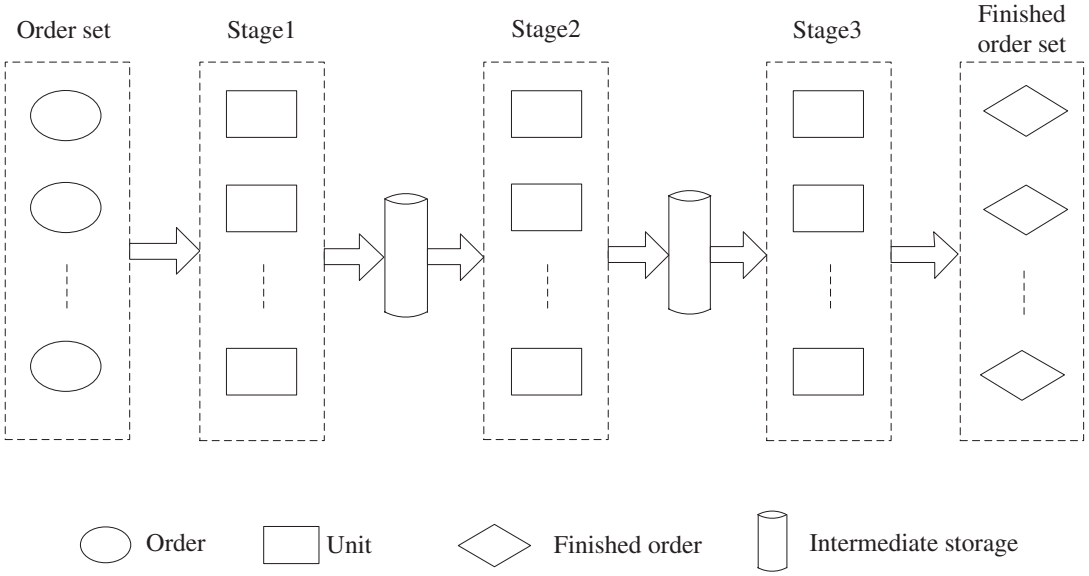


Fig. 1. Schematic of a multistage batch plant with parallel units.

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

Total flow time (F):

$$\text{Minimize } F \tag{1}$$

$$F = \sum_1^{NI} C_i i = 1, 2....NI \tag{2}$$

where, where $C_i$is the completion time of order i; F is the sum of completion time of all orders referred as total flow time.

Total weighted process time (PT):

$$\text{Maximize } PT \tag{3}$$

$$PT = \sum_{i=1}^{NI} \left(\sum_{s=1}^{S} a_{s,i} C_{s,i} - NC \times \max\{C_{s,i} - d_i, 0\} \right) i = 1, 2....NIs = 1, 2....S \tag{4}$$

where,$C_{s,i}$ is the completion time of order i in stage s; $a_{s,i}$ is the weighted factor of order i in stage s; and PT is the total weighted process time. The advantage of maximizing this objective is to keeping the completion time of each order approaching its due date as much as possible. This would reflect in minimum intermediate storage requirements. On the other hand, the weighted factor of each order represents the importance of each product, by adjusting the weighting factors, particular scheduling scheme can be obtained.

2.3. Heuristic order assignment rules

For solving the above scheduling objectives, two decisions must be made successively in each stage: (1) assigning unit for each order, and (2) arranging the production time for each order at the selected unit. The heuristic order assignment rules that proposed in our previous work [17] can be employed here to help achieve the scheduling objectives. However, it is noted that the scheduling objectives, total flow time and total weighted process time, considered in this work are two types of conflict objectives. The former pursues the entire production process that can be finished as soon as possible, while the latter emphasizes that the completion time of each order can approach its due date as closely as possible. Obviously, the heuristic order assignment rules described in Ref. [17] can still apply to the former scheduling objective, however, part of them need to be adjusted for the latter. The revised heuristic order assignment rules are describes as follows:

(1) Assign the order to the unit that makes the start time of order on the unit to be as late as possible, namely, last start time (last start time, LST);
(2) Assign the order to the unit that makes the completion time of order on the unit to be as late as possible (last completion time, LCT);

It is noted that the above two rules along with rule of shortest changeover time (SCT), rule of shortest process time (SPT), and rule of shortest total changeover and process time (SCTP) proposed in Ref. [17] can be used for order assignment under the scheduling objective of maximizing total weighted process time.

2.4. Hierarchical scheduling strategy

According to the heuristic rules under different scheduling objectives, the unit used for processing given order in each stage can be determined. Next, the start time and completion time of the order in the selected unit need to be arranged. Since the multi-stage multi-product batch process is mainly composed of a plurality of sequential and independent production stages, hence, this process can be divided into multiple sequentially connected single process stages, and then arranging the order production in each single stage. In this work, according to the different characteristics of scheduling objectives, the corresponding two different order assignment strategies: forward order assignment strategy and backward order assignment strategy, are presented as follows.

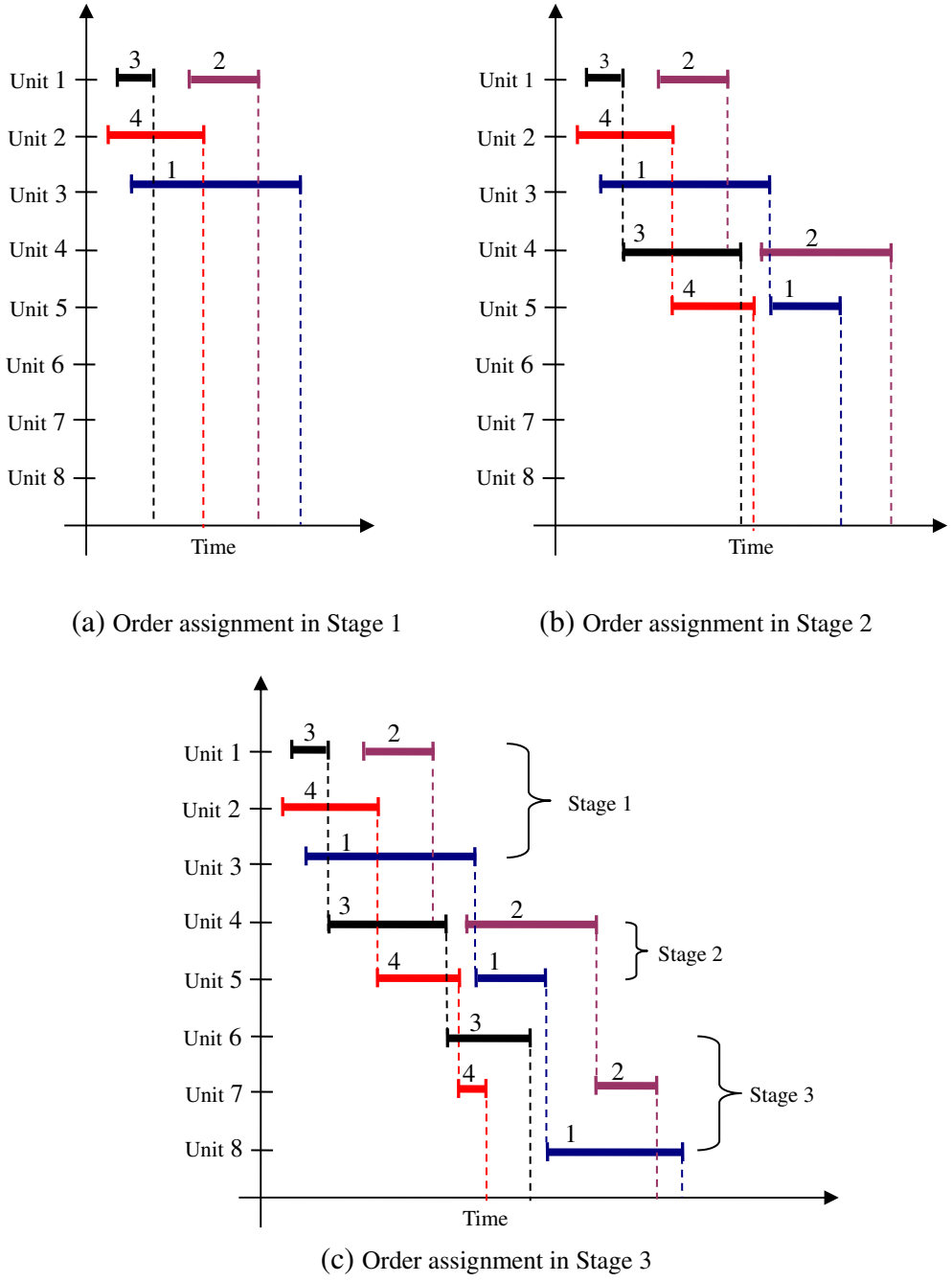
(1) Forward order assignment strategy: for solving the total flow time minimization problem, the production time of each order is sequentially arranged from the first stage to the final stage. At beginning, for a given order sequence, the start time and completion time of each order is determined sequentially in the first stage according to selected heuristic rule. Then, the completion of each order in first stage is treated as its virtual order release time, and the order completion sequence is used as the order processing sequence in second stage. And so on, arranging the order production in the subsequent stages.
(2) Backward order assignment strategy: For solving the total weighted process time maximization problem, the production time of each order is sequentially arranged from the final stage to the first stage. At the beginning, for the first order in the given order sequence, the production unit of this order in final stage is arranged according to the selected heuristic rule. And then, arranging the order production to make its completion time as close as possible to its due date. Using the same method above, the production of rest order in final stage is determined. Afterwards, taking the start time of each order in final stage as its virtual due date in the penultimate stage, and then, ordering the start time of all orders in descending sequence, this sequence is employed as production sequence of orders in the penultimate stage. And so on, arranging the order production in the previous stages one by one until the production of all orders in first stage is arranged.

