Pergamon

Mathl. Comput. Modelling Vol. 23, No. 11/12, pp. 203–214, 1996

0895-7177/96 $15.00 + 0.00

S0895-7177(96)00073-8

Scheduling of a Machining Center

J. WANG AND P. B. LUH
Department of Electrical and Systems Engineering
University of Connecticut, Storrs, CT 06269-3157, U.S.A.
jihua@brc.uconn.edu luh@farside.ese.uconn.edu

Abstract—A machining center is an advanced NC (Numerical Control) machine that has the capability to perform a variety of operations on a part by automatically changing the cutting tools. Because of its versatile processing capabilities, a machining center is often a production bottleneck, and effective scheduling can result in significant improvement of system performance. The problem, however, is very difficult since many factors such as machine setups, pallets, tool magazine, and possible tool overlapping among different part types, etc., have to be considered. This paper presents an optimization-based approach for the scheduling of a machining center with two pallets. A novel "separable" problem formulation that considers the above mentioned factors is presented. Lagrangian relaxation is applied to decompose the problem into simple subproblems, which are efficiently solved without encountering complexity difficulties. The subgradient method is then used to update the multipliers. Testing results indicate that the approach is effective, and the algorithm provides a valuable tool for solving stand-alone machining center problems. The approach also points out a direction on how to consider machining centers within a job shop environment.

Keywords—Scheduling, Machining center, Lagrangian relaxation.

1. INTRODUCTION

Manufacturing scheduling is concerned with the sequencing of operations on machines, and is directly linked to system performance such as on-time delivery of products, low inventory, and high productivity. A machining center is an advanced NC machine that has the capability to continuously perform a variety of operations on a part by automatically changing the cutting tools. Today, machining centers are used in a variety of settings, from job shops to flexible manufacturing systems (FMS). Because of their versatile processing capabilities, considerable numbers of operations are assigned to these expensive facilities. Consequently, machining centers are often production bottlenecks, and effective scheduling can result in significant improvement of system performance.

Problem Description

Different from traditional machines, a machining center has several pallets to hold fixtures which in turn clamp parts for processing, and a tool magazine to store the cutting tools needed (Figure 1). To process one type of parts, an appropriate fixture is first fixed on a pallet and calibrated (the "setup" process). The tools needed for this part type are also loaded onto the

This work was supported in part by the National Science Foundation under Grant DDM-9119074 and DMI-9500037, and the Advanced Technology Center for Precision Manufacturing, the University of Connecticut. The authors would like to thank B. Hu and Z. Feng of Xian Jiaotong University, P.R. China, for their invaluable insight and suggestions, and R. Tomastik and L. Gou of the University of Connecticut for their valuable help in the algorithm development process.

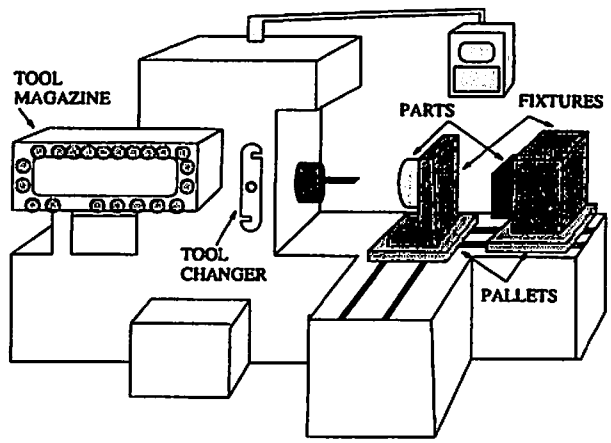

Figure 1.

tool magazine during the setup. This setup process may take away a significant amount of machine time. In a manufacturing environment where all parts are different, a setup is related to a particular part, and the setup time can be reasonably handled by lumping it with the part processing time. In medium to high volume production to be considered in this paper, parts of the same type are often processed in lots (collection of the same type of parts) to avoid excessive setups. A setup is thus associated with a lot, and a pallet is dedicated to the lot from its setup until completion. We shall say that the pallet is "occupied" by the lot, and the lot is "in-processing" during this period of time. Since a machining center usually has more than one pallet, there may be multiple lots "in-processing" at the same time. As a result, parts belonging to different "in-processing" lots may be processed alternatively. This alternative processing turns out to be beneficial since processing one part and loading another of a different lot onto its fixture can be done concurrently, although part loading generally does not take too much time.

