The value of multi-stage stochastic programming in capacity planning under uncertainty

Kai Huang and Shabbir Ahmed[*]

School of Industrial & Systems Engineering

Georgia Institute of Technology

April 26, 2005

Abstract

This paper addresses a general class of capacity planning problems under uncertainty, which arises, for example, in semiconductor tool purchase planning. Using a scenario tree to model the evolution of the uncertainties, we develop a multi-stage stochastic integer programming formulation for the problem. In contrast to earlier two-stage approaches, the multi-stage model allows for revision of the capacity expansion plan as more information regarding the uncertainties is revealed. We provide analytical bounds for the *value of multi-stage stochastic programming* (VMS) afforded over the two-stage approach. By exploiting a special lot-sizing substructure inherent in the problem, we develop an efficient approximation scheme for the difficult multi-stage stochastic integer program and prove that the proposed scheme is asymptotically optimal. Computational experiments with realistic-scale problem instances suggest that the VMS for this class of problems is quite high. Moreover the quality and performance of the approximation scheme is very satisfactory. Fortunately, this is more so for instances for which the VMS is high.

Key words: multi-stage stochastic programming, capacity planning, semiconductor tool planning, stochastic lot-sizing, analysis of algorithms.

Subject classification: Facility/equipment planning: capacity planning. Production/ scheduling: planning. Programming: stochastic programming.

Area of review: Optimization.

[*]Corresponding author. E-mail: sahmed@isye.gatech.edu

1 Introduction

Capacity planning, i.e., deciding the optimal timing and level of capacity acquisition and allocation, plays a crucial role in strategic level planning in a wide array of applications. This activity involves substantial commitment of capital resources and is marred by uncertainties in the long range forecasts, thereby making the associated decision problems very complex. For example, the initial investment in building a semiconductor wafer fab is close to two billion dollars, and every year the procurement of new tools to accommodate the high volatility in demand, product mix and technology could cost several million dollars (cf. Barahona et al. (2005), Hood et al. (2003), Swaminathan (2000)).

Owing to the inherent complexities, quantitative models for economic capacity planning under uncertainty have been the subject of intense research since the early 1960s (cf. Luss (1982)). Early approaches for solving stochastic capacity expansion problems are restricted to a single resource and based on simplifying assumptions on the underlying stochastic processes to render analytical tractability (cf. Bean et al. (1992), David et al. (1987), Freidenfelds (1980), Manne (1961)). More general stochastic programming based approaches that use scenarios to model the uncertain parameters within large-scale mathematical programs for multi-resource multi-item capacity planning have since been proposed (cf. Berman et al. (1994), Eppen et al. (1989), Fine and Freund (1990)). Most of these stochastic programming approaches are based on the two-stage paradigm, wherein the capacity acquisition schedule for the entire (multi-period) planning horizon is decided "here-and-now," and capacity allocations are made on a period-by-period basis based on realized uncertainties and acquired capacities. In the context of semiconductor tool planning, such two-stage models are investigated in Barahona et al. (2005), Hood et al. (2003), Karabuk and Wu (2003) and Swaminathan (2000). Multi-stage stochastic programming models extend the two-stage paradigm by allowing revised decisions in each time stage based upon the uncertainty realized so far (cf. Birge (1985)). A multi-stage stochastic capacity planning model involving continuous capacity allocation decisions and fixed charge expansion costs is considered in Ahmed and Sahinidis (2003). The authors develop a LP-relaxation based heuristic for this problem and proved, via a probabilistic analysis, that the heuristic is asymptotically optimal in the number of planning stages.

Motivated by applications in semiconductor tool planning, we address a general multi-stage stochastic capacity planning model involving discrete capacity acquisition decisions. Our model generalizes earlier two-stage approaches considered in Barahona et al. (2005), Hood et al. (2003), Swaminathan (2000), by allowing for revision of the capacity expansion plan as more information regarding the uncertainties is revealed. We provide analytical bounds for the *value of multi-stage stochastic programming* afforded over two-stage approaches. By exploiting a special lot-sizing substructure inherent in the problem, we develop an efficient approximation scheme for the multi-stage problem and prove that the proposed scheme is asymptotically optimal. Our asymptotic analysis is significantly different from that of Ahmed and Sahinidis (2003), since we consider discrete capacity acquisition levels and do not make any assumptions regarding the distributions of the underlying stochastic parameters. Finally, we present numerical results for a realistic-scale semiconductor tool planning problem to demonstrate the advantage of the proposed model and solution method.

2 Model development

In this section we present a mathematical formulation for the stochastic capacity planning under consideration. We first describe a specific deterministic capacity planning formulation related to semiconductor tool planning, and then discuss deterministic and stochastic generalizations of this model.

2.1 A deterministic model for semiconductor tool planning

Consider a wafer fab consisting of M tool types, that can process N types of wafers. Each product (wafer type) goes through a subset of K processing steps, each of which can be performed on one or more tool types. The products are measured in units of "wafer start." Let h_{ijk} denote the time (in hours per wafer start) required by processing step k $(1, \ldots, K)$ on wafer type j $(1, \ldots, N)$ on tool type i $(1, \ldots, M)$. We set $h_{ijk} = 0$ if step k is not needed for wafer type j, and $h_{ijk} = \infty$ if step k is required for wafer type j but cannot be performed on tool type i. Consider now a planning horizon of T periods. Let us use variables x_{it}, u_{jt},

v_{ijkt}, and w_{jt}, to denote the number of tools of type i purchased in period t $(1, \ldots, T)$, the shortage (in wafer starts) of wafer type j in period t, the number of wafer starts of type j whose k-th processing step is allotted to tool type i in period t, and the production of wafer type j (in wafer starts) in period t, respectively. In addition to h_{ijk}, let us also consider the problem parameters a_{it}, b_{jt}, c_i, and d_{jt} corresponding to the (discounted) cost of tool type i in period t, the penalty cost of unit shortage in wafer type j in period t, the per-period capacity (in hours) of one tool of type i, and the per-period demand (in wafer starts) of wafer type j in period t, respectively.

With the above notation, an optimization model for multi-period (deterministic) scheduling of tool purchases and the allocation of tool capacity to production so as to minimize total tool purchase costs and shortage penalties can be stated as follows:

$$\begin{array}{lll}
\min & \sum_{t=1}^{T} \Big(\sum_{i=1}^{M} a_{it} x_{it} + \sum_{j=1}^{N} b_{jt} u_{jt} \Big) & \\
\text{s.t.} & \sum_{j=1}^{N} \sum_{k=1}^{K} h_{ijk} v_{ijkt} \leq c_i \Big(\sum_{\tau=1}^{t} x_{i\tau} \Big) & \forall\ i,\ t \\
& \sum_{i=1}^{M} v_{ijkt} \geq w_{jt} & \forall\ j,\ k,\ t \\
& w_{jt} + u_{jt} \geq d_{jt} & \forall\ j,\ t \\
& u_{jt},\ v_{ijkt},\ w_{jt} \in \mathbb{R}_+ & \forall\ i,\ j,\ k,\ t \\
& x_{it} \in \mathbb{Z}_+ & \forall\ i,\ t.
\end{array} \tag{1}$$

For any period t, the first constraint in model (1) assures that the total processing requirement (in hours) allocated to tool i cannot exceed the installed capacity; the second constraint enforces that the actual production of wafer type j is equal to the number of wafer starts that has completed all of the required K processing steps; the third constraint enforces that the production and shortage together should exceed the demand; the fourth constraint enforces non-negativity of the production-allocation-shortage variables; and the fifth constraint enforces the integrality of the tool purchase decisions. Model (1) is a multi-period extension of the tool planning model described by Swaminathan (2000).

2.2 A generic capacity planning model

The semiconductor tool planning model (1) is a special case of the following generic capacity acquisition and allocation model:

$$
\begin{array}{lll}
\min & \sum_{t=1}^{T} \Big(\alpha_t x_t + \beta_t y_t\Big) & \\
\text{s.t.} & A_t y_t \leq \sum_{\tau=1}^{t} x_\tau & \forall\, t \\
 & B_t y_t \geq \delta_t & \forall\, t \\
 & y_t \in \mathbb{R}_+^J,\ x_t \in \mathbb{Z}_+^I & \forall\, t.
\end{array}
\qquad (2)
$$

In (2), the $I \times 1$ vector of variables x_t represent the capacity acquisition decisions for a set of I resources and the $J \times 1$ vector y_t represents the operational level allocation of capacity to a set of J tasks, in period t. The parameters α_t, β_t and δ_t represent acquisition costs, allocation costs, and demands, respectively. The matrices A_t and B_t represent resource-task utilization coefficients.

To see the connection between models (1) and (2), note that x_t and y_t correspond to the tool purchase and production-allocation-shortage decision vectors for period t, respectively; α_t, β_t, and δ_t correspond to the tool cost, shortage penalty, and demand vectors for period t, respectively; the matrix A_t corresponds to the coefficients of the first set of constraint in (1), and the matrix B_t corresponds to the coefficients of the second and third set of constraint in (1).

