Calhoun
Institutional Archive of the Naval Postgraduate School

Calhoun: The NPS Institutional Archive

DSpace Repository

Faculty and Researchers | Faculty and Researchers' Publications

2008-10

Scheduling Policies for an Antiterrorist Surveillance System

Kress, Moshe; Lin, Kyle Y.; Szechtman, Roberto

2009 “Scheduling Policies for an Antiterrorist Surveillance System”, (with K. Lin and R. Szechtman), Naval Research Logistics (NRL), V. 56, No. 2, pp 113-126.
http://hdl.handle.net/10945/38174

Downloaded from NPS Archive: Calhoun

http://www.nps.edu/library

Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

Scheduling Policies for an Antiterrorist Surveillance System

Kyle Y. Lin, Moshe Kress, and Roberto Szechtman

Operations Research Department

Naval Postgraduate School, Monterey, California 93943

October 31, 2008

Abstract

This paper concerns scheduling policies in a surveillance system aimed at detecting a terrorist attack in time. Terrorist suspects arriving at a public area are subject to continuous monitoring, while a surveillance team takes their biometric signatures and compares them with records stored in a terrorist database. Because the surveillance team can screen only one terrorist suspect at a time, the team faces a dynamic scheduling problem among the suspects. We build a model consisting of an M/G/1 queue with two types of customers—red and white—to study this problem. Both types of customers are impatient, but the reneging time distributions are different. The server only receives a reward by serving a red customer, and can use the time a customer has spent in the queue to deduce its likely type. In a few special cases, a simple service rule—such as first-come-first-serve—is optimal. We explain why the problem is in general difficult, and develop a heuristic policy motivated by the fact that terrorist attacks tend to be rare events.

Keywords: homeland security, counterterrorism, multiclass queue, reneging, dynamic scheduling.

1 Introduction

Terrorist attacks—such as bombing, assassination of political figures, and release of poison gas in a crowd—are serious threats in many regions of the world. A significant terrorist attack occured in 1972 at a ticket counter in Lod International airport near Tel Aviv, Israel, where a three-man hit squad from the Japanese Red Army killed 26 people and injured 78 more. More recent examples include the 9/11 attacks in 2001, the Bali bombings in 2002 and 2005, and the London bombings in 2005. Numerous instances in the past suggest that terrorists often aim their attacks at crowded locations—such as restaurants, transportation terminals, popular tourist spots, political rallies, and subway stations—to create chaos and cause damage. The consequences of such terrorist attacks are casualties, damaged property, and a major disruption of daily life.

Response actions for mitigating and countering terrorist attacks include political and social policies that aim at deterring recruitment of terrorists, as well as protection of potential targets by police or military forces. While the authorities spend considerable resources going after the sources of such attacks—the terrorist organizations and their infrastructure—it is still important to have the ability to thwart terrorist attacks by timely detection and effective response. In this paper, we focus on that last line of defense—the problem of detecting, as early as possible, a developing terrorist attack on a public target.

We consider a large public area—henceforth called *arena*—where people can come and go freely, such as an airport lobby or a popular tourist attraction. An array of video cameras monitors the arena and feeds real-time video streams to a control center, where a security team screens people in two phases. In the first phase, people entering the arena are examined visually and each person is immediately put into one of two groups: *nonsuspects* and *suspects*. Only suspects are subject to the second-phase screening, which includes taking their biometric signatures (such as face structure, hair color, etc.) and running them through a terrorist database for comparison. In case of a positive match, the suspect is classified as a *potential terrorist* and security forces are notified to take proper actions; otherwise, the suspect is reclassified as a nonsuspect and the security team moves on to conduct the second-phase screening on another suspect. Because in the second phase, the security team can screen only one suspect at a time, the team faces a dynamic scheduling problem among the suspects with the goal to maximize the probability of detecting a terrorist attack in time.

Two observations motivate our research problem. First, because the second-phase screening takes time, the security team may not be able to inspect all suspects before they leave the arena. Second, a terrorist's intention and action are different from those of other people in the arena, so that the distribution of the time he spends in the arena may be different too. Consequently, by carefully choosing which suspect to inspect next, one could increase the probability of detecting a terrorist attack in time. In this paper, we develop a queueing model with impatient customers of unknown identity to analyze this problem, and draw insights into the effect of scheduling policies on such a surveillance system. Several attempts have recently been made to model and analyze detection and response actions associated with counterterrorism and homeland security; for example, see [8, 10, 11, 12, 14, 20]. However, we are not aware of any work that addresses this type of situation.

The contributions of this paper are twofold. From a theoretical standpoint, we build a queueing model with impatient customers that describes the antiterrorist surveillance system. There are two types of customers—terrorists and nonterrorists. The novelty of this queueing model is that only one type of customer (terrorists) is worth serving, but the server does not know a customer's identify until service completion. From an application standpoint, we develop dynamic scheduling policies for an antiterrorist surveillance system that can improve the probability of detecting a terrorist in a crowded area.

The rest of this paper is organized as follows. In Section 2, we discuss the operational setting and develop a queueing model. In Section 3, we identify a few special cases where the optimal policy can be explicitly determined. In Section 4, we discuss why the optimal policy is difficult to derive in general, and develop a heuristic policy. Conclusions and future research directions are discussed in Section 5.

2 The Model

In this section, we build a mathematical model to study the second-phase screening discussed in Section 1. Suppose the suspects that are subject to the second-phase screening arrive according to a Poisson process with rate λ. Each suspect is independently a red customer (terrorist) with probability p, or a white customer (nonterrorist) with probability $1-p$. It is helpful to keep in mind that p is very small because terrorist attacks generally are rare events. Both types of customers are impatient, and will leave the arena after a random

amount of time regardless of whether the second-phase screening has started. The time a red customer spends in the arena—called *reneging time* in queueing theory—represents the time it takes for a terrorist to initiate an attack. The time a white customer spends in the arena represents the time an innocent civilian wanders in the arena. We assume that the reneging time distribution of red customers $F_R(\cdot)$ can be estimated from intelligence and past events, while that of white customers $F_W(\cdot)$ can be estimated from data.

The second-phase screening comprises a continuous monitoring of the suspect, while running the suspect's biometric signature through a terrorist database for comparison. In our queueing model, the security team is the *server* who provides *service*—second-phase screening—to customers one at a time. The service time follows a distribution function $F_S(\cdot)$, independent of the customer's identity. The objective of the server is to detect a red customer in time so that the security forces can take proper actions to prevent or mitigate the attack. In other words, only red customers are valuable for service. The server, however, cannot tell the identity of a customer until after the service.

To define an objective function for the problem, note that whereas a customer waiting in queue may depart the system due to his impatience, a customer in service will depart the system either due to his impatience or due to service completion, whichever occurs first. If the departing customer is white, the process continues; if the departing white customer was in service, the server becomes available and immediately chooses another customer in the queue to serve. The process ends as soon as a red customer departs the system for the first time, which includes three possible scenarios. First, if the red customer departs before service is initiated, the server *fails* and receives a reward of 0, because a terrorist attack takes place without warning and the damage will be at its greatest. Second, if the red customer departs due to service completion, the server *succeeds* and receives a reward of 1, because the screening team identifies the terrorist in time to prevent the attack. Last, if the red customer departs while in service—due to the initiation of an attack—the server *succeeds partially* and receives a reward of $r \in [0, 1]$, because the security team identifies the terrorist and can respond to it quickly. The rationale that the surveillance process stops as soon as a red customer departs is that in all three scenarios the police or military force will take charge immediately—locking down the arena, evacuating the civilians, etc.—which makes continual surveillance irrelevant. Therefore, the *objective* of the server is to schedule the service sequence in real time in order to maximize the expected reward when

the surveillance process ends. In other words, our objective is to maximize the probability of preventing a terrorist attack, if we interpret r as the probability that an attack can be prevented if a terrorist initiates the attack while being watched by the surveillance team. Note that the server may still fail even when the first arriving red customer enters service, because another red customer may arrive and renege before the first arriving red customer departs.

The server's problem is to decide which customer in the queue to serve each time the server becomes available. Specifically, we can delineate the state of the queue by

$$(t_1, t_2, \ldots, t_n), \qquad t_1 > t_2 > \cdots > t_n,$$

with the interpretation that there are n customers in the queue, and the ith customer has spent t_i time units in the queue. We do not need to include the time since the last customer arrival in the state space because the customer arrival process is a Poisson process. A feasible policy is a function that maps a vector $(t_1, \ldots, t_n)$ to an index $i \in \{1, \ldots, n\}$, for $n = 1, 2, \ldots$.

In queueing theory, there is extensive research that concerns dynamic scheduling of multiclass queueing networks. In a service center, different classes of customers bring in different revenue and require different service times; see, for example, Miller [15] and Harrison [6]. In a production system, switching from one customer class to the other may require setup times; see, for example, Reiman and Wein [18] and Olsen [16]. For real-time scheduling problems involving impatient customers, see Gaver et al. [2], Glazebrook et al. [4], Jouini et al. [9], and the references therein. More recently, there is a growing interest in multiclass queues in heavy traffic; for example, see Bertsimas and Mourtizinou [1], Plambeck et al. [17], and Harrison and Zeevi [7]. The major distinction between our model and these earlier works is that in our model, a customer does not reveal his identity upon arrival, and the server can gather information about a customer's identity by studying how long the customer has spent in the queue. To the best of our knowledge, our work is the first to address this type of problem.

