A Continuous-Time Strategic Capacity Planning Model*

Woonghee Tim Huh,[1] Robin O. Roundy[2]

[1] *Department of Industrial Engineering and Operations Research, Columbia University, New York, New York 10027*

[2] *School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York 14853*

Received 29 October 2003; revised 22 October 2004; accepted 23 January 2005
DOI 10.1002/nav.20081
Published online 15 March 2005 in Wiley InterScience (www.interscience.wiley.com).

Abstract: Capacity planning decisions affect a significant portion of future revenue. In the semiconductor industry, they need to be made in the presence of both highly volatile demand and long capacity installation lead-times. 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 by an application of the divide-and-conquer algorithm, convexity, submodularity, and the open-pit mining problem. © 2005 Wiley Periodicals, Inc. Naval Research Logistics 52: 329–343, 2005.

Keywords: capacity planning; stochastic demand; submodularity; semiconductor industry

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, up to two-thirds of which are attributed to tool costs. The lead-time for purchasing tools is between 6 months and 18 months, upon which tools will start becoming 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. A poorly planned sequence of tool purchases results that expensive tools idle due to the lack of complementary critical machines. Premature tool purchases incur unnecessary high purchase costs because of rapidly decreasing tool prices and result in overcapacity, 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 predetermined 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 6 months to 4 years. 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. We consider strategic capacity planning under demand uncertainty.

An extensive review of literature can be found in Luss [26], Çakanyıldırım, Roundy, and Wood [10], and Roundy et al. [29]. Most models are unable to handle problems of both realistic size and complexity arising from the semiconductor industry. The first set of papers, including Benavides, Duley, and Johnson [4], Berman, Ganz, and Wagner [5], Li, and Tirupati [11], and Li and Tirupati [24], uses simple (such as one-to-one) relationships between product families and tool types. Capacity planning with a single product and

* 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 as well as the Second Prize in the 2002 INFORMS Manufacturing & Service Operations Management Section Student Paper Competition. Its extended abstract appeared in Huh [20].

Correspondence to: W.T. Huh (huh@ieor.columbia.edu)

single resource type is relatively easy to solve. However, they fail to model the complexity of semiconductor manufacturing production. The second set of papers, including Escudero et al. [17], Bhatnagar [8], Swaminathan [30], and Ahmed and Sahinidis [1], present complex models, which are challenging to provably solve with problems of realistic size. For these problems, stochastic programming with full recourse is beyond today's computational ability.

An example of the current practice in capacity planning has been documented by Bermon and Hood [6] at IBM. Its Capacity Optimization Planning System (CAPS) is an expansion model for multiple products and multiple resource types, and are based on linear programming. It assumes deterministic demand forecast. A stochastic version of CAPS, called SCAPS (Barahona et al. [3]), has been subsequently developed, and models demand with several scenarios. SCAPS applies traditional integer programming techniques. Both CAPS and SCAPS, like a majority of capacity planning models, is based on *discrete-time* models—the planning period is divided into a finite number of time periods, and the decision variables are indexed by them. These decision variables are often binary indicators for whether a particular tool should be purchased in a given period under a demand scenario. As a result, the resulting formulation has a large number of decision variables, and becomes difficult to solve.

A notably exceptional discrete-time stochastic model for multiple resource types and multiple product families is due to Roundy et al. [29]. It exhibits a fast algorithm exploiting the maximum-flow and minimum-cut structure. However, because of its structural rigidity, this model cannot be generalized to cope with additional constraints or requirements.

As in Çakanyıldırım and Roundy [9] and Çakanyıldırım, Roundy, and Wood [10], we continue to explore alternative approaches based on *continuous-time models.* The time at which a machine is purchased becomes a continuous decision variable. Davis et al. [15] and Khmelnitsky and Kogan [23] have used continuous-time models in the context of optimal control theory. 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-time 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. The model in Çakanyıldırım et al. [10] assumes multiple resource types and a single product family with nondecreasing demand. Because there is only one product family, one can define a sequence of tools in the order they become bottleneck machines. The resulting formulation is minimization of a separable convex function with a chain constraint. This model is extended in Çakanyıldırım and Roundy [9] to allow decreasing demand over time and tool retirement.

In this paper, we present a capacity expansion model for multiple resource types and multiple product families under demand uncertainty. We allow decreasing demand. As in Bermon and Hood [6], Barahona et al. [3], Roundy et al. [29], Çakanyıldırım, Roundy, and Wood [10], and Çakanyıldırım and Roundy [9], we assume no backorders and negligible amount of inventory. When capacity is insufficient to meet all realized demand, then we use a capacity allocation policy that equalizes the fill rates of product families. It is a multiple product family version of Çakanyıldırım et al. [10] and Çakanyıldırım and Roundy [9]. Since these models consider a single product, it is possible to deduce a linear order on resource types depending on when the corresponding purchases become bottleneck constraints. Since multiple products require varying amounts and ratios of capacities, we cannot deduce the sequence of tool purchases from preprocessing. Our model is also a continuous-time version of the discrete-time model in Roundy et al. [29]. The continuous-time model is likely to allow more modeling extensions than the discrete-time model as demonstrated in a follow-up paper (Huh, Roundy, and Çakanyıldırım [21]).

We show that the objective function of the formulation is not separable, but *quasiseparable*—the objective function is separable within each subset of the domain where the ordering of decision variables is fixed. We present an efficient divide-and-conquer algorithm which finds a local optimal solution of this problem. A subroutine to this algorithm is the submodular function minimization problem. We also identify certain conditions under which the planning algorithm finds a global optimal solution; the continuous-time model under some assumptions becomes a convex program, a provably tractable instance of nonlinear programming.

In Section 2, we describe our model of planning capacity for the multiple-product and multiple-machine manufacturing system, and derive basic properties including quasiseparability. In Section 3, we present a policy by which we allocate capacity across multiple product families, and show how this policy is related to submodularity. 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 the demand model 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 as-

sumes 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 capacity allocation stochastic programming recourse, the production quantities at a given time instant are decided as a function of the capacity and demand at that time instant only, and not of those at other time instants. At time 0, all the capacity acquisition plans are made whereas production decisions are made at each time instant after instantaneous demands have been observed. We use this model as a part of a rolling-horizon implementation.

Section 2 assumes a general capacity allocation policy and a general demand model. (For a specific capacity allocation policy and demand model, see Sections 3.1 and 5.1.) Section 2.1 presents constraints and the objective function of our formulation. Section 2.2 elaborates on how the instantaneous lost sales cost is computed. Section 2.3 explores separability properties of the objective function. Section 2.4 characterizes local optimality.

2.1. Formulation

Let $t \in [0, T]$, where T is the planning horizon. We use $p \in \mathcal{P}$ and $m \in \mathcal{M}$ to index product families and tool types, respectively. For each tool of type $m \in \mathcal{M}$, we let n be its index in the ordered set $\mathcal{N}_m$ of tools of type m in the order that purchases will be made. There are partially ordered (acyclic) precedence relations between tools indicating certain tools should be purchased no earlier than other tools. They include the sequence of tool purchases of type m defined by the ordered $\mathcal{N}_m$. Let $\mathcal{J} = \cup_{m\in\mathcal{M}} \mathcal{N}_m$ be the (unordered) set of all tools of all types that we contemplate purchasing over the planning horizon. For easier referencing, we use j as well as (m, n) to index $\mathcal{J}$.

The price of purchasing tool j at time t is assumed to be a decreasing differentiable convex function $P_j(t)$ of t. This assumption holds, for example, when the discounting factor is constant over time. Such a function includes a linearly decreasing function and an exponentially decreasing function. 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, the associated tool capacity $\mu_m(Q)$ of type m is

$$\mu_m(Q) = \sum_{n:(m,n)\in Q'} u_{mn}$$

where $Q' \subseteq Q$ consists of all the tools that obey the precedence relations; i.e., Q' is the maximal initial set of Q. [The definition of μ_m ensures that tools purchases should be consistent with the precedence relations because any tool (m, n) purchased out of order 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. Our objective is to minimize the sum of tool purchase costs and expected lost sales costs. The tool purchase cost is

$$\eta^P(\tau) = \sum_{j\in\mathcal{J}} P_j(\tau_j),$$

which is separable in the τ_j's. 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. The value of $\xi(Q, t)$ depends on Q through the $\mu_m(Q)$'s, and is discussed in Section 2.2 (see also Section 3.1). We denote by $Q_t^\tau = \{j : \tau_j \le t\}$ 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

$$\eta^{LS}(\tau) = \int_{t=0}^{T} \xi(Q_t^\tau, t)\, dt. \tag{1}$$

The problem we want to solve is the following:

$$(P) \qquad \min_{\tau} \quad \eta(\tau) = \eta^P(\tau) + \eta^{LS}(\tau)$$

$$\text{s.t.} \qquad 0 \le \tau_j \le T \qquad \text{for all } j \in \mathcal{J}.$$

We do not explicitly write the precedence relations as constraints because the definition of $\mu_m(Q)$ incorporates them into the objective function η.

