Operations Research Perspectives 4 (2017) 49–55

Contents lists available at ScienceDirect

Operations Research Perspectives

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

Efficient mixed integer programming models for family scheduling problems

Meng-Ye Lin, Yarlin Kuo*

Department of Industrial Engineering and Management, Yunlin University of Science and Technology, Yunlin, Taiwan

ARTICLE INFO

Article history:
Received 9 November 2016
Revised 3 March 2017
Accepted 15 March 2017
Available online 23 March 2017

Keywords:
Family scheduling
Sequence independent setup
Total weighted completion time
Maximum lateness

ABSTRACT

This paper proposes several mixed integer programming models which incorporate optimal sequence properties into the models, to solve single machine family scheduling problems. The objectives are total weighted completion time and maximum lateness, respectively. Experiment results indicate that there are remarkable improvements in computational efficiency when optimal sequence properties are included in the models. For the total weighted completion time problems, the best model solves all of the problems up to 30-jobs within 5 s, all 50-job problems within 4 min and about 1/3 of the 75-job to 100-job problems within 1 h. For maximum lateness problems, the best model solves almost all the problems up to 30-jobs within 11 min and around half of the 50-job to 100-job problems within 1 h.

1. Introduction

Batching jobs that share the same setup on a machine to increase productivity is a common practice in manufacturing. Such batching of jobs is often classified as a family in scheduling. When processing two jobs belonging to different families consecutively, a setup is required between them. The batching operation often leads to job lateness and results in poor delivery performance. How to trade-off productivity and delivery performance is a tricky scheduling problem. This problem is called batch/family/group scheduling with setups in the literature. For past research on family scheduling see Refs. [12,16,23].

The solution approaches to solve family scheduling problems include mathematical programming [1,3,8,13], branch and bound [7,18,19], dynamic programming [6,9,11,17] and heuristic/meta heuristics [4,5,14,21,24,25]. The last approach, heuristic/meta heuristics, is often designed to find good solutions for large scale problems. Except for mathematical programming, time consuming and complicated coding is required to find optimal/good sequences. On the other hand, commercial solvers such as LINGO and CPLEX are available to solve mathematical models. Thus, the start-up task for mathematical programming is much less than those of the other three solution approaches. Often the mathematical programming approach is applied to find an optimal schedule with limited CPU time, say 1 h, and is often used to solve small size problems. Thus, the mathematical programming approach is applicable for small-size enterprise scheduling problems or as a basis for rolling-horizon based scheduling techniques.

Webster and Baker [23] give an overview of optimal properties on family scheduling on a single machine. Very often these optimal properties are included in the branch and bound, dynamic programming and heuristic solution approaches to achieve better computational efficiency. In this study, we include some optimal properties of the family scheduling problem into mathematical programming models and investigate the effects of this inclusion on computational efficiency. In solving scheduling problems with mathematical programming, there are several well-known mixed integer programming (MIP) formulations proposed in the literature, which include: (1) the disjunctive formulation developed by Manne [10] which contains precedence variables that define the precedence order of any two jobs and disjunctive constraints that relate the completion times between any two jobs (2) the time indexed formulation proposed by Sousa and Wolsey [20] which defines time variables that relate jobs to the corresponding processing starting times in a finite discrete time horizon, (3) the linear ordering formulation developed by Potts [15] which adds triangle inequalities among the precedence variables of any three jobs and (4) the sequence position formulation developed by Wagner [22] which contains sequence position variables that relate jobs to corresponding positions in a sequence. For the single machine scheduling problem with release dates and sequence dependent setup and the objectives of total weighted tardiness and total weighted completion time, respectively, Nogueira and Carvalho (2014) compares the performance of six MIP formulations,

* Corresponding author.
E-mail addresses: M10221311@yuntech.edu.tw (M.-Y. Lin), kuoyl@yuntech.edu.tw (Y. Kuo).

http://dx.doi.org/10.1016/j.orp.2017.03.001

which include the four formulations above and two improved formulations and finds that the disjunctive formulation solves a greater number of problems.

In this study, we adopt the mathematical programming approach to find the optimal sequence of a single machine family scheduling problem with family sequence independent setup time and the objectives of total weighted completion time (TWC) and maximum lateness ($L_{\max}$), respectively. Three MIP formulations which include the optimal sequence properties are proposed. The performance of the proposed three MIP formulations is compared with two MIP models that do without the optimal sequence properties by computational experiments under different operating scenarios. The rest of this paper is organized as follows. Section 2 gives problem statements and presents MIP formulations. Section 3 gives the computational results and the paper concludes in Section 4.

2. Problem statements and MIP formulations

Consider the problem of scheduling N jobs belonging to F families on a single machine. A family sequence independent setup is required when a machine switches from idle to busy and from processing jobs in one family to jobs in another family. The objectives are to minimize total weighted completion time (TWC) and maximum lateness, respectively. Before proposing the MIP models, we give two optimal sequence properties of the family scheduling problems first.

Bruno and Sethi's optimal property: For the TWC problems, there is an optimal sequence in which jobs in the same family are ordered by SWPT (shortest weighted processing time first) [2].

Monma and Potts's optimal property: For the $L_{\max}$ problems, there is an optimal sequence in which jobs in the same family are ordered by EDD (earliest due date first) [11].