Figs. 2–3 are the illustration of how FAU rule and LCT rule work according to above two strategies respectively. Other rules work similarly, taking a 3-stage/4-order production process for example. There are 3 parallel units in the first stage, 2 units in the second stage and 3 units in the third stage, respectively. Given a random order sequence $P =$ {4, 3, 1, 2}, and assuming the rule FAU is adopted to arrange order production according forward order assignment strategy. Fig. 2(a) shows the assignment result of these orders, it is clear that the completion sequence of all orders in the first stage is {3, 4, 2, 1}, this sequence is treated as the order processing sequence in the second stage. And the completion time of each order in the first stage is treated as its virtual order release time in the second stage. On this basis the order production in the second stage can be determined as shown in Fig. 2(b). The completion sequence of orders is {3, 4, 1, 2} which can be seen from this figure; this sequence is used as the order processing sequence in the third stage. Correspondingly, the completion time of each order in the second stage is taken as its virtual order release time in stage 3. Therefore the order production in stage 3 can also be determined as shown in Fig. 2(c).

In this section, it is assumed that the rule LCT is employed to arrange order production according to backward order assignment strategy. Fig. 3(a) shows the order assignment in the final stage and d_1, d_2, d_3, and d_4 denote the actual due date of Orders 1–4 respectively. It is clear from Fig. 3(a) that the sequence {4, 3, 2, 1} can be obtained by sorting the starting time of orders in descending order, this sequence will be treated as the order processing sequence in Stage 2. And the starting time of each order in Stage 3 can be taken as its virtual due date in Stage 2. On this basis, the order production in Stage 2 is determined as shown in Fig. 3(b). The similarity steps are adopted to arrange order production in the first stage, and Fig. 3(c) gives the final assignment.

In MMSP, besides the basic process constraints, there are also some complex constraints, such as order and unit release time, forbidden sequences and forbidden units, and sequence- and unit-dependent changeover time. These constraints can be transformed to non-constraints by using the complex constraints handling strategies proposed in our previous work [17].

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014



(a) Order assignment in Stage 1

(b) Order assignment in Stage 2

(c) Order assignment in Stage 3

Fig. 2. Schematic diagram of forward order assignment based FAU rule.

2.5. Order assignment procedure

Based on the above order assignment rules and hierarchical scheduling strategies, given arbitrary order sequence $P=\{p_1,p_2,\ldots,p_N\}$ and scheduling objective, a detailed schedule scheme can be obtained.

Table 1 gives the order assignment procedure for solving the total flow time minimization problem by using forward assignment strategy. The procedure can be described as follows:

(1) At the beginning of the scheduling duration, setting release time of each unit as its first available time. Setting the latest order finished in each unit is a virtual Order "0". For any giving Order i, the changeover time between Order i and Order 0 is 0, namely $CT_{0,i}=0$. Setting the first stage as current stage.

(2) Getting the current Order p_i from the given order sequence P, selecting a Unit $\tilde{j}$ from Order p_i's available Units set J_{s,p_i} according to given order assignment rule.

(3) Calculating the start time $TS_{p_i,\tilde{j}}$ and the completion time $TE_{p_i,\tilde{j}}$ of order p_i in Unit $\tilde{J}$.

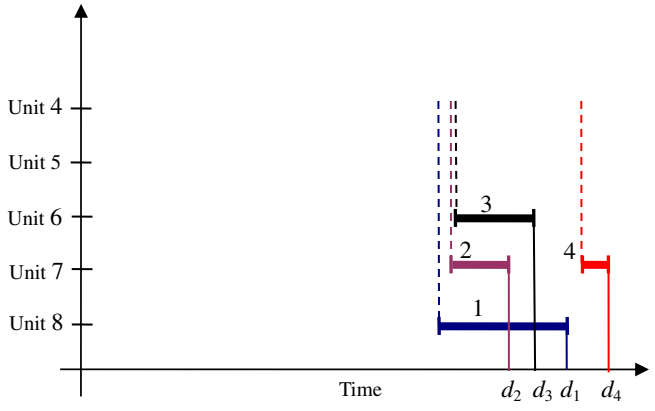
(4) Taking the completion time of order p_i in Unit $\tilde{j}$ as Unit $\tilde{j}$'s current available time, namely. $MCT_{\tilde{j}}=TE_{p_i,\tilde{j}}$

(5) Recording order p_i as the latest finished order Unit $\tilde{j}$, $MCP_{\tilde{j}}=p_i$.

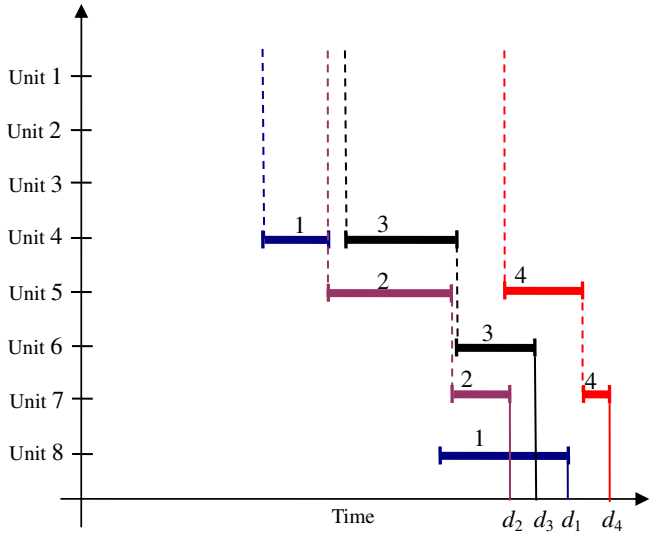
(6) Recording $TE_{P_{i,\tilde{j}}}$ as the completion time of Order p_i in current Stage s, $c_{s,p_i}=TE_{p_i,\tilde{j}}$

(7) If p_i is the final order in order sequence P, go to Step (8), if not, go to Step (2)

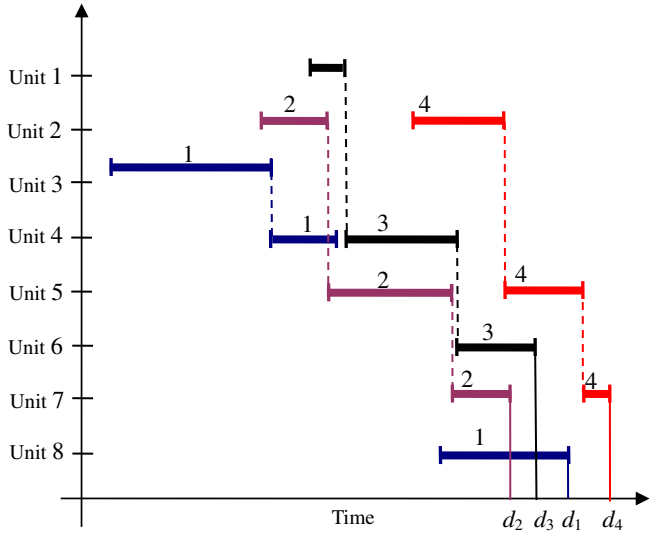
Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014



(a) Order assignment in Stage 3



(b) Order assignment in Stage 2



(c) Order assignment in Stage 1

Fig. 3. Schematic diagram of backward order assignment based LCT rule.

(8) Sorting the order completion time in current stage in ascending order $\tilde{P}$, then using this sequence to update the current order production sequence, $P = \tilde{P}$

(9) Setting the order completion time in current stage as its virtual release time in the next followed stage, $VOR_{s+1,i} = c_{s,i}, i \in I$

Table 1
Pseudo code of forward order assignment procedure

$MCT_j = UR_j, \forall j \in J$;
$MCP_j = 0, \forall j \in J$;
$VOR_{1,i} = OR_i, \forall i \in I$;
for $s = 1:S$
for $i = 1:NI$
switch (r_s)
case 1: $f_{p_{i,\tilde{J}}} = \min\{MCT_j\}, \forall j \in J_{s,pi}$://FAU
case 2: $f_{p_{i,\tilde{J}}} = \min\{CT_{MCP_j,p_i}\}, \forall j \in J_{s,p_i}$://SCT
case 3: $f_{p_{i,\tilde{J}}} = \min\{MCT_j + CT_{MCP_j,p_i}\}, \forall j \in J_{s,p_i}$://EST
case 4: $f_{p_{i,\tilde{J}}} = \min\{PT_{p_i,j}\}, \forall j \in J_{s,p_i}$://SPT
case 5: $f_{p,j} = \min\{\max(MCT_j + CT_{MCP_j,p_i}, VOR_{s,p_i}) + PT_{p_i,j}\}, \forall j \in J_{s,p_i}$;//ECT
case 6: $f_{p_{i,\tilde{J}}} = \min\{CT_{MCP_j,p_i} + PT_{p_i,j}\}, \forall j \in J_{s,p_i}$://SCPT
case 7: $f_{p_{i,\tilde{J}}} = \min\{MCT_j + PT_{p_i,j}\}, \forall j \in J_{s,p_i}$://SATP
end
$TE_{p_{i,\tilde{J}}} = \max\left(MCT_{\tilde{J}} + CT_{MCP_j,p_i}, VOR_{s,p_i}\right) + PT_{p_{i,\tilde{J}}}$;
$TS_{p_{i,\tilde{J}}} = TE_{p_{i,\tilde{J}}} - PT_{p_{i,\tilde{J}}}$;
$MCT_{\tilde{J}} = TE_{p_{i,\tilde{J}}}$;
$MCP_{\tilde{J}} = p_i$;
$c_{s,p_i} = TE_{p_{i,\tilde{J}}}$;
end
$\tilde{P} = sort(c_{s,1}, c_{s,2} \cdots c_{s,NI})$;
$P = \tilde{P}$
$VOR_{s+1,i} = c_{s,i}, \forall i \in I$
end

(10) If current stage is the final stage, finish the order assignment procedure, if not, update current stage, $s = s + 1$, then go to Step (2) for the next turn of order assignment.

