# A Practical Approach to Job-Shop Scheduling Problems

Debra J. Hoitomt, *Member, IEEE,* Peter B. Luh, *Senior Member, IEEE,* and Krishna R. Pattipati, *Senior Member, IEEE*

***Abstract*— Scheduling is one of the most important issues in the planning and operation of manufacturing systems, but the generation of consistently good schedules has proven to be extremely difficult. The problem is that optimal scheduling solutions involve costly and impractical enumeration procedures, while the performance of most heuristic techniques is difficult to estimate and varies considerably from one problem to the next. Recently, scheduling methodologies based on Lagrangian relaxation have proven to be computationally efficient and have provided near-optimal solutions to identical, parallel machine scheduling problems. In this paper, we explore the use of Lagrangian relaxation to schedule job shops, which include multiple machine types, generic precedence constraints, and simple routing considerations. Using an augmented Lagrangian formulation, the scheduling problem is decomposed into operation-level subproblems for the selection of operation beginning times and machine types, with given multipliers and penalty coefficients. The multipliers and penalty coefficients are then updated at the high level. The solution forms the basis of a list-scheduling algorithm that generates a feasible schedule. A procedure is also developed to evaluate the quality of this feasible schedule by generating a lower bound on the optimal cost. Numerical examples are taken from a representative industrial job shop. High-quality schedules are efficiently generated every other day over a three-week period, with costs generally within 4% of their respective lower bounds. The methodology compares favorably with a knowledge-based scheduling method used in the shop at the time of the numerical tests.**

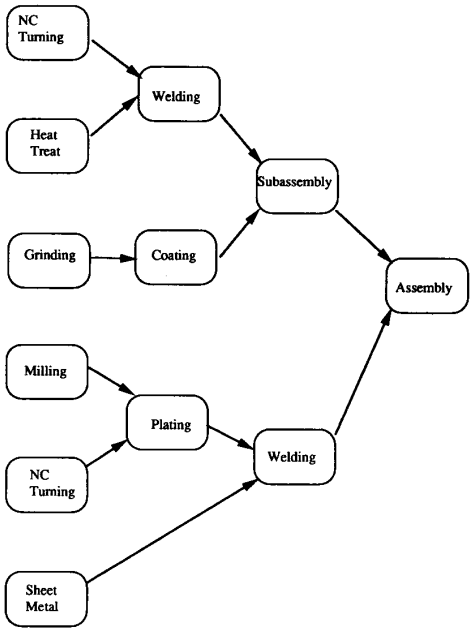



Fig. 1. A sample process plan.

## I. Introduction

TRANSFER lines have long been established as the most efficient method of producing goods in a high-volume/low-variety manufacturing environment. Low-volume/high-variety and mid-volume/mid-variety manufacturing, however, have always been plagued with difficulties. Lead times and work-in-process inventories are often excessive, and machine utilization is generally low. Many of these production problems are attributable to problems in the scheduling function: not having the right items when they are needed, not having equipment available when it is needed, using excess inventory to "hide" problems, and inflexibility and lack of responsiveness. In addition, the economic significance of these problems can be broadly defined. According to Dr. Eugene Merchant of Metcut Research Associates, Inc., approximately 50–75% of manufactured parts falls into the low volume/high variety and mid-volume/mid-variety categories and, with the trend toward variety in products, this percentage is likely to increase.[1]

This paper presents a solution methodology for the scheduling of job shops, a typical environment for the manufacture of low-volume/high-variety products. In a job shop, there are jobs with various levels of importance and due dates to be processed by many types of machines (grinding, drilling, etc.). Each job requires a sequence of operations for completion, and each operation requires a specific type(s) of machine for a specified duration of time. An operation may begin only when all its preceding operations have been completed. See Fig. 1 for a sample job with operations shown according to its process plan. The process plans are usually different for different jobs. The capacity of each type of machine is finite and may be time varying.

Because of the intrinsic difficulties of job-shop scheduling, optimal scheduling methodologies require expensive and time-

Manuscript received October 8, 1990; revised March 31, 1992. This work was supported in part by the Department of Higher Education of the State of Connecticut and Pratt & Whitney under the Cooperative High Technology Research and Development Grant Program, and by the National Science Foundation under Grants ECS-8717167 and DDM-9119074.

D. J. Hoitomt is with Pratt & Whitney, East Hartford CT 06108.

P. B. Luh and K. R. Pattipati are with the Department of Electrical and Systems Engineering, University of Connecticut, Storrs, CT 06269-3157.

IEEE Log Number 9203645.

[1] Metcut Research Associates, Inc., is a Cincinnati based consulting company. Cook [10] cites Dr. Merchant's original estimate in 1975, and an update was provided by Dr. Merchant in April 1989 via personal communication.

consuming enumeration [24]. Since a job shop may involve hundreds of machines and thousands of jobs to be scheduled over a period varying from several months to a few years, the size of the scheduling problem makes such methodologies impractical. Thus, there is a pressing need for a practical scheduling methodology [2]. Our goal, therefore, is not to obtain the optimal schedule. Rather, we want to obtain a *near-optimal* schedule with *quantifiable* performance and within *reasonable* computation time.

This paper addresses the job-shop scheduling problem by using an *augmented* Lagrangian relaxation approach. This method is unique (to the best of our knowledge) for job-shop scheduling problems, and it displays many advantages over other approaches. The scheduling problem is decomposed into operation-level subproblems for the selection of operation beginning times and machine types, with given multipliers and penalty coefficients. The multipliers and penalty coefficients are then updated at the high level. The solution forms the basis of a list-scheduling algorithm that generates a feasible schedule. A procedure is also developed to evaluate the quality of this feasible schedule by generating a lower bound on the optimal cost. Our method is designed for the practical generation of schedules in industrial job shops. The use of the method is demonstrated by scheduling a bank of numerical control (NC) machines in an actual job shop over a three-week period.

The job-shop scheduling problem has been the subject of intensive investigation for at least 30 years. As such, Section II is devoted to a description of the various approaches. An integer programming problem formulation of the job-shop scheduling problem is then presented in Section III. Section IV describes the Lagrangian relaxation framework, the decomposed solution methodology, and the derivation of a feasible schedule. To evaluate the quality of the schedule, the second problem formulation is described in Section V. Finally, numerical results presented in Section VI demonstrate the potential of the approach to schedule job shops of realistic sizes.

## II. Literature Review

In the scheduling of job shops, the most common methodology is materials requirement planning (MRP). However, MRP is mostly a planning tool and is not really designed for detailed-level scheduling [36]. In many companies, scheduling is performed by experienced shop-floor personnel with pencil, paper, a few graphical aids (such as a Gantt chart) and perhaps a modern industrial database [17], [29], [36]. Simple dispatching rules are often used for solving immediate problems, such as sequencing at the work-center level [29], [36]. The result can be scheduling chaos, where completion dates cannot be predicted and work-in-process (WIP) inventory builds. Sometimes, even high-level management must chase down high-priority jobs on the shop floor [2].

Many dispatching rules have been presented and implemented based on due dates, criticality of operations, processing times, and resource utilization [6]. The "critical ratio," defined by one definition as a ratio of remaining processing time over remaining time to due date, has been very popular in job shops [17]. More complicated heuristics take into account some combination of the above factors. For example, Viviers' [35] algorithm incorporates three priority classes in the shortest processing time (SPT) rule. Each job is assigned an index equal to its processing time plus a value graded to its priority class. High-priority jobs have low index values and are processed first according to the SPT rule. Heuristics have been comparatively evaluated (e.g., [3]). Many artificial intelligence (AI) approaches [16], [23], [31] also use dispatching rules or heuristics for scheduling. It is generally very difficult to evaluate the performance of schedules generated by these methods. The results may also depend upon the initial ordering of jobs. This implies that minor changes in jobs and/or resource availability from one day to the next may result in quite different schedules.

There has been a great deal of effort concentrated on optimization methodologies. The methods in this category include dynamic programming [19] and the branch-and-bound method [12]. These methods require at least partial enumeration of possible sequences. Because the number of possible sequences grows exponentially as the problem size increases [24], these methods become very computation intensive for even small-sized job shops. Carlier and Pinson [8], for example, solve *for the first time* a ten-job, ten-machine job-shop problem originally posed in a 1963 industrial engineering textbook. Though this is not a large problem compared to industry standards, the optimal solution required some 4 h (17 982 s) of CPU time on a PRIME 2655 computer. Additionally, the schedule will no longer be optimal after the arrival of a new job or the breakdown of a machine. Regeneration of the schedule means another excessively long computer run to obtain a new schedule.

Attempts to bridge the gap between heuristic approaches and optimization approaches have also been undertaken (e.g., Fisher *et al.* [15], Adams *et al.* [1], Luh *et al.* [27]). In Adams *et al.*, for example, a heuristic for a job-shop problem was developed based upon optimally solving single machine sequencing problems. A criterion for measuring machine busyness was developed, and the job sequence for the busiest machine (the bottleneck) was first developed. The job sequence for the next busiest machine was then determined, and the solution was fed back into the previously solved machine problem by a "local reoptimization." However, schedule evaluation could only be achieved through "selective enumeration." Recently, the Lagrangian relaxation technique has been used to solve scheduling problems. The method can decompose a problem into a number of smaller subproblems, which are easier to solve. It can also provide a tight lower bound on the optimal cost. Fisher [12] uses the Lagrangian relaxation lower bound to obtain a more efficient enumeration method for a class of job-shop scheduling problems. More recently, we have used the technique to obtain near-optimal solutions (within one scheduling of parallel, identical machines [20], [27]). In this paper, we explore the use of Lagrangian relaxation to schedule job shops that include multiple machine types, generic precedence constraints, and simple routing considerations.

The method developed here is an extension of our previous work on the scheduling of parallel, identical machines for single-operation jobs [27] and for multioperation jobs with simple fork/join-type precedence constraints [20]. Although it is more than likely that these scheduling problems are all NP-hard, the effort required to find schedules for the different problem types may vary considerably. In general, precedence constraints complicate the scheduling problem [25]. For example, in [20], a computationally intensive enumeration procedure was used to handle precedence constraints, a step not required by the parallel machines scheduling methodology of [27]. However, the approach of [20] is impractical if a job has more than three or four operations. In our current work, a decomposition approach is used to handle precedence constraints, making the approach amenable for jobs with generic precedence structures. Nonidentical machines and routing considerations further complicate the problem.

## III. Problem Formulation

The discrete-time, integer programming formulation developed here follows the model of [20] and [27]. It has also been influenced by the work of Bruvold and Evans [7] in some variable definitions; Norbis and Smith [30] in some constraint statements; and generally by Everett [11], Fisher [12], Fisher *et al.* [14], and Conterno and Ho [9]. First, the following variables will be defined, where operation $j$ of job $i$ is referred to as operation $(i, j)$.

