Approximation to Multistage Stochastic Optimization in Multiperiod Batch Plant Scheduling under Demand Uncertainty

J. Balasubramanian and I. E. Grossmann*

Department of Chemical Engineering, Carnegie Mellon University

Pittsburgh, PA 15213, USA

April 2003
Revised September 2003

Abstract

Abstract We consider the problem of scheduling under demand uncertainty a multiproduct batch plant represented through the State Task Network. Given a scheduling horizon consisting of several time-periods in which product demands are placed, the objective is to select a schedule that maximizes the expected profit. We present a multistage stochastic Mixed Integer Linear Programming (MILP) model, wherein certain decisions are made irrespective of the realization of the uncertain parameters, and some decisions are taken upon realization of the uncertainty. To overcome the computational expense associated with the solution of the large-scale stochastic multistage MILP for large problems, we examine an approximation strategy based on the solution of a series of a two-stage models within a shrinking horizon approach. Computational results indicate that the proposed approximation strategy provides an expected profit within a few percent of the multistage stochastic MILP in a fraction of the computation time, and provides significant improvement in the expected profit over similar deterministic approaches.

1 Introduction

Batch processes are the preferred mode of production for a number of specialty chemicals and pharmaceutical products due to the increased flexibility that they confer on the processing

*Author to whom all correspondence should be addressed plants. With steady growth in these industries, the design and scheduling of multiproduct batch plants have been an active area of research over the past decade (see Shah[1] and Pinto and Grossmann[2] for reviews). Most of the work in this area has been limited to deterministic approaches, wherein the problem parameters are assumed to be known with certainty. However, in reality there can be uncertainty in a number of factors such as processing times, costs, demands, etc. To cope with this, there has been increased interest in the development of different types of probabilistic models that explicitly take in to account the various uncertainties (Sahinidis[3] provides a review of optimization under uncertainty).

Research in the area of batch plants with demand uncertainty has focussed more on the design aspect than on operations scheduling. For example, Wellons and Reklaitis[4] proposed an MINLP model for the design of batch plants under uncertainty with staged capacity expansions. Based on the structure of multiproduct batch plants, Straub and Grossmann[5] developed an efficient procedure to evaluate the expected stochastic flexibility (a measure of the probability of meeting demands). This evaluation procedure was then incorporated within an optimization framework for selecting the design (size and number of parallel equipment) that maximizes the expected stochastic flexibility subject to a capital constraint. Two-stage stochastic programming approaches have also been applied for design under uncertainty (Ierapetritou and Pistikopolous[6]; Harding and Floudas[7]; Petkov and Maranas[8]; Cao and Yuan[9]). The scheduling of batch plants under demand uncertainty has emerged only recently as an area of active research. Work in this area has addressed the problems of both evaluating and optimizing schedules under uncertainty. Sand *et al.*[10] proposed an algorithm to approximate the performance of an ideal online scheduler for a multiproduct batch plant under demand uncertainty. These authors used a two-level hierarchical framework which consisted of a stochastic linear program for the purpose of long-term planning and a deterministic nonlinear model for short-term scheduling. Gupta *et al.*[11] used a chance-constrained approach in conjunction with a two-stage stochastic programming model to analyze the trade-offs between demand satisfaction and production costs for a mid-term supply chain planning problem. Vin and Ierapetritou[12] addressed the problem of quantifying schedule robustness under demand uncertainty. These authors used multiperiod programming to obtain schedules that were feasible over an entire range of parameters, and also proposed robustness metrics based on deviations from the deterministic schedule. They observed with flexibility analysis that the schedules from the multiperiod programming approach were more robust than the deterministic schedules. More recently, Engell *et al.*[13] reported an application of a scenario-decomposition algorithm, which along with some problem-dependent preprocessing procedures, provided very good computational results for the solution of a two-stage stochastic scheduling model for a polymer plant under demand uncertainty.

In the operations research literature, the stochastic lot scheduling problem (SLSP) is perhaps most similar to our problem. The SLSP deals with the scheduling of multiple products, each with random demand, on a single facility with limited production capacity and significant change-over costs. However, the critical difference between the SLSP and our current problem is that the SLSP does not incorporate the explicit material balances of chemical plants. Sox *et al.*[14] provide a comprehensive review of the various approaches used for solving the SLSP. These approaches can range from simple control rules derived from deterministic analyses to control policies derived from nonlinear optimization, or simulation or queueing analysis of the stochastic models.

Other approaches to handling uncertainty include the reactive modification of schedules upon realization of uncertainty (Cott and Macchietto[15]; Kanakamedala *et al.*[16]; Jain and Elmaraghy[17]; Vin and Ierapetritou[18]; Mendez and Cerda[19]). Of course, with the reactive modification strategy, there is the possibility that too many changes are made to the schedule. This has typically been handled through the use of penalty terms in the objective, which penalize significant changes from the current schedule.

It is the aim of this work to present a multistage stochastic programming approach to the scheduling of multiproduct batch plants represented as State Task Networks (STNs) (Kondili *et al.*[20]) under demand uncertainty. In this aproach certain decisions are made irrespective of the realization of the uncertain parameters, and some decisions are taken upon realization of the uncertainty. The multistage stochastic program is a large-scale Mixed Integer Linear Programming (MILP) model derived from the corresponding single time-period deterministic scheduling model. To overcome some of the difficulties associated with the solution of the large-scale MILP we examine an approximation strategy based on the solution of a series of a two-stage models within a shrinking horizon approach. We also compare these approaches with similar deterministic approaches with respect to solution quality and computational performance.

The paper is organized as follows. Section 2 presents a formal definition of the problem of interest. Section 3 illustrates the value of using multistage stochastic programming approaches over conventional methods. Section 4 discusses the multistage stochastic programming formulation, while Section 5 discusses the approximation strategies possible for the problem of interest. After presenting results from a number of examples in Section 6, we draw the conclusions on this work.

2 Problem Statement

We consider the problem of optimizing the short-term scheduling of a multiproduct batch plant represented as an STN, while taking into account uncertainties in product demands.

The description of the STN consists of, (i) the materials or states $s \in S$ (raw materials, intermediates and products) (ii) the recipes for the production of different products $s \in SP \subset S$ (i.e., the various tasks $i \in I$, the processing times and the amount of materials required for the production of each product), (iii) the available processing units $j \in J$ and their suitabilities for the tasks (I_j indicating the tasks that can be performed by unit j and J_i indicating the units that can be used to perform task i), (iv) storage capacities (CAP_s) for the various materials. The processing units can be operated in any of $l \in L$ different modes, each of which is characterized by a range of batch sizes $[B_{ijl}^{LO}, B_{ijl}^{UP}]$ and a task-dependent processing time τ_{ijl}. These L processing modes provide increased production flexibility to the plant. In this work we assume that there are no change-over costs or times between the processing modes, although it is possible to extend the models to the case with non-zero change-over costs.

In this work the scheduling model is formulated as a discrete-time STN where each time-period is divided into several time intervals, as shown in Figure 1. The scheduling horizon is of length H units and consists of time-periods $m \in M$. In this work we define a time-period as the shortest duration for which the demand uncertainty can be reliably described using probability distributions. As shown in Figure 1, a time-period m corresponds to the time interval $T_m = [H_{m-1}, H_m)$, except for the final time-period for which $T_{|M|} = [H_{|M|-1}, H_{|M|}]$ (i.e., the final time-period includes the time point at the end of the horizon). With this notation $H_0 = 0$ and $H_{|M|} = H$. We define the time-periods this way to reflect the fact that the uncertainty in the demand in the time-period m completely resolves itself by $t = H_m$, which implies that the decisions to be made at $t = H_m$ are recourse.

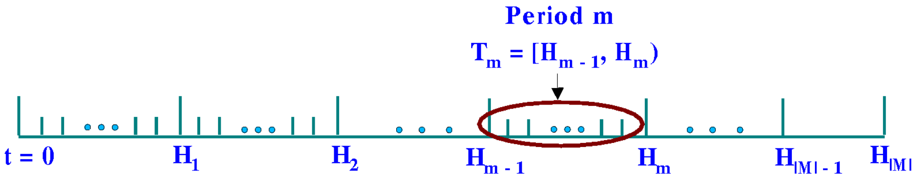

Figure 1: Scheduling Horizon

The demands for products are placed in the time-periods through the horizon and are required to be satisfied at the end of the horizon at $t = H$. The uncertainty in the product demands in each time-period is described by discrete probability distributions, i.e. any one of $k_m \in K_m$ possible events can occur during a time-period, each with an associated probability p_{m,k_m}. Thus D_{s,m,k_m} indicates the demand for product $s \in SP$ placed in time-period m if event k_m occurs in time-period m. We assume that $|K_m| = K \;\; \forall m \in M$.

The costs involved are those for, (i) holding inventory over the horizon (parameters HC_s denote the cost of holding one unit of state s for one time unit), (ii) excess inventory at the end of the horizon (parameters XC_s denote the cost of holding one unit of state s at the end of the horizon, i.e. a terminal cost), (iii) lost demand due to inadequate production (parameters LC_s denote the cost of losing one unit of demand for state s), while the revenues are from selling the products (parameters REV_s denote the revenues from selling one unit of state s).

The problem then is to obtain a schedule that maximizes the expected profit, where the scheduling decisions consist of the allocation of the equipment to the appropriate mode of production of different products over the time horizon, as well as choosing the batch size in the selected production mode. With the inventory costs, we would like to store as little as possible through the horizon and produce the final products as late as possible towards the end of the horizon, which is when the demands must be met. With the costs for excess- and under-production, we would like to match demand as closely as possible. Unlike previous work, we assume that some of the demand can actually be unsatisfied due to limited production capacity. Thus, the problem also consists in choosing the optimal amounts to be produced for each product in case there are conflicts between products for resources.

3 Motivating Example

Here we present a small example to illustrate the pitfalls of using deterministic approaches to scheduling optimization in the face of demand uncertainty. We also show the value of using a staged decision-making approach to solving this problem.

Consider a single-stage single-unit batch plant that can process two products, A and B. The processing unit can be operated in $|L| = 3$ processing modes which differ in the batch sizes. The processing time for a given mode is product dependent. For example, it takes 2 time units to produce 0-5 tons of product A, while it takes 3 time units to produce 0-5 tons of B. The processing times and batch-size ranges for the different modes for the two products, as well as the costs and revenues are given in Table 1. Product B is more profitable than A and takes longer to process for a given batch size. The excess costs denote the penalty for over-production, while the lost costs account for under-production. With these costs, we would like to match demand as closely as possible. Furthermore, for the sake of simplicity of presentation, we assume in this example that there are no holding (inventory) costs.

The horizon of interest in this problem is a period of 20 units (from $t = 0$ to $t = 20$), and is made up of $|M| = 2$ equally long time-periods (the first time-period from [0, 10) and the second from [10, 20]) for which the product demand distribution is available. For the sake of simplicity, we assume that the demand distributions for both time-periods are the same

Table 1: Data for Motivating Example

Product	Batch Size Range (tons)	Proc. Time (units)	REV_s (\$/tons)	XC_s (\$/tons)	LC_s (\$/tons)
A	0-5	2	100	10	20
	5-10	4			
	10-25	6			
B	0-5	3	250	20	50
	5-10	5			
	10-25	7			

(i.e., time independent distributions), although this assumption can be easily relaxed. The demand distribution for each time-period is given in Table 2. There are $|K| = 2$ possible events in each of the two time-periods, resulting in a total of 2^2 scenarios for the entire horizon. The mean time-period demands for A and B are 17.5 and 3.75 tons respectively.

Table 2: Time-period Demand Distribution for Motivating Example

Event	Probability	Demands (tons)
1	0.25	A: 10, B: 0
2	0.75	A: 20, B: 5

Given this data, the objective is to find a schedule that maximizes profit, defined as the difference of the revenues and the excess and lost costs. The scheduling decisions include determining the optimal amounts to be produced for each product, the modes of production (batch sizes) for that amount, as well as the timing of the production.

3.1 Deterministic Approach

The expected demands for products A and B for the entire horizon are 35 tons and 7.5 tons respectively. The optimal schedule obtained from solving an MILP model for these demands is as shown in Figure 2. Although the deterministic MILP model predicts a profit of \$5375 for this schedule and production that meets the expected demand, the actual expected profit is only \$4559 when evaluated under uncertainty given in Table 2.

3.2 Two-stage Approach

The two-stage approach treats all scheduling decisions (such as which processing mode is selected, the sequencing and sizing of various batches) as first-stage decisions that are com-

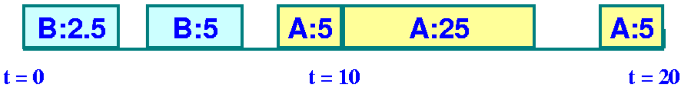

Figure 2: Optimal Deterministic Schedule

mon to all scenarios. However, decisions such as the amount to be sold and the amount of demand unsatisfied are regarded as second-stage decisions, which are taken at the end of the horizon ($t = 20$), after realization of the total demand, and are scenario-dependent. The amount to be sold in a particular scenario is calculated as the minimum of the amount produced and demand realized in that scenario. The optimization model for determining a schedule that maximizes expected profit over all scenarios can be formulated as an MILP as will be seen later in the paper. The optimal two-stage solution is as shown in Figure 3 and has an expected profit of \$5275. Note that in this solution we produce 40 tons of A and 10 tons of B since it is more profitable to produce in excess of the mean demands despite the penalties for over-production.

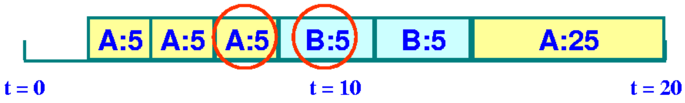

Figure 3: Optimal Schedule from Two-Stage Model

3.3 Three-stage Approach

Since the uncertainty in the demand resolves itself in 2 stages (at $t = 10$ when the first time-period's uncertainty is resolved, and at $t = 20$), it is natural to treat this problem as a three-stage stochastic MILP with the following structure:

(i) the scheduling decisions (timing and sizing of batches) for the first time-period [0, 10) are treated as first-stage decisions;

(ii) the scheduling decisions for the second time-period [10, 20) are second-stage decisions which are dependent on the events that occur in the first time-period, and

(iii) the continuous variables related to calculation of amounts sold, in excess, lost and revenues and costs are considered as third-stage decisions made at $t = 20$.

