Analysis of a Generalized Shortest Queue System by Flexible Bound Models

G.J. van Houtum
Faculty of Mechanical Engineering
University of Twente
E-mail: g.j.j.a.n.vanhoutum@wb.utwente.nl

Abstract

Motivated by a practical situation for the production/assembly of Printed Circuit Boards, we study a generalized shortest queue system. This system consists of parallel servers, which all have their own queue. The system serves several types of jobs, which arrive according to Poisson processes. Because of technical reasons, most or all types of arriving jobs can only be served by a restricted set of servers. All jobs have the same exponential service time distribution, and, in order to minimize its own service time, each arriving job joins (one of) the shortest queue(s) of all queue(s) where the job can be served. The behavior of the resulting queueing system may be described by a multi-dimensional Markov process. Since an analytical solution for this Markov process is hard to obtain, we present flexible bound models in order to find the most relevant performance measures, viz. the waiting times for each of the job types separately and for all job types together. The effectiveness of the flexible bound models is shown by some numerical results.

1 Introduction

To show the relevance of the queueing system studied in this paper, we first describe a queueing situation stemming from a flexible assembly system consisting of a group of parallel insertion machines, which have to mount vertical components on Printed Circuit Boards. We start the description with explaining how an *insertion machine* operates. An insertion machine mounts vertical components, such as resistors and capacitators, on a Printed Circuit Board (PCB) by the *insertion head.* The components are mounted in a certain sequence, which is prescribed by a Numerical Control program. The insertion head is fed by the *sequencer*, which picks components from tapes and transports them in the right order to the insertion head. Each tape contains only *one* type of components. The tapes are stored in the *component magazine*, which may contain 80 tapes, say. Each PCB needs, on average, 60 different types of components. If a machine has to mount components on a PCB, then all the components need to be available on that machine. That means that for all those components a tape must be placed in the magazine. So the set of components available on the machine completely determines which types of PCBs can be handled.

In general we have a group of parallel insertion machines which have to process a number of different types of PCBs at the same time. Each insertion machine has its own queue, and the PCBs are transported to the insertion machines by an Automatic Conveyor System. In Figure 1, we have depicted a system which consists of three insertion machines and which has to process three different types of PCBs. The machines are basically similar, but due to the fact that they may be loaded with different types of components, the classes of PCB-types that can be handled by the machines may be different. In the situation depicted in Figure 1, machine M_1 can handle PCBs of the types A and B, machine M_2 can handle the types A and C, and machine M_3 can handle the types B and C.

In fact, there are two decision problems: the *assignment problem* and the *routing problem.* We first describe the assignment problem, which is the major problem. The assignment problem concerns how the tapes with components have to be divided among the machines. One should try to allocate the tapes with components to the machines such that, for example, the waiting times (and/or sojourn times) of the PCBs are minimized. There would be no problem if the magazines were big enough to contain all components needed to process all types of PCBs. However, in general they can only contain the components needed for a small subset of the different types of PCBs.

In order to solve the assignment problem, we must be able to evaluate the performance characteristics of a *given assignment* of the components to the machines. These performance characteristics depend on how the second decision problem, i.e. the routing problem, is handled. This problem concerns to which machines the PCBs must be sent upon arrival. For an arriving PCB, we must select one of the machines which can handle that PCB. If for all types the mounting times are roughly the same, then it is reasonable to select the machine with the *shortest queue* (let ties be broken with equal probabilities); this at least (roughly) minimizes the waiting time of the arriving PCB itself, and it may be expected that this also roughly minimizes the average waiting time for all PCBs together, provided that we are in a balanced situation (i.e. a situation in which each server will have to handle the same amount of work on average). Assume that the shortest queue routing is used by the Automatic Conveyer System, and that, once arrived in a queue, the PCBs are served in a First-Come-First-Served (FCFS) manner. Then we have the following problem:

> *Given the shortest queue routing and the FCFS service discipline at each machine, we want to have an efficient method for the determination of the performance characteristics of the flexible assembly system for a given assignment of the components to the machines.*

The main performance characteristics we are interested in, are the waiting times for each type of PCBs separately and for all PCBs together. It is obvious that an efficient method for determining these measures can be exploited for selecting the best possible assignment of the components to the machines.

