



An improved ant colony optimization for scheduling identical parallel batching machines with arbitrary job sizes

Bayi Cheng^{a,b,*}, Qi Wang^a, Shanlin Yang^{a,b}, Xiaoxuan Hu^{a,b}

^a School of Management, Hefei University of Technology, Hefei 230009, PR China

^b Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education, Hefei University of Technology, Hefei 230009, PR China

ARTICLE INFO

Article history:

Received 11 February 2012

Received in revised form 20 July 2012

Accepted 26 October 2012

Available online 12 November 2012

Keywords:

Scheduling

Batching machines, Arbitrary job sizes

Ant colony optimization

ABSTRACT

In this paper we consider the problem of scheduling parallel batching machines with jobs of arbitrary sizes. The machines have identical capacity of size and processing velocity. The jobs are processed in batches given that the total size of jobs in a batch cannot exceed the machine capacity. Once a batch starts processing, no interruption is allowed until all the jobs are completed. First we present a mixed integer programming model of the problem. We show the computational complexity of the problem and optimality properties. Then we propose a novel ant colony optimization method where the Metropolis Criterion is used to select the paths of ants to overcome the immature convergence. Finally, we generate different scales of instances to test the performance. The computational results show the effectiveness of the algorithm, especially for large-scale instances.

Crown Copyright © 2012 Published by Elsevier B.V. All rights reserved.

1. Introduction

The problem of scheduling batching machines with arbitrary job sizes is often encountered in real industries. The typical applications are semiconductor manufacturing, food processing, metal working and ship scheduling at navigation locks. In the burn-in operation of semiconductor manufacturing, the chips have different sizes and processing times which are both required by the customers. The actual processing time of a chip can be longer than or equal to the required time, however, it cannot be shorter than the required time. The chips are processed in *batches* where a *batch* means a subset of chips. A burn-in oven has a limited capacity, i.e., the total size of jobs in a batch cannot exceed the capacity. During the burn-in operation, the oven is heated to a high temperature. Therefore, the operation cannot be interrupted until all the chips in a batch are completed. After the burn-in operation, the defective ones can be found and eliminated to guarantee the quality of products. Such applications are also encountered in food processing, metal working industries and ship scheduling at navigation locks.

In the related research, the scheduling problem of batching machines was proposed by Lee et al. [1]. They investigated the scheduling where a batching machine can process a fixed number of jobs and showed that the problem of minimizing makespan is NP-hard in the strong sense. Janiak et al. [2,3] and Bellanger et al. [4] proposed algorithm for minimizing total completion time. Then

the scheduling of a single batching machine with agreeable release times and due dates was proposed from the semiconductor manufacturing systems [5]. Then Jolai [6] investigated several algorithms for the single machine problem. Ji and Cheng [7] and Li et al. [8] considered the scheduling of batching machines with deteriorating jobs and release dates and provided polynomial time algorithms. Janiak et al. [9,10] considered the single-machine problems with resource dependent times and limited deterioration of reworkables. The two-machine shop problems were studied in Potts et al. [11] where the machines can process a bounded or unbounded number of jobs.

On the other hand, the problem of scheduling batching machines with arbitrary job sizes was introduced by Uzsoy [12]. The single-machine problem was investigated and heuristic methods were proposed to minimize makespan and total completion times respectively. Then the performances of the proposed heuristics were analysed by Zhang et al. [13], where an improved 7/4-approximation algorithm was proposed for minimizing makespan. Dupont and Flipo [14] introduced a branch and bound method for the problem. Kashan et al. [15] considered a general version of the single-machine problem, in which the processing times of the jobs with sizes greater than $1/m$ is not less than the other ones, where $m \geq 2$ and is an integer. They proposed an algorithm with absolute worst-case ratio of $3/2$ and asymptotic worst-case ratio of $(1+1/m)$. Cheng et al. [16] studied the fuzzy uncertainty in the real manufacturing and proposed a fuzzy model for single-machine problem. The single-machine problem with release time was studied in Ref. [17] and a $(2+\varepsilon)$ -approximation algorithm was designed, where $\varepsilon > 0$ and can be

* Corresponding author.

E-mail address: chengbayi@gmail.com (B. Cheng).

arbitrarily small. In recent years, meta-heuristic algorithms have been introduced to solve single-machine problems. Sevaux and Peres [18] applied genetic algorithm to minimize the weighted number of late jobs. Kashan et al. [19] and Damodaran et al. [20] modified the coding and decoding methods to improve the quality of the solutions. Other meta-heuristic algorithms have also been applied in the problem like simulated annealing [21] and branch and price algorithm [22]. In the research of multi-machine problems, Jula and Leachman [23] considered the coordination problem in supply chain where the production facility has batching machines.

Ant colony optimization (ACO) is a typical meta-heuristic algorithm inspired by the real ants' behaviour of searching food in nature. The algorithm was introduced by Dorigo et al. [24]. For it is simple to implement and it has a good performance in optimization, ACO has been widely used in combinatorial optimization problems, such as structural topology optimization [25], shortest path problem [26], Bayesian networks [27] and scheduling problems [28]. The readers can refer to Ref. [29] for a brief survey of the application of ACO. However, the determinate selection of paths for ants tends to immature convergence [30]. In the implementation of ACO, the paths of ants are generated according to two factors: the pheromone value of paths and the heuristic information of paths. The pheromone on a path is strengthened if it is selected by an ant and consequently, the behaviour of ants in the earlier iterations influences the ones in the later iterations. When the earlier ants concentrate on locally optimal paths, the pheromone on these paths will be reinforced repeatedly, which has a negative impact on the final solutions.

