Expert Systems with Applications 39 (2012) 10016–10021

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Flexible job shop scheduling with parallel machines using Genetic Algorithm and Grouping Genetic Algorithm

James C. Chen [a,*], Cheng-Chun Wu [b], Chia-Wen Chen [c], Kou-Huang Chen [d]

[a] Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu, Taiwan, ROC
[b] Data System Consulting Co., Taipei, Taiwan, ROC
[c] Department of Industrial and Information Management, National Cheng Kung University, Tainan, Taiwan, ROC
[d] Department of Industrial Engineering and Management, China University of Science and Technology, Taipei, Taiwan, ROC

ARTICLE INFO

Keywords:
Flexible job shop
Scheduling
Parallel machine
Genetic Algorithm
Grouping Genetic Algorithm
Makespan
Tardiness

ABSTRACT

Based on Genetic Algorithm (GA) and Grouping Genetic Algorithm (GGA), this research develops a scheduling algorithm for job shop scheduling problem with parallel machines and reentrant process. This algorithm consists of two major modules: machine selection module (MSM) and operation scheduling module (OSM). MSM helps an operation to select one of the parallel machines to process it. OSM is then used to arrange the sequences of all operations assigned to each machine. A real weapon production factory is used as a case study to evaluate the performance of the proposed algorithm. Due to the high penalty of late delivery in military orders and high cost of equipment investment, total tardiness, total machine idle time and makespan are important performance measures used in this study. Based on the design of experiments, the parameters setting for GA and GGA are identified. Simulation results demonstrate that MSM and OSM respectively using GGA and GA outperform current methods used in practice.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

A classic job shop scheduling problem (JSP) has a set of *n* jobs processed by a set of *m* machines. According to its production routine, each job is processed on machines with a given processing time, and each machine can process only one type of operation. In modern manufacturing plant, a machine may have the flexible capability to be set up to process more than one type of operations. This leads to a modified version of JSP called flexible JSP (FJSP). There are two types of FJSP (Brucker & Schlie, 1990). For type I FJSP, jobs have alternative operation sequences and alternative identical or non-identical machines for each operation. The problem is to select operation sequences for jobs and determine job processing orders on machines. For type II FJSP, jobs can have only fixed operation sequences but alternative identical or non-identical machines for each operation. The problem is to arrange jobs to machines according to their operation sequences.

In this paper, a scheduling algorithm for type I FJSP with parallel machines and reentrant process is developed and applied to a real factory. In a reentrant shop, jobs visit a certain machine or a set of machines more than once. As shown in Fig. 1, M_1, M_2, and M_3 represent machine types, and each type has parallel machines, e.g., M_1 has three parallel machines M_{11}, M_{12}, and M_{13}. Solid lines and dot lines represent the process flows of two different jobs. The dot line shows a reentrant flow from machine type M_2 to M_1 to M_3 and then to M_1 again. Each operation is processed by one machine and the processing time is the same for any of the identical parallel machines. In this paper, a setup depends only on the job to be processed and the machine processing it, and the setup time is included in the processing time. The transportation time is ignored in this study.

The proposed algorithm is developed in two major modules. Grouping Genetic Algorithm (GGA) is applied to develop machine selection module (MSM) to assign operations to machines. Then Genetic Algorithm (GA) is used to develop operation scheduling module (OSM) to determine the processing sequence of operations on machines. The objectives of the proposed algorithm are the minimization of multiple performance measures including makespan, total tardiness, and total machine idle time.

This paper is organized as follows. This section gives an introduction to this research. The next section reviews relevant literature. Section 3 describes the parameters setting for GA and GGA and the detailed procedure of the proposed algorithm. Next, Section 4 presents a case study in a real weapon production factory using the proposed algorithm. Section 5 shows the results and discussion. Finally, Section 6 draws conclusions and gives directions of future work.

* Corresponding author. Tel.: +886 3 574 2695.
E-mail addresses: james@ie.nthu.edu.tw, jameschen1686@gmail.com (J.C. Chen).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.01.211

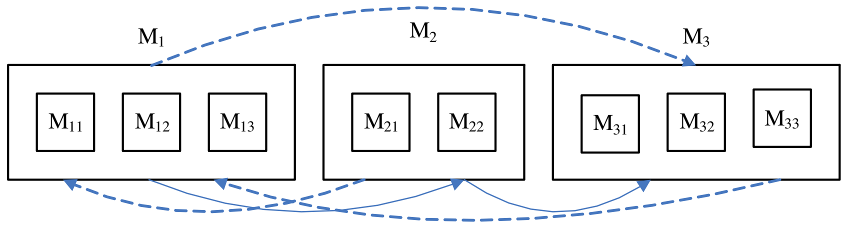

Fig. 1. A typical job shop layout showing the relationship among machine types, parallel machines, and jobs' process routings.

