Roser, Christoph, Masaru Nakano, and Minoru Tanaka. “Buffer Allocation Model Based on a Single Simulation.” In Winter Simulation Conference, edited by S. Chick, Paul J Sanchez, David Ferrin, and Douglas J Morrice, 1238–46. New Orleans, Louisiana, USA, 2003.

BUFFER ALLOCATION MODEL BASED ON A SINGLE SIMULATION

Christoph Roser
Masaru Nakano
Minoru Tanaka

TOYOTA Central Research and Development Laboratories
Nagakute, Aichi 480-1192 JAPAN

ABSTRACT

Allocating buffers in manufacturing systems is one of easiest ways to improve the throughput of the system, as changes can be implemented quickly and the initial cost of the change is low. Yet, while an increase in the buffer size usually increases the throughput, it often also increases the work in progress and the makespan, therefore increasing the inventory and the time to the customer. Subsequently, the trade off between the throughput, the work in progress, and the makespan are of significant research interest. This paper describes a general prediction model of these performance measures for different buffer size increases based on only a single simulation. A fully automated implementation of the simulation analysis and prediction model for manufacturing systems of any size and complexity is available. The method can be used for flow shops, job shops, and serial or parallel systems.

1 INTRODUCTION

The optimization of manufacturing systems and other discrete events systems is one of the most important and most researched subjects in discrete event simulation (Boesel et al. 2001; Fu et al. 2000). Subsequently, there is a large body of research in the area of discrete event optimization (Azadivar 1999; Swisher et al. 2000). One of the easiest ways to improve a manufacturing system is to adjust the buffer allocation, as the initial cost of adding or removing buffers is usually only a fraction of the cost for adding processing machines or changing the system layout.

Buffers allow a better utilization of the bottleneck machines by reducing the idle time (starved and blocked) thereof. In particular, buffers serve two purposes: They reduce the starving time of machines by providing additional parts and the blocking time of machines by providing additional free spaces, thus improving the throughput of the system. However, this improvement comes at a cost. Besides the cost of providing the buffer spaces, there will be an increased number of parts in the system, i.e. the work in progress (WIP) is increased, creating additional cost for the inventory. Even more significant, the makespan increases, and the system responds slower to production changes and customer orders, reducing the ability to produce *Just In Time* (JIT). Therefore there has to be a trade off between a small WIP & makespan (i.e. small buffers) and a fast production rate (i.e. large buffers). An excellent discussion of the effect of buffers can be found by Conway et al. (1988) and others (Brittan 1996; Caramanis, Pan, and Anli 2001).

There is a large body of research related to buffer allocation. Most of the methods are based on building a metamodel requiring numerous repetitions, for example by using simulated annealing and genetic algorithms (Spinellis and Papadopoulos 1999a; Spinellis and Papadopoulos 2000a; Spinellis and Papadopoulos 2000b), neural networks (Altiparmak, Dengiz, and Bulgak 2002), gradient based searches (Gershwin and Schor 2000; Levantesi, Matta, and Tolio 2001; Schor 1995), or tabu searches (Shi and Men 2002). However, in industry it is usually difficult to obtain the large number of replications needed to implement the model, and the use of these methods is inefficient. Other approaches are based on a functional approximation and evaluation (Enginarlar, Li, and Meerkov 2001; Enginarlar et al. 2002) and knowledge based methods (Vouros and Papadopoulos 1998), or combinations of analytical and simulation based methods (Nakano and Ohno 2000).

This paper focuses on the area of buffer allocation by creating a prediction model to estimate the effect of additional buffer capacity onto the system performance using only a single simulation. The presented model has the advantages that the approach is fully automated and therefore easy to use, and that the method is based on only a single simulation, therefore allowing the modeling of complex systems without the need of a large number of repetitions. This method works for large systems, balanced and unbalanced systems, and serial and parallel manufacturing systems and can be adapted to nonmanufacturing discrete event systems. The method is based in part on the shifting bottleneck detection method (Roser, Nakano, and Tanaka 2002a; Roser, Nakano, and Tanaka 2003).

2 SIMULATION EXAMPLE

The presented method will be demonstrated using a complex simulation example, consisting of 7 machines M1 to M7 and 2 different part types in a branched system as shown in Figure 1. The first machine M1 is never starved, and the last machine M7 is never blocked. Nine different buffer locations are considered, and buffers of capacity 1 have been added to BM3, AM4, BM5, and AM5. While the machine cycle times are constant, the machines are randomly delayed by exponential distributed failure times. The simulation time was 500 days, using the TOPQ simulation software (Kubota, Sato, and Nakano 1999; Nakano et al. 1994). The system is well balanced with M3 being the main bottleneck, and M2, M5 and M6 being secondary bottlenecks. The system completed one part every 55.7s, or an average of 64.7 parts per hour.

