Applied Mathematics and Computation 334 (2018) 254–268

Contents lists available at ScienceDirect

Applied Mathematics and Computation

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

A random-keys genetic algorithm for scheduling unrelated parallel batch processing machines with different capacities and arbitrary job sizes

Shengchao Zhou [a], Jianhui Xie [b,*], Ni Du [c], Yan Pang [a]

[a] School of Logistics, Central South University of Forestry and Technology, Changsha, Hunan 410004, China
[b] Lingnan College, Sun Yat-sen University, Guangzhou, Guangdong 510275 China
[c] School of Engineering, Hunan Agricultural University, Changsha, Hunan 410128, China

ARTICLE INFO

Keywords:
Scheduling
Unrelated parallel machines
Batch processing machines
Random-keys genetic algorithm

ABSTRACT

A batch processing machine (BPM) can simultaneously process several jobs and has wide applications in various industrial environments. This paper studies the problem of minimizing makespan on unrelated parallel BPMs with non-identical job sizes and arbitrary release times. In the environment of unrelated machines, each machine has a processing speed for each job. The unrelated BPM problem is the most general case of parallel BPM problems and is closer to actual production conditions. The problem under study is NP-hard. We present two lower bounds for the problem. Then a genetic algorithm based on random-keys encoding is proposed to solve the problem. The performance of the proposed algorithm is compared with a commercial solver (ILOG CPLEX) and two meta-heuristics published in the literature: a recent iterated greedy algorithm and a particle swarm optimization algorithm. Computational experiments show that the proposed algorithm produces better solutions compared to the other methods. The quality of the proposed lower bounds is evaluated as well.

1. Introduction

Batch processing machines (BPMs) are widely used in many industrial systems such as electronics manufacturing facilities [27] and steel-making plants [38]. A BPM can process a set of jobs simultaneously as a batch. Once processing of a batch starts on a BPM, it cannot be interrupted and no job can be added into the machine until processing is finished. The jobs in a batch start and complete processing at the same time. The processing time of a batch is given by the longest processing time of the jobs in the batch. The purpose of BPM scheduling problems is to group jobs into batches and scheduling the batches on the BPMs such that one or more time-based objectives are minimized.

A typical BPM scheduling problem occurs in the burn-in operation for final testing in semiconductor manufacturing. In the operation, burn-in ovens are used to test integrated circuits for a prolonged period of time at a high temperature in order to detect weak or fragile circuits. An oven is able to process a number of integrated circuits simultaneously. Because the processing time of the burn-in operation is much longer compared to other testing operations [27], the burn-in operation is usually a bottleneck in semiconductor manufacturing. Effective scheduling of the burn-in operations is very necessary to

* Corresponding author.
E-mail addresses: zhousc@mail.ustc.edu.cn (S. Zhou), xiejh23@mail.sysu.edu.cn (J. Xie), duni1204@163.com (N. Du).

https://doi.org/10.1016/j.amc.2018.04.024
0096-3003/

enhance the utilization of the ovens and throughput. The oven utilization can be enhanced by optimizing the makespan criterion.

In recent years, single or parallel BPM scheduling problems have been examined by many researchers [33]. When scheduling jobs on parallel BPMs, three interdependent decisions have to be made: grouping the jobs into batches, assigning the batches to the BPMs and sequencing the batches on each BPM. Parallel BPM scheduling problems can be classified into identical, uniform and unrelated parallel BPM problems. The unrelated BPM problem is the most general parallel BPM problem among them and is closer to actual production conditions. In the problem of unrelated machines, each machine has a processing speed for each job.

In this paper, we consider the unrelated parallel BPM scheduling problem with non-identical job sizes. The BPMs have different capacities. Each BPM can process a batch of jobs at a time as long as the total size of the jobs in the batch does not exceed its capacity. The jobs arrive at the production system dynamically instead of becoming available at time zero. A batch cannot be processed until all the jobs in the batch have arrived. The objective is to minimize the makespan (the largest completion time of the jobs) which reflects the machine utilization. This problem is NP-hard because its special case, the problem of minimizing makespan on a single BPM with non-identical job sizes is NP-hard [39]. Consequently, a hybrid random-keys genetic algorithm is proposed to solve this problem. Two lower bounds are also presented in this paper.

