Probability in the Engineering and Informational Sciences, **34**, 2020, 522–549.
doi:10.1017/S0269964819000238

PERFORMANCE MEASURES FOR THE TWO-NODE QUEUE WITH FINITE BUFFERS

YANTING CHEN
College of Mathematics and Econometrics, Hunan University, Changsha, Hunan 410082, P. R. China
E-mail: yantingchen@hnu.edu.cn

XINWEI BAI,* RICHARD J. BOUCHERIE and JASPER GOSELING
Stochastic Operations Research, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
E-mail: xinwei.bai@ibeo-as.com; r.j.boucherie@utwente.nl; j.goseling@utwente.nl

We consider a two-node queue modeled as a two-dimensional random walk. In particular, we consider the case that one or both queues have finite buffers. We develop an approximation scheme based on the Markov reward approach to error bounds in order to bound performance measures of such random walks. The approximation scheme is developed in terms of a perturbed random walk in which the transitions along the boundaries are different from those in the original model and the invariant measure of the perturbed random walk is of product-form. We then apply this approximation scheme to a tandem queue and some variants of this model, for the case that both buffers are finite. The modified approximation scheme and the corresponding applications for a two-node queueing system in which only one of the buffers has finite capacity have also been discussed.

Keywords: error bounds, finite state space, performance measure, product-form, random walk, two-node queue

1. INTRODUCTION

Van Dijk and Lamond [41] pioneered in developing error bounds for the throughput in a tandem queue using the product-form modifications and a Markov reward approach. The method has since been further developed by van Dijk [39] and van Dijk and Puterman [43] and has been applied to, for instance, Erlang loss networks [11], to networks with breakdowns [38], to queueing networks with non-exponential service [42], and to wireless communication networks with network coding [18]. An extensive description and overview of various applications of this method can be found in van Dijk [40]. A disadvantage of the error bound method mentioned above is that the verification steps that are required to apply the method can be technically quite complicated. Goseling, Boucherie, and van Ommeren [19] developed a general verification technique for the two-node queue, i.e., for

* Current address: Ibeo Automotive Eindhoven, High Tech Campus 69, 5656 AE Eindhoven, The Netherlands

© Cambridge University Press 2019 0269-9648/19 $25.00

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

random walks in the quarter-plane. This verification technique is based on formulating the application of the error bounds method as solving a linear program. In doing so, it avoids completely the induction proof required in van Dijk and Puterman [43].

It is of interest to extend and generalize the method of Goseling et al. [19] to more general queueing networks, for instance, with more than two nodes, with finite buffers at some of the nodes, and with overflow behavior. For these networks, currently no methods exist by which we can analyze them. The current work provides the necessary intermediate step in building up our approach from the first ideas in Goseling et al. [19] toward a completely general method.

The main contribution of the current work is to provide an approximation scheme which can be readily applied to approximate performance measures for any two-node queue in which one or both queues have finite buffer capacity. The essential difference between our work and Goseling et al. [19] can be summarized as follows. In Goseling et al. [19] a linear program is constructed in which some of parameters are derived from the structure of the network. More precisely, these parameters are derived by hand in Goseling et al. [19]. In the current work, since we consider finite buffers, we have a state space with more boundaries and, therefore, a more complicated structure. The most important part of our generalization of Goseling et al. [19] consists of deriving the parameters of the linear program by means of a second linear program. In addition, we demonstrate how both linear programs can be obtained in a methodological way using a mathematical programming language. We emphasize that the methods that are developed in this paper do not rely on the fact that the underlying state-space is two-dimensional. Therefore, it seems feasible, but part of future work, that our method can be further generalized to queueing networks with more than two nodes.

The two-node queue itself, has been extensively studied. In Section 7 we provide a comparison between our work and existing methods.

The remainder of this paper is organized as follows. In Section 2, we present the model and formulate the research problem. In Section 3, we provide an approximation scheme to bound performance measures for any two-node queue with finite buffers at both queues. We bound performance measures for a tandem queue with finite buffers and some variants of this model in Section 4. In Section 5, we extend the approximation scheme to any two-node queue with finite buffers at only one queue. In Section 6, this extended approximation scheme has been applied to a coupled-queue with finite buffers at only one queue. In Section 7, we compare our method with the existing methods. Finally, we provide concluding remarks in Section 8.

2. TWO-NODE QUEUE WITH FINITE BUFFERS AT BOTH QUEUES

2.1. Two-node queue with finite buffers at both queues

The two-node queue with finite buffers at both queues is a queueing system with two servers, each of them having finite storage capacity. If a job arrives at a server which does not have any more storage capacity, then the job is lost. In general, the two queues influence each other, i.e., the service rate at one of the queues depends on the number of jobs at the other.

Such a queueing system is naturally modeled as a two-dimensional finite random walk, which we introduce next. The connection between the continuous-time queueing system and the discrete-time random walk, obtained through uniformization, is made explicit for various examples in Sections 4 and 6.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

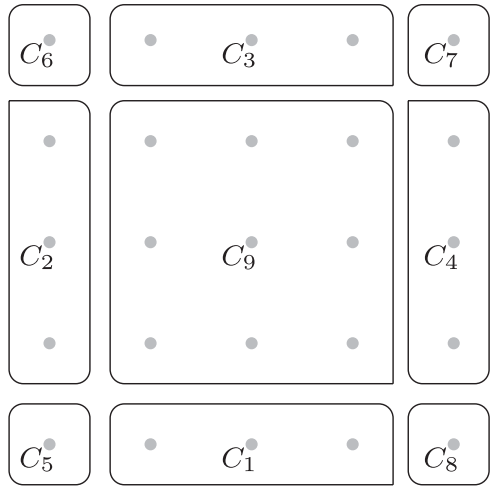


FIGURE **1.** C-partition of S with components $C_1, C_2, \ldots, C_9$.

2.2. Two-dimensional finite random walk on both axis

We consider a two-dimensional random walk R on S where

$$S = \{0, 1, 2, \ldots, L_1\} \times \{0, 1, 2, \ldots, L_2\}.$$

We use a pair of coordinates to represent a state, i.e., for $n \in S$, $n = (i, j)$. The state space is naturally partitioned in the following components (see Figure 1):

$$\begin{aligned}
&C_1 = \{1, 2, 3, \ldots, L_1 - 1\} \times \{0\}, \quad C_2 = \{0\} \times \{1, 2, 3, \ldots, L_2 - 1\},\\
&C_3 = \{1, 2, 3, \ldots, L_1 - 1\} \times \{L_2\}, \quad C_4 = \{L_1\} \times \{1, 2, 3, \ldots, L_2 - 1\},\\
&C_5 = \{(0, 0)\}, \quad C_6 = \{(0, L_2)\}, \quad C_7 = \{(L_1, L_2)\}, \quad C_8 = \{(L_1, 0)\},\\
&C_9 = \{1, 2, 3, \ldots, L_1 - 1\} \times \{1, 2, 3, \ldots, L_2 - 1\}.
\end{aligned}$$

We refer to this partition as the C-partition. The index of the component from the C-partition of state $n \in S$ is denoted by $k(n)$, i.e., $n \in C_{k(n)}$. Take for instance, $C_5 = (0, 0)$. Then the index of $(0, 0)$ is 5, hence, $k((0, 0)) = 5$, i.e., $(0, 0) \in C_5$.

Transitions for the states from S are restricted to the neighboring points (horizontally, vertically, and diagonally). The transition from a component of the C-partition to another component from the C-partition also has this restriction. For instance, let us consider C_5. The neighbors, N_5, is the product set $\{0, 1\} \times \{0, 1\}$, which denotes the coordinates of the transitions, either horizontally or vertically. For $k = 1, 2, \ldots, 9$, we denote by N_k the neighbors of a state in C_k. More precisely, $N_1 = \{-1, 0, 1\} \times \{0, 1\}$, $N_2 = \{0, 1\} \times \{-1, 0, 1\}$, $N_3 = \{-1, 0, 1\} \times \{-1, 0\}$, $N_4 = \{-1, 0\} \times \{-1, 0, 1\}$, $N_5 = \{0, 1\} \times \{0, 1\}$, $N_6 = \{0, 1\} \times \{-1, 0\}$, $N_7 = \{-1, 0\} \times \{-1, 0\}$, $N_8 = \{-1, 0\} \times \{1, 0\}$, and $N_9 = \{-1, 0, 1\} \times \{-1, 0, 1\}$. Also, let $N = N_9$.

Let $p_{k,u}$ denote the transition probability from state n in component C_k to $n + u$, where $u \in N_k$. For C_5, we now have $p_{k,u}$ from state $n = (0, 0)$ in component C_k, where $k = 5$, to $(0, 0) + u$, where $u \in N_5$. This means u could be $(0, 0), (0, 1), (1, 0)$, and $(1, 1)$. For instance, $p_{5,(1,0)}$ is the transition probability from state $(0, 0)$ in component C_5 to $(0, 0) + (1, 0)$, i.e., (1,0), transition to the right. The transition diagram of a two-dimensional finite random walk can be found in Figure 2. The transitions from a state to itself are omitted. The system

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

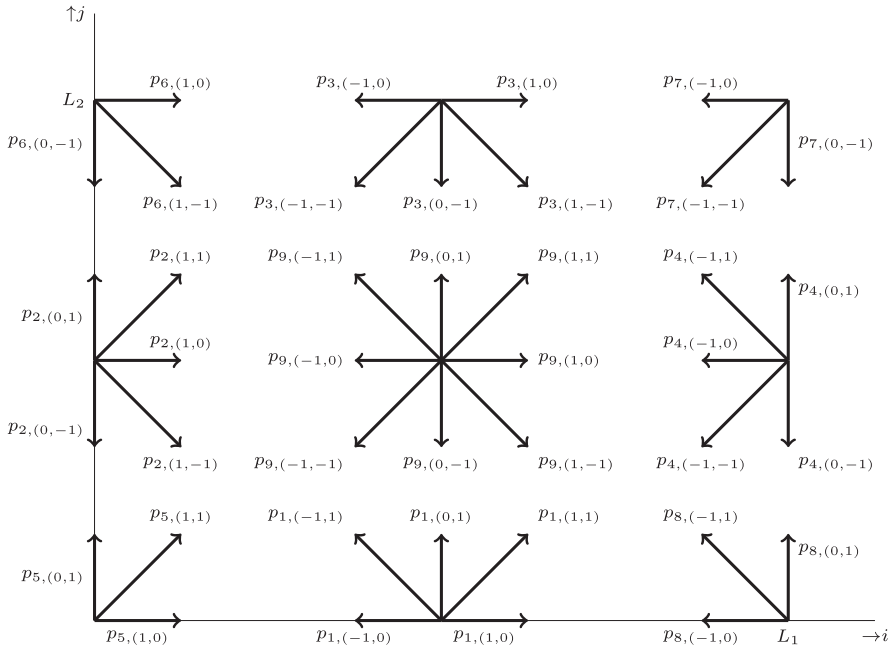


FIGURE **2.** Two-dimensional finite random walk on S. The transitions from a state to itself are omitted.

is homogeneous in the sense that the transition probabilities (incoming and outgoing) are translation invariant in each of the components, i.e.,

$$p_{k(n-u),u} = p_{k(n),u}, \quad \text{for } n-u \in S \text{ and } u \in N_{k(n)}. \tag{1}$$

Equation (1) not only implies that the transition probabilities for each part of the state space are translation invariant but also ensures that also the transition probabilities entering the same component of the state space are translation invariant.