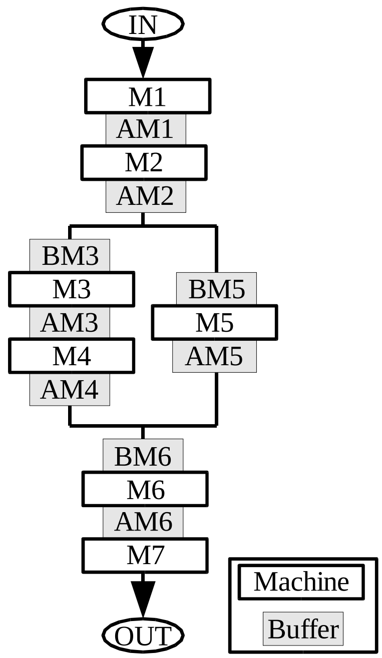

Figure 1: Example Layout

3 CAUSES OF STARVING AND BLOCKING SITUATIONS

Buffers improve the system throughput by reducing the idle time (blocking and starving) of the machines. Therefore, to understand the buffers it is crucial to understand the blocking and starving of the machines, the causes thereof, and, most important the path to the causes and the buffer locations in between. This method analyzes every starving or blocking occurrence of every machine in the simulation, and finds the cause of the starving and blocking, and, more important, the buffer locations on the path between the idle machine and the cause thereof. The time a possible buffer location is part of a path is determined for each machine.

There are also four possible modes how a buffer can affect another machine as shown in Table 1. A buffer can provide either additional parts or spaces. Usually, parts are given to starved machines downstream (Mode I), and spaces are provided to blocked machines upstream (Mode IV). However, a buffer may also relieve a blocked machine indirectly by providing parts to another machine (Mode II), or relieve a starved machine indirectly by providing spaces to another machine (Mode III). For example, in Figure 1, machine M2 is providing parts to both machines M3 and M5. In some cases, machine M3 is starved for parts. This may be due to the fact that M2 cannot deliver parts, because M2 is blocked by M5. Therefore, adding free spaces to the buffer BM5 will reduce the blocking of M2 and therefore the starving of M3 (Mode III).

Table 1: Effect Modes of Buffers on Machines

Effect	Machine:	Machine:

Modes	Starved	Blocked
Buffer: Provide Parts	I	II
Buffer: Provide Spaces	III	IV

The analysis of the causes of starving and blocking has to distinguish between these four modes. The percentage of the time a buffer j has an effect on a starved machine i is named $\%E^{I}_{i,j}$ and $\%E^{III}_{i,j}$ for modes I and III and the percentage of the time a buffer j has an effect on a blocked machine i is named $\%E^{II}_{i,j}$ and $\%E^{IV}_{i,j}$ for modes II and IV respectively. The effects can range from 100% (for example an adjacent downstream buffer is always in the path between the blocked machine and the cause thereof) to 0%, where the buffer is never in the path between the idle machine and the cause thereof.

3.1 Analysis Logic

To find the cause of an idleness of a machine, an algorithm has been developed that follows the cause from machine to machine or buffer until the cause of the idle period has been found. While depending on the detail of the available data, there may some ambiguity, the following set of rules provide a good estimate for the search of the cause of an idle period. In this algorithm, it is also assumed that the loading time of parts to and from a machine is negligible and that a buffer is always between two machines and two machines only (the branched system in the example uses a transfer machine to realize the branches).

A machine is always either active (A), blocked (B) or starved (S). The definition of active includes not only working machines, but also machines under repair or performing a tool change. For blocked machines, there are a total of 5 possible situations, determining the next machine/buffer in the search for the cause of the blocked machine. An overview of the situations is given in Figure 2, and the 5 cases are listed below.

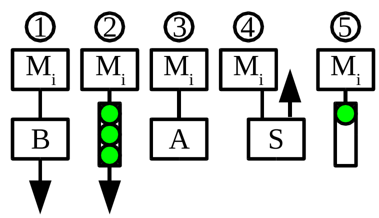

Figure 2: Situations with Blocking of Machine M_i

1. If the downstream machine is blocked, continue the search with the downstream machine.
2. If the downstream buffer is full, continue the search with the downstream buffer.
3. If the downstream machine is active, the machine is the cause of the block. Stop the search.
4. If the downstream machine is starved, continue the search with the downstream machine by looking upstream for the cause of the starve of the downstream machine.
5. If the downstream buffer is not full, the buffer is the cause of the block due to insufficient speed. Stop the search.

The numbers in the figure and the list also represents the ranking if more than one situation is possible due to more than one downstream machine/buffer. Always follow the situation with the lowest number. For example, if a blocked machine is followed by both a full buffer (Case 2) and a not full buffer (Case 5), the block is most likely caused by the full buffer (Case 2) or a downstream machine thereof.

For starved machines, there are also a total of 5 possible situations, determining the next machine/buffer in the search for the cause of the starved machine. An overview of the situations is given in Figure 3, and the 5 cases are listed below.

