Computers & Industrial Engineering 62 (2012) 946–952

Contents lists available at SciVerse ScienceDirect

Computers & Industrial Engineering

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

A GES/TS algorithm for the job shop scheduling ☆

Mohammad Mahdi Nasiri *, Farhad Kianfar

Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 25 March 2011
Received in revised form 24 October 2011
Accepted 15 December 2011
Available online 27 December 2011

Keywords:
Scheduling
Job shop
Path relinking
Tabu search
Global equilibrium search

ABSTRACT

The job shop scheduling problem is a difficult combinatorial optimization problem. This paper presents a hybrid algorithm which combines global equilibrium search, path relinking and tabu search to solve the job shop scheduling problem. The proposed algorithm used biased random sampling to have a better covering of the solution space. In addition, a new version of N6 neighborhood is applied in a tabu search framework. In order to evaluate the algorithm, comprehensive tests are applied to it using various standard benchmark sets. Computational results confirm the effectiveness of the algorithm and its high speed. Besides, 19 new upper bounds among the unsolved problems are found.

1. Introduction

The job shop scheduling problem (JSP) is a practical problem and has numerous applications in manufacturing and supply chain scheduling problems (Pinedo, 2008). In addition, the results obtained for JSP can be extended to be used in open shop and even in resource constrained project scheduling problems.

One of the most interesting criteria in JSP literature is the makespan. In fact, the simplicity of the computation of makespan makes it the most popular objective function. The problem (which is denoted by $J_m||C_{\max}$) can be described as follows: n jobs are given that should be done on m machines. Each job is composed of a set of operations which should be processed in a predetermined order. The set of all operations is denoted by $O = \{1, 2, \ldots, o\}$. Each operation u has to be processed without stoppage for a fixed duration (processing time) p_u on a given machine. Operations of the same job cannot be processed at the same time. Each job can visit each machine at most once and no machine can process more than one job at a time. A schedule is an allocation of operations to time periods on a machine. The objective of the problem is to locate a schedule that minimizes the makespan ($C_{\max}$), that is, the completion time of the last operation finished. Each schedule can be represented by a processing order that determines the sequence of operations on each machine.

In the scheduling literature, many algorithms (exact and approximation) are presented to solve the JSP with makespan criterion. Most of the exact algorithms (like Applegate & Cook, 1991) are focused on the branch and bound method. However, since it is proven that the problem is NP-hard (the proof can be found in Garey, Johnson, & Sethi, 1976) many approximation (heuristic and meta-heuristic) algorithms have been developed to find a near optimal solution. Among these efforts, those that used tabu search at least as a part of the algorithm had better results.

Nowicki and Smutnicki (1996) proposed a tabu search algorithm with back jump (TSAB) that performs very well. A comprehensive survey of older scheduling techniques can be seen from Blazewicz, Domschke, and Pesch (1996). Balas and Vazacopoulos (1998) proposed a guided local search with shifting bottleneck. Park, Choi, and Kim (2003) proposed a hybrid genetic algorithm. Nowicki and Smutnicki (2005) proposed *i*-TSAB which remained as a state of the art paper for JSP with makespan criterion for years. Pardalos and Shylo (2006) presented an algorithm based on global equilibrium search (GES) techniques. Sha and Hsu (2006) developed a hybrid particle swarm optimization (PSO) whose particle position is based on preference list-based representation, particle movement is based on swap operator, and particle velocity is based on the tabu list concept. Zhang, Li, Guan, and Rao (2007) tried to extend the neighborhood of Balas and Vazacopoulos (1998). Zhang, Li, Rao, and Guan (2008) combined tabu search (TS) and simulated annealing (SA) and proposed TSSA that outperforms most of the algorithms except *i*-TSAB. Pardalos, Shylo, and Vazacopoulos (2010) used the properties of the search space such as backbone and "big valley" to accelerate the search process. Nasiri and Kianfar (2011a) proposed a hybrid scatter search for an extension of the JSP. Nasiri and Kianfar (2011c) presented a combination of tabu search and path relinking which outperforms many well known algorithms.

☆ This manuscript was processed by Area Editor T.C. Edwin Cheng.
* Corresponding author. Tel.: +98 91 2223 8648; fax: +98 21 3330 6764.
E-mail addresses: mmnasiri@gmail.com (M.M. Nasiri), fkianfar@sharif.edu (F. Kianfar).

0360-8352/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2011.12.018

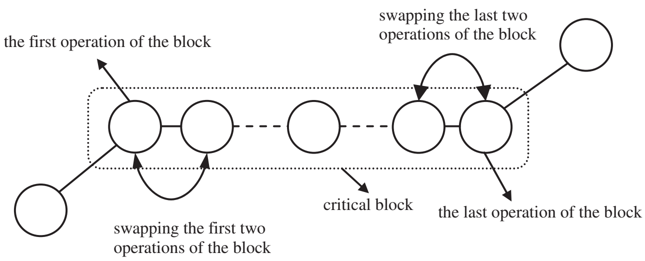

Fig. 1. The outline of the N5 neighborhood.

