ELSEVIER

European Journal of Operational Research 132 (2001) 528–538

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/dsw

Theory and Methodology

A flexible flowshop problem with total flow time minimization

Meral Azizoğlu [a,*], Ergin Çakmak [b], Suna Kondakci [a]

[a] Industrial Engineering Department, Middle East Technical University, Ankara, Turkey
[b] Industrial Engineering Department, Anadolu University, Eskişehir, Turkey

Accepted 12 May 1999

Abstract

In this study, we consider total flow time problem in a flexible flowshop environment. We develop a branch and bound algorithm to find the optimal schedule. The efficiency of the algorithm is enhanced by upper and lower bounds and a dominance criterion. Computational experience reveals that the algorithm solves moderate sized problems in reasonable solution times.

Keywords: Flexible flowshop; Parallel machines; Total flow time

1. Introduction

This paper addresses the problem of scheduling n jobs on w serial stages, each stage including several parallel identical machines. A job should be processed on any one of the parallel machines at each stage. Such an environment is called a flexible flowshop. Our scheduling objective is to minimize total flow time.

Flexible flowshops are generalizations of flowshops. The literature for the flowshop problem has grown after Johnson's (1954) well-known algorithm for the two stage maximum completion time, i.e. the makespan problem. Many of the studies on flowshops consider the minimization of makespan and total flow time. The studies by Gupta and Dudek (1971) and Panwalker et al. (1972) have revealed that total flow time problem is more representative of scheduling costs than makespan. The branch and bound algorithms by Ignall and Schrage (1965), Bansal (1977) and Ahmadi and Bagchi (1990) and heuristic approaches by Ho and Chang (1991), Miyazaki et al. (1978) and Rajendran and Chaudhuri (1991) are among various attempts to solve the total flow time problem on flowshops.

Recent studies have recognized the importance of flexible flowshops to reduce the delays caused by bottleneck stages. Flexible flowshop problems arise in a number of different settings including polymer, chemical, and petrochemical industries (Salvador, 1973). It has been encountered in certain manufacturing systems (Zijm and Nelissen, 1990) and in assembly lines with parallel machines at workstations (Brah and Hunsucker, 1991),

* Corresponding author. Tel.: +90-312-210-2281; fax: +90-312-210-1268.

E-mail address: meral@ie.metu.edu.tr (M. Azizoğlu).

0377-2217/01/$ - see front matter
PII: S0377-2217(00)00142-9

electronics industry (Guinet and Solomon, 1996) and textile industry (Guinet, 1991).

Many of the studies on flexible flowshops consider makespan minimization. Some special cases of the makespan problem are studied by Arthanari and Ramamurthy (1971), Mittal and Bagga (1972), Murty (1974), Rajendran et al. (1986), Gupta (1988), Gupta and Tunc (1994) and Kim et al. (1998). The studies by Sriskandarajah and Sethi (1989) and Brah and Hunsucker (1991) consider the general flexible flowshop makespan problem. Sriskandarajah and Sethi (1989) study a number of heuristics in terms of their worst and average case performances, whereas Brah and Hunsucker (1991) propose a branch and bound algorithm to find an optimal schedule.

To the best of our knowledge, the only reported research on total flow time problem in flexible flowshops is due to Rajendran and Chaudhuri (1992). The study proposes a branch and bound algorithm to obtain an optimal permutation schedule. In this study, we propose a branch and bound algorithm to find an optimal solution for the total flow time problem in flexible flowshops. This optimal schedule need not be a permutation schedule. The lower bounds and the dominance rule developed considerably reduce the size of the branch and bound tree.

The rest of the paper is organized as follows. In the next section, we define our notation and describe the problem. In Section 3, we present the branch and bound algorithm along with the lower and upper bounds and the dominance theorem. In Section 4, we give the results of our computational experience. We conclude in Section 5.

2. Problem definition

We consider the flowshop scheduling problem. We assume that there are identical parallel machines that are continuously available at each stage. Processing of a job on a machine cannot be interrupted.

Let w denote the number of stages, n the number of jobs, m_j the number of machines at stage j, p_{ij} the processing time of job i at stage j and C_{ij} the completion time of job i at stage j. All jobs follow the same processing order, $1, 2, \ldots, w$.

The completion time of job i in schedule S is $C_{iw}(S)$ and the objective is to find S^* that minimizes total flow time, i.e.

$$\sum_i C_{iw}(S^*) = \underset{S \in \pi}{\text{Min}} \left\{ \sum_i C_{iw}(S) \right\},$$

where π is the set of all feasible schedules.

Since a special case of our problem, where there is a single machine at each stage has been shown to be NP-hard (Garey et al., 1976) our problem is also NP-hard. When the problem has a single stage, it can be solved in $O(n \log n)$ complexity (Conway et al., 1967). Assigning jobs to the earliest available machine in the order they appear in the shortest processing time (SPT) list solves the problem.

A common approach to flowshop problems is to use permutation schedules which preserve the same order of assignment at each stage. In our problem, this corresponds to using the same list from which the jobs are assigned to the earliest available machine at each stage. Permutation schedules are not guaranteed to produce optimal solutions except in some restrictive special cases. Note that permutation schedules constitute only $n!$ of the $(n!)^w$ possible schedules.

Our approach does not restrict itself with permutation schedules and is guaranteed to find the optimal schedule.

3. The branch and bound algorithm

We solve a branch and bound algorithm for the first stage of our flexible flowshop problem. The end nodes correspond to complete solutions and are listed in nondecreasing order of their flow time values. Starting from the first node of the list, we proceed to the branch and bound procedure of the second stage. The completion times of the jobs at stage 1 are their ready times at stage 2. Similarly the end nodes of stage k provide ready times for stage $k+1$. A complete solution is obtained when the branch and bound solution of stage w is reached. We backtrack to a previous stage when