On Transfer Blocking and Minimal Blocking in Serial Manufacturing Systems — The Impact of Buffer Allocation

Alexander K. Schömig
Bayerische Julius–Maximilians Universität Würzburg,
Institut für Informatik, Lehrstuhl für Informatik III
Am Hubland, D-97074 Würzburg, Germany
Tel.: +49-931-8885511, Fax: +49-931-8884601
e-mail: schoemig@informatik.uni-wuerzburg.de

Abstract

In this paper we investigate the impact of buffer allocation in an unbalanced, asynchronous production system consisting of 12 stages arranged in series. We consider finite buffer tandem queue like arrangements as well as the singlecard kanban system. The first belongs to the class of manufacturing systems operating in *push* mode, while the latter is a means for the implementation of the *just–in–time* (JIT) philosophy. We discuss how given parameters influence the performance of the systems under consideration. Finally, we present guidelines how designers and operations managers of a flow shop manufacturing system can maximize throughput while keeping WIP and the cycle time at a low level.

Keywords

Tandem queues, kanban system, just–in–time, simulation

1 INTRODUCTION

Production systems often consist of a number of stages arranged in series. Each job is released to the first stage of the system and undergoes processing at each stage. The performance of a production system is mainly determined by variations in the service times at each stage. The higher those variations are, e.g. due to breakdowns of equipment, operator unavailability, temporal material shortage, or rework, the lower is the production rate, i.e. the amount of products finished in a given time period. A common means of increasing the production rate is providing buffer space between production stages.

This research was supported by the Deutsche Forschungsgemeinschaft (DFG), under grant #Tr 257/2–1.

These buffers decouple adjacent stages and reduce the probability of blocking the flow of material. However, increasing the buffer space means also increasing the work in process (WIP), i.e. the sum of all workpieces currently present in the production system.

Manufacturers in Europe and the U.S.A. traditionally organized their production systems as *push systems*. To protect the system against the consequences of incorrect forecasts, in–process safety stocks are often build up. Utiliziation of resources and throughput were considered as the key criteria in the design and operation of a production process.

During the past years much attention has been devoted to just–in–time (JIT) manufacturing systems. In the philosophy of JIT, safety stocks are seen as a unnecessary waste of resources, since with each workpiece costs for raw material, storage, and handling are involved. Additionally, problems in the production process are usually hidden by a high in–process inventory level. One way to implement a JIT–system, also called *pull system*, is the Japanese *kanban system*. Control cards, called 'kanban', circulate within a production stage and their occurence at specific positions indicate the status of this stage.

2 LITERATURE REVIEW

Much of the research on serial manufacturing systems is based on queueing network models. Buzacott and Shantikumar (1993) as well as Askin and Standridge (1993) summarize the state of research using finite buffer tandem queues. Most noteable are the efficient decomposition and calculation algorithms presented by Gershwin (1991) and Dallery and Frein (1993). Unfortunately, often unrealistic assumptions like exponentially distributed service times, unlimited buffer space, or only two or three stages that are in tandem have to be made for the sake of mathematical tractability.

The amount of literature on JIT and kanban systems has grown vast during the past decade. Golhar and Stamm (1991) list more than 200 references to research papers and books on JIT. Berkley (1992) reviews the the kanban production control research literature. He classfies the kanban models according to certain system features such as blocking mechanism and material handling. The kanban system used in this study belongs to the category of pull production systems as previously presented by DiMascolo et al. (1992), and Mitra and Mitrani (1990). The kanban blocking mechanism of these models is given by a single constraint which limits the input and output buffers of a production stage in contrast to two–card kanban systems. So and Pinault (1988) addressed the problem of buffer space, i.e. kanban allocation, in singlecard kanban systems using a queueing network approximation technique. Schömig and Gihr (1993) presented a peculiar kanban allocation scheme that reduces WIP while maintaining the throughput of the kanban system.