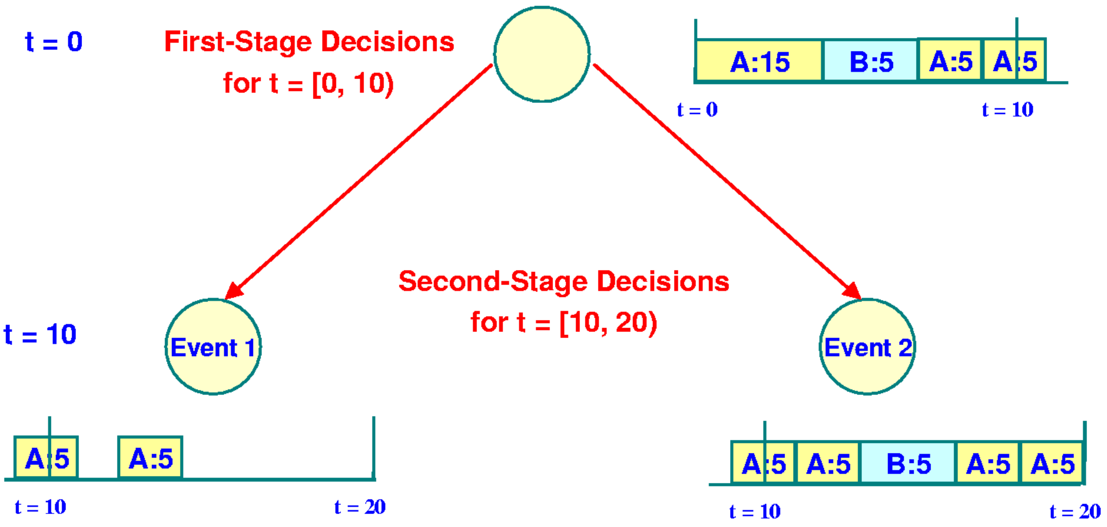

Figure 4: Optimal Schedules from Three-Stage Model

With the three-stage MILP model, the optimal schedule is embedded in a tree as shown in Figure 4. At the root node of the tree we have the tasks to be implemented in stage one. From Figure 4, we can see that these tasks are processing 15 tons of A at $t = 0$, 5 tons of B at $t = 4$ and two batches of 5 tons of A at $t = 7$ and $t = 9$. Note that we do not restrict the first-stage decisions to finish by the end of the first time-period (for e.g., a batch of 5 tons of A begins processing at $t = 9$ and continues till $t = 11$, beyond the end of the first time-period). At $t = 10$, when the uncertainty in demand in the first time-period has been completely resolved, there are two possible courses of action. If Event 1 has occured in time-period 1 (a demand of 10 tons of A and 0 tons of B), then we carry out the schedule in the left half of the tree in Figure 4. If Event 2 has occured, then we follow the schedule in the opposite part of the tree. We have not shown the third-stage decisions (amounts sold/lost/excess etc.) at the end of the horizon. This solution, which produces 30 tons of A and 5 of B if Event 1 occurs in the first time-period and 40 tons of A and 10 of B if Event 2 occurs, has an expected profit of $5325.

3.4 Comparison of Solutions

Table 3 presents a comparison of the performance of the different strategies in various scenarios. Here, Scenario 1 refers to a combination of Event 1 in the first time-period and Event 1 in the second time-period; Scenario 2, a combination of Event 1 in the first time-period and Event 2 in the second time-period and so on. The three-stage strategy not only

Table 3: Profits in Different Scenarios

Strategy	Profit in Scenario ($)				Expected Profit ($)
	1	2	3	4	
Deterministic	1700	4150	4150	5150	4559
Two-Stage	1600	4050	4050	6500	5275
Three-stage	1800	4250	4050	6500	5325

has the best overall expected profit, as expected ~~from the theory,~~ but also has significantly higher profit in the worst (Scenario 1) and best (Scenario 4) cases. Although the two-stage approach performs better than the deterministic approach, its performance suffers in comparison with the three-stage approach because the scheduling decisions are common to all scenarios. The three-stage aproach is also similar to reactive scheduling, wherein there is some rescheduling of activities as a response to the realization of an event.

Thus, there are significant incentives to anticipating demand uncertainty at the level of scheduling, by solving the corresponding multistage stochastic program.

4 Multistage Stochastic Programming Approach

In this section we discuss the various modeling approaches that are relevant to our problem. We begin with an overview of the concept of scenario trees, and then illustrate how the different models that we develop can be linked to these scenario trees.

4.1 Scenario Trees

Recall that there are $|M|$ time-periods over the horizon. The product demands in each of these time-periods are random variables that can take on $|K|$ possible values, each with an associated probability p_{m,k_m}. With this information structure, the external world (i.e., the demand) can be in any one of $|K|$ possible states after one time-period, in any one of $|K|^2$ possible states after two time-periods and so on. We can represent this information structure as a scenario tree, where the nodes at level m represent the states of the external world up to the time-period m (with the root node at level 0 corresponding to the state of the world at time $t = 0$). At level $|M|$ of the tree, we have $|K|^{|M|}$ nodes representing different scenarios. These scenarios correspond to a joint realization of the demands over the $|M|$ time-periods.

Figure 5 displays a scenario tree for a case when there are $|K| = 2$ possible realizations of the demands per time-period, and $|M| = 3$ time-periods, resulting in $2^3 = 8$ scenarios. For example, the random variable d_1 which represents the demand in the first time-period can take one of two values D_{11} or D_{12}, each with an associated probability, d_2 represents the demand in the second time-period etc.

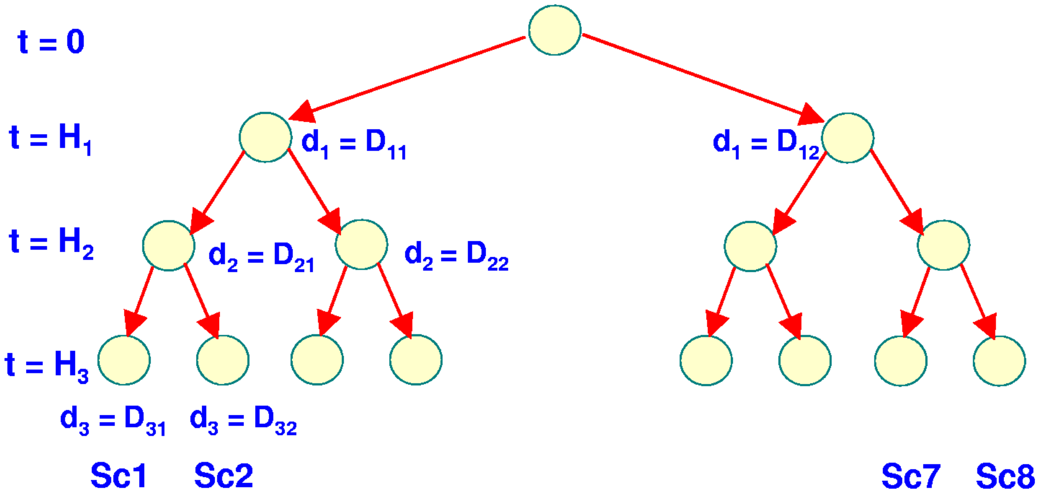

Figure 5: Scenario Tree

4.2 MILP Model

The multistage stochastic program captures the information structure associated with the scenario tree. Given that there are $|M|$ time-periods in the scheduling horizon, wherein the random variables are realized, we can formulate the scheduling problem as an $(|M| + 1)$-stage stochastic MILP. The solution of this MILP provides a set of scheduling decisions to implement at each node in the scenario tree (see Figure 6). Of course, at the terminal nodes (leaves) of the tree, the $(M + 1)^{th}$-stage decisions correspond only to the calculation of the amounts sold, lost, in excess, as well as the costs and revenues.

The $(|M| + 1)$-stage stochastic MILP model is given by (MSP), and represents an extension of the model of Kondili *et al.*[20] to the stochastic case. We refer the reader to the appendix for the nomenclature, but for completeness we provide a brief description of the notation used in model (MSP).

Indices k_m denote possible events in time-period m. The superscript m above a variable indicates that the latter is a stage-m variable or a variable in time-period m (as we shall see later while developing a two-stage model).

The index k_m^* represents the combination of events given by $(k_1, \ldots, k_{m-1})$, with $k_1 \in K_1, \ldots, k_{m-1} \in K_{m-1}$. For instance, k_2^* represents (k_1), k_3^* represents (k_1, k_2) and so on. Hence, a variable such as $x^m_{k^*_m}$ is a stage-m variable equivalent to $x^m_{k_1,\ldots,k_{m-1}}$, indicating that for every combination of events in the previous $m - 1$ stages, we define a possible recourse variable.

With this notation, k_1^* is assumed to be the empty set, $\{\}$: thus variables x^1 are stage-1 decision variables that are to be taken before a realization of any random variable.

We define K_m^* to be the set of $m - 1$ tuples representing all possible combinations

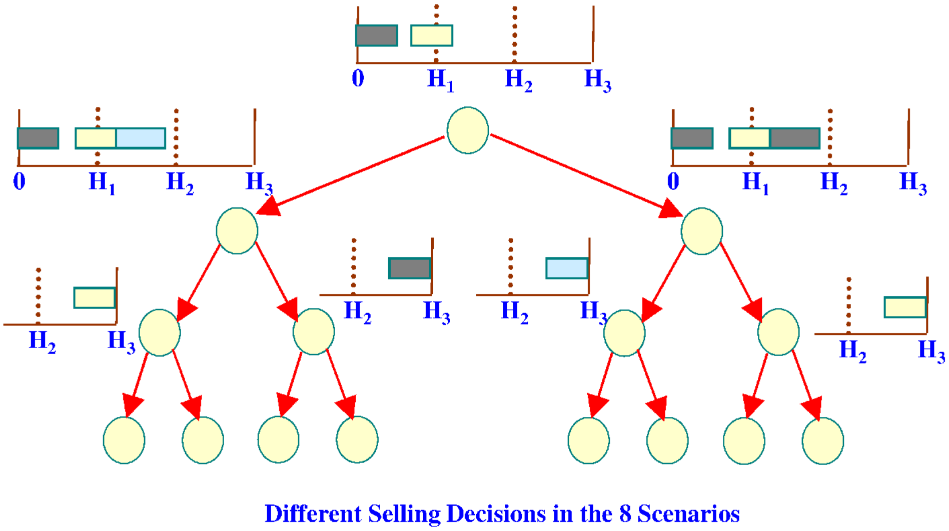

Figure 6: Multistage Solution

of $k_1, \ldots, k_{m-1} \in K$, i.e. $K^*_m = \{(k_1, \ldots, k_{m-1}) | k_1 \in K_1, \ldots, k_{m-1} \in K_{m-1}\}$. For instance, $K^*_3 = \{(k_1, k_2) | k_1 \in K_1, k_2 \in K_2\}$. With this notation, we can observe that the set K^*_m corresponds to the nodes at level $m-1$ of the scenario tree. Thus, the set $K^*_3 = \{(1,1),(1,2),(2,1),(2,2)\}$ corresponds to the four nodes at level 2 of the scenario tree shown in Figure 6. Also, $p_{k^*_{|M|+1}}$ indicates the probabilities of the $|K|^{|M|}$ scenarios that belong to set $K^*_{|M|+1}$.

In (MSP), the variables $y^m_{i,j,t,k^*_m,l}$ are stage-m binary variables that denote that task i starts in processing unit j at time t in processing mode l. The continuous variables $b^m_{i,j,t,k^*_m,l}$ and st^m_{s,t,k^*_m} are stage-m variables denoting the batch size of task i starting at time t in unit j in processing mode l and the amount of state s in storage at time t respectively. Continuous variables $sold$, xs, $lost$ are stage- $|M|+1$ variables that correspond to the amount of products sold, amount produced in excess and the amount lost due to insufficient production respectively. The profit in each scenario is calculated as the difference of the revenues ($sales$) and the costs of excess inventory ($txic$), holding inventory through the horizon (thc) and lost demands (tlc), which are all stage-$|M+1|$ variables.

The constraints in the model can be classified into two types: those related to the scheduling aspects (constraints (2) to (8)), and those related to the calculations of the revenues and costs (constraints (9) to (16)). The scheduling constraints are defined for every stage $m \in M$, and for every time point in interval T_m, as well as every k^*_m, while the revenue and cost calculations being stage-$|M|+1$ computations are defined only at the end of the horizon.

1. **Objective:** The objective in (1) is to maximize the expected profit defined as the probability weighted sum of the profit in each scenario.

$$\max EP = \sum_{k^*_{|M|+1} \in K^*_{|M|+1}} p_{k^*_{|M|+1}} \cdot (sales^{|M|+1}_{k^*_{|M|+1}} - txic^{|M|+1}_{k^*_{|M|+1}} - tlc^{|M|+1}_{k^*_{|M|+1}} - thc^{|M|+1}_{k^*_{|M|+1}}) \tag{1}$$

2. **Scheduling Constraints:**

In (2), we state that for every time point t, in a given unit j capable of processing tasks i, at most one task can begin processing in at most one mode.

$$\sum_{i \in I_j} \sum_{l \in L} y^m_{i,j,t,k^*_m,l} \leq 1 \qquad \forall (j \in J, m \in M, t \in T_m, k^*_m \in K^*_m) \tag{2}$$

In (3), we state that if a task i has been assigned to unit j in mode l at time t, i.e. $y_{ijtl} = 1$, then this unit cannot be assigned to tasks other than i during the interval [t-τ_{ijl}+1,t]. These are similar to the constraints in Shah *et al.*[21].

$$\sum_{i \in I_j} \sum_{l \in L} \sum_{t'=t-\tau_{ijl}+1}^{t} \sum_{m'=1}^{m} y^{m'}_{i,j,t',k^*_m,l} \leq 1 \qquad \forall (j \in J, m \in M, t \in T_m, k^*_m \in K^*_m) \tag{3}$$

Equations (4) and (5) bound the amount of material that has begun processing at any given time within the respective limits on the processing mode's size range.

$$B^L_{i,j} \cdot y^m_{i,j,t,k^*_m,l} \leq b^m_{i,j,t,k^*_m,l} \qquad \forall (i \in I, j \in J_i, m \in M, t \in T_m, k^*_m \in K^*_m) \tag{4}$$

$$b^m_{i,j,t,k^*_m,l} \leq B^U_{i,j} \cdot y^m_{i,j,t,k^*_m,l} \qquad \forall (i \in I, j \in J_i, m \in M, t \in T_m, k^*_m \in K^*_m) \tag{5}$$

The material balances for the different states are captured through equations (6) and (7), with (6) stating that the amount of state s in storage at $t = 1$ and the amount that has begun processing at $t = 1$ cannot be more than the initial amount available, $START_s$. In equation (7), the amount of state s at time t is equal to that at time $t-1$ adjusted by the amounts produced or consumed. Equation (8) imposes continuity of the amounts of states across the stages (time-periods).

