An Effective Approach for Job-Shop Scheduling with Uncertain Processing Requirements

Peter B. Luh, *Fellow, IEEE,* Dong Chen, *Student Member, IEEE,* and Lakshman S. Thakur

***Abstract*—Production systems often involve various uncertainties such as unpredictable customer orders or inaccurate estimate of processing times. Managing such uncertainties is becoming critical in the era of "time-based competition." For example, if a schedule is generated without considering possible orders in the future, new orders of significant urgency may interrupt those already scheduled, causing serious violation of their promised delivery dates. The consideration of uncertainties, however, has been proven to be very difficult because of the combinatorial nature of discrete optimization compounded further by the presence of uncertain factors.**

This paper presents an effective approach for job-shop scheduling considering uncertain arrival times, processing times, due dates, and part priorities. A separable problem formulation that balances modeling accuracy and solution method complexity is presented with the goal to minimize expected part tardiness and earliness cost. This optimization is subject to arrival time and operation precedence constraints (to be satisfied for each possible realization), and machine capacity constraints (to be satisfied in the expected value sense). A solution methodology based on a combined Lagrangian relaxation and stochastic dynamic programming is developed to obtain dual solutions. A good dual solution is then selected by using "ordinal optimization," and the actual schedule is dynamically constructed based on the dual solution and the realization of random events. The computational complexity of the overall algorithm is only slightly higher than the one without considering uncertainties. To evaluate the quality of the schedule, a dual cost is proved to be a lower bound to the optimal expected cost for the stochastic formulation considered here. Numerical testing supported by simulation demonstrates that near optimal solutions are obtained, and uncertainties are effectively handled for problems of practical sizes.

***Index Terms*—Job-shop scheduling, Lagrangian relaxation, stochastic dynamic programming, uncertainties.**

I. Introduction

PRODUCTION systems have various uncertainties. Materials may arrive late, the processing times of one-of-a-kind parts may substantially vary from estimates, and urgent orders may arrive requiring prompt attention. Organizations also have to deal with changes in part specifications, order quantities, delivery dates, and even cancellations. It is reported that such changes occur as frequently as every four to six hours on average for a typical job-shop [10]. In this era of "time-based competition," the impact of such changes can no longer be ignored. For example, if existing parts are scheduled without considering possible orders in the future, new orders of significant urgency may interrupt those already scheduled, rendering their planned processes delayed. Since many manufacturing activities are now tightly coupled in a complicated fashion, the delay of a single operation may have a domino effect, causing the delay of subsequent operations belonging to the same part and the delay of other parts sharing the same machines. The consideration of uncertainties in scheduling, however, has been proved to be very difficult because of the combinatorial nature of discrete optimization further compounded by the presence of uncertain factors. The main aim of this work is to provide a new problem formulation and a new methodology by considering key uncertain processing requirements within a job-shop context that is one of the most prevalent manufacturing environments.

A. Literature Review

There are many stochastic scheduling results which establish the rules to determine the sequence of parts to minimize an expected objective function (e.g., [8] and [20]). Not many results, however, have been obtained for the stochastic scheduling of more than two machines, as the problems are considerably harder [18]. Scheduling problems have also been considered within the queueing framework where parts arrive at random with random processing times. Most of the results obtained in this area concentrate on performance analysis of simple scheduling policies (e.g., the "first come first serve" and "last buffer first serve" policies) by using probability and statistics theory [11], rather than the generation of optimal or near-optimal schedules.

Another method is the so called "scenario analysis" [17], [19]. The idea is that by studying possible scenarios one may come up with a "well hedged" solution. When attempting to apply this method to job-shop scheduling, the number of possible scenarios grows exponentially as the number of uncertain events increases. The method is thus effective for problems of very small sizes. To solve larger problems, many dispatching rules combined with probabilistic or fuzzy theory have been investigated (e.g., [7]). These methods have the merit of being computationally efficient and can be applied to problems of practical sizes. However, results obtained are often of questionable quality, and there is no good way to systematically improve the results.

To avoid the difficulties associated with uncertainties in scheduling, an intuitive approach is to replace all random variables by their means, consequently converting the problem into a deterministic one [18]. Existing deterministic methods can then be used to solve the converted problem. The performance

Manuscript received March 2, 1998; revised December 10, 1998. This work was supported in part by the National Science Foundation under Grants DMI-9500037 and DMI9813176. This paper was recommended for publication by Associate Editor Y. Narahari and Editor R. Volz upon evaluation of the reviewers' comments.

The authors are with the University of Connecticut, Storrs, CT 06268 USA.

Publisher Item Identifier S 1042-296X(99)03391-1.

of this method (referred to as "mean method" in this paper), however, may not be good.

B. Scope of This Paper

To develop a practical method with near optimal performance, our idea is to seek a balance between modeling accuracy and solution method complexity. Specifically, a new "separable problem formulation" for scheduling job shops with uncertain arrival times, processing times, due dates, and part priorities is presented in Section II. These uncertain parameters are treated as random variables with given discrete distributions. The problem is to minimize expected part tardiness and earliness cost, subject to arrival time constraints, operation precedence constraints, and machine capacity constraints. Arrival time constraints and operation precedence constraints are required to be satisfied for each possible realization of random events to accurately model the uncertainties. Machine capacity constraints, however, are required to be satisfied in expected values to reduce computational complexity.

To solve the problem, expected machine capacity constraints are relaxed by using Lagrangian multipliers. The problem is thus decomposed into stochastic part-level subproblems, one for each part. A subproblem is solved by using stochastic dynamic programming, with stages corresponding to operations, precedence constraints embedded in allowable state transition patterns, and state transitions governed by probabilities and scheduling decisions as presented in Section III. The close-loop nature of dynamic programming is fully exploited so that arrival time constraints and operation precedence constraints are satisfied for each possible realization of random events. The multipliers are updated at the high level by using a conjugate subgradient method, with subgradient calculated from subproblem solutions. Finally, a good dual solution is selected by using "ordinal optimization" [4], [9], and the actual schedule is dynamically constructed based on the dual solution and the realization of random events. The complexity of the overall algorithm is only slightly higher than the one without considering uncertainties [22].

To evaluate the quality of the schedule, a dual cost is proved in Section IV to be a lower bound to the optimal expected cost for the stochastic problems considered here. The quality of the schedule obtained can be thus quantitatively measured.

The method has been implemented by using the object-oriented programming language C++ under a UNIX environment, and data sets based on Delta industries, a job shop in East Granby, CT, have been tested. It is observed that through the satisfaction of arrival time constraints and operation precedence constraints for each possible realization of random events, uncertainties are effectively managed. Through the satisfaction of expected capacity constraints in the optimization process, the computational complexity is well controlled without much loss of modeling accuracy and scheduling performance, enabling the method to solve problems of practical sizes.

II. Problem Formulation

The disjunction formulation of job-shop is commonly used in the literature (e.g., [1]). However, since it does not have a "separable structure," a large problem cannot be decomposed into small subproblems by using Lagrangian relaxation to obtain near-optimal schedules. The following formulation of a stochastic job-shop scheduling problem is an extension of our earlier separable formulation [6] without considering uncertainties. In the formulation, there are K discrete time units, with index k ranging from 0 to $K-1$. There are H machine types, and the available number of type h machines ($1 \le h \le H$) at time k is given and denoted as M_{kh}. There are I parts to be processed, and part i ($1 \le i \le I$) has its arrival time a_i, due date d_i, and priority (weight) w_i. Part i is assumed to require a series of J_i operations for completion without assembly requirements, and operation j ($1 \le j \le J_i$) of part i is denoted as (i, j). The first operation of part i, $(i, 1)$, can only be started after the arrival of the order or appropriate raw materials. Operation (i, j) has to be performed on a machine of type h belonging to a given set of "eligible" machine types H_{ij} for a specified duration of time t_{ijh}, and the processing may start only after its immediate preceding operation has been completed. For some parts, the arrival time a_i, processing time t_{ijh}, due date d_i, and priority (weight) w_i are not known exactly in advance. Such parameters are modeled as independent random variables with given discrete distributions. For simplicity, machine availability is assumed to be deterministic. The objective is to maximize on-time delivery of parts and to reduce work-in-process (WIP) inventory. The problem is formulated as follows with a list of symbols provided in Appendix A for easy reference.