2.3 Stochastic programming extensions

Let us now extend the deterministic capacity planning model (2) to a stochastic setting. We assume that the uncertain problem parameters $(\alpha_t, \beta_t, \delta_t, A_t, B_t)$ evolve as discrete time stochastic processes with a finite support. This information structure can be interpreted as a scenario tree where the nodes in stage (or level) t of the tree constitute the states of the world that can be distinguished by information available up to time stage t. There are in total T stages in the tree. Each node n of the scenario tree, except the root $(n = 1)$, has a unique parent $a(n)$, and each non-terminal node n is the root of a sub-tree $\mathcal{T}(n)$. The probability associated with the state of the world in node n is p_n. The set $\mathcal{S}_t$ denotes the nodes corresponding to time stage t, and t_n is the time stage corresponding to node n. The path from the root node to a node n will be denoted by $\mathcal{P}(n)$. If n is a terminal (leaf) node, i.e., $n \in \mathcal{S}_T$, then $\mathcal{P}(n)$ corresponds to a *scenario*, and represents a joint realization of the problem parameters over all periods. There are S leaf nodes corresponding to S scenarios, i.e., $S = |\mathcal{S}_T|$. We denote the whole tree $\mathcal{T}(1)$ by $\mathcal{T}$ and let N_T be the number of nodes in this tree. The stochastic problem parameters are then given by the sequence $\{\alpha_n, \beta_n, \delta_n, A_n, B_n\}_{n \in \mathcal{T}}$.

Let us first consider a two-stage model where the first-stage involves deciding the capacity acquisition plan for all periods, regardless of the state of the world, and the second-stage consists of deciding on the capacity allocation plan subject to available capacity and the realized state. Thus the capacity acquisition variables are only indexed by time periods (since these do not change with the realized state) while the allocation decisions are indexed by the nodes of the scenario tree. With an objective of minimizing the *expected* total costs, a two-stage stochastic programming extension of (2) is as follows:

$$
\begin{array}{lll}
\min & \sum_{t=1}^{T} \overline{\alpha}_t x_t + \sum_{n \in \mathcal{T}} p_n \beta_n y_n & \\
\text{s.t.} & A_n y_n \leq \sum_{\tau=1}^{t_n} x_\tau & \forall\, n \in \mathcal{T} \\
& B_n y_n \geq \delta_n & \forall\, n \in \mathcal{T} \\
& y_n \in \mathbb{R}_+^J & \forall\, n \in \mathcal{T} \\
& x_t \in \mathbb{Z}_+^I & \forall\, t,
\end{array}
\tag{3}
$$

where $\overline{\alpha}_t = \sum_{n \in \mathcal{S}_t} p_n \alpha_n$, i.e., the average capacity acquisition cost in period t. The stochastic programming model considered in Swaminathan (2000) is a special case of the model (3) when the number of periods $T = 2$. The models presented in Barahona et al. (2005) and Hood et al. (2003) are similar to (3), however, there, the uncertain parameters are defined over scenarios (paths in the scenario tree) rather than nodes of the scenario tree.

As mentioned earlier, the two-stage model (3) does not allow any flexibility in the capacity acquisition plan with respect to the realized state of the world. To formulate a multi-stage stochastic programming model, we need to have the capacity acquisition decisions to be dependent on the realized state, and hence the resulting model is as follows:

$$\begin{array}{lll} \min & \sum_{n\in\mathcal{T}} p_n\Big(\alpha_n x_n + \beta_n y_n\Big) & \\ \text{s.t.} & A_n y_n \le \sum_{m\in\mathcal{P}(n)} x_m & \forall\ n\in\mathcal{T} \\ & B_n y_n \ge \delta_n & \forall\ n\in\mathcal{T} \\ & y_n \in \mathbb{R}^J_+ & \forall\ n\in\mathcal{T} \\ & x_n \in \mathbb{Z}^I_+ & \forall\ n\in\mathcal{T}. \end{array} \tag{4}$$

2.4 Stochastic lot-sizing substructure

The multi-stage capacity planning problem (4) can be restated as follows:

$$\begin{array}{lll} \min & \sum_{n\in\mathcal{T}} p_n\beta_n y_n + \sum_{i=1}^{I} Q_i(y) & \\ \text{s.t.} & B_n y_n \ge \delta_n & \forall\ n\in\mathcal{T} \\ & y_n \in \mathbb{R}^J_+ & \forall\ n\in\mathcal{T}, \end{array} \tag{5}$$

where

$$\begin{array}{llll} Q_i(y) = & \min & \sum_{n\in\mathcal{T}} p_n\alpha_{in}x_{in} & \\ & \text{s.t.} & \sum_{m\in\mathcal{P}(n)} x_{im} \ge [A_n y_n]_i & \forall\ n\in\mathcal{T} \\ & & x_{in}\in\mathbb{Z}_+ & \forall\ n\in\mathcal{T}. \end{array} \tag{6}$$

The above reformulation decomposes the problem into two separate problems, one (5) involving the capacity allocation decisions, and the other (6) involving the capacity acquisition decisions. Note that we have used y to collectively denote the capacity allocation sequence $\{y_n\}_{n\in\mathcal{T}}$, and $[A_n y_n]_i$ to denote the i-th component of the I dimensional vector $A_n y_n$.

Observe that for a fixed sequence of capacity allocation decisions, the optimal capacity acquisition decisions can be obtained via solving (6) independently for each resource i. Problem (6) is equivalent to the following single-item, uncapacitated, stochastic lot-sizing problem with linear costs:

$$\begin{array}{lll} \min & \sum_{n\in\mathcal{T}} p_n\Big[c_n x_n + h_n s_n\Big] & \\ \text{s.t.} & s_{a(n)} + x_n = d'_n + s_n & \forall\ n\in\mathcal{T} \\ & x_n\in\mathbb{Z}_+,\ s_n\in\mathbb{R}_+ & \forall\ n\in\mathcal{T}. \end{array} \tag{7}$$

Above, x_n and s_n represent the production and ending inventory in node n, respectively; and the parameters c_n, h_n and d'_n represent the production cost, holding cost, and demand in node n, respectively. The objective is to minimize expected production and holding costs, subject to a inventory balance constraint for each n.

To see the equivalence of (6) and (7), note that we can eliminate the inventory variables s_n by substituting $s_n = \sum_{m \in \mathcal{P}(n)} x_m - d_n$, and obtain the following reformulation of (7):

$$\begin{array}{lll} \min & \sum_{n \in \mathcal{T}} p_n \alpha_n x_n - C & \\ \text{s.t.} & \sum_{m \in \mathcal{P}(n)} x_m \geq d_n & \forall\, n \in \mathcal{T} \\ & x_n \in \mathbb{Z}_+ & \forall\, n \in \mathcal{T}, \end{array} \tag{8}$$

where $\alpha_n = c_n + (1/p_n) \sum_{m \in \mathcal{T}(n)} p_m h_n$, $d_n = \sum_{m \in \mathcal{P}(n)} d'_m$, and $C = \sum_{n \in \mathcal{T}} p_n h_n d'_n$.

Key to the further developments in this paper is the study of (8). Note that this problem is a stochastic version of the classical dynamic deterministic lot-sizing problem which can be solved using a simple greedy algorithm (cf. Johnson (1957), Zipkin (2000)). The first observation is that even though (8) is a multi-stage stochastic integer program, the following advantageous property holds.

THEOREM 1 *With integer demand parameters $\{d_n\}_{n \in \mathcal{T}}$, the LP relaxation of (8) yields integral solutions.*

PROOF. Note that the constraint matrix of (8) is a $N_T \times N_T$ 0-1 matrix. Let us denote this matrix as $U = [u_{ij}]$, where $u_{ij} = 1$ if $j \in \mathcal{P}(i)$ in the scenario tree $\mathcal{T}$, and $u_{ij} = 0$ otherwise. Let $\mathcal{J}$ be any subset of the columns of U, i.e., a subset of the nodes in $\mathcal{T}$. Let $\{t_1, \ldots, t_K\}$ be the indices of time stages corresponding to the nodes in $\mathcal{J}$, and suppose that $t_1 < t_2 < \cdots < t_K$. Let $\overline{\mathcal{S}}_t = \mathcal{S}_t \cap \mathcal{J}$, i.e. the set of nodes in time stage t included in $\mathcal{J}$. We can then create a partitioning of the nodes (columns) in $\mathcal{J}$ as follows $\mathcal{J}_1 = \cup_{i=1,\ i \text{ is odd}}^{K} \overline{\mathcal{S}}_{t_i}$ and $\mathcal{J}_2 = \cup_{i=1,\ i \text{ is even}}^{K} \overline{\mathcal{S}}_{t_i}$. It is immediately verified that

$$\mid \sum_{j \in \mathcal{J}_1} u_{ij} - \sum_{j \in \mathcal{J}_2} u_{ij} \mid \leq 1 \ \forall\ i = 1, \ldots, N_T.$$

Thus for any subset of the columns in U, we can create a bi-partition such that the difference in the sum of coefficients of each partition along every row of U is at most 1. It then follows (cf. Nemhauser and Wolsey (1988)) that U is totally unimodular, and the claim holds. □

3 Value of multi-stage stochastic programming

Let the v^{TS} and v^{MS} denote the optimal objective function values of the two-stage (3) and multi-stage (4) models, respectively. For a given set of problem parameters, it is easily verified that any solution to (3) is feasible to (4), and the objective function values corresponding to this solution are equal in both problems, thus

$$v^{TS} \geq v^{MS}.$$

That is, the overall cost of the multi-stage solution is smaller than that of the two-stage solution. This should come as no surprise, since, the multi-stage solution offers more flexibility in the capacity acquisition decisions with respect to the uncertain states of the world. We refer to the difference between the optimal objective values of the two-stage and multi-stage formulations as the *value of multi-stage stochastic programming* (VMS):

$$\text{VMS} = v^{TS} - v^{MS}.$$

Unfortunately, the value of multi-stage stochastic programming comes at the expense of solving a much larger and difficult optimization model. Both (3) and (4) are stochastic integer programs, and in general, can be extremely difficult to solve. For our particular case, both models have the property that by fixing the capacity acquisition decisions (the x variables), we can break the problem down to independent capacity allocation problems (in the y variables) corresponding to each node of the scenario tree. Owing to this structure, Benders decomposition (cf. Benders (1962)) is particularly attractive for these problems. In case of (3) and (4), this would require us to solve master problems involving the integer variables x. While the two-stage model (3) involves $I \times T$ integer variables, the multi-stage model (4) involves $I \times N_T$ integer variables (recall that $N_T = |\mathcal{T}|$), and for any non-trivial scenario tree $N_T >> T$. Consequently the computational difficulty of (4) is significantly more than that of (3). If the VMS is small, then this additional computational effort may not be worthwhile. However, we need *a priori* estimates of VMS to analyze this tradeoff. Next, we first describe simple bounds on VMS for the stochastic lot-sizing problem (7) and then use these to get bounds on the VMS for the capacity planning problem (4).

3.1 VMS for the stochastic lot-sizing problem

Consider the linear relaxation of the multi-stage stochastic lot-sizing problem (8) and let v^M denote its optimal objective function value, i.e.,

$$\begin{array}{rll} v^M = & \min & \sum_{n \in \mathcal{T}} p_n \alpha_n x_n \\ & \text{s.t.} & \sum_{m \in \mathcal{P}(n)} x_m \geq d_n \quad \forall\ n \in \mathcal{T} \\ & & x_n \in \mathbb{R}_+ \qquad\qquad \forall\ n \in \mathcal{T}. \end{array} \tag{9}$$

Note that we have dropped the constant term C from the objective. A two-stage model for the stochastic lot-sizing problem would require that the production decisions for each time period be the same irrespective of the state realized, i.e.,

$$\begin{array}{rll} v^T = & \min & \sum_{n \in \mathcal{T}} p_n \alpha_n x_n \\ & \text{s.t.} & \sum_{m \in \mathcal{P}(n)} x_m \geq d_n \quad \forall\ n \in \mathcal{T} \\ & & x_n \in \mathbb{R}_+ \qquad\qquad \forall\ n \in \mathcal{T} \\ & & x_m = x_n \qquad\qquad \forall\ m, n \in \mathcal{S}_t,\ \forall\ t. \end{array} \tag{10}$$

THEOREM 2 *Let*

$$\begin{array}{rcl} \alpha^* & = & \max_{n \in \mathcal{T}} \alpha_n \\ \alpha_* & = & \min_{n \in \mathcal{T}} \alpha_n \\ d_T^* & = & \max_{n \in \mathcal{S}_T} (\max_{m \in \mathcal{P}(n)} d_m) \\ \overline{d}_T & = & \sum_{n \in \mathcal{S}_T} p_n (\max_{m \in \mathcal{P}(n)} d_m), \end{array}$$

then

$$\alpha_* d_T^* - \alpha^* \overline{d}_T \leq \text{VMS} = v^T - v^M \leq \alpha^* d_T^* - \alpha_* \overline{d}_T.$$

PROOF. Note that any feasible solution x for (9) has to satisfy

$$\begin{array}{rl} & \sum_{m \in \mathcal{P}(n)} x_m \geq \max_{m \in \mathcal{P}(n)} d_m \quad \forall\ n \in \mathcal{S}_T \\ \Rightarrow & \sum_{n \in \mathcal{S}_T} p_n \Big(\sum_{m \in \mathcal{P}(n)} x_m \Big) \geq \sum_{n \in \mathcal{S}_T} p_n (\max_{m \in \mathcal{P}(n)} d_m) \\ \Leftrightarrow & \sum_{t=1}^{T} \sum_{n \in \mathcal{S}_t} \Big(\sum_{m \in \mathcal{S}_T \cap \mathcal{T}(n)} p_m \Big) x_n \geq \overline{d}_T \\ \Leftrightarrow & \sum_{n \in \mathcal{T}} p_n x_n \geq \overline{d}_T, \end{array}$$

where the last step follows from the fact that

$$\sum_{m \in \mathcal{S}_T \cap \mathcal{T}(n)} p_m = p_n \quad \forall\ n \in \mathcal{T}.$$

Then if x^* is an optimal solution for (9), we have

$$v^M = \sum_{n\in\mathcal{T}} p_n \alpha_n x_n^* \geq \alpha_* \sum_{n\in\mathcal{T}} p_n x_n^* \geq \alpha_* \bar{d}_T. \tag{11}$$

Next, consider a feasible solution $\widehat{x}$ to (9), such that $\widehat{x}_n = \max_{m\in\mathcal{P}(n)} d_m - \max_{m\in\mathcal{P}(a(n))} d_m$ for all $n \in \mathcal{T}$, and $\max_{m\in\mathcal{P}(a(1))} d_m = 0$. Then

$$\begin{array}{rcl} v^M & \leq & \sum_{n\in\mathcal{T}} p_n \alpha_n \widehat{x}_n \\ & \leq & \alpha^* \sum_{t=1}^{T} \sum_{n\in\mathcal{S}_t} p_n (\max_{m\in\mathcal{P}(n)} d_m - \max_{m\in\mathcal{P}(a(n))} d_m) \\ & = & \alpha^* \sum_{n\in\mathcal{S}_T} p_n (\max_{m\in\mathcal{P}(n)} d_m) \\ & = & \alpha^* \bar{d}_T, \end{array} \tag{12}$$

where the third step follows the fact that

$$\sum_{m\in\mathcal{S}_{t+1}\cap\mathcal{T}(n)} p_m = p_n \ \ \forall\ t,\ n \in \mathcal{S}_t.$$

In the two-stage model (10), since the production decision is identical for all nodes in any stage, it has to satisfy the largest possible cumulative demand in that stage, i.e., d_n can be replaced with $\tilde{d}_n = \max_{m\in\mathcal{S}_{t_n}} d_m$ in (10). Then, by applying the same analysis used for problem (9) to problem (10) with $\tilde{d}_n$ replacing d_n, it can be shown that

$$\alpha_* d_T^* \leq v^T \leq \alpha^* d_T^*. \tag{13}$$

Combining (11), (12), and (13), the claim follows. □

Suppose that the cost parameters α_n are nearly constant, i.e., $\alpha^* \approx \alpha_* \approx \alpha$, then Theorem 2 implies

$$\text{VMS} \approx \alpha(d_T^* - \bar{d}_T).$$

Thus, VMS is directly related to the variability of the demand. If demand variability is high then VMS is high, and the two-stage approach is likely to produce poor quality solutions. On the other hand, if there is little variability in the demand data, then the multi-stage approach has little value.

3.2 VMS for the capacity planning problem

We shall now describe a lower bound on the VMS for the multi-stage capacity planning model (4) based on the analysis in the previous section and an optimal solution to the LP relaxation of the two-stage model (3). Since this LP relaxation can be solved fairly quickly, we can use this lower bound estimate to justify additional computational effort required to solve the difficult multi-stage model (4).

THEOREM 3 *Let $\{y_n^{TLP}\}_{n\in\mathcal{T}}$ be the capacity allocation decisions in an optimal solution to the linear relaxation of the two-stage model (3). Then for each resource $i = 1,\ldots,I$, let $d_{in} = [A_n y_n^{TLP}]_i$, $d_{iT}^* = \max_{n\in\mathcal{S}_T}(\max_{m\in\mathcal{P}(n)} d_{im})$, $\overline{d}_{iT} = \sum_{n\in\mathcal{S}_T} p_n(\max_{m\in\mathcal{P}(n)} d_{im})$, $\alpha_i^* = \max_{n\in\mathcal{T}} \alpha_{in}$, and $\alpha_{i*} = \min_{n\in\mathcal{T}} \alpha_{in}$, we have*

$$\text{VMS} \geq \sum_{i=1}^{I} \Big(\alpha_{i*} d_{iT}^* - \alpha_i^* \overline{d}_{iT}\Big) - \sum_{i=1}^{I} \alpha_{i1}.$$

PROOF. Note that

$$v^{TS} \geq \sum_{n\in\mathcal{T}} p_n \beta_n y_n^{TLP} + \sum_{i=1}^{I} v_i^T$$

where

$$\begin{array}{rcll} v_i^T = & \min & \sum_{n\in\mathcal{T}} p_n \alpha_{in} x_{in} & \\ & \text{s.t.} & \sum_{m\in\mathcal{P}(n)} x_{im} \geq d_{in} & \forall\, n \in \mathcal{T} \\ & & x_{in} \in \mathbb{R}_+ & \forall\, n \in \mathcal{T} \\ & & x_{im} = x_{in} & \forall\, m, n \in \mathcal{S}_t,\ \forall\, t. \end{array}$$

Since $\{y_n^{TLP}\}_{n\in\mathcal{T}}$ is a feasible capacity allocation for the multi-stage problem (4), we have

$$v^{MS} \leq \sum_{n\in\mathcal{T}} p_n \beta_n y_n^{TLP} + \sum_{i=1}^{I} o_i^M$$

where

$$\begin{array}{rcll} o_i^M & = & \min & \sum_{n\in\mathcal{T}} p_n \alpha_{in} x_{in} & \\ & & \text{s.t.} & \sum_{m\in\mathcal{P}(n)} x_{im} \geq d_{in} & \forall\, n \in \mathcal{T} \\ & & & x_{in} \in \mathbb{Z}_+ & \forall\, n \in \mathcal{T} \\ & = & \min & \sum_{n\in\mathcal{T}} p_n \alpha_{in} x_{in} & \\ & & \text{s.t.} & \sum_{m\in\mathcal{P}(n)} x_{im} \geq \lceil d_{in} \rceil & \forall\, n \in \mathcal{T} \\ & & & x_{in} \in \mathbb{R}_+ & \forall\, n \in \mathcal{T} \\ & = & \max & \sum_{n\in\mathcal{T}} [d_{in} + (\lceil d_{in} \rceil - d_{in})] \pi_{in} & \\ & & \text{s.t.} & \sum_{m\in\mathcal{T}(n)} \pi_{im} \leq p_n \alpha_{in} & \forall\, n \in \mathcal{T} \\ & & & \pi_{in} \in \mathbb{R}_+ & \forall\, n \in \mathcal{T}, \end{array}$$

in which the second equality comes from Theorem 1 and the third equality comes from linear program duality. We can further define:

$$\begin{array}{rlll} v_i^M & = \min & \sum_{n\in\mathcal{T}} p_n\alpha_{in}x_{in} & \\ & \quad\text{s.t.} & \sum_{m\in\mathcal{P}(n)} x_{im} \geq d_{in} & \forall\; n\in\mathcal{T} \\ & & x_{in}\in\mathbb{R}_+ & \forall\; n\in\mathcal{T} \\ & = \max & \sum_{n\in\mathcal{T}} d_{in}\pi_{in} & \\ & \quad\text{s.t.} & \sum_{m\in\mathcal{T}(n)} \pi_{im} \leq p_n\alpha_{in} & \forall\; n\in\mathcal{T} \\ & & \pi_{in}\in\mathbb{R}_+ & \forall\; n\in\mathcal{T}, \end{array}$$

Therefore,

$$\begin{array}{rllll} o_i^M & \leq v_i^M & + \max & \sum_{n\in\mathcal{T}}(\lceil d_{in}\rceil - d_{in})\pi_{in} & \\ & & \quad\text{s.t.} & \sum_{m\in\mathcal{T}(n)} \pi_{im} \leq p_n\alpha_{in} & \forall\; n\in\mathcal{T} \\ & & & \pi_{in}\in\mathbb{R}_+ & \forall\; n\in\mathcal{T} \\ & \leq v_i^M & + \max & \sum_{n\in\mathcal{T}} \pi_{in} & \\ & & \quad\text{s.t.} & \sum_{m\in\mathcal{T}(n)} \pi_{im} \leq p_n\alpha_{in} & \forall\; n\in\mathcal{T} \\ & & & \pi_{in}\in\mathbb{R}_+ & \forall\; n\in\mathcal{T} \\ & = v_i^M & + \min & \sum_{n\in\mathcal{T}} p_n\alpha_{in}x_{in} & \\ & & \quad\text{s.t.} & \sum_{m\in\mathcal{P}(n)} x_{im} \geq 1 & \forall\; n\in\mathcal{T} \\ & & & x_{in}\in\mathbb{R}_+ & \forall\; n\in\mathcal{T} \\ & = v_i^M & + \alpha_{i1}, & & \end{array}$$

where the second inequality comes from $\lceil d_{in}\rceil - d_{in} \leq 1$, the third equality comes from duality, the fourth equality comes from the fact that $p_1 = 1$ and an optimal solution to a stochastic lot-sizing problem with a cumulative demand of 1 unit in *every* node is to produce 1 unit in the root node. Therefore, we have:

$$\text{VMS} \geq \sum_{i=1}^{I}\left(v_i^T - v_i^M - \alpha_{i1}\right),$$

and the result follows from the bounds (12) and (13) derived in the proof of Theorem 2. □

4 An approximation algorithm

In this section we develop an approximation algorithm for the multi-stage capacity planning problem (4).

It can be easily shown that any instance of the NP-hard integer knapsack problem (cf. Garey and Johnson (1979)) with I items can be polynomially transformed to a single period instance of the deterministic capacity planning problem (2). Since (2) is just a single scenario instance of the stochastic models (3) and (4), we have the following result. The detailed proof is omitted in the interest of brevity.

THEOREM 4 *The deterministic capacity planning problem (2) and its stochastic counterparts (3) and (4) are NP-hard.*

Motivated by this intractability, we propose the approximation scheme outlined in Figure 1. The algorithm exploits the decomposable structure revealed by the reformulation (5)-(6) of the problem.

Step 1 of Algorithm 1 requires the solution of the LP relaxation of (4). This problem is a multi-stage stochastic linear program which can, in general, be solved by the Nested L-Shaped Decomposition algorithm (cf. Birge (1985)). Step 2 requires the solution of I stochastic lot-sizing problems (7) which are multi-stage stochastic integer programs. In following section (Section 4.1), we show that these problems can be solved extremely efficiently using a specialized scheme. Finally, Step 3 requires the solution of independent simple linear capacity allocation problems for each node in the tree.

4.1 An efficient algorithm for the stochastic lot-sizing problem

By virtue of Theorem 1 and the fact that the right-hand-sides of the stochastic lot-sizing problems solved in Step 2 of Algorithm 1 are integral, we only need to find an efficient scheme for the linear program

$$\begin{array}{lll} \min & \sum_{n\in\mathcal{T}} c_n x_n & \\ \text{s.t.} & \sum_{m\in\mathcal{P}(n)} x_m \geq d_n & \forall\, n \in \mathcal{T} \\ & x_n \in \mathbb{R}_+ & \forall\, n \in \mathcal{T}, \end{array} \tag{14}$$

Figure 1: An approximation scheme for the multi-stage capacity planning problem (4)

Algorithm 1

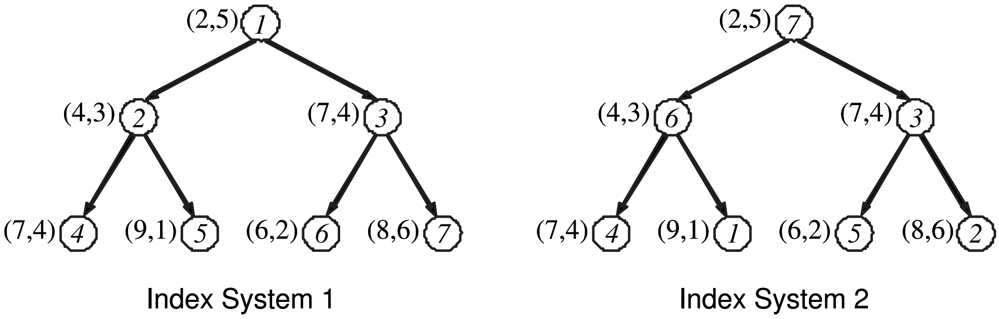
```
1: Solve the LP relaxation of (4). Let $\{(x_n^{LP}, y_n^{LP})\}_{n \in \mathcal{T}}$ be an optimal solution and $v_{MS}^{LP}$ be
   the optimal value. If $x_n^{LP}$ is integral for all $n$, stop and return $\{(x_n^{LP}, y_n^{LP})\}_{n \in \mathcal{T}}$.
2: For each resource $i = 1, 2, ..., I$, solve independent capacity acquisition (or stochastic
   lot-sizing) problems:
       $\begin{array}{lll} \min & \sum_{n \in \mathcal{T}} p_n \alpha_{in} x_{in} & \\ \text{s.t.} & \sum_{m \in \mathcal{P}(n)} x_{im} \geq \left\lceil [A_n y_n^{LP}]_i \right\rceil & \forall\, n \in \mathcal{T} \\ & x_{in} \in \mathbb{Z}_+ & \forall\, n \in \mathcal{T}, \end{array}$
   and let $x_{in}^H$ denote the corresponding solutions. Note that the integrality of the decision
   variables allows for the rounding up of $[A_n y_n^{LP}]_i$.
3: For each $n \in \mathcal{T}$, solve independent capacity allocation problems:
       $\begin{array}{ll} \min & \beta_n y_n \\ \text{s.t.} & A_n y_n \leq \sum_{m \in \mathcal{P}(n)} x_m^H \\ & B_n y_n \geq \delta_n, y_n \in \mathbb{R}_+^J, \end{array}$
   and let $y_n^H$ denote the corresponding optimal solution.
4: Return $\{(x_n^H, y_n^H)\}_{n \in \mathcal{T}}$.
```

where $d_n \in \mathbb{Z}$ and we have used c_n to succinctly denote $p_n \alpha_n$. The dual of (14) is

$$\begin{array}{lll} \max & \sum_{n \in \mathcal{T}} d_n \pi_n & \\ \text{s.t.} & \sum_{m \in \mathcal{T}(n)} \pi_m \leq c_n & \forall\, n \in \mathcal{T} \\ & \pi_n \in \mathbb{R}_+ & \forall\, n \in \mathcal{T}, \end{array} \tag{15}$$

Our proposed algorithm scheme is based on a greedy approach for solving the dual problem (15). The scheme takes advantage of complementary slackness conditions to recover primal optimal solutions. Figure 3 summarizes the proposed strategy. Here, we assume that the parameters c_n and d_n are strictly positive for all n. The scheme uses two different indexing schemes for the nodes in the tree $\mathcal{T}$:

Figure 2: The Indexing schemes. The numbers in parenthesis indicate (d_n, c_n)



Indexing scheme 1. The nodes in $\mathcal{T}$ are indexed $1, 2, \ldots, N_T$ in increasing order of their time stage, i.e., $t_1 \leq t_2 \leq \cdots \leq t_{N_T}$. No particular ordering is imposed on the indices of the nodes in the same time stage. Thus the root node has an index of 1.

Indexing scheme 2. The nodes in $\mathcal{T}$ are indexed $1, 2, \ldots, N_T$ in decreasing order of the corresponding cumulative demand, i.e., $d_1 \geq d_2 \geq \cdots \geq d_{N_T}$. If $d_m = d_n$, then $m < n$ if $t_m < t_n$.

The two indexing schemes corresponding to an exemplary scenario tree are illustrated in Figure 2.

