

Application Example 18

Designing the checkout system of a supermarket

In some cases, the performance of a stochastic system (or a deterministic system in a random environment) can be evaluated by analytical or numerical methods. In other cases, the most expedite and flexible approach is to use Monte Carlo (MC) simulation. The MC technique consists of simulating the random variables involved and evaluating the response of the system under each simulation. The results from repeated simulations are then treated as a statistical sample.

If the system is subjected to conditions that vary over time in a stationary way, a single long simulation may suffice to assess the system's performance, as time becomes a surrogate of sample size. Examples are a traffic light system that is meant to reduce traffic congestion at an intersection and the checkout system of a supermarket, which aims at efficiently serving the customers. The latter problem is the one used here for illustration.

Checkout System

Suppose that the manager of a supermarket wants to design the checkout system (how many counters to keep open, how to assign customers to different counters, how to direct customers with few purchased items to fast lanes, which checkout technology to use, etc.) to be inexpensive to operate, use a small amount of floor space, and serve the customers as efficiently as possible (i.e. with minimum waiting time). Since some of these objectives are in conflict, the manager is looking for a reasonable compromise.

For simplicity, consider just two design variables: (a) the number N of the counters that are open, and (b) the queuing strategy. Also let m_T and λ be the mean service time when a customer reaches a counter and the mean customer arrival rate, respectively. Notice that the ratio N/m_T is the maximum rate at which the system can serve customers when N counters are open. It is then useful to introduce the overcapacity ratio c as

$$c = \frac{N}{\lambda m_T} \quad (1)$$

In order to avoid the formation of ever increasing queues, it must be $c > 1$. The objective of the analysis is to find an appropriate value for c .

Regarding the allocation of customers to counters, we consider three alternative strategies:

- (1) *Random allocation*. Customers are allocated at random and with equal probability to different counters, irrespective of the existing queues at the counters,
- (2) *Shortest queue*. Upon arrival at the check-out area, customers select the shortest queue (if more than one such queue exists, the customer chooses at random one of them), and
- (3) *Single queue*. In this case there is just one queue. The first customer in the queue goes to the first available counter. This first-in, first-out (FIFO) scheme is more efficient than the other two, in terms of average waiting time.

In schemes (1) and (2), once a customer has chosen a line, he is not allowed to switch to a different line.

The key problem is to evaluate for each check-out system the mean waiting time as a function of c .

Simulation

Suppose that customers arrive at the checkout area at Poisson times T_i , with mean rate λ . This means that the interarrival times $\Delta T_i = T_i - T_{i-1}$ are independent and identically distributed (iid) exponential random variables with mean value $m_{\Delta T} = 1/\lambda$.

Also suppose that the service times of different customers at the counter are iid variables with common exponential distribution and mean value m_T . Hence the times when customers leave a busy counter form a Poisson process with mean rate $\lambda_1 = 1/m_T$. In general, while n counters are serving customers, the times when customers leave the checkout area define a Poisson point process with mean rate $\lambda_n = n/m_T$.

Under the above conditions, simulation of the checkout process might proceed as follows. Suppose that at the present time t there are N open counters of which $n(t)$ are serving customers. In the case of checkout system (1) or (2), denote by $m_i(t) \geq 0$, $i = 1, 2, \dots, N$, the number of customers at the various counters, including the customers who are being served. In the case of checkout system (3), there are $m(t) \geq 0$ customers in the only available queue, waiting to be served.

Whenever a customer arrives at the checkout area, he is assigned a sequential number k and his waiting time W_k is initialized as 0. Using the memoryless property of the Poisson process, the

“state” of the system at time t changes at the time $t + \tau$ when either a new customer arrives or a customer is served and leaves. Therefore, τ has exponential distribution with mean value:

$$m_\tau = \frac{1}{\lambda + \lambda_{n(t)}} = \frac{1}{\frac{1}{m_{\Delta T}} + \frac{n(t)}{m_T}} \quad (2)$$

The probability that the system changes state due to arrival of a new customer is

$$P_{new} = \frac{\lambda}{\lambda + \lambda_{n(t)}} = \frac{1}{1 + n(t)\frac{m_{\Delta T}}{m_T}} \quad (3)$$

Hence simulation proceeds from one state-change time to the next as follows:

- i. Simulate τ using equation (2);
- ii. Simulate whether the change is due to arrival or departure of a customer, using equation (3);
- iii. Update the waiting times:
 - $W_k^{(new)} = W_k^{(old)} + \tau$ for all “unlocked” customers (i.e. customers that have not been served yet; see below).
 - If a new customer arrived, initialize his waiting time W_k as 0,
 - If a customer departed, “lock” his updated waiting time W_k so it is not changed at any future step.
- iv. Update other variables depending on the checkout system considered:

For System 1:

- If a new customer arrived, select at random a counter i (with equal probability $1/N$ for each counter) and set $m_i^{(new)} = m_i^{(old)} + 1$. If $m_i^{(new)} = 1$, increase $n(t)$ by 1.
- If a customer departed, select at random the counter i that served him (assigning equal probability to all counters with $m_i > 0$) and set $m_i^{(new)} = m_i^{(old)} - 1$. If $m_i^{(new)} = 0$, decrease $n(t)$ by 1.

