Applied Soft Computing Journal 73 (2018) 283–305

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Minimizing the resource consumption of heterogeneous batch-processing machines using a copula-based estimation of distribution algorithm

Chuang Liu [a], Huaping Chen [a], Rui Xu [b,*], Yu Wang [a]

[a] School of Management, University of Science and Technology of China, Hefei 230026, China
[b] School of Business, Hohai University, Nanjing 210098, China

HIGHLIGHTS

- A flow shop scheduling problem on batch-processing machines is considered.
- The objective is to minimize resource consumption under the deadline constraint.
- An attempt is made to solve the scheduling problem by using the copula-based EDA.
- An advanced dynamical adjustment to correction algorithm is developed.
- Computational results show that the copula-based EDA outperforms GA, PSO and $MIMIC_c$.

ARTICLE INFO

Article history:
Received 9 May 2017
Received in revised form 2 August 2018
Accepted 29 August 2018
Available online xxxx

Keywords:
Flow shop
Batch-processing machines
Estimation of distribution algorithm
Resource consumption

ABSTRACT

The two-stage flow-shop scheduling problem investigated in this work aims minimize the resource consumption of non-identical job sizes. The flow shop consists of two batch-processing machines (BPMs): a parallel batch BPM and a sequential BPM. The makespan and resource consumption are considered together in this study, the makespan is the constraint condition, and the resource consumption is the objective. A copula-based Estimation of Distribution Algorithm (cEDA) is used to solve the problem. In this study, the individuals are coded by the allocated resource sequences of all jobs in two machines, and the convex resource consumption function is adopted to simulate the relationship between the processing time of the jobs and the resources allocated to the jobs. A Gaussian distribution is adopted as the marginal probabilistic distribution of all the components. The proposed copula function C_1 assumes independence among the components, whereas the Clayton copula function C_2 assumes that all components are interrelated and introduced for comparison. The computational experiments and comparisons verify the effectiveness of the proposed cEDA. In addition, the copula functions C_1 and C_2 adopted in the proposed cEDA approach are compared.

1. Introduction

Resource conservation and green manufacturing have become critical in the sustainable development of modern industrial companies. A higher resource efficiency result in less expenses cost and less environmental pollution. Production systems that consider resource allocation and enhance resource efficiency, have attracted the attention of researchers in the context of manufacturing industries, such as the steel industry Müller et al. [1], Dahlström and Ekins [2], the food processing industry Henningsson et al. [3], Pagotto and Halog [4], textile industries Martínez [5], chemical industries García et al. [6] and the coating/painting industry Alkaya and Demirer [7]. Jia et al. [8] investigated the problem of minimizing the makespan and the total electric power resource cost on a set of parallel identical batch-processing machines (BPMs). In problems that consider resource consumption, the processing time is generally determined by the amount of resources allocated.

In this work, the flow-shop scheduling problem treats resource consumption as the objective. The flow shop consists of two machines: the first machine, M_1, is a parallel BPM, and the second machine, M_2, is a sequential BPM. All of the jobs may have different sizes, and all of the jobs are ready to be processed at time zero. Each job must be processed successively on two machines. The processing time of each job on one machine depends on the amount of allocated resources. More resources, results in a shorter processing time for each job. The same amount of resources may result in different processing times for different jobs. On machine M_1, all of the jobs in the same batch can be processed simultaneously, and

* Corresponding author.
E-mail addresses: chuang@mail.ustc.edu.cn (C. Liu), hpchen@ustc.edu.cn (H. Chen), rxu@hhu.edu.cn (R. Xu), rainw@mail.ustc.edu.cn (Y. Wang).

https://doi.org/10.1016/j.asoc.2018.08.036
1568-4946/

each job can share the resource allocated to the jobs belonging to the same batch. On machine M_2, the setup time is considered. The setup operation can be performed only after the batch arrives at M_2. The setup time (T^s) is determined by the allocated resource (u^s). The batch cannot wait between two machines(i.e., no-wait). If M_2 is busy, the batch that has been processed completely on M_1 must be blocked. In addition, the scheduling scheme must follow the rule that the constituents of each batch and the batch processing sequence on the two machines are identical. The objective of this problem is the amount of resource consumed, and the makespan C_{max} is a constraint condition that cannot be greater than the deadline D^l. If C_{max} is greater than D^l, more resources must be invested to reduce the processing time and C_{max}. Lower resource consumption typically results in higher resource utilization.

