A Note on Approximate Iterative Solution of Open Tandem Networks with Blocking

Alexandre Brandwajn and Anil K. Sahai
Department of Computer Engineering, University of California, Santa Cruz, CA, USA

Received January 1989
Revised July 1989

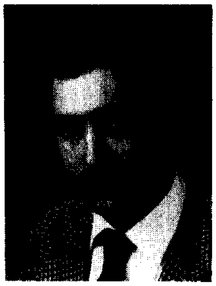
Open tandem networks of queues with finite buffers are useful as models of communication and production systems. In this note, we propose a speed up technique for the approximate iterative solution of such networks. The technique relies on a back and forth sweep of the network at each iteration so as to accelerate the return of the blocking information to the beginning of the network. Experimental evidence shows that a considerable speed up can be achieved, especially for larger networks for which the execution time can be virtually cut in half.

We also propose two variants of an iterative approach to the solution of the two-node cell used as the basic block in the tandem network solution. This approach uses directly conditional probability equations, and exploits the particular structure of a two-node tandem cell. Numerical results suggest that these two variants can be of particular value when dealing with cells with unbalanced buffer capacities.

Keywords: Tandem Queues, Finite Buffers, Blocking, Fixed Point Approximations.

1. Introduction

Open tandem networks of queues with finite buffers are useful as models of communication and production systems, and thus have received wide attention e.g. [1–4, 7, 10]. Various types of blocking resulting from the finite buffer space have been considered in the literature; see e.g., [11] for a brief description, or [12] for a more in depth discussion. In this note, our emphasis will be on the communica-



Alexandre Brandwajn received the Ingénieur Civil degree from the Ecole Nationale Supérieure des Télécommunications, Paris, France, in 1971, and the Doctor-Ingénieur and Docteur es-Sciences degrees from the University of Paris VI in 1972 and 1975 respectively.

From 1971 to 1975, he was a researcher at IRIA-Laboria, France, in the area of operating system performance evaluation and solution methods for queueing models of computer performance.

In 1975 he joined the the Ecole Nationale Supérieure de Télécommunications where he was a professor of computer science until 1979. From 1979 through 1985, he was with Amdahl Corporation, Sunnyvale, California, where he was the manager of Systems Analysis.

He is currently professor of Computer Engineering at the University of California, Santa Cruz. His research interests include the areas of analytical and numerical methods for queuing models, computer architecture, and fault tolerance techniques.

Dr. Brandwajn is the author of a number of articles and conference presentations, and a member of the Association for Computing Machinery.

Anil K. Sahai received the B.A. (Honors) degree in Mathematics from St. Stephen's College, Delhi, India, in 1980, the M.S. degree in Mathematics from Ohio University, Athens, OH in 1982, and the M.S. degree in computer science from University of Pittsburgh, Pittsburgh, PA.

From 1983 to 1986 he was an assistant professor in the department of computer science at Plymouth State College, Plymouth, NH. Presently, he is working towards a Ph.D. degree in the department of Computer and Information Science at University of California, Santa Cruz, CA. His research interests are performance modeling, communication networks, design and analysis of algorithms, and theory of computation.

tions type of blocking where start of service can only occur if and when there is space available in the downstream node buffer. In general, blocking precludes local balance properties, so that exact analysis of networks with blocking is unavailable except in a few special cases e.g. [3,9].

Several authors have proposed approximate solutions based on the idea to consider portions of the network at a time [4,7,13]. In particular, the iterative method described in [4] examines the system in pairs of neighboring nodes. At each iteration over the network, two-node cells of neighbors are considered, allowing to account precisely for potential blocking between the two neighbors. This leads to the generally high accuracy of this approximation method. Its computational complexity, although quite manageable, does deteriorate with the size of the network. Thus the goal of this note is twofold. Firstly, we propose a speed-up technique for the convergence of such an iterative scheme. Then, we discuss alternative techniques, based on the use of conditional probabilities, for the solution of the two-node cell invoked repeatedly in this iteration.

Section 2 is devoted to the convergence speed-up technique. In Section 3, we outline the alternative solutions for the basic cell of two-nodes. Section 4 summarizes the results of this note.

