

**COMPARING SYSTEM TIMES IN AN UNORDERED
TWO QUEUE NETWORK**

S. MOLINARO, M. HLYNKA,
DEPARTMENT OF MATHEMATICS AND STATISTICS
UNIVERSITY OF WINDSOR

Abstract:

A special customer must access two servers, each with its own $M/M/1$ queue. The customer can choose which queue to join first. No jockeying is permitted. Depending on which server is selected first, there will be two expected total system times. We compute the two expected system times and note the conditions for which one ordering gives a lower expected time than the other.

1. INTRODUCTION

Assume that there are two independent $M/M/1$ queueing systems. A special customer SP arrives and sees n customers in queue 1 and m customers in queue 2. The special customer is special in that it must receive service from both of the servers, and no other customer has the same requirement. The special customer has two choices:

- (a) Join queue 1 and then proceed to join queue 2.
- (b) Join queue 2 and then proceed to join queue 1.

We are interested in the total system time for the two different orderings.

Some related literature follows. He and Neuts (2002) consider transfer of customers in two $M/M/1$ queues. These customers seek service at only one of the queues. Parlakturk and Kumar (2004) analyzed the problem of self-interested routing in queueing networks. Their problem involves having each customer choose its route to minimize expected delay (there is more than one special customer). The system manager choose the scheduling rule. Winston (1977) proved that under certain conditions, a customer should join the shorter of two lines in a parallel queueing system. This was extended to a more general conditions by Weber (1978). However Whitt (1986) gave counterexamples where join-the-shortest-line does not hold.

Notation:

The following notation will be used throughout the report

- m is the number of customers in line at time 0 for queue 1.

- n is the number of customers in line at time 0 for queue 2.
- i is the number of customers in the next queue after completing one queue.
- $E(TST_1)$ is the expected total system time for SP to complete both queues if queue 1 is joined first.
- $E(TST_2)$ is the expected total system time for SP to complete both queues if queue 2 is joined first.
- ρ is the ratio $\frac{\lambda}{\mu}$.
- L is the expected number of customers in an $M/M/1$ queue; $L = \frac{\rho}{1-\rho}$.
- W is the expected time spent in an $M/M/1$ queue; $W = \frac{1}{\mu-\lambda}$ for a randomly arriving customer.

We first calculate transient probabilities of two $M/M/1$ queues $p_{r,i}^{(j)}(x)$ of having i customers in the queueing system j ($j = 1, 2$) at time x , beginning with r customers at time 0. The calculations are based on time 0 counts of m customers in line 1, n customers in line 2, the arrival rates λ_1 for system 1, λ_2 for system 2, and the service rates μ_1 for queue 1, μ_2 for queue 2. Transient probabilities are used to compute the expected total system times $E(TST_1)$ and $E(TST_2)$ for the two orders. The first, $E(TST_1)$, is calculated if the customer enters queue 1 first then proceeds to queue 2. The second, $E(TST_2)$, assumes queue 2 is completed first. We discover which queue should be joined first, based on different conditions. We will determine at what point the second option becomes more desirable based on arrival rates λ_1 , λ_2 ,

service rates μ_1, μ_2 , and the number of customers seen initially n, m . Any patterns of interesting results are noted.

For example, special customer SP arrives at an amusement park and wants to have two photographs, one with Superman and one with an alien, both of whom have separate lines for photographs. As SP is related to the owner of the amusement park, SP receives a pass for two photographs. Everyone else is given a pass with exactly one of the two celebrities specified on the pass. Type i people receive a pass for celebrity i , for $i = 1, 2$. The special customer must choose which line to enter first and makes a decision solely on minimizing total system time. Assuming all information about the queues are known, we can calculate the expected system times of both options (choosing photo 1 first or choosing photo 2 first).

In section 2, we present some preliminary analysis. In section 3, we present some of the numerical results. In section 4, we make some observations and conclusions. In section 4, we find a somewhat surprising result, that there exist situations where joining the shortest queue is not the best choice (when we must complete the queues in tandem, but have a choice of order). We observe patterns in joining preferences and make conclusions on which queue to join first.

2. PRELIMINARY ANALYSIS

The customer SP arrives at the system and observes m customers in the queueing system 1 (either $m = 0$ or if $m \geq 1$, one customer is in service and $m - 1$ customers are waiting). Similarly there are

n customers in the queueing system 2. After the arrival, we assume that the systems operate as independent $M/M/1$ queueing systems. In order to evaluate the system times for SP depending on the order chosen, we note that the second queueing system is not in steady state when the SP completes service from one of the queueing systems. In order to calculate the two expected total system times through both queues, for the two initial choices of which queue to join first, we must calculate the transient probabilities of the number of customers SP will observe in the second queue, when SP completes the first queue. Let $p_{m,i}^{(1)}(x)$ be the probability that queueing system 1 will have i customers at time x , given that queueing system 1 had m customers at time 0. Let $p_{n,i}^{(2)}(x)$ be the probability that queueing system 2 will have i customers at time x given that queueing system 2 had n customers at time time 0. We use the formula developed by Conolly and Langaris [6], for the transient probability $p_i(t)$ that there are i customers at time t given that there were n customers at time 0.

$$\begin{aligned}
p_i(t) = & (1 - \rho)\rho^i + e^{-(\lambda+\mu)t}\rho^i \sum_{k=0}^{\infty} \left(\frac{(\lambda t)^k}{k!} \sum_{r=0}^{k+i+n+1} (k-r) \frac{(\mu t)^{r-1}}{r!} \right) \\
& + e^{-(\lambda+\mu)t}\rho^i \sum_{k=0}^{\infty} \frac{(\lambda t)^{k+1}(\mu t)^{k+\max(i,n)}}{k!} \left(\frac{(\lambda t)^{-\min(i,n)-1}}{(k+|n-i|)!} - \frac{(\mu t)^{\min(i,n)+1}}{(k+n+i+2)!} \right).
\end{aligned} \tag{2.1}$$

This formula was checked in Hlynka and Molinaro (2009). The formula is used for both queue 1 and queue 2 (with the appropriate variable substitutions). Maple 12 was coded to perform the calculations.

Using the Erlang probability density function $f(x)$ we get an expression for the expected total system time TST_1 if queueing system 1 is joined first:

$$E(TST_1) = \int_0^\infty \left(x + \sum_{i=0}^{\infty} p_{m,i}^{(2)}(x) \frac{i+1}{\mu_2} \right) f(x) dx,$$

where $f(x) = \frac{x^m \mu_1^{m+1} e^{-\mu_1 x}}{\Gamma(m+1)}$ $x > 0$.