Tool loading is another complicated factor. Since the machining center must be shut down to load/unload tools onto the tool magazine, it is highly desirable to avoid frequent tool loadings/unloadings. This translates into the constraints that all the tools needed for in-processing lots (i.e., the "union" of the tools required for those lots) should be in the tool magazine at the same time. The possible overlapping of cutting tools for different in-processing lots and the limited tool magazine capacity lead to a set of nonlinear constraints which are difficult to handle [1]. A conservative assumption of no tool overlapping is often assumed to simplify the problem.

From the above description, it can be seen that the scheduling of a machining center involves the consideration of multiple resources including the machine, pallets and the tool magazine. In addition, setup times are significant, and in-processing times of various lots depend on the schedule itself in view of the above-mentioned alternative processing possibility.

Literature Review

Literature on the scheduling of stand-alone machining centers considering the above-mentioned factors is limited. A two-stage batching and sequencing algorithm for NC punch presses is presented in [2]. Batches are first formed to maximize tool magazine utilization, and a heuristic is then used to sequence the batches. Scheduling machining centers in flexible manufacturing systems (FMS) has been studied by many researchers. For example, an interactive decision-making framework for scheduling an FMS with two machining centers is presented in [3]. A branch and bound method is applied for the scheduling of two machining centers in [4].

Overview of the Paper

In view of the importance and complexity of scheduling a machining center, this paper addresses the scheduling of a machining center with two pallets. A novel integer programming

formulation considering all the above-mentioned factors is presented in Section 2. Setups are explicitly modeled, and are required to be performed before the processing of individual parts within lots. A set of "lot-level" variables is created to describe the in-processing status of individual lots, and is related to the beginnings of setups and the completions of the last parts within those lots. The difficulty caused by tool overlapping is handled by translating the tool magazine capacity constraints into "noncompatible pairs" of part types that cannot be in-processing at the same time. The objective of on-time delivery and low inventory is modeled as weighted penalties on back order and excessive finished part inventory. The resulting formulation is "separable" in that the objective function and constraints are additive in terms of basic decision variables.

By using Lagrangian relaxation, the above problem is separated or decomposed into inventory position, part, and lot subproblems as described in Section 3. A complication arises since one set of the coupling constraints cannot be relaxed: the requirement that the setup should be started before the completion of a lot cannot be relaxed, since otherwise the basic condition for the definition of lot-level variables may be violated. Consequently, the lot subproblems are solved by using dynamic programming [5,6]. Other subproblems can be solved without encountering complexity difficulties. The subgradient method is then used to update the Lagrange multipliers, and a heuristic is developed to construct feasible schedules based on subproblem solutions. Testing results presented in Section 4 indicate that the approach is effective, and the algorithm provides a valuable tool for solving stand-alone machining center problems.

Lagrangian relaxation is a decomposition and coordination technique, and has been proved to be effective for solving scheduling problems consisting of various types of machines. By following the step-by-step modeling and resolution procedure in [7], the approach presented here can be extended to handle job shops or FMSs consisting of multiple machining centers and other conventional machines.

2. PROBLEM FORMULATION

The following integer programming formulation is based on the models presented in [7], and considers all the key factors of a machining center. It is assumed that parts belong to I types, and parts of type i are processed in L_i lots. The l^{th} lot of part type i is denoted as (i,l), with lot-size Q_{il} given. The setup time for part type i and the processing time of a type i part are denoted by s_i and p_i, respectively. Part loading times (i.e., time required to load a part onto a fixture already calibrated), however, are ignored since they usually are insignificant. The demand for part type i at time k is denoted by d_{ik}, and is assumed given. Fixtures, tools and parts to be processed are assumed to be available, and the time horizon under consideration is K.

As mentioned, a set of lot-level variables is created to describe the in-processing status of individual lots, and is related to the beginnings of setups and the completions of the last parts within those lots. Constraints considered include machine capacity constraints, pallet capacity constraints, tool magazine capacity constraints, processing sequence constraints, setup and processing time requirements, and inventory position dynamics. The objective of on-time delivery and low inventory is modeled as weighted penalties on the number of back orders and excessive finished part inventory. The constraints and the objective function are presented in detail below.

Machine Capacity Constraints

The machine capacity constraints state that a machining center can either perform a setup or process one part at a time, i.e.,

$$\sum_{i=1}^{I}\sum_{l=1}^{L_i}\sum_{q=0}^{Q_{il}}\delta_{ilk}^{q} \le 1, \qquad k = 1,\ldots,K, \tag{1}$$

where δ_{ilk}^{q} is a 0-1 "part variable." It equals 1 if at time k part q of lot (i,l) is being processed $(0 \le q \le Q_{il})$, and 0 otherwise. In the above, setup is treated as "part 0" of the lot for notational

convenience. Equivalently, one has

$$\delta_{ilk}^{q} = \begin{cases} 1 & \text{if } b_{il}^{q} \le k \le c_{il}^{q}, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where b_{il}^{q} and c_{il}^{q} are, respectively, the beginning time and completion time of part q of lot (i,l) $(0 \le q \le Q_{il})$.