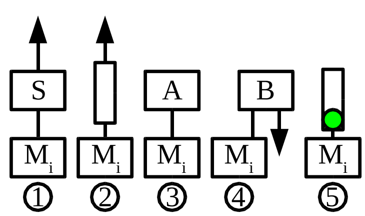

Figure 3: Situations with Starving of Machine M_i

1. If the upstream machine is starved, continue the search with the upstream machine.
2. If the upstream buffer is empty, continue the search with the upstream buffer.
3. If the upstream machine is active, the machine is the cause of the starve. Stop the search.
4. If the upstream machine is blocked, continue the search with the upstream machine by looking downstream for the cause of the block of the upstream machine.
5. If the upstream buffer is not empty, the buffer is the cause of the starve due to insufficient speed. Stop the search.

Again, the numbers in the figure and the list also represents the ranking if more than one situation is possible due to more than one upstream machine/buffer. Always follow the situation with the lowest number. For example, if a starved machine is preceded by both an empty buffer (Case 2) and a not empty buffer (Case 5), the starve is most likely caused by the empty buffer (Case 2) or an upstream machine thereof. Using this set of rules, it is possible to find the cause of an idle period for all idle periods of all machines, and to determine the period of time a buffer location was part of the path to the cause of an idle machine. This allows the conclusion of the effect of a buffer onto the different machines.

3.2 Analysis Results

The presented example has been analyzed, and the causes of the blocking and starving of the machines has been established. Figure 4 presents the results for machines M3 and M5 in graphical form, showing the path of the starves (cross-hatched) and blocks (diagonal-hatched) from machine M3 and machine M5 to the machine causing the starve or block. The width of the path represents the fraction of the starves/blocks following this path.

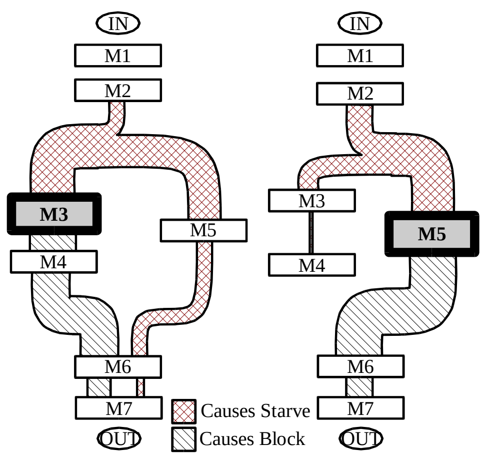

Figure 4: Causes of Blocking and Starving of Machines M3 and M5

For example, machine M3 is blocked 6.1% of the time. Whenever machine M3 is blocked, the path to the cause of the block leads to the next downstream machine M4 (100% of the blocked time). However, M4 itself is rarely the cause of the block. Most the paths continue to machine M6 (78% of the blocked time), and M7 (46% of the blocked time). Therefore, a buffer increase before machine M7 affects the blocking of machine M3 46% of the time. Machine M3 is also starved for 5.8% of the time. The path to the cause of the starve splits, with 38% of the starving periods caused by machine M2, and 62% following to machine M5. From machine M5 the paths continue to M6 (27% of the starving time), and from there to M7 (15% of the starving time).

The causes of the starving and blocking of machine M5 can be traced similarly, with the path to the cause of the blocks continue to machine M6 (99% of the blocked time) and M7 (57% of the blocked time). The path to the cause of the starves splits towards M2 (55% of the starved time), and M3 (40% of the starved time), continuing to M4 (7% of the starved time). The path to the causes of the blocked and starved periods has to be analyzed for all machines to estimate the effect of buffers.

3.3 Discussion of the Effect of Buffers

The path between the idle machines and the cause thereof allows an estimation of the effect of buffers. Only buffers in these path affect the machines. Furthermore, there are different modes in which a buffer can affect a machine as discussed in Table 1. For example in Figure 4, if the buffer before machine M3 can provide parts, the starving of M3 is reduced (Mode I). At the same time, if the buffer can provide additional spaces, the starving of machine M5 is also reduced (Mode III). The buffer before machine M6 has an especially interesting effect, as it not only reduces the blocking of machine M3 by providing spaces (Mode IV), but also reduces starving on the very same machine M3 by providing spaces to machine M5 (Mode III).

4 SINGLE SIMULATION PREDICTION MODEL

This section predicts the change in the system performance based on an increase in the buffer capacity of one or more buffers.