2. Speed-up of iteration convergence

The iterative method proposed in [4] explores the open tandem network of queures with finite buffers in pairs of neighboring nodes (see Fig. 1). At each iteration, all pairs of nodes $(i,\ i+1)$ are solved $(i=1,\ 2,\ \ldots,\ K-1$, where K denotes the number of nodes in the network) (see Fig. 2). In general, the arrival rate to the first node in such a pair is taken from the analysis of the preceding pair $(i-1,\ i)$, and the service rate of the second node is kept from the analysis of the following pair $(i,\ i+1)$ performed at the previous iteration. The number of iterations required by this method depends on network parameters. It has been our experience that, in larger systems (say, of six or more nodes), this number can grow quite substantially with the size of the network. Observation of the convergence pattern of this iterative scheme seems to indicate that blocking properties of the more remote nodes tend to take a long time to propagate to the lower numbered nodes. This suggests that improvement in convergence speed might be achieved by altering the order in which the pairs of stations are explored so as to '*force*' a faster return of the blocking information from the remote nodes.

Based on this, we propose to modify the scheme as follows. At each iteration, we '*sweep the network back and forth*', i.e., we consider the pairs of stations $(i,\ i+1)$, for $i=1,\ldots,\ K-1$ as previously, and then pairs $(j,\ j+1)$ for $j=K-2,\ K-3,\ \ldots,2$. In other words, once we have reached the last pair of nodes in our forward '*sweep*', we reconsider inner pairs of nodes in the reverse '*sweep*'. As an example, in a network with $K=5$ nodes, at each iteration we would thus solve pairs of nodes (1,2), (2,3), (3,4), (4,5), and then again (3,4), and (2,3) (see Fig. 3). By doing so, in the solution of the pair (3,4) on the reverse '*sweep*', the service rate of node 4 includes updated blocking information from the solution of (4,5). Similarly, the solution of (2,3) incorporates new information and also propagates it back in the form of the service rate that will be used for the pair (1,2) at the next iteration. As pointed out by a referee, a similar idea of '*back and forth sweep*' has been independently used by Gershwin [8].

Clearly, the network must consist of more than 3 stations for this scheme to differ from that described in [4]. Also, in comparing the two schemes, we must keep in mind that the '*back and forth sweep*' implies

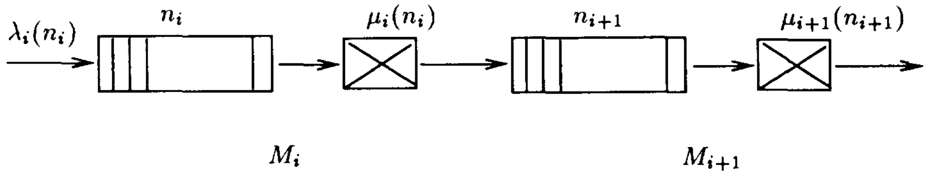


Fig. 1. A basic cell of two nodes.

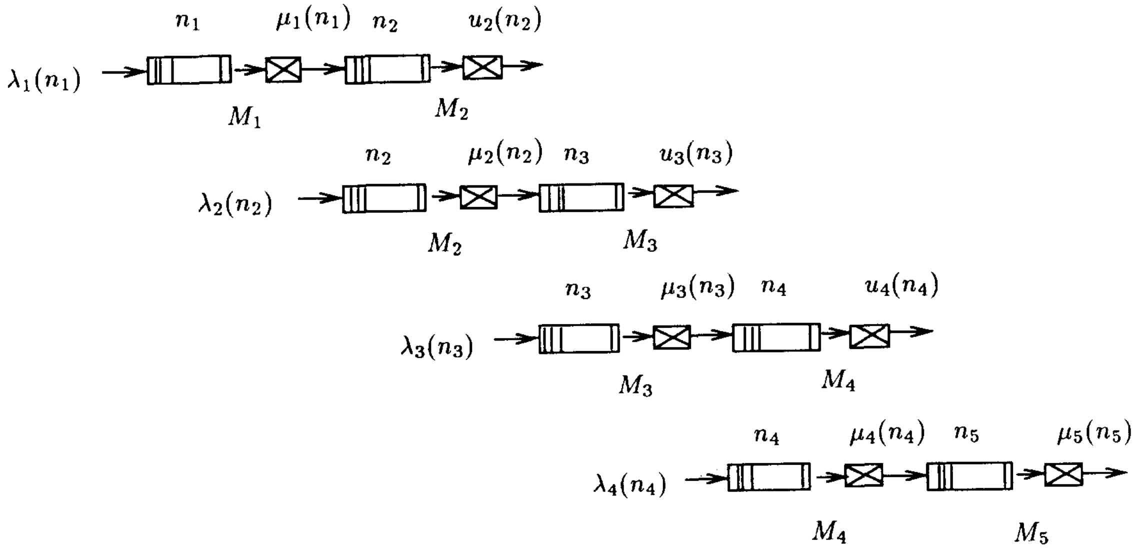


Fig. 2. Iteration of [4] in a network with 5 nodes.

