Computers & Industrial Engineering 59 (2010) 647–662

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

An effective hybrid tabu search algorithm for multi-objective flexible job-shop scheduling problems

Jun-qing Li [a], Quan-ke Pan [a,b], Yun-Chia Liang [c,*]

[a] College of Computer Science, Liaocheng University, Liaocheng 252059, People's Republic of China
[b] State Key Lab. of Digital Manufacturing Equipment & Technology in Huazhong University of Science & Technology, Wuhan 430074, People's Republic of China
[c] Department of Industrial Engineering and Management, Yuan Ze University, Yuan-Tung Road, Chung-Li 32003, Taiwan, ROC

ARTICLE INFO

Article history:
Received 10 November 2009
Received in revised form 2 May 2010
Accepted 16 July 2010
Available online 21 July 2010

Keywords:
Flexible job-shop scheduling problem
Multi-objective optimization
Variable neighborhood search
Public critical block
Tabu search

ABSTRACT

This paper proposes an effective hybrid tabu search algorithm (HTSA) to solve the flexible job-shop scheduling problem. Three minimization objectives – the maximum completion time (makespan), the total workload of machines and the workload of the critical machine are considered simultaneously. In this study, a tabu search (TS) algorithm with an effective neighborhood structure combining two adaptive rules is developed, which constructs improved local search in the machine assignment module. Then, a well-designed left-shift decoding function is defined to transform a solution to an active schedule. In addition, a variable neighborhood search (VNS) algorithm integrating three insert and swap neighborhood structures based on public critical block theory is presented to perform local search in the operation scheduling component. The proposed HTSA is tested on sets of the well-known benchmark instances. The statistical analysis of performance comparisons shows that the proposed HTSA is superior to four existing algorithms including the AL + CGA algorithm by Kacem, Hammadi, and Borne (2002b), the PSO + SA algorithm by Xia and Wu (2005), the PSO + TS algorithm by Zhang, Shao, Li, and Gao (2009), and the Xing's algorithm by Xing, Chen, and Yang (2009a) in terms of both solution quality and efficiency.

1. Introduction

As a branch of production scheduling problems, the classical job-shop scheduling problem (JSP) has been proven to be an NP-hard problem (Garey, Johnson, & Sethi, 1976). The flexible job-shop scheduling problem (FJSP) is an extension of the classical JSP, which allows one operation to be processed on one machine from a set of alternative machines. Therefore, FJSP is more complex than JSP because of the addition need to determine the assignment of machines for each operation. Flexibility of the FJSP can be generally categorized into partial flexibility and total flexibility. If each operation can be processed on any machine in the system, we have a total FJSP (T-FJSP); otherwise flexibility would be partial (P-FJSP).

The FJSP has recently captured the interests of many researchers. Due to the complexity of the FJSP, meta-heuristic algorithms have become a practical alternative of solving techniques. Brandimarte (1993) proposed a hybrid tabu search algorithm with some existing dispatching rules to solve the single-objective FJSP. Mastrolilli and Gambardella (2000) used local search techniques and developed two neighborhood functions for the problem. The experimental results of Mastrolilli and Gambardella (2000) are considered to be the best results for FJSP by tabu search approach so far (Pezzella, Morganti, & Ciaschetti, 2008). Gao, Peng, Zhou, and Li (2006) presented a general particle swarm optimization (PSO) algorithm to solve the FJSP. Liouane, Saad, Hammadi, and Borne (2007) solved the problem with a hybrid algorithm combining ant colony optimization (ACO) with TS. Saidi-mehrabad and Fattahi (2007) developed an improved TS algorithm for FJSP. Gao, Sun, and Gen (2008) proposed a hybrid genetic algorithm (GA) with a variable neighborhood descent (VND) algorithm to deal with the FJSP while Pezzella et al. (2008) introduced a GA integrating different strategies to generate the initial population for the FJSP.

Although the single-objective FJSP has been widely investigated, the research on the multi-objective FJSP is still considered limited. Kacem, Hammadi, and Borne (2002a, 2002b) developed an effective evolutionary algorithm controlled by an assigned model based on the approach of localization (AL). Xia and Wu (2005) presented a hierarchical solution approach by using a PSO algorithm to assign operations on machines and a simulated annealing (SA) algorithm to schedule operations on each machine. An algorithm hybridized with evolving dispatching rules and genetic programming was proposed by Tay and Ho (2008). Xing et al. (2009a) gave a simulation model for solving multi-objective FJSP, which is a general framework for the problem and easy to

* Corresponding author. Tel.: +886 3 4638800; fax: +886 3 4638907.
E-mail addresses: lijunqing.cn@gmail.com (J.-q. Li), panquanke@gmail.com (Q.-k. Pan), ycliang@saturn.yzu.edu.tw (Y.-C. Liang).