1) Arrival Time Constraints: The arrival time constraints state that the first operation of part i cannot be started until the arrival of order or appropriate raw materials, i.e.,

$$a_i \le b_{i1}, \qquad i = 1, \cdots, I \tag{1}$$

where b_{i1} is the beginning time of $(i, 1)$.

2) Operation Precedence Constraints: The operation precedence constraints state that operation $j+1$ of part i cannot be started before the completion of operation j of part i plus an elapse of "time-out" S_{ij} between the two operations, i.e.,

$$c_{ij} + S_{ij} + 1 \le b_{i,j+1}, i = 1, \cdots, I; j = 1, \cdots, J_i - 1 \tag{2}$$

where c_{ij} is the completion time of (i, j), and $b_{i,j+1}$ is the beginning time of $(i, j+1)$.

3) Processing Time Requirements: The processing time requirements state that operation j of part i must be assigned the required amount of processing time t_{ijh}, i.e.,

$$\begin{aligned} c_{ij} &= b_{ij} + t_{ijh} - 1, \qquad i = 1, \cdots, I; \\ & j = 1, \cdots, J_i; \quad h \in H_{ij}. \end{aligned} \tag{3}$$

The "+1" and "−1" are needed in (2) and (3), respectively, since the beginning times are assumed to refer to the beginning of a period and the completion times to the end of a period in the formulation. For example, $b_{ij} = 4$ and $c_{ij} = 5$ would imply a processing time of 2 time units, thus $c_{ij} = b_{ij} + 2 - 1$. When part i has uncertain arrival and/or processing times, (1)–(3) are required to be satisfied for each possible realization to accurately model the uncertainties. For different realizations of random parameters, the operation beginning times may be different. The beginning times may thus be random decision variables.

An uncertain customer order is an order for a part which may come with uncertain processing requirements or may not come at all. It can be formulated as a part with uncertain arrival time, processing times, due date, and part priority. The sum of probabilities associated with all possible arrival times might be less than one since the order may not come at all.

4) Machine Capacity Constraints: Machine capacity constraints state that the number of operations assigned to machine type h at time k should be less than or equal to M_{kh}, the number of machines available at that time, i.e.,

$$\sum_{ij} \delta_{ijkh} \le M_{kh}, \qquad k = 0, \cdots, K-1; \quad h \in H \tag{4}$$

where δ_{ijkh} is a 0–1 operation variable. It equals 1 if (i, j) is assigned to a machine of type h at time k, and 0 otherwise, i.e.,

$$\delta_{ijkh} = \begin{cases} 1, & \text{if operation } (i, j) \text{ is assigned to machine} \\ & \quad \text{type } h \text{ and } b_{ij} \le k \le c_{ij} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

In view of the uncertainties, what we are looking for is not a "static" schedule. Rather, we are looking for a "scheduling policy" indicating what to do under various realizable circumstances. An "implementable schedule" is a scheduling policy satisfying (1)–(4) for any realization of random parameters.

With random arrival and/or processing times, it is very difficult to handle machine capacity constraints (4) mathematically for all possible realizations of random events because of complexity. Machine capacity constraints are thus required to be satisfied in the expected sense, i.e.,

$$E\left[\sum_{ij} \delta_{ijkh}\right] \le M_{kh}, \qquad k = 0, \cdots, K-1; \quad h \in H. \tag{6}$$

The constraints (6) are approximations in the presence of uncertainties. A schedule satisfying (1)–(3) and (6) is called "model feasible."

5) Objective Function: The objective of on-time delivery of parts and low WIP inventory is translated to the minimization of penalties on part tardiness and on releasing the raw materials too early by selecting appropriate operation beginning times b_{ij} and machine types h_{ij} from a set of "eligible" machine types H_{ij}, i.e.,

$$\min_{\{b_{ij},\, h_{ij}\}} J, \text{ with } J \equiv E\left[\sum_{i=1}^{I} \left(w_i T_i^2 + \beta_i E_i^2\right)\right] \tag{7}$$

subject to constraints (1)–(3), and (6). In the above, tardiness T_i is the amount of overdue time, i.e., $\max(0, c_i - d_i)$, with $c_i = c_{iJ_i}$ the completion time of the last operation of part i. For a given part due date d_i, a desired part beginning time $\overline{b}_i$ can be roughly estimated based on the critical path of the part [6], i.e.,

$$\overline{b}_i \equiv d_i - \gamma \sum_{i=1}^{J_i} t_{ij}, \qquad 1 \le i \le N; \quad \gamma \ge 1 \tag{8}$$

where t_{ij} is the processing time of operation (i, j) along the critical path, and coefficient γ $(\gamma \ge 1)$ is related to the desired WIP level and usually chosen to be relatively small. Earliness E_i is then defined as the amount of time that part beginning time, $b_i = b_{i1}$, leads the desired beginning time, i.e., $\max(0, \overline{b}_i - b_i)$. The square on tardiness reflects the fact that a part becomes more critical with each time unit after passing its due dates and similarly for the square on earliness. Parameters w_i and β_i are given weights associated with tardiness and earliness penalties, accounting for the importance of meeting on-time completion and low work-in-process inventory. Since on-time completion is the foremost criterion in (7), β_i is usually an order of magnitude smaller than w_i. The expectation is taken with respect to random parameters and random decision variables.

A model feasible schedule satisfying (1)–(3), and (6) is usually not an implementable schedule since (4) is generally not satisfied. To obtain an implementable schedule, a list scheduling heuristic developed in Section III-D is used to dynamically construct the schedule based on optimization solution and the realization of random events.

Since (1)–(3) are linear, and (6) and (7) are additive, the formulation is thus "separable." Lagrangian relaxation (LR) technique can then be effectively applied as presented next.

III. Solution Methodology

Similar to the pricing concept of a market economy, the Lagrangian relaxation method replaces "hard" coupling constraints (expected machine capacity constraints) by "soft" prices (Lagrange multipliers) for the use of machines at each time. The original problem is thus decomposed into stochastic part-level subproblems which are effectively solved by using stochastic dynamic programming. The close-loop nature of dynamic programming is fully exploited so that arrival time constraints and operation precedence constraints are satisfied for each possible realization of random events. These prices or multipliers are then iteratively adjusted based on the degree of constraint violations following again the market economy mechanism, and these subproblems are resolved using the new set of multipliers. Finally, a good dual solution is selected by using "ordinal optimization," and an on-line heuristic is applied to adjust the dual solution selected to remove any infeasibilities and dynamically construct an implementable schedule based on the realization of random events. The overall complexity is only slightly higher than the one without considering uncertainties.

A. The Lagrangian Relaxation Framework

By using Lagrangian multipliers π_{kh} to relax expected machine capacity constraints (6), the following relaxed problem is obtained:

$$\min_{\{b_{ij},\, h_{ij}\}} L,$$

$$\begin{aligned} \text{with } L \equiv & E\left[\sum_i \left(w_i T_i^2 + \beta_i E_i^2\right)\right] \\ & + \sum_{kh} \pi_{kh} \left(E\left[\sum_{ij} \delta_{ijkh}\right] - M_{kh}\right) \\ = & E\left[\sum_i \left(w_i T_i^2 + \beta_i E_i^2\right) + \sum_i \sum_{jkh} \pi_{kh}\, \delta_{ijkh}\right] \\ & - \sum_{kh} \pi_{kh} M_{kh} \end{aligned} \tag{9}$$

subject to arrival time constraints (1), operation precedence constraints (2), and processing time requirements (3) for each possible realization. By using (5) and regrouping relevant terms, the relaxed problem can be decomposed into the following part-level subproblems:

$$\min_{\{b_{ij},h_{ij}\}} L_i, \text{ with}$$

$$L_i \equiv E\left[w_i T_i^2 + \beta_i E_i^2 + \sum_{j=1}^{J_i}\sum_{k=b_{ij}}^{c_{ij}} \pi_{kh}\right]$$
$$i = 1, \cdots, I \tag{10}$$

subject to (1)–(3).

