Robust Capacity Planning in Semiconductor Manufacturing

Francisco Barahona*
Stuart Bermon*
Oktay Günlük*
Sarah Hood†

October, 2001 (Revised February, 2005)

Abstract

We present a stochastic programming approach to capacity planning under demand uncertainty in semiconductor manufacturing. Given multiple demand scenarios together with associated probabilities, our aim is to identify a set of tools that is a good compromise for all these scenarios. More precisely, we formulate a mixed-integer program in which expected value of the unmet demand is minimized subject to capacity and budget constraints. This is a difficult two-stage stochastic mixed-integer program which can not be solved to optimality in a reasonable amount of time. We instead propose a heuristic that can produce near-optimal solutions. Our heuristic strengthens the linear programming relaxation of the formulation with cutting planes and performs limited enumeration. Analyses of the results in some real-life situations are also presented.

*IBM T. J. Watson Research Center, Yorktown Heights, NY 10598
†IBM Microelectronics Division

1 Introduction

In the semiconductor industry, the determination of the number of manufacturing tools needed to manufacture forecasted product demands, is particularly difficult because of its sensitivity to product mix, the uncertainty in future demand, the long lead time for obtaining tools and large tool costs. Tools used in the manufacturing process are highly customized and made to order with delivery lead times up to 18 months and costs ranging from below $1 million to over $13 million. The total capital investment for a plant is typically several billion dollars. The product demand is highly volatile and therefore it is difficult to predict the demand profile for the mix of products over several months or years. Planning for a single demand profile can result in a large gap between planned and needed capacity when the actual demand materializes. Our goal in this paper is to explore a stochastic programming approach to produce a tool set that is robust with respect to demand uncertainty. To achieve this, we consider multiple demand scenarios instead of a single one. We associate a probability with each scenario and formulate a mixed-integer programming model in which the expected value of the unmet demand is minimized subject to capacity and budget constraints. We solve the resulting two-stage stochastic mixed-integer program to near-optimality using a heuristic based on cutting planes and limited enumeration.

Capacity planning in the semiconductor industry is typically implemented using spreadsheets [3, 16, 19] or when considerations of cycle time are important, using discrete-event simulation [22]. These methods do not involve the application of optimization techniques and require multiple runs to find a solution which may effectively use the tools to "maximize" profit, revenue or throughput. Linear programming based techniques have been applied to this problem. Leachman [12] presents several models for production planning problems. Yang [23] describes a method for planning tool purchases and product mix to maximize throughput subject to space constraints in the lithography area of a semiconductor line. Bermon and Hood [4] present a method that 1) given a specified tool set, finds the optimal mix of products to maximize profit when product volumes are allowed to vary within an acceptable production range or 2) calculates the minimum number of tools required to manufacture a specified demand. The formulation deals with parallel, unrelated tool groups that may perform the same operation at different rates and captures the preferential order in which to use such tool groups. This linear programming

based system, called the Capacity Optimization Planning System (CAPS), has been the primary decision support system for capacity planning at IBM's largest semiconductor manufacturing line since 1996. Our work extends this system.

In all of the methods described above, the input consists of a single demand profile. In this paper we describe a method using stochastic integer programming that finds a tool set that performs well across a range of scenarios. We call our decision support system the Stochastic Capacity Optimization System (SCAPS). A version of this work, oriented for an audience with no background in optimization, has been presented in [9]. A similar model has been studied in [21]. We present four different solution methods, two of them are similar to the ones given in [21]. In addition we test our approach with real-world data. A simpler version of this model is also studied in [20]. A related model, although for a single period, also appears in [15]. Also see [1] for related work on capacity planning under uncertainty.

Planning under uncertainty goes back to the early 1950s, see [6] where a model for aircraft assignment under uncertainty is studied. Capacity planning under uncertainty has been studied in a variety of areas. See [14, 18] for heavy process industries; [11] for communications networks; [5] for automobile industries; and [17] for electronic goods.

The paper is organized as follows: In Sections 2 and 3 we describe the manufacturing process and how it influences the model; in Section 4 we present our stochastic integer programming model; in Section 5 we describe the generation of data; in Section 6 we present the solution approach; and in Section 7 we analyze the results.

2 The Manufacturing Process

Semiconductor integrated circuit manufacturing requires hundreds of process steps performed by several hundred unique tool groups and involving tolerances of significantly less than one ten-thousandth of a millimeter. Hundreds to thousands of identical circuit patterns, i.e., chips, requiring up to 25 patterned layers, are built on a single silicon wafer substrate of 5 to 12 inches diameter. The process flow through the manufacturing line is termed reentrant since each layer is built using the same or similar processes, with wafers returning to the same tools a number of times.