As mentioned earlier, the greedy dual step first assigns the largest dual value (as permitted by the constraints) to the node with the largest demand, and then considers the node with the next largest demand, and so on. The marker m_k is assigned the index of the node (closest on the path $\mathcal{P}(k)$) to k whose corresponding dual constraint becomes tight when the dual value for node k is set. Note that once k has a positive dual value, no other nodes in $\mathcal{T}(m_k)$ will be further considered (any other node in $\mathcal{T}(m_k)$ will satisfy the condition in line 4 of the dual step). Thus all nodes in $l \in \mathcal{T}(m_k)$ except for the ones with $m_l > 0$, will have $\pi_l^* = 0$. The marker m_k is used in the primal step to set the primal variables such that complementary slackness conditions are satisfied. Note that according to the algorithm, only a node k with $m_k > 0$ could have a positive dual value, and a node n could have a positive primal value only if $n = m_l$ for some node $l \in \mathcal{T}(n)$.

The following results establish the validity of Algorithm 2. For the remainder of this

Figure 3: An algorithm for the stochastic lot-sizing primal and dual problems (14) and (15).

Algorithm 2

The dual step:

```
label the nodes in $\mathcal{T}$ according to indexing scheme 2
initialize $\pi_n^* = 0$, $c_n^0 = c_n$ and $m_n = 0$ for all $n \in \mathcal{T}$
for $k = 1, \ldots, N_T$ do
    if there exists $n \in \mathcal{P}(k)$ such that $n = m_l$ for some $l \in \mathcal{T}(n)$ then
        break
    else
        set $\pi_k^* = \min_{n \in \mathcal{P}(k)} \{c_n^{k-1}\}$
        set $m_k = \mathrm{argmin}_{n \in \mathcal{P}(k)} \{c_n^{k-1}\}$ such that $c_l^{k-1} > c_{m_k}^{k-1}$ for all $l \in \mathcal{P}(k) \setminus \mathcal{P}(m_k)$
        set $c_n^k = c_n^{k-1} - \pi_k^*$ if $n \in \mathcal{P}(k)$ and $c_n^k = c_n^{k-1}$ otherwise
    end if
end for
```

The primal step:

```
transform the node indices as well as $m_n$ to indexing scheme 1
initialize $x_n^* = 0$ for all $n \in \mathcal{T}$
for $n = 1, \ldots, N_T$ do
    if there exists $l \in \mathcal{T}(n)$ such that $n = m_l$ then
        set $x_n^* = d_l - \sum_{k \in \mathcal{P}(n) \setminus \{n\}} x_k^*$
    end if
end for
```

section we use indexing scheme 2 for the node labels.

LEMMA 1 *In each iteration $k \in \{1, \ldots, N_T\}$ of the dual step of Algorithm 2, the dual solution π^* satisfies*

$$\sum_{m \in \mathcal{T}^k(n)} \pi^*_m \le c_n \quad \text{for all } n \in \mathcal{T}, \tag{16}$$

where $\mathcal{T}^k(n) = \mathcal{T}(n) \cap \{1, 2, \ldots, k\}$.

PROOF. By induction on k, it can be seen that $c^k_n = c_n - \sum_{m \in \mathcal{T}^k(n)} \pi^*_m$. Also, $n \in \mathcal{P}(k)$ if and only if $k \in \mathcal{T}(n)$ and $\pi^*_k \le c^{k-1}_n$ for all $n \in \mathcal{P}(k)$. Therefore, we always have $c^k_n \ge 0$ for any $n \in \mathcal{T}$. □

Lemma 1 guarantees the feasibility of the dual solution π^* (let $k = N_T$ in (16)). Furthermore, we also have $\sum_{l \in \mathcal{T}^k(n)} \pi^*_l = c_n$ for all $n \in \text{argmin}_{m \in \mathcal{P}(k)}\{c^{k-1}_m\}$. Since dual feasibility of π^* implies $\sum_{l \in \mathcal{T}(n)} \pi^*_l \le c_n$ and $\pi^*_n \ge 0$, we also have:

$$\sum_{l \in \mathcal{T}^k(n)} \pi^*_l = \sum_{l \in \mathcal{T}(n)} \pi^*_l = c_n \quad \forall\, n \in \text{argmin}_{m \in \mathcal{P}(k)}\{c^{k-1}_m\}. \tag{17}$$

That is, for any $l \in \mathcal{T}(n)$ such that $l \notin \{1, \ldots, k\}$, π^*_l=0.

LEMMA 2 *The primal solution $x^* = (x^*_1, x^*_2, \ldots, x^*_{N_T})$ produced by the primal step of Algorithm 2 is feasible.*

PROOF. By construction, if a node n is such that $m_n > 0$, then the following equalities hold:

$$\sum_{m \in \mathcal{P}(n)} x^*_m = \sum_{m \in \mathcal{P}(m_n)} x^*_m = d_n. \tag{18}$$

Now consider a node n such that $m_n = 0$. Let $l \in \mathcal{P}(n)$ be such that $l = m_k$ for some node $k \in \mathcal{T}(m_k)$. Note that such a node l must always exists, since the root node is one such node. Suppose l be the closest (on the path $\mathcal{P}(n)$) such node to n. Note that $n, k \in \mathcal{T}(l)$ while $m_n = 0$ and $m_k > 0$, thus in the dual step node k must have been considered before node n, i.e., $d_k \ge d_n$. According to (18), $\sum_{m \in \mathcal{P}(n)} x^*_m = \sum_{m \in \mathcal{P}(m_k)} x^*_m = \sum_{m \in \mathcal{P}(k)} x^*_m = d_k \ge d_n$ (the first equality holds since $x^*_m = 0$ for $m \in \mathcal{P}(n) \backslash \mathcal{P}(m_k)$). □

THEOREM 5 *The solutions x^* and π^* returned by Algorithm 2 are optimal solutions of* (14) *and* (15)*, respectively.*

PROOF. Lemmas 1 and 2 have proven the feasibility of π^* and x^*. Here we show that π^* and x^* satisfies the complementary slackness conditions:

$$\pi_n^* > 0 \implies \sum_{m \in \mathcal{P}(n)} x_m^* = d_n \tag{19}$$

$$\sum_{m \in \mathcal{T}(n)} \pi_m^* < c_n \implies x_n^* = 0. \tag{20}$$

We prove by induction on the nodes indexed according to scheme 2.

The base case: Consider node 1. Note that $\pi_1^* = c_{m_1} > 0$, then (19) follows from (18). For all other nodes $n \in \mathcal{T}(m_1)$, $\pi_n^* = 0$. On the other hand, $\sum_{m \in \mathcal{T}(m_1)} \pi_m^* = c_{m_1}$, thus $\{n \in \mathcal{T}(m_1) : \sum_{m \in \mathcal{T}(n)} \pi_m^* < c_n\} \subseteq \mathcal{T}(m_1) \backslash \{m_1\}$. Then (20) holds, since $x_n^* = 0$ for all $n \in \mathcal{T}(m_1) \backslash \{m_1\}$. Note that we have verified the complementary slackness conditions for all nodes in $\mathcal{T}(m_1)$, and not just node 1.

The induction step: Assume that we have checked nodes $1, ..., k$, and now consider node $k+1$. If $m_{k+1} = 0$, then this node has already been checked since then $k+1 \in \mathcal{T}(m_j)$ for some $j < k+1$. So we assume that $m_{k+1} > 0$. Denote $\{1, 2, ..., k\}$ by $\mathcal{H}(k)$ and $\{m_1, m_2, ..., m_k\}$ by $\mathcal{R}(k)$. Also define $\mathcal{F}(k) = \cup\{\mathcal{T}(m_n) : n \in \mathcal{H}(k), m_n > 0\}$. We now examine the nodes in $\mathcal{T}(m_{k+1}) \backslash \mathcal{F}(k)$. Notice for all nodes n in $\mathcal{T}(m_{k+1}) \backslash \mathcal{F}(k) \backslash \{k+1\}$, $\pi_n^* = 0$ since $m_n = 0$. Amongst the nodes in $\mathcal{T}(m_{k+1}) \backslash \mathcal{F}(k)$, only node $k+1$ could have a positive dual value. For node $k+1$, (19) then holds from (18). On the other hand, $\sum_{m \in \mathcal{T}(m_{k+1})} \pi_m^* = c_{m_{k+1}}$ from (17), so $\{l \in \mathcal{T}(m_{k+1}) \backslash \mathcal{F}(k) : \sum_{m \in \mathcal{T}(l)} \pi_m^* < c_l\} \subseteq \mathcal{T}(m_{k+1}) \backslash \mathcal{F}(k) \backslash \{m_{k+1}\}$. The conclusion then holds since $x_m^* = 0$ for all $m \in \mathcal{T}(m_{k+1}) \backslash \mathcal{F}(k) \backslash \{m_{k+1}\}$. □

It can be shown that by adopting an appropriate data structure, Algorithm 2 can be executed in no more than $O(N_T \log N_T \log \log N_T)$ operations. We omit details of this complexity calculation here.

