Fast Accurate Simulation of Physical Flows in Demand Networks

Karl Kempf
Decision Technologies
Intel Corporation
5000 W. Chandler Blvd.
Chandler, AZ 85226
karl.g.kempf@intel.com

Kraig Knutson, John Fowler,
Benjamin Armbruster,
Praveen Babu, Brett Duarte
College of Engineering and Applied Science
Arizona State University
Tempe, AZ 85287-5906

KEYWORDS
Supply Chain, Control, Factory, Warehouse, Transport

ABSTRACT
More efficient and effective control of supply chains (more appropriately called demand networks) is conservatively worth billions of dollars to the world economy. Developing improved control polices requires simulation of the physical, financial, decision, and data flows involved. This paper describes our initial work on modeling and simulating the physical flows. We show the level of abstraction that is appropriate, formulate and test a general model at this level, and show minor specializations to incorporate particular features of factories, warehouses, and transportation links.

INTRODUCTION
Improving the operational efficiency of demand networks (also known as supply chains) for physical goods is one way to boost the national and global economy. Understanding the control physics of this kind of demand networks is a key to providing continuous improvement of their efficiency. Given the inherent complexity of demand networks that involve production, developing such an understanding requires extensive experimentation to formulate and validate control theories. However, this type of demand network generates enormous wealth so direct experimentation would involve untenable financial risk. Therefore some form of modeling and simulation is required.

Simulating demand networks requires the modeling of a number of quite different but interrelated flows including physical, financial, decision, and data. The physical flow represents the goods being produced, stored, and shipped while the financial flow concerns the payments for the materials and services required and the products supplied. The data flow represents data about the past, present, and the forecast future state of the physical and financial flows. The decision flow uses the data available about the physical flow, advice from the financial system, and its decision policies to provide direction for the physical system. The overall goal is to maximize the four customer service factors - the right product, in the right quantity, in the right place, at the right time - while minimizing the four major costs - materials, production, storage, transport.

We have developed a software architecture to model and simulate these flows and the interactions between them as shown in Figure 1. Although a variety of software modules can be connected in this way with multiple modules of like or similar functionality, our current implementation includes just one of each type and utilizes commercial products.

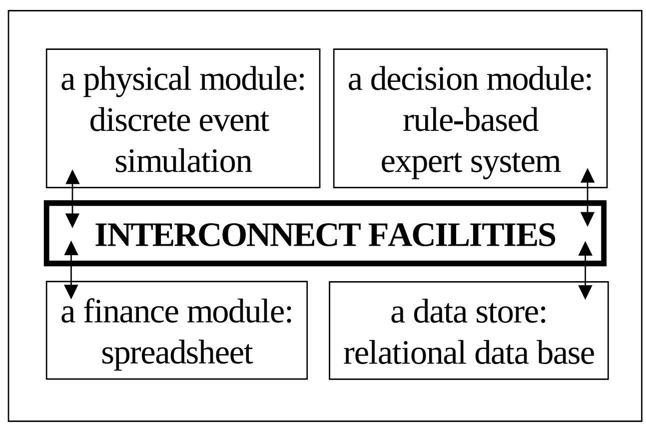

Figure 1. An architecture for demand network simulation

Using this architecture a wide variety of experiments can be imagined for exploring the behavior of demand networks. From the physical perspective, the number and connectivity of entities can be varied as well as the capacity and throughput time of each entity. With the financial module, calculations can be done at the end of a simulation for evaluating the overall run and at the beginning of or during a simulation to support decision-making. Using the decision module, various mathematical and heuristic control policies can be investigated to quantify their impact on the business. The advantage of one central controller versus many local controllers can be quantified by using one or multiple decision and financial modules. The utility of performing financial calculations and decision-making more or less frequently can be explored. The data that can flow between modules is the basis for their interaction, as is the associated database. Experiments can be imagined that test the advantage of having precise data immediately versus inaccurate stale data.

THE BASIC APPROACH

The focus of this paper is the modeling and simulation of the physical flow that carries raw materials, production work in progress (WIP), and finished goods. The entities involved in the flow include factories, warehouses, and transportation links. In the example shown in Figure 2, there are a number of suppliers of materials to the core company, the factories and warehouses owned by the core company, a number of types of downstream customers of the core company, and the transportation infrastructure.