3 Exponential Reneging Time Distribution

This section presents the case when both F_R and F_W are exponential. In Subsection 3.1 we study the *first-come-first-serve* rule, and in Subsection 3.2 we study the *last-come-first-serve*

rule. In Subsection 3.3 we consider the *random-selection* rule, and compare all three rules numerically. Although our primary interest is to study a nonpreemptive service system, in Subsection 3.4 we discuss a preemptive service system that complements our theoretical results.

3.1 First-Come-First-Serve (FCFS) Rule

With the FCFS rule, the server always serves the customer who has spent the longest time in the queue. If the reneging time distributions for both red and white customers are exponential, the next theorem presents a sufficient condition for the FCFS rule to be optimal. Note that the theorem does not require the service time distribution F_S to be exponential, nor does it require the arrival process to be a Poisson process.

Theorem 3.1 *If both F_R and F_W are exponential with respective rates $\theta_R < \theta_W$, then the FCFS rule is optimal for any $r \in [0,1]$, for an arbitrary distribution function F_S, and for an arbitrary arrival process.*

Proof: Consider an arbitrary state $(t_1, t_2, \ldots, t_n)$ such that $t_1 > t_2 > \cdots > t_n$. We first want to show that for any policy that does not start with customer 1, we can find a better policy by starting with customer 1. The proof relies on an argument that involves stochastic coupling between two sample paths. A reference to the stochastic coupling technique can be found in Section 9.2 in Ross [19].

Let $p(t)$ denote the probability that a customer in the queue is red if he has spent t time units in the queue. Using Bayes' rule, we can calculate that

$$p(t) = \frac{p\bar{F}_R(t)}{p\bar{F}_R(t) + (1-p)\bar{F}_W(t)}, \tag{1}$$

where $\bar{F}_R(t) \equiv 1 - F_R(t)$ is the tail distribution function of a red customer's reneging time, and $\bar{F}_W(t) \equiv 1 - F_W(t)$ is that of a white customer's reneging time. Because both F_R and F_W are exponential with respective rates $\theta_R < \theta_W$, it follows that $p(t)$ increases in t (the first derivative of Equation (1) is positive). Therefore, we have that $p(t_1) > p(t_2) > \cdots > p(t_n)$.

Consider two servers—server A and server B—each facing the state $(t_1, \ldots, t_n)$. Suppose server B uses a policy ϕ, in which $\phi(t_1, \ldots, t_n) = i \neq 1$. Consider a policy for server A as follows: Serve customer 1 first. If server A finds customer 1 to be white and no red

customer has left (unserved) yet, then (1) if customer i is not in the queue, switch to policy ϕ thereafter; (2) if customer i is still in the queue, then increment the state variable of that customer by $t_1 - t_i$ (so that customer will be treated as a customer who has spent an additional $t_1 - t_i$ time units in the queue) and switch to policy ϕ thereafter.

Because $p(t_1) > p(t_i)$, we are able to couple customer 1's identity and customer i's identity in queues A and B in five cases as follows (see Table 1 for a summary). Define a random variable I to indicate which case takes place, and let X denote the reward for server A, and Y the reward for server B. For brevity, we use A-i to denote customer i in queue A, and so on.

Table 1: The identities of customers can be coupled stochastically in five cases, used in the proof of Theorem 3.1.

	Queue A		Queue B	
Probability	Customer 1	Customer i	Customer 1	Customer i
$(1-p(t_1))(1-p(t_i))$	white	white	white	white
$p(t_1)p(t_i)$	red	red	red	red
$(1-p(t_1))p(t_i)$	red	white	white	red
$(1-p(t_1))p(t_i)$	white	red	red	white
$p(t_1)-p(t_i)$	red	white	red	white

1. With probability $(1-p(t_1))(1-p(t_i))$, A-1, A-i, B-1, and B-i are all white. Because of the memoryless property of the exponential distribution, we can couple A-1 and B-i such that they will renege at the same time. Similarly, we can couple A-i and B-1 such that they will renege at the same time. We further couple the service times for the two servers such that the kth service initiated by server A takes the same amount of time as the kth service initiated by server B, $k = 1, 2, \ldots$. Finally, we couple the identities of the other $n-2$ customers in the queue and their respective remaining times to renege, as well as the arrival times of future customers, their identities, and their reneging times. By doing so, we can see that both queues will follow the same sample path—except that customer labels 1 and i are swapped in the two queues (server A will serve customer i in queue A if and only if server B serves customer 1 in queue

B). Consequently, for each sample path, the two servers will earn an identical reward; therefore, $E[X|I = 1] = E[Y|I = 1]$.

2. With probability $p(t_1)p(t_i)$, A-1, A-i, B-1, and B-i are all red. By coupling the sample paths between the two queues exactly the same way as in case 1, we can see that servers A and B will earn an identical reward in each sample path. Therefore, we can conclude $E[X|I = 2] = E[Y|I = 2]$.

3. With probability $(1-p(t_1))p(t_i)$, A-1 and B-i are red, while A-i and B-1 are white. As in case 1, we can conclude $E[X|I = 3] = E[Y|I = 3]$.

4. With probability $(1-p(t_1))p(t_i)$, A-1 and B-i are white, while A-i and B-1 are red. As in case 1, we can conclude $E[X|I = 4] = E[Y|I = 4]$.

5. With probability $p(t_1)-p(t_i)$, A-1 and B-1 are red, while A-i and B-i are white. Because both A-1 and B-1 are red, we can couple A-1 and B-1 such that they will renege at the same time. Similarly, we can couple A-i and B-i such that they will renege at the same time. We further couple the service times for the two servers such that the kth service initiated by server A takes the same amount of time as the kth service initiated by server B, $k = 1, 2, \ldots$. Finally, we couple the identities of the other $n-2$ customers in the queue and their respective remaining times to renege, as well as the arrival times of future customers, their identities, and their reneging times. Consider the next event that occurs.

   (a) If the next event to occur is a service completion, then server A earns 1, while server B may eventually earn 0, r, or 1. The probability server B will earn 0 or r is nonzero.

   (b) If the next event to occur is the reneging of A-1 and B-1, then server A earns r while server B earns 0.

   (c) If the next event to occur is the reneging of A-i and B-i, then server B will choose another customer to serve. At that point, we can repeat the whole stochastic coupling argument for cases 1–5 listed in this proof.

   (d) If the next event to occur is the reneging of any of the other $n-2$ customers, then both servers will earn 0 if that customer is red. If that reneging customer is

white, then the process continues, and we can consider the *next* event and repeat the argument in cases (a)–(e).

(e) If the first event to occur is the arrival of a new customer, then the process continues, and we can consider the *next* event and repeat the argument in cases (a)–(e).

Consequently, we can see that in each sample path, server A will earn a reward greater than or equal to what server B will earn. Therefore, $E[X|I=5] > E[Y|I=5]$.

Taking all 5 cases together, we can write that

$$E[X] - E[Y] = \sum_{k=1}^{5} (E[X|I=k] - E[Y|I=k]) \cdot P\{I=k\} > 0.$$

Hence, any policy that does not select customer 1 cannot be optimal. In addition, staying idle and begin service at a later time, perhaps to a newly-arrived customer, cannot be optimal either, which can be proven by a similar coupling argument. Consequently, it must be optimal to select customer 1.

Finally, because the preceding argument applies each time the server becomes available, it follows that the FCFS rule is optimal. □

Although Theorem 3.1 holds for an arbitrary value of p, we are particularly interested in the case when $p \to 0$, because terrorist attacks tend to be rare events. To compute the expected reward as $p \to 0$, we first construct a queue with only white customers arriving according to a Poisson process with rate λ, and then let a red customer arrive in steady state. We can obtain a closed-form solution for the expected reward if the service time distribution is also exponential.

Suppose F_S is exponential with rate μ. With white customers arriving according to a Poisson process with rate λ, the steady-state probability that there are n customers in the system can be found by a birth-death process (see Gross and Harris [5], pages 122–124), and is given by

$$\frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} (\frac{\lambda}{\mu + i\theta_W})}, \qquad \text{for } n = 0, \tag{2}$$

and

$$\frac{\prod_{i=1}^{n} (\frac{\lambda}{\mu + i\theta_W})}{1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} (\frac{\lambda}{\mu + i\theta_W})}, \qquad \text{for } n = 1, 2, \ldots.$$

The queue is stable as long as $\theta_W > 0$, regardless of the values of μ and λ.

