Roser, Christoph, Masaru Nakano, and Minoru Tanaka. “Shifting Bottleneck Detection.” In Winter Simulation Conference, edited by Enver Yucesan, C. -H Chen, J. L Snowdon, and John M Charnes, 1079–86. San Diego, CA, USA, 2002.

SHIFTING BOTTLENECK DETECTION

Christoph Roser
Masaru Nakano
Minoru Tanaka

Software Science Laboratory
Toyota Central Research and Development Laboratories
Nagakute, Aichi, 480-1192, JAPAN

ABSTRACT

This paper provides a novel method for detecting bottlenecks in manufacturing systems and the shifting of these bottlenecks. All manufacturing systems are constrained by one or more bottlenecks. Improving the bottleneck will improve the whole system. Yet, finding the bottleneck is no trivial task. Furthermore, the system may change over time or due to random events, and subsequently the bottleneck may shift from one machine to another machine. The shifting bottleneck detection method determines the bottleneck based on the duration a machine is active without interruption. The method is very robust, easy to apply and able to detect the primary and secondary bottlenecks in a wide range of production systems. This allows the use of simulation to predict bottlenecks for both steady state and variable systems. The measurement of the likelihood of a machine being the bottleneck aids in the decision-making regarding the allocation of the available resources.

1 INTRODUCTION

This paper describes a method to detect and monitor the bottleneck in steady state and non-steady-state production system subject to random variation. There are numerous definitions as to what constitutes a bottleneck (Lawrence and Buss 1995). Within this paper, we define a bottleneck as a stage in a production system that has the largest effect on slowing down or stopping the entire system. The shifting bottleneck method further distinguishes between a momentary bottleneck, describing the bottleneck at any given point in time, and an average bottleneck, describing the bottleneck behavior over a selected period of time.

While the shifting bottleneck approach is based on the theory of constraints (Blackstone 2001; Goldratt 1992), the method is not limited to long-term average bottlenecks. Although most manufacturing systems usually have one main bottleneck, in all but the simplest applications bottlenecks are not static but rather shift between different machines (Lawrence and Buss 1994; Moss and Yu 1999). These shifts may for example be due to the sequence of random events or due to a gradual change in the manufacturing system. A non-bottleneck machine may become a bottleneck, for example due to a machine failure, and similarly a bottleneck machine may become a non-bottleneck machine. Over longer periods of time, a system therefore may not only have one primary bottleneck, but also secondary and tertiary bottlenecks, i.e., machines which are also occasional bottlenecks, yet to a lesser extent than the primary bottleneck. To improve the system throughput there is a two-pronged approach. One task is to reduce the cycle times of the main bottleneck machine. The other task is to reduce the idle time of the main bottleneck machine by ensuring a steady supply of parts of the bottleneck machine to achieve a utilization approaching 100%. The presented method detects both the main bottleneck for an improvement of the cycle time, and the secondary bottlenecks, whose improvement reduces the idle time of the main bottleneck.

Finding the bottleneck is no trivial task, and Cox and Spencer (1997) for example simply recommends that ‘… the best approach is often to go to the production floor and ask knowledgeable employees …’. Fortunately, there are a number of systematic methods available to find the bottleneck for production systems. One approach measures the utilization of the different machines of the production system (Law and Kelton 2000). The machine with the highest utilization is considered to be the bottleneck. However, the utilizations of different machines are often very similar, and it cannot be said for sure which machine is the bottleneck. Longer simulations may be required to generate meaningful and accurate results. Furthermore, this method is limited to steady state systems. The utilization method is also unable to determine the momentary bottleneck, but only the average bottleneck over long periods of time, making it unsuitable to detect and monitor shifting bottlenecks.

Another frequently used method analyses the queue lengths of the machines in the production systems. In this method, either the queue length or the waiting time is determined, and the entity with the longest queue length or waiting time is considered to be the bottleneck. This method has the advantage that a momentary bottleneck can be determined by simply comparing the queue lengths or waiting times. The average bottleneck can also be detected using the average queue length or waiting time. Yet, this method has a number of other shortcomings. First and foremost, many production entities have only a limited queue or no queue at all, in which case the queue length cannot be used to detect the bottleneck. Also, in a saturated production system where the supply of new parts exceeds the capacity of the system, the queue lengths and waiting times of all queues in front of the bottleneck approach the maximum buffer size and the queue lengths cannot be used to determine the bottleneck. Furthermore, if the batch sizes vary for different machines throughout the production system, the waiting time or queue length may give in some cases incorrect results. Finally, the waiting time of the parts or the queue length is a heavily dependent factor and not independent and identically (i.i.d.) distributed. Therefore, it is difficult to estimate the accuracy of the average queue length or waiting time over periods of time.