2. Literature review

There are far less literatures focusing on FJSP than on JSP. FJSP was first proposed by Brucker and Schlie (1990). They proposed a polynomial algorithm to solve the assignment and scheduling problems in FJSP with two jobs. For work on type I FJSP, Balin (2011) proposed new operations in GA to adapt non-identical parallel machine scheduling problem in order to minimum makespan. Baykasoğlu (2002) proposed a linguistic based meta-heuristic modeling and solution approach. Chen, Chen, Wu, and Chen (2008) developed a heuristic algorithm and applied it to solve real industrial FJSP. Choi and Choi (2002) developed a mixed-integer program model and local search algorithm for alternative operations and sequence-dependent setups in various manufacturing environments and in project scheduling. Kim, Park, and Ko (2003) proposed a new approach to simultaneously solve process planning and scheduling problem to balance machine loads and to minimize the makespan and mean flow time. Liu, Sun, Yan, and Kang (2011) presented an Adaptive Annealing Genetic Algorithm (AAGA) to deal with the job-shop planning and scheduling problem for the single-piece, small-batch, custom production mode, and computational results proved AAGA was more efficient than traditional GA. Moradi, Ghomi, and Zandieh (2011) proposed four algorithms to solve FJSP with preventive maintenance activities under the multi-objective optimization approaches, and used a benchmark with a large number of test instances to investigate the performance of the approaches. Yazdani, Amiri, and Zandieh (2010) proposed a parallel variable neighborhood search algorithm solving FJSP to minimize makespan time, and the results demonstrated that the proposed method is better than other algorithms in the past research.

For type II FJSP, Chan, Wong, and Chan (2006) proposed a GA-based approach to solve resource-constrained type II FJSP problem with multiple identical machines. Chan, Choy, and Bibhushan. (2011) presented a GA-based job-shop scheduler for a flexible multi-product, parallel machine sheet metal job shop with an objective to minimum makespan, and it yielded better than other simple sequencing rules. Mastrolilli and Gambardella (2000) proposed a neighborhood function using tabu search method incorporating two neighborhood functions to minimize the makespan. Norman and Bean (1999) developed a GA-based approach to minimize total tardiness given multiple machines, ready times, sequence dependent setups, machine downtime and scarce tools.

Genetic Algorithms are heuristic search techniques that utilize analogies to natural selection and survival of the fittest. They employ a population of solutions, combining those solutions in specific ways in an attempt to form better solutions. Introduced by Holland (1975), Genetic Algorithms have become a popular solution methodology for a variety of complex problems (Allahverdi, Gupta, & Aldowaisan, 1999). The population of solutions with which a GA works is comprised of encodings, known as chromosomes. Individual elements of the chromosomes are called genes. Based on the objective function of the problem at hand, each chromosome is evaluated and given a fitness score. Tsai, Liu, Ho, and Chou (2008) proposed a method of combining the traditional GA with Taguchi method to solve JSP.

The Grouping Genetic Algorithm, introduced by Falkenauer (1992), is a Genetic Algorithm designed to handle the special structure of grouping problems. The GGA varies from the standard GA in several ways. First, the encoding scheme includes an augmented chromosome. The groups are encoded on a one gene for one group basis, and this separate encoding is appended to the standard chromosome. Second, the genetic operators work with the group section of the chromosome, altering the composition of the groups. This, in turn, leads to alteration of the main chromosome. Third, the GGA operators of crossover and mutation do not function in the same manner as the classical operators. Brown, Ragsdale, and Carter (2007) applied GGA to solve the multiple traveling salesperson problems.

3. Proposed algorithm

In this research, an order is defined to have several jobs with precedence relations in a hierarchical structure based on Bill of Materials (BOM). Different jobs with the same parent node can be processed simultaneously at different machines. Each job includes several operations that must be processed in a specified sequence and different operations of the same job cannot be processed at different machines simultaneously. As shown in Fig. 2, order o has four jobs J_1 to J_4 with one to three operations for each job. The first operation of J_3 (OP_{31}) can be processed when the last operation of J_1 (OP_{12}) is finished. The first operation of J_1 (OP_{11}) can be processed by any machine of type M_1 including M_{11}, M_{12}, and M_{13}. The third operation of J_3 (OP_{33}) represents a reentrant feature as it is processed by the same machine type of its preceding operation OP_{31}.

The proposed algorithm (shown in Fig. 3) consists of two modules: MSM and OSM. Using MSM, an operation selects one of the parallel machines to process it, and OSM is then used to schedule the sequences and timing of all operations assigned to each machine. This paper integrates MSM using GGA and OSM using GA.

3.1. Machine selection module (MSM)

In MSM, each operation selects one of the parallel machines to process it. When GGA is applied, an encoding is selected that fits naturally with the type I FJSP with parallel machines and reentrant process problem. In particular, the chromosome representation consists of two sections: the group section and the machine section. The following notation is used for chromosome representation in this research:

$$G_1G_2\ldots G_N|12\ldots M$$

where G_i the machine used to process ith job, $i = 1$ to N, M the number of machines, N the number of jobs, $G_i \in \{1,2,\ldots,M\}$.

For example, there are three parallel machines that can process nine tasks. One solution to this problem is the chromosome $\overset{1}{1}\,\overset{2}{1}\,\overset{3}{2}\,\overset{4}{2}\,\overset{5}{3}\,\overset{6}{1}\,\overset{7}{3}\,\overset{8}{1}\,\overset{9}{2}|123$. In this chromosome, jobs 1, 2, 6, and 8 are

Download English Version:

https://daneshyari.com/en/article/383663

Download Persian Version:

https://daneshyari.com/article/383663

Daneshyari.com