Assignment #3 Solutions

1. The point of this problem is that if you model an arrival process as a Poisson process, as in part (a), but the true process either has interarrival times that are more or less variable than an exponential, as in parts (b) and (c), or if the interarrival times are exponential but dependent, as in parts (d) and (e), then your simulation estimate of average number in system can be very inaccurate. See the class website for a simulation program that will do the requested estimations. To generate a service time *V*, use problem 3 from Assignment #1: generate two uniforms U_1 and U_2, then set $V = 0.99(U_1 + U_2)$. Note that, in parts (d) and (e), the system is no longer, strictly speaking, a GI/G/1 queue, because interarrival times are *not* independent.

   For the questions in part (d), observe that, for each *n*,

   $$\mathrm{E}[Y_n] = \mathrm{E}[c(Z_n - Z_{n-1})] = c\mathrm{E}[Z_n] - c\mathrm{E}[Z_{n-1}] = c \cdot 0 - c \cdot 0 = 0,$$

   because each *Z* is $N(0,1)$. Similarly,

   $$\mathrm{Covar}[Y_n, Y_n] = \mathrm{Var}[Y_n] = \mathrm{E}[Y_n^2] = c^2\mathrm{E}[(Z_n - Z_{n-1})^2] = c^2\mathrm{E}[Z_n^2 - 2Z_nZ_{n-1} + Z_{n-1}^2] = 2c^2\mathrm{E}[Z_n^2] = 1,$$

   $$\mathrm{Covar}[Y_n, Y_{n-1}] = \mathrm{E}[Y_nY_{n-1}] = c^2\mathrm{E}[(Z_n - Z_{n-1})(Z_{n-1} - Z_{n-2})] = -c^2\mathrm{E}[Z_{n-1}^2] = -c^2\mathrm{Var}[Z_{n-1}] = -c^2 = -1/2,$$

   and

   $$\mathrm{Covar}[Y_n, Y_{n-j}] = \mathrm{E}[Y_nY_{n-j}] = c^2\mathrm{E}[(Z_n - Z_{n-1})(Z_{n-j} - Z_{n-j-1})] = c^2 \cdot 0 = 0 \text{ for } j \ge 2.$$

   Here we have repeatedly used the fact that $E[Z_iZ_j] = E[Z_i]E[Z_j] = 0 \cdot 0 = 0$ for $i \ne j$, by independence. Since the sum of normal random variables has a normal distribution, we have that Y_n has a $N(0,1)$ distribution for each *n*. By the result of problem 1(c) in Assignment #2, each $\Phi(Y_n)$ has a uniform $(0,1)$ distribution, and hence each $X_n = -\log(\Phi(Y_n))$ has an exp(1) distribution as in the standard inversion method. Since the Y_n's are correlated, so are the X_n's. (The autocorrelation between X_n and X_{n-1}—that is, the covariance divided by the variance—is -0.5.) For part (e), almost identical calculations show that each Y_n has a $N(0,1)$ distribution, but now the autocorrelation between Y_n and Y_{n-1} is 0.5, which is also the autocorrelation between X_n and X_{n-1}. This method of generating an autocorrelated sequence of random variables, where the random variables have a specified marginal distribution, was originally proposed using an autoregressive sequence $(Y_n : n \ge 0)$ of normal random variables, and was called the ARTA (autoregressive-to-anything) method. An autoregressive process of order *p* is defined by $Y_n = \sum_{h=1}^{p} \alpha_h Y_{n-h} + Z_n$ for $n > p$. In our example, $(Y_n : n \ge 0)$ is a "moving average" time series of order 1 rather than an autoregressive time series. The tricky part in using such methods is choosing the autocorrelation of the Y_n sequence to achieve a desired autocorrelation in the X_n sequence. For a discussion of this issue, see Biller and Nelson, "Fitting time-series input processes for simulation", *Oper. Res.* 53(3), 2005, 549-559.