Other methods are for example a very rigorous mathematical approach developed by Chiang, Kuo, and Meerkov (Chiang, Kuo, and Meerkov 1998; Chiang, Kuo, and Meerkov 2002; Kuo, Lim, and Meerkov 1996), analyzing the interaction between the machines in order to determine the effect of the machines onto the bottleneck. Adams, Balas, and Zawack (1988) uses disjunctive graphs to detect the bottleneck in order to optimize the scheduling in a shifting bottleneck procedure. Uzsoy and Wang (2000) compared the shifting bottleneck procedure to the theory of constraints.

In summary, there are various methods available, yet all of them have one or more disadvantages, for example in terms of reliability, usability, resolution and accuracy. The method described below is able to overcome these disadvantages, and allows the detecting and monitoring of both momentary and average bottlenecks over any selected period of time. This allows the use of the bottleneck detectio method for simulation analysis, predicting the bottleneck probabilities for both steady state and non-steady state discrete event systems, as for example manufacturing systems, computer networks or logistics. The method is also easy to implement, requiring no knowledge of the structure of the production system, and therefore very easy to implement in any existing simulation software. The method will be compared to the measurement of the queue length used by Lawrence and Buss (1994) to determine the shifting bottlenecks, where the longest queue at any given time determines the bottleneck.

2 BOTTLENECK DETECTION METHOD

The presented method will be able to detect and monitor the shifting momentary bottleneck of a production system, and also determine the average bottleneck over a selected period of time based on the duration the machines are active without interruption. This method is a continued development and improvement based on the method of the average active duration (Roser, Nakano, and Tanaka 2001), expanding the theory of constraints into momentary and shifting bottlenecks (Lawrence and Buss 1994), (Moss and Yu 1999).

2.1 The Active Duration

The presented method is based on the duration a processing machine is active without interruption. As a first step, it is necessary to group all possible machine states into two groups, being either active states or inactive states. A state is active whenever the machine may cause other machines to wait. For example working on one part may cause a subsequent idle machine to wait for the completion of the part, or a machine under repair may block previous machines. A state is inactive if the associated machine is not active but instead waiting for the completion of another task, for example the arrival of a part or service, or for the removal of a part. Table 1 shows a possible grouping of selected states for different entities of a production system into active and inactive.

Table 1: Active – Inactive States for Different Machines

Machine	Active	Inactive
Processing Machine	Working, in repair, changing tools, serviced	Starving, blocked
Automated Guided Vehicles (AGV)	Moving to a pickup location, moving to a drop off location, recharging, being repaired	Waiting
Factory Worker	Working, on scheduled break	Waiting

Figure 1 shows an example of the active (work, repair, tool change) and inactive (waiting) states of one machine during a brief period of a simulation. The active periods without interruption are shown.

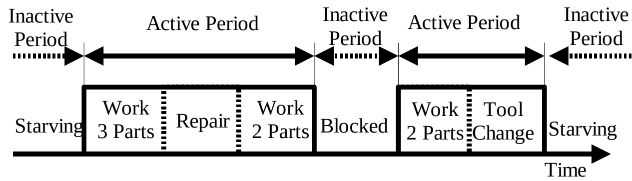

Figure 1: Active Periods of Machine During Simulation

The bottleneck detection method compares the durations of the active periods of the different machines. If the analysis is based on simulation data or historical data, it is possible to determine the durations of all active periods for all machines. However, if the analysis is used for real time monitoring, the future is unknown and the durations of the active periods are known only until the present. In this case, the active duration is measured until the present and may be updated if further information becomes available with time.

2.2 The Momentary Bottleneck

The underlying idea of the method is that at any given time the machine with the longest uninterrupted active period is the momentary bottleneck at this time. The overlap of the active period of a bottleneck with the previous or subsequent bottleneck represents the shifting of the bottleneck from one machine to another machine. In an interconnected production system, machines block and starve each other. If a machine is active, it is neither starved nor blocked. The longer a machine is active without interruption, the more likely it is that this machine blocks or starves other machines in the production system. The machine with the longest uninterrupted active period therefore has the biggest impact onto starving or blocking the other machines, therefore being the largest constraint a.k.a. the largest bottleneck. The following method describes how to determine which machine of a production system is the sole or part of a shifting bottlenecks at any time *t*.

If at time *t* no machines are active, then there is no bottleneck. If one or more machines are active at the time *t*, the machine with the longest active period at the time *t* is the momentary bottleneck machine, and the active period of this machine is the current bottleneck period. It is also necessary to find the previous and subsequent bottleneck machines before and after the current bottleneck period. The previous bottleneck machine is the machine with the longest active period just prior to the beginning of the current bottleneck period. Similarly, the subsequent bottleneck machine is the machine with the longest active period just after the end of the current bottleneck period.

The shifting of the bottleneck from the previous bottleneck machine to the current bottleneck machine happens during the overlap of the previous and the current bottleneck periods. Similarly, the shifting of the bottleneck from the current bottleneck machine to the subsequent bottleneck machine happens during the overlap of the current and the subsequent bottleneck periods. During the overlaps between the bottleneck periods no machine is the sole bottleneck, instead the bottleneck shifts between the two machines. If a bottleneck machine is not shifting, then this machine is the sole and only bottleneck at this time. Of course, if there are no other machines active just prior or after the current bottleneck period, then there is no overlap and subsequently no shifting bottleneck. Using this method, it can be determined at any given time if a machine is a non-bottleneck, a shifting bottleneck, or a sole bottleneck. This method allows the detection of the bottleneck, where and when the previous bottleneck was shifting to the current bottleneck, and where and when the current bottleneck is shifting to the next bottleneck.

Figure 2 illustrates the method using a simple example consisting of only two machines. The figure shows the active periods of the machines over a short period of time. At the selected time t, both machines M1 and M2 are active. Yet, as M1 has the longer active period, M1 is the bottleneck machine for the time t. As there is no machine active before the current bottleneck period, there is no overlap and no shifting at the beginning of the current bottleneck period. However, at the end of the current bottleneck period, M2 is active and has the longest active period. Therefore, the subsequent bottleneck machine is M2. During the overlap between the current bottleneck period and the subsequent bottleneck period the bottleneck shifts from M1 to M2. Now, M2 is the bottleneck machine. Similarly, at the end of the bottleneck period of M2, the bottleneck shifts back to M1. Processing all available data using this method shows at what time which machine is the bottleneck machine, when the bottleneck is shifting, and when there is no bottleneck at all. Therefore, it is possible to detect and monitor the bottleneck at all times.

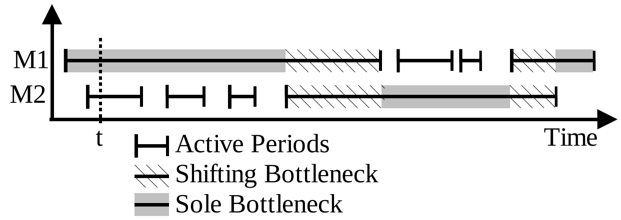

Figure 2: Shifting Bottlenecks

2.3 The Average Bottleneck

The above method detects and monitors the momentary bottleneck at any instant of time. However, in many cases it may be of interest not to investigate an instant of time but rather a period of time. This section describes how to compare different machines with respect to the bottleneck over a period of time. To determine the bottleneck during a period of time the available data is analyzed and the momentary bottlenecks are determined over the selected period of time. Next, the percentage of time a machine is the sole bottleneck machine and the percentage of the time a machine is part of a shifting bottleneck is measured for the selected period of time.

Figure 3 illustrates this method using the example with two machines as shown in Figure 2. The percentages of the machines being the sole bottleneck or the shifting bottleneck have been measured over the period of time shown in Figure 2. The larger the percentages, the larger is the effect of the respective machine onto slowing down or stopping the system. M1 is the sole bottleneck more often than M2, and is also involved in a number of shifting operations. M2 is the smaller constraint, i.e., a secondary bottleneck, having being the sole bottleneck for a smaller percentage of time. The graph below shows the overall effect of the machines in terms of being the bottleneck over a period of time by plotting the sum of the machines being the bottleneck or shifting. Overall, an improvement of the throughput of M1 would yield a larger overall improvement of the system throughput than an improvement of M2, as M1 is the primary bottleneck during the selected period of time.

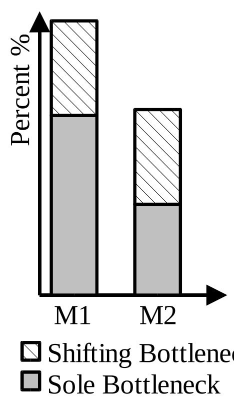

Figure 3: Average Bottleneck over Period of Time

3 COMPUTATIONAL EXAMPLES

This section will describe three computational examples. The first two examples are a flow shop and a job shop with four stations each, taken with small modifications from Lawrence and Buss (1994). The last example is a complex branched system with seven machines and two different part types. The shifting bottleneck detection method was implemented as software tool GAROPS Analyzer to analyze the simulation data from the GAROPS simulation software as shown by Kubota, Sato, and Nakano (1999) and Nakano et al. (1994). This tool was also adapted to analyze the simulation results from the ARENA simulation software (Kelton, Sadowski, and Sadowski 1997). The GAROPS Analyzer analyses the machine state information over time and creates an excel file containing a statistical description of the simulation including the change of the sole and shifting momentary bottlenecks over time and also the sole and shifting average bottlenecks of the complete simulation.

Lawrence and Buss (1994) also devised a bottleneck shiftiness measure β as shown in equation (1), where c_v is the coefficient of variation of the bottleneck probability of the different machines and n is the number of machines in the system. The bottleneck shiftiness measure β ranges from zero for a system with a unique bottleneck to one for a system where all machines are equally likely to be the bottleneck. The bottleneck shiftiness measure can also be applied to the active duration method and will be utilized in the examples below.

$$\beta = 1 - \frac{c_v}{\sqrt{n}} \tag{1}$$

3.1 Flow Shop

The flow shop example has an exponential inter arrival distribution with a mean inter arrival time of 1.25s. The processing times of the four machines has an exponential distribution with a mean service time μ_i of 1s for machines M1, M2, and M4, and 1.1s for machine M3. All parts are processed by all machines in sequence. The utilization p_i is 80% for machines M1, M2, and M4, and 88% for machine M3, which is subsequently the overall bottleneck. Figure 4 shows the layout of the flow shop system.

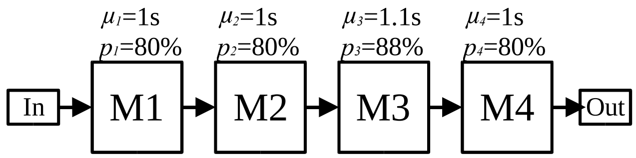

Figure 4: Flow Shop Layout

Similar to the aforementioned paper by Lawrence and Buss, the simulation was run for 120,000s, of which a warming up period of 20,000s was removed. The resulting simulation data was analyzed using the GAROPS Analyzer. The percentages of the time a machine was the sole bottleneck and the percentages of time a machine was part of a shifting bottleneck have been analyzed using the active duration method as described above. Table 2 shows the measured results of the simulation. For each machine, the utilization is given in column two. The percentages of the time a machine is the sole bottleneck and the percentage of the time a machine is part of a shifting bottleneck as described above are given in column three and four. The sum of the shifting and sole bottleneck percentages is given in the last column. The last row shows the bottleneck shiftiness measure β for the different bottleneck measurements according to equation (1). The results of Table 2 are also illustrated in Figure 5, including the confidence intervals with a 95% confidence level for the total bottleneck probability.

Table 2: Flow Shop Simulation Results

Machine	Utilization	%Sole	%Shifting	%Sum
M1	80.1%	12.67%	20.43%	33.1%
M2	80.2%	6.73%	15.93%	22.7%
M3	88.0%	32.54%	29.27%	61.8%
M4	80.0%	7.25%	15.23%	22.5%
Shiftiness Measure β		0.59	0.84	0.74

Machine M3 is clearly the bottleneck, as all measures in Table 2 indicate M3 as the main bottleneck. Machine M3 is the sole bottleneck for about 1/3rd of the time, and a shifting bottleneck for another 1/3rd of the time. This makes M3 a sole or shifting bottleneck for about 2/3rd of the time. Improving the main bottleneck M3 will improve the overall system throughput. Furthermore, improving the secondary bottlenecks M1, M2 and M4 would also improve the system throughput by reducing the idle time of the main bottleneck.