Table 2 gives the order assignment procedure for solving the total weighted process time maximization problem by using backward assignment strategy. The procedure can be described as follows:

(1) At the beginning of backward order assignment procedure, setting the final stage as current stage. Setting the latest order finished in each unit is a virtual order "0", $RMCP_j = 0$. For any

Table 2
Pseudo code of backward order assignment procedure

$MST_j = NC, \forall j \in J$;
$RMCP_j = 0, \forall j \in J$;
$VD_{S,i} = d_i, \forall i \in I$;
for $s = S:-1:1$
for $i = 1:NI$
switch (r_s)
case 1: $f_{p_{i,\tilde{J}}} = \max\left\{\min\left(MST_j - CT_{p_i,RMCP_j}, VD_{s,p_i}\right) - PT_{p_i,j}\right\}, \forall j \in J_{s,p_i}$://LST
case 2: $f_{p_{i,\tilde{J}}} = \max\left\{\min\left(MST_j - CT_{p_i,RMCP_j}, VD_{s,p_i}\right)\right\}, \forall j \in J_{s,p_i}$://LCT
case 3: $f_{p_{i,\tilde{J}}} = \min\left\{CT_{p_i,RMCP_j}\right\}, \forall j \in J_{s,p_i}$://SCT
case 4: $f_{p_{i,\tilde{J}}} = \min\left\{PT_{p_i,j}\right\}, \forall j \in J_{s,p_i}$://SPT
case 5: $f_{p_{i,\tilde{J}}} = \min\left\{CT_{p_i,RMCP_j} + PT_{p_i,j}\right\}, \forall j \in J_{s,p_i}$://SCPT
end
$TE_{p_{i,\tilde{J}}} = \min\left(MST_{\tilde{J}} + CT_{P_i,RMCP_{\tilde{J}}}, VD_{s,p_i}\right)$;
$TS_{p_{i,\tilde{J}}} = TE_{p_{i,\tilde{J}}} - PT_{p_{i,\tilde{J}}}$;
$MST_{\tilde{J}} = TS_{p_{i,\tilde{J}}}$;
$RMCP_{\tilde{J}} = p_i$;
$c_{s,p_i} = TE_{p_{i,\tilde{J}}}$;
$sta_{s,p_i} = TS_{p_{i,\tilde{J}}}$;
end
$\tilde{P} = dsort(sta_{s,1}, sta_{s,2} \cdots sta_{s,NI})$;
$P = \tilde{P}$
$VD_{s-1,i} = sta_{s,i}, \forall i \in I$
end

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

Table 3
Results of solving different cases in Example 1 by LCA

Size	Parameter of LCA		Result					
	Family number	Terminate criterion	Best	Mean best	Standard deviation	Success number	AEG[①]	Function evaluations
5	5	100	6852.76	6852.76	0	50	1.3	6.5
10	5	100	16,512	16,509	0.9	49	92.6	463
17	10	200	27681.34	27645.52	5.3	45	765	7650
22	20	200	36902.83	36897.7	12.5	43	1126	22520

① AEG denotes the average evolution generation required to obtain the best solution).

giving Order i, the changeover time between Order i and Order 0 is 0, namely $CT_{0,i}=0$. Assuming the starting time of Order 0 is big positive number $MST_j=NC$.

(2) Getting the current order p_i from the given order sequence P, selecting a Unit $\tilde{j}$ from Order p_i's available units set J_{s,p_i} according to the given order assignment rule.

(3) Arranging the completion time $TE_{p_i,\tilde{j}}$ of order p_i in Unit $\tilde{j}$ as close as possible to its due date, then calculating the start time $TS_{p_i,\tilde{j}}$ based on the completion time.

(4) Taking the start time of Order p_i in Unit $\tilde{j}$ as Unit $\tilde{j}$'s current start time, namely, $MST_{\tilde{j}}=TS_{p_i,\tilde{j}}$. Recording starting time of Order p_i in current Stage s, $sta_{s,p_i}=TS_{p_i,\tilde{j}}$.

(5) Recording order p_i as the latest finished order unit $\tilde{j}$,$RMCP_{\tilde{j}}=p_i$.

(6) Recording $TE_{P_{i,\tilde{j}}}$ as the completion time of Order p_i in current Sage s, $c_{s,p_i}=TE_{p_i,\tilde{j}}$

(7) If p_i is the final order in order sequence P, go to step (8), if not, go to Step (2)

(8) Sorting the order starting time in current stage in descending order $\tilde{P}$, then use this sequence to update the current order production sequence, $P=\tilde{P}$

(9) Setting the order starting time in current stage as the virtual order due date in the previous stage, $VD_{s-1,i}=sta_{s,i}, \forall i\in I$.

(10) If current stage is the first stage, finish the order assignment procedure, if not, update current stage, $s=s-1$, then go to step (2) for the next order assignment.

By using the above forward order assignment strategy, all process constraints can be met exactly. However, during the backward order assignment procedure, although the order completion time in final stage can be set earlier than its real due date, the start time of order in first stage may be earlier than its release time or than the unit release time.

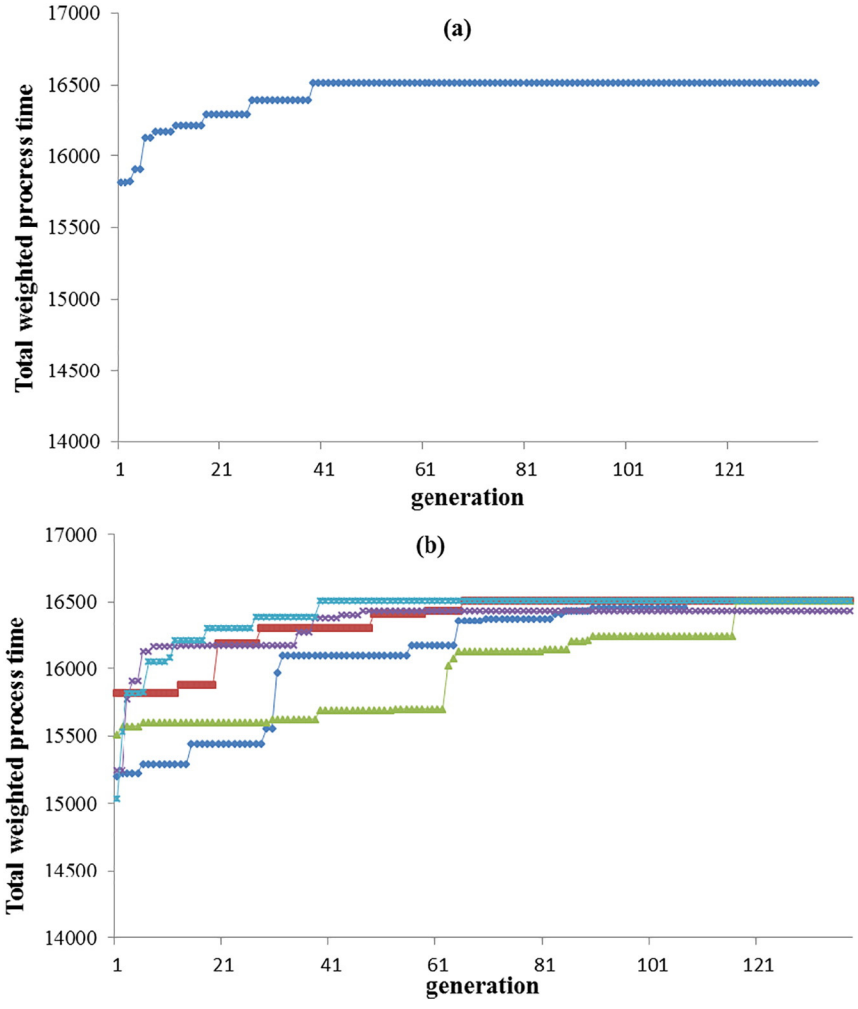


Fig. 4. Convergence curves of LCA for 10-order instance in Example 1.

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

Therefore, the formula of total weighted process time calculation Eq. (4) needs to be modified as follows:

$$\max\left\{\sum_{i=1}^{NI}\sum_{s=1}^{S} a_{s,i}c_{s,i} - NC\sum_{j=1}^{NJ}\max(UR_j - MST_j, 0) - NC\sum_{i=1}^{NI}\max(OR_i - sta_{1,i}, 0)\right\} \tag{5}$$

where, UR_j is the release time of unit j; MST_j is the start time of the latest finished order unit j; OR_i is the release time of order i. $sta_{1,i}$ is the start time of order i in first stage.

Through the above steps, an arbitrary order sequence can generate corresponding available scheduling scheme. Hence, the solution for MMSP with parallel units can be transform to determine optimal order sequence combined with proper heuristic order assignment rule to minimize total flow time or maximize total weighted process time.

3. Line-up Competition Algorithm for MMSP

It is clear that the determination of optimal order sequence and order assignment rule is a typical combinatorial optimization problem. Because of the complexity of this problem, the line-up competition algorithm (LCA) [18,19], which has been successfully applied to SMSP [17], is presented to optimize the alternative scheduling objectives, total flow time and total weighted process time. The procedures of LCA for solving MMSP are as follows.

(1) Generating m combined sequences, so-called m families, that consist of order sequence and order assignment rule to form the initial generation $X=\{Q_1, Q_2, \ldots, Q_m\}$ of LCA;
(2) Calculating the scheduling objective values $Y=\{f(Q_1), f(Q_2), \ldots, f(Q_m)\}$ corresponding to the m combined sequences $X=\{Q_1, Q_2, \ldots, Q_m\}$. Based on the objective values, the m combined sequences in X are ranked to form a line-up. For solving the problem of total flow time minimization, the m combined sequences are reordered as an ascending queue, and for solving the problem of total weighted process time maximization, they are reordered as a descending queue.
(3) Based on the position of each family in the line-up, its mutations times are therefore decided. The first family in the line-up has the fewest times of mutations, while the final family has the most times of mutations. Through this step, each combined sequence $Q_i (i=1,2,\ldots,m)$ in X, which is treated as parent here, can generate a corresponding new combined sequence $Qoffs_i (i=1,2,\ldots,m)$ as its child.
(4) One by one, comparing the objective function values of each Q_i and its child, and conserving the better one survives as parent in the next generation.

