Proceedings of the 2003 Winter Simulation Conference
S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds.

PROPERTIES OF DISCRETE EVENT SYSTEMS FROM THEIR MATHEMATICAL PROGRAMMING REPRESENTATIONS

Wai Kin Chan
Lee W. Schruben

Department of Industrial Engineering and Operations Research
University of California, Berkeley
4135 Etcheverry Hall
Berkeley, CA 94720 U.S.A.

ABSTRACT

An important class of discrete event systems, tandem queueing networks, are considered and formulated as mathematical programming problems where the constraints represent the system dynamics. The dual of the mathematical programming formulation is a network flow problem where the longest path equals the makespan of n jobs. This dual network provides an alternative proof of the reversibility property of tandem queueing networks under communication blocking. The approach extends to other systems.

1 INTRODUCTION

Schruben (2000) proposed a mapping of discrete event system models into mathematical programming formulations, where the solutions represent the system trajectories of the discrete event systems. The idea of this mapping is that a discrete event system is first modeled as an Event Graph (EG) (Schruben 1983), a powerful graphical representation for discrete event systems. Then the constraints of the system are derived from the edges of the EG along with the feasibility conditions of the state variables. It should be noted that there is another way of representing a discrete event system as a mathematical programming formulation which is based on a Petri Net Modeling technique (Yen 1999). However, since Event Graph Modeling (EGM) (Schruben and Schruben 2000) is more general than Petri Net Modeling (all Petri Net models can be transformed into EGs, but the reverse does not hold. See Schruben 2003), our derivation will be based on the EG representation. Perhaps the first reference on modeling queueing system dynamics as network optimization programs can be found in Maxwell and Wilson (1981).

The objective of this paper is two-fold: First, we show how the system dynamics of a tandem queueing network can be transformed into a set of linear programming (LP) constraints by means of EGM. Second, some properties of the system are investigated and proved using the linear programming formulation. Among these properties, the reversibility property is of special interest.

Specifically, we study a tandem queueing system with m consecutive stages, labeled $k = 1,..,m$. In each stage, there is a single server and a finite storage space for jobs. Job $i = 1,..,n$ is processed at all stages in sequence with service times s_{ki}. It is assumed that $\{(s_{1i}, s_{2i},\ldots,s_{mi}),\ i = 1,..,n\}$ are i.i.d. random variables.

Since the buffer spaces at each stage are limited, a control policy is needed to control the production process at the stages. The most common control policies that have been considered in the literature are the communication blocking control policy and the production blocking control policy (Buzacott and Shanthikumar 1993). In these two blocking schemes, a parameter a_k is associated with each stage, representing the maximum number of jobs at the stage at any time of the production process.

For the single server system, under communication blocking, the server at stage k will start processing a job whenever these three conditions are satisfied: C1) a job is available for processing, C2) the server at stage k is available, and C3) there is an empty space at the next stage. Under production blocking, the three conditions are: P1) a job is available for processing, P2) the server at stage k is available, and P3) there is no finished job blocked at stage k.

The only difference between these two control policies is the third condition: In communication blocking, the server will check the next stage for an empty buffer space before starting a service, therefore it is "*blocked before service.*" On the other hand, in production blocking, the server will check the next stage for an empty buffer space after finishing processing a job, therefore it is "*blocked after service*"; it may have at most one finished job blocked at the stage, which can be considered an additional buffer placed at the end of the stage: this makes production blocking superior to communication blocking in terms of throughput, all else being equal.

Cheng (1993) proposed another control policy called general blocking, which is an extension to Kanban blocking (Buzacott 1989). In general blocking, there are three parameters, a_k, b_k, and k_k, associated with each stage. They are, respectively, the maximum number of jobs waiting and in service, the maximum number of finished jobs allowed at the stage, and the maximum number of jobs allowed at the stage (including the jobs waiting, in service, and finished). To rule out triviality, the following conditions are necessary: $k_k \geq a_k \geq 1$, $k_k \geq b_k \geq 0$, and $b_k + a_k \geq k_k$ for $k = 1,..,m$. Despite its name, a system becomes more restrictive as its number of parameters increases.

Under general blocking, the server at stage k will start processing a job whenever these three conditions are satisfied: G1) a job is available for processing, G2) the server at stage k is available and G3) there are strictly less than b_k finished jobs blocked at stage k.

For all blocking schemes, blocking may happen from time to time, complicating the design and performance analysis of the system in terms of problem size or difficulty. In order to facilitate the analysis procedure, much research has been done to explore the structural properties of the system. Of particular interest is the well-known reversibility property: A tandem queue is said to be reversible if it has the same throughput as the corresponding reversed tandem queueing network (definitions in Section 3.2). Therefore, if the system is reversible, then the system properties of the original system can be inferred from those of the reversed system, thus simplifying the analysis process. Other motivations for studying the reversibility property of tandem queueing networks can be found in Chan and Schruben (2003).

The reversibility property of tandem queueing networks has been established and proved for communication blocking, production blocking (Yamazaki, Sakasegawa, and Kawashima 1978, Dattatreya 1978, and Muth 1979), as well as general blocking (Cheng 1995). All these proofs are based on the so-called "activity network," a directed graph with arcs representing the time required to process jobs in the system. In this paper, we propose another interesting proof of the reversibility property based on the dual network graph (a flow network) derived from the dual of a set of linear programming constraints representing the system dynamics. Since our dual graph is a network flow graph, algorithms and techniques in the field of network flows can now be used in the performance analysis of queueing systems: finding the makespan or bottleneck server can now be carried out by using the longest path algorithm (see Section 3.1 for details). Another potential application of our dual graph is that it can be used to prove the reversibility of multi-server tandem queues. This will be presented in a subsequent paper.

The paper is organized as follows. In Section 2, we derive the LP formulation of a tandem queueing network for communication blocking. We provide the proof of the reversibility property in Section 3. In Section 4, we remark on some ongoing research and possible future research directions.

2 MAPPING EVENT GRAPHS TO MATHEMATICAL PROGRAMMING FORMULATIONS

In this Section, we derive the LP formulation of a m-stage single-server tandem queue under communication blocking. For the derivation of production blocking and general blocking, see Chan and Schruben (2003).

Let TQmC($G/G/R_k/a_k$), $k = 1,..,m,$ denote a m-stage tandem queue with communication blocking, where stage k is a $G/G/R_k$ queue with raw job buffer a_k. Figure 1 is an example of 3-stage single server tandem queue with a communication blocking control policy.

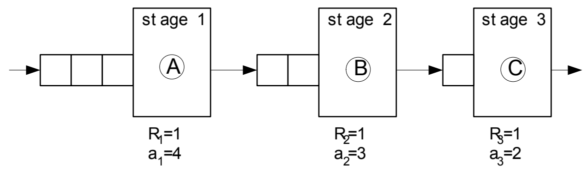

Figure 1: TQ3C($G/G/1/4, G/G/1/3, G/G/1/2$)

To model an m-stage tandem queue with n jobs being processed as an EG, the following notation is needed: ta_k is the time between the arrival of the i-1[th] job and the i^{th} job; s_{ki} is the service time of the i^{th} service at stage k; A_{ki}, S_{ki}, and F_{ki} are the time of the i^{th} *A*rrival event, *S*tart event, and *F*inish event at stage k respectively; $Q_k(t)$ is the number of jobs in front of stage k at time t; $R_k(t)$ is the number of available resources at stage k at time t; $a_k(t)$ is the number of available raw job spaces at stage k at time t; $C_E(t) = \lim_{\varepsilon \to 0} \max\{i; E_i \leq t + \varepsilon\}$ is the right-continuous event counting function that counts the number of times that event E has occurred by time t (see Schruben 2000 for a more detailed definition). Figure 2 is the EG of TQ3C($G/G/1/a_1, G/G/1/a_2, G/G/1/a_3$) where $R_1 = R_2 = R_3 = 1$. Notice that a_1 is irrelevant because we assume that there is an infinite input buffer in front of the first stage.