5 Analysis of the approximation algorithm

This section analyzes the optimality gap of the approximate solution produced by Algorithm 1. Given capacity acquisition-allocation solutions (x, y), let us denote the corresponding objective function value as

$$f(x,y) = \sum_{n\in\mathcal{T}} p_n\Big(\alpha_n x_n + \beta_n y_n\Big).$$

Recall that (x^{LP}, y^{LP}) denotes the capacity acquisition-allocation solutions corresponding to the LP relaxation of (4), and (x^H, y^H) denotes the capacity acquisition-allocation solutions returned by Algorithm 1. Let (x^*, y^*) denote an optimal solution to (4). Then the optimality gap of (x^H, y^H) is

$$\text{GAP} = f(x^H, y^H) - f(x^*, y^*).$$

THEOREM 6 GAP $\leq \sum_{i=1}^{I} \alpha_{i1}$, *where* 1 *is the root node of the scenario tree.*

PROOF. Note that

$$\begin{aligned} \text{GAP} &\leq f(x^H, y^H) - f(x^{LP}, y^{LP}) \\ &= f(x^H, y^H) - f(x^H, y^{LP}) + f(x^H, y^{LP}) - f(x^{LP}, y^{LP}) \\ &\leq f(x^H, y^{LP}) - f(x^{LP}, y^{LP}), \end{aligned}$$

where last inequality follows from the fact that $f(x^H, y^H) \leq f(x^H, y^{LP})$ (recall that y^H is an optimal capacity allocation corresponding to x^H, i.e., an optimal solution to capacity allocation problem solved in step 3 of Algorithm 1, whereas y^{LP} is just a feasible capacity allocation solution). Now

$$f(x^H, y^{LP}) - f(x^{LP}, y^{LP}) = \sum_{i=1}^{I}\sum_{n\in\mathcal{T}} p_n\alpha_{in}\Big(x^H_{in} - x^{LP}_{in}\Big). \tag{21}$$

Note that

$$\begin{aligned} \textstyle\sum_{n\in\mathcal{T}} p_n\alpha_{in}x^H_{in} \;=\; & \min && \textstyle\sum_{n\in\mathcal{T}} p_n\alpha_{in}x_{in} && \\ & \text{s.t.} && \textstyle\sum_{m\in\mathcal{P}(n)} x_{im} \geq \Big\lceil [A_n y^{LP}_n]_i \Big\rceil && \forall\, n\in\mathcal{T} \\ & && x_{in}\in\mathbb{R}_+ && \forall\, n\in\mathcal{T} \\ =\; & \max && \textstyle\sum_{n\in\mathcal{T}} \Big\lceil [A_n y^{LP}_n]_i \Big\rceil \pi_{in} && \\ & \text{s.t.} && \textstyle\sum_{m\in\mathcal{T}(n)} \pi_{im} \leq p_n\alpha_{in} && \forall\, n\in\mathcal{T} \\ & && \pi_{in}\in\mathbb{R}_+ && \forall\, n\in\mathcal{T}, \end{aligned} \tag{22}$$

and

$$\begin{aligned}\textstyle\sum_{n\in\mathcal{T}} p_n\alpha_{in}x_{in}^{LP} &= \min && \textstyle\sum_{n\in\mathcal{T}} p_n\alpha_{in}x_{in} \\ & \quad\ \text{s.t.} && \textstyle\sum_{m\in\mathcal{P}(n)} x_{im} \geq [A_n y_n^{LP}]_i && \forall\, n\in\mathcal{T} \\ & && x_{in}\in\mathbb{R}_+ && \forall\, n\in\mathcal{T} \\ &= \max && \textstyle\sum_{n\in\mathcal{T}}[A_n y_n^{LP}]_i\pi_{in} \\ & \quad\ \text{s.t.} && \textstyle\sum_{m\in\mathcal{T}(n)} \pi_{im} \leq p_n\alpha_{in} && \forall\, n\in\mathcal{T} \\ & && \pi_{in}\in\mathbb{R}_+ && \forall\, n\in\mathcal{T}.\end{aligned} \tag{23}$$

Thus

$$\begin{aligned}\textstyle\sum_{n\in\mathcal{T}} p_n\alpha_{in}\Big(x_{in}^{H}-x_{in}^{LP}\Big) &= \max && \textstyle\sum_{n\in\mathcal{T}}\Big(\Big\lceil [A_n y_n^{LP}]_i\Big\rceil - [A_n y_n^{LP}]_i\Big)\pi_{in} \\ & \quad\ \text{s.t.} && \textstyle\sum_{m\in\mathcal{T}(n)} \pi_{im} \leq p_n\alpha_{in} && \forall\, n\in\mathcal{T} \\ & && \pi_{in}\in\mathbb{R}_+ && \forall\, n\in\mathcal{T} \\ &\leq \max && \textstyle\sum_{n\in\mathcal{T}} \pi_{in} \\ & \quad\ \text{s.t.} && \textstyle\sum_{m\in\mathcal{T}(n)} \pi_{im} \leq p_n\alpha_{in} && \forall\, n\in\mathcal{T} \\ & && \pi_{in}\in\mathbb{R}_+ && \forall\, n\in\mathcal{T} \\ &= \min && \textstyle\sum_{n\in\mathcal{T}} p_n\alpha_{in}x_{in} \\ & \quad\ \text{s.t.} && \textstyle\sum_{m\in\mathcal{P}(n)} x_{im} \geq 1 && \forall\, n\in\mathcal{T} \\ & && x_{in}\in\mathbb{R}_+ && \forall\, n\in\mathcal{T} \\ &= && \alpha_{i1},\end{aligned} \tag{24}$$

where the first equality follows from (22) and (23), the next inequality follows from the fact that $\Big\lceil [A_n y_n^{LP}]_i\Big\rceil - [A_n y_n^{LP}]_i \leq 1$, the next equality follows from duality, and the last equality follows from the fact that an optimal solution to a stochastic lot-sizing problem with a cumulative demand of 1 unit in *every* node is to produce 1 unit in the root node. The result then follows from incorporating (24) in (21). □

Theorem 6 shows the surprising result that the optimality gap of Algorithm 1 is bounded above by a factor that is independent of the number of time stages, number of branches in the tree, number of tasks, or any problem data except for the sum of the capacity acquisition costs of the resources in the *first stage*. If we consider instances of (4) that have the same first-stage acquisition costs, but different topology of the scenario tree, then we have the following asymptotic quality guarantees for Algorithm 1. Corollary 1 is immediate, and the proof of Corollary 2 is provided in the appendix.

COROLLARY 1 $\lim_{T\to\infty} \frac{f(x^H,y^H)-f(x^*,y^*)}{T} = 0.$

COROLLARY 2 *Assume that,*

(i) there exists $\epsilon_1 > 0$ such that, for each $n \in \mathcal{T}$, there exists at least one product $k_n \in \{1, \ldots, K\}$ whose demand is at least ϵ_1, i.e., $[\delta_n]_{k_n} \geq \epsilon_1$; and

(ii) there exists $\epsilon_2 > 0$ such that, for each $n \in \mathcal{T}$, and any task $j \in \{1, \ldots, J\}$ and product $k \in \{1, \ldots, K\}$ with a positive demand-task allocation ratio, i.e., $[B_n]_{kj} > 0$, the allocation cost $[\beta_n]_j \geq \epsilon_2 [B_n]_{kj}$.

Then the following holds:

$$\lim_{T\to\infty} \frac{f(x^H, y^H) - f(x^*, y^*)}{f(x^*, y^*)} = 0.$$

Note that the assumptions in Corollary 2 are not particularly restrictive. These only require that, for every node of the scenario tree, there is always some positive demand, and that the unit allocation cost is never smaller than some level.

6 Computational results

In this section, we report on computational experiments with the proposed multi-stage stochastic programming approach for a realistic scale semiconductor tool planning problem. Our experiments focus on two objectives: (i) to investigate the value of multi-stage stochastic programming; and, (ii) to investigate the performance of the proposed approximation scheme. In the following, we first describe our experimental environment and then report on the experimental results in light of each of the above two objectives.

6.1 Experimental environment

Our test problem instances are derived from a realistic scale two-stage stochastic programming model for semiconductor tool planning from Barahona et al. (2005) and Hood et al. (2003). The formulation is very similar to (1) with an additional purchase budget constraint.

Numerical data for instances of the model with two periods and 2, 3, and 4 scenarios are available in Ahmed (2004) (see the SEMI test set). The instances consists of 306 machine tools, 40 wafer types (products) and 2575 processing steps. The only uncertain parameters are demands of 7 of the 40 products. The demand data for the uncertain products for each scenario varies around that of a "base" scenario (having the highest probability).

We generate our test problem instances from the above data set as follows. We ignore the budget constraint since our approach is not designed to handle such a constraint. The original cost and demand data corresponding to the first period base scenario is used for the root node (node 1) of our scenario tree. The demand data (for the 7 products with uncertain demand) for each subsequent node is independently generated by multiplying the root node data with a random number generated from a lognormal distribution $\lambda(\mu, \sigma)$, where μ is the expectation and σ is the standard deviation. We considered four trends of the demand with respect to the time period. These demand patters are indicated in Table 1. In Table 1, z_n is the demand of a product in node n, and z_1 is the demand of the product in the root node. Recall that t_n is the stage number of node n (if $n \in \mathcal{S}_t$, then $t_n = t$). So for all nodes in the same stage, we have the same demand distribution. The cost data is discounted at the rate of 5% for each stage of the scenario tree.

Table 1: Demand patterns

	Characteristic	Distribution
1	constant mean, constant standard deviation.	$z_n \sim z_1\lambda(1, 0.5)$
2	constant mean, increasing standard deviation.	$z_n \sim z_1\lambda(1, 0.5 + 0.1t_n)$
3	increasing mean, constant standard deviation.	$z_n \sim z_1\lambda(1 + 0.5t_n, 0.5)$
4	increasing mean, increasing standard deviation.	$z_n \sim z_1\lambda(1 + 0.5t_n, 0.5 + 0.1t_n)$

We consider scenario trees with the number of stages (T) varying from 2 to 5, and the number of branches (B) for each non-leaf node varying from 2 to 5. Thus, there are, in total, 7 scenario tree structures. The nodes in these trees vary from 3 to 31. For each tree structure and demand pattern combination, we generate 5 problem instances, and report statistics averaged over these 5 instances. A total of $(7 \times 4 \times 5 =)$ 140 problem instances are considered. To get a sense of the sizes of these instances, note that the smallest multi-stage instance with $T = 2$ and $B = 2$ consists of 8,763 constraints and 15,621 variables of which 918 are integers, and an instance with $T = 5$ and $B = 2$ consists 90,551 constraints and 161,417 variables of which 9,486 are integers.