$$st^1_{s,1} + \sum_{i \in I_s} \rho^{IN}_{is} \cdot \sum_{j \in J_i} \sum_{l \in L} b^1_{i,j,1,l} \leq START_s \qquad \forall s \in S \tag{6}$$

$$\begin{aligned} st^m_{s,t,k^*_m} + \sum_{i \in I_s} \rho^{IN}_{is} \cdot \sum_{j \in J_i} \sum_{l \in L} b^m_{i,j,t,k^*_m,l} \quad = \quad & st^m_{s,t-1,k^*_m} \\ & + \sum_{i \in I_s} \rho^{OUT}_{is} \cdot \sum_{j \in J_i} \sum_{l \in L} \sum_{m'=1}^{m} b^{m'}_{i,j,t-\tau_{ijl},k^*_{m'},l} \\ & \forall (s \in S, m \in M, t \in T_m, k^*_m \in K^*_m) \end{aligned} \tag{7}$$

$$st^{m-1}_{s,H_m-1,k^*_{m-1}} = st^m_{s,H_m-1,k^*_m} \qquad \forall (s \in S, m \in M, k^*_m \in K^*_m) \tag{8}$$

3. **Sales Calculations:**

   Constraints (9) and (10) specify that the amount of product s sold is the minimum of the amount available at the end of the horizon and the total demand for the product over the entire horizon (i.e., $D^{|M|+1}_{s,k^*_{|M|+1}} = \sum_{m \in M} D_{s,m,k_m}$). Constraints (11) and (12) calculate the amount of product that is in excess of demand, and less than the demand, respectively, while (13)-(15) compute the revenue and various costs. In (16) we compute the inventory cost using a rectangular approximation of the total area under the inventory-time curve for each state. In (14), we compute the cost of excess production of product as well as that of producing excess of intermediates. In (17), we specify that no task can be started if it cannot be finished before the end of the scheduling horizon.

$$
\begin{aligned}
sold^{|M|+1}_{s,k^*_{|M|+1}} &\le st^{|M|}_{s,H,k^*_{|M|}} && \forall (s \in SP, k^*_{|M|+1} \in K^*_{|M|+1}) && (9)\\
sold^{|M|+1}_{s,k^*_{|M|+1}} &\le D^{|M|+1}_{s,k^*_{|M|+1}} && \forall (s \in SP, k^*_{|M|+1} \in K^*_{|M|+1}) && (10)\\
xs^{|M|+1}_{s,k^*_{|M|+1}} &\ge st^{|M|}_{s,H,k^*_{|M|}} - D^{|M|+1}_{s,k^*_{|M|+1}} && \forall (s \in SP, k^*_{|M|+1} \in K^*_{|M|+1}) && (11)\\
lost^{|M|+1}_{s,k^*_{|M|+1}} &\ge D^{|M|+1}_{s,k^*_{|M|+1}} - st^{|M|}_{s,H,k^*_{|M|}} && \forall (s \in SP, k^*_{|M|+1} \in K^*_{|M|+1}) && (12)\\
sales^{|M|+1}_{k^*_{|M|+1}} &= \sum_{s \in SP} REV_s \cdot sold^{|M|+1}_{s,k^*_{|M|}} && \forall (k^*_{|M|+1} \in K^*_{|M|+1}) && (13)\\
txic^{|M|+1}_{k^*_{|M|+1}} &= \sum_{s \in SP} XC_s \cdot xs^{|M|+1}_{s,k^*_{|M|}} && && \\
&+ \sum_{s \in S \setminus SP} XC_s \cdot st^{|M|}_{s,H,k^*_{|M|}} && \forall (k^*_{|M|+1} \in K^*_{|M|+1}) && (14)\\
tlc^{|M|+1}_{k^*_{|M|+1}} &= \sum_{s \in SP} LC_s \cdot lost^{|M|+1}_{s,k^*_{|M|+1}} && \forall (k^*_{|M|+1} \in K^*_{|M|+1}) && (15)\\
thc^{|M|+1}_{k^*_{|M|+1}} &= \sum_{m \in M} \sum_{s \in S} \sum_{t \in T_m} HC_s \cdot st^m_{s,t,k^*_m} && \forall (k^*_{|M|+1} \in K^*_{|M|+1}) && (16)\\
y^m_{i,j,t',k^*_m,l} &= 0 && \forall (m \in M, i \in I, j \in J_i, k^*_m \in K^*_m, l \in L, && \\
& && t' | t' + \tau_{ijl} > H) && (17)
\end{aligned}
$$

With the integrality specifications on y^m and the bounds on the other variables, the model (MSP) is thus formulated as:

$$
\begin{aligned}
\text{(MSP)} \qquad\qquad & \max \quad EP && \\
\text{s.t.} \quad & (2) - (17) && \\
y^m_{i,j,t,k^*_m,l} &\in \{0,1\} && \forall (m \in M, i \in I, j \in J_i, t \in T_m, k^*_m \in K^*_m, l \in L)\\
b^m_{i,j,t,k^*_m,l} &\ge 0 && \forall (m \in M, i \in I, j \in J_i, t \in T_m, k^*_m \in K^*_m, l \in L)\\
0 \le st^m_{s,t,k^*_m} &\le CAP_s && \forall (m \in M, s \in S, t \in T_m, k^*_m \in K^*_m)
\end{aligned}
$$

$$
\begin{aligned}
sold^{|M|+1}_{s,k^*_{|M|+1}}, xs^{|M|+1}_{s,k^*_{|M|+1}}, lost^{|M|+1}_{s,k^*_{|M|+1}} &\geq 0 \qquad \forall (s \in SP, k^*_{|M|+1} \in K^*_{|M|+1}) \\
sales^{|M|+1}_{k^*_{|M|+1}}, thc^{|M|+1}_{k^*_{|M|+1}} &\geq 0 \qquad \forall (k^*_{|M|+1} \in K^*_{|M|+1}) \\
txic^{|M|+1}_{k^*_{|M|+1}}, tlc^{|M|+1}_{k^*_{|M|+1}} &\geq 0 \qquad \forall (k^*_{|M|+1} \in K^*_{|M|+1})
\end{aligned}
$$

Note that in (MSP) the number of variables and constraints grow exponentially with the number of events $|K|$ that can occur in any time-period. This is evident from the K^*_m subscripts on the variables, as well as the replication of constraints for each of the K^*_m combination of events at every stage $m \in M$. The y^m and b^m variables are fixed to zero for times $t \in T \backslash T_m$, i.e. the stage-m batch start and size variables are defined for only those time-points in interval T_m. However, this is not the case for the st^m variables due to the material continuity requirements (see (8)). The continuity constraint is also another reason why the final time point in the horizon is included in $T_{|M|}$. In time-periods other than the terminal time-period the continuity constraint links the amounts stored across stages; however, this is not the case for the terminal time-period. But since the amounts stored at $t = T_{|M|} = H$ are required for the revenue/cost calculations, we need the material balances and variables to defined for the final time-point also. Also, the variables y^m and b^m are defined only for compatible task-unit combinations.

4.3 Generating different models from (MSP)

We now illustrate how we can model the various approaches (deterministic, two-stage, multi-stage etc.) for the scheduling problem of interest. To simplify the presentation, we present models for a scheduling horizon with $|M| = 3$ time-periods and $|K| = 2$ possible events in each time-period. Let $T_1 = [0, H_1)$, $T_2 = [H_1, H_2)$ and $T_3 = [H_2, H_3]$. Let $k_1, k_2 \in K = \{1, 2\}$.

4.3.1 Formulation of constraints in (MSP): an illustration

To make the notation used in the formulation of (MSP) clearer, we show below the formulation of one set of constraints (and the associated variables) in model (MSP).

Consider constraints (2) in (MSP), which specify that for every time point t, in a given unit j capable of processing tasks i, at most one task can begin processing in at most one mode.

The stage-1 constraints are formulated as in (18). Note that there are no k indices on the variables and that there is no replication of the constraints for different events $k \in K$. These are identical to the type of constraints we would formulate for deterministic models.

$$
\sum_{i \in I_j} \sum_{l \in L} y^1_{i,j,t,l} \leq 1 \qquad \forall (j \in J, t \in T_1) \tag{18}
$$

The stage-2 constraints and variables are recourse to the event that occurred in the first time-period T_1, and are as shown in (19).

$$\sum_{i \in I_j} \sum_{l \in L} y^2_{i,j,t,k_1,l} \leq 1 \qquad \forall (j \in J, t \in T_2, k_1 \in K) \tag{19}$$

The stage-3 constraints and variables are recourse to the events that occurred in the first and second time-periods and are as shown in (20).

$$\sum_{i \in I_j} \sum_{l \in L} y^3_{i,j,t,k_1,k_2,l} \leq 1 \qquad \forall (j \in J, t \in T_3, (k_1, k_2) \in K^2) \tag{20}$$

Now consider constraints (3) in (MSP) which are more complicated than (2) in that they involve y variables across different stages (note the summation over $m' \leq m$ - see also (7)). The reason why these constraints involve y and b variables across stages is that we do not restrict a task that is started in the first time-period to finish by the end of the first time-period. Hence if a first-stage task has been started at $t = (H_1 - 1)$ and has a duration of 2 units, the associated processing unit is unavailable at $t = H_1$ and $t = H_1 + 1$. This in turn implies that the second-stage y^2 and b^2 variables are to be fixed to zero. The first-, second- and third-stage for (3) are as shown in (21)-(23):

$$\sum_{i \in I_j} \sum_{l \in L} \sum_{t'=t-\tau_{ijl}+1}^{t} y^1_{i,j,t',l} \leq 1 \quad \forall (j \in J, t \in T_1) \tag{21}$$

$$\sum_{i \in I_j} \sum_{l \in L} \sum_{t'=t-\tau_{ijl}+1}^{t} (y^1_{i,j,t',l} + y^2_{i,j,t',k_1,l}) \leq 1 \quad \forall (j \in J, t \in T_2, k_1 \in K) \tag{22}$$

$$\sum_{i \in I_j} \sum_{l \in L} \sum_{t'=t-\tau_{ijl}+1}^{t} (y^1_{i,j,t',l} + y^2_{i,j,t',k_1,l} + y^3_{i,j,t',k_1,k_2,l}) \leq 1 \quad \forall (j \in J, t \in T_3, (k_1, k_2) \in K^2) \tag{23}$$

4.3.2 Deterministic Model

The deterministic approach to solving the scheduling problem consists of determining the optimal schedule to meet the expected value of the product demand. Thus, as shown in the Figure 7, the scenario tree is collapsed to a pair of nodes, a root node ($T = 0$) and a single node that represents a state at $T = H_3$ where the total demand realized is the mean value of the random variable $d_1 + d_2 + d_3$. The deterministic model (DM) can be obtained from (MSP) by setting $|K| = 1$ and replacing the uncertain demands D_s by their mean values (and dropping the m superscripts and k^* subscripts on the variables).

The expected profit of the schedule may be obtained by fixing the y and b variables, and determining the appropriate sales and other flows, for every possible demand realization.

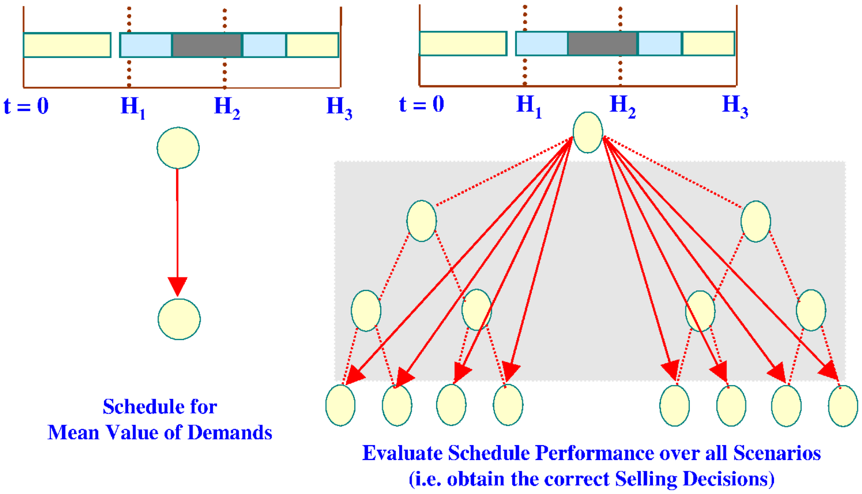
t = 0
H_1
H_2
H_3
t = 0
H_1
H_2
H_3
Schedule for
Mean Value of Demands
Evaluate Schedule Performance over all Scenarios
(i.e. obtain the correct Selling Decisions)

Figure 7: Deterministic Approach

4.3.3 Two-stage Model

Although the two-stage models explicitly consider the uncertainty in the demands, we restrict all the scheduling decisions (i.e., the binary variables y and continuous variables b and st) for the entire horizon (i.e., all time-periods) to be first-stage decisions (see Figure 8). The two-stage model (TSM) corresponds to dropping in (MSP) the k^* subscripts from y, b and st , as well as from the scheduling constraints (2)-(8). The second-stage decisions are the continuous variables $sold, lost, xs, sales, txic, tlc, thc$, which involve the calculation of the amounts sold, lost, in excess etc. Thus, we have a set of second-stage decisions for every scenario $k^*_{|M|+1} = (k_1, k_2, k_3)$, although there is only one set of scheduling decisions. We present a generic two-stage model (TSM($kk^*_{m'}$)) in Appendix A that will be used as part of our approximation strategy.

Note that the two-stage model has the same number of binaries as the deterministic model since all scheduling decisions are first-stage variables. In essence, the two-stage model is formulated to provide a robust schedule. Of course, if there is only one time-period, the two-stage model is equivalent to (MSP).