If a red customer finds n white customers in the system upon arrival, then the probability that he will enter service before reneging is the probability that all those n white customers depart—either due to impatience or due to service completion—before the red customer reneges. This probability can be obtained by the memoryless property of the exponential distribution:

$$\prod_{i=1}^{n}\Big(\frac{\mu + i\theta_W}{\mu + i\theta_W + \theta_R}\Big).$$

Therefore, with the FCFS rule, the probability that the red customer arriving in steady state will enter service before reneging is

$$\begin{aligned}
&\frac{1}{1+\sum_{k=1}^{\infty}\prod_{i=1}^{k}(\frac{\lambda}{\mu+i\theta_W})} + \sum_{n=1}^{\infty}\Big[\Big(\frac{\prod_{i=1}^{n}(\frac{\lambda}{\mu+i\theta_W})}{1+\sum_{k=1}^{\infty}\prod_{i=1}^{k}(\frac{\lambda}{\mu+i\theta_W})}\Big)\prod_{i=1}^{n}\Big(\frac{\mu + i\theta_W}{\mu + i\theta_W + \theta_R}\Big)\Big] \\
&= \frac{1+\sum_{k=1}^{\infty}\prod_{i=1}^{k}(\frac{\lambda}{\mu+i\theta_W+\theta_R})}{1+\sum_{k=1}^{\infty}\prod_{i=1}^{k}(\frac{\lambda}{\mu+i\theta_W})}
\end{aligned} \tag{3}$$

Once the red customer enters service, we can deduce, once again due to the memoryless property of the exponential distribution, that the service will complete before the red customer reneges with probability $\mu/(\mu+\theta_R)$. Consequently, the expected reward for the FCFS rule is Equation (3) multiplied by $(\mu + r\theta_R)/(\mu + \theta_R)$.

3.2 Last-Come-First-Serve (LCFS) Rule

With the LCFS rule, the server always serves the customer who most recently joined the queue. Somewhat surprisingly, the counterpart of Theorem 3.1 when $\theta_R > \theta_W$ is not true even if F_S is also exponential. For example, if there is only one customer in the queue, and that customer has been in the queue for a long time (so that the customer is most likely white), then the server may prefer waiting for the next new arrival rather than serving that very old customer, as shown in the next example.

Example 3.1

Suppose $\lambda = \mu = 1$, $\theta_R = 10$, $\theta_W = 0.1$, and $p = 0.8$. Consider a situation when there is only one customer in the queue—referred to as customer Z throughout this example—who joined the queue one time unit ago. According to Equation (1), customer Z is a red customer with probability $p(1) \approx 0.0002$.

With the LCFS rule, the server initiates service with customer Z. Let X denote the reward received by the server with the LCFS rule. Compute $P\{X=0\}$ by conditioning on the identity of customer Z:

$$\begin{aligned} P\{X=0\} &= p(1)P\{X=0|\text{Z is red}\} + (1-p(1))P\{X=0|\text{Z is white}\} \\ &> 0 + (1-p(1))\frac{\lambda}{\lambda+\mu+\theta_W}\, p\, \frac{\theta_R}{\mu+\theta_W+\theta_R}, \end{aligned} \tag{4}$$

where the inequality follows because conditional on customer Z being white, $X=0$ as long as the following three events occur sequentially: (1) a new customer arrives before customer Z departs (whether due to impatience or due to service completion); (2) the new customer is red; and (3) the new customer reneges (unserved) before customer Z departs.

To compute $P\{X=r\}$ and $P\{X=1\}$, note that to get a positive reward, the server needs to select a red customer at some point. Once that happens, there is still a chance for another red customer to renege before the red customer in service departs. However, if the process does end because the red customer in service departs, then the probabilities whether the ending is due to impatience or due to service completion are proportional to their respective exponential rates θ_R and μ. Therefore, we conclude that

$$\frac{P\{X=r\}}{P\{X=1\}} = \frac{\theta_R}{\mu}.$$

Hence, we have that

$$\begin{aligned} E[X] &= 1\cdot P\{X=1\} + r\cdot P\{X=r\} + 0\cdot P\{X=0\} \\ &= (1-P\{X=0\})\frac{\mu}{\mu+\theta_R} + r(1-P\{X=0\})\frac{\theta_R}{\mu+\theta_R} \\ &< \left(1-(1-p(1))\frac{\lambda}{\lambda+\mu+\theta_W}\, p\, \frac{\theta_R}{\mu+\theta_W+\theta_R}\right)\frac{\mu+r\theta_R}{\mu+\theta_R} \\ &\approx 0.05972(1+10r), \end{aligned} \tag{5}$$

where the inequality follows from Equation (4).

An alternative policy is for the server to stay idle until a new customer arrives, and then immediately serve the newly-arrived customer. Let Y denote the reward under this policy. Compute $P\{Y=1\}$ by conditioning on the identity of customer Z:

$$\begin{aligned} P\{Y=1\} &= p(1)P\{Y=1|\text{Z is red}\} + (1-p(1))P\{Y=1|\text{Z is white}\} \\ &> 0 + (1-p(1))\, p\, \frac{\mu}{\lambda+\mu+\theta_R}, \end{aligned}$$

where the inequality follows because conditional on customer Z being white, the server will receive a reward of 1 as long as the first arrival is a red customer, and the service for that red customer completes before the red customer reneges and before another new customer arrives. Similarly, we have that

$$P\{Y = r\} > (1 - p(1))\, p\, \frac{\theta_R}{\lambda + \mu + \theta_R}.$$

Therefore,

$$\begin{aligned} E[Y] &= 1 \cdot P\{Y = 1\} + r \cdot P\{Y = r\} + 0 \cdot P\{Y = 0\} \\ &> (1 - p(1))\, p\, \frac{\mu + r\theta_R}{\lambda + \mu + \theta_R} \\ &\approx 0.06665(1 + 10r). \end{aligned} \tag{6}$$

From Equations (5) and (6), we conclude that $E[Y] > E[X]$ for any $r \in [0, 1]$, which implies that the LCFS rule is not optimal. □

Although the LCFS is not optimal as seen by Example 3.1, it is indeed the optimal policy if the server is not allowed to stay idle when there are customers in the queue.

Theorem 3.2 *If both F_R and F_W are exponential with respective rates $\theta_R > \theta_W$ and if the server is not allowed to stay idle when there are customers in the queue, then the LCFS rule is optimal for any $r \in [0, 1]$, for an arbitrary distribution function F_S, and for an arbitrary arrival process.*

The proof is similar to that of Theorem 3.1 and is therefore omitted, as we can show that serving the customer who most recently joined the queue is better than serving anyone else. This argument also shows that even if strategic idling is allowed, whenever a customer is chosen for service, it should be the most recent arrival.

As we did for the FCFS rule in Section 3.1, we let $p \to 0$ and calculate the expected reward under the LCFS rule. As $p \to 0$, we can find this probability by first constructing a queue with only white customers and letting a red customer arrive in steady state. We first calculate the probability that the red customer arriving in steady state will *ever* enter service before reneging. On the one hand, if the server is idle when a red customer arrives, then the red customer enters service immediately. On the other hand, if the server is busy when a red customer arrives, then with the LCFS rule, the current number of white customers in the

system is irrelevant to whether the red customer will enter service before reneging. In this case, we construct a Markov chain to represent the state of the system when a red customer is present. Denote by k the state if the server is busy with a white customer, and there are $k-1$ white customers in the queue who arrived after the red customer, $k = 1, 2, \ldots$. Let the state become 0 when the red customer enters service, and -1 when the red customer reneges before entering service. Note that by definition, states 0 and -1 are absorbing, and that the Markov chain starts in state 1.

Let α_k, $k = 1, \ldots, \infty$, denote the probability that the Markov chain in state k will *ever* enter state $k-1$ before entering state -1 (the red customer reneges unserved). We need to determine α_1, the probability that a red customer will enter service before reneging if the server is busy upon his arrival.

To obtain α_1, we first find an expression for α_k by conditioning on whether the next event is a new arrival, a departure of a white customer, or the departure of the unserved red customer:

$$\alpha_k = \frac{\lambda}{\theta_R + \lambda + \mu + k\theta_W} \cdot \alpha_{k+1}\alpha_k + \frac{\mu + k\theta_W}{\theta_R + \lambda + \mu + k\theta_W} \cdot 1 + \frac{\theta_R}{\theta_R + \lambda + \mu + k\theta_W} \cdot 0,$$

for $k = 1, 2, \ldots$. Solving for α_k yields

$$\alpha_k = \frac{\mu + k\theta_W}{\theta_R + \lambda + \mu + k\theta_W - \lambda\alpha_{k+1}}. \tag{7}$$

Because $\alpha_{k+1} \in [0, 1]$, the preceding implies that

$$\frac{\mu + k\theta_W}{\theta_R + \lambda + \mu + k\theta_W} < \alpha_k < \frac{\mu + k\theta_W}{\theta_R + \mu + k\theta_W}. \tag{8}$$

Consequently, we can choose a large value of k, use Equation (8) to bound α_k, and then use Equation (7) to recursively compute the bounds for $\alpha_{k-1}, \alpha_{k-2}, \ldots, \alpha_1$. Because the bounds converge very quickly, we can approximate α_1 satisfactorily.