4.1 Expected number of Parts in a Buffer

The same buffer can have different effects depending on the number of parts and the number of spaces provided to the machines in the system. Therefore, the first step is to estimate the mean number of parts in a buffer, and subsequently the mean number of additional parts and the mean number of additional free spaces if a buffer is increased. There are a number of methods available in the literature, most of them based on a decomposition approach (Bouhchouch, Frein, and Dallery 1993; Dallery and Frein 1989; Spinellis and Papadopoulos 1999b). This paper uses a estimation of the mean number of parts based on the shifting bottleneck detection approach (Roser, Nakano, and Tanaka 2002a; Roser, Nakano, and Tanaka 2003), but the reader may choose any suitable method of his/her choice, as long as the additional number of parts $\Delta B^P{}_j$ and free spaces $\Delta B^S{}_j$ can be estimated based on the change in the buffer size ΔB_j of buffer j.

4.2 Additional Parts and Spaces for Each Machine

The next step estimates the number of additional parts $\Delta M^P{}_i$ available in front of machine i to reduce starving and the additional number of spaces $\Delta M^S{}_i$ available after machine i to reduce blocking. This estimation is based on the additional number of parts $\Delta B^P{}_j$ and free spaces $\Delta B^S{}_j$ available in all buffers j, and the effect of the buffer into the machines for the four modes $\%E^I{}_{i,j}$, $\%E^{II}{}_{i,j}$, $\%E^{III}{}_{i,j}$ and $\%E^{IV}{}_{i,j}$ as shown in equations (1) and (2).

$$\Delta M_i^P = \sum_{j=1}^{n} \Delta B_j^P \cdot \%E_{i,j}^I + \sum_{j=1}^{n} \Delta B_j^S \cdot \%E_{i,j}^{III} \qquad (1)$$

$$\Delta M_i^S = \sum_{j=1}^{n} \Delta B_j^P \cdot \%E_{i,j}^{II} + \sum_{j=1}^{n} \Delta B_j^S \cdot \%E_{i,j}^{IV} \qquad (2)$$

4.3 Possible Reduction in the Time per Part for Each Machine

After estimating the number of additional parts and free spaces available for each machine, the possible reduction in the time per part of the machines can be estimated. Each additional part available in front of the machine allows the machine to work longer by avoiding starving periods. Similarly, each available free space after the machine allows the machine to work longer by avoiding blocking periods. The maximum additional time that can be worked depends on the additional number of parts $\Delta M^P{}_i$, spaces $\Delta M^S{}_i$, and the mean cycle time $C^T{}_i$ needed to produce one part. For example, if there would be one additional part $\Delta M^P{}_3$ available in front of machine M3, then machine M3 with a cycle time $C^T{}_3$ of 140s could avoid starving periods up to 140s completely and reduce all remaining starving periods by 140s.

The mean time that can be reduced therefore depends on the distribution of the starving and blocking times of the machines, and the probability density function of the starving time distribution $pdfM^S{}_i(t)$ and the probability density function of the blocking time distribution $pdfM^B{}_i(t)$ are needed to estimate the reduction in the idle times of the machines. Figure 5 shows the cumulative density function of the idle time distributions of selected machines as measured in the example. As the example includes both deterministic events and random events, the resulting idle time distributions are a combination of deterministic and random distributions as can be seen from the deterministic steps in the otherwise random distribution.

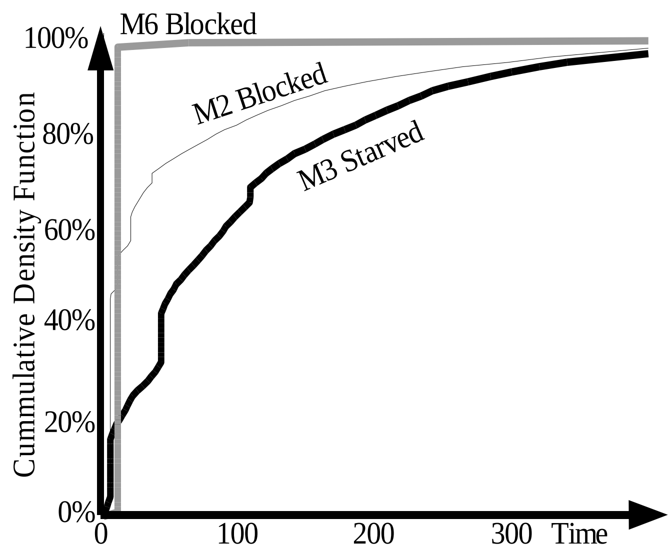

Figure 5: Selected Idle Time Distributions

The mean reduced idle time can be calculated by integrating the probability density functions $pdfM^S_i(t)$and $pdfM^B_i(t)$ multiplied by the time t between the time 0 and the upper limit defined by the cycle time C^T_i and the additional number of parts ΔM^P_i or spaces ΔM^S_i. The mean waiting time of the entire distribution can be calculated by setting the upper limit of the integral to infinite. The ratio of these two integrals is the percentage reduction of the waiting time. Combining this percentage reduction with the percent of the time a machine is starved $\%M^S_i$ or blocked $\%M^B_i$ gives the overall percentage reduction of the mean starving time per part $\%\Delta T^S_i$ and the mean blocking time per part $\%\Delta T^B_i$. This is shown in equations (3) and (4). The total percentage reduction in the time between parts $\%\Delta T_i^P$ for machine i is the sum of the percentage reduction of the starving times $\%\Delta T^S_i$ and blocking times $\%\Delta T^B_i$ as shown in equation (5).