5 Approximate Solution Approach

We refer the reader to Kall and Wallace[22] and Birge and Louveaux[23] as basic references for the theory and applications of multistage stochastic programs. As observed in Römisch and Schultz[24] and Ahmed *et al.*[25], much of the work in the area of multistage stochastic programs has focussed on stochastic linear programs (i.e., without integer requirements or

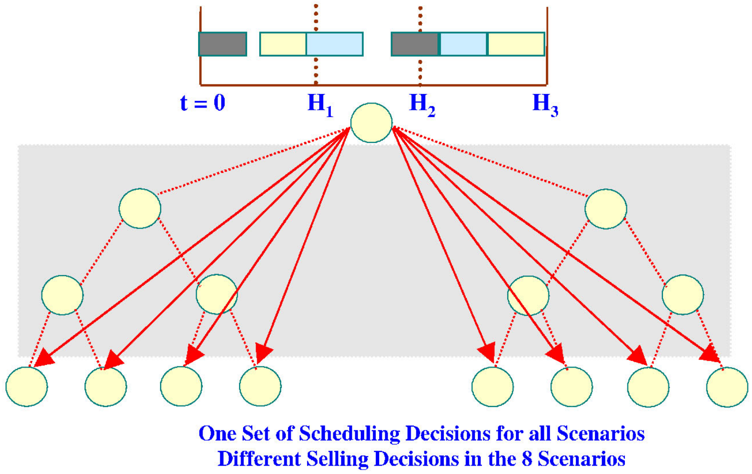
t = 0
H_1
H_2
H_3
One Set of Scheduling Decisions for all Scenarios
Different Selling Decisions in the 8 Scenarios

Figure 8: Two-stage Approach

nonlinearities). This is mainly due to the convexity of the value function of the stochastic linear programs, which breaks down in the case of stochastic integer programs. Thus, in our multistage stochastic programs, where a significant number of the recourse variables are binary variables (for example, the allocation of processing modes in the later time-periods), approaches based on convex programming which work for stochastic linear programs are not applicable. To reduce the computational expense associated with solving the multistage stochastic integer programs, we present an approximate solution strategy that relies on solving a number of smaller models. In particular, the two-stage stochastic models we will use in this work have a purely continuous second stage; however there are binary and continuous first stage variables.

5.1 Solution of Two-stage Models within a Shrinking Horizon framework

Some of the common approaches to the optimization and implementation of schedules over several time periods are rolling-horizon or shrinking-horizon strategies. These are widespread in process control, especially Model Predictive Control and are increasingly of interest in scheduling problems (see for instance, Wilkinson[26], Dimitriadis *et al.*[27], Elkamel and Mohindra[28], Pinedo[29], Perea-Lopez *et al.*[30]).

A typical shrinking-horizon algorithm works as follows. An approximate model (i.e., simplification of the original problem) is formulated for the entire horizon of $|M|$ time-periods. Although the solution of this model gives us decisions for the entire horizon, only the decisions for the first time-period are implemented. The state of the system is updated at the end of the first time-period, and an approximate model is solved for the remaining $|M| - 1$ time-periods (with decisions for the elapsed first time-period already fixed). The algorithm proceeds in this manner till decisions have been fixed for the entire horizon.

In this work we use two-stage models as the approximation models in the shrinking-horizon strategy. The two-stage models (TSM) have the structure as in Section 4.3.3, with the scheduling decisions as first-stage variables (common for all scenarios) and the calculations for sales, costs, revenues as second-stage variables at the end of the horizon. If we implement the shrinking-horizon strategy in real-time, we would obtain a feasible schedule for the entire horizon at each step (see Figure 9 for an illustration of the shrinking-horizon strategy for a three time-period example). Calculating the sales and other decisions at the end of the horizon would give us the profit for the particular scenario that was realized over the time horizon. If we were just interested in implementing a scheduling system in real-time, carrying out the steps similar to Figure 9 once as we move forward in time is sufficient. However, a more important objective would be knowledge of the performance of the strategy over all possible scenarios.

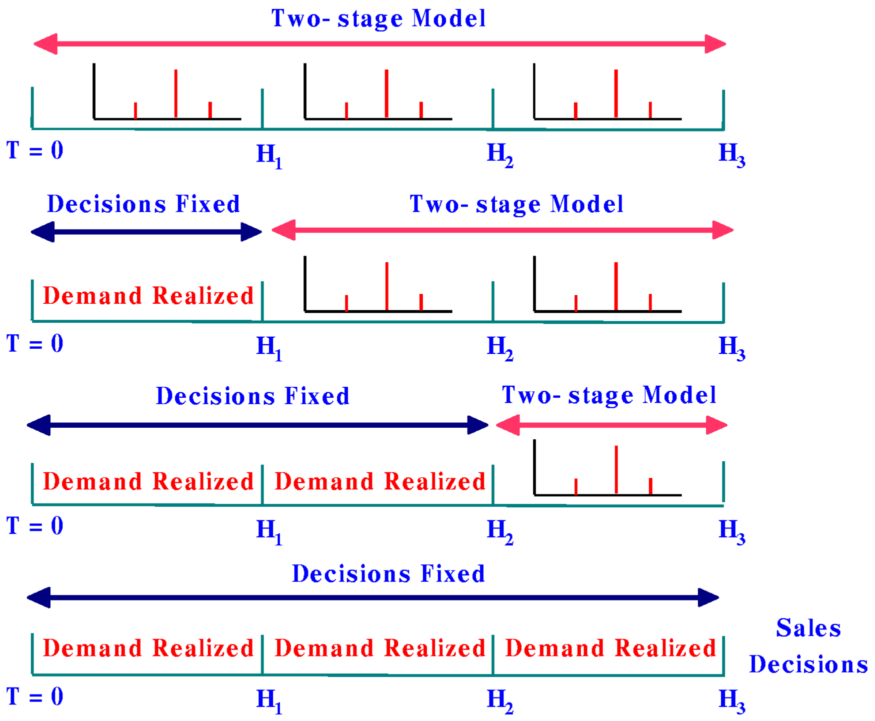

Figure 9: Application of Shrinking Horizon Strategy to a Scenario

To evaluate the performance of the proposed strategy (SHT), an algorithm can be constructed as shown below. This algorithm is mainly motivated by (i) the observation that the first two-stage model needs to be solved only once since the scheduling decisions for the first time-period will be the same for all scenarios with the shrinking-horizon strategy, and (ii) linking the decision-making process at the end of each time-period with the scenario tree. Thus, if we store the solutions (mainly, the scheduling variables y and b) of the two-stage models at level m of the tree, we can use these values for the solution of the models at level $m+1$. Furthermore, for convenience of implementation, rather than constructing different two-stage models (with different number of scenarios) that are to be solved at the different levels of the scenario tree, we make use of the structure of the two-stage model at the root node by appropriately modifying the values of the parameters in the model. We refer the reader to Appendix A for a description of the generic two-stage models that we solve at the different nodes of the tree.

Algorithm *SHT*

1. Initialization.

   Set $t = 0, ctr = 1$

   Solve $TSM(kk_1^*)$, the two-stage model for the entire horizon (corresponding to the root node).

   Store scheduling decisions y^1, b^1, st^1 (i.e., for time-period $ctr = 1$) in $PRED(kk_1^*)$.

2. Increment

   Set $ctr = ctr + 1$

3. Termination Check

   If $ctr > |M| + 1$, there are no more two-stage models to be solved. Go to Step 5.

4. Solving Two-stage models at level $ctr - 1$

   Solve the two-stage models corresponding to level $ctr - 1$ of the scenario tree, i.e. $TSM(kk_{ctr}^*)$. These models have the scenario demands adjusted by the demand realization till H_{ctr}, scheduling decisions fixed till H_{ctr} to values from $PRED(kk_{ctr-1}^*)$.

   Store scheduling decisions $y^{ctr}, b^{ctr}, st^{ctr}$ (i.e., for next time-period ctr) in $PRED(kk_{ctr}^*)$.

   Go to Step 2.

5. Termination

   Compute Expected Profit through probability-weighted sum of profits at nodes in level M of scenario tree. Stop.

Figures 10-13 illustrate how we would proceed for a three time-period problem. In these figures the arrows between the nodes at a given level m and the nodes at level 3 indicate the two-stage approximation being performed at level m. The hatched rectangles in the Gantt charts adjacent to the nodes at which the computations are being performed indicate the time-periods for which the scheduling decisions are already fixed and the demand realized, while the horizontally shaded rectangles in Figures 11-13 indicate levels of the tree at which the computations have already been performed. Thus in Figure 10 at the root node, no demand has been realized yet and no scheduling decision has been made, while at the nodes in Level 1 in Figure 11, decisions have already been fixed for the first time-period (to those from the solution of the two-stage model at the root node) and the demand for the first time-period already realized. In Figure 12, decisions have already been fixed for the first two time-periods and demands in the first two time-periods already realized. Finally, in Figure 13, note that the scheduling decisions have been fixed for the entire horizon and all demand realized, which implies that only LPs have to be solved to calculate the correct values of the sales and other variables.

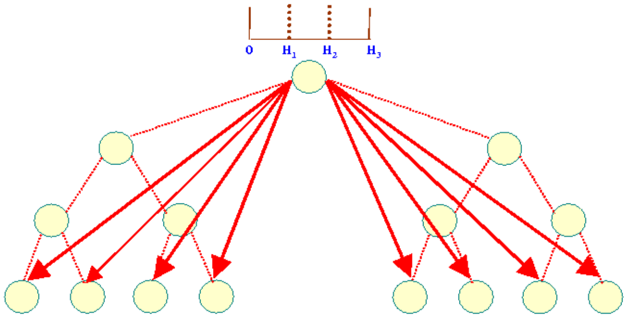

Figure 10: Initial Step

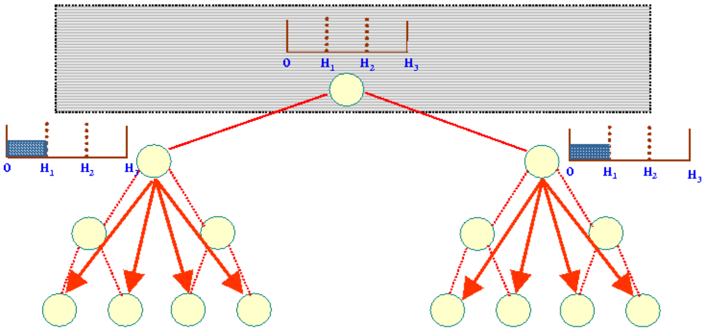

Figure 11: Solving 2 MILPs at Level 1

With this approximation strategy, we would have a look-up table of scheduling decisions for every node of the scenario tree (in a sense resembling the multistage solution). The

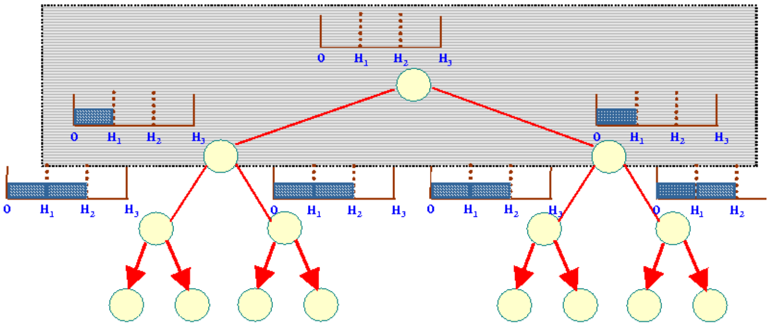

Figure 12: Solving 4 MILPs at Level 2

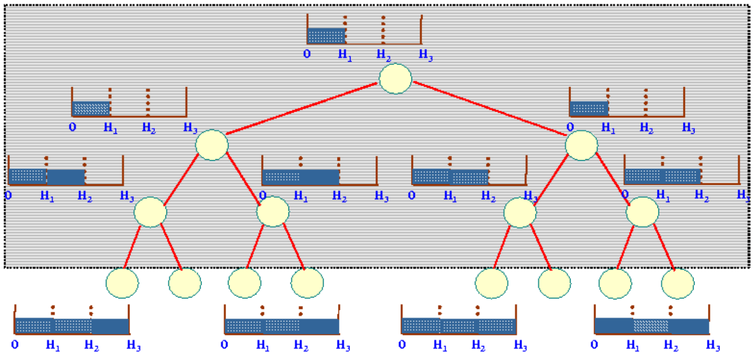

Figure 13: Solving 8 LPs at Level 3

computational advantages of the solution strategy are as follows. Assume that we require B binaries for modeling the scheduling of a time-period. A deterministic model with M time-periods then has $B \cdot M$ binaries. With K possible events per time-period, the number of binaries in a stochastic multistage model would be $B \cdot (K^M - 1)/(K - 1)$. The idea is that instead of solving the stochastic multistage MILP model with $B \cdot (K^M - 1)/(K - 1)$ binaries, we solve a series of MILPs that are smaller in size. Thus, initially we would solve a stochastic two-stage MILP with $B \cdot M$ binaries, corresponding to the root node of the scenario tree. We would then solve K two-stage MILPs with $B \cdot (M - 1)$ binaries and so on. At the lower levels of the tree, where we solve a number of MILPs, the number of binaries is quite small, given that most of the decisions for the horizon are already fixed. Also, the solutions of the model at a given node can be used as a starting solution for the model at its child nodes, thereby further reducing the computational effort.

5.2 Quality of approximation

Although no rigorous theoretical analysis is possible, the quality of the solution (i.e., the expected profit) provided by the approximation strategy can be compared with the following:

(i) the expected profit from a similar shrinking-horizon approach but based on solving deterministic models at each step (SHD), in which expected demands are considered;

(ii) the expected profit from the multistage stochastic program (i.e., the exact solution), or the LP relaxation, if it cannot be solved to optimality;

(iii) the expected profit from the Wait-and-See solutions (WSS);

In comparison (i), we would generally expect our strategy to perform better since the deterministic models do not have any information about the uncertainty. Furthermore, our strategy will clearly provide an expected profit less than or equal to those in (ii) and (iii). Recall that in (iii), we solve the scheduling problem for each scenario and then combine the profits from the scenarios using a probability-weighted average. This provides an upper bound on the expected profit, since it is clearly better to know the value of the random variable before making the decision than having to make the decision before knowing it (Kall and Wallace[22]). As we will see later, the upper bound provided in (iii) can be significantly higher than even the LP relaxation of the multistage stochastic program; thus we use the minimum of these two to provide the tightest upper bound.

5.3 Revisiting the Motivating Example

We revisit the motivating example discussed earlier and compare the solutions from the approximation strategies with that of the multistage model.

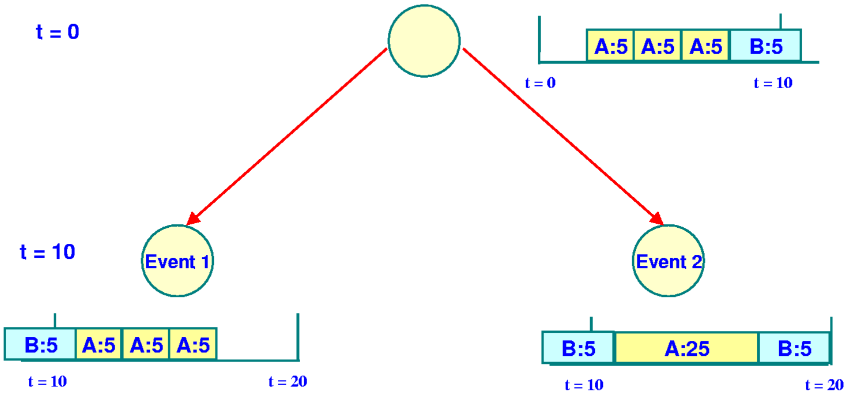

Figure 14: Motivating Example - Solution from SHT approach

Figure 14 presents the optimal schedules using the proposed shrinking horizon two-stage (SHT) approach which yields an expected profit of $5325, which is in fact identical to that of the three-stage solution. Note that for the first time-period between $t = 0$ and $t = 10$ we implement the scheduling decisions that are optimal to the two-stage model for the entire horizon (compare the schedule at the root node in Figure 14 to that shown in Figure 3. Furthermore, since there are no holding costs in this example, the timing and sizing of the batches is not as critical - for e.g., two batches of 5 tons each of A is equivalent to one batch of 10 tons of A. Thus, when we compare Figure 14 with the optimal three-stage solution in Figure 4, we see that we produce identical amounts for the different scenarios.