The expected total system time TST_2 if queueing system 2 is joined first is:

$$E(TST_2) = \int_0^\infty \left(x + \sum_{i=0}^{\infty} p_{n,i}^{(1)}(x) \frac{i+1}{\mu_1} \right) g(x) dx,$$

where $g(x) = \frac{x^n \mu_2^{n+1} e^{-\mu_2 x}}{\Gamma(n+1)}$ $x > 0$.

By substituting formula 2.1 into $E(TST_1)$ and $E(TST_2)$ we get the following two formulas:

$$\begin{aligned} E(TST_1) = & \int_0^\infty \left(x + \sum_{i=0}^{\infty} \left((1 - \rho_2) \rho_2^i + e^{-(\lambda_2 + \mu_2)t} \rho_2^i \sum_{k=0}^{\infty} \left(\frac{(\lambda_2 t)^k}{k!} \sum_{r=0}^{k+i+n+1} (k-r) \frac{(\mu_2 t)^{r-1}}{r!} \right) \right. \right. \\ & \left. \left. + e^{-(\lambda_2 + \mu_2)t} \rho_2^i \sum_{k=0}^{\infty} \frac{(\lambda_2 t)^{k+1} (\mu_2 t)^{k+max(i,n)}}{k!} \left(\frac{(\lambda_2 t)^{-min(i,n)-1}}{(k+|n-i|)!} - \frac{(\mu_2 t)^{min(i,n)+1}}{(k+n+i+2)!} \right) \right) \right. \\ & \left. \frac{i+1}{\mu_2} \right) \frac{x^m \mu_1^{m+1} e^{-\mu_1 x}}{\Gamma(m+1)} dx, \end{aligned}$$

$$\begin{aligned} E(TSS_2) = & \int_0^\infty \left(x + \sum_{i=0}^{\infty} \left((1 - \rho_1) \rho_1^i + e^{-(\lambda_1 + \mu_1)t} \rho_1^i \sum_{k=0}^{\infty} \left(\frac{(\lambda_1 t)^k}{k!} \sum_{r=0}^{k+i+m+1} (k-r) \frac{(\mu_1 t)^{r-1}}{r!} \right) \right. \right. \\ & \left. \left. + e^{-(\lambda_1 + \mu_1)t} \rho_1^i \sum_{k=0}^{\infty} \frac{(\lambda_1 t)^{k+1} (\mu_1 t)^{k+max(i,m)}}{k!} \left(\frac{(\lambda_1 t)^{-min(i,m)-1}}{(k+|m-i|)!} - \frac{(\mu_1 t)^{min(i,m)+1}}{(k+m+i+2)!} \right) \right) \right. \\ & \left. \frac{i+1}{\mu_1} \right) \frac{x^n \mu_2^{n+1} e^{-\mu_2 x}}{\Gamma(n+1)} dx. \end{aligned}$$

Using these formulas we calculate expected total system times for both cases (choosing queue 1 first and choosing queue 2 first). By comparing the total system times we can see when it is optimal to choose queue 2 first or queue 1 first. These formulas were coded into Maple 12 for different values of m , n , λ_1 , λ_2 , μ_1 and μ_2 .

3. NUMERICAL RESULTS

The following tables were calculated using Maple 12 and are set up as follows:

- (1) Each table represents the arrival and service rates of queue 1 (λ_1 and μ_1).
- (2) Each column represents the arrival and service rates of queue 2 (λ_2 and μ_2).
- (3) The rows represent different values of n and m the number of people observed in queue 1 and queue 2 respectively.
- (4) The columns are split into $E(TST_1)$ (denoted $ETST1$ in the tables), the expected total system time when entering queue 1 first, and $E(TST_2)$ (denoted $ETST2$ in the tables), the expected total system time when entering queue 2 first .

By calculating values for fixed μ we change λ in order to see its relationship. We also vary values of μ (holding and not holding ρ constant) in order to see how μ affects choosing queue 1 or queue 2. Taking into

consideration the lengths of the queues we observe relationships involving n and m .

Note: Finite summations are used to approximate infinite summations. Everywhere, assume $\rho < 1$, although this is not needed because we are dealing with a finite time situation.

Note that when customer SP chooses queue 2 first, the variable λ_2 is not used in the calculations. If SP enters queue 2, it does not matter who enters that queue 2 later as SP stays in queue 2 until its service is completed. Looking at cases where μ is constant, note that $E(TST_2)$ does not change when varying λ_2 .

Table 1: $\lambda_1 = 0.1, \mu_1 = 1, \rho_1 = 0.1, L_1 = 0.11$

m & n		$\lambda_2 = 0.05, \mu_2 = 0.5$ $\rho_2 = 0.1, L_2 = 0.11$		$\lambda_2 = 0.1, \mu_2 = 1$ $\rho_2 = 0.1, L_2 = 0.11$		$\lambda_2 = 0.75, \mu_2 = 1$ $\rho_2 = 0.75, L_2 = 3$		$\lambda_2 = 0.9, \mu_2 = 1$ $\rho_2 = 0.9, L_2 = 9$		$\lambda_2 = 0.2, \mu_2 = 2$ $\rho_2 = 0.1, L_2 = 0.11$	
		<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>
$m = 0$	$n = 0$	3.06740	3.06969	2.05125	2.05125	2.44300	2.05125	2.54659	2.05125	1.53485	1.53370
$m = 1$	$n = 5$	12.21878	13.11382	6.31712	7.13276	7.53292	7.13276	7.82433	7.13276	3.60360	4.20575
	$n = 1$	5.03160	5.22187	3.34720	3.34720	4.14131	3.34720	4.36030	3.34720	2.61094	2.51580
	$n = 0$	4.11331	3.41816	3.07815	2.56374	3.74227	2.56374	3.93203	2.56374	2.54753	2.20769
$m = 5$	$n = 20$	42.60000	43.11112	21.60179	22.11479	25.50003	22.11479	26.40001	22.11479	11.23456	11.76398
	$n = 5$	13.02706	13.33226	8.14330	8.14330	10.86656	8.14330	11.65150	8.14330	6.66613	6.51353
	$n = 1$	8.41151	7.20719	7.13276	6.31712	8.57349	6.31712	9.09560	6.31712	6.55691	6.10939
$m = 10$	$n = 20$	43.10021	43.11173	22.16845	22.22893	29.25107	22.22893	30.90035	22.22893	13.10424	13.40789
	$n = 10$	23.28838	23.29072	13.89720	13.89720	19.42352	13.89720	20.98959	13.89720	11.64536	11.64419
	$n = 5$	15.36442	3314.86379	12.29168	11.83095	15.43958	11.83095	16.67111	11.83095	11.55983	11.30552
	$n = 1$	13.25926	11.53860	12.11266	11.20503	13.92870	11.20503	14.75392	11.20503	11.55558	11.10006
$m = 20$	$n = 20$	44.13886	43.19832	25.16547	25.16547	36.78996	25.16547	39.91089	25.16547	21.59916	22.06943
	$n = 5$	23.52796	22.46912	22.11479	21.60179	25.12298	21.60179	26.94812	21.60179	21.55556	21.30000
	$n = 1$	23.22430	21.40312	22.11113	21.20001	24.32978	21.20001	25.65530	21.20001	21.55556	21.10000
$m = 30$	$n = 30$	65.10838	63.15044	36.30247	36.30247	54.25953	36.30247	58.90096	36.30247	31.57522	32.55419