- $\delta_{ijkh}$ Integer variable equal to one if operation $(i, j)$ is active on machine type h at time k.
- $b_{ij}$ Beginning time of operation $(i, j)$.
- $c_{ij}$ Completion time of operation $(i, j)$.
- $C_i$ Completion time of job $i$.
- $D_i$ Due date of job $i$.
- $H$ Number of machine types.
- $H_{ij}$ Set of machine types capable of performing operation $(i, j)$.
- $I_{ij}$ Set of operations of job $i$ immediately following operation $(i, j)$.
- $J$ Objective function to be optimized.
- $K$ Time horizon under consideration.
- $M_{kh}$ Capacity of machine type $h$ at time $k$.
- $m_{ij}$ Machine type selected to process operation $(i, j)$; $m_{ij} \epsilon H_{ij}$
- $N$ Number of jobs.
- $N_i$ Number of operations for job $i$.
- $S_{ijl}$ Fixed period of time between operations $(i, j)$ and $(i, l) \epsilon I_{ij}$ when the job is not available for scheduling (required timeout).
- $T_i$ Tardiness of job $i$, defined as $\max[0, C_i - D_i]$.
- $t_{ijh}$ Processing time of operation $(i, j)$ on machine type $h \epsilon H_{ij}$.
- $w_i$ Weight (value or importance) of job $i$.

It is assumed that the precedence constraints of a job form a directed acyclic graph and, without loss of generality, that each job ends with a single operation so that $C_i = c_{i,N_i}$. Job processing is assumed to be nonpreemptive so that a contiguous block of time of length $t_{ijh}$ is needed to process operation $(i, j)$ on machine type $h \epsilon H_{ij}$. All jobs are considered available for processing at time 1 (to be relaxed in Subsection IV-C), and the time horizon $K$ is assumed to be long enough to complete all the jobs (i.e., $C_i \leq K$ for all $i = 1, 2, \ldots, N$).

| Operation | Processing Time | Machine Type |
|---|---|---|
| 1 | 3 | 2 |
| Timeout | 2 | -- |
| 2 | 2 | 1 |

Weight = 1
Due Date = 9

Fig. 2. Work order for job $i$.

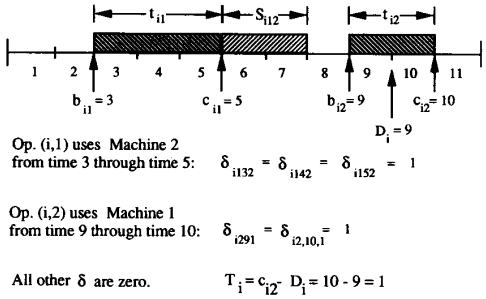



Fig. 3. A scheduled job.

Among the above variables, the time horizon $K$, the number of jobs $N$, the weights of the jobs $\{w_i\}_{i=1}^{N}$, precedence structures of jobs $\{I_{ij}\}_{i=1,j=1}^{N\ N_i-1}$, processing time requirements $\{t_{ijh}\}_{i=1,j=1,h\epsilon H_{ij}}^{N\ N_i}$, due dates $\{D_i\}_{i=1}^{N}$, the number of machine types $H$, machine capacity $\{M_{kh}\}_{k=1,h=1}^{N\ H}$, and the operation to machine capability mapping $\{H_{ij}\}_{i=1,j=1}^{N\ N_i}$ are assumed to be given. The required timeouts $S_{ijl}$ are assumed to be given, and are used to model time spent on machines or tasks that are not considered in the problem formulation (e.g., travel time, office work, or inspection). The decision variables are the beginning times of all operations $\{b_{ij}\}$ and the machine types $\{m_{ij}\}$. Once they are selected, $\{c_{ij}\}$, $\{C_i\}$, $\{T_i\}$, and $\{\delta_{ijkh}\}$ can be easily derived. Although $\delta_{ijkh}$ appears to have a high dimensionality, the fact that $\delta_{ijkh} = 0$ for $h \neq m_{ij}$ can be used to reduce memory requirements. See Fig. 2 for a sample job and Fig. 3 for its schedule.

The cost function to be minimized is a weighted quadratic tardiness function of the jobs.

$$J \equiv \Sigma_i w_i T_i^2. \tag{1}$$

This tardiness objective function accounts for the values of jobs, the importance of meeting due dates, and the fact that a job becomes more critical with each time unit after passing its due date.[2]

Research into practical scheduling has shown that this objective is likely to be more useful than, say, makespan

[2] Luh *et al.* [27] used the sum of weighted tardiness rather than weighted quadratic tardiness. For the weighted tardiness function, the incremental penalty of a job does not change as the tardiness increases. Thus, for example, for two jobs with weight one, both jobs one day late (objective function $J = 2$) is equivalent to one job two days late ($J = 2$). The weighted quadratic tardiness function resolves this ambiguity. Here, two jobs one day late has lower cost ($J = 2$) than the situation of one job being two days late ($J = 4$).

criteria in actual manufacturing systems [6]. In addition, the additivity of the cost function facilitates the decomposition approach.

A static and deterministic scheduling problem can now be formulated as follows:

$$P: \min_{\{b_{ij}\},\{m_{ij}\}} J, \quad \text{with } J \equiv \Sigma_i w_i T_i^2 \tag{2}$$

subject to precedence constraints:

$$c_{ij} + S_{ijl} + 1 \le b_{il} \quad (i = 1,2,\ldots,N; j = 1.2,\ldots,N_i - 1; l \epsilon I_{ij}) \tag{3}$$

capacity constraints:

$$\Sigma_{ij} \delta_{ijkh} \le M_{kh} \quad (k = 1,2,\ldots,K; h = 1.2,\ldots,H) \tag{4}$$

and processing time requirements:

$$c_{ij} - b_{ij} + 1 = t_{ijm_{ij}} \quad (i = 1.2,\ldots,N; j = 1.2,\ldots,N_i). \tag{5}$$

As noted in Section II, this problem formulation is similar to that of [20], except that capacity constraints (4) consider different types of machines, precedence constraints (3) are no longer restricted to the simple fork/join type, and constraint (5) reflects the fact that operation $(i, j)$ can be processed on a few machine types with different processing time requirements (i.e., simple routing for operations).

## IV. Solution Methodology

The complexity of the scheduling problem motivates a decomposition approach. Lagrangian relaxation (LR) has been used to relax the capacity constraints in [27] and in [20] to achieve a decomposition of the scheduling problem by job. In the latter paper, a cost function is computed for every possible combination of the operation beginning times of a job. The computational complexity in scheduling a job is thus an exponential function of the number of operations of the job. In this paper, a job may have a large number of operations, so that the approach of [20] is impractical. It was determined that both the precedence and capacity constraints would be relaxed by using Lagrange multipliers. This leads to additional complications, which will be explained in detail below as they occur and summarized at the end of this section.

### A. The Lagrangian Relaxation Approach

The capacity constraints (4) can be relaxed by using the nonnegative Lagrange multipliers $\pi_{kh}$, and the precedence constraints (3) can be relaxed by using the nonnegative Lagrange multipliers $\lambda_{ijl}$.

$$\min_{\{b_{ij}\},\{m_{ij}\}} \left\{ \Sigma_i \left\{ w_i T_i^2 + \Sigma_{j,l \epsilon I_{ij}} [\lambda_{ijl}(b_{ij} + t_{ijm_{ij}} + S_{ijl} - b_{il})] + \Sigma_{kh} \pi_{kh} (\Sigma_{ij} \delta_{ijkh} - M_{kh}) \right\} \right\} \tag{6}$$

subject to (5).

The Lagrangian dual to problem $P$ is

$$\max_{\pi,\lambda \ge 0} \left\{ -\Sigma_{kh} \pi_{kh} M_{kh} + \Sigma_i \min_{\{b_{ij}\},\{m_{ij}\}} \{ w_i T_i^2 + \Sigma_j [\Sigma_{l \epsilon I_{ij}} \lambda_{ijl}(b_{ij} + t_{ijm_{ij}} + S_{ijl} - b_{il}) + \Sigma_{k=b_{ij}}^{c_{ij}} \pi_{km_{ij}}] \} \right\} \tag{7}$$

where the fact that $\delta_{ijkh} = 0$ for all $h \ne m_{ij}$ has been used. The minimization operation in (6) has been brought inside the summation because the minimum of the sum is the sum of the minima when the jobs are independent. This results in a minimization subproblem for each job:

$$\min_{\{b_{ij}\},\{m_{ij}\}} \{ w_i T_i^2 + \Sigma_j [\Sigma_{l \epsilon I_{ij}} \lambda_{ijl}(b_{ij} + t_{ijm_{ij}} + S_{ijl}) - \Sigma_{l:j \epsilon I_{il}} \lambda_{ilj} b_{ij} + \Sigma_{b_{ij}}^{c_{ij}} \pi_{k,m_{ij}}] \}. \tag{8}$$

For a particular operation and a particular machine type $h \epsilon H_{ij}$, (8) can be further decomposed to the following operation level subproblems:

$$\min_{\{1 \le b_{ij} \le K\}} \{ w_i T_i^2 \Delta_{jN_i} + [\Sigma_{l \epsilon I_{ij}} \lambda_{ijl} - \Sigma_{l:j \epsilon I_{il}} \lambda_{ilj}] b_{ij} + \Sigma_{b_{ij}}^{c_{ij}} \pi_{kh} ] \} \tag{9}$$

where $\Delta_{jN_i} = 1$ if $j$ is the last operation of the job and 0 otherwise, and $t_{ijh}$ and $S_{ijl}$ are constants that do not impact the beginning time selection process.

The development (6)–(9) follows the methodology developed in [27], and includes the relaxation of precedence constraints. The operation level subproblem of (9) is enumeratively solved for every candidate machine type within $H_{ij}$. The beginning time and machine type associated with the smallest cost is then selected and used to update the multipliers, to be discussed in Subsection IV-D. The selection of $b_{ij}$, however, depends heavily upon the sign of $(\Sigma_{l \epsilon I_{ij}} \lambda_{ijl} - \Sigma_{l:j \epsilon I_{il}} \lambda_{ilj})$, which is a constant for this operation level subproblem. The result is that $b_{ij}$ is very large when this term is negative and very small when it is positive. For an operation constrained by preceding and succeeding operations, neither result is acceptable since a small beginning time overlaps with preceding operations and a large beginning time overlaps with succeeding operations. The *solution oscillation* between small and large beginning times also leads to oscillations in multiplier values, making convergence to any meaningful multipliers difficult. For this reason, quadratic penalty terms were utilized to add the fine tuning necessary to balance early and late scheduling and fit all operations of a job together.

### B. The Augmented Lagrangian Relaxation Approach

The augmented Lagrangian relaxation method (or the multiplier method) has generally been applied to continuous variable problems with a great deal of success. Basically, augmented LR is regular LR with constraint-related penalty terms added to the cost function. The idea is to add a term to the objective function that induces a high cost for violation of the constraints. Associated with the method is a penalty coefficient that determines the severity of the penalty. As this coefficient is increased, the solution of the augmented Lagrangian relaxation problem approaches the solution of the original problem [26]. Bertsekas [4] reports faster convergence

$$D: \max_{\pi,\lambda\geq 0} q(\pi,\lambda)$$

with

$$q(\pi,\lambda) \equiv \min_{\{b_{ij}\},\{m_{ij}\},\{s_{ijl}\}} L = \left\{ -\Sigma_{hk}\pi_{kh}M_{kh} + \Sigma_i \min_{\{b_{ij}\},\{m_{ij}\},\{s_{ijl}\}} \left\{ w_i T_i^2 + \Sigma_j \left[ \Sigma_{l\epsilon I_{ij}} \left[ \lambda_{ijl}(b_{ij} + t_{ijm_{ij}} + s_{ijl} - b_{il}) + \frac{p_{ijl}}{2}(b_{ij} + t_{ijm_{ij}} + s_{ijl} - b_{il})^2 \right] + \Sigma_{b_{ij}}^{c_{ij}} \pi_{k,m_{ij}} \right] \right\} \right\}. \quad (12)$$

when the penalty terms are added to the LR cost function. To the best of our knowledge, the method has not been applied to job-shop scheduling problems previously.