2.2. Instantaneous Lost Sales Cost

This section explains how we determine the expected value ξ of the instantaneous lost sales cost given a specific capacity allocation policy. Further discussion on various capacity allocation policies is provided in Section 3.1.

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. Given a set Q_t^τ of tools which are available at time t (which is determined by τ), tool type m's capacity is given by $\mu_m(Q_t^\tau)$. 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 $\mu(Q_t) = (\mu_m(Q_t) | m \in \mathcal{M})$ of all tool types and the *realized* demand $d_t = (d_{pt} | p \in \mathcal{P}) \ge 0$ of all product families at time t, we determine both the instantaneous production quantity $v_t = (v_{pt} | p \in \mathcal{P})$ of

product family p and the allocation $x_t = (x_{mpt} | m \in \mathcal{M}, p \in \mathcal{P})$, where x_{mpt} is the amount of tool type m's capacity allocated to p. A *capacity allocation policy* is a way of selecting x_t and v_t, given Q_t and d_t.

As in many papers in the capacity planning literature (e.g., Çakanyıldırım and Roundy [9] and Roundy et al. [29]) and the current capacity planning practice in the semiconductor industry (e.g., Bermon and Hood [5]), assume no finished goods inventory, and no backorders. In other words, demand at time t can be satisfied by what is produced at time t only. We remark that there are a number of papers that model joint capacity and inventory decisions (e.g., Rajagopalan and Swaminathan [28]; Atamturk and Hochbaum [2]). Under our modeling assumption, in any capacity allocation policy, production does not exceed demand, i.e.,

$$v_{pt} \le d_{pt} \qquad \text{for all } p \in \mathcal{P}. \tag{2}$$

There need not be a demand shortfall if the capacity $\mu_m(Q_t^\tau)$ is sufficient to meet the demand d_t, i.e.,

$$\sum_{p=1}^{P} U(m, p) d_{pt} \le \mu_m(Q_t^\tau) \qquad \text{for all } m = 1, \ldots, M.$$

Otherwise, we are unable to meet all demands. A capacity allocation policy determines how we allocate insufficient capacity to product families. Under any allocation policy, production v and allocation x must obey the capacity limit of each tool type:

$$\sum_{p=1}^{P} x_{mpt} \le \mu_m(Q_t) \qquad \text{for all } m \in \mathcal{M} \text{ and } t \in [0, T], \tag{3}$$

$$U(m, p) v_{pt} \le x_{mpt} \qquad \text{for all } p \in \mathcal{P} \text{ and } t \in [0, T]. \tag{4}$$

It is noted that these constraints are necessary conditions, and a capacity allocation policy may impose further constraints on x and v. Any capacity allocation policy defines a production function $v_t = \nu(d_t, Q_t) = (\nu_p(d_t, Q_t) | p \in \mathcal{P})$ in terms of d_t and Q_t (which depends on τ). This function, in general, is neither simple nor algebraic, which make analysis difficult. In Section 3.1, we present a capacity allocation policy which makes this dependency tractable.

The lost sales are the difference between demand and production. At time t, the lost sales of product family p are $(d_{pt} - v_{pt})^+$. Let demand $D_t = (D_{pt} | p \in \mathcal{P}) \ge 0$ be random vector corresponding to d_t. Then we can write

$$\xi(Q_t, t) = E_{D_t}\left[\sum_{p \in \mathcal{P}} c_{pt}(D_{pt} - \nu_p(D_t, Q_t))^+ \right]. \tag{5}$$

We remark that lost sales $(D_{pt} - \nu_p(D_t, Q_t))^+$ depend on the capacity allocation policy $\nu(\cdot)$, on demand D_t, and also on tool availability Q_t.

2.3. Quasiseparability

In this section, we derive another expression for the expected lost sales cost $\eta^{LS}(\tau)$ within a subset of the feasible region, and develop separability properties of η and additive properties of its directional derivatives.

Let $Q^0 \subseteq \mathcal{J}$ and $j \in \mathcal{J} \backslash Q^0$. Let $t \in [0, T]$. We denote by $g_j^{Q^0}(t)$ the amount of reduction in the expected instantaneous lost sales cost ξ at time t by adding the tool j to the set Q^o of available tools. Formally we define

$$g_j^{Q^o}(t) = \xi(Q^o, t) - \xi(Q^o + j, t), \tag{6}$$

where we write $Q^o + j$ for $Q^o \cup \{j\}$. 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 assume, throughout this paper, that $\xi(Q^1, t) \ge \xi(Q^2, t)$ for any t whenever $Q^1 \subseteq Q^2 \subseteq \mathcal{J}$. It follows that $g_j^{Q^o}(t) \ge 0$.

We generalize the definition (6) of g: For any disjoint sets $Q^o, Q \subseteq \mathcal{J}$ of tools, we define

$$g_Q^{Q^o}(t) = \xi(Q^o, t) - \xi(Q^o \cup Q, t). \tag{7}$$

This quantity corresponds to the marginal benefit of adding the tool set Q to the existing set Q_o at time t. A direct consequence of (7) is

$$g_{Q^1}^{Q^o}(t) + g_{Q^2}^{Q^o \cup Q^1}(t) = g_{Q^1 \cup Q^2}^{Q^o}(t) \tag{8}$$

provided that $Q^o, Q^1, Q^2 \subseteq \mathcal{J}$ are disjoint.

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. For each $R \subseteq \{1, \ldots, |\mathcal{J}|\}$, let $\pi(R) = \{\pi(r) | r \in R\}$.

PROPOSITION 1: For any $\pi \in \Pi$ and $r \in \{1, \ldots, |\mathcal{J}|\}$, the expected lost sales cost $\eta^{LS}(\tau)$ is continuous and separable within the permutation simplex $PS(\pi)$. If $j = \pi(r)$, then in the interior of $PS(\pi)$, we have

$$\frac{\partial}{\partial \tau_j} \eta^{LS}(\tau) = g_j^{Q^o}(\tau_j), \tag{9}$$

where $Q^o = \pi(\{1, \ldots, r-1\})$.

PROOF: 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 $\pi(\{1, 2, \ldots, r_0 - 1, r_0\})$ for some $r_0 \in \{1, \ldots, |\mathcal{J}|\}$. Thus, the telescoping sum of (6) implies

$$\xi(Q_t^\tau, t) = \xi(\mathcal{J}, t) + \sum_{r:\tau_{\pi(r)} > t} g_{\pi(r)}^{\pi(\{1,\ldots,r-1\})}(t).$$

From (1), we obtain

$$\eta^{LS}(\tau) = \int_{t=0}^{T} \xi(\mathcal{J}, t)\, dt + \sum_{r=1}^{|\mathcal{J}|} \int_{t=0}^{\tau_{\pi(r)}} g_{\pi(r)}^{\pi(\{1,\ldots,r-1\})} \times (t)\, dt, \qquad \tau \in PS(\pi). \tag{10}$$

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 in $\tau_{\pi(r)}$, the upper limit of the second integral. Differentiating $\eta^{LS}(\tau)$ with respect to $\tau_{\pi(r)} = \tau_j$ results in (9). □

Furthermore, $\eta^{LS}(\tau)$ is continuously differentiable in the interior of $PS(\pi)$ if each $g_{\pi(r)}^{\pi(1,\ldots,r-1)}$, $r \in \{1, \ldots, |\mathcal{J}|\}$, is continuous with respect to τ. We observe that the right-hand side expression of (9) is common across many permutation simplices.

COROLLARY 2: The partial derivative of $\eta^{LS}(\tau)$ with respect to τ_j is also given by (9) in the interior of $\{\tau \in [0, T]^{\mathcal{J}} : \tau_{j_1} \le \tau_j \le \tau_{j_2}$ for all $j_1 \in Q^o$ and $j_2 \in \mathcal{J} \backslash (Q^o + j)\}$.

A differentiable function is separable if its partial derivative with respective to one variable is independent of the values of the other variables. We say a function $f(\tau)$ is *quasiseparable* if its partial derivative with respective to one variable τ_j depends on the other variables only through the set $\{j' \in \mathcal{J} : \tau_{j'} < \tau_j\}$. The function $\eta^P(\tau)$ is separable, and thus we can define, without ambiguity, $h_j(t) = (\partial/\partial\tau_j)\eta^P(\tau)$, where $\tau_j = t$ for $j \in \mathcal{J}$. We also define $h_Q(t) = \sum_{j \in Q} h_j(t)$ for $Q \subseteq \mathcal{J}$. The function $\eta^{LS}(\tau)$ is not separable, but it is quasiseparable by Corollary 2. From the separability of $\eta^P(\tau)$, $\eta(\tau) = \eta^P(\tau) + \eta^{LS}(\tau)$ is quasiseparable.