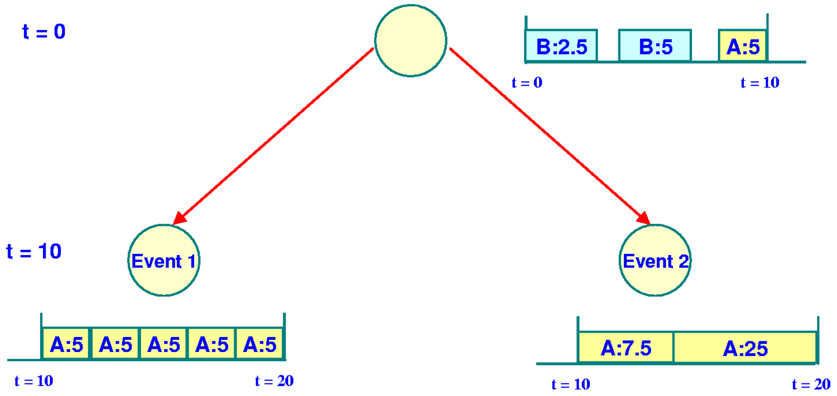

Figure 15: Motivating Example - Solution from SHD approach

In the simpler shrinking horizon deterministic (SHD) approach, we solve the deterministic model for the entire horizon, implement the decisions for the first time-period (i.e., fix the binary variables for the first time-period) and solve two different deterministic models corresponding to the nodes at Level 1 of the scenario tree (part of it shown in Figure 15. For instance, at Node 1 in Level 1 we would update the demand to be the sum of already-realized demand (i.e., 10 for A and 0 for B) and the mean value of the demand for the next time-period, while at Node 2 in Level 1, the demand would be the sum of (20 of A, 5 of B) and the mean value of the demand for the next time-period. At the nodes in Level 2, where the demand has been entirely realized, we would calculate the selling decisions (note that we have not shown these nodes at Level 2). Finally, we combine the resulting profits weighted with the scenario probabilities to get the expected value of this approach ($4681), which is considerably lower than with the proposed SHT strategy. The predicted schedules thus obtained are shown in Figure 15.

The Wait and See approach yields an expected profit of $5375, which represents an upper bound to the solution of the multistage model ($5325).

Figure 16 compares the profits of the different strategies in the four scenarios (see Section 3 for discussion of the scenarios), with the probabilities of the scenarios indicated above the

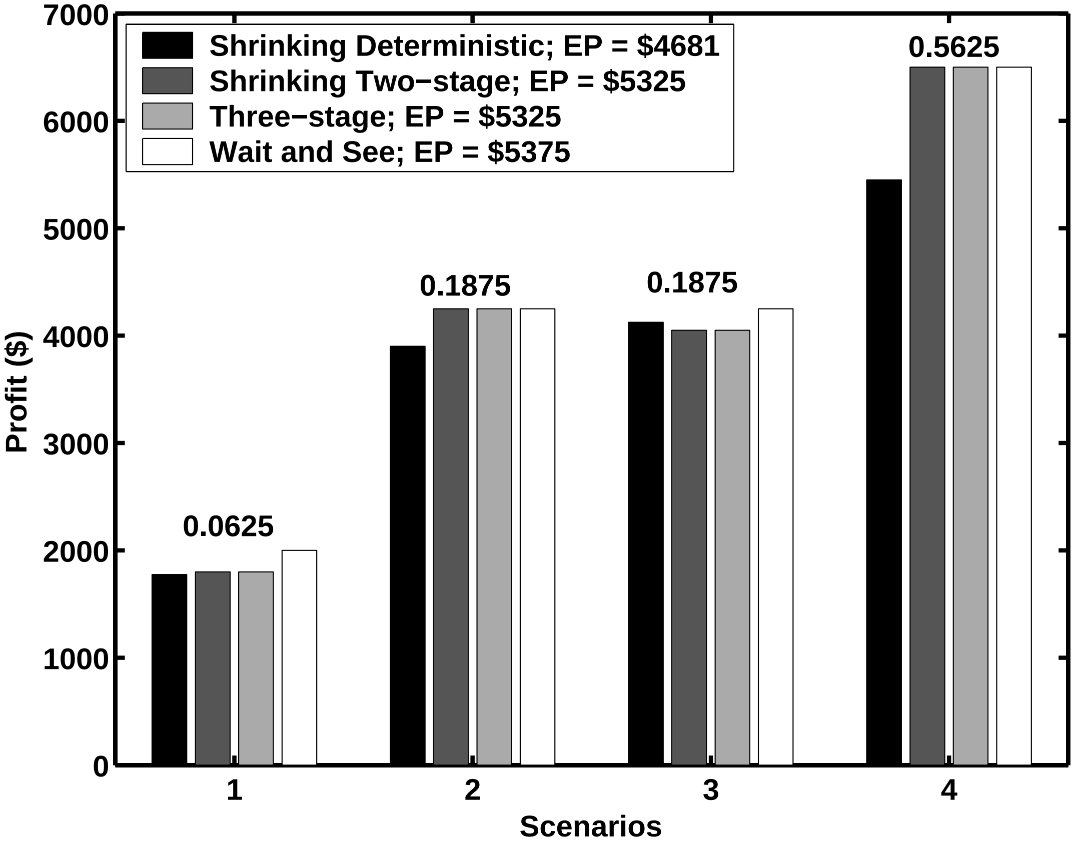

Figure 16: Motivating Example - Comparison of Scenario Profits

bars in the plot. The SHT approach matches the three-stage solution in every scenario, resulting in the same optimal expected profit. The shrinking horizon deterministic approach performs worse than the three-stage solution in scenarios 1 and 2 and considerably worse in Scenario 4 where the maximum demand is realized in each time-period, resulting in a very low expected profit (12 % worse than the optimal expected profit). Note that the shrinking-horizon deterministic approach performs even worse than the static two-stage solution which has an expected profit of $5275. Note that it is possible that the shrinking horizon two-stage approach can yield a worse profit than the shrinking-horizon deterministic approach in a particular scenario (see, for instance, Scenario 3 in Figure 16), but the expected profit from the SHT strategy will be greater than or equal to the expected profit from the SHD strategy.

6 Computational Results

In this section, we present several examples to illustrate the performance of the various strategies in the scheduling of STNs. In the following examples, the two-stage stochastic MILPs that were solved as part of the approximation strategy were not very expensive to solve. Thus, we did not have to resort to special algorithms. For problems with several thousands of scenarios, methods such as Lagrangean decomposition (Fisher[32]; Guignard and Kim[33]) can be easily adapted for the solution of the two-stage models in this work. All models were formulated and solved using GAMS with CPLEX-7.5 (ILOG CPLEX Inc.[31]) as the MILP solver on a Pentium III/930MHz machine running Linux.

6.1 Example 1

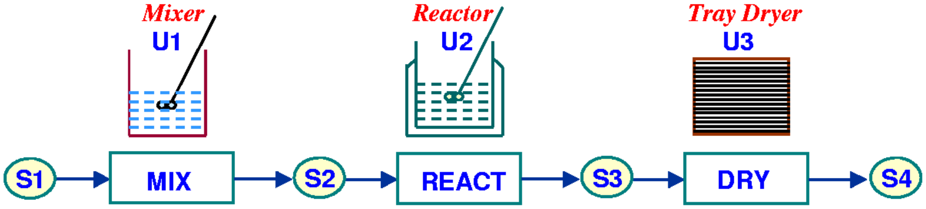

Figure 17: Example 1 - STN Representation

Figure 17 is an STN-representation of a process that produces a single product ($S4$) through three stages, viz. mixing, reaction and purification. Each task can be carried out in three possible processing modes. Each of these processing modes has a specific range for the batch size and a corresponding processing time over the range.

We present two examples which differ in the number of time-periods and the length of the time horizon. The data for these example are adapted from Ierapetritou and Floudas[34] and are presented in Tables 20-21 in Appendix B.

6.1.1 Example 1a

In this example the horizon of interest is of duration 18 time units (from $t = 0$ to $t = 18$), with there being 3 equally long time-periods. The demand for product $S4$ in each time-period has the same distribution which is shown in Table 4. Since there are 2 possible events for each of the 3 time-periods, there are 2^3 scenarios for the entire horizon.

At the beginning of the horizon only raw material $S1$ is available. The intermediate states $S2$ and $S3$ have limited storage capacity as shown in Table 21. There is no demand for the intermediate states; however, the cost of producing excess of these intermediate states is penalized through a non-zero excess-cost. There is a holding cost for the final product $S4$, so we would like to produce $S4$ as late as possible in the horizon.

Table 4: Example 1a - Time-period Demand Distribution

~~Event~~	~~Probability~~	~~Demand for Product S4~~
~~*1*~~	~~0.2~~	~~0~~
~~*2*~~	~~0.8~~	~~30~~

Table 5 displays the computational results from applying the various approaches. In this example the solution times are negligible (under a CPU second) for the different approaches, hence we emphasize the differences in the solutions obtained.

Although the optimal profit predicted by the deterministic model was \$70,200, the schedule when evaluated under demand uncertainty had a significantly smaller expected profit of \$52,690.

The two-stage model when solved to optimality provides a schedule with 13.2% larger expected profit than the deterministic approach (\$60,300). Note that although we seek only one schedule in both these approaches the two-stage approach performs significantly better since the two-stage model considers the performance of a given schedule in all scenarios and can trade-off a larger loss in a less probable scenario with a larger profit in a more probable scenario.

Unlike the four-stage model where we consider the possibility of implementing different schedules after each time-period, the three-stage model is formulated such that some time-periods are aggregated to form a stage. Two three-stage formulations are possible: (a) a formulation where the first stage corresponds to the first time-period, the second stage corresponds to the second and third time-periods, and the third stage consists of the sales calculations at the end of the horizon; (b) a formulation where the first stage corresponds to the first two time-periods, the second stage corresponds to the third time-period, and the third stage consists of the sales calculations at the end of the horizon.

Figure 18 displays the optimal solution for the first three-stage model, while Figure 19 displays the optimal solution for the second three-stage model. The latter model provides

Table 5: Example 1a - Computational Results

Model Type	**Equations**	**Variables**	**Binaries**	**CPU secs**	**E[Profit], \$**
Deterministic	541	426	144	0.07	52,690
Two-stage	605	483	144	0.05	60,300
Three-stage (a)	977	1024	234	0.47	63,600
Three-stage (b)	1209	1240	252	0.37	65,840
Four-stage	1385	1776	306	0.67	66,120

for more scheduling responsiveness to the demands, and hence has a larger expected profit of \$65,840 than the \$63,600 of the former. Also note that in Figure 19 the nodes at $t = 12$ display the *total* demand realized so far in the first two time-periods. Thus, the central node with total demand 30 represents two scenarios: (i) where the demand for $S4$ is 0 ($Event1$) in the first time-period and 30 ($Event2$) in the second, and (ii) the reverse.

The solution from the four-stage model has the maximum expected profit of \$66,120 since it allows for the possibility of reacting to every event over the horizon. Figure 20 displays the optimal 2^{nd} and 3^{rd}-stage decisions when the demand realized in the first time-period is 0 (i.e. $Event1$), while Figure 21 displays the optimal 2^{nd}- and 3^{rd}-stage decisions when $Event2$ occurs in the first time-period. From these figures it is clear that the four-stage model pushes for a production of 30 tons of $S4$ for every time-period, although it delays the production of $S4$ until the last time-period. Thus, we see that at the $(0, 0)$ node in Figure 20, having observed the demand in the first two time-periods to be 0, we produce only 30 tons of $S4$, since that is the maximum demand possible. Similarly at the $(0, 30)$ node, we produce 60 tons of $S4$, taking into account the 30 tons already placed in the second time-period and a probable 30 tons in the third time-period. This can also be observed in the schedules displayed in Figure 21.

It is also instructive to compare the four-stage solution with the upper bound from the Wait and See solutions. The optimal schedules for the various scenarios are shown in Figure 22. These show that it is possible to meet the demand exactly for each scenario, thus incurring no penalties for excess production. Combining the profits in these scenarios with the scenario probabilities as weights, we get the expected profit of the Wait and See strategy as \$69,696. The difference between the expected profit from the Wait and See solutions and the expected profit from the four-stage solution (i.e., $69, 696 - 66, 120 = 3576$) represents the Expected Value of Perfect Information (EVPI) Kall and Wallace[22], which is the maximum amount we would be willing to pay for information to resolve the uncertainty in demands.

Figure 23 presents the distribution of the profit of the various strategies. Here we have plotted the probability of making more than a given amount of profit. Although the deterministic schedule has a smaller worst-case loss than and dominates the two-stage schedule over the lower range of profit, the maximum profit it can make is substantially smaller than in the other models. The two-stage model takes into account the maximum possible demand and thus provides a schedule that can match this demand, which results in the same maximum profit as in the other strategies.