In order to overcome the disadvantage, we use the stochastic selection mechanism of the Metropolis Criterion and propose a novel ACO in this paper. In the early iterations, the restriction is relaxed for the paths. More paths are selected as elite ones. In the later evolution of the algorithm, the restriction is tightened gradually and is controlled by the Metropolis Criterion. Then the effectiveness of the algorithm is tested by instances in the experiments. The remainder of the paper is organized as follows. In Section 2, we describe the problem and present a mathematical model. In Section 3, we show the properties of optimal solutions and feasible solutions. In Section 4, we propose our algorithm and show the detailed implementation. In Section 5, we show the design of instances and report the computational results. Finally in Section 6, we conclude this paper and give future directions.

2. Problem statement

The problem under investigation can be described as follows. There are n jobs to be processed and the job set is $J = \{1, 2, \dots, n\}$. There are m parallel machines and the machine set is $M = \{M_1, M_2, \dots, M_m\}$. Each job needs only one operation on an arbitrary machine. The size of job j is s_j and the processing time of j is t_j . The actual processing time of j should be not less than t_j . The machines are identical and their size capacity is B , that is, once a batch of jobs are assigned to be processed, the total size should be no larger than B . The number of batches assigned on M_i is K_i and $K = \sum_{i=1}^m K_i$. The k th batch processed on M_i is b_{ik} . The processing time and completion time of b_{ik} are respectively T_{ik} and C_{ik} . The processing of b_{ik} is denoted by O_{ik} and $O_{ih} < O_{il}$ means that b_{ih} is processed before b_{il} . Using the general notation of scheduling problems [31], the problem under investigation can be denoted as $Pm|B, s_j, batch|C_{max}$. In the notation, Pm indicates that there are m identical parallel machines; B represents the machine capacity and s_j means the jobs have arbitrary sizes. *Batch* indicates that the jobs are processed in batches without pre-emption and C_{max} implies that the objective function is makespan.

Now set 0–1 decision variables λ_{ik} , μ_{ijk} and ξ_{ihl} as:

$$\lambda_{ik} = \begin{cases} 1 & \text{if } b_{ik} \text{ is created for } M_i; \\ 0 & \text{otherwise} \end{cases};$$

$$\mu_{ijk} = \begin{cases} 1 & \text{if } j \in b_{ik} \\ 0 & \text{otherwise} \end{cases};$$

$$\xi_{ihl} = \begin{cases} 1 & \text{if } O_{ih} < O_{il}; h, l = 1, \dots, K_i \\ 0 & \text{otherwise} \end{cases}.$$

Define a positive integer $X \rightarrow +\infty$. $Pm|B, s_j, batch|C_{max}$ can be described as the following mixed integer programme (MIP) model.

Minimize

$$C_{max} = \max_{i=1, \dots, m} \max_{k=1, \dots, K_i} \{C_{ik}\} \quad (1)$$

Subject to

$$K_i = \sum_{k=1}^n \lambda_{ik}, \quad i = 1, \dots, m \quad (2)$$

$$\mu_{ijk} \leq \lambda_{ik}, \quad i = 1, \dots, m; j = 1, \dots, n; k = 1, \dots, n \quad (3)$$

$$\sum_{i=1}^m \sum_{k=1}^n \mu_{ijk} = 1, \quad j = 1, \dots, n \quad (4)$$

$$\sum_{j=1}^n \mu_{ijk} s_j \leq B, \quad i = 1, \dots, m; k = 1, \dots, n \quad (5)$$

$$T_{ik} = \max\{t_j : \mu_{ijk} = 1\}, \quad i = 1, \dots, m; j = 1, \dots, n \quad (6)$$

$$C_{il} - T_{ih} + X(1 - \xi_{ihl}) \geq C_{ih}, \quad i = 1, \dots, m; h, l = 1, \dots, K_i \quad (7)$$

$$\lambda_{ik}, \mu_{ijk}, \xi_{ihl} \in \{0, 1\} \quad \forall i, j, k, h, l \quad (8)$$

Objective (1) means the makespan is the completion time of the latest completed batch. Constraint (2) generates the number of batches on the machines. Constraint (3) ensures that jobs can be assigned into a batch only when the batch is created and (4) ensures each job should be processed only one time. Constraint (5) ensures that the total size of jobs in a batch cannot exceed the machine capacity and (6) means the processing of a batch is non-preemptive. Constraint (7) ensures that a machine can process at most one batch simultaneously. Constraint (8) imposes binary restrictions on the decision variables.

3. Properties on solutions

Note that $1|B, s_j, batch|C_{max}$ is NP-hard in the strong sense and it is a special case of the parallel machine problem. Obviously we have the following conclusion.

Property 1. $Pm|B, s_j, batch|C_{max}$ is NP-hard in the strong sense.

Let π denotes a feasible solution of $Pm|B, s_j, batch|C_{max}$. Note that the generation of a feasible solution consists of two steps. The first is to assign jobs into feasible batches and the second is to order the batches on machines. Since in the special case where $n \leq m$, the jobs can be processed in arbitrary order and the makespan is always $C_{max} = \max_{j=1, \dots, n} \{t_j\}$. Consequently, we have the following property on the scale of feasible solutions.

Property 2. Let L_f and U_f represent the lower and upper bound of the number of feasible solutions respectively. Then $L_f = n!/m!$ and $U_f = n^n$.

Proof. In contrast to classical scheduling, the complexity of $Pm|B, s_j, batch|C_{max}$ is caused by the large number of combinations of jobs when they are assigned into batches. If each batch can accommodate at most one job, i.e., no combination exists, then the problem

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات