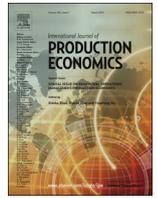


Contents lists available at [SciVerse ScienceDirect](http://www.sciencedirect.com)

Int. J. Production Economics

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

A bi-objective scheduling problem on batch machines via a Pareto-based ant colony system

Rui Xu^a, Huaping Chen^a, Xueping Li^{b,*}^a School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, PR China^b Department of Industrial and Systems Engineering, University of Tennessee, Knoxville, TN 37996-0700, USA

ARTICLE INFO

Article history:

Received 24 September 2012

Accepted 27 April 2013

Keywords:

Scheduling

Batch-processing machine

Makespan

Maximum tardiness

Multi-objective ant colony optimization

ABSTRACT

This research aims to minimize the bi-criteria of makespan and maximum tardiness on a set of identical batch-processing machines arranged in parallel. Each machine can process multiple jobs simultaneously as long as the machine capacity is not exceeded. Each job is defined by its processing time, ready time, due date, and size. The processing time and ready time of a batch are represented by the largest processing time and release time among all jobs in the batch, respectively. For this problem, a scheduling algorithm based on the framework of a multi-objective ant colony optimization (MOACO) approach called a Pareto-based ant colony system (PACS) was developed. Based on the constructive characteristics of PACS, a new mechanism of solution construction was introduced so that the proposed algorithm had the ability to explore the entire solution space. Moreover, corresponding to the new construction mechanism, a candidate list strategy and a form of dynamic heuristic information were developed to reduce the search space and direct the search toward the promising regions, respectively. Through extensive computational experiments with various problem instances, the effectiveness of the proposed algorithm was evaluated by measuring the computational efficiency and solution quality. The experiment results demonstrated that PACS had a superior performance compared to other benchmark algorithms, especially for large job instances.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Nowadays microprocessors, memory chips, and other semiconductor related devices are a part of daily life, appearing in electronics ranging from personal computers to cellular phones. The semiconductor industry has already become one of the largest industries in the world, and semiconductor manufacturers need to utilize their resources effectively to confront the huge demand and severe competition in the marketplace. Furthermore, there is increasing pressure for semiconductor manufacturers to improve their overall performance, such as machine utilization and on-time delivery. Effective scheduling is one of the most economical and executable potential solutions to such challenges.

Batch processing is a very common procedure in semiconductor manufacturing for the avoidance of setups and/or facilitation of material handling. A batch is defined as a group of jobs that have to be processed jointly (Brucker et al., 1998) and a batch scheduling problem consists of grouping jobs on each machine into

batches which are scheduled either in serial (called s-batching) or in parallel (called p-batching). On an s-batching machine, the processing time of a batch equals the sum of the processing times of all jobs that form the batch. On a p-batching machine, several jobs can be processed as a batch simultaneously, and the length of a batch is the largest processing time of its jobs.

In semiconductor manufacturing, p-batching is a much more important issue, occurring in many processes such as oxidation/deposition/diffusion of wafer fabs and the burn-in operation in test facilities (Moench et al., 2011). For instance, the burn-in operation of the final testing phase of semiconductor manufacturing tests the integrated circuits for defects by subjecting them to thermal stress for an extended period of time. In this way, latent defects may be discovered that would otherwise only be found in the operating environment. While very important, the operations' lengthy processing time compared with the other operations in semiconductor manufacturing creates a bottleneck in the final testing phase. Consequently efficient scheduling is of great concern to overall performance. In addition, p-batching is commonly observed in other modern industrial environments, such as in chemical, food and mineral processing, pharmaceutical, and metalworking industries as well as with wafer and environmental stress screening chamber fabrication.

This paper focuses on p-batching scheduling, which is also known as the scheduling of a batch-processing machine (BPM) or

* Corresponding author at: Department of Industrial and Systems Engineering, University of Tennessee, 408 East Stadium Hall, Knoxville, TN 37996-0700, USA. Tel.: +1 865 974 7648; fax: +1 865 974 0588.

E-mail addresses: ruixu@mail.usc.edu.cn (R. Xu), hpcchen@usc.edu.cn (H. Chen), Xueping.Li@utk.edu (X. Li).

batch machine in the literature (Mathirajan and Sivakumar, 2006; Moench et al., 2011). On a BPM, a fixed maximum batch size exists as the capacity of the batch machine, and a batch machine is capable of accommodating a group of jobs simultaneously as long as the sum of the job sizes in the batch is less than or equal to the machine capacity. The processing time of each batch is determined by the longest job processing time among the jobs contained in the batch. Finally, once a batch is being processed, it is non-preemptable during the long period of processing.

In this research, we consider the BPM problem of scheduling on identical batch machines arranged in parallel with dynamic job arrivals and non-identical job sizes to minimizing makespan (C_{max}) and maximum tardiness (T_{max}) simultaneously. Note that the target problem is complicated by the considerations of different constraints and multiple objectives. First, a machine environment involving multiple batch machines instead of one requires an additional decision about machine assignment to obtain a final solution. Secondly, dynamic job arrivals and non-identical job sizes, which reflect realistic conditions in most manufacturing systems, are considered. Lastly, two performance measures of C_{max} and T_{max} , which are among the most commonly used criteria in the batch scheduling research, are considered. Makespan is a measure of system utilization while maximum tardiness is a measure of performance in meeting customer due dates. This target problem can be denoted by $P_m|r_j, s_j, p\text{-batch}|C_{max}, T_{max}$, and the details of the problem statement and formulation will be given in Section 3.

