The semiconductor industry has been one of the driving forces of the “new” economy. It boasts of the exponentially growing performance of semiconductor devices, coupled with rapidly decreasing chip prices; however, it faces highly volatile demands, and copes with astronomical fab costs, most of which can be attributed to tool costs. The leadtime for purchasing tools is between 6 and 18 months, upon which tools quickly become obsolete. Thus, semiconductor companies need to recover their 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. 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 leadtimes. 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 Çakanyildirim et al. (1999) and Roundy et al. (2000). Typical methods of stochastic optimization include stochastic-linear programming, stochastic-integer programming, and Markov decisions processes; yet, they have not been able to solve real-world capacity planning problems on the scale faced by the semiconductor industry.

This paper takes the stochastic-optimization approach, which explicitly incorporates randomness in the model. We assume nonstationary stochastic demand, with the expected demand for product families increasing over time. We also assume lost sales and no finished good inventory. As in Çakanyildirim 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. 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. We present an efficient divide-and-conquer algorithm that will find a locally optimal solution of this problem. A subroutine to this algorithm is the parametric minimum-cut problem.

1. Formulation
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...
We derive another expression for \( \eta^L(\tau) \) within a subset of the feasible region and develop some properties of \( \eta \). Let \( \Pi \) be the set of all permutations on \( J \), or bijective maps from \([1, \ldots, |J|]\) to \( J \). Each \( \pi \in \Pi \) corresponds to a sequence of tool purchases, and the permutation simplex defined by \( \pi \) is \( PS(\pi) = \{ \tau \in [0, T]^J | \tau_{\pi(1)} \leq \tau_{\pi(2)} \leq \cdots \leq \tau_{\pi(|J|)} \} \), which corresponds to the set of valid \( \tau \)'s for that sequence.

Suppose \( \tau \in PS(\pi) \) where \( \pi \in \Pi \). For each \( r \in [1, \ldots, |J|] \), let \( \pi^{-}(r) = (\pi(r') | r' < r) \). Suppose that \( Q_\pi = \pi^{-}(r) \) for some \( r \). The amount of reduction in the expected instantaneous lost sales cost \( \xi \) at time \( t \) by adding the tool \( \pi(r) \) to the set of available tools is denoted by \( g^{\pi^{-}(r)}(t) \). Formally we define, for any \( Q' \subseteq J \) and \( j \in \mathbb{J} \),

\[
D_j^Q(t) = \xi(Q', t) - \xi(Q' \cup \{j\}, t).
\]

Note that \( D_j^Q(t) \) is the difference, in lost sales cost, of having the tool set \( Q' \) and that of having \( Q' \cup \{j\} \) at time \( t \).

Suppose \( \tau \in PS(\pi) \), i.e., \( \tau \) follows the sequence given by \( \pi \). Then for fixed \( t \), \( Q_\pi^t = \{ j | \pi(t) \leq t \} \) can be expressed as \( \pi^{-}(r) \cup \{\pi(r)\} \) for some \( r_o \in [1, \ldots, |J|] \). Within the permutation simplex \( PS(\pi) \), the expected lost sales cost \( \eta^L \) is continuous and separable. It is also differentiable and its partial derivative with respect to \( \tau_{\pi(r)} \) is \( \frac{\partial}{\partial \tau_{\pi(r)}} \eta^L(\tau) = g^{\pi^{-}(r)}(\tau_{\pi(r)}) \). Furthermore, \( \eta^L \) is continuously differentiable if each \( g^{\pi^{-}(r)}(\tau) \), \( r \in [1, \ldots, |J|] \), is continuous with respect to \( \tau \). Whenever \( \pi(r) = j \) and \( \pi^{-}(r) = \emptyset \), we have \( \frac{\partial}{\partial \tau_{\pi(r)}} \eta^L(\tau) = 0 \). This is a much stronger separability of the expected lost sales cost \( \eta^L \) than separability in each permutation simplex. We generalize the definition of \( g \) for any disjoint sets \( Q_1, Q_2 \subseteq \mathbb{J} \) of tools, we define \( g^{Q_1 \cup Q_2}(t) = \xi(Q_1, t) - \xi(Q_1 \cup Q_2, t) \). This quantity corresponds to the marginal benefit of adding the tool set \( Q_1 \) to the existing set \( Q_2 \) at time \( t \). It can be shown that \( g^{Q_1 \cup Q_2}(t) = g^{Q_1 \cup \{j\}}(t) \) if \( Q_2 \subseteq \mathbb{J} \). It is a strong additivity property of derivatives of \( \eta^L \) that spans many neighboring permutation simplices.

We let \( h_j(t) = (d/dt)P_j(\tau) \leq 0 \) be the rate of change in the tool cost at \( t \). By the convexity of the tool cost \( P_j \), \( h_j(t) \) is nondecreasing. For \( Q \subseteq \mathbb{J} \), we set \( h_Q(t) = \sum_{j \in Q} h_j(t) \). We remark that within the permutation simplex \( PS(\pi) \) defined by \( \pi \), the objective function \( \eta \) is separable and its partial derivative with respect to \( j \) is \( (d/\partial \tau_j) \eta(\tau) = h(\tau_j) + g_j^{\pi^{-}(r)}(\tau_j) \), provided \( j = \pi(r) \).
Suppose at time $t$, we have a partition $Q_L$, $Q_o$, and $Q_U$ of $f$, 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 $Q_o$, uniformly slide $Q \subseteq Q_o$ earlier and $Q \setminus Q$ later, then $\eta$ changes at the rate of $-\left[ h_Q(t) + s_Q^2(t) \right] + \left[ h_{Q_o \setminus Q}(t) + \tilde{s}_{Q_o \setminus Q}(t) \right]$. Minimizing this expression is called the cluster-splitting of $Q_o$ given $Q_L$ and $Q_U$.

