CONSTRAINT MANAGEMENT IN MANUFACTURING SYSTEMS [1]

Christoph Roser
Masaru Nakano
Minoru Tanaka
Toyota Central Research and Development Laboratories
Nagakute, Aichi, 480-1192, JAPAN
croser@robotics.tytlabs.co.jp
nakano@robotics.tytlabs.co.jp
tanaka@robotics.tytlabs.co.jp

Keywords:
Flexible Manufacturing System (FMS)
Manufacturing Bottlenecks
Operation Planning
Production Management

Word count: 4879 words

---

[1] This paper is based on the paper "Detecting Shifting Bottlenecks" by Christoph Roser, Masaru Nakano and Minoru Tanaka, presented at the "International Symposium on Scheduling" in Hamamatsu on June 4-6. The paper has been edited and expanded for the special issue of the "International Journal of the Japan Society of Mechanical Engineering" .

ABSTRACT

This paper provides a novel method for detecting production bottlenecks and the shifting of the production bottlenecks. All production systems are constrained by one or more bottlenecks. Improving the bottleneck will improve the whole production 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 presented active duration 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. The method is demonstrated using different examples. The measurement of the likelihood of a machine being the bottleneck aids in the decision-making regarding the allocation of the available resources.

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, both for flow shop and job shop systems. Within this paper, a bottleneck is seen as a stage or step in a production system that has the largest effect on slowing down or stopping the entire system, either for an instant in time or averaged over a longer time period. Therefore, it is of interest to determine the bottleneck in order to improve the throughput of the production system by improving the throughput of the bottleneck, also known as the theory of constraints [1, 2]. The paper 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. Yet, finding the bottleneck is no trivial task, and Cox et al. for example simply recommend that '… the best approach is often to go to the production floor and ask knowledgeable employees …' [3].

Furthermore, in all but the simplest applications the bottleneck is not static. Instead, the bottleneck shifts between different machines, depending on the preceding random events. 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. The method presented in this paper considers the shifting of both momentary and average bottlenecks.

Currently there are a number of methods in use to find the bottleneck for production systems. One approach measures the utilization of the different machines of the production system [4]. The machine with the highest utilization is considered to be the bottleneck. 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. The disadvantages of these methods will be described in more detail below. Chiang and Kuo et al. use the sensitivity of the machine performance to the overall throughput as a theoretical bottleneck measure [5, 6]. Adams et al. use disjunctive graphs to detect the bottleneck in order to optimize the scheduling in a shifting bottleneck procedure [7]. Uzsoy et al. compare the shifting bottleneck procedure to the theory of constraints [8].

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. This method is a continued development and improvement based on the method of the average active duration [9], as presented at the International Symposium on Scheduling in Hamamatsu, Japan 2002 [10]. The following section introduces the conventionally available methods and shows its shortcomings before introducing the new bottleneck detection method.

Conventional Bottleneck Detection Methods

Conventionally, there are two methods commonly used to determine the bottleneck in a manufacturing system. The first method is based on the machine utilization, i.e. the percentage of the time a machine is active. The idea is that the machine with the largest utilization is the bottleneck. Unfortunately, there are many situations where this approach does not work very well or not at all. First of all, in many cases it is rather difficult to measure the utilization with sufficient accuracy to determine the bottleneck. As the utilization is based on a time series, the variation of the utilization is difficult to measure, requiring for example complicated batch means methods [11]. Even with suitable tools to measure the variation [12, 13], large sets of data are required to reduce the confidence interval widths to a level required for meaningful conclusions. This also forbids the use of the utilization method for short or medium term bottleneck analysis in a flexible manufacturing system. Therefore, the accuracy is frequently inadequate to determine the bottleneck, and only a group of possible bottlenecks can be determined. Furthermore, even if the machine with the largest utilization can be determined, it is not necessarily the bottleneck [14, 15]. For the same reasons, this method is also frequently unable to distinguish between secondary bottlenecks. Finally, some machines may have no significant effect on the throughput at all, yet may have a nonzero utilization, adding further complications to this bottleneck detection method. And finally, due to the large sets of data needed to obtain a valid utilization, this method is clearly incapable of detecting the momentary bottleneck. In summary, the utilization is a flawed measure of the bottleneck, not only giving imprecise but also occasionally wrong results.

The second frequently used method for bottleneck detection is based on the waiting time of the parts, or alternatively on the queue length in front of a machine. The machine whose queue has the longest waiting time or the largest number of parts waiting is supposed to be the bottleneck. However, frequently it is not possible to measure the queue length or the waiting time due to system limitations. For example, if there is no buffer, then there are no parts waiting, and a waiting time cannot be determined. Even if there is a buffer, the capacity of this buffer is often limited, making conclusions onto the bottleneck behavior difficult. In some cases, there is a joint buffer for multiple machines, or multiple buffers for separate parts leading to a single machine, making it difficult to analyze the bottleneck. The queue length and waiting time may also fluctuate frequently, causing some bottleneck detections to be invalid. This fluctuation may also be increased due to outside effects as for example the batching of parts. If the parts arrive in batches, the queue length or waiting time peaks in synchronization with the batches, giving incorrect bottleneck detections.

In summary, neither the utilization nor the queue length/waiting times are very suitable tools for the bottleneck detection. The following two sections present a new bottleneck detection method, overcoming the problems of the conventional methods.

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.

(Insert Table 1 about here)

By grouping the machine states into active and inactive, the uninterrupted active period can be measured. It is important to note that the completion of one task (e.g. working on a part) and the subsequent start of a new task (e.g. working on another part or a tool change) is not an interruption but rather a continuous active period. A completion of a task is not an interruption if the next task is started immediately. Thus, the active period may extend over a number of different produced parts, tool changes or repair times until the machine is interrupted by an idle or blocking period.

The Momentary Bottleneck

The underlying idea is that the longer a machine is working without interruption, the more likely it is that this machine constrains the performance of other machines. Therefore, 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. The following method describes how to determine which machine of a production system is the sole bottleneck or part of a shifting bottleneck 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.

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. 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, and the shifting of the bottleneck can be monitored over time.

Figure 1 visualizes 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*. 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. Similarly, at the end of the bottleneck period of M2, the bottleneck shifts back to M1.

(Insert Figure 1 about here)

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. 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 2 visualizes this method using the example with two machines as shown in Figure 1. The percentages of the machines being the sole bottleneck or the shifting bottleneck have been measured over the period of time shown in Figure 1. M1 is more likely to be the bottleneck than M2, and therefore is the main bottleneck. Yet, M2 is also sometimes the bottleneck, although less likely than M1, and therefore is a secondary bottleneck. Overall, an improvement of the performance of M1 would yield a larger overall improvement of the system than an improvement of M2.

(Insert Figure 2 about here)

COMPUTATIONAL EXAMPLES

This section will describe two computational examples. The first example is a flow shop with four stations each, taken with small modifications from [16]. The second example is a complex branched system with seven machines and two different part types.

Lawrence et al. 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 [16]. 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.

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

The method was implemented as software tool GAROPS Analyzer to analyze the simulation data from the GAROPS simulation software as shown in [17] and [18]. The software tool analyses the machine status 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.

Flow Shop

The flow shop example has an exponential inter-arrival rate with a mean inter-arrival time of 1.25s. The processing times of the four machines has an exponential distribution with a mean service rate 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. Figure 3 shows the layout of the flow shop system.

(Insert Figure 3 about here)

The simulation was run for 120 000s, of which a warming up period of 20 000s was removed. The results of the analysis using the GAROPS Analyzer are shown in Table 2. 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 visualized in Figure 4, including the confidence intervals with a confidence level of 95%.

(Insert Table 2 about here)

(Insert Figure 4 about here)

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. However, due to random variations, machines M1, M2 and M4 are also occasional bottlenecks, although to a lesser extent than machine M3. Therefore, an improvement of the machines M1, M2 and M4 will also improve the overall system performance, although to a lesser extent than M3. The shifting bottleneck detection method was also applied to a job shop example with similar results.

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 practically identical with the flow shop example. The layout of the system is given in Figure 5.

(Insert Figure 5 about here)

Using the same settings as the example by Lawrence and Buss [16], 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. 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 column 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 visualized in Figure 6, including the confidence intervals with a confidence level of 95%.

(Insert Table 3 about here)

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.

(Insert Figure 6 about here)

Complex Example

The complex example consists of a branched system with seven machines and two different part types as shown in Figure 7, including different buffers. The simulation was run for 200 000s, of which the warming up period was removed.

(Insert Figure 7 about here)

Figure 8 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. Statistically it is not known if M3 or M5 has the larger utilization, and the primary bottleneck cannot be determined. Due to the small differences in utilization it is difficult to detect the primary bottleneck by measuring the utilization, let alone secondary and tertiary bottlenecks.

(Insert Figure 8 about here)

Table 4 and Figure 9 show the result of the bottleneck detection using the active period. Here 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. Calculating the 95% confidence intervals reveals that the results are statistically significant and M5 is indeed the bottleneck. This example also indicates that M3 is a potential secondary bottleneck and M7 is a potential tertiary bottleneck. Figure 9 includes the confidence intervals with a confidence level of 95%.

(Insert Table 4 about here)

(Insert Figure 9 about here)

In summary, an improvement of the performance M5 would improve the overall system performance. Machines M3, M7 and M2 may also be considered for improvements depending on the trade-off between the cost of the improvement and the benefit of the improved system performance. Furthermore, the bottleneck analysis determines that an improvement of M1, M4 and M6 is unlikely to increase the system performance, and no resources should be invested into an improvement of M1, M4 and M6 at this time.

CONSTRAINT MANAGEMENT TECHNIQUES

The presented bottleneck detection method is very well suited for detecting both short term and long-term bottlenecks in almost any manufacturing system. Not only does the method clearly identify the main bottlenecks, it also determines if a machine is rarely a bottleneck or no bottleneck at all. This allows the use of this method for a multitude of constraint management techniques. These constraint management techniques are aimed to improve the throughput of the system or reduce the cost while maintaining the throughput. A selection of these techniques is presented below.

Improve Bottlenecks

The most preferred way to improve the system throughput is to improve the throughput of the main bottlenecks. Taking the complex manufacturing example as presented above and shown in Figure 9, the main bottleneck is clearly machine M5. Therefore, improving the throughput of the main bottleneck M5 will also improve the throughput of the entire system. The exact improvement of course depends on the nature of this machine, as for example an increase in cutting speed or the replacement of the machine with a higher quality machine, but there are also general options, as for example scheduling the breaks of the operators in a way that the machine is operated continuously.

Looking closer at Figure 9, it can also be seen that machines M3 and M7 are sometimes the bottleneck, although not as often as machine M5. Therefore, improving machines M3 and M7 will not have the same effect as an improvement of machine M5, yet if there are cost effective ways to improve these machines, it may be economically feasible to also improve these machines. This will result in a better supply of parts to the main bottleneck Machine 5 and a resulting increased utilization of machine M5, which is currently at 94%.

Reduce Non-Bottlenecks

The shifting bottleneck detection method also determines which machines are non-bottlenecks. In the above example in Figure 9, machines M1 and M4 were no bottlenecks at all. This, in turn, may be used to reduce the cost at these machines by reducing the machine throughput without decreasing the system throughput. Again, the exact change depends on the nature of the machine, but for example it might be possible to reduce the cutting speeds of a cutting machine in order to increase the tool life. In comparison, conventional bottleneck detection methods are unable to point out machines for possible cost cutting.

Schedule around Bottlenecks

In many manufacturing systems, different products are produced at the same time. These products may require different processing times on different machines. Using the bottleneck detection method in combination with simulation techniques, it is possible to determine the different bottlenecks for the different products. Subsequently, it is possible to schedule the different products in a way to avoid an accumulation of the productwise bottlenecks on the same time at the same machine. Instead, the schedule can be arranged to have a more evenly distributed utilization across the machines. Further research in this area is in progress.

Buffer Main Bottlenecks

Usually one of the fastest and easiest methods to improve a manufacturing system is the addition of buffers. These buffers aim to provide a steady supply of parts to the production machine, increasing the utilization of these machines. The shifting bottleneck detection method is able to determine the main bottleneck. Subsequently, it is desirable to improve the utilization of the main bottleneck. In the above complex example, machine M5 is the main bottleneck, yet this machine has a utilization of only 94%. This means, that adding buffers before and after machine M5 can improve the utilization by about 6%, i.e. the machine can produce 6% more parts in a given time. As this machine is the bottleneck, this machine throughput improvement will also yield an overall system throughput improvement.

SUMMARY

The active period method has many advantages over other methods for bottleneck detection. For example, the measurement of the queue length or waiting time in order to detect the bottleneck cannot be used if the queue lengths are limited. In addition, the queue length may fluctuate frequently, complicating a reallocation of the resources in a “chase the bottleneck” approach. Using the utilization as a bottleneck detection method may give inaccurate results for the detection of the primary bottleneck, and it is usually impossible to detect secondary and tertiary bottlenecks.

The active period method as presented in this paper, however, is a very flexible tool and can be used for a wide range of job shop and flow shop systems as for example production systems, computer networks or traffic systems. The method is easy to apply, and the required data is usually readily available. 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. Both, short term and long term average bottlenecks can be detected. For non steady state systems there is no long-term average bottleneck. However, the likelihood of a machine being a bottleneck during the analyzed period of the non steady state system can be determined.

Knowing the likelihood of each machine to be the bottleneck aids the manager in making a trade-off between the effort of adding capacity to the machines and the benefits of improved throughput.

Research is in progress to adapt the active period method for the optimization of the production systems.

REFERENCES

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

(2) Goldratt, E.M., *The Goal: A Process of Ongoing Improvement*. 2nd ed. 1992: North River Press. 351.

(3) Cox, J.F.I. and M.S. Spencer, *The Constraints Management Handbook*. 1997, Boca Raton, Florida: CRC Press - St. Lucie Press. 300.

(4) Law, A.M. and D.W. Kelton, *Simulation Modeling & Analysis*. 3 ed. 2000: McGraw Hill.

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

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

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

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

(9) Roser, C., M. Nakano and M. Tanaka. *A Practical Bottleneck Detection Method*. in *Winter Simulation Conference*. 2001. Arlington, Virginia, USA: Institute of Electrical and Electronics Engineers.

(10) Roser, C., M. Nakano and M. Tanaka. *Detecting Shifting Bottlenecks*. in *International Symposium on Scheduling*. 2002. Hamamatsu, Japan.

(11) Schmeiser, B.W. and W.T. Song. *Batching methods in simulation output analysis: what we know and what we don't*. in *Winter Simulation Conference*. 1996. San Diego, CA USA.

(12) Roser, C. and M. Nakano. *Single Simulation Confidence Intervals using the Delta Method*. in *International Symposium on Scheduling*. 2002. Hamamatsu, Japan.

(13) Roser, C. and M. Nakano, *Confidence Intervals from a Single Simulation using the Delta Method.* International Journal of the Japan Society of Mechanical Engineering, Special Issue Advanced Scheduling, 2003.

(14) Roser, C., M. Nakano and M. Tanaka. *Utilization vs. Throughput: Bottleneck Detection in AGV Systems*. in *Logistic System Symposium*. 2002. Nagoya.

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

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

(17) Kubota, F., S. Sato and M. Nakano. *Enterprise Modeling and Simulation Platform Integrated Manufacturing System Design and Supply Chain*. in *IEEE Conference on Systems, Man, and Cybernetics*. 1999. Tokyo, Japan.

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

TABLES

Table 1: Active – Inactive Examples

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

Table 2: Flow Shop Simulation Results

Machine	Utilization	Bottleneck	Shifting	Sum
M1	80.1%	12.7%	20.4%	33.1%
M2	80.2%	6.7%	15.9%	22.7%
M3	88.0%	32.5%	29.3%	61.8%
M4	80.0%	7.3%	15.2%	22.5%
Shiftiness Measure β		0.59	0.84	0.74

Table 3: Job Shop Simulation Results

Machine	Utilization	Bottleneck	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

Table 4: Complex Example Simulation Results

Machine	Utilization	Bottleneck	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

FIGURES

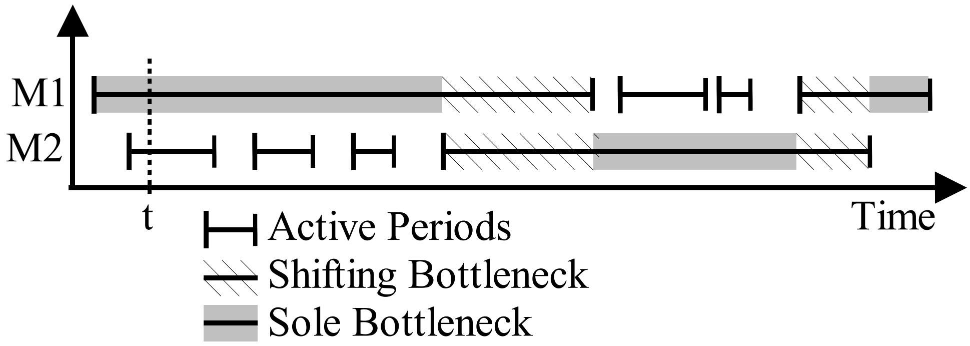


Figure 1: Shifting Bottlenecks

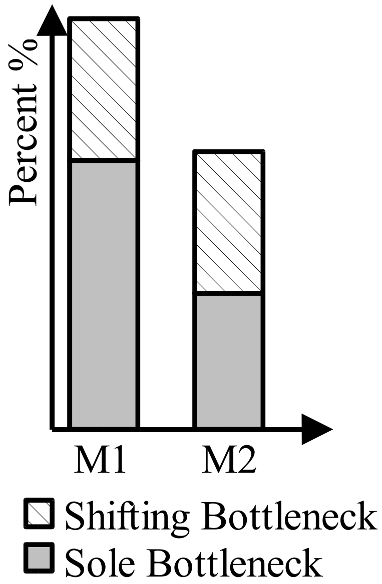


Figure 2: Average Bottleneck over Period of Time

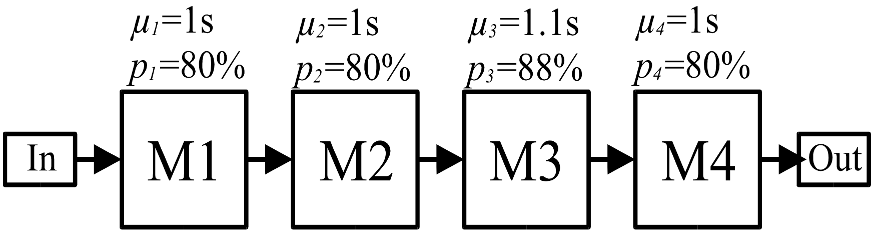


Figure 3: Flow Shop Layout

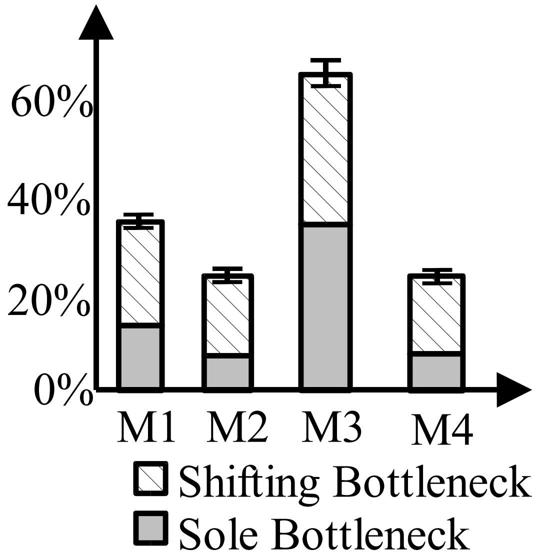


Figure 4: Flow Shop Bottlenecks

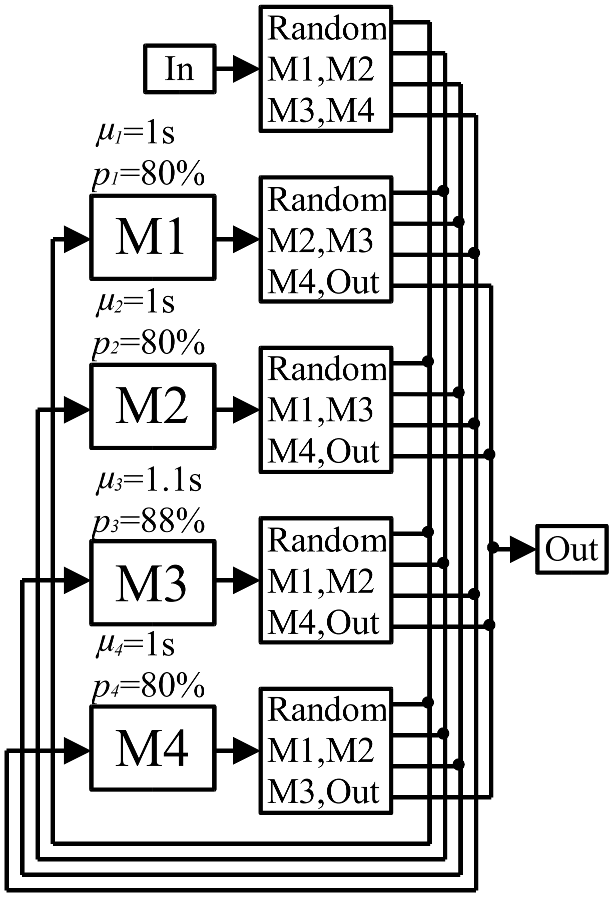


Figure 5: Job Shop Layout

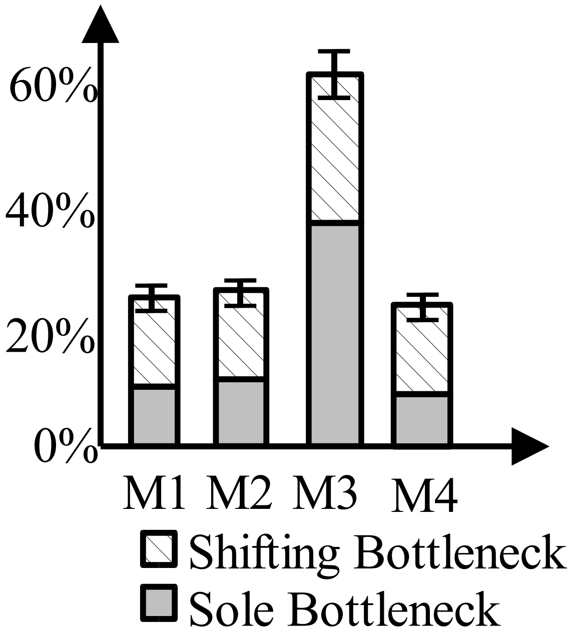
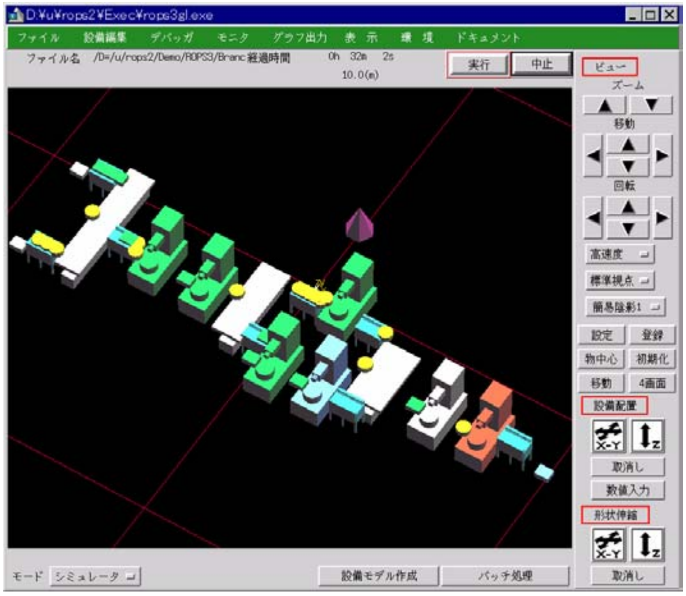