We assume that the random walk R that we consider is aperiodic, irreducible, positive recurrent, and has invariant probability measure $m(n)$, where $m(n)$ satisfies for all $n \in S$,

$$m(n) = \sum_{u \in N_{k(n)}} p_{k(n+u),-u} m(n+u).$$

2.3. Problem formulation

Our goal is to approximate the steady-state performance of the random walk R. The performance measure of interest is

$$\mathcal{F} = \sum_{n \in S} m(n) F(n),$$

where $F(n) : S \to [0, \infty)$ is linear in each of the components from C-partition, i.e.,

$$F(n) = f_{k(n),0} + f_{k(n),1} i + f_{k(n),2} j, \quad \text{for} \quad n = (i,j) \in S. \tag{2}$$

The constants $f_{k(n),0}$, $f_{k(n),1}$, and $f_{k(n),2}$ are allowed to be different for different components from the C-partition of S.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

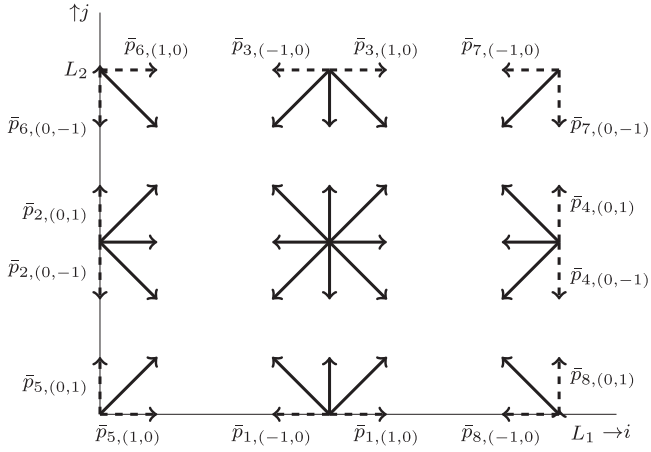


FIGURE **3.** Perturbed random walk $\bar{R}$.

When L_1 and L_2 are not too large, the invariant measure $m(n)$ could be obtained numerically by solving a system of linear equations. However, when L_1 or L_2 is relatively large, the complexity of solving a very large system of linear equations cannot be neglected anymore. Therefore, in general, we will use a perturbed random walk of which the invariant measure has a closed-form expression to approximate the performance measure $\mathcal{F}$.

We approximate the performance measure $\mathcal{F}$ in terms of the perturbed random walk $\bar{R}$. We consider the perturbed random walk $\bar{R}$ in which only the transition probabilities along the boundaries $(C_1, \ldots, C_8)$ are allowed to be different, i.e., for instance, $p_{1,(-1,0)}$, $p_{1,(1,0)}$, $p_{1,(0,0)}$ for the state from C_1 are allowed to be different in $\bar{R}$, $p_{2,(0,1)}$, $p_{2,(0,-1)}$, $p_{2,(0,0)}$ for the state from C_2 are allowed to be different in $\bar{R}$, etc. An example of a perturbed random walk $\bar{R}$ can be found in Figure 3.

We use $\bar{p}_{k,u}$ to denote the probability of $\bar{R}$ jumping from any state n in component C_k to $n + u$, where $u \in N_k$. Moreover, let $q_{k,u} = \bar{p}_{k,u} - p_{k,u}$. The probability measure $\bar{m}$ of $\bar{R}$ is of product-form,i.e.,

$$\bar{m}(n) = \alpha \rho^i \sigma^j,$$

where $n = (i, j)$ for some $(\rho, \sigma) \in (0, 1)^2$ and $\alpha \neq 0$. The measure $\bar{m}$ is the invariant measure of $\bar{R}$, i.e., it satisfies

$$\bar{m}(n) = \sum_{u \in N_{k(n)}} \bar{p}_{k(n+u),-u} \bar{m}(n + u), \tag{3}$$

for all $n \in S$.

In the following sections, we are going to find upper and lower bounds of $\mathcal{F}$ in terms of the perturbed random walk $\bar{R}$ defined above.

3. PROPOSED APPROXIMATION SCHEME

In this section, we establish an approximation scheme to find upper and lower bounds for performance measures of a two-dimensional finite random walk.

In Goseling et al. [19], an approximation scheme based on a linear programming problem is developed for a random walk in the quarter-plane. This approximation scheme has also been used in Chen, Boucherie, and Goseling [12]. We will show in this paper that the

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

technique can be extended to cover our model, i.e., a two-dimensional finite random walk. We will explain how this is achieved in the following sections.

3.1. Markov reward approach to error bounds

The fact that R and $\bar{R}$ differ only along the boundaries of S makes it possible to obtain the error bounds for the performance measures via the Markov reward approach. An introduction to this technique is provided in van Dijk [40]. We interpret F as a reward function, where $F(n)$ is the one-step reward if the random walk is in state n. We denote by $F^t(n)$ the expected cumulative reward at time t if the random walk starts from state n at time 0, i.e.,

$$F^t(n) = \begin{cases} 0, & \text{if } t = 0, \\ F(n) + \sum_{u \in N_{k(n)}} p_{k(n),u} F^{t-1}(n+u), & \text{if } t > 0, \end{cases} \tag{4}$$

For convenience, let $F^t(n+u) = 0$ where $u \in \{(s,t) | s,t \in \{-1,0,1\}\}$ if $n+u \notin S$. Terms of the form $F^t(n+u) - F^t(n)$ play a crucial role in the Markov reward approach and are denoted as *bias terms*. Let $D_u^t = F^t(n+u) - F^t(n)$. For the unit vectors $e_1 = (1,0)$, $e_2 = (0,1)$, let $D_1^t(n) = D_{e_1}^t(n)$ and $D_2^t(n) = D_{e_2}^t(n)$.

The next result in van Dijk [40] provides bounds for the approximation error for $\mathcal{F}$. We will use two non-negative functions $\bar{F}$ and G to bound the performance measure $\mathcal{F}$.

THEOREM 1 ([40]): *Let $\bar{F}$: $S \to [0,\infty)$ and G: $S \to [0,\infty)$ satisfy*

$$\left| \bar{F}(n) - F(n) + \sum_{u \in N_{k(n)}} q_{k(n),u} D_u^t(n) \right| \le G(n), \tag{5}$$

for all $n \in S$ and $t \ge 0$. Then

$$\sum_{n \in S} [\bar{F}(n) - G(n)] \bar{m}(n) \le \mathcal{F} \le \sum_{n \in S} [\bar{F}(n) + G(n)] \bar{m}(n). \tag{6}$$

3.2. A linear programming approach

In this section we present a linear programing approach to bound the errors. Due to our construction of $\bar{R}$, the random walks R and $\bar{R}$ differ only in the transitions that are along the unit directions, i.e.,

$$q_{k,u} = \bar{p}_{k,u} - p_{k,u} = 0 \quad \text{for} \quad u \neq \{e_1, e_2, -e_1, -e_2, (0,0)\}. \tag{7}$$

This restriction will significantly simplify the presentation of the result.

To start, consider the following optimization problem. We only consider how to obtain the upper bound for $\mathcal{F}$ here because the lower bound for $\mathcal{F}$ can be found similarly.

PROBLEM 1:

$$\textit{minimize} \quad \sum_{n \in S} [\bar{F}(n) + G(n)] \bar{m}(n), \tag{8}$$

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

$$\text{subject to} \quad \left|\bar{F}(n) - F(n) + \sum_{s=1,2} \left(q_{k(n),e_s} D_s^t(n) + q_{k(n),-e_s} D_s^t(n-e_s)\right)\right|$$

$$\leq G(n), \quad \text{for} \quad n \in S, t \geq 0, \tag{9}$$

$$\bar{F}(n) \geq 0, G(n) \geq 0, \quad \text{for} \quad n \in S. \tag{10}$$

The variables in Problem 1 are the functions $\bar{F}(n)$, $G(n)$ and the parameters are $F(n), \bar{m}(n), q_{k(n),e_s}$ and $D_s^t(n)$ for $n \in S$, $s = 1, 2$. Hence, Problem 1 is a linear programming problem over two non-negative variables $\bar{F}(n)$ and $G(n)$ for every $n \in S$.

This linear programming problem has infinitely many constraints because we have unbounded time horizon. We will first bound the bias term $D_s^t(n)$ uniformly over t. Then we have a linear programming problem with a finite number of variables and constraints. However, further reduction is still needed because the number of variables and constraints will increase rapidly if L_1 and L_2, which define the size of the state space, increase. Our contribution is to reduce Problem 1 to a linear programming problem where the number of variables and constraints does not depend on the size of the finite state space. By doing so, we will achieve a constant complexity in the parameters L_1 and L_2.

We now verify that the objective in Problem 1 is indeed an upper bound on the performance measure $\mathcal{F}$. Consider $D_{(0,0)}^t(n) = 0$, $D_{-e_s}^t(n) = -D_{e_s}^t(n - e_s)$ for $s = 1, 2$ and (7), it follows directly that (9) is equivalent to (5). Therefore, it follows from Theorem 1 that the objective of Problem 1 provides an upper bound on $\mathcal{F}$.

3.3. Bounding the bias terms

The main difficulty in solving Problem 1 is the unknown bias terms $D_s^t(n)$. It is in general not possible to find closed-form expressions for the bias terms. Therefore, we introduce two functions A_s: $S \to [0, \infty)$ and $B_s : S \to [0, \infty)$, $s = 1, 2$. We will formulate a finite number of constraints on functions A_s and B_s where $s = 1, 2$ such that for any t and $s = 1, 2$ we have

$$-A_s(n) \leq D_s^t(n) \leq B_s(n), \tag{11}$$

i.e., the functions A_s and B_s provide bounds on the bias terms uniformly over all $t \geq 0$. In the next section, we will find a finite number of constraints that imply (11). Our method is based on the method that was developed in Goseling et al. [19] for the case of an unbounded state space.

For notational convenience, as will become clear below, we define a finer partition of S, the Z-partition. This partition is depicted in Figure 4. For example, we have $Z_1 = \{(0, 0)\}$, $Z_2 = \{(1, 0)\}$, $Z_3 = \{2, \ldots, L_1 - 2\} \times \{0\}$, $Z_4 = \{(L_1 - 1, 0)\}$, and $Z_5 = \{(L_1, 0)\}$, the rest of the elements in the partition are determined similarly. Let $k^z(n)$ denote the label of component from Z-partition of state $n \in S$, i.e., $n \in Z_{k^z(n)}$. Similar to the definition of N_k, let N_k^z denote the neighbors of a state n in Z_k from the Z-partition of S.