Finally, we can compute the probability that the red customer enters service before leaving under the LCFS rule by

$$\begin{aligned} &1 \cdot P\{\text{server idle in steady state}\} + \alpha_1 \cdot P\{\text{server busy in steady state}\} \\ &= \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} (\frac{\lambda}{\mu + i\theta_W})} + \alpha_1 \Big(1 - \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=1}^{k} (\frac{\lambda}{\mu + i\theta_W})}\Big), \end{aligned} \tag{9}$$

where the steady-state probability is given by Equation (2). With the same reason given at the end of Section 3.1, the expected reward with the LCFS rule is Equation (9) multiplied by $(\mu + r\theta_R)/(\mu + \theta_R)$.

3.3 Random Selection (RS) Rule

Another service rule of interest is the RS rule, in which the server, when becoming available, randomly selects a customer in the queue to serve. If both reneging times are exponentially distributed and $\theta_R = \theta_W$, then all three rules—FCFS, LCFS, and RS—perform equally well for two reasons: (1) each customer in the queue has a probability p of being red regardless of the amount of time he has spent in the queue; and (2) the remaining times to renege for all customers in the queue are independent and identically distributed because of the memoryless property of the exponential distribution. If $\theta_R \neq \theta_W$, we would expect that the performance of the RS rule lies between those of the FCFS and the LCFS rules.

As $p \to 0$, we can formulate a Markov chain to compute the expected reward of the RS rule as we did in Sections 3.1 and 3.2. We omit the derivation. To compare the three service rules, note that in order for the server to earn a positive reward, the red customer arriving in steady state (of a system that consists of only white customers) needs to enter service before reneging. If the red customer does enter service, then the expected reward becomes $(\mu + r\theta_R)/(\mu + \theta_R)$ as derived at the end of Section 3.1. Therefore, the relative performance among the three rules is independent of r—the reward of partial success. For this reason, to compare the three service rules when $p \to 0$, we plot in Figures 1 and 2 the probability that the arriving red customer will ever enter service before reneging—namely Equation (3) for the FCFS rule and Equation (9) for the LCFS rule.

As seen in Figures 1 and 2, the FCFS rule is the best of the three when $\theta_R < \theta_W$ (in this case, the FCFS is optimal according to Theorem 3.1), while the LCFS rule is the best when $\theta_R > \theta_W$. Examples of arenas for the case $\theta_R < \theta_W$ include the vending machine corners, walkways, stairs, and parking lots, where a typical visitor tends to leave fairly soon. In this case, although the FCFS rule is optimal, its performance is quite sensitive to the changes in either θ_W or θ_R. Examples for the case $\theta_R > \theta_W$ include department stores, public parks, picnic areas, and other places where a typical visitor tends to spend a long time. In this case, although the LCFS rule is not optimal, its performance is relatively robust to the change in θ_W and θ_R, especially when $\theta_R \approx \theta_W$. This observation suggests that if the value of θ_R is highly uncertain—as θ_W is typically much easier to estimate—then the LCFS rule may be preferred.

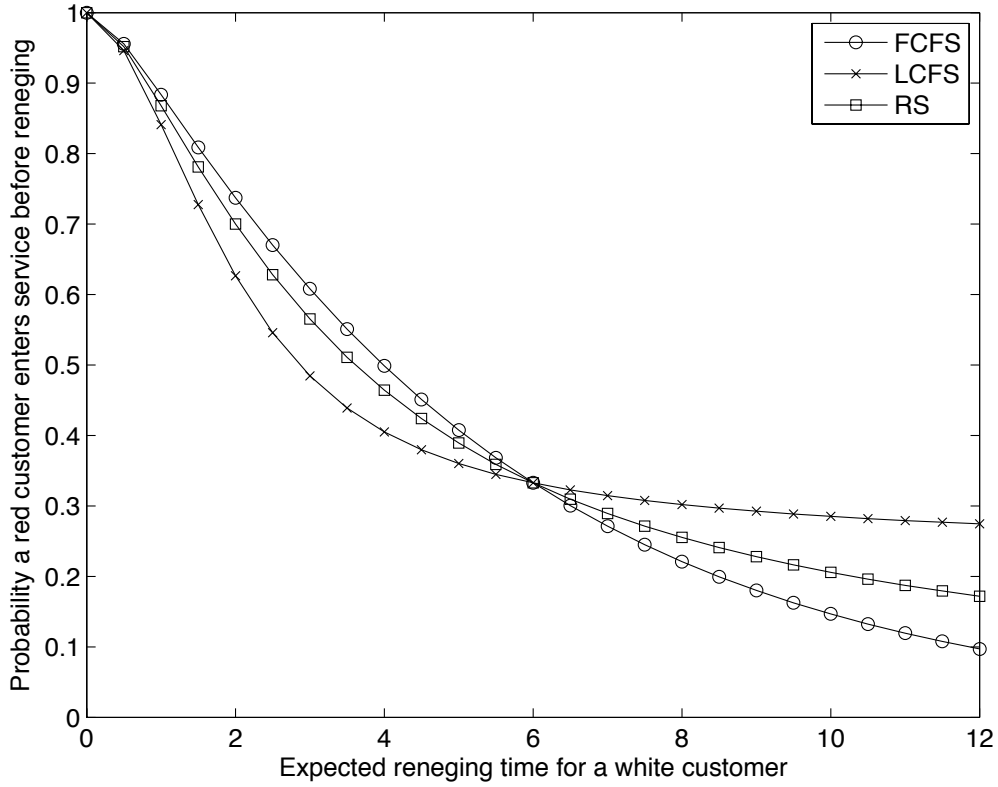

Figure 1: Comparison of three service rules when the expected reneging time for a white customer varies. F_R, F_W, and F_S are all exponential; $p \to 0$, $\lambda = 2$, $1/\theta_R = 6$, and $1/\mu = 2$.

3.4 Preemptive Service

Our queueing model assumes that the service is nonpreemptive because the screening process of a suspect cannot be interrupted. If preemptive service is allowed, the server can switch to another customer upon a new arrival or any departure by interrupting the current screening, and picks up where it left off when the service resumes. This subsection presents a theorem that complements Theorem 3.1 in the nonpreemptive service case.

Theorem 3.3 *If the service is preemptive and F_S is exponential, and both F_R and F_W are exponential with respective rates $\theta_R < \theta_W$ (respectively, $\theta_R > \theta_W$), then the FCFS (respectively, LCFS) rule is optimal for any $r \in [0, 1]$ and for an arbitrary arrival process.*

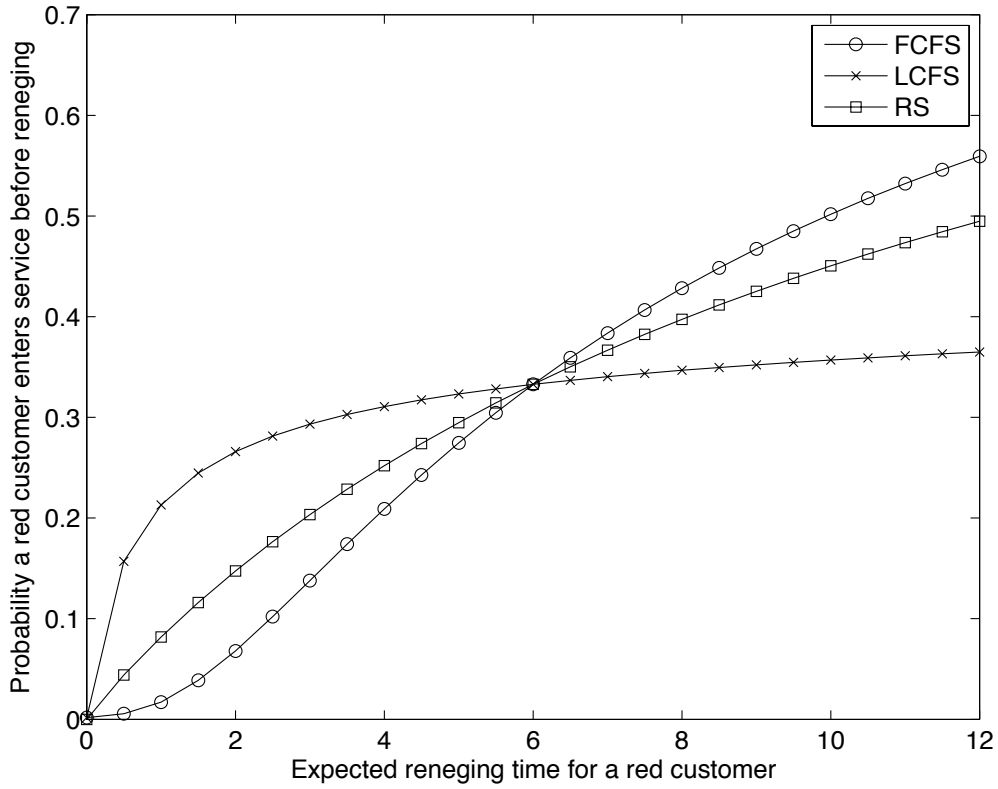

Figure 2: Comparison of three service rules when the expected reneging time for a red customer varies. F_R, F_W, and F_S are all exponential; $p \to 0$, $\lambda = 2$, $1/\theta_W = 6$, and $1/\mu = 2$.

