Applied Soft Computing 29 (2015) 184–195

Contents lists available at ScienceDirect

Applied Soft Computing

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

An improved scatter search algorithm for the single machine total weighted tardiness scheduling problem with sequence-dependent setup times

Qingxin Guo [a,b], Lixin Tang [a,*]

[a] Institute of Industrial Engineering & Logistics Optimization, Northeastern University, Shenyang 110819, China
[b] State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China

ARTICLE INFO

Article history:
Received 9 November 2012
Received in revised form
28 December 2014
Accepted 29 December 2014
Available online 6 January 2015

Keywords:
Improved scatter search
Variable neighborhood search
Variable-length reference set
Discrete differential evolution
Total weighted tardiness scheduling

ABSTRACT

In this paper, a scatter search algorithm with improved component modules is proposed to solve the single machine total weighted tardiness problem with sequence-dependent setup times. For diversification generation module, both random strategy based heuristics and construction heuristic are adopted to generate the diversified population. For improvement module, variable neighborhood search based local searches are embedded into the algorithm to improve the trial solutions and the combined solutions. For reference set update module, the number of edges by which the two solutions differ from each other is counted to measure the diversification value between two solutions. We also propose a new strategy in which the length of the reference set could be adjusted adaptively to balance the computing time and solving ability. In addition, a discrete differential evolution operator is proposed with another two operators constitute the combination module to generate the new trial solutions with the solutions in the subsets. The proposed algorithm is tested on the 120 benchmark instances from the literature. Computational results indicate that the average relative percentage deviations of the improved algorithm from the ACO_AP, DPSO, DDE and GVNS are −5.16%, −3.33%, −1.81% and −0.08%, respectively. Comparing with the state-of-the-art and exact algorithms, the proposed algorithm can obtain 78 optimal solutions out of 120 instances within a reasonable computational time.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

One of the most thoroughly studied scheduling problems is the single machine total weighted tardiness problem with sequence-dependent setup times (STWTSDS) [1]. Formally, this problem is generally expressed as $1/s_{ij}/\sum w_jT_j$ and can be stated as follows. A set of n jobs are available at time zero and have to be scheduled without preemption on a single machine that can handle at most one job at a time. Each job j ($j = 1, 2, \ldots, n$) has a processing time p_j, a due date d_j, a weight w_j and a setup time s_{ij} which takes place when job j immediately follows job i in the processing sequence. If job j is processed first, it is assumed that it requires a setup time s_{0j}. The tardiness of job j is defined as $T_j = \max\{0, C_j - d_j\}$, where C_j is the completion time of job j. The objective of this problem is to find a sequence of jobs that minimizes the total weighted tardiness, i.e., $\sum w_jT_j$. Both the features of the weighted tardiness and the sequence-dependent setup times are encountered in a number of real-world applications. In many manufacturing industries, such as iron and steel industry, chemical industry, the mode of their production is make-to-order, so an important objective function is to minimize the total tardiness as it can be used to differentiate between customers. The setup time between two consecutive jobs on the same machine is a function of the size, physical or chemical characteristics of the jobs.

Comparing the STWTSDS with the well-known TSP problem, it can be concluded that they are both permutation problem with the distances (setup times for STWTSDS) between two elements and their purposes are both to find a tour with specific objective functions. According the triplet-field notation in scheduling theory, TSP could be described as $1/s_{ij}/C_{\max}$. Therefore, STWTSDS is also similar with TSP from the perspective of their models in addition to the different objective function. Based on this consideration, the successful methods of TSP may be also effective for STWTSDS.

The STWTSDS problem has been proved to be strongly NP-hard [2]. Actually, $1//\sum w_jT_j$, the generic version of this problem has already been proven to be strongly NP-hard by Lawler [3]. The

* Corresponding author. Tel.: +86 024 83680169.
E-mail address: qhjytlx@mail.neu.edu.cn (L. Tang).

http://dx.doi.org/10.1016/j.asoc.2014.12.030
1568-4946/© 2015 Elsevier B.V. All rights reserved.

feature of sequence-dependent setup times greatly increases the difficulty of solving the problem. So there is a need to develop heuristic algorithms for obtaining near-optimal solutions within a reasonable computation time. The best known construction heuristic for this problem is the *Apparent Tardiness Cost with Setups* (ATCS) rule proposed by Lee et al. [4]. The ATCS rule combines the weighted shortest processing time (WSPT) rule, the minimum slack (MS) rule and the shortest setup time (SST) rule in a single ranking index. The ATCS rule is implemented by assigning jobs in non-increasing order of parameter $I_j(t, l)$ that is given by (1)

$$I_j(t,l) = \frac{w_j}{p_j} \exp\left[-\frac{\max(d_j - p_j - t, 0)}{k_1 \bar{p}}\right] \exp\left[-\frac{s_{lj}}{k_2 \bar{s}}\right] \tag{1}$$

where t denotes the current time and l is the index of the job just completed, $\bar{p}$ is the average processing time, $\bar{s}$ is the average setup time, k_1 and k_2 are the look-ahead parameters fixed as originally suggested in Lee et al. [4].

