Computers & Industrial Engineering 56 (2009) 1464–1473

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Beam search algorithm for capacity allocation problem in flexible manufacturing systems

Selin Bilgin Özpeynirci [a], Meral Azizoğlu [b,*]

[a] İzmir University of Economics, Department of Industrial Systems Engineering, 35330 Balçova, İzmir, Turkey
[b] Middle East Technical University, Industrial Engineering Department, 0653l Ankara, Turkey

A R T I C L E I N F O

Article history:
Received 13 November 2007
Received in revised form 16 April 2008
Accepted 9 September 2008
Available online 17 September 2008

Keywords:
Beam search
Capacity allocation
Flexible manufacturing systems
Tool assignment

A B S T R A C T

This study considers the operation assignment and tool allocation problem in flexible manufacturing systems. A set of operations together with their required tools are selected so as to maximize the total weight. The machines have limited time and tool magazine capacities and the tools are available in limited quantities. We develop a beam search algorithm and obtain near optimal solutions for large size problems very quickly.

1. Introduction

Flexible manufacturing systems (FMSs) are defined as the integrated systems of computer numerically controlled (CNC) machines connected with automated material handling mechanisms. They aim to combine flexibility of job shops and efficiency of flow lines through proper planning of operation assignments and tool allocations. In recent years, FMSs attract enormous attention both in industry and academia, due to its practical importance and technical challenge inherent in solving the associated problems. Moreover the high capital investment deserves extra attention via efficient planning techniques. Several problems are addressed in FMSs include, but not limited to, part selection, operation assignment and tool allocation.

The operation assignment problem can be defined as the selection of the operations belonging to the selected part types so as to achieve some prespecified goals. In the FMS literature, the operation assignment problem is studied both in the absence and presence of tooling decisions. Toktay and Uzsoy (1998), Akçalı, Üngör, and Uzsoy (2005) and Çatay, Erengüç, and Vakharia (2005) address the capacity allocation problem ignoring the tools. Toktay and Uzsoy (1998) assume that the machines can process a limited number of operations and they have limited time capacities. Their aim is to maximize the total assigned work. Akçalı et al. (2005) study the same problem, but they assume that each machine can process only a subset of operations. Çatay et al. (2005) allow machine duplications and aim to minimize the total cost due to capacity usages and machine purchases. Çatay et al. (2005) consider a capacity allocation problem that defines efficient tool procurement plans.

The most noteworthy studies that consider the operation assignment problem together with the tooling decisions are due to D'Alfonso and Ventura (1995), Ventura, Chen, Frank, and Leonard (1988), Berrada and Stecke (1986), Kim, Lee, Lim, and Choi (2003). D'Alfonso and Ventura (1995) aim to minimize the frequency of travels between the CNC machines and Ventura et al. (1988) study the maximum completion time problem. Berrada and Stecke (1986) consider time capacities over all machines and for each individual machine. Kim et al. (2003) assume deadlines on the completion times of the operations and aim to minimize the tool purchasing cost.

In this paper, we consider the operation assignment and tool allocation decisions simultaneously. There are a group of operations that have to be processed on a group of parallel CNC machines. We assume that any operation splitting is not allowed and the machines have limited time and tool magazine capacities. Moreover the number of tools available in the system is assumed to be limited. We assign the operations together with their required tools to the machines so as to maximize the total weight of the assigned operations. Bilgin and Azizoğlu (2006) work on a slightly different version of this problem, where the operations can be split. They suggest several lower bounds and upper bounds based on linear programming relaxation and Lagrangean relaxation. Bilgin and Azizoğlu (2007a) consider the problem without operation splitting and find near optimal solutions using a tabu

* Corresponding author. Tel.: +90 312 210 2281; fax: +90 312 210 4687.
E-mail address: meral@ie.metu.edu.tr (M. Azizoğlu).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2008.09.007

search algorithm. Bilgin and Azizoğlu (2007b) develop upper bounds by adding valid inequalities to the linear programming relaxation of the problem. They also suggest some easily computable and effective upper bounds that can be used to evaluate the partial solutions of an enumeration algorithm. Bilgin and Azizoğlu (2007c) use these upper bounds to find optimal solutions in their branch and bound algorithm. In addition, they employ the branch and bound tree structure in their classical beam search algorithm.

The rest of the paper is organized as follows. In the next section, we define our problem and give the mathematical formulation. In Section 3, we briefly explain our beam search algorithm. Section 4 defines the components of the algorithm together with our implementation details. We give the results of our experiments in Section 5 and conclude the paper in Section 6.

2. Problem definition

Consider n operations that are to be processed by a set of m parallel CNC machines. An operation is assigned to at most one machine, and when assigned, it should be processed for P_i time units without preemption. We let w_i be the weight of operation i which may represent the profit brought or the assignment cost when it is negative. C_j is the time capacity of machine j. P_i and C_j are measured in same units, say in minutes.