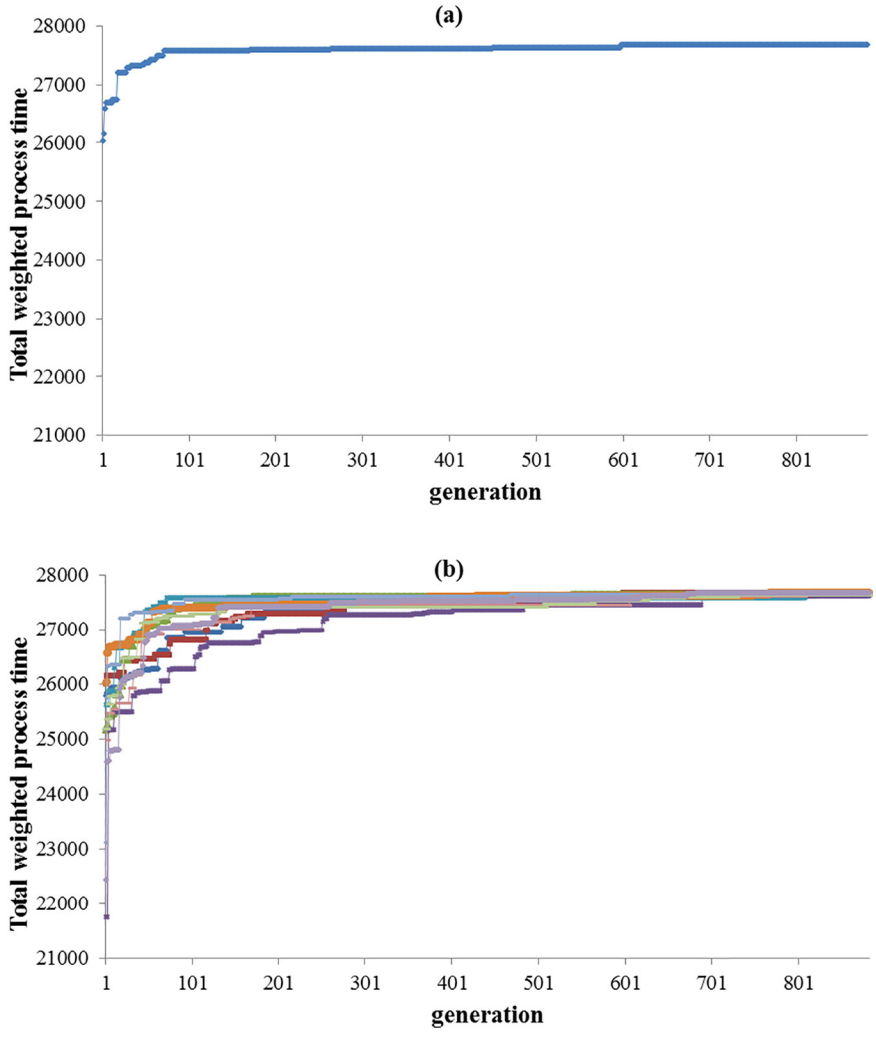


Fig. 5. Convergence curves of LCA for 17-order instance in Example 1.

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

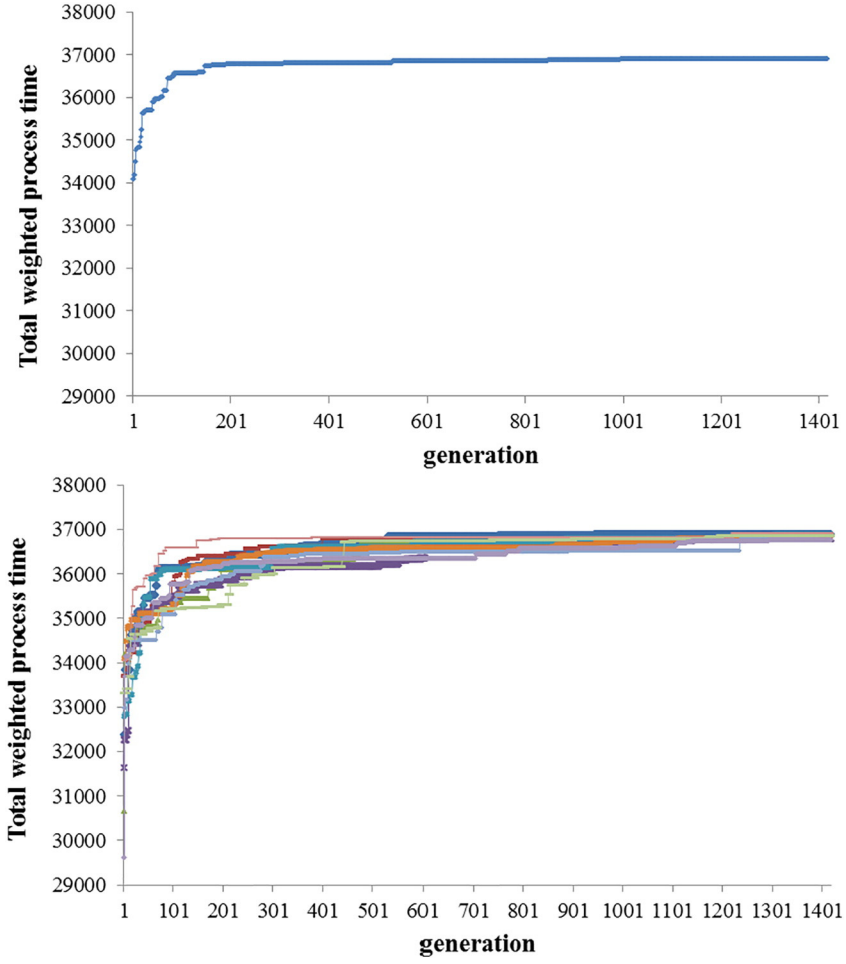


Fig. 6. Convergence curves of LCA for 22-order instance in Example 1.

(5) Testing the termination conditions, if they are satisfied, terminating the search process, outputting the best family in current generation as the final solution found by LCA; if not, go to Step (2) for continuing evolution.

Step (3) is the core of the algorithm. It balances local search and global search. The reason is that the first family in the line-up mutates only few times, which is beneficial to speed up local search for finding optimal solution quickly, and then the final family mutates many times, resulting in migration in a larger search space, which pays an important role in global search. In this study, according to the structure of individual, mutation of each combined sequence need to be separated. Since the size of order production sequence is only related to the number of orders. Hence, the swap-reverse mutation strategy proposed in Ref. [17] can still be employed for mutation. In MMSP, it is need to determine order assignment rule for every stage, therefore, the mutation method of proportional assigning search space is applied to mutation of order rules [19].

4. Case Study

In this section, the performance of the proposed approach for MMSP with parallel units is illustrated through the solution different size benchmark examples from literature. All of following scheduling cases were solved on Intel® Core™ i5 CPU and 4GB of memory using the MATLAB® 7.9 language environment. In some literature, the computational time of the algorithm is also adopted as a criterion for performance test. However, since the computer hardware and software used in our approach have varying differences with the ones used in the literature, it makes no sense to compare the absolute computational time directly. In this paper we use the average function evaluations for obtaining the best solution in each 50-test instead of the absolute computational time. In LCA, the number of families, m, is a very important parameter. It influences directly on the balance of local and global search. A smaller m is to benefit local search, but to go against global search. A larger one can provide better global search, but handicaps local search. We have to trade between the local and global search. Based on our computing experiences, it is appropriate for the number of families to be controlled between 5 and the number of integer variables related problem. A prespecified number, which denotes the duration generations of the current best solution, is took as terminate criterion.

4.1. Example 1

This example was given by Xue and Yuan [15]. It contains 5 stages and 25 parallel units, where Units 1–6 belong to Stage1, Units 7–9

Table 4
Results of two different methods for solving different instances in Example 1

Size	Method	Variable number		Constraint number	Best result
		Integer	Continuous		
5	MILP-x [15]	105	55	399	6852.76
	Proposed	5	5	30	6852.76
10	MILP-x [15]	218	110	1264	16,512
	Proposed	10	5	35	16,512
17	MILP-x [15]	364	187	3211	27,312.43
	Proposed	17	5	42	27,681.34
22	MILP-x [15]	4777	242	5260	36,290.69
	Proposed	22	5	47	36,902.83

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

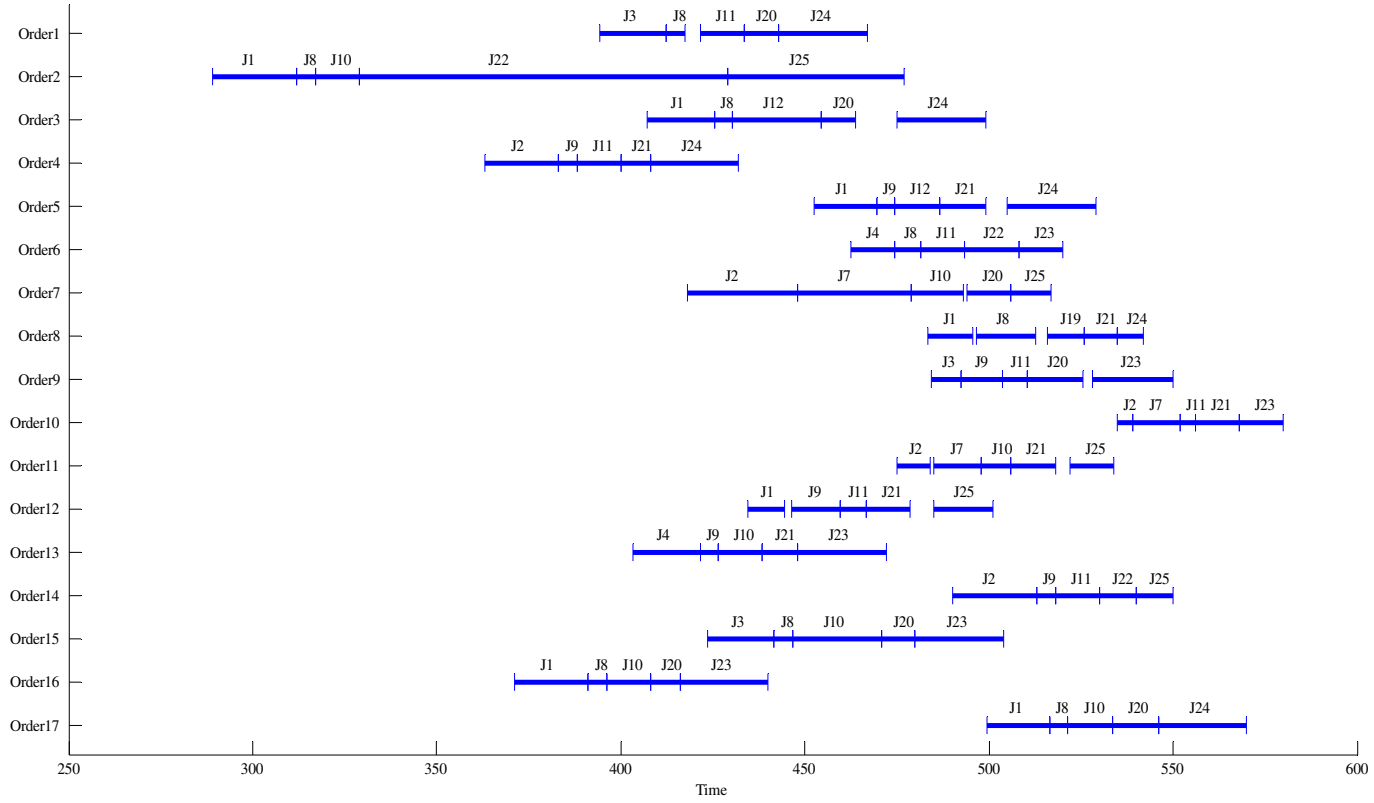


Fig. 7. Gantt chart of the best schedule for Example 1 (17 orders over 25 units).