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:

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

We say a feasible direction y is a *descent direction* provided that $\eta'(\tau; y) < 0$.

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. The following proposition shows that the directional derivatives of η are separable "by cluster."

PROPOSITION 3: Suppose η is quasiseparable, and $y \in \mathbb{R}^{\mathcal{J}}$ is a feasible direction at τ. Let $y^1, \ldots, y^K \in \mathbb{R}^{\mathcal{J}}$ satisfying

(i) $y = \sum_{k=1}^K y^k$,
(ii) the supports of y^k, $k = 1, \ldots, K$, are mutually exclusive, and
(iii) for $j_1, j_2 \in \mathcal{J}$, we have $\tau_{j_1} = \tau_{j_2}$ whenever there exists $k = 1, \ldots, K$ such that both $y_{j_1}^k$ and $y_{j_2}^k$ are nonzero.

Then,

$$\eta'(\tau; y) = \sum_{k=1}^{K} \eta'(\tau; y^k).$$

PROOF: We show the proof for the case when the supports of y^k's partition $\mathcal{J}$.

Let $t_k = \tau_j$, where $y_j^k \neq 0$. Without loss of generality, assume $t_1 < t_2 < \cdots < t_K$. Then $J_{t_k}(\tau)$ is the support of y^k. Choose $\pi \in \Pi$ such that, for sufficiently small $\varepsilon > 0$, $\tau + \varepsilon y$ is in the permutation simplex $PS(\pi)$ defined by π. For $k_1 < k_2, j_1 \in J_{t_{k_1}}(\tau)$ precedes $j_2 \in J_{t_{k_2}}(\tau)$ in the order defined by π. Since η is separable and differentiable within $PS(\pi)$, Proposition 1 and the ensuing discussion imply

$$\eta'(\tau; y) = \sum_{r=1}^{|\mathcal{J}|} y_{\pi(r)}[h_{\pi(r)}(\tau_{\pi(r)}) + g_{\pi(r)}^{\pi(\{1,\ldots,r-1\})}(\tau_{\pi(r)})]$$

$$= \sum_{k=1}^{K} \sum_{\pi(r) \in J_{t_k}(\tau)} y_{\pi(r)}[h_{\pi(r)}(\tau_{\pi(r)}) + g_{\pi(r)}^{\pi(\{1,\ldots,r-1\})}(\tau_{\pi(r)})]$$

$$= \sum_{k=1}^{K} \eta'(\tau; y^k). \qquad \square$$

As a result, to find a descent direction y, it suffices to look for a descent direction in one cluster at a time.

2.4. Characterization of Local Optimality

The previous section indicates that the function η behaves well within a permutation simplex. For example, η is separable, and 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. The major result of this section is Theorem 4, which characterizes local optimality.

Suppose at time t_0, we have a partition Q_L, Q_0, and Q_U of $\mathcal{J}$, where Q_L is the set of tools we have purchased prior to t_0, and Q_U is the set of tools we will purchase after t_0. Currently, we purchase tools in Q_o at t_o. 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

$$\vartheta_{t_o}(Q|Q_L, Q_o, Q_U) = \begin{cases} -[h_Q(t_o) + g_Q^{Q_L}(t_o)] + [h_{Q_o\backslash Q}(t_o) + g_{Q_o\backslash Q}^{Q_L\cup Q}(t_o)], \\ \qquad\qquad \text{if } t_o \in (0, T), \\ +[h_{Q_o\backslash Q}(t_o) + g_{Q_o\backslash Q}^{Q_L\cup Q}(t_o)], \quad \text{if } t_o = 0, \\ -[h_Q(t_o) + g_Q^{Q_L}(t_o)], \quad \text{if } t_o = T. \end{cases} \tag{11}$$

From (8), for $t_o \in (0, T)$, the above expression can be rewritten as

$$\vartheta_{t_o}(Q|Q_L, Q_o, Q_U) = -2[h_Q(t_o) + g_Q^{Q_L}(t_o)] + [h_{Q_o}(t_o) + g_{Q_o}^{Q_L}(t_o)], \tag{12}$$

or, equivalently, as

$$\vartheta_{t_o}(Q|Q_L, Q_o, Q_U) = -[h_{Q_o}(t_o) + g_{Q_o}^{Q_L}(t_o)] + 2[h_{Q_o\backslash Q}(t_o) + g_{Q_o\backslash Q}^{Q_L\cup Q}(t_o)]. \tag{13}$$

We also write $\vartheta_{t_o}(Q) = \vartheta_{t_o}(Q|\varnothing, \mathcal{J}, \varnothing)$.

We say a function η has a descent direction y at τ if $\eta'(\tau; y) < 0$ for some feasible direction $y \in \mathbb{R}^{\mathcal{J}}$. If a current solution τ satisfies $\tau_j = t_o$ for each $j \in \mathcal{J}$, computing (11) for all $Q \subseteq \mathcal{J}$ corresponds to the evaluation of the directional derivative of the objective function η in $2^{|\mathcal{J}|}$ directions. It may be plausible that while splitting a cluster into two clusters finds no descent direction in η, splitting it into three or more clusters may find one. The next theorem, however, shows that this is not the case.

THEOREM 4: Suppose $\tau \in [0, T]^{\mathcal{J}}$. The quasiseparable function η has a descent direction at τ if and only if there exist $t_o \in [0, T]$ and $Q \subseteq J_{t_o}(\tau)$ such that $\vartheta_{t_o}(Q|\cup_{t<t_o} J_t(\tau), J_{t_o}(\tau), \cup_{t>t_o} J_t(\tau)) < 0$.

PROOF: Let χ_Q, $Q \subseteq \mathcal{J}$, be the characteristic vector of Q, i.e., the jth component of χ_Q is 1 if $j \in Q$, and 0 otherwise.

Without loss of generality, assume $\tau_j = t_o \in (0, T)$ for all $j \in \mathcal{J}$ (see Proposition 3). It is easy to see that if $t_o \in [0, T]$ and $Q \subseteq \mathcal{J}$ satisfy $\vartheta_{t_o}(Q) = \vartheta_{t_o}(Q|\varnothing, \mathcal{J}, \varnothing) < 0$, then $-\chi_Q + \chi_{\mathcal{J}\backslash Q}$ is a descent direction of η at τ.

Suppose $\vartheta_{t_o}(Q) \geq 0$ for all $Q \subseteq \mathcal{J}$. Then, in particular, we have $\vartheta_{t_o}(\varnothing) \geq 0$ and $\vartheta_{t_o}(\mathcal{J}) \geq 0$, so (11) implies that $h_{\mathcal{J}}(t_o) + g_{\mathcal{J}}^{\varnothing}(t_o) = 0$. Consequently, for $Q \subseteq \mathcal{J}$, it follows from the expression (13) for $\vartheta_{t_o}(\mathcal{J}\backslash Q)$ that

$$0 \leq \tfrac{1}{2}\vartheta_{t_o}(\mathcal{J}\backslash Q) = -\tfrac{1}{2}[h_{\mathcal{J}}(t_o) + g_{\mathcal{J}}^{\varnothing}(t_o)] + [h_Q(t_o) + g_Q^{\mathcal{J}\backslash Q}(t_o)] = \eta'(\tau; \chi_Q). \tag{14}$$

Now we prove $\eta'(\tau; y) \geq 0$ for general $y \in \mathbb{R}^{\mathcal{J}}$. Let $\pi \in \Pi$ be such that y satisfies $y_{\pi(1)} \leq \cdots \leq y_{\pi(|\mathcal{J}|)}$. 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. Since $\tau + \varepsilon\beta_r\chi_{Q^r}$ belongs to the same permutation simplex $PS(\pi)$ for all $r = 1, \ldots, |\mathcal{J}|$,

$$\eta'(\tau; y) = \eta'\left(\tau; \sum_{r=1}^{|\mathcal{J}|} \beta_r\chi_{Q^r}\right) = \sum_{r=1}^{|\mathcal{J}|} \beta_r\eta'(\tau; \chi_{Q^r}).$$

The nonnegativity of $\eta'(\tau; \chi_{Q^r})$, $r = 1, \ldots, |\mathcal{J}|$, follows from (14). Furthermore, $\eta'(\tau; \chi_{Q^1}) = 0$. Thus, we conclude that $\eta'(\tau; y)$ is nonnegative. □

Selecting Q to minimize function ϑ in (11) is called the *cluster splitting* problem. Theorem 4 shows that there is a descent direction from a cluster if and only if the corresponding cluster splitting problem has a negative optimal value.

