Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 2892–2907
computers & operations research
www.elsevier.com/locate/cor

A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems

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

[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

Available online 2 February 2007

Abstract

This paper addresses the flexible job shop scheduling problem (fJSP) with three objectives: min makespan, min maximal machine workload and min total workload. We developed a hybrid genetic algorithm (GA) for the problem. The GA uses two vectors to represent solutions. Advanced crossover and mutation operators are used to adapt to the special chromosome structure and the characteristics of the problem. In order to strengthen the search ability, individuals of GA are first improved by a variable neighborhood descent (VND), which involves two local search procedures: local search of moving one operation and local search of moving two operations. Moving an operation is to delete the operation, find an assignable time interval for it, and allocate it in the assignable interval. We developed an efficient method to find assignable time intervals for the deleted operations based on the concept of earliest and latest event time. The local optima of moving one operation are further improved by moving two operations simultaneously. An extensive computational study on 181 benchmark problems shows the performance of our approach.

Keywords: Flexible job shop scheduling; Genetic algorithms; Variable neighborhood descent; Critical path

1. Introduction

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} for 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 this paper we study a generalization of JSP called the flexible job shop scheduling problem (fJSP), which provides a closer approximation to a wide range of scheduling problems encountered in real manufacturing systems.

In a flexible job shop, each job i consists of a sequence of n_i operations $(o_{i1}, o_{i2}, \ldots, o_{in_i})$. The fJSP extends JSP by allowing an operation o_{ik} to be executed by one 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 fJSP 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.

* Corresponding author. Tel./fax: +81 93 692 5273.
E-mail addresses: calebgao@yahoo.com (J. Gao), lysun@mail.xjtu.edu.cn (L. Sun), gen@waseda.jp (M. Gen).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2007.01.001

fJSP is therefore made more complex than JSP by the need to determine a routing policy (i.e., which machine to assign for each operation) other than the traditional scheduling decisions (i.e., to determine the starting time of each operation). In this study, we consider minimizing the following three criteria:

(1) Minimize the makespan (C_M) of the jobs.
(2) Minimize the 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) Minimize the total workload (W_T), which represents the total working time assigned over all machines. This objective is of interest if machine efficiencies differ.

In contrary to those multiobjective optimization problems in which there are no priorities among the objectives, makespan is given the first importance, maximal machine workload is given the secondary importance, and total workload is given the least importance in this study. When two solutions with different makespans are compared, we always prefer the solution with a smaller makespan regardless of its maximal machine workload and total workload.

In this paper, a hybrid genetic and variable neighborhood descent (VND) algorithm is used to treat the fJSP problem. The genetic algorithm (GA) uses two vectors to express the machine assignment and operation sequence information for fJSP solution candidates. The operation sequence vector uses two representation methods: Gen et al.'s [1] representation and permutation representation. Under the framework of the GA, a VND is applied to each newly generated offspring to improve its quality before injecting it into the population. The VND improves a solution first by moving one operation. When the local optimum of moving one critical operation is found, it is further improved by moving two operations simultaneously.

In Section 2, an overview of previous research is provided. Section 3 presents the representation method, decoding procedure and genetic operators of the GA. The VND and the framework of the hybrid GA are presented in Section 4. In Section 5, we provide an extensive computational study on 181 benchmark problems where our approach is compared with previous approaches. Some final concluding remarks are given in Section 6.

2. Literature review

Bruker and Schlie [2] gave a polynomial algorithm for solving flexible job shop scheduling problems with two jobs. Based on the observation that fJSP turns into the classical job shop scheduling problem when a routing is chosen, early literature [3–6] proposed hierarchical strategies for the complex scheduling problem, where job routing and sequencing are studied separately.

Jurisch [7] considered the routing and scheduling decisions simultaneously. Hurink et al. [8] and Chambers [9] developed tabu search algorithms to solve the problem. Dauzère-Pérès and Paulli [10] extended the classical disjunctive graph model for job shop scheduling to take into account the fact that operations have to be assigned to machines in the fJSP problem. Based on the extended disjunctive graph, a new neighborhood structure is defined and a tabu search procedure is provided to solve the problem. Mastrolilli and Gambardella [11] proposed two neighborhood functions for this problem. They proposed a tabu search procedure and provided an extensive computational study on 178 fJSP problems and 43 JSP problems. Their approach found 120 new better upper bounds and 77 optimal solutions over the 178 fJSP benchmark problems and it was outperformed in only one problem instance.

Yang [12] presented a genetic algorithm (GA)-based discrete dynamic programming approach. Kacem and Borne [13] proposed a localization approach to solve the resource assignment problem, and an evolutionary approach controlled by the assignment model for the fJSP problem. Zhang and Gen [14] proposed a multistage operation-based GA to deal with the problem from a point view of dynamic programming. Xia and Wu [15] treated this problem with a hybrid of particle swarm optimization (PSO) and simulated annealing (SA) as a local search algorithm. Wu and Weng [16] considered the problem with job earliness and tardiness objectives, and proposed a multiagent scheduling method.

Vaessens [17] defined a neighbor for the fJSP problem by deleting an operation r from the machine ordering, identifying a set of feasible insertions containing the optimal one and inserting r in the best-possible way. Vaessens' algorithm spends a considerable amount of computing time defining such a set of feasible insertions. Brucker and Neyer [18] also proposed the best insertion of an operation in their neighborhood function, but the algorithm they suggested is too time consuming. In order to reduce the computational effort, they proposed a faster algorithm that guarantees only