   Using 10,000 replications per experiment, our simulation results were as follows:

Arrival process	Poisson	Weibull $\sigma = 0.5\mu$	Weibull $\sigma = 2\mu$	Negative-correlated exponential	Positive-correlated exponential
E[avg # in system]	11.95	7.13	21.84	7.50	15.46

To explain these results, note that $\rho = \text{E[service time]} / \text{E[interarrival time]} < 1$, so that the system is stable. If there were no variability in the system, so that the service time is less than the interarrival time with probability 1, then there would be at most one job in the system at any time. It is the variability in interarrival and service times that causes congestion. In other words, every once in a while the server gets temporarily overwhelmed by a clump of customers, either because the customers all came in at roughly the same time (short interarrival times) or a service was extra long. It then takes a while for the server to process this clump of customers, and the system is congested. For a Poisson arrival process, the standard deviation of the interarrival time equals the mean. When the interarrival times are less variable but with the same mean, as in part (b), the clumping effect is decreased, and the congestion goes down, leading to a lower average number in system. Conversely, when the interarrival times are more variable, as in part (c), the clumping effect is increased, and the congestion goes up, leading to a higher average number in system. When interarrival times are negatively correlated, as in part (d), short interarrival times tend to be followed by longer interarrival times, which reduces clumping, so congestion goes down. When interarrival times are positively correlated, a short interarrival time tends to be followed by another short interarrival time, which leads to clumping, which increases congestion.

2. Derivation of MLE estimators (problem 6.10 parts (a), (c), and (d) in Law)

(a) $U(0,b)$: The likelihood function is $L(b) = \prod_{i=1}^{n} f(X_i) = \frac{1}{b^n} \prod_{i=1}^{n} 1_{[0,b]}(X_i)$, where 1_A is an indicator function as before. Clearly, $L(b) = 0$ if $X_i > b$ for any *i*. So we must have $\hat{b} \ge \max_{1\le i\le n} X_i$. Indeed, $L(b)$ is maximized by taking $\hat{b} = \max_{1\le i\le n} X_i$. (Any larger choice of $\hat{b}$ would yield a smaller likelihood value because of the factor of b^{-n}.)

(b) $U(a,b)$: The likelihood function is $L(b) = \prod_{i=1}^{n} f(X_i) = \frac{1}{(b-a)^n} \prod_{i=1}^{n} 1_{[a,b]}(X_i)$. By reasoning similar to part (a), we must have $\hat{b} \ge \max_{1\le i\le n} X_i$ and $\hat{a} \le \min_{1\le i\le n} X_i$, and $L(b)$ is maximized by taking *a* as large as possible and *b* as small as possible to maximize the factor of $(b-a)^{-n}$. Thus the MLE estimates are $\hat{a} = \min_{1\le i\le n} X_i$ and $\hat{b} = \max_{1\le i\le n} X_i$.

(c) $N(\mu,\sigma^2)$: Since the normal density function is

$$f(x;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

the log-likelihood function is (ignoring the constant term $-n\log\sqrt{2\pi}$)

$\tilde{L}(\mu,\sigma) = -n\log\sigma - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(X_i-\mu)^2$. So we can find the MLE estimates by taking derivatives and setting them equal to 0:

$$\frac{\partial}{\partial\mu}\tilde{L}(\mu,\sigma) = \frac{1}{\sigma^2}\sum_{i=1}^{n}(X_i-\mu) = 0$$

$$\Rightarrow \sum_{i=1}^{n} X_i - n\mu = 0 \Rightarrow \mu = \frac{1}{n}\sum_{i=1}^{n} X_i \triangleq \bar{X}_n$$

$$\frac{\partial}{\partial \sigma}\tilde{L}(\mu,\sigma) = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{i=1}^{n}(X_i-\mu)^2 = 0$$

$$\Rightarrow \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i-\mu)^2 = \frac{1}{n}\sum_{i=1}^{n}(X_i-\bar{X}_n)^2$$

The last step is obtained by substituting the optimal value of μ.