We omit the proof of Theorem 3.3 because it is similar to that of Theorem 3.1. Note that contrary to Theorem 3.1, however, Theorem 3.3 does not hold for an arbitrary service time distribution F_S, which can be understood intuitively as follows. Suppose that F_R and F_W are exponential with respective rates $\theta_R < \theta_W$, and that F_S has a decreasing failure rate—that is, $f_S(t)/\bar{F}_S(t)$ decreases in t, for $t > 0$—so that the longer a customer has been in service, the longer (stochastically) his remaining service time becomes. Granted, with the FCFS rule, the server always serves the customer who has the highest probability of being red. However, after serving the same customer for a long time without a conclusion, the remaining service time tends to be even longer (in the regular stochastic sense). At that point, the server may prefer to switch to another customer for a fresh service time, even though this other customer is less likely a red customer.

4 General Reneging Time Distributions

This section presents the case when F_R and F_W do not follow exponential distributions. In Subsection 4.1, we explain why it is difficult to find the optimal policy for general reneging time distributions. In Subsection 4.2, we develop a heuristic policy. In Subsection 4.3, we use Monte Carlo simulation to numerically evaluate the heuristic policy.

4.1 Special Cases and Counterexamples

We first investigate whether we can relax the exponential assumption on F_R and F_W so that the FCFS rule remains optimal under weaker conditions. Intuitively, for the FCFS rule to be optimal, two conditions need to hold: (a) the longer a customer has spent in the queue, the more likely he is a red customer; and (b) the longer a white customer has spent in the queue, the sooner he tends to leave the queue according to his reneging distribution.

For condition (a) to hold, we need $p(t)$ in Equation (1) to increase in t. Differentiating $p(t)$ shows that a sufficient condition for $p(t)$ to increase in t is for F_R to have a smaller failure rate than F_W. In other words, we need

$$\frac{f_R(t)}{\bar{F}_R(t)} \leq \frac{f_W(t)}{\bar{F}_W(t)}, \qquad \text{for } t > 0, \tag{10}$$

where f_R and f_W are the density functions, and $\bar{F}_R$ and $\bar{F}_W$ are the tail distribution functions, for the reneging times of red customers and white customers, respectively. For condition (b) to hold, we need the random variable $W_t \equiv (W - t \,|\, W > t)$ to decrease in t in the regular stochastic sense, where W denotes a random variable with distribution function F_W. That is, we need W to be increasing in failure rate (IFR); see Chapter 9 in Ross [19].

Let R denote a random variable with distribution function F_R, and define $R_t \equiv (R - t \,|\, R > t)$. Next, we examine whether the FCFS rule remains optimal in two cases: R is decreasing in failure rate (DFR), and R is IFR.

Red customer's reneging time is DFR

When R is DFR, R_t increases in t in the regular stochastic sense. In other words, the longer a red customer has spent in the queue, the longer his additional reneging time tends to be. On the one hand, the probability of completing service for a red customer increases with the

time a red customer has spent in the queue, which makes the FCFS rule appealing, especially when $r = 0$ so that a partial success is worthless. On the other hand, the server may prefer to serve the customer who joined the queue most recently, because that customer—if red—tends to renege the soonest. The example below shows that R being DFR is not a sufficient condition for the FCFS rule to be optimal.

Example 4.1

Suppose that the reneging time of a white customer is exponentially distributed with rate equal to 1, and that of a red customer has the following failure rate function:

$$\frac{f_R(t)}{\bar{F}_R(t)} = \begin{cases} 1, & \text{for } 0 \le t < 2, \\ 0.01, & \text{for } t \ge 2. \end{cases}$$

Also suppose that $p = 0.5$, and the service time distribution F_S is deterministic and equal to 0.5. In addition, assume λ is extremely small (say 10^{-6}), so that the effect of future arrivals is negligible. Consider a scenario when the server finds two customers in the queue with $t_1 = 2$ and $t_2 = 1$.

Compare two service orders 1, 2 (FCFS) and 2, 1. If both customers are white, then with either service order the service process continues after both customers depart. If at least one of the two customers is red, then the service process ends when (or *before*) both customers depart. With some analysis (see Appendix), it turns out that, conditional on at least one customer being red, the expected reward is $0.6711 + 0.0927r$ for service order 1, 2, and $0.7337 + 0.2637r$ for service order 2, 1. Therefore, for any $r \in [0, 1]$, the FCFS rule is not optimal. This conclusion is somewhat intuitive, because the server still has a great chance to serve customer 1 by starting with customer 2, but not vice versa. □

Red customer's reneging time is IFR

When R is IFR, R_t decreases in t in the regular stochastic sense. If a red customer tends to leave sooner the longer he has spent in the queue, then one may argue that the server should give the priority to the customer who has spent the longest time in the queue, especially when $r = 1$. Therefore, if both conditions (a) and (b) hold, and if R is IFR, it makes intuitive sense for the FCFS rule to be optimal. However, this conjecture is not true even in two special cases, when $r = 0$ and $r = 1$, as shown in Examples 4.2 and 4.3, respectively.

Example 4.2

Suppose $r = 0$. Let F_R and F_W be identical with the following failure rate function:

$$\frac{f_R(t)}{\bar{F}_R(t)} = \frac{f_W(t)}{\bar{F}_W(t)} = \begin{cases} 0, & \text{for } 0 \leq t < 2, \\ \infty, & \text{for } t \geq 2. \end{cases}$$

In other words, both F_R and F_W are deterministic and equal to 2. Suppose that the service time distribution F_S is also deterministic and equal to 1. In addition, assume λ is extremely small (say 10^{-6}), so that the effect of future arrivals is negligible. Consider a scenario when the server finds two customers in the queue with $t_1 = 1.1$ and $t_2 = 0.5$.

If the server follows the service order 1, 2 (FCFS), then both customers will renege before service completion, regardless of their identities. The service order 2, 1 is better, because the server will receive a reward of 1, if customer 1 is white and customer 2 is red. □

Example 4.3

Suppose $r = 1$. Let F_R and F_W be identical with the following failure rate function:

$$\frac{f_R(t)}{\bar{F}_R(t)} = \frac{f_W(t)}{\bar{F}_W(t)} = \begin{cases} 0, & \text{for } 0 \leq t < 2, \\ 1, & \text{for } 2 \leq t < 4, \\ \infty, & \text{for } t \geq 4. \end{cases}$$

Also suppose that the probability of a red customer is $p = 0.5$. Because F_R and F_W are identical, the server cannot learn about a customer's identity from the amount of time the customer has spent in the queue. Hence, $p(t) = p = 0.5$ for all $t > 0$.

Suppose there are three customers with $t_1 = 2.99$, $t_2 = 2$, and $t_3 = 1$, and that the service time distribution F_S is deterministic and equal to 1. In addition, assume λ is extremely small (say 10^{-6}), so that the effect of future arrivals is negligible. Compare two service orders 1, 2, 3 (FCFS) and 2, 1, 3. If all three customers are white, then with either service order the service process continues after all three customers depart. If at least one of the three customers is red, then the service process ends when (or *before*) all three customers depart. With some analysis and Monte Carlo simulation (see Appendix), it turns out that, conditional on at least one customer being red, the expected reward is 0.720 for the service order 1, 2, 3, and 0.752 for the service order 2, 1, 3 (standard error approximately 10^{-5}). Therefore, it is better to start with customer 2 rather than with customer 1, and the FCFS rule is not optimal. □

To gain some intuition about this example, first note that the failure rate function remains a constant for $2 \leq t < 4$, and the service time is deterministic and equal to 1. Because $[t_i, t_i + 1] \subset [2, 4)$ for $i = 1, 2$, the time it takes for the server to become available is identically distributed regardless of whether the server starts with customer 1 or customer 2. In addition, if the server starts with customer 1, the probability that customer 2 is still in the queue when the server becomes available is the same as the probability that customer 1 is still in the queue if the server starts with customer 2. Consequently, the number of customers between customers 1 and 2 that the server can serve by following the order 2, 1, 3 is identically distributed to that number when the server follows the order 1, 2, 3.

However, by starting with customer 2, the time it takes for the server to become available for customer 3 is stochastically smaller than by starting with customer 1, because as soon as a customer spends 4 time units in the queue, he will leave immediately. Consequently, by starting with customer 2, the server has a better chance to serve customer 3.

The three examples in this section show that even in the special cases when $r = 0$ and $r = 1$, the FCFS rule is not optimal under some plausible conditions. To determine the optimal policy for an arbitrary r can only be more difficult. Therefore, we next turn our attention to a heuristic policy.

