Applied Soft Computing 9 (2009) 362–376

Contents lists available at ScienceDirect

Applied Soft Computing

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

Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling

Li-Ning Xing *, Ying-Wu Chen, Ke-Wei Yang

Department of Management Science & Engineering, College of Information System and Management, National University of Defense Technology, Changsha 410073, PR China

ARTICLE INFO

Article history:
Received 30 June 2007
Received in revised form 4 April 2008
Accepted 20 April 2008
Available online 26 April 2008

Keywords:
Simulation modeling
Combinatorial optimization
Flexible job shop scheduling
Ant colony optimization
Multi-objective optimization
Scheduling knowledge

ABSTRACT

Flexible job shop schedule is very important in both fields of combinatorial optimization and production management. In this paper, a simulation model is presented to solve the multi-objective flexible job shop scheduling problem. The proposed model has been coded by Matlab which is a special mathematical computation language. After modeling the pending problem, the model is validated by five representative instances based on practical data. The results obtained from the computational study have shown that the proposed approach is a feasible and effective approach for the multi-objective flexible job shop scheduling problem.

1. Introduction

The job shop scheduling problem (JSSP) is generally defined as decision-making problems with the aim of optimizing one or more scheduling criteria. JSSP is a branch of production scheduling which is among the hardest combinatorial optimization problems. Many different approaches have been successfully applied to JSSP, such as, simulated annealing (SA [1,2]), tabu search (TS [3,4]), genetic algorithm (GA [5–9]), ant colony optimization (ACO [10,11]), neural networks (NN [12,13]), evolutionary algorithm (EA [14,15]) and other heuristic approach [16–20].

Flexible job shop scheduling problem (FJSSP) is an extension of the classical JSSP which allows an operation to be processed by any machine from a given set. It is more complex than JSSP because of the addition need to determine the assignment of operations to machines. Bruker and Schlie [21] were among the first to address this problem. For solving the realistic case with more than two jobs, two types of approaches have been employed: hierarchical approaches and integrated approaches [22].

In hierarchical approaches, assignment of operations to machines and sequencing of operations on machines are treated separately. Brandimarte [23] was the first to use the decomposition for the FJSSP. He solved the routing sub-problem using some existing dispatching rules and then solved the scheduling sub-problem using a TS heuristic. Tung et al. [24] developed a similar approach for scheduling a flexible manufacturing system. Recently, Kacem et al. [25,26] proposed a GA controlled by the assigned model which is generated by the approach of localization. Xia and Wu [22] makes use of particle swarm optimization (PSO) to assign operations on machines and SA algorithm to schedule operations.

Integrated approaches were used by considering assignment and scheduling at the same time. Hurink et al. [27] proposed a TS heuristic in which reassignment and rescheduling are considered as two different types of moves. The integrated approach presented by Dauzere-Peres and Paulli [28] was defined a neighborhood structure for the problem where there is no distinction between reassigning and resequencing an operation, and the TS procedure is proposed based on the neighborhood structure. Mastrolilli and Gambardella [29] improved Dauzere-Peres' TS techniques and presented two neighborhood functions. Most researchers were interested in applying TS and GA techniques to FJSSP in the past [22].

A simulation model is proposed in our work for the FJSSP with multi-objectives of minimizing makespan, total workload of machines and workload of the critical machine. The remainder of this paper is organized as follows. The notation and problem formulation are introduced in Section 2. Section 3 describes the proposed simulation modeling. In Section 4, we present some improvements for the simulation model. In Section 5, computational experiments performed with our approach for some representative instances of FJSSP are reported followed by the

* Corresponding author.
E-mail address: xinglining@gmail.com (L.-N. Xing).

1568-4946/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2008.04.013

comparison to other heuristic methods. Some concluding remarks are made in Section 6.

2. Problem formulation

The flexible job shop scheduling problem can be formulated as follows:

(1) There is a set of n jobs that plan to process on m machines.
(2) The set machine is noted M, $M = \{M_1, M_2, \ldots, M_m\}$.
(3) Each job i consists of a sequence of n_i operations $O_{i1}, O_{i2}, \ldots, O_{in_i}$;
(4) The execution of operation O_{ij} requires one machine out of a set of given machines called $M_{ij} \subseteq M$.