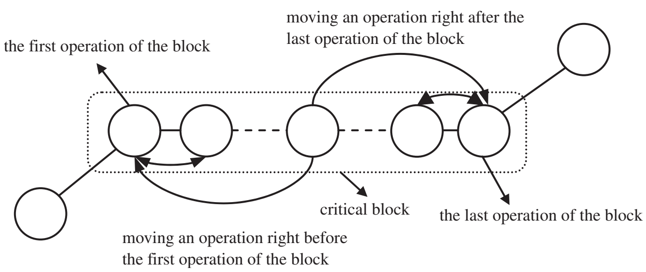

Fig. 2. The outline of the N6 neighborhood.

This paper combined the most effective approaches. Global equilibrium search (GES), path relinking (PR) and tabu search are used to develop a new algorithm (GESTS) that can be used for continuing the search process in long runs. The biased random sampling and path relinking are used for diversification purposes and in order to cover all the search space while the global equilibrium search and tabu search are applied for intensification intentions and in order to escape from local optimums. Therefore, the proposed algorithm (GESTS) is composed of three components: a tabu search (NKTS), the path relinking (NIS) of Nowicki and Smutnicki (2005) and a global equilibrium search.

The remainder of the paper is organized as follows. In Section 2, the neighborhood structures used in the algorithm are explained. Section 3 describes the tabu search procedure. Section 4 hybridizes tabu search and path relinking procedures. In Section 5, the algorithm GESTS is illustrated. Section 6 presents the computational results. Finally, Section 7 concludes the paper.

2. Neighborhood structures

The neighborhood structure is a mechanism which can obtain a new set of neighbor solutions by applying a small perturbation to a given solution (Glover & Laguna, 1997). A given solution is modified to a neighbor solution by applying a move. Since the proposed algorithm used some neighborhood structures, it is necessary to define them. For a comprehensive review of job shop neighborhoods, see Zhang et al. (2007).

The N1 neighborhood structure for the JSP was introduced by Van Laarhoven, Aarts, and Lenstra (1992). The properties of the N1 neighborhood facilitate its application in path relinking procedures. In this neighborhood, a move is defined by the interchange of two successive operations on a critical path that should be processed on the same machine. According to Zhang et al. (2007), two important properties of this neighborhood are:

- Given a feasible solution, the exchange of two adjacent critical operations cannot yield an infeasible solution.
- The permutation of non-critical operations cannot improve the objective function and even may create an infeasible solution.

In order to explain N5 and N6 neighborhoods, the term "block" should be defined. Simply, one can say that the block is a maximal subsequence of a critical path, which contains operations processed on the same machine. In order to define N5 neighborhood, only a single critical path is considered. In every block except the first block and the last block, the first two and the last two operations are swapped. In the first block, only the last two operations and by symmetry, in the last block only the first two operations are swapped (Nowicki & Smutnicki, 1996). Fig. 1 shows the N5 neighborhood.

The experimental results for N6 neighborhood show that it is very effective in deep search. In this neighborhood, an operation within a critical block can be moved right before the first or right after the last operation of the block. The N6 neighborhood is illustrated in Fig. 2.

3. Tabu search procedure

The best results for the job shop scheduling problem are obtained by using local search methods. In fact, TS which is introduced by Glover (1986) is very effective for intensification purposes. In addition, TS has the ability of escaping from local optima. The most effective applications of TS to JSP are TSAB of Nowicki and Smutnicki (1996), *i*-TSAB of Nowicki and Smutnicki (2005) and TSSA of Zhang et al. (2008).

It should be noted that if the initial solution of our tabu search algorithm (NKTS) is obtained from deterministic rules (like SPT or LPT) the algorithm is not a randomized one. Some characteristics that specify NKTS are as follows.

In contrast with TSAB that uses N5 neighborhood, NKTS uses N6 with cycle prevention conditions of Nasiri and Kianfar (2011b).

In addition, NKTS stores *maxl* promising solutions so that if a predetermined number (*Improvelter*) of moves without improvement is done the search process is returned to the promising solutions.

3.1. Move evaluation

When a set of feasible moves has been formed, each move should be evaluated to calculate or estimate the makespan. Since a fast and authentic estimate is available for N6 from Balas and Vazacopoulos (1998), this estimate is utilized. It should be noted that new heads and tails are needed for the estimation of the objective function. Thus, a topological order with the procedure of Nasiri and Kianfar (2011c) is used to calculate only the heads and tails that need to be updated.

3.2. Tabu list

The tabu list eludes returning to the formerly visited solutions. The algorithm TSAB stores moves in tabu list. This may be suitable for N5 neighborhood that it uses. Furthermore, the algorithm TSSA uses attributes of solutions instead of using attributes of moves. Since our algorithm uses N6 neighborhood like TSSA, the same structure for the tabu list is used. Consequently, if a move consists of interchange on operations u and v, operations between u and v, and their positions are saved before implementing the interchange. For the duration that these operations and positions recorded as "tabu", a move that would achieve the same sequence of operations and positions is not permitted. It should be noted that if the solution after the move has the same operations and positions as one of tabu list items but the move has no role in creating this situation, then the move will not be considered as "tabu". In addition, the length of tabu list has important effect on quality of the solutions. This length can be configured by trial and error.