$$\%\Delta T_i^S = \%M_i^S \cdot \frac{\int_0^{\Delta M_i^P \cdot C_i^T} pdfM_i^S(t) \cdot t \cdot dt}{\int_0^{\infty} pdfM_i^S(t) \cdot t \cdot dt} \qquad (3)$$

$$\%\Delta T_i^B = \%M_i^B \cdot \frac{\int_0^{\Delta M_i^S \cdot C_i^T} pdfM_i^B(t) \cdot t \cdot dt}{\int_0^{\infty} pdfM_i^B(t) \cdot t \cdot dt} \qquad (4)$$

$$\%\Delta T_i^P = \%\Delta T_i^S + \%\Delta T_i^B \qquad (5)$$

The above equations estimate the possible reduction in the time between parts $\%\Delta T_i^P$ for all machines i based on the additional number of parts before and free spaces after the machine and the blocking and starving time distributions. However, this estimation does not yet take the complex interactions in the system into account, and the predicted machine improvement may not be realized because other machines continue to block and starve this machine. The transition from a possible machine improvement to the actual system improvement depends on the bottlenecks and is described below.

4.4 System Performance Estimation

The previous step estimated the improvement in the machine performances based on the change in the buffers. However, this improvement may not be realized because other machines continue to block or starve this machine. To estimate the system improvement based on the individual machine improvements, the contribution of the individual machines to the system performance has to be determined, i.e. which machines constrain the system and by how much.

In prior research (Roser, Nakano, and Tanaka 2002a) we have developed a bottleneck detection method based on the active periods of machines that reliably and accurately detects not only the main bottlenecks, but also secondary bottlenecks and non-bottlenecks. The likelihood of a machine i constraining the system is described as the bottleneck probability $\%BN_i$, which is given as the percentage of the time a machine constrains the system. This probability $\%BN_i$ can range from 0% (never a bottleneck) to 100% (always a bottleneck).

This bottleneck probability has been used to estimate the effect of a machine improvement onto the system performance (Roser, Nakano, and Tanaka 2002b) as part of a machine performance prediction model based on a single simulation. The bottleneck probability is now used in a very similar approach to estimate the system performance improvement based on the machine performance improvement. While the shifting bottleneck detection method distinguishes between sole (unique) bottlenecks and shifting bottlenecks (bottlenecks in the process of changing from one machine to another), this method uses the sole bottleneck probability.

The bottleneck probability $\%BN_i$ of a machine i describes what effect a percentage improvement of the time between parts $\%\Delta T_i^P$ of machine i would have on the percentage improvement of the time between parts of the system $\%\Delta T^P$. The improvement of the system $\%\Delta T^P$ is simply the sum of the individual machine improvements $\%\Delta T_i^P$ weighted by the bottleneck probability $\%BN_i$ as shown in equation (6).

$$\%\Delta T^P = \sum_{i=1}^{n} \%\Delta T_i^P \cdot \%BN_i \qquad (6)$$

To get from the initial time between parts of the system T^P to the improved time between parts of the system T^{P*} simply reduce the initial time T^P by the percentage reduction $\%\Delta T^P$ as shown in equation (7). This predicted time per part for increased buffers T^{P*} can then easily be used to predict other system performance measures like the make span or the work in progress.

$$T^{P*} = T^P \cdot (1 - \%\Delta T^P) \qquad (7)$$

5 VALIDATION

The prediction model has been tested for various simulation systems, as for example an eight machine straight manufacturing line, a three machine assembly line, a two parallel machines with a shared buffer, etc. This section presents the results of the example as shown in section 2. This example was selected because it is the most complex example out of the investigated manufacturing systems, and comparable to real manufacturing systems found on the factory floor. Not only are there two part types in a branched system, the system is also well balanced with 4 different machines out of 7 being primary and secondary bottlenecks. Different buffer have been simulated for various sizes, and the simulation results have been compared with the predicted results based on the initial simulation.

The initial system contains very few buffers, and different buffer increases have been studied. Figure 6 shows the comparison of the predicted time per part to the measured time per part for increases of the buffer AM2 located after machine M2. This buffer has been increased from the initial capacity of one up to a capacity of 10. The continuous line shows the measured data, including the 95% confidence intervals, and the dotted lines shows the predicted system performance. Overall, the predicted results show approximately the same tendency as the measured data, and, while the prediction is not perfectly accurate, the prediction model is reasonably close to the measured data. The overall root mean squared error $RMSE$ was 0.34s.