that a larger number of basic two-node cells are solved at each iteration. Therefore, we use the number of basic cell solutions (and not the number of iterations) as a measure of actual convergence speed of the two schemes. We have studied a large number of examples of networks ranging from 4 to 20 nodes. The

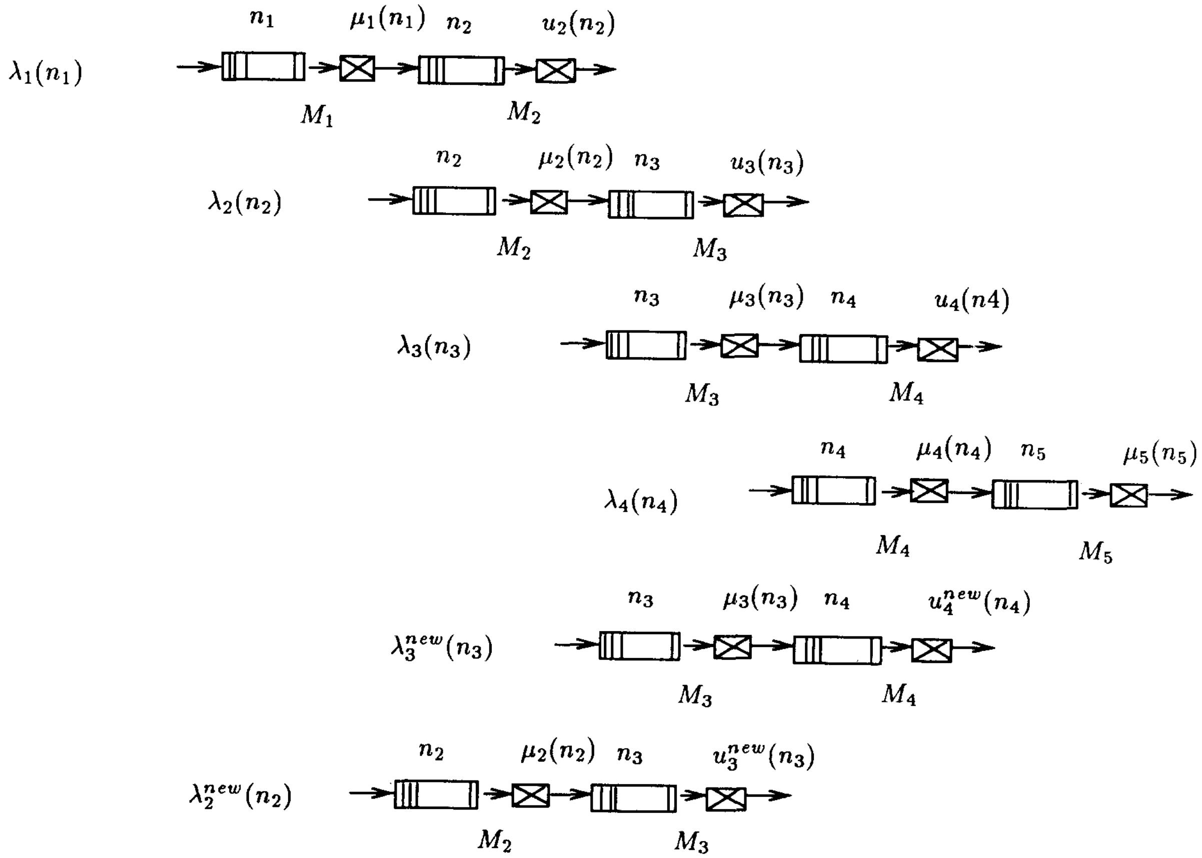


Fig. 3. Iteration with *back and forth sweep* for a network with 5 nodes.