Table 2: $\lambda_1 = 0.5, \mu_1 = 5, \rho_1 = 0.1, L_1 = 0.11$

m & n		$\lambda_2 = 0.3, \mu_2 = 3$ $\rho_2 = 0.1, L_2 = 0.11$		$\lambda_2 = 0.5, \mu_2 = 5$ $\rho_2 = 0.1, L_2 = 0.11$		$\lambda_2 = 3.75, \mu_2 = 5$ $\rho_2 = 0.75, L_2 = 3$		$\lambda_2 = 4.5, \mu_2 = 5$ $\rho_2 = 0.9, L_2 = 9$		$\lambda_2 = 1, \mu_2 = 10$ $\rho_2 = 0.1, L_2 = 0.11$	
		<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>
$m = 0$	$n = 0$	0.54601	0.54633	0.41025	0.41025	0.48860	0.41025	0.50932	0.41025	0.30697	0.30674
$m = 1$	$n = 5$	2.04644	2.22322	1.26342	1.42655	1.50658	1.42655	1.56487	1.42655	0.72072	0.84115
	$n = 1$	0.88983	0.91650	0.66944	0.66944	0.82826	0.66944	0.87206	0.66944	0.52219	0.50316
	$n = 0$	0.75415	0.62430	0.61563	0.51275	0.74845	0.51275	0.78641	0.51275	0.50951	0.44154
$m = 5$	$n = 20$	7.12000	7.22223	4.32036	4.42296	5.10001	4.42296	5.28000	4.42296	2.24691	2.35280
	$n = 5$	2.24793	2.29380	1.62866	1.62866	2.17331	1.62866	2.33030	1.62866	1.33323	1.30271
	$n = 1$	1.59296	1.36925	1.42655	1.26342	1.71470	1.26342	1.81912	1.26342	1.31138	1.22188
$m = 10$	$n = 20$	7.22025	7.22287	4.43369	4.44579	5.85021	4.44579	6.18007	4.44579	2.62085	2.68158
	$n = 10$	3.96520	3.96515	2.77944	2.77944	3.88470	2.77944	4.19792	2.77944	2.32907	2.32884
	$n = 5$	2.81904	2.71266	2.45834	2.36619	3.08792	2.36619	3.33422	2.36619	2.31197	2.26110
	$n = 1$	2.57390	2.27959	2.42253	2.24101	2.78574	2.24101	2.95078	2.24101	2.31112	2.22001
$m = 20$	$n = 20$	7.44993	7.28227	5.03309	5.03309	7.35799	5.03309	7.98218	5.03309	4.31983	4.41389
	$n = 5$	4.59364	4.41820	4.42296	4.32036	5.02460	4.32036	5.38962	4.32036	4.31111	4.26000
	$n = 1$	4.57050	4.26682	4.42223	4.24000	4.86596	4.24000	5.13106	4.24000	4.31111	4.22000
$m = 30$	$n = 30$	10.96499	10.59808	7.26049	7.26049	10.85191	7.26049	11.78019	7.26049	6.31504	6.51084

Table 3: $\lambda_1 = 3.75, \mu_1 = 5, \rho_1 = 0.75, L_1 = 3$

m & n		$\lambda_2 = 2.25, \mu_2 = 3$ $\rho_2 = 0.75, L_2 = 3$		$\lambda_2 = 0.5, \mu_2 = 5$ $\rho_2 = 0.1, L_2 = 0.11$		$\lambda_2 = 3.75, \mu_2 = 5$ $\rho_2 = 0.75, L_2 = 3$		$\lambda_2 = 4.5, \mu_2 = 5$ $\rho_2 = 0.9, L_2 = 9$		$\lambda_2 = 7.5, \mu_2 = 10$ $\rho_2 = 0.75, L_2 = 3$	
		<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>	<i>ETST1</i>	<i>ETST2</i>
$m = 0$	$n = 0$	0.63618	0.65637	0.41025	0.48860	0.48860	0.48860	0.50932	0.48860	0.36861	0.35380
$m = 1$	$n = 5$	2.30218	2.57068	1.26342	1.71470	1.50658	1.71470	1.56487	1.71470	0.91953	1.05365
	$n = 1$	1.08157	1.12484	0.66944	0.82826	0.82826	0.82826	0.87206	0.82826	0.63575	0.60434
	$n = 0$	0.91233	0.75480	0.61563	0.60673	0.74845	0.60673	0.78641	0.60673	0.60889	0.49727
$m = 5$	$n = 20$	7.90000	7.80116	4.32036	5.02460	5.10001	5.02460	5.28000	5.02460	3.00048	2.98936
	$n = 5$	2.93431	2.90631	1.62866	2.17331	2.17331	2.17331	2.33030	2.17331	1.64250	1.66223
	$n = 1$	1.97945	1.72549	1.42655	1.50658	1.71470	1.50658	1.81912	1.50658	1.49598	1.35069
$m = 10$	$n = 20$	8.65000	8.12068	4.43369	5.56499	5.85021	5.56499	6.18007	5.56499	3.75728	3.80047
	$n = 10$	5.32456	5.11051	2.77944	3.88470	3.88470	3.88470	4.19792	3.88470	2.87675	3.02704
	$n = 5$	3.79137	3.73102	2.45834	3.74532	3.08792	3.10457	3.33422	3.10457	2.61251	2.65015
	$n = 1$	3.07882	2.70131	2.42253	2.50012	2.78574	2.50012	2.95078	2.50012	2.53410	2.35000
$m = 20$	$n = 20$	10.15046	9.53493	5.03309	7.35799	7.35799	7.35799	7.98218	7.35799	5.31753	5.77506
	$n = 5$	4.59316	5.10001	4.42296	5.10001	7.26049	5.10001	5.38962	5.10001	4.59603	4.65000
	$n = 1$	4.56206	4.50000	4.42223	4.50000	4.86596	4.50000	5.13106	4.50000	4.56742	4.35000
$m = 30$	$n = 30$	9.02770	10.85191	7.26049	10.85191	10.85191	10.85191	11.78019	10.85191	7.74228	8.52500

4. OBSERVATIONS

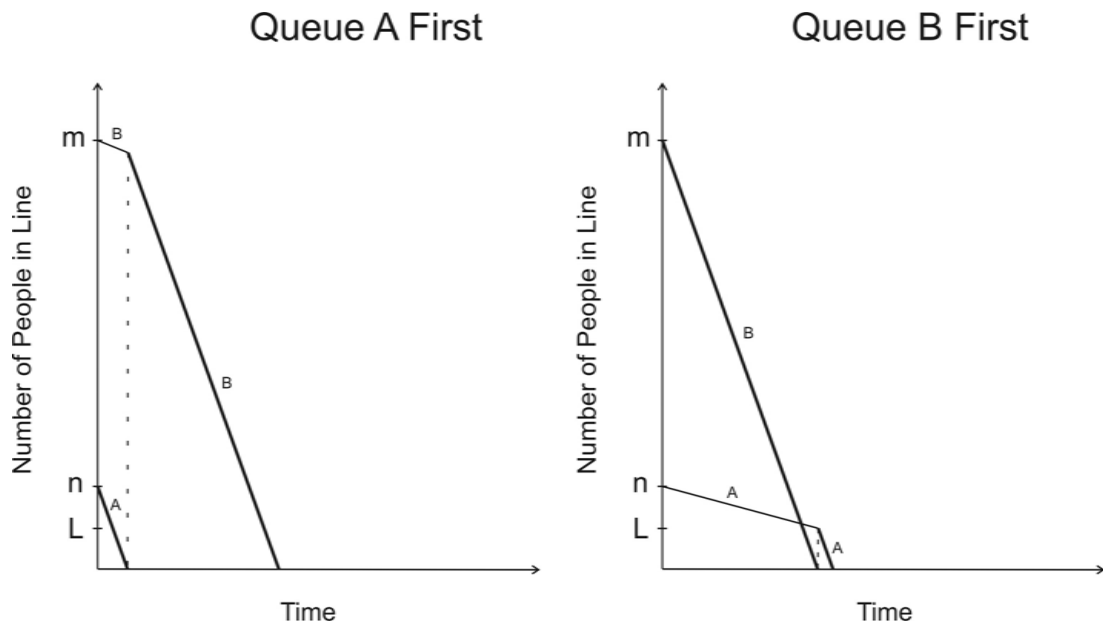
CASE 1: Both queues have the same rates.

Suppose that the two queueing systems have parameters $\lambda_1 = \lambda_2$, $\mu_1 = \mu_2$ and $\rho_1 = \rho_2$.

This case yields one of the most surprising properties. The initial counts for line 1 and line 2 are m and n respectively. Intuitively, it is reasonable to believe that an arriving special customer should choose the shorter of the two systems. Certainly if $n > L > m$, one would expect that by choosing the shorter line first, the longer line would be likely decrease toward the mean length L by the time the short service is done. This is good. If the longer line is chosen first, then the short line would likely increase toward L by the time that the long line is done. This is bad. Even if L is not between m and n , one would expect the same pattern. However, we will show that it is not always better to choose the shorter line first.

When the two lines have identical parameters $\lambda_1 = \lambda_2$, $\mu_1 = \mu_2$, and $n > L > m$ or $L > n > m$, the special customer SP should always choose line 1. When the two queues have the same number of customers $m = n$, SP is indifferent between the choice of the first queue (it is irrelevant which one is entered first). What happens when $n > m > L$ depends on the values of λ and μ . Normally the best strategy is to join the shorter queue. However, as m gets large (and $n > m \gg L$) we have a switch in the form of the best strategy for

SP, and it is better to join the longer queue. If we look at the case where $\lambda_1 = 0.1 = \lambda_2$ and $\mu_1 = 1 = \mu_2$ (so $\lambda \ll \mu$), the value of m must be at least 10 before this switch occurs. When $m = 20$, SP switches to the longer line first when $m \approx 27$. This switch occurs much earlier when λ is closer to μ . Looking at $\lambda_1 = 3.75 = \lambda_2$, $\mu_1 = 5 = \mu_2$, and $\rho = .75$, SP should choose the longer line when m is as small as 5, and $n > 8$, which is much earlier than the case where $\lambda \ll \mu$. When $\lambda_1 = 0.75 = \lambda_2$ and $\mu_1 = 1 = \mu_2$ the value of ρ is still 0.75, and we still first choose the longer line when $m = 5$. This indicates that ρ is important in determining if and when the best choice is the longer line.



This interesting phenomenon that it can be better to choose the longer line first, even for queues with the same parameters, is illustrated by the following an example:

Suppose we have queue 1 and queue 2 (identical parameter queues) such that $\lambda_A = 0.9 = \lambda_B$, $\mu_A = 1 = \mu_B$, $\rho_A = 0.9 = \rho_B$, $L_A = 9 = L_B$, $m = 15$ (initial queue 1 length) and $n = 100$ (initial queue 2 length). Note that ρ is fairly close to 1 and n is much larger than m . One would think SP should join the shortest queue (since they are identical) and hope that the longer queue decreases by the time SP finishes the short queue. SP is actually better off joining the longer queue first. See Figure 1. The left portion of the figure represents joining queue 1 first, and the right portion joining queue 2 first. The horizontal axis represents time. Assume that $n \gg m > L$. Let r represent the number of people in queue 2 when we finish queue 1, if queue 1 is joined first. The following condition is intuitively reasonable, although it may not be completely precise. SP should choose queue 2 first if

$$n - r < m - L.$$

In our example we have: $100 - r < 15 - 9 = 6$. As long as queue 2 drops less than 6 people (when SP completes queue 1 first) SP saves time by joining queue 2 first. The reason the queue would drop by so few people is because the time to complete queue 1 is relatively short (queue 2 is not in steady state) and our value of ρ is close to 1 (causing the queue to drop slowly). We assume that by choosing queue 2 first,

queue 1 will get close to L (since the line in queue 2 is long enough to push allow queue 1 to get close to steady state).

CASE 2: One system's parameters are double those of the other ($2\lambda_1 = \lambda_2$, $2\mu_1 = \mu_2$ and $\rho_1 = \rho_2$).

If there are two empty queues, and $2\mu_1 = \mu_2$, $2\lambda_1 = \lambda_2$, SP should choose the higher parameter queue. This makes sense because the lower parameter queue is less likely to have customers by the time SP finishes the faster queue.

