Computers & Operations Research 53 (2015) 154–164

Contents lists available at ScienceDirect

Computers & Operations Research

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

A tabu search/path relinking algorithm to solve the job shop scheduling problem

Bo Peng [a], Zhipeng Lü [a,b,*], T.C.E. Cheng [b]

[a] SMART, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, PR China
[b] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

Available online 19 August 2014

Keywords:
Scheduling
Job shop
Metaheuristics
Tabu search
Path relinking
Hybrid algorithms

ABSTRACT

We present an algorithm that incorporates a tabu search procedure into the framework of path relinking to generate solutions to the job shop scheduling problem (JSP). This tabu search/path relinking (TS/PR) algorithm comprises several distinguishing features, such as a specific relinking procedure to effectively construct a path linking the initiating solution and the guiding solution, and a reference solution determination mechanism based on two kinds of improvement methods. We evaluate the performance of TS/PR on almost all of the benchmark JSP instances available in the literature. The test results show that TS/PR obtains competitive results compared with state-of-the-art algorithms for JSP in the literature, demonstrating its efficacy in terms of both solution quality and computational efficiency. In particular, TS/PR is able to improve the upper bounds for 49 out of the 205 tested instances and it solves a challenging instance that has remained unsolved for over 20 years.

1. Introduction

The job shop scheduling problem (JSP) is not only one of the most notorious and intractable NP-hard problems, but also one of the most important scheduling problems that arise in situations where a set of activities that follow irregular flow patterns have to be performed by a set of scarce resources. In job shop scheduling, we have a set $M=\{1,\ldots,m\}$ of m machines and a set $J=\{1,\ldots,n\}$ of n jobs. JSP seeks to find a feasible schedule for the operations on the machines that minimizes the makespan (the maximum job completion time), i.e., C_{max}, the completion time of the last completed operation in the schedule. Each job $j\in J$ consists of n_j ordered operations $O_{j,1},\ldots,O_{j,n_j}$, each of which must be processed on one of the m machines. Let $O=\{0,1,\ldots,o,o+1\}$ denote the set of all the operations to be scheduled, where operations 0 and $o+1$ are dummies, have no duration, and represent the initial and final operations, respectively. Each operation $k\in O$ is associated with a fixed processing duration P_k. Each machine can process at most one operation at a time and once an operation begins processing on a given machine, it must complete processing on that machine without preemption. In addition, let p_k be the predecessor operation of operation $k\in O$. Note that the first operation has no predecessor. The operations are interrelated by two kinds of constraints. First, operation $k\in O$ can only be scheduled if the machine on which it is processed is idle. Second, precedence constraints require that before each operation $k\in O$ is processed, its predecessor operation p_k must have been completed.

Furthermore, let S_o be the start time of operation o ($S_0=0$). JSP is to find a starting time for each operation $o\in O$. Denoting E_h as the set of operations being processed on machine $h\in M$, we can formulate JSP as follows:

$$\text{Minimize } C_{max}=\max_{k\in O}\{S_k+P_k\}, \tag{1}$$

subject to

$$S_k\geq 0;\quad k=0,\ldots,o+1, \tag{2}$$

$$S_k-S_{p_k}\geq P_{p_k};\quad k=1,\ldots,o+1, \tag{3}$$

$$S_i-S_j\geq P_i \text{ or } S_j-S_i\geq P_j;\quad (i,j)\in E_h,\ h\in M. \tag{4}$$

In the above problem, the objective function (1) is to minimize the makespan. Constraints (2) require that the completion times of all the operations are non-negative. Constraints (3) stipulate the precedence relations among the operations of the same job. Constraints (4) guarantee that each machine can process no more than one single operation at a time.

Over the past few decades, JSP has attracted much attention from a significant number of researchers, who have proposed a large number of heuristic and metaheuristic algorithms to find optimal or near-optimal solutions for the problem. One of the most famous algorithms is the tabu search (TS) algorithm TSAB proposed by Nowicki and Smutnicki [14]. Nowicki and Smutnicki

* Corresponding author at: SMART, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, PR China.
E-mail addresses: zhipeng.lui@gmail.com (Z. Lü), Edwin.Cheng@polyu.edu.hk (T.C.E. Cheng).

http://dx.doi.org/10.1016/j.cor.2014.08.006
0305-0548/

[15] later extend algorithm TSAB to algorithm i-TSAB, which Beck et al. [5] combine with a constraint programming based constructive search procedure to create algorithm CP/LS. Pardalos and Shylo [16] propose algorithm GES, which is based on global equilibrium search techniques. Zhang et al. [23] extend the N6 neighborhood proposed by Balas and Vazacopoulos [4] to a new neighborhood and Zhang et al. [24] combine TS with SA to create algorithm TS/SA, which outperforms almost all the algorithms. Nagata and Tojo [11] present a local search framework termed guided ejection search, which always searches for an incomplete solution for JSP. Recently, Gonćalves and Resende [9] present the biased random-key genetic algorithm BRKGA, which is able to improve the best known results for 57 instances and outperforms all the reference algorithms considered in their paper. From all these algorithms, it is apparent that the recent state-of-the-art algorithms either hybridize several strategies instead of using a single algorithm or employ a population-based algorithm instead of a single-solution based one.