3. Asymmetric triangular distributions

(a) Note that we can write $Z_1 = (b-a)U_1 + a$ and $Z_2 = (b-a)U_2 + a$, where U_1, U_2 are Uniform[0,1]. Let $x \in [a,b]$. Then

$$\begin{aligned} F_{Z_1 \vee Z_2}(x) &= P(Z_1 \vee Z_2 \le x) = P(Z_1 \le x \text{ and } Z_2 \le x) \\ &= P(Z_1 \le x)P(Z_2 \le x) = P^2(Z_1 \le x) \\ &= P^2\left(U_1 \le \frac{x-a}{b-a}\right) = \left(\frac{x-a}{b-a}\right)^2 \end{aligned}$$

The third equality uses the independence assumption and the fourth equality uses the identically-distributed assumption. Take the derivative of $F_{Z_1 \vee Z_2}(x)$ with respect to x to obtain the final answer. Thus we can generate a sample from the given distribution as

$$Y_1 = \max\big((b-a)U_1 + a, (b-a)U_2 + a\big) = a + (b-a)\max(U_1,U_2)$$

Alternately, observe that, as a consequence of the above calculations, $F_{Y_1}^{-1}(u) = a + (b-a)\sqrt{u}$, so that we can use the inversion method: $Y_1 = a + (b-a)\sqrt{U}$.

(b) In a similar manner, write $Z_1 = (c-b)U_1 + b$ and $Z_2 = (c-b)U_2 + b$. Then

$$\begin{aligned} 1 - F_{Z_1 \wedge Z_2}(x) &= P(Z_1 \wedge Z_2 > x) = P(Z_1 > x \text{ and } Z_2 > x) \\ &= P(Z_1 > x)P(Z_2 > x) = P^2(Z_1 > x) \\ &= P^2\left(U_1 > \frac{x-b}{c-b}\right) = \left(1 - \frac{x-b}{c-b}\right)^2 = \left(\frac{c-x}{c-b}\right)^2 \end{aligned}$$

so that $F_{Z_1 \wedge Z_2}(x) = 1 - \left(\frac{c-x}{c-b}\right)^2$. Again, take the derivative to get the final answer. Thus we can generate $Y_2 = b + (c-b)\min(U_1,U_2)$. Alternatively, we have $F_{Y_2}^{-1}(u) = c - (c-b)\sqrt{1-u}$, so we can use inversion to generate $Y_2 = c - (c-b)\sqrt{U}$.

4. (a) Proceeding similarly as in problem 2(c), the log-likelihood function is (ignoring constant terms)

$$\begin{aligned} \tilde{L}(\alpha,\mu_1,\mu_2) &= N_1 \log(\alpha) - \sum_{j\in A_1}(X_j-\mu_1)^2/2 \\ &+ (n-N_1)\log(1-\alpha) - \sum_{j\in A_2}(X_j-\mu_2)^2/2. \end{aligned}$$

Setting partial derivatives equal to 0 and solving, we find that $\hat{\alpha} = f_1(N_1,S_1,S_2) = N_1/n$, $\hat{\mu}_1 = f_2(N_1,S_1,S_2) = S_1/N_1$, and $\hat{\mu}_2 = f_3(N_1,S_1,S_2) = S_2/(n-N_1)$.

(b) This iterative procedure is an example of an *expectation-maximization* (EM) algorithm. Such algorithms are widely used for maximum likelihood estimation in the presence of "hidden" or "latent" data, as in this example, where the labels are hidden from the observer. Step (ii) is the E-step, where we estimate (functions of) the hidden data as conditional expectations, given the observed data and the current parameter estimates. Step (iii) is the M-step, where we compute maximum-likelihood estimates based on the observed data plus our estimates of (functions of) the missing data. Following the hint, we have