4.2 Heuristic Policy

One possible greedy policy is for the server to always select the customer that yields the highest expected reward. The drawback of this policy, however, is that the server may waste too much time on a customer whose expected reward is only marginally higher than the other customers. Because the server's time is valuable, we propose a heuristic policy where the server selects the customer with the highest reward rate—the ratio between the expected reward and the expected time spent if the customer is served. The idea of indexing each customer by the reward rate is reminiscent of the Gittins index used in other problems, where the effort is sequentially allocated among a number of competing projects; see Gittins [3]. The difference, however, is that in our problem the existing projects (customers) may disappear before service, while new projects may show up in the future.

Let R, W, and S denote random variables with respective probability distribution functions F_R, F_W, and F_S. Suppose a customer has spent t time units in the queue, then serving

that customer yields a reward rate equal to

$$\begin{aligned}\gamma(t) &\equiv \frac{E[\text{reward received from serving a customer who has spent } t \text{ time units in queue}]}{E[\text{time spent on serving a customer who has spent } t \text{ time units in queue}]} \\ &= \frac{p(t)\cdot(P\{R_t > S\} + rP\{R_t \le S\}) + (1-p(t))\cdot 0}{p(t)E[\min(R_t,S)] + (1-p(t))E[\min(W_t,S)]},\end{aligned}$$

where $p(t)$ is given by Equation (1), the probability a customer is red if he has spent t time units in the queue. Letting $p \to 0$, we can compare the reward rate between any two customers by

$$\lim_{p\to 0}\frac{\gamma(t_1)}{\gamma(t_2)} = \frac{\left(\dfrac{\bar{F}_R(t_1)(r+(1-r)P\{R_{t_1} > S\})}{\bar{F}_W(t_1)E[\min(W_{t_1},S)]}\right)}{\left(\dfrac{\bar{F}_R(t_2)(r+(1-r)P\{R_{t_2} > S\})}{\bar{F}_W(t_2)E[\min(W_{t_2},S)]}\right)}. \tag{11}$$

Therefore, we define a score function for a t-time-unit-old customer as

$$s(t) \equiv \frac{\bar{F}_R(t)(r+(1-r)P\{R_t > S\})}{\bar{F}_W(t)E[\min(W_t,S)]}, \tag{12}$$

and let the server choose the customer who has the highest score.

To further compute Equation (12), we calculate

$$\begin{aligned}P\{R_t > S\} &= \int_0^\infty P\{R - t > x | R > t\} f_S(x)dx \\ &= \int_0^\infty \frac{\bar{F}_R(t+x)}{\bar{F}_R(t)} f_S(x)dx,\end{aligned} \tag{13}$$

and

$$\begin{aligned}E[\min(W_t,S)] &= \int_0^\infty P\{\min(W_t,S) > x\}dx \\ &= \int_0^\infty P\{W_t > x \text{ and } S > x\}dx \\ &= \int_0^\infty P\{W_t > x\}P\{S > x\}dx \\ &= \int_0^\infty \frac{\bar{F}_W(t+x)}{\bar{F}_W(t)}\bar{F}_S(x)dx.\end{aligned} \tag{14}$$

Consequently, putting Equations (12)–(14) together gives

$$s(t) = \frac{r\bar{F}_R(t) + (1-r)\int_0^\infty \bar{F}_R(t+x)f_S(x)dx}{\int_0^\infty \bar{F}_W(t+x)\bar{F}_S(x)dx}. \tag{15}$$

The score in Equation (15) is computed for each customer individually, based on the time a customer has spent in the queue. One advantage of this score is that it is easy to

compute. In practice, we can compute the score $s(t)$ for all values of t beforehand, which allows easy implementation in real time. Observe that a customer's score does not depend on the number of other customers in the queue nor the customer arrival rate λ. Therefore, this heuristic cannot be optimal in general. In particular, when the traffic is relatively light, the server may be more concerned with the actual reward earned—as opposed to the reward rate—when choosing the next customer, because the server's time may not be a significant constraint. Hence, in a light-traffic system it is possible to devise a policy that is specifically tailored for given distributions F_R, F_W, and F_S. On the other hand, in a surveillance system under heavy traffic—the case we expect to see in applications—there are many customers to choose from each time the server becomes available. Because the server would be kept busy most of the time, it should focus on selecting the customer that yields the highest reward rate. Consequently, we expect our heuristic policy to be effective for a system in heavy traffic.

When F_R, F_W, and S follow exponential distributions with respective rates θ_R, θ_W, and μ, the score function in Equation (15) becomes

$$s(t) = (\theta_W + \mu)\left(\frac{r\theta_R + \mu}{\theta_R + \mu}\right)e^{-(\theta_R - \theta_W)t}.$$

When $\theta_R < \theta_W$, the preceding increases in t, so the heuristic policy coincides with the FCFS rule—the optimal policy according to Theorem 3.1. When $\theta_R > \theta_W$, the heuristic policy coincides with the LCFS rule—the optimal nonidling policy.

4.3 Numerical Experiment

This subsection presents a numerical example. Let F_R follow the Erlang distribution with shape parameter 6 and scale parameter 1, and F_W the Erlang distribution with shape parameter 2 and scale parameter 3. We choose the two distributions to have the same expected value, namely 6, because if they are much different, the heuristic policy resembles either the FCFS rule or the LCFS rule. We also choose the distributions so that F_W has a larger variance, because the visiting purposes of the white customers are more diverse. The choice of the Erlang distribution is primarily due to its unimodal density function. The service time is largely deterministic, but we let F_S follow a uniform distribution to allow for a small variance.

Instead of varying r, we plot the score function in Equation (15) for $r = 0$ in Figure 3, and for $r = 1$ in Figure 4. Because $s(t)$ is a linear function in r, for an arbitrary $r \in [0, 1]$, the score function is basically a weighted average between these two extreme cases.

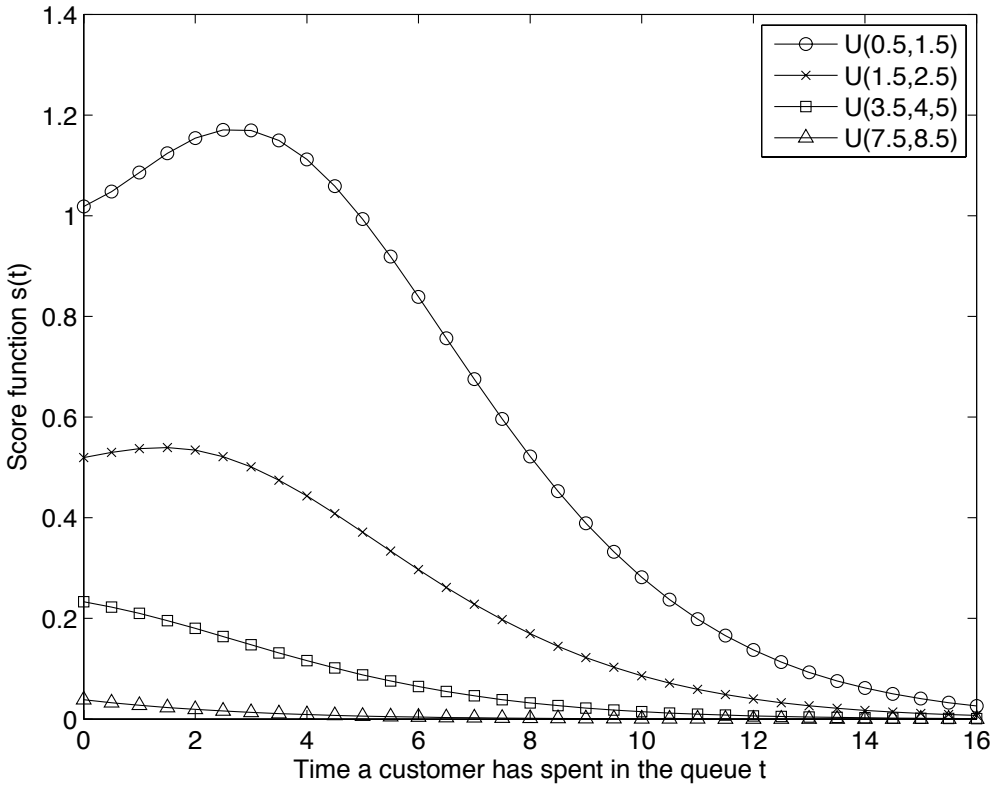

Figure 3: Score function $s(t)$ for $r = 0$; $F_R \sim$ Erlang$(6, 1)$, $F_W \sim$ Erlang$(2, 3)$, and F_S follows four different uniform distributions.

In the case $r = 0$, the server does not earn any reward if a red customer reneges during service. As seen in Figure 3, when $F_S \sim U(0.5, 1.5)$, the score function $s(t)$ largely coincides with

$$\frac{p(t)}{p(0)} = \frac{\bar{F}_R(t)}{p\bar{F}_R(t) + (1-p)\bar{F}_W(t)} \approx \frac{\bar{F}_R(t)}{\bar{F}_W(t)}, \qquad \text{as } p \to 0.$$

In other words, when the service time is small, the server selects the next customer primarily based on the likelihood of the customer being red, because most likely the service will complete before the customer reneges. When the service time is large, however, it becomes less desirable to serve a customer who has spent longer in the queue (the Erlang distribution is

IFR), because the chance of reneging during service becomes larger. When $F_S \sim U(7.5, 8.5)$, the service time is so great that the heuristic policy coincides with the LCFS rule.