Among the metaheuristic approaches used to generate solutions to JSP, a powerful local search procedure is always necessary. This observation is especially noticeable for the state-of-the-art algorithms for JSP. As one of the most popular local search algorithms, TS has been widely used by researchers to generate solutions to JSP, e.g., Nowicki and Smutnicki [15], Zhang et al. [23], Nasiri and Kianfar [12], Shen and Buscher [19], and Gonçalves and Resende [9].

On the other hand, Aiex et al. [2] apply path relinking within a GRASP procedure as an intensification strategy to generate solutions to JSP. Furthermore, Nowicki and Smutnicki [15] improve their famous algorithm TSAB by introducing a new initial solution (NIS) generator based on path relinking. Recently, Nasiri and Kianfar [13] take advantage of the N1 neighborhood to construct the path in the path relinking procedure that is identical to i-TSAB's NIS generator.

The above observations and considerations motivate us to develop a more robust algorithm for JSP by combining the more global relinking approach and the more intensive TS. In this vein, we design the tabu search and path relinking (TS/PR) algorithm to strike a better balance between the exploration and the exploitation of the search space in a flexible manner.

We summarize the main contributions of TS/PR as follows: Compared with the state-of-the-art algorithms for generating solutions to JSP, TS/PR uses a specific mechanism to effectively construct the path linking the initiating solution and the guiding solution, as well as using two kinds of improvement methods to determine the reference solution.

The remaining part of the paper is organized as follows: Section 2 describes in detail the components of TS/PR. Section 3 presents the detailed computational results and comparisons between TS/PR and some best performing algorithms in the literature for generating solutions to six sets of a total of 205 challenging benchmark JSP instances. Finally, we conclude the paper and suggest future research topics in Section 4.

2. The TS/PR algorithm

2.1. Main framework

In principle, TS/PR repeatedly operates between a path relinking method that is used to generate promising solutions on the trajectory set up from an initiating solution to a guiding solution, and a TS procedure that improves the generated promising solution to a local optimum. Algorithm 1 presents the main procedure of TS/PR.

Table 1
The description of the symbols used in TS/PR.

Symbols	Description
$j_{k,i}$	The ith operation executed on machine k.
S	A schedule for JSP is represented by permutations of operations on the machines: $\{(j_{1,1}, j_{1,2}, \ldots, j_{1,n}), (j_{2,1}, j_{2,2}, \ldots, j_{2,n}), \ldots, (j_{m,1}, j_{m,2}, \ldots, j_{m,n})\}$.
S^I	The initiating solution for the relinking procedure.
S^G	The guiding solution in the relinking procedure.
S^C	The current solution during the relinking procedure.
$CS(S^I, S^G)$	The common sequence between S^I and S^G: $\{j^I_{k,i} \mid j^I_{k,i} = j^G_{k,i}, k \in M, i \in N\}$.
$NCS(S^I, S^G)$	The set of elements not in the common sequence between S^I and S^G: $\{j^I_{k,i} \mid j^I_{k,i} \neq j^G_{k,i}, k \in M, i \in N\}$.
$Dis(S^I, S^G)$	The distance between S^I and S^G: $\lvert NCS(S^I, S^G)\rvert$.
PairSet	A set that stores the candidate solution pairs for path building.
PathSet	A set that stores the candidate solutions on a single path that will be optimized by the improvement method.
α	The minimum distance between the initiating (or guiding) solution and the first (or last) solution in the *PathSet*.
β	The interval for choosing the candidate solutions in *PathSet*.

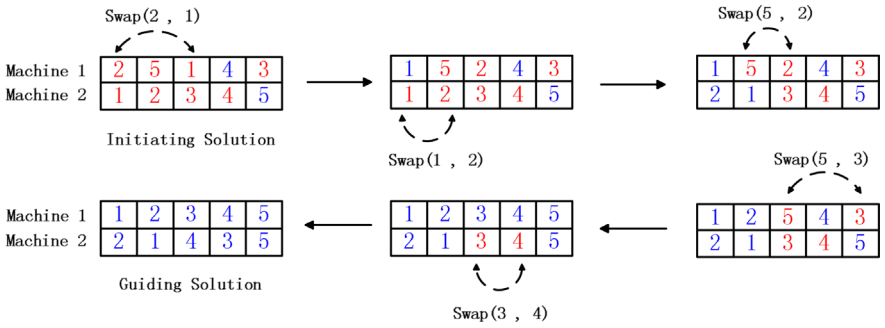

Fig. 1. Path construction procedure.