0360-8352/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2010.07.014

apply different algorithms for it. A local search algorithm was introduced by Xing, Chen, and Yang (2009b) for solving multi-objective T-FJSPs and P-FJSPs. The experimental results show that Xing's algorithms (Xing et al., 2009a, 2009b) are efficient especially for solving large scale T-FJSP problems. Lastly, Zhang et al. (2009) introduced a hybrid PSO and TS algorithm to solve the multi-objective FJSP.

In this paper, we propose a hybrid tabu search algorithm (HTSA) for the multi-objective flexible job-shop scheduling problem. In the proposed HTSA, two adaptive neighborhood search rules are presented for performing local search in the machine assignment module. Meanwhile, three insert and swap neighborhood structures based on the public critical blocks theory are developed to produce optimal adjustment in the operation scheduling module. In addition, an efficient left-shift function is designed to decode a solution to an active schedule.

The rest of this paper is organized as follows: In Section 2, we briefly describe the problem formulation. Then, the framework of the hybrid algorithm is presented in Section 3. The TS algorithm for the machine assignment component is shown in Section 4 while Section 5 introduces the variable neighborhood search algorithm for the operation scheduling component. Section 6 shows the experimental results and compares with other algorithms in the literature to demonstrate the superiority of the HTSA performance. Finally, the last section presents conclusions of our work.

2. Problem formulation

The FJSP problem is divided into two closely related sub-problems. The first one is to assign a proper machine from a set of candidate machines to perform each operation, which can be considered as a parallel machine problem; the second one is to sequence each operation on each machine, which is equivalent to a classical job-shop scheduling problem. Both of them were proven to be NP-hard problems (Garey et al., 1976). The notations and assumptions of the FJSP are summarized as follows:

- Let $J = \{J_i\}_{1 \leqslant i \leqslant n}$, indexed i, be a set of n jobs to be scheduled. q_i denotes total number of operations of job J_i.
- Let $M = \{M_k\}_{1 \leqslant k \leqslant m}$, indexed k, be a set of m machines.
- Each job J_i consists of a predetermined sequence of operations. Let $O_{i,h}$ be operation h of J_i.
- Each operation $O_{i,h}$ can be processed without interruption on one of a set of machines $M(O_{i,h})$. Let $p_{i,h,k}$ be the processing time of $O_{i,h}$ on machine M_k.
- Each machine can process only one operation at a time.
- Two categories of the FJSP are dealt: T-FJSP and P-FJSP.

$$FJSP = \begin{cases} T-FJSP, & \text{if } M(O_{i,h}) = M; \quad \forall i,h \\ P-FJSP, & \text{if } M(O_{i,h}) \subset M; \quad \forall i,h \end{cases}$$

- Decision variables

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

$c_{i,h}$: completion time of the operation $O_{i,h}$.

- Three objectives to minimize
  (1) Makespan denoted by c_M, which is the maximal completion time of machines.
  (2) Total workload of machines denoted by w_T, which is the total working time of all machines in the system. This objective is of interest assigning the machine with relative small processing time for each operation to improve economic efficiency.
  (3) Critical machine workload denoted by w_M, which is the machine with the biggest workload. This objective considers the workload balance among all machines to prevent too much work been assigned to a single machine.

Lastly, the formulation of the multi-objective FJSP in this study is given in Fig. 1.

Eq. (4) ensures the operation precedence constraints. Eq. (5) states that one machine must be selected from the set of available machines for each operation. Eq. (6) ensures the set of available machines for each operation comes from the given machine set M.

Many approaches have been developed to solve the multi-objective optimization. These approaches can be classified into three categories (Hsu, Dupas, Jolly, & Goncalves, 2002).

(1) Transform the multi-objective problem to a mono-objective problem by assigning different weight coefficient for each objective.

$$\min c_M = \max_{1 \le i \le n} \{c_{i,q_i}\} \tag{1}$$

$$\min w_T = \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{h=1}^{q_i} p_{i,h,k} x_{i,h,k} \tag{2}$$

$$\min w_M = \max_{1 \le k \le m} \{\sum_{i=1}^{n} \sum_{h=1}^{q_i} p_{i,h,k} x_{i,h,k}\} \tag{3}$$

$$\text{s.t.} \quad c_{i,h} - c_{i,h-1} \ge p_{i,h,k} x_{i,h,k}, \quad h = 2,....,q_i; \forall i,k \tag{4}$$

$$\sum_{k \in M(O_{i,h})} x_{i,h,k} = 1, \forall i,h \tag{5}$$

$$M(O_{i,h}) \subseteq M, \forall i,h \tag{6}$$

$$x_{i,h,k} \in \{0,1\}, \forall i,h,k \tag{7}$$

$$c_{i,h} \ge 0, \forall i,h \tag{8}$$

Fig. 1. Problem formulation of the multi-objective FJSP.