Next we look at two queues with the same number of customers in line ($m = n > 1$). The speed of the second queue is exactly double that of the first. From our calculations, ; we notice that SP joining the higher parameter queue first is more efficient than joining the lower parameter queue first if the number of customers in line is small (e.g. 1 person in each line). As the number of people in the lines increases (e.g. 10 customers in each line) SP lowers its expected total system time by joining the lower parameter queue first. An example of this pattern can be found by looking at the results for the following parameters: $\lambda_1 = 0.1, \mu_1 = 1\rho_1 = 0.1$ and $\lambda_2 = 0.2, \mu_2 = 2, \rho_2 = 0.1$.

This interesting pattern is related to the value of L in comparison to m and n . If the values of m and n are below L , choosing the lower parameter queue first will allow the higher parameter queue to move up to L . On the other hand, if we choose the higher parameter queue first, then the lower parameter queue does not have as much time to move upward to L . This means choosing queue 2 (higher parameter queue) first will save the customer SP some time. When the values of

m and n are above L , the opposite holds. In this case the queues are moving down toward L (instead of up) indicating that the more time the queue has to reach L , the better. In this case SP should choose the lower parameter queue first.

When the lines are not the same length we notice a different pattern. If we look at cases where $\lambda_i \ll \mu_i$, $i = 1, 2$, SP should generally choose the shorter line when it is considerably shorter. As the lines get longer there comes a point in which SP should choose the longer of the two lines. As λ_i approaches μ_i the switch happens sooner (i.e. we choose the longer line sooner). Look at the following three cases:

- (1) **Case 1:** $\lambda_1 = 0.1, \mu_1 = 1; \lambda_2 = 0.2, \mu_2 = 2$.
- (2) **Case 2:** $\lambda_1 = 0.5, \mu_1 = 5; \lambda_2 = 1, \mu_2 = 10$.
- (3) **Case 3:** $\lambda_1 = 3.75, \mu_1 = 5; \lambda_2 = 7.5, \mu_2 = 10$.

If we consider the situation where queue 1 has 20 people in line and vary the number of people in queue 2, we get the following:

- (1) **Case 1:** Choose the shorter (and faster) queue, up until there are 11 people in queue 2.
- (2) **Case 2:** Choose the shorter (and faster) queue, up until there are 11 people in queue 2.
- (3) **Case 3:** Choose the shorter (and faster) queue, up until there are 5 people in queue 2.

If we start with m smaller than 20, we make the switch sooner. Notice that when ρ is constant we still switch at the same point. As λ gets closer to μ we switch to the longer line sooner.

This can be explained by the same analysis as in the case where the two queues are identical (a more extreme example).

CASE 3: μ is constant and λ_2 is larger than λ_1
($\lambda_1 < \lambda_2$ and $\mu_1 = \mu_2$).

When looking at two empty queues it is obvious that SP should choose the queue with the largest arrival rate first (i.e. choose system 2 first). This is indicated in the sample calculations. A natural question to ask is “Is choosing the queue with the smaller arrival rate first *ever* more efficient?”

When $\lambda_1 \ll \lambda_2$ (and initial systems lengths are $m = n = 0$), customer SP has a smaller expected total system time if it joins queue 2 first. The expected total system time remains shortest by joining queue 2 first when the number of customers in queue 2 is: the same, much larger ($m = 1$ vs. $n = 30$) or much smaller ($m = 30$ vs. $n = 1$) than the number of customers in queue 1. An example of this situation occurs when $\lambda_1 = 0.1, \lambda_2 = 0.75$ and $\mu_1 = 1 = \mu_2$. Even more drastically, we see this when $\lambda_1 = 0.1, \lambda_2 = 0.9$ and $\mu_1 = 1 = \mu_2$. The difference between the two expected total system times (as a function of n) begins at a particular value for $n = 1$, and then decreases. After a certain value of n , the difference in the total system times begins to increase (in the $\lambda_2 = 0.75$ case, that point is near $n = 12$ customers in queue 2).

Since the two arrival rates are so far apart $E(TST1)$ is never less than $E(TST2)$.

The following table illustrates this observation:

$\lambda_1 = 0.1, \mu_1 = 1$				
$\rho_1 = 0.1, L_1 = 0.11$				
m & n		$\lambda_2 = 0.75, \mu_2 = 1$		$ETST1-ETST2$
		$\rho_2 = 0.75, L_2 = 3$		
		$ETST1)$	$ETST2$	
$m = 1$	$n = 0$	3.74227	2.56374	1.17853
$m = 1$	$n = 1$	4.14131	3.34720	0.79411
$m = 1$	$n = 5$	7.53292	7.13276	0.40016
$m = 1$	$n = 10$	12.50059	12.11266	0.38793
$m = 1$	$n = 15$	17.50001	17.11125	0.38876
$m = 1$	$n = 20$	22.50000	22.11113	0.38888

We observe an interesting pattern when λ_1 and λ_2 are close together. Consider the case with $\lambda_1 = 3.75, \lambda_2 = 4.5$ and $\mu_1 = 5 = \mu_2$. Fix the number of customers people in queue 1 to be small ($m < 5$). Begin with the number of customers in queue 2 small ($n = 2$) and increase n . There comes a point (by increasing the number of people in queue 2) where the expected total system time is smaller when the customer enters queue 1 first (the queue with the smaller arrival rate). As the

number of people in queue 1 increases it remains more efficient to join queue 2 first longer (the queue with the larger arrival rate). After the number of people in queue 1 reaches a certain point it no longer ever becomes more efficient to join queue 1 first. This observation is similar because of the initial point where the total system times approach each other. The following table illustrates this observation:

$\lambda_1 = 3.75, \mu_1 = 5$				
$\rho_1 = 0.75, L_1 = 3$				
m & n		$\lambda_2 = 4.5, \mu_2 = 5$		$ETST1-ETST2$
		$\rho_2 = 0.9, L_2 = 9$		
		$ETST1$	$ETST2$	
$m = 1$	$n = 0$	0.78641	0.60673	0.17967
$m = 1$	$n = 1$	0.87206	0.82826	0.04380
$m = 1$	$n = 2$	1.01310	1.05220	-0.03911
$m = 1$	$n = 5$	1.56487	1.71470	-0.14983
$m = 1$	$n = 10$	2.56007	2.78574	-0.22567
$m = 1$	$n = 20$	4.56000	4.86596	-0.30596
$m = 1$	$n = 30$	6.56000	6.90992	-0.34992
$m = 5$	$n = 1$	1.81912	1.50658	0.31254
$m = 5$	$n = 5$	2.33030	2.17331	0.15699
$m = 5$	$n = 10$	3.28253	3.08792	0.19461
$m = 5$	$n = 20$	5.28000	5.02460	0.25541
$m = 5$	$n = 30$	7.28000	7.00527	0.27473

We can explain this observation by again suggesting its relationship with L_1 and L_2 . If we look at the example where $\lambda_1 = 0.75$, $\lambda_2 = 0.9$, $\mu_1 = 1 = \mu_2$, $m = 1$ and $n = 10$, we notice that $m < L_1$ and $n > L_2$. If SP chooses queue 1 first, queue 2 will tend to drop towards L_2 , after SP finishes queue 1. If SP chooses queue 2 first, then queue 1 will increase towards L_1 . In this situation it is better to choose queue 1 first. One can see that this is not so obvious when m starts to increase and will likely stop happening when $m > L$ (which is illustrated in our calculations). The transfer of preference to the queue with the lower arrival rate is not obvious when λ_1 is small (seen in the example where $\lambda_1 = 0.1$ and $\lambda_2 = 0.75$). This has to do with the fact that L_1 is quite small (meaning it is unlikely that $n < L_1$).