In order to apply the method, the inequality precedence constraint (3) is converted to the following equality constraint by using a slack variable $s_{ijl}$:

$$b_{ij} + t_{ijm_{ij}} + s_{ijl} = b_{il} \quad (i = 1,2,\ldots,N; j = 1,2,\ldots,N_i - 1; l\epsilon I_{ij}) \quad (10)$$

where (5) has been utilized to substitute out $c_{ij}$. The slack variable $s_{ijl}$ is the time between operations and becomes a decision variable with the requirement that $s_{ijl} \geq S_{ijl}$. A penalty term is then formed using the above equality constraint, with penalty coefficient $p_{ijl}$. This leads to the following relaxed problem:

$$R: \min_{\{b_{ij}\},\{m_{ij}\},\{s_{ijl}\}} L$$

with

$$L \equiv \Big\{ \Sigma_i \Big\{ w_i T_i^2 + \Sigma_{j,l\epsilon I_{ij}} \Big[ \lambda_{ijl}(b_{ij} + t_{ijm_{ij}} + s_{ijl} - b_{il}) + \frac{p_{ijl}}{2}(b_{ij} + t_{ijm_{ij}} + s_{ijl} - b_{il})^2 \Big] \Big\} + \Sigma_{ij}\Sigma_{k=b_{ij}}^{c_{ij}} \pi_{km_{ij}} - \Sigma_{kh}\pi_{kh}M_{kh} \Big\}. \quad (11)$$

The multiplier method mandates that $p_{ijl}$ be positive and not decrease during the course of the optimization. The Lagrangian dual to problem $P$ is given by (12) at the top of this page. In deriving (12), the minimization operation in (11) has again been brought inside the summation because the jobs are independent. This results in a minimization subproblem for each job:

$$R_i: \min_{\{b_{ij}\},\{m_{ij}\},\{s_{ijl}\}} L_i$$

with

$$L_i \equiv \Big\{ w_i T_i^2 + \Sigma_j \Big[ \Sigma_{l\epsilon I_{ij}} \Big[ \lambda_{ijl}(b_{ij} + t_{ijm_{ij}} + s_{ijl}) + \frac{p_{ijl}}{2}(b_{ij} + t_{ijm_{ij}} + s_{ijl} - b_{il})^2 \Big] - \Sigma_{l:j\epsilon I_{il}} \lambda_{ilj} b_{ij} + \Sigma_{b_{ij}}^{c_{ij}} \pi_{k,m_{ij}} \Big] \Big\}. \quad (13)$$

At this point, further decomposition to operation-level subproblems requires some examination. In (9), the selection of $b_{ij}$ was independent of any other operation beginning time of job $i$. In (13), however, a *cross term* involving both $b_{ij}$ and $b_{il}$ is derived from the quadratic penalty function. The selection of $b_{ij}$ therefore depends upon the solutions of other operations and vice versa, and any global minimum requires an enumeration procedure like [20]. Since that is impractical, the next subsection describes a forced decomposition approach designed to obtain a good schedule for what could be a sizable problem.

### C. Scheduling Individual Operations

In the first step toward decomposing the problem of (13), all terms involving $b_{ij}$ are gathered. These consist of the following:

$$\begin{aligned} & w_i T_i^2 \Delta_{jN_i} + \Sigma_{l\epsilon I_{ij}} \Big[ \lambda_{ijl}(b_{ij} + t_{ijh} + s_{ijl} - b_{il}) + \frac{p_{ijl}}{2}(b_{ij} + t_{ijh} + s_{ijl} - b_{il})^2 \Big] \\ & + \Sigma_{l:j\epsilon I_{il}} \Big[ \frac{p_{ilj}}{2}(b_{il} + t_{ilm_{il}} + s_{ilj} - b_{ij})^2 \\ & + \lambda_{ilj}(b_{il} + t_{ilm_{il}} + s_{ilj} - b_{ij}) \Big] + \Sigma_{k=b_{ij}}^{c_{ij}} \pi_{kh} \end{aligned} \quad (14)$$

where $\Delta_{jN_i} = 1$ if $j$ is the last operation of the job and 0 otherwise. Note that two terms are derived from interaction with preceding operations and two from interaction with succeeding operations. The Gauss–Seidel *iterative* approach [28] is adopted to overcome this interdependency. A Gauss–Seidel iteration for a job consists of solving *all* operation-level subproblems from the *first to the last operation of* the job. The initialization of the beginning times may be the earliest start dates of all the operations, considering processing times and required timeouts. In solving a particular subproblem relating to operation $(i,j)$, the latest available $\{b_{il}\}$ are used and are treated as constant. Because the operations are solved in a particular order, the beginning times of all preceding $(l < j)$ operations have been solved in the current Gauss–Seidel iteration, while the beginning times of all succeeding $(l > j)$ operations are taken from the previous iteration (or initialization if it is the first iteration). At the end of a Gauss–Seidel iteration, if the solutions to the current iteration are the same as the previous iteration, the Gauss–Seidel is said to have converged. Otherwise, the operation beginning times obtained are used to start the next Gauss–Seidel iteration. If convergence is not obtained after a fixed number of iterations, the solution generating the smallest cost is used. However,

whether the Gauss–Seidel converges or not, the cost generated may not be a global minimum [5].[3]

Let $\bar{b}_{il}$ represent the beginning time of the succeeding operation $(i,l)$, computed in the previous iteration. Given all $\pi$, $\lambda$, and $h\epsilon H_{ij}$, the operation cost function to be minimized is

$$\bar{R}_(ijh): \min_{\{b_{ij}\},\{s_{ijl}\}} \bar{L}_{ijh}$$

with

$$\begin{aligned}\bar{L}_{ijh} \equiv\ & w_i T_i^2 \Delta_{jN_i} + \Sigma_{l\epsilon I_{ij}}\left[\lambda_{ijl}(b_{ij}+t_{ijh}+s_{ijl}-\bar{b}_{il})\right.\\ & \left.+\frac{p_{ijl}}{2}(b_{ij}+t_{ijh}+s_{ijl}-\bar{b}_{il})^2\right]\\ & +\Sigma_{l:j\epsilon I_{il}}\left[\frac{p_{ilj}}{2}(b_{il}+t_{ilm_{il}}+s_{ilj}-b_{ij})^2\right.\\ & \left.+\lambda_{ilj}(b_{il}+t_{ilm_{il}}+s_{ilj}-b_{ij})\right]+\Sigma_{k=b_{ij}}^{c_{ij}}\pi_{kh}.\end{aligned} \tag{15}$$

In determining the $s_{ijl}$ that gives the minimum value of (15), we first note that $\bar{L}_{ijh}$ in (15) is quadratic with respect to $s_{ijl}$. Given $b_{ij}$, it can be seen that $s_{ijl}=\max[S_{ijl}, s^r_{ijl}]$, where $s^r_{ijl}$ is the rounded integer value of $[\bar{b}_{il}-(b_{ij}+t_{ijm_{ij}}+\lambda_{ijl}/p_{ijl})]$. Likewise, $s_{ilj}=\max[S_{ilj}, s^r_{ilj}]$ . The variables $b_{ij}$ are then obtained by enumerating the beginning times from 1 through the time horizon $K$ for each possible machine type $h\epsilon H_{ij}$ and directly comparing the values of $\bar{L}_{ijh}$ derived from each possibility. The machine type that generates the smallest solution is defined as $m_{ij}$, and its cost is $\bar{L}_{ijm_{ij}}$ or simply $\bar{L}_{ij}$. The complexity of obtaining the beginning time and machine type for each operation for one Gauss–Seidel iteration is $O(K*|H_{ij}|)$, where $|H_{ij}|$ is the cardinality of $H_{ij}$. Note that if a job is not available until an earliest start time, this enumeration process begins at the earliest start time rather than at time 1.

The costs generated by the solution of (15) are now added together to obtain the job subproblem cost of (13). Note that, because of the way (14) is formed, the terms due to precedence constraints are doubly counted. Therefore, the sum of $\bar{L}_{ij}$ must be adjusted to form $\bar{L}_i$, as follows:

$$\begin{aligned}\bar{L}_i =& \Sigma_j\left\{\bar{L}_{ij}-\Sigma_{l\epsilon I_{ij}}\left[\frac{p_{ijl}}{4}(b_{ij}+t_{ijm_{ij}}+s_{ijl}-\bar{b}_{il})^2\right.\right.\\ & \left.+\frac{\lambda_{ijl}}{2}(b_{ij}+t_{ijm_{ij}}+s_{ijl}-\bar{b}_{il})\right]\\ & -\Sigma_{l:j\epsilon I_{il}}\left[\frac{p_{ilj}}{4}(b_{il}+t_{ilm_{il}}+s_{ilj}-b_{ij})^2\right.\\ & \left.\left.+\frac{\lambda_{ilj}}{2}(b_{il}+t_{ilm_{il}}+s_{ilj}-b_{ij})\right]\right\}.\end{aligned} \tag{16}$$

Also, since the Gauss–Seidel technique may not produce the minimum operation costs, the value $\bar{L}_i$ is an approximation to the optimal job cost of (13). Nevertheless, $\bar{L}_i$ is used to maximize the Lagrangian dual cost $\bar{q}(\pi,\lambda)$ following (12):

$$\bar{D}: \max_{\pi,\lambda\geq 0} \bar{q}(\pi,\lambda)$$

[3]If the function is convex and continuous, the Gauss–Seidel iterative technique is guaranteed to converge to an optimal solution [5]. No convergence result can be found for the class of problems considered here.

with

$$\bar{q}(\pi,\lambda)=\{-\Sigma_{hk}\pi_{kh}M_{kh}+\Sigma_i\bar{L}_i\}. \tag{17}$$

The maximization problem of (17) will be discussed in more detail in the next subsection.

### D. Solving the Dual Problem

To solve the dual problem related to (17), the subgradient method of our previous work [20], [27] is adopted in updating the multiplier $\pi$, and the multiplier method [4] in updating the multiplier $\lambda$ and penalty coefficient $p$. The subgradient method was originally presented by Shor [34], further explored by Polyak [33], and is commonly used to solve this type of problem (see, e.g., [32]). The Lagrange multiplier $\pi$ is updated by

$$\pi^{n+1}=\pi^n+\alpha^n g(\pi^n) \tag{18}$$

where $\alpha^n$ is the step size at the $n$th iteration, and $g(\pi)$ is the subgradient of $\bar{q}$. The subgradient component of machine type $h$ at time $k$ is equal to $(\Sigma_{ij}\delta_{ijkh}-M_{hk})$. The step size $\alpha^n$ is given by

$$\alpha^n=\beta\frac{q^U-\bar{q}^n}{g(\pi^n)^T g(\pi^n)} \tag{19}$$

where $q^U$ is an estimate of the optimal solution of (17), and $\bar{q}^n$ is the value of $\bar{q}$ at the $n$th iteration. The parameters $\beta$ and $q^U$ are changed adaptively as the algorithm converges, with the rate of change determined by testing experiences (see also [33]). This has generally been shown to improve the convergence of the subgradient algorithm (see [13]). In particular, if the value of $\bar{q}^n$ remains approximately the same over several iterations, the parameter values are decreased. The number of iterations before decrease and the percentage decrease are the values to be selected or "tuned." Since the input data may change very little from day to day in the manufacturing scheduling situation, the "tuning" of these parameters is a fairly straightforward task.