There are only few similar studies like the one presented here that compare the performance of a finite buffer tandem queue model with an equivalent kanban system. El–Rayah (1979) investigated the efficiency of balanced and unbalanced production lines taking into consideration some early results, e.g. Hillier and Bolling's (1966) notion of the *bowl phenomenon*. Berkley (1991) compared the two–card kanban system appearing in the production literature to the standard tandem queueing model. He showed that a large number of kanban systems, but not all, are equivalent to a tandem queue. Sarker and Fitzsimmons (1989) investigated the effects of production variabilities on the performance of kanban and tandem queue systems by means of simulation. Their results were doubted by Barker et al. (1990) and they stated that Sarker and Fitzsimmons' results were flawed.

To the authors' knowledge this is the first study that compares a kanban system with finite buffer tandem queues paying attention to possible interactions of factors that could be important for the question of buffer space allocation between production stages.

3 MODELS OF SERIAL MANUFACTURING SYSTEMS

Serial manufacturing systems are usually modeled as finite buffer tandem queue networks. Each machine is represented by a server and the storage space in front of or after the machine are represented by a finite queue. In a crude conception of such a manufacturing flow line consisting of N machines, raw parts enter the facility from an inexhaustible source, move from machine #1 to machine #N, undergo processing at each machine, and leave the manufacturing line to meet an insatiable demand. The processing times on each machin are assumed to be random to account for variability due to setup times, downtimes, material shortage, and/or operator unavailability and the like.

First we look at the standard flowline–like layout with limited buffers: If a part cannot proceed to the next machine because the downstream buffer is full, this part remains in the machine and blocks it until buffer space becomes available. If so, this part moves to the downstream buffer and releases the machine. This can now resume service with the next available part. This blocking mechanism is called *transfer blocking*. Figure 1 shows our model consisting of 12 machines and one feeder machine. Note, that this model is coherent to previous studies, e.g. by Dallery and Frein (1993).

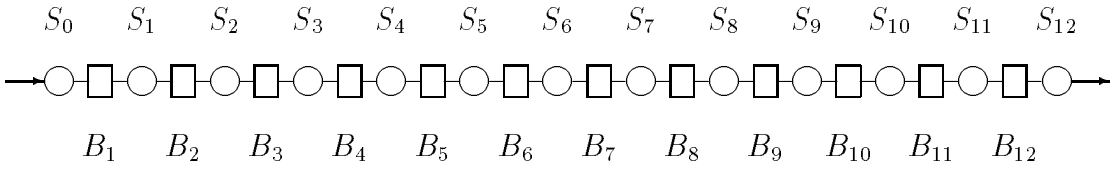

Figure 1: Model of a manufacturing line consisting of 12 machines and one feeder machine with finite buffers

We use the model of a singlecard kanban system as introduced by Mitra and Mitrani (1990), p. 1555, Fig. 4. We only sketch the basic mechanism and refer to the original paper for details. The production line is partitioned into stages. In the trivial case each stage consists of only one machine, but stages may also contain several machines. There is storage space for a limited number of parts awaiting processing or authorization to move to the next stage. C_i kanbans are allocated to stage i . These kanbans are stored in the *bulletin board (BB)*. A part may only enter a stage when there is a kanban available from the BB. In this case, a kanban is taken from the BB and attached to the entering part; this kanban is not removed until this part leaves the stage. Thus, the total number of parts in stage i is limited to C_i. For this reason, the total number of parts in the line may not exceed $\sum_{i=1}^{N} C_i$. Upon completion of processing in stage i, a part and its attached kanban proceed to the *output hopper (OH)* of stage i. Two possible actions can now take place: If the BB of stage $i+1$ is empty this part has to wait in the OH until a free kanban of stage $i+1$ becomes available. This only can occur, when a part leaves stage $i+1$. If there is a free kanban in the BB of stage $i+1$, the part may leave stage i and enter