CASE 4: μ doubled and λ is constant

($2\mu_1 = \mu_2$ and $\lambda_1 = \lambda_2$).

Looking at the example when $\mu_1 = 1$, $\mu_2 = 2$ and $\lambda_1 = 0.1 = \lambda_2$ we see that the service rate has doubled and the arrival rate has remained constant. If there are no customers in line SP should join queue 1 first (the queue with the slower service rate). This case has the same pattern as the case where the speed of the system is doubled. When the number of customers in both lines is initially the same, SP should

to join queue 2 (for small $m = n$ except when there are 0 customer in each line). As the number of customers ($m = n$) increases, SP should eventually switch to join queue 1 first (the same as the case where system speed is doubled).

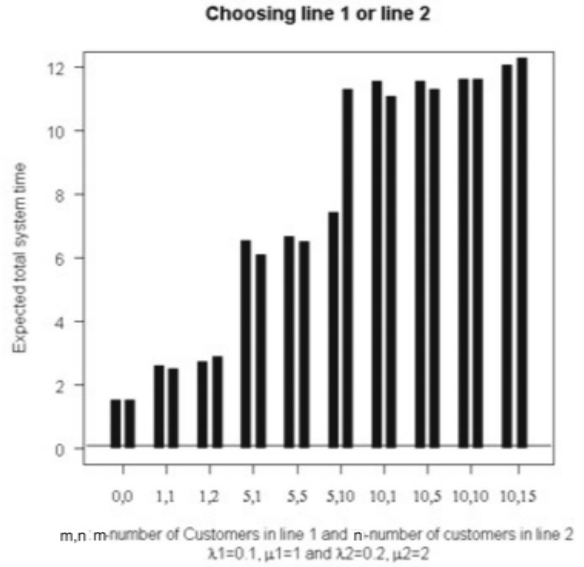
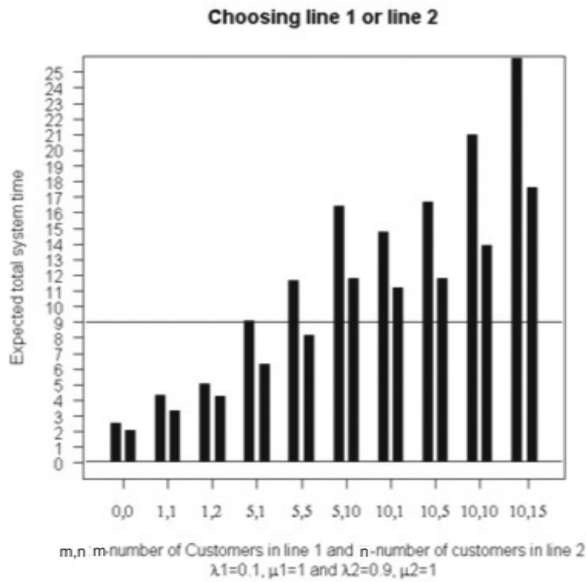
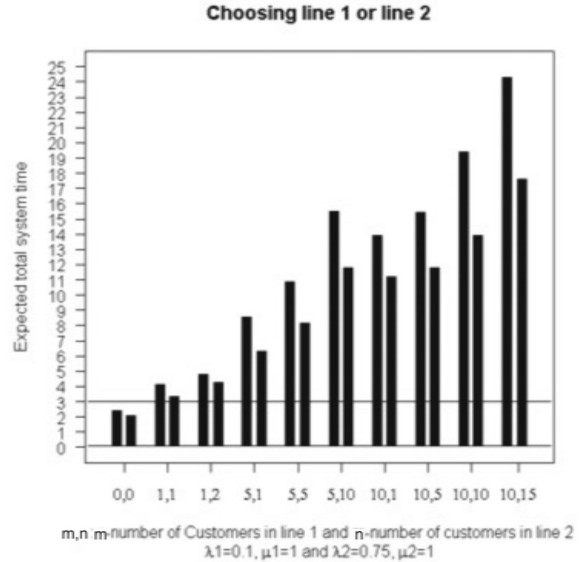
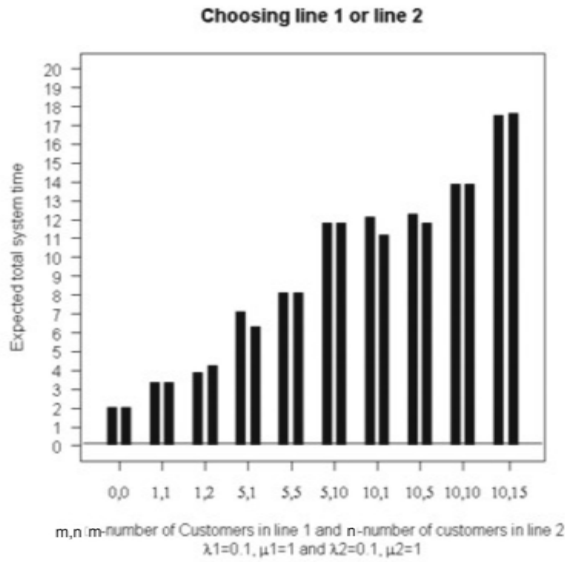
Note: All other cases follow similar patterns($\lambda_1 > \lambda_2$, queue speed halved, etc.). The case where the system speed is doubled represents the same pattern as other multiples of system speed. A summary of the cases can be found in the conclusion section of this report.