The 8 major processing areas are chemical clean, oxidation, photo-lithography,

plasma/chemical etch, ion implant, chemical-mechanical polishing planarization, metal/ insulator deposition, and anneal (heat treatment). Initially a cleaned silicon wafer is exposed to oxygen in a furnace in order to form a layer of silicon dioxide on its surface. An additional nitride insulating layer on top of the oxide is grown by chemical vapor deposition. A film (the photo-resist layer) is deposited on the oxide/nitride surface and exposed to ultraviolet light through a very-high resolution patterned mask. The areas that have been exposed to the light undergo a chemical transformation that allows them to be washed away in a chemical solvent called the developer. The uncovered oxide/nitride layer is then etched away to expose the silicon surface beneath. Charged impurity atoms or ions, called dopants, are then implanted by acceleration in an intense electric field through the "windows" in the thick oxide/nitride film into the otherwise insulating silicon surface to form conducting regions. In these regions electrical current may be carried by negative charges (n-type regions) or by positive charges (p-type regions) depending on the type of dopant. Such heavily-doped implanted areas are used to form the source and drain regions of a metal-oxide-semiconductor transistor. The source and drain are separated by a thin, lightly-doped region of opposite polarity, above which, photo-resist patterning is used to form a very thin oxide layer. This region becomes the gate or channel region of the transistor. Metal-layer contacts or electrodes are then deposited onto these source, insulated-gate, and drain regions to complete the transistor. With a voltage applied between the source and drain, but no voltage applied to the gate, the transistor is normally off. When a small voltage is applied to the gate, however, electrical current flows between the source and drain, thus providing for the transistor amplifying and switching action.

Repeated oxidation, masking, etching, implanting, and metal deposition steps are used to form the "front-end" layers of the wafer that contain all the active devices. When this is completed, the individual devices are interconnected by a system of fine metal lines and vias (vertical conducting plugs between metal lines on different layers) using a series of photo-resist-patterned metal and insulator deposition and etching steps. Up to eight layers of such metal interconnect layers are employed with metal line widths as small as 0.25 microns (a human hair is 100 microns in diameter).

Following the patterning of the last metal layer, the wafer is covered by a final dielectric layer for passivation (prevention of chemical reaction with the environment) with openings etched in this film for making electrical contacts. The wafer is then diced into individual chips, each of which is assembled into a module which provides for the

attachment of wire bonds to the chip. Modules undergo heat treatment stressing (burn-in) and final electrical test before shipping.

3 Model Considerations Influenced by The Manufacturing Process

The IBM semiconductor manufacturing line that we study in this paper can process up to 50 different product families (a collection of products with the same routing on the line), each of which require between 400 and 600 process steps in sequence for a total of approximately 25,000 separate operations. For capacity planning purposes, we ignore the sequence that these operations should follow for a particular product. Many of the operations on a given product are actually the same type of operation repeated many times on the same tool group - such as photo-resist patterning on different levels. The number of times a part visits the same tool to undergo the same type of operation is called the number of *passes* for that operation for that product. Note that the same type of operation may also be performed on different products. By aggregating such similar operations within a single product, as well as across products, it is possible to reduce the number of *distinct* operations by an order of magnitude. This reduces the size of the model described in Section 4. To this end, we define a bill-of-materials (BOM) for each product, which comprises the number of passes of each type of distinct operation (hereafter simply referred to as an *operation*) utilized by the product.

As wafers progress through the line, a certain fraction is lost due to breakage and problems in processing. In our model, the BOM parameter incorporates a yield factor that accounts for this wafer yield loss. That is, for every operation and product, this parameter is used to convert the primary decision variable of wafers per day that enter the production line to wafers per day that each tool actually processes. The BOM would usually be constant over time, but because of the yield factor, which generally improves over time, it is time dependent. The yield factors are obtained statistically and are quite reliable. These are on the order of 0.98-0.99.

It is important to understand that the requirements placed on the manufacturing line's capacity may vary greatly from one product to another. Some products may require specialized tools that are of little use in producing other products. Process times may be

widely different for different products on the same tool. That is, the capacity of the line is mix sensitive. Two different mixes with the same number of total wafers per day may require radically different tool sets. One cannot simply say that the total capacity of the line is X wafers per day without stating what the specific mix is.

In the following discussion, the term tool set refers to the entire manufacturing facility. The tool set consists of hundreds of different tool groups each made up of one or more identical tools. The term identical means that each tool in such a tool group is qualified to perform the same set of operations at the same speed with the same reliability. From the perspective of capacity planning, tools belonging to a given tool group are viewed as indistinguishable and therefore a tool group consisting of N identical tools has N times the capacity of a single tool. Different tool groups that can perform at least one operation in common are referred to as parallel and unrelated. They are called parallel because they can be used to perform the common operations in parallel, and unrelated because they perform those operations at different speeds and with different reliability. A tool group may be designated as the preferred or *primary* tool group for the set of operations it can perform or it may be designated as the back-up or *secondary* tool group. In general, there can be several secondary tool groups for every primary tool group. Secondary tools are usually older, slower and/or poorer yielding tools that have been retained for additional capacity. In a large factory, the number of tool groups may exceed 300. The assignment of products to the various operations (BOM) and the assignment of operations to different tool groups is indicated schematically in Figure 1.

4 The Model

It is possible to model this problem as a deterministic integer program if a reliable forecast for future demand is available. In our application, this is not the case. In order to deal with uncertainty, we use a two-stage stochastic integer program. The first stage involves the capacity expansion decisions which have to be made before a reliable demand forecast is available. The second stage involves the actual production decisions, which can be finalized after the demand profile is known with certainty. Our planning horizon is longer than the average tool delivery lead time and consequently a multi-stage stochastic integer programming model would be more appropriate. However, we have not pursued this due to the computational difficulty of solving even the two-stage model.

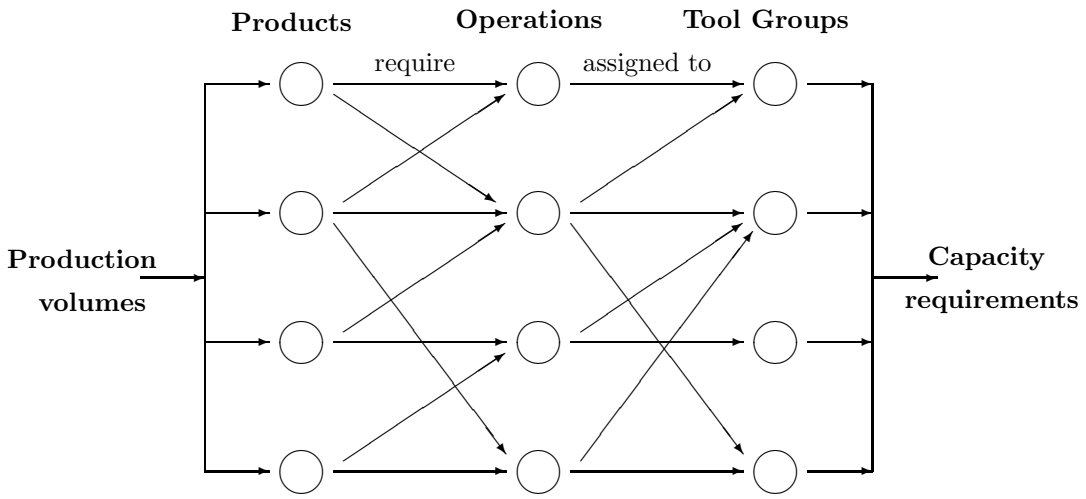

Figure 1: From production volumes to capacity requirements

In our model we assume that several demand profiles, called scenarios, together with associated probabilities are available. Each scenario specifies demand volumes of different products for a number of planning periods. Later in Section 5 we discuss how capacity planners generate the demand data.

Single-period deterministic model

Before describing the detailed model, we next present a simplified version that has a single period and a single scenario, i.e., deterministic demands. We find it easier to describe the dynamics of the model on this simplified version. Below we present the list of indices, variables and data used, and then the model:

Indices of the model

p : products
j : operations
i : tool groups

Data

γ_p	:	expected number of wafers completed per wafer started for product p
d_p	:	demand for product p in wafers per day
$b_{j,p}$	:	number of passes, adjusted for yield, of operation j on product p (BOM)
μ_i	:	initial capacity for tool group i in hrs/day
c_i	:	unit capacity of a tool in tool group i in hrs/day
$h_{i,j}$	:	number of hours to process one wafer through operation j on tool group i
m_i	:	cost of purchasing a new tool in tool group i
β	:	total budget available for buying new tools
α_p	:	upper bound for the unmet demand for product p
$I(j)$	:	collection of tool groups that can perform operation j
$J(i)$	:	collection of operations that can be performed on tool group i

Variables of the model

U_p	:	unmet demand for product p in wafers per day
W_p	:	number of wafers per day that enter the production line for product p
$O_{j,i}$	:	number of wafers per day that require operation j, on tool i
N_i	:	number of new tools bought for tool group i

$$\text{minimize } \sum_p U_p$$

$$\gamma_p W_p + U_p = d_p \quad \text{for all } p \tag{1}$$

$$\sum_p b_{j,p} W_p = \sum_{i \in I(j)} O_{j,i} \quad \text{for all } j \tag{2}$$

$$\sum_{j \in J(i)} h_{i,j} O_{j,i} \leq \mu_i + c_i N_i \quad \text{for all } i \tag{3}$$

$$\sum_i m_i N_i \leq \beta \tag{4}$$

$$U_p \leq \alpha_p \quad \text{for all } p \tag{5}$$

$$\text{all variables} \geq 0 \tag{6}$$

$$N_i \text{ integer valued for all } i. \tag{7}$$

The objective of the model is to minimize the total unmet demand for all products. Equations (1) relate the demand for each product with production variables and unmet demand. The coefficients $\gamma_p \leq 1$ represent a yield factor that accounts for the wafers lost during the manufacturing process of a particular product.

For every operation, the left-hand side of equation (2) gives the total number of wafers that require the operation based on production levels. The right-hand side requires this total to be distributed to the tool groups that can perform the operation.

Constraint (3) requires the total production load not to exceed the available capacity, measured in terms of hours per day of production. Inequality (4) is the budget constraint, and inequalities (5) set upper bounds for the unmet demand for each product. Finally, inequalities (6) require the decision variables to be non-negative, and (7) requires that the number of new tools delivered to be integer valued.

Multi-period two-stage stochastic model

We next present the complete model that has multiple scenarios and multiple time periods. Time periods typically span a few months or a year. Each scenario consists of a demand profile for each time period. These scenarios are generated by capacity planners and each one has a probability associated with it. The resulting stochastic integer program has the objective of minimizing a weighted sum of total expected unmet demand and penalty terms that discourage purchasing primary and secondary tools. The penalties of secondary tools are higher than those of primary tools. If one has some prediction of revenue or profit per unit given by each product, one could multiply the variables U by such values.

The new tool purchase decisions constitute the first-stage variables that are common to all scenarios. If these variables are fixed, then the model decomposes into several deterministic models, one for each scenario. We are assuming that the demand of a particular period should be satisfied with production from the same period. This is a common practice in semiconductor manufacturing industry where inventory is kept very low. It is easy to modify the model so it carries over production from one period to the next, but we have not investigated the computational difficulty of this approach.

We next present the additional indices and data used, and the complete model:

Additional Indices

s : scenarios
t : time periods

Additional Data

π_s : probability of scenario s
q_1 : penalty for buying a primary tool
q_2 : penalty for buying a secondary tool
PT : the set of primary tool groups
ST : the set of secondary tool groups

The remaining data items and all of variables used in the stochastic model are similar to the ones used in the deterministic model except they now have extra indices to account for different scenarios and time periods.

$$\text{minimize} \sum_s \pi_s \sum_{p,t} U_{s,p,t} + q_1 \sum_{i \in PT} \sum_t N_{i,t} + q_2 \sum_{i \in ST} \sum_t N_{i,t} \tag{8}$$

$$\gamma_{p,t} W_{s,p,t} + U_{s,p,t} = d_{s,p,t} \quad \text{for all } s,p,t \tag{9}$$

$$\sum_p b_{j,p,t} W_{s,p,t} = \sum_{i \in I(j)} O_{s,j,i,t} \quad \text{for all } s,j,t \tag{10}$$

$$\sum_{j \in J(i)} h_{i,j,t} O_{s,j,i,t} \leq \mu_{i,t} + c_{i,t} \sum_{\tau=1}^{t} N_{i,\tau} \quad \text{for all } s,i,t \tag{11}$$

$$\sum_i m_{i,t} N_{i,t} \leq \beta_t \quad \text{for all } t \tag{12}$$

$$U_{s,p,t} \leq \alpha_{s,p,t} \quad \text{for all } s,p,t \tag{13}$$

$$\text{all variables} \geq 0, \tag{14}$$

$$N_i \text{ integer valued for all } i. \tag{15}$$

5 Data

The scenarios are typically created by capacity planners to reflect the market outlook based on their knowledge of the industry. They consider several factors such as advances in technology, existing product mix in the marketplace and economic outlook. In our application the scenarios were prepared by a separate team and we can not describe the actual process in detail.

In our examples, we had between 4 and 6 scenarios. One scenario was regarded as the primary or most likely scenario and the remaining scenarios were variants. The number of products was in the range of 30 with approximately 2500 operations requiring 300 tool-groups. The number of periods ranged between 4 and 8. The periods were either a quarter, or a half a year, or a year, and later periods were usually longer. This is a common practice in production planning, see for instance [8].

For a representative instance this model has 2500 integer and 230,000 continuous variables; and 140,000 constraints. When we tried CPLEX 8.1 on a small instance with half as many variables and constraints, we observed that the run terminated with an integrality gap of 41.7% after enumerating more than 30,000 nodes in 20 hours. The branch-and-bound tree required 550 Megabytes of storage. In this experiment we used an RS6000 44P model 270 running at 375 Mhz. Since solving these problems to optimality in a reasonable time seems unrealistic, we developed faster heuristic approaches that we describe in the following section.

6 Solution Process

Our solution procedure consists of the following steps:

Step 1: Relax the integrality requirements (15) and solve the Linear Programming (LP) relaxation;

Step 2: Strengthen this relaxation with valid inequalities and resolve;

Step 3: Fix some of the variables to zero and impose bounds on others using the optimal fractional solution to the strengthened relaxation;

Step 4: Apply branch-and-bound based heuristics to find integer solutions.

The second step of this procedure provides a lower bound on the objective function value of the mixed-integer program, and the last step provides an upper bound. We next discuss these steps in detail.

Strengthening the LP-relaxation

In our earlier experiments we observed that the value of the LP-relaxation of our model was approximately 50% of the value of the best solution we can find to the integer program. To strengthen the LP-relaxation of the model, we use the so-called *residual capacity inequalities* [13] applied to capacity inequalities (11). Our objective here is not only to increase the lower bound on feasible solutions but also to obtain a tighter formulation that might yield better integral solutions when used in the branch-and-bound based heuristic.

Let F be the set of points $[x, y] \in R^n \times R$ that satisfy

$$\sum_{i \in Q} a_i x_i \leq a_0 + y \tag{16}$$

$$0 \leq x_i \leq 1, \text{ for } i \in Q, \quad y \geq 0, \tag{17}$$

$$y \geq 0, \quad \text{integer} \tag{18}$$

where $Q = \{1, \ldots, n\}$. Magnanti et. al., [13] show that the convex hull of F is defined by (16), (17), and the residual capacity inequalities:

$$\sum_{i \in S} a_i x_i - ry \leq a(S) - r\nu, \quad \text{for all } S \subseteq Q, \tag{19}$$

where $a(S) = \sum_{i \in S} a_i$, $\nu = \lceil a(S) - a_0 \rceil$, and $r = a(S) - a_0 - \lfloor a(S) - a_0 \rfloor$. Although there is an exponential number of these inequalities, they admit the following simple separation algorithm [2]. For a given point $(\bar{x}, \bar{y})$ that does not belong to the convex hull of F, the most violated residual capacity inequality (19) is indexed by the set:

$$\bar{S} = \{i \in Q \, : \, \bar{x}_i > \bar{y} - \lfloor \bar{y} \rfloor\}.$$

We obtain a set of the form F for each s, t and p as follows: First, re-write inequality (11) as follows:

