A Continuous-Time Strategic Capacity Planning Model *

Woonghee Tim Huh and Robin O. Roundy

School of Operations Research and Industrial Engineering

Cornell University

Ithaca, NY, 14853, USA

(607) 255-2981

{huh, robin}@orie.cornell.edu

August 23, 2002

Abstract

The sequencing and timing of machine purchases in a semiconductor fabrication facility are challenging because of the vast amount of capital investment, rapidly evolving technologies, and uncertainty in the market. The strategic capacity planning problem is a difficult problem to solve, and there are many opportunities within current practices for improved efficiencies and monetary savings. In contrast to traditional discrete-time models, we present a continuous-time stochastic programming model for multiple resource types and product families. We show how this approach can solve capacity planning problems of reasonable size and complexity with provable efficiency. This is achieved through an application of the divide-and-conquer algorithm, submodularity, and the parametric minimum-cut problem.

*Supported by Semiconductor Research Corporation Task ID: 490.003 and Graduate Fellowship Program; and National Sciences and Engineering Research Council of Canada Postgraduate Scholarship B. An earlier version of this paper has received the First Prize in the Open Category of the Canadian Operational Research Society 2002 Student Paper Competition.

1 Introduction

The semiconductor industry has been one of the driving forces of the "new" economy; in the United States, it creates more value than any other manufacturing industry. The exponentially growing performance of semiconductor devices, coupled with rapidly decreasing chip prices, has fueled incentives for innovation and progress across many sectors. The semiconductor industry, however, continues to face challenges even as it sets the pace of technological advancement. It faces highly volatile demands, and copes with astronomical fab costs, most of which are attributed to tool costs. The lead-time for purchasing tools is between six months and eighteen months, upon which tools quickly become obsolete. Thus, semiconductor companies need to recover capital investment in the tools over a short period of time.

We develop models and algorithms for strategic capacity planning, which is to determine the sequence and timing of acquiring tools. Fabs do not want to purchase expensive tools that would become idle due to the lack of critical or bottleneck machines. Premature tool purchases incur unnecessary high purchase costs and result in over-capacity; whereas tardy purchase decisions lose customer demands (especially at the early stage of a product's life cycle when the margin is highest).

Strategic capacity planning is contrasted with *tactical* planning, which allocates the usage of a pre-determined capacity to a group of operations and products. In *strategic* capacity planning, decisions need to be made well in advance of capacity utilization, and its planning horizon ranges from six months to four years. International Technology Roadmap for Semiconductors (ITRS) Semiconductor Industry Association (1999) has identified the following difficult challenges in the "factory integration" area: complexity management; factory optimization; and extendibility, flexibility and scalability (EFS). Developing models and methods for strategic capacity planning is one of several ways to tackle these challenges.

Strategic planning decisions are made in the presence of high uncertainty. Uncertainty comes from factors such as technology, the market and its products, and becomes amplified by long lead-times. Although capacity planning decisions need to be made in the presence

of high uncertainty, early research and even some current practices overlook the stochastic nature of planning, with the exception of simple case analyses. An extensive review of literature can be found in Çakanyıldırım et al. (1999) and Roundy et al. (2000) Recent papers include Rajagopalan and Yu (2001), Chand et al. (2000), Ryan (1999), and Ryan (2000).

There are only a handful of papers that incorporate stochastic demand in strategic planning. Nearly all models assume continuous increments for capacity, and therefore they do not account for the discrete capacity jumps that result from purchasing a set of tools. Many papers, like Benavides et al. (1999), Berman et al. (1994), Chen et al. (1998) and Li and Tirupati (1994), use simple (such as one-to-one) relationships between product families and tool types, failing to model the complexity of semiconductor manufacturing production.

Typical methods of stochastic optimization include stochastic linear programming, stochastic integer programming, and Markov decisions processes; however, they have not been able to solve real-world capacity planning problems on the scale faced by the semiconductor industry. For example, the Markov decision process model by Bhatnagar et al. (1999) is flexible and general, but it cannot solve complex problems efficiently. The complexity is partially due to the large number of decision variables, many of which are associated with time periods of discrete-time models.

A notably exceptional discrete-time stochastic model for multiple resource types and multiple product families is due to Roundy et al. (2000). It models general product resource requirement relationships. and exhibits a fast algorithm which exploits the maximum-flow and minimum-cut structure. However, this model is for expansion only, and does not easily cope with variations in policies or additional requirements. For multiple resource types and single product family with non-decreasing demand, Çakanyıldırım et al. (1999) have developed an efficient continuous-time model.

This paper takes the stochastic optimization approach, which explicitly incorporates randomness in the model. We assume non-stationary stochastic demand, with the expected demand for product families increasing over time. We also assume lost sales, and no fin-

ished good inventory. As in Çakanyıldırım et al. (1999), we continue to explore alternative approaches based on *continuous-time models*. The time at which a machine is purchased becomes a continuous decision variable. These models are more compact than traditional stochastic programming methods based on discrete-time models. For example, for 50 tools and 20 time periods, these discrete models require 1000 binary variables; whereas, our model uses 50 continuous variables that can take values between 0 and 20. It is hoped that the small dimensionality of continuous time models will make the strategic capacity planning problem computationally tractable.

In this paper, we model multiple resource types used for multiple product families. The resulting problem is related to the continuous relaxation of the lot sizing problem with submodular ordering costs as in Federgruen et al. (1992) and Federgruen and Zheng (1992). We present an efficient divide-and-conquer algorithm which will find a locally optimal solution of this problem. A subroutine to this algorithm is the parametric minimum-cut problem. We also identify certain circumstances under which the planning algorithm finds the globally optimal solution; the continuous-time model under some assumptions becomes a convex program, a provably tractable instance of nonlinear programming.

This paper takes the novel approach of using a continuous-time model with continuous variables for problems that have typically been addressed by discrete-time models with binary variables. We successfully solve a continuous-time version of the discrete-time multiple-product and multiple-resource capacity planning problem presented by Roundy et al. (2000). Our model yields a better solution than the discrete-time model because its solution space is not restricted to a discrete set. As numerical tests show, it runs much faster. The continuous-time model is more likely to allow more modeling extensions than the discrete-time model in Roundy et al. (2000). This issue is further discussed in Section 6. It is a substantial generalization of a single product family model by Çakanyıldırım et al. (1999). One can also view this model as a generalization of a family of lot-sizing problems (e.g. Federgruen and Zheng (1992)). In the context of strategic capacity planning, our model and algorithm can handle problems that are much larger and more complex than has been possible in the

past. This approach can accommodate different assumptions more readily than Roundy et al. (2000).

In Section 2, we describe our model of planning capacity for the multiple-product and multiple-machine manufacturing system, and derive basic properties. A demand model for multiple product families is also described. In Section 3, we present a policy by which we allocate capacity across multiple product families, and show how our model uses a variation of the open-pit mining problem. A divide-and-conquer algorithm for finding a solution with the first-order optimality condition is described in Section 4. This algorithm finds a globally optimal solution under assumptions given in Section 5, which includes computational results. Section 6 concludes this paper.

2 Model

In this section, we provide a mathematical formulation of the strategic capacity planning problem. Due to the high rate of obsolescence, industries such as the semiconductor industry have low finished goods inventory. This model assumes that negligible amounts of finished-goods inventories are held. Motivated by current industry practices, it also assumes that backorders are negligible. These assumptions imply that in the recourse, the production quantities at a given time instance are functions of the capacity and demand at that time instance only, and not at those of other time instances. At time 0, all the capacity acquisition plans are made whereas production decisions are made at each time instance after instantaneous demands have been observed. We use this model as a part of a rolling-horizon implementation.

Section 2.1 presents constraints and the objective function. Section 2.2 develops an alternate expression for the objective function, from which some properties are observed, and Section 2.3 elaborates on how the instantaneous lost sales cost is to be computed. Section 2.4 describes how we model stochastic demand.

2.1 Formulation

We denote by $t \in [0, T]$ a continuous time between 0 and T where T is the planning horizon. We use p and m to index product families $\mathcal{P}$ and tool types $\mathcal{M}$ respectively. For each tool of type $m \in \mathcal{M}$, we let n be its index in the set $\mathcal{N}_m$ of tools of type m in the order that purchases will be made. The ordered set $\mathcal{N}_m$ determines the sequence of tool purchases of type m. We also use $j = (m, n) \in \mathcal{J}$ to index all tools of all types that we contemplate purchasing over the planning horizon.

The price of purchasing tool j at time t is given by a decreasing convex function $P_j(t)$ of t. The instantaneous lost sales cost is c_{pt} per unit of product family p at time t. Let u_{mn} be the capacity of the n'th tool of the tool type m. For any given subset $Q \subseteq \mathcal{J}$ of tools and a given tool group m, let $n = \min\{n' : (m, n') \notin Q\}$, and let the associated tool capacity of the type m tools be $\mu_m(Q) = \sum_{n'<n} u_{mn'}$. The definition of μ_m ensures that tools of the same type should be purchased in the given order because any tool purchased out of sequence does not contribute to the tool capacity of type m. To produce one unit of product family p, we utilize $U(m, p)$ units of capacity from each tool type m.

