Computers & Industrial Engineering 57 (2009) 1150–1153

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Short Communication

Generating subproblems in branch and bound algorithms for parallel machines scheduling problem ☆

Sang-Oh Shim

Department of Business Administration, Hanbat National University, Daejeon, South Korea

ARTICLE INFO

Article history:
Received 30 July 2008
Received in revised form 2 February 2009
Accepted 27 February 2009
Available online 9 March 2009

Keywords:
Scheduling
Branch and bound algorithm
Parallel machines
Branching

ABSTRACT

A branch and bound algorithm (B&B) has been widely used in various discrete and combinatorial optimization fields. To obtain optimal solutions as soon as possible for scheduling problems, three tools, which are branching, bounding and dominance rules, have been developed in the B&B algorithm. One of these tools, a branching is a method for generating subproblems and directly determines size of solution to be searched in the B&B algorithm. Therefore, it is very important to devise effective branching scheme for the problem.

In this note, a survey of branching schemes is performed for parallel machines scheduling (PMS) problems with n independent jobs and m machines and new branching schemes that can be used for identical and unrelated PMS problems, respectively, are suggested. The suggested branching methods show that numbers of generated subproblems are much smaller than that of other methods developed earlier and therefore, it is expected that they help to reduce a lot of CPU time required to obtain optimal solutions in the B&B algorithm.

1. Introduction

This note focuses on a method for generating subproblems during the B&B algorithm for the problem of scheduling n jobs on m parallel machines. In conventional PMS problems, each job can be processed on only one machine during processing and no preemption is allowed. The objective of the PMS problem is to find optimal (or near optimal) schedules which satisfy the customers' various demands such as maximizing throughput and minimizing total tardiness of jobs (Alidaee & Rosa, 1997; Barnes & Brennan, 1977; Bruno, Coffman, & Sethi, 1974; Cheng & Chen, 1994; Dogramaci, & Surkis, 1979; Gupta, Ruiz-Torres, & Webster, 2003; Ho & Chang, 1991; Horn, 1973).

Generally, parallel machines can be classified into three types depending on the processing time of a job on a machine (Liaw, Lin, Cheng, & Chen, 2003; Pinedo, 1995; Shim & Kim, 2007a). If processing times of a job are the same on different machines, they are called identical and denoted as P_m. On the other hand, the processing times of jobs on a machine are proportional to a ratio given for the machine on the uniform parallel machines (Q_m). If processing times of a job on different machines are arbitrary, the machines are called unrelated (R_m). In this note, we consider scheduling problems on the identical and unrelated parallel machines.

As shown in several studies (Du & Leung, 1990; Garey & Johnson, 1979; Koulamas, 1994; Lenstra, Rinnooy Kan, & Brucker, 1977), many PMS problems are known to be NP-hard. Therefore, to obtain optimal solutions, B&B algorithm or dynamic programming has been usually developed. Azizoglu and Kirca (1998), Shim and Kim (2007b) and Yalaoui and Chu (2002) find dominance properties and effective bounds and present B&B algorithms using them in identical PMS problems with the objective of minimizing total tardiness. On the other hand, for unrelated parallel machines, Azizoglu and Kirca (1999) study the B&B algorithm to minimize total weighted flow time. Also, Liaw et al. (2003) and Shim and Kim (2007a) suggest B&B algorithms using newly suggested dominance rules and bounding schemes for minimizing total tardiness of jobs.

Since introduced by Land and Doig (1960) first, B&B algorithm has been widely used for solving discrete and combinatorial optimization problems. Since B&B algorithm is based on the total enumeration method which evaluates every feasible solution to the problem and select the best, one may be necessary to fathom unpromising nodes (and not to generate unpromising nodes) effectively for finding optimal solutions in a reasonable time. Three tools, which are branching, bounding and dominance rules, have been usually used in the scheduling problems to reduce the CPU time required to obtain optimal solution in the B&B algorithm. One of the tools, branching scheme is the method for generating subproblems (or candidate problems) and directly determines size of solution domain (i.e. the set of subproblems) which would be searched during the algorithm if no node is pruned by dominance rules and bounds during the algorithm.

☆ This manuscript was processed by Area Editor Maged M. Dessouky.
E-mail address: soshim@hanbat.ac.kr

0360-8352/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2009.02.013

Since there may be several branching schemes even for the same problem, sizes of the solution domain might be different. Some branching schemes might generate redundant and meaningless feasible subproblems. In other words, even though better dominance rules and bounding schemes are developed and used in the B&B algorithm, it may take too much time to check dominance rules and compute bounds repeatedly due to bad branching schemes. Therefore, proposing a good branching scheme/method is very essential as well as suggesting better dominance rules and bounds (lower and upper). In this paper, after some branching schemes for PMS problems are introduced, we suggest two new branching schemes that can help to reduce much time required to obtain optimal solutions in the B&B algorithm for the identical and unrelated PMS problems.

In this note, we suggest two effective branching schemes for identical and unrelated PMS problems, respectively. The remainder of this paper is organized as follows. In the next section, some branching schemes developed earlier are introduced with numerical examples. In the succeeding section, we present new branching methods, and the paper is concluded with a short summary and discussions on possible extensions.