Let L_i^* denote the resulting minimal subproblem cost. The high level dual problem is then obtained as

$$\max_{\{\pi_{kh}\}} D, \text{ with } D \equiv \sum_i L_i^* - \sum_{kh} \pi_{kh} M_{kh}. \tag{11}$$

B. Dynamic Programming for Solving Subproblems

Recently, forward dynamic programming (DP) was imbedded within the LR framework in [5] and [22] to solve part subproblems to avoid algorithm convergence difficulties as reported in [6]. However, it is well known that forward DP can not be used to solve stochastic subproblems with uncertain processing times. In this paper, backward stochastic dynamic programming is used to solve part subproblems (10) to manage uncertainties. In this procedure, each DP stage corresponds to an operation, and at each stage, the states (or nodes) are the possible operation beginning times. The subgradient component

$$\left(E\left[\sum_{ij} \delta_{ijkh}\right] - M_{kh}\right)$$

which will be needed to update the multipliers, is calculated based on subproblem results. To better illustrate the DP procedure, the deterministic case is first presented as follows.

1) DP for Deterministic Case: In this case, all parameters of part i are deterministic. The DP algorithm starts with the last stage having the following terminal cost:

$$V_{iJ_i}(b_{iJ_i}, h_{iJ_i}) = w_i T_i^2 + \sum_{k=b_{iJ_i}}^{c_{iJ_i}} \pi_{kh_{iJ_i}}. \tag{12}$$

The cumulative cost when moving backward is then obtained recursively as follows:

$$V_{ij}(b_{ij}, h_{ij}) = \min_{\{b_{i,j+1}, h_{i,j+1}\}} \left(\beta_i E_i^2 \Delta_{ij} + \sum_{k=b_{ij}}^{c_{ij}} \pi_{kh_{ij}} + V_{ij+1}(b_{i,j+1}, h_{i,j+1})\right)$$
$$= \beta_i E_i^2 \Delta_{ij} + \sum_{k=b_{ij}}^{c_{ij}} \pi_{kh_{ij}} + \min_{\{b_{i,j+1}, h_{i,j+1}\}} \cdot V_{ij+1}(b_{i,j+1}, h_{i,j+1}), \quad 1 \le j \le J_i - 1 \tag{13}$$

where Δ_{ij} is an integer variable equal to one if operation (i, j) is the first operation of part i, and zero otherwise. The second equality in (13) is derived because $\beta_i E_i^2 \Delta_{ij} + \sum_{k=b_{ij}}^{c_{ij}} \pi_{kh_{ij}}$ is a fixed value for the given b_{ij} and h_{ij}. The optimal L_i^* is then obtained as the minimal cumulative cost at the first stage, subject to the arrival time constraint. Finally, the optimal beginning times and the corresponding machine types can be obtained by tracing the stages forward. The computational complexity of the above DP technique for part i is

$$O\left(K \sum_j |H_{ij}|\right)$$

where $|H_{ij}|$ is the cardinality of H_{ij} [5].

2) DP for Uncertain Case: Similar to the deterministic case, the terminal cost for the stochastic case is given by

$$V_{iJ_i}(b_{iJ_i}, h_{iJ_i}) = E\left[w_i T_i^2 + \sum_{k=b_{iJ_i}}^{c_{iJ_i}} \pi_{kh_{iJ_i}}\right]. \tag{14}$$

The expectation is taken with respect to all possible processing times of the last operation, due dates, and weights. The recursive DP equation is

$$V_{ij}(b_{ij}, h_{ij}) = E\left[\beta_i E_i^2 \Delta_{ij} + \sum_{k=b_{ij}}^{c_{ij}} \pi_{kh_{ij}} + \min_{\{b_{i,j+1}, h_{i,j+1}\}} V_{ij+1}(b_{i,j+1}, h_{i,j+1})\right]$$
$$1 \le j \le J_i - 1 \tag{15}$$

subject to operation precedence constraints for each possible processing time of operation j. This expectation is taken with respect to all possible processing times. Finally

$$L_i^* = E[V_{i1}(b_{i1}^*, h_{i1}^*)] \tag{16}$$

subject to the arrival time constraints for each possible arrival time. This expectation is taken with respect to all possible arrival times to obtain the minimal subproblem cost.

To better understand the above, the special case with uncertain processing times only is illustrated next.

3) Solving Subproblem with Uncertain Processing Times: When the processing times $tijh$ are random and other parameters of part i are deterministic, the algorithm is as follows. For a particular b_{iJ_i} and h_{iJ_i} at the last stage, the cost is calculated by (12) for each possible processing time. The terminal cost is the expected value of all these possible costs

$$V_{iJ_i}(b_{iJ_i}, h_{iJ_i}) = E\left[w_i T_i^2 + \sum_{k=b_{iJ_i}}^{c_{iJ_i}} \pi_{kh_{iJ_i}}\right]. \tag{17}$$

To move backward to a node at stage j from stage $j+1$, the decision of which node should be selected at stage $j+1$ can be made for each possible processing time of operation j, subject to the operation precedence constraint. The associated

TABLE I
MULTIPLIERS FOR EXAMPLE 3.1

π_{01}	π_{11}	π_{21}	π_{31}	π_{41}	π_{51}	π_{61}
2	10	2.1	10	1	9	0
π_{02}	π_{12}	π_{22}	π_{32}	π_{42}	π_{52}	π_{62}
7	1	12	0	6	0	0

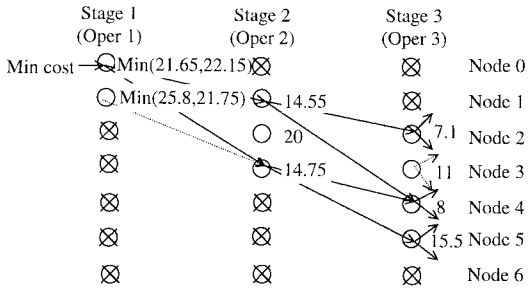


Fig. 1. DP for uncertain processing times.

cost can be obtained as in (13). The cumulative cost of the node is then the expected value of all the above costs, i.e.,

$$V_{ij}(b_{ij}, h_{ij}) = E\left[\beta_i E_i^2 \Delta_{ij} + \sum_{k=b_{ij}}^{c_{ij}} \pi_{kh_{ij}} + \min_{\{b_{i,j+1}, h_{i,j+1}\}} V_{ij+1}(b_{i,j+1}, h_{i,j+1})\right]. \tag{18}$$

This procedure continues until the cumulative costs for all the nodes at the first stage are obtained. Finally, L_i^* is selected as the minimum of the first stage cumulative costs subject to the arrival time constraint. The complexity of the algorithm is

$$O\left(K \sum_j |H_{ij}||T_{ijh}|\right)$$

where T_{ijh} is the set of all possible processing times $\{t_{ijh}\}$, and $|T_{ijh}|$ is its cardinality. This complexity is only slightly higher than that for the deterministic case.

4) Example 3.1—DP Procedure for Uncertain Processing Times: This example illustrates the DP procedure for solving a subproblem with uncertain processing times. Part i has three operations with $a_i = 0, d_i = 3, w_i = 1$, and no time-out between operations. Its processing times t_{i1h}, t_{i2h}, and t_{i3h} are either 1 with probability 0.5 or 2 with probability 0.5. Operation 1 can be performed on machine type either 1 or 2. Operation 2 needs to be performed on machine type 2 and operation 3 on machine type 1. The multipliers π_{kh} are assumed given (either from initialization or a dual solution) as shown in Table I with planning horizon $K = 7$. The state transition diagram for the DP algorithm is shown in Fig. 1.

Since the smallest processing time for operation 1 is 1, the earliest possible beginning time for operations 2 is 1. This implies that node 0 at stage 2 need not be considered. Similarly, nodes 0 and 1 at stage 3 need not be considered. Since the largest processing time for operation 3 are 2, the latest possible beginning time is 5 to complete the part within the planning horizon 7. Thus node 6 at stage 3 need not be considered. Similarly, nodes 4 to 6 at stage 2 and nodes 2 to 6 at stage 1 need not be considered.