Table 1
Selected network parameters

Blocking Level	Key	No. of Nodes	Node Capacities	Service Rates
Low	1	4	$M_1 = M_2 = \cdots = 4$	$\mu_1 = \mu_2 = \cdots = 1.1$
	2	8	$M_1 = M_2 = \cdots = 4$	$\mu_1 = \mu_2 = \cdots = 1.1$
	3	12	$M_1 = M_2 = \cdots = 4$	$\mu_1 = \mu_2 = \cdots = 1.1$
	4	16	$M_1 = M_2 = \cdots = 4$	$\mu_1 = \mu_2 = \cdots = 1.1$
	5	20	$M_1 = M_2 = \cdots = 4$	$\mu_1 = \mu_2 = \cdots = 1.1$
Medium	1	4	$M_1 = M_3 = 4,\ M_2 = M_4 = 2$	$\mu_1 = \mu_3 = 1.0,\ \mu_2 = \mu_4 = 0.8$
	2	8	$M_1 = M_3 = \cdots = 4,\ M_2 = M_4 = \cdots = 2$	$\mu_1 = \mu_3 = \cdots = 1.0,\ \mu_2 = \mu_4 = \cdots = 0.8$
	3	12	$M_1 = M_3 = \cdots = 4,\ M_2 = M_4 = \cdots = 2$	$\mu_1 = \mu_3 = \cdots = 1.0,\ \mu_2 = \mu_4 = \cdots = 0.8$
	4	12	$M_1 = M_3 = \cdots = 4,\ M_2 = M_4 = \cdots = 2$	$\mu_1 = \mu_3 = \cdots = 1.0,\ \mu_2 = \mu_4 = \cdots = 0.8$
	5	20	$M_1 = M_3 = \cdots = 4,\ M_2 = M_4 = \cdots = 2$	$\mu_1 = \mu_3 = \cdots = 1.0,\ \mu_2 = \mu_4 = \cdots = 0.8$
High	1	4	$M_1 = M_3 = 2,\ M_2 = M_4 = 1$	$\mu_1 = \mu_3 = 1.0,\ \mu_2 = \mu_4 = 0.2$
	2	8	$M_1 = M_3 = \cdots = 2,\ M_2 = M_4 = \cdots = 1$	$\mu_1 = \mu_3 = \cdots = 1.0,\ \mu_2 = \mu_4 = \cdots = 0.2$
	3	12	$M_1 = M_3 = \cdots = 2,\ M_2 = M_4 = \cdots = 1$	$\mu_1 = \mu_3 = \cdots = 1.0,\ \mu_2 = \mu_4 = \cdots = 0.2$
	4	16	$M_1 = M_3 = \cdots = 2,\ M_2 = M_4 = \cdots = 1$	$\mu_1 = \mu_3 = \cdots = 1.0,\ \mu_2 = \mu_4 = \cdots = 0.2$
	5	20	$M_1 = M_3 = \cdots = 2,\ M_2 = M_4 = \cdots = 1$	$\mu_1 = \mu_3 = \cdots = 1.0,\ \mu_2 = \mu_4 = \cdots = 0.2$

network parameters used, as well as the level of blocking in the network, are described in Table 1. The parameters include the arrival rate to the network, and the capacities and service rates for each node. Table 2 compares the corresponding number of solutions of basic two-node cell required to achieve the same accuracy by each of the two iterative schemes considered. We observe that the proposed '*back and forth sweep*' consistently outperforms the method of [4]. The performance advantage grows rather rapidly with the number of nodes in the network. For larger networks, the number of basic cell solutions can be reduced by a factor of two or three. As a final points in this section, note that the computer program implementation of the '*back and forth sweep*' represents a minimal programming change as compared to the orginal scheme of iteration.

Table 2
Comparison of two iterative methods

Blocking Level	Key	Iteration of [4] Number of Calls [a]	Iteration using 'back and forth' sweep Number of Calls [a]
Low	1	18	13
	2	105	61
	3	385	141
	4	735	253
	5	1349	397
Medium	1	15	13
	2	126	85
	3	341	201
	4	870	337
	5	1197	541
High	1	18	17
	2	91	61
	3	297	201
	4	540	337
	5	779	577

[a] Calls to solution of basic two-node cells.