Pallet Capacity Constraints

The pallet capacity constraints state that the number of in-processing lots may not exceed 2, the number of pallets available, i.e.,

$$\sum_{i=1}^{I} \sum_{l=1}^{L_i} \theta_{ilk} \le 2, \qquad k = 1, \ldots, K, \tag{3}$$

where θ_{ilk} is a 0-1 "lot variable." It equals 1 if lot (i,l) is in-processing at time k, and 0 otherwise. This lot variable can be specified in terms of setup beginning time b_{il}^{0} and last part completion time $c_{il}^{Q_{il}}$, i.e.,

$$\theta_{ilk} = \begin{cases} 1 & \text{if } b_{il}^{0} \le k \le c_{il}^{Q_{il}}, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

under the following basic conditions:

$$b_{il}^{0} \le c_{il}^{Q_{il}}, \qquad i = 1, 2, \ldots, I; \quad l = 1, 2, \ldots, L_i. \tag{5}$$

Tool Magazine Capacity Constraints

The tool magazine capacity constraints state that the tool slots needed for the two in-processing lots may not exceed the total number of slots in the tool magazine. Let T_i be the set of tools required for processing part type i, the constraints can then be written as

$$\text{slot}\left[\cup_{i=1}^{I} \cup_{l=1}^{L_i} T_{ilk}\right] \le T, \qquad k = 1, \ldots, K, \tag{6}$$

where the set T_{ilk} equals T_i if θ_{ilk} is 1, and ϕ (an empty set) otherwise. The *slot* operator returns the number of slots needed to hold the tools, and T is the total number of slots in the tool magazine. Note that the above are nonlinear constraints.

For a machining center with two pallets, as is commonly the case, the above constraints state that if the number of tool slots needed for a pair of part types exceeds the tool magazine capacity, these two part types may not be in-processing at the same time. The constraints can thus be translated into a set of "noncompatible pairs" of part types. For example, supposing that part types 1 and 2 constitute the only "noncompatible pair" as will be assumed for simplicity for the remaining derivation, the above nonlinear constraints become

$$\sum_{l_1=1}^{L_1} \theta_{1l_1k} + \sum_{l_2=1}^{L_2} \theta_{2l_2k} \le 1, \qquad k = 1, \ldots, K. \tag{7}$$

By using (7) as the tool magazine capacity constraints, the nonlinear difficulty caused by tool overlapping is avoided. It is clear that the number of such constraints increases as the number of noncompatible pairs grows. The approach is thus reasonable when the number of noncompatible pairs of part types is not too large.

Processing Sequence Constraints

The processing sequence constraints state that a machine should be set up for a lot before any part in the lot can be processed, representing physical constraints. Without loss of generality, they also specify the processing sequence of parts within a lot, representing artificial constraints but essential for solving the problem. That is,

$$c_{il}^{q} + 1 \leq b_{il}^{q+1}, \tag{8}$$

where $i = 1, \ldots, I$; $l = 1, \ldots, L_i$; and $q = 0, \ldots, Q_{il} - 1$.

Setup and Processing Time Requirements

The setup (or processing) time requirements state that the elapsed difference between the beginning time and completion time of a setup (or part processing) should equal the required setup time s_i (or part processing time p_i). That is,

$$c_{il}^{0} = b_{il}^{0} + s_i - 1; \qquad c_{il}^{q} = b_{il}^{q} + p_i - 1, \tag{9}$$

where $i = 1, \ldots, I$; $l = 1, \ldots, L_i$; and $q = 1, \ldots, Q_{il}$.

Although the "lot definition conditions" (5) are implied by (8) and (9) above, they are explicitly included in the problem formulation since they have to hold even after processing sequence constraints (8) are relaxed.

Inventory Position Dynamics

The inventory position dynamics are used to describe the evolution of inventory positions in the medium to high volume production environment. They state that the current inventory position x_{ik} equals the inventory position of the previous period plus the number of parts finished at the current time and minus the current demand d_{ik}. That is,

$$x_{ik} = x_{i(k-1)} + \sum_{l=1}^{L_i} \sum_{q=1}^{Q_{il}} \psi_{ilk}^{q} - d_{ik}, \qquad i = 1, \ldots, I; \quad k = 1, \ldots, K, \tag{10}$$

where ψ_{ilk}^{q} equals 1 if $c_{il}^{q} = k$, and 0 otherwise. A negative inventory position represents back orders. The initial inventory position x_{i0} is assumed to be given.