The expected costs for node 2 to node 5 at stage 3 can first be calculated by (14). For stage 2, consider node 3 for example. Since operation precedence constraints have to be satisfied, only node 4 and node 5 at stage 3 can be selected for $t_{i2h} = 1$, and the one with a smaller cost is chosen. For $t_{i2h} = 2$, only node 5 can be selected. The expected cumulative cost for node 3 at stage 2 can be obtained by (15). This procedure then repeats. The optimal L_i^* is then selected as the minimum 21.65 among all cumulative costs at stage 1 subject to the arrival time constraint. The optimal beginning times and machine types can be determined by tracing forward the optimal DP paths based on the realizations of random arrival and processing times. The optimal beginning time b_{i1}^* for the first operation is 0 with corresponding machine type 1. The optimal beginning time for the second operation depends on the realization of random t_{i1h}: $b_{i2}^* = 1$ if $t_{i1h} = 1$, and $b_{i2}^* = 3$ if $t_{i1h} = 2$, and they correspond to two DP paths. Similarly, the optimal beginning time of the third operation depends on the realization of t_{i2h} and b_{i2}^* with four possibilities as shown in Fig. 1.

5) Calculating Subgradients: The subgradient

$$\left(E\left[\sum_{ij} \delta_{ijkh}\right] - M_{kh}\right)_{K \times H}$$

is needed to update the multipliers as will be presented in the next subsection. A key step to obtain the subgradient is to calculate the expected machine utilization $E[\delta_{ijkh}]$ for stage j of part i. To obtain this, all the nodes on the optimal DP paths for part i are located, and the probabilities that they are selected as beginning times are determined. For a node on the optimal paths at stage j, the machine utilization δ_{ijkh} can be calculated for the optimal machine type and each possible processing time. The machine utilization $E[\delta_{ijkh}]$ associated with the node is then the expected value of all these δ_{ijkh} multiplied by the probability that this node will be selected. Finally, the expected machine utilization $E[\delta_{ijkh}]$ for this stage j is the sum of the machine utilization $E[\delta_{ijkh}]$ for all these nodes. This procedure is illustrated as follows.

Let $N(1)$ denote the set of all nodes on the optimal paths at stage 1. For any node $n_1 \in N(1)$, let $A(n_1)$ denote the set of all possible arrival times having n_1 as the optimal beginning time. The probability p_{n_1} that the node is selected as beginning time is the sum of probabilities p_a associated with all these possible arrival times, i.e.,

$$p_{n_1} = \sum_{a \in A(n_1)} p_a, \qquad n_1 \in N(1). \tag{19}$$

When moving from stage j to stage $j+1$ ($1 \le j \le J_i - 1$), let $N(j+1)$ denote the set of all nodes on the optimal paths at stage $j+1$. For any node $n_{j+1} \in N(j+1)$, let $NT(n_{j+1})$ denote the set of all pairs of $n_j \in N(j)$ and possible processing time t having n_{j+1} as the optimal beginning time for operation $j+1$. The probability $p_{n_{j+1}}$ that the node n_{j+1} is selected as beginning time is thus the sum of all probabilities

$p_{n_j} p_t$ associated with these pairs (n_j, t), i.e.,

$$p_{n_{j+1}} = \sum_{(n_j,t)\in NT(n_{j+1})} p_{n_j}\, p_t, \qquad n_{j+1} \in N(j+1) \quad (20)$$

where p_t is the probability associated with the possible processing time t.

Once all these nodes are located and associated probabilities determined, the expected machine utilization $E[\delta_{ijkh}]$ for a particular (i, j) can be calculated as follows. For a node $n_j \in N(j)$ at stage j, the machine utilization δ_{ijkh} can be derived for the optimal machine type and each possible processing time. The machine utilization $E[\delta_{ijkh}]$ associated with the node is then the expected value of all these δ_{ijkh} multiplied by the probability that this node will be selected. Finally, the expected machine utilization $E[\delta_{ijkh}]$ for this stage j is the sum of the machine utilization $E[\delta_{ijkh}]$ for all the nodes $n_j \in N(j)$.

The subgradient is finally calculated by

$$E\left[\sum_{ij} \delta_{ijkh}\right] - M_{kh} = \sum_{ij} E[\delta_{ijkh}] - M_{kh}. \quad (21)$$

The complexity of calculating

$$E\left[\sum_{j} \delta_{ijkh}\right]$$

is

$$O\left(K \sum_{j} |T_{ijh}|\right)$$

because the number of the nodes at a stage is at most K. When this method is used to calculate the subgradient for a deterministic case, the complexity is the same as that of the deterministic method [15].

C. Solving the Dual Problem

The dual function D in (11) is concave, piece-wise linear, and consists of many facets [21]. Each facet corresponds to a possible scheduling policy of the relaxed problem. Because of the combinatorial nature of discrete optimization further compounded by various possible realizations of uncertain factors, the number of possible scheduling policies increases drastically as the problem size increases. Therefore for a practical problem the number of facets is extremely large, and the dual function approaches a smooth function especially near its maximum. This smoothness of the dual function motivates the use of conjugate gradient method [3] to iteratively solve the high level dual problem (11), using subgradient instead of gradient in the conjugate gradient formula. For a given set of multipliers, subproblems are solved as explained above to obtain the optimal subproblem solutions, and multipliers are then updated based on the degrees of constraint violation using the conjugate subgradient method. This iterative procedure repeats until some stopping criteria are met. The overall solution is of semi close-loop nature since close-loop subproblem solutions are obtained by using DP given Lagrangian multipliers.

D. Selecting a Dual Solution and Implementing a Schedule

Since expected machine capacity constraints (6) are relaxed in the LR process, the solutions of subproblems when put together generally do not provide a model feasible schedule, i.e., the expected machine capacity constraints (6) might be violated at particular time units. Furthermore, since the expected machine capacity constraints are a kind of approximation, a model feasible schedule satisfying (6) is generally not implementable, i.e., machine capacity constraints (4) might be violated. To obtain an implementable schedule, a list scheduling heuristic is used based on a selected dual solution and the realization of random events. How to select a good dual solution is presented first.

1) Selecting a Dual Solution: In view of the heuristic nature of how feasible schedules are constructed, a dual solution with a high dual cost may not necessarily be associated with a good feasible schedule. One therefore has to try out several candidate dual solutions having high dual costs to find one which generates a good feasible schedule. In the stochastic setting, each dual or feasible solution is in fact a policy, indicating what to do under which circumstances. To obtain the expected value of the objective function (7) for a single dual solution thus involves simulation, and is very time consuming. The idea of ordinal optimization [4], [9] is employed to perform short simulation runs on selected candidate dual solutions to determine the "ranking" of their expected costs. A winner of the short tryout is then the dual solution selected to generate implementable schedules, and rigorous simulation runs are then performed to obtain performance statistics.

2) Implementing a Schedule: To obtain an implementable schedule, the list scheduling heuristic developed is a modified version of what was presented in [15]. The difference is that the schedule here is dynamically constructed based on the realization of random events by exploiting the close-loop nature of DP solutions. In the heuristic, a list of "assignable" operations is created at time 0 and updated at each subsequent time unit based on the realization of random events and DP solutions. Operations are then scheduled on the required machine types according to this list as machines become available. If there are not enough machines of a particular type for operations, the incremental changes in cumulative DP costs of the operations are used to determine which operations should be assigned at the time slot and which ones are to be delayed by one time unit. Once an operation is delayed and scheduled for a later time, the beginning time of its succeeding operation is determined based on this scheduled time rather than its original beginning time. The procedure is illustrated as follows.

- *Step 1*: For all the operations whose raw materials are available at time 0, an operation list is created in the ascending order of their beginning times.
- *Step 2*: Operations are scheduled on the required machine types according to this operation list as machines become available.
- *Step 3*: If machine capacity constraint (4) for a particular machine type is violated at the time slot,

the operations with small incremental change in optimal cost-to-go are to be delayed by one time unit, where the incremental change in optimal cost-to-go $f(i, j)$ for b_{ij} is defined as

$$f(i, j) \equiv \min_{\{h_{ij}\}} V_{ij}(b_{ij}+1, h_{ij}) - \min_{\{h_{ij}\}} V_{ij}(b_{ij}, h_{ij}). \tag{22}$$

Step 4: Terminate the process if all operations are assigned to required machine types. Otherwise, go to next time slot.