2. Demand Modeling

As in Roudy 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., $D_t = b_t + \Delta_{b-t} + \phi_{b-t}$, where $b_t = \{b_{pt} \mid p \in \mathcal{P} \} \in \mathbb{R}^m$ is a deterministic nonnegative vector that is nondecreasing in $t$; $\phi_{b-t}$ is a deterministic nonnegative unit-norm direction vector in $\mathbb{R}^m$; and $\Delta_{b-t}$ is a deterministic nonnegative random scalar along $\phi_{b-t}$. Intuitively, the demand $D_t$ is determined by starting at $b_t$, randomly selecting a direction by observing $\phi_{b-t}$ and moving a random distance $\Delta_{b-t}$ in the direction $\phi_{b-t}$.

Currently, most models of high-dimensional random vectors are either continuous (e.g., multivariable normal) or discrete (e.g., multinomial). Our demand model is a hybrid of both: No point in $\mathbb{R}^m$ 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 Roudy et al. (2000) that by a variance-reduction technique called conditioning, our demand model can approximate a continuous distribution in $\mathbb{R}^m$ more accurately than the conventional method of sampling points, provided that the number of vectors is the same as the number of points.

There is no demand shortfall if the capacity $\mu_m(Q_t')$ is sufficient to meet the demand $d_t$, i.e., $\sum_{p=1}^m U(m, p) d_{pt} \leq \mu_m(Q_t')$ for all $m = 1, \ldots, M$. Otherwise, we are unable to meet all demands. The following section outlines a policy we use to allocate insufficient capacity to product families.

3. Shortfall Allocation

This section explains how we determine the expected value $\xi$ 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. Given a set $Q$ of tools that are available at time $t$ (which is determined by $\tau$), tool type $m$’s capacity is given by $\mu_m(Q)$. Given that the capacity $\mu_m(Q) = \{\mu_m(Q) \mid m \in \mathcal{M}\}$ of all tool types and the realized demand $d_t = \{d_{pt} \mid p \in \mathcal{P}\}$ of all product families at time $t$, we determine both the production quantity $v_t = \{v_{pt} \mid p \in \mathcal{P}\}$ of product family $p$ and the allocation $x_t = \{x_{mpt} \mid m \in \mathcal{M}, p \in \mathcal{P}\}$ 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 Roudy et al. (2000), we 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. Thus, in any capacity allocation policy, production should not exceed demand, i.e., $v_{pt} \leq d_{pt}$ for all $p \in \mathcal{P}$. Production $v$ and allocation $x$ must obey the capacity limit of each tool type: $\sum_{p=1}^m x_{mpt} \leq \mu_m(Q)$ for all $m \in \mathcal{M}$ and $t \in [0, T]$, and $U(m, p) v_{pt} \leq x_{mpt}$ for all $p \in \mathcal{P}$ and $t \in [0, T]$.

We conceptually divide the demand into a deterministic portion $b_t \geq 0$ and a stochastic portion $\Delta_{b-t}$, $\phi_{b-t} \geq 0$. 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 $\tau$. Because $D_t \geq b_t$, our allocation policy meets the deterministic part $b_t$ of demand before allocating resources to the stochastic part.

We use an allocation policy that determines production quantities $v_t$, which 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 $d_t = b_t + \Delta_{b-t}$ is realized, this implies that we select production quantities $v_t = b_t + \xi \phi_{b-t}$ for some $\xi \in [0, \delta_{b-t}]$. Thus, $v_t$ also lies on the ray defined by the starting point $b_t$ and the direction $\phi_{b-t}$. The value $\xi$ 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 $(v_{pt} - b_{pt})/(d_{pt} - b_{pt}) = \xi/\delta_{b-t}$, which is independent of the product family $p$. If $b_t = 0$, then this corresponds to the classical fill rate.
4. Divide-and-Conquer Algorithm

We outline an efficient divide-and-conquer algorithm to minimize the total cost $\eta$. 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. 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 $\tau^*$ where $lb(C) \leq \tau_j^* \leq ub(C)$ for all $j \in C$ such that $\tau^*$ satisfies the first-order necessary condition of (P).

We note that if $lb(C) = ub(C)$, then we have found the desired purchase times $\tau_j^*$ 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 $\omega_C$ given $Q_L$ and $\mathcal{J}\setminus(Q_L \cup C)$, and let $S \subseteq C$ be its optimal solution, i.e., let $S$ minimize $\varphi_{\omega_C}(\cdot|Q_L, C, \mathcal{J}\setminus(Q_L \cup C))$, 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 $\bar{S} = C \setminus S$. 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 $C \in \mathcal{C}$.

In general, finding the minimizer of $\varphi_i$ may not be easy. Using explicit enumeration takes $O(2^{|Q_L|})$ computational time. Yet, under our modeling assumptions, we can minimize $\varphi_i$ efficiently by constructing a minimum-flow network of $O(\mathcal{J}|\mathcal{J}|)$ nodes and arcs. The network, similar to one found in Roundy et al. (2000), exploits the separability properties of the expected lost sales cost as well as the order in which tools become bottlenecked along each demand ray. This divide-and-conquer algorithm resembles the algorithm of Gusfield and Martel (1992), for the monotone parametric minimum-cut networks, and the algorithm of Hochbaum and Queyranne (2000), for the convex cost-closure problem.

Theorem 1. At each iteration of the algorithm, there exists some solution $\tau^*$ 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.

Under some assumptions, our continuous-time model becomes a minimization of a convex function and the above algorithm find the globally optimal solutions.

References