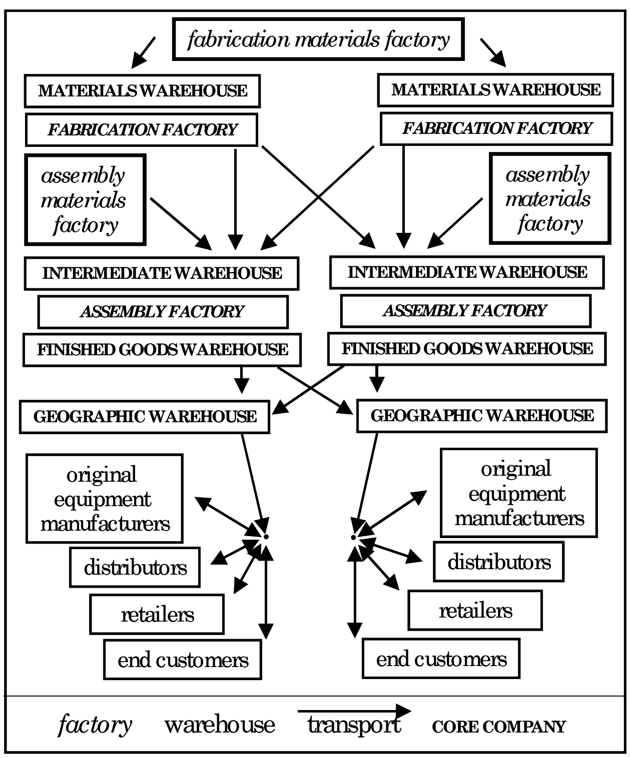

Figure 2. An example demand network physic flow

As with any modeling and simulation exercise, finding the appropriate level of abstraction is necessary. That level of abstraction will provide solutions that are accurate enough for the problem being studied in as short a time as the current state of computer technology will allow. One approach that is not appropriate here is tool level discrete event simulation. The answers are very accurate, but in a demand network with multiple entities in the physical flow, run times would be prohibitive. Another approach would be the assignment of a few static parameters to model the dynamic performance of each entity. This produces answers very quickly, but answers that are not accurate enough for our purposes.

The key insight leading to the appropriate abstraction is that all of the entities in the physical flow of a demand network are capacitated. This means that as they are loaded nearer and nearer to their capacity, the cycle time required for them to perform their function rises nonlinearly and their throughput and cycle time performance becomes increasingly stochastic as shown in Figure 3. Each entity in Figure 2 behaves in this way although each has a different capacity, throughput variability, cycle time, and cycle time variability. In addition, each may be operated at a different point on its characteristic curve depending upon whether speed (on the lower left of Figure 3) or volume (on the upper right of Figure 3) is the focus.

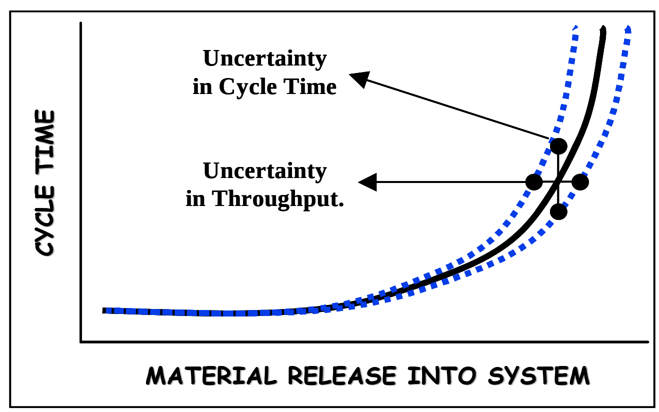

Figure 3. The performance of all demand network entities

THE RESULTING IMPLEMENTATION

This abstraction leads to the core module that we use to model and simulate physical flow in demand networks. This core module contains two components.

The first sub-module deals with capacity. At the beginning of each time period, some amount of capacity becomes available. If fewer units arrive during the time period than can be processed by the available capacity, all of the units are passed to the second sub-module and the remaining capacity is lost. If more units arrive during the time period than can be processed, units equal to the capacity are passed to the second sub-module, the capacity is exhausted, and the remaining units stand in queue until enough capacity becomes available to process them. Since capacity is variable, this sub-module contains a distribution from which a capacity is randomly drawn at the beginning of each time period. This distribution is skewed to the low capacity side since there are usually more ways to temporarily lose capacity than there are to gain it.

The second sub-module deals with cycle time. Again a distribution is used to represent variability, this time skewed to the high cycle time side since there are usually more ways to slow a unit down that there are to speed it up. For each unit that is passed from the first sub-module, a cycle time is randomly drawn from the distribution and assigned. After being held for the appropriate time, units emerge from our core module.

Notice that this abstraction and its implementation span the approaches initially considered and rejected. If one of our modules is used to represent each tool in a demand network entity, then a tool level discrete event simulation results. If one of our modules is used to represent a whole demand network entity and the capacity and cycle time distributions are collapsed to single numbers, then the approach relying on a few static parameters results. Our module with its distributions in tact supplies results approaching the accuracy of tool level discrete event simulation in run times approaching the use of a few static parameters. We will demonstrate these claims in the next three sections.

SINGLE MODULE RESULTS

The data shown here are for a single module with a discrete triangular distribution for capacity (min= 75.0, most probable= 150.0, mean= 142.5, max= 195.0, in units/day) and a continuous triangular distribution for cycle time (min= 21.25, most probable= 22.50, mean= 25.00, max= 31.25, in days). Figure 4 shows the relationship between throughput and the loading of the system relative to its maximum capacity. The separation of the 3 lines at lower loadings shows the effect of the capacity distribution. The nonlinear separation of the lines at higher loadings shows the impact of units queuing in front of the capacity sub-module.

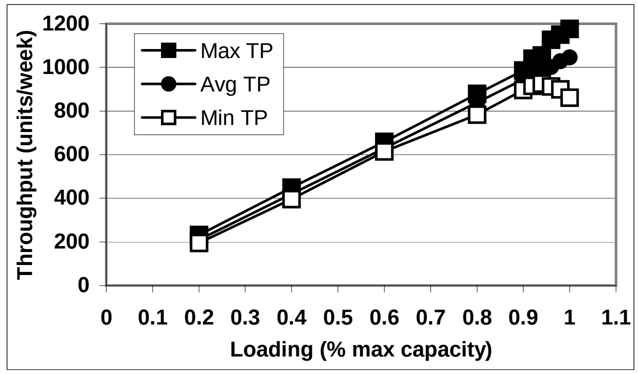

Figure 4. Single Module Throughput versus Loading

The relationship between cycle time and loading is shown in Figure 5. The separation of the three lines shows the effect of the cycle time distribution, and the nonlinear separation of the lines at higher loadings shows the impact of units queuing in front of the capacity sub-module. The separation of the lines at the lower loadings is a reflection of drawing less number from the cycle time distribution.

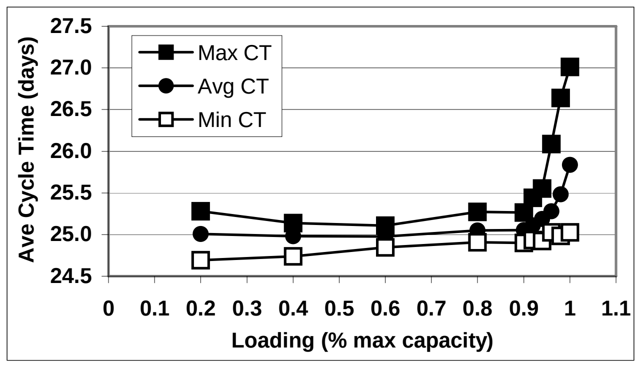

Figure 5. Single Module Ave. Cycle Time versus Loading

The variability of throughput and cycle time over time is shown in Figures 6 and 7 respectively for 94% loading.

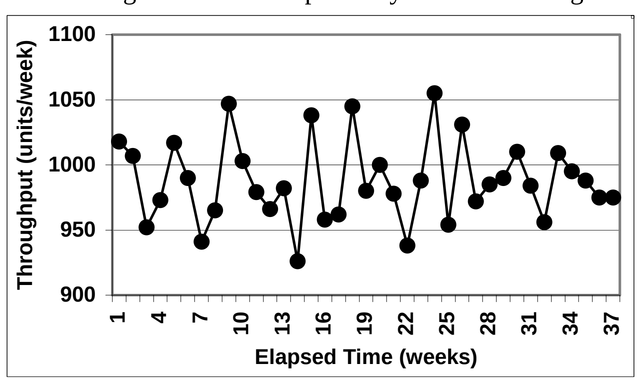

Figure 6. Single Module Throughput versus Time

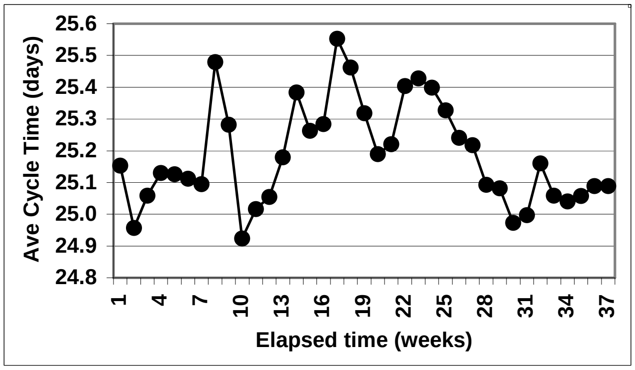

Figure 7. Single Module Ave. Cycle Time versus Time

All of these plots are consistent with the behavior required by Figure 3, and are qualitatively in alignment with data from actual fabrication and assembly factories.

MULTIPLE MODULE RESULTS

The data shown here are for a simple demand network composed of 7 of our modules including, in material flow sequence, a fabrication factory (2 modules, one for fabrication, one for final test), a transportation link, an intermediate warehouse, an assembly factory (2 modules, one for assembly, one for final test), and a finished goods warehouse. The capacity and cycle time distribution parameters are listed in Table 1.

Table 1 – Capacity and Cycle Time Parameters

	min	most prob	mean	max
CAPACITY				
fabrication	75	150	142.5	195
test-1	100	200	190	260
transport	125	250	237.5	325
inter. w-house	125	250	237.5	325
assembly	105	210	199.5	273
test2	80	160	152	208
f-goods w-house	125	250	237.5	325
CYCLE TIME				
fabrication	42.5	45	50	62.5
test1	4.25	4.5	5	6.25
transport	2.55	2.7	3	3.75
inter. w-house	1.7	1.8	2	2.5
assembly	4.25	4.5	5	6.25
test-2	1.7	1.8	2	2.5
f-goods w-house	0.85	0.9	1	1.25

The relationship between throughput and cycle time versus loading is shown in Figures 8 and 9 respectively. The line separations now show the effects of 7 sequential capacity distributions, 7 sequential cycle time distributions, and sequential queuing in front of 7 capacity sub-modules.

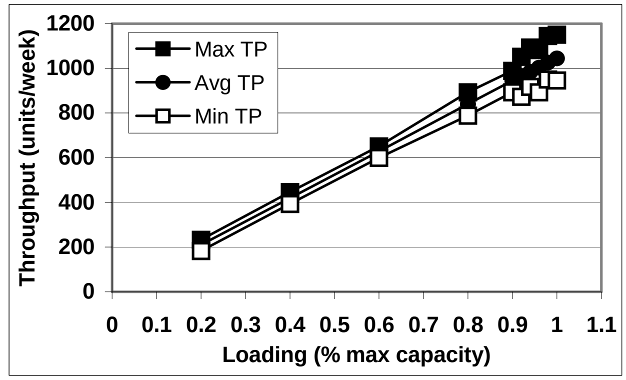

Figure 8. Multiple Modules Throughput versus Loading

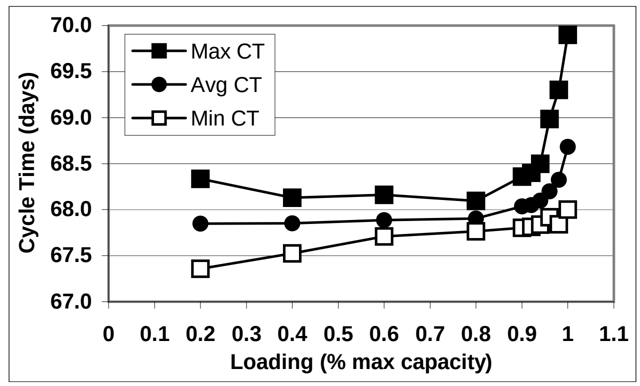

Figure 9. Multiple Modules Ave. Cycle Time vs. Loading

The variability of throughput and cycle time over time is shown in Figures 10 and 11 respectively for 94% loading.