Objective Function

The weighted tardiness and earliness penalties on individual parts are effective to meet on-time delivery and reduce inventory for small volume production [7]. In medium to high volume production as considered in this paper, a customer order may contain a large quantity of parts, and there may exist a large number of orders of the same part type. It is difficult and unnecessary to trace the tardiness and earliness of individual parts. In view of this, the objective of on-time delivery and low inventory is modeled as weighted penalties on back order and excessive finished part inventory, i.e.,

$$J = \sum_{i=1}^{I} \sum_{k=1}^{K} \left[p_{ik}^{-} x_{ik}^{-2} + p_{ik}^{+} x_{ik}^{+2} \right], \tag{11}$$

where $x_{ik}^{-} \equiv \min[0, x_{ik}]$ is the number of back orders for part type i at time k, and $x_{ik}^{+} \equiv \max[0, x_{ik}]$ the number of parts in the inventory. The coefficients p_{ik}^{-} and p_{ik}^{+} represent penalty coefficients on back orders and excess inventory, respectively. Usually, p_{ik}^{-} is much greater than p_{ik}^{+} since on-time delivery is much more important than low inventory, especially for a bottleneck machine.

The scheduling problem is to minimize the objective function J subject to the above-mentioned constraints (1), (3), (5), (7)–(10). The basic decision variables are setup beginning times and part processing beginning times. Once these variables are determined, other variables can be easily derived. Since the objective function and constraints are additive in terms of these decision variables, the problem formulation is "separable." The problem can therefore be "separated" or decomposed into individual subproblems after coupling constraints (constraints (1), (3), (7), (8), (10) are relaxed by using Lagrange multipliers).

3. SOLUTION METHODOLOGY

Lagrangian relaxation is an effective method for constrained optimization for problems with a separable structure, and has been successfully used in the scheduling of job shop problems [7]. The special factors for the problem under consideration, however, complicate the formation and resolution of subproblems. Complications are caused by the existence of both part and lot variables (δ^q_{ilk} and θ_{ilk}) and the unspecified lot in-processing times. To handle these complications, setup of a lot and the processing of the last part of the lot are jointly considered, and are solved by using dynamic programming subject to the "lot definition conditions" (5).

Lagrangian Relaxation Framework

By using Lagrange multipliers π_k, ξ_k, τ_k, η_{ilq} and σ_{ik} to relax coupling constraints (1), (3), (7), (8), (10), respectively, the following relaxed problem is obtained:

$$
\begin{aligned}
R: \min_{\{x_{ik},\delta^q_{ilk},\theta_{ilk}\}} L, \qquad \text{with } L \equiv & \sum_{i,k} p^-_{ik} {x^-_{ik}}^2 + p^+_{ik} {x^+_{ik}}^2 + \sum_k \pi_k \left(\sum_{i,l,q} \delta^q_{ilk} - 1 \right) \\
& + \sum_k \xi_k \left(\sum_{i,l} \theta_{ilk} - 2 \right) + \sum_k \tau_k \left(\sum_{l_1} \theta_{1l_1k} + \sum_{l_2} \theta_{2l_2k} - 1 \right) \\
& + \sum_{i,l,q} \eta_{ilq} \left(c^q_{il} - b^{q+1}_{il} + 1 \right) + \sum_{i,k} \sigma_{ik} \left(x_{ik} - x_{i,k-1} - \sum_{l,q} \psi^q_{ilk} + d_{ik} \right),
\end{aligned} \tag{12}
$$

subject to lot definition conditions (5), and setup and processing time requirements (9).

Since inventory position dynamics (10) are relaxed, the inventory positions $\{x_{ik}\}$ become decision variables. After regrouping relevant terms, the relaxed problem (12) can be decomposed into inventory position subproblems, part subproblems and lot subproblems by using the definitions of δ^q_{ilk} and θ_{ilk} in (2), (4) and (5). The subproblems are presented next.

