PERFORMANCE ANALYSIS OF A MULTIPLE VEHICLE TANDEM SYSTEM WITH INTER-VEHICLE BUFFERS AND BLOCKING

Matthew E. H. Petering[1], Jaeyoung Seo[2], and Chulung Lee[3]

1 ISE Department, National University of Singapore, (65) 6874-1929, isemehp@nus.edu.sg
IOE Department, University of Michigan—Ann Arbor, mpeterin@umich.edu
2 ISE Department, National University of Singapore, (65) 6874-4606, isesjy@nus.edu.sg
3 ISE Department, National University of Singapore, (65) 6874-6431, iseleecu@nus.edu.sg

Abstract: *We analyze a new tandem system that consists of a series of vehicles that move back and forth between each other, passing goods forward along a chain from an origin to a destination. Each link in the chain is maintained by a single vehicle with unit capacity that travels between two fixed points—an upstream point and a downstream point. Each vehicle travels with an item in the forward (downstream) direction and without an item in the reverse (upstream) direction. Buffers of non-identical capacities are allowed between vehicles. The system is mathematically equivalent to a multi-station, reliable series production line where each station has an exponential service time followed by exponential cleaning time. An item is free to leave a station when service is complete, but the station cannot begin processing another item until both service and cleaning are complete. We model this system as a continuous time Markov chain and develop a recursive algorithm for generating the state transition matrix. After solving for the steady state probabilities using the Gauss-Seidel method, we compute the average throughput rate, average number of items in the system, and average sojourn time for various system configurations.*

Keywords: *tandem system, production line, material handling system, vehicle, buffer, queue, throughput*

1. Introduction and Literature Review

In recent years much academic research has focused on the design and analysis of manufacturing systems. Overviews of this research are provided by Papadopoulos and Heavey [4] and Govil and Fu [2]. Material handling systems have also been heavily studied. Ventura and Lee [8] is an example of recent research in this area. Although the literature on these topics now comprises hundreds of articles, there are still relatively few exact analytical results for manufacturing systems or material handling systems of arbitrarily large size. The articles which do obtain such results typically use ideas from queueing theory to study serial production lines with blocking phenomena. Papadopoulos and O'Kelly [7] are the first researchers to present a general recursive method for generating the continuous time Markov chain (CTMC) state transition matrix for such systems. They assume exponential service times at all stations and no buffers between stations. Papadopoulos *et al.* [5,6] extend these results to allow for finite buffers of non-identical capacities between stations and Erlang distributed service times. More recently, Heavey *et al.* [3] analyze the performance of a production line where stations may be either reliable or unreliable. The time to failure for an unreliable machine is exponentially distributed and the repair time is Erlang distributed. Also, Vidalis and Papadopoulos study a reliable series production line where the service time at each station follows a Coxian-2 distribution [9]. All of these studies, however, assume machines are stationary. The movement of items from one machine to the next is not explicitly modeled.

The present paper, on the other hand, obtains exact results for a tandem system in which the physical transport of items is explicitly modeled. The system can be interpreted as a manufacturing system or a material handling system; it consists of a series of vehicles that move back and forth between each other, passing goods forward along a chain from an origin to a destination. Each link in the chain is maintained by a single vehicle with unit capacity that travels between two fixed points—an upstream point and a downstream point. The vehicle travels with an item at a Poisson rate in the downstream direction and without an item at a (possibly different) Poisson rate in the upstream direction. Buffers of non-identical capacities are allowed between vehicles. We model this system as a continuous time Markov chain and develop a recursive algorithm for generating the state transition matrix. After solving for the steady state probabilities using the Gauss-Seidel method, we compute the average throughput rate, average number of items in the system, and average sojourn time for various system configurations.

The system studied here is similar to a multi-station reliable series production line where service times follow the type-2 Erlang distribution [6] or the Coxian-2 distribution [9]. However, the system studied here is *mathematically* different than these systems. In particular, the mechanics by which a totally blocked-up system becomes unblocked are quite unique for the system studied here. In [6] and [9], consecutive stations that are blocked become unblocked simultaneously when the station ahead of them finally finishes service. In the present paper, however, the "stations" are actually vehicles that travel back and forth. Therefore, consecutive vehicles that are blocked (i.e. are idle at their forward positions) do not become unblocked simultaneously when the vehicle ahead of them finally returns to pick up another item. Instead, blocked vehicles are "unblocked" one at a time in a cascade that begins with the last blocked vehicle, then the 2nd last blocked vehicle, then the 3rd last blocked vehicle, etc.

The system studied here is mathematically equivalent to a multi-station, reliable series production line where each station has an exponential service time followed by exponential cleaning time. An item is free to leave a station when service is complete, but the station cannot begin processing another item until cleaning is complete. Such a system is similar to, but not equivalent to, the unreliable system in [3].

Overall, the authors of the present paper tried unsuccessfully to locate any paper which explicitly modeled the movement of items through a tandem system in a fashion similar to that done here. In addition, we did not find any model of a tandem queueing system that could be mathematically translated into, or qualitatively reinterpreted as, the multiple vehicle tandem system considered here.

2. System Description

The system we study is as follows. An infinite population of items is to be transported from an origin to a destination. At time 0, all items are waiting at the origin. In order to reach the destination, the items must pass through a series of N intermediate buffers ($N \geq 1$). Every item passes through each of the intermediate buffers in the same fixed sequence. Let "Buffer 1" refer to the buffer closest to the origin, "Buffer 2" refer to the buffer second-closest to the origin, and so on, so that "Buffer N" refers to the N^{th}, or final, intermediate buffer. These buffers can either be thought of as passenger stations in a transportation network or as buffers in an assembly line that lie between machines that have to be cleaned every time they are used. In this paper, we adopt the former interpretation of the system.

A fleet of $N+1$ vehicles is responsible for transporting the items from the origin to the destination. Each vehicle in the fleet connects a unique pair of locations. Vehicle 1 is responsible for transporting items from the origin to Buffer 1, Vehicle 2 is responsible for transporting items from Buffer 1 to Buffer 2, and so on, so that Vehicle $N+1$ is responsible for transporting items from Buffer N to the destination. Each vehicle can transport only one item at a time. Vehicle i's pure travel time in the forward (downstream) direction, from (the origin/Buffer $i-1$) to (Buffer i/the destination), is an exponential random variable with mean λ_i^+, rate R_i^+ ($1/\lambda_i^+ = R_i^+$). The vehicle always carries an item during this journey. The vehicle's pure return travel time from (the destination/Buffer i) to (Buffer $i-1$/the origin) is an exponential random variable with mean λ_i^-, rate R_i^- ($1/\lambda_i^- = R_i^-$). During this journey, the vehicle is always empty. All pure travel times are independent of each other. Overall, the pure travel times of all vehicles are encapsulated by $2N+2$ parameters.

Items may be blocked during various stages of their journey from the origin to the destination. In particular, each buffer has a finite capacity, i.e. a maximum number of items that can be occupying the buffer at any given time. Let C_i be the capacity of Buffer i ($i=1, 2, ..., N$). Normally, we assume that the loading of items onto vehicles and the discharging of items from vehicles happen instantaneously. However, if a vehicle is attempting to drop off an item at a buffer that is already filled to capacity, both the item and vehicle must wait together beside the buffer until room to accommodate the item becomes available. The amount of time spent waiting is exactly equal to the time that elapses before the vehicle that is next in sequence returns to the buffer to pick up an item. In other words, if Vehicle i finds Buffer i full upon its arrival (i.e. there are already C_i items at Buffer i), Vehicle i must wait until Vehicle $i+1$ returns before it can drop off its item. The loading of Vehicle $i+1$ and unloading of Vehicle i then happen simultaneously at Buffer i, at the exact instant when Vehicle $i+1$ arrives. If $C_i=0$, the above still takes place and we imagine that the item is instantaneously transferred from Vehicle i to Vehicle $i+1$.

Overall, there are two types of vehicle idling in this system: drop-off idling and pick-up idling. As mentioned above, idling of the former type occurs because of buffer congestion and is directly related to the blocking of items. Idling of the latter type occurs because buffers may be empty and therefore no immediate pick-up is possible; such idling does not correspond to the blocking of items. In this paper, we assume that $C_0=C_{N+1}=\infty$. This means that Vehicle 1 never experiences pick-up idling and Vehicle $N+1$ never experiences drop-off idling. The overall system is depicted in Figure 1.