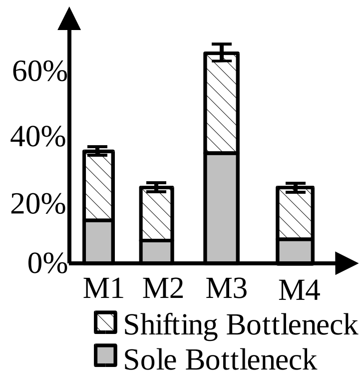

Figure 5: Flow Shop Shifting and Sole Bottleneck Probabilities, including 95% Confidence Interval on Total

However, there is an interesting phenomenon at the non-bottleneck machines M1, M2 and M4. These machines have different bottleneck probabilities despite having an identical utilization. Machines at the beginning of the flow shop are more likely to be the bottleneck than at the end of the flow shop for equal utilization. It appears that in a system with unlimited demand the likelihood of starving a subsequent machine increases with the number of subsequent machines. As the presented flow shop has an unlimited buffer size, there is no blocking of previous machines, which may balance the bottleneck probabilities. This also implies that the utilization is not always a suitable measurement to detect the bottleneck in a system. Thus, machine M1 is the secondary bottleneck, and an improvement of M1 may also improve the overall system. The large bottleneck shiftiness measure β indicates that the bottlenecks in the flow shop are not very distinct.

3.2 Job Shop

The job shop example is very similar to the flow shop example, except for the processing sequence. The job shop example also has an exponential inter arrival distribution with a mean inter arrival time of 1.25s. The processing times of the four machines have an exponential distribution with a mean service time μ_i of 1s for machines M1, M2, and M4, and 1.1s for machine M3. An arriving part has a probability of 25% to go to any of the four machines. After a machine processes a part, there is a 25% chance of the part going to any of the other three machines, and a 25% chance of the part leaving the system. This random sequencing approach avoids the effects of a flow shop as shown in the previous example. The utilization rates are identical with the flow shop example. The layout of the system is given in Figure 6.

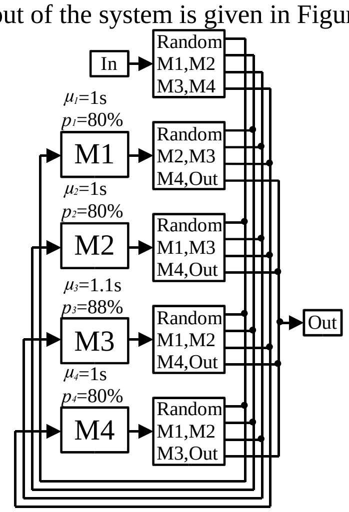

Figure 6: Job Shop Layout

Using the same settings as the example by Lawrence and Buss, the simulation was run for 120,000s, of which a warming up period of 20,000s was removed. The resulting simulation data was analyzed using the GAROPS Analyzer. Table 3 shows the results of the simulation. The layout is very similar to Table 2. For each machine, the utilization is given in column two. The percentages of the time a machine is the sole bottleneck and the percentage of the time a machine is part of a shifting bottleneck as described above are given in column three and four. The fifth row shows the sum of the percentages being a shifting and sole bottleneck. The last row shows the bottleneck shiftiness measure β for the different bottleneck measurements according to equation (1). The results of Table 3 are also illustrated in Figure 7, including the confidence intervals with a 95% confidence level for the total bottleneck probability.

Table 3: Job Shop Simulation Results

Machine	Utilization	%Sole	%Shifting	%Sum
M1	80.2%	10.7%	15.8%	26.5%
M2	80.0%	10.3%	14.9%	25.2%
M3	87.6%	33.6%	22.1%	55.6%
M4	79.8%	11.4%	15.1%	26.6%
Shiftiness Measure β		0.65	0.90	0.78

As expected, machine M3 is again clearly the bottleneck, as all measures in Table 3 find M3 to be the main bottleneck. Overall, M3 is a sole or shifting bottleneck for about ½ of the time. Improving the main bottleneck M3 will improve the overall system throughput. In addition, improving the secondary bottlenecks M1, M2 and M4 would also improve the system throughput by reducing the idle time of the main bottleneck. As there is no fixed sequence in the job shop, all non-bottleneck machines M1, M2 and M4 have an equal likelihood of being the bottleneck at any given time. The large bottleneck shiftiness measure β indicates that the bottlenecks in the job shop are also not very distinct.

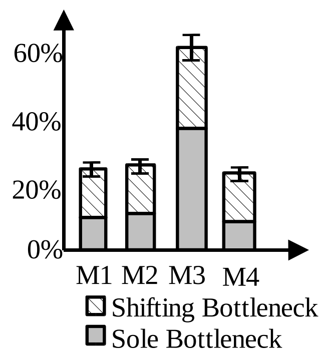

Figure 7: Job Shop Shifting and Sole Bottleneck Probabilities, including 95% Confidence Interval on Total

3.3 Complex Example

The complex example consists of a branched system with seven machines and an infinite supply of two different part types as shown in Figure 8. The buffer size for the different machines ranges from zero (no buffer at all) to five, depending on the buffer location. The simulation was run for 200,000s, of which the first half was removed as the warming up period. This warming up period of 100,000s was selected very conservatively to achieve a steady state, and the analyzed simulation time of 100,000s allows a comparison of the simulation accuracy with the previous examples having the same analyzed simulation time.

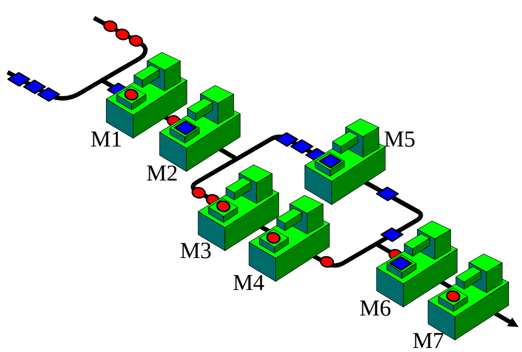

Figure 8: Complex Example Layout

Figure 9 shows the utilization of the seven machines, including the ranges of the 95% confidence intervals. The potential primary bottlenecks are shaded. Based on this simulation, it cannot be said for sure which machine is the primary bottleneck. It appears that M5 is the primary bottleneck, yet as the 95% confidence intervals of M3 and M5 overlap there is no statistical confidence that M5 is truly the bottleneck. Therefore, it is difficult to detect the primary bottleneck by measuring the utilization, let alone secondary and tertiary bottlenecks.

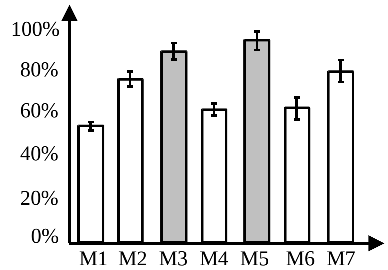

Figure 9: Utilization of Complex Example

Figure 10 and Table 4 show the result of the shifting bottleneck detection using the active period. The confidence intervals with a confidence level of 95% for of the total bottleneck probability are shown. Due to the fact that the complex example includes rare events, the confidence intervals are comparatively wider than the previous examples without machine failures as shown in Figure 5 and Figure 7. Still, the results are very clear, showing that M5 is indeed the main bottleneck, being a sole bottleneck for 45% of the time and a shifting bottleneck for 37% of the time, i.e., M5 is part of a bottleneck for 82% of the time. This example also indicates that M3 is a potential secondary bottleneck and M7 is a potential tertiary bottleneck, although the confidence intervals of M2, M3 and M7 are too wide to make an exact distinction.

Table 4: Complex Example Simulation Results

Machine	Utilization	%Sole	%Shifting	Sum
M1	54%	0.0%	0.1%	0.1%
M2	76%	2.2%	3.3%	5.6%
M3	89%	1.2%	29.3%	30.5%
M4	62%	0.1%	0.0%	0.1%
M5	94%	45.1%	37.3%	82.4%
M6	63%	1.5%	3.6%	5.1%
M7	80%	7.0%	12.5%	19.5%
Shiftiness Measure β		0.24	0.54	0.46

In summary, an improvement of the throughput M5 would improve the overall system throughput , as M5 is the main bottleneck. Machines M3, M7 and M2 may also be considered for improvements, as this would improve the system by reducing the idle time of the main bottleneck M5, depending on the trade off between the cost of the improvement and the benefit of the improved system throughput . Furthermore, the bottleneck analysis determines that an improvement of M1, M4 and M6 is unlikely to increase the system throughput , and no resources should be invested into an improvement of M1, M4 and M6 at this time. The small bottleneck shiftiness measure β also indicates that the bottlenecks of the complex system are more pronounced than the bottlenecks of the flow shop and the job shop example.