The assembly of PCBs is often characterized by relatively few job types, large production batches and small processing times (see Zijm [13]). Therefore, a queueing model approach seems natural. The flexible assembly system can be modeled as a queueing system consisting of parallel servers, each with a own queue, and serving several types of jobs, where each job upon arrival joins the shortest queue of all queues that can handle this job. We call this system a *Generalized Shortest Queue System (GSQS).*

Apart from the situation described above, the GSQS is also relevant for many other practical situations; for example, in a job shop with a group of identical, parallel machines which are loaded with different sets of tools, in a computer system where each information

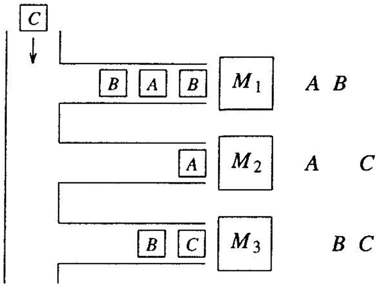

Figure 1: A flexible assembly system consisting of three parallel insertion machines, on which three types of PCBs are made.

file is available on a restricted set of a number of parallel disks and requests for information files have to be handled by only one disk, and at a banking office where each clerk is able to carry out a restricted set of tasks. Nevertheless, the GSQS has hardly been studied in the literature. To our knowledge, the only contribution is made by Adan, Wessels and Zijm [3], who, for a simplified situation (see the next paragraph), present rough approximations for the waiting times. Further, closely related systems have been studied by Schwartz [10] (see also Roque [9]), Green [5] and Hassin and Haviv [6].

In this paper, we make the following assumptions for the GSQS (cf. [3]): (i) all jobs arrive according to Poisson streams; (ii) the service times are exponentially distributed; (iii) the service times are job-independent; (iv) all insertion machines work equally fast. The assumptions (ii)-(iv) imply that all service times are exponentially distributed with the same parameter. Even under these assumptions, the GSQS constitutes a hard problem. The behavior of the GSQS is described by a continuous-time Markov process with multi-dimensional states where each component denotes the queue length at one of the servers. However, because of the shortest queue routing, the structure of the transitions is rather complicated and hence an analytical solution seems hard to obtain in general. In fact, an analytical solution is only known for the special case with two parallel servers and one type of jobs that can be handled by both servers; in this case the GSQS reduces to the two-dimensional symmetric shortest queue system, for which a generalized product-form solution has been derived by using a compensation approach (see [4]). For all other cases, even a standard numerical method is not available. Therefore, for the general case of the GSQS, we propose to use truncation models which: (i) have a truncated state space with a flexible size (i.e. depending on one or more truncation parameters); (ii) can be solved efficiently; (iii) lead to upper/lower bounds for the waiting times. Such models are called *solvable flexible bound models*. We shall define one lower bound and one upper bound model. By solving these two models for increasing sizes of the truncated state space, we can determine the waiting times of the original GSQS as accurately as desired. Numerical results for two series of instances will show that this method may work quite well. It is noted that flexible bound models previously have been successfully applied to the symmetric shortest queue system (with ≥ 2 servers), the symmetric longest queue system and an $M|M|c$ system with critical jobs (see [2, 12, 1]).

This paper is organized as follows. In Section 2, we give a precise description of our model for the GSQS. Next, in Section 3, we describe the flexible bound models that can be used to determine the waiting times for the GSQS. Finally, in Section 4, we present numerical results in order to show the effectiveness of the flexible bound models. For simplicity and in order to save space, in the remaining part of this paper we shall restrict ourselves to the two-dimensional case, i.e. to a GSQS consisting of two servers. Nevertheless, the whole analysis can easily be generalized to the case with two or more servers; for this generalization the reader is referred to [11].