3. UNIFORM FILL RATE PRODUCTION

In Section 3.1, we introduce a specific capacity allocation policy, called the uniform fill rate production policy, that assigns capacity to product families in the case of demand shortfall. Under this policy, Section 3.2 shows the submodularity of the objective function in the cluster-splitting problem. Submodularity is related to global convexity and polynomial provability of splitting. With an alternative linear program based allocation, the objective function in the splitting problem is not submodular.

3.1. Description

This section explains the uniform fill rate production policy, and the computation of instantaneous lost sales cost.

When the capacity is insufficient to meet 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 determine the lost sales $((d_{pt} - v_{pt})^+|p \in \mathcal{P})$. One plausible way of determining the lost sales at t is to minimize the total instantaneous lost sales cost by solving an allocation linear program: to minimize $\sum_{p\in\mathcal{P}} c_{pt}(d_{pt} - v_{pt})^+$ subject to constraints (2)–(4). IBM uses a similar linear program to reflect the available manufacturing capacity (see Bermon and Hood [6]). 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. One possible way to avoid this problem is to add more constraints or a penalty function in the objective to ensure that products with low profitability are treated reasonably well. The second consequence is that the objective function η does not admit a structure that enables an efficient solution method.

For the rest of this paper, we use an alternative allocation policy that equalizes the instantaneous fill rates of stochastic portion of demand at time t across all products. It approximates the current practice of at least one major U.S. semiconductor manufacturer. Computational results in Huh, Roundy, and Çakanyıldırım [21] suggest that the choice between these two allocation policies does not significantly affect the output of the capacity planning algorithm.

We conceptually divide the demand D_t into a deterministic portion $b_t \geq 0$ and a stochastic portion $D_t - b_t$. 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. The demand at time t can be written as a vector $d_t = b_t + \delta\phi$, where $\phi = (\phi_p|p \in \mathcal{P}) \in \mathbb{R}^{\mathcal{P}}$ is a directional unit vector with $|\phi| = 1$, and $\delta \in \mathbb{R}$ is the magnitude of $d_t - b_t$ along ϕ. In the recourse at time t, after the demand d_t is realized, we select production quantities such that

$$\nu(d_t, Q_t^\tau) = b_t + \zeta\phi \qquad \text{for some } \zeta \in [0, \delta], \quad (15)$$

where $Q_t^\tau = \{j : \tau_j \leq t\}$ is the set of tools available at time t. Thus, the production vector $\nu(d_t, Q_t^\tau)$ also lies on the ray defined by the starting point b_t, and the direction ϕ. The value ζ indicates the magnitude of production along this ray. It is easy to see that the fill rate of the stochastic part of the demand for product p is $(\nu_p(d_t, Q_t^\tau) - b_t)/(d_{pt} - b_t) = \zeta/\delta$, 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* $b_t = 0$. Thus, Eq. (15) becomes

$$\nu(d_t, Q_t^\tau) = \zeta\phi \qquad \text{for some } \zeta \in [0, \delta]. \quad (16)$$

We choose ζ as to maximize production along the ray without exceeding the capacity of Q_t^τ. Thus, from (2)–(4), it can be shown that $\zeta = \min\{\zeta_\phi^{Q_t^\tau}, \delta\}$, where ζ_ϕ^Q for $Q \subseteq \mathcal{J}$ and unit vector ϕ is defined as

$$\zeta_\phi^Q = \min_{m\in\mathcal{M}} \frac{\mu_m(Q)}{\sum_{p\in\mathcal{P}} U(m,p)\phi_p} = \min_{m\in\mathcal{M}} \frac{\sum_{n:(m,n)\in Q'} u_{mn}}{\sum_{p\in\mathcal{P}} U(m,p)\phi_p}, \quad (17)$$

where $Q' \subseteq Q$ is the maximal initial set of Q given the precedence relations between tools. The last equality comes from the definition of $\mu_m(Q)$ in Section 2.1. We remark that ζ_ϕ^Q represents the maximum demand magnitude along ϕ that tool set Q can support. This capacity allocation policy is called the *uniform fill rate production* policy. While this policy may not maximize the expected profit, it ensures that no product family has a fill rate lower than other product families. Having established the capacity allocation policy, we want to find the instantaneous lost sales cost ξ introduced in Section 2.2. Suppose we fix the demand direction ϕ; i.e., we condition on the event that demand falls along the ray defined by ϕ. Let Δ_t be the random scalar corresponding to the magnitude of D_t conditioned on $D_t/|D_t| = \phi$. Equations (5) and (16) show that the lost sales cost at time t given $D_t/|D_t| = \phi$ is

$$\begin{aligned}\xi\left(Q_t^\tau, t \middle| \frac{D_t}{|D_t|} = \phi\right) &= E_{\Delta_t}\left[\sum_{p\in\mathcal{P}} c_{pt}(D_{pt} - \nu_p(d_t, Q_t^\tau))^+ \middle| \frac{D_t}{|D_t|} = \phi\right] \\ &= \left(\sum_{p\in\mathcal{P}} c_{pt}\phi_{pt}\right) E_{\Delta_t}[\Delta_t - \zeta_\phi^{Q_t^\tau}]^+. \quad (18)\end{aligned}$$

The expression $E_{\Delta_t}[\Delta_t - K]^+$ for any fixed scalar K, is a common function in inventory theory. It is typically easy to evaluate (18). If $f_{\Phi_t}(\phi)$ is the probability density of $D_t/|D_t| = \phi$, then

$$\xi(Q_t^\tau, t) = \int_\phi \xi\left(Q_t^\tau au, t \middle| \frac{D_t}{|D_t|} = \phi\right) \cdot f_{\Phi_t}(\phi)\, d\phi,$$

or otherwise if $p_{\Phi_t}(\phi)$ is the probability mass function of $D_t/|D_t| = \phi$, then

$$\xi(Q_t^\tau, t) = \sum_\phi \xi\left(Q_t^\tau, t \middle| \frac{D_t}{|D_t|} = \phi\right) \cdot p_{\Phi_t}(\phi).$$

3.2. Cluster Splitting and Submodularity

This section shows that the cluster splitting problem is a submodular function minimization problem under the uniform fill rate production policy. It also proves that the directional derivatives are nondecreasing across the boundaries of permutation simplices.

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_0; 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 $[0, T]$. We recall from Section 2.4 that the splitting problem is to choose $Q \subseteq \mathcal{J}$ as to minimize ϑ, which becomes by (12)

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

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 $h_\mathcal{J}(t_o) + g_\mathcal{J}^\varnothing(t_o)$ is independent of Q, our problem is equivalent to finding $Q \subseteq \mathcal{J}$ minimizing $-h_Q(t_o) - g_Q^\varnothing(t_o)$.

Thus, in order to find a descent direction, we want to solve the cluster splitting problem. We present an efficient way of solving the cluster splitting problem under uniform the fill rate production policy (Section 3.1). We achieve this by reducing the cluster splitting problem to a submodular function minimization problem.

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 $Q_1, Q_2 \subseteq \mathcal{W}$. It can be easily shown that ρ is submodular if and only for any $Q_o \subseteq \mathcal{W}$ and $w_1, w_2 \in \mathcal{W}$, we have $\rho(Q_o + w_1) + \rho(Q_o + w_2) \geq \rho(Q_o + w_1 + w_2) + \rho(Q_o)$ (see, for example, Nemhauser and Wolsey [27]). We also say that ρ is *modular* if the above inequality is replaced with an equality, and that ρ is *supermodular* if $-\rho$ is submodular. The following proposition is used in Theorem 6 to show that the splitting problem is a submodular function minimization problem.

PROPOSITION 5: For any unit vector $\phi \in \mathbb{R}^\mathcal{P}$, let ζ_ϕ^Q, $Q \subseteq \mathcal{J}$, be defined as in (17). Then, ζ_ϕ^Q is supermodular in Q.

PROOF: Suppose $Q_o \subseteq \mathcal{J}$ and $j_1, j_2 \in \mathcal{J}\backslash Q_o$ such that $j_1 \neq j_2$. From (17), ζ_ϕ^Q is monotone nondecreasing in Q, and it follows $\zeta_\phi^{Q_o+j_1}, \zeta_\phi^{Q_o+j_2} \leq \zeta_\phi^{Q_o+j_1+j_2}$.