3. Alternative solutions for a two-node cell

It is apparent from Table 2 that the iterative solution of a tandem network may involve a large number of solutions of a basic two-node cell. Since, the exact analytical solution of such a cell is not available in general, some type of numeric approach or approximation has to be used. It has been our observation that for larger networks the accuracy of the solution of these two node cells can be of crucial importance in the overall interative scheme. In particular, an inaccurate basic cell solution can actually slow down, or even preclude overall convergence. This puts into question the suggested use of approximations for the solution of a basic cell (cf. [4]), and emphasizes the need for a numerically fast and accurate approach.

[4] suggests the use of an iterative method developed for two-dimensional *birth and death* processes [5] to handle the solution of a basic cell. Our experience shows that this method tends to be slow when the queue capacities of a cell are strongly out of balance. It is possible to design a solution approach which works especially well in such cases.

Note that, in the iterative solution of the tandem network, conditional probabilities of the state of a node given that of its neighbor are used to obtain the equivalent arrival and service rates (cf. [4]). The solution of cell $(i,\ i+1)$ must produce the equivalent arrival rate to node $i+1$,

$$\lambda_{i+1}(n_{i+1}) = \sum_{n_i>0} \mu_i(n_i) \cdot P\{n_i \mid n_{i+1}\} \tag{1}$$

as well as the equivalent service rate for node i,

$$u_i(n_i) = \mu_i(n_i) \cdot \sum_{n_{i+1}<M_{i+1}} P\{n_{i+1} \mid n_i\}. \tag{2}$$

Hence, a solution that directly produces conditional probabilities is of interest. Note from the familiar balance equations for the basic two-node cell (cf. [4,6]), it is relatively simple to obtain equations for both conditional probability distributions. For $P\{n_{i+1} \mid n_i\}$ we get

$$\begin{aligned} P\{n_{i+1} \mid n_i\} &= D \cdot \lambda_i(n_i) \cdot P\{n_{i+1} - 1 \mid n_i + 1\} / \left[1 - P\{M_{i+1} \mid n_i + 1\}\right] \\ &\quad + E \cdot P\{n_{i+1} \mid n_i - 1\} \cdot \mu_i(n_i) \cdot \left[1 - P\{M_{i+1} \mid n_i\}\right] \\ &\quad + F \cdot P\{n_{i+1} + 1 \mid n_i\} \cdot u_{i+1}(n_{i+1} + 1) \end{aligned} \tag{3}$$

where $A = B = C = D = E = F = 1/\Psi$ with $\Psi = [A \cdot \lambda_i(n_i) + B \cdot \mu_i(n_i) + C \cdot \mu_{i+1}(n_{i+1})]$, except

$$\begin{array}{llll} A = 0 & \text{for } n_i = M_i, & D = 0 & \text{for } n_i = M_i, \text{ and } n_{i+1} = 0, \\ B = 0 & \text{for } n_i = 0, \text{ and } n_{i+1} = M_{i+1}, & E = 0 & \text{for } n_i = 0, \\ C = 0 & \text{for } n_{i+1} = 0, & F = 0 & \text{for } n_2 = M_2. \end{array}$$

Considering these equations in the order of increasing n_i for $(n_i = 0, 1, \ldots, M_i)$, we can solve them iteratively as **(Method I)**

$$\begin{aligned} P^j\{n_{i+1} \mid n_i\} &= D \cdot \lambda_i(n_i) \cdot P^{j-1}\{n_{i+1} - 1 \mid n_i + 1\} / \left[1 - P^{j-1}\{M_{i+1} \mid n_i + 1\}\right] \\ &\quad + E \cdot P^j\{n_{i+1} \mid n_1 - 1\} \cdot \mu_i(n_i) \cdot \left[1 - P^j\{M_{i+1} \mid n_i\}\right] \\ &\quad + F \cdot P^j\{n_{i+1} + 1 \mid n_i\} \cdot u_{i+1}(n_{i+1} + 1) \end{aligned} \tag{4}$$

where the superscript j denotes the iteration number. Note that, for each n_i, we view the equation as a simple one-step recurrence for $P\{n_{i+1} \mid n_i\}$. When convergence has been attained, the other conditional $P\{n_i \mid n_{i+1}\}$ can be found using

$$P\{n_i \mid n_{i+1}\} = \frac{P\{n_i\} \cdot P\{n_{i+1} \mid n_i\}}{\sum_{n_i=0}^{M_i} P\{n_i\} \cdot P\{n_{i+1} \mid n_i\}} \tag{5}$$

where $P\{n_i\}$ can be found using the standard relation implicit in the equivalence

$$\frac{P\{n_i+1\}}{P\{n_i\}} = \frac{\lambda_i(n_i)}{u_i(n_i+1)} \quad \text{with} \sum_{n_i=0}^{M_i} P\{n_i\} = 1 \tag{6}$$

where $u_i(n_i) = \mu_i(n_i) \cdot P\{n_{i+1} < M_{i+1} \mid n_i\}$.