belong to Stage2, Units 10–19 belong to Stage 3, Units 20–22 belong to Stage 4, and Units 23–25 belong to Stage 5. All production orders were processed sequentially from Stage 1 to Stage 5. The detailed process data for this example can be found in Ref. [15]. There were only unit release times, due date and forbidden units constraints, but no other complex constraints. Table 3 lists the results of maximizing the total weighted process time of different cases in Example 1 using the proposed approach.

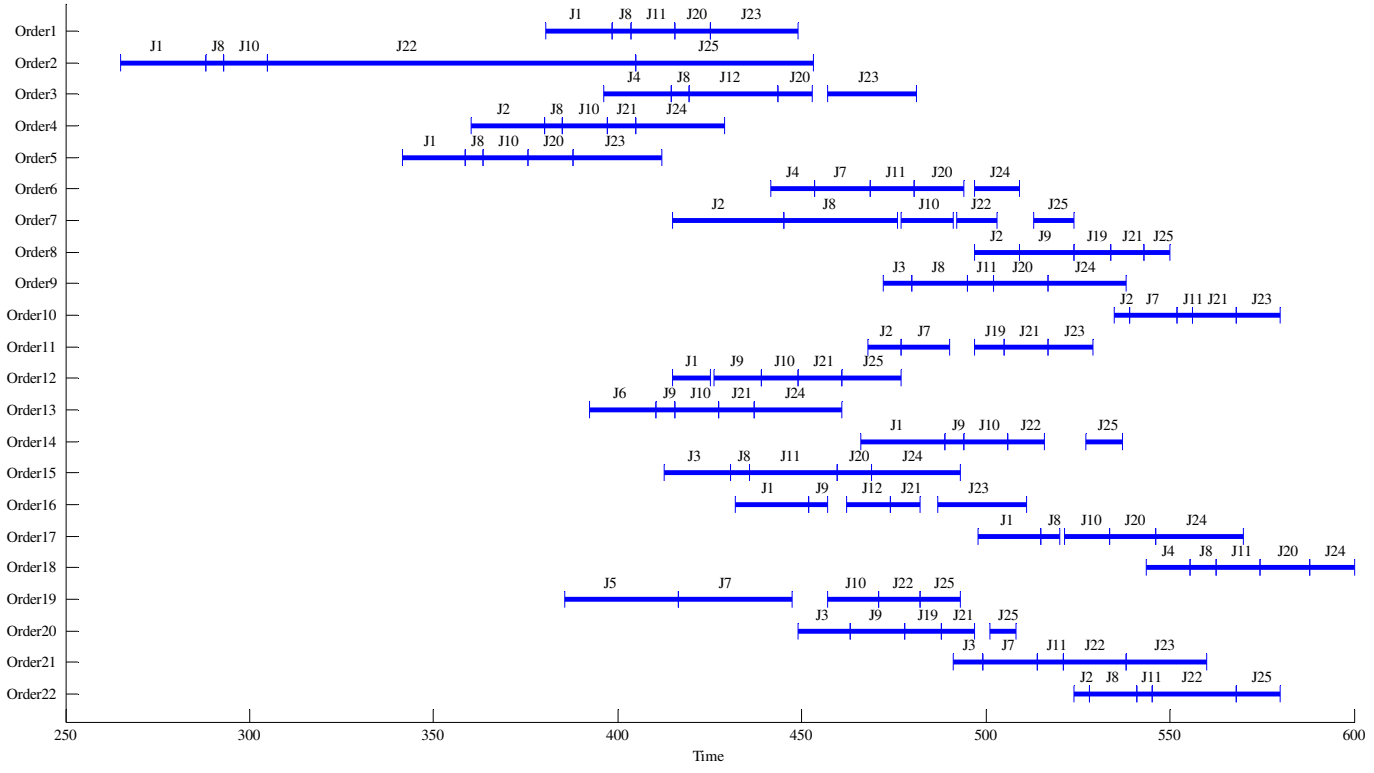


Fig. 8. Gantt chart of the best schedule for Example 1 (22 orders over 25 units).

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

Table 5
Results of solving different instances in Example 2 by LCA

Size	Parameter of LCA		Result					
	Family number	Terminate criterion	Best	Mean best	Standard deviation	Success number	AEG	Function evaluations
10	5	100	1344.8	1348.6	0.3	49	126	630
12	10	100	1650.4	1656.6	2.5	47	197	1970
16	10	150	2649.3	2675.2	12.6	46	558	5580
20	20	200	3614.8	36774	15.5	44	984	19680
24	20	400	4642.8	47094	26.6	46	1720	34400

As can be seen from Table 3, both the standard deviation of solutions and the differences between average solution and best solution are small when conducting each 50-test for solving different size instances by LCA. Especially, for solving 5-order case, the best solution can be obtained at no more than averagely 1.3 generations, where the objective function is evaluated only 6.5 times in average. With the problem size increasing, from 10-order to 22-order case, although the average function evaluations increase evidently, the algorithm can be terminated in time by using the proposed termination criterion, leading to avoiding redundant calculations. It is also clear from Table 3 that, during each 50-independent test, the number of success search are 50, 49, 45, and 43 respectively for solving 5-, 10-, 17-, 22-order cases, which indicate that the average success rate of our algorithm is over 93%. Figs. 4–6 show the converge process of LCA for solving 10-, 17-, and 22-order cases respectively. The sub-figures (a) in Figs. 4–6 are the evolution process of the best solutions in random tests, and the sub-figures (b) are the corresponding evolution process of different families in LCA. From each sub-figure (b) one can see that the objective values of different families in LCA increase alternately during the search process. It indicates that all the families compete for the first place in LCA, leading to the algorithm to converge rapidly to an optimal or near-optimal solution.

Xue and Yuan maximized the total weighted process time of different instances in this example by using pre-ordering MILP. Table 4 presents the best computational results by their methods and our proposed LCA respectively.

As can be seen from Table 4 two methods achieved the same best solution for solving for 5-order and 10-order instance. While for solving 17-order instance, a better solution was got by our proposed approach, the best schedule increased by nearly 1.4% compared to that found by pre-ordering MILP. Moreover, for solving 22-order scheduling problem, the best solution of our approach increased nearly 1.7%. It also can be seen from Table 4, the scheduling model of each instance built in Ref. [15] contained large number of integer variables, continuous variables and constraints. Moreover, the size of these variables and constraints increases sharply with the problem size increasing. While, in

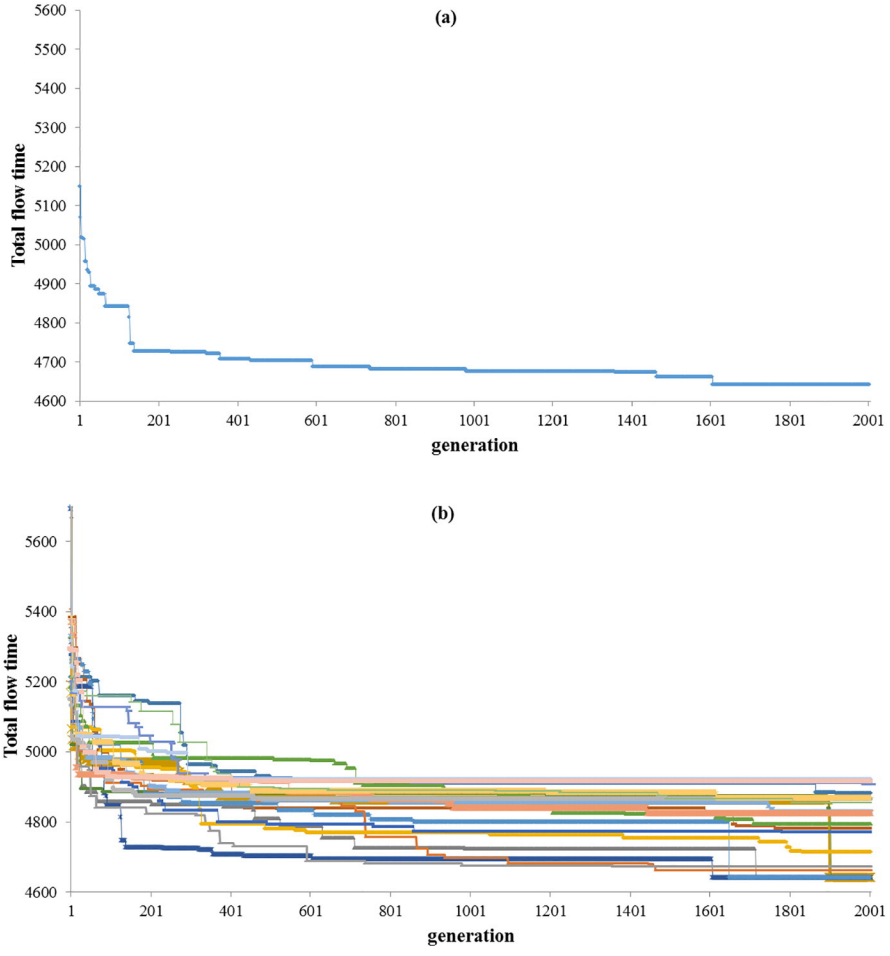


Fig. 9. Convergence curves of LCA for 24-order instance in Example 2.

