Computers and Chemical Engineering 87 (2016) 63–67

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Note

Simultaneous subtour elimination model for single-stage multiproduct parallel batch scheduling with sequence dependent changeovers

Yingzong Liang, Chi Wai Hui*

Department of Chemical and Biomolecular Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 21 July 2015
Received in revised form
24 December 2015
Accepted 29 December 2015
Available online 7 January 2016

Keywords:
Batch scheduling
MILP
Parallel non-uniform production
Subtour elimination

ABSTRACT

In this paper a mixed-integer linear programming (MILP) model is presented to minimize makespan of single-stage multiproduct parallel batch production with sequence dependent changeovers. The computational inefficiency and suboptimal problems are addressed by the tight and rigorous formulation of the proposed model. Subtours (subcycles) are eliminated simultaneously so that the optimal solution is obtained in one step. The proposed model is tested with two examples. The results show that the model obtains the global optimal solutions with significant improvement in solution time.

1. Introduction

Despite the optimization of batch scheduling with mathematical models has been intensively studied, the solution of this problem is still computationally expensive or cannot ensure the optimality for large-scale scheduling problems.

Cerdá et al. (1997) presented continuous time approach and formulated tri-index model for single-stage multiproduct batch scheduling problem. Hui and Gupta (2001) proposed bi-index model with further variables reduction. Although the models are rigorous and the optimal solution is guaranteed, the main drawback of these approaches is that the formulation involves many big-M constraints in calculating the finishing time of the orders, which enlarges the relaxation area and the relaxation gap of the solution. As a consequence, the model becomes computationally expensive to solve.

Castro et al. (2006) proposed a continuous time four-index (CT4I) model for batch scheduling with sequence dependent changeovers. In their model, time grids are applied to represent the position of individual order and consecutive orders. Since the sequence of the time grids is fixed, the time difference between time grids can be calculated without using big-M formulation. The authors found that the CT4I model solved single-stage problem efficiently with good solutions thanks to tighter formulation of the time constraints and better relaxation of the model. However, since the model contains four-index variables (immediate precedence, units, time grids), the size of the model grows significantly when solving large problems, which may influence its solution efficiency.

Erdirik-Dogan and Grossmann (2007) proposed E-D&G model for planning and scheduling optimization. The model is decomposed into planning level and scheduling level. In scheduling level, makespan is calculated by the longest processing time of the units instead of the finishing time of each task, and thus big-M constraints are avoided and the model's relaxation gap is tightened. This results in better lower bound and improvement in solution time compared with the bi-index and tri-index model. However, since its solution may feature subtours, they proposed an iteration method to eliminate the subtours. While the feasible and optimal solutions may be pruned by using the iterations, the resultant solution can be suboptimal (Castro et al., 2008).

To address computational inefficiency and suboptimal solution problems of the existing methods, a simultaneous subtour elimination model is presented to minimize makespan for single-stage batch scheduling with parallel units and sequence dependent changeovers. The proposed MILP model does not require iteration for subtours elimination so that the optimality of the solution is guaranteed. The model shares similar subtour elimination strategy of Miller et al.'s (1960) model in traveling salesman problem. Besides, the proposed model is tighter in relaxation gap, which results in better solution performance.

* Corresponding author. Tel.: +852 2358 7137; fax: +852 2358 0054.
E-mail addresses: yliangam@connect.ust.hk (Y. Liang), kehui@ust.hk (C.W. Hui).

http://dx.doi.org/10.1016/j.compchemeng.2015.12.024
0098-1354/

Nomenclature

Indices

i, j	orders
u	units

Sets

I	orders to be processed
U	units available
I_u	orders can be processed in unit u
U_i	units available to process order i
SU_i	feasible successor of order i
PR_i	feasible predecessor of order i

Parameters

TP_{iu}	processing time of order i in unit u
C_{ij}	changeover time from order i to order j
N	total number of orders

Variables

H	makespan of the batch process

Binary variables

w_{iu}	assignment of order i to unit u
s_{iu}	assignment of order i as the starting order of unit u
r_{iu}	assignment of order i as the finishing order of unit i
x_{iju}	immediate precedence of order i over order j at unit u

Integer variables

a_i	position number of order i

2. Problem definition

The batch scheduling problem addressed in this paper is determined by following specifications.

