Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 53 (2007) 149–162
computers & industrial engineering
www.elsevier.com/locate/dsw

A hybrid of genetic algorithm and bottleneck shifting for multiobjective flexible job shop scheduling problems ☆

Jie Gao [a,*], Mitsuo Gen [b], Linyan Sun [a], Xiaohui Zhao [c]

[a] *School of Management, Xi'an Jiaotong University, Xi'an 710049, China*
[b] *Graduate School of Information, Production & Systems, Waseda University, Kitakyushu 808-0135, Japan*
[c] *School of Mechanical & Electrical Engineering, Xi'an Polytechnic University, Xi'an 710048, China*

Received 3 November 2006; received in revised form 29 April 2007; accepted 30 April 2007
Available online 6 May 2007

Abstract

Flexible job shop scheduling problem (fJSP) is an extension of the classical job shop scheduling problem, which provides a closer approximation to real scheduling problems. This paper addresses the fJSP problem with three objectives: min makespan, min maximal machine workload and min total workload. We develop a new genetic algorithm hybridized with an innovative local search procedure (bottleneck shifting) for the problem. The genetic algorithm uses two representation methods to depict solution candidates of the fJSP problem. Advanced crossover and mutation operators are proposed to adapt to the special chromosome structures and the characteristics of the problem. The bottleneck shifting works over two kinds of effective neighborhood, which use interchange of operation sequences and assignment of new machines for operations on the critical path. In order to strengthen search ability, the neighborhood structure can be adjusted dynamically in the local search procedure. The performance of the proposed method is tested by numerical experiments on a large number of representative problems.

Keywords: Flexible job shop scheduling problem; Genetic algorithm; Bottleneck shifting; Neighborhood structure

1. Introduction

Flexible job shop is a generalization of the job shop and the parallel machine environment, which provides a closer approximation to a wide range of real manufacturing systems. In particular, there are a set of work centers in a flexible job shop. Each work center has a set of parallel machines with possibly different efficiency. An operation can be performed by any machine in a work center. Consequently, this results in two problems. The first one is the routing problem (i.e., the assignment of operations to machines), and the second one is the scheduling problem (i.e., determining the starting time of each operation). The combination of the two

☆ This manuscript was processed by Area Editor Maged M. Dessouky.

* Corresponding author. Tel./fax: +86 29 82668700.
E-mail addresses: calebgao@yahoo.com (J. Gao), gen@waseda.jp (M. Gen), lysun@mail.xjtu.edu.cn (L. Sun), xhuizhao@263.net (X. Zhao).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2007.04.010

decisions presents additional complexity and a new problem called flexible job shop scheduling problem (fJSP). The fJSP problem is NP-hard since it is an extension of the job shop scheduling problem (JSP) that has been proven to be NP-hard (Garey, Johmson, & Sethi, 1976).

Bruker and Schlie (1990) developed a polynomial algorithm for solving the flexible job shop scheduling problem with two jobs. Chambers (1996) developed a tabu search algorithm to solve the problem. Mastrolilli and Gambardella (2000) proposed two neighborhood functions for the fJSP problem. Yang (2001) presented a new genetic algorithm (GA)-based discrete dynamic programming approach. Kacem, Hammadi, and Borne (2002a) proposed the approach by localization to solve the resource assignment problem, and an evolutionary approach controlled by the assignment model for the fJSP problem. Wu and Weng (2005) considered the problem with job earliness and tardiness objectives, and proposed a multiagent scheduling method. Xia and Wu (2005) treated this problem with a hybrid of particle swarm optimization and simulated annealing as a local search algorithm. Zhang and Gen (2005) proposed a multistage operation-based genetic algorithm to deal with the fJSP problem from a point view of dynamic programming.

In this paper, a hybrid genetic algorithm (hGA) is used to solve the fJSP problem. The genetic algorithm uses two representation methods. One is used in initialization and mutation, and the other is used for crossover operation. In order to strengthen the search ability, bottleneck shifting serves as a kind of local search method under the framework of GA. The local search only investigates neighbor solutions that have possibilities to improve the incumbent one.

The fJSP problem is described in Section 2. Section 3 presents the representation method, decoding procedure and genetic operators of the proposed GA. The details of the bottleneck shifting are presented in Section 4. In Section 5, we present computational study on a number of well-known fJSP benchmark problems and compare our results with those obtained by previous authors. Some final concluding remarks and future research directions are given in Section 6.

2. The flexible job shop scheduling problem

In the job shop scheduling problem (JSP), there are n jobs that must be processed on a group of m machines. Each job i consists of a sequence of m operations $(o_{i1}, o_{i2}, \ldots, o_{im})$, where o_{ik} (the kth operation of job i) must be processed without interruption on a predefined machine m_{ik} during p_{ik} time units. The operations $o_{i1}, o_{i2}, \ldots, o_{im}$ must be processed one after another in the given order and each machine can process at most one operation at a time.

In a flexible job shop, each job i consists of a sequence of n_i operations $(o_{i1}, o_{i2}, \ldots, o_{in_i})$, and each operation o_{ik} can be processed on any machine out of a set A_{ik} of given machines. The processing time of operation o_{ik} on machine j is $p_{ikj} > 0$. The scheduling problem is to choose for each operation o_{ik} a machine $M(o_{ik}) \in A_{ik}$ and a starting time s_{ik} at which the operation must be performed. In this study, we consider to minimize the following three criteria:

(1) Makespan (c_{M}) of the jobs, which represents the completion time of all the jobs.
(2) Maximal machine workload (w_{M}), i.e., the maximum working time spent at any machine. This objective is to prevent a solution from assigning too much work on a single machine and to keep the balance of work distribution over the machines.
(3) Total workload (w_{T}), which represents the total working time over all machines. This objective is of interest if machine efficiencies differ.

3. Hybrid genetic algorithm for fJSP

3.1. Two-vector Gen et al.'s representation

The GA's structure and parameter setting affect its performance. However, the primary determinants of a GA's success or failure are the coding by which its genotypes represent candidate solutions and the interaction of the coding with the GA's recombination and mutation operators.