The constraints which ensure (11) are obtained based on an induction in t. More precisely, we express D_s^{t+1} as a linear combination of D_1^t and D_2^t as

$$D_s^{t+1}(n) = F(n + e_s) - F(n) + \sum_{v=1,2} \sum_{u \in N_{k(n)}^z} c_{s,k^z(n),v,u} D_v^t(n + u), \tag{12}$$

where the $c_{s,k^z(n),v,u}$, $s \in \{1, 2\}, k \in \{1, 2, \ldots, 25\}, v \in \{1, 2\}, u \in N_k^z$ are constants. We now assume that such constants $c_{s,k^z(n),v,u}$ always exist. In the next section, we will explain how to obtain these constants $c_{s,k,v,u}$ based on a linear programming problem.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

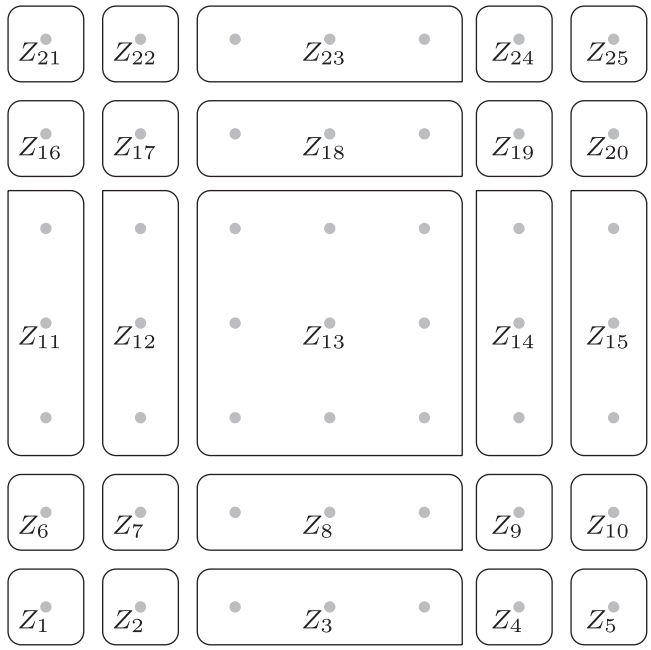


FIGURE **4.** Z-partition of S with components $Z_1, Z_2, \ldots, Z_{25}$.

We are now ready to bound the bias terms based on (12). The result, which is easy to verify, states that if $A_s\colon S \to [0,\infty)$ and $B_s\colon S \to [0,\infty)$ where $s = 1, 2$ satisfy

$$\begin{aligned} &F(n+e_s) - F(n) \\ &\quad + \sum_{v=1,2} \sum_{u \in N^z_{k(n)}} \max\{-c_{s,k^z(n),v,u} A_s(n+u), c_{s,k^z(n),v,u} B_s(n+u)\} \le B_s(n), \\ &F(n) - F(n+e_s) \\ &\quad + \sum_{v=1,2} \sum_{u \in N^z_{k(n)}} \max\{-c_{s,k^z(n),v,u} B_s(n+u), c_{s,k^z(n),v,u} A_s(n+u)\} \le A_s(n), \end{aligned}$$

for all $n \in S$, then

$$-A_s(n) \le D^t_s(n) \le B_s(n),$$

for $s = 1, 2$, $n \in S$ and $t \ge 0$.

After bounding the bias terms, we are able to rewrite the linear programing Problem 1 into Problem 2 with plugging in the upper and lower bounds for $D^t_s(n)$.

PROBLEM 2:

$$\begin{aligned} \textit{minimize} \quad & \sum_{n \in S} [\bar{F}(n) + G(n)] \bar{m}(n), \\ \textit{subject to} \quad & \bar{F}(n) - F(n) + \sum_{s=1,2} \max\{q_{k(n),e_s} B_s(n) + q_{k(n),-e_s} A_s(n-e_s), \\ & \quad - q_{k(n),e_s} A_s(n) - q_{k(n),-e_s} B_s(n-e_s)\} \le G(n), \end{aligned}$$

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

$$
\begin{aligned}
&F(n) - \bar{F}(n) + \sum_{s=1,2} \max\{q_{k(n),e_s} A_s(n) + q_{k(n),-e_s} B_s(n-e_s),\\
&- q_{k(n),e_s} B_s(n) - q_{k(n),-e_s} A_s(n-e_s)\} \le G(n)\\
&F(n+e_s) - F(n) + \sum_{v=1,2} \sum_{u \in N^z_{k(n)}} \max\{-c_{s,k^z(n),v,u} A_s(n+u),\\
&c_{s,k^z(n),v,u} B_s(n+u)\} \le B_s(n),\\
&F(n) - F(n+e_s) + \sum_{v=1,2} \sum_{u \in N^z_{k(n)}} \max\{-c_{s,k^z(n),v,u} B_s(n+u),\\
&c_{s,k^z(n),v,u} A_s(n+u)\} \le A_s(n),\\
&\bar{F}(n) \ge 0, G(n) \ge 0, A_s(n) \ge 0, B_s(n) \ge 0,\\
&\textit{for} \quad n \in S, s \in \{1,2\}.
\end{aligned}
$$

3.4. Constants $c_{s,k^z(n),v,u}$ based on a linear programming

Unlike the procedure to find the constants $c_{s,k^z(n),v,u}$ manually in Goseling et al. [19], we find the constants required to bound the bias terms automatically here based on linear programming. The reason is that due to the large number of C and Z components finding these constants manually is cumbersome and error-prone. Moreover, automating the search procedure for these constants is a necessary intermediate step in building up our approach from the first idea in Goseling et al. [19] toward a completely general method that can be used in analyzing two-dimensional random walks with more complex behavior.

Next, we formulate the sufficient conditions on $c_{s,k^z(n),v,u}$ such that (12) holds. Using (4), we have

$$
\begin{aligned}
D_s^{t+1}(n) = {} & F(n+e_s) - F(n)\\
& + \sum_{d \in k(n+e_s)} p_{k(n+e_s),d} F^t(n+e_s+d) - \sum_{u \in N_{k(n)}} p_{k(n),u} F^t(n+u). \qquad \textbf{(13)}
\end{aligned}
$$

Thus, (12) holds if and only if

$$
\begin{aligned}
&\sum_{d \in k(n+e_s)} p_{k(n+e_s),d} F^t(n+e_s+d) - \sum_{u \in N_{k(n)}} p_{k(n),u} F^t(n+u)\\
&= \sum_{v=1,2} \sum_{u \in N^z_{k(n)}} c_{s,k^z(n),v,u} D_v^t(n+u)\\
&= \sum_{v=1,2} \sum_{u \in N^z_{k(n)}} c_{s,k^z(n),v,u} \left[F^t(n+u+e_v) - F^t(n+u)\right]. \qquad \textbf{(14)}
\end{aligned}
$$

Then, the sufficient condition for (14) to hold is that for each $w \in N_{k(n)} \cup e_s + N_{k(n+e_s)}$, the coefficient of $F^t(n+w)$ at the LHS is equal to its coefficient at the RHS. We can interpret $c_{s,k^z(n),v,u}$ as a flow from $n+u$ to $n+u+e_v$. For each $w \in N_{k(n)} \cup e_s + N_{k(n+e_s)}$, $n+w$ has a demand of $\mathbf{1}(w - e_s \in N_{k(n+e_s)}) p_{k(n+e_s),w-e_s} - \mathbf{1}(w \in N_{k(n)}) p_{k(n),w}$ from the LHS of (14). Therefore, (14) holds if the demand at each node is equal to the difference between

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

the inflow and outflow of the node, i.e.,

$$\sum_{v=1,2} \mathbf{1}(w - e_v \in N_{k(n)}) c_{s,k^z(n),v,w-e_v} - \sum_{v=1,2} \mathbf{1}(w \in N_{k(n)}) c_{s,k^z(n),v,w} \tag{15}$$

$$= \mathbf{1}(w - e_s \in N_{k(n+e_s)}) p_{k(n+e_s),w-e_s} - \mathbf{1}(w \in N_{k(n)}) p_{k(n),w}, \tag{16}$$

for $k^z(n) \in \{1, 2, \ldots, 25\}$, $s \in \{1, 2\}$, and $w \in N_{k(n)} \cup e_s + N_{k(n+e_s)}$. We formulate a linear programming problem with (15) as the constraints. Then, any feasible solution of the linear programming problem guarantees that (12) holds.

3.5. Fixed number of variables and constraints

The final step is to reduce Problem 2 to a linear programming problem with fixed number of variables and constraints regardless of the size of the state space.

We first introduce the notion of a piecewise-linear function on the Z-partition. A function $F : S \to [0, \infty)$ is called Z-linear if the function is linear in each of the components from Z-partition, i.e.,

$$F(n) = f_{k^z(n),0} + f_{k^z(n),1} i + f_{k^z(n),2} j, \quad \text{for} \quad n = (i, j) \in S.$$

where $f_{k^z(n),0}$, $f_{k^z(n),1}$, and $f_{k^z(n),2}$ are the constants that define the function. In similar fashion we define C-linear functions on the C-partition of S.

Now, in Problem 2 we put the additional constraint that the variables $\bar{F}$, G, A_s, and B_s are C-linear functions. Hence, these functions are defined in terms of variables, the number of which is independent on L_1 and L_2. Hence, the number of variables in the resulting linear programming problem is independent of L_1 and L_2.

It remains to show that the number of constraints is independent of L_1 and L_2. Following the reasoning on the properties of Z-partition below (12) it is easy to see that all constraints in Problem 2 can be formulated as a non-negativity constraint on a Z-linear function. Such a constraint on a Z-linear function induces at most 4 constraints per component in the Z-partition, one constraint for each corner of the component. This indicates that the number of constraints does not depend on the size of the state space, since the number of constraints are fixed as well.

3.6. The optimal solutions

We are now able to find the upper and lower bounds of $\mathcal{F}$ based on the linear programming problem here.

Let $\mathcal{P}$ denote the set of $(\bar{F}, G)$ for which we are able to find functions A_s and B_s where $s = 1, 2$ such that all constraints in Problem 2 are satisfied. Then, we find the upper and lower bounds for $\mathcal{F}$ as follows,

$$\mathcal{F}_{up} = \min \left\{ \sum_{n \in S} [\bar{F}(n) + G(n)] \bar{m}(n) | (\bar{F}, G) \in \mathcal{P} \right\},$$

and

$$\mathcal{F}_{low} = \max \left\{ \sum_{n \in S} [\bar{F}(n) - G(n)] \bar{m}(n) | (\bar{F}, G) \in \mathcal{P} \right\}.$$

We have now presented the complete approximation scheme to obtain the upper and lower bounds for $\mathcal{F}$ using the perturbed random walk $\bar{R}$ of which the probability measure is of product-form.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

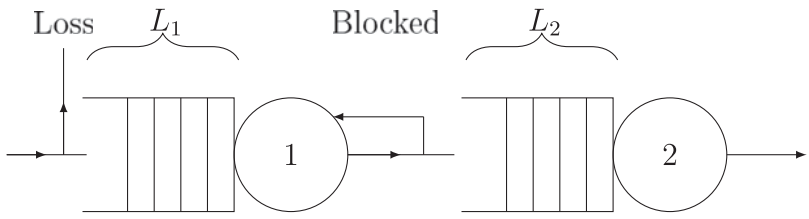


FIGURE **5.** Tandem queue with finite buffers.

3.7. The perturbed random walk with product-form invariant measure

Our construction of $\bar{R}$ is based on queueing networks with blocking. Sufficient conditions for these networks to have a product-form invariant are given in, for instance, Balsamo and De Nitto-Persone; Berezner, Krzesinski, and Taylor; Economou and Fakinos [3–5,14]. More details of the conditions from Balsamo and De Nitto-Persone; Berezner et al.; Economou and Fakinos [3–5,14], which are used to preserve product-form invariant measures for blocking systems, are given when constructing specific perturbed random walks in the applications of the approximation schemes.

In the next section, we will consider some examples: a tandem queue with finite buffers and some variants of this model (Figure 5).

4. APPLICATION TO THE TANDEM QUEUE WITH FINITE BUFFERS

In this section, we investigate the applications of the approximation scheme proposed in Section 3.

4.1. Model description

Consider a two-node tandem queue with Poisson arrivals at rate λ. Both nodes have a single server. At most a finite number of jobs, say L_1 and L_2 jobs, can be present at nodes 1 and 2. This includes the jobs in service. An arriving job is rejected if node 1 is saturated, i.e., there are L_1 jobs at node 1. The service time for the jobs at both nodes is exponentially distributed with parameters μ_1 and μ_2, respectively.