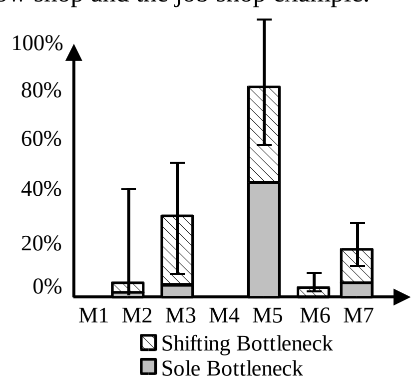

Figure 10: Complex Example Shifting and Sole Bottleneck Probabilities, including 95% Confidence Interval on Total

4 ADVANTAGES

The active period method has many advantages over other methods for bottleneck detection. While, for example, methods based on the queue length or waiting time are restricted by the length of the queue or the batch sizes as described in the introduction, the presented active period method has no such restrictions and can be used regardless of the buffer sizes.

In addition, the flow shop and job shop queuing systems in the above examples can also be seen as having the bottleneck in the supply of parts. If the supply would be able to provide more parts, the throughput would increase until the bottleneck is at 100% utilization. Yet, the length of the queues in front of the bottleneck approach infinity, and the queue length cannot be used to detect the bottleneck. The active period method can be adapted to detect bottlenecks in a wide variety of production system configurations and entities, as for example the supply, the demand, Automated Guided Vehicles (AGVs), computer networks or factory workers.

Measuring the utilization also introduces potential errors as this method ignores the processing sequence. In the flow shop example above, machines M1, M2 and M4 had identical utilizations. Yet, machine M1 at the beginning of the system is much more likely to be the bottleneck than machine M4 at the end of the system. The active period method detects the bottleneck with respect to the position of the machine in the sequence, even though the machining sequence is not known to the active period algorithm.

Furthermore, a detailed comparison of the shifting bottleneck based on the active period and the queue length as used by (Lawrence and Buss 1994) reveals that the queue length fluctuates much more than the active periods. Figure 11 shows the shifting bottleneck for a short period of the flow line example described above. The upper graph uses the active period method to detect the sole and shifting bottlenecks. Machine M1 is the first bottleneck, which then shifts to machine M3, and later shifts back to machine M1. The lower graph uses the queue length to detect the bottleneck. Due to the random changes in the queue length, the bottleneck shifts back and forth rapidly between machines M1, M2 and M3. Yet, merely because a machine has the longest queue for a very short time interval, this does not indicate that this machine is the bottleneck for this short time interval. In addition, during the transition of the bottleneck from machine M1 to machine M3, machine M2 has temporarily a longer queue than machine M1 and M3. Yet, as the active period is shorter than for machines M1 and M3, machine M2 is not the bottleneck.

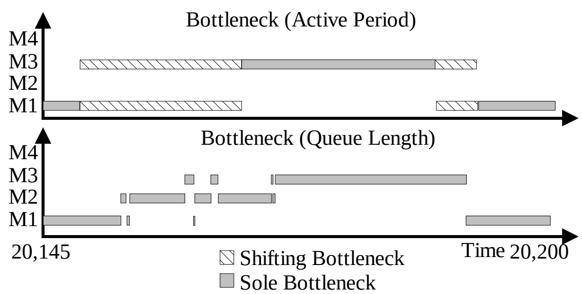

Figure 11: Shifting Bottleneck According to Active Period and Queue Length for Flow Shop Example

The fluctuations of the bottleneck based on the queue length method can cause difficulties for the control of the production system. Many production systems use a "chase the bottleneck" approach to allocate resources to the bottleneck machine in order to improve the overall system throughput. In this case, the fluctuations of the queue length method require frequent reallocations of the available resources. Yet, reallocations also require time and effort, during which the resources do not contribute to the production system throughput . In comparison, the active period method has much less fluctuations. Over the 100,000s simulation period of the flow shop example, there have been a total of 16,000 shifts in the bottleneck for the active period method. The queue length method, however, shifts 27,000 times during the same period, almost twice as often. This results into a much larger effort to "chase the bottleneck".