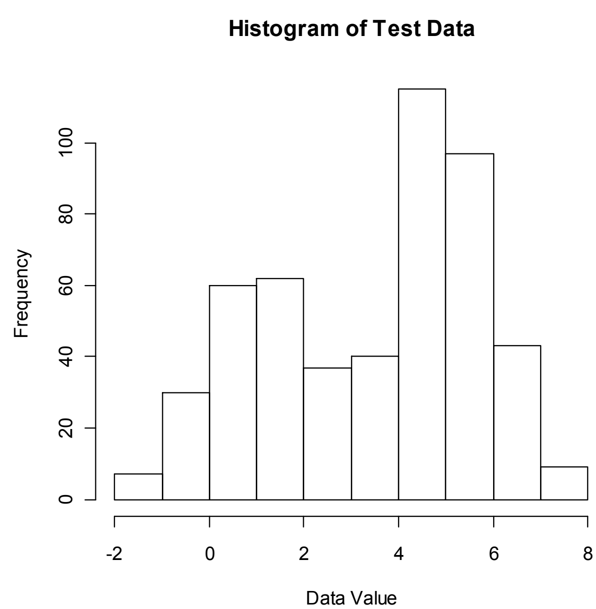
$$\begin{aligned} P(L_j = 1 \mid \hat\alpha^{(m)}, \hat\mu_1^{(m)}, \hat\mu_2^{(m)}, X_j) &= \frac{P(L_j = 1, X_j \mid \hat\alpha^{(m)}, \hat\mu_1^{(m)}, \hat\mu_2^{(m)})}{P(X_j \mid \hat\alpha^{(m)}, \hat\mu_1^{(m)}, \hat\mu_2^{(m)})} \\ &= \frac{P(L_j = 1 \mid \hat\alpha^{(m)}, \hat\mu_1^{(m)}, \hat\mu_2^{(m)}) P(X_j \mid \hat\alpha^{(m)}, \hat\mu_1^{(m)}, \hat\mu_2^{(m)}, L_j = 1)}{P(X_j \mid \hat\alpha^{(m)}, \hat\mu_1^{(m)}, \hat\mu_2^{(m)})}, \\ &= g(\hat\alpha^{(m)}, \hat\mu_1^{(m)}, \hat\mu_2^{(m)}, X_j) \end{aligned}$$

where

$$g(\alpha, \mu_1, \mu_2, x) = \frac{\alpha (2\pi)^{-1/2} e^{-(x-\mu_1)^2/2}}{\alpha (2\pi)^{-1/2} e^{-(x-\mu_1)^2/2} + (1-\alpha)(2\pi)^{-1/2} e^{-(x-\mu_2)^2/2}}.$$

So $\hat N_1^{(m)} = \sum_{j=1}^n g(\hat\alpha^{(m)}, \hat\mu_1^{(m)}, \hat\mu_2^{(m)}, X_j)$, $\hat S_1^{(m)} = \sum_{j=1}^n g(\hat\alpha^{(m)}, \hat\mu_1^{(m)}, \hat\mu_2^{(m)}, X_j) X_j$, and $\hat S_2^{(m)} = \sum_{j=1}^n \bar g(\hat\alpha^{(m)}, \hat\mu_1^{(m)}, \hat\mu_2^{(m)}, X_j) X_j$, where $\bar g(\alpha, \mu_1, \mu_2, x) = 1 - g(\alpha, \mu_1, \mu_2, x)$.

(c) As can be seen from the histogram below, the data has two modes, so a mixture model is a reasonable choice of distribution type.



The following R code will read in the data and run the EM algorithm for 10 iterations.

```
# read in data and convert to a vector
ww = read.table(file="c:\\hw3.dat"); w = ww[,1]
# intialize parameter estimates
alpha = 0.5; mu1 = 0; mu2 = 10; n = 500
# run EM algorithm for 10 iterations
# note that dnorm is the normal density function
for (i in c(1:10)) {
  n1 = alpha*dnorm(w,mean=mu1)/(alpha*dnorm(w,mean=mu1)
         +(1-alpha)*dnorm(w,mean=mu2))
  s1 = sum(w * n1); s2 = sum(w * (1-n1))
  n1 = sum(n1)
  alpha = n1/n;   mu1 = s1/n1;   mu2 = s2/(n-n1)
  print(c(i, alpha, mu1, mu2))
}
```

The output is given below. Note that values of alpha = 0.4 mu1 = 1 mu2 = 5 were used to generate the data. Also note that the accuracy of the method is limited by the number of data points used.)

Iter	alpha	mu1	mu2
1	0.7360416	2.2503899	5.8331326
2	0.6056252	1.7521410	5.4134904
3	0.5146083	1.3215125	5.1834916
4	0.4715885	1.1064114	5.0610442
5	0.4544805	1.0243922	5.0053551
6	0.4479828	0.9945666	4.9827001
7	0.4455399	0.9835934	4.9739469
8	0.4446240	0.9795148	4.9706312
9	0.4442809	0.9779918	4.9693842
10	0.4441523	0.9774222	4.9689166