The other Lagrange multiplier $\lambda$ is adjusted according to the multiplier method update formula [4]

$$\lambda^{n+1}=\lambda^n+p^n g(\lambda^n) \tag{20}$$

where $p^n$ is the penalty coefficient at the $n$th iteration, and $g(\lambda)$ is the subgradient of $\bar{q}$ with respect to $\lambda$. The subgradient component relating the $j$th operation to the $l$th operation of job $i$ is $(b_{ij}+t_{ijm_{ij}}+s_{ijl}-b_{il})$. The penalty coefficient $p_{ijl}$ is multiplied by a factor $\gamma$ periodically (every five or ten iterations) if the subgradient component $g(\lambda_{ijl})$ is nonzero at that iteration. The parameter $\gamma$ is selected so that $p_{ijl}$ is nondecreasing, but not so large that the problem becomes ill-conditioned (Bertsekas suggests $\gamma\epsilon$ [4,7] [4, p. 123]).

The quality of the updates in (18) and (20) depends on the convergence of the Gauss–Seidel iteration of (15). As mentioned, the Gauss–Seidel method is not guaranteed to converge to the optimum. The quality of the updates is enhanced by performing a limited Armijo-type line search for

the best update values of $\pi$ and $\lambda$. The multipliers that generate the largest function value $\bar{q}$ are chosen as the updates. The algorithm is stopped when a fixed number of iterations has been reached. The results provide the basis for the construction of a feasible schedule, the topic of the next subsection.

### *E. Construction of a Feasible Schedule*

Because of the discrete decision variables involved and the stopping criterion used, the solution to the dual problem is generally associated with an infeasible schedule, i.e., some of the precedence constraints (3) and/or capacity constraints (4) might be violated. Note that the processing time requirements (5) are always satisfied in view of the way subproblems (15) are solved. To construct a feasible schedule, the dual solution is first modified to ensure that precedence constraints are satisfied. This is done by pushing all $b_{ij}$ that violate precedence constraints forward in time, starting with the second operation of each job. The list-scheduling approach of [20] is then applied. In this list-scheduling procedure, a list is created by arranging operations of all jobs in the ascending order of the modified operation beginning times. Operations are then scheduled on the required machine types according to this list as machines become available. If the capacity constraint for a particular machine type is violated at time $k$, a greedy heuristic based on the incremental change in $J$ (the original cost function (1)) determines which new operations should begin at that time slot and which ones are to be delayed by one time unit. The subsequent operations of those delayed ones are then delayed by one time unit if precedence constraints are violated. The process then repeats. The pseudocode for this heuristic is provided in Appendix I.

## V. Performance Evaluation

### *A. A Related Problem Formulation*

Since the Gauss–Seidel technique may not produce the minimal operation level costs, the value of the dual function $\bar{q}$ in (17) may not be a lower bound on the optimal cost. To evaluate the schedule, a slightly different problem formulation is employed. When standard Lagrangian relaxation is applied to this problem formulation, an effective lower bound on the optimal cost can be obtained. To further explain the technique, the following additional variables are introduced:

$\omega_{ijlk}$ integer variable equal to one for every time unit $k \leq c_{ij} + S_{ijl}$, $l \epsilon I_{ij}$, and zero otherwise;

$\sigma_{ijk}$ integer variable equal to one for every time unit $k \geq b_{ij}$, and zero otherwise.

With these new variables, the precedence constraints (3) can be replaced by the following:

$$\omega_{ijlk} + \sigma_{ilk} \leq 1 \quad (i = 1, \ldots, N; j = 1, \ldots, N_i; k = 1, \ldots, K; l \epsilon I_{ij}). \quad (21)$$

Note that $\omega_{ijlk}$ is one for every time unit less than $c_{ij} + S_{ijl}$, and $\sigma_{ilk}$ is one for every time unit greater than $b_{il}$. Therefore, the constraint (21) is satisfied if and only if the subsequent operation $(i, l)$ does not begin until the completion of the current operation $(i, j)$ and the required timeout.

With constraint (21) in place of (3), the problem formulation remains valid, and standard Lagrangian relaxation can be applied. The disadvantage of this approach, however, is that the set of Lagrange multipliers that relaxes (21) has extremely high dimensionality. It is therefore very difficult to solve the scheduling problem based on this problem formulation alone. Our idea is to use this problem formulation after solving the dual problem of Section IV, and use the multipliers $\pi$ obtained there. It has been shown that, for convex programming problems, the optimal Lagrange multipliers from the augmented Lagrangian procedure have the same values as in LR [4, p. 322]. Although no corresponding results have been found for integer programming problems, the multipliers $\pi$ obtained from Section IV are nevertheless used here to avoid computational difficulties. With $\pi$ fixed, the multipliers associated with (21) can be obtained in a jobwise manner, and the dual cost provides a lower bound on the optimal cost. The multipliers $\pi$ can also be updated a few times to improve the lower bound by using the subgradient method, although the Lagrange multipliers associated with (21) might have to be reinitialized each time if adequate memory is not available. Details of the Lagrangian relaxation framework and solution methodology are provided in Appendix II.

### *B. Evaluation of the Feasible Solution via the Approximate Duality Gap*

Once a feasible schedule is obtained, the corresponding value of the objective function $J$ is an upper bound on the optimal objective $J*$. The value of the dual function $q*(\mu, \pi)$ based on (21) is a lower bound on $J*$ [18]. The difference between $J*$ and $q*$ is known as the duality gap. An upper bound of the duality gap is provided by $J - q*$, which is a measure of the suboptimality of the feasible schedule. The approximate relative duality gaps $(J - q*)/q*$ for the examples in the next section are less than 10%. This is somewhat larger than the parallel machine results reported in [20] and [27], since the current problem is much more complicated.

### *C. Overall Structure of Algorithm*

The overall structure of the algorithm is presented in Fig. 4. The basic building blocks are the subgradient and multiplier methods for updating the multipliers and penalty coefficients, the Gauss–Seidel iterative technique for solving operation-level subproblems, the list-scheduling method to generate a feasible schedule, and the evaluation methodology with job-related dual problems discussed in Section V. The multipliers, penalty coefficients, and beginning times of all operations must be initialized. In the next section, the methodology is illustrated through three examples, where initializations used to solve each problem are given. The last example shows the day-to-day usage of the algorithm for an actual job-shop over a three-week period.

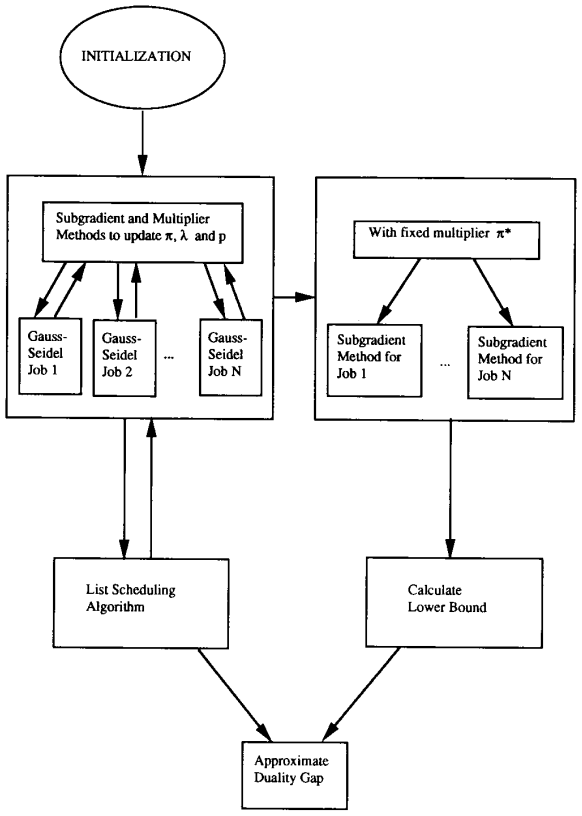



Fig. 4. A schematic of the algorithm.

TABLE I
DATA FOR EXAMPLE 1

| Job $i$ | Op. $j$ | Mach. $h$ | $t_{ij}$ | $I_{ij}$ |
|---|---|---|---|---|
| 1 | 1 | 1 | 4 | — |
| | 2 | 2 | 3 | 1 |
| | 3 | 3 | 2 | 2 |
| 2 | 1 | 2 | 1 | — |
| | 2 | 1 | 4 | 1 |
| | 3 | 3 | 4 | 2 |
| 3 | 1 | 3 | 3 | — |
| | 2 | 2 | 2 | 1 |
| | 3 | 1 | 3 | 2 |
| 4 | 1 | 2 | 3 | — |
| | 2 | 3 | 3 | 1 |
| | 3 | 1 | 1 | 2 |

TABLE II
GANTT CHART OF THE SCHEDULE FOR EXAMPLE 1

| Time | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Machine 1 | 1,1* | 1,1 | 1,1 | 1,1 | 2,2 | 2,2 | 2,2 |
| Machine 2 | 4,1 | 4,1 | 4,1 | 2,1 | 1,2 | 1,2 | 1,2 |
| Machine 3 | 3,1 | 3,1 | 3,1 | 4,2 | 4,2 | 4,2 | |
| Time | 8 | 9 | 10 | 11 | 12 | 13 | |
| Machine 1 | 2,2 | 4,3 | 3,3 | 3,3 | 3,3 | | |
| Machine 2 | 3,2 | 3,2 | | | | | |
| Machine 3 | 1,3 | 1,3 | 2,3 | 2,3 | 2,3 | 2,3 | |

*The notation "$i$, $j$" refers to operation $j$ of the job $i$.

## VI. TEST RESULTS

### A. Example 1

In the first example, there are three different machines and four equally weighted jobs ($w_i = 1$). The planning horizon is 25 days (i.e., $K = 25$, and the time unit is days). All machines are available on day 1 and throughout the planning horizon. All jobs are available for processing on day 1, but are due on day 0. Each job is composed of three serial operations to be processed on three machines without required timeouts between operations. Data are shown in Table I.

All the multipliers were initialized at zero, and all penalty coefficients $p$ were initialized at 0.5 and doubled every five iterations if the associated precedence constraint was violated at the fifth iteration. The lower bound is obtained at 469.7, and the feasible schedule has a cost $J = 475.0$ with a relative duality gap of 1.13 %; this cost is really optimal. The resulting schedule is shown in the form of a Gantt chart in Table II. The time to solve the problem is 12.7 CPU seconds on an IBM 3090 mainframe computer. Note that if several new jobs were to arrive after this schedule has been generated, the current multipliers and penalty coefficients can be used to initiate the rescheduling process. This has generally been shown to reduce the computation time compared to the case where all multipliers are initialized at zero (see [20] and [27]).

### B. Example 2

This example draws data from Pratt & Whitney's Development Operations shop. The data come from about 20 work centers relating to numerical control (NC) machines. There are a total of 32 different machines, and all of them are different (32 machine types). There are 228 jobs, each consisting of one to seven operations, for a total of 335 operations. An operation may be performed on one of a set of machine types, with the number of eligible machine types ranging from one to six. Required timeouts of varying lengths also exist between some operations. The jobs are characterized by different due dates and may have one of five different weights. The planning horizon is 414 working days, or about 1.5 years. Not all the machines are immediately available as some of them are busy processing jobs already started.