Alternatively, for $P\{n_i \mid n_{i+1}\}$ we have the following set of equations

$$\begin{aligned} P\{n_i \mid n_{i+1}\} &= D \cdot P\{n_i+1 \mid n_{i+1}-1\} \cdot \mu_i(n_i+1) \cdot u_{i+1}(n_{i+1})/\lambda_{i+1}(n_{i+1}-1) \\ &\quad + E \cdot P\{n_i-1 \mid n_{i+1}\} + F \cdot P\{n_i \mid n_{i+1}+1\} \cdot \lambda_{i+1}(n_{i+1}) \end{aligned} \tag{7}$$

where $A = B = C = D = E = F = 1/\Psi$ with $\Psi = [A \cdot \lambda_i(n_i) + B \cdot \mu_i(n_i) + C \cdot u_{i+1}(n_{i+1})]$, except

$$\begin{array}{llll} A = 0 & \text{for } n_i = M_i & D = 0 & \text{for } n_i = M_i, \text{ and } n_{i+1} = 0, \\ B = 0 & \text{for } n_i = 0, \text{ and } n_{i+1} = M_{i+1}, & E = 0 & \text{for } n_i = 0, \\ C = 0 & \text{for } n_{i+1} = 0, & F = 0 & \text{for } n_2 = M_r. \end{array}$$

Considering this set in the order of increasing n_{i+1}, we can solve it iteratively as (**Method II**)

$$\begin{aligned} P^j\{n_i \mid n_{i+1}\} &= D \cdot P^j\{n_i+1 \mid n_{i+1}-1\} \cdot \mu_i(n_i+1) \cdot u_{i+1}(n_{i+1})/\lambda_{i+1}(n_{i+1}-1) \\ &\quad + E \cdot P^j\{n_i-1 \mid n_{i+1}\} + F \cdot P^{j-1}\{n_i \mid n_{i+1}+1\} \cdot \lambda_{i+1}(n_{i+1}) \end{aligned} \tag{8}$$

where the superscript j denotes the iteration number. Note that, here, we view the equations as a recurrence for $P\{n_i \mid n_{i+1}\}$, for each value of $n_{i+1} = 0, \ldots, \mathrm{M_i} + 1$.

Table 3
Comparsion of alternative solutions of a two-node cell

Blocking Level	λ	μ_1	M_1	μ_2	M_2	GCM [4,5]	Method I	Method II
Low	1.0	1.0	2	1.0	2	0.28	0.12	0.18
	1.0	1.0	2	1.0	5	1.48	1.18	0.34
	1.0	1.0	2	1.0	10	5.80	5.60	0.60
	1.0	1.0	2	1.0	20	18.60	18.40	1.40
	1.0	1.0	5	1.0	2	0.74	0.30	0.92
	1.0	1.0	10	1.0	2	3.60	0.60	3.80
	1.0	1.0	20	1.0	2	10.00	1.00	21.00
	1.0	1.0	5	1.0	5	2.72	2.64	2.64
	1.0	1.0	10	1.0	10	24.20	33.60	24.40
Medium	1.0	1.0	2	0.8	2	0.24	0.16	0.18
	1.0	1.0	2	0.8	5	1.54	1.24	0.34
	1.0	1.0	2	0.8	10	8.00	7.40	0.80
	1.0	1.0	2	0.8	20	38.80	37.60	1.60
	1.0	1.0	5	0.8	2	0.62	0.32	0.92
	1.0	1.0	10	0.8	2	2.60	0.60	3.40
	1.0	1.0	20	0.8	2	5.80	1.20	18.80
	1.0	1.0	5	0.8	5	2.68	2.60	2.28
	1.0	1.0	10	0.8	10	23.20	28.80	17.40
High	1.0	1.0	2	0.4	2	0.24	0.12	0.16
	1.0	1.0	2	0.4	5	1.26	1.12	0.34
	1.0	1.0	2	0.4	10	4.00	5.00	0.60
	1.0	1.0	2	0.4	20	8.60	15.60	1.20
	1.0	1.0	5	0.4	2	0.74	0.28	0.94
	1.0	1.0	10	0.4	2	1.20	0.60	2.40
	1.0	1.0	20	0.4	2	1.60	1.00	11.00
	1.0	1.0	5	0.4	5	1.80	2.12	1.64
	1.0	1.0	10	0.4	10	6.60	15.40	12.80

