

Optimal Capacity Expansion for Multi-Product, Multi-Machine Manufacturing Systems with Stochastic Demand

Robin Roundy, Feng Zhang,
Metin Çakanyıldırım, and Woonghee Tim Huh

School of Operations Research and Industrial Engineering
Cornell University, Ithaca NY, 14853

May 30, 2002

Abstract

We consider a discrete-time capacity expansion problem involving multiple product families, multiple machine types, and non-stationary stochastic demand. Capacity expansion decisions are made to strike an optimal balance between investment costs and lost sales costs. Motivated by current practices in the semiconductor and other high-tech industries, we assume that only minimal amounts of finished-goods inventories are held, due to the risk of obsolescence. We assume that when capacity is in short supply, management desires to ensure that a minimal service level for all product families is obtained. Our approach uses a novel assumption that demand can be approximated by a distribution whose support is a collection of rays emanating from a point and contained in real multi-dimensional space. These assumptions allow us to solve the problem as a max-flow, min-cut problem. Computational experiments show that large problems can be solved efficiently.

1 Introduction

The semiconductor industry has become one of the leading industries in US economy. A 1998 study commissioned by SIA (the Semiconductor Industry Association) [24] shows that chip sales increased 15 percent annually from 1987 to 1996, moving the industry from twentieth to fourth on the list of top manufacturing industries. One of the features of semiconductor manufacturing process is intensive capital investment. A modern fabrication facility (fab) being built today costs more than 1 billion dollars. More than 60% of the total cost is attributed to the cost of tools alone. In addition, in most existing fabs millions of dollars are spent on tool procurement each year to accommodate changes in technology. Therefore, making efficient usage of current tools and carefully planning the purchase of new tools are of great importance.

However, capacity planning decisions in the semiconductor industry are very challenging. Some of the main reasons for this are as follows.

- The consumer electronics market: In the consumer electronics business, product design cycles and life cycles are rapidly shrinking. Competition is fierce and the pace of product innovation is high. Consequently, the demand for new semiconductor products is becoming increasingly difficult to predict.
- Rapid changes in technology: In recent years the semiconductor industry has seen line width shrinkages approximately once every 18 months. Fabs dedicated to 300 millimeter wafers have been recently announced by most large semiconductor manufacturers. These and other technological advances require companies to continually replace many of their tools that are used to manufacture semiconductor products.
- Long procurement lead time: The lead time for procuring a new tool usually ranges from 3 months to a year. Many of the most expensive tools require 18 months. As a result, plans for expanding and upgrading capacity must be made based on demand forecasts reaching 2-3 years into the future. These demand forecasts are subject to a very high degree of uncertainty.
- High cost of tools: It is reported that, in 1996, the industry reinvested 23% of its total revenue in capacity expenses, mostly (60-70%) for tool

purchases. It is also projected that the cost of semiconductor manufacturing tools will be steadily increasing in the future. Thus, a small improvement in the tool purchase plan could lead to huge impact on the manufacturers.

In this paper, we will propose a discrete-time, finite-horizon analytical model for optimizing tool purchase plans under non-stationary stochastic demand. Capacity expansion decisions strike an optimal balance between lost sales costs and discounted tool purchase costs. Motivated by current practices in the semiconductor and other high-tech industries we assume that backorders are negligible¹ and that negligible amounts of finished-goods inventories are held due to the risk of obsolescence. This assumption leads to important simplifications. This assumption closely approximates the practices of many semiconductor companies, but not all of them. There are semiconductor companies that hold substantial semiconductor finished-goods inventory in order to smooth out production.

We aggregate products into the same product family if they consume identical or nearly proportional amounts of capacity on each of the key tool groups. For example, two logic products might have 4 and 5 layers of metal respectively. One wafer of the 4-layer product might consume nearly the same amount of capacity on key tool groups as 0.8 wafers of the 5-layer product. If so, for long range capacity planning purposes, they could be aggregated into a single product family by adjusting units of measure appropriately.

We use a novel demand model. If there are K different product families, then the demand $\mathbf{D}(t)$ is a random vector in K -dimensional real space, with a finite mean. In each time period t we assume that a collection of rays emanates from a point. These rays are a subset of the nonnegative orthant of K -dimensional real space. Most of our results depend on an assumption that the random demand vector $\mathbf{D}(t)$ has support on these rays. Computational results suggest that our demand model can effectively approximate a multi-dimensional demand vector with support on all of \mathfrak{R}_+^K .

¹When semiconductor companies lack capacity they often postpone delivery dates to their clients. This is formally treated like a backorder. However deferred delivery dates frequently prompt the client to defer subsequent orders, so the end result can be closer to lost sales than to backorders.

Whereas capacity decisions are made in period 1, production decisions are made at each period adaptively. One consequence of the no-back-order, no-inventory assumption is that lost sales can be computed by comparing production in a period to the demand in the same period. We assume that when capacity is in short supply, management desires to ensure a minimal service level for all product families. Specifically, we assume that if $\mathbf{D}(t)$ lies on one of the given rays as described above, and if there is not enough capacity to meet the demand in period t , then the point on the ray that is capacity-feasible and as close to $\mathbf{D}(t)$ as possible will correspond to production in period t . If the rays emanate from the origin, this strategy equalize the fill rate of the different product families. If they emanate from a different point, this policy equalize a service measure that is calibrated differently for different product families².

In a classical approach to stochastic programming models, a continuous distribution is approximated by a finite collection of points sampled from the distribution. If we had taken this approach, it would be easy to formulate our problem as a stochastic integer linear programming problem. However, a large number of points would be necessary to model the demand adequately. Our approach is different. By using distributions that are supported on rays instead of points, the number of rays needed to model a continuous distribution may be smaller. We prove that our formulation of this problem may be transformed to a max-flow network problem, which can be efficiently solved by available software.

The rest of this paper is organized as follows. In section 2, related literature is reviewed. Section 3 presents our demand model, which is used to formulate our problem in section 4. In section 5 we provide details of the numerical tests, and we conclude in section 6.

2 Literature Review

In this section we present a brief discussion of stochastic capacity expansion problems, especially single-location models. An extensive survey of early

²See Figure 2 below

work, capturing application areas and multi-location models as well as single-location models, is provided in Luss [19]. A discussion of the models that appeared after [19] is given in [11].

A general approach to discrete-time capacity planning models under uncertainty is stochastic programming (Wets [25], Birge and Louveaux [7]). Generally demand uncertainty is represented in terms of demand scenarios. Eppen, Martin and Schrage [15], Escudero, Kamesam, King and Wets [16], and Chen, Li and Tirupati [8] use scenarios in studying capacity planning for manufacturing industries. However, in order to make the scenario-based model solvable, only a limited number of scenarios can be used. Consequently it is usually impossible to accurately model multi-dimensional demand processes.

Karabuk and Wu [18] treat both demand and capacity as uncertain. A multi-stage stochastic program is formulated. Demand and capacity uncertainty is incorporated through a scenario structure. Then the problem is decomposed to reflect a decentralized decision-making process. For the service industries Berman, Ganz and Wagner [4] provide a stochastic programming model to expand capacities at multiple loosely coupled locations.

Another approach for discrete-time models is Markov Decision Processes (MDP). Using MDPs, Bhatnagar, Fernandez-Gaucherand, Fu, He and Marcus [6] lay out a capacity expansion model capturing machine disposals and inventory holding. Like stochastic programming, Markov decision process is a general purpose tool. It can provide great flexibility in modeling; however, such flexibility makes it extremely difficult (or impossible) to find solutions to problems of realistic size and complexity.

If time is a continuous variable rather than a discrete one, the capacity expansion problem may be formulated as an optimal control problem. For a single product with a stochastic demand process, Davis, Dempster, Sethi and Vermes [12] regulate the expansion rate with an investment rate function which is the control variable. As soon as the cumulative investment reaches the random price of a discrete capacity unit, that unit becomes available. For multiple products with deterministic demand rates, Sethi, Taksar and Zhang [23] study machine capacity expansion when machines can break down randomly. They propose to plan capacities using the mean available capacity

numbers. When breakdowns and repairs happen sufficiently fast, they show that the cost of their expansion plan asymptotically converges to the optimal cost.

In the economics community, the capacity expansion problem is recently addressed by works of Dixit [13], and Eberly and Van Mieghem [14]. They establish structural properties for expansion and contraction policies for multiple factors contributing to capacity. The latter paper provides a closed form solution for the optimal policy in the case of IID stochastic processes and stationary costs. For certain capacity expansion cases, Rocklin, Kashper and Varvaloucas [21] establish the optimality of (s, S) policies.

There have been efforts to convert stochastic expansion problems to equivalent deterministic problems. One of them is Bean, Hagle and Smith [3], which provides such a conversion by modifying the interest rate when demand is either a Brownian motion or a transformed birth-death process.

Over long horizons, with the introduction of new technologies, new types of machines become available for purchase. Rajagopalan, Singh and Morton [20] study the replacement of old machines with new ones, under both certain and uncertain technology arrival times, and with deterministic, nondecreasing demand. They show some structural properties of the optimal solution and exploit those with a dynamic program.

We now turn to the capacity expansion studies specifically targeting the semiconductor industry. Although it is a deterministic rather than a stochastic capacity planning model, we mention the capacity optimization planning system (CAPS) (Bermon and Hood [5]) because it does an excellent job of capturing the capacity relationships between products and tools. Bard, Srinivasan and Tirupati [2] formulate the capacity expansion problem as a nonlinear integer mathematical program that finds a tool-set configuration to minimize average cycle time, within a given budget. Their model is based on the assumption of a small number of products, and it uses queuing network approximations. Chou and Everton [9] use simulation to study tool capacity and operator availability at development wafer fabs. Angelus, Porteus and Wood [1] consider capacity expansion with fixed costs, and stochastically increasing and correlated demand. They show that the optimal expansion policy is (s, S) type where both parameters depend on the most recently ob-

served demand. Jayashankar [17] captures demand uncertainty with a two stage stochastic mixed integer linear program. Since this problem becomes very large even with a few scenarios, he provides Lagrangian based bounds and heuristics.