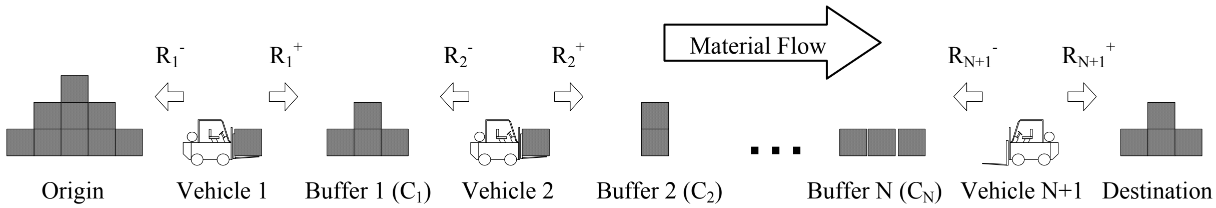

Figure 1: A Multiple Vehicle Tandem System with Inter-vehicle Buffers

3. A CTMC Model of the System

Our analysis closely follows [3], [5], [6], [7], and [9]. In these papers, the authors look at various tandem queueing systems. They model these systems using CTMCs and then develop a procedure to automatically generate the state balance equations, solve for the steady state probabilities, and compute performance measures for different system configurations. Here, we do the same for the system at hand.

A continuous time Markov chain representation of the system at hand is straightforward. First of all, we represent the status of each component of the system (vehicle or buffer) with a single letter or number. The status of a buffer at any moment is the number of items occupying the buffer. The status of a vehicle is given by one of the four terms M^+, M^-, I^+, I^-. The meaning of these terms is given below.

Vehicle status	Meaning
M^+	moving downstream, towards the drop-off location
M^-	moving upstream, towards the pick-up location
I^+	idle at downstream location (i.e. waiting to drop off a container)
I^-	idle at upstream location (i.e. waiting to pick up a container)

Let v_1, v_2, ..., v_{N+1} be the status of Vehicles 1, 2, ..., $N+1$ respectively and b_1, b_2, ..., b_N be the status of Buffers 1, 2, ..., N respectively. Then we can write the state of the entire system as (v_1, b_1, v_2, b_2, ..., v_N, b_N, v_{N+1}). In other words, we can write the state of the entire system by listing the status of its components in downstream order from left to right, omitting the status of the origin and destination. For example, consider a system with $N=2$, $C_1=2$, $C_2=4$. This system has 3 vehicles, an origin, two intermediate buffers, and a destination. The following holds if the system is in state (M^+, 1, I^+, 4, M^-): Vehicle 1 is moving an item from the origin to Buffer 1; Buffer 1 holds one item; Vehicle 2 is waiting to drop off an item at Buffer 2; Buffer 2 holds 4 items; and Vehicle 3 is empty and is moving upstream towards Buffer 2. With the above notation, the state of the system is well-defined. The state changes when a vehicle completes its journey in either the downstream or upstream direction. Since all vehicle journeys are exponentially distributed, the process of transitioning between states is a competing exponentials process that is memoryless. The evolution of the system therefore follows the Markov property and we have a continuous time Markov chain.

4. Computing the Number of CTMC States

The number of states in the CTMC representation of the system depends on the number of vehicles and buffer sizes. Theorem 1 gives a recursive method for computing the number of states for any system.

Theorem 1. Let $S_{n+1}^{C_1C_2\cdots C_n}$ be the number of states in the CTMC representation of a system with n+1 vehicles and buffer capacities $C_1, C_2, ..., C_n$ from the origin to the destination respectively. Assuming $S_0 = 0$ and $S_1 = 2$, the following is true for all $n \geq 1$:

$$S_{n+1}^{C_1C_2\cdots C_n} = (2C_1 + 4)(S_n^{C_2C_3\cdots C_n}) - S_{n-1}^{C_3C_4\cdots C_n}\text{ . } \blacksquare$$

Theorem 1 is used to calculate the number of states for the systems in Table 1. Table 1 shows that the number of states increases tremendously as the number of buffers and the buffer capacities increase.

