Computers & Industrial Engineering 72 (2014) 274–281

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Minimizing total tardiness for scheduling identical parallel machines with family setups

Jeffrey E. Schaller *

Eastern Connecticut State University, Department of Business Administration, 83 Windham ST., Willimantic, CT 06250, United States

ARTICLE INFO

Article history:
Received 22 August 2013
Received in revised form 6 February 2014
Accepted 2 April 2014
Available online 13 April 2014

Keywords:
Scheduling
Batch sizing
Heuristics
Parallel machines
Sequencing

ABSTRACT

This paper presents several procedures for scheduling identical parallel machines with family setups when the objective is to minimize total tardiness. These procedures are tested on several problem sets with varying numbers of families, jobs and machines, varying setup time distributions and various levels of due date tightness and variability. The results show that genetic algorithms are the most effective algorithms for the problem.

1. Introduction

In many operations the existence of changeover times, or setup times, on a machine motivates the grouping of jobs in order to obtain economies of scale. In scheduling, efficiencies that lead to economies of scale are gained by grouping similar jobs together. For example, jobs may belong to families where the jobs in each family tend to be similar in some way, such as their required tooling. As a result of this similarity, a job does not need a setup when following another job from the same family, but a known "family setup time" is required when a job follows a job that is a member of some other family. This is called a family scheduling model. Typically, there are a large number of jobs, but a relatively small number of families.

When a set of jobs is to be scheduled and need to be processed by a type of machine an important consideration is completing each job on or before the customer's due date. To help complete jobs in a timely manner there may be two or more identical machines of a type. These machines are referred to as identical parallel machines and the scheduling of jobs on the machines is referred to as parallel machine scheduling. To address this consideration, this paper seeks to identify methods for assigning and sequencing a set of jobs on identical parallel machines with significant family setup times that will minimize the total tardiness of the jobs. The tardiness of a job is defined as the completion time of the job minus the due date for the job if the job is completed after the due date and the tardiness is equal to zero if the job is completed on or before the due date. When jobs are scheduled in this environment, one approach is to sequence jobs in the same family as a batch on one of the machines (jobs in the same family are sequenced next to each other) to reduce setup time. The batching of jobs could cause some jobs to be processed before they are needed while at the same time delaying other jobs and causing them to be tardy. A second approach is to stop processing jobs in one family so a job in another family can be processed and completed by its due date. This causes an additional setup to be required and this additional setup increases the overall time to complete jobs that could have the effect of causing jobs that are at the end of the sequence on a machine to be very tardy. The trade-off just described causes this sequencing problem to be very challenging and as yet the only research that has been published for sequencing and scheduling in the described environment with an objective to minimize total tardiness was done by Shin and Leon (2004).

Formally, suppose there is a set of n jobs belonging to F setup families to be processed on M identical machines. Let p_j, $f(j)$, S_j, C_j, and d_j represent the processing time, the setup family, the setup time, the completion time, and the due date of job j ($j = 1, \ldots, n$) respectively. The tardiness of job j, T_j is defined as: $T_j = \max\{C_j - d_j, 0\}$, for $j = 1, \ldots, n$. The objective function, Z, can be expressed as: $Z = \sum_{j=1}^{n} T_j$. Also, note that if the job to be sequenced in position j of machine $m (m = 1, \ldots, M)$ is denoted as $[j]m$ then $C_{[j]m} = C_{[j-1]m} + p_{[j]}$ if $f[j]m = f\,[j-1]m$ and $C_{[j]m} = C_{[j-1]m} + S_{[j]} + p_{[j]}$ if $f\,[j]m \neq f\,[j-1]m$. This notation is summarized in Table 1.

* Tel.: +1 860 465 5226.
E-mail address: schallerj@easternct.edu

http://dx.doi.org/10.1016/j.cie.2014.04.001
0360-8352/

Five papers have addressed the problem of scheduling jobs on parallel machines to minimize total tardiness without family setups using branch-and-bound algorithms. Azizoglu and Kirca (1998) were the first to develop an optimal branch-and-bound algorithm. They developed several dominance properties and a lower bound that includes jobs that are not yet in a partial schedule. Their lower bound is based on a relaxed problem that allows jobs to be simultaneously processed on more than one machine. Yalaoui and Chu (2002) developed a branch-and-bound algorithm that included additional dominance properties and a new lower bound. Shim and Kim (2007) developed a branch-and-bound algorithm that included additional dominance properties and a lower bound that improves the lower bound developed by Azizoglu and Kirca (1998). Schaller (2009) shows how the lower bounds developed by Shim and Kim (2007) can be improved. Tanaka and Araki (2008) incorporated a Lagrangian relaxation into a branch-and-bound procedure for the problem. Sen, Sulek, and Dileepan (2003) include scheduling heuristics for the parallel machine problem to minimize total tardiness in their survey. More recently Biskup, Herrmann, and Gupta (2008) developed an insertion based heuristic for the problem.