5 SUMMARY

The active period method as presented in this paper is a very flexible tool and can be used for a wide range of production systems containing a wide range of entities as for example machines, AGVs, factory workers, computer networks, and supply and demand logistics. The method is easy to apply, and the required data is usually readily available. The internal structure of the simulation is not needed, merely a history of the machine activities. As the active period is measured directly at the machine, there are no errors due to outside limitations as for example in the indirect measurement of the machine activity using the queue length. Knowing the likelihood of each machine to be the bottleneck aids the manager in making a trade off between the effort of adding capacity and the benefits of improved throughput.

Research is in progress to adapt the active period method to detect the bottleneck in real time, allowing the monitoring of the bottleneck as it shifts between different machines over time, and to improve the scheduling to optimize the throughput of the shifting bottleneck machines. This gives the manager of the production system valuable information in order to improve the overall system performance.

REFERENCES

Adams, J., Balas, E., and Zawack, D. 1988. The Shifting Bottleneck Procedure for Job-Shop Scheduling. *Management Science*, 34(3), 391-401.

Blackstone, J. H. 2001. Theory of constraints - a status report. *International Journal of Production Research*, 39(6), 1053-1080.

Chiang, S.-Y., Kuo, C.-T., and Meerkov, S. M. 1998. Bottlenecks in Markovian Production Lines: A Systems Approach. *IEEE Transactions on Robotics and Automation*, 14(2), 352-359.

Chiang, S.-Y., Kuo, C.-T., and Meerkov, S. M. 2002. c-Bottlenecks in Serial Production Lines: Identification and Application. *Mathematical Problems in Engineering*, to appear 2002.

Cox, J. F. I., and Spencer, M. S. 1997. *The Constraints Management Handbook.* Boca Raton, Florida: CRC Press - St. Lucie Press.

Goldratt, E. M. 1992. *The Goal: A Process of Ongoing Improvement.* North River Press.

Kelton, D. W., Sadowski, R. P., and Sadowski, D. A. 1997. *Simulation With Arena.* McGraw Hill College Div.

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*, ed., 511-515, Tokyo, Japan.

Kuo, C.-T., Lim, J.-T., and Meerkov, S. M. 1996. Bottlenecks in Serial Production Lines: A System-Theoretic Approach. *Mathematical Problems in Engineering*, 2, 233-276.

Law, A. M., and Kelton, D. W. 2000. *Simulation Modeling & Analysis.* McGraw Hill.

Lawrence, S. R., and Buss, A. H. 1994. Shifting Production Bottlenecks: Causes, Cures, and Conundrums. *Journal of Production and Operations Management*, 3(1), 21-37.

Lawrence, S. R., and Buss, A. H. 1995. Economic Analysis of Production Bottlenecks. *Mathematical Problems in Engineering*, 1(4), 341-369.

Moss, H. K., and Yu, W. B. 1999. Toward the Estimation of Bottleneck Shiftiness in a Manufacturing Operation. *Production and Inventory Management Journal*, 40(2), 53-58.

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*, ed., 201-208, Kobe, Japan.

Roser, C., Nakano, M., and Tanaka, M. 2001. A Practical Bottleneck Detection Method. In *Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 949-953, Arlington, Virginia, USA: Institute of Electrical and Electronics Engineers.

Uzsoy, R., and Wang, C.-S. 2000. Performance of decomposition procedures for job shop scheduling problems with bottleneck machines. *International Journal of Production Research*, 38(6), 1271-1286.

AUTHOR BIOGRAPHIES

CHRISTOPH ROSER is a researcher at the Toyota Central Research and Development Laboratories, Nagoya, Japan. Current research topics are simulation output analysis, bottleneck detection and risk and sensitivity analysis. 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. His email address is <`croser@robotics.tytlabs.co.jp`>

MASARU NAKANO is a senior 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 Research Domain. 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. His email address is <`nakano@robotics.tytlabs.co.jp`>

MINORU TANAKA is an associate 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, manufac-

turing system modeling, and discrete event simulation. He is a member of The Robotics Society of Japan. His email address is
<tanaka@robotics.tytlabs.co.jp>