For System 2:

- If a customer arrived, select at random one counter i among those for which m is minimum and set $m_i^{(new)} = m_i^{(old)} + 1$. If $m_i^{(new)} = 1$, increase n by 1.
- If a customer departed, do the same as for System 1.

For System 3:

- If a customer arrived and there is a counter i with $m_i = 0$, set $m_i = 1$. Otherwise increase m by 1.
- If a customer departed, select the first counter i with $m_i = 1$. If $m > 0$ set $m = m - 1$ (i.e. move the first waiting customer to counter i). Otherwise set $m_i = 0$.

Provided that $c \geq 1$, after an initial transient period the system reaches a steady state condition called stationarity. Under stationarity, one may use the “locked” waiting times W_k as a sample to find the mean checkout waiting time $m_W = E[W_k]$ under the conditions considered.

Figure 1 shows simulation results for the three checkout systems described above, under two different conditions. In the simulations we kept both the number of open counters N and the mean value of the customer inter-arrival time m_{AT} constant ($N = 5$ counters, $m_{AT} = 5\text{min}$) and varied the overcapacity parameter c . According to equation (1), this is equivalent to varying the mean service time m_T of the system.

Figures 1.a and 1.b show the estimated mean checkout time, $m_W = E[W_k]$, for the three checkout systems described as a function of simulation time. Notice that for $c > 1$ the average waiting time of all systems tends to stabilize as simulation time increases. This is expected since for $c > 1$ the average rate at which the system can serve customers exceeds the mean rate of customer arrival. Hence the checkout system approaches a condition of stationarity.

It is important to notice that, as c increases, m_W decreases. This is physically expected since for values of c close to 1 the system reaches its capacity limit. Those limits are associated with very long queues and very long waiting times. This is obvious if one compares Figure 1.a, where $c = 1.2$, with Figure 1.b, where $c = 2$. Independently of the type of check-out system, the waiting times in Figure 1.a are much higher than those in Figure 1.b.

Another important observation is that under the same simulation conditions (number of open counters, mean rate of customer arrival, and mean service time), system 1 is the most inefficient one and system 3 is the most efficient one.

To study the effect of the overcapacity ratio c on the average checkout time m_W , we define the *inefficiency ratio*,

$$IR = \frac{E[W_k]}{m_T} \quad (4)$$

as the ratio of the mean customer waiting time $m_W = E[W_k]$ and to the mean service time m_T . IR is always greater than 1, and all systems tend to have $IR = 1$ as $c \rightarrow \infty$.

Figure 2 shows the ratio IR as a function of c for the 3 checkout systems. Irrespective of the type of checkout system, IR decreases as c increases and tends to 1 as $c \rightarrow \infty$. System 1 is by far the most inefficient checkout scheme, especially for small values of c , whereas systems 2 and 3 have nearly the same efficiency.

In deciding about the number of counters to be kept open, one must further balance the inefficiency ratio in Figure 2 with the cost of operating the counters.

