Measurement and Improvement of Manufacturing Capacity (MIMAC) Designed Experiment Report

SEMATECH
Technology Transfer 95062860A-TR

Measurement and Improvement of Manufacturing Capacity (MIMAC)
Designed Experiment Report

Technology Transfer # **95062860A-TR**
SEMATECH
July 20, 1995

Abstract: This document presents a methodology and the results of an 11-factor simulation experiment designed to measure the effects and interactions of major factors that cause a loss in fab efficiency. The dataset contained actual manufacturing data from semiconductor manufacturing facilities organized into a standard format. Each dataset contained the minimum information necessary to model a factor, including product routing and process times, rework routings, equipment availability, operator availability, and product starts. A low- and high-capacity setting were selected for the 11 modeled factors. Four factors were identified as consistently significant: downtime, yield, dispatch rule, and setup. This work was done under the Measurement and Improvement of Manufacturing Capacity (MIMAC) project and was designed to allow JESSI and SEMATECH member companies to conduct similar analyses using their own data.

Keywords: Factory Modeling, Design of Experiments, Factor Cost Analyses, Fab Capacity, Cycle Time, Planning

Authors: John Fowler, Jennifer Robinson

Approvals:
John Fowler, Author, Project Manager
Walt Trybula, Manager, Operational Modeling
Peter Lloyd, Director, MCSM
Eugene Woodall, Technical Information Transfer Team Leader

Table of Contents

List of Figures

List of Tables

1 EXECUTIVE SUMMARY

This document presents a methodology and the results of an 11-factor simulation experiment designed to identify some factors and interactions that have a major effect in the capacity planning process.

The methodology had the following four steps:

1. Dataset preparation
2. Experimental design
3. Simulation runs and analysis for each dataset
4. Summary of results across the datasets

The dataset contained actual manufacturing data from wafer fabrication facilities that were organized into a standard format. Each dataset contained the minimum information necessary to model a factor, including product routing and process times, rework routings, equipment availability, operator availability, and product starts.

Two levels, high capacity setting and low capacity setting, were selected for the 11 modeled factors. The factors and settings are shown in Table 1.

Table 1 Factor Settings

Factor	High Capacity Setting	Low Capacity Setting
Downtime	None	Dataset default
Setup	None	Dataset default
Yield	No loss	Dataset default
Batching Policy	Greedy	Force full batches
Operator Availability	Infinite number	Dataset default
Rework	None	Dataset default
Operator Cross-training	Single operator pool	Dataset operators
Dispatch Rule	Setup avoidance	FIFO
Hot Lots	None	5% of avg.WIP @ 85%
Downtime Distribution	Low variability	High variability
Downtime Frequency	Shorter/more frequent	Dataset default

Four factors were consistently identified as significant capacity loss factors. These were downtime, setup, yield, and dispatch rule. In one of the logic factories and one of the application-specific integrated circuit (ASIC) factories, downtime was the most significant factor. For the other two datasets, setup was the most significant factor. In the cases where setup was the most important factor, the bottleneck tool group was the implanter. The batching policy was very significant for one of the datasets. In addition to these factors, four interactions (setup and dispatch rule, downtime and dispatch rule, downtime and yield, and setup and downtime) were significant.

2 BACKGROUND AND METHODOLOGY

2.1 Motivation

Increasing competition in semiconductor manufacturing coupled with shorter market windows and higher costs have made designing good products insufficient to guarantee success. Manufacturing products well is also important. Unfortunately, unlike semiconductor product design, manufacturing methods do not have an established science base. The situation is particularly critical when it comes to operations. The innovations in fab operation strategies and policies have not kept pace with innovations in the design of manufacturing equipment, materials handling systems, and integrated equipment. Why semiconductor fabs operate below their theoretical capacity is not fully understood, and methods for measuring or predicting fab performance need to be improved. The effect of maintenance, batching, dispatching, setup, and yield loss are just some of the factors believed to influence manufacturing capacity.

JESSI and SEMATECH have been working together to identify and measure the effects and interactions of major factors that cause loss in fab efficiency. Once these factors are understood, methods can be developed to predict fab performance under different design and operation strategies better and to improve fab capacity. This work is being done under the Measurement and Improvement of Manufacturing Capacity (MIMAC) project.

The MIMAC project team began by compiling a list of factors believed to contribute to capacity loss in wafer fabs. They requested opinions from SEMATECH and JESSI member companies concerning these factors via an extensive survey and interview process (refer to SEMATECH Technology Transfer #94052374A-XFR, *MIMAC Survey and Interview Results*). This input helped to refine and rank the list of factors being studied. The factors studied under MIMAC include alternative equipment (equipment dedication), batching, breakdowns, dispatch/sequencing, end-of-shift effect, factory shutdown, hot lots/engineering lots, inspection/yield, level of operator cross-training, lot sizes, mix, operator availability, order release, redundant equipment, re-entrant flow, rework, setup, and time bound sequences.

The team laid the groundwork for the factor studies by defining the problem domain and terminology and summarizing the existing body of knowledge in the area (refer to SEMATECH Technology Transfer #94062424A-XFR, *MIMAC Bibliography*). They then showed the local effects of single factors by developing small models to study each factor in isolation. Sometimes these models were expanded to evaluate particular interaction effects. The results of the single factor studies are in the *Local Effect Experiment Report*, available from John Fowler at SEMATECH. Next, the team evaluated the global (factory-level) effects of the factors and their interactions via a designed experiment using a series of full factory models. The designed experiment is the subject of this report. The full factory models used were part of a testbed of real factory datasets gathered by SEMATECH and available to the public via anonymous ftp.

Several factors supported the need for a global analysis. The MIMAC team wanted to show how to identify the factors that have a major effect in the capacity planning process. They also wanted to identify significant interactions between factors. Finally, they wanted to develop and deliver a methodology that would allow people from the JESSI and SEMATECH member companies to perform similar experiments with their own data.

2.2 Methodology: Cycle Time-Constrained Capacity

Cycle time emerged as the number one performance metric for MIMAC survey and interview respondents. In response to this focus on cycle time, the MIMAC team developed a performance measure called cycle time-constrained capacity. Simply put, cycle time-constrained capacity is the maximum rate at which a system can complete work under the constraint that average cycle time does not exceed a pre-specified target. For a precise definition, denote the average throughput rate of a system by τ, and the expected average cycle time resulting from this throughput rate by $C(\tau)$. It may seem odd to consider the throughput rate of a system as an independent variable (one that can be directly controlled). However, the throughput rate is directly related to the release rate λ via a multiplication by the average line yield Y as follows:

$$\tau = \lambda Y$$

Thus, either a release rate or a throughput rate may be specified.

In any system with limited capacity to perform work, there will be some maximum throughput rate μ that the system can achieve. The MIMAC team defined cycle time-constrained capacity as the maximum throughput rate τ^* that can be achieved while limiting the mean cycle time to fall below some target value C^*,

$$\tau^* = \max_{\tau \le \mu} \{\tau : C(\tau) \le C^*\}$$

In graphical terms, this is equivalent to plotting cycle time $C(\tau)$ versus throughput τ, and drawing a horizontal line across the graph at height C^*. At the point where this horizontal line intersects the curve $C(\tau)$, a vertical line is then dropped to the x-axis. The point on the x-axis where this vertical line falls is the cycle time-constrained capacity τ^*. This is shown in Figure 1.