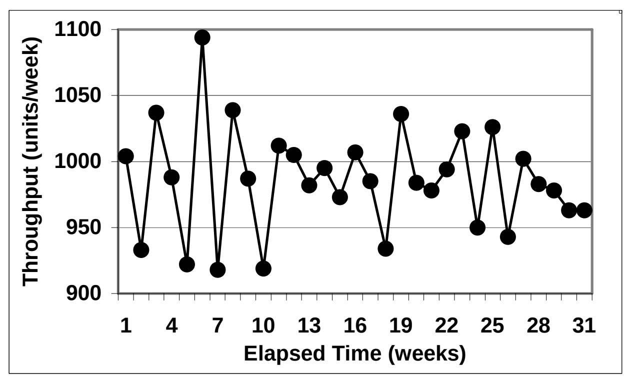

Figure 10. Multiple Modules Throughput versus Time

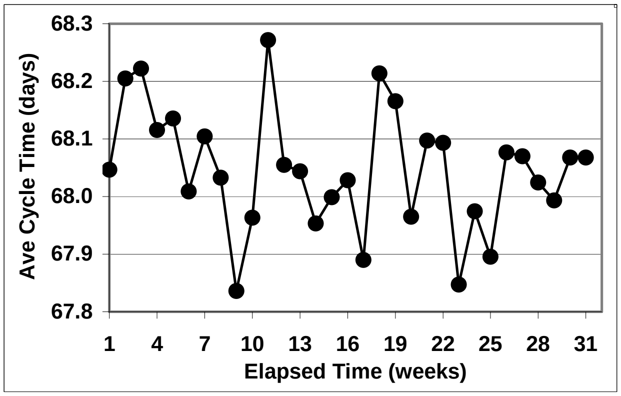

Figure 11. Multiple Modules Ave. Cycle Time versus Time

Rerunning the experiments with the distributions widened generates a more detailed demonstration of the impact of the capacity and cycle time distributions on the 7-module demand network. Specifically the difference between the means and the max and min values of each of the 14 triangular distributions were doubled. Figure 12 shows the impact on throughput at the high loading levels that demand networks often experience. The widened distributions result in higher variability in throughput at each loading as well as higher variability between loadings. Similar results are obtained for cycle times as shown in Figure 13.

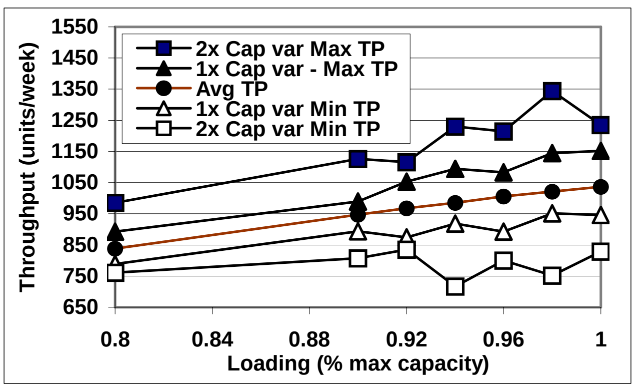

Figure 12. Throughput at Levels of Capacity Variability

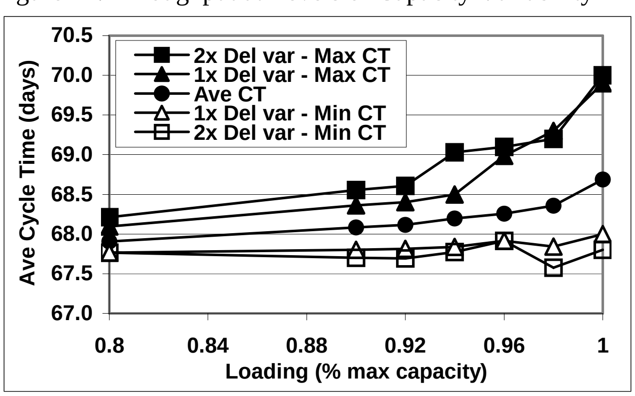

Figure 13. Cycle Time at Levels of Cycle Time Variability

TIMING RESULTS

Execution times were collected from experiments conducted on a computer running Windows 98™ on a 400 MHz Pentium III™ with 4 MB of primary memory. The module used in the experiments had a capacity distribution with a min of 40, a mean of 50, and a max of 60 units per day, and a cycle time distribution with a min of 16, a mean of 20, and a max of 24 days. Chains were built using from 1 to 7 of this