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

our model, the number of continuous variables remains constant, only the number of integer variables and constraints changes synchronously with the number of order. It indicates that the proposed approach can reduce integer variables and continuous variables, simplify the complex production constraints, and therefore reduce the difficulty of problem solving. Figs. 7–8 are the Gantt charts of best schedules for 17-order and 22-order instances found by LCA respectively.

4.2. Example 2

This example was given by He and Hui [16], the production process is same as Example 1. The detailed process data of this example can be found in ref. [16]. The constraints of order and unit release time, forbidden units and forbidden sequence constraints, sequence-dependent and unit-depend changeover time constraints were all considered in this example. Table 5 presents the results of minimizing total flow time of different cases in Example 2 by LCA

Table 6
Results of GA and LCA for solving different instances in Example 2

Size	GA [16]		Proposed	
	Best result	Standard deviation	Best result	Standard deviation
10	1425.2	2.28	1344.8	0.3
12	1823.90	3.11	1650.4	2.5
16	3000.70	3.92	2649.3	5.7
20	4477.80	4.74	3614.8	10.3
24	5864.3	6.11	4642.8	19.2

It is clear that the similar trends with respect to results can be observed here as in Example 1. The standard deviation of solutions during each 50-test is still small. For solving the cases with less than 16 orders, LCA can rapidly find the best solutions, while for solving other cases over 16 orders, it also can obtain good solutions within acceptable generations through increasing the family number and extending the number of generations used for confirming the best result. The average success rate of our algorithm can still maintain over 92%.

Fig. 9 shows the converge process of LCA for solving 24-order case. Fig. 9(a) is the evolution process of the best solutions in a random test, and Fig. 9(b) is the corresponding evolution process of different families in LCA. It is evident from Fig. 9(b) that although some families are trapped to local minimum, there are still families that jumped out from local optima, and through competing for the first place in LCA, finally found the global minimum.

He and Hui minimized the total flow time of five typical instances with up to 24 orders by using genetic algorithm (GA). In this section, the same instances were studied by using our proposed approach, and then the results were compared with those obtained by GA in Ref. [16]. Table 6 presents the results comparison of Example 2 by GA and LCA. Since many complex constraints exist in Example 4, the difficulty for solving all instances increases evidently. For comparison purposes, 10 independent tests of computation for each instance were performed as Ref. [16] did. The deviation between the best solution and the results of 10-independent tests are used to evaluate the performance of the algorithms.

It is clear from Table 6, our approach can always obtain better solutions than those gotten by GA for solving all the instances. Moreover, the

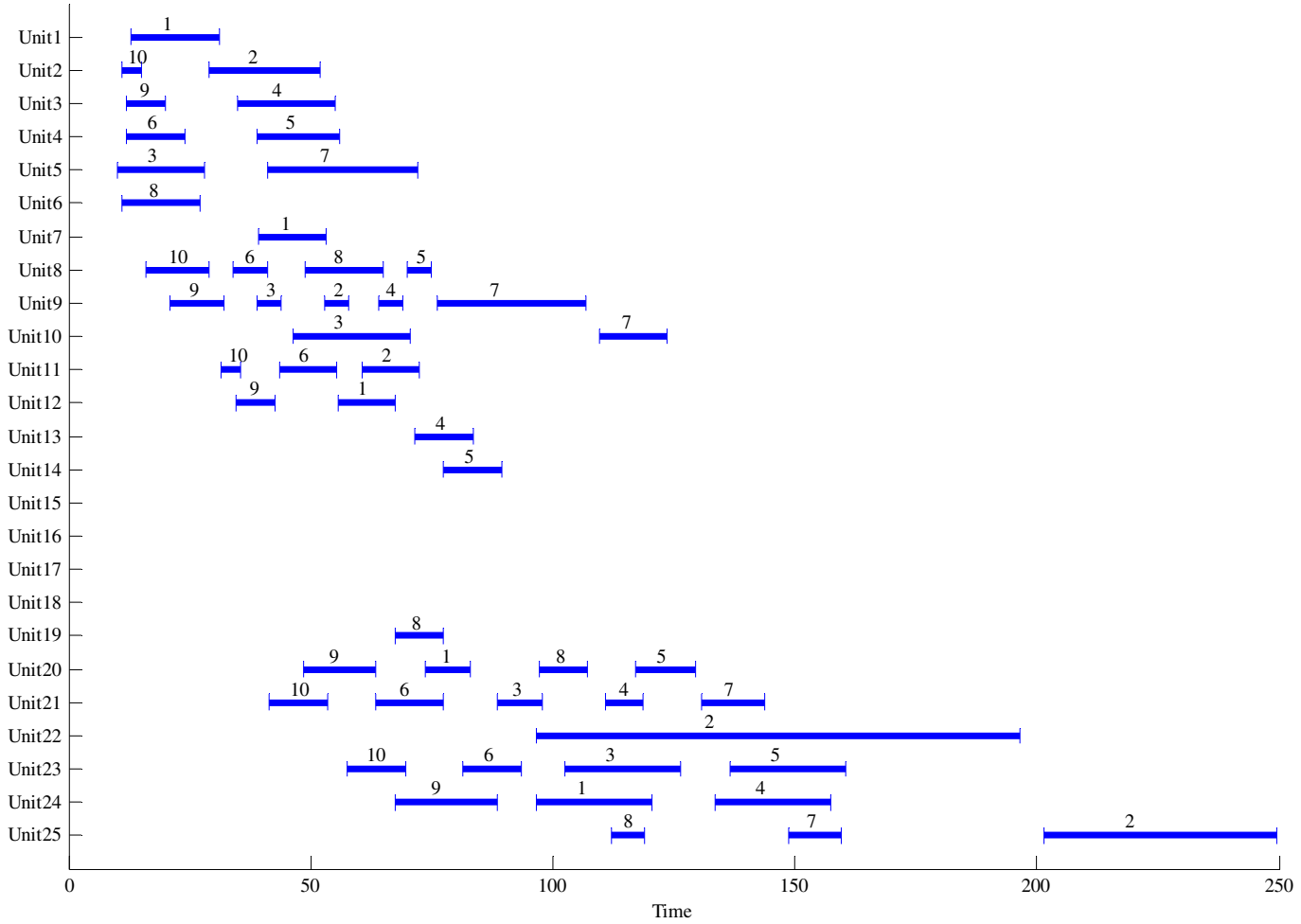


Fig. 10. Gantt chart of the best schedule for Example 2 (10 orders over 25 units).

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

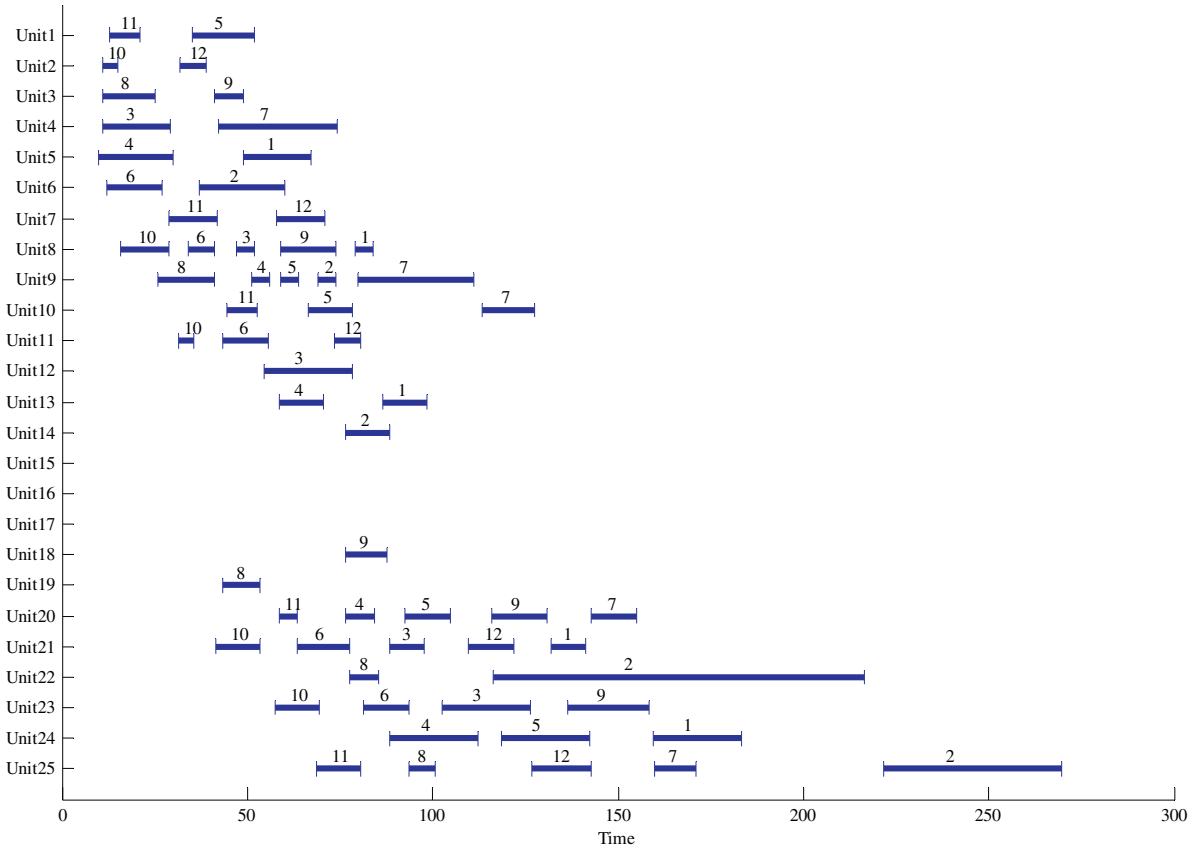


Fig. 11. Gantt chart of the best schedule for Example 2 (12 orders over 25 units).