2 Model

We consider a GSQS consisting of two parallel servers. For this system we distinguish three types of jobs: jobs of type A, which can be served by both servers, jobs of type B, which can only be served by server 1, and jobs of type C, which must be served by server 2; see Figure 2. The jobs of the types A, B and C arrive according Poisson processes with intensities λ_A, λ_B and λ_C (all ≥ 0). The total arrival intensity is denoted by $\lambda = \lambda_A + \lambda_B + \lambda_C$. All service times are assumed to be exponentially distributed with parameter $\mu = 1$. Upon arrival, jobs of type B join the queue at server 2, jobs of type C join the queue at server 3, and jobs of type A join the shortest queue (if both queues have equal length, then each queue is chosen with probability $\frac{1}{2}$).

The behavior of the GSQS is described by a continuous-time Markov process with states (m_1, m_2), where m_i denotes the length of the queue at server i, $i = 1, 2$ (jobs in service are included). So, the state space is equal to

$$M = \{m \mid m = (m_1, m_2) \text{ with } m_i \in I\!N_0 \text{ for } i = 1, 2\} .$$

In order to obtain an irreducible Markov process, we assume that $\lambda_A + \lambda_B > 0$ and $\lambda_A + \lambda_C > 0$. The transition rates are denoted by $q_{m,n}$. These rates have been depicted in Figure 3.

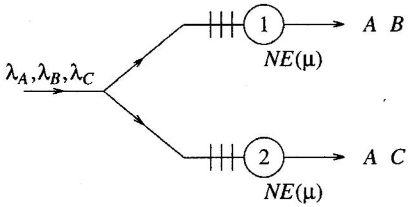

Figure 2: The GSQS with two servers and three job types.

The average workload per server is given by $\rho = \lambda/2$. The GSQS obviously can only be ergodic if $\rho < 1$ and if each of the servers can handle the job type that always has to be served by him, i.e. if

$$\lambda_B < 1, \quad \lambda_C < 1 \quad \text{and} \quad \lambda < 2. \tag{1}$$

We *conjecture* that this condition is not only necessary, but *also sufficient* for the ergodicity. This conjecture is based on: (i) the idea that the *dynamic* shortest queue routing gives a better performance than a *static* routing; (ii) the property that if condition (1) is satisfied, then there exists a static routing under which the system is ergodic. The latter property

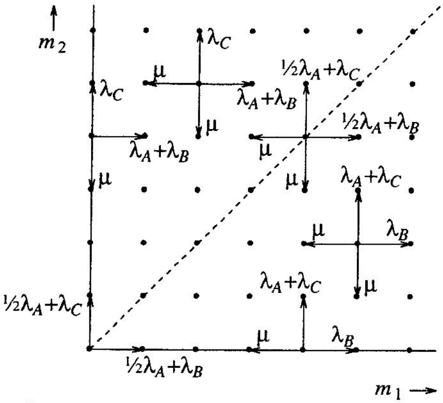

Figure 3: The transition rate diagram for the GSQS.

is seen as follows. Under a static routing, upon arrival, a job of type A joins the queue at server 1 with some given probability x, $0 \leq x \leq 1$, and it joins the queue at server 2 with probability $1 - x$. Then the two servers behave as two independent $M|M|1$ queues with workloads $x\lambda_A + \lambda_B$ and $(1 - x)\lambda_A + \lambda_C$, respectively, and the system is ergodic if $x\lambda_A + \lambda_B < 1$ and $(1 - x)\lambda_A + \lambda_C < 1$. It may be shown that this latter condition always can be satisfied for some choice of x if condition (1) is satisfied. From now on, we assume that condition (1) is satisfied.

The performance measures we are interested in are the mean waiting times $W^{(A)}$, $W^{(B)}$, $W^{(C)}$ for each of the job types A, B and C separately and the mean waiting time W for all job types together, which is equal to

$$W = \frac{\lambda^{(A)}}{\lambda} W^{(A)} \frac{\lambda^{(B)}}{\lambda} W^{(B)} \frac{\lambda^{(C)}}{\lambda} W^{(C)} .$$

It is easily seen that $W^{(B)}$ and $W^{(C)}$ are equal to the mean queue lengths L_1 and L_2 at the servers 1 and 2, respectively, and that $W^{(A)}$ is equal to the mean L_{sq} of the length of the shortest queue.