Table 1: Number of States in the CTMC Representation of the System

Number of buffers	Size of each buffer								
	0	1	2	3	4	5	6	7	8
0	2	2	2	2	2	2	2	2	2
1	8	12	16	20	24	28	32	36	40
2	30	70	126	198	286	390	510	646	798
3	112	408	992	1960	3408	5432	8128	11592	15920
4	418	2378	7810	19402	40610	75658	129538	208010	317602

5. Recursive Method for Generating the CTMC State Transition Matrix

We found a recursive method for generating the transpose of the CTMC state transition matrix for any system configuration. The method begins by creating the transition matrix corresponding to a single-buffer subsystem of the original system and then recursively creates the transition matrix for a (B+1)-buffer subsystem based on the transition matrix for the B-buffer subsystem for $B=1,2,...,N-1$.

Before continuing, we define the terms "B-buffer subsystem" and "A_B." The term "B-buffer subsystem" refers to the (B+1)-vehicle, B-buffer system whose vehicle rates are equivalent to the rates of the *final* B+1 vehicles in the original system (from origin to destination respectively) and whose buffer capacities are equivalent to the capacities of the *final* B buffers in the original system (from origin to destination respectively). A_B represents the transpose of the transition matrix for the B-buffer subsystem.

Due to space limitations, our method for generating A_1 cannot be explained in detail here. However, it is important to mention that the method orders the states in a special way so that all transitions with the same rate parameter appear on the same diagonal of A_1. A_{i+1} is constructed recursively from A_i. The overall procedure for constructing A_N, the transition matrix for the entire N-buffer system, is as follows:

1) Create A_1, the transpose of the transition matrix for the 1-buffer subsystem. For the time being, omit the main diagonal (i.e. negative) entries of this matrix. The columns of this matrix should correspond to the sources of the transitions and the rows should correspond to the states that result from the transitions. In other words, row r column c of the matrix should be the rate at which State c transitions into State r. All entries on the main diagonal should be zero. Let B = 1.
2) If $B=N$, go to Step 8. Otherwise, let D_B be equal to A_B except that the entries with rate R_{N-B+1}^- are made zero. (These entries are in the top (1/2)($S_{B+1}^{C_{N-B+1}C_{N-B+2}\cdots C_N} - S_B^{C_{N-B+2}C_{N-B+3}\cdots C_N}$) rows of A_B..)
3) Let E_B be the matrix that is equal to the top (1/2)($S_{B+1}^{C_{N-B+1}C_{N-B+2}\cdots C_N} - S_B^{C_{N-B+2}C_{N-B+3}\cdots C_N}$) rows of A_B, with all entries except those equal to rate parameter R_{N-B+1}^- made zero.

4) Let F_B be R_{N-B}^- multiplied by the identity matrix that has $(C_{N-B}+1)(S_{B+1}^{C_{N-B+1}C_{N-B+2}\cdots C_N}) + (1/2)(S_{B+1}^{C_{N-B+1}C_{N-B+2}\cdots C_N} - S_B^{C_{N-B+2}C_{N-B+3}\cdots C_N})$ rows and columns.
5) Let G_B be R_{N-B}^+ multiplied by the identity matrix that has $(C_{N-B}+1)(S_{B+1}^{C_{N-B+1}C_{N-B+2}\cdots C_N}) + (1/2)(S_{B+1}^{C_{N-B+1}C_{N-B+2}\cdots C_N} - S_B^{C_{N-B+2}C_{N-B+3}\cdots C_N})$ rows and columns.
6) Let H_B be the square sub-matrix of A_B that consists of the $(1/2)(S_{B+1}^{C_{N-B+1}C_{N-B+2}\cdots C_N} - S_B^{C_{N-B+2}C_{N-B+3}\cdots C_N})$ upper-left-most rows and columns of A_B.
7) Construct A_{B+1} by pasting together the matrices constructed in Steps 2 through 6 according to Figure 2. In particular, the diagonal portion of A_{B+1} should consist of $C_{N-B}+1$ consecutive copies of D_B, followed by H_B, followed by $C_{N-B}+2$ consecutive copies of D_B, followed by H_B. Also, the $(1/2)(S_{B+1}^{C_{N-B+1}C_{N-B+2}\cdots C_N} - S_B^{C_{N-B+2}C_{N-B+3}\cdots C_N})$ rows directly beneath each D_B should be equal to E_B and the upper-right portion of A_{B+1} should be equal to F_B. Finally, G_B should be pasted into A_{B+1} so that G_B's upper-left element corresponds to the element in Row $w+1$, Column 1 of A_{B+1}, where w is the final row occupied by F_B. A nice property of A_{B+1} is that all transitions with the same rate parameter appear on the same diagonal of A_{B+1}. Increase B by 1 and return to Step 2.
8) To make the matrix a proper transition matrix, make each diagonal entry in A_N equal to the negative of the sum of all elements in the same column as the diagonal entry. A_N now equals the transpose of the CTMC state transition matrix for the N-buffer system. A nice property of A_N is that all transitions with the same rate parameter appear on the same diagonal of A_N.