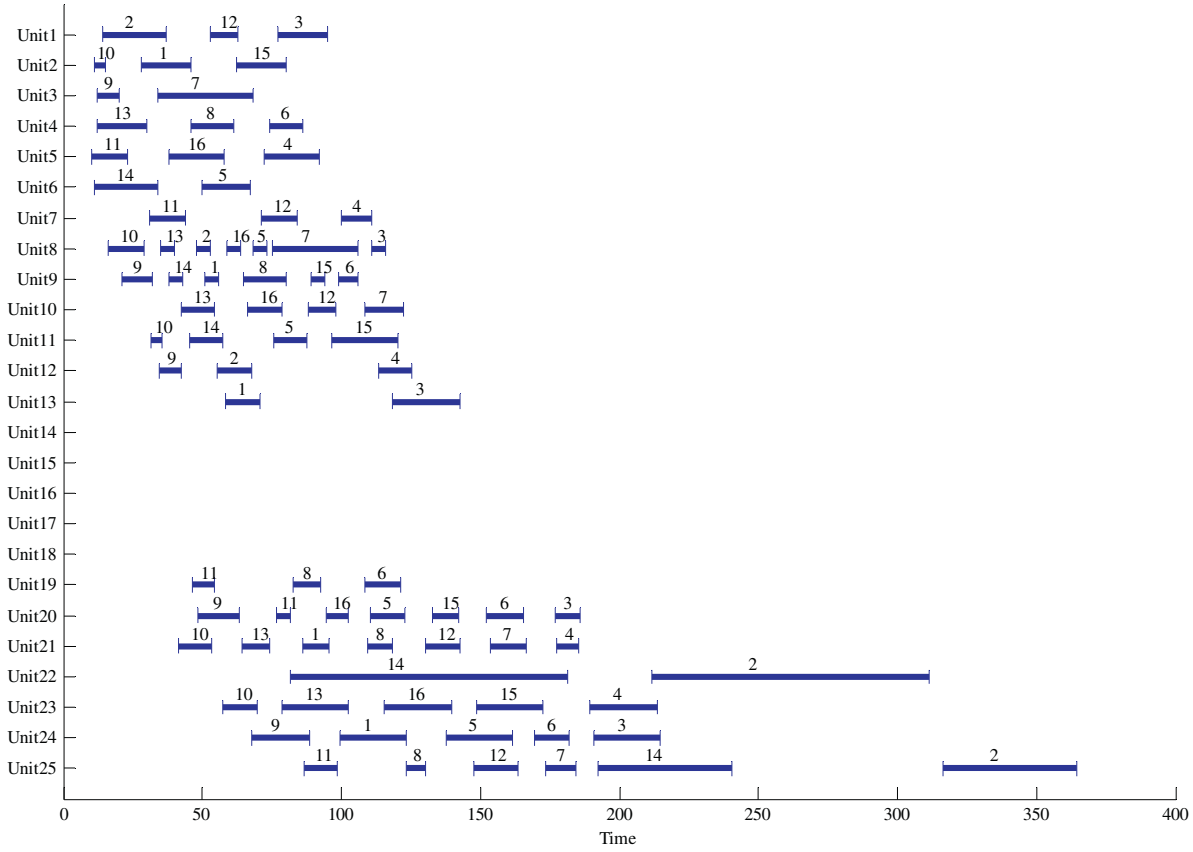


Fig. 12. Gantt chart of the best schedule for Example 2 (16 orders over 25 units).

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

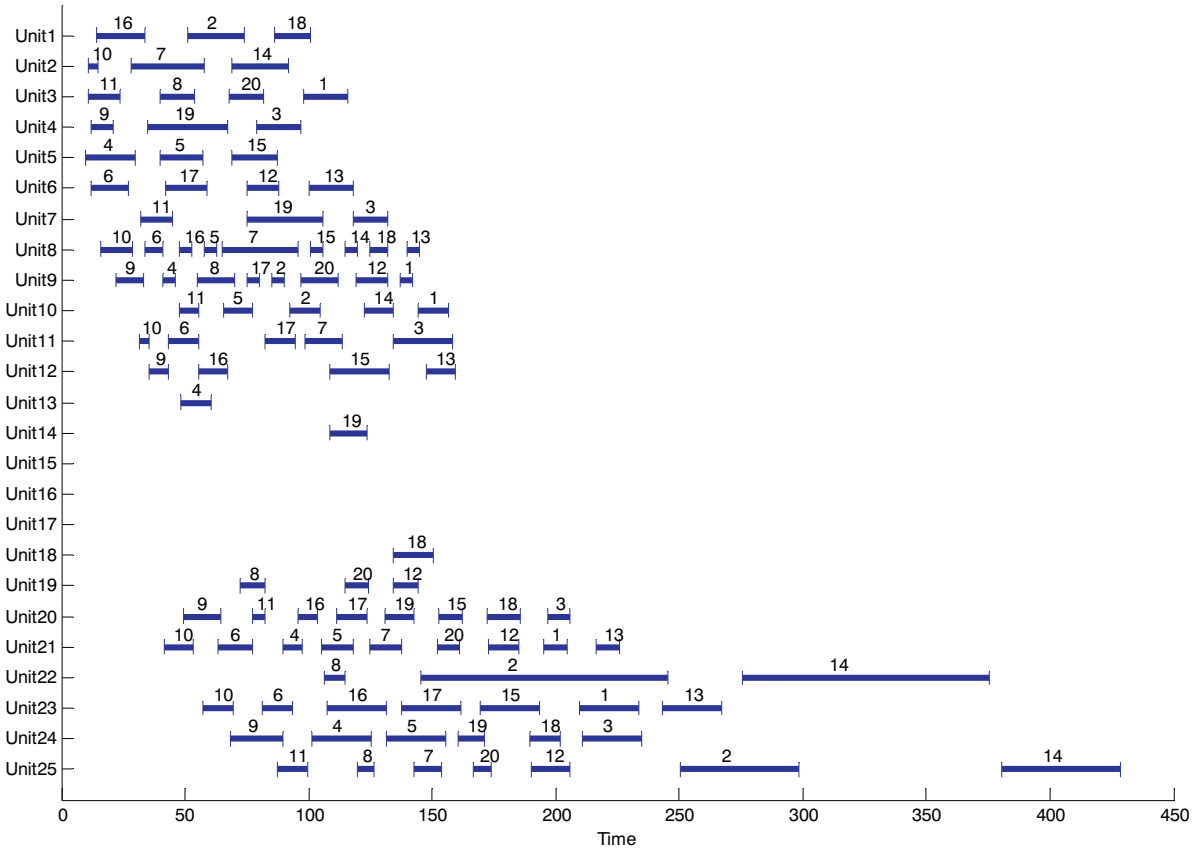


Fig. 13. Gantt chart of the best schedule for Example 2 (20 orders over 25 units).

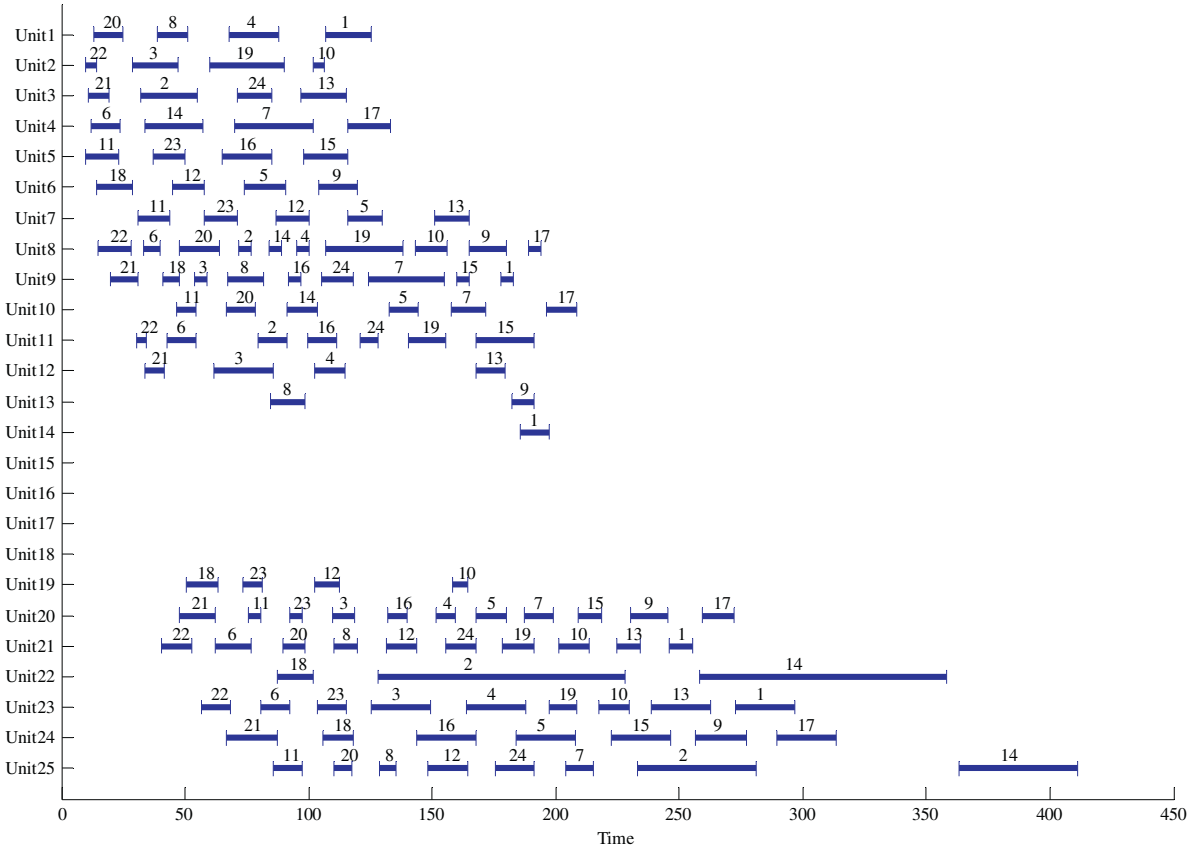


Fig. 14. Gantt chart of the best schedule for Example 2 (24 orders over 25 units).