Table 1 Mean service times

Server:	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
'–'	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
'/'	0.64	0.67	0.70	0.73	0.76	0.79	0.82	0.85	0.88	0.91	0.94	0.97	1.00
'\'	1.00	0.97	0.94	0.91	0.88	0.85	0.82	0.79	0.76	0.73	0.70	0.67	0.64
'∨'	1.00	0.97	0.94	0.91	0.88	0.85	0.82	0.85	0.88	0.91	0.94	0.97	1.00
'∧'	0.82	0.85	0.88	0.91	0.94	0.97	1.00	0.97	0.94	0.91	0.88	0.85	0.82

stage $i+1$, after returning its kanban to the BB of stage i. This kanban is replaced by one of stage $i+1$. The part goes to the input hopper of stage $i+1$ and awaits processing. We assume that the actions and movements of parts and kanbans described above are done without any delay. Mitra and Mitrani (1990) refer to the blocking mechanism introduced by their system as *minimal blocking*.

4 METHODOLOGY

Research Objective

Sarker and Fitzsimmons (1989) point out that the findings of previous studies are contradictory. In view of this fact and the lack of reliable work on the performance of pull *and* push systems, we defined the following main objectives. (1) How does a kanban system compare to a finite buffer tandem queue? (2) Does the interaction of factors influence the buffer space allocation scheme? (3) Do findings for adviseable buffer space allocation schemes for kanban systems hold for finite buffer tandem queues?

Simulation Parameters

The simulation programs of the model were coded in MODULA–2, using special simulation subroutines. For each parameter set 15 independent replications of a simulation run of the system under consideration were conducted. Each run had a total length of 2,000 time units and statistical data from the initial transient period were discarded according to the recommendations given by Law and Kelton (1991). We observed three performance measures: System throughput ('Thp'), cycle time, and WIP. All measures are normalized to one time unit. For each result the 95% confidence interval of the mean value was calculated. The columns containing the results in the following tables from these simulation runs also show the lower and upper bound of the confidence interval, in the graphs, the confidence intervals are depicted appropiately.

Experimental Design

According to our research objectives, we selected three experimental factors, i.e. factors that may unbalance the line: (1) service times, (2) variation of the service times, and (3) buffer space allocation and kanban stage segmentation.

Table 1 lists five basic patterns how we defined the mean service times: '–' denotes the trivial case where each machine has the same mean service time, '/' and '\' denote the cases when the machines are order in acending order of the service times and, respectively

Table 2 Coefficient of variation of service times

Server	S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
'/'	0.70	0.75	0.80	0.85	0.90	0.95	1.00	1.05	1.10	1.15	1.20	1.25	1.30
'\'	1.30	1.25	1.20	1.15	1.10	1.05	1.00	0.95	0.90	0.85	0.80	0.75	0.70
'$\wedge$'	0.85	0.90	0.95	1.00	1.05	1.10	1.15	1.10	1.05	1.00	0.95	0.90	0.85
'$\vee$'	1.15	1.10	1.05	1.00	0.95	0.90	0.85	0.90	0.95	1.00	1.05	1.10	1.15

falling service times. Finally we consider the bowl–shaped pattern, '$\vee$', and the reverse bowl, '$\wedge$'. Note that the biggest mean service time for each configuration is always 1.00. Hence, the machine with a mean service rate of 1.00 is the bottleneck.

To account for possible variations of the service times we assumed in the simple case exponentially distributed service times, i.e. we have a coefficient of variation (c.v.) of 1.00. We also defined certain patterns of parameter assignment for the c.v.s analogous to before. The same symbols apply. The actual data is displayed in Tab. 2.

In the following we denote the basic finite buffer tandem queue system as '**A**'. The notion of kanban stage *segmentation* is the grouping of machines into stages. In the most simple case, here denoted '**B**1', there is only one machine within a stage. This configuration is often called *operational kanban*. We also consider sytems, where we have equal sized stages with two (system '**B**2'), three ('**B**3'), four ('**B**4'), six ('**B**5'), or 12 machines ('**B**6') in each stage. The last case is equivalent to the *closed loop principle*. The feeder machine is in any case a single machine in a kanban stage.

5 DISCUSSION OF RESULTS

Due to the size of this paper we are only able to present few results chosen from the multitude of possible configurations and parameter sets.