$$\sum_{j \in J(i)} \frac{h_{i,j,t}}{c_{i,t}} \; O_{s,j,i,t} \leq \frac{\mu_{i,t}}{c_{i,t}} + \sum_{\tau=1}^{t} N_{i,\tau} \tag{20}$$

to obtain an inequality of the form (16). In this inequality, we treat $(\sum_{\tau=1}^{t} N_{i,\tau})$ as a single integer variable that corresponds to variable y in (16)-(18). We define $O'_{s,j,i,t} = O_{s,j,i,t}/UB_{s,j,i,t}$, where $UB_{s,j,i,t} = \sum_p b_{j,p,t} d_{s,p,t}/\gamma_{p,t}$. Note that $UB_{s,j,i,t}$ is a valid upper bound on $O_{s,j,i,t}$ that has been obtained by combining inequalities (9) and (10). Next we replace the variables O with O' in (20), and treat variables O' as the continuous variables x in (16)-(18). This gives us a set of the type F and we can define residual capacity inequalities accordingly.

We add these valid inequalities in a cutting plane fashion as follows: We first solve the continuous relaxation of (8)-(15) and based on the solution we separate violated inequalities as described above. We generate at most one violated cut for each inequality (11) and include it in the formulation. We then repeat this procedure until no significant improvement is observed in the objective function value. Typically, this took up to 20 iterations resulting in approximately 700 cuts.

Fixing and Bounding the Variables

At the end of the cutting plane phase, we fix to zero and remove all integer variables taking the value zero. This eliminates roughly 90% of the integer variables. For each remaining variable $N_{i,t}$, we impose an upper bound that is equal to the smallest integer which is greater than the current value of $N_{i,t}$. In other words, if the variable had a value of $\delta > 0$, we set its upper bound to $\lceil \delta \rceil$.

Effects of Strengthening, Fixing and Bounding

In order to see the computational performance of this procedure, we apply it to five single-period problems. Our purpose here is to verify that fixing and bounding some of the variables does not result in a significant loss of quality in the solution. The results of the experiments are shown in Table 1. Under the label LP we display the value of the LP relaxation. Under the label LP+cuts we show the LP value obtained after adding residual capacity inequalities. Note that both of these values give a lower bound on the value of the integer program. To obtain a stronger lower-bound, we run branch-and-bound on the strengthened formulation without fixing or bounding any integer variables. We run this with a limit of 50,000 branch-and-bound nodes, and we display the lower bound obtained under the label LB in Table 1. Notice that this was done only for experimental purposes,

since it takes between 6 and 8 hours, and it is not a part of the decision support system (SCAPS) that will be used in practice. Finally, we run branch-and-bound after fixing and bounding the integer variables with a node limit of 5,000 nodes. The value of the best integer solution found at the end of this limited branch-and-bound procedure gives an upper bound on the value of the integer program and it appears under IS in Table 1. We report the resulting gap between IS and LB in the last column. All computational experiments presented in Tables 1-3 were done on an RS6000-590 machine with the OSL package [10].

Case	LP	LP+cuts	LB	IS	Gap
Case 1	482	770	1007	1043	3.6%
Case 2	470	699	859	859	0.0%
Case 3	394	517	655	669	2.1%
Case 4	608	762	1019	1140	12.0%
Case 5	867	925	1073	1146	6.8%

Table 1: Comparison of upper and lower bounds.

As seen in Table 1, the value of the LP-relaxation is far from the lower bound obtained after enumerating many nodes. The cutting planes improve the quality of the LP-relaxation significantly but a noticeable gap still remains. The fixing and bounding procedure, which helps reduce the computing time, does not appear to deteriorate the quality of the solution significantly.

Branch-and-bound based heuristics

To deal with the multi-period models, we tried several heuristic approaches. In all of them we start with the formulation strengthened with cutting planes and restricted with variable fixing.

The first approach is quite straightforward: we simply apply the branch-and-bound procedure with a limit of 5,000 nodes and use the best integral solution found during this enumeration. We call this the *Basic* method.

In the second approach, we delete all variables associated with periods other than the first period to obtain a single period problem and apply the methodology above (i.e., limited branch-and-bound). We then fix the decision variables related with the first period to their value in the best integral solution obtained. We then repeat this procedure with

the second period, fix again and so on. We call this the *Serial* method. This is similar to a heuristic proposed by Swaminathan [21].

The third approach is a generalization of the Serial method where we now deal with two consecutive periods at a time. At every iteration the variables of the first period are treated as integral variables and the variables of the second period are treated as continuous variables. We then fix the variables for the first period and repeat this procedure with later periods. We call this the *Window* method.

In the fourth approach, we again solve two-period models at every iteration where the second period aggregates all future periods. In particular, we start with keeping the first period as it is and aggregate periods two to n into a second period. The variables related to the (aggregate) second period are treated as continuous. After applying the branch-and-bound procedure (with a limit of 5,000 nodes) we fix the variables for the first period. We then repeat the same aggregation procedure for periods 2 to n and so on. This is called the *Aggregate* method. The window and aggregate methods are similar to heuristics used by Forrest [7] in deterministic production planning applications.