Inventory Position Subproblems

$$
\min_{x_{ik}} L_{ik}, \qquad \text{with } L_{ik} \equiv p^-_{ik} {x^-_{ik}}^2 + p^+_{ik} {x^+_{ik}}^2 + (\sigma_{ik} - \sigma_{i,k+1}) x_{ik}. \tag{13}
$$

The above subproblem is to minimize a *piecewise quadratic* penalty function on back order, inventory carrying, and the violation of inventory position dynamics. The sign of the linear coefficient $(\sigma_{ik} - \sigma_{i,k+1})$ determines which "piece" the minimum x^*_{ik} should be in. When $(\sigma_{ik} - \sigma_{i,k+1})$ is negative, x^*_{ik} is positive, and the p^+_{ik} term is active. The minimum can thus be obtained by setting the derivative of this quadratic envelope to zero, comparing subproblem costs of the rounded up and truncated down integer solutions, and choosing the one with lower cost. Similarly, the minimum can be found when $(\sigma_{ik} - \sigma_{i,k+1})$ is positive. There is no enumeration in solving the subproblem.

Part Subproblems

The following part subproblem applies to all parts in a lot except part 0 (the setup) and the last part in the lot ($q = Q_{il}$):

$$\min_{b_{il}^q} L_{ilq}, \qquad \text{with } L_{ilq} \equiv \sum_{k=b_{il}^q}^{c_{il}^q} \pi_k + (\eta_{ilq} - \eta_{il,q-1}) b_{il}^q - \sigma_{ic_{il}^q}, \quad q = 1, 2, \ldots, Q_{il} - 1, \tag{14}$$

subject to processing time requirements.

The part subproblem is thus a balance among machine utilization cost, the penalty for violating processing sequence, and the contribution to the balance of inventory position dynamics. The subproblem can be solved by enumerating through all possible part beginning times, and the complexity is $O(K)$.

Lot Subproblems

By grouping the terms related to setup (part 0, corresponding to lot beginning) and the last part in a lot (part Q_{il}, corresponding to lot completion), the following lot subproblem is obtained:

$$\begin{aligned} \min_{b_{il}^0, b_{il}^{Q_{il}}} L_{il}, \qquad \text{with } L_{il} \equiv & \sum_{k=b_{il}^0}^{c_{il}^0} \pi_k + \sum_{k=b_{il}^{Q_{il}}}^{c_{il}^{Q_{il}}} \pi_k + \sum_{k=b_{il}^0}^{c_{il}^{Q_{il}}} (\xi_k + \Delta_i \tau_k) \\ & + b_{il}^0 \eta_{il0} - b_{il}^{Q_{il}} \eta_{il,Q_{il}-1} - \sigma_{ic_{il}^{Q_{il}}}, \end{aligned} \tag{15}$$

subject to lot definition conditions (5), and setup and processing time requirements (9).

In the above, Δ_i is a 0-1 integer variable. It equals one if $i = 1$ or 2, and zero otherwise. Solving the subproblem by simple enumeration will require $O(K^2)$ computations. By using dynamic programming, the complexity can be significantly reduced [5,6]. To do this, we note that

$$\sum_{k=b_{il}^0}^{c_{il}^{Q_{il}}} (\xi_k + \Delta_i \tau_k) = \sum_{k=b_{il}^0}^{K} (\xi_k + \Delta_i \tau_k) - \sum_{k=c_{il}^{Q_{il}}+1}^{K} (\xi_k + \Delta_i \tau_k), \tag{16}$$

assuming that the lot definition conditions (5) are satisfied. The lot subproblem (15) can thus be written as

$$\min_{b_{il}^0, b_{il}^{Q_{il}}} L_{il}, \qquad \text{with } L_{il} \equiv L_{b_{il}^0} + L_{c_{il}^{Q_{il}}}, \tag{17}$$

where

$$L_{b_{il}^0} \equiv \sum_{k=b_{il}^0}^{c_{il}^0} \pi_k + \sum_{k=b_{il}^0}^{K} (\xi_k + \Delta_i \tau_k) + b_{il}^0 \eta_{il0}, \tag{18}$$

and

$$L_{c_{il}^{Q_{il}}} \equiv \sum_{k=b_{il}^{Q_{il}}}^{c_{il}^{Q_{il}}} \pi_k - \sum_{k=c_{il}^{Q_{il}}+1}^{K} (\xi_k + \Delta_i \tau_k) - b_{il}^{Q_{il}} \eta_{il,Q_{il}-1} - \sigma_{ic_{il}^{Q_{il}}}. \tag{19}$$

Solving the above subproblem therefore entails the following two steps.

STEP 1. For each possible value of $c_{il}^{Q_{il}}$ ($1 \le c_{il}^{Q_{il}} \le K$), decide b_{il}^{0*} that minimizes $L_{b_{il}^0}$ while satisfying the lot definition conditions (5). Compute the associate subproblem cost L_{il}.