Finally, note that the GSQS is *symmetric* if $\lambda_B = \lambda_C$. For such a system, the ergodicity condition (1) reduces to $\rho < 1$ and the shortest queue routing used by the jobs of type A can be shown to minimize the total number of jobs in the system and hence also the mean waiting time W (this may be done by the technique used by Hordijk and Koole [7]).

3 Solution by flexible bound models

We now define two truncation models: one leading to lower bounds for the waiting times $W^{(A)}$, $W^{(B)}$, $W^{(C)}$ and W, and another one leading to upper bounds.

Since the shortest queue routing in general will cause a drift to the states with equal queue lengths, for both the lower and the upper bound model the truncated state space is defined by

$$M' = \{m \in M \mid m = (m_1, m_2),\ m_1 \le m_2 + T_1 \text{ and } m_2 \le m_1 + T_2\},$$

where $T_1, T_2 \in I\!N$ are so-called threshold parameters. For this choice of the truncated state space, there are four types of transitions pointing from states inside M' to states outside M':

(i) for the states $m = (m_1, m_1 + T_1) \in M'$ with $m_1 > 0$, a service completion at server 1 occurs with rate μ and leads to a transition from m to state $n = m - e_1 \notin M'$;

(ii) for the states $m = (m_2, m_2 + T_2) \in M'$ with $m_2 > 0$, a service completion at server 2 occurs with rate μ and leads to a transition from m to state $n = m - e_2 \notin M'$;

(iii) for the states $m = (m_1, m_1 + T_1) \in M'$ with $m_1 \ge 0$, an arrival of a job of type C occurs with rate λ_C and leads to a transition from m to state $n = m + e_2 \notin M'$;

(iv) for the states $m = (m_2, m_2 + T_2) \in M'$ with $m_2 \ge 0$, an arrival of a job of type B occurs with rate λ_B and leads to a transition from m to state $n = m + e_1 \notin M'$.

In the lower bound model, these transitions are redirected from the states n to states n' which correspond to situations with a smaller number of jobs at one of the two servers. With respect to waiting times and queue lengths these states are more attractive. In the upper bound model, redirections are made to less attractive states corresponding to situations with a larger number of jobs at one of the two servers.

In the lower bound model, the transitions described under (i) and (ii) are redirected to the states $n' = n - e_2 = m - e_1 - e_2 \in M'$ and $n' = n - e_1 = m - e_1 - e_2 \in M'$, respectively. The physical interpretation of these redirections is that a departure of a job at a non-empty shortest queue is accompanied by a destruction or killing of one job at the other queue. Further, the transitions described under (iii) and (iv) are redirected to the states $n' = n - e_2 = m \in M'$ and $n' = n - e_1 = m \in M'$, i.e. to the states m itself. The physical interpretation of these redirections is that a new job arriving at one of the servers is rejected. Because of the physical interpretations, the lower bound model is called the *Threshold Killing and Rejection (TKR) model.*

In the upper bound model, the transitions described under (i)-(iv) are redirected to $n' = n + e_1 = m$, $n' = n + e_2 = m$, $n' = n + e_1 = m + e_1 + e_2$ and $n' = n + e_2 = m + e_1 + e_2$, respectively. The meaning behind the first two types of redirections is that if for one queue the difference with respect to the shortest queue has already reached its maximum value, then a service completion at the other queue is not accompanied by a departure, and the job in service has to be served once more; this is equivalent to saying that then the other server is blocked. The meaning behind the latter two types of redirections is that an arrival of a new job at a queue for which the difference with respect to the shortest queues has already reached its maximum value, is accompanied by the addition of one extra job at each of the shortest queues. Hence, the upper bound model is called the *Threshold Blocking and Addition (TBA) model.*

In Figure 4, we have depicted the redirections for both the lower and upper bound model.