In Table 2 we display the solution values given by these approaches. Case 1 involves 6 demand scenarios for 26 products over 8 time periods with the unmet demand held to 0 for 3 key products and the maximum unmet demand for all other products set to 40% of the individual demands. Case 2 utilizes the same set of scenarios, but with the maximum unmet demand globally set to 30% of the demand for all products in all periods. In addition, the budget for the last 5 periods is increased by a cumulative amount of \$100M compared to Case 1. Cases 3 and 4 are separate distinct sets of 4 scenarios representing earlier projections for 24 products over 6 time periods with maximum unmet demand set to 50% of the demand over all products and periods. Case 5 is a variation of Case 4 run over the first 4 of the 6 periods of Case 4, but with increased variance among the scenarios and a significantly larger budget in period 4. Here the globally set value of the maximum unmet demand was 80% of the demand. Under the label OBJ we show the value of the objective function which is a linear combination of unmet demand and penalties for purchasing secondary tools; under time we show the computing time in hours:minutes.

As seen in Table 2, the serial method is the fastest one, and the window method gives better solutions. The basic method failed to produce an integer solution within 44 hours for the first case. Clearly the basic method is neither reliable nor practical.

Case	Serial		Window		Aggregate		Basic	
	OBJ	time	OBJ	time	OBJ	time	OBJ	time
Case 1	935	1:09	896	7:44	917	5:06	-	> 44:00
Case 2	502	1:17	501	9:17	500	8:33	1134	32:31
Case 3	318	1:24	303	4:51	315	5:20	283	24:36
Case 4	343	2:35	312	10:43	314	9:34	336	22:58
Case 5	349	3:32	348	11:34	349	10:02	380	17:18

Table 2: Comparison of the four solution approaches.

7 Analysis of the results

One possible way to quantify the advantage of using the stochastic approach is to pick a possible scenario and compare its unmet demand when planning decisions are made using SCAPS with the unmet demand when planning decisions are made based on a single most likely scenario. We present below this kind of analysis for Case 5 described in the previous section. In this section we present the results in terms of met demand.

In Figure 2 we show the demand met by the tool set generated by SCAPS (labeled SCAPS) versus the demand met by a tool set obtained by planning only for a particular scenario (labeled BAU for "business as usual"), for 4 scenarios over 4 successive yearly time periods. The topmost demand curve is the total forecasted volume expressed in wafers per day (w/d) for each scenario. Scenario SCN1 is the plan of record profile, that is, the forecasted deterministic demand that is the single scenario against which tools would be purchased in the business as usual case. The other scenarios, SCN2 and SCN3 have roughly the same total demand as SCN1, but constitute different mixes that respectively represent a faster and slower introduction of a new technology compared to SCN1. SCN4 represents a case with higher demands. The dollar investment (in $M) to completely meet the demand for each scenario is shown listed under the respective scenario label along the x-axis. Figure 3 shows a histogram for period 3 for the demand volumes of the 10 most important products for each of the 4 scenarios, from which one may gauge the extent of the variation by product from scenario to scenario. The units here are wafers per day (w/d).

The bottom (BAU) dashed curve in Figure 2 shows the met demand resulting from purchasing a tool set that just satisfies the demand for the deterministic scenario SCN1

