



An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem

Guohui Zhang, Xinyu Shao*, Peigen Li, Liang Gao

The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science & Technology, Wuhan, Hubei Province 430074, China

ARTICLE INFO

Article history:

Received 20 April 2007

Received in revised form 25 July 2008

Accepted 28 July 2008

Available online 6 August 2008

Keywords:

Multi-objective optimization

Flexible job-shop scheduling

Particle swarm optimization

Tabu search

ABSTRACT

Flexible job-shop scheduling problem (FJSP) is an extension of the classical job-shop scheduling problem. Although the traditional optimization algorithms could obtain preferable results in solving the mono-objective FJSP. However, they are very difficult to solve multi-objective FJSP very well. In this paper, a particle swarm optimization (PSO) algorithm and a tabu search (TS) algorithm are combined to solve the multi-objective FJSP with several conflicting and incommensurable objectives. PSO which integrates local search and global search scheme possesses high search efficiency. And, TS is a meta-heuristic which is designed for finding a near optimal solution of combinatorial optimization problems. Through reasonably hybridizing the two optimization algorithms, an effective hybrid approach for the multi-objective FJSP has been proposed. The computational results have proved that the proposed hybrid algorithm is an efficient and effective approach to solve the multi-objective FJSP, especially for the problems on a large scale.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Job-shop scheduling problem (JSP), which is among the hardest combinatorial optimization problems (Sonmez & Baykasoglu, 1998), is a branch of production scheduling. It is well known that this problem is NP-hard (Garey, Johnson, & Sethi, 1976). The classical JSP schedules a set of jobs on a set of machines with the objective to minimize a certain criterion, subjected to the constraint that each job has a specified processing order through all machines which are fixed and known in advance.

Flexible job-shop problem (FJSP) is an extension of the classical JSP that allows one operation which can be processed on one machine out of a set of alternative machines. It is closer to the real manufacturing situation. Because of the additional needs to determine the assignment of operations on the machines, FJSP is more complex than JSP, and incorporates all the difficulties and complexities of JSP. The problem of scheduling jobs in FJSP could be decomposed into two sub-problems: a routing sub-problem, which is assigning each operation to a machine out of a set of capable machines and a scheduling sub-problem, which is sequencing the assigned operations on all selected machines in order to obtain a feasible schedule with optimized objectives (Xia & Wu, 2005). The scheduling problem in a flexible job shop is at least as hard as the job shop problem, which is considered one of the most difficult problems in combinatorial optimization (Lawler, Lenstra, Rinnooy Kan, & Shmoys, 1993). Although an optimal solution algo-

rihm for the classical JSP has not been developed yet, there is a trend in the research community to model and solve much more complex version of the FJSP (Baykasoglu, Ozbakir, & Sonmez, 2004).

The research on the multi-objective FJSP is much less than the mono-objective FJSP. Brandimarte (1993) was the first to apply the decomposition approach into the FJSP. He solved the routing sub-problem by using some existing dispatching rules and then focused on the scheduling sub-problem, which was solved by using a tabu search heuristic. Hurink, Jurisch, and Thole (1994) proposed a tabu search heuristic. They considered the reassignment and rescheduling as two different types of moves. Mastrolilli and Gambardella (2000) proposed some neighborhood functions for the FJSP, which could be used in meta-heuristic optimization techniques. The most important issue in employing meta-heuristics for combinatorial optimization problems is to develop an effective “problem mapping” and “solution generation” mechanism. If these two mechanisms are devised successfully, it is possible to find good solutions to the given optimization problem in acceptable time. Parsopoulos and Vrahatis (2002) conducted the first research on the particle swarm optimization method in multi-objective optimization problems. Kacem, Hammadi, and Borne (2002a, 2002b) proposed a genetic algorithm controlled by the assigned model which was generated by the approach of localization (AL) to mono-objective and multi-objective FJSP. They used the integrated approach considering assignment and scheduling at the same time. Rigao (2004) developed two heuristics based on tabu search: a hierarchical procedure and a multiple start procedure. Recently, Xia and Wu (2005) proposed

* Corresponding author.

E-mail address: shaoxinyu68@163.com (X. Shao).

a hybrid algorithm using particle swarm optimization (PSO) assignment and simulated annealing (SA) scheduling to optimize multi-objective FJSP, respectively. To our knowledge, research on multi-objective FJSP is rather limited, and most traditional optimal approaches used only one optimization algorithm for solving multi-objective FJSP.

In this paper, the search mechanism of the particle swarm optimization and tabu search is taken full advantage of. An effective solution approach is proposed for solving multi-objective FJSP. The proposed approach uses PSO to assign operations on machines and to schedule operations on each machine, and TS is applied to local search for the scheduling sub-problem originating from each obtained solution. The objectives which are considered in this paper are to minimize maximal completion time, the workload of the critical machine and the total workload of machines simultaneously.

The remainder of this paper is organized as follows. The formulation and notion of multi-objective FJSP are introduced in Section 2. Section 3 describes the traditional PSO algorithm and summarizes the local search scheme by TS, and then the model of the proposed hybrid algorithm (PSO + TS) is given. At the same time, how to apply it to solve the multi-objective FJSP is described. Section 4 shows the four representative instances of multi-objective FJSP and the corresponding computational results with the proposed hybrid algorithm. Then, the computational results are compared with other heuristic algorithms in this section. Finally, concluding remarks and further research are given in Section 5.