The decision variables we are interested in are the purchase times $\tau = (\tau_j | j \in \mathcal{J})$ of the tools. We minimize the sum of tool purchase costs and expected lost sales costs. The tool purchase cost is

$$\eta^P(\tau) = \sum_{j=1}^{J} P_j(\tau_j).$$

Let $\xi(Q, t)$ be the expected instantaneous lost sales cost provided that $Q \subseteq \mathcal{J}$ is the subset of tools available at time t. We denote by $Q_t^\tau = \{j : \tau_j \leq t\}$ the set of tools available at time t given purchase times τ. We can write the expected lost sales cost η^{LS} as an integral of instantaneous lost sales cost

$$\eta^{LS}(\tau) \quad = \quad \int_{t=0}^{T} \xi(Q_t^\tau, t) dt. \tag{2.1}$$

The problem we want to solve is the following:

$$\begin{array}{lll} (P) \qquad \min & \eta(\tau) = \eta^P(\tau) + \eta^{LS}(\tau) & \\ \text{s.t.} & 0 \le \tau_j \le T & \text{for all } j \in \mathcal{J}. \end{array}$$

2.2 Properties

In this section, we derive another expression for $\eta^{LS}(\tau)$ within a subset of the feasible region, and develop some properties of η. Let Π be the set of all permutations on $\mathcal{J}$, or bijective maps from $\{1, \ldots, |\mathcal{J}|\}$ to $\mathcal{J}$. Each $\pi \in \Pi$ corresponds to a sequence of tool purchases, and the *permutation simplex* defined by π is

$$PS(\pi) = \{\tau \in [0,T]^{\mathcal{J}} | \tau_{\pi(1)} \le \tau_{\pi(2)} \le \cdots \le \tau_{\pi(|\mathcal{J}|)}\},$$

which corresponds to the set of valid τ's for that sequence. Suppose $\tau \in PS(\pi)$ where $\pi \in \Pi$. For each $r \in \{1, \ldots, |\mathcal{J}|\}$, let $\pi^-(r) = \{\pi(r') | r' < r\}$, and suppose that $Q_t^\tau = \pi^-(r)$. The amount of reduction in the expected instantaneous lost sales cost ξ at time t by adding the tool $\pi(r)$ to the set of available tools is $g_{\pi(r)}^{\pi^-(r)}(t)$. Formally we define, for any $Q^o \subseteq \mathcal{J}$ and $j \in \mathcal{J} \backslash Q^o$,

$$g_j^{Q^o}(t) = \xi(Q^o, t) - \xi(Q^o + j, t). \tag{2.2}$$

Note that $g_j^{Q^o}(t)$ is the difference, in lost sales cost, of having the tool set Q^o and that of having $Q^o + j$ at time t. It is reasonable to expect that the more tools we have, the less expected lost sales cost we incur. We say that the instantaneous lost sales cost ξ is *regular* provided that $\xi(Q^1, t) \ge \xi(Q^2, t)$ for any t whenever $Q^1 \subseteq Q^2 \subseteq \mathcal{J}$. We assume, throughout this paper, that ξ is regular. It follows that $g_j^{Q^o}(t) \ge 0$.

Suppose $\tau \in PS(\pi)$; i.e., τ follows the sequence given by π. Then for fixed t, $Q_t^\tau = \{j \in \mathcal{J} | \tau_j \le t\}$ can be expressed as $\pi^-(r_o) \cup \{\pi(r_o)\}$ for some $r_o \in \{1, \ldots, |\mathcal{J}|\}$. Thus, the telescoping sum of (2.2) implies

$$\xi(Q_t^\tau, t) = \xi(\mathcal{J}, t) + \sum_{r : \tau_{\pi(r)} > t} g_{\pi(r)}^{\pi^-(r)}(t).$$

From (2.1), we obtain

$$\eta^{LS}(\tau) = \int_{t=0}^{T} \xi(\mathcal{J}, t)dt + \sum_{r=1}^{|\mathcal{J}|} \int_{t=0}^{\tau_{\pi(r)}} g_{\pi(r)}^{\pi^-(r)}(t)dt, \qquad \tau \in PS(\pi). \tag{2.3}$$

We note that the first term is a constant. Within the permutation simplex $PS(\pi)$, the expected lost sales cost η^{LS} is continuous and separable. It is also differentiable and its partial derivative with respect to $\tau_{\pi(r)}$ is $\frac{\partial}{\partial \tau_{\pi(r)}}\eta^{LS}(\tau) = g_{\pi(r)}^{\pi^-(r)}(\tau_{\pi(r)})$, in the interior of $PS(\pi)$. Furthermore, η^{LS} is continuously differentiable if each g_r^{π}, $r \in \{1, \ldots, |\mathcal{J}|\}$, is continuous with respect to τ.

Whenever $\pi(r) = j$ and $\pi^-(r) = Q$, we have $\frac{\partial}{\partial \tau_j}\eta^{LS}(\tau) = g_j^Q(\tau_j)$. This is a much stronger separability of the expected lost sales cost η^{LS} than separability in each permutation simplex. We generalize the definition (2.2) of g: for any disjoint sets $Q^o, Q \subseteq \mathcal{J}$ of tools, we define

$$g_Q^{Q^o}(t) = \xi(Q^o, t) - \xi(Q^o \cup Q, t). \tag{2.4}$$

This quantity corresponds to the marginal benefit of adding the tool set Q to the existing set Q_o at time t. For $Q^1, Q^2 \subseteq \mathcal{J}$, we say $Q^1 <_\pi Q^2$ provided $\pi^{-1}(j_1) < \pi^{-1}(j_2)$ for any $j_1 \in Q^1$ and $j_2 \in Q^2$. It follows easily that for any $\pi \in \Pi$ satisfying $Q^o <_\pi Q <_\pi \mathcal{J}\backslash(Q^o \cup Q)$, we get

$$g_Q^{Q^o}(t) = \sum_{r'=|Q^o|+1}^{|Q^o \cup Q|} g_{\pi(r')}^{\pi^-(r')}(t) = \sum_{j \in Q} \frac{\partial}{\partial \tau_j}\eta^{LS}(\tau), \tag{2.5}$$

where partial derivatives are taken in $PS(\pi)$, and $\tau_j = t$ for all $j \in Q$. Like (2.4), (2.5) is a strong additivity property of derivatives of η^{LS} that spans many neighboring permutation simplices. A direct consequence of (2.5) is

$$g_{Q^1}^{Q^o}(t) + g_{Q^2}^{Q^o \cup Q^1}(t) = g_{Q^1 \cup Q^2}^{Q^o}(t) \text{ if } Q^o, Q^1, Q^2 \subseteq \mathcal{J} \text{ are disjoint.} \tag{2.6}$$

The remainder of this section presents some properties of the objective function related to the divide-and-conquer method in Section 4. We introduce some new notation. We let $h_j(t) = \frac{d}{dt}P_j(t) \leq 0$ be the rate of change in the tool cost at t. By the convexity of the tool cost P_j, $h_j(t)$ is nonnegative. For $Q \in \mathcal{J}$, we set $h_Q(t) = \sum_{j \in Q} h_j(t)$. We remark that within

the permutation simplex $PS(\pi)$ defined by π, the objective function η is separable and its partial derivative with respect to j is $\frac{\partial}{\partial \tau_j}\eta(\tau) = h(\tau_j) + g_{\pi^{-}(r)}(\tau_j)$.

Suppose at time t, we have a partition Q_L, Q_o and Q_U of $\mathcal{J}$ where Q_L is the set of tools we have purchased prior to t, and Q_U is the set of tools we will purchase after t. Currently, we purchase tools in Q_o at t. If we split Q_o, and uniformly slide $Q \subseteq Q_o$ earlier and $Q_o \backslash Q$ later, then η changes at the rate of

$$\varphi_t(Q|Q_L, Q_o, Q_U) = -[h_Q(t) + g_Q^{Q_L}(t)] + [h_{Q_o \backslash Q}(t) + g_{Q_o \backslash Q}^{Q_L \cup Q}(t)]. \tag{2.7}$$

From (2.6), the above expression can be rewritten as

$$\varphi_t(Q|Q_L, Q_o, Q_U) \;=\; -2[h_Q(t) + g_Q^{Q_L}(t)] + [h_{Q_o}(t) + g_{Q_o}^{Q_L}(t)], \tag{2.8}$$

or, equivalently as

$$\varphi_t(Q|Q_L, Q_o, Q_U) \;=\; -[h_{Q_o}(t) + g_{Q_o}^{Q_L}(t)] + 2[h_{Q_o \backslash Q}(t) + g_{Q_o \backslash Q}^{Q_L \cup Q}(t)]. \tag{2.9}$$

Selecting Q as to minimize the above function φ is called the *cluster splitting* problem. The following proposition shows that splitting Q_o so as to minimize the above function is good – if Q minimizes φ_t, and if we partition Q_o by setting τ_Q to $t - \varepsilon$, and $\tau_{Q_o \backslash Q}$ to $t + \varepsilon$, then no subset of either set will want to cross t. In Section 4.2, this proposition is used to establish how the cluster splitting problem justifies the divide-and-conquer method.

Proposition 2.2.1. *For fixed t, let Q^* minimize φ_t. Then we have*

$$h_B(t) + g_B^{Q_L \cup Q^* \backslash B}(t) \geq 0 \text{ for } B \subseteq Q^*; \text{ and } h_B(t) + g_B^{Q_L \cup Q^*}(t) \leq 0 \text{ for } B \subseteq Q_o \backslash Q^*.$$

Proof. For the simplicity of exposition, we show only the first result, and that result only for $Q_o = \mathcal{J}$ and $Q_L = Q_U = \emptyset$. Let $\bar{B} = Q^* \backslash B$. By the optimality of Q^* and (2.6),

$$\begin{aligned}
0 \;&\geq\; \varphi_t(Q^*|\emptyset, \mathcal{J}, \emptyset) - \varphi_t(\bar{B}|\emptyset, \mathcal{J}, \emptyset) \\
&=\; [-h_{Q^*}(t) - g_{Q^*}^{\emptyset}(t) + h_{\mathcal{J} \backslash Q^*}(t) + g_{\mathcal{J} \backslash Q^*}^{Q^*}(t)] - [-h_{\bar{B}}(t) - g_{\bar{B}}^{\emptyset}(t) + h_{\mathcal{J} \backslash \bar{B}}(t) + g_{\mathcal{J} \backslash \bar{B}}^{\bar{B}}(t)] \\
&=\; -[g_{Q^*}^{\emptyset}(t) - g_{\bar{B}}^{\emptyset}(t)] - [g_{\mathcal{J} \backslash \bar{B}}^{\bar{B}}(t) - g_{\mathcal{J} \backslash Q^*}^{Q^*}(t)] - [h_{Q^*}(t) - h_{\bar{B}}(t)] - [h_{\mathcal{J} \backslash \bar{B}}(t) - h_{\mathcal{J} \backslash Q^*}(t)] \\
&=\; 2[-g_B^{Q^* \backslash B}(t) - h_B(t)],
\end{aligned}$$

indicating that $h_B(t) + g_B^{Q^* \backslash B}(t) \geq 0$. □

In general, finding the minimizer of φ_t may not be easy. Using explicit enumeration takes $O(2^{|Q_o|})$ computational time. In Section 3.3 we show, under our modeling assumptions, how to minimize φ_t efficiently.

2.3 Determination of Instantaneous Lost Sales Cost

This section explains how we determine the expected value ξ of the instantaneous lost sales cost. The lost sales at time t depend on demands for product families at time t, capacities of tool types at time t, and the allocation of tool capacities to product families. The next section describes how we model stochastic demand. Given a set Q of tools which are available at time t (which is determined by τ), tool type m's capacity is given by $\mu_m(Q)$. Whereas the tool purchase times τ are all determined at the beginning of the horizon, we allow for the *dynamic* allocation of tools. Given the capacity $\mu(Q) = (\mu_m(Q)|m \in \mathcal{M})$ of all tool types and the *realized* demand $d_t = (d_{pt}|p \in \mathcal{P})$ of all product families at time t, we determine both the production quantity $v_t = (v_{pt}|p \in \mathcal{P})$ of product family p and the allocation $x_t = (x_{mpt}|m \in \mathcal{M}, p \in \mathcal{P})$ of tool type m's capacity to p. A *capacity allocation policy* is a way of selecting x_t and v_t.

As in Çakanyıldırım et al. (1999) and Roundy et al. (2000), we assume no finished goods inventory, and no backorder. In other words, demand at time t can be satisfied by what is produced at time t only. Thus, in any capacity allocation policy, production should not exceed demand, i.e.,

$$v_{pt} \leq d_{pt} \text{ for all } p \in \mathcal{P}. \tag{2.10}$$

Production v and allocation x must obey the capacity limit of each tool type:

$$\sum_{p=1}^{P} x_{mpt} \leq \mu_m(Q) \text{ for all } m \in \mathcal{M} \text{ and } t \in [0,T] \tag{2.11}$$

$$U(m,p)v_{pt} \leq x_{mpt} \text{ for all } p \in \mathcal{P} \text{ and } t \in [0,T]. \tag{2.12}$$

It is noted that a capacity allocation policy may impose further constraints on x and v. (See Section 3.1, for example.) Any capacity allocation policy defines functions x and v in

terms of Q (which depends on τ) and d. These functions, in general, are neither simple nor algebraic, which make analysis difficult. In Section 3.1, we present a capacity allocation policy which makes these dependencies tractable.

The lost sales is the difference between demand and production. At time t, the lost sales l_{pt} of product family p is

$$l_{pt} = d_{pt} - v_{pt}. \tag{2.13}$$

Let production $V_t = (V_{pt}|p \in \mathcal{P})$ and lost sales $L_t = (V_{pt}|p \in \mathcal{P})$ be random vectors corresponding to v_t and l_t respectively. Then we can write

$$\xi(Q,t) \;\; = \;\; E_{D_t}[\sum_{p\in\mathcal{P}} c_{pt}L_{pt}]. \tag{2.14}$$

We remark that L_{pt} depends on demand D_t, and also on tool availability Q through (2.11) and (2.12).

2.4 Demand Modeling

As in Roundy et al. (2000), we model the random demand vector D_t at time t as a sum of a deterministic part and a stochastic part, i.e.,

$$D_t = b_t + \Delta_{I_t,t}\phi_{I_t,t}, \tag{2.15}$$

where $b_t = (b_{pt}|p \in \mathcal{P}) \in \mathbb{R}^{\mathcal{P}}$ is a deterministic nonnegative vector which is nondecreasing in t; I_t is a discrete random variable whose support is a finite set $\mathcal{I}_t$ such that $P[I_t = i] = w_{it}$ for each $i \in \mathcal{I}_t$; $\phi_{it} = (\phi_{ipt}|p \in \mathcal{P})$ is a deterministic non-negative unit-norm directional vector in $\mathbb{R}^{\mathcal{P}}$; and Δ_{it} is a continuous non-negative random scalar along ϕ_{it}. Intuitively, the demand D_t is determined by starting at b_t, randomly selecting a direction by observing I_t, and moving a random distance $\Delta_{I_t,t}$ in the direction $\phi_{I_t,t}$

Currently, most models of high dimensional random vectors are either continuous (e.g. multi-variable normal) or discrete (e.g. multinomial). Our demand model is a hybrid of both: no point in $\mathbb{R}^{\mathcal{P}}$ has any nonzero probability mass. The support of D_t is a finite collection

of rays emanating from b_t, and has measure zero. It is shown in Roundy et al. (2000) that by a variance-reduction technique called conditioning, our demand model can approximate a continuous distribution in $\Re^{\mathcal{P}}$ more accurately than the conventional method of sampling points, provided that the number of vectors is the same as the number of points. As we shall see, useful theoretical and algorithmic properties follow. As the number of rays increases, we have a better approximation of a continuous distribution. Numerical results in Roundy et al. (2000) indicate that in a 4-dimensional space, 64 rays provided an approximation to a multi-variable log normal distribution that was sufficiently accurate for a capacity planning problem. Growth in the required number of rays as the number $|\mathcal{P}|$ of product families seems moderate.

There is no demand shortfall if the capacity $\mu_m(Q_t^\tau)$ is sufficient to meet the demand d_t, i.e.

$$\sum_{p=1}^{P} U(m,p)d_{pt} \leq \mu_t(Q_t^\tau) \text{ for all } m = 1, \cdots, M.$$

Otherwise, we are unable to meet all demands. The following section outlines a policy we use to allocate insufficient capacity to product families.

3 Uniform Fill-Rate Production

In Section 3.1, we introduce a specific capacity allocation policy that assigns capacity to product families in the case of demand shortfall. An extension of the classical open-pit mining problem, described in Section 3.2, is used in Section 3.3 to solve the cluster-splitting problem. Section 3.4 shows how cluster-splitting is related to the optimality of our problem.

3.1 Description and Reformulation

When the capacity is insufficient in meeting realized demand $d_t = (d_{pt}|p \in \mathcal{P})$, we need a capacity allocation policy in order to determine the production quantities $v_t = (v_{pt}|p \in \mathcal{P})$, which in turn determines the lost sales $l_t = (l_{pt}|p \in \mathcal{P})$. One plausible way of determining l_t

is to minimize the total instantaneous lost sales by solving an allocation linear program: to minimize $\sum_{p\in\mathcal{P}} c_{pt} l_{pt}$ subject to constraints (2.10) through (2.13). IBM uses a similar linear program to reflect the available manufacturing capacity. (See Bermon and Hood (1999).) However, this approach suffers from two consequences; one is a modeling issue, and the other is an analytical and algorithmic issue. The first consequence is that the production quantities of certain product families, especially those with low profitability, are much more sensitive to the total capacity availability than those with high profitability. The importance of this consequence is dependent upon context. The second consequence is that the objective function η does not admit a structure that enables an efficient solution method.

We conceptually derive the demand into a deterministic portion $b_t \geq 0$ and a stochastic portion $\Delta_{I_t,t} \cdot \phi_{I_t,t} \geq 0$. We assume that there is enough capacity to meet the deterministic part b_t of the demand. We may ensure this assumption by imposing upper bounds on purchase times τ. Since $D_t \geq b_t$, our allocation policy meets the deterministic part b_t of demand before allocating resources to the stochastic part.

For the rest of this paper, we use an allocation policy which determines production quantities v_t that equalizes the instantaneous fill rates of stochastic portion of demand at time t across all products. In the recourse at time t, after the demand $d_{it} = b_t + \delta_{it}\phi_{it}$ is realized, this implies that we select production quantities

$$v_t = b_t + \zeta\phi_{it} \text{ for some } \zeta \in [0, \delta_{it}]. \tag{3.16}$$

Thus, v_t also lies on the ray defined by the starting point b_t, and the direction ϕ_{it}. The value ζ indicates the magnitude of production along this ray. It is easy to see that the fill-rate of the stochastic part for product p is $(v_{pt} - b_{it})/(d_{ipt} - b_{it}) = \zeta/\delta_{it}$, which is independent of the product family p. If $b_t = 0$ then this corresponds to the classical fill rate.

To simplify our notation, we proceed by assuming $b_t = 0$. Thu,s equations (2.15) and (3.16) become

$$D_t = \Delta_{I_t,t}\phi_{I_t,t}, \text{ and} \tag{3.17}$$

$$v_t = \zeta\phi_{it} \text{ for some } \zeta \in [0, \delta_{it}]. \tag{3.18}$$

The allocation of capacity to meet these production quantities is $x_{mpt} = U(m,p)v_{pt}$. After the demand has been realized, we choose ζ as to maximize production along the ray without exceeding the capacity of Q_t^τ; i.e., we select $\zeta^* = \min\{\zeta_{it}^{Q_t^\tau}, \delta_{it}\}$, where

$$\zeta_{it}^{Q} = \max\{\zeta \geq 0 | \zeta \sum_{p \in \mathcal{P}} U(m,p)\phi_{ipt} \leq \mu_m(Q) \text{ for all } m \in \mathcal{M}\}, \qquad Q \subseteq \mathcal{J}.$$

We remark that ζ_{it}^{Q} represents the maximum demand magnitude along ϕ_{it} that tool set Q can support. This capacity allocation policy is called the *uniform fill-rate production* policy. This policy ensures that no product family has a fill-rate lower than other product families. It approximates the current practice of at least one major U.S. semiconductor manufacturer. We will show that this policy has many interesting and useful properties, some of which enable us to use an efficient divide-and-conquer method.

Having established the capacity allocation policy, we want to find the instantaneous lost sales cost ξ under our demand model introduced in Section 2.4. Suppose we fix the demand direction ϕ_{it}; i.e., we condition on $I_t = i$ in (3.17). Along this demand ray, any tool set that does not contain tool $j = (m,n)$ cannot support a production vector v_t whose magnitude $|v_t| = \zeta$ is greater than

$$\beta_{it}(m,n) = \max\{\zeta \geq 0 | \zeta \sum_{p \in \mathcal{P}} U(m,p)\phi_{ipt} \leq \sum_{n' < n} u_{mn'}\}.$$

It is easy to see $\zeta_{it}^{Q} = \min_m\{\beta_{it}(m,n_m)\}$ where $n_m \in \mathcal{N}_m$ is the largest index such that Q contains the first n_m tools of type m; i.e., $(m,n') \in Q$ for all $n' \leq n_m$, and $(m,n_m+1) \notin Q$.

Along a fixed demand ray ϕ_{it}, each tool (m,n) constrains production capacity along this ray at a certain magnitude $\beta_{it}(m,n)$ We use $\psi_{it} : \{1,\ldots,|\mathcal{J}|\} \to \mathcal{J}$ to define a sequence of tools in the order in which they constrain capacity along demand ray ϕ_{it}; thus $\beta_{it}(\psi_{it}(1)) \leq \beta_{it}(\psi_{it}(2)) \leq \cdots \leq \beta_{it}(\psi_{it}(|\mathcal{J}|))$. Purchasing tool $\psi_{it}(r)$ before $\psi_{it}(r-1)$ does not contribute to the maximum magnitude along ϕ_{it} that a tool set can support. If Q_1 is the largest set of the form $\{\psi_{it}(1), \psi_{it}(2), \ldots, \psi_{it}(r)\}$, $r \in \{1,\ldots,|\mathcal{J}|\}$, that is a subset of Q, then Q_1 is the *effective tool subset* of Q for demand ray i, that contributes to the capacity magnitude along the demand ray ϕ_{it} at time t; i.e. $\zeta_{it}^{Q} = \zeta_{it}^{Q_1}$.

Before the demand $D_t = \Delta_{I_t,t}\phi_{I_t,t}$ is realized, production $V_t = (\min\{\zeta_{it}^{Q_t^\tau}, \Delta_{it}\})\phi_{I_t,t}$ is random. Equations (2.13) and (2.14) show

$$\xi(Q_t^\pi, t) = E_{D_t} \sum_{p\in\mathcal{P}} c_{pt} L_{pt} = \sum_{i\in\mathcal{I}_t} w_{it} \sum_{p\in\mathcal{P}} c_{pt}\phi_{ipt} E_{\Delta_{it}}[\Delta_{it} - \zeta_{it}^{Q_t^\tau}]^+. \quad (3.19)$$

$$= \sum_{i\in\mathcal{I}_t} w_{it} \sum_{p\in\mathcal{P}} c_{pt}\phi_{ipt} E_{\Delta_{it}}[\Delta_{it} - \zeta_{it}^{\{\psi_{it}(1),\ldots,\psi_{it}(r_i)\}}]^+ \quad (3.20)$$

where $\{\psi_{it}(1), \ldots, \psi_{it}(r_i)\}$ is the effective tool subset of Q_t^τ for demand ray i. The expression $E_{\Delta_{it}}[\Delta_{it} - K]^+$ for any fixed scalar K, is a common function in the inventory theory. It is easy to evaluate (3.20).

We assume that each item in the righthand side of (3.20) is continuous in t for any $Q \subseteq \mathcal{J}$. This implies that η is continuously differentiable in each permutation simplex.

3.2 Open-pit Tooling Problem

In this section, we present an extension to the classical open-pit mining problem in the network flow theory, which we will use later in Section 3.3 to solve the cluster splitting problem introduced in Section 2.2. We will prove two basic propositions of this extension. This section can be read independent of any other part of this paper.

Suppose we are given a set B of blocks, and a precedence set $A \subseteq B \times B$ of blocks. The precedence set A is acyclic, and $(b_1, b_2) \in A$ indicates that block b_2 cannot be excavated unless block b_1 is also excavated. For each block $b \in B$, the net loss ϱ_b associated with block b is the cost of excavating block b minus the revenue generated by the ore found in block b. Thus, the net profit associated with a pit $R \subseteq B$ is $-\sum_{b\in R} \varrho_b$. The open-pit mining problem is to find a set $R \subseteq B$ of blocks to be excavated as to minimize $\sum_{b\in R} \varrho_b$. The set R must be *initial*, which means there does not exist any $(b_1, b_2) \in A$ with $b_1 \notin R$ and $b_2 \in R$. This problem is easily solved via a minimum-cut problem. (See Chavátal (1983).)

Now we make the following modification to the open-pit mining problem. In addition to B and A, there is a set W of tools that can be purchased. Let ς_w be the purchase cost of tool $w \in W$. Associated with each block b is a set $\phi(b) \subseteq W$ of tools required to excavate

that block. Having purchased a set $Q \subseteq W$ of tools, the set of blocks that can be excavated is $B(Q) = \{b \in B : \phi(b) \subseteq Q\}$. If we fix the set Q of tools to be purchased and we restrict R to a subset of $B(Q)$, then we obtain a corresponding open-pit mining problem $\mathbf{P}^Q$. In this problem the total cost, consisting of the tool purchase cost and the net loss of excavation, is

$$C(Q) = \sum_{w \in Q} \varsigma_w + \min_{R \subseteq B(Q) \text{ initial}} \sum_{b \in R} \varrho_b, \tag{3.21}$$

where $\bar{Q} = W \backslash Q$. We want to find a subset Q of W as to minimize $C(Q)$. We say this problem is the *open-pit tooling problem.*

The following proposition demonstrates a property of the objective function. For any set $\mathcal{W}$, a real valued function ρ on $2^{\mathcal{W}}$ is *submodular* provided $\rho(Q_1) + \rho(Q_2) \geq \rho(Q_1 \cup Q_2) + \rho(Q_1 \cap Q_2)$ for all $Q_1, Q_2 \subseteq \mathcal{W}$. Equivalently, ρ is submodular if for any $Q_o \subseteq \mathcal{W}$ and $w_1, w_2 \in \mathcal{W}$, we have $\rho(Q_o + w_1) + \rho(Q_o + w_2) \geq \rho(Q_o + w_1 + w_2) + \rho(Q_o)$. We also say that ρ is *modular* if the above inequality is replaced with the equality.

Proposition 3.2.1. *The cost function $C(\cdot)$ is submodular.*

Proof. Let $Q \subseteq W$ and $i, j \in W \backslash Q$ such that $i \neq j$. We first observe that the union and intersection of $B(Q + i)$ and $B(Q + j)$ are subsets of $B(Q + i + j)$ and $B(Q)$ respectively.

Since the first term in (3.21) is modular, it suffices to show that the sum of the optimal values of $\mathbf{P}^{Q+i}$ and $\mathbf{P}^{Q+j}$ is at least the sum of the optimal values of $\mathbf{P}^{Q+i+j}$ and $\mathbf{P}^Q$. Let R^{Q+i} and R^{Q+j} be the minimizers of $\mathbf{P}^{Q+i}$ and $\mathbf{P}^{Q+j}$ respectively. Then from the observation, we conclude that $R^{Q+i+j} = R^{Q+i} \cup R^{Q+j}$ is a subset of $B(Q + i + j)$ and $R^Q = R^{Q+i} \cap R^{Q+j}$ is a subset of $B(Q)$. Since R^{Q+i} and R^{Q+j} are initial, and the union and the intersection of initial sets are also initial, it follows that R^{Q+i+j} and R^Q are initial. Thus, R^{Q+i+j} and R^Q are feasible in $\mathbf{P}^{Q+i+j}$ and $\mathbf{P}^Q$ respectively. From

$$\sum_{b \in R^{Q+i}} \varrho_b + \sum_{b \in R^{Q+j}} \varrho_b = \sum_{b \in R^{Q+i+j}} \varrho_b + \sum_{b \in R^Q} \varrho_b,$$

we obtain the required result. □

Since each evaluation of $C(\cdot)$ can be done efficiently using the minimum-cut algorithm, it is possible to use a submodular function minimization algorithm to solve the open-pit

tooling problem in polynomial time. Minimizing a general submodular function can be done in polynomial time, but, it takes a prohibitively long time. The following result shows that the open-pit tooling problem can be efficiently solved.

Proposition 3.2.2. *The open-pit tooling problem reduces to an open-pit mining problem.*

Proof. Suppose that the open-pit tooling problem is given by parameters B, ϱ, A, W, ς and ϕ. Let $B' = B \cup W$. We assign net loss $\varrho'_b = \varrho_b$ if $b \in B$, and $\varrho'_b = \varsigma_w$ if $b = w \in W$. The new precedence set $A' = A \cup \{(w, b) : w \in W \text{ and } b \in B \text{ such that } w \in \phi(b)\}$ is obtained by adding to A the relationships between blocks and their required tools. Then, the set of parameters B', ϱ' and A' defines an open-pit mining problem equivalent to the given open-pit tooling problem. □

We conclude this section with the following observation. In the minimum-cut problem to which we eventually reduce an open-pit tooling problem, all the arcs with the finitely capacity are incident with either source or sink. Furthermore, suppose we parameterize the costs $\varrho_b = \varrho_b(t), b \in B$ and $\varsigma_w = \varsigma_w(t), w \in W$ as functions of t, and obtain a *parametric minimum-cut network*. If the incoming arcs into the sink have nondecreasing capacities, and the outgoing arcs from the source have nonincreasing capacities in t, the parametric minimum-cut network becomes *monotonic* (Gusfield and Martel (1992)). This observation is useful in devising an efficient divide-and-conquer algorithm, as we will see in the following sections.

3.3 Solving Cluster Splitting Problem

We have seen results indicating that the function η behaves well within a permutation simplex. For example, it is easy to find a descent direction at an interior point of a permutation simplex. When the current solution lies along the boundary of a permutation simplex, some of the inequalities defining the permutation simplex are tight, and the solution belongs to more than one permutation simplex simultaneously. Thus, in order to find a descent direction, we want to solve the cluster splitting problem presented in Section 2.2. In this section,

we present an efficient way of solving the cluster splitting problem under uniform fill-rate production (Section 3.1). We achieve this by reducing the cluster splitting problem to the open-pit tooling problem (Section 3.2).

Even though $\eta(\tau)$ is separable inside each permutation simplex, in general it is neither separable nor continuously differentiable across permutation simplices. The following definition of the *directional derivative* of η at τ with respect to a feasible direction y is conventional:

$$\eta'(\tau; y) = \lim_{\varepsilon \to 0^+} \frac{\tau(x + \varepsilon y) - \tau(x)}{\varepsilon}.$$

We say a feasible direction y is a *descent direction* provided $\eta'(\tau; y) < 0$. We define a *cluster* $J_t(\tau)$ of τ at t as the set $\{j : \tau_j = t\}$ of tools whose corresponding τ values are t.

We claim that the directional derivatives of η are separable by cluster. We know η^P is separable. Let $\{J^k = J_{t_k}(\tau), k = 1, \ldots, K\}$ be a set of pairwise disjoint clusters that partition $\mathcal{J}$ such that $t_1 < t_2 < \cdots < t_K$. For any feasible direction $y \in \mathbb{R}^{\mathcal{J}}$, we can write $y = \sum_{k=1}^{K} y^k$ where $y^k \in \mathbb{R}^{\mathcal{J}}$ may have nonzero values only at components associated with J^k; i.e. $y^k_j = 0$ for all $j \in \mathcal{J} \backslash J^k$. Let $\pi \in \Pi$ such that for sufficiently small $\varepsilon > 0$, $\tau + \varepsilon y$ is in the permutation simplex $PS(\pi)$ defined by π. We have $J^1 <_\pi \cdots <_\pi J^K$. Since η^{LS} is separable and differentiable within $PS(\pi)$, equation (2.5) implies

$$\eta'(\tau; y) = \sum_{k=1}^{K} h_{J_k}(t_k) + g_{J_k}^{J_1 \cup \cdots \cup J_{k-1}}(t_k) = \sum_{k=1}^{K} \eta'(\tau; y^k). \tag{3.22}$$

As a result, to find the directional derivative of η at τ along any direction y, it suffices to look for a descent direction in one cluster at a time.

For simplicity of argument, in this section, we assume without loss of generality that *there is only one cluster and it is located at* t_o; i.e., $\tau_j = t_o$ for each $j \in \mathcal{J}$, or $J_{t_o}(\tau) = \mathcal{J}$. We assume that t_o is in the interior of $[0, T]$. The splitting problem is to choose $Q \subseteq \mathcal{J}$ as to minimize φ, which becomes by (2.8)

$$\varphi_{t_o}(Q) = \varphi_{t_o}(Q|\emptyset, \mathcal{J}, \emptyset) = -2[h_Q(t_o) + g_Q^{\emptyset}(t_o)] + [h_{\mathcal{J}}(t_o) + g_{\mathcal{J}}^{\emptyset}(t_o)].$$

We recall Q is the set of tools whose purchase times are perturbed to the left (earlier) and $\mathcal{J} \backslash Q$ is the set of tools whose purchase times are perturbed to the right (later). Since

$h_{\mathcal{J}}(t_o) + g^{\emptyset}_{\mathcal{J}}(t_o)$ is independent of Q, our problem is equivalent to finding $Q \subseteq \mathcal{J}$ minimizing $-h_Q(t_o) - g^{\emptyset}_Q(t_o) = \eta'(\tau, -\chi_Q)$, where $\chi_Q \in \{0,1\}^{\mathcal{J}}$ is a binary indicator vector of Q. We reduce this problem to the open-pit tooling problem.

Suppose we want to purchase a set $Q \subseteq \mathcal{J}$ of tools earlier by $\varepsilon > 0$. Then the rate of increase in the purchase cost η^P is $-\sum_{j \in Q} \frac{dP_j(t)}{dt}|_{t=t_o} = -h_Q(t_o)$, and the rate of decrease in the expected lost sales cost η^{LS} is $g^{\emptyset}_Q(t_o) = \xi(\emptyset, t_o) - \xi(Q, t_o)$. (See (2.4).) By (3.20), the cluster splitting problem reduces to the open-pit tooling problem in which blocks in the initial set corresponds to effective tool subset for each demand ray. The parameters of the reduced open-pit tooling problem are:

$$
\begin{aligned}
B &= \{(i,r) | i \in I_{t_o}, r \in \{1, \ldots, |\mathcal{J}|\}\}, \\
\varrho_{(i,r)} &= w_{it_o} \sum_{p \in \mathcal{P}} c_{pt} \phi_{ipt_o} E_{\Delta_{it_o}}[(\Delta_{it_o} - \zeta_{it_o}^{\{\psi_{it_o}(1), \ldots, \psi_{it_o}(r)\}})^+ - (\Delta_{it_o} - \zeta_{it_o}^{\{\psi_{it_o}(1), \ldots, \psi_{it_o}(r-1)\}})^+] \leq 0, \\
&\quad \text{for each } r \in \{1, \ldots, |\mathcal{J}|\} \\
A &= \{((i, r-1), (i, r)) | r \in \{2, 3, \ldots, |\mathcal{J}|\}\}, \\
W &= \mathcal{J}, \\
\varsigma_w &= -\frac{dP_w(t)}{dt}|_{t=t_o} = -h_w(t_o) \geq 0, \text{ for each } w \in W, \text{ and} \\
\phi(w) &= \{(i,r) | \psi_i(r) = w\}, \text{ for each } w \in W.
\end{aligned}
$$

The objective function becomes $C(Q) = -h_Q(t_o) - g^{\emptyset}_Q(t_o)$, for $Q \subseteq W$. Any feasible solution of this open-pit tooling problem corresponds to a feasible solution of the cluster-splitting problem. An optimal solution of either problem optimizes both problems. If the optimal value of the open-pit tooling problem is at least $-\frac{1}{2}[h_{\mathcal{J}}(t_o) + g^{\emptyset}_{\mathcal{J}}(t_o)]$, then by (2.8), there is no way to split the cluster that corresponds to a descent direction.

We remark that in the maximum-flow minimum-cut flow network to which the open-pit tooling problem eventually is reduced, the number of nodes and the number of arcs are proportional to the product of the number $|\mathcal{J}|$ of tools and the maximum number $|\mathcal{I}|$ of rays used in modeling demand.

The submodularity result of the open-pit tooling problem implies the following result:

Proposition 3.3.1. *The objective function $\varphi_{t_o}(\cdot)$ is submodular.*

Proof. The objective function $C(Q) = -h_Q(t_o) - g_Q^{\emptyset}(t_o)$ of the open-pit tooling problem is submodular in $Q \subseteq \mathcal{J}$ by Proposition 3.2.1. Therefore, from (2.8), we see that $\varphi_{t_o}(Q)$ is submodular in Q. □

This result is used in the next section to show a property of directional derivatives of η.

3.4 Cluster Splitting and Optimality

Given a current solution τ such that $\tau_j = t_o$ for each $j \in \mathcal{J}$, cluster splitting checks the descent of the objective function η in $2^{|\mathcal{J}|}$ directions, each of which corresponds to the splitting of the cluster into at most two clusters. It may be plausible that while splitting it into two clusters finds no descent direction in η, splitting into three or more clusters may find one. This section however shows that this is not the case. The following theorem shows that there is a descent direction from a cluster if and only if the cluster splitting problem has a negative optimal value.

Theorem 3.4.1. *Given $\tau \in [0,T]^{\mathcal{J}}$ and $t_o \in [0,T]$, the cluster splitting problem of cluster $J_{t_o}(\tau)$ has a nonnegative optimal value, i.e.*

$$\varphi_{t_o}(Q) \geq 0 \text{ for all } Q \subseteq J_{t_o}(\tau),$$

if and only if

$$\eta'(\tau; y) \geq 0,$$

for all $y \in \mathbb{R}^{\mathcal{J}}$ such that $y_j = 0$ for all $j \notin J_{t_o}(\tau)$.

Proof. Without loss of generality, assume $\tau_j = t_o$ for all $j \in \mathcal{J}$. (See (3.22) and the ensuing discussion.) Suppose $\varphi_{t_o}(Q) \geq 0$ for all $Q \subseteq \mathcal{J}$. Then, in particular, $\varphi_{t_o}(\emptyset) \geq 0$ and $\varphi_{t_o}(\mathcal{J}) \geq 0$ imply that $h_{\mathcal{J}}(t_o) + g_{\mathcal{J}}^{\emptyset}(t_o) = 0$. Consequently, it follows from the expression (2.9) for $\varphi_{t_o}(\bar{Q}) = \varphi_{t_o}(\bar{Q}|\emptyset, \mathcal{J}, \emptyset)$, $\bar{Q} = \mathcal{J} \backslash Q$, that

$$\eta'(\tau; \chi_Q) = [h_Q(t_o) + g_Q^{\bar{Q}}(t_o)] = \frac{1}{2}\varphi_{t_o}(\bar{Q}) \geq 0, \qquad Q \subseteq \mathcal{J}.$$

Now we prove the result for general $y \in \mathbb{R}^{\mathcal{J}}$. Let $\pi \in \Pi$ be such that y is in its permutation simplex $PS(\pi)$, and set $Q^r = \{\pi(r), \pi(r+1), \ldots, \pi(|\mathcal{J}|)\}$, for $r \in \{1, \ldots, |\mathcal{J}|\}$. Then there exist multipliers $\beta_r \in \mathbb{R}$, $r \in \{1, \ldots, |\mathcal{J}|\}$, such that $y = \sum_{r=1}^{|\mathcal{J}|} \beta_r \chi_{Q^r}$, and all β_r's are nonnegative except possibly for β_1. By (3.22),

$$\eta'(\tau; \sum_{r=0}^{|\mathcal{J}|} \beta_r \chi_{Q^r}) = \sum_{r=0}^{|\mathcal{J}|} \beta_r \eta'(\tau; \chi_{Q^r}).$$

Since $\eta'(\tau; \chi_{Q^r})$ is nonnegative for all Q^r and vanishes for $Q^1 = \mathcal{J}$, we conclude that the above expression is nonnegative. □

We introduce a classical result of Lovász, that relates submodularity and convexity. For the set function φ_{t_o} on $2^{\mathcal{J}}$, we define a *linear extension* $\hat{\varphi}_{t_o} : [0,1]^{\mathcal{J}} \to \mathbb{R}$ such that

i. $\hat{\varphi}_{t_o}(\chi_Q) = \varphi_{t_o}(Q)$ for any $Q \subseteq \mathcal{J}$, and

ii. For any z in the permutation simplex $PS(\pi)$ defined by some $\pi \in \Pi$, let $\hat{\varphi}_{t_o}(z)$ be the value at z of the hyperplane defined by $\{\chi_{\pi(\{1,2,\ldots,r\})} | r = 0, 1, \ldots, N\}$.

The following proposition relates submodularity and convexity.

Proposition 3.4.2. *(Lovász (1983)) Any set function is submodular if and only if its linear extension is convex.*

Proposition 3.4.3. *For any $\tau \in [0,T]^{\mathcal{J}}$ and $y \in \mathbb{R}^{\mathcal{J}}$, we have*

$$\eta'(\tau; y) + \eta'(\tau; -y) \geq 0,$$

whenever the first two terms are defined.

We remark that since η is differentiable in each permutation simplex, the above expression holds with equality if τ is an interior point of a permutation simplex. This proposition examines the behavior of the directional derivative as it crosses the boundary of permutation simplices, and shows that it is nondecreasing across the boundary.

Proof. Without loss of generality, assume $\tau_j = t_o$ for all $j \in \mathcal{J}$. (See (3.22) and the ensuing discussion.) Since $\varphi_{t_o}(\cdot)$ is submodular by Proposition 3.3.1, $\hat{\varphi}_{t_o}(\cdot)$ is convex by Proposition 3.4.2. Thus,

$$\varphi_{t_o}(\bar{Q}) + \varphi_{t_o}(Q) \geq \hat{\varphi}_{t_o}(\frac{1}{2}\chi_Q + \frac{1}{2}\chi_{\bar{Q}}) = \hat{\varphi}_{t_o}(\frac{1}{2}\chi_{\mathcal{J}}).$$

By the definition of linear extension and (2.7),

$$\hat{\varphi}_{t_o}(\frac{1}{2}\chi_{\mathcal{J}}) = \frac{1}{2}\varphi_{t_o}(\chi_{\mathcal{J}}) + \frac{1}{2}\varphi_{t_o}(\chi_{\emptyset}) = 0.$$

Now from the expression (2.8) applied to $\varphi_{t_o}(Q)$, and the expression (2.9) applied to $\varphi_{t_o}(\bar{Q})$,

$$\begin{aligned} \varphi_{t_o}(Q) + \varphi_{t_o}(\bar{Q}) &= -2[h_Q(t_o) + g_Q^{\emptyset}(t_o)] + 2[h_Q(t_o) + g_Q^{\bar{Q}}(t_o)] \\ &= 2[\eta'(\tau; -\chi_Q) + \eta'(\tau; \chi_Q)]. \end{aligned}$$

Thus, $\eta'(\tau; -\chi_Q) + \eta'(\tau; \chi_Q) \geq 0$ for all $Q \subseteq \mathcal{J}$, and it holds with equality if $Q = \mathcal{J}$.

□

4 Divide-and-Conquer Algorithm

4.1 Description

In this section, we outline an efficient divide-and-conquer algorithm to minimize the total cost η. This algorithm finds a solution that satisfies the first-order necessary condition for the optimality of (P) – namely, this solution has no feasible descent direction. We also describe the correctness and computational complexity for our algorithms.

Our algorithm tracks and modifies clusters C that have the following properties: (1) C is a subset of the set $\mathcal{J}$ of all tools; and (2) there exists a lower bound $lb(C)$ and an upper bound $ub(C)$ such that we know there exists a solution τ^* where $lb(C) \leq \tau_j^* \leq ub(C)$ for all $j \in C$ such that τ^* satisfies the first-order necessary condition of (P). We note that if $lb(C) = ub(C)$, then we have found the desired purchase times τ_j^* for all $j \in C$. At the start of each iteration of the algorithm, we maintain an ordered collection $\mathcal{C}$ of sets, each of which

has the above two properties. We note that $\mathcal{C}$ is a partition of the set $\mathcal{J}$ of all tools, and the intervals $[lb(C), ub(C)]$ defined for these clusters are mutually disjoint except possibly at endpoints. If C_1 and C_2 are two members of $\mathcal{C}$ such that C_1 precedes C_2, then we have $ub(C_1) \leq lb(C_2)$.

Here are the steps of the divide-and-conquer algorithm:

0. Initially, set $\mathcal{C} = \{\mathcal{J}\}$, $lb(\mathcal{J}) = 0$ and $ub(\mathcal{J}) = T$.

1. Choose some $\omega_C \in [lb(C), ub(C)]$, for each $C \in \mathcal{C}$.

2. Choose some $C \in \mathcal{C}$ such that $lb(C) < ub(C)$. Perform cluster splitting of C at ω_C and let $S \subseteq C$ be its optimal solution; i.e., let S minimize $\varphi_{\omega_C}(\cdot|Q_L, C, \mathcal{J}\backslash(Q_L \cup C))$ where Q_L is the union of all clusters preceding C in $\mathcal{C}$, and $S \subseteq C$. If the optimal value is nonnegative, set $lb(S) = ub(S) = \omega_C$. Otherwise, replace C with S and $\bar{S}$ in $\mathcal{C}$, where S precedes $\bar{S} = C\backslash S$. Let $lb(S) = lb(C)$, $ub(S) = \omega_C$, $lb(\bar{S}) = \omega_C(C)$ and $ub(\bar{S}) = ub(C)$.

3. Go to Step 1 unless $lb(C) = ub(C)$ for all $C \in \mathcal{C}$.

Step 1 of the algorithm does not completely specify the choice of C and ω_C. In the *bisection-based method*, we pick $C \in \mathcal{C}$ with the maximum value of $ub(C) - lb(C)$. We choose ω_C to be the midpoint between $lb(C)$ and $ub(C)$. This method traverses the divide-and-conquer tree in a width-first search manner (one depth at a time). Alternatively, we can pick $C \in \mathcal{C}$ arbitrarily, and choose ω_C to be a local minimizer of $\int_{t=0}^{\omega_C} g_C^{Q_L}(t) + h_C(t)dt$ as we vary the value $\omega_C = \tau_j, j \in C$ over the interval $[lb(C), ub(C)]$. (See (2.3) and (2.5).) We call this the *optimization-based method*. The choice of ω_C, in general, determines the output of the algorithm when there are multiple solutions satisfying the first-order necessary conditions.

4.2 Correctness and Complexity

In this section, Theorem 4.2.1 shows that the lower bound and the upper bound used in the above algorithm are valid, justifying the correctness of our algorithm. We also present results regarding the time complexity of our algorithm.

Theorem 4.2.1. *At each iteration of the algorithm, there exists some solution τ^* with no descent direction in (P) such that $\tau_j^* \in [lb(C), ub(C)]$ where for all $j \in C$ and $C \in \mathcal{C}$. If the algorithm terminates, we have found such a solution.*

Proof. Consider one step of the divide-and-conquer method, in which Q^* minimizes $\varphi_t(\cdot|Q_L, Q_o, Q_U)$. Let $\bar{Q}^* = Q_o \backslash Q^*$. If the algorithm does not terminate, assume without loss of generality that the maximum gap between ub and lb converges to 0, and let τ^* be the limit. Let $B_1 = \{j \in Q^* | \tau_j^* = t\}$, and $B_2 = \{j \in \bar{Q}^* | \tau_j^* = t\}$. It suffices to show that $B = B_1 \cup B_2$ has no descent direction at t.

By the algorithm, B_1 and B_2 satisfy

$$h_{B_1}(t) + g_{B_1}^{Q_L \cup Q^* \backslash B_1}(t) \leq 0 \text{ and } h_{B_2}(t) + g_{B_2}^{Q_L \cup Q^*}(t) \geq 0. \tag{4.23}$$

For simplicity of argument, we assume $(Q_L, Q_o, Q_U) = (\emptyset, \mathcal{J}, \emptyset)$. (See (3.22) and the ensuing discussion.) By Proposition 2.2.1, we can assume that the inequalities in (4.23) hold with equality. Let $\bar{B}_1 = Q^* \backslash B_1$ and $\bar{B}_2 = \bar{Q}^* \backslash B_2$. Then,

$$\varphi_t(B_1 | \bar{B}_1, B, \bar{B}_2) = -(h_{B_1}(t) + g_{\bar{B}_1}^{\bar{B}_1}(t)) + (h_{B_2}(t) + g_{\bar{B}_2}^{Q^*}(t)) = 0.$$

For any subset S of $B = B_1 \cup B_2$, we have

$$\varphi_t(\bar{B}_1 \cup S | \emptyset, \mathcal{J}, \emptyset) = -(h_{\bar{B}_1}(t) + g_{\bar{B}_1}^{\emptyset}(t)) + \varphi_t(S | \bar{B}_1, B, \bar{B}_2) + (h_{\bar{B}_2}(t) + g_{\bar{B}_2}^{Q^* \backslash \bar{B}_2}(t)).$$

By the definition of $Q^* = B_1 \cup \bar{B}_1$, we get

$$\begin{aligned} \varphi_t(\bar{B}_1 \cup S | \emptyset, \mathcal{J}, \emptyset) &\geq \varphi_t(\bar{B}_1 \cup B_1 | \emptyset, \mathcal{J}, \emptyset) \\ &= -(h_{\bar{B}_1}(t) + g_{\bar{B}_1}^{\emptyset}(t)) + \varphi_t(B_1 | \bar{B}_1, B, \bar{B}_2) + (h_{\bar{B}_2}(t) + g_{\bar{B}_2}^{Q^* \backslash \bar{B}_2}(t)). \end{aligned}$$

Therefore we obtain $\varphi_t(S | \bar{B}_1, B, \bar{B}_2) \geq \varphi_t(B_1 | \bar{B}_1, B, \bar{B}_2) = 0$ which concludes this proof. □

In the analysis of the algorithm, as in Gallo et al. (1989), we use that the running time of finding a minimum-cut of a network (V, A) is $O(|V||A|\log(|V|^2/|A|))$. We recall that the number of nodes and the number of arcs in the minimum-cut network we construct for cluster splitting are both $O(|\mathcal{I}||\mathcal{J}|)$.

In the optimization-based method, the algorithm terminates in at most $|\mathcal{J}|$ iterations. The number of subsets of $\mathcal{J}$ ever included in $\mathcal{C}$ during the course of running this algorithm is bounded by $2|\mathcal{J}|$. Thus, it performs at most $2|\mathcal{J}|$ optimization and $|\mathcal{J}|$ minimum-cut computations. Let γ be the time required to find a local minimizer of a real-valued one-dimensional function of the form $\int_{t=0}^{\tau} g_C^{Q_L}(t) + h_C(t)dt$ where τ ranges in an interval. Then, the running time of the algorithm is bounded by $O(|\mathcal{J}|\gamma + |\mathcal{J}|^3|\mathcal{I}|^2(\log|\mathcal{J}| + \log|\mathcal{I}|))$.

In the bisection-based method, the size of the interval defined by ub and lb of each cluster in $\mathcal{C}$ is reduced by half at each depth of the divide-and-conquer tree. We note that the total running time of all the minimum-cut computations of a given depth of the divide-and-conquer tree is bounded by that of one minimum-cut computation on a network with $O(|\mathcal{I}||\mathcal{J}|)$ nodes and $O(|\mathcal{I}||\mathcal{J}|)$ arcs. We say τ is an ε-*close* solution if there exists a solution τ^* satisfying the first order necessary condition and $|\tau^* - \tau|_\infty < \varepsilon$. The bisection-based method obtains an ε-close solution in time complexity of $O(|\mathcal{J}|^2|\mathcal{I}|^2(\log|\mathcal{J}| + \log|\mathcal{I}|)\log\varepsilon^{-1})$ at the most.

However, we can achieve better bounds on time complexity. The divide-and-conquer algorithm can be also be stated in terms of the parametric minimum-cut network for cluster splitting. In either the bisection-based or the optimization-based method, an iteration can be described as follows. We have $Q_1 \subseteq Q_2 \subseteq \cdots \subseteq Q_k$ and $t_1 < t_2 < \cdots < t_k$. They are related to the clusters in $\mathcal{C}$ and the corresponding lower and upper bounds. We select $\omega \in [t_r, t_{r+1}]$, and we find a minimum cut Q_o of the parametric minimum-cut network at ω subject to $Q_r \subseteq Q_o \subseteq Q_{r+1}$. Then we add ω to $\{t_1, t_2, \ldots, t_k\}$, and Q_o to $\{Q_1 \subseteq Q_2 \subseteq \cdots \subseteq Q_k\}$, and relabel elements in both sets. This description of the divide-and-conquer algorithm resembles the algorithm of Gusfield and Martel (1992), for monotone parametric minimum-cut networks.

The correctness proof for their algorithm can be extended to build a version of our divide-and-conquer algorithm with better time complexity bounds: $O(|\mathcal{J}|\gamma + |\mathcal{J}|^2|\mathcal{I}|^2(\log|\mathcal{J}| + \log|\mathcal{I}|))$ for the optimization-based method, and $O(|\mathcal{J}||\mathcal{I}|(\log|\mathcal{J}| + \log|\mathcal{I}|)(|\mathcal{J}||\mathcal{I}| + \log\varepsilon^{-1})$ for the bisection-based method. For a fixed γ and ε, these bounds are asymptotically same as the time complexity of one max-flow computation on a graph with $O(|\mathcal{J}||\mathcal{I}|)$ nodes and $O(|\mathcal{J}||\mathcal{I}|)$ arcs.

5 Stationary Product Mix Assumption on Demand

5.1 Description and Global Convexity

In this section, we introduce some assumptions on the demand distribution which enable us to show the global convexity of η. Global convexity is desirable because any local minimizer globally minimizes a convex function. Without the convexity of the objective function, finding a global minimizer of $\eta(\tau)$ is very difficult.

Since $P_j(t)$ is a convex function in t, the tool purchase cost $\eta^P(\tau) = \sum_{j\in\mathcal{J}} P_j(\tau_j)$ is convex and separable in the τ_j's. However, $\eta^{LS}(\tau)$ is not convex in general. If we want to obtain the convexity of $\eta^{LS}(\tau)$ in each permutation simplex $PS(\pi)$, it is sufficient, under uniform fill-rate production, to show that the instantaneous benefit $g_j^{\{j'\in\mathcal{J}|j'<_\pi j\}}(t)$ of having tool $j \in \mathcal{J}$ at time t is non-decreasing with respect to t. (See (2.3) and Proposition 3.4.3.) As we show later in this section, this occurs if the demand distribution has certain properties.

We say that the demand $D_t = \Delta_{I_t,t}\phi_{I_t,t}$ satisfies the *stationary product mix assumption* provided

1. The ray direction ϕ_{it} and ray probability $w_i(t)$ are independent of t.
2. Demand magnitude Δ_{it} is stochastically nondecreasing in t.
3. The lost sales cost c_{pt} per unit is nondecreasing in t.

Note that under this assumption, the distribution of $\frac{1}{|D_t|}D_t = \phi_{I_t,t}$ is stationary. However $\frac{1}{|E(D_t)|}E(D_t)$ is not necessarily stationary.

These assumptions are quite strong, but necessary to show theoretical results. We will show, in fact, that the above assumptions are sufficient not only for convexity in each permutation simplex, but also for the global convexity.

Proposition 5.1.1. *Under the uniform fill-rate production and the stationary product mix assumption, the expected lost sales cost $\eta^{LS}(\tau)$ is convex with respect to τ_j's in each permutation simplex.*

Proof. Suppose a permutation cone $PS(\pi)$ is defined by $\pi \in \Pi$. Since the partial derivative of η^{LS} with respect to $\pi(r)$, $r \in \{1, \ldots, |\mathcal{J}|\}$, is $g_{\pi(r)}^{\pi^-(r)}(t)$ (see (2.3)), it suffices to show that this function is nondecreasing.

By the stationary product mix assumption, we can write $w_{it} = w_i$, and $\phi_{ipt} = \phi_{ip}$. We note that $\zeta_{it}^{\pi\{1,\ldots,r\}}$ is the maximum magnitude we can produce along the ray $\phi_i = (\phi_{ip}|p \in \mathcal{P})$ provided that we have purchased r tools $\{\pi(1), \ldots, \pi(r)\}$ in the sequence π. From (2.2) and (3.20),

$$\begin{aligned} g_{\pi(r)}^{\{\pi(1),\ldots,\pi(r-1)\}}(t) &= \xi(\{\pi(1), \ldots, \pi(r-1)\}, t) - \xi(\{\pi(1), \ldots, \pi(r)\}, t) \\ &= \sum_{i \in \mathcal{I}} w_i (\sum_{p \in \mathcal{P}} c_{pt}\phi_{ip}) E_{\Delta_{it}}[(\Delta_{it} - \zeta_{it}^{\{\pi(1),\ldots,\pi(r-1)\}})^+ - (\Delta_{it} - \zeta_{it}^{\{\pi(1),\ldots,\pi(r)\}})^+]. \end{aligned}$$

Since $\zeta_{it}^{\{\pi(1),\ldots,\pi(r-1)\}} \leq \zeta_{it}^{\{\pi(1),\ldots,\pi(r)\}}$ and Δ_{it} is stochastically nondecreasing, we conclude that $g_{\pi(r)}^{\{\pi(1),\ldots,\pi(r-1)\}}(t)$ is nondecreasing in t. □

Since η^P is convex, the previous proposition shows that η is convex in any permutation simplex $PS(\pi)$, $\pi \in \Pi$. A continuous function that is convex in each permutation simplex may in general not be globally convex. This following proposition provides a sufficient condition for global convexity to hold.

