Computers & Industrial Engineering 60 (2011) 699–705

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

An efficient memetic algorithm for solving the job shop scheduling problem

Liang Gao [a], Guohui Zhang [b,*], Liping Zhang [a], Xinyu Li [a]

[a] The State Key Laboratory of Digital Manufacturing Equipment and Technology, 1037 Luoyu Road, Huazhong University of Science & Technology, Wuhan 430074, China
[b] Zhengzhou Institute of Aeronautical Industry Management, Middle Daxue Road, Zhengzhou 450015, China

ARTICLE INFO

Article history:
Received 31 July 2009
Received in revised form 29 December 2010
Accepted 4 January 2011
Available online 26 January 2011

Keywords:
Job shop scheduling
Memetic algorithm
Local search
Neighborhood structure

ABSTRACT

The job shop scheduling problem (JSP) is well known as one of the most complicated combinatorial optimization problems, and it is a NP-hard problem. Memetic algorithm (MA) which combines the global search and local search is a hybrid evolutionary algorithm. In this paper, an efficient MA with a novel local search is proposed to solve the JSP. Within the local search, a systematic change of the neighborhood is carried out to avoid trapping into local optimal. And two neighborhood structures are designed by exchanging and inserting based on the critical path. The objective of minimizing makespan is considered while satisfying a number of hard constraints. The computational results obtained in experiments demonstrate that the efficiency of the proposed MA is significantly superior to the other reported approaches in the literature.

1. Introduction

The job shop scheduling problem is well known as one of the most complicated combinatorial optimization problems (Garey, Johnson, & Sethi, 1976), and it is also well known for its practical applications in many manufacturing industries. A classical JSP is combined with n different jobs and m different machines. Each job consists of a set of operations and each operation requires a different machine. All the operations of each job are processed in a fixed processing order. And each operation has a given processing time. Some assumptions and constraints on jobs and machines are made while solving the JSP. A solution is to determine the operation sequences on the machines to satisfy some constraints. And the objectives usually considered in JSP are the minimization of makespan, the minimization of tardiness, and the minimization of mean flow time, etc. In this paper, the minimization of makespan is the objective. It is defined as the total time between the starting of the first operation and the ending of the last operation in all jobs.

JSP with the objective of minimizing makespan is one of the best known and strongly NP-hard problems (Garey et al., 1976). As a matter of fact, only small size instances among the benchmark problems within the literature could be solved within a reasonable computational time by exact optimization algorithm such as shifting bottleneck procedure (Adams, Balas, & Zawack, 1988), fast taboo search algorithm (Nowicki & Smutnicki, 1996), branch and bound (Akkan & Karabati, 2004) and dynamic programming (Lorigeon, 2002), etc. However, with the problem scale increasing, the computational time of the exact methods grows exponentially. By contrast with the exact methods, the approximate and heuristic algorithms make better tradeoff between solution quality and computational time. These methods mainly include dispatching priority rules (Canbolat & Gundogar, 2004), shifting bottleneck heuristic (Huang & Yin, 2004), lagrangian relaxation (Chen & Luh, 2003). The shifting bottleneck heuristic is one of the successful heuristics for minimizing the makespan in JSP. During the processing, each unscheduled machine is considered as a separate single machine, and the machine that gives the maximum delay in a single job is identified as a bottleneck machine, then the machine is scheduled first. In recent years, much attention has been devoted to the meta-heuristics with emergence of new techniques from the field of artificial intelligence such as tabu search (TS) (Zhang, Li, Guan, & Rao, 2007), simulated annealing (SA) (Kolonko, 1999), greedy randomized adaptive search procedure (GRASP) (Binato, Hery, Loewenstern, & Resende, 2002), ant colony optimization (ACO) (Udomsakdigool & Kachitvichyanukul, 2008), artificial neural network (ANN) (Yang & Wang, 2001), genetic algorithm (GA) (Tsai, Liu, Ho, & Chou, 2008), artificial immune system (AIS) (Luh & Chueh, 2009), particle swarm optimization (PSO) (Sha & Hsu, 2006), memetic algorithm (MA) (Hasan, Sarker, Essam, & Cornforth, 2009) and so on. These meta-heuristic algorithms could be regarded as independent approaches. However, no single algorithm is suitable for solving all kinds of JSP with both a reasonably good enough solution and within a reasonable computational time (Hasan et al., 2009). GA has an efficient exploration ability to search a wide range of search space. However, it does not have an effective local search mechanism for accurately searching near a good solution. So, many researchers improve the performance of GA by incorporating different search and heuristic techniques for solving the JSP.

* Corresponding author.
E-mail address: linghuizhang80@163.com (G. Zhang).

