Computers & Operations Research 38 (2011) 809–815

Contents lists available at ScienceDirect

Computers & Operations Research

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

Minimization of maximum lateness on parallel machines with sequence-dependent setup times and job release dates

Shih-Wei Lin [a], Zne-Jung Lee [b], Kuo-Ching Ying [c,*], Chung-Cheng Lu [d]

[a] Department of Information Management, Chang Gung University, Taoyuan, Taiwan
[b] Department of Information Management, Huafan University, Taipei, Taiwan
[c] Department of Industrial Engineering and Management, National Taipei University of Technology, Taipei, Taiwan
[d] Graduate Institute of Information and Logistics Management, National Taipei University of Technology, Taipei, Taiwan

ARTICLE INFO

Available online 8 October 2010

Keyword:
Parallel machines scheduling problems
Sequence-dependent setup times
Maximum lateness
Iterated greedy heuristic

ABSTRACT

In this paper, we consider an identical parallel machine scheduling problem with sequence-dependent setup times and job release dates. An improved iterated greedy heuristic with a sinking temperature is presented to minimize the maximum lateness. To verify the developed heuristic, computational experiments are conducted on a well-known benchmark problem data set. The experimental results show that the proposed heuristic outperforms the basic iterated greedy heuristic and the state-of-art algorithms on the same benchmark problem data set. It is believed that this improved approach will also be helpful for other applications.

1. Introduction

The scheduling problem addressed in this study aims to minimize maximum lateness on identical parallel machines with sequence-dependent setup times (SDSTs) and job release dates constraints. According to the standard three-field notation introduced by Graham et al. [1], this problem is denoted as $P/ST_{sd},r_j/L_{max}$. As a result of wide applications in practice, the study of scheduling problems with sequence-dependent setup times or job release dates has attracted considerable attentions in recent years. Surprisingly, while there is a large body of works devoting to the scheduling problems on parallel machines, very few have considered SDSTs or job release dates. Lenstra et al. [2] showed that parallel machine scheduling problems (PMSP) with release dates constraint are *NP-hard in the ordinary sense* whatever the performance criterion considered. Also, Ullman [3] indicated that PMSPs with SDSTs are *NP-hard in the strong sense*. Therefore, the increasing complexity of PMSPs with SDSTs and/or job release dates is the main reason why most of the previous studies did not consider SDSTs or release dates.

Allahverdi et al. [4,5] conducted an extensive review on machine scheduling problems; they revealed that PMSPs with processing characteristics such as SDSTs and release dates remain largely unstudied. Particularly, to our current knowledge, none of the previous studies has attempted to solve the $P/ST_{sd},r_j/L_{max}$ problem by using exact solution methods. Currently available literature on this strongly NP-hard problem [6] solely focuses on the development of heuristic (or approximate) solution procedures. One of the pioneering works on the $P/ST_{sd},r_j/L_{max}$ problem is by Ovacik and Uzsoy [6], who proposed a family of rolling horizon procedures (*RHPs*) to address this problem. Their computational results showed that *RHPs* consistently outperformed dispatching rules combined with local search techniques. However, as indicated by Lee et al. [7], some solutions obtained by Ovacik and Uzsoy were infeasible. Separated for many years, Kim and Shin [8] presented a restricted Tabu search (*RTS*) approach for the same problem. It was found that, on the whole, the performance of the *RTS* was superior to that of the *RHPs*, the basic Tabu search (*TS*), and the simulated annealing (*SA*). Nevertheless, the performance improvement of *RTS* is gained at the expense of the extra work on tuning and customization, which may prevent it from being generalized to other problems.

Recently, Lee et al. [7] proposed a restricted simulated annealing (*RSA*) algorithm that incorporates a restricted search strategy for solving the $P/ST_{sd},r_j/L_{max}$ problem. Based on extensive computational experiments, Lee et al. demonstrated that the *RSA* algorithm is highly effective as compared to the best solutions found in the literature on the same benchmark problem data set. The aforementioned state-of-the-art meta-heuristics serve as effective and efficient approaches for solving the $P/ST_{sd},r_j/L_{max}$ problem. However, these meta-heuristics are very sophisticated algorithm-wise, hence requiring arduous coding efforts for their implementation. For practical purposes, a simple and effective approach is preferable. Therefore, it is necessary to develop approaches that are easy to be comprehended and have a greater potential of being used effectively.

* Corresponding author. Tel.: +866 2 2771 2171; fax: +866 2 2731 7168.
E-mail address: kcying@ntut.edu.tw (K.-C. Ying).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2010.09.020

The iterated greedy (*IG*) heuristic, proposed by Jacobs and Bruscos [9], represents one of the simplest meta-heuristics both conceptually and practically. In spite of its simplicity, *IG* heuristics have been applied successfully to a number of combinatorial optimization problems without relying on too much problem-specific knowledge. For example, satisfactory results have been reported from the applications to set covering problems [9,10] and some scheduling problems [11–16]. With a particular emphasis on developing an algorithm that is easy to be implemented and effective for solving real-world $P/\mathrm{ST_{sd}},r_j/L_{\max}$ problems, an improved *IG* heuristic with a sinking temperature is proposed in this study.

