Applied Soft Computing 10 (2010) 79–89

Contents lists available at ScienceDirect

Applied Soft Computing

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

A hybrid immune simulated annealing algorithm for the job shop scheduling problem

Rui Zhang *, Cheng Wu

Department of Automation, Tsinghua University, Beijing 100084, PR China

ARTICLE INFO

Article history:
Received 31 October 2007
Received in revised form 28 January 2009
Accepted 25 June 2009
Available online 2 July 2009

Keywords:
Job shop scheduling problem
Simulated annealing
Artifical immune system
Fuzzy inference system

ABSTRACT

A hybrid simulated annealing algorithm based on a novel immune mechanism is proposed for the job shop scheduling problem with the objective of minimizing total weighted tardiness. The proposed immune procedure is built on the following fundamental idea: the bottleneck jobs existing in each scheduling instance generally constitute the key factors in the attempt to improve the quality of final schedules, and thus, the sequencing of these jobs needs more intensive optimization. To quantitatively describe the bottleneck job distribution, we design a fuzzy inference system for evaluating the bottleneck level (i.e. the criticality) of each job. By combining the immune procedure with a simulated annealing algorithm, we design a hybrid optimization algorithm which is subsequently tested on a number of job shop instances. Computational results for different-sized instances show that the proposed hybrid algorithm performs effectively and converges fast to satisfactory solutions.

1. Introduction

The job shop scheduling problem (JSSP) has been an important research focus in both theoretical and industrial fields ever since the 1950s. Concerning its difficulty, JSSP was shown to be $\mathcal{NP}$-hard in the strong sense [18]. Therefore, it is dramatically difficult to find the absolutely optimal solution even for very small instances. Actually, the well-known JSSP benchmark problem labeled as "FT10" or "MT10" which contains only 10 jobs and 10 machines was not solved to optimality until two decades after the problem was initially constructed [1].

After recognizing the remarkable difficulty of JSSP and the poor performance of exact solution methods, a lot of researchers now put their effort into meta-heuristic and local search optimization strategies for solving JSSP, such as simulated annealing (SA) [15,27,3], genetic algorithm (GA) [28,7], tabu search (TS) [20], scatter search (SS) [8,25], particle swarm optimization (PSO) [26,17] and ant colony optimization (ACO) [5,11]. These intelligent optimization algorithms are relatively easy to implement and they could conveniently be adapted for different kinds of scheduling problems. This has made the research on them increasingly popular in the recent years.

However, the converging processes of the standard versions of these algorithms tend to be too slow for practical-scale JSSPs. Specifically, in order to find satisfactory solutions, standard SA needs to start with a sufficiently high initial temperature and the cooling process must be slow enough. Only in this way can it be guaranteed that a large enough number of samples are obtained from the whole search space. Likewise, standard GA should employ a reasonably large population and an adequately long evolution process in order to produce enough offspring individuals. Under such a situation, it would be not difficult to observe a severe decline of the performance of these algorithms when they face larger scale JSSP instances. Because of the $\mathcal{NP}$-hard nature of JSSP, the solution space will expand exponentially with the problem size. Finally, the actual limit on the available computational time and the memory size determines that the performance of heuristic local search methods alone will be unsatisfactory. To make the algorithms more powerful, two major sorts of approaches have been adopted and reported in the existing literature:

(1) Great effort has been devoted to the modification and improvement of the local search algorithm itself, independently of the specific characteristics of the pending optimization problem. For example, the probabilistic model-building genetic algorithms (PMBGA) [22,29] produce new individuals by estimating the probability distribution of a selected set of high-quality solutions. This procedure actually replaces the conventional crossover and mutation operators in GA. Also remarkably, many hybrid algorithms have been devised in an attempt to combine the merits of two different local search methods. For example, the genetic local search algorithm presented in [4] makes use of the exploration ability of GA and the exploitation ability of iterated local search (ILS).

* Corresponding author. Tel.: +86 10 62446011.
E-mail address: zhang-r@mails.tsinghua.edu.cn (R. Zhang).

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2009.06.008

(2) Due to the famous "No Free Lunch Theorem" (NFLT) [31] and much recent literature, researchers have gradually realized the importance of mining and utilizing problem-specific or instance-specific information during the optimization process of local search algorithms for hard problems. For example, the shifting bottleneck (SB) heuristic is based on the critical path theory for JSSP with makespan criterion. SB is therefore combined with GA in [6] in order to provide guidance for the searching process of GA. Moreover, the problem knowledge can also be introduced into the initial solution or the initial population, such as in [30], which also proves efficient in accelerating the convergence.