We claim that either $\zeta_\phi^{Q_o} = \zeta_\phi^{Q_o+j_1}$ or $\zeta_\phi^{Q_o} = \zeta_\phi^{Q_o+j_2}$ holds. Assume, by way of contradiction, that both $\zeta_\phi^{Q_o} < \zeta_\phi^{Q_o+j_1}$ and $\zeta_\phi^{Q_o} < \zeta_\phi^{Q_o+j_2}$ hold. Let $m_o \in \mathcal{M}$ minimize the right-side expression of $\zeta_\phi^{Q_o}$ in (17). Then,

$$\frac{\sum_{n:(m_o,n)\in(Q_o)'} u_{m_o n}}{\sum_{p\in\mathcal{P}} U(m_o, p)\phi_p} = \zeta_\phi^{Q_o} < \zeta_\phi^{Q_o+j_1} \leq \frac{\sum_{n:(m_o,n)\in(Q_o+j_1)'} u_{m_o n}}{\sum_{p\in\mathcal{P}} U(m_o, p)\phi_p},$$

where $(\cdot)'$ indicates the initial set. Thus,

$$\sum_{n:(m_o,n)\in(Q_o)'} u_{m_o n} < \sum_{n:(m_o,n)\in(Q_o+j_1)'} u_{m_o n}.$$

Let n_1 be the smallest index of tool group m_o such that $n_1 \notin (Q_o)'$. It follows $(m_o, n_1) \in (Q_o + j_1)'\backslash(Q_o)'$, which indicates that (m_o, n_1) and all of its preceding tools are in $Q_o + j_1$, but not in Q_o. Thus, it follows that $j_1 = (m_o, n_1)$. Similarly we can show $j_2 = (m_o, n_1)$, which however contradicts $j_1 \neq j_2$.

Thus, we conclude $\zeta_\phi^{Q_o} + \zeta_\phi^{Q_o+j_1+j_2} \geq \zeta_\phi^{Q_o+j_1} + \zeta_\phi^{Q_o+j_2}$. □

THEOREM 6: Under the uniform fill rate production policy, the cluster splitting function $\vartheta_{t_o}(\cdot)$ is submodular.

PROOF: Since $h_Q(t_o)$ is modular in Q and (7) holds, it suffices to show that $\xi(Q, t_o)$ is submodular in Q. We condition on $D_t/|D_t| = \phi$. Defining ζ_ϕ^Q as in (17), Proposition 5 shows that ζ_ϕ^Q is supermodular in Q. The sum of submodular functions is also submodular. Thus, from (18), we obtain $\xi(Q) + \xi(Q + i + j) \leq \xi(Q + i) + \xi(Q + j)$ as desired. □

Two directional unit vectors are called antiparallel if their sum is a zero vector. Since η is differentiable in each permutation simplex, the sum of any pair of antiparallel directional derivatives results in a zero vector at an interior point of a permutation simplex. The following proposition examines the behavior of the directional derivative as it crosses the boundary of permutation simplices, and shows that it does not decrease across the boundary.

PROPOSITION 7: Under the uniform fill rate production policy, 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.

PROOF: Without loss of generality, assume $\tau_j = t_o$ for all $j \in \mathcal{J}$. (See Proposition 3). Let $\chi_Q \in \{0, 1\}^\mathcal{J}$ be the

binary indicator vector of Q. For the set function ϑ_{t_o} on $2^{\mathcal{J}}$, we define a *linear extension* $\hat{\vartheta}_{t_o} : [0, 1]^{\mathcal{J}} \to \mathbb{R}$ such that:

i. $\hat{\vartheta}_{t_o}(\chi_Q) = \vartheta_{t_o}(Q)$ for any $Q \subseteq \mathcal{J}$.
ii. For any z in the permutation simplex $PS(\pi)$ defined by some $\pi \in \Pi$, let $\hat{\vartheta}_{t_o}(z)$ be the value at z of the hyperplane defined by $N + 1$ points in $\{\chi_{\pi(\{1,2,\ldots,r\})} | r = 0, 1, \ldots, N\}$, i.e. $\hat{\vartheta}_{t_o}(z) = H_{t_0}(z)$ for some matrix H satisfying i.

Lovász [25] shows that any set function is submodular if and only if its linear extension is convex. By Theorem 6, $\vartheta_{t_o}(\cdot)$ is submodular, and it follows $\hat{\vartheta}_{t_o}(\cdot)$ is convex. Thus, for any $Q \subseteq \mathcal{J}$,

$$\vartheta_{t_o}(\mathcal{J}\backslash Q) + \vartheta_{t_o}(Q) \geq \hat{\vartheta}_{t_o}(\tfrac{1}{2}\chi_Q + \tfrac{1}{2}\chi_{\mathcal{J}\backslash Q}) = \hat{\vartheta}_{t_o}(\tfrac{1}{2}\chi_{\mathcal{J}}).$$

By the definition of linear extension and (11), the right-most expression of the above becomes

$$\hat{\vartheta}_{t_o}(\tfrac{1}{2}\chi_{\mathcal{J}}) = \tfrac{1}{2}\vartheta_{t_o}(\chi_{\mathcal{J}}) + \tfrac{1}{2}\vartheta_{t_o}(\chi_{\varnothing}) = 0.$$

However, from the expression (12) applied to $\vartheta_{t_o}(Q)$, and the expression (13) applied to $\vartheta_{t_o}(\mathcal{J}\backslash Q)$,

$$\vartheta_{t_o}(Q) + \vartheta_{t_o}(\mathcal{J}\backslash Q) = -2[h_Q(t_o) + g_Q^{\varnothing}(t_o)] + 2[h_Q(t_o) + g_Q^{\mathcal{J}\backslash Q}(t_o)] = 2[\eta'(\tau; -\chi_Q) + \eta'(\tau; \chi_Q)].$$

Thus, $\eta'(\tau; -\chi_Q) + \eta'(\tau; \chi_Q) \geq 0$, and it holds with equality if $Q = \mathcal{J}$. Now, apply the argument in the proof of Theorem 4 to extend this to an arbitrary direction y. □

4. DIVIDE-AND-CONQUER ALGORITHM

In Section 4.1, we outline an efficient divide-and-conquer algorithm for the problem of minimizing the total cost η. Our algorithm resembles that of Hochbaum and Queyranne [19] for the convex cost closure problem. This algorithm finds a solution that satisfies the first-order necessary and sufficient condition for the local optimality of (P) in Section 2.1—namely, this solution has no feasible descent direction. Section 4.2 describes the correctness and computational complexity for our algorithm.

4.1. Description

The algorithm in Hochbaum and Queyranne [19] finds a global minimizer of a separable convex function subject to precedence constraints as in our formulation. We relax the separability assumption to quasiseparability. Our algorithm finds a local minimizer of a quasiseparable, nonconvex objective function. Section 5.1 shows that the same algorithm finds a global minimizer of a quasiseparable, convex function.

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 require 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 clusters, 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 $Q \subseteq C$ be its optimal solution; i.e., let Q minimize $\rho_{\omega_C}(\cdot | Q_L, C, Q_U))$, where $Q_L(Q_U)$ is the union of all clusters preceding (succeeding) C in $\mathcal{C}$ and $Q \subseteq C$. If the optimal value is nonnegative, set $lb(C) = ub(C) = \omega_C$. Otherwise, replace C with Q and $C\backslash Q$ in $\mathcal{C}$, where Q precedes $C\backslash Q$. Let $lb(Q) = lb(C)$, $ub(Q) = \omega_C$, $lb(C\backslash Q) = \omega_C(C)$ and $ub(C\backslash Q) = ub(C)$.
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 breadth-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 $\int_{t=lb(C)}^{\omega_C} g_C^{Q_L}(t) + h_C(t)\, dt$ as we vary the value $\omega_C = \tau_j$, $j \in C$ over the interval $[lb(C), ub(C)]$ [see (10)]. 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 9 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.

PROPOSITION 8: For fixed t, let Q^* be a local minimizer of the cluster splitting function $\vartheta_t(\cdot|Q_L, Q_o, Q_U)$. Then we have

$$h_B(t) + g_B^{Q_L \cup Q^* \backslash B}(t) \geq 0 \qquad \text{for } B \subseteq Q^*$$

and

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

PROOF: For 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 $B \subseteq Q^*$ and $\bar{B} = Q^* \backslash B$. By the optimality of Q^* and (8),

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

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

THEOREM 9: At each iteration of the divide-and-conquer algorithm, there exists some solution τ^* with no descent direction in (P) such that $\tau_j^* \in [lb(C), ub(C)]$ for all $j \in C$ and $C \in \mathcal{C}$. If the algorithm terminates, we have found such a solution.

PROOF: We consider one step of the divide-and-conquer method, in which the algorithm splits cluster C at time t, where Q_L is the union of the clusters preceding cluster C. Let Q^* minimize $\vartheta_t(\cdot|Q_L, C, \mathcal{J}\backslash(Q_L \cup C))$, and let $\bar{Q}^* = C \backslash Q^*$. Let lb^0 and ub^o (lb' and ub') be the lower bound lb and upper bound ub before (after) splitting C. Let τ^* be the output of the algorithm if it terminates. Otherwise, it can be shown that the algorithm is bisection-based, and the maximum gap between ub and lb converges to 0; let τ^* be the limit. We continue this proof assuming the optimization-based method is used; the other case is similar.