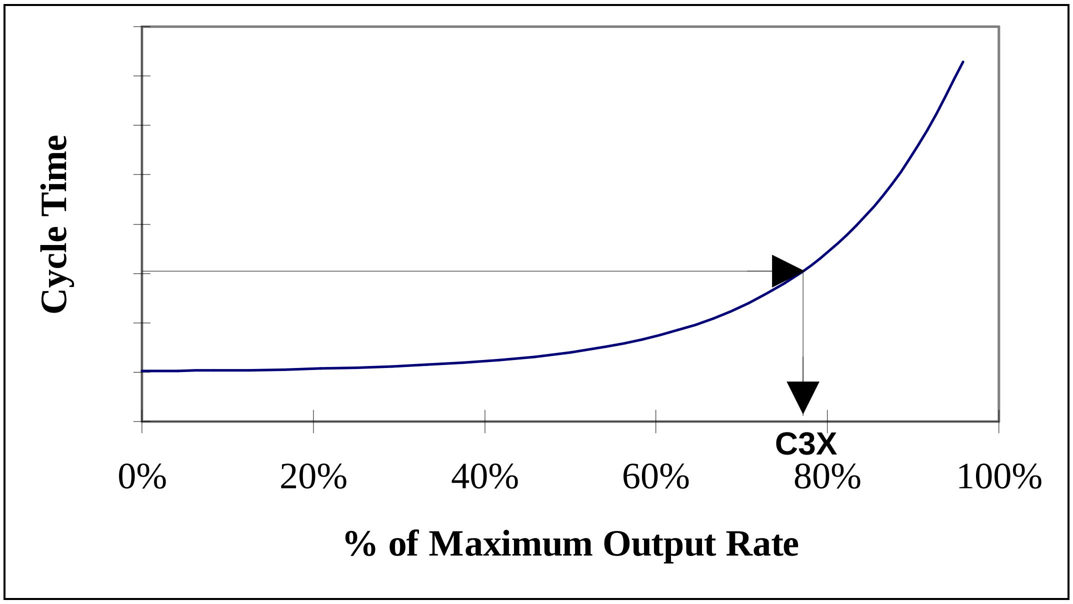

Figure 1 Cycle Time-Constrained Capacity

For systems with multiple products, the total system throughput can still be treated as an independent variable as long as a constant product mix is maintained. Denote the number of product lines by *n*, the throughput of product *j* as τ_j, the release rate of product *j* as λ_j, and the line yield of product *j* as Y_j. This leads to the relationship:

$$\tau = \sum_{j=1}^{n} \tau_j$$

$$= \sum_{j=1}^{n} \lambda_j Y_j$$

To obtain meaningful results, it is necessary to hold either the input product mix or the output product mix constant. Although holding the input mix constant is easier in practice, the MIMAC team chose to maintain the output mix because this represented demand in real factories. Under the assumption that the output product mix does not vary when changing throughput rate τ, it is possible to solve for the appropriate release rates λ_j when given a throughput value τ. To see this fact, define the output product mix *P* as a vector with each entry being the ratio of the output rate for product *j* to the total output rate,

$$P = \begin{pmatrix} \tau_1 / \tau \\ \tau_2 / \tau \\ \vdots \\ \tau_n / \tau \end{pmatrix}$$

Denote the *j*th element of *P* by p_j. Holding the output mix constant means the value p_j does not change as the throughput rate τ is varied. The following discussion shows that specifying a throughput rate τ determines the value of λ_j for all products. Suppose a new throughput rate τ' is given. Since the output mixture is constant, it is possible to solve for the new throughput rates of individual products τ',

$$\tau'_j = p_j \tau'$$

Once τ' is known, it is possible to solve for λ'_j,

$$\tau'_j = \frac{\tau'_j}{Y_j}$$

Thus in a system with multiple products and a fixed product mix, the total system throughput τ can be treated as an independent variable.

The MIMAC team used cycle time-constrained capacity as a performance measure through most of their experiments. It should be noted that the definition of capacity in terms of cycle time also has relevance for those not directly interested in cycle time through the relationship of cycle time and work in process (WIP). A mathematical relationship known as Little's Law states that at the same start rate, cycle time and WIP are directly proportional to one another. This means that factors that lead to increased cycle times (and decreased cycle time-constrained capacities) also lead to increased WIP. Details concerning the use of cycle time-constrained capacity in the MIMAC designed experiments are provided in Section 5.

2.3 Background: Design of Experiments

Experimental design provides a systematic means of investigating the effect of one or more independent variables on a dependent variable. Designing an experiment requires selecting factors of interest, choosing two or more settings (or levels) for each factor, and deciding which combinations of the factors and settings to evaluate. Each combination of factor settings is called a design point. For example, in a study with two factors, A and B, and two levels of each factor, high and low, the four possible design points are high A - high B; high A - low B; low A - high B; and low A - low B. If a researcher makes evaluations at all four design points, he or she can learn the effects of factors A and B and the interaction effect that results from A and B together. Two factors are said to have a positive interaction if one factor amplifies the impact of the other. They are said to have a negative interaction if one factor reduces the impact of the other. Experiments that explore all combinations of factor settings are called full factorial experiments. A drawback to full factorial experiments is that as the number of factors increases, the number of design points grows exponentially. To study all possible combinations of ten factors, for example, where each factor has two settings, requires evaluating $2^{10} = 1024$ design points. Design of experiments (DOE) allows researchers to extract much of the information of interest with a much smaller number of evaluations. This is accomplished by selecting certain combinations of factors from the space of possible combinations and sacrificing the ability to quantify all possible interactions between the factors. Often, higher order interactions are difficult to interpret anyway. Designed experiments have been used successfully to study many factors in semiconductor manufacturing (Hood and Welch, 1992); they are also used in semiconductor process design.

The MIMAC team used DOE to study the effect of 11 loss factors on the cycle time-constrained capacity of four testbed datasets (described in Section 3). A series of eight simulation runs was required to generate the characteristic curve for each design point. A full factorial experiment would have required 2048 design points for each of the four factories, with a total of 65,536 simulation runs. Such an experiment was beyond the computational resources of the project. Instead, a fractional factorial design was selected that allowed the MIMAC team to estimate the main effects of the 11 factors, and all two-way interactions, using 128 design points for each dataset. For more information on DOE, refer to Box, Hunter, and Hunter (1978) or Montgomery (1976).

2.4 Methodology: MIMAC Designed Experiments

The methodology used to perform the MIMAC global experiments is included in Figure 2. The four main steps were 1) dataset preparation; 2) experimental design; 3) simulation runs and analysis for each dataset; and 4) summary of results across the four datasets. These are discussed in detail in Sections 3 through 6 of this report. Section 7 contains a discussion of how people from JESSI and SEMATECH member companies might apply the MIMAC methodology to their own factories. Conclusions are presented in Section 8.

MIMAC DOE Methodology

1. Prepare datasets (including validation)
2. Determine:
   a) Factors to study (or not study)
   b) Modeling assumptions about factors
   c) Factor settings
3. For each dataset:
   a) Prepare simulation input files for all design points
   b) Make simulation runs
   c) Generate characteristic curve
   d) Compute capacities and analyze output
   e) Look for problems with data, design points, and simulation runs
   f) Make confirmation runs
4. Summarize results across all datasets

Figure 2 Global Experiment Methodology

3 DATASETS

In a separate effort, SEMATECH assembled several factory-level datasets. European data was added and validated under the MIMAC project. The purpose of collecting the datasets was to aid academics and suppliers in developing new models and tools for industry. The datasets contain actual manufacturing data from both ASIC and logic wafer fabrication facilities, organized into a standard format. They include no real product names, company names, or other nomenclature that could serve to identify the source of the data. Each dataset contains the minimum information necessary to model a factory, including product routings and processing times, rework routings, equipment availability, operator availability, and product starts. For a more detailed description of the datasets, refer to Fowler, Leachman, and Feigin (1995).[1]

Four of the six datasets (two ASIC and two logic) were selected for the MIMAC project. Characteristics of the four datasets are summarized in Table 2.