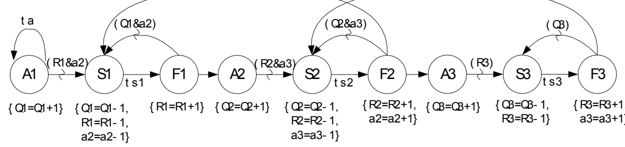

Figure 2: EG of TQ3C($G/G/1/a_1, G/G/1/a_2, G/G/1/a_3$)

2.1 Derivation of LP Formulation for TQ3C(.)

Following the method given in Schruben (2000) and using the Finish event times as the variables, we derive the LP formulation for the TQ3C(*G/G/1/a_1,G/G/1/a_2,G/G/1/a_3)* as the following.

2.1.1 Unconditional Edges:

In Figure 2, there are six unconditional edges, giving us the following six equations.

Edge A1-A1:	$A_{1,i+1} = A_{1i} + ta_i$	(1)
Edge S1-F1:	$F_{1i} = S_{1i} + s_{1i}$	(2)
Edge F1-A2:	$F_{1i} = A_{2i}$	(3)
Edge S2-F2:	$F_{2i} = S_{2i} + s_{2i}$	(4)
Edge F2-A3:	$F_{2i} = A_{3i}$	(5)
Edge S3-F3:	$F_{3i} = S_{3i} + s_{3i}$	(6)

2.1.2 Conditional Edges:

There are eight conditional edges and eight state variables. The two events associated with the conditional edge have only one state variable that gets incremented and decremented between them. Here, there are eight constraints (one constraint corresponds to the feasibility condition of a state variable). The first feasibility condition of the state variable Q_1 gives us the first constraint derived as follows: At any time of the simulation, since Q_1 is equal to the number of A1 events that have occurred minus the number of S1 events that have occurred, it can not be less than zero at any time t. Using the event counts $C_E(t)$ defined earlier, we have:

$$\begin{aligned} Q_1(t) = C_{A1}(t) - C_{S1}(t) &\geq 0 \\ C_{A1}(S_{1i}) - C_{S1}(S_{1i}) &\geq 0 \\ C_{A1}(S_{1i}) &\geq i \\ A_{1i} &\leq S_{1i} \end{aligned}$$

In the third inequality, we have used the fact that the number of S1 events that have occurred at time S_{1i} is i. The last inequality is because of the well-known relationship between an event and its counting point process, $C_{A1}(S_{1i}) \geq i \Leftrightarrow A_{1i} \leq S_{1i}$ (Ross 1997). Combining the last inequality and equation (2) gives the constraint,

$$F_{1i} \geq A_{1i} + s_{1i}$$

Similarly, the feasibility condition of state variable R_1 gives us the next constraint: At any time of the simulation, since R_1 is equal to the number of F1 events that have occurred minus the number of S1 events that have occurred plus 1, it cannot be less than zero at any time t (We add 1 because it is the original number of available resources at stage 1), we have:

$$\begin{aligned} R_1(t) = 1 + C_{F1}(t) - C_{S1}(t) &\geq 0 \\ 1 + C_{F1}(S_{1i}) - C_{S1}(S_{1i}) &\geq 0 \\ C_{F1}(S_{1i}) &\geq i-1 \\ F_{1,i-1} &\leq S_{1i} \end{aligned}$$

The third and fourth inequalities are derived in a similar manner. Combining the last inequality and equation (2) gives the constraint,

$$F_{1i} - F_{1,i-1} \geq s_{1i}$$

Using a similar procedure to derive constraints for the other state variables, along with the objective of executing all events as soon as possible (this objective function ensures that the server will work on the jobs as soon as possible), we end up with the following LP formulation.

<u>TQ3C-LP1:</u>

$$\min \sum_{i=1}^{n} \left(F_{1i} + F_{2i} + F_{3i} \right)$$

$st.$

F_{1i}	$\geq A_{1i} + s_{1i}$		$, i = 1,\ldots,n$	(U_{1i})
$F_{1i} - F_{1,i-1}$	$\geq$	s_{1i}	$, i = 2,\ldots,n$	(V_{1i})
$F_{1i} - F_{2,i-a_2}$	$\geq$	s_{1i}	$, i = a_2 + 1,\ldots,n$	(W_{12i})
$F_{2i} - F_{1i}$	$\geq$	s_{2i}	$, i = 1,\ldots,n$	(U_{2i})
$F_{2i} - F_{2,i-1}$	$\geq$	s_{2i}	$, i = 2,\ldots,n$	(V_{2i})
$F_{2i} - F_{3,i-a_3}$	$\geq$	s_{2i}	$, i = a_3 + 1,\ldots,n$	(W_{23i})
$F_{3i} - F_{2i}$	$\geq$	s_{3i}	$, i = 1,\ldots,n$	(U_{3i})
$F_{3i} - F_{3,i-1}$	$\geq$	s_{3i}	$, i = 2,\ldots,n$	(V_{3i})

where U_{ki}, V_{ki}, and $W_{k,k+1,i}$ are the corresponding dual variables for each constraint.

2.2 Extension to *m*-Stage Tandem Queue with Communication Blocking

The LP for 3-stage tandem queue with communication blocking is easily extended to an LP for *m*-stage tandem queue. The third, fourth, and fifth inequalities of TQ3C-LP1 are equivalent to the necessary and sufficient conditions C1, C2, and C3 introduced in Section 1. Therefore, the m-stage formulation should be based on these conditions, which leads to:

TQmC-LP1:

$$\min \sum_{k=1}^{m}\sum_{i=1}^{n} F_{k,i}$$

st.

$$F_{k,i} - F_{k-1,i} \geq s_{ki} \quad , k = 1,..,m, i = 1,...,n \quad (U_{k,i})$$
$$F_{k,i} - F_{k,i-1} \geq s_{ki} \quad , k = 1,..,m, i = 2,...,n \quad (V_{k,i})$$
$$F_{k,i} - F_{k+1,i-a_{k+1}} \geq s_{ki}, k = 1,..,m-1, i = a_{k+1}+1,...,n (W_{k,k+1,i})$$

where $F_{0i} = A_{1i}$, $i = 1,..,n$. We can also write the constraints as the following:

$$F_{k,i} = \max\left\{F_{k-1,i}, F_{k,i-1}, F_{k+1,i-a_{k+1}}\right\} + s_{ki}$$

which is the same as equation 5.38 in Buzacott and Shanthikumar (1993).

2.3 Comparison of Communication Blocking and Production Blocking

Some results given in Buzacott and Shanthikumar (1993) can be shown easily using the linear programming formulations (complete proofs are presented in Chan and Schruben 2003).

1. The departure time under communication blocking is not less than the departure time under production blocking.
   Proof:
   This result is immediate when one compares the two linear programs for these systems (see Chan and Schruben 2003 for the LP of production blocking). □
2. The departure time under production blocking with buffer capacities $\{a_1, a_2,\ldots, a_m\}$ is not less than the departure time under communication blocking with buffer capacities $\{a_1+1, a_2+1,\ldots, a_m+1\}$.
   Proof:
   This is true because the LP of production blocking with buffer capacities $\{a_1, a_2,\ldots, a_m\}$ has more constraints than the LP of communication blocking with buffer capacities $\{a_1+1, a_2+1,\ldots, a_m+1\}$, all else being equal. □
3. The throughput of the communication blocking tandem queue with buffer capacity $\{a_1, a_2,\ldots, a_m\}$ is not greater than that of the production blocking tandem queue with buffer capacity $\{a_1, a_2,\ldots, a_m\}$, which is, in turn, not greater than that of communication blocking tandem queue with $\{a_1+1, a_2+1,\ldots, a_m+1\}$.
   Proof:
   The proof is based on the above first and second results. □

3 REVERSIBILITY OF TANDEM QUEUEING NETWORKS

The well-known reversibility property of single-server tandem queues has been proven by many researchers using different methods. Many of the proofs are based on a characterization of the activity network associated with the queue, see Yamazaki, Sakasegawa, and Kawashima (1978), Dattatreya (1978), Muth (1979), Liu and Buzacott (1992) and Cheng (1995). An activity network for a tandem queue is a directed graph, representing the time required to process n jobs.

In this Section, we propose a dual network graph (a flow network) that yields not only the proof of the reversibility but also some important information on the system, e.g. the number of jobs in each busy period. Our dual network is a network flow model derived directly from the dual linear programming formulation, not from the activity of the system, which makes it fundamentally different from the activity networks in the literature. Because of its mathematical form, it can also be used in the performance analysis of the queueing system.

3.1 Dual Network (Flow Network)

To derive the dual network using the 3-stage communication blocking tandem queue as an example, we first take the dual of TQ3C-LP1. Since all service times are positive (except for a dummy server who has zero service time, which actually can be expressed as an empty buffer), all the F_{ki}, $k = 1,..,m$, $i = 1,..,n,$ in the primal are strictly positive, forcing all the inequality constraints in the dual to be equalities. By examining the structure of the dual, one can see that it is, in fact, the network flow problem of finding the longest paths from the first node (source) to all other nodes (sink), where each constraint represents the flow conservation constraint of a node in the network. Since the constraints are equality, summing over all constraints will yield the flow balance constraint for the source node, $\sum_{i=1}^{n} U_{1i} = m*n$. Therefore, a total of $m*n$ units of flow (one unit flow is one Finish event) will be sent from the source node to all other $m*n$ sink nodes (one event for each sink node).

The dual LP and the dual network of TQ3C-LP1($G/G/1/\bullet, G/G/1/3, G/G/1/2$) are given in the following and in Figure 3 respectively:

TQ3C-LP1-Dual:

$$\max \quad \sum_{i=1}^{5}(A_{1i}+s_{1i})U_{1i}+\sum_{i=2}^{5}s_{1i}V_{1i}+\sum_{i=4}^{5}s_{1i}W_{12i}+\sum_{i=1}^{5}s_{2i}U_{2i}+\sum_{i=2}^{5}s_{2i}V_{2i}+\sum_{i=3}^{5}s_{2i}W_{23i}+\sum_{i=1}^{5}s_{3i}U_{3i}+\sum_{i=2}^{5}s_{3i}V_{3i}$$

st.

$$\begin{aligned}
&U_{11}-U_{21}-V_{12} &&=1\\
&U_{12}-U_{22}+V_{12}-V_{13} &&=1\\
&U_{13}-U_{23}+V_{13}-V_{14} &&=1\\
&U_{14}-U_{24}+V_{14}-V_{15}+W_{124} &&=1\\
&U_{15}-U_{25}+V_{15}+W_{125} &&=1\\
&U_{21}-U_{31}-V_{22}-W_{124} &&=1\\
&U_{22}-U_{32}+V_{22}-V_{23}-W_{125} &&=1\\
&U_{23}-U_{33}+V_{23}-V_{24}+W_{233} &&=1\\
&U_{24}-U_{34}+V_{24}-V_{25}+W_{234} &&=1\\
&U_{25}-U_{35}+V_{25}+W_{235} &&=1\\
&U_{31}-V_{32}-W_{233} &&=1\\
&U_{32}+V_{32}-V_{33}-W_{234} &&=1\\
&U_{33}+V_{33}-V_{34}-W_{235} &&=1\\
&U_{34}+V_{34}-V_{35} &&=1\\
&U_{35}+V_{35} &&=1\\
&U_{ik}\geq 0 \quad , i=1,\dots,5, k=1,\dots,3
\end{aligned}$$

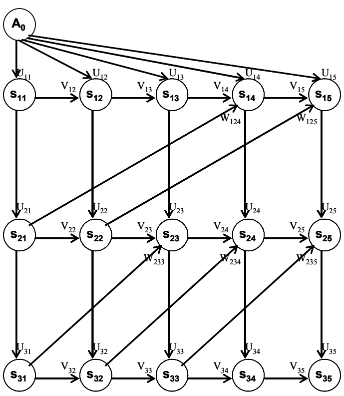

Figure 3: Dual Network Graph of TQ3C-LP1($G/G/1/\bullet, G/G/1/3, G/G/1/2$)