2. Problem formulation

A general multi-objective minimization problem, which could be defined as: minimize a function $f(x)$, with P ($P > 1$) decision variables and Q objectives ($Q > 1$), subject to several equality or inequality constraints. More precise definitions of the terms about the multi-objective optimization can be found in Deb (2001).

$$\text{Minimize } f(x) = (f_1(x), f_2(x), \dots, f_q(x), \dots, f_Q(x)) \quad (1)$$

where Ω is the feasible solution space, $x = \{x_1, x_2, \dots, x_p, \dots, x_P\}$ is the set of p -dimensional decision variables (continuous, discrete or integer) ($1 \leq p \leq P$), i.e., a possible solution for the considered problem; $f_q(x)$ is the q th objective function ($1 \leq q \leq Q$). It is obvious that it does not exist an exact solution to such a problem. However, the multi-objective optimization problem should optimize the different objective functions simultaneously.

The FJSP could be formulated as follows. There is a set of n jobs and a set of m machines. M denotes the set of all machines. Each job i consists of a sequence of n_i operations. Each operation O_{ij} ($i = 1, 2, \dots, n; j = 1, 2, \dots, n_i$) of job i has to be processed on one machine M_k out of a set of given compatible machines M_{ij} (for $M_k \in M_{ij}, M_{ij} \subseteq M$). The execution of each operation requires one machine selected from a set of available machines. In addition, the FJSP needs to set each operation's starting and ending time. The FJSP is needed to determine both an assignment and a sequence of the operations on the machines in order to satisfy the given criteria. If there is $M_{ij} \subset M$ for at least one operation, it is partial flexibility FJSP (P-FJSP); while there is $M_{ij} = M$ for each operation, it is total flexibility FJSP (T-FJSP) (Kacem et al., 2002a, 2002b).

In this paper, the following criteria are to be minimized:

- (1) the maximal completion time of machines, i.e., the makespan;
- (2) the maximal machine workload, i.e., the maximum working time spent on any machine;
- (3) the total workload of machines, which represents the total working time over all machines.

During the process of solving this problem, the following assumptions are made:

- (1) Each operation cannot be interrupted during its performance (non-preemptive condition);
- (2) Each machine can perform at most one operation at any time (resource constraint);
- (3) The precedence constraints of the operations in a job can be defined for any pair of operations;
- (4) Setting up time of machines and move time between operations are negligible;
- (5) Machines are independent from each other;
- (6) Jobs are independent from each other.

The notation used in this paper is summarized in the following:

- n : total number of jobs;
- m : total number of machines;
- n_i : total number of operation of job i ;
- O_{ij} : the j th operation of job i ;
- M_{ij} : the set of available machines for the operation O_{ij} ;
- P_{ijk} : processing time of O_{ij} on machine k ;
- t_{ijk} : start time of operation O_{ij} on machine k ;
- C_{ij} : completion time of the operation O_{ij} ;
- i, h : index of jobs, $i, h = 1, 2, \dots, n$;
- k : index of machines, $k = 1, 2, \dots, m$;
- j, g : index of operation sequence, $j, g = 1, 2, \dots, n_i$;
- C_k is the completion time of M_k ;
- W_k is the workload of M_k .

$$x_{ijk} = \begin{cases} 1, & \text{if machine } k \text{ is selected for the operation } O_{ij} \\ 0, & \text{otherwise} \end{cases}$$

Our model is then given as follows:

$$\min f_1 = \max_{1 \leq k \leq m} (C_k) \quad (2)$$

$$\min f_2 = \max_{1 \leq k \leq m} (W_k) \quad (3)$$

$$\min f_3 = \sum_{k=1}^m W_k \quad (4)$$

$$\text{s.t. } C_{ij} - C_{i(j-1)} \geq P_{ijk} x_{ijk}, j = 2, \dots, n_i; \forall i, j \quad (5)$$

$$[(C_{hg} - C_{ij} - t_{hgk}) x_{hgk} x_{ijk} \geq 0] \vee [(C_{ij} - C_{hg} - t_{ijk}) x_{hgi} x_{ijk} \geq 0], \quad (6)$$

$$\sum_{k \in M_{ij}} x_{ijk} = 1, \forall i, j \quad (7)$$

Inequity (5) ensures the operation precedence constraint. Inequity (6) ensures each machine could process only one operation at each time. Equation (7) states that one machine could be selected from the set of available machines for each operation.

Many studies have been devoted to the subject of multi-objective meta-heuristic optimization and the developed methods to solve multi-objective optimization can be generally classified into three different types (Hsu, Dupas, Jolly, & Goncalves, 2002):

- (1) The transformation towards a mono-objective problem consists of combining the different objectives into a weighted sum. Methods in this class are based on utility functions or E-Constraint and goal programming;
- (2) The non-Pareto approach utilizes operators for processing the different objectives in a separated way;
- (3) The Pareto approaches are directly based on the Pareto optimality concept. They aim at satisfying two goals: converging towards the Pareto front and also obtaining diversified solutions scattered all over the Pareto front.

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات