KTH Mathematics

Suggested solutions for the exam in SF2863 Systems Engineering.
June 3, 2013 14.00–19.00

Examiner: Per Enqvist, phone: 790 62 98

---

1. We can think of the F&H office as a Jackson network. Heathcliff is a $M|M|1$ queue with arrival intensity λ_H and service intensity $\mu_H = 1/10$. Frasse's service is a $M|M|1$ queue with arrival intensity λ_F and service intensity $\mu_F = 1/8$.

(a) Let $\lambda_R = 1/20$ be the arrival intensity from the outside, i.e. the intensity in to the reception. The traffic balance equations are $\lambda_F = 0.6\lambda_H + 0.1\lambda_F$ and $\lambda_H = \lambda_R + 0.2\lambda_H + 0.1\lambda_F$ which yields, $\lambda_F = 1/22$ and $\lambda_H = 3/44$. We can now check that the low traffic requirements are satisfied, i.e., that $\lambda_H = 3/44 < \mu_H = 1/10$ and $\lambda_F = 1/22 < \mu_F = 1/8$. So the queues converge to a stationary state.
$L_F = \rho_F/(1-\rho_F) = 4/7$, and $L_H = \rho_H/(1-\rho_H) = 15/7$.
The average total number of patents in the systems is then $L = L_H + L_F = 19/7$

(b) The probability is $0.2\lambda_H/\lambda_R = 0.2\frac{3/44}{1/20} = 3/11$.

(c) Let V_F and V_H be the average time it takes from a call arrives to one of the service stations until it leaves it, i.e., $V_F = L_F/\lambda_F = 88/7$, and $V_H = L_H/\lambda_H = 220/7$.
Letting W_H be the average time from a patent arrives to Heathcliff until it exits the system, W_F be the average time from a patent arrives to Frasse until it exits the system, then

$$W_F = V_F + 1/10W_H + 1/10W_F$$
$$W_H = V_H + 0.2W_H + 0.6W_F$$

then $W_F = 20$ and $W_H = 54.3$.
So the average time for passing through is 54.3 hours.

(d) If a patent is interrupted during a readthrough no time is lost due to the memory less property of the exponential distribution. If the patent has not be read through at some particular time the probability that Heathcliff will finish reading through within the next t time units is the same as if he started from the beginning. Therefore, when the prioritized patent has been finished, no time was wasted on the patent that was currently being read and therefore the average queue length is not changed.

2. (a) This can be solved using the economic order quantity model Let $K = 150$, $d = 100$, $h = 1/100$ and $c = 0$.
Optimal D is $D = \sqrt{\frac{2dK}{h}} = 100\sqrt{300}$
The time between to visits to the loan shark is $Q/d = \sqrt{300}$.

(b) This can be modelled as an EOQ model with planned shortage. The SMS-loan is taken at the moment that the money borrowed from the loan shark becomes zero.

Now let $p = 1/10$ denote the shortage cost, S the amount from the SMS-loan. The total cost per cycle is

$$T_c = K + \frac{hD^2}{2d} + \frac{pS^2}{2d}.$$

Let $Q = D + S$, then the length of a cycle is Q/d and the total cost per time unit is

$$T = \frac{Kd}{Q} + \frac{hD^2}{2Q} + \frac{p(Q-D)^2}{2Q}.$$

The optimum is obtained at the stationary point, where derivatives of T w.r.t. Q and D are zero.

$$-\frac{Kd}{Q^2} - \frac{hD^2}{2Q^2} - \frac{p(Q-D)^2}{2Q^2} + \frac{p(Q-D)}{Q} = 0$$

$$\frac{hD}{Q} - \frac{p(Q-D)}{Q} = 0.$$

From which

$$D^* = \sqrt{\frac{2dK}{h}}\sqrt{\frac{p}{p+h}} \approx 5450$$

$$Q^* = \sqrt{\frac{2dK}{h}}\sqrt{\frac{p+h}{p}} \approx 5505$$

3. (a) Let s_i denote the number of servers at facility i.

Let $f(s_1, s_2, s_3, s_4) = \sum_{i=1}^{N} W_q(s_i)$ and $g(s_1, s_2, s_3, s_4) = \sum_{i=1}^{N} s_i$.

Then f and g are separable functions, f is decreasing and g is increasing. Furthermore, $\Delta^2 g = 0$ since the function is linear, so it is integer-convex.

$\Delta W_q(s)$	$\rho = 0.2$	$\rho = 0.5$	$\rho = 0.7$	$\rho = 0.8$
$s = 1$	-4.1116	-14.7597	-36.07011	-65.6154
$s = 2$	-0.1904	-1.1206	-2.1852	-2.8944
$s = 3$	-0.0080	-0.1094	-0.2759	-0.3962

Since the values are increasing in each column the function f is integer-convex for the tabulated values.