We use the backward induction to show the nonexistence of a descent direction at τ^* subject to bounds lb and ub. We assume that the result holds for lb' and ub'. (Clearly, it holds for the final lb and ub.) During the iteration, the interval defined by $[lb^o(C), ub^o(C)]$ is subdivided into $[lb^o(C), t]$ and $[t, ub^o(C)]$. Thus, it suffices to show that there is no descent direction $y \in \mathbb{R}^{\mathcal{J}}$ such that the support of y is included in $B = J_t(\tau^*)$. Furthermore, by Theorem 4, it suffices to show $\vartheta_t(S|Q_L \cup (Q^* \backslash B), B, (\mathcal{J}\backslash(Q_L \cup C)) \cup (\bar{Q}^* \backslash B)) \geq 0$ for all $S \subseteq B$. We let $B_1 = Q^* \cap B$ and $B_2 = \bar{Q}^* \cap B$.

For simplicity of argument, we assume $(Q_L, C, \mathcal{J}\backslash(Q_L \cup C)) = (\emptyset, \mathcal{J}, \emptyset)$ (see Proposition 3). Applying the KKT condition associated with the bounds lb' and ub', B_1 and B_2 satisfy

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

The above inequalities hold with equality by Proposition 8. It follows that $\vartheta_t(B_1|\bar{B}_1, B, \bar{B}_2) = 0$, where $\bar{B}_1 = Q^* \backslash B_1$ and $\bar{B}_2 = \bar{Q}^* \backslash B_2$. Let $S \subseteq B$. By (8), (11), and the choice of Q^*,

$$\begin{aligned} -(h_{\bar{B}_1}(t) + g_{B_1}^{\emptyset}(t)) + \vartheta_t(B_1|\bar{B}_1, B, \bar{B}_2) + (h_{\bar{B}_2}(t) + g_{B_2}^{Q^* \backslash \bar{B}_2}(t)) \\ = \vartheta_t(\bar{B}_1 \cup B_1|\emptyset, \mathcal{J}, \emptyset) \leq \vartheta_t(\bar{B}_1 \cup S|\emptyset, \mathcal{J}, \emptyset) = -(h_{\bar{B}_1}(t) \\ + g_{B_1}^{\emptyset}(t)) + \vartheta_t(S|\bar{B}_1, B, \bar{B}_2) + (h_{\bar{B}_2}(t) + g_{B_2}^{Q^* \backslash \bar{B}_2}(t)). \end{aligned}$$

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

In the analysis of the complexity of the algorithm, we use the fact that the running time of a submodular function minimization on $|\mathcal{J}|$ variables is $O(|\mathcal{J}|^7)$ as shown in Iwata [22]. [This assumes the evaluation of a function is at most $O(|\mathcal{J}|)$.]

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 $|\mathcal{J}|$ submodular function minimization computations. Let γ be the time required to find a local minimizer of a real-valued one-dimensional function of the form $\int_{t=lb(C)}^{\tau} g_C^{Q_L}(t) + h_C(t)\, dt$, where τ ranges in an interval $[lb(C), ub(C)]$. Then, the running time of the algorithm is bounded by $O(|\mathcal{J}|\gamma + |\mathcal{J}|^8)$.

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 submodular function minimization computations of a given depth of the divide-and-conquer tree is bounded by that of one submodular function minimization on $|\mathcal{J}|$ variables. 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 $O(|\mathcal{J}|^7 \log T\varepsilon^{-1})$.

These running times can be reduced significantly if we employ the demand model as given in Section 5.

5. THE DEMAND MODEL AND COMPUTATIONAL RESULTS

In Section 5.1, we present a specific demand model which enables our algorithm to find a global minimum solution. Section 5.2 presents a sufficient condition for demand, under which the splitting problem eventually reduces to a variation of the max-flow min-cut problem, a special case of a submodular function minimization problem. Given this condition and demand model, some computational results are reported in Section 5.3.

5.1. Stationary Product Mix Assumption 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.

The following proposition presents a sufficient condition for convexity in each permutation simplex.

PROPOSITION 10: Within any permutation simplex $PS(\pi)$, $\pi \in \Pi$, $\eta(\cdot)$ is convex if for all $r \in \{1, \ldots, |\mathcal{J}|\}$, the instantaneous benefit $g_{\pi(r)}^{\pi(\{1,2,\ldots,r-1\})}$ of having tool $\pi(r) \in \mathcal{J}$ at time t is nondecreasing with respect to t.

PROOF: Since the price $P_j(t)$ of tool j 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 $\tau \in [0, T]^{\mathcal{J}}$. If $g_{\pi(r)}^{\pi(\{1,2,\ldots,r-1\})}$ is nondecreasing in t for any $r \in \{1, \ldots, |\mathcal{J}|\}$, then the convexity of $\eta^{LS}(\tau)$ follows from Proposition 1. □

We say that the demand D_t satisfies the *stationary product mix assumption* provided that:

1. The probability density of $\Phi_t = D_t/|D_t|$ is independent of t.
2. The demand magnitude $\Delta_t = |D_t|$, given that $\Phi_t = \phi_t$ 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 the product mix $\Phi_t = D_t/|D_t|$ is stationary. However, Δ_t is not stationary, so $E(D_t)/|E(D_t)|$ is not necessarily stationary. We model demand at the product family level, where a product family is a group of products which require the same or proportional utilization of tool groups, so the second assumption is plausible. These three assumptions are quite strong, but seem needed 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 global convexity.

PROPOSITION 11: 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 the interior of each permutation simplex.

PROOF: Suppose a permutation simplex $PS(\pi)$ is defined by $\pi \in \Pi$. Proposition 1 implies that the partial derivative of η^{LS} with respect to $\tau_{\pi(r)}$, $r \in \{1, \ldots, |\mathcal{J}|\}$, is $g_{\pi(r)}^{\pi(\{1,\ldots,r-1\})}(t)$, and it suffices to show that this function is nondecreasing.

We condition on $\Phi_t = \phi = (\phi_p | p \in \mathcal{P})$. By the stationary product mix assumption, the probability density of such an event is independent of t. Let $Q^r = \pi(\{1, \ldots, r\})$ and $Q^{r-1} = \pi(\{1, \ldots, r-1\})$. From (18),

$$\xi\left(Q^{r-1}, t \middle| \frac{D_t}{|D_t|} = \phi\right) - \xi\left(Q^r, t \middle| \frac{D_t}{|D_t|} = \phi\right) = \sum_{p \in \mathcal{P}} c_{pt} \phi_p E_{\Delta_t}[(\Delta_t - \zeta_\phi^{Q^{r-1}})^+ - (\Delta_t - \zeta_\phi^{Q^r})^+ \Big| \frac{D_t}{|D_t|} = \phi$$

is nondecreasing in t since $\zeta_\phi^{Q^{r-1}} \le \zeta_\phi^{Q^r}$, both $\zeta_\phi^{Q^{r-1}}$ and $\zeta_\phi^{Q^r}$ are independent of t, and Δ_t is stochastically nondecreasing. It follows from (5) and (6) that $g_{\pi(r)}^{\pi(\{1,\ldots,r-1\})}(t)$ 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 shows the global convexity of η.

PROPOSITION 12: With the uniform fill rate production and the stationary product mix assumption, if η is convex in each permutation simplex, then η is globally convex.

PROOF: Let $\eta_L(s)$ be the function η restricted to some line segment $L \subseteq \mathbb{R}^{\mathcal{J}}$, parametrized by $s \in [0, 1]$. It suffices to show that η_L is convex. In each permutation simplex, η 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 7. Therefore, by Proposition B.2 in Bertsekas [7] or alternatively by the monotonicity of the "presubdifferential" in Correa, Jofré, and Thibault [14], we conclude that η_L is convex. □

From Propositions 11 and 12, we have the following theorem:

THEOREM 13: 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. Without the uniform fill rate production and the stationary product mix assumption, one can show that η is not globally convex. Thus, we make a progress towards defining the dividing point between polynomial solvability and hard versions of the capacity planning problem.

5.2. Faster Cluster Splitting Based on the Minimum Cut Problem

In this section, we assume a finite support for the directional vector $\Phi_t = (D_t - b_t)/|D_t - b_t|$ (but not for D_t), and show that the cluster splitting problem can be reduced to the classical minimum cut problem. We also improve the running time of the divide-and-conquer algorithm by applying a parametric optimization result.