Step 5: If a part arrives, the operation list is updated by inserting the part's first operation in the ascending order of beginning times. If operation (i, j) is completed and it has a succeeding operation $(i, j+1)$, the beginning time of $(i, j+1)$ is determined by the scheduled time and processing time of (i, j) according to the DP solution. The operation list is updated by inserting the succeeding operation in the ascending order of beginning times. Then go to Step 2.

Because of the semi close-loop nature of the subproblem solutions, rescheduling is needed periodically or after a major random event occurs with the latest information. Rescheduling can achieve better result without requiring much additional computation time if the multipliers are initialized at their previous values.

IV. Performance Evaluation

A. Performance Evaluation via Simulation

To analyze algorithm performance for large size problems, a simulation shell has been developed. Random numbers are generated according to the discrete distributions of random parameters, and Monte Carlo simulation is performed based on the dual solution selected as described in Section III-D-1. After N runs, sample cost J_n is available for run n, $1 \le n \le N$. The expected cost J can then be estimated as $J \approx \overline{J} \equiv (1/N) \sum_{n=1}^{N} J_n$. The accuracy of the estimate can be statistically evaluated based on the confidence region for a given probability of error α. Furthermore, confidence region can be used as a simple way to compare two algorithms using the same set of random variables. If the confidence regions of two methods do not overlap, the one having smaller $\overline{J}$ is better with confidence $1-\alpha$. Otherwise, a so-called "optimal" comparison technique can be used based on hypothesis testing [2].

B. Evaluation of the Solution via Duality Gap

Although simulation can be used to obtain statistics of a schedule, it cannot tell how close the schedule is to the optimal one. For the deterministic case, the dual cost has been proved to be a lower bound to the optimal cost following the "weak duality theorem" [3]. For the formulation considered here, the following theorem provides a stochastic version of the important result.

Theorem 4.1: If J^+ is the expected tardiness and earliness cost of an optimal implementable schedule, then $J^+ \ge D$.

The proof of the theorem is provided in Appendix B. The theorem states that a dual cost D is a lower bound to the expected cost J^+ of an optimal implementable schedule. The (relative) duality gap $(\overline{J} - D)/D$ can thus be used as another measure of schedule quality.

V. Numerical Results

This algorithm was implemented in C++ under a UNIX environment. Testing has been performed on a SUN ULTRA 1 workstation to compare the performance of our method with that of the "mean method." In the mean method, all random variables are replaced by their means, and the converted deterministic problems are solved by using our previous LR/DP technique [22]. In the testing, all the multipliers are initialized at zero. The conjugate subgradient algorithm for multiplier updating is terminated after a fixed amount of computation time. Based on the dual solution selected, the two methods then use the same heuristics as described in Section III-D-2 without rescheduling. In the numerical results to be presented, the first two small examples are used to show the solutions in detail and the insights obtained. Their expected costs are obtained by enumerating all possible events and determining their associated probabilities. The other examples draw data from Delta industries, an engine part manufacturer in Connecticut, to demonstrate that our method can effectively handle uncertainties for problems of practical sizes. The expected costs are obtained by Monte Carlo simulation. Each example may include several cases, and the percentage difference of the expected costs (J) between the two methods

$$\text{Difference of } J = \left(\frac{J \text{ of Mean Method} - J \text{ of Our Method}}{J \text{ of Our Method}} \right) \times 100\% \tag{23}$$

is given for each case.

Example 5.1—Scheduling with Uncertain Orders: In this example, five existing parts and a new order are to be scheduled on two different machines over a planning horizon of 14 time units. The new order (Part 6) may come on time 2 with probability 0.9, or it may not come at all. If it comes, its single operation has an uncertain processing time of either 1 time unit with probability 0.3 or 3 with probability 0.7. Data are shown in Table II, and the parameter γ in (8) is set to 1, meaning that the desired WIP level is very low.

The problem is first solved by using our method in 0.09 CPU s. The Gantt charts of resulting implementable schedule are presented in Fig. 2 with an expected cost 4.37. The lower bound D is obtained at 4.361 with a relative duality gap 0.2%. For this small example, it can be shown that the schedule is optimal by exhaustive search.

The problem is also solved by using the mean method with the same 0.09 CPU s. In the mean method, the new order is treated as coming with certainty having a processing time of 2

TABLE II
DATA FOR EXAMPLE 5.1

Part i	Op. j	Mach. h	a_i	t_{ijh}	d_i	w_i	β_i
1	1	1	0	3	2	1	0.1
2	1	1	0	1	4	1.5	0.1
	2	2		1			
3	1	1	0	1	5	1	0.05
	2	2		1			
4	1	1	0	1	6	1	0.05
	2	2		1			
5	1	1	0	1	8	1	0.05
	2	2		2			
6	1	2	6	1(0.3) or 3(0.7)	1	0.5	0.1

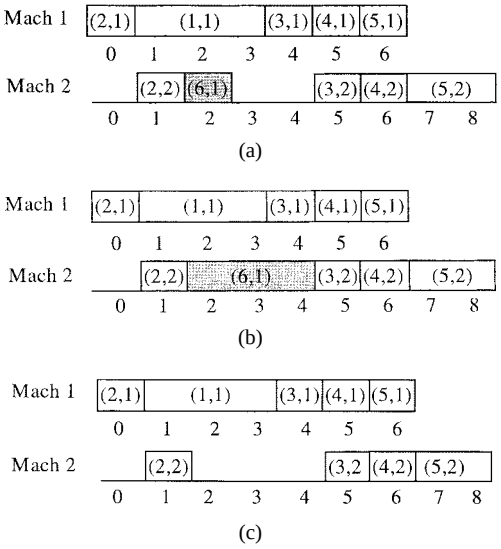


Fig. 2. Gantt charts with uncertain orders, our method.

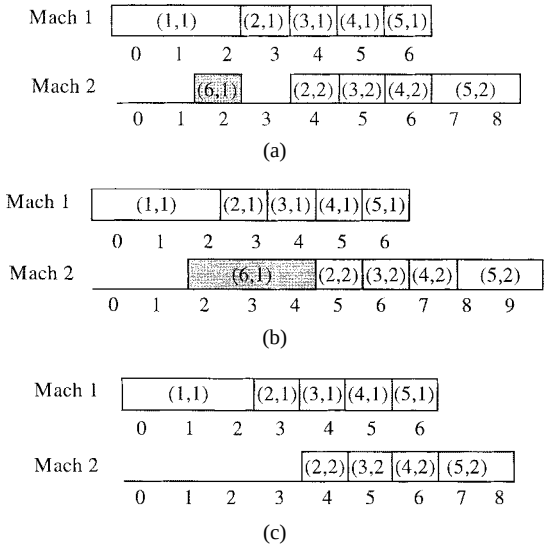


Fig. 3. Gantt charts with uncertain orders, mean method.

TABLE III
DATA FOR EXAMPLE 5.2

Part i	Op. j	Mach. H	t_{ijh}	d_i	w_i	β_i
1	1	1	1	6	1.5	0.1
	2	1	4			
2	1	2	3	11	0.5	0.1
	2	2	1			
	3	1	2			
3	1	1	4	8	1	0.1
	2	2	3			

units. The Gantt charts of the resulting implementable schedule are presented in Fig. 3, and the expected cost is 5.805.

For this example, our method has 32.8% lower expected cost than that of the mean method. In the mean method, the new order is treated as coming with certainty and with processing time 2. In this circumstance, Part 1 should start before Part 2 to avoid the tardiness penalty of Part 1 and the earliness penalty of Part 2. This sequence, however, fails to consider the tardiness penalties of Parts 2–5 when the processing time of Part 6 is 3.

The above example demonstrates a case in which our method outperforms the mean method when on-time delivery is more important than low WIP inventory. The following two variational cases show that our method also outperforms the mean method when low WIP inventory becomes important or more uncertain orders are involved.