The rest of the paper is organized as follows. Section 2 presents the literature review of BPM scheduling problems and applications of genetic algorithms for scheduling problems. Section 3 describes the problem under study formally. Section 4 presents lower bounds for the problem. Section 5 introduces details of the proposed genetic algorithm. Experimental results are given in Section 6. Section 7 presents some conclusions and future research directions.

2. Literature review

2.1. Batch processing machine scheduling

The problems related to BPM scheduling have received a lot of attention in the literature. At the start, researchers mainly focused on the scheduling problems of single BPM with unit job sizes. [17] firstly considered the single BPM scheduling problem. Assuming that job processing times are constant, job sizes are identical, and release times and due dates are agreeable, they proposed a polynomial time algorithm for minimizing makespan. Lee et al. [27] studied the BPM problem with equal job processing times and agreeable release times and due dates. Several dynamic programming algorithms were presented to minimize maximum tardiness and minimize the number of tardy jobs. Li and Lee [28] proved that the problem of scheduling a single BPM with agreeable release times and due dates is strongly NP-hard when the objective is to minimize maximum tardiness or to minimize the number of tardy jobs. They provided dynamic programming algorithms when job processing times are also agreeable with release times and due dates. Brucker et al. [4] derived several dynamic programming algorithms for minimizing regular scheduling criteria that are non-decreasing in the job completion times on a BPM without job release times.

Some researchers addressed the single BPM problems in the presence of arbitrary job release times, but identical job sizes [26,40]. Sung and Choung [36] investigated the problem of minimizing makespan on a single BPM with different release times, for which a branch and bound algorithm and several heuristics were exploited. Sung et al. [37] constructed a dynamic programming algorithm for minimizing makespan on a BPM with job families and dynamic job arrivals. Deng et al. [15] analyzed the problem of scheduling jobs on a BPM with release times to minimize total completion time with a polynomial time approximation scheme.

Some works considered the single BPM problems with non-identical job sizes and equal release times [7,13,18,34]. Uzsoy [39] proved that minimizing makespan and minimizing total completion time on a BPM with non-identical job sizes are NP-hard. Several heuristics and a branch and bound algorithm were developed. Jolai Ghazvini and Dupont [22] provided several heuristics to minimize mean flow times. To minimize makespan, Zhang et al. [45] presented an approximation algorithm with worst-case ratio 7/4 and proved that the worst-case ratio of algorithm FFLPT given by Uzsoy [39] is 2. Dupont and Dhaenens-Flipo [16] proposed a branch and bound method for the BPM problem of minimizing makespan. Kashan and Karimi [23] considered the problem of minimizing total weighted completion time with incompatible job families and proposed an ant colony optimization (ACO) meta-heuristic.

Several studies dealt with the single BPM problems considering non-identical job sizes and unequal release times. Li et al. [29] proposed an approximation algorithm for the problem with makespan minimization. Chou et al. [10] introduced two versions of a genetic algorithm (GA) to solve the single BPM scheduling problem and Chou [9] developed an approach by combining dynamic programming and GA. Zhou et al. [48] proposed a lower bound and a number of heuristics to minimize makespan.

Most of the works mentioned above concentrated on the single BPM problem. Some researchers have examined the problem of scheduling identical parallel BPMs [8,19,21,47]. Chang et al.[5] presented a simulated annealing (SA) algorithm to minimize makespan on identical parallel BPMs. Chung et al. [11] addressed the parallel BPM problem with non-identical job sizes and arbitrary release times. A mixed integer programming model and several heuristics were proposed to minimize makespan. Chen et al.[6] presented two lower bounds and two meta-heuristics GA and ACO for the same problem and Velez-Gallego [14] provided a SA. Some authors considered the scheduling problem of parallel BPMs with non-identical capacities [12,20,44].