(a) A fixed number of parallel non-uniform production units are available for given number of orders.
(b) Each order involves a single product, each order's processing time is fixed and unit dependent.
(c) Each order must be and must only be processed in one unit.
(d) Changeover time is required for orders' transition, and it is fixed and sequence dependent.
(e) Each unit must process at least one order.

3. Model formulation

In this section, the simultaneous subtour elimination model is developed for makespan minimization. Integer variables a_i and corresponding constraints are introduced to exclude all the solutions with subtour(s). The usage of big-M constraints is avoided by calculating the total processing time of units instead of the finishing time of the orders.

(a) *Each unit contains a unique starting and finishing order*. Logically, a unit has only one starting order and only one finishing order of its processing orders. This logic is assured by Eqs. (1) and (2).

$$\sum_{i \in I_u} s_{iu} = 1 \quad \forall u \in U \tag{1}$$

$$\sum_{i \in I_u} r_{iu} = 1 \quad \forall u \in U \tag{2}$$

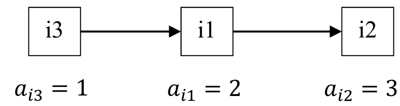

Fig. 1. Position number of orders.

where binary variables s_{iu} and r_{iu} denote order i is assigned as the starting and finishing order of unit u, respectively.

(b) *Each order must be processed by one unique unit*. According to the problem definition (c) in Section 2, each order must be assigned to one unique unit for processing. The requirement is accomplished by Eq. (3).

$$\sum_{u \in U_i} w_{iu} = 1 \quad \forall i \in I \tag{3}$$

where binary variable w_{iu} represents the assignment of order i to unit u.

(c) *Relationship between unit assignment and immediate precedence*. Eq. (4) enforces that if and only if an order is the last order of the unit or has a successor in the unit, it is assigned to the unit. Eq. (5) is similar to Eq. (4) that if and only if an order is the first order of the unit or has a predecessor in the unit, it is assigned to the unit.

$$r_{iu} + \sum_{j \in SU_i} x_{iju} = w_{iu} \quad \forall i \in I, \quad u \in U_i \tag{4}$$

$$s_{ju} + \sum_{i \in PR_j} x_{iju} = w_{ju} \quad \forall j \in I, \quad u \in U_j \tag{5}$$

where binary variable x_{iju} represents the immediate precedence of order i over order j at unit u.

(d) *Simultaneous subtour elimination*. Instead of using iteration, a new constraint is presented in Eq. (7) to eliminate subour(s) simultaneously.

$$a_i - a_j + 1 \le N \left(1 - \sum_{u \in U_j \cap U_i} x_{iju}\right) \quad \forall i \in I, \quad j \in SU_i \tag{6}$$

where integer variable a_i denotes the position number of order i among the orders (see Fig. 1).

Eq. (6) states that if order j is successor of order i $\left(\sum_{u \in U_j \cap U_i} x_{iju} = 1\right)$, then position number of j must be greater than that of $i(a_i + 1 \le a_j)$; conversely, if the position number of i is greater than or equal to that of $j(a_i \ge a_j)$, then j cannot be the successor of i $\left(\sum_{u \in U_j \cap U_i} x_{iju} = 0\right)$ in order to satisfy the constraint. Eq. (6) prevents the following orders from being assigned as the predecessors of the preceding orders, or in other words, avoids forming subtour. Thus, subtours are eliminated simultaneously. Interestingly, since Eq. (6) only enforces the difference of position number to be greater than 1, variable a_i can be defined as integer variable or continuous variable.

A simple example illustrates the idea of simultaneous subtour elimination (see Fig. 2(a)). Without loss of generality, it is assumed that $i2$ is the successor of $i1(x_{i1i2} = 1)$, and so on. According to Eq. (6), there is $a_{i1} < a_{i2} < a_{i3} < a_{i4}$, which avoids $i2$ from being the successor of $i3(x_{i3i2} = 0)$ as $a_{i3} > a_{i2}$. Therefore, subtour cannot be formed with the constraint of Eq. (6).

Fig. 2(b) shows a solution with subtour that $i2$ and $i3$ are successors of each other; $i1$ is the 'starting' order of the sequence, $i4$ is the 'finishing' order, and $i4$ is the successor of $i1$. The solution