These jobs are currently scheduled by using a knowledge-based scheduler at Pratt & Whitney. The knowledge-based scheduler operates interactively with humans. The human scheduler for the NC machine group had years of experience as a scheduler in the shop and a significant amount of experience in using the interactive scheduler. Thus we can compare our results with the current schedule.

All the multipliers and penalty coefficients were initialized and adjusted as described in Example 1. The lower bound is obtained at 4 634 114.6, and the feasible schedule has a cost

TABLE III
SCHEDULE GENERATION AND RECONFIGURATION FOR NC MACHINES

| Date | Problem Size | Cost, Existing Schedule | Cost, Our Schedule | Lower Bound | Relative* Difference (%) | CPU** Time |
|---|---|---|---|---|---|---|
| 4/17 | 32×126 | 304589.0 | 274735.5 | 264488.0 | 3.9 | 2.6 |
| 4/19 | 32×143 | 305990.0 | 278108.5 | 267093.8 | 4.1 | 2.8 |
| 4/22 | 34×141 | 307934.0 | 280746.0 | 271100.4 | 3.6 | 2.9 |
| 4/24*** | 33×133 | 564532.0 | 531851.0 | 531203.2 | 0.1 | 3.0 |
| 4/26 | 32×135 | 296671.0 | 268827.5 | 266665.0 | 0.8 | 2.8 |
| 4/29 | 32×128 | 298825.5 | 268504.5 | 266122.0 | 0.9 | 2.2 |
| 5/1 | 32×124 | 303477.5 | 270584.5 | 267442.3 | 1.2 | 2.3 |
| 5/3 | 33×127 | 282516.0 | 247843.5 | 243192.0 | 1.9 | 2.1 |

*Relative difference refers to relative difference between our schedule cost and the lower bound.

**CPU time is in minutes, and was performed on an IBM 3090 mainframe computer.

***A few single-operation, high priority tardy jobs arrived on this day, which temporarily drove up the schedule cost. Since these operations immediately went into processing, the schedule cost returns to normal on 4/26.

$J = 4\,981\,197.5$ with a relative duality gap 7.49%. The time to solve the problem is 7.44 CPU minutes on an IBM 3090 mainframe computer. The existing schedule generated by the other scheduler was evaluated at 6 504 614.5, which is 40.36% above the lower bound.

### C. Example 3

As a third example, the NC machine group at the Development Operations shop was scheduled over a three-week period in Spring 1991, where the schedule is updated to reflect the arrival and departure of jobs and the latest information regarding process plans and processing times. With each schedule reconfiguration, the multipliers from the previous schedule are used to initialize the algorithm. The results are shown in Table III.

The problem size is $M \times N$, where $M$ is the number of machine types and $N$ is the number of jobs. Here, $M$ varies slightly because only those machines required by jobs were included in the problem. The time horizon $K$ is 300 days in each case. As in Example 2, each machine type has one machine, each part may have up to seven operations, and each operation may be performed on one of several (up to six) different machine types. The data and schedule for May 3, 1991 are provided in Table IV.

Day-to-day scheduling is undertaken by initializing the multipliers $\pi$, $\lambda$, and the penalty coefficients $p$ based on values from the previous schedule. The penalty coefficients $p$ were decreased by half upon initialization and subsequently increased according to the strategy discussed in Example 1. Beginning times were initialized by using backward scheduling from the due date or using forward scheduling from today if the due date could not be met. The consistently near-optimal quality of the schedules is demonstrated by the fact that all the schedules are within 5 %, whereas the quality of the schedules generated by the knowledge-based system is 15–20% above the lower bound (third column of Table III). The computation time has also been significantly improved over Example 2, since the multipliers and penalty coefficients were initialized based on results from the previous schedule.

## VII. CONCLUSION

A new and effective job-shop scheduling methodology has been presented that encompasses most of the classic features of job shops: jobs with multiple operations, generic precedence constraints, capacity constraints, and simple routing decisions. The augmented Lagrangian relaxation approach and the Gauss–Seidel iterative technique were used to facilitate a decomposition into a set of operation-level subproblems, each with the complexity of $O(K * |H_{ij}|)$. From the dual solution, a list-scheduling method was employed to generate a feasible schedule. Because of the Gauss–Seidel iterative technique employed, the dual solution did not yield a lower bound on the optimal cost. The Lagrangian relaxation solution of a second problem formulation resulted in an effective lower bound on the optimal cost, and facilitates the performance evaluation of the schedule. Compared to our previous results on parallel machines, the duality gaps and CPU times are larger but are still within reasonable limits. This is to be expected because of the additional complexity of the problem.

This approach has many advantages over existing job-shop scheduling algorithms and is well-suited for implementation in actual manufacturing environments. First, it generates a schedule and a lower bound efficiently, and the resulting schedule has generally been shown to be near optimal. In day-to-day scheduling operations, the Lagrange multipliers and penalty coefficients from the previous schedule can initialize the algorithm to generate a new schedule. Since the status of a shop may not change much from day to day, the computation time can be vastly reduced by using this initialization procedure as demonstrated in Section VI. In addition, the decomposed structure of the algorithm facilitates distributed implementation using workstations [22]. A distributed implementation can mimic the actual decision-making structure of the shop. These features have led to adoption of the method by the Development Operations shop of Pratt & Whitney as the backbone of its new scheduling system. In this implementation, only bottleneck and potential bottleneck machines are considered for scheduling. Nonbottleneck machine processing times and travel times are aggregated and considered as required timeouts. In this way, large-scale job-shop problems can be effectively handled, even in a rapidly changing environment.

## APPENDIX I
## PSEUDOCODE OF THE HEURISTIC PROCEDURE

To obtain a feasible schedule, the dual solution is first modified to ensure that the precedence constraints are satisfied, as discussed in Subsection IV-E. A list $U$ is then created by arranging operations of all jobs in the ascending order of the modified operation beginning times. Operations are scheduled on the required machine types $\{m_{ij}\}$ according to this list as machines become available. If the capacity constraint for a particular machine type is violated at time $k$, a greedy

TABLE IV
NC MACHINES DATA/SCHEDULE FOR MAY 3, 1991
(Time horizon: 300. Number of machine types: 33. Number of jobs: 124. All jobs denoted with an asterisk have weights of 0.5; otherwise, the job has a weight of 1.0.)