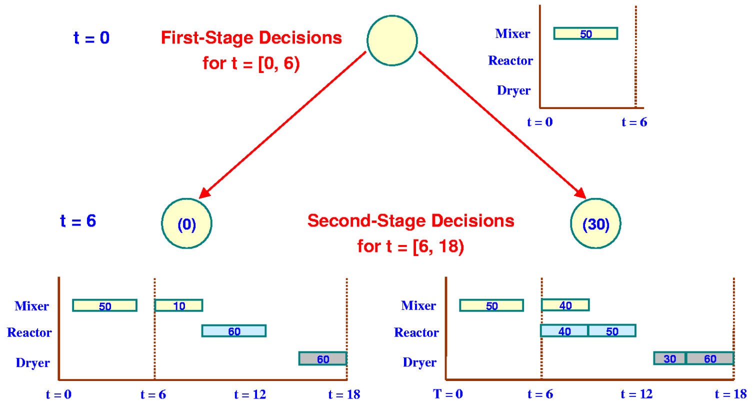

Figure 18: Example 1a - Optimal solution to Three-stage(a) model

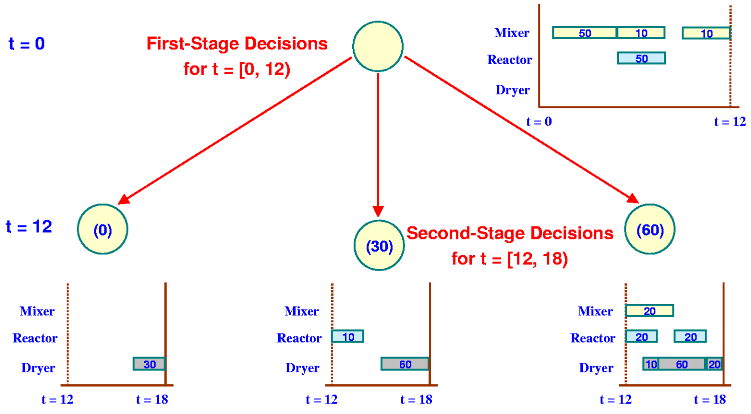

Figure 19: Example 1a - Optimal solution to Three-stage(b) model

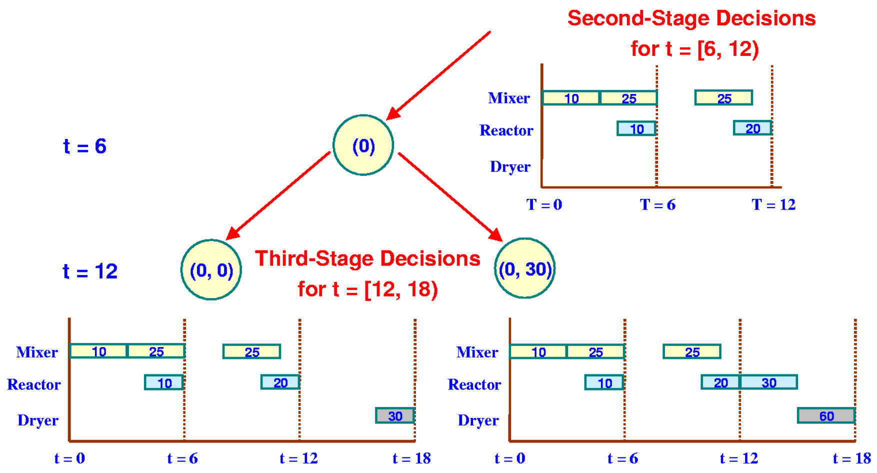

Figure 20: Example 1a - Optimal four-stage Solution - Left half of tree

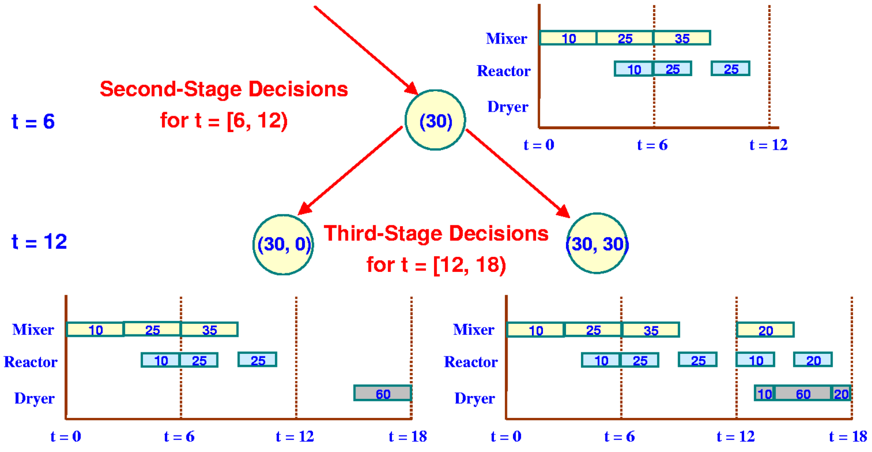

Figure 21: Example 1a - Optimal four-stage Solution - Right half of tree

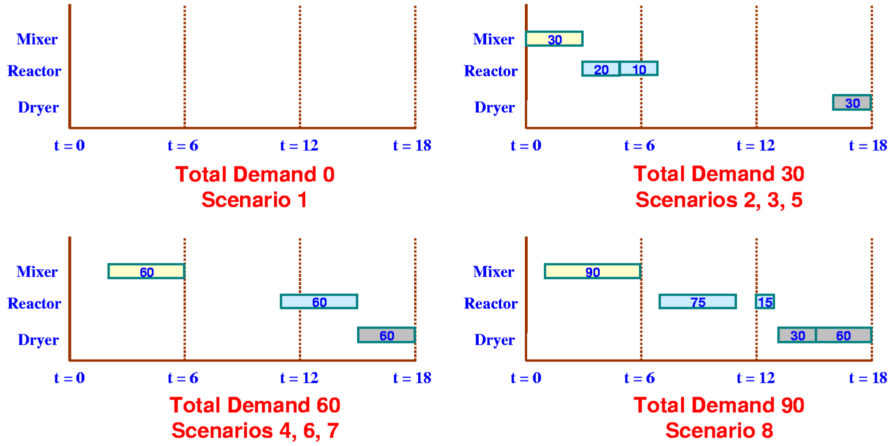

Figure 22: Example 1a - Wait and See Schedules

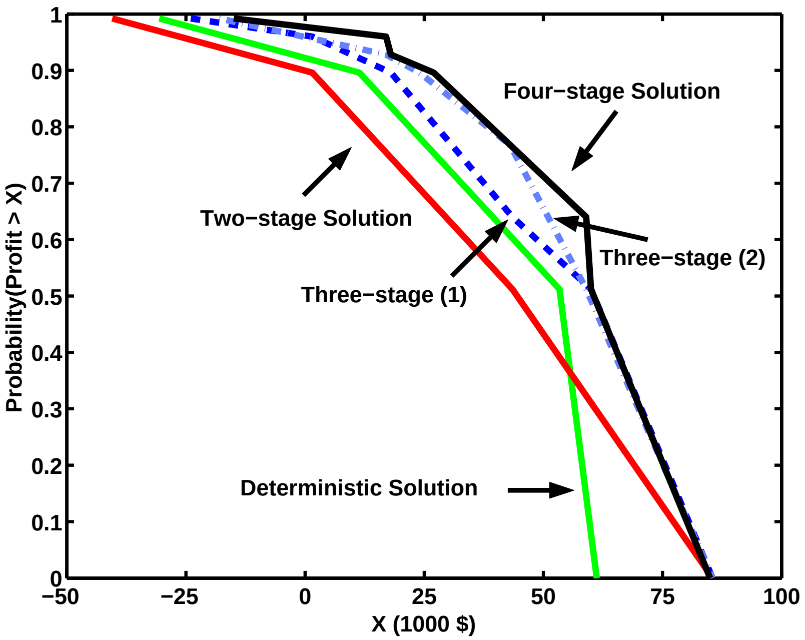

Figure 23: Example 1a - Comparison of Distributions of Profit

Comparison with shrinking-horizon approaches

The shrinking-horizon approach with deterministic models provided an expected profit of $58,589 while the shrinking-horizon with two-stage models provided an expected profit of $64,920 (see Table 6). Note that the shrinking horizon two-stage strategy provides an expected profit within 2% of the optimal solution of the four-stage model ($66,120), while the shrinking-horizon deterministic approach provided an expected profit ($58,589) even less than the static two-stage approach. The largest models solved in these approximation strategies have the same size as the deterministic and two-stage models shown in Table 5. Even though the deterministic and two-stage models at different levels of the tree were very inexpensive to solve, we had to solve many of these models, and therefore the total CPU time for these approaches exceeds the time taken to solve the four-stage model (see Table 6). The CPU time for obtaining the SHD and SHT solutions represents the time required for solution of 1 MILP at the top of the tree, 2 MILPs at level 2, 4 MILPs at level 3 and 8 LPs at level 4. Note that at level 4 we have LPs rather than MILPs since we have fixed all the binary variables.

Table 6: Example 1 - Comparison of Approximation Strategies

Approach	**E[Profit], $**	**CPU Time (secs)**
SHD	58,589	0.71
SHT	64,920	0.69
Four-stage	66,120	0.67

A comparison of the scenario profits from the approximation strategies with the four-stage solution and Wait and See solutions is provided in Figure 24. The numbers above the bars in the plot indicate the probability of each scenario. As can be seen from the plot, the performance of the shrinking horizon two-stage approach is comparable to the four-stage solution in almost all scenarios, and overall much superior to the shrinking-horizon deterministic approach which has a smaller expected profit than even the static two-stage solution (compare Table 5 with Table 6).

6.1.2 Example 1b

In this example the horizon of interest is of duration 60 units, with five equally long time-periods. The time-period demands for product $S4$ have the distribution shown in Table 7. There are 3 possible events for each of the 5 time-periods, resulting in a total of 243 scenarios for the entire horizon. Note that the probability distributions are different for the

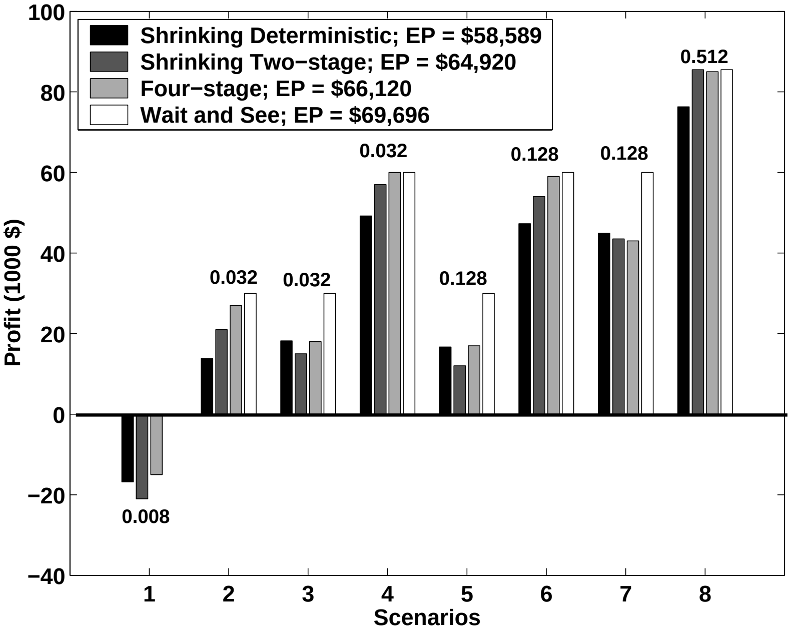

Figure 24: Example 1a: Comparison of Scenario Profits

five time-periods. We can see that the second time-period has the highest mean demand while the first and last time-periods have comparatively high variances (deviation about the mean) relative to the mean demand. The high variances could reflect the case where there is a history of rush orders being placed towards the end of the horizon.

As before, only raw material $S1$ is available at the beginning of the horizon. The intermediate states $S2$ and $S3$ have limited storage capacity, but larger capacities than in Example 1a (i.e., 500, instead of 100). There is no demand for the intermediate states; however, the cost of producing excess of these intermediate states is penalized through a non-zero excess-cost. There is a holding cost for the intermediate states and the final product. The price and cost data for the different states are presented in Table 22 in Appendix B.

Computational results from applying the static approaches to this example are summarized in Table 8. Although the schedule from the solution of the deterministic model was predicted to have a profit of around $767,350, the actual expected profit under uncertainty was only $656,230. The two-stage model provided a solution with 2.4% larger expected profit, with a small increase in computational time. Both these models were solved with an optimality gap of under 0.5%.

The six-stage model's solution reported in Table 8 represents the best solution obtained after different attempts at solving the model (these attempts differed in the CPLEX options such as branching and cut-generation strategies). The best options were found to be those

Table 7: Example 1b: Time-period Demand Distributions

Time-period	Probability	Demand for S4
1	0.5	25
	0.35	125
	0.15	200
2	0.3	50
	0.35	150
	0.35	200
3	0.5	25
	0.3	100
	0.2	150
4	0.5	50
	0.3	75
	0.2	100
5	0.6	25
	0.2	75
	0.2	100

that placed an emphasis on finding feasible solutions, and even with these, the model could not be solved to optimality. Nevertheless the reported solution is still within 0.75% of the best possible (i.e., the best possible solution was $727,800 when the resource limit was exceeded). Despite the dramatic increase in CPU time (50,000 seconds versus 20 seconds for the other two models) required for solving the six-stage model to (near) optimality, the six-stage solution provides a 7.4% improvement over the two-stage solution, and a 10% improvement over the deterministic solution.

Table 8: Example 1b - Computational Results

Model Type	Equations	Variables	Binaries	CPU secs	E[Profit], 1000$
Deterministic	1717	1350	522	3	656.23
Two-stage	2732	2248	522	20	672.04
Six-stage	45592	53719	11610	> 50,000	722.43

Comparison with shrinking-horizon approaches

In this example the models at the lower levels of the tree were solved to within 0.5% optimality in both SHD and SHT approaches. We tried two different strategies in CPLEX-7.5 for solving the sub-problems: with the emphasis on finding, (i) good feasible solutions, without proving optimality, (ii) good feasible solutions with proving optimality. From the computations we observed that (i) worked better than (ii) for the SHT strategy while there was not much of a difference for the SHD strategy with either (i) or (ii). Indeed, for the SHT strategy, with (i) most of the sub-problems were solved to within the desired optimality gap within a minute of CPU time, while with (ii) it took over 1000 CPU seconds for some sub-problems. For the SHD strategy, although some of the solutions of the sub-problems at the higher levels of the scenario tree were different, there was not much of a difference in the overall expected profit with either (i) or (ii). Below, we present the results using (i) for both SHT and SHD strategies.

The shrinking-horizon approach with deterministic models provided an expected profit of \$697,000, while the shrinking-horizon with two-stage models provided an expected profit of \$717,320, an improvement of about 2.8% over the SHD. Note that the shrinking horizon two-stage strategy also provides an expected profit within 0.7% of the best solution obtained from solving the six-stage model in a fraction of the CPU time.

