Contents lists available at ScienceDirect

Computers & Operations Research

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

A study of hybrid evolutionary algorithms for single machine scheduling problem with sequence-dependent setup times

Hongyun Xu [a], Zhipeng Lü [a,*], Aihua Yin [b], Liji Shen [c], Udo Buscher [c]

[a] SMART, School of Computer Science and Technology, Huazhong University of Science and Technology, 430074 Wuhan, PR China
[b] School of Software Communication Engineering, Jiangxi University of Finance and Economics, 330013 Nanchang, PR China
[c] Faculty of Business and Economics, Technische Universitaet Dresden, D-01062 Dresden, Germany

ARTICLE INFO

Available online 30 April 2014

Keywords:
Single machine scheduling
Sequence-dependent setup times
Hybrid evolutionary algorithm
Crossover operator
Population updating

ABSTRACT

We present a systematic comparison of hybrid evolutionary algorithms (HEAs), which independently use six combinations of three crossover operators and two population updating strategies, for solving the single machine scheduling problem with sequence-dependent setup times. Experiments show the competitive performance of the combination of the linear order crossover operator and the similarity-and-quality based population updating strategy. Applying the selected HEA to solve 120 public benchmark instances of the single machine scheduling problem with sequence-dependent setup times to minimize the total weighted tardiness widely used in the literature, we achieve highly competitive results compared with the exact algorithm and other state-of-the-art metaheuristic algorithms in the literature. Meanwhile, we apply the selected HEA in its original form to deal with the unweighted 64 public benchmark instances. Our HEA is able to improve the previous best known results for one instance and match the optimal or the best known results for the remaining 63 instances in a reasonable time.

1. Introduction

Given a job set $J=\{1, 2, \ldots, N\}$ of N jobs which should be processed on a single machine, each job $j \in J$ has a processing time p_j, a due date d_j, a weight w_j, and an initial setup time s_{0j} when job j is the first job processed on the machine. In addition, there is a sequence-dependent setup time s_{ij} for any two jobs i and j that will be incurred if job j is processed immediately after job i, where s_{ij} is not necessarily equal to s_{ji}. Let $\pi=\{\pi_1,\pi_2,\ldots,\pi_N\}$ $(\pi_i \neq \pi_j,\ i \neq j)$ represent a processing sequence of the N jobs, where π_j denotes the job processed in the jth position of π. Then, the completion time and tardiness of job π_j are denoted as C_j and T_j respectively:

$$C_j = C_{j-1} + s_{\pi_{j-1}\pi_j} + p_{\pi_j}, \tag{1}$$

$$T_j = \max(0, C_j - d_{\pi_j}). \tag{2}$$

The total weighted tardiness of the sequence π is given as

$$F(\pi) = \sum_{j=1}^{N} w_{\pi_j} T_j. \tag{3}$$

Therefore, the objective of the single machine scheduling problem with sequence-dependent setup times, known as $1|s_{ij}|\sum w_jT_j$ according to the notation of [1], is to find a job sequence which minimizes the total weighted tardiness $F(\pi)$.

The single machine scheduling (SMS) problem attracts extensive attention in the scheduling areas for almost five decades and has been widely used in the actual industry production. The $1|s_{ij}|\sum w_jT_j$, $1|s_{ij}|\sum T_j$, and $1||\sum w_jT_j$ problems are three special cases of the SMS problem.

The $1|s_{ij}|\sum w_jT_j$ problem is NP-hard since its special case, $1||\sum w_jT_j$, has proved to be NP-hard in [2,3]. Given the interest of the SMS problem and its NP-hard nature, a large number of solution procedures, including exact algorithms, heuristics, and metaheuristics, have been reported in the literature. We briefly review below some of the most representative algorithms.

There are several exact methods, for instance, dynamic programming algorithms and branch and bound algorithms, developed to solve the SMS problems in [4–9]. Recently, Tanaka and Araki [10] proposed an exact algorithm which is able to optimally solve almost all the well-known benchmark instances of the $1|s_{ij}|\sum w_jT_j$ and $1|s_{ij}|\sum T_j$ problems.

On the other hand, a large number of heuristic and metaheuristic algorithms have shown to be highly effective for finding high-quality solutions in an acceptable amount of time. The priority dispatching rules are the first heuristic approaches put forward to solve the scheduling problems. The apparent tardiness cost (ATC) rule was proposed by Vepsalainen and Morton [11] for the $1\|\sum w_jT_j$ problem, based on which Lee et al. [12] presented the apparent tardiness cost with setups (ATCS) rule for the $1|s_{ij}|\sum w_jT_j$ problem.

* Corresponding author.
E-mail address: zhipeng.lui@gmail.com (Z. Lü).

http://dx.doi.org/10.1016/j.cor.2014.04.009
0305-0548/© 2014 Elsevier Ltd. All rights reserved.

For the $1|s_{ij}|\sum w_jT_j$ problem, some representative metaheuristic methods include Stochastic Sampling Approaches [13], Genetic Algorithms [14,15], Ant Colony Optimization [16,17], Tabu Search and Simulated Annealing [14], Beam Search [18], Discrete Particle Swarm Optimization approach [19], Discrete Differential Evolution algorithm [20], and Variable Neighborhood Search algorithm [21].

Recently, Xu et al. [22] presented an Iterated Local Search (ILS) to deal with the $1|s_{ij}|\sum w_jT_j$ problem. In the ILS algorithm, a new neighborhood structure called *Block Move* and a fast incremental evaluation technique for evaluating neighborhood solutions were proposed. Applying the proposed ILS algorithm to solve the well-known instances of the $1|s_{ij}|\sum w_jT_j$ problem, the results of the ILS are competitive compared to the previously proposed exact algorithm and five sets of best solutions of state-of-the-art metaheuristic algorithms in the literature.