The TKR model leads to stochastically smaller lengths for the queue at server 1, the queue at server 2 and the shortest queue, and hence also to smaller means than obtained for the original model. Further, it may be shown that the larger the values of T_1 or T_2 the

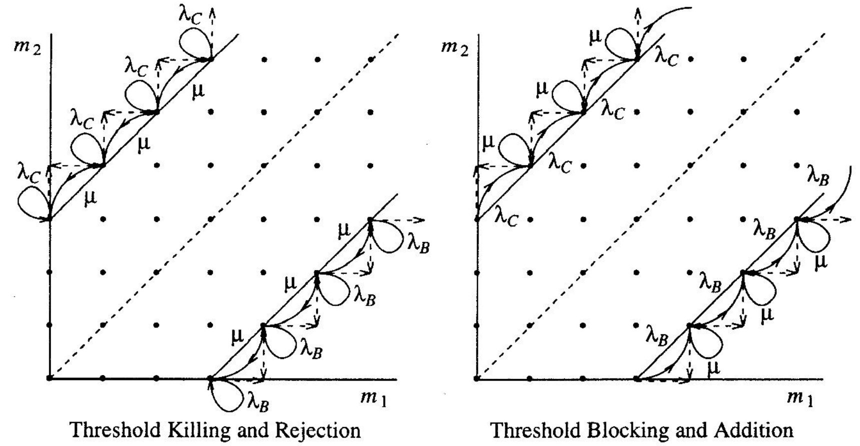

Figure 4: The redirections for the TKR and TBA model. For both models, T_1 and T_2 have been taken equal to 3.

smaller the difference between the queue lengths in the TKR model and the queue lengths in the original model. The lower bounds for the mean queue lengths immediately lead to lower bounds $W^{(A)}_{TKR}(\hat{T})$, $W^{(B)}_{TKR}(\hat{T})$, $W^{(C)}_{TKR}(\hat{T})$ and $W_{TKR}(\hat{T})$ for the mean waiting times; here $\hat{T} = (T_1, T_2)$. Similarly, the TBA model leads to larger queue lengths and waiting times. The upper bounds for the mean waiting times are denoted by $W^{(A)}_{TBA}(\hat{T})$, $W^{(B)}_{TBA}(\hat{T})$, $W^{(C)}_{TBA}(\hat{T})$ and $W_{TBA}(\hat{T})$. A formal proof of all these monotonicity results may be given by using the *precedence relation method.* This method is based on Markov reward theory and has been developed in [11].

For both the TKR and TBA model, the steady-state distribution can be determined by the *matrix-geometric approach*, as described in [8]. This enables an efficient computation of the corresponding lower and upper bounds for the waiting times; see [11] for appropriate matrix formulae that can be used for this computation.

4 Numerical results

In this final section, numerical results for two series of instances are presented in order to show how well the waiting times of the original GSQS can be determined by using the bound models. The instance with

$$\rho \;=\; 0.9, \;\; \lambda \;=\; 2\rho, \;\; \lambda_A \;=\; p\lambda \text{ with } p = \frac{1}{2}, \;\; \lambda_B \;=\; \lambda_C \;=\; \frac{1}{2}(1-p)\lambda$$

has been chosen as a basic instance. In the first series, we have varied the value of the workload ρ. In the second series, we have varied the value of the fraction p of jobs that can be handled by both servers.

Since all instances concern symmetric cases, we can take $T_1 = T_2 = T$ and the waiting times $W^{(A)}$, $W^{(B)}$, $W^{(C)}$ and W can be determined by solving the TKR and TBA model

ρ	T	$W^{(A)}$	$\Delta^{(A)}(\hat{T})$	$W^{(B)}$	$\Delta^{(B)}(\hat{T})$	W	$\Delta(\hat{T})$
0.1	2	0.0146	0.0006	0.1059	0.0007	0.0603	0.0006
0.2	3	0.0558	0.0006	0.2282	0.0008	0.1420	0.0007
0.3	3	0.1281	0.0034	0.3746	0.0043	0.2514	0.0038
0.4	4	0.2351	0.0030	0.5577	0.0038	0.3964	0.0034
0.5	5	0.3966	0.0034	0.7977	0.0042	0.5971	0.0038
0.6	7	0.6468	0.0018	1.1337	0.0021	0.8902	0.0019
0.7	8	1.0723	0.0039	1.6532	0.0044	1.3628	0.0041
0.8	11	1.9222	0.0027	2.6142	0.0029	2.2682	0.0028
0.9	15	4.4516	0.0032	5.2782	0.0033	4.8649	0.0033
0.95	18	9.4729	0.0048	10.3800	0.0048	9.9265	0.0048
0.98	23	24.4883	0.0032	25.4495	0.0032	24.9689	0.0032
0.99	26	49.4939	0.0031	50.4742	0.0031	49.9841	0.0031