| Job No. | Due Date | Rte. No. | Process Time | Earliest Start Date | Start | Mach. | Other Machines Eligible to Perform Operation | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 10 | 1 | 1 | 1 | 7 | 1 | | | | | |
| 2 | 10 | 1 | 1 | 1 | 6 | 1 | | | | | |
| 3 | 9 | 1 | 3 | 1 | 11 | 1 | | | | | |
| 4 | 61 | 1 | 3 | 2 | 36 | 2 | 3 | 4 | 5 | 6 | |
| 5 | 61 | 1 | 3 | 2 | 33 | 2 | 3 | 4 | 5 | 6 | |
| 6 | -3 | 1 | 3 | 2 | 33 | 2 | 3 | 4 | 5 | 6 | |
| 7* | -85 | 1 | 2 | 1 | 1 | 9 | | | | | |
| | | 2 | 4 | 5 | 5 | 10 | | | | | |
| | | 3 | 2 | 11 | 11 | 10 | | | | | |
| 8* | 10 | 1 | 1 | 9 | 9 | 11 | 12 | 13 | 14 | | |
| 9 | 18 | 1 | 1 | 49 | 49 | 15 | | | | | |
| 10 | 145 | 1 | 4 | 49 | 56 | 15 | | | | | |
| | | 2 | 2 | 53 | 60 | 10 | | | | | |
| | | 3 | 1 | 101 | 135 | 15 | | | | | |
| 11 | 145 | 1 | 4 | 49 | 52 | 15 | | | | | |
| | | 2 | 2 | 53 | 56 | 10 | | | | | |
| | | 3 | 1 | 101 | 134 | 15 | | | | | |
| 12 | -169 | 1 | 28 | 78 | 78 | 15 | | | | | |
| | | 2 | 3 | 156 | 156 | 16 | | | | | |
| 13 | -114 | 1 | 1 | 14 | 14 | 18 | 19 | 20 | 21 | 22 | 23 |
| 14 | -123 | 1 | 1 | 14 | 14 | 22 | 18 | 19 | 20 | 21 | 23 |
| 15 | 91 | 1 | 1 | 18 | 27 | 2 | 3 | 4 | 5 | 6 | |
| | | 2 | 1 | 20 | 46 | 2 | | | | | |
| | | 3 | 1 | 24 | 50 | 5 | 2 | 4 | 5 | 6 | |
| 16 | 91 | 1 | 1 | 18 | 26 | 2 | 3 | 4 | 5 | 6 | |
| | | 2 | 1 | 20 | 32 | 2 | 3 | 4 | 5 | 6 | |
| | | 3 | 1 | 24 | 36 | 3 | 2 | 4 | 5 | 6 | |
| 17 | 91 | 1 | 1 | 18 | 25 | 2 | 3 | 4 | 5 | 6 | |
| | | 2 | 1 | 20 | 31 | 2 | 3 | 4 | 5 | 6 | |
| | | 3 | 1 | 24 | 35 | 3 | 2 | 4 | 5 | 6 | |
| 18 | 91 | 1 | 1 | 18 | 20 | 2 | 3 | 4 | 5 | 6 | |
| | | 2 | 1 | 20 | 30 | 2 | 3 | 4 | 5 | 6 | |
| | | 3 | 1 | 24 | 34 | 3 | 2 | 4 | 5 | 6 | |
| 19 | 91 | 1 | 1 | 18 | 19 | 2 | 3 | 4 | 5 | 6 | |
| | | 2 | 1 | 20 | 29 | 2 | 3 | 4 | 5 | 6 | |
| | | 3 | 1 | 24 | 33 | 3 | 2 | 4 | 5 | 6 | |
| 20 | 91 | 1 | 1 | 18 | 18 | 2 | 3 | 4 | 5 | 6 | |
| | | 2 | 1 | 20 | 28 | 2 | 3 | 4 | 5 | 6 | |
| | | 3 | 1 | 24 | 32 | 3 | 2 | 4 | 5 | 6 | |
| 21 | -65 | 1 | 1 | 1 | 1 | 4 | 2 | 3 | 5 | 6 | |
| | | 2 | 1 | 3 | 3 | 6 | 2 | 3 | 4 | 5 | |
| | | 3 | 2 | 6 | 6 | 5 | 2 | 3 | 4 | 6 | |
| 22 | -51 | 1 | 1 | 1 | 1 | 2 | 3 | 4 | 5 | 6 | |
| | | 2 | 1 | 3 | 3 | 2 | 3 | 4 | 5 | 6 | |
| | | 3 | 2 | 6 | 6 | 2 | 3 | 4 | 5 | 6 | |
| 23 | -47 | 1 | 1 | 1 | 1 | 5 | 2 | 3 | 4 | 6 | |
| 24 | -35 | 1 | 1 | 2 | 2 | 24 | | | | | |
| 25* | 2 | 1 | 2 | 1 | 1 | 25 | 26 | 27 | 28 | 29 | |
| 26 | 6 | 1 | 2 | 1 | 4 | 2 | 3 | 4 | 5 | 6 | |
| 27 | -47 | 1 | 28 | 78 | 106 | 15 | | | | | |
| | | 2 | 3 | 156 | 184 | 16 | | | | | |
| 28 | 38 | 1 | 1 | 68 | 69 | 1 | | | | | |
| 29 | 38 | 1 | 1 | 68 | 68 | 1 | | | | | |
| 30 | 161 | 1 | 1 | 77 | 119 | 1 | | | | | |
| 31 | -45 | 1 | 1 | 1 | 1 | 15 | | | | | |
| 32 | -53 | 1 | 10 | 18 | 18 | 9 | | | | | |
| 33 | 37 | 1 | 2 | 41 | 41 | 19 | 18 | 20 | 21 | 22 | 23 |
| | | 2 | 2 | 67 | 69 | 30 | | | | | |
| 34 | 37 | 1 | 2 | 41 | 41 | 18 | 19 | 20 | 21 | 22 | 23 |
| | | 2 | 2 | 67 | 67 | 30 | | | | | |
| 35 | 159 | 1 | 2 | 41 | 85 | 18 | 19 | 20 | 21 | 22 | 23 |
| | | 2 | 2 | 67 | 134 | 30 | | | | | |
| 36 | 8 | 1 | 14 | 2 | 2 | 21 | 18 | 19 | 20 | 22 | 23 |
| 37 | 12 | 1 | 7 | 2 | 4 | 18 | 19 | 20 | 21 | 22 | 23 |
| 38 | 20 | 1 | 7 | 2 | 12 | 19 | 18 | 20 | 21 | 22 | 23 |
| 39 | 40 | 1 | 7 | 2 | 19 | 18 | 19 | 20 | 21 | 22 | 23 |
| 40 | 62 | 1 | 7 | 2 | 43 | 18 | 19 | 20 | 21 | 22 | 23 |
| 41 | 84 | 1 | 7 | 2 | 57 | 18 | 19 | 20 | 21 | 22 | 23 |
| 42 | 35 | 1 | 2 | 38 | 38 | 18 | 19 | 20 | 21 | 22 | 23 |
| 43 | 35 | 1 | 2 | 38 | 38 | 22 | 18 | 19 | 20 | 21 | 23 |
| 44 | 157 | 1 | 2 | 18 | 87 | 18 | 19 | 20 | 21 | 22 | 23 |
| 45 | 157 | 1 | 2 | 23 | 89 | 18 | 19 | 20 | 21 | 22 | 23 |
| 46 | 157 | 1 | 2 | 92 | 126 | 18 | 19 | 20 | 21 | 22 | 23 |
| 47 | 157 | 1 | 2 | 92 | 124 | 18 | 19 | 20 | 21 | 22 | 23 |
| 48 | 35 | 1 | 2 | 45 | 45 | 23 | 18 | 19 | 20 | 21 | 22 |
| 49 | 35 | 1 | 2 | 49 | 49 | 23 | 18 | 19 | 20 | 21 | 22 |
| 50 | 7 | 1 | 2 | 9 | 9 | 23 | 18 | 19 | 20 | 21 | 22 |
| 51 | -53 | 1 | 1 | 1 | 1 | 3 | 2 | 4 | 5 | 6 | |
| 52 | 70 | 1 | 7 | 2 | 39 | 2 | 3 | 4 | 5 | 6 | |
| 53 | 3 | 1 | 2 | 11 | 1 | 22 | 18 | 19 | 20 | 21 | 23 |
| 54 | -4 | 1 | 2 | 12 | 12 | 18 | 19 | 20 | 21 | 22 | 23 |
| 55 | 12 | 1 | 1 | 8 | 8 | 20 | 18 | 19 | 21 | 22 | 23 |
| 56 | 29 | 1 | 4 | 2 | 4 | 9 | | | | | |
| | | 2 | 3 | 24 | 26 | 22 | 18 | 19 | 20 | 21 | 23 |
| 57 | 132 | 1 | 13 | 2 | 61 | 32 | | | | | |
| 58 | -22 | 1 | 10 | 18 | 28 | 9 | | | | | |
| 59 | 13 | 1 | 13 | 1 | 1 | 12 | 11 | 13 | 14 | | |
| 60 | -10 | 1 | 3 | 1 | 1 | 7 | 8 | | | | |
| 61 | 12 | 1 | 13 | 5 | 15 | 7 | 8 | | | | |
| 62 | 12 | 1 | 13 | 8 | 24 | 8 | 7 | | | | |
| 63 | 6 | 1 | 1 | 1 | 3 | 1 | | | | | |
| | | 2 | 1 | 7 | 9 | 1 | | | | | |
| | | 3 | 1 | 8 | 10 | 1 | | | | | |
| | | 4 | 2 | 14 | 16 | 1 | | | | | |
| | | 5 | 1 | 16 | 18 | 1 | | | | | |
| | | 6 | 5 | 24 | 26 | 1 | | | | | |
| 64* | 35 | 1 | 2 | 13 | 23 | 16 | | | | | |
| 65 | -81 | 1 | 1 | 16 | 16 | 23 | 18 | 19 | 20 | 21 | 22 |
| 66 | 37 | 1 | 1 | 9 | 27 | 18 | 19 | 20 | 21 | 22 | 23 |
| 67 | 37 | 1 | 1 | 9 | 26 | 18 | 19 | 20 | 21 | 22 | 23 |
| 68 | 71 | 1 | 5 | 3 | 36 | 11 | 12 | 13 | 14 | | |
| | | 2 | 1 | 8 | 56 | 18 | 19 | 20 | 21 | 22 | 23 |
| 69 | 88 | 1 | 5 | 3 | 10 | 11 | 12 | 13 | 14 | | |
| | | 2 | 1 | 8 | 15 | 23 | 18 | 19 | 20 | 21 | 23 |
| 70 | 18 | 1 | 1 | 51 | 51 | 15 | | | | | |
| 71 | 8 | 1 | 3 | 26 | 26 | 30 | 31 | | | | |
| | | 2 | 1 | 29 | 31 | 1 | | | | | |
| 72 | 31 | 1 | 1 | 12 | 22 | 28 | 29 | | | | |
| | | 2 | 1 | 14 | 28 | 29 | 28 | | | | |
| 73 | 35 | 1 | 1 | 12 | 24 | 28 | 29 | | | | |
| | | 2 | 1 | 14 | 32 | 29 | 28 | | | | |
| 74 | 36 | 1 | 1 | 12 | 26 | 28 | 29 | | | | |
| 75 | 39 | 1 | 1 | 12 | 28 | 28 | 29 | | | | |
| | | 2 | 1 | 14 | 35 | 29 | 28 | | | | |
| 76 | 40 | 1 | 1 | 12 | 30 | 28 | 29 | | | | |
| | | 2 | 1 | 14 | 37 | 29 | 28 | | | | |
| 7 | 42 | 1 | 1 | 12 | 32 | 28 | 29 | | | | |
| | | 2 | 1 | 14 | 39 | 29 | 28 | | | | |
| 78 | 44 | 1 | 1 | 12 | 34 | 28 | 29 | | | | |
| | | 2 | 1 | 14 | 41 | 29 | 28 | | | | |
| 79 | 31 | 1 | 1 | 12 | 21 | 28 | 29 | | | | |
| | | 2 | 1 | 14 | 27 | 29 | 28 | | | | |
| 80 | 35 | 1 | 1 | 10 | 23 | 28 | 29 | | | | |
| | | 2 | 1 | 12 | 31 | 29 | 28 | | | | |
| 81 | 36 | 1 | 1 | 11 | 25 | 28 | 29 | | | | |
| | | 2 | 1 | 13 | 33 | 29 | 28 | | | | |
| 82 | 39 | 1 | 1 | 12 | 27 | 28 | 29 | | | | |
| | | 2 | 1 | 14 | 34 | 29 | 28 | | | | |
| 83 | 40 | 1 | 1 | 12 | 29 | 28 | 29 | | | | |
| | | 2 | 1 | 14 | 36 | 29 | 28 | | | | |
| 84 | 42 | 1 | 1 | 11 | 31 | 28 | 29 | | | | |
| | | 2 | 1 | 13 | 38 | 29 | 28 | | | | |
| 85 | 44 | 1 | 1 | 12 | 33 | 28 | 29 | | | | |
| | | 2 | 1 | 14 | 40 | 29 | 28 | | | | |
| 86 | 71 | 1 | 1 | 8 | 51 | 18 | 19 | 20 | 21 | 22 | 23 |
| 87 | 71 | 1 | 1 | 8 | 50 | 18 | 19 | 20 | 21 | 22 | 23 |
| 88 | 88 | 1 | 1 | 8 | 54 | 18 | 19 | 20 | 21 | 22 | 23 |
| 89 | 88 | 1 | 1 | 8 | 52 | 18 | 19 | 20 | 21 | 22 | 23 |

*(Continued on page 11)*

TABLE IV *(continued)*
( Time horizon: 300. Number of machine types: 33. Number of jobs: 124. All jobs denoted with an asterisk have weights of 0.5; otherwise, the job has a weight of 1.0.)