Treating tool purchase times as continuous decision variables, Çakanyıldırım, Roundy and Wood [11] provide a nonlinear program to minimize the expected lost sales costs and capacity costs, for a fab producing a single product but experiencing uncertain demand. In the single product case, the analysis is simplified after observing that the optimal sequence of machine purchases is independent of demands. However, this is not true for the multi-product case because the allocation of capacity to products is not straightforward.

The current paper extends [11] to the multi-product discrete-time case by proposing a specific capacity allocation scheme. It presents a practical solution approach for a problem similar to the one studied in [17]. Unlike the other models described above, our model is capable of dealing with problem instances of realistic scale, i.e., large numbers of time periods, products and tools. It captures the uncertainty in non-stationary demand using an innovative model. When combined with our capacity allocation scheme, this demand model is not only tractable but much more compelling than using a small number of demand scenarios in order to represent a multi-dimensional random vector.

3 Notation and Demand Model

We use the following notation³.

t : Time index; $t \in \{1, \dots, T\}$ where T is the time horizon we need to consider in the model.

m : Tool type index; $m \in \{1, \dots, M\}$.

k : Product family index; $k \in \{1, \dots, K\}$.

$\lambda(m)$: Purchase lead time for a type m tool.

³In the rest of this paper, uppercase letters are used to represent random variables. The corresponding lowercase letter is used to represent a realization of a random variable. Thus $\mathbf{d}(t)$ is a specific realization of $\mathbf{D}(t)$. Vectors and matrices are in boldface.

$\mathbf{D}(t)$: Random demand vector in period t , $\mathbf{D}(t) \in \mathfrak{R}^K$.

\mathbf{u} : Utilization matrix; the (m, k) 'th component represents the units of capacity-time required at a type m tool if one unit of product family k is produced.

$\kappa(t)$: Capacity at time period t ; $\kappa(t) \in \mathfrak{R}^M$. Also, $\kappa_m(t)$ is the capacity on tool type m in period t .

$\mathbf{P}(t)$: Production plan in period t ; $\mathbf{P}(t) \in \mathfrak{R}^K$. Specifically $\mathbf{P}_k(t)$ is the production of product family k in period t .

To evaluate the average lost sale cost, we need to know the distribution of the random demand vector $\mathbf{D}(t) \in \mathfrak{R}^K$. We use the following novel approach to model demand.

Assumption 1 (*Ray-based demand model*) *The random demand $\mathbf{D}(t)$ can be expressed as*

$$\mathbf{D}(t) = \mathbf{b}(t) + \Delta_{I(t)}(t)\phi_{I(t)}(t)$$

where

$\mathbf{b}(t)$ is a deterministic vector in \mathfrak{R}_+^K ,

$I(t)$ is a discrete random variable with finite support such that $p_i(t)$ is the probability that $I(t) = i$,

$\phi_i(t)$ is a deterministic unit-norm vector in \mathfrak{R}_+^K , and

$\Delta_i(t)$ is a continuous non-negative random scalar, independent of $I(t)$.

In each time period t , a finite number of vectors $\phi_i(t)$ define a collection of rays that emanate from $\mathbf{b}(t)$ and are contained in \mathfrak{R}_+^K . The support of $\mathbf{D}(t)$ is the union of these rays. (See Figure 1.) The random variable $I(t)$ determines the ray on which $\mathbf{D}(t)$ lies. After the ray has been selected (i.e., if we condition on $I(t) = i$), the continuous random variable $\Delta_i(t)$ determines how far one moves from $\mathbf{b}(t)$ in the direction $\phi_i(t)$ before reaching the demand $\mathbf{D}(t)$. Thus $\Delta_{I(t)}(t) = |\mathbf{D}(t) - \mathbf{b}(t)|$. If we set $\mathbf{b}(t) = \mathbf{0}$, then the product mix in the demand vector is determined by the random variable $I(t)$, because $\mathbf{D}_k(t)/\mathbf{D}_1(t) = \phi_{ki}(t)/\phi_{1i}(t)$ for all product families k . With $\mathbf{b}(t) = \mathbf{0}$, the magnitude of the demand vector is $|\mathbf{D}(t)| = \Delta_{I(t)}(t)$. The magnitude is

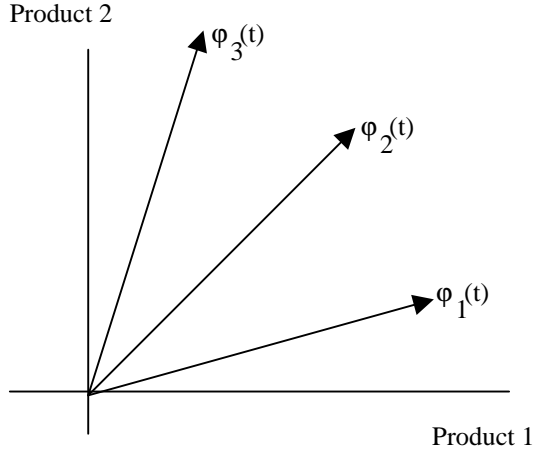


Figure 1: The support of $\mathbf{D}(t)$ with $\mathbf{b}(t) = \mathbf{0}$.

correlated with the product mix because the distribution of $\Delta_i(t)$ depends on i .

In the remainder of this section, we discuss one motivation for introducing the ray-based demand model (Assumption 1). We also describe a practical approach for obtaining parameters for the ray-based demand model. Consider a demand vector \mathbf{D}^* that has a continuous density with support in \mathfrak{R}_+^K . Suppose we want to compute an approximation to $E[f(\mathbf{D}^*)]$ for some function f . (In the capacity planning problem presented in this paper, the function f corresponds to the lost sales in period t .) One common approach is to approximate $E[f(\mathbf{D}^*)]$ with $\frac{1}{|S|} \sum_{s \in S} f(s)$, where S is a set of points sampled from \mathbf{D}^* . The variance of the estimate is $\frac{1}{|S|} \text{var } f(\mathbf{D}^*)$. This classical approximation is often embedded in optimization algorithms, usually formally by assuming that $\mathbf{D}(t)$ has support on S .

Our ray-based demand model can be derived from improved estimate of $E[f(\mathbf{D}^*)]$, based on a variance-reduction technique called conditioning. Define $\Phi(t) = \mathbf{D}^*/|\mathbf{D}^*|$, and define Δ_{Φ}^* by $\mathbf{D} = \Delta_{\Phi}^* \cdot \Phi$. The conditioning approach is viable because we can analytically evaluate $E_{\Delta_{\Phi}^*} [f(\Delta_{\Phi}^* \cdot \phi)]$ for a given unit vector ϕ (see section 4). We randomly sample points $s \in S$ from

\mathbf{D}^* , and we compute $\phi_s = \frac{1}{|S|}s$. The ray-based estimate of $E[f(\mathbf{D}^*)]$ is

$$\frac{1}{|S|} \sum_{s \in S} E_{\Delta^*_{\phi_s}} [f(\Delta^*_{\phi_s} \cdot \phi_s)]. \quad (1)$$

This is an unbiased estimator of $E[f(\mathbf{D}^*)]$. The variance of this estimator is $\frac{1}{|S|} \text{var}_{\Phi(t)} [E_{\Delta^*_{\Phi}} [f(\Delta^*_{\Phi} \cdot \Phi)]]$. The standard conditioning inequality states that (see Law and Kelton[26]⁴)

$$\begin{aligned} & \text{var}_{\Phi(t)} [E_{\Delta^*_{\Phi}} [f(\Delta^*_{\Phi} \cdot \Phi)]] \\ &= \text{var}[f(\mathbf{D}^*(t))] - E_{\Phi(t)} [\text{var}_{\Delta^*_{\Phi}} [f(\Delta^*_{\Phi} \cdot \Phi)]] \\ &\leq \text{var}[f(\mathbf{D}^*(t))]. \end{aligned}$$

Thus, the ray-based estimate (1) of $E[f(\mathbf{D}^*)]$ has lower variance than the more traditional estimate $\frac{1}{|S|} \sum_{s \in S} f(s)$.

To see the relationship between the ray-based estimate of $E[f(\mathbf{D}^*)]$ and our ray-based demand model, assume that we obtain the parameters for $\mathbf{D}(t)$ from \mathbf{D}^* in the following manner.

1. $\mathbf{b}(t) = \mathbf{0}$,
2. $P(I(t) = i) = p_i = \frac{1}{|S|}$,
3. $\{\phi_s = \frac{s}{|S|} \mid s \in S\} = \{\phi_i(t) \mid 1 \leq i \leq |S|\}$, and
4. The distribution of $\Delta_i(t)$ is equal to that of $\Delta^*_{\phi_i}$ (see Appendix C).

Then

$$\begin{aligned} Ef(\mathbf{D}(t)) &= E_{I(t)} \{E[f(\mathbf{D}(t)) \mid I(t) = i]\} \\ &= \frac{1}{|S|} \sum_i E_{\Delta_i(t)} [f(\Delta_i(t)\phi_i(t))], \end{aligned}$$

which is the ray-based estimate of $E[f(\mathbf{D}^*(t))]$. Thus the ray-based demand model can arise from the variance reduction technique called conditioning. In this paper, we formally assume that the demand is ray-based; i.e., that Assumption 1 holds.

⁴The authors acknowledge Shane Henderson for pointing out this reference.

In section 5, we will generate parameters for the ray-based demand model. We will assume that $\mathbf{D}^*(t)$ is a multi-variate lognormal random variable with known parameters. We will set $\mathbf{b}(t) = \mathbf{0}$. We will obtain $\phi_i(t)$'s by drawing an appropriate number of random samples from $\mathbf{D}^*(t)$ and normalizing them. We will set $\Delta_i(t)$ to the conditional distribution of $\mathbf{D}^*(t)$ given that the demand lies on the ray corresponding to $\phi_i(t)$ (see Appendix C). We could set $p_i(t) = \frac{1}{r}$ where $r = |\text{supp}(I(t))|$ is the number of rays. However the recommended procedure is to adjust $p_i(t)$'s so that the mean vector of the ray-based demand $\mathbf{D}(t)$ coincides with the mean vector $\boldsymbol{\mu}(t) = E[\mathbf{D}^*(t)]$. This can be done, for example, by solving the following quadratic program:

$$\begin{aligned} \text{Minimize} \quad & \sum_i \left(\frac{1}{r} - p_i(t)\right)^2 \\ \text{such that} \quad & \boldsymbol{\mu}(t) = \mathbf{b}(t) + \sum_i p_i(t) \mu_i(t) \phi_{I(t)}(t) \end{aligned}$$

where $\mu_i(t)$ is the expected value of $\Delta_i(t)$.

4 Optimization Model

In this section we provide an integer programming formulation of the capacity expansion problem. We consider the problem in multiple discrete time periods. Our objective is to find the optimal tool purchase plan, so as to minimize the sum of lost sale costs and tool purchase costs.

Assumptions

We assume that all tool purchase decisions are made at the beginning of time period 1. Thus, the capacity $\kappa(t)$ is based on the distribution of the demand $\mathbf{D}(t)$, which is available at the time of planning, but not on the realization $\mathbf{d}(t)$ of $\mathbf{D}(t)$. In practice a dynamic rolling horizon approach would be utilized; thus new plans would be generated periodically, and only the earliest portion of the computed plans would be implemented. We note that both the capacity $\kappa(t)$ and the production plan $\mathbf{P}(t)$ are intermediate variables that depend on the tool purchase decisions. The production plan $\mathbf{P}(t)$ in period t also depends on the realization $\mathbf{d}(t)$ of demand $\mathbf{D}(t)$ in period t , as shown in the following assumptions.

Assumption 2 *In each time period t , the following events are assumed to happen in the given order*

1. *Observe the state.*
2. *Newly-installed tools become available for production.*
3. *Demand is observed.*
4. *Production decisions are made, and production occurs.*
5. *Costs are incurred.*

We have assumed that both inventories and backorders are negligible. This assumption has an important consequence, namely, that after tool purchase decisions have been made the problem decomposes by time periods. Thus, after $\boldsymbol{\kappa}(t)$ has been selected for all t , the production $\mathbf{P}(t)$ in period t can be selected independently of $\mathbf{P}(t - 1)$, $\mathbf{P}(t + 1)$, etc.

According to Assumption 2, the production plan $\mathbf{P}(t)$ is made after the demand $\mathbf{D}(t)$ has been observed. There are many ways to choose $\mathbf{P}(t)$. In our model, we use a service-based policy called proportional production which, as we described in the introduction, achieves an equal fill rate for all product families if $\mathbf{b}(t) = \mathbf{0}$. Proportional production is illustrated in Figure 2. If demand $\mathbf{d}(\mathbf{t})$ falls in the shaded region, then production is equal to demand. If the realized demand is outside of the shaded region, $\mathbf{p}(t)$ will be the production plan and there is a shortfall. Mathematically we can define our proportional production plan as follows.

Assumption 3 (*Proportional Production*) *We apply proportional production in each time period, i.e.*

$$\mathbf{P}(t) = \mathbf{b}(t) + \min [\sigma_{I(t)}^*(t), \Delta_{I(t)}(t)] \boldsymbol{\phi}_{I(t)}(t),$$

where

$$\sigma_i^*(t) = \max\{\sigma : \mathbf{u}[\mathbf{b}(t) + \sigma \boldsymbol{\phi}_i(t)] \leq \boldsymbol{\kappa}(t)\}.$$

Consequently, we observe that

1. If $\mathbf{b}(t) = \mathbf{0}$, the ratio of unsatisfied demand to total demand is the same across all products. Thus the fill rates among the different products are equalized. If $\mathbf{b}(t) \neq \mathbf{0}$, then the fill rates are different from product to product.

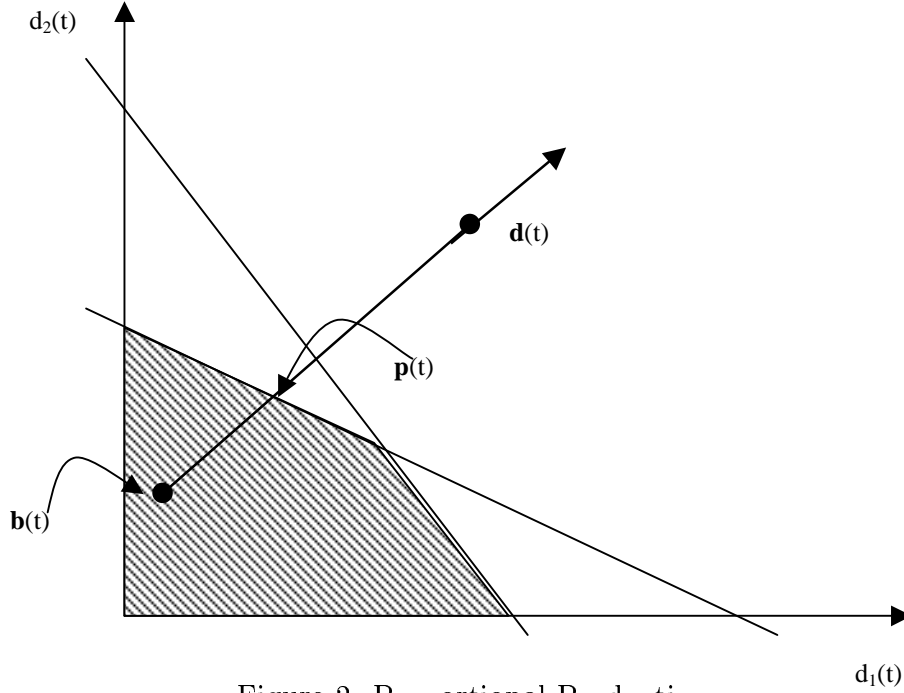


Figure 2: Proportional Production

2. The production plan $\mathbf{P}(t)$ always lies on the same ray as the demand $\mathbf{D}(t)$.
3. The maximum production plan coefficient $\sigma_{I(t)}^*(t)$ is determined by the demand direction $\phi_{I(t)}(t)$ and the current capacity $\kappa(t)$.

We point out that there are other plausible alternatives to proportional production. For example, we could select the production plan that maximizes the revenue obtained.

Lost Sales Costs

The demand distribution (Assumption 1) and proportional production (Assumption 3) allow us to derive a simple expression for the lost sales. Given a demand $\mathbf{d}(t)$ which is a realization of the random vector $\mathbf{D}(t)$, the production plan $\mathbf{p}(t)$ is fixed. Assume that the production plan $\mathbf{p}(t)$ is applied and that the lost sales cost is linear in the quantity of unsatisfied demand⁵. Then the lost sales cost is $\mathbf{c} \cdot [\mathbf{d}(t) - \mathbf{p}(t)]^+$. The k 'th component of vector \mathbf{c} is the unit

⁵Note that we do not produce more than the demand, so there are no inventory costs. As we mentioned earlier, substantial inventories of finished goods constitute a risk that many companies prefer to avoid.

lost sales cost for the k 'th product. Consequently the expected lost sales cost in period t will be

$$E_{\mathbf{D}(t)}\{\mathbf{c} \cdot [\mathbf{D}(t) - \mathbf{P}(t)]^+\}.$$

Let $n_{\Delta_i(t)}(\cdot)$ be the partial expectation, defined by

$$n_{\Delta_i(t)}(\sigma) = E_{\Delta_i(t)}(\Delta_i(t) - \sigma)^+.$$

Then the expected lost sales cost in t can be rewritten as ⁶

$$\sum_i p_i(t) \cdot n_{\Delta_i(t)}(\sigma_i^*(t)) \mathbf{c} \cdot \boldsymbol{\phi}_i(t). \quad (2)$$

Variables and Constraints

Having defined a convenient expression for the lost sales, we now formulate the variables and constraints of our integer programming model. We also relate the variables to our ability to meet the demand. We use (m, n) to index tool purchases, where (m, n) represents the n 'th purchase of a type m tool. We also write $j \sim (m, n)$ where j ($1 \leq j \leq J$) indexes purchases of all tools of any type. Define a binary variable $x(j, t)$ as follows:

$$x(j, t) = \begin{cases} 1 & \text{if tool } j \text{ is available for production by time } t \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Obviously $x(j, t)$ is non-decreasing with respect to time, i.e.

$$x(j, t) \leq x(j, t + 1) \quad \forall j, t \quad (4)$$

We refer to (4) as *time monotonicity* in the rest of this paper. Since $\lambda(m)$ is the purchase lead time for a type m tool, if $t \leq \lambda(m)$ then the value of $x(j, t)$ is fixed because the purchase decision has already been made.

The most intricate part of the formulation is relating the lost sales cost (2) to the variables $x(j, t)$. Intuitively, the variable $x(j, t)$ determines whether or not the purchase of tool j has been made by time period t . However the purchase of tool j may not increase the actual capacity at all. This can happen if the capacity of a type m tool is currently the tight constraint, and the purchase of tool j increases the capacity of a type $m' (\neq m)$ tool. In

⁶See Appendix A for a proof.

order to understand the actual capacity, we denote the demand ray that has direction $\phi_i(t)$ as demand ray i . For each time period t and each demand ray i , we need to understand which tool is currently limiting the capacity. We let $\sigma(i, j, t)$ be the distance along ray i from $\mathbf{b}(t)$ to the point where tool j limits capacity. For $j \sim (m, n)$, we have

$$\sigma(i, j, t) = \max\{\sigma : \mathbf{u}_{m,:}[\mathbf{b}(t) + \sigma\phi_i(t)] \leq \kappa(j)\},$$

where $\mathbf{u}_{m,:}$ is the m 'th row of utilization matrix \mathbf{u} and $\kappa(j)$ is the capacity of the type m tool group just before $j \sim (m, n)$ is available. Thus, when tool j becomes available for production, the capacity of the type m tool group jumps from $\kappa(j)$ to $\kappa(j')$, where $j' \sim (m, n + 1)$.

Suppose, at time period t , that the realized demand falls on ray i and is

$$\mathbf{d}(t) = \mathbf{b}(t) + \delta_i(t)\phi_i(t). \quad (5)$$

Lemma 1 *Let (5) hold and let $\sigma(i, j, t) < \delta_i(t) \leq \sigma(i, j', t)$ where $j \sim (m, n)$ and $j' \sim (m, n + 1)$. Then the capacity of the type m tool group is adequate to meet the demand $\mathbf{d}(t)$ if and only if $x(j, t) = 1$.*

Proof: A direct consequence of the definitions of $\sigma(i, j, t)$ and $x(j, t)$. QED

Along demand ray i , the actual capacity is determined by the sequence in which the values $\{\sigma(i, j, t) : 1 \leq j \leq J\}$ fall. We sort $\{\sigma(i, j, t) : 1 \leq j \leq J\}$ in increasing order. We denote by $\gamma_t^i(j)$ the *immediate successor* of tool j along ray i ; i.e. we write

$$\gamma_t^i(j) = j'$$

if (i, j', t) comes immediately after (i, j, t) in the sorted list. We also define the *precedence set* $\pi_t^i(j)$ of tool j along demand ray i as follows:

$$\pi_t^i(j) = \{j' : j' = j \text{ or } (i, j', t) \text{ comes before } (i, j, t) \text{ in the sorted list}\}.$$

Figure 3 illustrates the definition of γ and π using an example with 2 product families, 2 types of tools and 3 demand rays. The definition of π implies that $\sigma(i, j', t) \leq \sigma(i, j, t)$ if and only if $j' \in \pi_t^i(j)$.

If (5) holds, then in order to meet the demand, every type of tool must have adequate capacity. Assume that

$$\sigma(i, j, t) < \delta_i(t) < \sigma(i, \gamma_t^i(j), t). \quad (6)$$

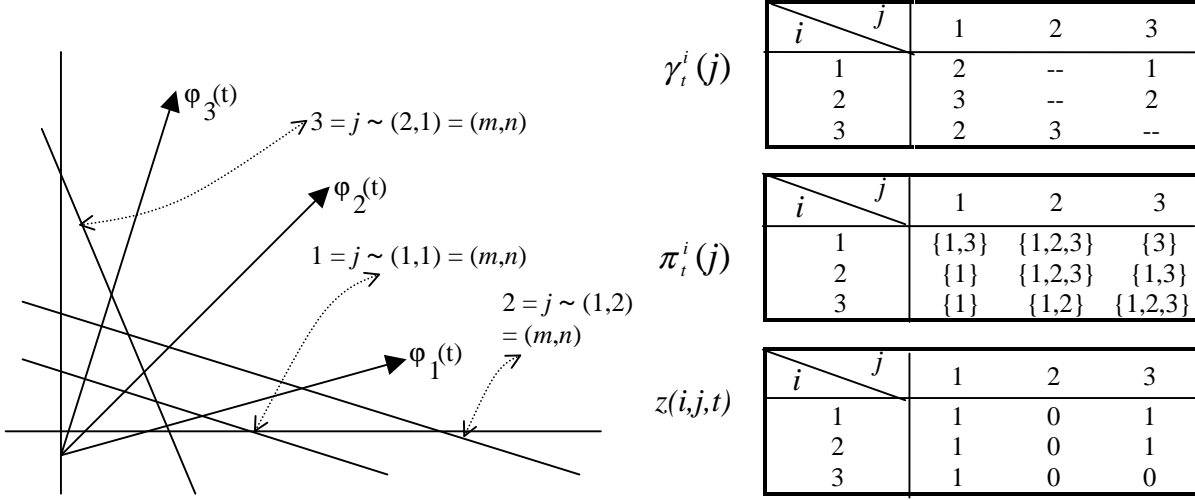


Figure 3: $\gamma_t^i(j)$, $\pi_t^i(j)$ and $z(i, j, t)$ when $x(1, t) = x(3, t) = 1$ and $x(2, t) = 0$

By Lemma 1, to meet the demand we must have

$$x(j', t) = 1 \quad \forall j' \text{ s.t. } \sigma(i, j', t) < \delta_i(t), \text{ i.e., } \forall j' \in \pi_t^i(j).$$

To capture this, we define another group of binary indicator variables $z(i, j, t)$ by

$$z(i, j, t) = \begin{cases} 1 & \text{if } x(j', t) = 1 \quad \forall j' \in \pi_j^i(j), \\ 0 & \text{otherwise.} \end{cases}$$

Thus if (5) and (6) hold, we can meet demand if and only if $z(i, j, t) = 1$. Figure 3 illustrates $z(i, j, t)$ for the case when the first and third purchases have been made, i.e., $x(1, t) = x(3, t) = 1$ and $x(2, t) = 0$. The decision variables $z(i, j, t)$ satisfy the following two properties: (1) along demand ray i , the precedence set of j is contained in the precedence set of $\gamma_t^i(j)$, so

$$z(i, j, t) \geq z(i, \gamma_t^i(j), t). \quad (7)$$

(2) By definition of the precedence set, $j \in \pi_t^i(j)$. Therefore

$$z(i, j, t) \leq x(j, t). \quad (8)$$

Subsequently we refer to (7) as *ray monotonicity* and to (8) as *consistency*. Note that $\sigma(i, j, t) \leq \sigma(i, j', t)$ implies $z(i, j, t) \geq z(i, j', t)$.

Lemma 2 *Let $\mathbf{x} = (x(j, t), \forall j, t)$ and $\mathbf{z} = (z(i, j, t), \forall i, j, t)$. Assume that the demand in period t satisfies (5) where $\sigma(i, j', t) < \delta_i(t) \leq \sigma(i, \gamma_t^i(j'), t)$.*

- (i) *Suppose that the vectors \mathbf{x} and \mathbf{z} satisfy (4), (7) and (8). If $z(i, j', t) = 1$, then there is adequate capacity to meet the demand in period t .*
- (ii) *Conversely, if there is adequate capacity to meet the demand in period t , then there exist \mathbf{x} and \mathbf{z} satisfying (4), (7) and (8), with $z(i, j', t) = 1$.*

Proof: (i) follows directly from (3), (7), (8), and Lemma 1. To prove (ii), assume that there is adequate capacity and that x is given by (3). Note that (3) implies (4). From (5), Lemma 1 implies that $x(j, t) = 1$ if $\sigma(i, j, t) < \delta_i(t)$. Set $z(i, j, t) = 1$ if $\sigma(i, j, t) < \delta_i(t)$ and $z(i, j, t) = 0$ otherwise. It's easy to verify that \mathbf{z} satisfies (7) and (8). QED

The Objective Function

We now relate the expected lost sales cost to the variables \mathbf{z} . Suppose that the demand $\mathbf{D}(t)$ lies on demand ray i (i.e. we condition on $I(t) = i$). Furthermore, suppose that in period t , all the tools in the precedence set of tool j' along demand ray i are available but the immediate successor j_o of j' is not. In other words,

$$z(i, j', t) = 1 \text{ and } z(i, j_o, t) = 0 \text{ where } j_o = \gamma_t^i(j')$$

If we make the purchase of tool j_o , we increase $x(j_o, t)$ and $z(i, j_o, t)$ from 0 to 1, and then by (2) and Lemma 2, the expected lost sales cost will decrease by

$$\nu(i, j, t) = [n_{\Delta_i(t)}(\sigma(i, j', t)) - n_{\Delta_i(t)}(i, \sigma(\gamma_t^i(j')), t)] \mathbf{c} \cdot \boldsymbol{\phi}_i(t)$$

Therefore the lost sales cost incurred in period t , given that the demand ray is $i = I(t)$, is

$$\sum_{\{j: z(i, j, t)=0\}} \nu(i, j, t) = \sum_j \nu(i, j, t) - \sum_j \nu(i, j, t) z(i, j, t). \quad (9)$$

We now consider tool purchases costs. Define $\mu(j, t)$ to be the decrease in tool purchase cost by deferring the purchase of tool j from time period t to $t + 1$, including the effects of discounting. Let $\mu(j, T)$ include the fixed cost of purchasing tool j . Under (3), the total cost of the purchase of tool j is $\sum_t \mu(j, t)x(j, t)$.

Formulation

Our objective function trades off expected lost sales costs against tool purchase costs. The constraints are time monotonicity (4), ray monotonicity (7) and consistency (8). Mathematically, using (9) to formulate our objective, we have the following ILP model:

$$\begin{aligned}
 (ILP1) \quad \min \quad & \sum_t \left\{ \sum_j [\mu(j, t)x(j, t)] - \sum_{i,j} [p_i(t)\nu(i, j, t)z(i, j, t)] \right\} \\
 \text{such that} \quad & x(j, t) \leq x(j, t + 1) \quad \forall j, t & (4) \\
 & z(i, j, t) \leq x(j, t) \quad \forall i, j, t & (8) \\
 & z(i, \gamma_t^i(j), t) \leq z(i, j, t) \quad \forall i, j, t & (7) \\
 & x(j, t), z(i, j, t) \quad \text{binary} \quad \forall i, j, t & (10)
 \end{aligned}$$

Clearly (ILP1) is equivalent to a min-cut problem, which is a dual to a max-flow problem in the capacity expansion network (see Figure 4⁷). This network has four levels of nodes in the network. The first level contains only one node: *SOURCE*. The fourth level also contains only one node: *SINK*. Every node in the second level corresponds to a variable $z(i, j, t)$, and is called a *z-node*. Every node in the third level corresponds to a variable $x(j, t)$, and is called an *x-node*. We use the variable names as node labels.

The network is constructed to capture the objective function and the constraints in (ILP1). More precisely, the flow on the arc leading from *SOURCE* to a node $z(i, j, t)$ in the second level is constrained to lie between 0 and $p_i(t)\nu(i, j, t)$, the corresponding lost sales cost. These arcs are referred as

⁷For simplicity, $T = 3$, the rays are i and i' , and the tool purchases are j and j' . Also assume no purchase lead time; i.e., $\lambda(j) = \lambda(j') = 0$. Along those rays and in all time periods, j and j' fall in the same sequence. Thus $\gamma_t^i(j') = (i, j, t)$ and $\gamma_t^{i'}(j) = (i', j, t) \forall t$. $\{(i, j', 1), (i', j, 1), (i', j', 1)\}$.

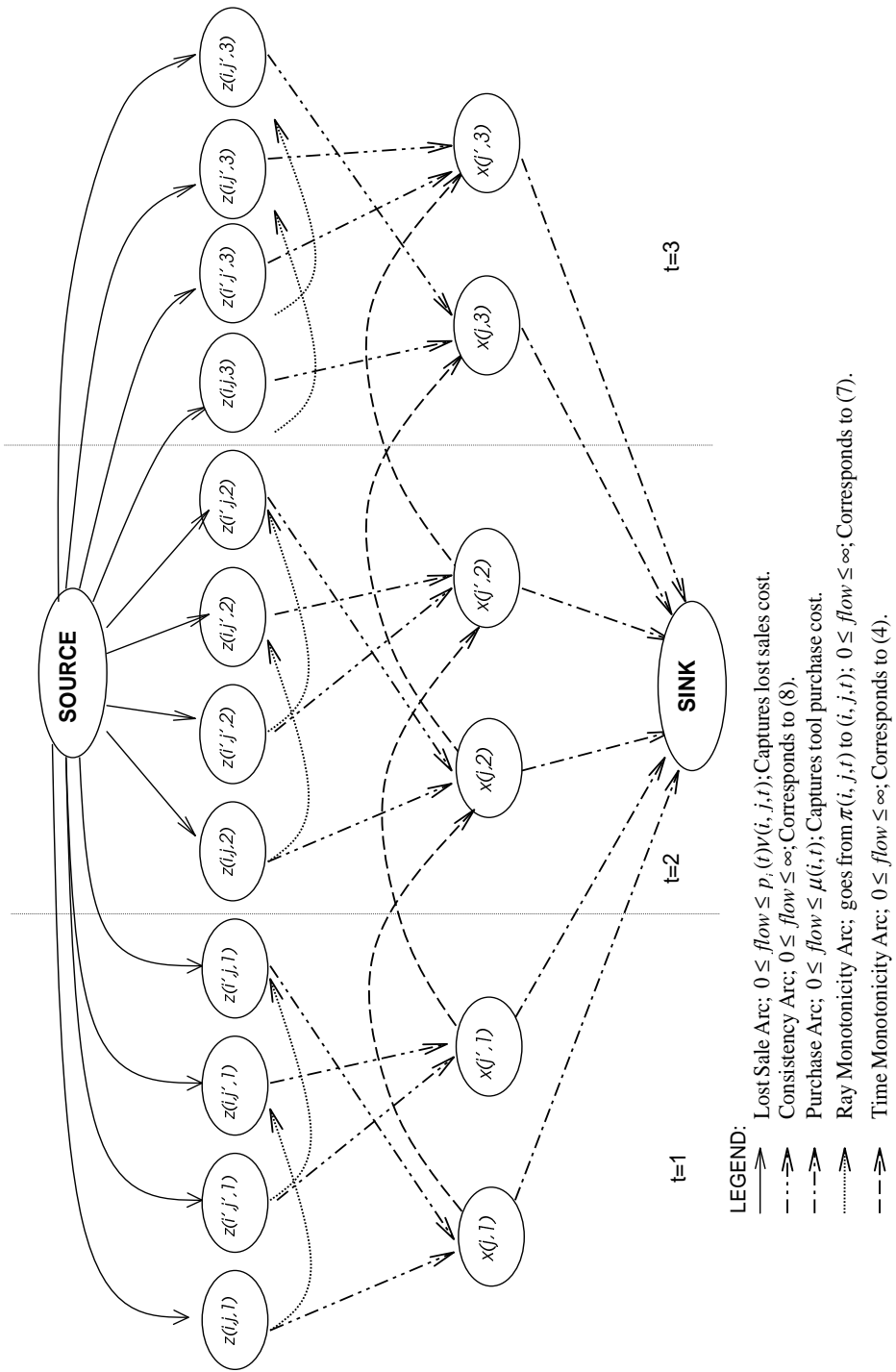


Figure 4: Capacity Expansion Network

Lost Sales Arcs. Similarly, the arc from a node $x(j, t)$ in the third level to SINK is a *Purchase Arc*. The flow on the arc must be between 0 and $\mu(j, t)$, the purchase cost. In addition, every constraint of the form $u \leq v$ in (ILP1) corresponds to an arc leading from node u to node v . Thus, for example, a *Consistency Arc* leads from $z(i, j, t)$ to $x(j, t)$, a *Time Monotonicity Arc* leads from $x(j, t)$ to $x(j, t + 1)$, and a *Ray Monotonicity Arc* leads from $z(\gamma_t^i(j))$ to $z(i, j, t)$. On all of these arcs, the flow is constrained to lie between 0 and ∞ . The infinite-capacity arcs ensure that every finite capacity cut in the network corresponds to a feasible tool purchase plan. We have the following result.

Theorem 1 *(ILP1) is equivalent to a minimum-cut problem.*

We note that some variables in the formulation of (ILP1) are fixed because of tool purchase lead times. This results in the deletion of some nodes and arcs in the Capacity Expansion Network. A full discussion and a proof of Theorem 1 are given in the appendix.

5 Numerical Testing

The primary goal of our numerical experiments is to test the performance of our algorithm for capacity planning problems of practical size and complexity. The approach we have developed to solve the optimal capacity planning problem is based on Assumption 1. If the demand $\mathbf{D}(t)$ is a continuous random variable, then in accordance with Assumption 1, we can approximate $\mathbf{D}(t)$ with a distribution which has support on a finite number of demand rays as discussed in the previous section. As the number of demand rays increases, we obtain a more accurate approximation to the true demand. However the size of the Capacity Expansion Network also increases, requiring more computation. We want to determine how many demand rays are required to approximate a continuous demand distribution effectively.

We combined data from two industrial data sources. We obtained a tool data file from the SEMATECH (SEmiconductor MANufacturing TECHNOlogy) database. Tool data includes purchase prices and capacities for each tool type. The forecast and demand data files come directly from a semiconductor manufacturer. We used SeDFAM [22] to analyze this forecast data

and to generate the variance and covariance of future demands.

Based on the properties of the industrial data, we assume that the demand $\mathbf{D}(t)$ has a multi-variate log-normal distribution. For each t , we use the distribution of $\mathbf{D}(t)$ to randomly generate a pre-determined number of future demands. After being normalized, these become our demand rays. Given any demand ray, the distribution and partial expectation of $\Delta_i(t)$ are easy to derive (see Appendix C).

We use two steps to compute an optimal expansion plan. A Capacity Expansion Network is constructed in the first step. More specifically, tool data and demand data are read in, and an intermediate network data file is generated using MATLAB. The network file is of DIMACS format [10] and describes the Capacity Expansion Network. We refer to the first step as NG (network generation) hereafter. In the second step we use HIPR, software developed by B. Cherkassky and A. Goldberg, to solve the maximum flow/minimum cut problem [10]. HIPR is an efficient implementation of the push-relabel method, and is designed to run on UNIX/LINUX. It reads the network information from the DIMACS format data file and writes the results to standard output. The second step is called HIPR in the rest of paper.

In our simulation we assume that generation of the purchase plan is done quarterly, and we use a 4-year planning horizon. Demand approximately doubles over the 4-year horizon. The lost sale costs vary across the product families. The average is about \$10,000 per wafer and the deviation is $\pm 10\%$. Purchase prices for tools range from \$100K to \$15M. For each tool group the initial capacity is set to be greater than or equal the expected demand in first time period.

There are three different cases in our simulation. In the first case, which we call the base case, there are 4 product families and 43 tool types. In the second one we double the number of product families and keep other parameters unchanged. Therefore this case is referred as the Double Product Family case. In the last one, the number of product families is 43 again but the number of tool types is doubled. The case is called the Double Tool Type case. For each case, we test our algorithm with 2, 4, 8, 16, 32, 64 and 128 demand rays. For each case and for each number of rays, we randomly generate demand rays and run our algorithm four times.

Every time when we run through NG and HIPR, we record following information: number of rays, number of nodes and arcs in the Capacity Expansion Network, total cost, lost sale cost, purchase cost, total number of tools purchased, fill rate, CPU time spent in NG and HIPR and size of the network files. Some of the above items are easy to understand and some deserve more explanation. The purchase cost and the lost sale cost are the first and second terms of the objective function of (ILP2), respectively. The total cost is the sum of these two costs. The unit of measure for all costs is millions of dollars. The fill rate is the expected percentage of satisfied demand, which is one minus the ratio of the total lost sales cost to the total demand, measured in dollars. All times are measured in seconds. The unit of File Size is Megabytes. Our simulations are run on a Sun Microsystems ULTRA 10 workstation with 256MB of memory.

We are interested in the convergence rate of our algorithm as the number of rays increases is the major concern of our simulation. For each case and each number of demand rays we generate and solve four problem instances. Then we compute the mean and standard deviation of the total cost and plot them against the number of rays for each case (see Figure 4, 5 and 6). The plots of fill rate, total number of tools purchased and purchase cost are obtained similarly. It is remarkable that the total cost and other measures plotted are not more sensitive to the number of demand rays. When 64 rays are used, the maximum ratio of standard deviation to mean in all plots is less than $\frac{1}{60}$.

We pay special attention to how fast total cost, fill rate, total number of tools purchased and the purchase cost converge as we increase the number of demand rays. The standard deviation indicates that the number of rays required to achieve convergence is between 64 and 128, uniformly. Surprisingly, the solution of the Double Product Family case converges as fast as it does in the other two cases. Note that the demand vector is in \mathfrak{R}_+^8 in this case, but only \mathfrak{R}_+^4 elsewhere. We expected that more demand rays would be needed to get a satisfactory approximation in the double product family case. However that did not happen. The fact that capacity is added in discrete rather than continuous increments probably helps.

In Tables 1, 2 and 3, we summarize the results of each case. Since we

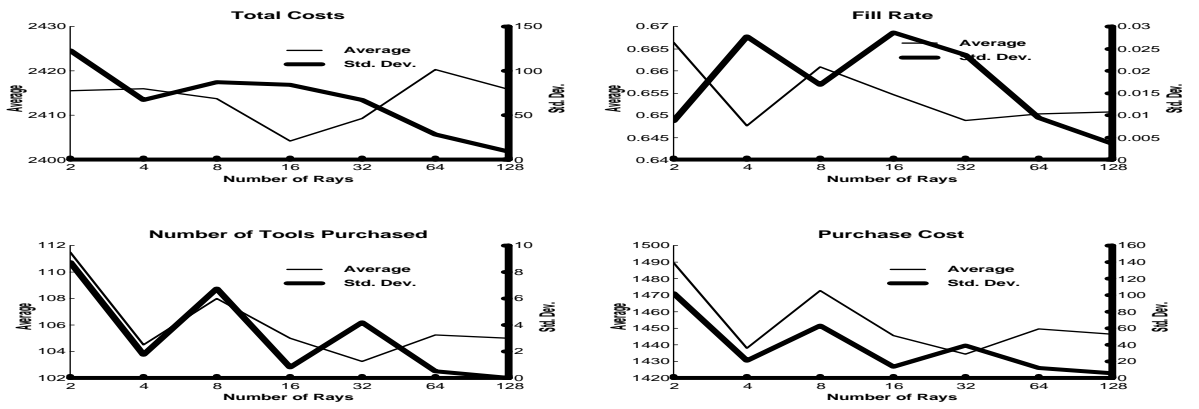


Figure 5: Base Case

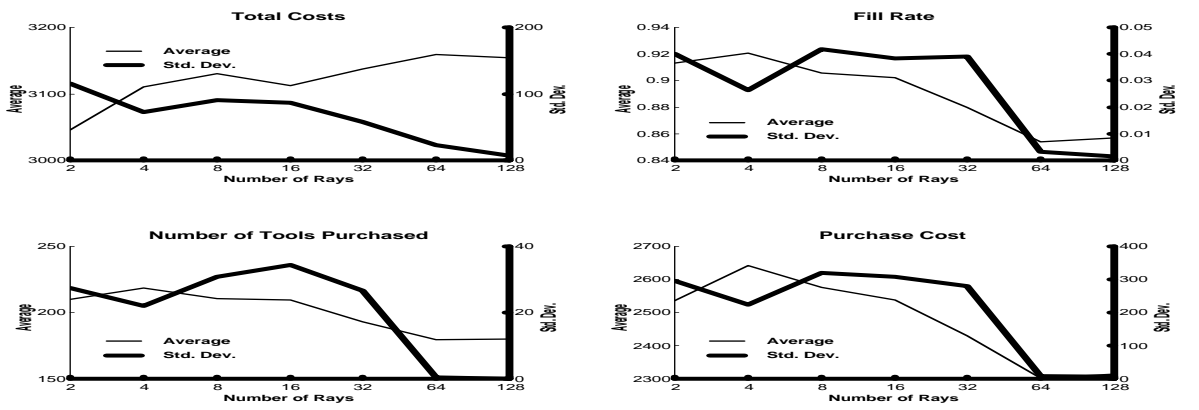


Figure 6: Double Product Family Case

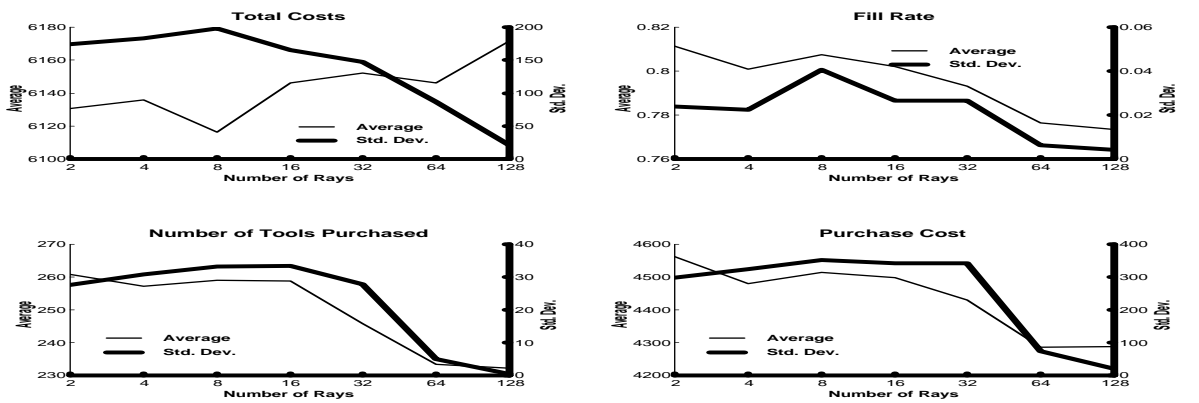


Figure 7: Double Machine Type Case

| | | | | | | | |
|----------------------|-------|-------|-------|-------|-------|-------|-------|
| Demand Rays | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| Nodes (in thousands) | 33 | 55 | 99 | 187 | 363 | 716 | 1,420 |
| Arcs (in thousands) | 73 | 124 | 227 | 431 | 842 | 1,663 | 3,304 |
| Total Cost | 2,416 | 2,416 | 2,414 | 2,404 | 2,409 | 2,420 | 2,416 |
| Lost Sale Cost | 926 | 978 | 941 | 958 | 975 | 970 | 969 |
| Purchase Cost | 1,490 | 1,438 | 1,473 | 1,446 | 1,434 | 1,450 | 1,447 |
| Number of Purchases | 112 | 105 | 108 | 105 | 103 | 105 | 105 |
| Fill Rate | 0.666 | 0.648 | 0.661 | 0.655 | 0.649 | 0.650 | 0.651 |
| NG CPU Seconds | 63.6 | 121 | 234 | 459 | 907 | 1,816 | 3,629 |
| HIPR CPU Seconds | 0.32 | 0.66 | 1.48 | 3.03 | 6.16 | 13.17 | 33.07 |
| File Size | 1.9 | 3.3 | 6.2 | 12.3 | 24.3 | 48.3 | 99 |

Table 1: Base Case (4 product families and 43 tool types)

| | | | | | | | |
|----------------------|-------|-------|-------|-------|-------|-------|-------|
| Demand Rays | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| Nodes (in thousands) | 33 | 55 | 99 | 187 | 363 | 716 | 1,420 |
| Arcs (in thousands) | 73 | 125 | 230 | 438 | 854 | 1,687 | 3,352 |
| Total Cost | 3,046 | 3,110 | 3,130 | 3,113 | 3,137 | 3,160 | 3,155 |
| Lost Sale Cost | 511 | 467 | 553 | 574 | 707 | 857 | 841 |
| Purchase Cost | 2,535 | 2,643 | 2,577 | 2,539 | 2,430 | 2,303 | 2,314 |
| Number of Purchases | 210 | 219 | 211 | 210 | 193 | 180 | 180 |
| Fill Rate | 0.913 | 0.920 | 0.906 | 0.902 | 0.880 | 0.854 | 0.857 |
| NG CPU Seconds | 72.6 | 136 | 262 | 515 | 1,023 | 2,034 | 3,927 |
| HIPR CPU Seconds | 0.63 | 1.1 | 2.04 | 4.89 | 10.08 | 24.34 | 49.09 |
| File Size (in MB) | 2.1 | 3.5 | 6.5 | 13.0 | 25.7 | 51.1 | 100.0 |

Table 2: Double Product Family Case (8 product families and 43 tool types)

repeat each case 4 times, every number in the following tables is the average of 4 data points. Now consider the running time of both NG and HIPR. The size of the network, both the number of nodes and the number of arcs, is $O(\eta)$, where $\eta = TRMN$, T is the number of time periods, R is the number of demand rays, M is the number of tool types and N is the average number of potential tool purchases per tool type. Thus, the number J of tools is $O(MN)$. The running time of NG is $O(\eta)$. Recalling that the time-complexity of max-cut problem with V nodes and $O(V)$ arcs is $O(V^2 \log V)$, [27] the running time of HIPR is at most $O(\eta^2 \log \eta)$. Tables 1, 2 and 3 fully support this time-complexity with respect to R , M , and the lack of dependence on the number of product families.

| | | | | | | | |
|----------------------|-------|-------|-------|-------|-------|-------|--------|
| Demand Rays | 2 | 4 | 8 | 16 | 32 | 64 | 128 |
| Nodes (in thousands) | 66 | 110 | 198 | 374 | 727 | 1,431 | 2,840 |
| Arcs (in thousands) | 158 | 273 | 502 | 962 | 1,881 | 3,720 | 7,398 |
| Total Cost | 6,131 | 6,136 | 6,117 | 6,146 | 6,152 | 6,146 | 6,172 |
| Lost Sale Cost | 1,569 | 1,657 | 1,602 | 1,648 | 1,721 | 1,861 | 1,885 |
| Purchase Cost | 4,562 | 4,479 | 4,515 | 4,498 | 4,431 | 4,285 | 4,287 |
| Number of Purchases | 261 | 257 | 259 | 259 | 246 | 234 | 232 |
| Fill Rate | 0.812 | 0.801 | 0.808 | 0.802 | 0.793 | 0.777 | 0.774 |
| NG CPU Seconds | 205 | 395 | 773 | 1,537 | 3,047 | 6,162 | 12,109 |
| HIPR CPU Seconds | 1.96 | 3.20 | 7.42 | 15.70 | 27.19 | 82.04 | 169.70 |
| File Size (in MB) | 4.2 | 7.3 | 13.9 | 26.9 | 53.4 | 107.8 | 218 |

Table 3: Double Machine Type Case (4 product families and 86 tool types)

Also note that the CPU time for running HIPR is much shorter than the CPU time for running NG, especially when the number of rays is large. The MATLAB code used to generate Capacity Expansion Network is not designed for efficiency; re-coding NG with such programming languages as C would dramatically improve the CPU times for NG.⁸ The reported CPU times for our capacity planning algorithm are viable.

6 Conclusions

In this paper, we develop a capacity expansion model for multi-product, multi-machine manufacturing systems with stochastic demand. We assume that the demand can be approximated by a distribution whose support is a collection of a finite number of rays. We also make an assumption that capacity is allocated proportionally if it is not sufficient. Those two assumptions are essential in our model formulation. We prove that the optimal capacity planning problem is equivalent to a maximum flow/minimum cut problem. We construct the Capacity Expansion Network and do numerical simulation on industrial data. Our experiment shows that a reasonable number of rays

⁸In this paper, we did not implement NG in C for two reasons: (1) this code is intended to be a research prototype, and (2) the asymptotic running time of the algorithm is not bounded by NG, but by HIPR.

(< 128) are sufficient to get a stable plan, and that the CPU times are very predictable and very reasonable for a planning problem of this type.

References

- [1] Angelus, A., E.L. Porteus and S.C. Wood, (1997). *Optimal sizing and timing of capacity expansions with implications for modular semiconductor wafer fabs*. Research Paper No.1479. Graduate School of Business, Stanford University.
- [2] Bard, J., K. Srinivasan and D. Tirupati (1999). *An optimization approach to capacity expansion in semiconductor manufacturing facilities*. International Journal of Production Research, Vol.37, No.15 : 3359-3382.
- [3] Bean, J.C., J.L. Higle and R.L. Smith (1992). *Capacity expansion under stochastic demands*. Operations Research Vol.40 Supp. No.2: S210-S216.
- [4] Berman, O., Z. Ganz and J.M. Wagner (1994). *A stochastic optimization model for planning capacity expansion in a service industry under uncertain demand*. Naval Research Logistics, Vol.41 : 545-564.
- [5] Bermon, S. and S.J. Hood (1999). *Capacity Optimization Planning System (CAPS)*. Interfaces, Vol.29, No.5 : 31-50.
- [6] Bhatnagar, S., E. Fernandez-Gaucherand, M.C. Fu, Y. He and S.I. Marcus (1999). *A Markov decision process for capacity expansion and allocation*. Technical report, ISR, University of Maryland, College Park, MD.
- [7] Birge, J.R. and F. Louveaux (1997). *Introduction to Stochastic Programming*. Springer Verlag, New York.
- [8] Chen, Z.-L., S. Li and D. Tirupati (1998). *A scenario based stochastic programming approach for technology and capacity planning*. Working paper 98-04, Department of Systems Engineering, University of Pennsylvania, Philadelphia, PA.
- [9] Chou, W. and J. Everton (1996). *Capacity planning for development wafer fab expansion*. IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop. ASMC 96 Proceedings, Cambridge, MA: 17-22.
- [10] Nguyen, Q.C. and V. Venkateswaran (1993). *Emplementation of Goldberg-Tarjan Maximum Flow Algorithm*. Network Flows and Matching: First DIMACS Implementation Challenge, pages 19-42. AMS.

- [11] Çakanyıldırım, M, R. Roundy and S.C. Wood (1999). *Machine purchasing strategies under demand- and technology-driven uncertainties*. Technical paper no: 1250, SORIE, Cornell University, NY.
- [12] Davis, M.H.A., M.A.H. Dempster, S.P. Sethi and D. Vermes (1987). *Optimal capacity expansion under uncertainty*. Advances in Applied Probability Vol.19: 156-176.
- [13] Dixit, A. (1997). *Investment and employment dynamics in the short run and the long run*. Oxford Economic Papers, Vol.49 No.1: 1-20.
- [14] Eberly, J.C. and J.A. Van Mieghem (1997). *Multi-factor dynamic investment under uncertainty*. Journal of Economic Theory Vol.75: 345-387.
- [15] Eppen, G.D., R.K. Martin and L. Scrage (1989). *A scenario approach to capacity planning*. Operations Research, Vol. 37, No.4: 517-527.
- [16] Escudero, L.F., P.V. Kamesam, A.J. King and R.J.B. Wets (1993). *Production planning with scenario modelling*. Annals of Operations Research, Vol 43: 311-335.
- [17] Swaminathan, J.M. (2000). *Tool capacity planning for semiconductor fabrication facilities under demand uncertainty*. European Journal of Operational Research, Vol.120, No.3: 545-558.
- [18] Karabuk, S. and S.D. Wu (1999). *Coordinating Strategies Capacity Planning in the Semiconductir Industry*. Technical Report 99T-11, Department of IMSE, Lehigh University.
- [19] Luss, H. (1982). *Operations research and capacity expansion problems: a survey*. Operations Research, Vol.30, No.5: 907-947.
- [20] Rajagopalan, S., M.R. Singh and E.M. Morton (1998). *Capacity expansion and replacement in growing markets with uncertain technological breakthroughs*. Management Science, Vol.44, No.1 : 12-30.
- [21] Rocklin, S.M., A. Kashper and G.C. Varvaloucas (1984). *Capacity expansion / contraction of a facility with a demand augmentation dynamics*. Operations Research, Vol.32, No.1: 133-147.

- [22] Çakanyıldırım, M and R. Roundy (1999). *SeDFAM: Semiconductor Demand Forecast Accuracy Model*. Technical paper no: 1230, SORIE, Cornell University, NY.
- [23] Sethi, S.P., M. Taksar and Q. Zhang (1995). *Hierarchical capacity expansion and production planning decisions in stochastic manufacturing systems*. Journal of Operations Management Vol.12: 331-352.
- [24] 1998 Annual Report & Directory. Semiconductor Industry Association, San Jose, California.
- [25] Wets, R.J.B. (1989). *Stochastic programming*. Chapter VIII of Optimization: Handbooks in OR & MS ed. G.L. Nemhauser et al. Elsevier Science Publishers B.V. North-Holland.
- [26] Law, A.V. and W.D. Kelton. (2000). *Simulation Modeling and Analysis -3rd ed.*. McGraw-Hill.
- [27] Gallo, G., M.D. Grigoriadis and R.E. Tarjan. (1989) A fast parametric maximum flow algorithm and applications. SIAM J. Comput., Vol.18, No.1:30-55, 1989.

Appendix

A Proof of (2)

$$\begin{aligned}
E_{\mathbf{D}(t)}\{\mathbf{c} \cdot [\mathbf{D}(t) - \mathbf{P}(t)]^+\} &= E_{I(t)}\{E_{\mathbf{D}(t)}\{\mathbf{c} \cdot [\mathbf{D}(t) - \mathbf{P}(t)]^+ \mid I(t)\}\} \\
&= E_{I(t)}\{E_{\Delta_i(t)}\{[\Delta_i - \sigma_i^*(t)]^+ \mathbf{c} \cdot \boldsymbol{\phi}_i(t) \mid I(t) = i\}\} \\
&= E_{I(t)}\{E_{\Delta_i(t)}\{n_{\Delta_i(t)}(\sigma_i^*(t)) \mathbf{c} \cdot \boldsymbol{\phi}_i(t) \mid I(t) = i\}\} \\
&= \sum_i p_i \{n_{\Delta_i(t)}(\sigma_i^*(t)) \mathbf{c} \cdot \boldsymbol{\phi}_i(t)\}.
\end{aligned}$$

B Detailed Proof of Theorem 1

The formulation (ILP1) is simple in appearance. We can eliminate some variables and constraints by considering lead times. Recall that $\lambda(m)$ is the purchase lead time for a type m tool, and that $x(j, t)$ has already been fixed if $t \leq \lambda(m)$. These dependencies can be made explicit. We denote by λ_j the lead time of the purchase of tool j , and assume that the lead time depends on the type of tool; i.e.,

$$\lambda_j = \lambda(m) \text{ where } j \sim (m, n).$$

Then, $x(j, t)$ is input data if $t \leq \lambda_j$, and it is a variable for $t > \lambda_j$.