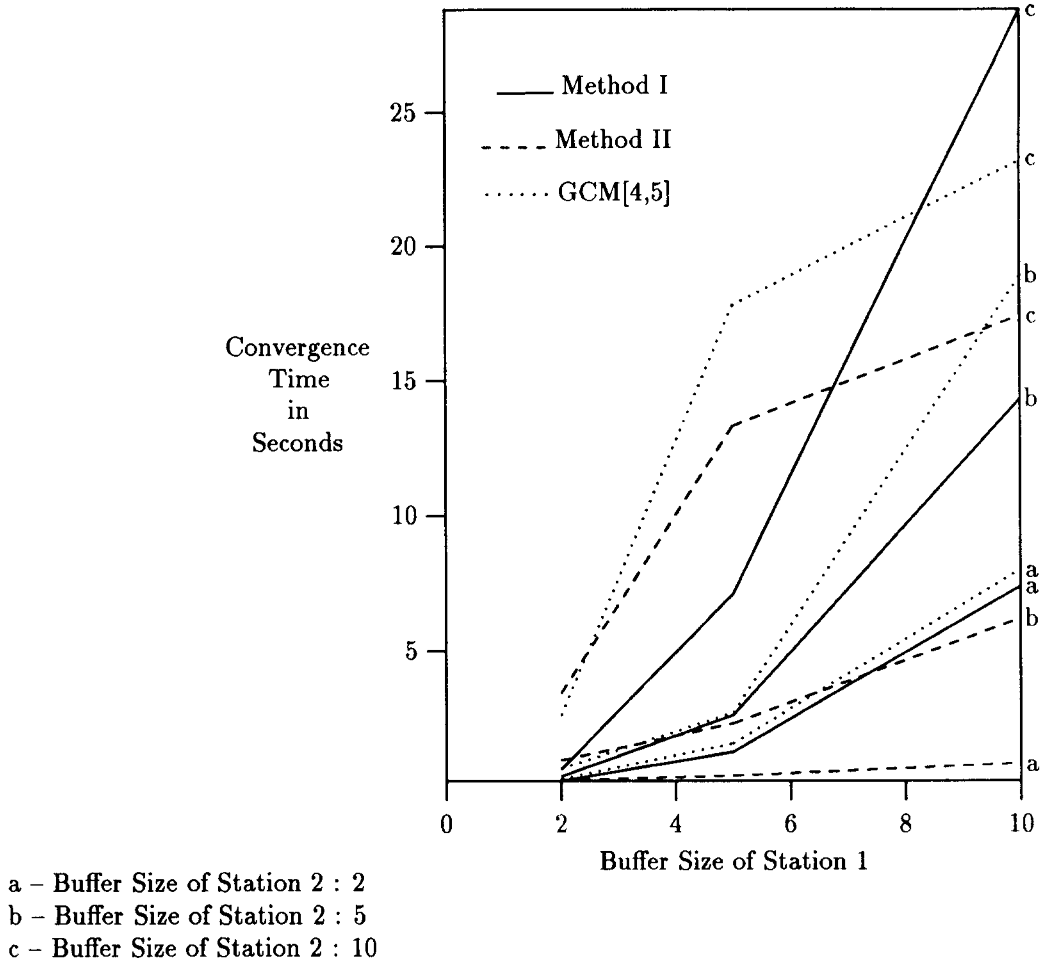


Fig. 4. Comparing convergence times of methods I and II.

We have not succeeded in establishing theoretically the convergence properties of these two iterative solutions. In the many examples considered in practice, we have found that Method I works particularly well when the buffer size at the first node is larger than the buffer at the second node of the pair: the more unbalanced the buffer sizes, the faster the convergence. Method II tends to exhibit just the reverse properties. Its convergence is fastest when the second node buffer is larger than the first node buffer.