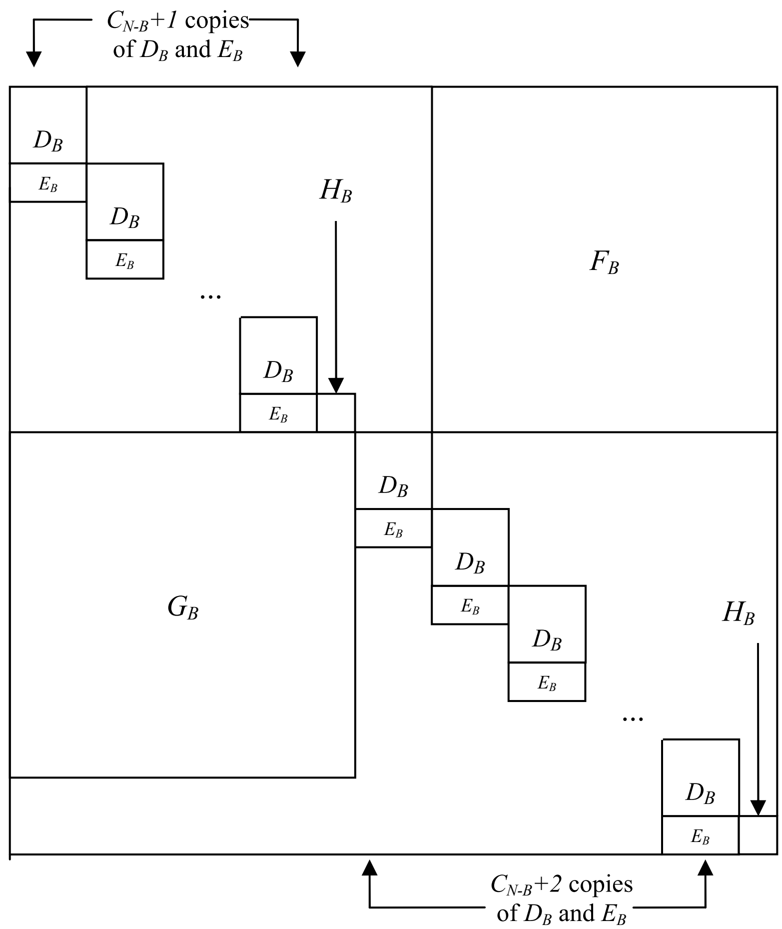

Figure 2: A_{B+1}, the state transition matrix for the $(B+1)$-buffer subsystem, is constructed out of various components of A_B, the state transition matrix for the B-buffer subsystem.

6. Computing the Steady State Probabilities and Various System Performance Measures

The recursive procedure above generates the CTMC state transition matrix for any system configuration, i.e. for any values of the parameters N, C_1, C_2, ..., C_N, R_1^+, R_1^-, R_2^+, R_2^-, ..., R_{N+1}^+, R_{N+1}^-. Let A be the CTMC state transition matrix for the overall system (i.e. $A = A_N$), x be the column vector of steady state probabilities (i.e. x_i is the steady state probability of being in State i), and e be a column vector of size equal to x with each entry equal to 1. Then the steady state probability vector x satisfies Ax=0, e^Tx=1, where one of the equations from matrix A is redundant. Once the steady state probabilities are in hand, T, the average throughput rate of the system, can be computed as follows. For any i from 1 to N+1, the average throughput rate of the system is the rate at which Vehicle i moves forward times the steady state probability that Vehicle i is moving forward:

$$T = (R_i^+) * \text{Prob}(v_i = \text{M}^+) \qquad (1).$$

L, the average number of items in the system, is the steady state probability of being in state i times the number of items in the system when the system is in state i, summed over all states i:

$$L = \sum_{i=1}^{S_{N+1}^{C_1C_2\cdots C_N}} (x_i) * (\text{number of items in system for State } i) \qquad (2).$$