0360-8352/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2011.01.003

In this paper, an efficient memetic algorithm (MA) is proposed to solve the JSP. MA was introduced by Moscato and Norman (1992) to describe evolutionary algorithms in which local search plays a significant part. MA combines the global search and local search by using GA (Holland, 1975) to perform exploration and a local search method to perform exploitation. MA keeps well the balance between intensification and diversification. Radcliffe and Surry (1994) gave a formal description of MA. MA has been applied on many combinatorial optimization problems successfully such as the quadratic assignment problem (QAP) (Merz & Freisleben, 2000), the traveling sales man problem (TSP) (Buriol, Franca, & Moscato, 2004), the partitioning problem of tandem AGV systems (Elmekkawy & Liu, 2009) and so on. Franca, Mendes, and Moscato (2001) proposed a MA for the total tardiness single machine scheduling (SMS) problem with due dates and sequence-dependent setup time. Yang, Sun, Lee, Qian, and Liang (2008) proposed a clonal selection based MA for solving JSP. In the clonal selection mechanism, clonal selection, hypermutation and receptor edit theories are presented to construct an evolutionary searching mechanism. Caumond, Lacomme, and Tchernev (2008) introduced a framework based on a disjunctive graph to model the JSP and use a MA for job sequence generation on machine. Hasan et al. (2009) proposed a MA with three priority rules to improve the performance of traditional MA for solving JSP.

In this research, a novel local search is proposed and combined with GA to improve the MA for solving the JSP. As the above mentioned, GA has an efficient exploration ability to search a wide range of search space, but GA is not well suited for local optimization. Combining the GA and local search, GA is used to perform global exploration among the population, and local search is used to perform local exploitation around chromosomes. Some benchmark problems of JSP were solved by the proposed MA. The computational results show that the MA is effective and efficient through compared with the reported approaches in several literatures.

The remainder of the paper is organized as follows. Section 2 describes the job shop scheduling problem. Section 3 presents the framework based on a memetic algorithm search scheme. Section 4 deals with the computational evaluation of the framework including benchmark problems. Section 5 is the conclusion and addressed several promising research directions.

2. Problem representations

In this research, the JSP consists of a set of jobs Job = $\{J_1, J_2, \ldots, J_n\}$ and a set of machines Machine = $\{M_1, M_2, \ldots, M_m\}$. The objective is to minimize the makespan, i.e., the completion time of the last job being completed in the system. In the JSP, several constraints and assumptions are made as follows:

- Each machine could process at most one job at a time.
- Each job is only processed by one machine at a time.
- The sequence of machines which a job visits is completely fixed and has a linear precedence structure.
- All jobs must be processed by each machine only once and there are at most m operations for a job.
- There are no precedence constraints among the operations of different jobs.
- The machines are always available at zero and never break down.
- Processing time of all operations is known.

3. Job shop scheduling with memetic algorithm

MA could well balance its diversification and intensification to find high quality solutions of the optimization problem. Diversification is a search of different areas of the search space to find the most promising regions. Intensification is a search of the neighborhoods of the individuals to produce better solutions (Ong & Keane, 2004). In the proposed MA, the local search procedure is applied to each child to search for a better solution. The flowchart of the proposed MA in this paper is shown in Fig. 1.

Step 1: Generate initial population. Set parameters of GA including population size, max iteration, mutation probability, crossover probability, etc. Then encode an initial solution into a chromosome. Repeat this step until the number of individual equals to the population size.
Step 2: Apply the local search procedure to improve the quality of each individual.
Step 3: Decode each individual of population to obtain the makespan corresponding with each individual. And compare them to obtain the best solution.
Step 4: Check the termination criteria. If one of the criteria is satisfied, then stop the algorithm and output the best solution; otherwise, go to step 5.
Step 5: Generate new population for the next generation. Genetic evolution with three operators including selection, crossover and mutation is applied to create offspring for the next population. Following this, the algorithm goes back to step 2.

In the following sub-sections, chromosome representation and decoding, initial population, genetic evolution, local search procedure, and termination criteria will be discussed.

3.1. Chromosome representation and decoding

Better efficiency of MA-based search could be achieved by modifying the chromosome representation and its related operators so as to generate feasible solutions and avoid repair mechanism. In order to apply the MA to the JSP, a proper chromosome represen-

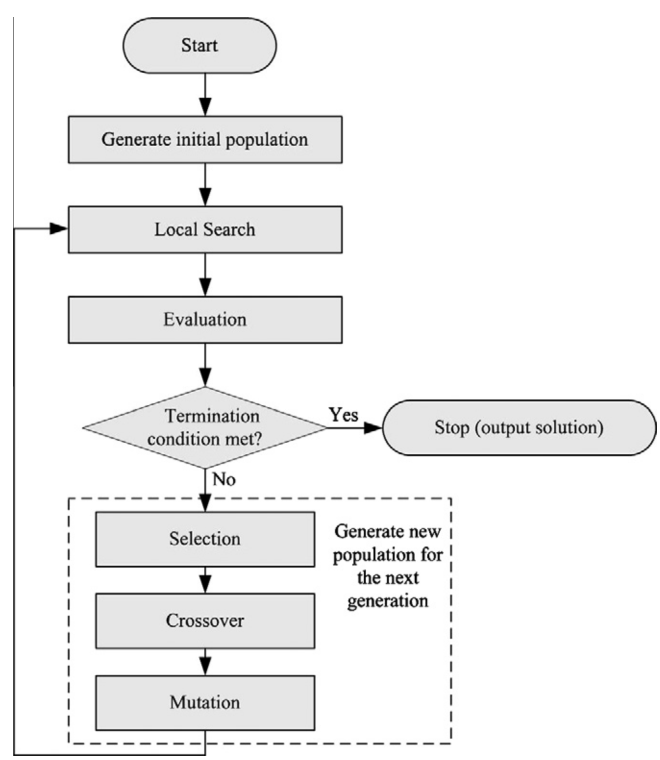

Fig. 1. Flowchart of the proposed MA.