STEP 2. Find the minimum among the K subproblem costs resulting from Step 1. The corresponding b_{il}^0 and $c_{il}^{Q_{il}}$ are optimal subproblem solutions.

In fact, the above two steps can be combined with an overall complexity $O(K)$.

The High Level Dual Problem

Based on subproblem solutions, the dual problem for updating Lagrange multipliers is obtained as

$$\max_{\{\pi_k,\xi_k,\tau_k,\eta_{ilq},\sigma_{ik}\}} D, \qquad \text{with } D \equiv \sum_{i,k} L^*_{ik} + \sum_{i,l} L^*_{il} + \sum_{i,l,q} L^*_{ilq} - \sum_k (\pi_k + 2\xi_k + \tau_k) + \sum_{i,l} \eta_{il0} s_i + \sum_{i,l,q} \eta_{ilq} p_i + \sum_i \left(\sum_k \sigma_{ik} d_{ik} - \sigma_{i1} x_{i0} \right). \tag{20}$$

The above dual problem is solved by using the subgradient method, a method commonly used to solve this type of dual problems. The number of dual variables equals $K(\pi) + K(\xi) + K(\tau) + \sum_{i,l} Q_{il}(\eta) + I * K(\sigma)$. The solution process thus involves the iterative resolution of subproblems and the updating of multipliers. This process is stopped when the maximum of the dual function is reached, or after a fixed number of iterations has been executed. The resulting schedule is generally infeasible, i.e., some of the once relaxed constraints might be still violated. A heuristic presented next is then used to construct a feasible schedule based on subproblem solutions.

The Heuristics

The heuristic consists of the following three steps:

STEP 1. Create a lot sequence list. Arrange all the lots in the ascending order of their lot beginning times (i.e., setup beginning times). If two lots have the same beginning time, the one with the shorter part processing time is placed first. If they have the same lot beginning time and the same part processing time, the one with the larger back order penalty coefficient (p^-) or the smaller inventory penalty coefficient (p^+) is placed first. Set the first two lots on the list as two in-processing lots, and remove them from the list. Initialize machine earliest available time k_a to one.

STEP 2. Check compatibility of the two in-processing lots and sequence parts. If the two in-processing lots are noncompatible, set up the first in-processing lot and schedule the parts of that lot sequentially starting with time k_a. Update k_a when a setup or a part is scheduled based on the setup or processing time requirements (9). Then set the other lot as the first in-processing lot, and go to Step 3. Otherwise, schedule the setups or part processing of these two lots in the ascending order of their setup or part beginning times, subject to the requirements that setup should be performed before part processing within each lot. If two parts have the same beginning time, part processing time and the penalty coefficients (p^- and p^+) are used to resolve the conflict as in Step 1. After one of the two lots is finished, group the remaining parts of the other lot as the first in-processing lot with or without setup requirement, and go to Step 3.

STEP 3. Determine the second in-processing lot. If there is no more lot on the lot sequence list, sequence all the parts in the first in-processing lot, and stop. Otherwise, set the first lot on the lot sequence list as the second in-processing lot, remove it from the list, and go to Step 2.

The cost of the feasible schedule obtained can be easily calculated. Although the optimal cost J^* may not be obtainable, the values of the dual function L are lower bounds on J^*. The quality of the feasible schedule can therefore be quantitatively evaluated by calculating the approximate duality gap $(J - L^*)/L^*$, where L^* is the maximum of the dual values obtained. A feasible schedule can also be generated at the end of each high level iteration, and in this case the approximate duality gap may serve as one of the stopping criteria.

Lagrange multipliers represent the "prices" or sensitivity information with respect to constraints, and can be utilized to reconfigure a schedule after changes occurred in the system (e.g., change in part processing time or demand [7]). This will be demonstrated in the next section.

4. TESTING RESULTS

Three examples are used to test the problem formulation and the solution methodology. In each of the examples, three part types with a total of 20 parts are considered. All the parts are assumed to be available for processing at the beginning, and the initial inventory positions (x_{i0}) for all part types are zero. The machining center is also assumed to be available throughout the time horizon. In the first example, it is assumed for simplicity that there is no noncompatible pair of part types. The second example is based on the first one, and is used to demonstrate the schedule reconfiguration capability. The third example is to show the ability to handle "noncompatible pairs" of part types. The testing results are obtained on a SUN SPARCstation 10.

EXAMPLE 1. In this example, there is no noncompatible pair of part types; therefore the tool magazine capacity constraints are ignored. Data are shown in Table 1. The time horizon under consideration is 30, and the total number of Lagrange multipliers is 170.

Table 1. Data for Example 1.