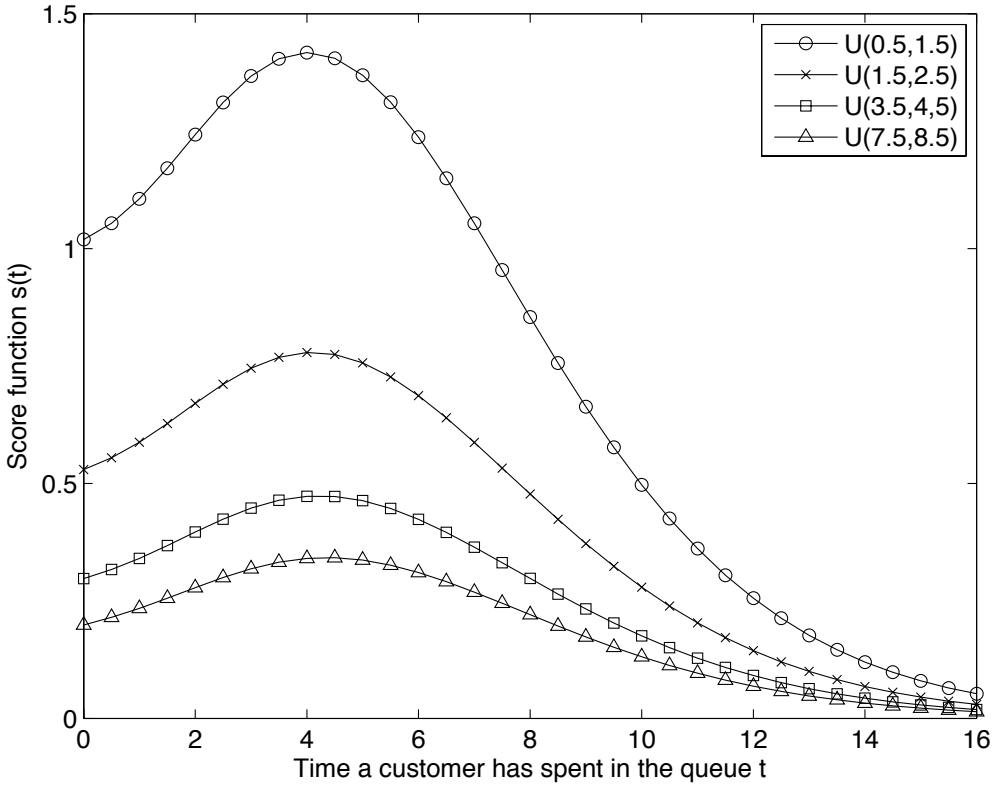

Figure 4: Score function $s(t)$ for $r = 1$; $F_R \sim \text{Erlang}(6, 1)$, $F_W \sim \text{Erlang}(2, 3)$, and F_S follows four different uniform distributions.

In the case $r = 1$, the server earns a reward of 1 for a partial success. As seen in Figure 4, $s(t)$ still follows a similar shape to $\bar{F}_R(t)/\bar{F}_W(t)$ when $F_S \sim U(0.5, 1.5)$. When the service time becomes larger, the peak of $s(t)$ shifts to the right slightly. A customer who has spent a longer time in the queue becomes more attractive because the server can save time if it is a white customer; in addition, the server does not need to complete service to earn a reward of 1 if it is a red customer.

We next compare the performance of the heuristic policy with the other three naive service rules—FCFS, LCFS, and RS rules—using the same example. To simulate the performance of a policy, note that as $p \to 0$, a red customer will arrive in steady state to a queueing

system that consists of only white customers arriving according to a Poisson process with rate λ. However, it is inefficient to collect only one estimate each time we generate a steady state. To overcome this issue, we generate a sample path of the queueing system where white customers arrive according to a Poisson process with rate λ for the first n (a large number) arrivals, and let the server process customers according to a given service rule—FCFS, LCFS, RS, or heuristic. After generating the sample path, we turn our attention to each customer one at a time. For the jth arriving customer, define the random variable Z_j as the reward the server would have earned had customer j been a red customer, while all other customers remain white. Hence, our estimator is $(\sum_{j=1}^{n} Z_j)/n$.

Our simulation algorithm uses steady-state simulation to collect multiple estimates in a single simulation run. There are two issues related to a steady-state simulation. First, there is initial bias because the system is not in steady state when we start the simulation with an empty queue. Second, the random variables Z_j and Z_{j+1} are not independent. If $Z_j = 0$, it becomes more likely for the queue to have many customers, which in turn makes Z_{j+1} more likely to also take on value 0. To resolve these two issues, we allow a prolonged warm-up period before collecting data, and use batch means to estimate the standard error of our estimate; see, for example, Law and Kelton [13]. We choose the batch size so that with probability close to 1 the first customers in consecutive batches will never coexist in the system.

In the simulation experiment, we choose $F_S \sim U(1.5, 2.5)$, and simulate three cases for $r = 0$, 0.5, and 1. Table 2 compares the expected reward as $p \to 0$ for four policies when the arrival rate λ varies from 1 to 5. In each case, we use a sufficiently large n so that the standard error is about 10^{-3} of the estimate. We choose the performance of the RS rule as the benchmark, and report the performance of the other three rules as ratios to the benchmark.

As seen in Table 2, the heuristic policy always yields the highest expected reward. In addition, the heuristic policy's relative improvement over the RS rule gradually increases as λ increases. This observation is not surprising, as we argued in Section 4.2, the heuristic policy is particularly suitable for a system in heavy traffic, in which the server can often select a high-score customer from a full spectrum of customers.

The FCFS rule performs well when $\lambda = 1$, especially in the case $r = 1$. As seen in Figure 4, the score function $s(t)$ ($F_S \sim U(1.5, 2.5)$) increases when t is small. When λ is

Table 2: Expected reward for different policies as $p \to 0$; $F_R \sim$ Erlang$(6,1)$, $F_W \sim$ Erlang$(2,3)$, and $F_S \sim U(1.5, 2.5)$.

			Ratio to RS Rule			
r	λ	RS	RS	FCFS	LCFS	Heuristic
	1	0.464	1.000	0.927	1.076	1.096
	2	0.217	1.000	0.617	1.210	1.233
0	3	0.141	1.000	0.417	1.243	1.273
	4	0.104	1.000	0.299	1.260	1.294
	5	0.083	1.000	0.228	1.270	1.304
	1	0.538	1.000	1.028	0.987	1.082
	2	0.264	1.000	0.821	1.028	1.170
0.5	3	0.173	1.000	0.610	1.033	1.197
	4	0.128	1.000	0.473	1.039	1.211
	5	0.102	1.000	0.374	1.042	1.223
	1	0.612	1.000	1.106	0.922	1.132
	2	0.311	1.000	0.963	0.900	1.230
1	3	0.205	1.000	0.747	0.890	1.257
	4	0.153	1.000	0.586	0.888	1.270
	5	0.122	1.000	0.473	0.886	1.278

small, often there are only a few customers who are new to the system, so the FCFS rule often selects the same customer as does the heuristic policy. When $\lambda = 5$, however, often the queue is full of customers, many of which have spent a long time in the queue. In that case, the FCFS rule would select a customer that has been in the queue for a long time, whereas the heuristic policy tends to select a customer who has spent about 4 time units in the queue. For $r = 0$ and $r = 0.5$, the FCFS rule also performs poorly for $\lambda = 5$, because $s(t)$ is decreasing for larger values of t.

The LCFS rule performs relatively well in the case $r = 0$, because in Figure 3 the score function $s(t)$ ($F_S \sim U(1.5, 2.5)$) is largely decreasing in t, so the LCFS rule and the heuristic policy often make the same decision. In the case $r = 1$, $s(t)$ in Figure 4 is unimodal with the

maximum occurring at about 4. When λ increases, the LCFS rule often selects a customer that just entered the queue, while the heuristic policy tends to select a customer that has spent about 4 time units in the queue. Therefore, the performance of the LCFS rule drops as λ increases.

5 Concluding Remarks

In this paper we developed a single-server queueing model with impatient customers to study a surveillance system aimed at detecting terrorists in real time. Two types of customers—terrorist and nonterrorist—arrive at the system, but a customer does not reveal his identity upon arrival. The server, however, can infer a customer's likely identity based on the time the customer has spent in the system. We presented a few cases in which the optimal policy can be explicitly determined, and studied a heuristic policy that performs well for a system in heavy traffic.

Because our study focused on the scheduling aspect of the screening operation, we assumed that the surveillance system has perfect sensitivity and perfect specificity. If the surveillance system were to erroneously classify a terrorist as a nonterrorist (false negative) with a certain probability, then the performance of the surveillance system described in this paper would simply be discounted by that probability. If false positive errors are also possible, then the actions taken by the authorities would incur a social cost associated with the disruption of normal daily life. This cost, however, is typically much smaller than that of a successful terrorist attack.