[1] The datasets are available via anonymous ftp from ftp.sematech.org, or by contacting John Fowler at SEMATECH or Ben Rodriguez at Nimble.

Table 2 Dataset Characteristics

Type of Factory	Logic1 Commodity	Logic2 Commodity	ASIC1 ASIC	ASIC2 ASIC/Pilot
# of Process Flows Modeled	2	11	21	9
Approx. WSPM	16,000	21,400	10,000	5,500
# of Tool Groups	83	73	85	104
Average Downtime per Tool	10.2%	1.8%	3.2%	13.6%
# of Operator Groups	32	Not Modeled	4	7
Is Rework Modeled?	Yes	Yes	No	No
Are Setups Modeled?	Yes	Yes	Yes	Yes
Avg. Line Yield Modeled	86%	90%	100%	93%
Avg. Steps/Mask Layer	**15**	**35**	**30**	**30**

The MIMAC project was the first real use of the testbed datasets. Under the project, the datasets were validated both internally and externally. For example, the MIMAC team uncovered and fixed several problems while preparing for and running simulation experiments. They also reviewed the simulation output with several industry experts. The experts were asked questions such as the following:

- Do the expected tools appear as bottlenecks for each factory?
- Is the downtime and yield data reasonable?
- Does the overall performance of the dataset seem realistic?
- Is the information contained in the process flows sufficiently detailed to provide interesting results?

Several problems uncovered involved converting from the testbed format to the input format required by the simulator used for MIMAC. Some were data problems. For example, some field widths specified in the testbed were not originally followed consistently. This caused difficulties when the datasets were converted to the simulation input format. Also, tools were sometimes specified as per batch tools at some steps and per wafer tools at others, something that the simulator could not handle easily. Other problems stemmed from the conversion program itself. These included errors such as not converting from minutes to hours consistently, not reading in all of the downtime information for a factory, and reading in numbers as integers that should have been real (and truncating). These problems have all been since corrected.

Several lessons learned through the data validation effort are listed in Figure 3. The validation effort required considerable time and resources. Ultimately, the result was a collection of datasets and an accompanying simulator with a functioning program to convert between them. These are currently being used at a number of universities and member companies.[2]

---

[2] For those familiar with the testbed, the datasets referred to here as Logic1, Logic2, ASIC1, and ASIC2 are called Set1, Set3, Set5, and Set6 (respectively) in the testbed.

Lessons Learned About Data Validation

1. Document everything carefully. This is especially important when several people working from different sites are using the data. Making a quick fix now, and then forgetting it at the next iteration, causes unnecessary work.
2. Look for the obvious first when there is a problem. This includes things like using days instead of weeks, lots instead of wafers, etc.
3. When the problem is not obvious, check to see if operators are constraining the system.
4. Run test simulations long enough to allow problems to arise.
5. As early as possible, ask people familiar with the real factory to review the data.

Figure 3 Lessons Learned

4 FACTORS

4.1 Factor Selection

Overall, 18 factors were targeted under the MIMAC project and are as follows:

- Alternative equipment (equipment dedication)
- Batching
- Breakdowns
- Dispatch/sequencing
- End-of-shift effect
- Factory shutdown
- Hot lots/engineering lots
- Inspection/yield
- Level of operator cross-training
- Lot sizes
- Mix
- Operator availability
- Order release
- Redundant equipment

- Re-entrant flow
- Rework
- Setup
- Time bound sequences.

Each factor is defined in the Local Effect Experiment Report. Not all of the 18 factors were studied as part of the global experiments. Time-bound sequences and tool dedication were eliminated because they were not modeled easily with the chosen simulator (Delphi). Although the simulator could have been made to handle these effects, the MIMAC team decided instead to study the factors in separate “side experiments.” Side experiments also used the factory-level data and are described as part of the Local Effect Experiment Report. Several other factors would have affected the coherency of the global experiment. These were lack of redundancy, lot size, product mix, and order release. For example, process changes typically accompany lot size changes, making lot size difficult to look at on a “level playing field” with the other factors. These four factors were also studied in side experiments. Finally, the local effect experiments and survey results had suggested that factory shutdown, end-of-shift effect, and re-entrant flow would have a less significant effect than the other factors. Therefore, they were not studied using the datasets.

Elimination of the above factors from study left the following nine original factors for the global experiment:

- Downtime
- Setup
- Yield
- Batching
- Operator availability
- Rework
- Operator cross-training
- Dispatch rule
- Hot lots

Based on the survey results, and the local effect experiments, the MIMAC team expected down-time to have a significant impact at the factory level. They decided to add two additional factors related to downtime: downtime distribution and downtime frequency. These are discussed in more detail in Section 4.3.

4.2 Modeling Assumptions About the Factors

After selecting the factors for study in the global experiments, the MIMAC team made various modeling assumptions. Some of these assumptions are listed in Figure 4. Several assumptions resulted from a lack of available data. For example, the datasets included only mean processing times. Not all of the datasets distinguished between preventive maintenance and random failures. Therefore, when the MIMAC team studied the effect of eliminating downtime from factories, they eliminated both the random failures and the preventive maintenance events. The team assumed no correlation between downtime and yield or between cycle time and yield, because they had no information with which to quantify these relationships. Similarly, no consistent data on how look-ahead information was used was available. Because no look-ahead data on the arrival of hot lots was available, machines were never held idle for imminent hot lots. The last two assumptions, which concerned hot lots and "soft" failures, were made because the simulator did not easily model preemptions. Although the simulator could have been modified to do this, the extra effort was not deemed worthwhile, particularly given that the majority of failures are actually soft failures.

- Processing times are constant.
- The time between lot releases is constant.
- No splitting of lots.
- No distinction between preventive maintenance events and random failures.
- No correlation between downtime and step yield.
- No correlation between cycle time and step yield.
- No look-ahead information is available.
- Machines are not held idle for approaching hot lots.
- Hot lots are non-preemptive, but go to the front of the queue.
- No failures while the tool is running ("soft" failures only).

Figure 4 Modeling Assumptions

4.3 Factor Settings

The nine original factors along with the two additional downtime-related factors resulted in an eleven-factor experiment. The MIMAC team selected two levels of each factor, a high capacity setting and a low capacity setting. Where possible, the low capacity settings reflected the default conditions in the datasets, while the high settings represented an "ideal" factory. In other cases, the MIMAC team selected "reasonable" good and bad cases. Their aim was to estimate the capacity loss effect of the factors on real factories. The factors and settings are shown in Table 1. Each factor is then discussed in more detail below.

4.3.1 Downtime

For the high-capacity settings, all random failures and preventive maintenance events were removed from the models. All four datasets contained downtime data, although the information provided with some datasets was more detailed than with others. Across the datasets, the downtime percentage at individual tools ranged from 0% to 40%.

4.3.2 Setup

All four datasets contained setup information for two to five tools (typically implanters and steppers). For the high-capacity setting, all setups were removed from the models (that is, setup times were set to zero). Delphi and the datasets modeled up to two group-dependent setups and one product-dependent setup per process step. This allowed modeling major and minor setups at implant workstations and mask changes at steppers. Sequence dependent setups, however, were not modeled.

4.3.3 Yield

Three of the four datasets included line yield. Average line yields across all products for the three datasets ranged from 86% to 93%. The high-capacity setting assumed no yield loss for each dataset. Because the line yields were not the same for each product, the high capacity setting resulted in an output mix different from the original output mix. The MIMAC team decided to hold mix constant between the two yield settings by using the output mix for the yield case and the product mix (both input and output) in the no yield case.

4.3.4 Batching Policy

Batching policy does not necessarily have a high and a low capacity setting, because the impact of the policy depends on how busy each tool is. The greedy policy usually results in significantly lower cycle times for lightly loaded tools, and results in cycle times similar to those obtained by using a full batch policy at heavily loaded tools. Because it tends to do better under a wide range of start rates, the greedy policy was selected here as the high capacity setting.

Forcing full batches resulted in a significant capacity loss for dataset ASIC1. This 21-product factory had many batch ID's that could not be processed together. Some products comprised very small proportions of the overall mix. At low to moderate start rates, under a full batch policy, these low volume products sometimes waited literally months to make a full batch. As a result, the cycle time increased with lower start rates. This is illustrated in Figure 5, which shows the characteristic curve for the factory under greedy and full batch policies. The weighted average cycle time never went below three times the raw process time for the factory under the full batch policy. Since the factor was obviously significant and made it impossible to detect the effects of the other factors for dataset ASIC1, the batching factor was removed from the analysis for this dataset. Batching policy is discussed in more detail in the Local Effect Experiment Report.

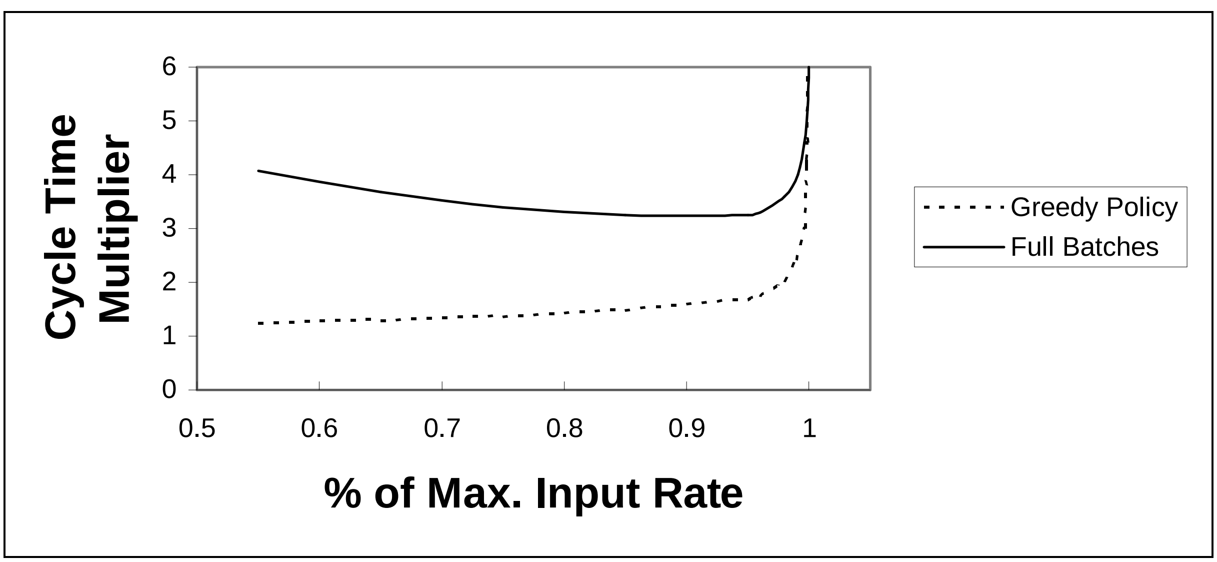

Figure 5 Impact of Batch Decision Policy on Cycle Time of Dataset ASIC1

4.3.5 Operator Availability

Three of the four datasets contained information about operator availability. Operators did not constrain the theoretical capacity of any of these datasets. However, the MIMAC team felt that they might have an impact on cycle time-constrained capacity. The default numbers of operators were therefore used as the low-capacity settings. The high-capacity settings assumed an infinite supply of operators for each dataset. A separate experiment in which the number of operators was reduced is described in the Local Effect Experiment Report.

4.3.6 Rework

Only two of the datasets contained rework sequences (the two logic factories), and neither included a high percentage of reworked lots. A high-capacity setting of no rework for any of the datasets was selected for the experiments.

4.3.7 Operator Cross-Training

Of the three datasets with operators, one had four operator groups (or work areas), one had seven groups, and one had 32. The MIMAC team believed that capacity might increase if all of the operators were fully cross-trained. Therefore, the high capacity setting was selected to be a single operator pool for each dataset. A separate study with an intermediate level of cross-training for one dataset (Logic1) is included in the Local Effect Experiment Report.

4.3.8 Dispatch Rule

No consensus has been reported on what makes up a high capacity versus low-capacity dispatch rule. However, for tools with significant setups, a setup avoidance policy usually results in higher capacities than a first in, first out (FIFO) policy. Therefore, a setup avoidance policy was used as the high capacity setting for the global experiments. Under a setup avoidance policy, when a machine completes processing a lot the operator checks the queue for lots with matching setup ID's. Only if no lots with matching ID's are available will the machine be set up for a different setup ID. Other dispatch rules were studied in a separate experiment and reported in the Local Effect Experiment Report.

4.3.9 Hot Lots

None of the datasets contained information about hot lots. Here the default (no hot lots) was used for the high capacity setting. For the low capacity setting, a constant number of hot lots was used for all runs of each dataset. This number was computed by taking 5% of the average WIP for a run of the factory at 85% of its theoretical maximum (unconstrained) capacity. This resulted in thirteen, twelve, eleven, and eight hot lots for datasets Logic1, Logic2, ASIC1, and ASIC2 (respectively).

4.3.10 Downtime Distribution

To simulate random failures, sampling from some distribution for time between failures and time to repair is necessary. The testbed datasets included only information about mean time between failures (MTBF) and mean time to repair (MTTR), so the actual distributions used constitute modeling assumptions.

Local effect experiments had shown that at the tool level, more variable downtime distributions can lead to lower cycle time-constrained capacity. To see if this had an impact at the factory level, the MIMAC team selected a high variability distribution as the low capacity setting (exponential), and a low variability distribution (triangular) as the high capacity setting. The coefficients of variation of the two distributions were 1.0 and 0.4 respectively.

4.3.11 Downtime Frequency

The local effect experiments had also shown that longer, less frequent failures lead to more system variability than shorter, more frequent failures (at the same overall availability percentage). To see whether this had an effect at the factory level, the MIMAC team used the dataset defaults for the low capacity setting. They then cut both the MTBF and the MTTR in half (maintaining the availability percentage) for the high-capacity setting.

5 SIMULATION AND ANALYSIS

5.1 Prepare Simulation Input Files

As described above, the DOE required 128 design points for each dataset. Editing simulation input files by hand for each design point would have been a tedious and error-prone process. Instead, the MIMAC team wrote UNIX shell scripts that could generate the input files for each design point. For example, scripts were written to create a new set of input files with no downtime, rework, or setup, using a greedy batching policy, etc. The scripts are available as a deliverable from the MIMAC project.