Part Type ID	1	2	3
Setup time	1	1	1
Part processing time	1	1	1
Back order penalty coefficients	1	1	1
Inventory penalty coefficients	1	1	1
Number of lots	2	2	2
Size of lot 1	4	3	5
Size of lot 2	3	2	3
Demand 1 (d/t^*)	4/3	3/11	5/19
Demand 2 (d/t^*)	3/7	2/15	3/24

d/t^*: demand for parts/time

All the multipliers are initialized at zero (schedule from scratch), and the results at a few different iteration numbers are presented in Table 2. It can be seen that the computation time increases almost linearly with the number of iterations. In the process, the dual cost increases fairly slowly after the first 1000 iterations. Nevertheless, since the demands for different part types are scattered over the time horizon, a good feasible schedule is obtained in the first few hundred iterations. After 2000 iterations, the lower bound obtained is 22.36, and the feasible schedule has a cost $J = 23$ with a relative duality gap of 2.87%. The feasible schedule is presented in Table 3. Since the penalty coefficients and inventory positions are integers, the feasible schedule must have an integer cost. The above feasible schedule is therefore optimal.

Table 2. Results for Example 1.

Iter. no.	Dual cost	Feasible cost	Duality gap	CPU sec.
500	16.04	23	43.40%	3.16
1000	20.45	23	12.44%	6.27
1500	21.66	23	6.19%	9.48
2000	22.36	23	2.87%	12.67

EXAMPLE 2. Suppose that the schedule for Example 1 had been obtained, but for some reasons the demands were modified. The task is now to reconfigure the schedule to meet the following revised demands: the demands for the first part type are 4 at time 3 and 3 at time 8, and the demands for the second part type are 3 at time 10 and 2 at time 16. All other data are assumed to remain the same as in Example 1.

Table 3. Feasible schedule for Example 1.

Part type	Lot id.	Beginning times					
		Setup	Part 1	Part 2	Part 3	Part 4	Part 5
1	1	1	2	3	4	5	
	2	6	7	8	9		
2	1	10	11	12	13		
	2	14	15	16			
3	1	17	18	19	20	21	22
	2	23	24	25	26		

Using the multipliers obtained from Example 1 to initialize the algorithm, the results are presented in Table 4. Results obtained by initializing all multipliers at zero (schedule from scratch) are also included in the table for comparison purpose. It can be seen that the Lagrange multipliers from the previous schedule can be used to initialize the algorithm to significantly reduce the number of iterations. With the new demands, different lots tend to compete for resources at the same time periods. The feasible schedule (for rescheduling) presented in Table 5 shows that the algorithm can smooth out the conflict among different lots by employing alternative processing of parts belonging to different types.

Table 4. Results for Example 2.

	Iter. no.	Dual cost	Feasible cost	Gap
Reschedule	200	20.70	31	49.8%
	500	22.80	28	22.8%
	1000	23.01	27	17.3%
Schedule from scratch	500	15.30	31	102.5%
	1000	21.41	28	30.7%
	2000	22.57	27	19.6%

Table 5. Feasible schedule for Example 2.

Part type	Lot id.	Beginning times					
		Setup	Part 1	Part 2	Part 3	Part 4	Part 5
1	1	1	3	5	6	7	
	2	2	4	9	10		
2	1	8	11	12	13		
	2	14	16	17			
3	1	15	18	19	20	21	22
	2	23	24	25	26		

EXAMPLE 3. In this example, part types 1 and 2 form a "noncompatible pair" of part types. Data are shown in Table 6. The time horizon is 30, and the total number of multipliers is 200.

All the multipliers are initialized at zero, and the results at a few different iteration numbers are presented in Table 7. After 2000 iterations, the lower bound obtained is 22.56, and the feasible schedule has a cost $J = 26.70$ with a relative duality gap of 18.37%. With the emerging of a noncompatible pair, the additional constraints lead to a larger number of multipliers and more terms in subproblems. The function evaluation, subgradient computation and multiplier updating for each iteration now take more time as observed from Tables 2 and 7. The corresponding feasible schedule is listed in Table 8. It can be seen that although demands for part types 1 and 2

Table 6. Data for Example 3.

Part Type ID	1	2	3
Setup time	1	1	1
Part processing time	1	1	1
Back order penalty coefficients	1.2	1	1
Inventory penalty coefficients	0.5	1	1
Number of lots	2	2	2
Size of lot 1	4	3	5
Size of lot 2	3	2	3
Demand 1 (d/t^*)	4/4	3/6	5/19
Demand 2 (d/t^*)	3/11	2/15	3/24

d/t^*: demand for parts/time

Table 7. Results for Example 3.