The demand model with a *finite number of rays* can be written as

$$D_t = b_t + \Delta_{I_t,t}\phi_{I_t,t}, \tag{19}$$

where 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 nonnegative unit-norm directional vector in $\mathbb{R}^{\mathcal{P}}$; and Δ_{it} is a continuous nonnegative random variable representing the magnitude of $D_t - b_t$ along ϕ_{it}. The support of D_t consists of a finite number of rays indexed by I_t and emanating from b_t. This demand model was first introduced in Roundy et al. [29]. For simplicity of exposition, assume $b_t = 0$ as before.

Currently, most models of high-dimensional random vectors are either continuous (e.g., multivariable normal) or discrete (e.g., multinomial). This 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 in the direction of ϕ_{it}, and has measure zero. Note $\Delta_{I_t,t} = |D_t - b_t|$. Thus Δ_{it} is the conditional magnitude of $D_t - b_t$, conditioned on ray i being chosen. It is shown in Roundy et al. [29] 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. [29] indicate that in a 4-dimensional space, 64 rays provided an approximation to a multivariable log normal distribution that was sufficiently accurate for a capacity planning problem.

We show how this demand model enables us to solve the cluster splitting problem faster by reducing it to a classical open-pit mining problem (see, for example, Chvátal [13]). The open-pit mining problem is to find a subset of blocks to be excavated as to minimize the sum of costs associated with each excavated block. The subset should obey a set of precedence relationships.

THEOREM 14: Under the uniform fill rate production, the cluster splitting problem is reduced to the open-pit mining problem.

PROOF: Without loss of generality, assume that there is only one cluster $J_{t_o}(\tau) = \mathcal{J}$ at t_o. The cluster splitting problem is to minimize $\vartheta_{t_o}(Q)$ where $Q \subseteq \mathcal{J}$. From (12), we want to purchase a subset $Q \subseteq \mathcal{J}$ of tools earlier by $\varepsilon > 0$ from t_o to $t_o - \varepsilon$. The instantaneous rate of increase in the purchase cost η^P is $-\sum_{j\in Q} dP_j(t)/dt|_{t=t_o} = -h_Q(t_o)$, and the rate of decrease in the expected lost sales cost η^{LS} is $g_Q^{\varnothing}(t_o) = \xi(\varnothing, t_o) - \xi(Q, t_o)$ [see (7)].

Suppose $I_{t_o} = i$. Along the fixed demand ray ϕ_{it_o}, any tool set that does not contain tool j cannot support a production vector v_{t_o} whose magnitude $|v_{t_o}|$ is greater than $\zeta_{\phi_{it_o}}^{\mathcal{J}\setminus j}$ by Definition (17). We use $\psi_{it_o} : \{1, \ldots, |\mathcal{J}|\} \to \mathcal{J}$ to define a permutation sequence of tools in the order in which they constrain capacity along demand ray ϕ_{it_o}; thus $\zeta_{\phi_{it_o}}^{\mathcal{J}\setminus\phi_{it_o}(1)} \le \zeta_{\phi_{it_o}}^{\mathcal{J}\setminus\phi_{it_o}(2)} \le \cdots \le \zeta_{\phi_{it_o}}^{\mathcal{J}\setminus\phi_{it_o}(|\mathcal{J}|)}$. Purchasing tool $\psi_{it_o}(r)$ before $\psi_{it_o}(r-1)$ does not contribute to the maximum magnitude along ϕ_{it_o} that a tool set can support.

The blocks of the open-pit mining problem consist of both the tool set $\mathcal{J}$, and the set $\{(i, r)|i \in I_{t_o}, r \in \{1, \ldots, |\mathcal{J}|\}\}$ of pairs. The pair (i, r) represents the subset $\{\psi_{it_o}(1), \ldots, \psi_{it_o}(r)\}$ of tools of size r along the demand ray i. (We use the subscript i to represent values associated with the corresponding demand ray.) The cost associated with block $j \in \mathcal{J}$ is $-dP_j(t)/dt|_{t=t_o} = -h_j(t_o) \ge 0$. The cost associated with block (i, r) is the negative of the marginal benefit of the last tool of the effective tool set for demand ray i, which is, by (18),

$$w_{it_o} \sum_{p\in\mathcal{P}} c_{pt}\phi_{ipt_o}E_{\Delta_{it_o}}[(\Delta_{it_o} - \zeta_{it_o}^{\{\psi_{it_o}(1),\ldots,\psi_{it_o}(r)\}})^+ - (\Delta_{it_o} - \zeta_{it_o}^{\{\psi_{it_o}(1),\ldots,\psi_{it_o}(r-1)\}})^+] \le 0.$$

The precedence arcs specify that the benefit of a tool cannot be obtained before it is purchased; i.e., we have the arc $j \to (i, r)$, where $\psi_{it_o}(r) = j$. They also specify that the order of effective tools should be maintained; i.e., we have the arc $(i, r-1) \to (i, r)$ for $r = 2, 3, \ldots, |\mathcal{J}|$.

The objective function becomes $C(Q) = -h_Q(t_o) - g_Q^{\varnothing}(t_o)$, for $Q \subseteq \mathcal{J}$. Any feasible solution of this open-pit

Table 1. Computational result for the capacity planning models: With demand satisfying stationary product mix assumption.

Number of demand rays		2	4	8	16	32	64	128
Discrete time model	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 (s)*	0.92	1.42	3.75	8.50	12.81	26.81	50.22
	MATLAB CPU time (s)	29.82	53.54	104.33	199.22	407.03	803.58	1613.25
Continuous time model	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 (s)	1.42	1.35	1.76	2.15	2.32	3.51	5.25
	MATLAB CPU time (s)	8.84	12.18	25.33	44.97	74.51	162.38	331.67

* This includes, unlike Roundy et al. [29], the time required for file input/output.

mining 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 mining problem is at least $-(1/2)[h_{\mathcal{J}}(t_o) + g^{\varnothing}_{\mathcal{J}}(t_o)]$, then by (12), 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 mining problem eventually is reduced, the number of nodes and the number of arcs are proportional to the product of the number of tools in the cluster and the maximum number $|\mathcal{I}|$ of rays used in modeling demand. Thus, using that the running time of finding a minimum-cut of a network (V, A) is $O(|V|\,|A|\log(|V|^2/|A|))$ the running time of our divide-and-conquer algorithm becomes $O(|\mathcal{J}|\gamma + |\mathcal{J}|^3|\mathcal{I}|^2(\log|\mathcal{J}| + \log|\mathcal{I}|))$ for the optimization-based method, and $O(|\mathcal{J}|^2|\mathcal{I}|^2(\log|\mathcal{J}| + \log|\mathcal{I}|)\log T\varepsilon^{-1})$ for the bisection-based method.

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 [18] 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 $O(|\mathcal{J}|\,|\mathcal{I}|(\log|\mathcal{J}| + \log|\mathcal{I}|)(|\mathcal{J}|\,|\mathcal{I}| + \log T\varepsilon^{-1})$ for the bisection-based method. For a fixed γ and ε, these bounds are asymptotically the 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.

Furthermore, if the stationary product mix assumption holds, it can be seen that our problem (P) reduces to the minimization of the sum of separable convex functions where the variables x_b are times associated with each block b of the open-pit mining formulation. It is subject to precedence arc set constraints; i.e., arc $b_1 \to b_2$ specifies $x_{b_1} \leq x_{b_2}$. The resulting problem is a convex cost closure problem, which can be solved in the same asymptotic running time as our divide-and-conquer algorithm with monotonic parametric min-cut (see Hochbaum and Queyranne [19]).

5.3. 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. [29]. 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 the solution of 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. [29] by modifying the cost for tool purchases such that it is close to zero at the end of the planning horizon. We test both cases where the stationary product mix assumption on demand holds and does not hold. For the first case, the demand data is modified to satisfy the stationary product mix assumption with a finite number of rays. 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

Table 2. Computational result for the capacity planning models: Without demand satisfying stationary product mix assumption.

Number of demand rays		2	4	8	16	32	64	128
Discrete time model	Lost sales cost (million $)	53.32	55.81	64.65	68.14	65.80	64.92	65.69
	Purchase cost (million $)	155.94	155.82	146.59	146.27	153.26	157.26	159.19
	Total cost (million $)	209.26	211.63	211.24	214.41	219.06	222.18	224.89
	HIPR CPU time (s)*	0.79	1.34	2.33	5.17	8.88	18.89	53.17
	MATLAB CPU time (s)	24.06	48.09	93.18	184.01	364.49	750.66	1616.55
Continuous time model	Lost sales cost (million $)	48.56	63.23	62.95	62.90	53.54	57.69	55.89
	Purchase cost (million $)	173.33	158.49	152.94	155.96	168.45	166.69	127.31
	Total cost (million $)	221.89	221.73	215.89	218.86	221.99	224.37	227.20
	HIPR CPU time (s)	4.87	5.56	5.36	4.58	6.62	7.42	11.43
	MATLAB CPU time (s)	40.52	68.13	96.48	131.06	242.96	515.20	1055.70

* This includes, unlike Roundy et al. [29], the time required for file input/output.