To the best of our knowledge, the BPM scheduling problem for optimizing multi-objectives has not been fully investigated in the literature. Only Kashan et al. (2010) studied a NP-hard scheduling problem on a BPM while optimizing the same objectives considered in the target problem. However, they do not consider the constraints of dynamic job arrivals and multiple BPMs, which often arises from real-world production systems and occurs in a more general environment. Besides, this then leads to the problem becoming much more complicated and being NP-hard as well. In view of the complexity of the target problem, we extend the framework of ant colony system, which has been proven effective for solving complex single objective optimization problems, to explore its potential for solving multi-objective scheduling problems.

The rest of this paper is outlined as follows: Section 2 provides a literature review, including batch machine scheduling problems with different constraints. Section 3 illustrates the problem under study and presents the bi-objective integrated mathematical model for the problem. Section 4 describes the proposed multi-objective ant colony optimization algorithm, which employs several distinctive features based on the problem-specific knowledge. Then, taking into account different test instances and evaluation metrics, we conduct experimental studies as presented in Section 5. The paper concludes in Section 6.

2. Related work

In recent years, a body of research has started to address the scheduling problem of a BPM. Initially, researchers started from the simplest model—scheduling for a single BPM. Ikura and Gimple (1986) were probably the first authors study the BPM problem. They proposed an $O(n^2)$ algorithm for a single BPM problem with identical job processing times, unit job sizes, dynamic job arrivals, and an objective of minimizing makespan. Lee et al. (1992) first presented a detailed description for burn-in operation and then studied a problem in which the processing time of jobs were identical and the relationship between release time (r_i) and due date (d_i) was agreeable, i.e., $r_i \leq r_j$ implying $d_i \leq d_j$.

They considered minimizing maximum lateness, L_{max} , and the number of tardy jobs, $\sum U_i$, using dynamic programming algorithms. Uzsoy and Yang (1997) considered the problem of minimizing total weighted completion time, $\sum w_i C_i$, and provided several heuristics and a branch-and-bound algorithm. Jolai (2005) showed that the problem of minimizing the number of tardy jobs, $\sum U_i$, is NP-hard. He developed a dynamic programming algorithm with polynomial time complexity for a fixed number of job families and limited BPM capacity.

Real-world production systems usually require BPMs arranged in parallel in order to prevent the system from being blocked by the unavailability (e.g., breakdown) of a single BPM. Accordingly, parallel systems have been the focus on several studies. Lee et al. (1992) examined the worst-case error bound of any list-scheduling algorithm for minimizing C_{max} with unit job sizes and used the longest processing time (LPT) rule to the case of identical BPMs. Chang et al. (2004) proposed a simulated annealing (SA) algorithm to minimize C_{max} on BPMs in parallel with non-identical job sizes and compared the results obtained from the CPLEX Solver. Kashan et al. (2008) developed a hybrid genetic algorithm (GA) to minimize makespan when scheduling different size jobs. A simulated annealing approach was employed as a comparator algorithm. Damodaran and Velez-Gallego (2010) proposed a constructive heuristic to schedule jobs with non-identical job sizes and arbitrary release times and compared the proposed algorithm with several heuristics in the literature with respect to solution quality and computational cost. Wang and Chou (2010) provided a mixed integer programming model for identical BPMs with non-identical job sizes and dynamic job arrivals. They also proposed two meta-heuristics based on SA and GA, which were combined with a dynamic programming algorithm.

To make a BPM scheduling problem closer to the real-world situation, the constraints of non-identical job sizes and dynamic job arrivals should be considered. Considering jobs with non-identical sizes, Uzsoy (1994) presented complexity results for C_{max} and $\sum C_i$ criteria. He also provided several heuristics and a branch-and-bound algorithm. To schedule a BPM to minimize C_{max} with different job sizes, Melouk et al. (2004) and Damodaran et al. (2006) proposed two different algorithms: a simulated annealing approach and a genetic algorithm, respectively. Kashan et al. (2006) proposed two different genetic algorithms based on different encoding schemes to minimize makespan with non-identical job sizes. Kashan and Karimi (2008) developed an ant colony framework in two versions, varying according to the type of embedded heuristic information, to minimize total weighted completion time of a single BPM with incompatible job families and arbitrary job sizes. Considering jobs with arbitrary release times, Lee and Uzsoy (1999) first presented polynomial and pseudo polynomial-time algorithms for several special cases and developed various heuristics to minimize makespan on a single BPM in the presence of dynamic job arrivals. Sung and Choung (2000) proposed some heuristics to minimize C_{max} on a single burn-in oven in semiconductor manufacturing with different job release times. Wang and Uzsoy (2002) combined a dynamic programming algorithm with a random key-based representation genetic algorithm to minimize L_{max} on a single BPM in the presence of job release times. Malve and Uzsoy (2007) considered the problem of minimizing maximum lateness L_{max} on identical BPMs with dynamic job arrivals. They developed a family of iterative improvement heuristics and combined them with a genetic algorithm.

Clearly, although BPM scheduling with different constraints has been extensive researched, these studies have largely been limited to BPM problems with only a single objective function (e.g., C_{max}), and optimizing a single objective generally may lead to deterioration of other possible objectives. In semiconductor manufacturing

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