The machines are flexible in the sense that they function according to the tools loaded on their tool magazines. Machine j has a tool magazine capacity of s_j tool slots and has a time capacity of C_j time units. The tools are available in limited quantities due to the technological restrictions or budget limitations. There are t tool types and r_k tools of type k are available. A set of tools $l(i)$ should be available on the tool magazine before processing operation i. We assume each tool uses one tool slot.

The problem is to select a subset of operations, allocate the time capacity of the machines to the selected operations and their tool magazine capacities to the required tools so as to maximize the total weight.

Our decision variables are:

$$X_{ij} = \begin{cases} 1 & \text{if operation is assigned machine } j \\ 0 & \text{otherwise} \end{cases}$$

$$Z_{ij} = \begin{cases} 1 & \text{if tool } k \text{ is loaded on machine } j \\ 0 & \text{otherwise} \end{cases}$$

The mathematical programming formulation of the problem is given below.

$$\text{Maximize} \quad Z = \sum_{i=1}^{n}\sum_{j=1}^{m} w_i X_{ij} \tag{2.1}$$

$$\sum_{j=1}^{m} X_{ij} \leqslant 1 \quad \forall i \tag{2.2}$$

$$\sum_{i=1}^{n} P_i X_{ij} \leqslant C_j \quad \forall j \tag{2.3}$$

$$\sum_{k=1}^{t} Z_{kj} \leqslant S_j \quad \forall j \tag{2.4}$$

$$\sum_{j=1}^{m} Z_{kj} \leqslant r_k \quad \forall k \tag{2.5}$$

$$X_{ij} \leqslant Z_{kj} \quad \forall i, j, k \in l(i) \tag{2.6}$$

$$X_{ij}, Z_{kj} \in \{0, 1\} \quad \forall i, j, k \tag{2.7}$$

Our objective function requires maximizing the total weight (2.1). Constraint set (2.2) guarantees that operation i is assigned to at most one machine. The processing amount of operations assigned to machine j cannot exceed its capacity (2.3). Constraint set (2.4) makes sure that the number of tools loaded on machine j does not exceed its tool magazine capacity. Constraint set (2.5) states that the number of tool type k assigned cannot exceed its available number. Constraint set (2.6) ensures that an operation is assigned to a machine only if all required tools are loaded. By constraint set (2.7) partial operation assignments and tool allocations are not allowed.

In the absence of the tooling decisions, our problem reduces to the classical multiple-knapsack problem. The multiple-knapsack problem is strongly NP-hard (Martello & Toth, 1990), so is our problem, with additional tool allocation decisions.

3. Beam search algorithm

Beam search algorithm is a heuristic branch and bound algorithm that returns a solution by searching only some promising partial solutions. At each level in the branch and bound tree, only a prespecified number of nodes are selected for branching, the rest of the tree is permanently discarded. Thus, the number of nodes evaluated is polynomial in the problem size.

At each level, the number of nodes kept for further branching, i.e., the number of promising nodes, is called beam width, β. The evaluation function that determines the promising β nodes is called the beam evaluation function.

Beam search technique sacrifices the guarantee of optimality for reduced solution times and memory requirements. The increase in β value and use of powerful elimination functions enhance the solution quality, however at an expense of higher solution times.

In the literature, many successful applications of the beam search technique are reported. Some recent noteworthy studies are due to Morton and Pentico (1993), Valente and Alves (2006), Della Croce, Ghirardi, and Tadei (2004), Ghirardi and Potts (2005). Morton and Pentico (1993) give the details and several variations of the beam search technique. Valente and Alves (2006) apply different versions of the beam search algorithm to the single machine total weighted tardiness scheduling problem. Della Croce et al. (2004) apply a recovering beam search technique to some well-known combinatorial optimization problems. Ghirardi and Potts (2005) suggest recovering beam search approach for the makespan scheduling problem on unrelated parallel machines.

4. The components of the beam search algorithm

In this section, we define the components of the beam search algorithm and explain the details of our implementation.

4.1. Beam search tree

Our beam search tree has the same structure with the branch and bound tree used in Bilgin and Azizoğlu (2007c). Each level of the tree corresponds to an operation assignment related decision. At the first level, we consider the operation with the largest w_i/P_i value. At each level i, there are $m + 1$ nodes for operation i. The first m nodes represent the assignment of operation to each of the m machines. The $(m + 1)$st node represents the decision of not assigning operation i to any machine, i.e., not processing it. Accordingly, the first node sets the assignment variable X_{i1} to 1, the mth node sets $X_{im} = 1$ and the $(m + 1)$st node assumes $\Sigma_j X_{ij} = 0$.The following figure illustrates our beam search tree (see Fig. 1).

4.2. Beam evaluation function

Beam evaluation function is used as a guide to determine the quality of the solution provided by a node. Our evaluation function