Available online at www.sciencedirect.com
ScienceDirect
Advances in Engineering Software 39 (2008) 47–54
ADVANCES IN ENGINEERING SOFTWARE
www.elsevier.com/locate/advengsoft

Parallel robot scheduling to minimize mean tardiness with precedence constraints using a genetic algorithm

Tarık Çakar [a,*], Raşit Köker [b], H. İbrahim Demir [a]

[a] Sakarya University, Engineering Faculty, Department of Industrial Engineering, 54187 Sakarya, Turkey
[b] Sakarya University, Engineering Faculty, Computer Engineering Department, 54187 Sakarya, Turkey

Received 4 October 2005; accepted 21 November 2006
Available online 17 January 2007

Abstract

Identical parallel robot scheduling problem for minimizing mean tardiness with precedence constraints is a very important scheduling problem. But, the solution of this problem becomes much difficult when there are a number of robots, jobs and precedence constraints. Genetic algorithm is an efficient tool in the solution of combinatorial optimization problems, as it is well known. In this study, a genetic algorithm is used to schedule jobs that have precedence constraints minimizing the mean tardiness on identical parallel robots. The solutions of problems, which have been taken in different scales, have been done using simulated annealing and genetic algorithm. In particular, genetic algorithm is found noteworthy successful in large-scale problems.
© 2006 Elsevier Ltd. All rights reserved.

Keywords: Parallel robot scheduling; Genetic algorithms; Precedence constraints; Mean tardiness

1. Introduction

In this study, a genetic algorithm is used to schedule jobs that have precedence constraints minimizing the mean tardiness on n-number of job and m-number of identical parallel robots. There are many algorithms and heuristics related to the scheduling problem of parallel machines and robots. In this study, a genetic algorithm has been used to find the job schedule, which minimizes the mean tardiness. We know that this problem is in the class of NP-hard combinatorial problem.

Kashara and Narita [1] developed a heuristic algorithm and optimization algorithm for parallel processing of robot arm control computation on a multiprocessor system. Chen et al. [2] developed a state-space search algorithm coupled with a heuristic for robot inverse dynamics computation on a multiprocessor system. An assignment rule noted traffic priority index (TPI) was built in 1991 by Ho and Chang [3]. In this method, SPT and EDD rules are combined using by using a new measurement named as traffic congestion ratio (TCR). Then, for the cases with one or identical machine they built heuristics. Their heuristics consist of building a first solution by scheduling jobs in increasing order of their priority index. Then they improved this solution using permutation technique of WI method, which was developed previously by Wilkerson and Irwin [4].

Koulamas [5] developed a heuristic noted hybrid simulated annealing (HAS) based on simulated annealing. Chen et al. [6] have developed highest priority job first (HPJF) method, which is based on extension of the WI method extended with various priority rules such as minimum processing time first (priority = 1/processing time), maximum processing time first (priority = processing time), minimum deadline first (priority = 1/due date) and maximum deadline first (priority = due date). Alidaee and Rosa [7], in 1997, proposed a heuristic which is based on extending the modified due date (MDD) method belonging Baker and Bertrand [8]. Their method is quite effective for parallel machine problem according to their reports. Azizoglu and

* Corresponding author.
E-mail address: tcakar@sakarya.edu.tr (T. Çakar).

0965-9978/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.advengsoft.2006.11.003

Kirca [9] proposed a branch and bound (BAB) approach to solve the same problem mentioned in this paper. Another example can be given by considering identical due dates and processing times, Elmaghraby and Park in 1974 [10], developed an algorithm based on a branch and bound to minimize a function of penalties belonging to tardiness. In 1977, Barnes and Brennan [11] evaluated and improved their method again.