Table 9: Example 1b - Comparison of Approximation Strategies

Strategy	**Profit (1000\$)**			
	Minimum	**Maximum**	**Expected Value**	**CPU secs**
SHD	324	966	697.49	180
SHT	299	1030	717.32	360
Six-stage	296	1108	722.43	> 50,000
WSS	341	1290	762.09	3090

6.2 Example 2

Figure 25 is an STN-representation of a multiproduct batch plant that can produce 4 products ($P1$, $P2$, $P3$ and $P4$) through 8 tasks from 3 feeds ($F1$, $F2$, $F3$) and various intermediates ($I4$ to $I9$). Six processing units are available to carry out the tasks, and there are 3 processing modes for each unit/task combination. The unit-task compatibilities and the batch size/processing time data are adapted from Ierapetritou and Floudas[34] and can be found in Balasubramanian[36]. Each task can be carried out in three possible processing modes in suitable processing units, with each processing mode accommodating a given range of batch sizes and requiring a specific processing time. The intermediate states $S2$ and $S3$ have limited storage capacity.

We present two examples which differ in the demand distributions. In both examples the horizon of interest is of duration 18 time units (from $t = 0$ to $t = 18$), consisting of 3

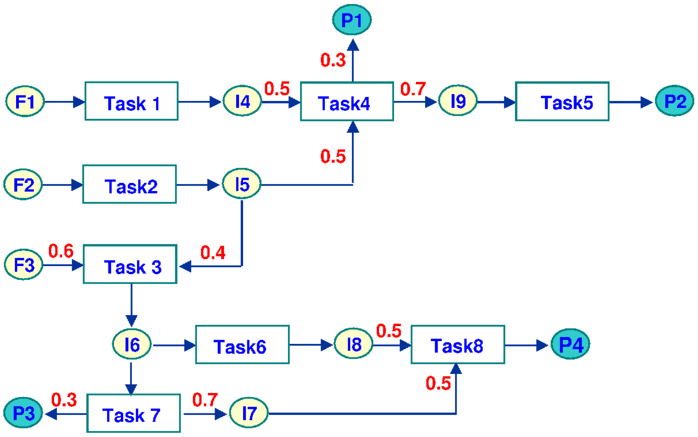

Figure 25: STN Representation for Example 2

equally long time-periods.

6.2.1 Example 2a

Table 10: Example 2a: Time-period Demand Distributions

Event	Probability	Demands (P1, P2, P3, P4)
1	0.5	(360, 600, 144, 264)
2	0.5	(1440, 2400, 576, 1056)

In this example the time-period demands for the products have the same probability distribution for all three time-periods (see Table 10). The demands have considerable variance.

Computational results from applying the static approaches to this example are summarized in Table 11. Although the length of the time horizon is the same as in Example 1, there are considerably more variables and equations in this example due to the larger number of tasks, units, states, etc. The CPU time reported for the deterministic model in Table 11 includes the time for solving the deterministic model and evaluating the obtained schedule under uncertainty; the evaluation of a given schedule is done under 1 CPU second. The four-stage model was solved to within about 0.6% of optimality (i.e., the best possible solution had an expected profit of $116,421).

Comparison with shrinking-horizon approaches

Table 11: Example 2a - Computational Results

Model Type	Equations	Variables	Binaries	CPU secs	E[Profit], $
Deterministic	1408	1176	408	42	98,380
Two-stage	1549	1289	408	25	105,238
Four-stage	3605	4605	912	> 50,000	115,769

Table 12: Example 2a - Comparison of Approximation Strategies

Strategy	Profit ($)			
	Minimum	Maximum	Expected Value	CPU secs
SHD	15,368	165,701	111,104	250
SHT	12,308	157,875	114,208	200
Four-stage	10,027	166,371	115,769	> 50,000
WSS	61,916	204,116	141,117	2800

As seen in Table 12 the shrinking-horizon approach with deterministic models provided an expected profit of $111,104 while the shrinking-horizon with two-stage models provided an expected profit of $114,208, an improvement of about 2.8%. Note that the shrinking horizon two-stage strategy also provides an expected profit about 1.3% lower than the best solution obtained from solving the four-stage model for orders of magnitude larger CPU time. The SHD strategy performed significantly better than static two-stage approach, thus providing incentive to revise the schedule after the time-periods. In this example the models at the lower levels of the tree were solved to within 0.1% optimality in both SHD and SHT approaches. The large CPU time for obtaining all the Wait and See solutions is mainly due to the large solution time for Scenario 8 which has maximum demand (the solution was terminated after about 2600 CPU seconds with a gap of 0.3%). The CPU time for solving the deterministic model at node 2 at level 1 of the scenario tree was 220 seconds, while it was only 150 seconds for the corresponding two-stage model; the remaining solution times were comparable for both approaches.

6.2.2 Example 2b

The time-period demands for the products has the distribution shown in Table 13. Note that the probability distributions are different for the three time-periods. In this example, since there are 4 possible events for each of the 3 time-periods, there are a total of 64 scenarios for the entire horizon.

Computational results from applying the static approaches to this example are summa-

Table 13: Example 2b: Time-period Demand Distributions

Time-period	Probability	Demands (P1, P2, P3, P4)
1	0.2	(0, 0, 0, 0)
	0.3	(600, 1000, 250, 150)
	0.3	(1200, 1500, 500, 300)
	0.2	(1800, 2500, 700, 450)
2	0.1	(300, 500, 100, 50)
	0.4	(600, 1000, 250, 100)
	0.4	(1200, 1500, 500, 200)
	0.1	(1500, 2000, 600, 300)
3	0.4	(150, 250, 50, 50)
	0.4	(300, 500, 250, 100)
	0.1	(600, 1000, 250, 150)
	0.1	(900, 1500, 500, 200)

rized in Table 14. The CPU time reported for the deterministic model in Table 14 includes the time for solving the deterministic model and evaluating the obtained schedule under uncertainty; the evaluation of a given schedule is done under 1 CPU second.

Table 14: Example 2b - Computational Results

Model Type	Equations	Variables	Binaries	CPU secs	E[Profit], $
Deterministic	1408	1176	408	10	59,573
Two-stage	2732	2248	408	35	62,945
Four-stage	11971	13179	2640	> 100,000	75,851

A few words about the solution of the four-stage model: the solution reported in Table 14 represents the best solution obtained after several attempts which differed in the branching strategies as well as the cut generation in CPLEX 7.5. With the default options and an emphasis on finding good feasible solutions (CPLEX option: *mipemphasis*1), the four-stage model was solved repeatedly (time limit of 15000 CPU seconds for each run), with a lower cut-off as the best solution obtained in the previous run. The resulting solution with an expected profit of $75,851 is the one reported in Table 14. From the various runs the smallest upper bound (best node remaining) obtained had a value of $77,652.

Comparison with shrinking-horizon approaches

Table 15: Example 2b - Comparison of Approximation Strategies

Strategy	Profit ($)			
	Minimum	Maximum	Expected Value	CPU secs
SHD	8,272	124,101	74,747	60
SHT	11,852	112,017	75,452	180
Four-stage	8,895	121,411	75,851	> 100,000
WSS	20,992	153,631	85,884	650

As seen in Table 15 the shrinking-horizon approach with deterministic models provided an expected profit of $74,747 while the shrinking-horizon with two-stage models provided an expected profit of $75,452, an improvement of about 1% (see Table 15). Note that the shrinking horizon two-stage strategy also provides an expected profit only about 0.5% lower than the best solution obtained from several attempts at solving the four-stage model. The SHD strategy performed significantly better than static two-stage approach, thus providing incentive to revise the schedule after the time-periods. In this example the models at the lower levels of the tree were solved to within 0.5% to 1% optimality in both SHD and SHT approaches.

6.3 Example 3

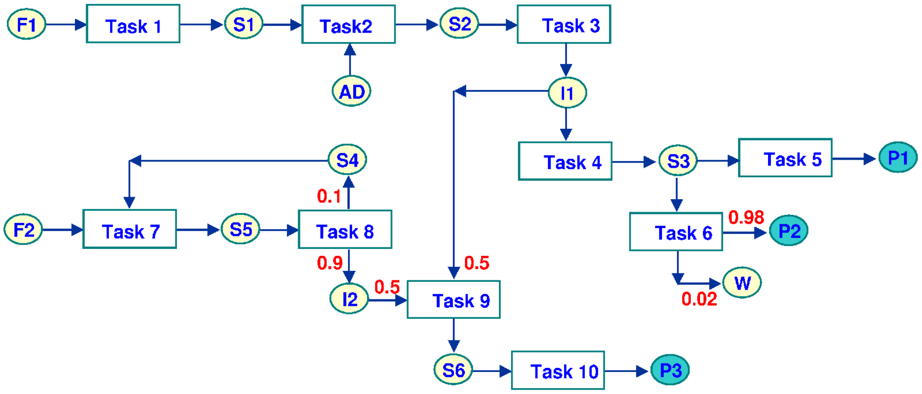

Figure 26: STN Representation for Example 3

Figure 26 is an STN-representation of a multiproduct batch plant that can produce 3 products ($P1$, $P2$ and $P3$) through 10 tasks from 2 feeds ($F1$ and $F2$) and various intermediates. Six processing units are available to carry out the tasks, and there are 3 processing modes for each unit/task combination. The unit-task compatibilities and the batch size/processing time data are adapted from Papageorgiou and Pantelides[35] and can be found in Balasubramanian[36]. The intermediate states $S2$ and $S3$ have limited storage capacity. The horizon of interest is of duration 60 time units (from $t = 0$ to $t = 60$), with 5 equally long time-periods. There are 3 possible events for each of the 5 time-periods, resulting in a total of $3^5 = 243$ scenarios (see Table 16 for the demand data).

Table 16: Example 3: Time-period Demand Distributions

Time-period	Probability	Demands (P1, P2, P3)
1	0.50	(150, 100, 100)
	0.15	(250, 150, 250)
	0.35	(600, 350, 300)
2	0.15	(200, 150, 150)
	0.50	(300, 200, 300)
	0.35	(500, 400, 400)
3	0.30	(200, 200, 200)
	0.50	(250, 300, 250)
	0.20	(300, 400, 350)
4	0.20	(100, 100, 100)
	0.50	(150, 150, 150)
	0.30	(200, 200, 200)
5	0.40	(50, 50, 0)
	0.50	(100, 100, 100)
	0.10	(300, 150, 150)

Table 17 summarizes the key model statistics and computational results for this example. Given that there are 10 tasks and 3 processing modes for compatible unit/task combinations and the length of the time horizon (60 units), even the deterministic model requires a significant number of binary variables (1739) and about 4500 continuous variables. The two-stage model for the entire horizon requires more continuous variables and equations, but the same number of binary variables. The deterministic and two-stage models were solved to within 0.5% optimality in reasonable time (a few CPU minutes). The six-stage model requires over 38,600 binary variables and over 100,00 equations and variables; thus approximation strategies are especially relevant for this problem. Although we were not able to solve the six-stage model to optimality or obtain a feasible solution even after 50,000 CPU seconds of computation, the best LP bound we were able to obtain had an expected profit of \$28,841 (we did not provide an initial solution to the solver for the six-stage model).

Table 17: Example 3 - Computational Results

Model Type	Equations	Variables	Binaries	CPU secs	E[Profit], \$
Deterministic	5263	4528	1739	50	25,844
Two-stage	9136	7675	1739	179	26,527
Six-stage	143,782	177,285	38,619	> 50,000	-

The sub-problems within the shrinking horizon approaches were all solved with a 2000 CPU sec time limit and a 0.5% optimality gap. In the SHD approach, all but one of the sub-problems terminated well within the 2000 CPU sec time limit with optimality gaps less than 0.5% (the sub-problem which exceeded the time limit was solved to within 1% of optimality). In contrast, in the SHT strategy all sub-problems were solved to within 0.5% gap, with the largest time required for a sub-problem being around 800 CPU seconds. As seen in Table 18 the SHT strategy provides an expected profit about 1.6% better than the SHD strategy (\$28,487 vs \$28,041) and better still, comes to within 1.2% of the LP relaxation of the six-stage model (\$28,841), requiring a fraction of the computing time. The computation of the WSS solutions required nearly 11,000 CPU seconds mainly due to Scenarios 228-243 for which the models on an average took nearly 1200 CPU seconds to solve.

7 Conclusions

We have addressed the problem of short-term scheduling of multiproduct batch plants represented by STNs under demand uncertainty. The STN under our consideration is characterized by the presence of different processing modes, which differ in the range of batch sizes

Table 18: Example 3 - Comparison of Approximation Strategies

Strategy	Profit (\$)			
	Minimum	Maximum	Expected Value	CPU secs
SHD	17,106	37,002	28,041	3130
SHT	16,772	37,733	28,497	3110
Six-stage	-	-	-	> 50,000
WSS	17,782	41,849	29,434	37,840

that they can accommodate and their processing times, while the scheduling horizon is made up of several time-periods during which the product demands are placed.

With the evolution of product demand over the horizon, a multistage stochastic programming approach to scheduling, wherein some scheduling decisions are taken as recourse to the realization of some demand, is found to be the optimal strategy for maximizing expected profit. To overcome the large computational expense associated with the solution of the multistage stochastic program, we have proposed an approximation strategy that consists of solving a series of two-stage stochastic programs within a shrinking-horizon framework. The approximation strategy is shown from numerical examples to provide an expected profit that is within a few percent of the optimal multistage stochastic solution, and with order magnitude reduction in computation time for the larger examples. In comparison with similar deterministic approaches or the commonly-used two-stage programming approach, our approach is shown to provide consistently superior performance in terms of solution quality without significant increase in computational effort.

Acknowledgement

The authors gratefully acknowledge financial support from the National Science Foundation under Grant CTS-9810182.

Nomenclature

Indices

t	Time points
m	Time-periods
i	Tasks
l	Processing Modes
j	Units
s	States
k_m	Events that could occur in time-period m
k_m^*	Combination of events that could occur before time-period m

Sets

M	Time-periods
T	Time points within the horizon
T_m	Time points within a time-period

I	Tasks
L	Production Modes
J	Units
S	States
SP	States that are final products
I_s	Tasks that produce or consume state s
I_j	Tasks that can be processed in unit j
J_i	Units capable of processing task i
K_m	Possible events in time-period m
K_m^*	Combinations of possible events before time-period m

Parameters

H_m	Final time point in time-period m
$\rho_{i,s}^{in}$	Proportion of state s consumed by task i
$\rho_{i,s}^{out}$	Proportion of state s produced by task i
$B_{i,j,l}$	Batch size of task i in unit j for production mode l
$\tau_{i,j,l}$	Processing time of task i in unit j for production mode l
V_j	Capacity of unit j
CAP_s	Storage capacity for state s
$START_s$	Initial availability of state s
REV_s	Revenue for state s
XC_s, LC_s, HC_s	Excess-, lost- and inventory cost for state s
D_{s,m,k_m}	Demand for state s in event k_m in time-period m
p_{m,k_m}	Probability of event k_m in time-period m
$D_{s,k_1,\ldots,k_{\|M\|}}^{\|M\|+1}$	Demand for state s in scenario $(k_1, \ldots, k_{\|M\|})$

Variables

Stage m

$y_{i,j,t,k_m^*,l}^m$	Binary denoting start of task i in unit j in mode l at time t
$st_{s,k_m^*,t}^m$	Amount of state s at time t
$b_{i,j,k_m^*,t,l}^m$	Amount started in task i in unit j in mode l at time t

Stage $|M|+1$

xs_s, $sold_s$, $lost_s$	Amount of state s in excess, sold and lost respectively
$sales$	Total revenue from sales
thc, tlc, $txic$	Total inventory, lost- and excess-costs respectively
$profit$	Profit

Appendix A Generic Two-Stage Model

Here we present the generic two-stage model $TSM(kk^*_{m'})$ $\forall(m' \in M \cup |M|+1)$ used in Chapter 4. This two-stage model represents the approximation of the multistage stochastic program at any node of the scenario tree.