Iter. no.	Dual cost	Feasible cost	Duality gap	CPU sec.
500	15.28	31.70	107.46%	3.8
1000	19.65	31.70	61.28%	7.8
1500	21.22	31.70	49.41%	11.6
2000	22.56	26.70	18.37%	15.4

Table 8. Feasible schedule for Example 3.

Part type	Lot id.	Beginning times					
		Setup	Part 1	Part 2	Part 3	Part 4	Part 5
1	1	1	2	3	4	5	
	2	10	11	12	13		
2	1	6	7	8	9		
	2	14	15	16			
3	1	17	18	19	20	21	22
	2	23	24	25	26		

compete for resources, their lots cannot be processed alternatively since part types 1 and 2 are noncompatible pairs.

It can be seen that the resulting duality gaps in Examples 2 and 3 are much larger than that in Example 1. It is also observed that for Examples 2 and 3, the dual cost rises slowly, and the zigzagging phenomenon reported in [8] is observed over the iterations. The number of iterations needed to converge depends on the geometric characteristics of dual function, the number of multipliers and the algorithm used. The characteristics of dual function are currently under investigation, and a more effective algorithm to update the multipliers is currently under development.

5. CONCLUSION

In this paper, the scheduling of a machining center with two pallets is addressed. A novel integer programming formulation considering machine setups and tool overlapping is presented. The difficulty caused by tool overlapping is handled by translating tool magazine capacity constraints into "noncompatible pairs" of part types. With the "separable" problem formulation, a solution methodology based on Lagrangian relaxation has been developed, and the decomposed subproblems have plausible interpretations and can be efficiently solved. Effective schedules with quantifiable quality can thus be generated by iteratively solving the subproblems and updating

the multipliers. Testing results indicate that the method is promising, and the algorithm provides a valuable direction for solving stand-alone machining center problems.

It can be shown that the dual function D is concave and piecewise linear, and composed by many facets. The subgradient method may exhibit slow convergence because multipliers tend to zigzag across intersections of facets (as described in [8] and evidenced by Example 2 and 3). To overcome this difficulty, the recently developed "reduced-complexity bundle method" (RCBM) will be used to update the multipliers [9], and the algorithm is currently under development. Problems of larger sizes will then be tested.

Since Lagrangian relaxation is essentially a decomposition and coordination technique, the method presented here can be extended to handle multiple machining centers within a larger environment, e.g., within a job shop or a flexible manufacturing system. The idea is based on the fact that machines in such a system are essentially independent, but are coupled through operation precedence constraints. By introducing Lagrange multipliers to relax these constraints as in [7], or by applying the combined Lagrangian relaxation and dynamic programming method as presented in [6], the overall problem can be decomposed into small subproblems which can be solved without encountering major complexity difficulties. The method developed in this paper can then be applied, and the subproblem solutions are coordinated through the iterative updating of multipliers.

REFERENCES

1. K.E. Stecke, Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems, *Management Science* **29**, 273–288, (1983).
2. C.Y. Lee, S.D. Liman and A. Wirakusumah, Product batching and batch sequencing for NC punch presses, *Int. J. Prod. Res.* **31** (5), 1143–1156, (1993).
3. J. Slomp and J.N.D. Gupta, Interactive tool for scheduling jobs in a flexible manufacturing environment, *Computer-Integrated Manufacturing Systems* **5** (4), 291–299, (1992).
4. E. Aanen, G.J. Gaalman and W.M. Nawijn, A scheduling approach for a flexible manufacturing system, *Int. J. Prod. Res* **31** (10), 2369–2385, (1993).
5. M. Pinedo, *Scheduling—Theory, Algorithms and Systems*, Prentice Hall, (1995).
6. H. Chen and J.M. Proth, A more efficient Lagrangian relaxation approach to job-shop scheduling problems, *Proceedings of the 1995 IEEE International Conference on Robotics and Automation*, May 1995, Nagoya, Japan, (to appear); and personal communication.
7. P.B. Luh and D.J. Hoitomt, Scheduling of manufacturing systems using the Lagrangian relaxation technique, *IEEE Trans. on Automat. Contr.* **38** (7), 1066–1079, (1993).
8. C. Czerwinski and P.B. Luh, Scheduling products with bills of materials using an improved Lagrangian relaxation technique, *IEEE Transactions on Robotics and Automation* **10** (2), 99–111, (April 1994).
9. R.N. Tomastik and P.B. Luh, A reduced-complexity bundle method for maximizing concave nonsmooth functions (submitted).