Table 1: The mean waiting times $W^{(\cdot)}$ and W determined within an absolute accuracy of $\epsilon_{abs} = 0.005$ for increasing values of ρ and with $\lambda = 2\rho$, $\lambda_A = \lambda$, $\lambda_B = \lambda_C = \lambda$.

for increasing values of T. Here, for each T, the values of $(W^{(A)}_{TKR}(\hat{T}) + W^{(A)}_{TBA}(\hat{T}))/2$ and $\Delta^{(A)}(\hat{T}) = (W^{(A)}_{TBA}(\hat{T}) - W^{(A)}_{TKR}(\hat{T}))/2$, where $\hat{T} = (T, T)$, are used as an approximation for $W^{(A)}$ and an upper bound for the corresponding absolute inaccuracy; and, similarly for $W^{(B)}$, $W^{(C)}$ and W. For each instance, we have determined the smallest value of T for which each of the waiting times was determined within an absolute accuracy $\epsilon_{abs} = 0.005$.

The numerical results have been gathered in the Tables 1 and 2. The first column of Table 1 denotes the chosen values for ρ, while the second column depicts the value found for T. In the third, fifth and seventh column, we have listed the approximations which for this T have been obtained for $W^{(A)}$, $W^{(B)} = W^{(C)}$ (because of the symmetry, also the waiting times for the types B and C are equal) and W; and, in the fourth, sixth and eighth column, we have listed the upper bounds $\Delta^{(A)}(\hat{T})$, $\Delta^{(B)}(\hat{T}) = \Delta^{(C)}(\hat{T})$ and $\Delta(\hat{T})$ for the corresponding absolute inaccuracies. Table 2 consists of the same columns, except that in this table the first column denotes the chosen values of p.

The results in Table 1 show that, as expected, the threshold parameter T which is needed to approximate the mean waiting times within the desired absolute accuracy, is increasing as a function of the workload ρ. Further, the results in the Table 2 show that the required value for T strongly depends on the strength of the drift to the states with equal queue lengths, i.e. to the states on the diagonal. In this table, a smaller value for p corresponds to a weaker drift to the states on the diagonal. It follows that the weaker the drift to the diagonal, the larger the required value for T. In the extreme case with $p = 0.0$, in which the corresponding SQS-JDP consists of 2 independent $M|M|1$ queues, T has to be equal to 85 in order to reach the desired accuracy, while in the other extreme case with $p = 1.0$, in which we have a pure symmetric shortest queue system, T only has to be equal to 8.

From the values found for T, it may be concluded that the TKR and TBA model only lead to tight bounds, if the drift to the states with equal queue lengths is sufficiently strong. This will also hold for GSQSs with more than two servers. It is noted that the existence of a certain drift to the states with equal queue lengths has been a point of departure when we constructed the TKR and TBA model. So, if there is only a weak drift to the states with equal queue lengths, then the probability mass will not be concentrated around these states,

p	T	$W^{(A)}$	$\Delta^{(A)}(\hat{T})$	$W^{(B)}$	$\Delta^{(B)}(\hat{T})$	W	$\Delta(\hat{T})$
0.0	85	4.2648	0.0024	8.9976	0.0046	8.9976	0.0046
0.1	43	4.3594	0.0038	6.8002	0.0046	6.5561	0.0045
0.2	29	4.4027	0.0041	6.0435	0.0045	5.7154	0.0044
0.3	22	4.4266	0.0040	5.6619	0.0042	5.2913	0.0041
0.4	18	4.4414	0.0033	5.4320	0.0034	5.0357	0.0034
0.5	15	4.4516	0.0032	5.2782	0.0033	4.8649	0.0033
0.6	13	4.4589	0.0027	5.1682	0.0028	4.7426	0.0028
0.7	11	4.4645	0.0034	5.0856	0.0035	4.6509	0.0034
0.8	10	4.4688	0.0025	5.0212	0.0025	4.5793	0.0025
0.9	9	4.4722	0.0021	4.9697	0.0021	4.5220	0.0021
1.0	8	4.4751	0.0022	4.9275	0.0022	4.4751	0.0022