5.2 Make Simulation Runs

All MIMAC global experiments were performed using Delphi, a discrete-event simulator modified under the MIMAC project. Delphi models the data described in the testbed datasets, including breakdowns, setups, rework, scrap, and operators. It also includes a set of queuing formulas that approximate workstation utilizations before each simulation run. The formulas include rework, yield, setups, and equipment downtime, though they do not include operators. Delphi was chosen for MIMAC because of its speed (running on UNIX workstations), and because the team had access to the source code to make modifications during the project. Each simulation was run for a very long time (five years, with a one-year warm-up) to obtain long run capacity estimates.

5.3 Generate Characteristic Curves

The MIMAC method for estimating cycle time-constrained capacities is based on two intuitions about manufacturing cycle times. The first is that for any manufacturing system that is not completely deterministic, cycle times will grow larger and larger as the system throughput approaches capacity. Expressed symbolically, $C(\tau) \rightarrow \infty$ as $\tau \rightarrow \mu$ (where τ is the system throughput rate, and μ is the maximum system throughput). When there is more variability in a system, the rapid cycle time increase appears at a lower start rate. This is shown in Figure 6, which shows the characteristic curve for Dataset Logic1 under three different levels of variability. In the low variability system, lot release times, processing times, MTBFs, and MTTRs were all treated as constants. In the medium variability system, MTBFs and MTTRs were exponentially distributed. The high variability system had additional variability in the processing times and the times between lot releases.

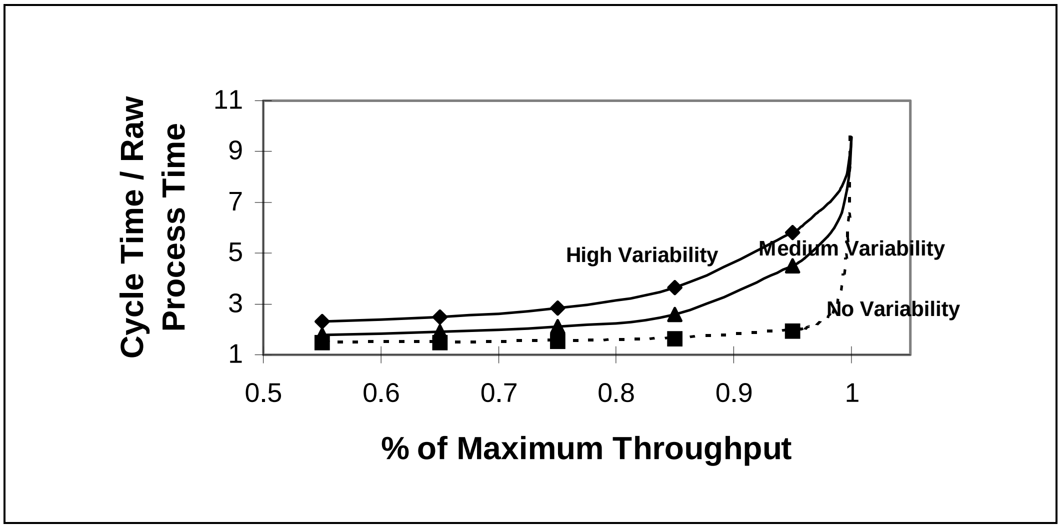

Figure 6 Impact of Increasing Variability in Dataset Logic1

The second intuition about cycle times is that below a certain value, small changes in throughput rate τ will usually result in small changes in cycle time $C(\tau)$ and that the function $C(\tau)$ is relatively smooth. Suppose τ_0 and τ_1 are similar (i.e., the difference τ_1 - τ_0 is small). Given the prior intuitions, it is probably not too great an error to approximate the true cycle-time function $C(\tau)$ for τ between τ_0 and τ_1 by a straight line connecting the points $\hat{C}(\tau_0)$ and $\hat{C}(\tau_1)$. Note that as the difference τ_1 - τ_0 decreases, the error in this linear approximation also decreases.

Continuing, choose a third throughput rate τ_2, that exceeds τ_1 by a small amount, and approximate the true function $C(\tau)$ for τ between τ_1 and τ_2 by a straight line connecting $\hat{C}(\tau_1)$ and $\hat{C}(\tau_2)$. Continuing this process, the result is a piece-wise linear approximation to the true function. Given a target cycle time C^*, the cycle time-constrained capacity τ^* may be approximated by first searching for a pair of throughput rates τ_j and τ_{J+1} with $C(\tau_j) \leq C^* \leq C(\tau_{j+1})$. Then, assuming a straight line approximation between $C(\tau_j)$ and $C(\tau_{j+1})$, linear interpolation can be used to find τ^*. This is equivalent to finding a desirable cycle time on the y-axis of the characteristic curve, then drawing a line across to the curve down to the x-axis to find the corresponding output rate.

Two complications can arise while using this method. First, suppose the simulation estimates $\hat{C}(\tau_0)$, ..., $\ddot{C}(\tau_n)$ are available, but $\ddot{C}(\tau_n) < \hat{C}*$. Finding a pair of simulation estimates bounding $\hat{C}*$ will not be possible. In that case, the MIMAC team found that it works well to approximate the right-hand tail of $C(\tau)$ by the function

$$T(\tau) = \alpha_0 + \alpha_1 \left(\frac{1}{\mu - \tau} \right)^2$$

Choose α_0 and α_1 so that $T(\tau)$ matches the right end of the estimated cycle-time function and its first derivative $T'(\tau)$ matches the slope of the right-most linear segment. Once the function $T(\tau)$ has been derived, it can be solved for τ^*, where τ^* satisfies

$$T(\tau^*) = C^*$$

A second complication arises if the true cycle-time function $C(\tau)$ is not monotonic; i.e., it does not always increase as τ increases. This situation can occur if there are minimum batch sizes of greater than one component specified in the system. In this case, there may be a region of relatively low throughput rates where the cycle time actually decreases as the throughput rate increases, since less time is spent waiting to form batches. See Figure 3.1 for an example of this phenomenon. If the capacity estimation code specifically searches for pairs of bounding points with $C(\tau_j) \leq C(\tau_{j+1})$, and the throughput rates are listed in increasing order, $\tau_0 \ldots \tau_n$, then there is no risk that the algorithm will find one of the artificially low capacities on the left-hand side of the U-shaped cycle-time curve. Also, when a U-shaped cycle-time curve occurs, it may happen that no pair of simulation estimates bound the target cycle time C^* (C^* may fall below the bottom of the U). In that case, the algorithm should notify the user of the situation. When this happened on one of the MIMAC datasets (set ASIC2), the team simply chose a different target cycle time that was feasible for all design points.

The MIMAC team believes that the benefits of this 'brute force' method outweigh the errors caused by the piece-wise linear approximation. In particular, the method is easy to program, is easy to understand, and performs very reliably. In their experience, it works best to simulate between 7 and 10 different throughput rates τ_j, with a concentration of points between 75% and 95% of capacity (μ). At lower capacities the points do not need to be as close together since the cycle-time curve will probably be relatively flat. As part of its capacity analysis, Delphi calculates an approximate value for μ. The MIMAC team chose to make runs at 35%, 45%, 55%, 65%, 75%, 80%, 85%, 90%, and 95% of μ.

Another advantage of the brute force method is that all of these simulation runs can be made in parallel when a network of machines is available. For the MIMAC experiments, simulations were run on a network of workstations. A final advantage of the method arises if, after the simulation estimates have been generated, one wishes to change the target cycle time C^* and search for a new cycle time-constrained capacity τ^*. With this method, it is not necessary to make any more simulation runs — simply proceed with the search process outlined above, using the new C^*.