There are 15 constraints in the dual LP, consequently there are 15 sink nodes in the dual graph. The constraint for the source node (node A_0) is obtained by summing over all 15 constraints, which is $\sum_{i=1}^{5}U_{1i}=15$. Graphically, node A_0 generates 15 units of flow and sends them out along its 5 incident arcs to nodes s_{11}, s_{12}, s_{13}, s_{14}, and s_{15}. If jobs arrive at the system according to a random process, i.e. $\{A_{1i}, i = 1,..,5\}$, then the length of each of these 5 arcs equals the arrival time of the corresponding job, i.e. the length of arc (A_0, s_{13}) is A_{13}, the time epoch of the 3rd job's arrival. If all jobs are waiting at stage 1 when the system starts (at time 0), then the length of these five arcs will be zero. In this case, U_{12}, U_{13}, U_{14}, and U_{15} will be zero and U_{11} will be 15 because the server can only serve one job at a time and thus arcs (A_0, s_{12}), (A_0, s_{13}), (A_0, s_{14}), and (A_0, s_{15}) can be eliminated. This is important in proving the property of reversibility.

The length of all other arcs equals the service time of the corresponding job at that stage, i.e. the length of arc (s_{22}, s_{23}) equals s_{23}, the service time of the 3rd job at stage 2. This is also true for arcs (s_{13}, s_{23}) and (s_{31}, s_{23}), meaning that all arcs that enter the same node have the same length. Hence, we can modify the network slightly by letting the "length" of each node equal the service time of the corresponding job at that stage, i.e. the "length" of node s_{23} equals s_{23}, the service time of the 3rd job at stage 2. This change will facilitate our proof of reversibility later.

There are three types of flows in the network, U_{ki}, V_{ki}, and $W_{k,k+1,i}$. U_{ki} is the flow from stage k to stage k+1; V_{kz} is the flow within stage k; and $W_{k,k+1,i}$ is the flow from stage k+1 back to stage k. The primal formulation tells us that one and only one constraint regarding F_{ki} is binding if there is no time tie. However, if there is a time tie, we can arbitrary pick one of the constraints that are blinding (e.g. always pick the server with the smallest index). This will not change our proof. Therefore, one and only one of U_{ki}, V_{ki}, and $W_{k,k+1,i}$ is positive.

The flow of each arc is shown at the end of the arc, i.e. W_{233} is the flow from node s_{31} to node s_{23} and its value represents the number of tasks (a "task" is finishing processing a job at one stage, e.g. a job has three tasks in a 3-stage tandem queue) within the same busy period. If a service time is changed, it may affect several tasks. For example, if $W_{233} = 4$, a change in the service time s_{23} will affect 4 tasks, including the 3rd job itself.

3.2 Reversibility of Communication Blocking Tandem Queues

We start the proof by giving some definitions related to reversibility. All notation in this Section with a "~" refers to the reversed system.

3.2.1 Definition 3.1.

The *throughput, TH,* of a tandem queueing network is defined as the number of jobs per unit time released from the last station in the long run: $TH = \lim_{n\to\infty} \frac{n}{E\left[F_{mn}\right]}$.

3.2.2 Definition 3.2.

Given a tandem queueing network, the corresponding *reversed tandem queueing network* is obtained by reversing the order of all stations.

For communication blocking, since a_k is defined as the buffer between stage k-1 and stage k plus the one space for a job during service, this reversal means $\tilde{a}_k = a_{m+2-k}$, k = 2,..,m.

3.2.3 Definition 3.3.

A tandem queueing network is said to be *reversible* if it has the same throughput as its corresponding reversed tandem queueing network.

In the following, we will show that, under job reversal ($\tilde{s}_{ki} = s_{m+1-k,n+1-i}, k = 1,..,m, i = 1,..,n$), $\tilde{F}_{mn}$ is identical to F_{mn}. So, if the service times $\{(s_{1i}, s_{2i},\ldots, s_{mi}), i = 1,..,n\}$ are i.i.d. random variables, the completion time of processing n jobs in the reversed queue has the same distribution as the completion time of processing n jobs in the original queue (i.e. $\text{Prob}(\tilde{F}_{mn} \le x) = \text{Prob}(F_{mn} \le x)$ for any positive integer n and real number x), thus the throughputs of the two systems are the same and consequently the reversibility property is proved.