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

Table 7
Best scheduling scheme of 24 orders

Order	Stage 1			Stage 2			Stage 3			Stage 4			Stage 5		
	U	ST	CT	U	ST	CT	U	ST	CT	U	ST	CT	U	ST	CT
1	1	107	125.1	9	178	183	14	185.5	197.5	21	246	255.5	23	272.5	296.5
2	3	32	55	8	72	77	11	79.5	91.5	22	128	228	25	233	281
3	2	29	47.1	9	54	59	12	61.5	85.5	20	109.5	118.8	23	125.5	149.5
4	1	68	88	8	95	100	12	102.5	114.5	20	151.7	159.6	23	163.6	187.6
5	6	74	91	7	116	130	10	132.5	144.5	20	167.6	180.1	24	184.1	208.1
6	4	12	24	8	33	40	11	42.5	54.5	21	62.5	76.5	23	80.5	92.5
7	4	70	102	9	124	155	10	157.5	171.5	20	187.1	199.1	25	204.1	215.1
8	1	39	51	9	67	82	13	84.5	98.5	21	110.5	119.5	25	128.5	135.5
9	6	104	120	8	165	180	13	182.5	191.5	20	230.4	245.4	24	256.4	277.4
10	2	102.1	106.1	8	143	156	19	158.5	164.5	21	201.5	213.5	23	217.5	229.5
11	5	10	23	7	31	44	10	46.5	54.5	20	75.5	80.5	25	85.5	97.5
12	6	45	58	7	87	100	19	102.5	112.5	21	131.5	143.5	25	148.5	164.5
13	3	97	115.1	7	151	165	12	167.5	179.5	21	224.5	234	23	238.5	262.5
14	4	34	57	8	84	89	10	91.5	103.5	22	258	358	25	363	411
15	5	98	116.1	9	160	165	11	167.5	191.5	20	209.1	218.4	24	222.4	246.4
16	5	65	85	9	92	97	11	99.5	111.5	20	131.8	139.7	24	143.7	167.7
17	4	116	133	8	189	194	10	196.5	208.5	20	259.4	271.9	24	289.4	313.4
18	6	14	29	9	41	48	19	50.5	63.5	22	87.5	102	24	106	118
19	2	60.1	90.1	8	107	138	11	140.5	155.5	21	178.5	191.5	23	197.6	208.6
20	1	13	25	8	48	64	10	66.5	78.5	21	89.5	98.5	25	110.5	117.5
21	3	11	19	9	20	31	12	33.5	41.5	20	47.5	62.5	24	66.5	87.5
22	2	10	14	8	15	28	11	30.5	34.5	21	40.5	52.5	23	56.5	68.5
23	5	37	50	7	58	71	19	73.5	81.5	20	92.5	97.5	23	103.5	115.5
24	3	71	85	9	105	118	11	121	128	21	155.5	167.5	25	175.5	191.5

U: Unit number; ST: Start time; CT: Completion time

solution quality gap between the two methods grows rapidly with the problem size increasing. For instance, when solving 10-order instance, the difference between the result obtained by our approach and that by GA is 5.98%. While for solving 24-order scheduling problem, the difference between them is up to 26.31%.

Figs. 10–14 show the Gantt charts of the best schedules found by our approach for the five instances in this example. It can be seen from these Gantt charts that the arrangement of orders in this example is completely different from the previous example. Since the scheduling objective is total flow time minimization, the forward order assignment strategy is adopted, thus the arrangements of all production steps are as close as possible to the beginning of the scheduling duration.

In addition, as also can be seen clearly from the figures, some devices is not adopted in the whole production processes. The main reasons may be that the number of units in certain stage is adequate and the performances of some devices are evidently worse than others in the same stage, so these devices were automatically avoided by our heuristic order assignment rule, *e.g.* in this example, Stage 3 has 10 units, from Unit 10 to Unit 19. And the order processing times in Units 15–18 are obviously longer than those in other units of the same stage. Hence these devices were seldom used in our best scheduling schemes. Table 7 shows the detailed scheduling results of optimal scheduling scheme of 24 orders.

5. Conclusions

This paper presents an improved scheduling model for MMSP with parallel units. According to the characteristics of multi-stage multi-product batch process, the productions of this process are divided into multiple sequentially connected single process stages by bringing two concepts of 'virtual due date' and 'virtual order release time'. In each stage the principle of 'first come first dealt' is insisted. First of all, processing unit is assigned for the coming order by heuristic rule, and then forward or backward order assignment strategy is employed to arrange the order production according to different scheduling objectives. Based on the above hierarchical scheduling strategies, an arbitrary order sequence can always be converted to a legal schedule with the help of heuristic rules. The optimal sequencing of orders and the best rule are explored by the line-up competition algorithm (LCA), where different mutation strategies are proposed to improve the efficiency of LCA. The effectiveness of proposed method is illustrated by solving a group typical benchmark MMSP cases. Computational results show that the proposed approach can easily obtain the same optimal solutions of small size problems involving up to 10 orders as those reported in literature. While, for relatively large size problems from 10 orders up to 24 orders with different complex constraints, the proposed approach obtained new solutions better than the MILP and GA did. And with the problem size increasing, the solutions obtained by the proposed approach are improved remarkably. Through comparative study, the proposed approach has shown the potential for solving large size MMSP. It is apparent that significant advances have been made in the area of scheduling of chemical processes in the past decade. Therefore further research efforts should aim at (a) developing mathematical models and algorithms that reduce the burden of computing for solving large-scale industrial scheduling applications; (b) considering the uncertainty may occurred in real industrial applications, such as the uncertainty in processing times, prices, changes in product demands, and equipment failure/breakdown; and (c) integrating the scheduling with design, synthesis, control and planning.

Appendix A. Supplementary data

Supplementary data to this article can be found online at http://dx.doi.org/10.1016/j.cjche.2017.03.014.

References

[1] C.A. Floudas, X. Lin, Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review, *Comput. Chem. Eng.* 28 (11) (2004) 2109–2129.
[2] A. Carlos, C.A. Mendez, J. Cerda, I.E. Grossmann, *et al.*, State-of-the-art review of optimization methods for short-term scheduling of batch processes, *Comput. Chem. Eng.* 30 (2006) 913–946.
[3] Y. Liu, I.A. Karimi, Novel continuous-time formulations for scheduling multi-stage batch plants with identical parallel units, *Comput. Chem. Eng.* 31 (2007) 1671–1693.
[4] M.E. Dogan, I.E. Grossmann, Slot-based formulation for the short-term scheduling of multistage multiproduct batch plants with sequence-dependent changeovers, *Ind. Eng. Chem. Res.* 47 (4) (2008) 1159–1183.

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014

[5] L.J. Zeballos, J.M. Novas, G.P. Henning, A CP formulation for scheduling multiproduct multistage batch plants, *Comput. Chem. Eng.* 35 (2011) 2973–2989.
[6] S. Velez, C.T. Maravelias, Multiple and nonuniform time grids in discrete-time MIP models for chemical production scheduling, *Comput. Chem. Eng.* 53 (2013) 70–85.
[7] I. Harjunkoski, C. Maravelias, P. Bongers, *et al.*, Scope for industrial applications of production scheduling models and solution methods, *Comput. Chem. Eng.* 62 (2014) 161–193.
[8] J.M. Pinto, I.E. Grossmann, A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants, *Ind. Eng. Chem. Res.* 34 (9) (1995) 3037–3051.
[9] J.M. Pinto, I.E. Grossmann, An alternate MILP model for short-term scheduling of batch plants with preordering constraints, *Ind. Eng. Chem. Res.* 35 (1) (1996) 338–342.
[10] C.W. Hui, A. Gupta, Harke A.J. van der Meulen, A novel MILP formulation for short-term scheduling of multi-stage multi-product batch plants with sequence-dependent constraints, *Comput. Chem. Eng.* 24 (12) (2000) 2705–2717.
[11] C.W. Hui, A. Gupta, A novel MILP formulation for short-term scheduling of multi-stage multi-product batch plants, *Comput. Chem. Eng.* 24 (2–7) (2000) 1611–1617.
[12] S. Gupta, I.A. Karimi, An improved MILP formulation for scheduling multiproduct multistage batch plants, *Ind. Eng. Chem. Res.* 42 (2003) 2365–2380.
[13] P.M. Castro, I.E. Grossmann, New continuous-time MILP model for the short-term scheduling of multistage batch plants, *Ind. Eng. Chem. Res.* 44 (2005) 9175–9190.
[14] P.M. Castro, I.E. Grossmann, A.Q. Novais, Two new continuous-time models for the scheduling of multistage batch plants with sequence dependent changeovers, *Ind. Eng. Chem. Res.* 45 (2006) 6210–6226.
[15] Y.F. Xue, J.Q. Yuan, Scheduling model for a multi-product batch plant using a pre-ordering approach, *Chem. Eng. Technol.* 31 (3) (2008) 433–439.
[16] Y. He, C.W. Hui, Genetic algorithm for large-size multi-stage batch plant scheduling, *Chem. Eng. Sci.* 62 (5) (2007) 1504–1523.
[17] B. Shi, L.X. Yan, W. Wu, Rule-based scheduling of single-stage multiproduct batch plants with parallel units, *Ind. Eng. Chem. Res.* 51 (2012) 8535–8549.
[18] L.X. Yan, Solving combinatorial optimization problems with line-up competition algorithm, *Comput. Chem. Eng.* 27 (2) (2003) 251–258.
[19] L.X. Yan, K. Shen, S.H. Hu, Solving mixed integer nonlinear programming problems with line-up competition algorithm, *Comput. Chem. Eng.* 28 (12) (2004) 2647–2657.

Please cite this article as: B. Shi, *et al.*, Rule-based scheduling of multi-stage multi-product batch plants with parallel units, *Chin. J. Chem. Eng.* (2017), http://dx.doi.org/10.1016/j.cjche.2017.03.014