2. Survey of branching schemes

Consider a branching scheme of the problem with n independent jobs and m parallel machines, where n and m are the number of jobs and the number of machines, respectively. Each node in the B&B tree represents a partial schedule, and a node in the ith level of the tree corresponds to a partial schedule in which i jobs are scheduled at the front part of a complete schedule. Then, $m{\cdot}(n-i)$ child nodes are generated from each parent node by assigning one of the unscheduled jobs to each of the m machines. Without being fathomed by any dominance properties and any bounds, the total number of *complete schedules*, which is represented by the last generated nodes in the last level, i.e. level n, to be generated is $m^n n!$. Here, let us denote this branching scheme as a *full enumeration* branching. Fig. 1 shows an example of a B&B tree for a problem with two machines and two jobs by using this method. Each node represents a pair of indexes of a job to be scheduled and a machine to be assigned at the front part of a complete schedule, respectively (a, b in the node implies that job a is scheduled on machine b in the front part of the complete schedule).

This method seems to be very natural; however, it can be seen that this B&B tree includes redundant schedules, as can be seen in Fig. 1. For example, nodes N_7 and N_{12} (in which job 1 and 2 are assigned to the machine 1 and 2, respectively) represent the same schedules and nodes N_8 and N_{11} also do. Consequently, if this method is used in the B&B algorithm for PMS problems, it may take much time to find optimal solutions since redundant schedules should be checked repeatedly.

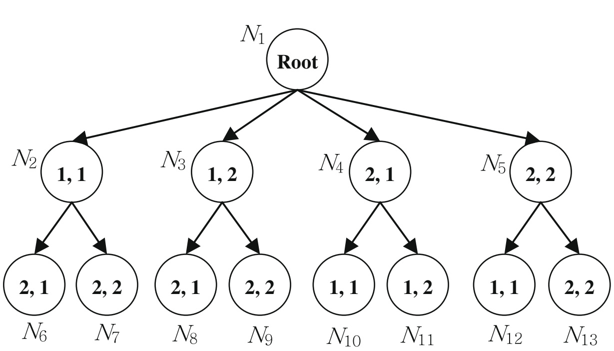

Fig. 1. A B&B tree obtained by the *full enumeration* branching.

2.1. Branching method for $R_m||\Sigma f(C_i)$

To avoid generation of redundant nodes, Shim and Kim (2007a) develop a new branching scheme for the unrelated parallel machines for *minimizing total regular costs*, i.e. a total regular cost is a function of job completion time and increases only if at least one of the job completion times in the schedule increases (Baker, 1974). From the three fields notation (Pinedo, 1995), this problem can be referred to as $R_m||\Sigma f(C_i)$, where $f(C_i)$ is a non-decreasing function of C_i which is a completion time of job i.

In this scheme, a new job associated with a certain node is assigned to a machine whose index is not less than that of the machine associated with the parent node. With n independent jobs and m unrelated PMS problem, the number of complete schedules is $\binom{n+m-1}{n} n!$ and let us call this scheme an R_m branching. Fig. 2 shows the B&B tree obtained by this scheme with the same problem instance of Fig. 1.

2.2. Branching method for $R_m||\Sigma w_i F_i$

Azizoglu and Kirca (1999) suggest the branching scheme for the unrelated parallel machines with the objective of minimizing total weighted flow time, i.e. $R_m||\Sigma w_i F_i$. Based on the optimal solution property (suggested by Smith (1956)) in which jobs on each machine should be ordered by the weighted shortest processing time (WSPT) rule, for each parent node, m child nodes are generated and each child node k represents the first unscheduled jobs ordered by WSPT rule on a machine k. A node at the ith level corresponds to a partial schedule with i jobs scheduled. This method generates only m^n complete schedules and in this study, we call this branching scheme an R_m *WSPT* branching.

Fig. 3 shows a branching tree with two machines and three jobs, whose weights are the same and whose processing times on each machine are 1, 2 and 3 on machine 1 and 3, 2 and 1 on machine 2, respectively. Although this method has a strong point in which only dominant schedules are generated for the considered problem, it may generate redundant schedules. For example, schedules represented by nodes N_9, N_{10} and N_{12} are the same ones (Also, nodes N_{11}, N_{13} and N_{14} are the same).

2.3. Branching method for $P_m||\Sigma T_i$

For the identical parallel-machine scheduling problems with the objective of minimizing total regular costs, i.e. $P_m||\Sigma f(C_i)$, an active schedule, in which no job can be started earlier without delaying any other job, is dominant in the problem (Baker, 1974). Therefore, a dominant schedule with a permutation of jobs may be obtained. Thus, from a permutation of jobs, one can construct a schedule by assigning each job to a machine with the earliest start time while ties are broken by choosing a machine with the lowest index.

Azizoglu and Kirca (1998), Shim and Kim (2007b) and Yalaoui and Chu (2002) use this branching scheme for solving their prob-

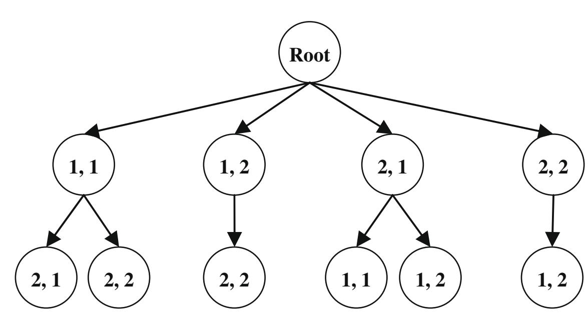

Fig. 2. A B&B tree obtained by the R_m branching.