If the simulation estimates $\hat{C}(\tau_j)$ have a high variance, then the resulting estimated cycle-time curve will not be a good approximation to the true curve. For the MIMAC experiments, very long simulation runs were made to decrease the variance of the estimates $C(\tau_j)$. To check the effect of variability, the entire experimental design was repeated with different random number seeds. The outputs of the analysis were quite similar, indicating that the effect of variability had been successfully minimized.

5.4 Compute Capacities

The MIMAC team used the method described above to generate characteristic curves for each design point for each dataset. The queuing formulas in Delphi were particularly useful for determining the maximum theoretical capacity μ of a factory. For a given product mix, the code finds the maximum achievable throughput rate (that will drive the bottleneck utilization to 1.0). The MIMAC team calls this the infinite X capacity. This infinite X capacity is difficult to find using simulation alone, because as cycle time and WIP levels grow very large, run length and memory requirements grow cumbersome. The numbers provided by the queuing formulas were validated for the datasets by making simulation runs just above and just below the infinite X capacities. The runs just below were stable, while for those just above, a queue would continue to grow at the bottleneck.

After the characteristic curves had been generated, the 2X (twice theoretical capacity) and 4X capacities were obtained by interpolating from the characteristic curve (as described in the equations above). The curve generation and capacity computation were automated using Splus (a statistics package) scripts. Section 5 describes the analysis of the capacity results.

5.5 Look for Problems

Once the output for each dataset was obtained, the MIMAC team reviewed it to look for potential problems. In some cases, problems were uncovered concerning the data, design points, or simulation runs. These sometimes required making revisions, and repeating the experiment. For example, initial test experiments (using a smaller experiment with fewer factors) showed an occasional positive effect of yield loss on cycle time-constrained capacity. This occurred when the yield loss happened to be greater for products with longer theoretical cycle times. When the longer cycle-time products were scrapped, processing more of the shorter cycle-time products was possible, thus increasing the cycle time-constrained capacity. The problem was eliminated by modifying the experiments to hold output mix constant (as described in Section 4).

Initial experiments also resulted in a negative effect of the greedy batching policy for one of the datasets (indicating that the full batch policy resulted in higher capacities). Further analysis showed that the effect was due to the strict implementation of the greedy policy in Delphi. As originally modeled, the first lot in the queue at a batch machine was always processed, even when it was the only lot with a particular batch ID. This resulted in a capacity loss when full batches of other batch ID's were forced to wait. Delphi was modified to allow checking the queue first for full batches of available product, even under a greedy policy. With this change, the negative batching effect disappeared.

5.6 Make Confirmation Runs

The capacity analysis resulted in some effects that were not immediately intuitive. For example, dataset Logic1 showed a small interaction effect of yield and setup. After some discussion, the MIMAC team surmised that this effect might be due to a three-way interaction with hot lots. When a hot lot is scrapped, the next lot started into the system is a hot lot of the same type (to replace the lost lot). As the hot lot passes through the factory, it incurs more setups than a regular lot and leads to additional capacity loss. This hypothesis was tested by building a small simulation model to do a full factorial experiment on yield loss, hot lots, and setup. In this experiment, the interaction of yield and setup was only apparent for the cases with hot lots present. This supported the conjecture. Confirmation experiments were also run to understand other interactions, as described in Section 6.3.

6 GLOBAL EXPERIMENT RESULTS

6.1 Individual Dataset Results

The 2X, 4X, and infinite X capacity numbers for all design points were obtained from the simulation and queuing model output. A regression was then done to estimate the impact of the different factors on the dataset capacity. The next four charts (Figures 7 through 10) show the relative ranking of the effects for each dataset. The top eight factors are shown for 2X, 4X, and infinite X constraints.

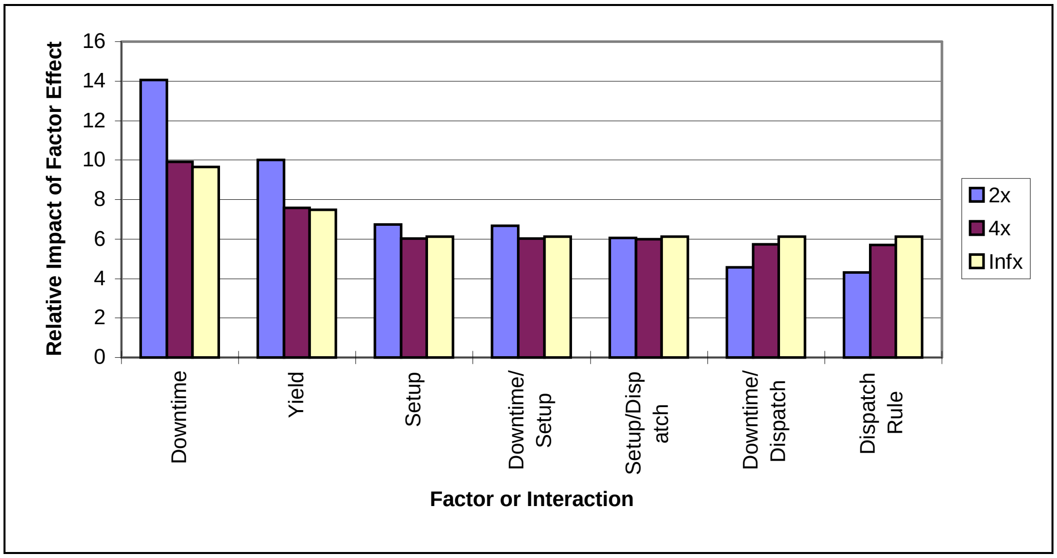

Figure 7 Relative Ranking of Factor Effects from Global Experiments Using Dataset Logic1

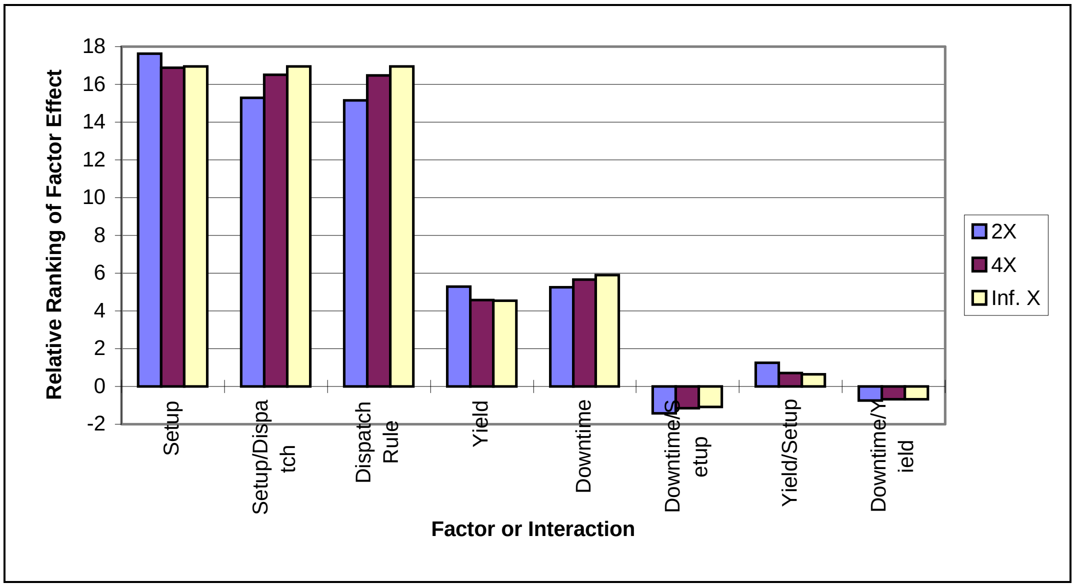

Figure 8 Relative Ranking of Factor Effects from Global Experiments Using Dataset Logic2