Practical applications of the problem addressed in this work can be easily found in industry. For example, Muthuswamy et al. [9] described a cleaning process observed at a sensor manufacturing facility. There is a boning operation, which establishes the electronic circuitry in a wire bonding machine. The substrates used for sensors are made of ceramic. Because residuals in the substrate could lead to defects in subsequent operations, the residuals must be removed. A substrate with residuals requires two cleaning operations in two sets of baths. In the first bath, because multiple parts that are attached to the same fixture can be cleaned at the same time, the cleaning time of a fixture is equal to the time required to clean the part with the longest cleaning time. Next, all parts belonging to the same batch move across a water jet to wash the chemicals and any additional residuals on the substrate in the second bath. Because one jet can wash only one part, a batch's cleaning time is equivalent to the sum of all of the parts in that batch. Similar practical applications can be found in [10–12]. In the manufacturing industry, optimizing resource consumption objective can reduce the cost of production, including the financial outlay, cost of manpower, cost of equipment and cost of fossil fuels. Decreasing the cost of manpower means that the manufactured products have greater competitiveness in the market, and a lower consumption of fossil fuel can result in fewer pollutants and lower waste gas purification costs [13–16].

The remainder of this study is organized as follows. Section 2 reviews the flow shop with BPMs, the resource consumption function (RCF) and the resource allocation policy of resources in a batch. We define and formulate the problem in Section 3. In Section 4, the solution representation and population initialization, the marginal probabilistic model, the proposed copula function, and the copula updating approach are explained in succession. Next, the approach for forming and sequencing batches as well as the correction algorithm for individuals are presented as well. In addition, the algorithm representation of the copula-based estimation of distribution algorithm (cEDA) is described. Computational experiments and comparisons among the different algorithms are provided in Section 5. Finally, the conclusions of this work are provided in Section 6.

2. Literature review

2.1. Flow shop with BPMs

Batch-processing problems began from the research that Lee et al. [17] initially introduced in their study of the batch-processing problem. This problem was abstracted from an industrial model of a burn-in oven in a semiconductor burn-in operation. The batch-processing operation aims to reduce the amount of resources consumed in the economic production procedure. Since then, BPMs have been widely applied in various industry sectors, ranging from traditional manufacturing to electronic chip processing. Lee et al. [17] examined the BPM of scheduling semiconductor burn-in operations. Tang et al. [18] presented a single BPM scheduling problem considering transportation and deterioration in a steel production setting. Zhou et al. [19] considered a parallel BPM scheduling problem in the presence of non-identical job sizes and arbitrary release times.

There are two BPMs: a parallel BPM (p-BPM), which can process jobs simultaneously in the same batch and a sequential BPM (s-BPM), which can process jobs serially in the same batch. The notations p-BPM and s-BPM are adopted from Oulamara [12] to represent the type of BPM. More specially, p-BPM means that the first machine, M_1, is a parallel BPM, and s-BPM means that the second machine, M_2, is a sequential BPM. The aforementioned BPMs are all p-BPMs. Mathirajan and Sivakumar [20] reviewed p-BPM and s-BPM problems in the semiconductor manufacturing industry.

The studies described above seldom considered the controllable processing time of jobs with resource constraints. In addition, batch processing with a controllable processing time is more complex than normal single-machine scheduling due to the resource consumption involved. However, resource consumption, which is an economically influential factor related to the processing cost, plays an increasingly significant role in making economic and industrial decisions.

The two types of BPMs (p-BPM and s-BPM) have distinct practical applications in many fields of industrial production. The two type of BPMs have different approaches to allocating resources to jobs: p-BPM typically adopts an even resource allocation policy (ERA, such as Ng et al. [21]), whereas a s-BPM employs a manually controllable resource allocation policy (MCRA, such as Oron [22]). Batch-processing scheduling problems, which consider resource consumption, will lead to the machines' controllable processing time or setup time. In addition, the controllable processing time may increase the complexity of the problem.

There are more significant and practical multi-stage flow-shop scheduling problems that are more complex than scheduling problems with a single machine. Flow-shop scheduling problems require consideration of the flow operation from the upstream machine to the downstream machine. Multi-stage flow-shop scheduling problems are widely used in the production procedure environment.

The work in this paper can be regarded as an extension of the classical two-machines flow-shop scheduling problems with two BPMs that do not consider controllable processing time, such as [9,12,23,24]. It is easily verified that the extended problem addressed in this work can be reduced to the classical problem after affixing a resource allocation to all jobs. Once the resource allocation to all jobs is fixed, an optimal job-processing order must be determined from all possible job orders to obtain the minimum value of the makespan. In other words, the problem in this work has been converted to the classical scheduling problem, which is similar to the studies above. Those studies all addressed the two-stage flow shop, in which one machine was a parallel BPM and the other machine was a sequential BPM. However, they examined only the objective of makespan, not resource consumption. In addition, only a few researchers studied problems that consider resource consumption, and they typically treated resource consumption in the problem as the constraint, not as an objective. For example, Shabtay et al. [25] considered a no-wait two-machine flow-shop scheduling problem with convex resource-dependent processing times. Janiak et al. [26] presented a survey of resource management in machine-scheduling problems.

Some researchers have studied the resource-related objectives. For instance, Vickson [27] studied the linear combination of job-processing cost and flow cost in a single machine, where the job-processing cost was related to the amount of resources consumed. Vickson [28] analyzed a similar problem involving a tardiness criterion in two single machines. Nowicki and Zdrzałka