When node 2 is saturated, i.e., there are L_2 jobs at node 2, node 1 stops serving. When it is not blocked, it instantly routes to node 2. All service times are independent. We also assume that the service discipline is first-in first-out.

The tandem queue with finite buffers can be represented by a continuous-time Markov process whose state space consists of the pairs (i, j) where i and j are the number of jobs at node 1 and node 2, respectively. We now uniformize this continuous-time Markov process to obtain a discrete-time random walk. We assume without loss of generality that $\lambda + \mu_1 + \mu_2 \leq 1$ and uniformize the continuous-time Markov process with uniformization parameter 1. We denote this random walk by R_T. All transition probabilities of R_T, except those for the transitions from a state to itself, are illustrated in Figure 6.

4.2. Perturbed random walk of R_T

We now present a perturbed random walk $\bar{R}_T$. The invariant measure of the perturbed random walk $\bar{R}_T$ is of product-form and only the transitions along the boundaries in $\bar{R}_T$ are different from those in R_T.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

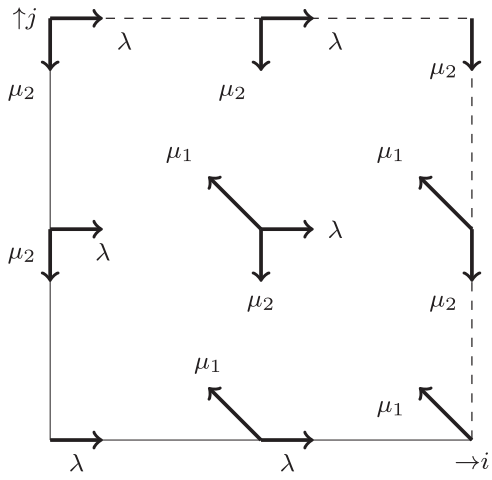


FIGURE **6.** Transition diagram of R_T.

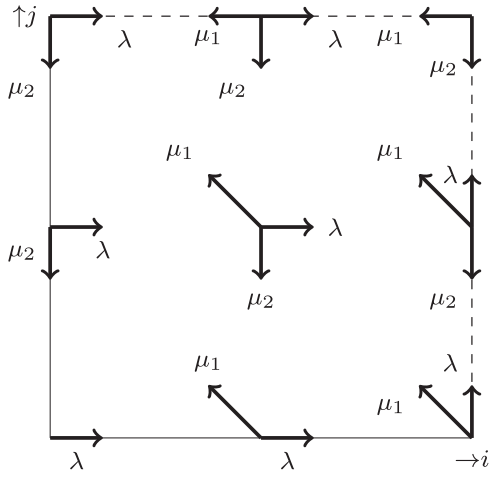


FIGURE **7.** Transition diagram of $\bar{R}_T$.

In the perturbed random walk $\bar{R}_T$, the transition probabilities in the components C_3, C_4, C_7, C_8 are different from those in R_T. More precisely, we have $\bar{p}_{3,(-1,0)} = \mu_1$, $\bar{p}_{4,(0,1)} = \lambda$, $\bar{p}_{7,(-1,0)} = \mu_1$, $\bar{p}_{8,(0,1)} = \lambda$, see Figure 7. Here we have used "the overtake full stations" protocol in Economou and Fakinos [14, Example 1] to construct the perturbed random walk $\bar{R}_T$. In particular, this protocol means that in a tandem queue, if a job overtakes a full station and moves forward with the same probabilities, then using the results from Economou and Fakinos [14] we conclude that the product-form invariant measure is retained. In our example, we see that in the perturbed random walk $\bar{R}_T$ for which the transition probabilities are demonstrated in Figure 7, when the first queue is full, i.e., $i = L_1$, the arrivals will skip the first queue and join the second queue. Moreover, when the second queue is full, i.e., $j = L_2$, the customers which finish their services in the first queue will skip the second queue and leave the two-station tandem queue system. Therefore, the invariant measure of the perturbed random walk $\bar{R}_T$ is of product-form.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

Example 1
$\lambda = 0.1$
$\mu_1 = 0.2$
$\mu_2 = 0.2$
$L_1 = L_2$

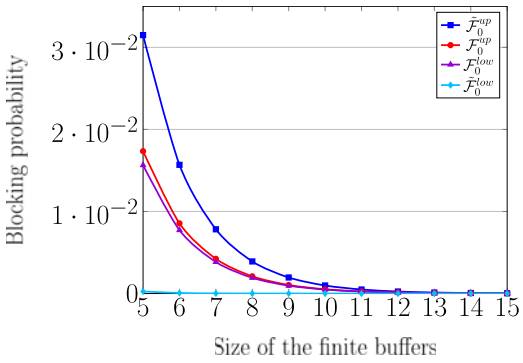


FIGURE **8.** The blocking probability $\mathcal{F}_0$.

With the normalizing constant α which depends on L_1 and L_2, we obtain

$$\bar{m}(i,j) = \alpha \left(\frac{\lambda}{\mu_1}\right)^i \left(\frac{\lambda}{\mu_2}\right)^j.$$

It can be readily verified that $\bar{m}(i,j)$ is the probability measure of the perturbed random walk $\bar{R}_T$ by substitution into the global balance equations (3) together with the normalization requirement.

4.3. Bounding the blocking probability

In this section, we provide error bounds for the blocking probability for the tandem queue with finite buffers using our approximation scheme provided in Section 3. Moreover, we show that our results are better than those obtained by van Dijk and Lamond [41].

For a given performance measure $\mathcal{F}$, we use $\mathcal{F}^{up}$, $\mathcal{F}^{low}$ to denote the upper and lower bounds for $\mathcal{F}$ obtained based on our approximation scheme and $\tilde{\mathcal{F}}^{up}$, $\tilde{\mathcal{F}}^{low}$ to denote the upper and lower bounds based on the method suggested by van Dijk and Lamond [41].

We use $\mathcal{F}_0$ to denote the blocking probability, i.e., the probability that an arriving job is rejected. We now consider an example that has also been considered in van Dijk and Lamond [41].

EXAMPLE 1: *Consider a tandem queue with finite buffers, we have* $\lambda = 0.1$, $\mu_1 = 0.2$, $\mu_2 = 0.2$.

We would like to compute the blocking probability of the queueing system. Hence, for the performance measure function $F(n)$, defined in (2), we set the coefficients $f_{k,d}$ where with $k = 1, 2, \ldots, 9$, $d = 0, 1, 2$ to be $f_{8,0} = 1$, $f_{4,0} = 1$, $f_{7,0} = 1$ and others 0. The error bounds can be found in Figure 8. Clearly, our results outperform the error bounds obtained in van Dijk and Lamond [41]. Moreover, the difference between the upper and lower bounds of $\mathcal{F}_0$ are captured in Figure 9. This indicates that our error bounds are tighter than those in van Dijk and Lamond [41].

In addition to the improved bounds, there is another advantage to our method. There is a limitation to the model modification approach that is used in van Dijk and Lamond [41]. This method requires a different model modification for each specific performance measure. For instance, the specific model modifications which are used to find error bounds for the blocking probability of a tandem queue with finite buffers in van Dijk and Lamond [41]

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

Example 1
$\lambda = 0.1$
$\mu_1 = 0.2$
$\mu_2 = 0.2$
$L_1 = L_2$

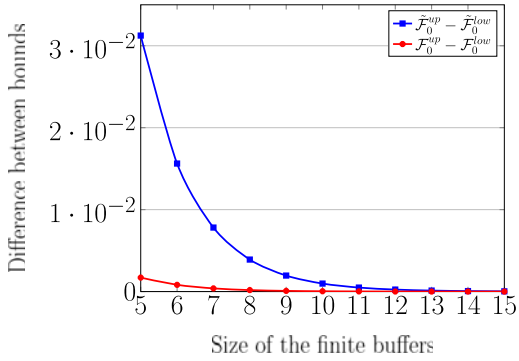


FIGURE **9.** The difference between bounds of $\mathcal{F}_0$.

cannot be used to obtain error bounds for the average number of jobs in the first node. In addition, extra effort is needed to verify that the model modifications are indeed valid for a specific performance measure. In the next section, we will show that our method can easily provide error bounds for other performance measures without extra effort.

4.4. Bounds for other performance measures

In this section, we will demonstrate the error bounds for other performance measures for Example 1, i.e., a tandem queue with finite buffers.

Let $\mathcal{F}_1$ be the average number of jobs at node 1 and $\mathcal{F}_2$ be the average number of jobs at node 2.

In general, the models (i.e., the perturbed systems), used to bound the blocking probability in van Dijk and Lamond [41] cannot be used to bound $\mathcal{F}_1$ and $\mathcal{F}_2$. The method in van Dijk and Lamond [41] requires different upper and lower bound models for different performance measures. Moreover, this method also requires an effort to verify that they are indeed the upper and lower bound models for this specific performance measure. Our approximation scheme does not have this disadvantage. For different performance measures, we only need to change the coefficients $f_{k,d}$ where $k = 1, 2, \ldots, 9$ and $d = 0, 1, 2$ in $F(n)$, which is defined in (2).

It can be readily verified that the performance measure $\mathcal{F}$ is $\mathcal{F}_1$ if and only if we assign the following values to the coefficients: $f_{1,1} = 1, f_{8,1} = 1, f_{9,1} = 1, f_{4,1} = 1, f_{3,1} = 1, f_{7,1} = 1$ and others 0. Figure 10 presents the error bounds of $\mathcal{F}_1$. Similarly, the performance measure $\mathcal{F}$ is $\mathcal{F}_2$ if and only if we assign the following values to the coefficients: $f_{2,2} = 1, f_{9,2} = 1, f_{4,2} = 1, f_{6,2} = 1, f_{3,2} = 1, f_{7,2} = 1$ and others 0. Figure 11 presents the error bounds of $\mathcal{F}_2$.

The results show that tight bounds have been achieved with our approximation scheme. Moreover, the only thing we need to change for different performance measures is the input function, which does not require further model modifications. In the next section, we will show that our approximation scheme could also give error bounds for the performance measures of the tandem queue with finite buffers which has a slower or faster server when another node is idle or saturated, respectively, without model modifications as well.

4.5. Tandem queue with finite buffers and server slow-down/speed-up

In this section, we consider two variants of the tandem queue with finite buffers. More specifically, we provide error bounds for the blocking probabilities when one server in the

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

Example 1
$\lambda = 0.1$
$\mu_1 = 0.2$
$\mu_2 = 0.2$
$L_1 = L_2$

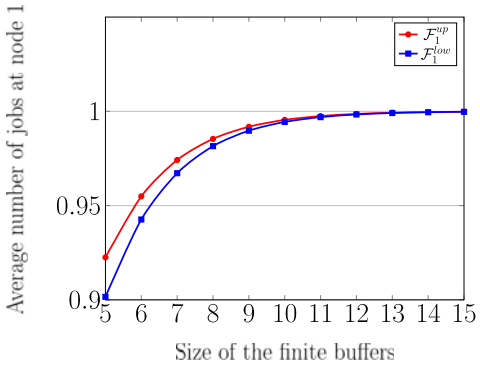


FIGURE **10.** Average number of jobs at node 1, $\mathcal{F}_1$.