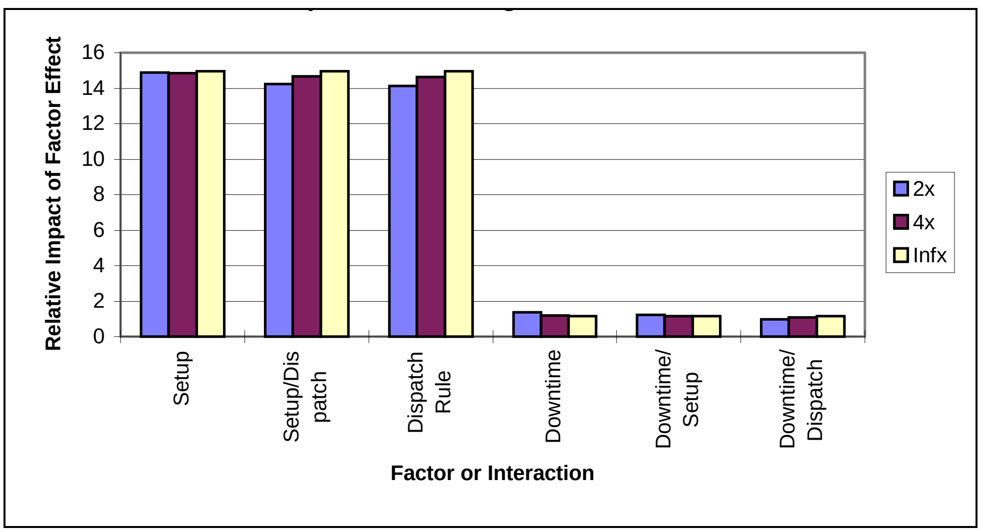

Figure 9 Relative Ranking of Factor Effects from Global Experiments Using Dataset ASIC1

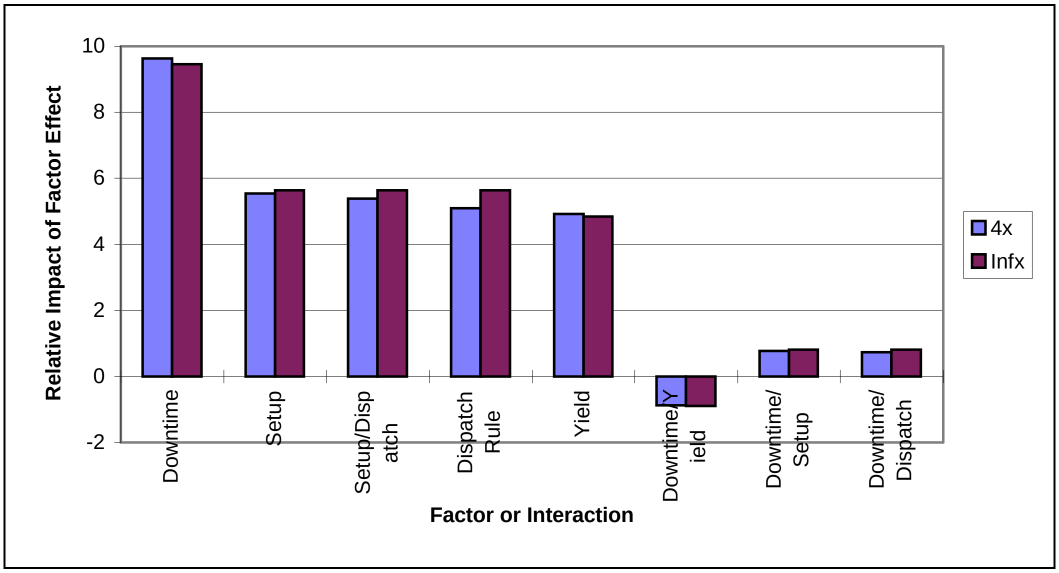

Figure 10 Relative Ranking of Factor Effects from Global Experiments Using Dataset ASIC2

Each bar in the above figures shows the percentage difference between the average across all simulation runs where the factor was at its high setting, and the average across all runs where the factor was at its low setting. For example, Figure 7 shows that under a 2X cycle-time constraint, the average capacity of all runs with no downtime is 14% higher than the average of all runs with downtime. Two factors listed together, for example downtime/setup, indicate an interaction effect. The presence of a downtime/setup interaction implies an additional capacity loss from having downtime and setup together, so that the total capacity loss is more than the sum of the individual losses for downtime and setup. Because of the resolution of the experiment, however, the interaction bars are not additive. Referring again to Figure 7, under a 2X constraint, downtime is attributed a 14% loss in capacity and a 7% loss in setup. The bar for the setup/downtime interaction does not show an additional 7% loss, but rather that the interaction is significant. Interactions are discussed further in Section 6.3.

Some additional caveats are necessary. The batching factor was removed from the analysis of dataset ASIC1 (refer to Section 4.3). Therefore, although batching significantly affected the cycle time-constrained capacity for this dataset, it does not appear in Figure 9. The second ASIC factory also had trouble at low start rates when full batches were used. Batching was still included in that analysis. However, recording 2X results was not possible, because for some design points the cycle time never reached as low as 2X. Therefore, Figure 10 shows only 4X and infinite X results. Some other factors did not appear significant for particular factories because they were not included in the datasets at all. For example, yield is not significant for dataset ASIC1, because no yield data was included. Similarly, not all datasets included operators or rework (refer to Table 2).

6.2 Summary Across the Datasets

Looking across all the datasets, four factors were consistently significant: downtime, yield, dispatch rule, and setup. Batching was also significant for dataset ASIC1, but it only had a slight effect on the other three datasets. Though not shown in the figures, distribution of downtimes was mildly significant for dataset Logic1 (ranked tenth among significant factors). No other factors appeared as significant for the datasets in this analysis. The following interactions were each significant for more than one dataset:

- Setup and dispatch rule had a positive interaction (additional capacity loss) for all four datasets.
- Downtime and dispatch rule had a slight to moderate positive interaction for all four datasets.
- Downtime and yield had a slight negative interaction (less capacity loss when both are present) for three of the datasets.
- Setup and downtime evidenced both positive and negative interaction effects.

Looking at the results in terms of cycle-time constraint, batching had very little effect when no cycle-time constraint was imposed, but had more effect under a tight cycle-time constraint. Downtime had a larger relative impact in the infinite X case because other factors like batching had less effect. The setup avoidance dispatch rule had the most significant positive effect in the infinite X case and less effect under a tighter cycle-time constraint. The latter can be explained by the fact that setup avoidance rules, while reducing time spent in setup, increase the cycle time of some of the lots.

6.3 Explanations

The appearance of downtime and yield as two of the most significant loss factors seems straightforward. Even in a non-cycle time-constrained environment, time spent in repairs and time spent processing lots that are later scrapped reduce the capacity of a factory. The results for setup and dispatch rule are related to their interaction. The dispatch rule studied was a setup avoidance policy. For all four datasets, the results show that when a FIFO rule alone is followed for a factory that experiences setups, a significant capacity loss results. Under a setup avoidance rule, setups cause a much smaller capacity loss.