In addition to these previous studies, there are a few more studies, which deal with parallel machine scheduling problem. But these studies are interested in alternatives. A few examples are given in the following for the minimization of the total weighted tardiness: Emmons and Pinedo [12], Arkin and Roundy [13]; for uniform or unspecified parallel machines scheduling, the example studies are: Emmons [14] or Guinet [15]. Karp [16] has shown that even the total tardiness minimization in two identical machine scheduling problem was NP-hard. A branch and bound algorithm to minimize maximum lateness considering due dates, family setup times and release dates has been presented by Shutten and Leussink [17]. A genetic algorithm was used to find a scheduling policy for identical parallel machine with setup times in Tamimi and Rajan's study [18]. Armento Yamashita [19] applied tabu search into parallel machine scheduling. A scheduling problem for unrelated parallel machine with sequence dependent setup times was studied by Kim et al. [20] using simulated annealing. SA was used to determine a scheduling policy to minimize total tardiness. Min and Cheng [21] proposed an algorithm for identical parallel machine problem. Their algorithm is based on using GA and SA to minimize makespan. According to their studies, it is seen that GA proposed is efficient and fit for larger scale identical machine scheduling problem to minimize the makespan. Kanjo and Ase [22] studied about scheduling in a multi robot welding system. Sun and Zhu [23] applied a genetic algorithm for scheduling dual resources with robots. Zacharia and Asparagatos [24] proposed a method on GAs for optimal robot task scheduling. In this study, the job with n-number of precedence constraints is assigned minimizing mean tardiness on m-number of parallel robot using genetic algorithms. A sample work station of robots working in parallel is shown in Fig. 1.

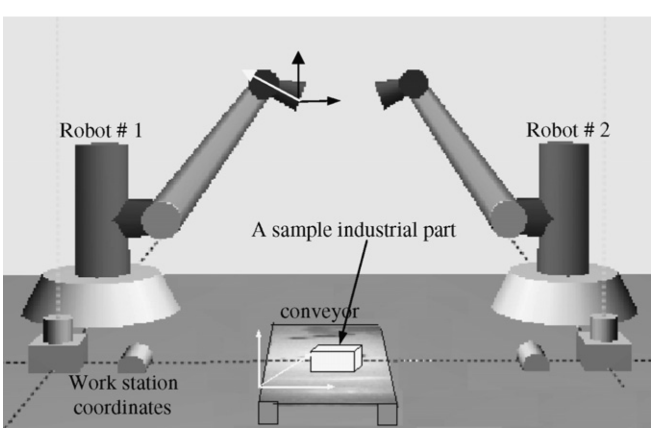

Fig. 1. A sample work station of robots working in parallel.

2. Definition of the problem

In this study, the job with n-number of precedence constraints is scheduled minimizing mean tardiness on m-number of parallel robots. There are process time and due date for each job. There is not any ready time that belongs to jobs. A robot can do just one job at the same time. The processing is non-preemptive. The target function, which will be minimized, is given below in Eq. (1).

$$\text{Mean Tardiness} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} R(i,j)T_j}{n} \quad (1)$$

Here, $T_j = \max\{0, C_j - d_j\}$ is the tardiness of job j. C_j being the completion time and d_j being due date for job j. $R(i,j)$, represents processing or unprocessing of j job on i robot. If j job is being processed on i robot, $R(i,j) = 1$, otherwise (if not being processed) $R(i,j) = 0$.

3. Genetic algorithm

The advantages of the genetic algorithms have been mentioned in the previous section. In this section, the modeling and the application of the GA are explained. From the view point of the working principle, genetic algorithms firstly needs the coding of the problem with the condition that it should be fitting with the GA. After coding process, GA operators are applied on chromosomes. It is not guaranteed that the obtained new offsprings are good solutions by the working of crossover and mutation operators. Feasible solutions are evaluated, and others are left out of evaluation. The feasible ones of the obtained offsprings are taken and new populations are formed by reproduction process using these offsprings. Crossover, mutation and reproduction processes go on until an optimal solution is found. The modeling of the defined problem using genetic algorithm has been presented below with its details.

3.1. Coding

The scheduling of the jobs on each robot forms the chromosomes. Here, the chromosomes give the number of robots too. The gene code are c_1, c_2, $c_3, \ldots, c_j, \ldots, c_n$, where $c_j \in [1, m]$. c_j is positive integer number. Here, each parallel robot represents a chromosomes; and gene in chromosome, represents ordered jobs on a robot. The assigned of jobs on robots when forming initial population is done randomly, and while this ordering is done, precedence constraints are taken under care. For instance, let us suppose that there are 8 jobs and 2 robots, and their precedence constraints are given in Fig. 3. Sample list representation of the schedule of the jobs on M1 and M2 robots has been given in Fig. 3. The sample schedule gives also a sample gene code.

Here, the scheduling of the jobs on robots also shows chromosomes code. M job can be scheduled on N robots

Download English Version:

https://daneshyari.com/en/article/566618

Download Persian Version:

https://daneshyari.com/article/566618

Daneshyari.com