Case 2: The earliness weights of all parts are changed to 0.5. The new order (Part 6) may come on time 2 with probability 0.6, or it may not come at all. If it comes, its single operation has an uncertain processing time of either 2 time units with probability 0.2, or 3 with probability 0.8. The rest of the data is the same as above.

For this case, the schedule generated by the mean method has an expected cost 5.4 among which 3.4 is associated with tardiness and 2.0 is associated with earliness. The schedule generated by our method does not yield any earliness penalty, and the expected cost associated with tardiness is 4.56. Overall, the expected cost 5.4 obtained by the mean method is 18.42% higher than the one 4.56 obtained by our method, implying that a better balance between on-time delivery and low WIP inventory is achieved by our method.

Case 3: Suppose that parts 3–5 are also new orders, any of which may arrive on time 4 with probability 0.5 or it may not come at all. The rest of the data is the same as for Case 2.

The schedule in Case 3 is much more complicated than the one in Case 2, with a total of 24 possible realizations as opposed to three realizations in Case 2 for either method. The schedule generated by our method has an expected cost 3.54 which is 52.54% smaller than 5.4 generated by the mean method. The difference between the two methods in Case 3 is larger than 18.42% of Case 2 since more uncertainties are involved.

Example 5.2—Scheduling with Uncertain Arrival Times: In this example, three parts are scheduled on two different machines over a planning horizon of 30 time units. Parts 1 and 2 are available for processing starting from time zero. The arrival time for Part 3 is random: either 1 with probability 0.7 or 2 with probability 0.3. Data are shown in Table III, and the parameter γ in (8) is set to 2.

The problem is first solved by using our method in 0.06 CPU s. The Gantt charts of resulting implementable schedule

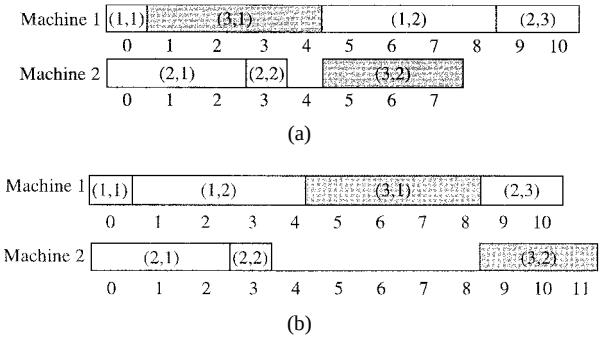


Fig. 4. Gantt charts with uncertain arrival, our method.

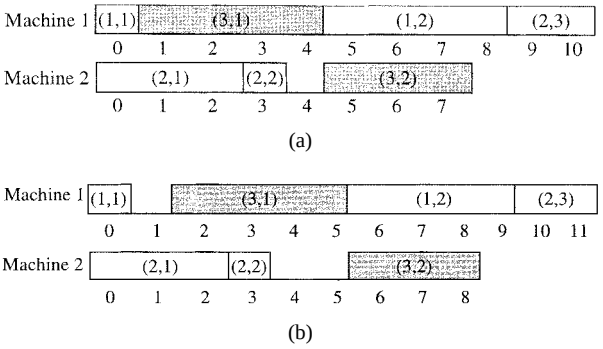


Fig. 5. Gantt charts with uncertain arrival, mean method.

are presented in Fig. 4 with an expected cost 6.9. The lower bound D obtained is 6.897 with a relative duality gap 0.043%. For this small example, it can also be shown that the schedule is optimal by exhaustive search.

The problem is resolved by using the mean method with the same 0.06 CPU s where the arrival time of Part 3 is replaced by its mean 1. The Gantt charts of the resulting implementable schedule are presented in Fig. 5 with an expected cost 8.25.

For this example, our method has an expected cost 19.57% lower than that of the mean method. The reason can be explained as follows. In the mean method when the arrival time of Part 3 is 1, operation (3, 1) starts before (1, 2). When Part 3 does not arrive at time 1, operation (1, 2) waits for its arrival. In our method, operation (3, 1) has two possible beginning times, corresponding to the two possible arrival times. Operations (3, 1) and (1, 2) can thus be switched to respond to different arrival times, leading to a smaller expected cost.

To examine the impact of various probability distributions, the following three cases are considered where Part 3 arrives either at time 1 with probability p or at time 2 with probability $1 - p$.

Case 2: $p = 0.5$.
Case 3: $p = 0.1$.
Case 4: p is randomly generated 100 times based on a uniform distribution over [0, 1].

The schedules generated by our method are identical to those shown in Fig. 4 for both Case 1 or Case 2. The schedule generated by the mean method is identical to the one shown in Fig. 5 for Case 1. For Case 2, the mean method has a different Gantt chart as presented in Fig. 6.

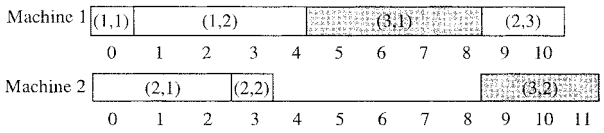


Fig. 6. Gantt chart for Case 2 with uncertain arrival, mean method.

TABLE IV
NUMERICAL RESULTS FOR EXAMPLE 5.2

Case	Our Method Expected Costs	Mean Method Expected Costs	Difference of J
1 (p = 0.7)	6.9	8.25	19.57%
2 (p = 0.5)	7.5	9	20%
3 (p = 0.1)	8.7	9	3.45%
4 (100 runs)	7.51382	8.37892	11.5%

For each case including the original one ($p = 0.7$), the expected costs associated with the two methods and their differences are summarized in Table IV.

Among the first three cases with $p = 0.7$, 0.5, 0.1, Case 3 with $p = 0.1$ has the least uncertainty resulting the smallest difference (3.45%) between the two methods. Our method, however, significantly outperforms the mean method when uncertainties are substantial.

Example 5.3: This example uses data from Delta Industries with contrived uncertainties, and is to show the effects of various levels of uncertainties on scheduling performance. In this example, 87 parts with a total of 461 operations are scheduled on 48 machines belonging to 27 machine types over a planning horizon of 100 time units. Three cases are considered, having 10%, 30%, and 45% parts with uncertainties, respectively. These uncertain parts are randomly selected, and for an uncertain part, each parameter has 50% probability to be uncertain with three possible discrete values. The algorithm is terminated after 5 min. Ordinal optimization then conducts 30 simulation runs for the last 15 dual solutions, and the one yielding the lowest average cost is selected. One thousand Monte Carlo runs are performed to obtain the expected cost for the particular setting. The procedure is repeated five times (five subcases), each having different sets of uncertain parts and uncertain parameters. Testing results are summarized in Table V, including for each subcase whether the confidence regions of the two methods overlap or not with error probability $\alpha = 0.05$. For comparison purposes, the deterministic problem without uncertainties is also tested and presented as Case 0.

It can be observed that near-optimal schedules are generated by our method for this practical problem within a reasonable amount of CPU time. Our method is substantially better than the mean method in subcases 1.4, 2.4, 3.2, and 3.5, and is moderately better than the mean method for subcases 1.1, 2.1, 2.2, and 3.4 having lower expected costs with nonoverlapping confidence regions. For other subcases, since uncertain parts are randomly selected and their parameters randomly generated, insignificant uncertainties may not have major impact. The two methods thus lead to similar results as expected with overlapping confidence regions, and the result of "optimal comparison technique" is also mostly inconclusive.

TABLE V
NUMERICAL RESULTS FOR EXAMPLE 5.3

Case	Subcase	Difference in J	Confidence Regions	Gap of our method
Case 0		0.05%		14.54%
Case 1	1.1	**4.6%**	No overlap	14.27%
	1.2	0.35%	Overlap	15.2%
	1.3	0.73%	Overlap	9.68%
	1.4	**13.17%**	No overlap	12.35%
	1.5	1.12%	Overlap	15.96%
Case 2	2.1	**4.3%**	No overlap	18.79%
	2.2	**5.12%**	No overlap	12.72%
	2.3	0.79%	Overlap	21.63%
	2.4	**15.18%**	No overlap	17.57%
	2.5	0.87%	Overlap	12.13%
Case 3	3.1	1.08%	Overlap	26.74%
	3.2	**11.42%**	No overlap	14.85%
	3.3	0.62%	Overlap	17.56%
	3.4	**6.9%**	No overlap	24.8%
	3.5	**19.2%**	No overlap	20.53%

This is consistent with what was mentioned in the literature (e.g., [19]).