The rest of the article is divided into four sections. The $P/\mathrm{ST_{sd}},r_j/L_{\max}$ problem dealt within this study is formulated in Section 2. In Section 3, the proposed *IG* heuristic is elaborated. In Section 4, the proposed IG heuristic is empirically evaluated by a famous benchmark problem data set, and the comparison of the algorithmic performances against the state-of-the-art meta-heuristics from the relevant literature is reported. Finally, conclusions are drawn together with recommendations for future research in Section 5.

2. Problem formulation

We consider the problem of scheduling n independent jobs $N=\{1, 2, \ldots, n\}$ on a set of m identical parallel machines, $M=\{1, 2, \ldots, m\}$. Each job $j \in N$ is to be processed by exactly one of the machines, and has a processing time p_j, a release date r_j, and a due date d_j. A sequence-dependent setup time s_{ij} independent of the machine used is required if job j is the immediate successor of job i on a machine. Without loss of generality, we assume the jobs are numbered in non-decreasing order of their release dates and the value of s_{ij} is set to zero when $i=j$. Further, s_{0j} represents the setup time of job j when it is the first job to be processed on a machine.

The objective is to find a feasible schedule $\pi=\{\pi_1, \pi_2, \ldots, \pi_m\}$ that minimizes the maximum lateness $L_{\max}=\max_{1 \le j \le n}\{L_j\}$, where π_k ($k \in M$) is the sequence of jobs on machine k, and $L_j=C_j-d_j$ is the lateness of job j, which is defined as the offset between its completion time C_j and its due date d_j. Moreover, the following assumptions are made for the $P/\mathrm{ST_{sd}},r_j/L_{\max}$ problem considered in this study:

- The number of machines is smaller than the number of jobs, i.e., $m<n$ in order to avoid trivial cases.
- All the parameters are deterministic non-negative integers except that the due date of each job is allowed to be a negative quantity.
- All machines are continuously available and each machine can handle one job at a time without preemption.
- Each job can be processed only once on any one machine.

3. Development of the proposed iterated greedy heuristic

Stochastic local search (*SLS*) algorithms have long been regarded as powerful approaches for solving computationally difficult problems. Some famous *SLS* such as Tabu search (*TS*), simulated annealing (*SA*), memetic algorithms (*MA*), ant colony optimization (*ACO*), evolutionary computation (*EC*), variable neighborhood search (*VNS*), and iterated local search (*ILS*) have been successfully applied in many areas of operations research. Among the above *SLS* methods, iterated greedy (*IG*) heuristic is a novel method that builds a sequence of solutions by iterating greedy construction heuristics through two main steps: *destruction* and *construction*.

As depicted in Fig. 1, a generic *IG* heuristic usually starts from an initial solution ξ_0 and then generates a sequence of solutions by iterating the greedy constructive heuristic through the *destruction* and *construction* phases [11]. In the *destruction* phase, a partial candidate solution ξ_d is obtained by removing a fixed number of elements from a current candidate solution ξ. Then, in the *construction* phase, a greedy constructive heuristic is utilized to sequentially insert the removed elements into the partial solution (ξ_d) until a full solution ξ_c is re-constructed. Once a new full solution is re-constructed, an acceptance criterion is applied to judge whether the incumbent solution will be replaced by the new solution or not. The process iterates through the *destruction* and *construction* phases until some termination conditions have been met. In addition, another local search heuristic can be applied before the main loop and the acceptance test to improve the initial solution and the re-constructed full solution, respectively.

In the literature [9–16], a simple simulated annealing-like acceptance criterion with a constant temperature was integrated into *IG* heuristics. However, an *IG* algorithm using the acceptance criterion with a constant temperature may be stagnated due to insufficient diversification. As an alternative, in this study, we considered an acceptance criterion with a sinking temperature value that is frequently used in simulated annealing algorithms. Based on the framework of the generic *IG* heuristic, the solution representation and the main steps of the proposed improved *IG* heuristic with a sinking temperature, named IG_{ST}, are discussed further in the following subsections.

```
Procedure Iterated_Greedy
begin
    ξ0 := Generate_Initial _Solution
    ξ := Local_Search (ξ0)                     % optional
    while termination conditions not meet do
        ξd := Destruction (ξ)
        ξc := Construction (ξd)
        ξ' := Local_Search (ξc)                % optional
        ξ := Acceptance_Criterion (ξ, ξ')
    return ξbest
end
```

Fig. 1. An outline of a generic *IG* heuristic.