Example 1
$\lambda = 0.1$
$\mu_1 = 0.2$
$\mu_2 = 0.2$
$L_1 = L_2$

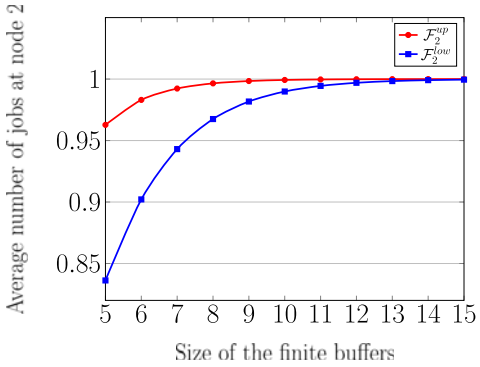


FIGURE **11.** Average number of jobs at node 2, $\mathcal{F}_2$.

tandem queue with finite buffers is slower or faster if another node is idle or saturated, respectively.

4.5.1. Tandem queue with finite buffers and server slow-down Tandem queue with server slow-down has been previously studied in, for instance, Miretskiy, Scheinhardt, and Mandjes; van Foreest et al. [31,45]. A specific type of tandem queue with finite buffers and server slow-down has been considered in Miretskiy et al.; van Foreest et al. [31,45]. More precisely, the service speed of node 1 is reduced as soon as the number of jobs in node 2 reaches some pre-specified threshold because of some sort of protection against frequent overflows.

We consider a different scenario with server slow-down. In our case, the service rate at node 2 reduces when node 1 is idle. This comes from a practical situation that when node 1 is idle, the working pressure for node 2 decreases and can shift some working capacity to other tasks. Therefore, we consider a two-node tandem queue with Poisson arrivals at rate λ. Both nodes have a single server. At most a finite number of jobs, say L_1 and L_2 jobs, can be present at nodes 1 and 2, respectively. An arriving job is rejected if node 1 is saturated. The service time for the jobs at both nodes are exponentially distributed with parameters μ_1 and μ_2, respectively. While node 2 is saturated, node 1 stops serving. When it is not blocked, it instantly routes to node 2. While node 1 is idle, the service rate of node 2 becomes $\tilde{\mu}_2$ where $\tilde{\mu}_2 < \mu_2$. All service times are independent. We also assume that the service discipline is first-in first-out.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

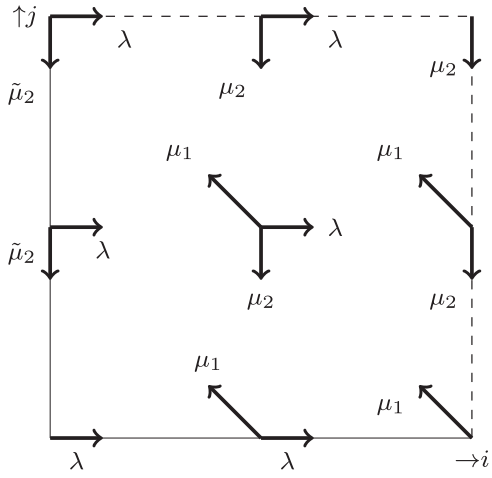


FIGURE **12.** Tandem queue with server slow-down and blocking.

Example 2
$\lambda = 0.1$
$\mu_1 = 0.2$
$\mu_2 = 0.2$
$L_1 = L_2$
$\tilde{\mu}_2 = 0.5\mu_2$

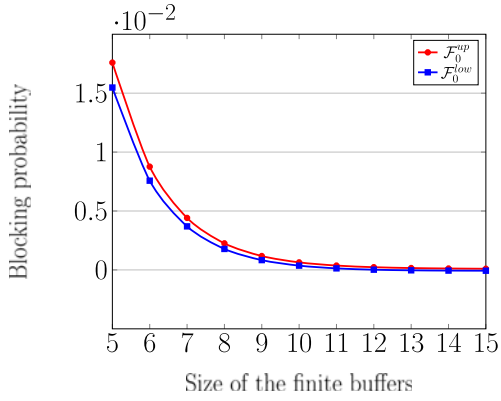


FIGURE **13.** Blocking probability with server slow-down.

The tandem queue with finite buffers and server slow-down can be represented by a continuous-time Markov process whose state space consists of the pairs (i, j) where i and j are the number of jobs at node 1 and node 2, respectively. We assume without loss of generality that $\lambda + \mu_1 + \mu_2 \leq 1$ and uniformize this continuous-time Markov process with uniformization parameter 1. Then we obtain a discrete-time random walk. We denote this random walk by R_T^{sd}, all transition probabilities of R_T^{sd}, except those for the transitions from a state to itself, are illustrated in Figure 12.

It can be readily verified that the random walk $\bar{R}_T$ as defined in Section 4.2 is a perturbed random walk of R_T^{sd} as well, i.e., the transition probabilities in $\bar{R}_T$ only differ from those in R_T^{sd} along the boundaries. We next consider a numerical example.

EXAMPLE 2 (slow-down): *Consider a tandem queue with finite buffers and server slow-down, we have* $\lambda = 0.1$, $\mu_1 = 0.2$, $\mu_2 = 0.2$, *and* $\tilde{\mu}_2 = 0.5\mu_2$.

The error bounds for the blocking probability of Example 2 are illustrated in Figure 13.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

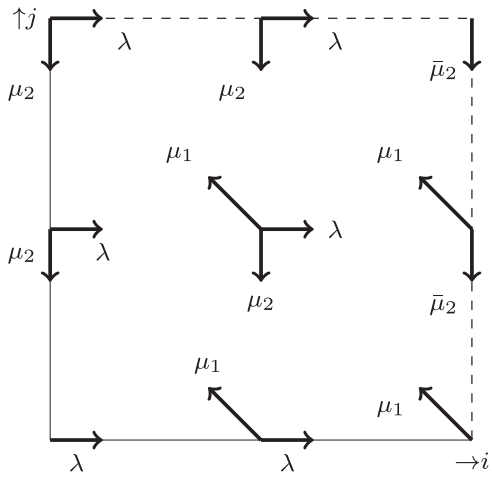


FIGURE **14.** Tandem queue with finite buffers and server speed-up.

Notice that our approximation scheme is sufficiently general in the sense that the error bounds for the performance measures of all tandem queue with server slow-down and blocking mentioned in the previous paragraphs can be obtained with our approximation scheme. There are no restrictions on the input random walk.

4.5.2. Tandem queue with finite buffers and server speed-up It is also of great interest to consider a tandem queue with finite buffers and server speed-up.

We consider the following scenario with server speed-up: The service rate at node 2 increases when node 1 is saturated. This comes from a practical situation, for instance, when node 1 is saturated, the working pressure for node 2 increases to eliminate the jobs in the queueing system. Therefore, we consider a two-node tandem queue with Poisson arrivals at rate λ. Both nodes have a single server. At most a finite number of jobs, say L_1 and L_2 jobs, can be present at nodes 1 and 2, respectively. An arriving job is rejected if node 1 is saturated. The service time for the jobs at both nodes are exponentially distributed with parameters μ_1 and μ_2, respectively. When node 2 is saturated, node 1 stops serving. When it is not blocked, it instantly routes to node 2. When node 1 is saturated, the service rate of node 2 becomes $\bar{\mu}_2$ where $\bar{\mu}_2 > \mu_2$. All service times are independent. We also assume that the service discipline is first-in first-out.

Tandem queue with finite buffers and server speed-up can be represented by a continuous-time Markov process whose state space consists of the pairs (i, j) where i and j are the number of jobs at node 1 and node 2, respectively. We assume without loss of generality that $\lambda + \mu_1 + \bar{\mu}_2 \leq 1$ and uniformize this continuous-time Markov process with uniformization parameter 1. Then we obtain a discrete-time random walk. We denote this random walk by R_T^{su}, all transition probabilities of R_T^{su}, except those for the transitions from a state to itself, are illustrated in Figure 14.

Again, it can be readily verified that the random walk $\bar{R}_T$ as defined in Section 4.2 is a perturbed random walk of R_T^{su} because only the transitions along the boundaries in $\bar{R}_T$ are different from those in R_T^{su}. We next consider the following numerical example.

EXAMPLE 3 (speed-up): *Consider a tandem queue with finite buffers and server speed-up, we have* $\lambda = 0.1$, $\mu_1 = 0.2$, $\mu_2 = 0.2$, *and* $\bar{\mu}_2 = 1.2\mu_2$.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

Example 3
$\lambda = 0.1$
$\mu_1 = 0.2$
$\mu_2 = 0.2$
$L_1 = L_2$
$\bar{\mu}_2 = 1.2\mu_2$

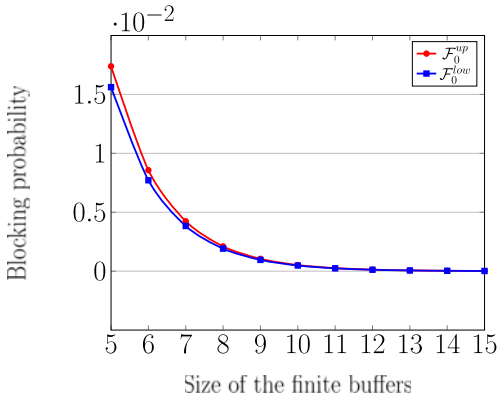


FIGURE **15.** Blocking probability with server speed-up.

The error bounds for the blocking probability of Example 3 can be found in Figure 15.

In the next section, we will extend our approximation scheme to the two-dimensional random walk in which one dimension is finite and another dimension is infinite.

5. TWO-NODE QUEUE WITH FINITE BUFFERS AT ONE QUEUE

The two-node queue with finite buffers at one queue is a queueing system with two servers, one of them having finite storage capacity. Without loss of generality, we assume node 1 has finite capacity. If a job arrives at node 1 when it does not have any more storage capacity, then the job is lost. There is no restriction to the capacity of node 2. In general, the two queues influence each other. In particular, the service rate at node 2 depends on the number of jobs at node 1. Again we model this queueing system as a two-dimensional random walk for which the state space is finite in one dimension.

5.1. Model

We consider a two-dimensional random walk $\tilde{R}$ on $\tilde{S}$ where

$$\tilde{S} = \{0, 1, 2, \ldots, L_1\} \times \{0, 1, 2, 3, \ldots\}.$$

The transition diagram of the two-dimensional random walk $\tilde{R}$ on $\tilde{S}$ can be found in Figure 17. The transitions from a state to itself are omitted. The C-partition of the state space $\tilde{S}$ can be found in Figure 16.

5.2. The modified approximation scheme

Next, we introduce the modified approximation scheme which will be used to find the upper and lower bounds.

We define the notations which would be needed in the approximation scheme later similarly as those in Section 2. For the perturbed random walk, we now consider the perturbed random walk $\tilde{\bar{R}}$ as depicted in Figure 18. Similarly to the perturbed random walk used in Section 2, only the transitions along the boundaries (the dashed transition probabilities) are allowed to be different from those in the original model.

For the approximation scheme, we again use the Markov reward approach to obtain the error bounds. Moreover, the construction of the linear programming remain the same.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

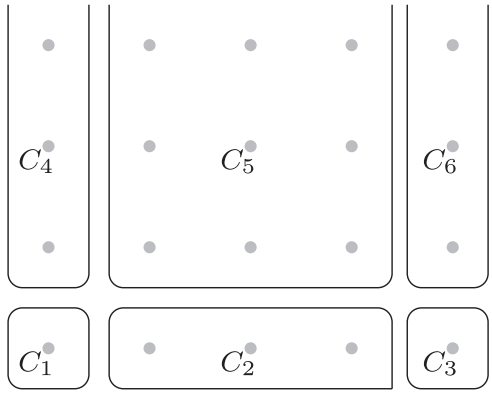


FIGURE **16.** C-partition of $\tilde{S}$ with components $C_1, C_2, \ldots, C_6$.