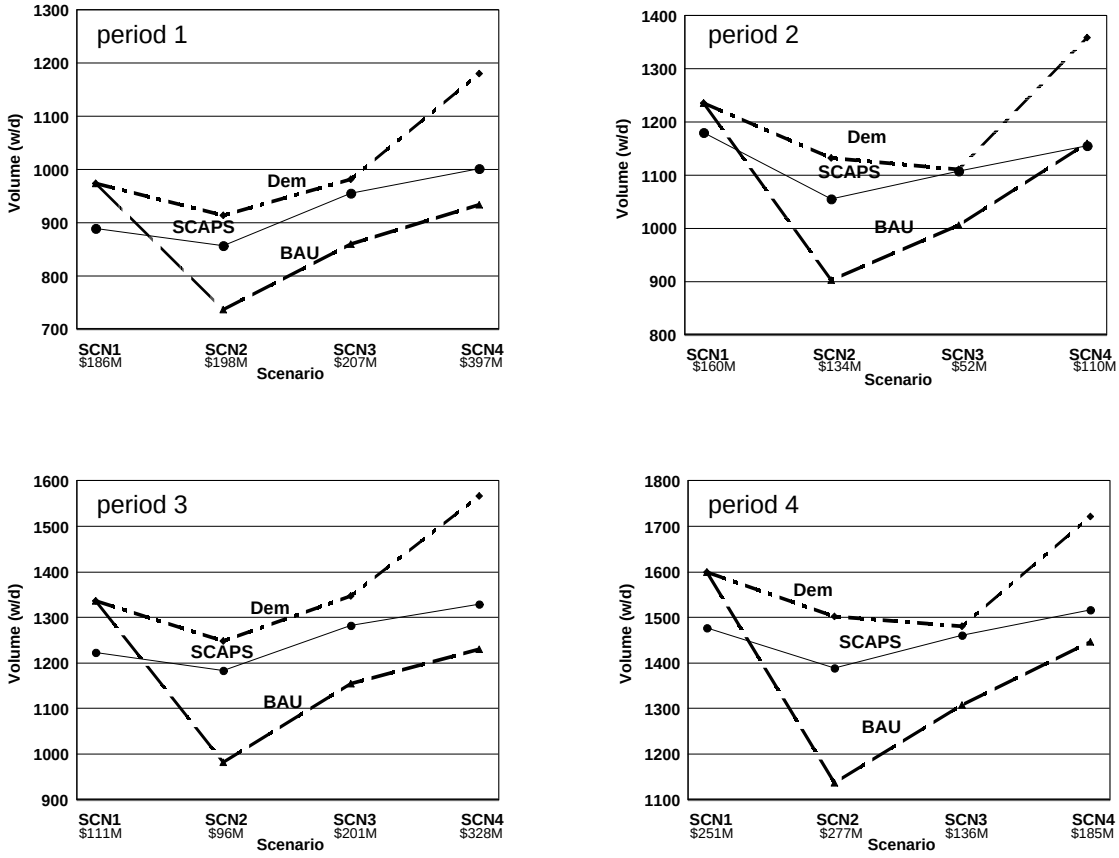

Figure 2: Comparison of met demands for 4 scenarios using the tool set generated by SCAPS to the BAU tool set designed to just produce scenario SCN1. The demand units are wafers per day (w/d).

at the indicated cost for SCN1 in each period. Of course, the unmet demand is zero for SCN1 for this BAU tool set, but can be quite large (up to several hundred wafers per day) should one of the alternate scenarios materialize. On the other hand, the tool set produced by SCAPS, which minimizes the expected value of the unmet demand across all the scenarios, although leaving some unmet demand for SCN1 (6 -10%), behaves much better across all of the other scenarios. The budget level in the stochastic case, which is an input to the model, has been set equal to what is required for SCN1 to meet the demand. In other words, we obtain this markedly improved performance across the range

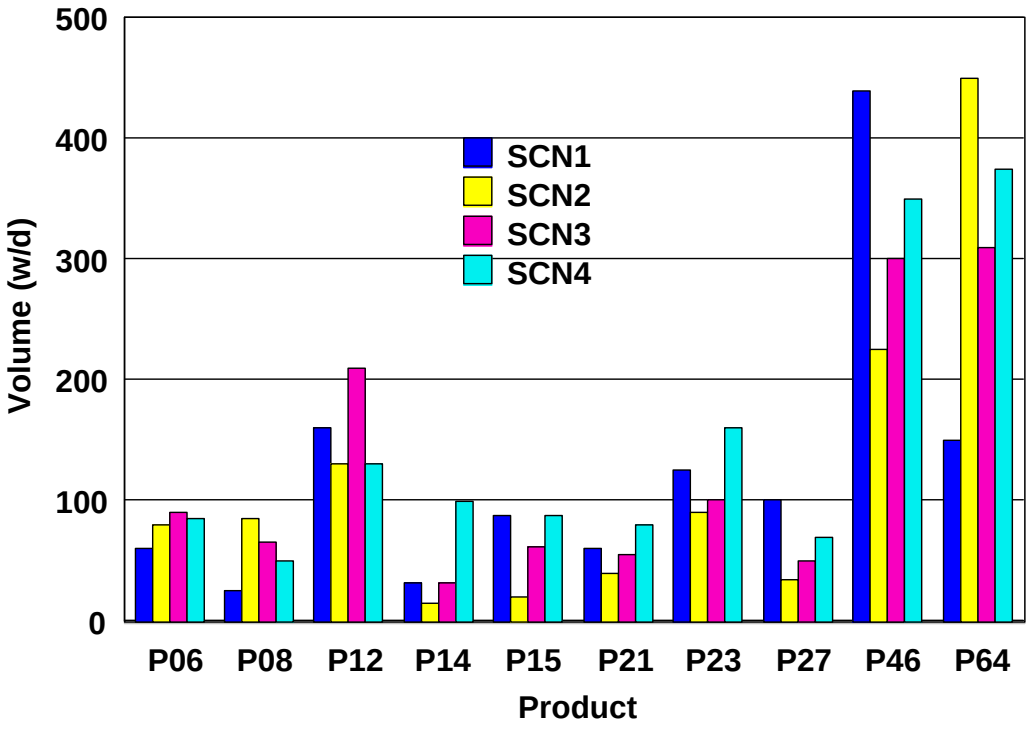

Figure 3: Comparison for period 3 of the demand volumes in wafers per day (w/d), for the 10 most important products for the 4 scenarios exhibited in Figure 2.

of scenarios at no extra cost compared to the BAU case.

The probabilities used in this example were: SCN1 (0.40); SCN2 (0.30); SCN3 (0.20); SCN4 (0.10). The relatively high value of 0.4 for SCN1 reflects the fact that it is the primary scenario, which, in the deterministic case, would be the only scenario considered. The last scenario, SCN4, has the largest deviation from the BAU scenario SCN1 and requires higher capital investment across time periods. SCN4 is assigned a relatively low probability (0.10). The relatively lower probability value for SCN4 is reflected in the smaller percentage improvement in the unmet demand for the SCAPS tool set versus the BAU tool set compared to scenarios SCN2 and SCN3. Indeed, in one case (period 2) the BAU tool set does as well for SCN4 as the SCAPS tool set. On the other hand, we note

how well the SCAPS tool set does for SCN3, particularly in periods 2 and 4, where it essentially meets all of the demand.