Example 5.4: This example also uses data from Delta Industries with contrived uncertainties, and is used to show the improvement in schedule quality as computation time increases. In this example, 38 parts with a total of 130 operations are scheduled on 35 machines belonging to 14 machine types over a planning horizon of 90 time units, where eight parts have uncertain arrival times and processing times. Four cases are considered in which the algorithm is terminated after 1, 3, 5, and 7 min, respectively. For each case, ordinal optimization conducts 30 simulation runs for last a few dual solutions before the termination of the algorithm and selects the best one. Based on the selected dual solution, 1000 Monte Carlo runs are then conducted to obtain the expected cost. Testing results for each case are shown in Fig. 7, where the confidence regions are obtained with an error probability $\alpha = 0.05$. The dual costs associated with the mean method are not included in the figure since they are generally not lower bounds to the optimal expected costs.

From the testing, it can be seen that our method becomes better than the mean method as computation time is reasonably long, and better dual costs are obtained as the computation time increases. Good expected costs, however, can be obtained within a reasonable CPU time (5 min in this example).

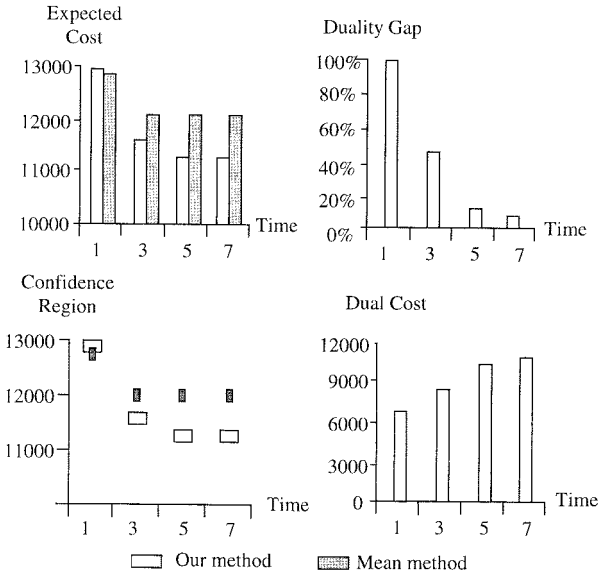


Fig. 7. Testing results and confidence regions for Example 5.4.

VI. CONCLUSION

The job-shop scheduling problem with considering uncertain arrival times, processing times, due dates and part priorities is addressed. A novel methodology that balances modeling accuracy and solution methodology complexity is presented. Through the satisfaction of arrival time constraints and operation precedence constraints for each possible realization of random events, uncertainties are effectively managed. The expected capacity constraints reduce the computational complexity without much loss of modeling accuracy and scheduling performance. The fundamental "weak duality theorem" of LR is proved to hold for the stochastic formulation considered here, providing a quantifiable measure of schedule quality. Testing results supported by simulation demonstrate that our method can be substantially better than the mean method, and near optimal schedules are generated by our method for problems of practical sizes within reasonable CPU times.

Although only uncertain arrival times, processing times, due dates, and part priorities are considered, the formulation and methodology can be extended to handle other kinds of uncertainties, e.g., uncertain number of iterations for a particular operation (representing rework, [16]). The handling of unpredictable machine breakdowns is also an important issue, however, does not fall directly into current framework, and is a subject for future research.

APPENDIX A
List of Symbols

a_i	Arrival time of part i.
b_i	Beginning time of part i.
$\overline{b}_i$	Desired beginning time of part i.
b_{ij}	Beginning time of operation (i, j).
b^*_{ij}	Optimal beginning time of operation (i, j) in subproblem.
b^+_{ij}	Beginning time of operation (i, j) in an optimal implementable schedule.
c_i	Completion time of part i.
c_{ij}	Completion time of operation (i, j).
c^+_{ij}	Completion time of operation (i, j) in an optimal implementable schedule.
d_i	Due date of part i.
D	Dual cost.
E_i	Earliness of part i.
E^+_i	Earliness of part i in an optimal implementable schedule.
$f(i, j)$	Change of cumulative DP cost for b_{ij}.
h	Machine type index.
h_{ij}	Selected machine type index for (i, j).

h_{ij}^*	Optimal machine type index selected for (i, j).
H	Set of machine types.
H_{ij}	Set of machine types eligible for (i, j).
$\|H_{ij}\|$	Cardinality of H_{ij}.
i	Part index $1 \le i \le I$.
I	Total number of parts to be scheduled.
(i, j)	The jth operation of part i.
j	Operation index, $1 \le j \le J_i$;
J	Expected tardiness and earliness cost.
J_i	Number of operations of part i.
J_n	Tardiness and earliness cost for Monte Carlo run n.
$\overline{J}$	Tardiness and earliness cost estimated from Monte Carlo runs.
J^+	Tardiness and earliness cost of an optimal implemenetable schedule.
k	Time index, $0 \le k \le K-1$.
K	Time horizon of scheduling.
L	Lagrangian function.
L_i	Subproblem i.
L_i^*	Optimal cost for subproblem i.
M_{kh}	Number of type h machines available at time k.
S_{ij}	Required "timeout" between operation (i, j) and $(i, j+1)$.
t_{ijh}	Processing time of operation (i, j) on machine type h.
T_i	Tardiness of part i.
T_i^+	Tardiness of an optimal implementable schedule.
T_{ijh}	Set of all possible values of random processing times t_{ijh}.
$\|T_{ijh}\|$	Cardinality of T_{ijh}.
$V_{ij}(b_{ij}, h_{ij})$	Cumulative cost of state j at stage i for b_{ij}, h_{ij}.
w_i	Weight of tardiness penalty for part i.
β_i	Weight of earliness penalty for part i.
δ_{ijkh}	0–1 operation variable identifying if (i, j) is active on machine type h at time k.
δ_{ijkh}^+	0–1 operation variable of an optimal implementable schedule.
π_{kh}	Lagrangian multiplier of machine type h at time k.
Δ_i	0–1 integer variable distinguishing if (i, j) is the first operation of part i.

APPENDIX B
PROOF OF DUAL COST AS LOWER BOUND

Proof of Theorem 4.1: Let $b_{ij}^+, c_{ij}^+, \delta_{ijkh}^+, T_i^+$, and E_i^+ be the beginning time, completion time, 0–1 operation variable, tardiness, and earliness associated with an optimal implementable schedule, respectively. According to (4), we have

$$\sum_{ij} \delta_{ijkh}^+ \le M_{kh}. \tag{24}$$

Thus for any $\pi_{kh} \ge 0$

$$\begin{aligned} J^+ &= E\left[\sum_i \left(w_i T_i^{+2} + \beta_i E_i^{+2}\right)\right] \\ &\ge E\left[\sum_i \left(w_i T_i^{+2} + \beta_i E_i^{+2}\right) \right. \\ &\qquad \left. + \sum_{kh} \pi_{kh}\left(\sum_{ij} \delta_{ijkh}^+ - M_{kh}\right)\right]. \end{aligned} \tag{25}$$

Furthermore, since L_i^* is the minimal cost of subproblem (10)

$$E\left[w_i T_i^{+2} + \beta_i E_i^{+2} + \sum_{j=1}^{J_i} \sum_{k=b_{ij}^+}^{c_{ij}^+} \pi_{kh}\right] \ge L_i^*. \tag{26}$$

Therefore

$$\begin{aligned} J^+ &\ge E\left[\sum_i \left(w_i T_i^{+2} + \beta_i E_i^{+2}\right) \right. \\ &\qquad \left. + \sum_{kh} \pi_{kh}\left(\sum_{ij} \delta_{ijkh}^+ - M_{kh}\right)\right] \\ &= \sum_i E\left[w_i T_i^{+2} + \beta_i E_i^{+2} + \sum_{j=1}^{J_i} \sum_{k=b_{ij}^+}^{c_{ij}^+} \pi_{kh}\right] \\ &\qquad - \sum_{kh} \pi_{kh} M_{kh} \\ &\ge \sum_i L_i^* - \sum_{kh} \pi_{kh} M_{kh} = D. \end{aligned} \tag{27}$$

ACKNOWLEDGMENT

The authors would like to thank J. Berrigan and C. Gonyea, Delta Industries, and F. Liu, University of Connecticut, for their valuable help, X. Zhao, University of Connecticut, for providing the module of conjugate subgradient method, Dr. Y. C. Ho, Harvard University, and Dr. C.-H. Chen, University of Pennsylvania, for ordinal optimization ideas, and the anonymous referees for their insightful and constructive comments.