The former approach is apparently independent of problem types, and therefore, such improvement can be directly applied to any optimization problem. However, in the latest research, the latter approach which is built on specific problem characteristics proves to be more successful for scheduling problems. The immune mechanism designed in this work also belongs to the latter approach because immunity is used to embed problem-specific knowledge into the general optimization process.

Among the numerous studies for JSSP, the concept of "bottleneck" has been regarded as a significant type of characteristic information that could be utilized to generate promising solutions [10,19,34]. In this paper, "bottleneck jobs" refer to those jobs whose processing sequence can have an obvious impact on the quality of final schedules. For most scheduling instances, the ultimate scheduling performance could be remarkably improved if these bottleneck jobs have been wisely scheduled. However, how to correctly identify the bottleneck level of each job under different scheduling objectives and in different optimization stages still belong to the unsolved problems which are currently under study in the scheduling community.

In this research, we propose a new bottleneck job identification method which uses a fuzzy inference system to evaluate the bottleneck characteristic value for each job. Then, we design an immune mechanism based on the obtained bottleneck information for improving JSSP solutions. Finally, the immune mechanism is combined with SA to form a hybrid optimization algorithm called immune simulated annealing (ISA). In ISA, the simulated annealing mechanism is used to explore the solution space, while at the same time, the immune mechanism aims at producing high-quality solutions from the current solution. Computational results for problems of different sizes show that ISA is effective and robust.

The rest of the paper is organized as follows. The discussed job shop scheduling problem is formulated in Section 2. Section 3 describes the immune algorithm for JSSP. Section 4 deals with the design and implementation of the hybrid ISA algorithm. The relevant computational results are provided in Section 5. Finally, some concluding remarks are given in Section 6.

2. Problem formulation

JSSP is one of the most frequently applied models in the current studies for production scheduling problems. In a JSSP instance, a set of n jobs (denoted by $\mathcal{J} = \{J_j\}_{j=1}^{n}$) are waiting to be processed on a set of m machines (denoted by $\mathcal{M} = \{M_l\}_{l=1}^{m}$) under the following basic assumptions:

- Machine breakdown does not occur, which means all the machines are continuously available throughout the production stage.
- Preemption of operations is not allowed, which means the processing of an operation cannot be interrupted once started.
- All jobs are released at time $r = 0$.
- The transportation time to deliver relevant jobs between different machines is neglected.
- The setup time for the machines to switch between different jobs is neglected.
- Each machine can process only one job at a time.
- Each job can be processed by only one machine at a time.

In JSSP, each job is associated with a fixed processing route which traverses all the m machines in a predetermined order, which is also known as the technological constraint. The manufacturing procedure to be performed on any one machine is called an operation of the job. Besides, a preset due date and a weight are given for each job. Due date reflects the urgency level of a job, and weight measures the importance of a job to the manufacturer.

For any manufacturing enterprise, it is vitally important to deliver goods in time according to customers' requirements. Hence, due-date-related criteria are usually used with high priority in practical scheduling. In this study, we adopt the total weighted tardiness as the optimization objective. In other words, the scheduling objective considered in this paper is to determine the processing sequence of the operations on each machine such that the total weighted tardiness could be minimized, i.e.,

$$\text{minimize} \sum_{j \in \mathcal{J}} w_j (F_j - d_j)^+,$$

where F_j is the completion time of job j; w_j and d_j respectively denote the weight and the due date of job j; $(x)^+ = \max\{x, 0\}$. This type of JSSP is officially noted as $J//\sum w_j T_j$ according to the three-field notation [12].

3. The immune mechanism for JSSP

3.1. Calculation of bottleneck characteristic values by fuzzy inference

The bottleneck characteristic value (denoted by Γ) is defined in order to describe each job's bottleneck (urgency) level in a quantitative manner. A fuzzy inference system is employed to evaluate all the Γ values. The feature extraction and implementation of the fuzzy inference system is detailed below.

In local search algorithms such as SA, we could obtain the completion time of each job under the current schedule after the current solution is decoded. Therefore, the following indices can be defined and directly calculated:

- The relative distance between job j's completion time and its due date:

  $$g_j = \frac{\tilde{F}_j - d_j}{d_j},$$

  where d_j and $\tilde{F}_j$ respectively denote job j's due date and completion time under the current schedule.
- The relative slack time of job j:

  $$h_j = \frac{d_j - C_{[j]} - \sum_{i \in JS([j])} p_i}{d_j},$$

  where $[j]$ denotes the currently considered operation of job j; $C_{[j]}$ refers to the completion time of the operation and italso equals the release time of $JS([j])$ which is the set of succeeding operations of $[j]$ in job j. Note that h_j corresponds to specific operations of job j and thus it reflects the characteristics of different processing stages of the considered job.