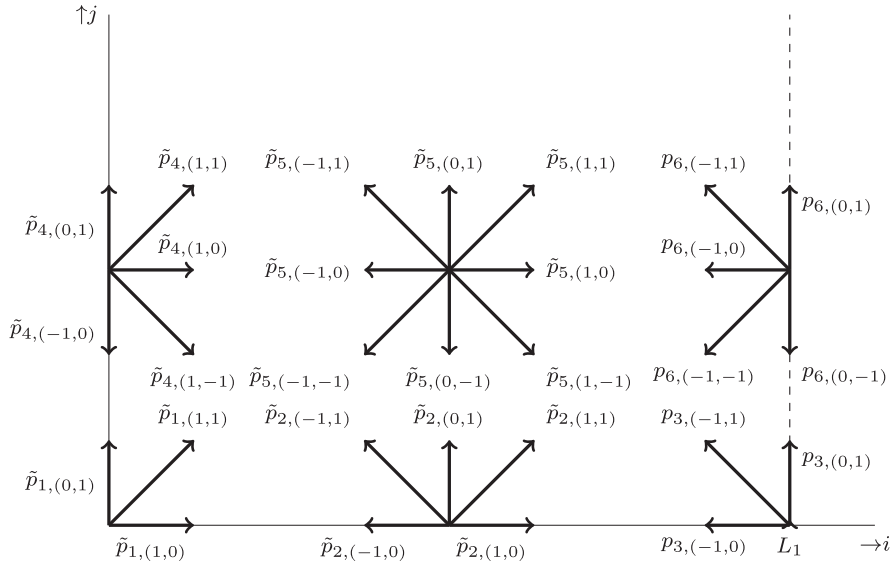


FIGURE **17.** Two-dimensional finite random walk $\tilde{R}$ on $\tilde{S}$.

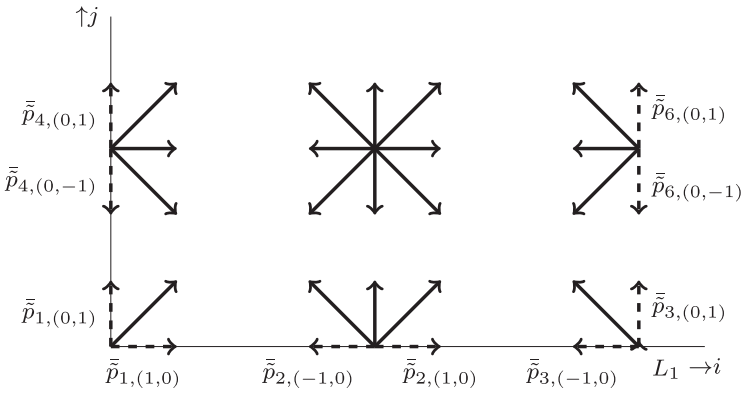


FIGURE **18.** Perturbed random walk $\bar{\tilde{R}}$ on state space $\tilde{S}$.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

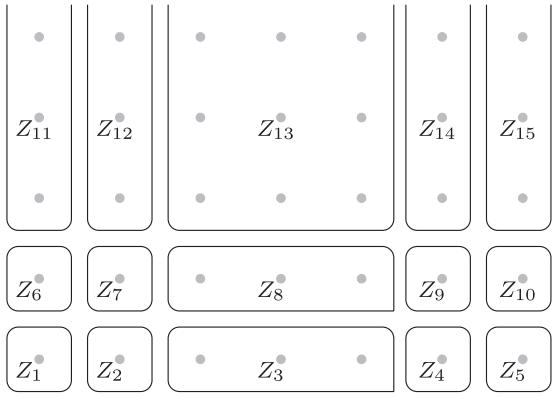


FIGURE **19.** $\tilde{Z}$-partition of $\tilde{S}$ with components $\tilde{Z}_1, \tilde{Z}_2, \ldots, \tilde{Z}_{25}$.

However, the procedure of bounding the bias terms becomes different for the model considered here due to the different state space $\tilde{S}$. More specifically, we now use the $\tilde{Z}$-partition of $\tilde{S}$, which is depicted in Figure 19, for bounding the bias terms.

The constants $c_{s,k^z(n),v,u}$ which are required to find the biased terms can be obtained from a linear programming problem automatically. This procedure is again similar to that in Section 3. More precisely, we solve the linear program with the constraints (15) to obtain $c_{s,k^z(n),v,u}$ for which (12) holds.

After bounding the biased terms, we are able to find the optimal solutions based on the linear programming problem with fixed number of variables and constraints, similarly to Section 3.

Although the number of states now becomes countably infinite, the number of constraints in the induced linear programming problem remains finite. The reason of this is the same as that in Goseling et al. [19] which deals with two-dimensional unbounded random walks. For instance, if we require a linear function $a + bi + cj$ where $(i, j) \in S$ to be nonnegative in C_6 from Figure 16, we would have finite linear constraints $a + b \times L_1 + c \times 1 \geq 0$ and $c \geq 0$. More details of building these linear constrains can be found in Gosling et al. [19, Lemma 4].

In the following section, we will consider some applications of the model discussed here.

6. APPLICATION TO THE COUPLED-QUEUE WITH FINITE BUFFERS AT ONE QUEUE

In this section, we apply the approximation scheme to a coupled-queue with finite buffers at one queue. The two coupled processors problem has been extensively studied. In particular, Fayolle and Iasnogorodski [16] reduce the problem of finding the generating function of the invariant measure to a Riemann–Hilbert problem. However, when we have finite buffers, the methods developed in Fayolle and Iasnogorodski [16] for a coupled-queue with infinite buffers are no longer valid. Knessl and Morrison considered a coupled-queue with each having a finite capacity of customers in Knessl and Morrison [25]. In particular, the capacity of the first queue is scaled to be large, while that of the second queue is held constant. Asymptotic limit of heavy traffic situation appears to be quite accurate numerically in Knessl and Morrison [25]. For the queueing system of two queues which can be modeled as quasi-birth-and-death process, the decay rates are also investigated, for instance, in Kroese, Scheinhardt, and Taylor; Latouche, Nguyen, and Taylor; Miyazawa [26,28,32].

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

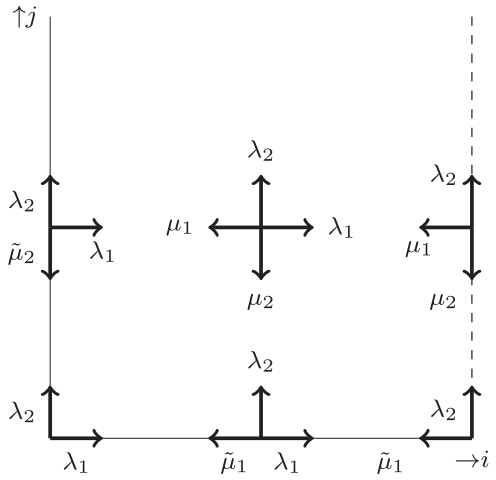


FIGURE **20.** Coupled-queue with finite buffers at one queue.

Our work aims at obtaining relatively tight numerical bounds efficiently. Also, the aim is to further develop these methods such that they can be applied more widely.

6.1. Model description

Consider a two-node queue with Poisson arrivals at rate λ_1 for node 1 and λ_2 for node 2. Both nodes have a single server and at most L_1 jobs can be present at nodes 1 and there is no restriction for the capacity of node 2. When neither of the nodes is empty they involve independently, but when one of the queues becomes empty the service rate at another queue changes. An arriving job for node 1 is rejected when node 1 is saturated. The service time at both nodes is exponentially distributed with parameters μ_1 and μ_2, respectively, when neither of the queue is empty. When node 1 is empty, the service rate at node 2 becomes $\tilde{\mu}_2$ where $\tilde{\mu}_2 > \mu_2$. When node 2 is empty, the service rate at node 1 becomes $\tilde{\mu}_1$ where $\tilde{\mu}_1 > \mu_1$. All service requirements are independent. We also assume that the service discipline is first-in first-out.

This coupled-queue with finite buffers at one queue can be represented by a continuous-time Markov process whose state space consists of the pairs (i, j) where i and j are the number of jobs at node 1 and node 2, respectively. We assume without loss of generality that $\lambda_1 + \lambda_2 + \tilde{\mu}_1 + \tilde{\mu}_2 \leq 1$ and uniformize this continuous-time Markov process with uniformization parameter 1. Then we obtain a discrete-time random walk. We denote this random walk by R_C. All transition probabilities of R_C, except those for the transitions from a state to itself, are illustrated in Figure 20.

6.2. Perturbed random walk $\bar{R}_C$

We now display a perturbed random walk $\bar{R}_C$ of R_C such that the probability measure of $\bar{R}_C$ is of product-form and only the transitions along the boundaries in $\bar{R}_C$ are different from those in R_C.

For the coupled-queue, the requirement of independence would be enough to guarantee a product-form invariant measure. Therefore, if we force both queues involve independently, then the invariant measure of the perturbed random walk $\bar{R}_C$ in Figure 21 is of product-form.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

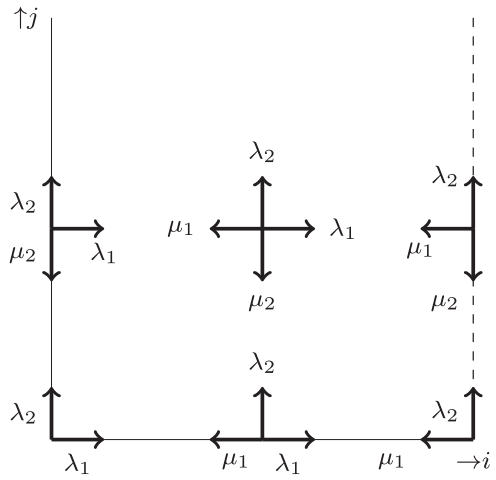


FIGURE **21.** Transition diagram of the perturbed random walk $\bar{R}_C$.

With the normalizing constant α which depends on L_1, we obtain

$$\bar{m}(n) = \alpha \left(\frac{\lambda_1}{\mu_1}\right)^i \left(\frac{\lambda_2}{\mu_2}\right)^j \quad \text{where} \quad n = (i, j). \tag{17}$$

It can be readily verified that $\bar{m}(n)$ is the probability measure of the perturbed random walk $\bar{R}_C$ by substituting it into the global balance equations (3) together with the normalization requirement.

We next illustrate a numerical example of a coupled-queue with finite buffers at one queue.

6.3. Numerical results

EXAMPLE 4: *Consider a coupled-queue with finite buffers at one queue, we have* $\lambda_1 = \lambda_2 = 0.15, \mu_1 = \mu_2 = 0.2$, $\tilde{\mu}_1 = \tilde{\mu}_2 = 0.25$.

We approximate the average number of jobs in node 1. We use F_1 to denote the average number of jobs in node 1. The upper and lower bounds of F_1, which are denoted by F_1^{up} and F_1^{low}, can be found in Figure 22.

We see from the results in Figure 22 that our approximation scheme can also be extended to finite random walks at one axis. Moreover, note that when L_1, i.e., the size of the first dimension, is increasing, the values of the upper and lower bounds reach a limit.

In the next numerical example, we will fix the service rate. We present the error bounds for the corresponding performance measure when the occupation rate, i.e., $\rho = \frac{\lambda}{\mu}$ increases, even close to 1.

EXAMPLE 5: *Consider a coupled-queue with finite buffers at one queue, we have* $\mu_1 = \mu_2 = 0.2$, $\tilde{\mu}_1 = \tilde{\mu}_2 = 0.25$, $L_1 = 20$. *Let* ρ *changes from* 0.5 *to* 0.95.