We include these two optimal properties in the constraints of MIP formulations and investigate the effects of these inclusions. Five MIP formulations are proposed to solve this problem, they are (1) family linear ordering formulation, FLO, (2) FLO with jobs in the same family sequenced in SWPT for TWC problems, FLO_{swpt}, and in EDD for $L_{\max}$ problems, FLO_{edd}, (3) ordered linear ordering formulation, OLO, where jobs in the same family are sequenced in SWPT for TWC problems and in EDD for $L_{\max}$ problems, (4) disjunctive formulation, DJ and (5) DJ with jobs in same family sequenced in SWPT for TWC problems, DJ_{swpt}, and in EDD for $L_{\max}$ problems, DJ_{edd}. The FLO and DJ formulations are proposed for the purpose of comparison. The notation and parameters used in the MIP formulations are as follows.

Notation and parameters

$a_{(i,j)}$: job j of family i.
F: set of all families.
N: set of all jobs.
s_i: setup time of family i.
$s_{(i,k)}$: setup time from family i to family k, $s_{(i,k)} = s_k$, $s_{(i,k)} = 0$ if $i = k$.
$p_{(i,j)}$: processing time of $a_{(i,j)}$.
$w_{(i,j)}$: weight of $a_{(i,j)}$.
$d_{(i,j)}$: due date of $a_{(i,j)}$.
f: total number of families.
n_i: total number of jobs in family i.
n: total number of jobs.
M: a big number.

Decision variables

$$\delta_{(i,j)(k,l)} = \begin{cases} 1 \text{ if } a_{(i,j)} \text{ is scheduled before} a_{(k,l)}. \\ 0 \text{ otherwise.} \end{cases}$$

$$x^i_{j'j} = \begin{cases} 1 \text{ if } a_{(i,j')} \text{ is } \textit{scheduled immediately before } a_{(i,j)}. \\ 0 \text{ otherwise.} \end{cases}$$

$$y_{(i,j)} = \begin{cases} 1 \text{ if a setup occurs immediately before } a_{(i,j)} \\ 0 \text{ otherwise.} \end{cases}$$

$$z_{(i,j)(k,l)} = \begin{cases} 1 \text{ if } a_{(i,j)} \text{ is } \textit{scheduled before } a_{(k,l)} \textit{ and } \text{a setup} \\ \quad s_i \text{ occurs immediately before } a_{(i,j)}. \\ 0 \text{ otherwise.} \end{cases}$$

$C_{(i,j)}$: completion time of $a_{(i,j)}$.

2.1. FLO formulation

The linear ordering formulation is to solve a single machine scheduling problem without setups. The problem studied here is a family scheduling problem with setups, and thus a family linear ordering formulation, FLO, is proposed. In the FLO formulation, jobs in each family are arbitrarily numbered. The constraints of FLO are:

$$\delta_{(i,j)(k,l)} + \delta_{(k,l)\,(i,j)} = 1 \;\; \forall (i,j), (k,l) \in N, \;\; (i,j) \neq (k,l) \tag{A1}$$

$$\delta_{(i,j)(k,l)} + \delta_{(k,l)\,(o,p)} + \delta_{(o,p)\,(i,j)} \le 2$$
$$\forall (i,j), (k,l), (o,p) \in N \quad \text{and} \quad (i,j) \neq (k,l) \neq (o,p) \tag{A2}$$

$$\left(1 - x^i_{j'j}\right) \le \left(\sum_{k=1}^{f} \sum_{l=1}^{n_k} \delta_{(k,l)(i,j)} - \sum_{o=1}^{f} \sum_{p=1}^{n_o} \delta_{(o,p)(i,j')} - 1 \right)$$
$$+ M_1\left(1 - \delta_{(i,j')(i,j)}\right) \quad \forall (i,j), \left(i,j'\right) \in N, j < j' \tag{A3}$$

$$\left(\sum_{k=1}^{f} \sum_{l=1}^{n_k} \delta_{(k,l)(i,j)} - \sum_{o=1}^{f} \sum_{p=1}^{n_o} \delta_{(o,p)(i,j')} - 1 \right)$$
$$\le M_2\left(1 - x^i_{j'j}\right) \quad \forall (i,j), \left(i,j'\right) \in N, j < j' \tag{A4}$$

$$x^i_{j'j} \le \delta_{(i,j')(i,j)} \quad \forall (i,j), \left(i,j'\right) \in N, j < j' \tag{A5}$$

$$y_{(i,j)} = 1 - \sum_{j'=1}^{n_i} x^i_{j'j} \quad \forall (i,j) \in N \tag{A6}$$

$$z_{(i,j)(k,l)} \ge \delta_{(i,j)(k,l)} + y_{(i,j)} - 1 \quad \forall (i,j), \; (k,l) \in N \tag{A7}$$

$$C_{(i,j)} = \sum_{k=1}^{f} \sum_{l=1}^{n_k} \left(p_{(k,l)} \delta_{(k,l)(i,j)} + s_k z_{(k,l)(i,j)} \right)$$
$$+ p_{(i,j)} + s_i y_{(i,j)} \quad \forall (i,j) \in N \tag{A8}$$

$$C_{(i,j)} \ge 0 \quad \forall (i,j) \in N \tag{A9}$$

$$\delta_{(i,j)(i,j)} = 0 \quad \forall (i,j) \in N \tag{A10}$$