5. GRAPHICAL ANALYSIS

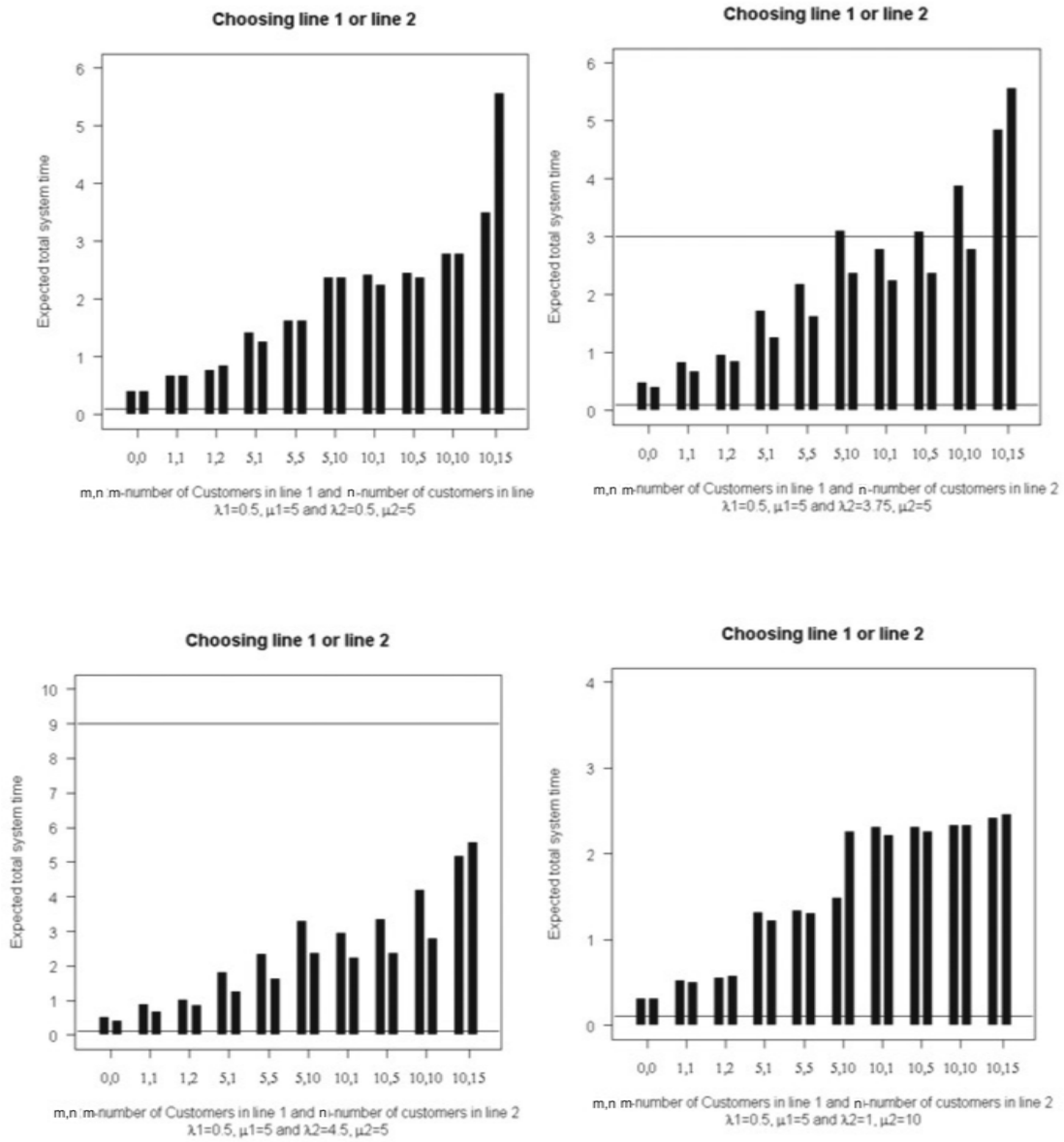
The graphs compare total expected system times $E(TST_1)$ and $E(TST_2)$ for various λ, μ, m and n .

The double bar graphs are set up as follows:

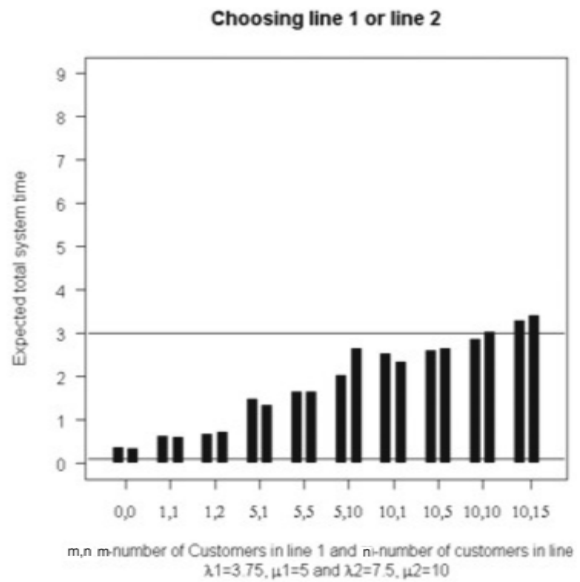
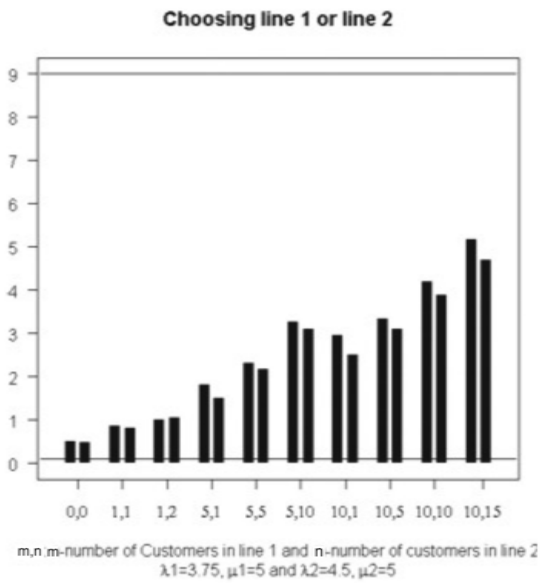
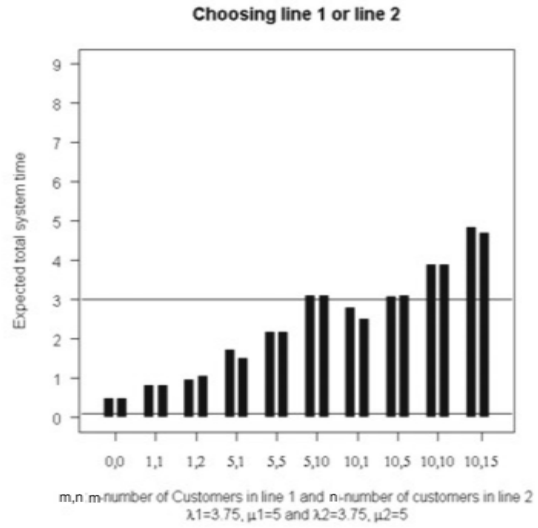
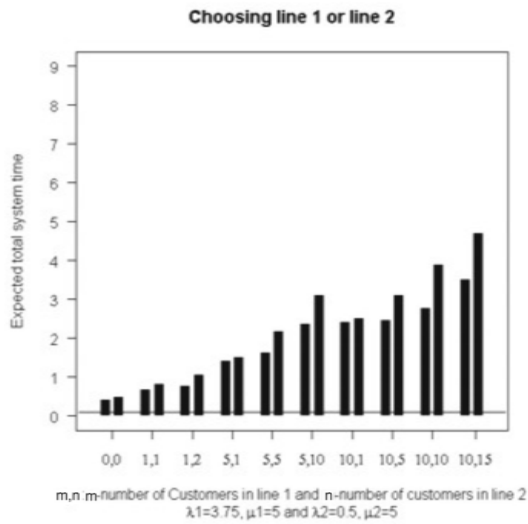
- (1) Each page has specific values of λ_1 and μ_1 .
- (2) Each individual double bar graph represents specific values of λ_2 and μ_2 .
- (3) On the horizontal axis pairs (m, n) are given. The vertical axis gives the two total expected system times .
- (4) The two bars represent the $ETST1$ and $ETST2$ for the given values of m and n .
- (5) the horizontal line(s) represents L_1 and L_2 . Note: when $L_1 = L_2$ you will see only one line drawn on the graph.