| Job No. | Due Date | Rte. No. | Process Time | Earliest Start Date | Start | Mach. | Other Machines Eligible to Perform Operation | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 90 | 71 | 1 | 5 | 3 | 31 | 11 | 12 | 13 | 14 | | |
| | | 2 | 1 | 8 | 53 | 18 | 19 | 20 | 21 | 22 | 23 |
| 91 | 81 | 1 | 5 | 3 | 3 | 11 | 12 | 13 | 14 | | |
| | | 2 | 1 | 8 | 8 | 23 | 18 | 19 | 20 | 21 | 22 |
| 92 | 11 | 1 | 2 | 2 | 6 | 20 | 18 | 19 | 21 | 22 | 23 |
| 93 | 8 | 1 | 2 | 2 | 4 | 20 | 18 | 19 | 21 | 22 | 23 |
| 94 | 8 | 1 | 2 | 2 | 6 | 19 | 18 | 20 | 21 | 22 | 23 |
| 95 | 11 | 1 | 2 | 2 | 8 | 19 | 18 | 20 | 21 | 22 | 23 |
| 96 | 8 | 1 | 2 | 2 | 4 | 19 | 18 | 20 | 21 | 22 | 23 |
| 97 | 14 | 1 | 1 | 7 | 11 | 19 | 18 | 20 | 21 | 22 | 23 |
| 98 | 14 | 1 | 1 | 7 | 10 | 19 | 18 | 20 | 21 | 22 | 23 |
| 99 | 37 | 1 | 4 | 1 | 21 | 2 | 3 | 4 | 5 | 6 | |
| 100* | -1 | 1 | 2 | 1 | 2 | 5 | 2 | 3 | 4 | 6 | |
| 101* | 2 | 1 | 2 | 1 | 3 | 3 | 2 | 4 | 5 | 6 | |
| 102* | -4 | 1 | 1 | 1 | 2 | 4 | 2 | 3 | 5 | 6 | |
| 103* | 4 | 1 | 2 | 1 | 3 | 4 | 2 | 3 | 5 | 6 | |
| 104* | 7 | 1 | 2 | 1 | 5 | 3 | 2 | 4 | 5 | 6 | |
| 105 | 53 | 1 | 1 | 9 | 11 | 18 | 19 | 20 | 21 | 22 | 23 |
| | | 2 | 4 | 22 | 38 | 9 | | | | | |
| 106 | 95 | 1 | 1 | 16 | 16 | 18 | 19 | 20 | 21 | 22 | 23 |
| | | 2 | 4 | 29 | 42 | 9 | | | | | |
| 107 | 132 | 1 | 1 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| | | 2 | 4 | 29 | 46 | 9 | | | | | |
| 108 | 174 | 1 | 1 | 16 | 18 | 18 | 19 | 20 | 21 | 22 | 23 |
| | | 2 | 4 | 29 | 50 | 9 | | | | | |
| 109 | 218 | 1 | 1 | 16 | 28 | 18 | 19 | 20 | 21 | 22 | 23 |
| | | 2 | 4 | 29 | 54 | 9 | | | | | |
| 110 | 261 | 1 | 1 | 16 | 29 | 18 | 19 | 20 | 21 | 22 | 23 |
| | | 2 | 4 | 29 | 58 | 9 | | | | | |
| 111 | 304 | 1 | 1 | 16 | 156 | 18 | 19 | 20 | 21 | 22 | 23 |
| | | 2 | 4 | 29 | 232 | 9 | | | | | |
| 112 | 28 | 1 | 2 | 2 | 9 | 9 | | | | | |
| | | 2 | 1 | 13 | 23 | 23 | 18 | 19 | 20 | 21 | 23 |
| 113 | 2 | 1 | 1 | 3 | 3 | 19 | 18 | 20 | 21 | 22 | 23 |
| 114 | 2 | 1 | 1 | 2 | 2 | 19 | 18 | 20 | 21 | 22 | 23 |
| 115 | 111 | 1 | 1 | 1 | 60 | 2 | 3 | 4 | 5 | 6 | |
| 116 | 111 | 1 | 1 | 1 | 59 | 2 | 3 | 4 | 5 | 6 | |
| 117 | -6 | 1 | 1 | 2 | 2 | 16 | | | | | |
| | | 2 | 1 | 4 | 4 | 16 | | | | | |
| 118 | 17 | 1 | 1 | 5 | 15 | 18 | 19 | 20 | 21 | 22 | 23 |
| 119 | -1 | 1 | 1 | 5 | 5 | 23 | 18 | 19 | 20 | 21 | 22 |
| 120 | -27 | 1 | 1 | 1 | 1 | 6 | 2 | 3 | 4 | 5 | |
| 121 | -6 | 1 | 1 | 1 | 2 | 2 | 3 | 4 | 5 | 6 | |
| 122 | -6 | 1 | 1 | 1 | 2 | 3 | 2 | 4 | 5 | 6 | |
| 123 | 1 | 1 | 1 | 1 | 2 | 6 | 2 | 3 | 4 | 5 | |
| 124* | 56 | 1 | 1 | 3 | 29 | 16 | | | | | |
| 125* | 78 | 1 | 1 | 1 | 39 | 16 | | | | | |
| 126* | 98 | 1 | 1 | 3 | 50 | 16 | | | | | |
| 127 | 19 | 1 | 1 | 3 | 11 | 33 | | | | | |

heuristic based on the incremental change in $J$ (the original cost function (1)) determines which new operations should begin at that time slot and which ones are to be delayed by one time unit.

The incremental cost function for job $i$ is defined as

$$f(i,j) = w_i[(T_i+1)^2 - T_i^2] \tag{A1}$$

if the tardiness $T_i$ would increase when $b_{ij}$ is delayed by one time unit; otherwise, $f(i,j) = 0$. For the sake of simplicity, only operations belonging to $I_{ij}$ are checked for slackness to determine $f(i,j)$. Operations in $U$ are ordered in such a way that if $(i,j)$ is before another operation $(u,v)$, then

1) $b_{ij} \leq b_{uv}$; and

2) if $b_{ij} = b_{uv}$, then $f(i,j) \geq f(u,v)$.

A few more indices and variables required by the pseudocode will now be defined. Let $H_{ij}$ be an ordered set of machine types, each of which is capable of performing operation $(i,j)$. Let $r$ be an index of this set of machines, with $h_r$ the $r$th element of the set. Additionally, define $n$ as a time index tracking machine availability, and let $E$ be a set of unscheduled operations that cannot be scheduled between time $n$ and their respective modified beginning times $b_{ij}$. The optimal routing $m_{ij}$ from the dual solution will be stored as a temporary variable called "temp" in searching for alternative routings. Given the sequence $U$ and $\{M_{hk}\}$, the greedy heuristic algorithm works as follows:

**Step 0:** *(Initialization.)* Set $E = \emptyset$ and go to Step 1.

**Step 1:** *(Get First Job on List U.)* Determine the job and operation indices $i$ and $j$ of the first operation in $U$. Set $m = m_{ij}$, $b = b_{ij}$, temp $= m$, and $r = 1$; go to Step 2.

**Step 2:** *(Determine Machine Earliest Available Time.)* Determine the first time $l$ such that $M_{lm} > 0$, set $n = l$, and go to Step 3.

**Step 3:** *(Check Machine Availability.)* If $M_{lm} \neq 0$ for $l = n, n+1, n+t_{ijm}-1$, go to Step 4; otherwise, go to Step 5.

**Step 4:** *(Modify Machine Availability.)* If the precedence constraints related to preceding operations are not violated, schedule operation $(i,j)$ to begin at time $n$, set $M_{lm} = M_{lm} - 1$ for $l = n, n+1, \ldots, n+t_{ijm}-1$ and go to Step 9; otherwise, go to Step 5.

**Step 5:** *(Advance Time n.)* Set $n = n+1$. If $n > b$, go to Step 6; otherwise, go to Step 3.

**Step 6:** *(Select Another Machine.)* If $h_r =$ temp, then set $r = r+1$. If $r > |H_{ij}|$, set $m_{ij} =$ temp and go to Step 7; otherwise, set $m = h_r$, $r = r+1$, and go Step 2.

**Step 7:** *(Get Next Operation on List.)* If operation 2 on list $U$ has beginning time $b$, then set sequence $U = U - \{(i,j)\}$, re-index the sequence, and set $E = E \cup \{(i,j)\}$, and go to Step 1; otherwise, go to Step 8.

**Step 8:** *(Update Sequence U.)* For any unscheduled operations $(i,j) \epsilon E$ such that $b_{ij} \leq b$, modify $b_{ij} = b+1$; check all subsequent operations of job $i$ and reset those beginning times that violate precedence constraints; set $U = U \cup E$ and reform sequence $U$; set $E = \emptyset$ and go to Step 1.

**Step 9:** *(Stopping Criterion.)* Set $U = U - \{(i,j)\}$; if $U = \emptyset$, stop; otherwise, go to Step 1.

## APPENDIX II
## SOLUTION METHODOLOGY FOR OBTAINING A LOWER BOUND

In Section V, constraint (3) is replaced by a new constraint (21). When this new problem formulation is solved via Lagrangian relaxation with the multipliers $\pi$ fixed at values obtained from Section IV, a set of decomposed dual problems are obtained, each relating to a job. The details of this solution methodology are given below.

The constraints (21) and (4) are first relaxed by using Lagrange multipliers $\mu_{ijlk}$ and $\pi_{kh}$ to form the relaxed problem:

$$R' : \min_{\{b_{ij}\},\{m_{ij}\}} L'$$

with

$$L' \equiv \Big\{\Sigma_i\big\{w_i T_i^2 + \Sigma_{j,k,l\epsilon I_{ij}} \mu_{ijlk}(\omega_{ijlk} + \sigma_{ilk} - 1)\big\} + \Sigma_{ij}\Sigma_{k=b_{ij}}^{c_{ij}} \pi_{km_{ij}} - \Sigma_{kh}\pi_{kh}M_{kh}\Big\}. \quad \text{(B1)}$$

For convex programming problems, the Lagrange multipliers $\{\pi\}$ resulting from the solution of an augmented Lagrangian (corresponding to eq. (12) in our case) are equal to those obtained from the unaugmented Lagrangian (eq. (B1) in our case), as discussed in [4, p. 322]. This may not be true here because the problem we considered is nonconvex and also because the Gauss–Seidel technique is used. In view of the complexity of the problem, however, the multipliers $\{\pi\}$ in (B1) are fixed at $\{\pi*\}$, the solution of the augmented Lagrangian of (17), as a good guess. The Lagrangian dual of (B1) then becomes

$$D' : \max_{\mu \geq 0} q(\mu, \pi*)$$

with

$$q(\mu,\pi*) \equiv \min L' = \Big\{-\Sigma_{kh}\pi *_{kh} M_{kh} - \Sigma_{ijk,l\epsilon I_{ij}}\mu_{ijlk} + \min_{\{b_{ij}\},\{m_{ij}\}} \Sigma_i\Big\{w_i T_i^2 + \Sigma_j\big[\Sigma_{l\epsilon I_{ij}}\Sigma_{k=1}^{c_{ij}+S_{ijl}}\mu_{ijlk} + \Sigma_{l:j\epsilon I_{il}}\Sigma_{k=b_{ij}}^{K}\mu_{iljk} + \Sigma_{b_{ij}}^{c_{ij}}\pi*_{k,m_{ij}}\big]\Big\}\Big\}. \quad \text{(B2)}$$

In deriving (B2), the fact that $\omega_{ijlk} = 0$ for $k > c_{ij} + S_{ijl}$ and $\sigma_{ijk} = 0$ for $k < b_{ij}$ has been used. With $\{\pi*\}$ fixed, the first term becomes a constant and (B2) breaks down into $N$ separable dual problems, one for each job:

$$D_i' : \max_{\mu_i \geq 0} q_i(\mu_i, \pi*)$$

with

$$q_i(\mu_i,\pi*) \equiv \Big\{-\Sigma_{jk,l\epsilon I_j}\mu_{ijlk} + \min_{\{b_{ij}\},\{m_{ij}\}} \Big\{w_i T_i^2 + \Sigma_j\big[\Sigma_{l\epsilon I_{ij}}\Sigma_{k=1}^{c_{ij}+S_{ijl}}\mu_{ijlk} + \Sigma_{l:j\epsilon I_{il}}\Sigma_{k=b_{ij}}^{K}\mu_{iljk} + \Sigma_{b_{ij}}^{c_{ij}}\pi^*_{k,m_{ij}}\big]\Big\}\Big\}. \quad \text{(B3)}$$

Comparing (B3) to (9), we see that $b_{ij}$ in (B3) is not a linear function of the multipliers $\{\mu_{ijlk}\}$, as was the case with $\{\lambda_{ij}\}$. Therefore, the oscillation phenomenon as discussed before no longer exists. Intuitively, both sets of multipliers are associated with the overlap of precedence constrained operations. The multiplier $\lambda_{ijl}$ indicates whether or not overlap exists, whereas the multipliers $\mu_{ijlk}$ indicate where in time the overlap occurs. Because no oscillation occurs, no quadratic penalty function is required, and the minimization subproblems of (B3) are completely separable by operation. Once the optimal solution for job $i$ is generated, the $\{\mu^*_{ijlk}\}$ are disposable, and the memory required for them can be reused to solve the next job. A lower bound on the optimal cost is thus given by

$$q*(\mu,\pi*) = -\Sigma_{h,k}\pi_{hk} * M_{hk} + \Sigma_i q_i*(\mu,\pi*) \quad \text{(B4)}$$

(See [18].) The quality of this lower bound is dependent upon the multipliers $\{\pi*\}$ derived from (17).