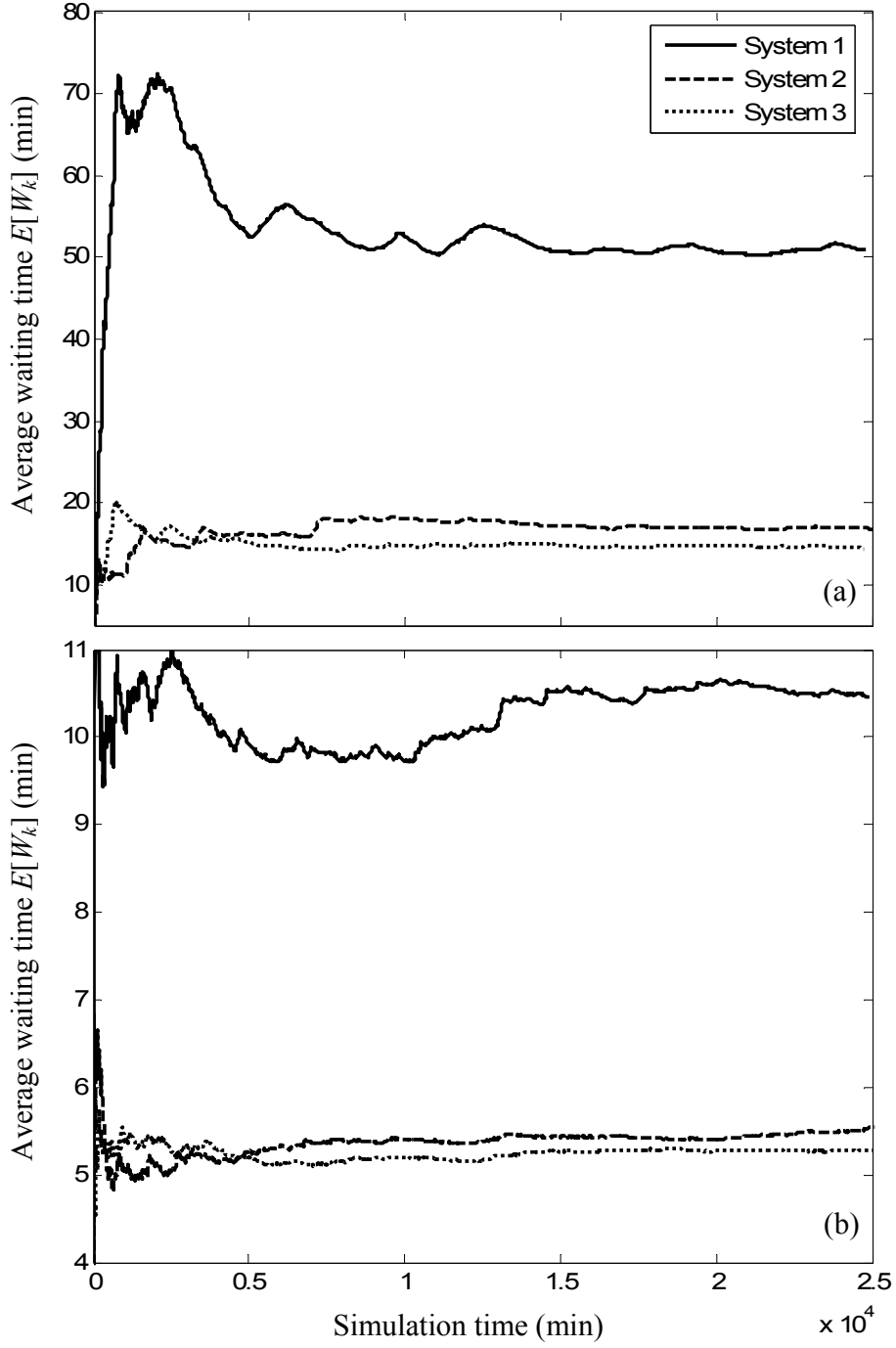


Figure 1: Mean waiting time, $m_W = E[W_k]$, for the three checkout systems: random allocation (system 1), shortest queue (system 2) and single queue (system 3), under two example conditions: (a) $N=5$, $m_{AT}=5\text{min}$, $c=1.2$, and (b) $N=5$, $m_{AT}=5\text{min}$, $c=2.0$.

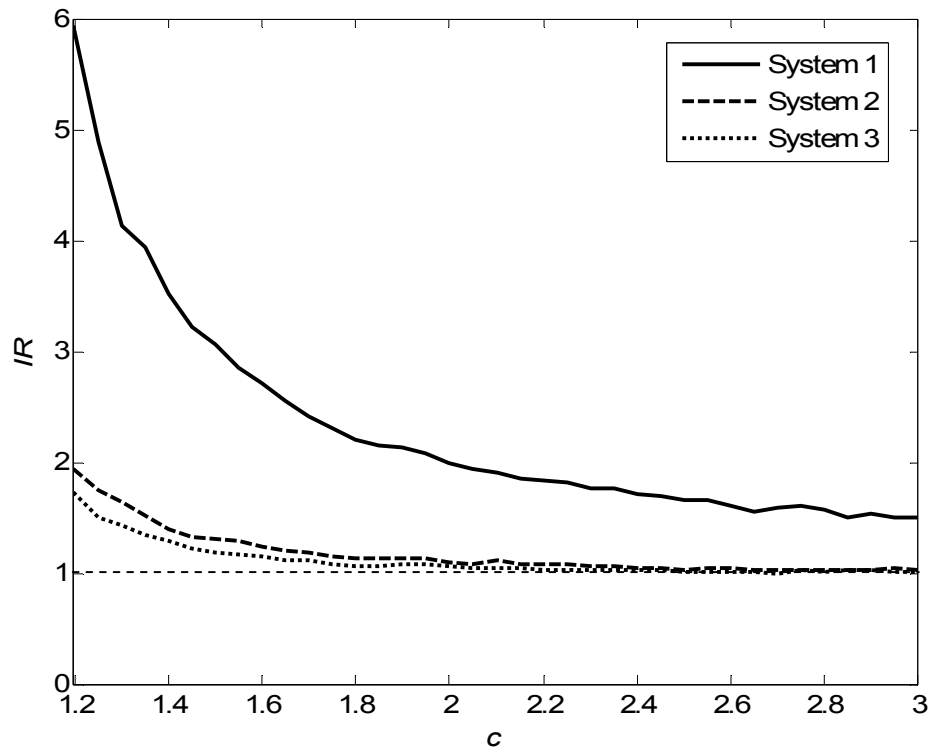


Figure 2: Inefficiency ratio, $IR = \frac{E[W_k]}{m_T}$, as a function of the system overcapacity ratio c for the three checkout systems studied.