W, the average amount of time an item spends in the system (i.e. the average sojourn time for an item), is computed from T and L using Little's Law:

$$\text{W} = L \,/\, T \qquad (3).$$

Another measure of interest is the fraction of instances in which Vehicle i must wait at Buffer i before dropping off a container at Buffer i. This equals the probability of being in a state where Vehicle i is M^+, Buffer i is full, and Vehicle i+1 is not I^-, divided by the probability that Vehicle i is M^+:

$$\text{Probability Vehicle } i \text{ Must Wait} = \text{Prob}(v_i = \text{M}^+, b_i = C_i, v_{i+1} \neq \text{I}^-) \,/\, \text{Prob}(v_i = \text{M}^+) \qquad (4).$$

7. Numerical Solution Method, Results, and Observations

We generated the state transition matrix and solved the system Ax=0, e^Tx=1 in the Microsoft Visual C++ 6.0 environment. Special data structures were used to keep track of the positions and values of the nonzero entries of A without actually storing A itself. In particular, a nontrivial solution to Ax=0 (assuming A is MxM) was found by setting x_1 equal to 1 and then solving the resulting nonsingular (M-1)x(M-1) system $A'x'=b'$ using the Gauss-Seidel method. Cooper and Gross prove that this method always converges [1]. This nontrivial solution was then renormalized so the sum of the steady state probabilities was 1. The Gauss-Seidel method was initialized with $x^0 = e$ and was continued until $|x_i^{n+1} - x_i^n|$ was less than $(\sum_i x_i^{n+1})\,(.00001)\,/\,(S_{N+1}^{C_1C_2\cdots C_N})$ for all i. Thus, the steady state probabilities for a 10-state system were revised until none of them changed more than 10^{-6} and the probabilities for a 100-state system were revised until none of them changed more than 10^{-7}. This very low tolerance guaranteed accurate results.

Results are presented in Table 2. For each system configuration, we display the number of CTMC states, the number of iterations required for Gauss-Seidel convergence, the average throughput rate, average number of items in the system, and average sojourn time. To verify that the results were correct, the average throughput rate was computed in at least two different ways for each system configuration using (1). For some systems we also show for each vehicle the fraction of instances in which the vehicle must wait at its forward buffer before dropping off a container at the buffer. Note that the term

$V^{C_1 C_2 \cdots C_{V-1}}{}_{R_1^+ R_1^- R_2^+ R_2^- \cdots R_V^+ R_V^-}$ stands for a V-vehicle system with buffer capacities C_1, C_2, ..., C_{V-1} (from origin to destination respectively) where the rates of vehicle movement are R_1^+, R_1^-, R_2^+, R_2^-, ..., R_V^+, R_V^-.