The operation-level subproblems are solved by direct enumeration, the same as for solving (15). The dual problem is then solved by using the subgradient method of Subsection IV-D for each job. In this case, the subgradient of $q_i(\mu_i, \pi*)$ is given by $\omega_{ijlk} + \sigma_{ilk} - 1$.

## ACKNOWLEDGMENT

The authors would like to thank G. Andrus, S. Cochran, S. Bailey, C. Cook, R. Lopatka, and J. Gibbs of Pratt & Whitney for their invaluable suggestions and support.

## REFERENCES

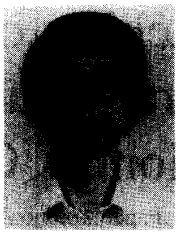
[1] J. Adams, E. Balas, and D. Zawack, "The shifting bottleneck procedure for job shop scheduling," *Management Sci.*, vol. 34, no. 3, pp. 391–401, Mar. 1988.
[2] J. E. Ashton, and F. W. Cook, Jr., "Time to reform job shop manufacturing," *Harvard Bus. Rev.*, pp. 106–111, Mar.-Apr. 1989.
[3] K. R. Baker and J. W. M. Bertrand, "A comparison of due-date selection rules," *Amer. Inst. Indust. Eng. Trans.*, vol. 13, no. 2, pp. 123–131, June 1981.
[4] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. New York: Academic, 1982.
[5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[6] J. H. Blackstone, D. T. Phillips, and G. L. Hogg, "A state-of-the-art survey of dispatching rules for manufacturing job shop operations," *Int. J. Production Res.*, vol. 20, pp. 27–45, Jan. 1982.
[7] N. T. Bruvold and J. R. Evans, "Flexible mixed-integer programming formulations for production scheduling problems," *IIE Trans.*, vol. 17, no. 1, pp. 2–7, Mar. 1985.
[8] J. Carlier and E. Pinson, "An algorithm for solving the job-shop problem," *Manag. Sci.*, vol. 35, no. 2, pp. 164–176, Feb. 1989.
[9] R. Conterno and Y. C. Ho, "Order scheduling problem in manufacturing systems," *Int. J. Production Res.*, vol. 26, no. 9, pp. 1487–1510, Sept. 1988.
[10] N. H. Cook, "Computer-managed parts manufacture," *Sci. Amer.*, vol. 232, no. 2, pp. 22–28, 1975.
[11] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Res.*, vol. 11, pp. 399–417, 1963.
[12] M. L. Fisher, "Optimal solution of scheduling problems using Lagrange multipliers, Part I," *Operations Res.*, vol. 21, pp. 1114–1127, 1973.
[13] ——, "Lagrangian relaxation method for solving integer programming problems," *Manag. Sci.*, vol. 27, pp. 1–18, 1981.
[14] M. L. Fisher, B. J. Lageweg, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Surrogate duality relaxation for job shop scheduling," *Discrete Appl. Math.*, vol. 5, pp. 65–75, Jan. 1983.
[15] M. L. Fisher, R. Jaikumar, and L. N. Van Wassenhove, "A multiplier adjustment method for the generalized assignment problem," *Manag. Sci.*, vol. 32, pp. 1095–1103, 1986.
[16] M. S. Fox and S. F. Smith, "ISIS — A knowledge-based system for factory scheduling," *Expert Syst.*, vol. 1, pp. 25–49, 1984.
[17] S. C. Graves, "A review of production scheduling," *Operations Res.*, vol. 18, pp. 841–852, 1981.
[18] A. M. Geoffrion, "Lagrangean relaxation for integer programming," *Math. Programm. Study*, vol. 2, pp. 82–114, 1974.
[19] M. Held and R. M. Karp, "A dynamic programming approach to sequencing problems," *J. Soc. Indust. Appl. Math.*, vol. 10, pp. 196–210, 1962.
[20] D. J. Hoitomt, P. B. Luh, E. Max, K. R. Pattipati, "Scheduling jobs with simple precedence constraints on parallel machines," *IEEE Contr. Syst. Mag.*, vol. 10, no. 2, pp. 34–40, 1990; also in *Dynamics of Discrete Event Systems*, Y. C. Ho, Ed. Piscataway, NJ: IEEE Press, 1992, pp. 271–277.

[21] D. J. Hoitomt, P. B. Luh, K. R. Pattipati, "Job shop scheduling with simple precedence constraints," in *Proc. Automat. Contr. Conf.* (San Diego, CA, May 1990), pp. 1–6.
[22] D. J. Hoitomt, J. B. Perkins, and P. B. Luh, "Distributed scheduling of job shops," in *Proc. IEEE Int. Conf. Robotics Automat.* (Sacramento, CA, Apr. 1991), pp. 1067–1072.
[23] A. Kuziak, *Intelligent Manufacturing Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
[24] J. K. Lenstra, A. H. G. Rinnooy Kan, and P. Brucker, "Complexity of machine scheduling problems," *Ann. Discrete Math.*, vol. 7, pp. 343–362, 1977.
[25] J. K. Lenstra and A. H. G. Rinnooy Kan, "Complexity of scheduling under precedence constraints," *Operations Res.*, vol. 26, no. 1, pp. 22–35, Jan.-Feb. 1978.
[26] D. G. Luenberger, *Linear and Nonlinear Programming*, 2md ed. Reading, MA: Addison-Wesley, 1984.
[27] P. B. Luh, D. J. Hoitomt, E. Max, and K. R. Pattipati, "Schedule generation and reconfiguration for parallel machines," *IEEE Trans. Robotics Automat.*, vol. 6, no. 6, pp. 687–696, Dec. 1990.
[28] J. H. Matthews, *Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[29] K. N. McKay, F. R. Safayeni, and J. A. Buzacott, "Job-shop scheduling theory: What is relevant?" *Interfaces*, vol. 18, pp. 84–90, July/Aug. 1988.
[30] M. I. Norbis and J. M. Smith, "Two level heuristic for the resource constrained scheduling problem," *Int. J. Production Res.*, vol. 24, pp. 1203–1219, 1986.
[31] P. S. Ow, S. F. Smith, and R. Howie, "A cooperative scheduling system," in *Proc. 2nd Int. Conf. Expert Syst. Leading Edge in Production Planning Contr.* (Charleston, SC, May 1988).
[32] K. R. Pattipati *et al.*, "CONFIDANTE: A computer-based design aid for the optimal synthesis, analysis and operation of maintenance facilities," in *Proc. IEEE AUTOTESTCON*, Nov. 1984.
[33] B. T. Polyak, "Minimization of unsmooth functionals," *USSR Comput. Math. Math. Phys.*, vol. 9, pp. 14–29, 1969.
[34] N. Z. Shor, "On the structure of algorithms for the numerical solution of optimal planning and design problems," Ph.D. dissertation, Cybernetics Inst., Academy of Sciences, USSR, 1964.
[35] F. Viviers, "A decision support system for job shop scheduling," *European J. Operational Res.*, vol. 14, no. 1, pp. 95–103, Sept. 1983.
[36] K. P. White, Jr., "Advances in the theory and practice of scheduling," in *Advances in Industrial Systems, Control and Dynamic Systems*, vol. 37, C. T. Leondes, Ed. San Diego, CA: Academic, 1990, pp. 115–158.

**Debra J. Hoitomt** (S'88–M'90) received the B.S. degree in natural science education from the University of Wisconsin, Madison, in 1977, the M.S. degree in systems and industrial engineering from the University of Arizona, Tucson, in 1984, and the Ph.D. degree in electrical and systems engineering from the University of Connecticut, Storrs, in 1990.

Beginning in 1987, she has worked with Pratt & Whitney managers and engineers to realize a next-generation scheduler for P&W's strategically important Product Development Center. In 1990, she became a Post-Doctoral Research Associate at the University of Connecticut in order to facilitate the transfer of technology from the Booth Research Center's Manufacturing Systems Laboratory to the P&W scheduling project. Her research interests include modeling and analysis of manufacturing systems, practical scheduling, large-scale integer optimization, analysis of dynamics in DEDS, queueing networks, and simulation.

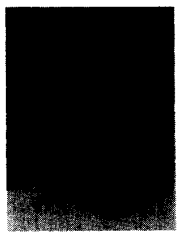
Dr. Hoitomt is the Chairperson of the Robotics and Automation Society's Computer Aided Production Systems Committee and organized a session at the Soceity's 1991 conference on practical scheduling. She is a member of Eta Kappa Nu.



**Peter B. Luh** (S'76–M'80–SM'91) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, Republic of China, in 1973, the M.S. degree in aeronautics and astronautics engineering from the Massachusetts Institute of Technology, Cambridge, in 1977, and the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA, in 1980.

Since 1980, he has been with the University of Connecticut, Storrs, where he is currently a Professor in the Department of Electrical and Systems Engineering. His major research interests include schedule generation and reconfiguration for manufacturing systems, scheduling of power systems, distributed decision making, game theory, and hierarchical planning and control of large-scale systems. He has been a principal investigator and consultant to many industry- and government-funded projects in the above areas, and has published more than 130 papers. He has made significant contributions in manufacturing by developing an efficient and near-optimal schedule generation and reconfiguration methodology currently in use in a major job shop, and reshaping directions for the scheduling research community. He has contributed to power systems by developing an efficient and near-optimal unit commitment and hydro-thermal coordination methodology currently in use by a major New England electric utility company. In the area of distributed decision making, he developed normative-descriptive models to describe and advance the knowledge of human team decision making and coordination processes in distributed task processing environments. He has made significant contributions in game theory by advancing the knowledge of incentives and by designing and implementing an incentive scheme for a large petrochemical company. His contributions to large-scale dynamic optimization include developing hierarchical algorithms with parallel processing structures and coordination mechanisms.

Dr. Luh is a Technical Editor for IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION (1990–1993), Chairperson of the Video Proceedings Committee that produced the first and second video proceedings for the IEEE International Conference on Robotics and Automation (1991, 1992), Director of Microprecision Manufacturing within the Advanced Technology Center for Precision Manufacturing of the State of Connecticut, and an Associate Editor for IEEE TRANSACTIONS ON AUTOMATIC CONTROL (1989–1991). He won the 1987 Joint Command and Control Research Symposium Best Paper Award, and has served on program committees and operating committees of many major national, international, and intersociety conferences. He is a member of Sigma Xi and is listed in *Who's Who in Engineering*, *Who's Who in the East*, and *Who's Who in American Education*.



**Krishna R. Pattipati** (S'77–M'80–SM'91) received the B.Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, India, in 1975, and the M.S. and Ph.D. degrees in systems engineering from the University of Connecticut in 1977 and 1980, respectively.

He was employed by Alphatech, Inc., Burlington, MA, from 1980 to 1986, where he supervised and performed research on human decision modeling, multitarget tracking, queuing networks, automated testing, and large-scale mixed-integer optimization. Since September 1986, he has been with the University of Connecticut, Storrs, where he is a Professor in the Department of Electrical and Systems Engineering. He has served as a consultant to Alphatech, Inc., and to the IBM Thomas J. Watson Research Center.

Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics Society as the Outstanding Young Engineer in 1984 and received the Centennial Key to the Future award. He won the best technical paper awards at the 1985 and 1990 IEEE AUTOTEST Conferences. He is currently serving at an Associate Editor of IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS (SMC). He served as the Finance Chairman of the 1989 International Conference on Control and Applications and as the Vice-Chairman for invited sessions of the 1989 IEEE International Conference on SMC. He is a member of the IEEE SMC Society AdCom.