Figure 2 shows the throughput of system '**A**' and the kanban system under different

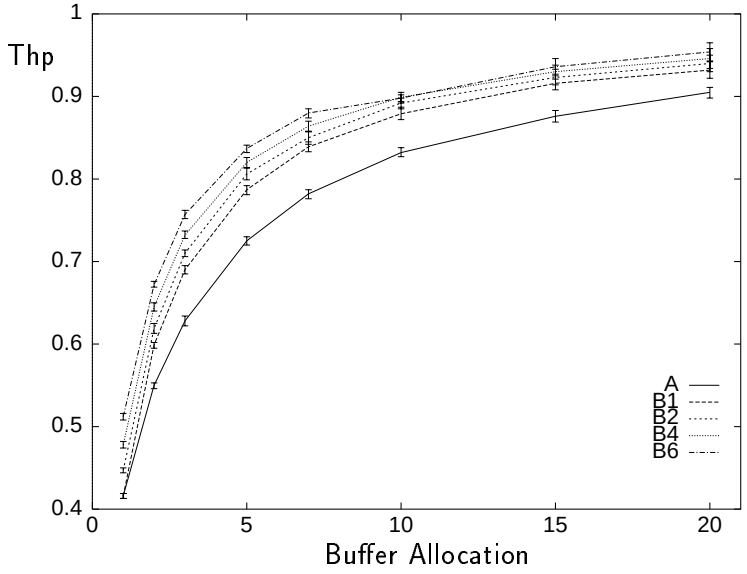

Figure 2: Throughput of several system configurations

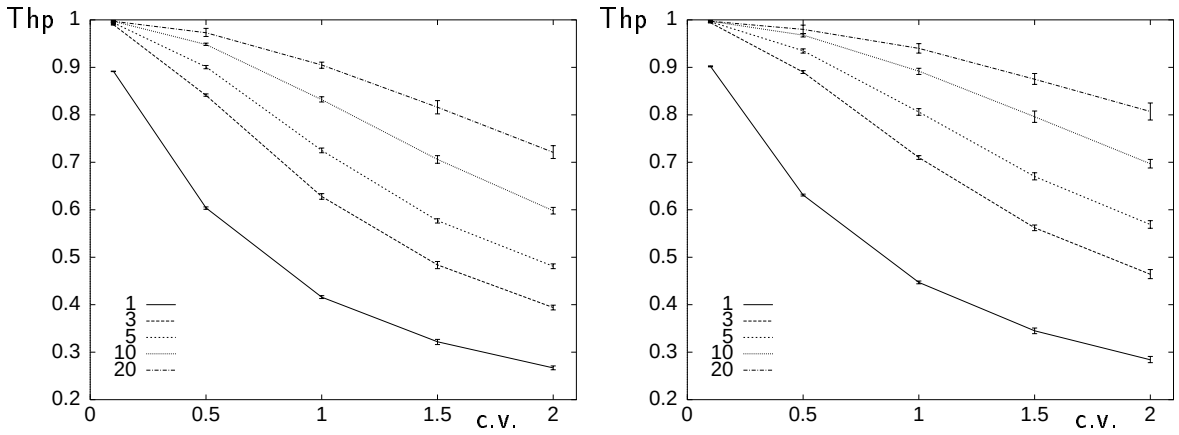

Figure 3: Throughput of system '**A**' and system '**B**2' for several c.v.'s of the service time distribution, mean service time pattern '–'

segmentations depending on the buffer size. Given are parameter sets '–' for mean service times and c.v.'s. Note, that the maximal throughput of these configurations equals 1.00, the mean service time of all machines. The curves exhibit the typical form of the throughput of serial manufacturing lines with finite buffers and idendical machines as discussed by Askin and Standridge (1993), pp. 87. However, we observe a significant difference in the performance of these systems: The closed loop mechanism, system '**B**6' achieves the highest throughput, while system '**A**' performs worst.