Proposition 5.1.2. *Suppose that η is convex in each permutation simplex. Then, η is globally convex.*

Proof. Let $\eta_L(s)$ be the function f restricted to some line segment $L \subseteq \mathbb{R}^{\mathcal{J}}$, where $s \in [0, 1]$ is a parameter. It suffices to show that η_L is convex. In each permutation simplex, f is

convex, and thus η'_L is nondecreasing. When L intersects with the boundaries of permutation cones, the left derivative of η_L is no more than the right derivative η_L by Proposition 3.4.3, Therefore, by Proposition B.2 in Bertsekas (1995) or alternatively by the monotonicity of the "presubdifferential" in Correa et al. (1995), we conclude that η_L is convex. □

From Propositions 5.1.1 and 5.1.2, we have the following theorem:

Theorem 5.1.3. *Under the uniform fill-rate production and the stationary product mix assumption, the objective cost function η is globally convex.*

Consequently, the divide-and-conquer algorithm produces globally optimal purchase times. We also observe that the parametric minimum-cut network of cluster splitting is monotonic.

5.2 Numerical Testing

This section compares the performance of the divide-and-conquer algorithm with that of the discrete-time model presented in Roundy et al. (2000) for the case when demand satisfies the stationary product mix assumption. We solve capacity planning problems of practical size and complexity. Computations indicate that the divide-and-conquer method for the continuous-time model finds a solution that is very close to solution found by the discrete-time model. The divide-and-conquer method runs much faster than the discrete-time algorithm.

We modify the data used in Roundy et al. (2000) by eliminating the fixed cost for tool purchases. The demand data is modified to satisfy the stationary product mix assumption. There are 4 product families and 43 tool types. The algorithms are tested with 2, 4, 8, 16, 32, 64 and 128 demand rays. Simulations are carried out on a Dell Inspiron 5000 notebook with a Pentium III 600 MHz processor and 128 MB of memory.

Table 1 summarizes the performance of the discrete-time model, which makes quarterly decisions for the next four years. The algorithm in Roundy et al. (2000) consists of two parts: generating a network using MATLAB, and solving the resulting network using a push-relabel algorithm for the minimum cut problem. We use a min-cut implementation called HIPR due

to Cherkassky and Goldberg (1997). Each running time in the table is an average of four trials.

Table 2 reports the performance of the optimization-based divide-and-conquer method for the continuous-time model. To evaluate costs associated with the resulting solution, we round-off the solution to the nearest quarter. This method uses HIPR several times, and the sum of running times on all HIPR calls are reported. All other operations including descent methods, are included in the MATLAB running time. Each of four repetitions yielded the same solution, and their average running times are reported.

Tables 1 and 2 show that both models find solutions that are very close. The differences in the cost are due to sub-optimality. They are less than a fraction of one percent. However, we see that as the number of demand rays increases, the continuous-time model has a dramatic advantage over the discrete-time model in running time. This is probably due to the size of the min-cut network; the discrete-time model solves by one big min-cut problem, whereas the continuous-time model generates many minimal cuts ons smaller networks.

6 Conclusions

This paper addresses the capacity planning for multiple tool types that are shared by multiple product families. Unlike the majority of past research, the uncertainty in the demand of these products is explicitly captured via stochastic programming. We have shown how problems of realistic size and complexity can be modeled and how these models can be solved efficiently. The models and theoretical results presented in this paper may serve as a prototype in constructing more complex and robust strategic capacity planning systems.

It would be nice to extend our model to use stochastic programming with full recourse; however, for problems of realistic size, that is beyond today's computational ability. Our future work includes extending our model to incorporate some generalizations that are of interest to the semiconductor industry. Possible generalizations include: capacity contraction, tool retirement, mean demand that is not necessarily increasing, complex tool-product

relationships, and LP-based shortfall allocation (see Section 3.1). These extensions may spoil many of the theoretical properties presented in this paper. Yet, there are many non-convex minimization problems for which local descent algorithms consistently yield very good solutions. We have some concrete indications that a modification of the algorithm presented in this paper will obtain good solutions with these generalizations.

Acknowledgements

The authors would like to thank Metin Çakanyıldırım, Lisa Fleischer, Tom McCormick, Michael Todd, and Feng Zhang for their assistance and insightful comments in preparing this document.

References

Benavides, D. L., Duley, J. R., and Johnson, B. E. (1999, August), "As Good As It Gets: Optimal Fab Design and Deployment," *IEEE Transactions on Semiconductor Manufacturing*, 12(3), 281–287.

Berman, O., Ganz, Z., and Wagner, J. M. (1994), "A Stochastic Optimization for Planning Capacity Expansion in a Service Industry under Uncertain Demand," *Naval Research Logistics*, 41, 545–564.

Bermon, S. and Hood, S. J. (1999, September-October), "Capacity Optimization Planning System (CAPS)," *Interfaces*, 29(5), 31–50.

Bertsekas, D. P. (1995), *Nonlinear Programming*, Belmont, Massachusetts: Athena Scientific.

Bhatnagar, S., Fernández-Gaucherand, E., Fu, M. C., He, Y., and Marcus, S. I. (1999), "Dynamic Capacity Expansion Problem with Multiple Products: Technology Selection and Timing of Capacity Additions," in *Proceedings of the 38th IEEE Conference on Decision and Control*.

Çakanyıldırım, M., Roundy, R. O., and Wood, S. C. (1999, September), "Machine Purchasing Strategies under Demand- and Technology-driven Uncertainties," Technical Report TR1250, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York.

Chand, S., McClurg, T., and Ward, J. (2000), "A model for parallel machine replacement with capacity expansion," *European Journal of Operational Research*, 121(3), 519–531.

Chavátal, V. (1983), *Linear Programming*, New York: W. H. Freeman and Company.

Chen, Z. L., Li, S., and Tirupati, D. (1998), "A Scenario Based Stochastic Programming Approach for Technology and Capacity Planning," Technical Report 98-04, Department of Systems Engineering, University of Pennsylvania.

Cherkassky, B. V. and Goldberg, A. V. (1997), "On Implementing Push-Relabel Method for the Maximum Flow Problem," *Algorithmica*, 19(4), 390–410.

Correa, R., Jofré, A., and Thibault, L. (1995), "Subdifferential Characterization of Convexity," in *Recent Advances in Nonsmooth Optimization*, eds. D.-Z. Du, L. Qi, and R. S. Womersley, 18–23, World Scientific.

Federgruen, A., Queyranne, M., and Zheng, Y. S. (1992, November), "Simple Power-of-2 Policies Are Close to Optimal in a General-Class of Production Distribution Networks with General Joint Setup Costs," *Mathematics of Operations Research*, 17(4), 951–963.

Federgruen, A. and Zheng, Y. S. (1992, March-Paril), "The Joint Replenishment Problem with General Joint Cost Structures," *Operations Research*, 40(2), 384–403.

Gallo, G., Grigoriadis, M., and Tarjan, R. E. (1989, February), "A Fast Parametric Maximum Flow Algorithm," *SIAM Journal on Computing*, 18(1), 30–55.

Gusfield, D. and Martel, C. (1992), "A Fast Algorithm for the Generalized Parametric Minimum Cut Problem and Applications," *Algorithmica*, 7, 499–519.

Li, S. and Tirupati, D. (1994, September-October), "Dynamic Capacity Expansion Problem with Multiple Products: Technology Selection and Timing of Capacity Additions," *Operations Research*, 42(5), 958–976.

Lovász, L. (1983), "Submodular functions and convexity," in *Mathematical Programming: The State of the Art, Bonn 1982*, eds. A. Bachem, M. Groötschel, and B. H. Korte, Berlin, 235–257, Springer–Verlag.

Rajagopalan, S. and Yu, S. (2001), "A Capacity Planning Model with Congestion Costs," *European Journal of Operational Research*, 134(2), 137–149.

Roundy, R. O., Zhang, F., Çakanyıldırım, M., and Huh, W. T. (2000, October), "Optimal Capacity Expansion for Multi-Product, Multi-Machine Manufacturing Systems with Stochastic Demand," Technical Report TR1271, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York.

Ryan, S. M. (1999), "Capacity Expansion with Lead Times and Correlated Random Demand," Technical Report 99-107, Industrial and Manufacturing Systems Engineering, Iowa State University.

Ryan, S. M. (2000), "Capacity Expansion for Random Exponential Demand Growth with Lead Times," Technical Report 00-109, Industrial and Manufacturing Systems Engineering, Iowa State University.

Semiconductor Industry Association (1999), *International Technology Roadmap for Semiconductors: 1999 edition*, Austin, TX: International SEMATECH.

Number of Demand Rays	2	4	8	16	32	64	128
Lost Sales Cost (million $)	20.66	22.85	29.83	31.16	31.07	28.96	31.27
Purchase Cost (million $)	202.92	205.55	197.98	194.90	194.36	199.04	196.67
Total Cost (million $)	223.58	228.40	227.82	226.05	225.43	228.00	227.94
HIPR CPU Time (sec)*	1.92	3.27	7.21	15.25	25.86	56.51	3717.02 †
MATLAB CPU Time (sec)	34.43	64.91	125.53	251.11	499.56	1044.31	2206.15 †

* It includes, unlike Roundy et al. (2000), the time required for file input/output.

† With 128 rays, a low virtual memory warning was issued.

Table 1: Computational Result for the Discrete-Time Capacity Planning Model

Number of Demand Rays	2	4	8	16	32	64	128
Lost Sales Cost (million $)	12.20	15.29	25.27	21.90	20.83	22.01	22.00
Purchase Cost (million $)	212.42	213.82	202.82	204.79	205.22	206.64	206.60
Total Cost (million $)	224.62	229.11	228.09	226.70	226.05	228.66	228.60
HIPR CPU Time (sec)	0.94	1.02	1.64	2.52	7.03	8.40	15.94
MATLAB CPU Time (sec)	30.74	36.82	67.79	119.47	196.69	428.94	770.49

Table 2: Computational Result for the Continuous-Time Capacity Planning Model