Optiplex GX270 personal computer with a Pentium 4 3.00 GHz processor and 2.0 GB of RAM.

Table 1 compares the performances of the discrete-time model with the continuous-time model when the stationary product mix assumption holds. The first half of the table summarizes the performance of the discrete-time model, which makes quarterly decisions for the next four years. The algorithm in Roundy et al. [29] is based on transforming the problem into one big network flow problem that includes a set of nodes for each discrete time period. It consists of two parts: generating a network using MATLAB 5.3, 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 [12]. Each running time in the table is an average of four trials.

The second half of Table 1 reports the performance of the optimization-based divide-and-conquer method for the continuous-time model, without taking advantage of either the parametric minimum cut network, or the convex cost closure set problem. Once the optimal solution τ^* is obtained, it is difficult to evaluate $\eta(\tau^*)$ since $\eta^{LS}(\tau^*)$ in (1) involves an integral. We approximate the optimal value $\eta^{LS}(\tau^*)$ by computing $\xi(Q_t^{\tau^*}, t)$ for $t = 1, 2, \ldots, T$ and taking a summation. Thus, we expect the approximate optimal values to be close to, but not as good as, those reported by the discrete model. This method uses HIPR several times, and the sum of running times on all HIPR calls are reported. All other operations, including the descent methods, are included in the MATLAB running time. Each of the four repetitions yielded the same solution, and their average running times are reported.

Table 1 shows that both models find solutions that are very close. The differences (less than 0.5%) in the cost are due to the approximation of the optimal values. They are less than a fraction of 1%. 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 on smaller networks.

Table 2 shows similar results for the case when the stationary product mix assumption does *not* hold, as in the original demand data of Roundy et al. [29]. In such case, the discrete-time model yields a globally optimal solution whereas the continuous-time model does not. However, the optimality gap is within 2% when the number of rays is greater than 8.

Computational results suggest that the divide-and-conquer method produces the optimal solution for the given data set, as verified by the discrete-time model. Compared to the discrete-time model, its running time grows moderately. Reduction in running time implies that a bigger and more complex model can be solved, and that the capacity planning model can be effectively used to support real-time decision-making such as lead-time quotation (Dietrich [16]).

6. CONCLUSIONS

This paper addresses the capacity planning for multiple tool types that are shared by multiple product families. Under demand uncertainty, we have shown how problems of realistic size and complexity can be modeled and how these models can be solved efficiently using a continuous-time model. 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. Possible extensions of our model to incorporate some generalizations that are of interest to the semiconductor industry. Possible generalizations include: capacity contraction, tool re-

tirement, mean demand that is not necessarily increasing, complex tool-product relationships (such as technology evolution a tool group), 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 nonconvex minimization problems for which local descent algorithms consistently yield very good solutions. It is hoped that a modification of the algorithm presented in this paper will obtain good solutions with these generalizations.

ACKNOWLEDGMENTS

The authors would like to thank Metin Çakanyıldırım, Lisa Fleischer, Dorit Hochbaum, Tom McCormick, Michael Todd, and Feng Zhang for their assistance and insightful comments in preparing this document. We also thank anonymous referees for their careful and thorough comments. Research was supported by the Semiconductor Research Corporation Task ID: 490.001 and Graduate Fellowship Program, as well as the Natural Science and Engineering Council of Canada Postgraduate Scholarship.

REFERENCES

[1] S. Ahmed and N. Sahinidis, An approximation scheme for stochastic integer programs arising in capacity expansion, Oper Res 51 (2003), 461–471.
[2] A. Atamturk and D.S. Hochbaum, Capacity acquisition, subcontracting, and lot sizing, Management Sci 47 (2001), 1081–1100.
[3] F. Barahona, S. Bermon, O. Gunluk, and S. Hood, Robust capacity planning in semiconductor manufacturing, Technical Report RC22196, IBM Research Division, Yorktown Heights, NY, 2001.
[4] D.L. Benavides, J.R. Duley, and B.E. Johnson, As good as it gets: Optimal fab design and deployment, IEEE Trans Semiconductor Manuf 12 (1999), 281–287.
[5] O. Berman, Z. Ganz, and J.M. Wagner, A stochastic optimization for planning capacity expansion in a service industry under uncertain demand, Naval Res Logist 41 (1994), 545–564.
[6] S. Bermon and S.J. Hood, Capacity Optimization Planning System (CAPS), Interfaces 29 (1999), 31–50.
[7] D.P. Bertsekas, Nonlinear programming, Athena Scientific, Belmont, MA, 1995.
[8] S. Bhatnagar, E. Fernández-Gaucherand, M.C. Fu, Y. He, and S.I. Marcus, A Markov decision process model for capacity expansion and allocation, Proc 38th IEEE Conf Decision Control, 1999, pp. 1380–1385.
[9] M. Çakanyıldırım and R. Roundy, Optimal capacity expansion and contraction under demand uncertainty, Technical Report, School of Management, University of Texas at Dallas, Richardson, TX, 2001.
[10] M. Çakanyıldırım, R.O. Roundy, and S.C. Wood, Machine purchasing strategies under demand- and technology-driven uncertainties, Technical Report 2, School of Management, University of Texas of Dallas, Richardson, TX, 2004.
[11] Z.-L. Chen, S. Li, and D. Tirupati, A scenario based stochastic programming approach for technology and capacity planning, Comput Oper Res 29 (1998), 781–806.
[12] B.V. Cherkassky and A.V. Goldberg, On implementing push-relabel method for the maximum flow problem, Algorithmica 19(4) (1997), 390–410.
[13] V. Chvátal, Linear programming, Freeman, New York, 1983.
[14] R. Correa, A. Jofré, and L. Thibault, "Subdifferential characterization of convexity," Recent advances in nonsmooth optimization, D.-Z. Du, L. Qi, and R.S. Womersley (Editors), World Scientific, Singapore, 1995, pp. 18–23.
[15] M.H.A. Davis, M.A.H. Dempster, S.P. Sethi, and D. Vermes, Optimal capacity expansion under uncertainty, Adv Appl Probab 19 (1987), 156–176.
[16] B. Dietrich, Use of optimization with IBM's supply chain, Keynote presentation at the Second Meeting of the Value Chain Academic-Industry Consortium, Lucent Technologies, Bell Labs, Murray Hill, NJ, 22 May 2003.
[17] L.F. Escudero, P.V. Kamesam, A.J. King, and R.J.B. Wets, Production planning with scenario modelling, Ann Oper Res 43 (1993), 311–335.
[18] D. Gusfield and C. Martel, A fast algorithm for the generalized parametric minimum cut problem and applications, Algorithmica 7 (1992), 499–519.
[19] D.S. Hochbaum and M. Queyranne, Minimizing a convex cost closure set, SIAM J Discrete Math 16 (2003), 192–207.
[20] W.T. Huh, A continuous-time strategic capacity planning model based on the minimum-cut problem, Manuf Serv Oper Management 5 (2003), 63–66.
[21] W.T. Huh, R.O. Roundy, and M. Çakanyıldırım, A general strategic capacity planning model under demand uncertainty, Technical Report 1379, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 2003.
[22] S. Iwata, A faster scaling algorithm for minimizing submodular functions, SIAM J Comput 32 (2003), 833–840.
[23] E. Khmelnitsky and K. Kogan, Optimal policies for aggregate production and capacity planning under rapidly changing demand conditions, Int J Prod Res 34(7) (1996), 1929–1941.
[24] S. Li and D. Tirupati, Dynamic capacity expansion problem with multiple products: Technology selection and timing of capacity additions, Oper Res 42(5) (1994), 958–976.
[25] L. Lovász, "Submodular functions and convexity," Mathematical programming: The state of the art, Bonn 1982, A. Bachem, M. Grötschel, and B.H. Korte (Editors), Springer, Berlin, 1983, pp. 235–257.
[26] H. Luss, Operations research and capacity expansion problems: A survey, Oper Res 30(5) (1982), 907–947.
[27] G.L. Nemhauser and L.A. Wolsey, Integer and combinatorial optimization, Wiley, New York, 1988, Chap. III.3, pp. 659–819.
[28] S. Rajagopalan and J.M. Swaminathan, Coordinated production planning model with capacity expansion and inventory management, Management Sci 47 (2001), 1562–1580.
[29] R.O. Roundy, F. Zhang, M. Çakanyıldırım, and W.T. Huh, Optimal capacity expansion for multi-product, multi-machine manufacturing systems with stochastic demand, IIE Trans 36 (2004), 23–36.
[30] J.M. Swaminathan, Tool capacity planning for semiconductor fabrication facilities under demand uncertainty, European J Oper Res 120 (2002), 545–558.