Figures 3 *a*) and *b*) compare the throughput of system '**A**' and system'**B**2'. Again, we find the same *qualitative* behaviour of both systems, but system '**A**' performs worse than the kanban system for all values given for c.v. and buffer size. We conjecture that findings for kanban systems hold for tandem queue networks when the respective parameters are considered.

Figures 4 and 5 show the throughput of system '**B**2' when unbalanced due to varying service times or c.v.'s. The black boxes on top of the columns denote the confidence interval. It is obvious that reverse–bowl buffer allocation only achieves a suboptimal throughput in case of a unbalanced line. However, the best results are obtained when more buffer space is provided to the bottleneck machines.

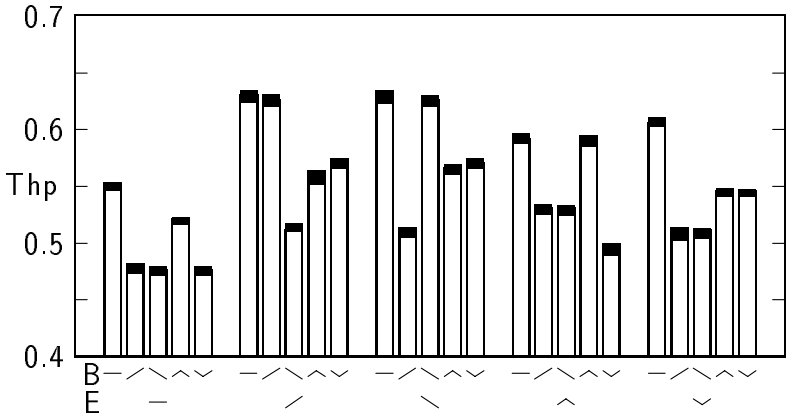

Figure 4: Impact of buffer allocation ('B') when the line is unbalanced by variations in mean service times ('E')

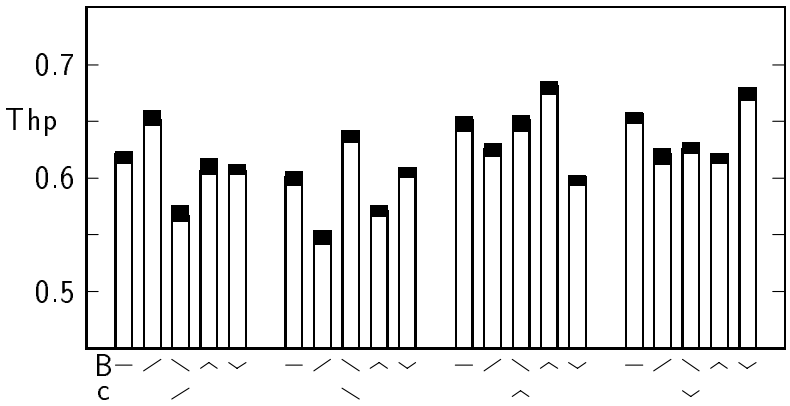

Figure 5: Impact of buffer allocation ('B') when the line is unbalanced by variations in coefficients of variation ('c')

Table 3 Unbalancing by buffer space variations of system **A**

A	Buffer space												
	B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}
'–'	1	5	5	5	5	5	5	5	5	5	5	5	5
E 1	1	4	4	4	4	5	5	5	5	5	5	5	5
E 2	1	3	3	4	4	4	5	5	5	5	5	5	5
E 3	1	2	3	3	4	4	5	5	5	5	5	5	5

Finally, we consider some special buffer allocation patterns for system '**A**' as proposed for kanban systems by Schömig and Gihr (1993). Table 4 contains the allocated buffer sizes, Tab. 5 the results obtained. If there are little reductions in the buffer space of the machines at the beginning of the line, throughput is reduced only by a few percent, but WIP and cycle times are cut drastically.