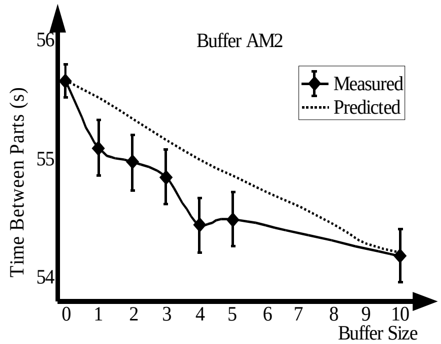

Figure 6: Performance Prediction for Buffer AM2

Figure 7 shows the comparison of the predicted time per part to the measured time per part for the buffer BM3 located before machine M3. Again, the continuous line shows the measured data including the 95% confidence intervals, and the dotted lines shows the predicted system performance. The predicted performance follows the measured data very nicely. The overall root mean squared error *RMSE* was only 0.24s.

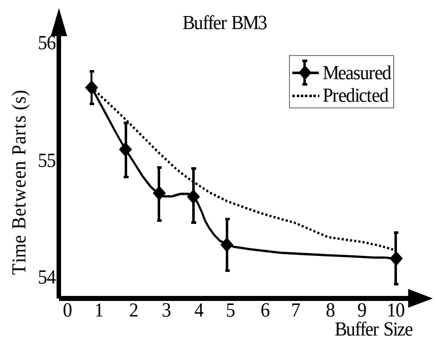

Figure 7: Performance Prediction for Buffer BM3

Figure 8 shows the comparison of the predicted time per part to the measured time per part for the buffer AM3 located after machine M3. While the prediction is close to the measured values for small changes, the prediction overshoots the measured data for larger changes. The overall root mean squared error *RMSE* was 0.38s.

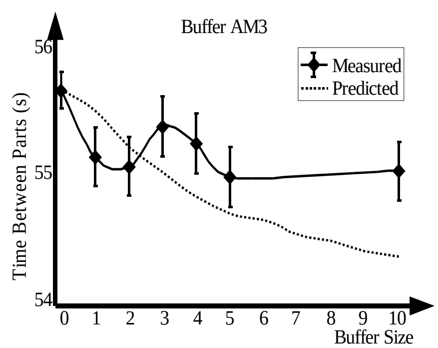

Figure 8: Performance Prediction for Buffer AM3

Overall, the prediction model is sufficiently accurate in view of the complexity of the model. The prediction model performed also very well for simpler models as for example an eight machine sequential system shown in Figure 9. The prediction model can therefore be used as part of a buffer allocation procedure, where the buffer prediction model is used to compare a large number of systems rapidly for a fast optimization, with the resulting optimal system being verified using a conventional simulation.

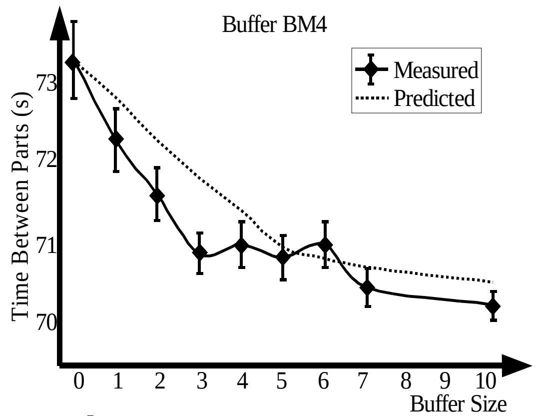

Figure 9: Performance Prediction for an Eight Machine Serial System

6 IMPLEMENTATION

The method has been implemented in a software analysis tool for the TOPQ simulation engine. A screenshot of the software is shown in Figure 10. Besides a thorough statistical analysis and a bottleneck detection, this software also produces a complete prediction model as a MS Excel worksheet. This allows the further use of the prediction model as for example for optimization. Selected charts of the excel output sheet are shown in Figure 11. The software is currently used by selected companies of the TOYOTA group.