FJSSP is to determine an assignment and a sequence of operations on machines to minimize

(1) Makespan or maximal completion time of machines.
(2) Total workload of machines, which represents the total working time of all machines.
(3) Critical machine workload, which is the machine with the biggest workload.

The weighted sum of the above three objective values is taken as the objective function:

$$F(c) = 0.5 \times F_1(c) + 0.2 \times F_2(c) + 0.3 \times F_3(c) \quad (1)$$

where $F(c)$ denotes the objective value of schedule c, $F_1(c)$, $F_2(c)$ and $F_3(c)$ denote the makespan, total workload of machines and workload of critical machine of schedule c, respectively. The weight of different objectives is determined by the empirical experience. If the decision maker pays more attention to a certain objective, then we can define a large weight to it. Otherwise, we can define a small weight to the given objective. There are three advantages of using weighted sum approach to deal with the multi-objective optimization. At first, it is easy for decision makers to understand the weighted sum method. At second, it is convenient for developers to implement the weighted sum approach. At third, it is available to modify the weight of different objectives for satisfying the requirement of decision makers. In our work, the importance of objectives $F_1(c)$, $F_2(c)$ and $F_3(c)$ are considered as 'most important', 'important' and 'more important', so the weight of them is defined as 0.5, 0.2 and 0.3, respectively.

Hypotheses considered in this paper are listed as follows:

(1) Jobs are independent from each other.
(2) Machines are independent from each other.
(3) Setting up time of machines is negligible.
(4) Move time between operations is negligible.
(5) At a given time, a machine can execute at most one operation.
(6) No more than one operation of the same job can be executed at a time.
(7) There are no precedence constraints among the operations of different jobs.

3. Simulation modeling of FJSSP

The first step in creating a simulation model must be to define its purpose [30]. In our case, the short-term goal of this simulation model was to use it for assigning the operations to machines and scheduling the operations on each machine. That is, it can obtain an optimal or near-optimal production schedule in the flexible manufacturing environment using this simulation model. The long-term goal of this simulation model was to use it for generating, selection, and analysis of alternative production schedules and to discover some available schedule knowledge for assisting production scheduling decision.

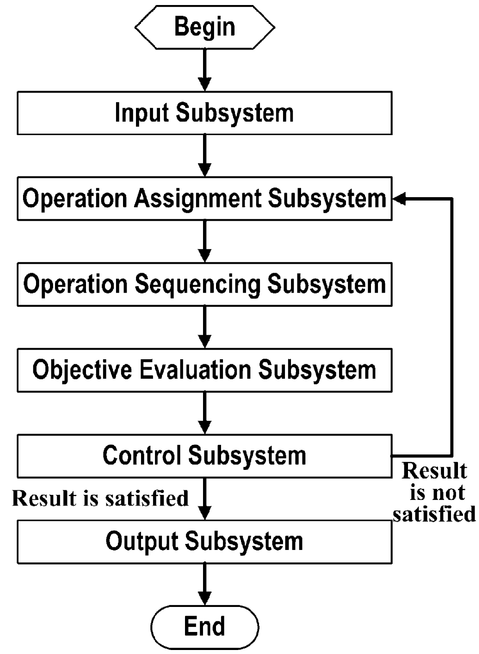

Fig. 1. Framework of our proposed simulation model.

Then a suitable simulation language should be chosen and definition of the entities used in that language can be undertaken [31]. In our work, the simulation model has been coded by Matlab which is a special mathematical computation language. Matlab is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

3.1. Simulation model framework

We now describe the framework of our proposed simulation model (see Fig. 1). From Fig. 1, we can see that there are six subsystems in the proposed simulation model. These six subsystems work together for obtaining an optimal or near-optimal production schedule in the flexible manufacturing environment.

3.2. Input subsystem

The leading function of input subsystem is inputting all necessary data of FJSSP. In this paper, we apply file mode to implement the data inputting. We provide the necessary data to input subsystem using a file with specific format and the input subsystem obtains these data through reading the given file. Please note, it should have a verify function after data reading, such as, each input data should be a positive integer. Limit to the paper length, we do not elaborate this verify function in this paper.

3.3. Operation assignment subsystem

The primary mission of operation assignment subsystem is achieving an excellent assignment of operations to machines. There are three parts in the operation assignment subsystem. (1) Data collection part, it collects all the necessary input data for the