In the literature, several meta-heuristics have also been proposed for the STWTSDS problem. Cicirello and Smith [5] generated 120 benchmark instances and applied five stochastic sampling approaches: Limited Discrepancy Search (LDS), Heuristic-Biased Stochastic Sampling (HBSS), Value Biased Stochastic Sampling (VBSS), Value Biased Stochastic Sampling seeded Hill-Climber (VBSS-HS) and Simulated Annealing. These instances were next improved by the following meta-heuristics algorithms. Lin and Ying [6] proposed a simulated annealing (SA) method with swap and insertion search, a generic algorithm (GA) with mutation operator performed by a greedy local search and a tabu search (TS) with a swap and an insertion tabu list. Liao and Juan [7] presented an ant colony optimization (ACO) algorithm for this problem. They introduced a new parameter for the initial pheromone trail and adjusted the timing of applying local search. Another ACO with a new global pheromone update mechanism and a new type of asymptotic pheromone trails is described by Anghinolfi and Paolucci [8] and compared with Liao and Juan [7]. Anghinolfi and Paolucci [9] proposed a new discrete particle swarm optimization (DPSO) approach and used a discrete model both for particle position and velocity and a coherent sequence metric that is different from previous DPSO. A discrete differential evolution (DDE) algorithm is presented by Tasgetiren et al. [10], speed-up methods are proposed to facilitate the greedy job insertions in their algorithm. Kirlik and Oguz [11] formulated this problem to a mathematical model and proposed a general variable neighborhood search (GVNS) heuristic to solve it. Chao and Liao [12] presented a discrete electromagnetism-like mechanism (DEM) algorithm for this problem with an attraction–repulsion mechanism involving crossover and mutation operators. A greedy randomized adaptive search procedure (GRASP) combined with path relinking (PR) is proposed by Luo and Hu [13] for this problem. Subramanian et al. [14] studied this problem and developed an iterated local search (ILS) and variable neighborhood descent with random neighborhood ordering meta-heuristic and compare its performance with the state-of-the-art meta-heuristic algorithms. Another ILS including a new neighborhood structure called *BlockMove* and a fast incremental evaluation technique is proposed by Xu et al. [15] to solve this problem. Deng and Gu [16] presented an enhanced iterated greedy (EIG) algorithm with elimination rules and a perturbation operator for this problem. Xu et al. [17] presented a systematic comparison of hybrid evolutionary algorithms (HEAs), which independently use six combinations of three crossover operators and two population updating strategies, for solving this problem and the unweighted 64 public benchmark instances. Additionally, Liao et al. [18] improve the time complexities of searching the interchange, insertion and twist neighborhoods for this problem which could be used to speed-up other algorithms.

1. Use both construction and random heuristics to generate the population
while (doesn't meet the stopping criteria) {
  2. Generate the initial reference set and set *gbest* to be the best solution found so far. Make *NewElements* = TRUE.
  while (*NewElements*) {
    3. Generate subsets from the current reference set. Make *NewElements* = FALSE.
    4. Combine solutions in each subset
    5. Use the appropriate local search to improve each combined solution and update *gbest*
    6. Update the reference set. If reference set is updated, make *NewElements* = TRUE.
  }
  go to Step 1.
}

Fig. 1. The outline of the improved SS algorithm.

Tanaka and Araki [19] proposed an exact algorithm for the STWTSDS problem. The algorithm is based on the SSDP (Successive Sublimation Dynamic Programming) [20,21]. The proposed algorithm is applied to Cicirello's benchmark instances and all of them are optimally solved. However, it still takes a long computation time even with 20GB memory size (for the two hardest instance, 2 weeks and 1 month were taken, respectively). Therefore, for the practical problems or larger size problems than Cicirello's benchmark instances, the computation time of Tanaka and Araki's exact algorithm may increase exponentially. The motivation for our work is the implementation of a new meta-heuristic algorithm with the aim of generating solutions which are closer to the optimal ones for the STWTSDS problem within a reasonable time.

Scatter search (SS) was first introduced in Glover [22] as a meta-heuristic for integer programming. SS is an evolutionary method which operates on a reference set of solutions by intelligently combining these solutions to yield better solutions. The purpose of these combination mechanisms is to incorporate both diversity and quality. Then the combined new solutions must undergo the improvement module which results in the reference set update. The algorithm continues until the stopping criteria are satisfied. Possible stopping criteria include the maximum number of iterations or the elapsed time. The algorithm may also terminate when the objective function reaches a predetermined value. SS is very flexible and has successful applications in several application areas, since each of its elements can be implemented in a variety of ways and degrees of sophistication. Laguna [23] summarized the basic SS framework as five component modules: diversification generation module, improvement module, reference set update module, subset generation module and combination module. The outline of our improved SS algorithm is shown in Fig. 1. The symbol *gbest* is used to describe the best solution found so far. A boolean variable *NewElements* is defined to distinguish whether the reference set is updated. For an overview of the features of SS, we refer to Glover and Laguna [24] and Martí et al. [25].

SS has also provided very promising results for scheduling problems. Jain and Meeran [26] applied a SS and path relinking method to general flow-shop problem. The SS and path relinking strategies were embedded within a core and shell framework that is able to provide substantially better results than a tabu search approach. Nowicki and Smutnicki [27] proposed a modified SS algorithm for the flow-shop problem with makespan criterion. Good properties were scalable and had been confirmed by a large number of tests on common benchmarks. Manikas and Chang [28] used SS algorithm to solve the job shop scheduling problem with sequence-dependent setup times. The experiments showed that SS produces excellent results comparing with solutions obtained by common heuristics,

Download English Version:

https://daneshyari.com/en/article/495237

Download Persian Version:

https://daneshyari.com/article/495237

Daneshyari.com