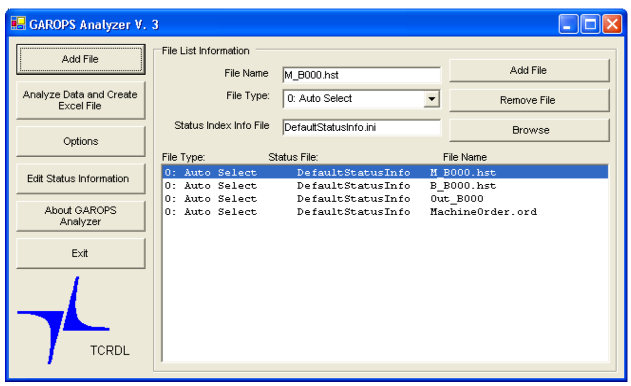

Figure 10: TOPQ Analyzer Screenshot

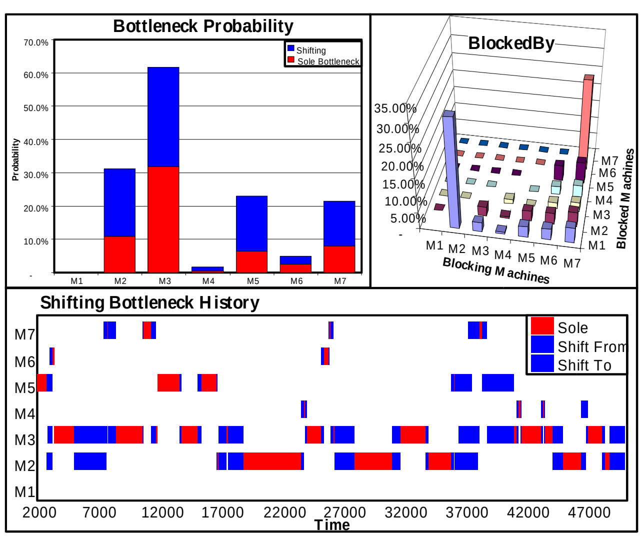

Figure 11: Selected Excel Output Charts

7 CONCLUSIONS

This paper describes a prediction model to estimate the effect of increased buffer capacity onto the system performance based on only a single simulation. The method does not require complex mathematical modeling and can be used for nearly any kind of discrete event system, including balanced and unbalanced lines, serial and/or parallel systems, flow shops, and job shops. There are two main steps to this method.

The first step analyzes the causes of the idle (starving or blocking) periods for all machines, and determines which buffer locations would reduce the idle time. This analysis also includes the indirect effect of buffers that are not in the part flow of a machine M_X, but may be in the part flow of another machine M_Y, which in turn affects the idle periods on the machine M_X. The analysis and graphical display of the path to the causes of the idle periods and the buffer locations in between greatly improves the understanding of the system and the effect of the buffers.

The second step estimates the performance improvement due to an increase in the buffer sizes, using the information gained in the previous step, combined with a statistical analysis of the simulation data. The prediction is accurate and can be used for a further optimization of the manufacturing system to determine the optimal buffer allocation. Furthermore, the method can be used for almost all discrete event systems, including complex lines with a combination of parallel and serial machines.

Overall the presented prediction model is very suitable for use in industry as it is applicable to typical manufacturing systems. The method is fully automated, and the prediction accuracy allows the system optimization using very few simulations for verification.

REFERENCES

Altiparmak, F., Dengiz, B., and Bulgak, A. A. 2002. Optimization of Buffer Sizes in Assembly Systems Using Intelligent Techniques. In *Winter Simulation Conference*, ed. E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 1157-1162, San Diego, CA., USA.

Azadivar, F. 1999. Simulation Optimization Methodologies. In *Winter Simulation Conference*, ed. P. A. Farrington, D. T. Nembhard, D. T. Sturrock, and G. W. Evans, 93-100, Phoenix, AZ, USA.

Boesel, J., Jr., R. O. B., Glover, F., Kelly, J. P., and Westwig, E. 2001. Future of Simulation Optimization. In *Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 1466-1469, Arlington, Virginia, USA: Institute of Electrical and Electronics Engineers.

Bouhchouch, A., Frein, Y., and Dallery, Y. 1993. A Decomposition Method for the Analysis of Tandem Queueing Networks with Blocking Before Service. *Queueing Networks with Finite Capacity*, 97-112.

Brittan, D. 1996. When Bad Things Happen to Good Factories. *Technology Review*.

Caramanis, M., Pan, H., and Anli, O. 2001. Is there a Trade off between Lean and Agile Manufacturing? A Supply Chain Investigation. In *Third Aegean International Conference on "Design and Analysis of Manufacturing Systems"*, ed., Tinos Island, Greece.

Conway, R. W., Maxwell, W., McClain, J. O., and Thomas, L. J. 1988. The role of work-in-process inventory in serial production lines. *Operations Research*, 36(2): 229-241.

Dallery, Y., and Frein, Y. 1989. A Decomposition Method for the Approximation Analysis of Closed Queueing Networks with Blocking. *Queueing Networks with Blocking*, 193-215.

Enginarlar, E., Li, J., and Meerkov, S. M. 2001. A Potpourri on the Theme of Lean Buffering. In *Third Aegean International Conference on "Design and Analysis of Manufacturing Systems"*, Tinos Island, Greece.

Enginarlar, E., Li, J., Meerkov, S. M., and Zhang, R. Q. 2002. Buffer Capacity for Accommodating Machine Downtime in Serial Production Lines. *International Journal of Production Research*, 40(3): 601-624.

Fu, M., Andradottir, S., Carson, J. S., Glover, F., Harrell, C. R., Ho, Y.-C., Kelly, J. P., and Robinson, S. M. 2000. Integrating Optimization and Simulation: Research and Practice. In *Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. R. Barton, 610-615, Orlando, Florida, USA.

Gershwin, S. B., and Schor, J. E. 2000. Efficient Algorithms for Buffer Space Allocation. *Annals of Operations Research*, 93: 117-144.