Our experiments utilize C/C++ implementations of Algorithms 1 and 2. CPLEX 9.0 is used to solve the linear programs in Steps 1 and 3 of Algorithm 1. All numerical experiments are conducted on an IBM PC with 1024 MB RAM and a PENTIUM4 1.6GHz processor.

6.2 Value of multi-stage stochastic programming

To compare two-stage and multi-stage models, we define the Relative Value of Multi-stage Stochastic Programming as:

$$\text{RVMS} = \frac{v^{TS} - v^{MS}}{v^{TS}},$$

where v^{TS}, v^{MS} are the optimal values of two-stage model and multi-stage model, respectively. However, since it is hard to solve the two-stage and multi-stage models to optimality, we consider the following lower bound:

$$\text{RVMS} \geq \frac{v_{LP}^{TS} - v_{HR}^{MS}}{v_{LP}^{TS}},$$

where v_{LP}^{TS} is the optimal value of the linear relaxation of two-stage model and v_{HR}^{MS} is the objective function value of an approximate solution (obtained using Algorithm 1) for the multi-stage model.

In Figure 4, we observe the behavior of the lower bound on RVMS (averaged over 5 instances) with respect to the number of stages T for each of the 4 demand patterns. The number of branches B is fixed at 2. Our first observation is that for all the four demand patterns, the RVMS lower bound increases as the number of stages increases. This implies that the value of multi-stage stochastic programming increases with the planning horizon. Our second observation is that, consistent with the theoretical analysis of Section 3, the value of multi-stage stochastic programming increases with the variability of demand (the RVMS lower bound is larger for demand patterns 2 and 4 that have increasing variability). Moreover, the rate at which the value increases with the planning horizon length also increases with demand variability.

Figure 4: The value of multi-stage stochastic programming with increasing planning horizon

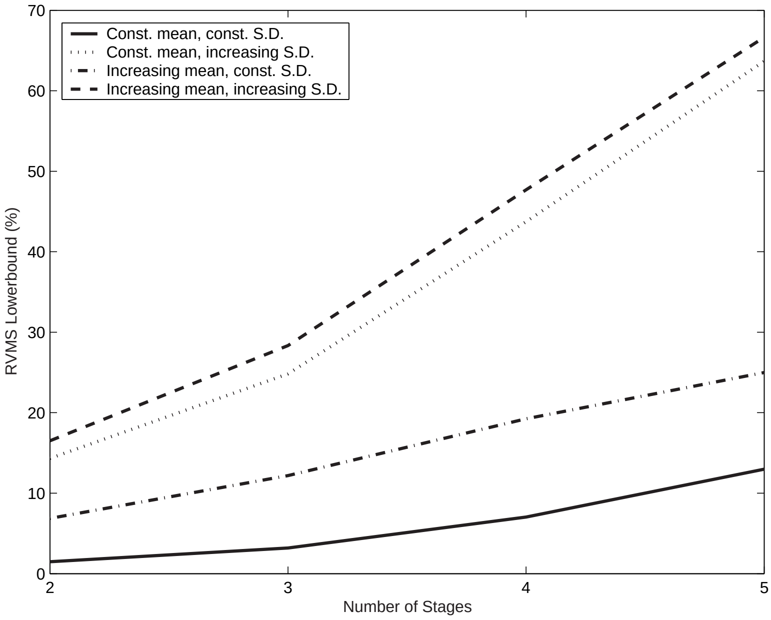
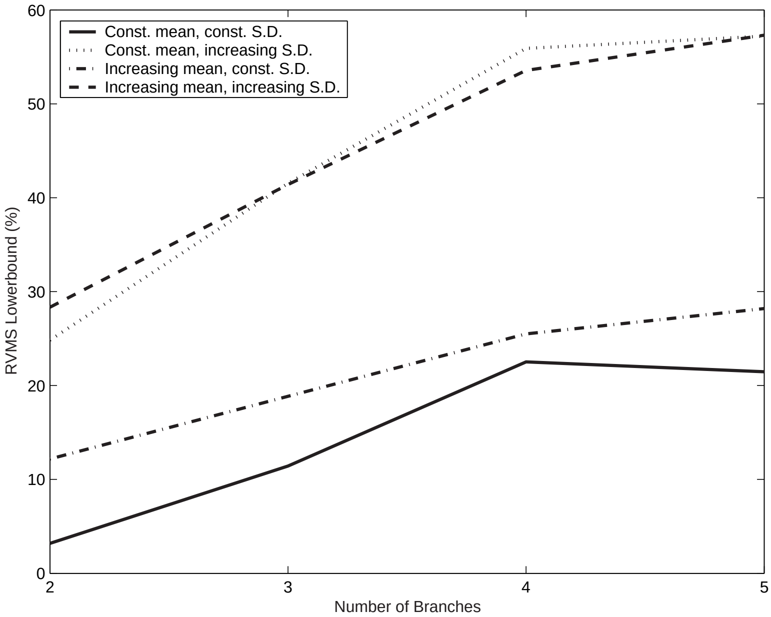


Figure 5: The value of multi-stage stochastic programming with increasing number of branches



In Figure 5, we observe the behavior of the lower bound on RVMS (averaged over 5 instances) with respect to the number of branches B for each of the 4 demand patterns. The number of stages T is fixed at 3. We observe that in most cases, the value of multi-stage stochastic programming increases with the number of branches since the variability of the demand data increases. Also, as before, the rate at which the value increases with the number of branches also increases with demand variability.

6.3 Performance of the approximation scheme

In this section, we report on the solution quality and computational efficiency of Algorithm 1.

A measure of the quality of an approximate solution to the multi-stage model is the relative gap defined as:

$$\text{RGAP} = \frac{v^{MS}_{HR} - v^{MS}}{v^{MS}},$$

where v^{MS}_{HR} and v^{MS} denote the objective value of the approximate solution and that of an optimal solution of the multi-stage model, respectively. To avoid solving the multi-stage model to optimality, we consider an upper bound on RGAP:

$$\text{RGAP} \leq \frac{v^{MS}_{HR} - v^{MS}_{LP}}{v^{MS}_{LP}},$$

where v^{MS}_{LP} is the optimal value of the linear programming relaxation of the multi-stage model.

In Figure 6, we observe the behavior of the upper bound on RGAP (averaged over 5 instances) with respect to the number of stages T for each of the 4 demand patterns. The number of branches B is fixed at 2. Our first observation is that typically the upper bound on RGAP decreases, hence the approximate solution quality increases, with the increase in the number of stages. This is consistent with the theoretical analysis in Corollary 2. Our second observation is that the upper bound on RGAP decreases, hence the approximate solution quality increases, with increase in the demand variability. Comparing Figures 4 and 6, we find that, fortunately, the instance with high VMS are precisely the ones for which the approximation schemes provide good quality solutions.

Figure 6: The quality of the approximate solution with increasing planning horizon

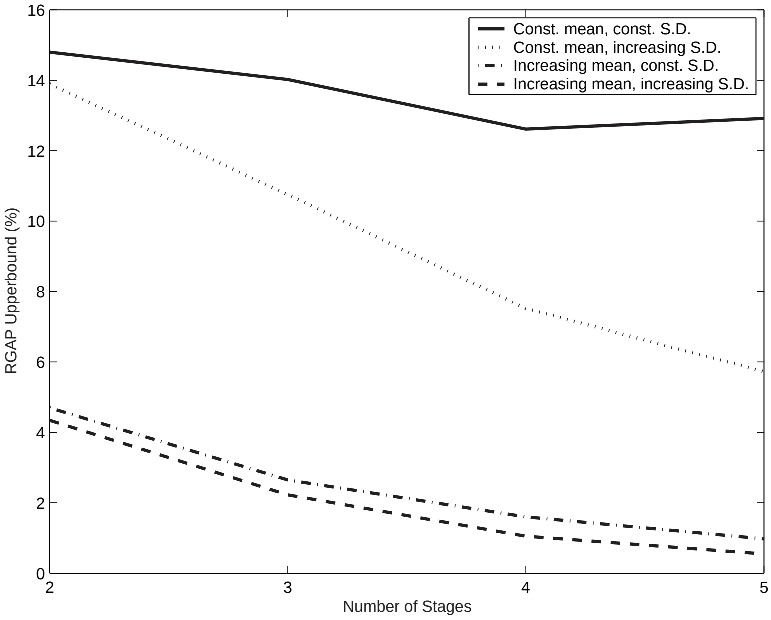
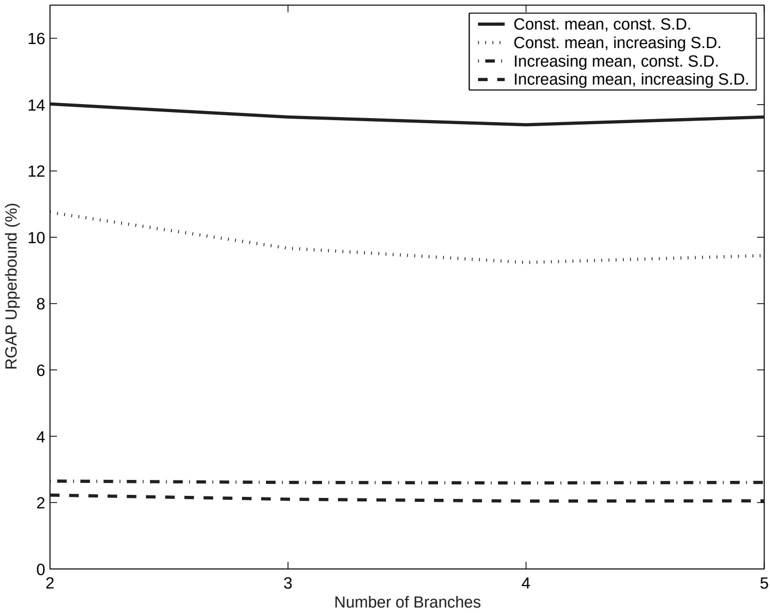


Figure 7: The quality of the approximate solution with increasing number of branches



In Figure 6, we observe the behavior of the upper bound on RGAP (averaged over 5 instances) with respect to the number of branches B for each of the 4 demand patterns. The number of stages T is fixed at 2. In this case, we observe that the upper bound on RGAP is quite independent on the number of branches. This can be explained by the fact that the optimal value of the multi-stage model is little affected by the number of branches. On the other hand the effect of demand variability is again clear, higher demand variability leads to smaller upper bound on RGAP, i.e., better quality approximate solutions.