We see from Figure 23 that the error bounds are quite tight as well.

Next, we present several examples for blocking probability, which is again denoted by F_0, based on Example 5 in which the size of the buffers in the first dimension increases from 20 to 10000.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

Example 4
$\lambda_1 = 0.15$
$\lambda_2 = 0.15$
$\mu_1 = 0.2$
$\mu_2 = 0.2$
$\tilde{\mu}_1 = 1.25\mu_1$
$\tilde{\mu}_2 = 1.25\mu_2$

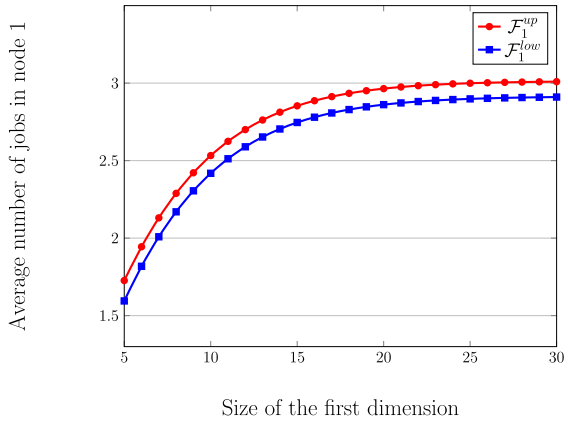


FIGURE **22.** Average number of jobs in node 1.

Example 5
$\rho = 0.5, \ldots, 0.95$
$\mu_1 = 0.2$
$\mu_2 = 0.2$
$\tilde{\mu}_1 = 1.25\mu_1$
$\tilde{\mu}_2 = 1.25\mu_2$
$L_1 = 20$

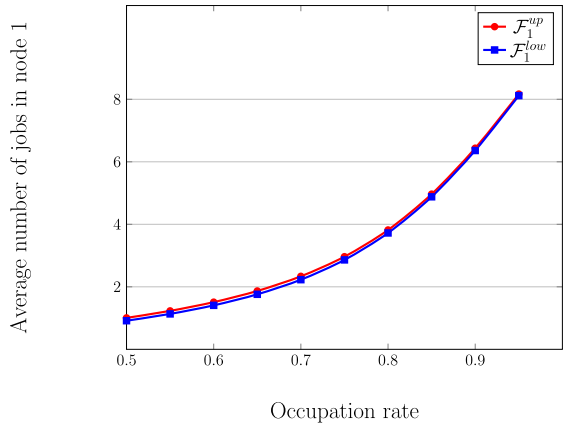


FIGURE **23.** Average number of jobs in node 1 when ρ increases.

EXAMPLE 6: *Consider a coupled-queue with finite buffers at one queue, we have* $\mu_1 = \mu_2 = 0.2$, $\tilde{\mu}_1 = \tilde{\mu}_2 = 0.25$, $L_1 = 20$ *and the occupation rate increases from* 0.5 *to* 0.95.

The bounds for blocking probabilities are very close in this case, hence, we convert these probabilities by applying logarithm to the y axis in Figure 24 and also in following examples (Figures 25 and 26).

EXAMPLE 7: *Consider a coupled-queue with finite buffers at one queue, we have* $\mu_1 = \mu_2 = 0.2$, $\tilde{\mu}_1 = \tilde{\mu}_2 = 0.25$, $L_1 = 500$ *and the occupation rate increases from* 0.98 *to* 0.99.

Next, we also extend these numerical results to the case when $L_1 = 10000$.

EXAMPLE 8: *Consider a coupled-queue with finite buffers at one queue, we have* $\mu_1 = \mu_2 = 0.2$, $\tilde{\mu}_1 = \tilde{\mu}_2 = 0.25$, $L_1 = 10000$ *and the occupation rate increases from* 0.98 *to* 0.99.

We see from the above examples that relatively tight bounds are obtained efficiently based on our approach.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

Example 6
$\rho = 0.5, \ldots, 0.95$ $\mu_1 = 0.2$ $\mu_2 = 0.2$
$\tilde{\mu}_1 = 1.25\mu_1$ $\tilde{\mu}_2 = 1.25\mu_2$
$L_1 = 20$

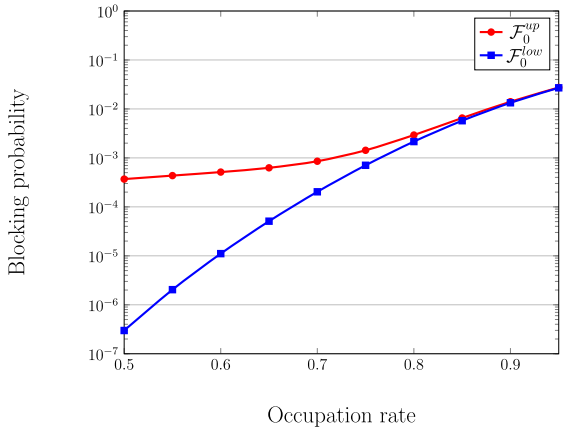


FIGURE **24.** The converted blocking probability ($y = \log Y$), $L_1 = 20$.

Example 7
$\rho = 0.980, \ldots, 0.995$ $\mu_1 = 0.2$ $\mu_2 = 0.2$
$\tilde{\mu}_1 = 1.25\mu_1$ $\tilde{\mu}_2 = 1.25\mu_2$
$L_1 = 500$

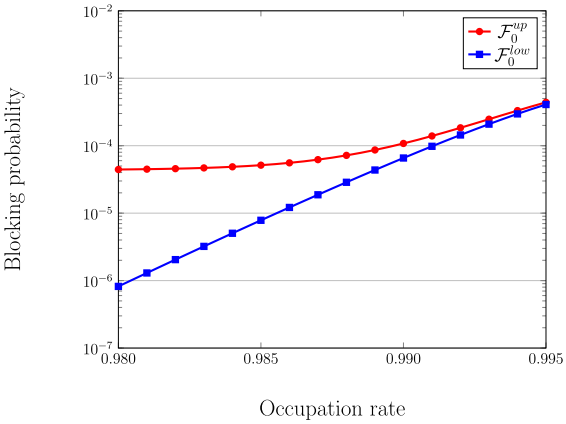


FIGURE **25.** The converted blocking probability ($y = \log Y$), $L_1 = 500$.

Example 8
$\rho = 0.9990, \ldots, 0.9998$ $\mu_1 = 0.2$ $\mu_2 = 0.2$
$\tilde{\mu}_1 = 1.25\mu_1$ $\tilde{\mu}_2 = 1.25\mu_2$
$L_1 = 10000$

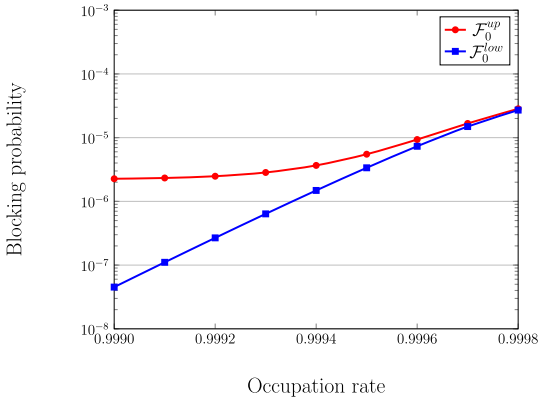


FIGURE **26.** The converted blocking probability ($y = \log Y$), $L_1 = 10000$.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

7. RELATED LITERATURE

For the two-node queue with finite buffers at both queues or at one queue, there exist other methods to obtain the equilibrium distribution. The most prominent methods are direct solution of the global balance equations and the matrix analytical method.

7.1. Two-node queue with finite buffers at both queues

When both buffers are finite, we may directly solve the global balance equations or use modified matrix analytical methods.

When the buffer size is relatively small, we may directly solve the global balance equations to obtain the equilibrium distribution and corresponding performance measures. The complexity of this approach is at least $O(L_1^2)$, where L_1 is the size of the smaller buffer. Our approach has complexity that is $O(1)$, i.e., is constant in the buffer size.

For the two-node queue with finite buffers at both queues, the modified matrix analytical method may be used to obtain the equilibrium distribution of an irreducible finite quasi-birth-and-death process (QBD), see de Nitto Persone and Grassi; Elhafsi and Molle; Grassmann and Tavakoli; Gun and Makowski; Hajek; Le Boudec; Li [13,15,20–22,29,30] and the references therein. A comprehensive comparison for variations of the matrix analytical method for solving finite QBDs can be found in Elhafsi and Molle [15, Section 4]. The common element of these methods is reduction of the global balance equations to a smaller (finite) system of equations and to express the equilibrium distribution as a function of the solution of this reduced system. The methods differ in both the way the reduced system is obtained and the way the equilibrium distribution is computed from this solution. Strong assumptions are required for the approach in de Nitto Persone and Grassi; Elhafsi and Molle; Grassmann and Tavakoli; Gun and Makowski; Hajek; Le Boudec; Li [13,15,20–22,29,30]. The common element is the assumption that some intermediate matrices must be non-singular. Before each application of the matrix analytical method we must verify non-singularity of these matrices and the approach fails when the assumptions are violated. Our approach serves as an efficient alternative to obtain performance bounds. The complexity of the methods in de Nitto Persone and Grassi; Elhafsi and Molle; Grassmann and Tavakoli; Gun and Makowski; Hajek; Le Boudec; Li [13,15,20–22,29,30] is cubic and in some special cases quadratic, see Elhafsi and Molle [15, Section 4]. The complexity of our approach is $O(1)$.

7.2. Two-node queue with finite buffers at one queue

For the standard QBD, the matrix analytic method is a mature method to obtain the equilibrium probabilities. Implementations of the matrix analytical method are available in Bini and coworkers [8–10]. The computational complexity for QBDs is in general $O(N^3)$, where N is the number of phases in the QBD, see Latouche and Ramaswami [27]. There are methods to reduce the complexity of the matrix manipulations required in using matrix analytical methods to analyze QBDs, see Bini et al.; He et al.; Perez and Van Houdt; Poloni [6,7,23,33,35]. For our approach, the complexity is a $O(1)$. The matrix analytical method achieves high accuracy. For engineering purpose, our approach can be used to obtain upper and lower bounds for performance measures within a few seconds regardless the size of the system.

7.3. Tandem queue

A special case of the two-node queue with finite buffers at both queues which has been extensively studied so far, is the tandem queue with finite buffers. An extensive survey of

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

results on this topic is provided in Balsamo; Perros [2,34]. Most of these papers focus on the development of approximations or algorithmic procedures to find steady-state system performance such as throughput and the average number of customers in the system. A popular approach used in such approximations is decomposition, see Asadathorn and Chao; Gershwin [1,17]. The main variations of a two-node queue with finite buffers at both queues are: three or more stations in the tandem queue [36], multiple servers at each station [44,46], optimal design for allocating finite buffers to the stations [24], general service times [37,42], etc. Numerical results of such approximations often suggest that the proposed approximations are indeed bounds on the specific performance measure, however rigorous proofs are not always available. Moreover, these approximation methods cannot be easily extended to a general method, which determines the steady-state performance measure of any two-node queue with finite buffers at both queues.

We have applied our approximation scheme to a tandem queue with finite buffers at both queues. We have shown that the error bounds for the blocking probability are improved compared to the error bounds for the blocking probability provided in van Dijk and Lamond [41]. The method in van Dijk and Lamond [41] is based on specific model modifications. Apart from this, our approximation scheme is more general in the sense that other interesting performance measures could also be obtained easily. This is an advantage over the methods used in van Dijk; van Dijk and Lamond; van Dijk and Puterman [39,41,43] where different model modifications are necessary for different performance measures. Moreover, we have shown that the error bounds can also be obtained for variations of the tandem queue with finite buffers. In particular, we considered the case that one server slows-down or speeds-up when another server is idle or saturated.