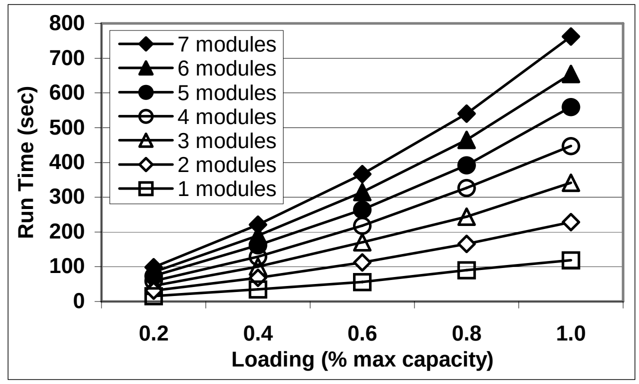

Figure 14. Run Time with Number of Modules and Loading

module and run at various loadings for 420 days of simulated time. The execution times are shown in Figure 14. The data contained in Figures 8 through 14 demonstrate that high quality results can be generated for a demand network of moderate complexity running for over a year near its maximum capacity in less than 15 minutes on a computer of modest performance.

ENHANCEMENTS FOR FACTORIES

While the generic module described above can be parameterized to represent factories, warehouses, and transportation links, each of these three entities does exhibit special characteristics. For example, Figure 15 shows the partial autocorrelation of cycle time data that we observe for actual factories and tool level discrete event simulations of factories. While our generic module does not exhibit this characteristic, this is easily remedied. To generate a series of cycle times with a predetermined autocorrelation we use the fact that a random walk has a partial autocorrelation of lag 1. If Z is an independently distributed random variable with mean $\mu\,(1-\alpha)$ and standard deviation $\sigma(1-\alpha^2)^{1/2}$, then the sequence X_n generated by $X_n=\alpha X_{n-1}+Z_n$ has a mean μ, a standard deviation σ, and a partial autocorrelation α. We implemented this rule in our cycle time sub-module. The results are given in Figure 16 and show a qualitatively correct autocorrelation in our factory cycle times.

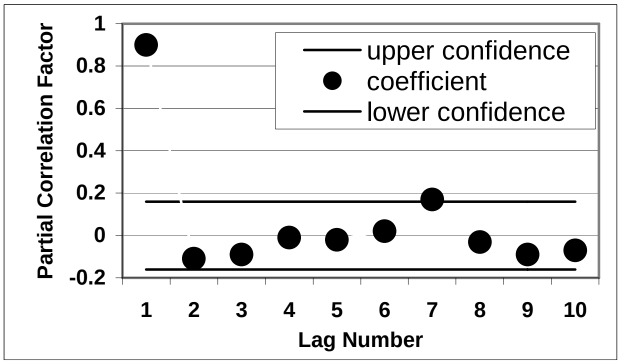

Figure 15. Cycle Time Autocorrelation in Actual Factories

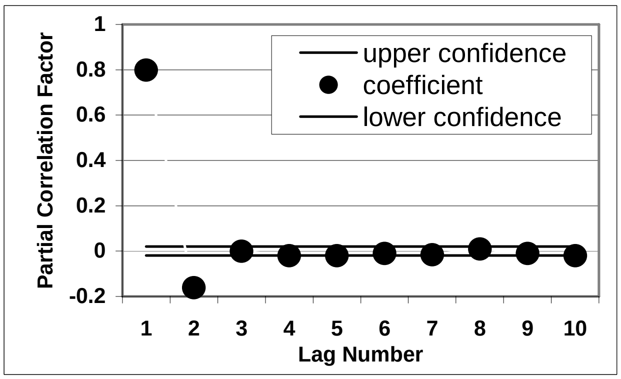

Figure 16. Cycle Time Autocorrelation in a Factory Module

In addition, actual factories have yield losses. To incorporate this feature, we simply added an addition sub-module with a random number generator and an overall factory yield target. Each unit leaving the cycle time sub-module enters this yield sub-module to determine whether it is scrap or leaves the factory as product. Figure 17 shows the throughput of the 7-module demand network without any yield loss against the throughput of the same system with an 80% yield sub-module at the exit of the test-1 factory after fabrication and a 95% yield sub-module at the exit of the test-2 factory after assembly. The cycle time of this system is little altered since the bulk of the time is spent in the fabrication module and the yield losses are suffered after the production units have left this module.