Scheduling parallel machines when family setup times exist has been addressed for a variety of objectives such as minimizing flowtime (Azizoglu and Webster) and minimizing tardiness. Shin and Leon (2004) address the problem of minimizing total tardiness on parallel machines when family setup times exist. They developed a two-phase solution procedure. The first phase uses a MULTIFIT method based on the method developed by Coffman, Garey, and Johnson (1978) to develop an initial solution. The initial solution assigns jobs to machines and creates a sequence of jobs on the machines that can be defined in terms of batches of jobs from the same family. The second phase uses a tabu search to create an improved solution. Eom, Shin, Kwun, Shim, and Kim (2002) developed an efficient heuristic for minimizing weighted tardiness for parallel machine scheduling with sequence-dependent family setup times. Shin and Kang (2010) and Kang and Shin (2010) developed heuristic procedures that include total tardiness as an objective for parallel machine scheduling with rework and sequence-dependent setups. Chen and Chen (2008) use bottleneck-based heuristics to minimize the number of tardy jobs in a hybrid flexible flow line with unrelated parallel machines and Behnamian et al. (2009) developed a genetic algorithm for scheduling a hybrid flowshop with sequence-dependent setups and an objective that includes total tardiness.

Schutten (1996) showed that list schedules are dominant to minimize any regular cost function for parallel machine scheduling with sequence dependent setups if each job is assigned to the machine that it will complete the earliest instead of start the earliest. In this paper Schutten's (1996) property is used to develop tabu searches and genetic algorithms based on list schedules. The tabu searches, including the one developed by Shin and Leon (2004), are described in section two. In section three the genetic algorithms are described. A branch-and-bound algorithm to obtain optimal solutions for the problem is described in section four. The procedures were tested on randomly generated problems and the results of these tests are presented in section five. Section six concludes the paper.

Table 1
Summary of notation used throughout the paper.

Indexes	
j	Part index ($j = 1, \ldots, n$)
m	Machine index ($m = 1, \ldots, M$)
f	Setup family index ($f = 1, \ldots, F$)
Parameters	
n	Total number of parts
M	Total number of machines
F	Total number of families
p_j	Processing time of job j
$f(j)$	Setup family job j belongs to
S_j	Setup time for job j
d_j	Due date for job j
Decision variables	
C_j	Completion time of job j
T_j	Tardiness of job j
$[j]m$	Job sequenced in the jth position of machine m
$C_{[j]m}$	Completion time of the job sequenced in the jth position of machine m

2. Tabu searches

In this section four tabu searches are described for the problem. An initial solution is needed to start the tabu search procedures. Procedures for generating initial solutions are described in the next subsection.

2.1. Initial solution

The algorithm to develop an initial solution presented in this section was developed by Shin and Leon (2004). This algorithm uses a MULTIFIT method based on the method developed by Coffman et al. (1978) and is referred to as the GM algorithm. The algorithm schedules jobs within a family in EDD order and attempts to allocate families to machines so total tardiness is minimized. The GM algorithm is described in the appendix.

In this paper a modified version of the GM algorithm is proposed. This version is referred to as GM′. The GM′ algorithm is the same as the GM algorithm except when scheduling the jobs for a family on a machine. In the original GM version jobs belonging to the same batch are sequenced in EDD order. While EDD order would be appropriate in situations where due dates are relatively loose (most jobs are early) it would not be good if due dates are tight (most jobs are tardy). In fact if due dates are very tight and all of the jobs belonging to a batch would be tardy then SPT order would be appropriate. In order to address this issue the revised algorithm initially sequences jobs belonging to a batch in EDD order and then checks a condition developed by Emmons (1969) to identify exchanges of jobs that belong to the same family that could reduce tardiness. If two jobs j and k belong to the same family and are assigned to the same machine m and job k is currently sequenced before job j then the Emmon's condition is checked: if $p_j < p_k$ and $d_j < \max\{d_k, C_k\}$ then the positions of the two jobs on machine m are swapped. The following lemma states that if jobs j and k satisfy the above conditions then job j should precede job k. The following notation is used in the lemma. Let $\sigma(m)$ be a sequence of jobs on machine m in which job k precedes job j and $\sigma'(m)$ be a sequence of jobs on machine m that is the same as the sequence $\sigma(m)$ with the exception that the positions of jobs k and j are swapped. Let C_j and C_k be the completion times of jobs j and k in sequence $\sigma(m)$ and C'_j and C'_k be the completion times of jobs j and k in sequence $\sigma'(m)$.

Lemma. *If two jobs j and k belong to the same family and are to be assigned to the same machine m if $p_j < p_k$ and $d_j < \max\{d_k, C_k\}$ then the job j precedes job k in at least one optimal sequence.*

Proof. Let T_j and T_k be the tardiness of jobs j and k in sequence $\sigma(m)$ and T'_j and T'_k be the tardiness of jobs j and k in sequence $\sigma'(m)$. Let $B_{kj}(m)$ be the set of jobs that are sequenced before job k in $\sigma(m)$, $A_{kj}(m)$ be the set of jobs that are sequenced after job j in $\sigma(m)$, and $Q_{kj}(m)$ be the set of jobs that are sequenced between jobs k and j in $\sigma(m)$. Since jobs j and k belong to the same family, the total tardiness of the jobs in the sets $B_{kj}(m)$ and $A_{kj}(m)$ will