Overall, the relative optimality gap of the approximation scheme is at most 15% and could be as small as 1%. Moreover, as mentioned earlier, it is consistently observed that, for instances with high VMS the relative optimality gap is small.

Finally, to appreciate the computational efficiency of the proposed approximation scheme, note that for all of the instances considered, the approximation scheme never requires more than 2 CPU minutes. By contrast, exact optimization of a multi-stage instance, with just 3 nodes in the scenario tree ($T = 2$ and $B = 2$), using the MIP solver of CPLEX 9.0 requires over an hour.

7 Summary of contributions

In this paper, we propose a generic multi-period capacity planning problem under uncertainty involving multiple resources, tasks and products.

First, we compare two-stage and multi-stage stochastic integer programming approaches for this problem. The concept of value of multi-stage stochastic programming (VMS) is discussed and informative analytical bounds are developed.

Second, by identifying and exploiting a key lot-sizing substructure in the problem, we propose an efficient approximation scheme for the difficult multi-stage model. We show that the absolute optimality gap of the approximation scheme is bounded above by a factor that is independent of the number of time stages, number of branches in the scenario tree, number of tasks, or any problem data except for the sum of the capacity acquisition costs of the resources in the first stage. This leads to an asymptotic optimality guarantee of the approximation scheme with respect to the number of planning stages.

Finally, we present numerical results using realistic-scale problem instances corresponding to semiconductor tool planning. Our numerical results indicate that a lower bound on the relative VMS can be as high as 70%. Recall that this lower bound is obtained by comparing the cost of an approximate solution to the multi-stage model to that of a lower bound on the cost of an optimal solution of the two-stage model. Therefore, this suggests that even an approximate solution to the multi-stage model may be far superior to any optimal solution to the two-stage model. These results confirm that the VMS for these problems is quite high. Moreover the quality and performance of the approximation scheme is very satisfactory, more so, for cases where the VMS is high.

Acknowledgments

A preliminary version of this work was presented at the International Conference on Modelling and Analysis in Semiconductor Manufacturing in Phoenix, AZ in 2002, and has appeared in the unrefereed proceedings of this conference. This research has been supported by the National Science Foundation under grants DMI-0099726 and DMI-0133943.

Appendix

PROOF of COROLLARY 2

We only need to show that $f(x^*, y^*) \to \infty$ as $n \to \infty$. Since $f(x^*, y^*) \geq f(x^{LP}, y^{LP})$, we only need to show $f(x^{LP}, y^{LP}) \to \infty$ as $n \to \infty$. For this purpose, we rewrite the linear relaxation of (4) as follows:

$$\begin{array}{lll}
\min & \sum_{n\in\mathcal{T}} p_n \Big[\alpha_n x_n + \beta_n y_n\Big] & \\
\text{s.t.} & \sum_{m\in\mathcal{P}(n)} x_m - A_n y_n \geq 0 & \forall\; n \in \mathcal{T} \\
 & B_n y_n \geq \delta_n & \forall\; n \in \mathcal{T} \\
 & y_n \in \mathbb{R}^J_+ & \forall\; n \in \mathcal{T} \\
 & x_n \in \mathbb{R}^I_+ & \forall\; n \in \mathcal{T},
\end{array}$$

whose dual program is:

$$\begin{array}{lll} \max & \sum_{n\in\mathcal{T}} \eta_n\delta_n & \\ \text{s.t.} & \sum_{m\in\mathcal{T}(n)} \gamma_n \le p_n\alpha_n & \forall\ n\in\mathcal{T} \\ & -\gamma_n A_n + \eta_n B_n \le p_n\beta_n & \forall\ n\in\mathcal{T} \\ & \gamma_n \in \mathbb{R}^I_+ & \forall\ n\in\mathcal{T} \\ & \eta_n \in \mathbb{R}^K_+ & \forall\ n\in\mathcal{T}, \end{array}$$

where $\alpha_n, \beta_n, \delta_n, \gamma_n, \eta_n$ are $1\times I, 1\times J, K\times 1, 1\times I, 1\times K$ vectors, respectively; and A_n, B_n are $I\times J$ and $K\times J$ matrices respectively. Let the objective function of the dual program be $g(\gamma_n, \eta_n)$. We will try to find a feasible solution $(\tilde{\gamma}_n, \tilde{\eta}_n)$ such that $g(\tilde{\gamma}_n, \tilde{\eta}_n) \ge T\epsilon$ for some $\epsilon > 0$. First, we assign $\tilde{\gamma}_n = 0$. From assumption (i) that $[\delta_n]_{k_n} \ge \epsilon_1$, and the constraint $B_n y_n \ge \delta_n$, it follows that there exists at least one j_{k_n} such that $[B_n]_{k_n,j_{k_n}} > 0$, otherwise the problem is not feasible (we can always make the problem feasible by adding a very expensive artificial resource that can satisfy all demand). Now assign $[\tilde{\eta}_n]_{k_n} = \frac{p_n[\beta_n]_{k_n}}{[B_n]_{k_n,j_{k_n}}}$, and $[\tilde{\eta}_n]_k = 0$ for all $k \ne k_n$. The constructed solution $(\tilde{\gamma}_n, \tilde{\eta}_n)$ is clearly dual feasible. Furthermore, from assumption (ii), $[\tilde{\eta}_n]_{k_n} \ge p_n\epsilon_2$. Thus $f(x^{LP}, y^{LP}) \ge g(\tilde{\gamma}_n, \tilde{\eta}_n) = \sum_{n\in\mathcal{T}} \tilde{\eta}_n\delta_n \ge \sum_{n\in\mathcal{T}} p_n\epsilon_1\epsilon_2 = T\epsilon_1\epsilon_2$, where the first inequality follows from weak duality and the last equality follows from the fact that $\sum_{n\in\mathcal{S}_t} p_n = 1$ for all $1 \le t \le T$. This completes the proof. □

References

S. Ahmed. SIPLIB: A stochastic integer programming test problem library. `http://www.isye.gatech.edu/~sahmed/siplib/`, 2004.

S. Ahmed and N.V. Sahinidis. An approximation scheme for stochastic integer programs arising in capacity expansion. *Operations Research*, 51:461–471, 2003.

F. Barahona, S. Bermon, O. Gunluk, and S. Hood. Robust capacity planning in semiconductor manufacturing. Technical Report RC22196, IBM Corporation, 2005.

J.C. Bean, J.L. Higle, and R.L. Smith. Capacity expansion under stochastic demands. *Operations Research*, 40:S210–S216, 1992.

J.F. Benders. Partitioning procedures for solving mixed variables programming problems. *Numersiche Mathematik*, 4:238–252, 1962.

O. Berman, Z. Ganz, and J. M. Wagner. A stochastic optimization model for planning capacity expansion in a service industry under uncertian demand. *Naval Research Logistics*, 41:545–564, 1994.

J.R. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33:989–1007, 1985.

M.H.A. David, M.A.H. Dempster, S.P. Sethi, and D. Vermes. Optimal capacity expansion under uncertainty. *Advances in Applied Probability*, 19:156–176, 1987.

G.D. Eppen, R.K. Martin, and L. Schrage. A scenario approach to capacity planning. *Operations Research*, 37:517–527, 1989.

C.H. Fine and R.M. Freund. Optimal investment in product-flexible manufacturing capacity. *Management Science*, 36:449–466, 1990.

J. Freidenfelds. Capacity expansion when demand is a birth-death random process. *Operations Research*, 28:712–721, 1980.

M.R. Garey and D.S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman and Co., 1979.

S.J. Hood, S. Bermon, and F. Barahona. Capacity planning under demand uncertainty for semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 16 (2), 2003.

S.M. Johnson. Sequential production planning over time at minimum cost. *Management Science*, 3(4):435–437, 1957.

S. Karabuk and S.D. Wu. Coordinating strategic capacity planning in the semiconductor industry. *Operations Research*, 51(6):839–849, 2003.

H. Luss. Operations research and capacity expansion problems: A survey. *Operations Research*, 30:907–947, 1982.

A.S. Manne. Capacity expansion and probabilistic growth. *Econometrica*, 29:632–649, 1961.

G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley and Sons, 1988.

J. Swaminathan. Tool capacity planning for semiconductor fabrication facilities under demand uncertainty. *European Journal of Operational Research*, 120:545–558, 2000.

P.H. Zipkin. *Foundations of Inventory Management*. McGraw-Hill, 2000.