Computers and Mathematics with Applications 63 (2012) 110–117

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

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

An improved heuristic for parallel machine weighted flowtime scheduling with family set-up times

Ching-Jong Liao *, Chien-Wen Chao, Liang-Chuan Chen

Department of Industrial Management, National Taiwan University of Science and Technology, Taipei 106, Taiwan

ARTICLE INFO

Article history:
Received 10 February 2011
Received in revised form 30 October 2011
Accepted 31 October 2011

Keywords:
Parallel machines
Family set-up times
Weighted flowtime
Heuristic

ABSTRACT

This paper studies the identical parallel machine scheduling problem with family set-up times and an objective of minimizing total weighted completion time (weighted flowtime). The family set-up time is incurred whenever there is a switch of processing from a job in one family to a job in another family. A heuristic is proposed in this paper for the problem. Computational results show that the proposed heuristic outperforms an existing heuristic, especially for large-sized problems, in terms of both solution quality and computation times. The improvement of solution quality is as high as 4.753% for six-machine problem and 7.822% for nine-machine problem, while the proposed heuristic runs three times faster than the existing one.

1. Introduction

A production system with identical parallel machines is a common machine environment in the real world because a single machine usually cannot achieve the desired capacity, cost and/or revenue. On the other hand, family set-up times are incurred quite often in real production systems because jobs are often grouped into family for processing to improve the production efficiency. In this paper, the above two important scheduling elements in the real world are combined by investigating a parallel machine scheduling problem with family set-up times. The chosen objective is minimizing the total weighted completion time (often referred to in the literature as the weighted flowtime), which is a measure of work-in-process inventory. The weights may represent the actual cost of keeping different jobs in the system. Following the three-field notation [1], the considered problem can be denoted as $Pm|s_i|\sum wC$, where Pm represents the m identical parallel machines, s_i represents the sequence-independent set-up time for family i, and $\sum wC$ denotes the total weighted completion time.

In what follows, the literature related to the single and parallel machine problems with set-ups is reviewed. The $1|s_i|\sum C$ problem has been proved to be an NP-hard problem [2]. Gupta [3] proposed a heuristic for the problem but with sequence-dependent set-up times. Williams and Wirth [4] developed a polynomial-time heuristic for the problem and showed that it is quite effective in finding a good solution to even large problems in acceptable time limits. Liao and Liao [5] proposed a tabu search for the problem where a major set-up time is required when processing is switched from one family to another, while a minor set-up time is necessary when it is switched from one class to another.

To solve the weighted problem $1|s_i|\sum wC$, Ghosh [6] developed a dynamic programming (DP) algorithm while Dunstall et al. [7] developed lower bounds and incorporated into a branch and bound (BAB) algorithm which is efficient solving

* Correspondence to: Department of Industrial Management, National Taiwan University of Science and Technology, 43 Keelung Road, Section 4, Taipei 106, Taiwan.

E-mail address: cjliao@mail.ntust.edu.tw (C.-J. Liao).

0898-1221/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2011.10.077

problems with up to 70 jobs. On a large set of test problems, several metaheuristics were developed by Crauwels et al. [8]. The best results were obtained with the tabu search method for smaller numbers of families and with the genetic algorithm for larger numbers of families. Wang et al. [9] considered the problem with exponential time-dependent learning effect and proved that the problem can be solved in polynomial time under certain conditions.

Most of the literature on parallel machine scheduling focuses on the objective of minimizing the makespan. Even without set-up times, the $Pm \parallel C_{\max}$ problem is NP-hard. Williams [10] used SPT list-scheduling of families to machines followed by heuristic sequencing at each machine for the $Pm|s_i|C_{\max}$ and $Pm|s_i|\sum C$ problems. Webster [11] established that $Pm|s_i|\sum C$ is a strongly NP-hard problem. For the same problem, Yi and Wang [12] proposed a tabu search, while Yi and Wang [13] presented a lower bound.