As a basis for re-formulating (ILP1), we define subsets of indices which will correspond to variables in the reformulation. Let

$$\mathcal{J} = \{(j, t) : t > \lambda_j \text{ and } x(j, \lambda_j) = 0\}$$

index tools j that do not become available by lead time λ_j in the periods after λ_j . Let

$$\mathcal{I} = \{(i, j, t) : \text{either } t > \lambda_{j'} \text{ or } x(j', t) = 1 \forall j' \in \pi_t^i(j)\}$$

index (i, j, t) such that every tool in the precedence set $\pi_t^i(j)$ of tool j along demand ray i is or may be available for purchase in period t . For $(i, j, t) \in \mathcal{I}$,

depending on when other tools are purchased, the purchase j of in period t can increase capacity along ray i . We re-formulate (ILP1) as follows.

$$(ILP2) \quad \min \sum_t \{ \sum_j [\mu(j, t)x(j, t)] - \sum_{i, j} [p_i(t)\nu(i, j, t)z(i, j, t)] \}$$

$$\text{such that } x(j, t) \leq x(j, t+1) \quad \forall (j, t) \in \mathcal{J} \quad (11)$$

$$z(i, j, t) \leq x(j, t) \quad \forall (i, j, t) \in \mathcal{I}, \forall (j, t) \in \mathcal{J} \quad (12)$$

$$z(i, \gamma_t^i(j), t) \leq z(i, j, t) \quad \forall (i, j, t) \in \mathcal{I} \quad (13)$$

$$x(j, t) \quad \text{binary } \forall (j, t) \in \mathcal{J} \quad (14)$$

$$z(i, j, t) \quad \text{binary } \forall (i, j, t) \in \mathcal{I} \quad (15)$$

Lemma 3 *(ILP1) and (ILP2) are equivalent.*

Proof: Note that if $(j, t) \notin \mathcal{J}$ or $(i, j, t) \notin \mathcal{I}$, then the value of $x(j, t)$ or $z(i, j, t)$ is determined by the input data and by (7), (8). Furthermore, feasible solutions to (ILP1) are also feasible to (ILP2). We claim that every feasible solution to (ILP2) can be augmented by the input data to yield a solution feasible to (ILP1). From the input data, we impute values of $x(j, t)$ for $(j, t) \notin \mathcal{J}$, and also of $z(i, j, t)$ for $(i, j, t) \notin \mathcal{I}$.

We note that the complement of \mathcal{J} is the disjoint union of

$$\mathcal{J}_0^c = \{(j, t) \mid t \leq \lambda_j \text{ and } x(j, t) = 0\}, \text{ and}$$

$$\mathcal{J}_1^c = \{(j, t) \mid x(j, \min(t, \lambda_j)) = 1\}.$$

If $(j, t) \in \mathcal{J}_0^c$, then $x(j, t) = 0$ is input data and (4) holds trivially. If $(j, t) \in \mathcal{J}_1^c$, then by (11), we get $x(j, t) = 1$. Since $(j, t) \in \mathcal{J}_1^c$ implies $(j, t+1) \in \mathcal{J}_1^c$, we obtain (4) for $(j, t) \in \mathcal{J}_1^c$. Let

$$\mathcal{I}^c = \{(i, j, t) \mid \exists j' \in \pi_j^i(t) \text{ s.t. } x(j', t) = 0 \text{ and } t \leq \lambda_{j'}\}$$

be the complement of \mathcal{I} . We set $z(i, j, t) = 0$ for all $(i, j, t) \in \mathcal{I}^c$, and show that (8) holds if either $(i, j, t) \in \mathcal{I}^c$ or $(j, t) \notin \mathcal{J}$.

If $(i, j, t) \in \mathcal{I}^c$, then we have $z(i, j, t) = 0$; if $(j, t) \in \mathcal{J}_1^c$, then we have $x(j, t) = 1$. In either case, (8) holds. The only remaining case is $(j, t) \in \mathcal{J}_0^c$, which actually implies $(i, j, t) \in \mathcal{I}^c$ (set $j' = j$ in the definition of \mathcal{I}^c). Thus (8) holds. Furthermore, if $(i, j, t) \in \mathcal{I}^c$, then the immediate successor $\gamma_t^i(j)$ of j is also in \mathcal{I}^c , satisfying (7). Thus every feasible solution for (ILP2) with

the input data and the imputed values of $x(j, t)$ and $z(i, j, t)$ corresponds to a feasible solution to (ILP1).

Note that $x(j, t) = 1$ if $(j, t) \in \mathcal{J}_1^c$. Since $z(i, j, t) = 0$ for $(i, j, t) \in \mathcal{I}^c$ in a feasible solution to (ILP1), the objective of (ILP1) is equal to

$$\sum_{\mathcal{J}} [\mu(j, t)x(j, t)] + \sum_{\mathcal{J}_1^c} \mu(j, t) - \sum_{\mathcal{I}} [p_i(t)\nu(i, j, t)z(i, j, t)],$$

which differs from the objective of (ILP2) by a constant. QED

Theorem 1 *(ILP1) is equivalent to a minimum-cut problem in the Capacity Expansion Network.*

Proof: By the previous lemma, it suffices to show that (ILP2) is equivalent to a minimum cut problem. Suppose there is a cut of the Capacity Expansion Network which separates SOURCE and SINK. Use R to represent the set of z -nodes which are on the same side of the cut as SINK. Obviously, R can be decomposed by time period into T disjoint subsets R_t , one for each time period. Similarly, S represents the set of x -nodes which are on the same side of the cut as SINK, and $S = \cup_{t=1}^T S_t$.

The correspondence between the cut and the original variables is that all the x -nodes and z -nodes on the same side as SOURCE assume one; i.e.,

- $j \notin S_t$ iff $x(j, t) = 1$, and
- $(i, j) \notin R_t$ iff $z(i, j, t) = 1$.

Since the capacity of some arcs is infinite, the capacity of the cut could be infinite. The cut capacity is finite if and only if the following 3 conditions are satisfied:

1. If node $z(i, j, t) \notin R_t$, then node $x(j, t) \notin S_t$. Equivalently, $x(j, t) \geq z(i, j, t)$.
2. If node $z(\gamma_t^i(j)) \notin R_t$ then $z(i, j, t) \notin R_t$. Equivalently, $z(\gamma_t^i(j)) \leq z(i, j, t)$.
3. If node $x(j, t) \notin S_t$ then node $x(j, t+1) \notin S_{t+1}$. Equivalently, $x(j, t) \leq x(j, t+1)$.

Thus the capacity of the cut is finite if and only if \mathbf{x} and \mathbf{z} are feasible for (ILP2). If the cut capacity is finite, then it is equal to

$$\begin{aligned} \sum_t \left[\sum_{j \notin S_t} \mu(j, t) + \sum_{(i,j) \in R_t} p_i(t) \nu(i, j, t) \right] \\ = \sum_t \sum_{j \notin S_t} \mu(j, t) + \sum_{\mathcal{I}} p_i(t) \nu(i, j, t) - \sum_t \sum_{(i,j) \notin R_t} p_i(t) \nu(i, j, t). \end{aligned}$$

Note that the middle term is constant, and is independent of $x(j, t)$ and $z(i, j, t)$. Therefore, minimizing the cut capacity is equivalent to minimizing

$$\begin{aligned} \sum_t \sum_{j \notin S_t} \mu(j, t) - \sum_t \sum_{(i,j) \notin R_t} p_i(t) * \nu(i, j, t) \\ = \sum_{\mathcal{J}} \mu(j, t) x(j, t) - \sum_{\mathcal{I}} p_i(t) \nu(i, j, t) z(i, j, t). \quad QED \end{aligned}$$

C Derivation of the Distribution of $\Delta_i(t)$

Given any demand ray $\phi_i(t)$, the density function of $\Delta_i(t)$ can be expressed as follows:

$$f_{\Delta_i}(r) = \frac{f_{\mathbf{D}(t)}(r\phi_i(t))r^{K-1}}{\int_0^\infty f_{\mathbf{D}(t)}(s\phi_i(t))s^{K-1}dr}$$

where $f_{\mathbf{D}(t)}(\mathbf{x})$ is the density function of $\mathbf{D}(t) \in \Re^K$. The factor r^{K-1} is required because the demand rays spread apart as one moves away from the origin, as is the case in polar coordinates. For example, assume that $\mathbf{D}(t) \in \Re^4$, which has a log-normal multi-variate distribution with parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Then

$$f_{\mathbf{D}(t)}(\mathbf{x}) = \prod_{j=1}^4 x_j^{-1} (2\pi)^{-2} [Det(\boldsymbol{\Sigma})]^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\log \mathbf{x} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\log \mathbf{x} - \boldsymbol{\mu})\right]$$

Therefore,

$$f_{\Delta_i}(r) = \frac{\sqrt{b}}{\sqrt{2\pi r}} \exp\left[-\frac{1}{2}((\log r)\mathbf{e} + \mathbf{a})' \boldsymbol{\Sigma}^{-1} ((\log r)\mathbf{e} + \mathbf{a}) + \frac{d - \frac{c^2}{4b}}{2}\right]$$

where

$$\mathbf{a} = \log \phi_i(t) - \boldsymbol{\mu},$$

$$\begin{aligned}
b &= \sum_{i,j} (\boldsymbol{\Sigma}^{-1})_{i,j}, \\
c &= \sum_{i,j} (a_i + a_j) (\boldsymbol{\Sigma}^{-1})_{i,j}, \\
d &= \sum_{i,j} (a_i a_j) (\boldsymbol{\Sigma}^{-1})_{i,j},
\end{aligned}$$

and (e) is a vector consisting of 1's only. In other words, the random variable $\Delta_i(t)$ has log-normal distribution with parameters $(-\frac{c}{2b}, \frac{1}{b})$. Now we are able to compute the partial expectation.

$$\begin{aligned}
n_{\Delta_i(t)}(r) &= \int_r^\infty (x - r) f(x) dx \\
&= \exp\left(-\frac{c}{2b} + \frac{1}{2b}\right) [1 - P_1(\log r)] - r[1 - P_2(r)]
\end{aligned}$$

where P_1 is the cumulative density function for the normal distribution with parameters $(-\frac{c}{2b} + \frac{1}{b}, \frac{1}{b})$ and P_2 is the cumulative density function of the log-normal distribution with parameters $(-\frac{c}{2b}, \frac{1}{b})$.