Besides, there are also some metaheuristic algorithms for the $1|s_{ij}|\sum T_j$ problem, such as Genetic Algorithm [23], Memetic algorithm [24], Ant Colony Optimization [25], Greedy Randomized Adaptive Search Procedure [26], Iterated Greedy heuristic [27]. In addition, a lot of studies on the $1\|\sum w_jT_j$ problem have been presented in the literature, (see for examples [28–32]).

In this paper, we study the hybrid evolutionary algorithms for solving the single machine scheduling problem with sequence-dependent setup times. For hybrid evolutionary algorithms, one of the most effective approaches is to embed a local search procedure into the framework of the population based evolutionary algorithm, where two of the most important factors are the crossover operator and population updating strategy. The crossover operator is used to generate promising offspring which should not only be different from their parents but also inherit some good components from their parents. The population updating strategy is employed to maintain the health of the population. Therefore, the main purpose of our paper is to demonstrate how to choose suitable crossover operators and updating strategies in a hybrid evolutionary algorithm. For this purpose, we take the single machine scheduling problem as a case study.

Specifically, we compare the performance of our HEAs, independently with the six combinations of the three crossover operators and two population updating strategies, for solving the $1|s_{ij}|\sum w_jT_j$ problem. We find that the HEA with the combination of the linear order crossover operator and the similarity-and-quality based population updating strategy outperforms other HEA versions. We evaluate the effectiveness and efficiency of the HEA by applying it to solve 120 public problem instances of $1|s_{ij}|\sum w_jT_j$ and 64 public problem instances of $1|s_{ij}|\sum T_j$.

The remaining part of the paper is organized as follows. In Section 2, the ingredients of our hybrid evolutionary algorithm are described, including the initial population, the local search procedure, three different crossover operators, and two population updating strategies. In Section 3, we compare the six combinations of three crossover operators and two population updating strategies. Moreover, the computational results and comparisons between our HEA and other state-of-the-art algorithms for solving the well-known instances of the $1|s_{ij}|\sum w_jT_j$ and $1|s_{ij}|\sum T_j$ problems are presented. In Section 4, we analyze the performance of our HEA by comparing with its two variants: one is to generate the initial population using the ATCS heuristic; the other is to choose parents for the crossover operator using the roulette wheel strategy. Finally, concluding remarks are given in Section 5.

2. Hybrid evolutionary algorithm

2.1. Main scheme

Hybrid evolutionary algorithms, also known as memetic algorithms, are considered as highly effective means to solve a large number of constraint satisfaction and optimization problems [33–37]. In a hybrid evolutionary algorithm, a better balance between the exploration and the exploitation of the search space would be achieved by combining a more global recombing search and a more intensive local search.

Generally, our HEA repeatedly operates between a crossover operator that is used to generate new offspring solutions and a local search procedure that brings individual offspring to a better solution. As soon as the offspring solutions are improved by the local search, the population is updated based on the population updating strategy. One observes that the crossover operator and the population updating strategy are the most important ingredients of the hybrid evolutionary algorithm. Thus, choosing proper crossover operator and population updating strategy has direct influence on the performance of the hybrid evolutionary algorithm in solving difficult optimization algorithms.

We present the general architecture of the HEA in Algorithm 1. It is composed of four main components: population initialization, a local search procedure, a crossover operator, and a population updating rule. Starting from an initial random population, HEA uses the local search procedure to optimize each individual to reach a local optimum (lines 4–6). Then, the crossover operator is employed to generate new offspring solutions (line 10). Accordingly, a new round of local search is launched again to optimize the objective function. Subsequently, the population updating rule decides whether such an improved solution should be inserted into the population and which existing individual should be replaced (line 15). In the following subsections, the four components of our HEA are described in detail.

Algorithm 1. Pseudo-code of the HEA for the $1|s_{ij}|\sum w_jT_j$ problem.

```
1:  Input: $J$, $s_{ij}$
2:  Output: the best job sequence $x^*$ found so far
3:  $P = \{x^1, \dots, x^p\} \leftarrow$ Population_Initialization()
4:  for $i = \{1, \dots, p\}$ do
5:    $x^i \leftarrow$ Local_Search($x^i$)
6:  end for
7:  $x^* = \text{argmin}\{f(x^i) | i = 1, \dots, p\}$
8:  repeat
9:    randomly choose two individuals $x^j$ and $x^k$ from $P$
10:   $x^0 \leftarrow$ Crossover_Operator($x^j, x^k$)
11:   $x^0 \leftarrow$ Local_Search($x^0$)
12:   if $f(x^0) < f(x^*)$ then
13:     $x^* = x^0$
14:   end if
15:   $\{x^1, \dots, x^p\} \leftarrow$ Population_Updating($x^0, x^1, \dots, x^p$)
16: until the stop criterion is met
```

2.2. Initial population

In our HEA, the individuals of the initial population are generated randomly. Then, each individual is further optimized by a local search procedure. Note that although the ATCS heuristic from Lee et al. [12] is a well-known constructive heuristic approach and has been widely used to generate initial solutions for advanced metaheuristic algorithms for the single machine scheduling problem, we yield the individuals of initial population randomly, since the quality of the solutions of the ATCS heuristic procedure is not as diversified as the pure random procedure.

2.3. Local search procedure

In this paper, we employ a steepest descent algorithm as our local search procedure. In each iteration, the algorithm chooses the best