We can quantify the improvement, pictured in Figure 2, provided by the SCAPS tool set over the BAU tool set, in meeting the forecasted demand across all of the scenarios by comparing the expected value for the unmet demand in wafers per day for the two cases as shown displayed in Table 3. The third data column shows the difference and the fourth column the additional profit per year realized by the SCAPS tool set being better able to meet the demand that may actually materialize. This amount is based upon an average profit of $2000 per wafer. The additional profit realized number is on the order of 10's of millions of dollars per year.

Period	Exp. value of unmet demand			Profit
	BAU	SCAPS	Diff	$M/year
1	102.3	74.5	27.8	20.3
2	107.6	63.6	44.0	32.1
3	152.1	101.8	50.3	36.7
4	171.1	107.7	64.0	46.7

Table 3: Expected unmet demand and additional profit realized

8 Conclusions

We have presented a mixed-integer, two-stage, stochastic programming model for capacity planning under uncertainty in semiconductor manufacturing. Due to its large size, the straightforward approach of just using a commercial solver does not work. We have used cutting planes and a heuristic approach to produce "good" solutions in a reasonable amount of time. The robustness of the tool set obtained has been shown by our analysis. The expected additional profit is in the order of 10's of millions of dollars per year. A remaining technical challenge is to improve the algorithmic component to efficiently handle a larger number of scenarios and periods. More importantly, a business challenge is to be able to produce a large number of scenarios that capture the possible tendencies of the semiconductor market.

Acknowledgments. We are grateful to Alan King and Samer Takriti for several helpful discussions. We are also grateful to the members of the Capacity, Tool, and Space Planning Group in IBM Burlington, particularly Otto Funke, manager and Scott Smith, capacity planner for their active participation and feedback, which was essential to the success of this project.

References

[1] Ahmed, S., King, A., and Parija, G. A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. Research Report RC22282, IBM, 2001.

[2] Atamturk, A., and Rajan, D. On splittable and unsplittable flow capacitated network design arc-set polyhedra. Tech. rep., Dept. of Industrial Engineering and Operations Research, University of California, Berkeley, 2000.

[3] Baudin, M., Mehrotra, V., Tillis, B., Yeaman, D., and Hughes, R. From spreadsheets to simulations: A comparison of analysis methods for IC manufacturing performance. In *IEEE/SEMI International Semiconductor Manufacturing Symposium* (1992), pp. 94–99.

[4] Bermon, S., and Hood, S. Capacity optimization planning system (CAPS). *Interfaces 29*, 5 (1999), 31–50.

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

[6] Ferguson, A. R., and Dantzig, G. B. The allocation of aircraft to routes—an example of linear programming under uncertain demand. *Management Sci. 3* (1956), 45–73.

[7] Forrest, J. personal communication.

[8] Hax, A., and Meal, H. Hierrachical integration of production planning and scheduling. In *Studies in Management Sciences, Vol. 1: Logistics* (New York, NY, 1975), Elsevier.

[9] Hood, S., Bermon, S., and Barahona, F. Capacity planning under demand uncertainty for semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing 16* (2003), 273–280.

[10] IBM Corp. *Optimization Solutions and Library.* `http://service2.boulder.ibm.com/sos/osl/`, 2001.

[11] Laguna, M. Applying robust optimization to capacity expansion of one location in telecommunications with demand uncertainty. *Management Science 44* (1998), S101–S110.

[12] Leachman, R. C. Production planning for multiresource network systems. *Naval Res. Logist. Quart. 29*, 1 (1982), 47–54.

[13] Magnanti, T. L., Mirchandani, P., and Vachani, R. The convex hull of two core capacitated network design problems. *Mathematical Programming 60* (1993), 233–250.

[14] Manne, A. S. *Investments for Capacity Expansion.* The MIT Press, Cambridge, MA, 1967.

[15] Morton, D. P., and Wood, R. K. Restricted-recourse bounds for stochastic linear programming. *Oper. Res. 47*, 6 (1999), 943–956.

[16] Occhino, T. Capacity planning model: The important inputs, formulas, and benefits. In *IEEE/SEMI International Semiconductor Manufacturing Symposium* (2000), pp. 455–458.

[17] Rajagopalan, S., Singh, M. R., and Morton, T. E. Capacity expansion and replacement in growing markets with uncertain technological breakthroughs. *Management Science 44* (1998), 12–30.

[18] Sahinidis, N. V., and Grossman, I. E. Reformulation of the multiperiod MILP model for capacity expansion of chemical processes. *Operations Research 40* (1992), S127–S144.

[19] Spence, A., and Welter, D. Capacity planning of a photolithography work cell. In *Proceedings IEEE International Conference on Robotics and Automation* (1987), pp. 702–708.

[20] Swaminathan, J. M. Tool capacity planning for semiconductor fabrication facilities under demand uncertainty. *European Journal of Operations Research 120*, 3 (2000), 545–558.

[21] Swaminathan, J. M. Tool procurement planning for wafer fabrication facilities: a scenario-based approach. *IIE Transactions 34* (2002), 145–155.

[22] Witte, J. Using static capacity modeling techniques. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop* (1996), pp. 31–35.

[23] Yang, J. An approach to determine appropriate fab development plans by taking space constraints and cost-effectiveness into consideration. In *Proceedings of the Ninth International Symposium on Semiconductor Manufacturing* (2000), pp. 217–220.