Table 3 illustrates the convergence speed of both methods as measured by the execution time required to achieve a given equivalent level of accuracy. For comparison, we also report the time required to solve the same basic cell using the iterative method mentioned in [4] whose theoretical convergence is guaranteed (referred to as GCM). The results in Table 3 have been obtained using distributions with all states equally probable as the starting point for the iteration. Other starting points have been tried without major impact on convergence speed (cf. [6]). Fig. 4 shows graphically the convergence behavior of the methods considered as a function of the station buffer sizes for a set of parameters corresponding to medium level of blocking. We observe that, for large and balanced cells, GCM can be faster than Methods I or II. However, for unbalanced cells, Methods I or II outperform GCM. The time required to solve a cell can actually be reduced by an order of magnitude.

4. Conclusions

We have presented a speed up technique for the iterative solution of open tandem networks of queues. The technique relies on a *back and forth sweep* of the network at each iteration so as to *force* a faster return of the blocking information. Experimental evidence shows that a considerable speed up can be achieved, especially for larger networks for which the number of basic cell solutions can be virtually cut in half. Although this speed up technique is targeted here for the iterative scheme of [4], it is quite likely that it is applicable to other similar iterative solutions.

We have also proposed two variants of an iterative approach to the solution of the two-node cell used as the basic block in the iteration of [4]. This approach uses directly conditional probability equations, and results in a relatively simple computation owing to the particular structure of a two-node tandem cell. Numerical results suggest that these two variants can be of particular value when dealing with cells with unbalanced buffer capacities.

As suggested by a referee, additional improvement in the convergence speed of the iterative solution of the whole tandem network can be achieved by retaining the probability distribution computed for each two-node cell iteration, and using it as the starting point for the iterative solution of that cell during the next *sweep*. The extent of the additional improvement can be significant (e.g., a factor of two in the overall compute time to solve a network). The drawback of this approach, however, is the amount of storage required to retain the state probability distributions for all pairs of neighboring nodes.

Acknowledgment

We wish to thank the referees for their helpful remarks and suggestions.

References

[1] I.F. Akyildiz and H. Von Brand, Dual and selfdual networks of queues with rejection blocking, *Comput. J.* (Fall 1989) to appear.

[2] T. Altiok and H.G. Perros, Approximate analysis of network queues with blocking: tandem configuration, *IEEE Trans. Softw. Eng.* **SE-12** (2) (1986) 450–461.

[3] S. Balsamo and G. Iazeolla, Some equivalence properties for queueing networks with and without blocking, in: Agrawala and Tripathi:, eds., *Performance '83* (North-Holland, Amsterdam, 1983) 351–360.

[4] Alexandre Brandwajn and Yung-Li L. Jow, An approximation method for tandem queues with blocking, *Oper. Res.* **36** (1) (1988) 73–83.

[5] Alexandre Brandwajn, An iterative solution of two-dimensional bith and death processes, *Oper. Res.* **27** (1979) 595–605.

[6] Alexandre Brandwajn and Anil K. Sahai, *A Note on Iterative Solution of Open Tandem Networks with Blocking*, Technical Report, University of California, Santa Cruz, CA, March 1989.

[7] B.S. Gershwin, An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking, *Oper. Res.* **35** (2)(1987) 291–305.

[8] S.B. Gershwin, An efficient decomposition algorithm for unreliable tandem queueing systems with finite buffers, *Proc. First International Workshop on Queueing Networks with Blocking* (North-Holland, Amsterdam, 1989) 127–146.

[9] A.G. Konheim and M. Reiser, A queueing model with finite waiting room and blocking, *J. ACM* 23 (1976) 328–341.

[10] G. Latouche and M. Neuts, Efficient algorithm solutions to exponential tandem queues with blocking, *SIAM J. Algebraic Discrete Methods* **1** (1980) 93–106.

[11] Y.C. Liu, A.A. Nilsson, and H.G. Perros, Approximate analysis of product-form type queueing networks with blocking and deadlock, *Perform. Eval.* **8** (1988) 19–39.

[12] R. Onvural and H.G. Perros, On equivalencies of blocking mechanisms in queueing networks with blocking, *Oper. Res. Lett.* **5** (1986) 293–297.

[13] Y. Takahashi, A new type aggregation method for large markov chains and its applications to queuing networks, *ITC Tokyo*, 3.4A-1-1-3.4A-1-5, September 1985.