Now assume that, at time 0, there are n jobs waiting for processing at stage 1 and there are no jobs at the other stages. Using the argument in Section 3.1, we can eliminate all arcs emanating from A_0 except the one that ends at node s_{11}. The dual problem is to find the longest paths from node s_{11} to all other nodes, especially to node s_{35}, which is the makespan for processing all jobs and is usually called the critical path.

Notice that since exactly one of U_{ki}, V_{ki}, and $W_{k,k+1,i}$ is positive, for each node in the network there is exactly one arc with positive flow going into it (the others have zero flow), and the longest path from the beginning node to the last node is continuous and unique (this can be easily seen if one starts from the last node, going backward to the first).

To show that the queue is reversible, we reverse the order of all stages and let the jobs enter the system in the reversed order. This is same as reversing the direction of all arrows and letting the jobs enter the system from the last node, node s_{35}. It is obvious that this reversal does not change the structure of the network. The reversed dual problem is now to find the longest paths form node s_{35} to all other nodes. Since the structure of network does not change, the longest path from node s_{35} to node s_{11} remains unchanged. However the longest paths from s_{35} to other nodes in the reversed network are not the same as the longest paths starting from s_{11} in the original network. For example, the length of the path from node s_{35} to node s_{23} in the reversed network is not equal to the length of the path from node s_{11} to node s_{23} in the original network. But these differences do not affect the makespan of processing n jobs. The longest path also reveals the bottleneck servers.

The fact that the structure of the reversed dual network is the same as that of the original dual network can be verified by reversing the order of stages and the service times in formulation TP3C-LP1 (the reversed queue is denoted as TQ3CR-LP1(*G/G*/1/•,*G/G*/1/3,*G/G*/1/2), where the "*R*" stands for "*R*eversed"), taking the dual of it, and drawing the dual network graph from that reversed formulation; this is also a tandem queue and can be expressed as TQ3C-LP1(*G/G*/1/•,*G/G*/1/2,*G/G*/1/3). Figure 4 shows that structure remains unchanged.

Therefore, we have $\tilde{F}_{mn} = F_{mn}$, which implies the tandem queue is reversible.

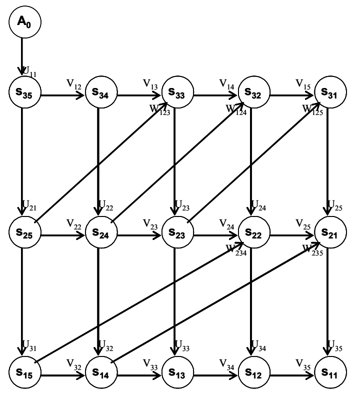

Figure 4: Dual Network Graph of TQ3CR-LP1(*G/G*/1/•,*G/G*/1/3,*G/G*/1/2)

3.3 Reversibility of Production Blocking and General Blocking Tandem Queues

Cheng (1995) gives a sufficient condition for the reversibility of a general blocking tandem queue, which is the interchangeability of a_k and b_k ($b_k = a_k$). This condition is restrictive. In Chan and Schruben (2003), the authors examine the reversibility of production blocking and general blocking tandem queues and provide a more relaxed condition for the reversibility of a general blocking tandem queue, which is stated as the following:

- *Condition:* A general blocking tandem queue is reversible if it satisfies the following condition:

$$a_k - b_k = Z, \quad k = 1,..,m,$$

where Z is an arbitrary integer. This condition implies Cheng (1995)'s condition when Z is 0. The intuitive interpretation of this condition is that, as long as the buffer size between stage k-1 and stage k in the reversed queue is same as the buffer size between stage m+1-k and stage m+2-k in the original queue, the blocking scenario is the same and thus the queue is reversible.

4 CONCLUSION AND EXTENSION

Although we study only single-server tandem queues here, it is possible to extend our LP method to do the performance analysis of multi-server tandem queues. It is a potential tool for proving the reversibility of multi-server tandem queues. Another possible use of our methodology is to prove the reversibility of closed tandem queueing networks.

While we focus our attention on tandem queueing networks, the method given in Section 2 can also be used in deriving the mathematical programming formulation for other discrete event systems.