There are a few related research directions that can follow from our study. First, it is possible to extend the queueing model to allow multiple servers and more than two types of customers (catching a terrorist is more rewarding than catching a criminal fugitive). Second, the probability of classification errors can be modeled as a function of the time a target is under surveillance. The longer the surveillance system monitors a target, the more likely the classification will be correct. In this case, the service time becomes a controlled variable rather than a random parameter. We believe that mathematical modeling along these research lines has the potential to advance the effort on counterterrorism and homeland security.

Appendix

Derivation of Example 4.1

Let $I_j = 1$ if customer j is red, and $I_j = 0$ if customer j is white, $j = 1, 2$. Let $d = 0.5$ denote the deterministic service time. Let T_j denote the time until customer j reneges, provided that customer j is red, for $j = 1, 2$. Note that because both customers will depart (either due to reneging or service completion) within the next 1 ($= 2d$) time unit, for the purpose of this example it is sufficient to assume that T_1 is exponentially distributed with rate $\mu_1 = 0.01$, while T_2 is exponentially distributed with rate $\mu_2 = 1$. In addition, let T denote the time until a white customer reneges; T is exponentially distributed with rate $\mu = 1$.

Let X denote the reward received by the service order 1, 2. To compute $E[X]$, condition on the identities of the two customers. The case both customers are white is irrelevant, because we are interested in the expected reward conditional on at least one customer being red. When the first customer is red and the second is white, we have that

$$\begin{aligned} P\{X = 1 | I_1 = 1, I_2 = 0\} &= P\{T_1 > d\} = e^{-\mu_1 d} \approx 0.9950, \\ P\{X = r | I_1 = 1, I_2 = 0\} &= P\{T_1 < d\} \approx 0.0050. \end{aligned}$$

When both customers are red, we have that

$$\begin{aligned} P\{X = 1 | I_1 = 1, I_2 = 1\} &= P\{T_1 > d, T_2 > d\} = e^{-\mu_1 d} e^{-\mu_2 d} \approx 0.6035, \\ P\{X = r | I_1 = 1, I_2 = 1\} &= P\{T_1 < \min(T_2, d)\} \\ &= P\{T_1 < T_2, T_1 < d\} \\ &= P\{T_1 < T_2\} \cdot P\{T_1 < d | T_1 < T_2\} \\ &= \frac{\mu_1}{\mu_1 + \mu_2}(1 - e^{-(\mu_1+\mu_2)d}) \approx 0.0039, \end{aligned}$$

where the last equality follows because $(T_1 | T_1 < T_2)$ is exponentially distributed with rate $\mu_1 + \mu_2$.

When the first customer is white and the second is red, we have that

$$\begin{aligned} P\{X = 0 | I_1 = 0, I_2 = 1\} &= P\{T_2 < \min(T, d)\} \approx 0.3161, \\ P\{X = 1 | I_1 = 0, I_2 = 1\} &= P\{T_2 > \min(T, d) + d\} \\ &= P\{T_2 > \min(T, d)\} \cdot P\{T_2 - \min(T, d) > d \,|\, T_2 > \min(T, d)\} \\ &= \left(1 - \frac{\mu_2}{\mu + \mu_2}(1 - e^{-(\mu+\mu_2)d})\right) \cdot e^{-\mu_2 d} \approx 0.4148. \end{aligned}$$

Finally, according to Equation (1), $p(1) = p(2) = 0.5$. Because $P\{I_1 = 0, I_2 = 0\} = P\{I_1 = 0, I_2 = 1\} = P\{I_1 = 1, I_2 = 0\} = P\{I_1 = 1, I_2 = 1\} = 0.25$, if at least one customer is red, then the conditional expected reward is

$$\frac{E[X|I_1 = 0, I_2 = 1] + E[X|I_1 = 1, I_2 = 0] + E[X|I_1 = 1, I_2 = 1]}{3} \approx 0.6711 + 0.0927r.$$

With a similar approach, we can compute the conditional expected reward for the service order 2,1, to be approximately $0.7337 + 0.2637r$, if at least one customer is red.

Derivation of Example 4.3

We compute the expected reward by conditioning on the identities of the 3 customers in the queue. In some cases, the expected reward can be analytically computed, while in the other cases we use Monte Carlo simulation with 10^8 independent runs. Because the derivation is similar to that in Example 4.1, we omit it and summarize the results in Table 3.

Table 3: Expected reward conditional on the identities of the 3 customers in Example **??**.

			Service order	
1	2	3	1, 2, 3	2, 1, 3
W	W	W	—	—
W	W	R	0.883469^a	0.996387^a
W	R	W	$0.5 + 0.5e^{-2}$	1
W	R	R	0.451127^a	1
R	W	W	1	$0.5 + 0.5e^{-2}$
R	W	R	1	0.563970^a
R	R	W	$0.5 + 0.5e^{-2}$	$0.5 + 0.5e^{-2}$
R	R	R	$0.5 + 0.5e^{-2}$	$0.5 + 0.5e^{-2}$

[a]Simulation result with standard error less than 5×10^{-5}.

To compute the expected reward conditional on at least one customer being red, we take the arithmetic average over 7 cases by excluding the case where all 3 customers are white. Because for either service order, in 2 out of 7 cases we use Monte Carlo simulation to estimate

the expected reward, the standard error of the arithmetic average is equal to

$$\sqrt{\frac{2 \times (5 \times 10^{-5})^2}{7^2}} \approx 1.01 \times 10^{-5}.$$

Acknowledgments

This material is based upon work supported by the Research Initiation Program at the Naval Postgraduate School. The authors thank Sheldon Ross and Kevin Glazebrook for helpful discussions, and an associate editor and two referees for many valuable comments.

References

[1] Bertsimas, D. and Mourtzinou, G. 1997. Multiclass queueing systems in heavy traffic: An asymptotic approach based on distributional and conservation laws. *Operations Research* **45,** 470–487.

[2] Gaver, D. P., Jacobs, P. A., Samorodnitsky, G. and Glazebrook, K. D. 2006. Modeling and analysis of uncertain time-critical tasking problems. *Naval Research Logistics* **53,**.

[3] Gittins, J. C. 1989. *Multi-armed Bandit Allocation Indices*. John Wiley & Sons.

[4] Glazebrook, K. D., Ansell, P. S., Dunn, R. T. and Lumley, R. R. 2004. On the optimal allocation of service to impatient tasks. *Journal of Applied Probability* **41,** 51–72.

[5] Gross, D. and Harris, C. M. 1985. *Fundamentals of Queueing Theory* 2nd ed. Wiley.

[6] Harrison, J. M. 1975. Dynamic scheduling of a multiclass queue: Discount optimality. *Operations Research* **23,** 270–282.

[7] Harrison, J. M. and Zeevi, A. 2004. Dynamic scheduling of a multiclass queue in the Halfin-Whitt heavy traffic regime. *Operations Research* **52,** 243–257.

[8] Jacobson, S. H., McLay, L. A., Kobza, J. E. and Bowman, J. M. 2005. Modeling and analyzing multiple station baggage screening security system performance. *Naval Research Logistics* **52,** 30–45.

[9] Jouini, O., Pot, A., Dallery, Y. and Koole, G. 2007. Real-time scheduling policies for multiclass call centers with impatient customers. Working paper.

[10] Kaplan, E. and Kress, M. 2005. Operational effectiveness of suicide bomber detector schemes: A best-case analysis. *Proceedings of the National Academy of Sciences* **102,** 10399–10404.

[11] Kaplan, E. H., Patton, C. A., FitzGerald, W. P. and Wein, L. M. 2003. Detecting bioterror attacks by screening blood donors: A best-case analysis. *Emerging Infectious Diseases* **9,** 909–914.

[12] Kress, M. 2005. The effect of crowd density on the expected number of casualties in a suicide attack. *Naval Research Logistics* **52,** 22–29.

[13] Law, A. M. and Kelton, W. D. 2000. *Simulation Modeling and Analysis* 3rd ed. McGraw-Hill, New York City, NY.

[14] McLay, L. A., Jacobson, S. H. and Kobza, J. E. 2006. A multilevel passenger screening problem for aviation security. *Naval Research Logistics* **53,** 183–197.

[15] Miller, B. L. 1969. A queueing reward system with several customer classes. *Management Sciences* **16,** 234–245.

[16] Olsen, T. L. 1999. A practical scheduling method for multiclass production systems with setups. *Management Science* **45,** 116–130.

[17] Plambeck, E., Kumar, S. and Harrison, J. M. 2001. A multiclass queue in heavy traffic with throughput time constraints: Asymptotically optimal dynamic controls. *Queueing Systems* **39,** 23–54.

[18] Reiman, M. I. and Wein, L. M. 1998. Dynamic scheduling of a two-class queue with setups. *Operations Research* **46,** 532–547.

[19] Ross, S. M. 1996. *Stochastic Processes* 2nd ed. Wiley.

[20] Wein, L. M. and Liu, Y. 2005. Analyzing a bioterror attack on the food supply: The case of botulinum toxin in milk. *Proceedings of the National Academy of Sciences* **102,** 9984–9989.