Figure 6: Job Shop Bottlenecks


D:¥u¥rops2¥Exec¥rops3gl.exe

ファイル名 /D=/u/rops2/Demo/ROPS3/Branc 経過時間 0h 32m 2s
10.0(m)

モード シミュレータ

Figure 7: GAROPS Screenshot

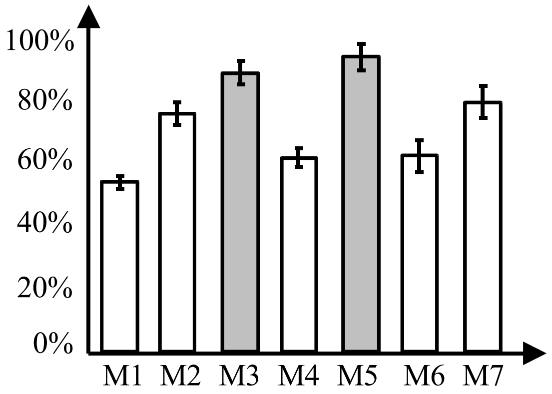


Figure 8: Utilization of Complex Example

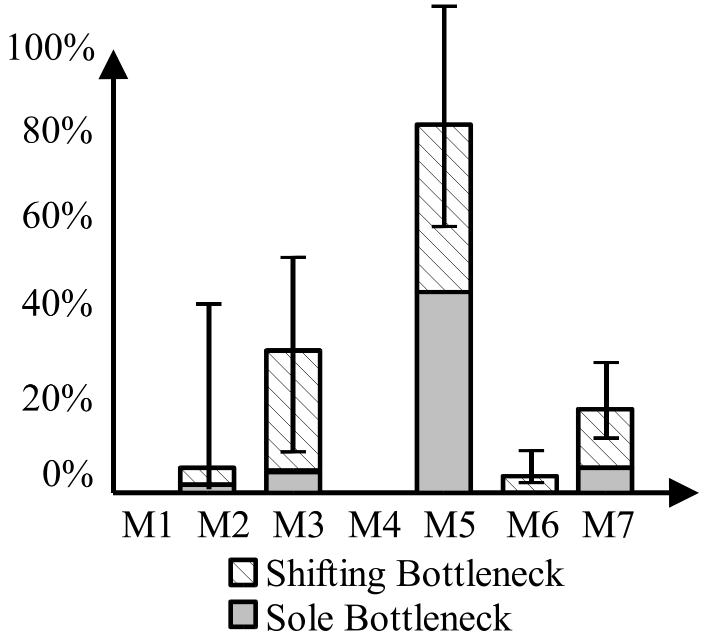


Figure 9: Complex Example Bottlenecks