The interaction of setup and downtime appears as both a positive and a negative effect. The MIMAC team did additional local effect experiments (described in the Local Effect Experiment Report) to understand this phenomenon. Essentially, when a machine goes down for repairs, a queue often builds up in front of the machine. Under a setup avoidance policy, when the machine comes back up the probability of finding lots with matching setup ID's in the queue will be high. Fewer setups will be done, thus “making up” for some capacity loss due to the failure. This also explains the dispatch rule/downtime interaction. However, downtime and setup can lead to additional capacity loss when they occur together by disrupting dedication schemes. Often machines can be dedicated to particular setup ID's. However, when one machine goes down, another may be forced to change over, leading to additional setup time and further decreased capacity.

The negative interaction of downtime and yield can be attributed to a “double-counting” of the capacity loss. When a machine goes down, it has less time available for processing and, therefore, fewer opportunities to cause yield loss.

The fact that none of the other factors appeared as significant in these experiments should not be taken to mean that they might not be significant in other contexts. First, the modeling assumptions made under MIMAC may have influenced some of the results. For example, hot lots were not as hot as they might have been. Machines were not held idle, or preempted, for hot lots.

Minimum batch sizes were still enforced, even when hot lots were present. Hot lots might have had a more significant effect if modeled differently. Also, the factor levels in the MIMAC DOE were not normalized. It is not surprising that removing all downtime from the models had more effect than merely changing downtime distribution or downtime frequency. The results were meant to show, at a qualitative level, the capacity lost due to some factors present in a few representative datasets. Different factors might have significance for other factories.

7 FOLLOWING THE MIMAC METHODOLOGY AT A MEMBER COMPANY

The primary goal of the MIMAC team in conducting their global experiments was not to find out exactly how much capacity was lost due to each factor in the testbed datasets. Rather, the goals were to identify some of the factors and interactions that have a major effect in the capacity planning process, and to develop a methodology. The MIMAC team hopes that people from the JESSI and SEMATECH member companies can use this methodology to conduct similar analyses using their own data. Data availability and some experience in using simulation are the primary requirements. The MIMAC results suggest that downtime, yield, setup, and dispatch rule might be worth investigating. However, for a particular factory, other issues like product mix or lot size might be of more pressing interest. In the remainder of this section, suggestions are made for adapting the MIMAC methodology to a member company factory.

7.1 Prepare Data

Data collection and validation are never trivial tasks. The importance of accurate data in modeling for capacity planning cannot be overemphasized. Even the most carefully designed and executed experiment can fall victim to the phenomenon of “garbage-in-garbage-out.” Data gathering will probably be less labor-intensive, and perhaps more accurate, if data can be collected automatically from a shop floor control system. Similarly, if people familiar with the operation of the factory help to validate the data, errors are likely to be reduced.

7.2 Determine Factors, Assumptions, and Settings

An 11-factor experiment requires many simulation runs. People doing a DOE of their own factory probably already have some idea of which factors might be important and which are not of interest. This information should help in setting up reasonable-sized experiments. Modeling assumptions should be discussed with people from the fab, if possible. This can help ensure the relevance of the assumptions selected, and the eventual acceptance of the results. Meanwhile, selecting factor settings will probably be an iterative process. Early in the study, best and worst case factor levels such as those used in the MIMAC study may make sense. This can help to identify areas of maximum potential benefit. Later, narrower factor ranges can be used to study specific scenarios. Even for small experiments, using DOE to select design points can significantly reduce the number of simulation runs required.[3]

[3] Refer to Box, Hunter, and Hunter (1978) or Montgomery (1976) for more information on DOE.

7.3 Make Simulation Runs and Analyze Results

While Delphi is available as a deliverable from the MIMAC project, any simulator could be used to generate the characteristic curve for a factory. The availability of queuing formulas makes it easier to find the infinite X capacity than otherwise. However, the maximum capacity can be estimated by incrementally increasing the input rate into a simulation model until the cycle time and WIP become very large. Someone with simulation experience should be consulted at this stage to help with decisions about run length and model warm-up.

The UNIX scripts provided by the MIMAC team can be used to analyze cycle time-constrained capacity numbers for experiments. For smaller experiments, however, a spreadsheet can be used nearly as easily to graph characteristic curves and linearly interpolate to find capacity numbers. Some statistical expertise will be necessary to determine significant effects from the capacity results.

7.4 Summarize Results

Summarizing results across datasets is hardly necessary for someone conducting an analysis for a single factory. However, it might be worthwhile for a company with several different fabs to attempt some standardized set of experiments across factories.

8 CONCLUSIONS

The MIMAC global experiments have highlighted several factors that may be important in capacity planning. Different factors might be important for specific factories. A high-level DOE could be used by people at the JESSI and SEMATECH member companies to decide which loss factors should be the focus of detailed capacity planning and which can be approximated. Experiments of this type require significant amounts of computer and human resources. The MIMAC team hopes that the methodology and lessons learned identified in this report will reduce false starts among JESSI and SEMATECH members who perform similar experiments. The use of the characteristic curve to evaluate cycle time-constrained capacity can eliminate problems in making comparisons across different scenarios. Even when a full-blown designed experiment is not needed, the characteristic curve can be used by itself to obtain more information about a factory.

9 REFERENCES

G.E.P. Box, W.G. Hunter, and J.S. Hunter, *Statistics for Experimenters. An Introduction to Design, Data Analysis, and Model Building*. New York: Wiley, 1978.

F. Chance, Delphi: *A Manufacturing Analysis and Simulation Tool*. Available from Chance Industrial Solutions, chance@chances.com.

F. Chance, J.W. Fowler, and J.K. Robinson, *Semiconductor Manufacturing Testbed: The Delphi Simulator*. Available from John Fowler at SEMATECH.

J.W. Fowler, R.C. Leachman, and G. Feigin, *Semiconductor Manufacturing Testbed: Testbed Dataset*s. Available from John Fowler at SEMATECH.

J.W. Fowler and J.K Robinson, *Measurement and Improvement of Manufacturing Capacity (MIMAC) Project Bibliography*, SEMATECH Technology Transfer #94062424A-XFR, 1994.

S.J. Hood and P.D. Welch, "Experimental Design Issues in Simulation with Examples from Semiconductor Manufacturing," *Proceedings of the 1992 Winter Simulation Conference*, eds. J.J. Swain, D. Goldsman, R.C. Crain, and J.R. Wilson, 255-263, 1992.

D.C. Montgomery, *Design and Analysis of Experiments.* New York: Wiley, 1976.

E. Neacy, S. Brown, R. McKiddie, *Measurement and Improvement of Manufacturing Capacity (MIMAC) Survey and Interview Results*, SEMATECH Confidential Technology Transfer #94052374A-XFR.

Local Effect Experiment Report distributed by John Fowler, SEMATECH.

10 ADDITIONAL INFORMATION

For additional information and other MIMAC documents, contact:

SEMATECH

John Fowler
SEMATECH
2706 Montopolis Drive
Austin, TX 78741
U.S.A.
Phone:+1(512)356 3755
Fax: +1(512)356 3083
Email: john.fowler@sematech.org

JESSI

Ben Rodriguez
Nimble
Maaltecenter Blok G
Derbystraat 313
B-9051 Gent, Belgium
Phone: +32(9)222 39 18
Fax: +32(9)222 96 90
Email: nvnimble@mcimail.com

SEMATECH Technology Transfer
2706 Montopolis Drive
Austin, TX 78741

http://www.sematech.org