Kubota, F., Sato, S., and Nakano, M. 1999. Enterprise Modeling and Simulation Platform Integrated Manufacturing System Design and Supply Chain. In *IEEE Conference on Systems, Man, and Cybernetics*, 511-515, Tokyo, Japan.

Levantesi, R., Matta, A., and Tolio, T. 2001. A new algorithm for Buffer Allocation in Production Lines. In *Third Aegean International Conference on "Design and Analysis of Manufacturing Systems"*, Tinos Island, Greece.

Nakano, M., and Ohno, K. 2000. An Integrated Analytical/Simulation Approach for Economic Design of an AGV System. *Journal of the Operations Research Society of Japan*, 43(3): 382-395.

Nakano, M., Sugiura, N., Tanaka, M., and Kuno, T. 1994. ROPSII: Agent Oriented Manufacturing Simulator on the basis of Robot Simulator. In *Japan-USA Symposium on Flexible Automation*, 201-208, Kobe, Japan.

Roser, C., Nakano, M., and Tanaka, M. 2002a. Shifting Bottleneck Detection. In *Winter Simulation Conference*, ed. E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, San Diego, CA, USA.

Roser, C., Nakano, M., and Tanaka, M. 2002b. Throughput Sensitivity Analysis using a single simulation. In *Winter Simulation Conference*, ed. E. Yucesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes.

Roser, C., Nakano, M., and Tanaka, M. 2003. Constraint Management in Manufacturing Systems. *International Journal of the Japan Society of Mechanical Engineering, Series C, Special Issue Advanced Scheduling*, 46(1): 73-80.

Schor, J. E. 1995. Efficient Algorithms for Buffer Allocation, Masters of Science, Massachusetts Institute of Technology, Boston.

Shi, L., and Men, S. 2002. Optimal Buffer Allocation in Production Lines. *Submitted to IIE Transactions*.

Spinellis, D. D., and Papadopoulos, C. T. 1999a. Explore: A Modular Architecture for Production Line Optimisation. In *Proceedings of the 5th International Conference of the Decision Science Institute*, ed. D. K. Despotis and C. Zopounidis, 1446-1449, Athens, Greece.

Spinellis, D. D., and Papadopoulos, C. T. 1999b. Production Line Buffer Allocation: Genetic Algorithm versus Simulated Annealing. In *Second International Aegean Conference on the Analysis and Modelling of Manufacturing Systems*, 89-101, Tinos, Greece: University of the Aegean, Department of Business Administration.

Spinellis, D. D., and Papadopoulos, C. T. 2000a. A Simulated Annealing Approach for Buffer Allocation in Reliable Production Lines. *Annals of Operations Research*, 93: 373.

Spinellis, D. D., and Papadopoulos, C. T. 2000b. Stochastic Algorithms for Buffer allocation in Reliable Production lines. *Mathematical Problems in Engineering*, 5: 441-458.

Swisher, J. R., Hyden, P. D., Jacobson, S. H., and Schruben, L. W. 2000. A Survey of Simulation Optimization Techniques and Procedures. In *Winter Simulation Conference*, ed. P. A. Fishwick, K. Kang, J. A. Joines, and R. R. Barton, 119-128, Orlando, Florida, USA.

Vouros, G. A., and Papadopoulos, H. T. 1998. Buffer Allocation in unreliable production lines using a knowledge based system. *Computers and Operations Research*, 25(12): 1055-1067.

AUTHOR BIOGRAPHIES

CHRISTOPH ROSER is a researcher at the Toyota Central Research and Development Laboratories, Nagoya, Japan since 2000. Current research topics are simulation output analysis, simulation optimization and logistic systems. He graduated from the Fachhochschule Ulm, Germany with a diploma in automation engineering, and received his Ph.D. in Mechanical Engineering at the University of Massachusetts, Amherst, studying flexible and robust design methods. He can be contacted by email at
`<croser@robotics.tytlabs.co.jp>`

MASARU NAKANO is a principal researcher at Toyota central Research and Development Laboratories, Inc. (TCRDL) in Japan. He joined TCRDL in 1980 and since then has studied in the field of robotics, computer vision, and manufacturing system design. He is currently research leader and manager of the Digital Engineering Laboratory. He received his B.S. and M.S. in operations research from Kyoto University, and his Dr. Eng. in manufacturing systems from the Nagoya Institute of technology. He can be contacted by email at
`<nakano@robotics.tytlabs.co.jp>`

MINORU TANAKA is a researcher at the Toyota Central Research and Development Laboratories, Nagoya, Japan. He graduated from Matsue Technical College, joined TCRDL in 1985, and specialized in the field of robotics, manufacturing system modeling, and discrete event simulation. He is a member of the robotics society of Japan. He can be contacted by email at
`<tanaka@robotics.tytlabs.co.jp>`