Table 2: Performance of Various Multiple Vehicle Tandem Systems

							Likelihood of Vehicles Idling				
Grp	System	#States	#Iter	T'put	#Items	Sojrn	1st	2nd	3rd	4th	5th
1	$2^{0}{}_{2,1,2,1}$	8	54	.4800	.76	1.583	.5	0			
	$2^{7}{}_{2,1,2,1}$	36	1625	.6253	4.187	6.696	.0887	0			
	$2^{77}{}_{2,1,2,1}$	316	111296	.6620	39.17	59.17	.0095	0			
	$2^{0}{}_{2,1,8,9}$	8	18	.6628	.4201	.6338	.0545	0			
	$2^{7}{}_{2,1,8,9}$	36	53	.6667	.4233	.6350	.0000	0			
	$2^{0}{}_{2,1,98,99}$	8	9	.6667	.3401	.5102	.0006	0			
2	$2^{0}{}_{1,1,1,1}$	8	54	.3636	1	2.75	.5	0			
	$3^{0,0}{}_{1,1,1,1,1,1}$	30	90	.3195	1.5	4.695	.6102	.3898	0		
	$4^{0,0,0}{}_{1,1,1,1,1,1,1,1}$	112	340	.2976	2	6.721	.6562	.5	.3438	0	
	$5^{0,0,0,0}{}_{1,1,1,1,1,1,1,1,1,1}$	418	2076	.2845	2.5	8.788	.681	.5518	.4482	.319	0
	$6^{0,0,0,0,0}{}_{1,1,1,1,1,1,1,1,1,1,1,1}$	1560	17857	.2757	3	10.88					
	$7^{0,0,0,0,0,0}{}_{1,1,1,1,1,1,1,1,1,1,1,1,1,1}$	5822	191274	.2695	3.5	12.99					
3	$2^{1}{}_{2,2,2,2}$	12	144	.8214	1.5	1.826	.3043	0			
	$3^{1,1}{}_{3,3,3,3,3,3}$	70	248	1.141	2.5	2.191	.3875	.2251	0		
	$4^{1,1,1}{}_{4,4,4,4,4,4,4,4}$	408	1335	1.459	3.5	2.398	.4263	.3009	.1951	0	
	$5^{1,1,1,1}{}_{5,5,5,5,5,5,5,5,5,5}$	2378	10575	1.777	4.500	2.532	.4488	.3402	.2644	.1793	0
4	$3^{0,0}{}_{1,2,1,2,9998,9999}$	30	64	.4800	1.240	2.583	.5000	.0000	0		
	$3^{0,0}{}_{1,2,9998,9999,1,2}$	30	146564	.5417	1.771	3.269	.3077	.6923	0		
	$3^{0,0}{}_{9998,9999,1,2,1,2}$	30	194	.4800	2.240	4.667	1	.5	0		
5	$4^{0,0,0}{}_{3,3,3,3,3,3,3,3}$	112	340	.8927	2	2.240	.6562	.5	.3438	0	
	$4^{0,0,0}{}_{2,6,2,6,2,6,2,6}$	112	142	.8566	2.596	3.030	.6836	.5298	.3650	0	
	$4^{0,0,0}{}_{6,2,6,2,6,2,6,2}$	112	1764	.8566	1.404	1.639	.6351	.4702	.3164	0	
	$4^{0,0,0}{}_{2,6,2,6,6,2,6,2}$	112	292	.8562	2	2.336	.6723	.5	.3277	0	
	$4^{0,0,0}{}_{6,2,6,2,2,6,2,6}$	112	596	.8562	2	2.336	.6432	.5000	.3568	0	
6	$5^{1,0,0,0}{}_{1,1,1,1,1,1,1,1,1,1}$	642	1730	.2950	3.483	11.81	.6173	.6341	.4895	.3389	0
	$5^{0,1,0,0}{}_{1,1,1,1,1,1,1,1,1,1}$	658	2882	.3034	3.206	10.57	.6288	.3953	.5382	.3589	0
	$5^{0,0,1,0}{}_{1,1,1,1,1,1,1,1,1,1}$	658	5070	.3034	2.794	9.210	.6411	.4618	.2070	.3712	0
	$5^{0,0,0,1}{}_{1,1,1,1,1,1,1,1,1,1}$	642	2391	.2950	2.517	8.534	.6611	.5105	.3659	.0746	0
	$5^{1,1,1,0}{}_{1,1,1,1,1,1,1,1,1,1}$	1562	4897	.3352	4.682	13.97	.4971	.4195	.3899	.4365	0
	$5^{1,1,0,1}{}_{1,1,1,1,1,1,1,1,1,1}$	1538	4107	.3314	4.242	12.80	.5117	.4504	.4851	.1235	0
7	$4^{0,1,1}{}_{1,1,1,1,1,1,1,1}$	268	1170	.3395	2.492	7.342	.5553	.1945	.1502	0	
	$4^{0,2,0}{}_{1,1,1,1,1,1,1,1}$	240	1269	.3341	3	8.978	.5630	.2066	.4370	0	
	$4^{4,4,4}{}_{1,1,1,1,1,1,1,1}$	3408	8025	.4324	8.000	18.50	.2017	.1341	.0815	0	
	$4^{4,5,3}{}_{1,1,1,1,1,1,1,1}$	3316	6878	.4315	8.518	19.74	.2037	.1343	.1281	0	
	$4^{3,6,3}{}_{1,1,1,1,1,1,1,1}$	3160	9472	.4301	8.000	18.60	.2124	.0977	.1267	0	
8	$3^{0,5}{}_{1,6,3,5,4,2}$	110	242	.7820	1.713	2.191	.2809	.0022	0		
	$3^{1,4}{}_{1,6,3,5,4,2}$	142	536	.8343	2.017	2.418	.0828	.0153	0		
	$3^{2,3}{}_{1,6,3,5,4,2}$	158	1900	.8482	2.127	2.508	.0294	.0450	0		
	$3^{3,2}{}_{1,6,3,5,4,2}$	158	4599	.8516	2.169	2.547	.0156	.1021	0		
	$3^{4,1}{}_{1,6,3,5,4,2}$	142	7857	.8520	2.224	2.611	.0134	.2176	0		
	$3^{5,0}{}_{1,6,3,5,4,2}$	110	4684	.8484	2.510	2.958	.0212	.4748	0		