Table 2: The mean waiting times $W^{(\cdot)}$ and W determined within an absolute accuracy of $\epsilon_{abs} = 0.005$ for the GSQS with $\rho = 0.9$, $\lambda = 2\rho$, $\lambda_A = p\lambda$, $\lambda_B = \lambda_C = (1-p)\lambda$, and varying p.

and one should focus on bound models with alternative truncated state spaces.

The values presented in the Tables 1 and 2 for the mean waiting times itself, also deserve some attention. The results in Table 1 show that only a small difference between the waiting times for the types B and C and the waiting time for type A is obtained, even for high workloads. From the results in Table 2, it follows that the mean waiting time W for all job types together is more than proportionally decreasing as a function of the fraction p of jobs which can be served by both servers. In fact, a small fraction p of jobs that can be handled by both servers, already leads to a considerable reduction for W, compared to the situation with $p = 0$. From this, we can draw the following important conclusion for the production of Printed Circuit Boards by the flexible assembly system, as described in Section 1: *In order to obtain small mean waiting times for the given total workload, the assignment of the components to the insertion machines should be such that for the resulting GSQS a strong drift to the states with equal queue lengths is obtained.* Note that these assignments are precisely the ones for which our bound models work well. Hence, after having selected a small number of assignments which are expected to have the strong drift to the states with equal queue lengths, the bound models can be well used to compute the performance for each of the selected assignments, and subsequently the best assignment can be easily determined.

References

[1] ADAN, I.J.B.F., AND HOOGHIEMSTRA, G., The $M|M|c$ with critical jobs. Memorandum COSOR 96-20, Eindhoven University of Technology, Dept. of Math. and Comp. Sci., 1996.

[2] ADAN, I.J.B.F., VAN HOUTUM, G.J., AND VAN DER WAL, J. Upper and lower bounds for the waiting time in the symmetric shortest queue system. *Annals of Operations Research 48* (1994), 197–217.

[3] ADAN, I.J.B.F., WESSELS, J., AND ZIJM, W.H.M. Queueing analysis in a flexible assembly system with a job-dependent parallel structure. In *Operations Research Proceedings 1988*, Springer-Verlag, Berlin, 1989, pp. 551–558.

[4] ADAN, I.J.B.F., WESSELS, J., AND ZIJM, W.H.M. Analysis of the symmetric shortest queue problem. *Stochastic Models 6* (1990), 691–713.

[5] GREEN, L. A queueing system with general-use and limited-use servers. *Operations Research 33* (1985), 168–182.

[6] HASSIN, R., AND HAVIV, M. Equilibrium strategies and the value of information in a two line queueing system with threshold jockeying. *Stochastic Models 10* (1994), 415–435.

[7] HORDIJK, A., AND KOOLE, G. On the assignment of customers to parallel queues. *Probability in the Engineering and Informational Sciences 6* (1992), 495–511.

[8] NEUTS, M.F. *Matrix-Geometric Solutions in Stochastic Models.* Johns Hopkins University Press, Baltimore, 1981.

[9] ROQUE, D.R. A note on "Queueing models with lane selection". *Operations Research 28* (1980), 419–420.

[10] SCHWARTZ, B.L. Queueing models with lane selection: a new class of problems. *Operations Research 22* (1974), 331–339.

[11] VAN HOUTUM, G.J. *New Approaches for Multi-Dimensional Queueing Systems.* Ph.D. Thesis, Eindhoven University of Technology, Eindhoven, 1995.

[12] VAN HOUTUM, G.J., ADAN, I.J.B.F., AND VAN DER WAL, J. The symmetric longest queue system. *Stochastic Models 13* (1997), 105–120.

[13] ZIJM, W.H.M. Operational control of automated PCB assembly lines. In *Modern Production Concepts: Theory and Applications*, G. Fandel and G. Zaepfel, Eds. Springer-Verlag, Berlin, 1991, pp. 146–164.