

First Exam Solutions

1. The model of a closed system consists of six resources where jobs receive service. Four of the resources are queueing centers, where a single server provides service and jobs must wait while another job is receiving its service. Two of the centers are delay centers; jobs at these centers are only delayed by their own service demands, and never have to wait while the server is busy with some other job.

The total demands for a job (not per-visit demands) at the four queueing centers are 10, 50, 25, and 15 ms. The total demands of each job (again, not per-visit demand) at the two delay centers are 100 and 300 ms. Note that this model is NOT a central server model.

- (a) At how many jobs does the system saturate?

That this is not a central server model does not affect the definition of the saturation point on the throughput curve.

$$N^* = \frac{D + Z}{D_{\max}}$$

where $D = 10 + 50 + 25 + 15 = 100$, $D_{\max} = 50$, and $Z = 100 + 300 = 400$. Hence,

$$N^* = \frac{100 + 400}{50} = 10$$

- (b) Suppose you decided to approximate the throughput curve (a plot of system throughput vs. number of jobs) by a curve that lies exactly halfway between the upper and lower bounds for throughput. What is the maximum possible relative error, both positive and negative, for this approximation, and at how many jobs does this occur? If $app(N)$, $up(N)$, and $low(N)$ are the approximation, upper bound, and lower bound, respectively, for N jobs in the system, the positive relative error is $(up(N)/app(N)) - 1$ and the negative relative error is $(low(N)/app(N)) - 1$.

The maximum relative error, positive and negative, must occur somewhere between $N = 1$ and $N = N^*$. It is easy to verify that the relative errors will be maximized at $N = N^*$.

The upper and lower bounds at $N = N^*$ are $10/50 = 1/50$ and $10/(100 \cdot 10 + 400) = 1/140$, respectively.

The average of the upper and lower bounds at $N = N^*$ is

$$\frac{10}{500} + \frac{10}{100(10) + 400} = \frac{19}{1400}$$

The maximum positive relative error is $(1/50)/(19/1400) - 1 = 9/19$ or 47.4%.

The maximum negative relative error is $(1/140)/(19/1400) - 1 = -9/19$ or 47.4%.

- (c) What can be done to the system in order to double the number of jobs at saturation? Your options are to speed up (or slow down) one or more of the queueing centers. Speeding up a queueing center increases its cost; slowing one down decreases its cost. Discuss the ramifications of each solution that you propose.

First try eliminating the bottleneck:

$$N^* = 20 = \frac{D + Z}{D_{\max}} = \frac{D_{\max} + 50 + 400}{D_{\max}}$$

$$20D_{\max} = D_{\max} + 450$$

$$D_{\max} = \frac{450}{19} = 23.68$$

$$\text{speedup} = \frac{D_{\max}(\text{original}) - D_{\max}(\text{enhanced})}{D_{\max}(\text{original})} = \frac{50 - 23.68}{50} = 52.63\%$$

This will not work, because the bottleneck resource, when sped up this much, is no longer the bottleneck. Furthermore, speeding up the bottleneck even further, once it is no longer the bottleneck, only decreases the number of jobs at saturation.

Once the demand for the bottleneck device reaches 25 ms, speeding it up further (reducing the demand for it) is counterproductive without also speeding up the device with an original demand of 25 ms. To increase the number of jobs at saturation beyond what can be achieved by reducing the bottleneck resource demand from 50 ms to 25 ms, reduce the demand for the 25 ms resource at the same rate.

$$N^* = 20 = \frac{D + Z}{D_{\max}} = \frac{2D_{\max} + 25 + 400}{D_{\max}}$$

$$20D_{\max} = 2D_{\max} + 425$$

$$D_{\max} = \frac{425}{18} = 23.61$$

Note that there is another way to achieve an increase in the number of jobs at saturation to 20. If you reduce the bottleneck resource demand to 25 and

increase the demand for the other devices (slow them down) to 25 as well, the number of jobs at saturation is

$$N^* = \frac{D + Z}{D_{\max}} = \frac{100 + 400}{25} = 20$$

as required.

The first solution costs more, since two devices must be made faster, while in the second solution only device is faster and two are slower. However, the second "solution" also is likely to increase response time compared to the first device. At least, the envelope in which the response time curve must lie for the second solution has higher upper and lower bounds than the envelope for the first solution.

- Suppose you want to use a *truncated exponential distribution* over the range (0,1) in a stochastic simulation. A truncated exponential distribution over this range has the form $F_X(x) = k(1 - e^{-\lambda x})$ for $0 \leq x \leq 1$ and $F_X(x) = 0$ otherwise. Describe two algorithms for generating random numbers from this distribution. Be precise. You may assume that you have a $U(0,1)$ random number generator to use. Compare your algorithms on the basis of efficiency, i.e., on the amount of computation required to generate a single random number.