The mathematical programming representation of discrete event systems makes it possible for us to use the techniques or algorithms of mathematical programming for the analysis of discrete event systems; i.e. when doing a simulation of a tandem queue, one may solve the mathematical programming problem, find the bottleneck server, and do sensitivity analysis using the dual variables.

ACKNOWLEDGMENTS

UC Berkeley graduate students Theresa Roeder and Debbie Pederson provided helpful suggestions and criticisms. This work was supported in part by the Semiconductor Research Corporation and the National Science Foundation.

REFERENCES

Buzacott, John A., "Queueing models of Kanban and MRP controlled production systems", *Engineering cost and Production Economics,* 17(1989), 3-20.

Buzacott, John A. and Shanthikumar, J. George, "Stochastic Models of Manufacturing Systems", New Jersey: Prentice Hall, 1993.

Chan, Wai Kin and Schruben Lee W., "Reversibility of Tandem Queueing Networks From Their Mathematical Programming Representations", *Technical Report,* IEOR Department, UC Berkeley, CA, 2003.

Cheng, Dinah W., "Tandem Queues with General Blocking: A Unified Model and Comparison Results", *J. Discrete Event Dynamic Systems: Theory and Applications*, 2 (1993), 207-234.

Cheng, Dinah W., "Line Reversibility of Tandem Queues with General Blocking", *Management Sci.* 41(1995), 864-873.

Dattatreya, Eswarahalli Sundararajan, "Tandem Queueing Systems with Blocking", Ph.D. dissertation, IEOR Department, University of California, Berkeley, CA, January 31, 1978.

Liu, Xiao-Gao and Buzacott, John A., "The Reversibility of cyclic queues", *Operations Research Letters*, 11(1992), 233-242.

Maxwell, W. L. and R. C. Wilson (1981). "Dynamic Network Flow Modeling of Fixed Path Material Handling Systems." *IIE Transactions* 13(1): 12-21.

Muth, Eginhard J., "The Reversibility Property of Production Lines", *Management Sci*., 25(1979), 152-158.

Ross, Sheldon M., "Introduction to Probability Models", Six Edition, 1997, Academic Press.

Schruben, Lee W., "Simulation Modeling with Event Graphs", *Communications of the Association of Computing Machinery,* 26(11), 1983, 957-963.

Schruben, Lee W., "Mathematical Programming Models of Discrete Event System Dynamics", *2000 Winter Simulation Conference Proceedings*, IEEE. Part vol.1, 2000, pp.381-5 vol.1.

Schruben, D and Schruben, Lee W., Graphical Simulation Modeling using SIGMA, Custom Simulation, 2000.

Schruben, Lee W., "Conditional Parametric Petri Nets and their Mapping to Simulation Event Graphs", submitted to *Europe Simulation and Modeling Conference*, Italy Oct. 2003.

Yamazaki, Genji, Sakasegawa, Hirotaka and Kawashima, Takeshi, "Production Rate Estimated with Flow-Shop Reversibility", *Bulletin of the JSME*, 21(1978), 167-171.

Yamazaki, Genji, Sakasegawa, Hirotaka and Shanthikumar, J. George, "On Optimal Arrangement of Stations in a Tandem Queueing System with Blocking", *Management Science,* 38.1(1992), 137-153.

Yen, H.-C., "Integer Linear Programming and the Analysis of Some Petri Net Problems", Theory Comput. Systems 32, 467-485(1999)

AUTHOR BIOGRAPHIES

WAI KIN CHAN is a PhD student in the Industrial Engineering and Operations Research Department at the University of California, Berkeley. His research interests include designs of simulation experiments, queueing theory, and computational optimization. His e-mail is <kin@ieor.berkeley.edu> and his web address is <www.ieor.berkeley.edu/~kin>.

LEE W. SCHRUBEN is the Chairman of the Department of Industrial Engineering and Operations Research at the University of California, Berkeley. He joined the faculty at UC Berkeley after spending 25 years as a member of the School of Operations Research and Industrial Engineering at Cornell where he held the Andrew S. Schultz Endowed Professorship. His research is primarily on discrete event simulation modeling methodology and in the design and statistical analysis of simulation optimization experiments. His e-mail is <schruben@ieor.berkeley.edu>