8. CONCLUSION AND OUTLOOK

In this paper, we presented a general approximation scheme for a two-node queue with finite buffers at either one or both queues, which establishes error bounds for a large class of performance measures. Our work is an extension of the linear programming approach developed in Goseling et al. [19] to approximate performance measures of random walks in the quarter-plane. However, we emphasize once again, that the main goal of our work is not aiming at providing a superior method for the two-node queue. Instead, we aim to develop a method for the two-dimensional model which is extendible to more general models, for instance, higher dimensional models.

We first developed an approximation scheme for a two-node queue with finite buffers at both queues. We then apply this approximation scheme to obtain bounds for the performance measures of a tandem queue in which both buffers are finite and some variants of this model. We also extend the approximation scheme to deal with a two-node queue with finite buffers at only one queue. We applied our approximation scheme to a coupled-queue with finite buffers at one queue. The approximation scheme gives tight bounds for various performance measures, like the blocking probability and the average number of jobs at node 1. We also obtain error bounds for the blocking probabilities when the size of the buffers in one dimension is really large.

The numerical results we have obtained indicate that the performance bounds for our examples are relatively tight. This matches our expectation because a linear programming problem is deployed to find the best performance bound, which would be tighter than van Dijk's performance bound which provides an arbitrary feasible solution from our linear programming problem.

A limitation of our approach is that there is no guarantee for the existence of a feasible solution for the linear programming problems. Indeed, in some cases no bounds can be

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

established. As part of future work we will establish sufficient conditions under which bounds are guaranteed to exist. Also, we will provide insight into the quality of these bounds.

Generalization of the approximation scheme used in this paper is possible. For instance, it seems feasible to extend our approximation scheme to the random walks with non-nearest neighbours or in higher dimensions. However, the construction of the perturbed random walks with product-form invariant measures of these cases would require theoretical investigations. An extension to random walks with state-dependent transitions also seems feasible. In particular, for a specific two-dimensional model with state-dependent transitions, the performance bounds obtained via an approximation scheme based on the Markov reward approach are given in van Dijk and van der Wal [44].

Acknowledgement

The authors are grateful to the reviewers as well as the editor for providing valuable suggestions that helped to improve this paper. Yanting Chen acknowledges support through the NSFC grant 71701066, the Fundamental Research Funds for the Central Universities and a CSC scholarship [No. 2008613008]. Xinwei Bai acknowledges support through a CSC scholarship [No. 201407720012]. This work is partly supported by the Netherlands Organization for Scientific Research (NWO) grant 612.001.107.

References

1. Asadathorn, N. & Chao., X. (1999). A decomposition approximation for assembly-disassembly queueing networks with finite buffer and blocking. *Annals of Operations Research* 87: 247–261.
2. Balsamo, S. (2011) Queueing networks with blocking: Analysis, solution algorithms and properties. In *Network Performance Engineering*. Kouvatsos D.D (ed). Berlin, Germany: Springer, pp. 233–257.
3. Balsamo, S. & De Nitto-Personé., V. (1991). Closed queueing networks with finite capacities: blocking types, product-form solution and performance indices. *Performance Evaluation* 12(2): 85–102.
4. Balsamo, S. & De Nitto-Personé., V. (1994). A survey of product form queueing networks with blocking and their equivalences. *Annals of Operations Research* 48(1-4): 31–61.
5. Berezner, S.A., Krzesinski, A.E., & Taylor., P.G. (1997). A product-form "loss network" with a form of queueing. *Journal of Applied Probability* 34(4): 1075–1078.
6. Bini, D.A., Latouche, G., & Meini., B. (2002). Solving matrix polynomial equations arising in queueing problems. *Linear Algebra and its Applications* 340: 225–244.
7. Bini, D.A., Latouche, G., & Meini., B. (2005). *Numerical methods for structured Markov chains*. Oxford, England: Oxford University Press.
8. Bini, D.A., Meini, B., Steffe, S., & Van Houdt., B. (2006) Structured Markov chains solver: software tools. *Proceedings from the 2006 Workshop on Tools for Solving Structured Markov Chains*.
9. Bini, D.A., Meini, B., Steffe, S., & Van Houdt., B. (2009) Structured Markov chains solver: tool extension. *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*.
10. Bini, D.A., Meini, B., Steffe, S., Perez, J.F., & Van Houdt., B. (2012). SMCsolver and Q-MAM: tools for matrix-analytic methods. *ACM SIGMETRICS Performance Evaluation Review* 39(4): 46–46.
11. Boucherie, R.J. & van Dijk., N.M. (2009). Monotonicity and error bounds for networks of Erlang loss queues. *Queueing Systems* 62(1–2): 159–193.
12. Chen, Y., Boucherie, R.J., & Goseling., J. (2016). Invariant measures and error bounds for random walks in the quarter-plane based on sums of geometric terms. *Queueing Systems* 84(1–2): 21–48.
13. de Nitto Persone, V. & Grassi., V. (1996). Solution of finite QBD processes. *Journal of Applied Probability* 33(4): 1003–1010.
14. Economou, A. & Fakinos., D. (1998). Product form stationary distributions for queueing networks with blocking the rerouting. *Queueing Systems* 30: 251–260.
15. Elhafsi, E.H. & Molle, M. (2007). On the solution to QBD processes with finite state space. *Stochastic Analysis and Applications* 25: 763–779.
16. Fayolle, G. & Iasnogorodski., R. (1979). Two coupled processors: the reduction to a riemann-hilbert problem. *Probability Theory and Related Fields* 47(3): 325–351.
17. Gershwin., S.B. (1987). An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. *Operations Research* 35(2): 291–305.
18. Goseling, J., Boucherie, R.J., & van Ommeren., J.C.W. (2013). Energy–delay tradeoff in a two-way relay with network coding. *Performance Evaluation* 70(11): 981–994.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press

19. Goseling, J., Boucherie, R.J., & van Ommeren., J.C.W (2016). A linear programming approach to error bounds for random walks in the quarter-plane. *Kybernetika (Prague)* 52(5): 757–784.
20. Grassmann, W.K. & Tavakoli., J. (2005). Two-stations queueing networks with moving servers, blocking, and customer loss. *Electronic Journal of Linear Algebra* 13: 72–89.
21. Gun, L. & Makowski, A.M. (1988). Matrix-geometric solution for finite capacity queues with phase-type distributions. In Courtouis, P.J. & Latouche, G., (eds.), *Performance '87*, Brussels, Belgium, 269–282.
22. Hajek., B.E. (1982). Birth-and-death processes on the integers with phases and general boundaries. *Journal of Applied Probability* 19(3): 488–499.
23. He, C., Meini, B., Rhee, N.H., & Sohraby., K. (2004). A quadratically convergent Bernoulli-like algorithm for solving matrix polynomial equations in Markov chains. *Electronic Transactions on Numerical Analysis* 17: 151–167.
24. Hillier, F.S. & So., K.C. (1995). On the optimal design of tandem queueing systems with finite buffers. *Queueing Systems* 21(3–4): 245–266.
25. Knessl, C. & Morrison., J.A. (2012). Asymptotic analysis of two coupled queues with vastly different arrival rates and finite customer capacities. *Studies in Applied Mathematics* 128(2): 107–143.
26. Kroese, D.P., Scheinhardt, W.R.W., & Taylor., P.G. (2004). Spectral properties of the Tandem Jackson network seen as a quasi-birth-and-death process. *The Annals of Applied Probability* 14(4): 2057–2089.
27. Latouche, G. & Ramaswami, V. (1999). *Introduction to Matrix-Analytic Methods in Stochastic Modeling*. Philadelphia: ASA-SIAM.
28. Latouche, G., Nguyen, G.T., & Taylor., P.G. (2011). Queues with boundary assistance and the many effects of truncations. *Queueing Systems* 69(2): 175–197.
29. Le Boudec, J.Y. (1991). An efficient solution method for Markov models of ATM links with loss priorities. *IEEE Journal on Selected Areas in Communications* 9(3): 408–417.
30. Li., S.Q. (1989). Overload control in a finite message storage buffer. *IEEE Transactions on Communications* 37(12): 1330–1338.
31. Miretskiy, D.I., Scheinhardt, W.R.W., & Mandjes., M.R.H. (2011). State-dependent importance sampling for a slowdown tandem queue. *Annals of Operations Research* 189(1): 299–329.
32. Miyazawa., M. (2009). Tail decay rates in double QBD processes and related reflected random walks. *Mathematics of Operations Research* 34(3): 547–575.
33. Perez, J.F. & Van Houdt., B. (2011). Quasi-birth-and-death processes with restricted transitions and its applications. *Performance Evaluation (Special issue QEST 2009)* 68(2): 126–141.
34. Perros., H.G. (1994). *Queueing networks with blocking*. Oxford, England: Oxford University Press, Inc.
35. Poloni., F. (2010) *Algorithms for quadratic matrix and vector equations*. PhD thesis, University of Pisa.
36. Shanthikumar, J.G. & Jafari., M.A. (1994). Bounding the performance of tandem queues with finite buffer spaces. *Annals of Operations Research* 48(2): 185–195.
37. van Dijk., N.M. (1987). A formal proof for the insensitivity of simple bounds for finite multi-server non-exponential tandem queues based on monotonicity results. *Stochastic Processes and Their Applications* 27: 261–277.
38. van Dijk., N.M. (1988). Simple bounds for queueing systems with breakdowns. *Performance Evaluation* 8(2): 117–128.
39. van Dijk., N.M. (1998). Bounds and error bounds for queueing networks. *Annals of Operations Research* 79: 295–319.
40. van Dijk, N.M. (2011). Error bounds and comparison results: The Markov reward approach for queueing networks. In Boucherie, R.J. & Van Dijk, N.M., (eds.), *Queueing Networks: A Fundamental Approach*, volume 154 of *International Series in Operations Research & Management Science*. Berlin, Germany: Springer.
41. van Dijk, N.M. & Lamond., B.F. (1988). Simple bounds for finite single-server exponential tandem queues. *Operations Research* 36(3): 470–477.
42. van Dijk, N.M. & Miyazawa., M. (2004). Error bounds for perturbing nonexponential queues. *Mathematics of Operations Research* 29(3): 525–558.
43. van Dijk, N.M. & Puterman., M.L. (1988). Perturbation theory for Markov reward processes with applications to queueing systems. *Advances in Applied Probability* 20(1): 79–98.
44. van Dijk, N.M. & van der Wal., J. (1989). Simple bounds and monotonicity results for finite multi-server exponential tandem queues. *Queueing Systems* 4(1): 1–15.
45. van Foreest, N.D., van Ommeren, J.C.W., Mandjes, M.R.H., & Scheinhardt., W.R.W. (2005). A tandem queue with server slow-down and blocking. *Stochastic Models* 21(2-3): 695–724.
46. van Vuuren, M., Adan, I.J.B.F., & Resing-Sassen, S.A.E. (2006). Performance analysis of multi-server tandem queues with finite buffers and blocking. In Liberopoulos, G., Papadopoulos, C.T., Tan, B., Smith, J.M., Gershwin, S.B. (eds.), *Stochastic Modeling of Manufacturing Systems*. Berlin, Germany: Springer, 169–192.

https://doi.org/10.1017/S0269964819000238 Published online by Cambridge University Press