One of the simplest and most direct ways to do this is to simply generate random numbers from an *untruncated* exponential distribution with the same parameter λ , and reject any values that fall outside the (0,1) interval. Since $e^{-\lambda}$ of the probability mass of the untruncated distribution falls outside this interval, the probability of rejecting a value is $p = e^{-\lambda}$. The expected number of values that must be generated in order to accept a value is just the mean of a geometric distribution with this probability of failure.

$$\begin{aligned} E[\# \text{ of values before success}] &= \sum_{i=0}^{\infty} ip^i(1-p) \\ &= \frac{p}{1-p} = \frac{e^{-\lambda}}{1-e^{-\lambda}} \end{aligned}$$

The expected number of values generated per value accepted is therefore

$$\frac{e^{-\lambda}}{1-e^{-\lambda}} + 1 = \frac{1}{1-e^{-\lambda}}$$

Hence, this algorithm has an efficiency that grows with λ .

An alternative procedure is to use inversion. In order to do this, we must first evaluate k (note that in the first procedure, it was unnecessary to explicitly

calculate k). k is determined by making the probability mass of the truncated distribution sum to 1:

$$F_x(1) = k(1 - e^{-\lambda}) = 1 \Rightarrow k = \frac{1}{1 - e^{-\lambda}}$$

Let z be a value returned by a $U(0,1)$ random number generator. Then

$$z = \frac{1 - e^{-\lambda x}}{1 - e^{-\lambda}}$$

which leads to

$$x = -\frac{\ln[1 - z(1 - e^{-\lambda})]}{\lambda}$$

We cannot simplify the argument of the natural log function, as we did in the case of the exponential distribution. However, the increase in computation cost is only a single multiplication and a single subtraction. This is likely to be a good tradeoff with respect to the first procedure for many values of λ .

You could also try an acceptance-rejection algorithm, but this is almost certainly going to be less efficient computationally than either of the two procedures described above. If you must try it, a linear bounding function

$$g(x) = -\lambda x + \frac{\lambda}{1 - e^{-\lambda}}$$

would seem worth investigating.

3. (a) A performance analyst simulated a computer system a total of 10 times, each simulation run independent of all the others. She calculated and recorded the sample means for system response time from each of the 10 runs, coming up with the following data:

2,5,17,3,9,6,4,25,8,1

What is the confidence interval with a 95% confidence level for the mean response time?

$$\bar{X} = 8, s = 7.5277, 1 - \alpha = 0.95 \Rightarrow 1 - \frac{\alpha}{2} = 0.975, t_9(0.975) = 2.2622$$

The confidence interval with a 95% confidence level is

$$\left(\bar{X} - t_9(0.975) \frac{s}{\sqrt{10}}, \bar{X} + t_9(0.975) \frac{s}{\sqrt{10}} \right) = (2.6149, 13.3851)$$

- (b) The performance analyst was concerned that the width of the confidence interval was too large, and consulted one of her colleagues (who never took Elec 428). He assured her that the width of the confidence interval was large because of the two large "outliers," the values 17 and 25, and that she could reduce the width of the confidence interval simply by discarding these values. She tried this, and discovered that the width of the interval did indeed go down substantially.

Is this a valid method? Why or why not? Be specific.

The "confidence interval" according to this procedure is

(2.3960, 7.1040)

However, the t-test assumes that all of the random variables are independent and identically distributed. By discarding some of the sample means based on their values, she was introducing correlation into the sample set. The procedure is therefore not valid.

- (c) Another colleague suggested that she try "smoothing" the data, by replacing each value with the average of itself and the preceding and succeeding values, and discarding the first and last values since the first has no predecessor and the last has no successor. She tried this, and was delighted to find that it did make the interval significantly smaller.

Is this a valid method? Why or why not? Be specific.

This is also invalid, since the use of overlapping "windows" to compute the new, average values introduces correlation.

- (d) Yet another colleague (she apparently has many) advised her to take each non-overlapping pair of successive data points (sample means) and replace them by a single value which was their average. This would also "smooth" the data, as it reduced the number of data points to five. What did she determine that this did to the confidence interval width, for the same confidence level? Is this a valid procedure?

The new set of data points is 3.5, 10, 7.5, 14.5, 4.5.

$$\bar{X} = 8, s = 4.4441, t_4(0.975) = 2.7764$$

The confidence interval with a 95% confidence level is

$$\left(\bar{X} - t_4(0.975) \frac{s}{\sqrt{5}}, \bar{X} + t_4(0.975) \frac{s}{\sqrt{5}} \right) = (2.4820, 13.5180)$$

This is a valid method, because the five random variables are still independent and identically distributed.

(e) In general, what effect would you expect this procedure to have on the width of the interval?

If the number of simulation runs, were large enough (10 or more), one wouldn't expect that this would have any advantage or disadvantage over using the original data. This is because the expected width of the confidence interval is proportional to $1/\sqrt{mn}$, where m is the number of simulation runs (number of sample means) and n is the number of data points that go into the computation of each run's sample mean. Combining two sample means is in effect halving the number of runs while doubling the number of points per run, and the net effect is that the expected width of the confidence interval will remain unchanged.

However, m is only 4 for the procedure in (d), and $t_4(.975)$ is significantly larger than $t_9(.975)$. Since this isn't approximately constant (as it would be if the number of simulation runs were much larger), we should expect an increase in the width of the confidence interval.