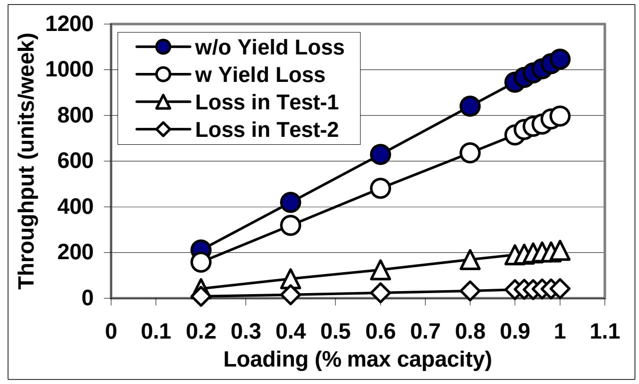

Figure 18. Factory Yield Losses

Finally, it is possible to alter the cycle time performance of actual factories by re-prioritizing the work in progress. We can realize this effect in our factory module by dynamically shifting cycle time distributions. A single factory module was run for 2 products with identical continuous triangular distribution for cycle time (min= 21.25, most probable= 22.50, mean= 25.00, max= 31.25, in days). In the 10th week of simulated operation, all of the distribution parameters of product A were shifted to be 5 days longer and all of the distribution parameters of product B were shifted to be 5 days shorter. The results of dynamic re-prioritization are shown in Figure 19.

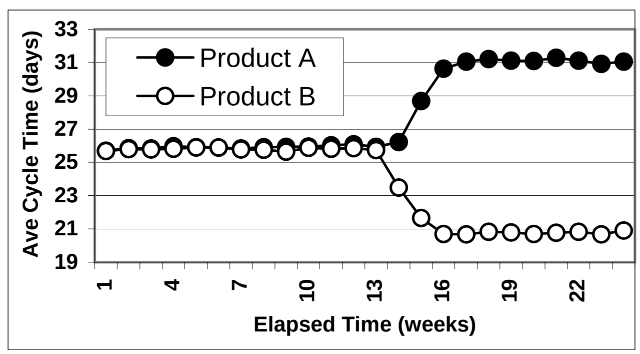

Figure 19. Factory Work in Progress Prioritization

TRANSPORT AND WAREHOUSE ENHANCEMENTS

Specializing our generic module to the individual needs of transportation links and warehouses is the focus of our current research.

Transportation links are like factories in that once work is released into them, they are usually expected to process the work as soon as possible. Their capacities vary periodically, but they differ in the pattern in which their capacities are available. In a factory, the capacity is available throughout the day or until it is exhausted. In a transportation link, once the truck, boat, or airplane is loaded, it departs and no capacity may be available until the next transporter arrives.

Warehouses differ from factories and transport links in that they are designed to hold units, not process them as soon as possible. Warehouses have a capacity to move units in and out and a capacity to hold units, and these two capacities are not necessarily related. We have thus far used our capacity and cycle time modules to model getting units out of warehouses and are considering whether this is adequate.

CONCLUSIONS AND FUTURE WORK

We have shown that a simple set of modules can quickly and accurately provide performance data for demand networks constructed from nonlinear stochastic entities including factories, transportation links, and warehouses. We have shown that the basic module can be simply extended to include the particular features of factories and believe that the same will be true for transportation links and warehouses.

Our future direction is to develop techniques to parameterize the capacity, cycle time, and yield distributions for our modules to statistically match the performance of actual demand network components. We are actively collecting performance data from factories, transportation links, and warehouses to support this effort.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation through grant DMI-0075655 for “A Modular, Scalable Approach to Modeling and Analysis of Semiconductor Manufacturing Supply Chains” from the Scalable Enterprise Systems Initiative.