REFERENCES

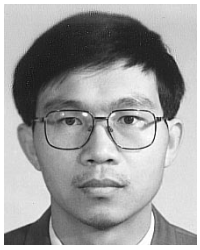
[1] D. Applegate and W. Cook, "A computational study of the job-shop scheduling problem," *ORSA J. Comput.*, vol. 3, no. 2, pp. 149–156, 1991.
[2] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software.* Norwood, MA: Artech, 1993.
[3] D. P. Bertsekas, *Nonlinear Programming.* New York: Athena Scientific, 1995.
[4] C. H. Chen, "A effective approach to smartly allocate computing budget for discrete event simulation," in *Proc. 34th IEEE Conf. Decision Contr.*, 1995, pp. 2598–2605.
[5] H. Chen, C. Chu, and J. M. Proth, "An improvement of the Lagrangean relaxation approach for job shop scheduling: A dynamic programming method," *IEEE Trans. Robot. Automat.*, vol. 14, pp. 786–795, Oct. 1998.
[6] C. Czerwinski and P. B. Luh, "Scheduling products with bills of materials using an improved Lagrangian relaxation technique," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 99–111, Apr. 1994.
[7] L. M. M. Custodio, J. J. S. Sentieiro, and C. F. G. Bispo, "Production planning and scheduling using a fuzzy decision system," *IEEE Trans. Robot. Automat.*, vol. 10, pp. 160–167, Apr. 1994.

[8] P. De, J. B. Ghosh, and C. E. Wells, "On the minimization of the weighted number of tardy job with random processing times and deadline," *Comput. Oper. Res.*, vol. 18, no. 5, pp. 457–463, 1991.
[9] Y. C. Ho, "A new paradigm for stochastic optimization and parallel simulation," in *Discrete Event Systems, Manufacturing Systems, and Communication Networks.* New York: Springer-Verlag, 1995, pp. 59–72.
[10] R. M. Kerr and R. V. Ebsary, "Implementation of an expert system for production scheduling," *Euro. J. Oper. Res.*, vol. 33, no. 1, pp. 17–29, 1988.
[11] P. R. Kumar and S. P. Meyn, "Stability of queuing networks and scheduling policies," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 251–260, Feb. 1995.
[12] S. Kumar and P. R. Kumar, "Performance bounds for queuing networks and scheduling policies," *IEEE Trans. Automat. Contr.*, vol. 39, pp. 1600–1611, Aug. 1994.
[13] V. J. Leon, S. D. Wu, and R. H. Storer, "A game-theoretic control approach for job shops in the presence of disruptions," *Int. J. Prod. Res.*, vol. 32, no. 6, pp. 1451–1476, 1994.
[14] P. B. Luh, D. Chen, and L. S. Thakur, "Modeling uncertainty in job shop scheduling," in *Proc. 1st Int. Conf. Oper. Quantitative Manag.*, 1997, pp. 490–497.
[15] P. B. Luh and D. J. Hoitomt, "Scheduling of manufacturing systems using the Lagrangian relaxation technique," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 1066–1079, Jan. 1993.
[16] P. B. Luh, F. Liu, and B. Moser, "Scheduling of design projects with uncertain number of iterations," in *Eur. J. Oper. Res.*, 1998.
[17] J. M. Mulvey and A. Ruszczynsky, "A new scenario decomposition method for large-scale stochastic optimization," *Oper. Res.*, vol. 43, no. 3, pp. 477–490, 1995.
[18] M. L. Pinedo, *Scheduling—Theory, Algorithms and Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1995.
[19] R. T. Rockafellar and R. J. B. Wets, "A scenarios and policy aggregation in optimization under uncertainty," *Math. Oper. Res.*, vol. 16, no. 1, pp. 119–147, 1991.
[20] H. M. Soroush, "Optimal sequence in stochastic single machine shops," *Comput. Oper. Res.*, vol. 23, no. 7, pp. 705–721, 1996.
[21] R. N. Tomastik and P. B. Luh, "The facet ascending algorithm for integer programming problems," in *Proc. IEEE Conf. Decision Contr.*, 1993, pp. 2880–2884.
[22] J. Wang, P. B. Luh, X. Zhao, and J. Wang, "An optimization-based algorithm for job shop scheduling," *Sadhana*, vol. 22, pt. 2, pp. 241–256, 1996.
[23] X. Zhao, P. B. Luh, and J. Wang, "The surrogate gradient algorithm for Lagrangian relaxation," *J. Optim. Theory Appl.*, 1999.

Peter B. Luh (S'77–M'80–SM'91–F'95) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., 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. He is the Director of the Booth Center for Computer Applications and Research and is a Professor in the Department of Electrical and Systems Engineering. He is interested in planning, scheduling, and coordination of design and manufacturing activities, and has developed a near-optimal and efficient schedule generation and reconfiguration methodology to improve on-time delivery of products and reduce work-in-progress inventory. He is also interested in schedule and transaction optimization for power systems, and has developed near-optimal and efficient unit commitment, hydro-thermal coordination, and power transaction methodologies to minimize total fuel costs. He owns one U.S. patent.

Dr. Luh received three Best Paper Awards. He is a member of the Connecticut Academy of Science and Engineering. He has been on the Editorial Board of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION since 1990 (Technical Editor, Associate Editor, and now Editor since 1995), is an Editor of IEEE ROBOTICS AND AUTOMATION MAGAZINE since 1996, and was an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 1989 to 1991.



Dong Chen (S'98) received the B.S. degree in mathematics and the M.S. degree in computer and systems sciences from Nankai University, Tianjin, China, in 1984 and 1987, respectively, and is currently pursuing the Ph.D. degree in the Department of Electrical and System Engineering, University of Connecticut, Storrs.

From 1987 to 1990, he was an Assistant Professor in the Department of Computer and Systems Sciences, Nankai University, Tianjin. From 1990 to 1995, he was an Assistant Professor and Associate Professor in the Department of Finance, Dongbei University of Finance and Economics, Dalian, China. His major interests include manufacturing planning and scheduling, control theory and applications, and optimization.

Mr. Chen received the Science and Technology Advancement Award honored by the State Education Commission, China, and the Best Student Paper Award in the First International Conference and Quantitative Management. He is a student member of INFORMS.

Lakshman S. Thakur received the B.Sc. degree in Mathematics and Physics, from Bombay University, Bombay, India, in 1963, and the Ph.D. degree in operations research and industrial engineering from Columbia University, New York, NY, in 1971.

He is an Associate Professor of Operations and Information Management at the University of Connecticut, Storrs. He has more than 20 years of experience in teaching, consulting, and research on optimization under resource constraints with numerous refereed publications. He has been a Consultant to IBM on manpower planning with risk assessment and product warranty systems, as well as a Senior Consultant and Director of Management Science in a consulting organization. He was a Visiting Professor of Operations Research at Yale School of Management, Yale University, New Haven, CT, from 1985 to 1987. He was published in *Management Science*, *Mathematics of Operations Research*, *SIAM Journal on Applied Mathematics*, *SIAM Journal on Optimization and Control*, *Journal of Mathematical Analysis and Applications*, *Naval Research Logistics*, and other journals. His primary research interests are in the development and applications of linear, nonlinear, and integer optimization methods in management science and spline function approximation in mathematics. His current research focuses on production scheduling, product design, and facility location problems. His research on scheduling is supported by National Science Foundation Grants.

Dr. Thakur is an Associate Editor of the *Naval Research Logistics Journal* and a member of INFORMS (Operations Research Society of America and The Institute of Management Sciences) and the Mathematical Programming Society.