This set of graphs represent when $\lambda_1 = 0.1, \mu_1 = 1, \rho_1 = 0.1$ and $L_1 = 0.11$. Each graph represents different values for λ_2, μ_2, ρ_2 , and L_2 . With μ fixed in the first three graphs, they represents how λ_2 changes the expected total system times. The last graph shows what happens when the entire system speed is doubled (by doubling μ_2 , doubling λ_2 and keeping ρ constant).



This set represents when $\lambda_1 = 0.5$, $\mu_1 = 5$, $\rho_1 = 0.1$ and $L_1 = 0.11$. Like the previous set of graphs, each graph represents different values for λ_2 , μ_2 , ρ_2 , and L_2 . μ is fixed in the first three graphs representing how λ_2 changes the expected total system times. The last graph represents what happens when the entire system speed is doubled (by doubling μ_2 , doubling λ_2 and keeping ρ constant).



The last set of graphs represents when $\lambda_1 = 3.75$, $\mu_1 = 5$, $\rho_1 = 0.75$ and $L_1 = 3$. Each graph represents different values of λ_2 , μ_2 , ρ_2 , and L_2 with μ fixed in the first three graphs (representing what happens when λ_2 varies). The last graph doubles the entire system speed (by doubling μ_2 , doubling λ_2 and keeping ρ constant).

The graphs that represent the identical queues are graph 1 in set 1, graph 1 in set 2, and graph 2 in set 3. We can observe the pattern when the queues are identical more clearly in the graphs. We can see that like in the calculations we always choose the shorter line when $m > n$ and if the lines are the same $m = n$ we are indifferent (the expected total system time is the same). We can also observe the switch to the longer line that occurs when $m < n$ (this is clear in graph 2 in set 3).

Looking at the last graph in each section we see the case where the speed of the queue has doubled. It is interesting to observe the bars where m and n are equal. Notice as m and n increase the bars get closer together. This represents what eventually becomes a switch of preference between which queue to choose first. We initially choose the faster queue first and eventually save time by choosing the slower queue first. The switch in the first two sets happens just after 10 people so is not shown in the graphs, but you can see in the last set this switch comes sooner.

The last thing to observe in the sets is the differences in the first three graphs. This represents the case where we let λ vary. The graphs show what happens when $\lambda_1 < \lambda_2$, when $\lambda_1 = \lambda_2$ and when $\lambda_1 > \lambda_2$.

6. CONCLUSION AND ACKNOWLEDGEMENTS

Performing calculations we found several interesting patterns. Each of the following cases represents a unique pattern:

- (1) $\lambda_1 = \lambda_2$, $\mu_1 = \mu_2$ and $m > n$: We always choose the shorter queue (queue 2).

- (2) $\lambda_1 = \lambda_2, \mu_1 = \mu_2$ and $m = n$: We are indifferent between the queues.
- (3) $\lambda_1 = \lambda_2, \mu_1 = \mu_2$ and $m < n$: We initially choose the shorter queue and later switch to longer queue (speed of switch is based on ρ).
- (4) $2\lambda_1 = \lambda_2, 2\mu_1 = \mu_2, \rho_1 = \rho_2$ and $m = n$: initially choose queue 2 then switch to queue 1.
- (5) $2\lambda_1 = \lambda_2, 2\mu_1 = \mu_2, \rho_1 = \rho_2$ and $m > n$: initially choose queue 2 then switch to queue 1.
- (6) $\lambda_1 \ll \lambda_2$ and $\mu_1 = \mu_2$: always choose the line with the higher arrival rate first.
- (7) $\lambda_1 < \lambda_2$ and $\mu_1 = \mu_2$: initially choose the line with the higher arrival rate when m is small we eventually switch and choose the line with the lower arrival rate first.
- (8) $2\mu_1 = \mu_2$ and $\lambda_1 = \lambda_2$: follows the same pattern as the doubled speed case.

This research was partially funded by NSERC-Natural Sciences and Engineering Research Council (of Canada).

References

- (1) He, Q.-M. and Neuts, M.F. (2002) Two $M/M/1$ queues with transfers of customers, *Queueing Systems*, 42, 377-400, 2002.
- (2) Parlakturk, A. and Kumar. S. (2004). Self-interested routing in queueing networks. *Management Science* 50(7), 949967.
- (3) Winston, W. (1977). Optimality of the shortest line discipline. *J. Appl. Prob.*, 14:181–189.
- (4) Weber, R.R. (1978). On the optimal assignment of customers to parallel servers. *J. Appl. Prob.* 15, 406–413.
- (5) Whitt, W. (1986). Deciding which queue to join: some counterexamples. *Operations Research*, 34(1), 55-62.
- (6) Conolly, B.W. and Langaris, C. (1993). On a new formula for the transient state probabilities for $M/M/1$ queues and computational implications, *J. Appl. Prob.* 30, 237-246.
- (7) Hlynka, M. and Molinaro, S. (2009) Calculating Transient Probabilities of an $M/M/1$ Queue: R Program and Test Bank. University of Windsor Technical Report WMSR #09-07. 15 pp.