Extension to the weighted version, Bruno et al. [14] established the NP-hardness of the $P2 \parallel \sum wC$ problem. Barnes and Laguna [15] showed that the Shortest Weighted Processing Time (SWPT) list-scheduling is a simple and reliable method for generating near-optimal solutions for the $Pm \parallel \sum wC$ problem. For the $Pm|s_i|\sum wC$ problem, a backward and a forward DP algorithm were proposed by Webster and Azizoglu [16]. When the numbers of machines and families are fixed, the backward DP is polynomial in the sum of the weights and the forward DP is polynomial in the sum of processing and set-up times. Therefore, the backward DP is more attractive when the sum of processing and set-up times is greater than the sum of the weights. In addition to the DP approach, there exist some BAB algorithms for the $Pm|s_i|\sum wC$ problem. Azizoglu and Webster [17] presented a BAB for the problem and generated optimal solutions with up to 15–25 jobs, depending on the number of machines. Chen and Powell [18] proposed column generation based BAB which can solve problems with up to 40 jobs, 4 machines, and 6 families. Dunstall and Wirth [19] presented another BAB with up to 25 jobs and 8 families. As indicated by Allahverdi et al. [20], the above two BAB algorithms remain to be compared. Dunstall and Wirth [21] also proposed several heuristics for the problem and evaluated performance of the heuristics relative to lower bounds and optimal solutions. In this paper, we continue the research by developing an improved heuristic for the problem and evaluating its performance relative to the heuristics of Dunstall and Wirth [21].

2. Problem formulation

Denote by (G, N, M) an instance set for the $Pm|s_i|\sum wC$ problem where G is the number of families, N is the total number of jobs, and M is the number of machines. A processing time $p_{i[j]}$ and a positive weight $w_{i[j]}$ are assigned into the jth job of family i. The set-up time for family i is denoted by s_i, which is sequence-independent. A family set-up time is sequence-independent if its duration depends only on the family of the current batch to be processed, and is sequence-dependent if its duration depends on the families of both the current and the immediately preceding batches.

The problem is considered under the following assumptions.

- An initial set-up time for each machine is required.
- A sequence-independent family set-up time is incurred whenever there is a switch of processing from a job in one family to a job in another family.
- There are m identical machines in parallel. A job may be processed on any one of the m machines.
- All machines are available to process jobs at time zero.
- No machine may process more than one job at a time.
- All jobs are ready to be processed at time zero.
- Preemptions are not allowed.

In the literature, Chen and Powell [18] developed a set partitioning type formulation for the considered problem. They showed that the problem is equivalent to a network problem and used the network structure to derive a set partitioning type formulation.

According to Chen and Powell [18], a directed network $G = (\mathbf{N}, \mathbf{A})$ is constructed as follows. The node set consists of $n + 2$ nodes $\mathbf{N} = \{0, 1, \dots, n, n + 1\}$, where 0 is a source node, $n + 1$ is a sink node and $1, 2, \dots, n$ is other nodes, called job nodes. Each job node j corresponds to job j. The arc set $\mathbf{A}$ consists of one arc from the source node to each job node, one arc from each job node to the sink node, and one arc from each job node j to each of the nodes. A directed path ω is from the source to the sink, denoted as $\omega = \{0, j_1, j_2, \dots, n + 1\}$, where $\{j_1, j_2, \dots\} \in \mathbf{N} \setminus \{0, n + 1\}$. If there is no cycle in the path, then this path ω is an acyclic path; otherwise, ω is a cyclic path. Let $a_{j\omega}$ be the number of times node j is visited by path ω. In this network problem, find m or less feasible acyclic paths from the source to the sink in the network G such that each job node is visited exactly once and the total cost of these paths is minimum.

Let Ω_a and Ω_c be the set of all feasible acyclic paths and the set of all feasible cyclic paths, respectively, from the source to the sink. Given any path $\omega \in \Omega_a \cup \Omega_c$, the cost of the path, c_ϖ is known. Define a binary variable x_ω for each $\omega \in \Omega_a$ as 1 if path ω is selected and 0 otherwise. Then, an integer programming formulation of a set partitioning type can be modeled as follows:

$$\min \sum_{\omega \in \Omega_a} c_\omega x_\omega \tag{1}$$

$$\text{s.t.} \sum_{\omega \in \Omega_a} a_{j\omega} x_\omega = 1, \quad \forall j, \tag{2}$$