6 CONCLUSION

In this paper we have discussed the problem of buffer allocation in an unbalanced, asynchronous production system. In contrast to common expectation, according to our findings the 'bowl pattern' achives only suboptimal results in a long serial manufacturing line. Ne-

Table 4 Results for buffer space variations

	Thp			WIP			Cycle Time		
'–'	0.752	0.756	0.760	40.63	41.78	42.93	53.64	55.27	56.90
E 1	0.740	0.746	0.752	36.18	37.16	38.14	48.48	49.82	51.16
E 2	0.726	0.732	0.738	31.68	32.52	33.36	43.45	44.45	45.46
E 3	0.692	0.695	0.698	25.39	26.18	26.97	36.55	37.61	38.67

vertheless, we are able to formulate three guidelines for buffer allocation:
(1) Buffer space of the buffers of the first half of the line should not exeed the buffers of the second half. (2) Additional buffer space should always first be provided for the bottleneck machines. (3) The buffer space allocation pattern should not be decending, however, there is no gain if there is more buffer at the backend machines of the line than in the middle part.

Certainly we do not deny the fact that the actual buffer allocation may depend on peculiarities of the manufacturing system under consideration. Last but not least operation managers have to find in any case a tradeoff between throughput and cycle time or WIP.

REFERENCES

Askin, R. G. and Standridge, C. R. (1993). *Modeling and Analysis of Manufacturing Systems*. John Wiley & Sons, Inc., New York, NY.

Baker, K. R., Powell, S. G., and Pyke, D. F. (1990). The performance of push and pull systems: A corrected analysis. *Int. Journal of Prod. Res.*, **28**(9):1731–1736.

Berkley, B. J. (1991). Tandem queues and kanban-controlled lines. *Int. Journal of Prod. Res.*, **29**(10):2057–2081.

Berkley, B. J. (1992). A review of the kanban production control research literature. *Prod. and Oper. Management*, **1**(4):393–411.

Buzacott, J. A. and Shanthikumar, J. G. (1993). *Stochastic Models of Manufacturing Systems*. Prentice-Hall, Englewood Cliffs, NJ.

Dallery, Y. and Frein, Y. (1993). On decomposition methods for tandem queueing networks with blocking. *Oper. Res.*, **41**(2):386–399.

Di Mascolo, M., Frein, Y., and Dallery, Y. (1992). Queueing network modeling and analysis kanban systems. In *Third Int. Conference On Computer Intergrated Manufacturing*. IEEE / Rensselaer Polytechnic Institute, IEEE Computer Society Press.

El-Rayah, T. E. (1979). The efficiency of balanced and unbalanced production lines. *Int. Journal of Prod. Res.*, **17**(1):61–75.

Gershwin, S. B. (1991). Assembly/disassembly systems: An efficient decomposition algorithm for tree-structured networks. *IIE Transactions*, **23**(4):302–314.

Golhar, D. Y. and Stamm, C. L. (1991). The just-in-time philosophy. *The Int. Journal of Prod. Res.*, **29**(4):656–676.

Hillier, F. S. and Boling, R. W. (1966). The effect of some design factors on the efficieny of production lines with variable operation times. *The Journal of Industrial Engineering*, **17**(12):651–658.

Law, A. M. and Kelton, W. D. (1991). *Simulation Modeling & Analysis*. McGraw–Hill, New York, 2nd edition.

Mitra, D. and Mitrani, I. (1990). Analysis of a kanban discipline for cell coordination in production lines. *Management Science*, **36**(12):1548–1566.

Sarker, B. R. and Fitzsimmons, J. A. (1989). The performance of push and pull systems: a simulation and comparative study. *Int. Journal of Prod. Res.*, **27**(10):1715–1731.

Schömig, A. and Gihr, O. (1993). Leistungsuntersuchung von Schedulingverfahren in Produktionssystemen. In Walke, B. and Spaniol, O., editors, *Messung, Modellierung und Bewertung von Rechen– und Kommunikationssystemen. Kurzberichte und Werkzeugvorstellungen*, pages 74–85. 7. ITG/GI–Fachtagung Aachen. (In German.)

So, K. C. and Pinault, S. C. (1988). Allocating buffer storages in a pull system. *Int. Journal of Prod. Res.*, **26**(12):1959–1980.