The results are intuitively understandable for the most part. Results for the first group of systems demonstrate that adding extra buffer space or increasing the speed of a vehicle increases overall throughput. However, these two means of improving system performance are only partially substitutable. In particular, faster vehicles will reduce the average sojourn time but larger buffers will increase the average sojourn time. The latter scenario appears to be less desirable. Results from group 2 indicate that throughput deteriorates as more buffers and vehicles of the same type are added to a system. These experiments also demonstrate nicely that the tandem system in this paper has a slightly reduced throughput compared to the system with type-2 Erlang service distribution in [6] which most closely approximates this system. Results from group 3 indicate that throughput improves almost linearly as more components are added to a system given a constant expected pure sojourn time from the origin to the destination. Group 4 shows that a single fast vehicle should be placed at the center of a system to maximize throughput. Group 5 shows that variation in a vehicle's forward and reverse rates can affect system throughput even though the expected pure, round-trip travel time for each vehicle is constant. Group 6 indicates that the capacities of the most central buffers in an identical-vehicle system should be increased before those of the outlying buffers in order to maximize throughput. However, group 7 shows that it is harmful to concentrate too much buffer capacity in the central buffers in such a system. Group 8 shows how five units of buffer capacity should be allocated among two buffers to maximize throughput.

8. Conclusions and Future Research

A new tandem system was analyzed and new numerical results regarding the performance of the system were presented. Future work will focus on other tandem systems that have vehicles moving back and forth between buffers and on various buffer allocation problems related to this system. Finally, we hope to apply these results to the study of industrial systems in a real setting. In particular, this work might help to evaluate the performance of double-trolley quay cranes at marine container ports. In these cranes, the job of transferring a container from vessel to truck is divided among two different trolleys that are separated by an intermediate platform. The intermediate platform has space to accommodate several containers. Double trolley quay cranes are a recent innovation in container terminal infrastructure and have the potential to dramatically speed up the loading and unloading of containerships.

References

1. R.B. Cooper and D. Gross , On the convergence of Jacobi and Gauss-Seidel iteration for steady-state probabilities of finite-state continuous-time Markov chains. *Stochastic Models* **7** (1991), pp. 185-189.
2. M.K. Govil and M.C. Fu , Queueing theory in manufacturing: A survey. *Journal of Manufacturing Systems* **18** (1999), pp. 214-240.
3. C. Heavey, H.T. Papadopoulos and J. Browne , The throughput rate of multistation unreliable production lines. *European Journal of Operational Research* **68** (1993), pp. 69–89.
4. H.T. Papadopoulos and C. Heavey , Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *European Journal of Operational Research* **92** (1996), pp. 1–27.
5. H.T. Papadopoulos,C. Heavey and M.E.J. O'Kelly,Throughput rate of multistation reliable production lines with inter station buffers: (I) Exponential case. *Computers in Industry* **13** (1989), pp. 229–244.
6. H.T. Papadopoulos,C. Heavey and M.E.J. O'Kelly,Throughput rate of multistation reliable production lines with inter station buffers: (II) Erlang case. *Computers in Industry* **13** (1990), pp. 317–335.
7. H.T. Papadopoulos and M.E.J. O'Kelly , A recursive algorithm for generating the transition matrices of multistation series production lines. *Computers in Industry* **12** (1989), pp. 227–240.
8. J.A. Ventura and C. Lee , Optimally locating multiple dwell points in a single loop guide path system. *IIE Transactions* **35** (2003), pp. 727-737.
9. M.I. Vidalis and H.T.Papadopoulos , Markovian analysis of production lines with Coxian-2 service times. *International Transactions in Operational Research* **6** (1999), pp. 495–524.