Recall from the earlier discussion in Section 4 that $kk^*_{m'}$ is an index used to represent the nodes of the scenario tree. At the node $kk^*_{m'}$, the events $KK_1, \ldots, KK_{m'-1}$ have already been realized, which implies that the corresponding demands $D_{s,1,KK_1}, \ldots, D_{s'_m,KK_{m'-1}}$ have been placed. Thus, these demands are guaranteed and the two-stage model to be solved at the node should take these into account for the remaining scenarios. However the demands to be placed in the future time-periods, i.e. $m \geq m'$ are still uncertain. This distinction between demands already-placed and uncertain demands is reflected in constraints (24), which is substituted into constraints (28)-(30). Note that in $TSM(kk^*_{m'})$, we have dropped the subscripts k^*_m from equations (2)-(8) since all scheduling decisions are considered to be first-stage. Furthermore, variables st, y, b are fixed to values $PRED(kk^*_{m'})$ up to the beginning of the current time-period as in (31), since these decisions have already been made.

$$D^{|M|+1}_{s,k_1\ldots k_{|M|}} = \sum_{m<m'} D^m_{s,KK_m} + \sum_{m\geq m'} D^m_{s,k_m} \qquad \forall(s \in SP, k_1 \in K_1, \ldots, k_{|M|} \in K_{|M|}) \tag{24}$$

As an illustration of constraint (24), consider the three time-period four-stage model depicted in Figure 6. In the two-stage model at the root node, the demand parameter D^4_{s,k_1,k_2,k_3} is defined as follows:

$$D^4_{s,k_1,k_2,k_3} = D_{s,1,k_1} + D_{s,2,k_2} + D_{s,3,k_3} \tag{25}$$

When solving the two-stage model at the left node in level 1 of the scenario tree in Figure 6, where $D_{s,1}$ has already taken on the first value of its distribution, the demand parameters are suitably modified as in (24) so as to reflect the realized value of $D_{s,1}$. Thus, even though the subscripts (k_1, k_2, k_3) indicate eight scenarios, there are in effect only four, when we consider the probabilities and demands.

$$D^4_{s,k_1,k_2,k_3} = D_{s,1,1} + D_{s,2,k_2} + D_{s,3,k_3} \tag{26}$$

The generic two-stage model is as shown below:

$$(TSM(kk^*_{m'})) \quad \max EP$$

$$\text{s.t.} \quad (2) - (8) \qquad \text{without } k^{*} \text{ subscripts}$$

$$sold^{|M|+1}_{s,k_1,\ldots,k_{|M|}} \leq st^{|M|}_{s,H} \qquad \forall (s \in SP, k_1 \in K_1, \ldots, k_{|M|} \in K_{|M|}) \tag{27}$$

$$sold^{|M|+1}_{s,k_1,\ldots,k_{|M|}} \leq D^{|M|+1}_{s,k_1 \ldots k_{|M|}} \qquad \forall (s \in SP, k_1 \in K_1, \ldots, k_{|M|} \in K_{|M|}) \tag{28}$$

$$xs^{|M|+1}_{s,k_1,\ldots,k_{|M|}} \geq st^{|M|}_{s,H} - D^{|M|+1}_{s,k_1 \ldots k_{|M|}} \qquad \forall (s \in SP, k_1 \in K_1, \ldots, k_{|M|} \in K_{|M|}) \tag{29}$$

$$lost^{|M|+1}_{s,k_1,\ldots,k_{|M|}} \geq D^{|M|+1}_{s,k_1 \ldots k_{|M|}} - st^{|M|}_{s,H} \qquad \forall (s \in SP, k_1 \in K_1, \ldots, k_{|M|} \in K_{|M|}) \tag{30}$$

$$\begin{aligned} y^{m}_{i,j,t',l} &= 0 \qquad \forall (m \in M, i \in I, j \in J_i, l \in L, t' | t' + \tau_{ijl} > H) \\ y^{m}_{i,j,t',l}, b^{m}_{i,j,t',l}, st^{m}_{s,t} \ \text{ fixed to } \ & PRED(kk^{*}_{m'-1}) \qquad \forall m < m' \end{aligned} \tag{31}$$

$$y^{m}_{i,j,t,l} \in \{0,1\} \qquad \forall (m \in M, i \in I, j \in J_i, t \in T_m, l \in L)$$

$$b^{m}_{i,j,t,l} \geq 0 \qquad \forall (m \in M, i \in I, j \in J_i, t \in T_m, l \in L)$$

$$0 \leq st^{m}_{s,t} \leq CAP_s \qquad \forall (m \in M, s \in S, t \in T_m)$$

$$sold^{|M|+1}_{s,k_1,\ldots,k_{|M|}}, xs^{|M|+1}_{s,k_1,\ldots,k_{|M|}}, lost^{|M|+1}_{s,k_1,\ldots,k_{|M|}} \geq 0 \qquad \forall (s \in SP, k_1 \in K_1, \ldots, k_{|M|} \in K_{|M|})$$

$$sales^{|M|+1}_{k_1,\ldots,k_{|M|}}, thc^{|M|+1}_{k_1,\ldots,k_{|M|}} \geq 0 \qquad \forall (k_1 \in K_1, \ldots, k_{|M|} \in K_{|M|})$$

$$txic^{|M|+1}_{k_1,\ldots,k_{|M|}}, tlc^{|M|+1}_{k_1,\ldots,k_{|M|}} \geq 0 \qquad \forall (k_1 \in K_1, \ldots, k_{|M|} \in K_{|M|})$$

Appendix B Data for Example 1

Table 20 displays the task data and unit compatibilities which are common for Examples 1a and 1b. The state data are different for Examples 1a and 1b, however: please use Table 21 for Example 1a and Table 21 for Example 1b.

Table 20: Example 1 - Task Data

Task	Suitable Unit	Batch Size Range (units)	Processing Time (units)
Mix	Unit1	10-40	3
		40-80	4
		80-120	5
React	Unit2	10-25	2
		25-50	3
		50-75	4
Dry	Unit3	10-20	1
		20-40	2
		40-60	3

Table 21: Example 1a - State Data

State	CAP_s (tons)	REV_s (\$/tons)	HC_s (\$/tons/unit)	XC_s (\$/tons)	LC_s (\$/tons)
S1	Unlimited	0	0	0	0
S2	100	0	0	100	0
S3	100	0	0	200	0
S4	Unlimited	1000	50	400	500

Table 22: Example 1b - State Data

State	CAP_s (tons)	REV_s (\$/tons)	HC_s (\$/tons/unit)	XC_s (\$/tons)	LC_s (\$/tons)
S1	Unlimited	0	0	0	0
S2	500	0	5	100	0
S3	500	0	10	200	0
S4	Unlimited	2000	15	400	100

References

(1) Shah, N. Single- and multisite planning and scheduling: current status and future challenges. In Proceedings of the Conference on Foundations of Computer Aided Process Operations FOCAPO98, Snowbird, USA, **1998**, 75.

(2) Pinto, J. M.; Grossmann, I.E. Assignment and sequencing models for the scheduling of process systems. *Annals of Operations Research*, **1998**, *81*, 433.

(3) Sahinidis, N.V. Optimization under uncertainty: state-of-the-art and opportunities. In Proceedings of the Conference on Foundations of Computer Aided Process Operations FOCAPO 2003, Coral Springs, USA, **2003**, 153.

(4) Wellons, H.S.; Reklaitis, G.V. The design of multiproduct batch plants under uncertainty with staged expansion. *Computers and Chemical Engineering*, **1989**, *13*, 115.

(5) Straub, D.A.; Grossmann, I.E. Evaluation and optimization of stochastic flexibility in multiproduct batch plants. *Computers and Chemical Engineering*, **1992**, *16*, 69.

(6) Ierapetritou, M.G.; Pistikopolous, E.N. Batch plant design and operations under demand uncertainty. *Industrial Engineering and Chemistry Research*, **1996**, *35*, 772.

(7) Harding, S.T.; Floudas, C.A. Global optimization in multiproduct and multipurpose batch design under uncertainty. *Industrial Engineering and Chemistry Research*, **(1997)**, *36*, 1644.

(8) Petkov, S.V.; Maranas, C.D. Design of single-product campaign batch plants under demand uncertainty. *AIChE Journal*, **1998**, *44*, 896.

(9) Cao, D-M.; Yuan, X.-G. Optimal design of batch plants with uncertain demands considering switch over of operating modes of parallel units. *Industrial Engineering and Chemistry Research*, **2002**, *41*, 4616.

(10) Sand, G.; Engell, A.; Märkert, A.; Schultz, R.; Schulz, C. Approximation of an ideal online scheduler for a multiproduct batch plant. *Computers and Chemical Engineering*, **2000**, *24*, 361.

(11) Gupta, A.; Maranas, C.D.; McDonald, C.M. Midterm supply chain planning under demand uncertainty: customer demand satisfaction and inventory management. *Computers and Chemical Engineering*, **2000**, *24*, 2613.

(12) Vin, J.P.; Ierapetritou, M.G. Robust short-term scheduling of multiproduct batch plants under demand uncertainty. *Industrial Engineering and Chemistry Research*, **2001**, *40*, 4543.

(13) Engell, S.; Märkert, A.; Sand, G.; Schultz, R. Scheduling of a multiproduct batch plant by two-stage stochastic integer programming. *Preprint 520-2002*, Institut für Mathematik, Gerhard-Mercator-Universität Duisburg, 2002.

(14) Sox, C.R.; Jackson, P.L.; Bowman, A.; Muckstadt, J.A. A review of the stochastic lot scheduling problem. *International Journal of Production Economics*, **1999**, *62*, 181.

(15) Cott, B.J.; Macchietto, S. Minimizing the Effects of Batch Process Variability using On-Line Schedule Modification. *Computers and Chemical Engineering*, **1989**, *13, 1/2*, 105.

(16) Kanakamedala, K.B; Reklaitis, G.V.; Venkatasubramanian, V. Reactive Schedule modification in Multipurpose Batch Chemical Plants *Industrial Engineering and Chemistry Research*, **1994**, *33*, 77.

(17) Jain, A.K.; Elmaraghy, A. Reactive schedule modification in multipurpose batch chemical plants. *International Journal of Production Research*, **1997**, *35*, 281.

(18) Vin, J.P.; Ierapetritou, M.G. A new approach for efficient rescheduling of multiproduct batch plants. *Industrial Engineering and Chemistry Research*, **2000**, *39*, 4228.

(19) Mendez, C.; Cerda, J.C. An MILP framework for reactive scheduling of resource-constrained multistage batch facilities. In Proceedings of the Conference on Foundations of Computer Aided Process Operations FOCAPO 2003, Coral Springs, USA, **2003**, 335.

(20) Kondili, E.; Pantelides, C.C.; Sargent, R.W.H. A general algorithm for short-term scheduling of batch operations-I. MILP formulation. *Computers and Chemical Engineering*, **1993**, *17*, 211.

(21) Shah, N.; Pantelides, C.C.; Sargent, R.W.H. A general algorithm for short-term scheduling of batch operations. II. Computational issues. *Computers and Chemical Engineering*, **1993**, *17(2)*, 229.

(22) Kall, P.; Wallace, S.W. *Stochastic Programming*, John Wiley and Sons, Chichester, England, 1994.

(23) Birge, J. R.; Louveaux, F. *Introduction to stochastic programming*, Springer, New York, NY, 1997.

(24) Römisch, W.; Schultz, R. Multistage stochastic integer programming: an introduction. In *Online Optimization of Large Scale Systems*; Grötschel, M., Krumke, S.O., Rambau, J., Eds.; Springer-Verlag: Berlin, 2001, 581.

(25) Ahmed, S.; Tawarmalani, M.; Sahinidis, N.V. A finite branch and bound algorithm for two-stage stochastic integer programming. Submitted for publication in *Mathematical Programming.*

(26) Wilkinson, S.J. Aggregate formulations for large-scale process scheduling problems. Ph.D. Dissertation, University of London, London, 1996.

(27) Dimitriadis, A.D.; Shah, N.; Pantelides, C.C. RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. *Computers and Chemical Engineering*, **1997**, *21*, S1061.

(28) Elkamel, A.; Mohindra, A. A rolling horizon heuristic for reactive scheduling of batch process operations. *Engineering Optimization*, **1999**, *31*, 763.

(29) Pinedo, M. *Scheduling: Theory, Algorithms and Systems, Second Edition*, Prentice Hall: Upper Saddle River, NJ, 2002, p 360.

(30) Perea-Lopez, E.; Grossmann, I.E.; Ydstie B.E.; Tahmassebi T. Rolling horizon scheduling and dynamic control for supply chain management. *Computers and Chemical Engineering*, in press.

(31) ILOG CPLEX 7.5 Users Manual. *ILOG Inc*, 2001.

(32) Fisher, M.L. The Lagrangean relaxation method for solving integer programming problems. *Management Science*, **1981**, *27*, 1.

(33) Guignard, M.; Kim, S. Lagrangean decomposition: a model yielding stronger Lagrangean bounds. *Mathematical Programming*, **1987**, *39*, 215.

(34) Ierapetritou, M.G.; Floudas, C.A. Effective continuous-time formulation for short-term scheduling. I. Multipurpose batch processes. *Industrial Engineering and Chemistry Research*, **1998**, *37*, 4341.

(35) Papageorgiou, L.G.; Pantelides, C.C. Optimal Campaign Planning/Scheduling of Multipurpose Batch/Semicontinuous Plants. 2. Mathematical Decomposition Approach. *Industrial Engineering and Chemistry Research* , **1996**, *35*, 510.

(36) Balasubramanian, J. Optimization Models and Algorithms for Batch Process Scheduling under Uncertainty. Ph.D. Dissertation, Carnegie Mellon University, Pittsburgh, PA, **2003, pp 198-202.**