We can then apply the Marginal Allocation algorithm.

Note that $\Delta f_i(s) = \Delta W_q(s_i)$ and $\Delta g_i(x) = 1$, so the quotients $-\Delta f_i(s)/\Delta g_i(s) = \Delta W_q(s_i)$ are given in the table above.

The efficient allocations are therefore,

$S^{(0)} = (s_1 = 1, s_2 = 1, s_3 = 1, s_4 = 1)$, $\mathrm{f}(\mathrm{s}^{(0)}) = 127.8, g(s^{(0)}) = 4$

$\mathrm{S}^{(1)} = (s_1 = 1, s_2 = 1, s_3 = 1, s_4 = 2)$, $\mathrm{f}(\mathrm{s}^{(1)}) = 62.2, g(s^{(1)}) = 5$

$S^{(2)} = (s_1 = 1, s_2 = 1, s_3 = 2, s_4 = 2)$, $f(s^{(2)}) = 26.2, g(s^{(2)}) = 6$
$S^{(3)} = (s_1 = 1, s_2 = 2, s_3 = 2, s_4 = 2)$, $f(s^{(3)}) = 11.4, g(s^{(3)}) = 7$
$S^{(4)} = (s_1 = 2, s_2 = 2, s_3 = 2, s_4 = 2)$, $f(s^{(4)}) = 7.3, g(s^{(4)}) = 8$

(b) Let $\Lambda_0 = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 0.11$. The mean wailting time is given by $\frac{\lambda_1}{\Lambda_0} W_q(s_1) + \frac{\lambda_2}{\Lambda_0} W_q(s_2) + \frac{\lambda_3}{\Lambda_0} W_q(s_3) + \frac{\lambda_4}{\Lambda_0} W_q(s_4)$
So we change the objective function to $f(s_1, s_2, s_3, s_4) = \sum_{i=1}^{N} \frac{\lambda_1}{\Lambda_0} W_q(s_1)$
It is still integer-convex and decreasing since the intensities λ_i are positive, and the only thing that changes is that column i in the table is multiplied with λ_i/Λ_0.
This will change the last allocation to be $S^{(4)} = (s_1 = 1, s_2 = 2, s_3 = 2, s_4 = 3)$.

4. (a) We need to keep track how many cakes we have at beginning of day k, define the state

$$s_k = \begin{cases} 0 & \text{if number of cakes at beginning of day } k \text{ is } 0 \\ 1 & \text{if number of cakes at beginning of day } k \text{ is } 1 \\ 2 & \text{if number of cakes at beginning of day } k \text{ is } 2 \end{cases}$$

Define the decisions

$$x_k = \begin{cases} 0 & \text{if Frasse orders 0 cakes day } k \\ 1 & \text{if Frasse orders 1 cakes day } k \\ 2 & \text{if Frasse orders 2 cakes day } k \end{cases}$$

The state update equation is $s_{k+1} = s_k - d_k + x_k$, where $0 \le s_k \le 2$, so larger values than 2 are counted as 2, and $d_k = 0$ if $s_k + x_k = 0$ and otherwise

$$d_k = \begin{cases} 1 & \text{with probability } 0.2 \\ 0 & \text{with probability } 0.8 \end{cases}$$

The transition probabilities are
$p_{ij}(x) =$ *the probability of jumping from state i to j if we make decision x.*

$$P(x = 0) = \begin{bmatrix} 1 & 0 & 0 \\ 0.2 & 0.8 & 0 \\ 0 & 0.2 & 0.8 \end{bmatrix}$$

$$P(x = 1) = \begin{bmatrix} 0.2 & 0.8 & 0 \\ 0 & 0.2 & 0.8 \\ 0 & 0 & 1 \end{bmatrix}$$

$$P(x = 2) = \begin{bmatrix} 0 & 0.2 & 0.8 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Let $q_{ij}(x) =$ expected cost incurred when the state is in state i decision x is made and the system evolves to state j.

$$Q(x = 0) = \begin{bmatrix} 0 & - & - \\ -100 & 0 & - \\ - & -100 & 0 \end{bmatrix}$$

$$Q(x=1) = \begin{bmatrix} -50 & 50 & - \\ - & -50 & 50 \\ - & - & 30 \end{bmatrix}$$

$$Q(x=2) = \begin{bmatrix} - & -20 & 80 \\ - & - & 60 \\ - & - & 60 \end{bmatrix}$$

Then the expected "cost" of making decision x_k at state s_k is $C_{s_k,x_k} = \sum_{j=0}^{3} q_{s_k,j} p_{s_k,j}(x_k)$.

$$C_{00} = q_{00}(0)p_{00}(0) + q_{01}(0)p_{01}(0) + q_{02}(0)p_{02}(0) = 0$$

$$C_{01} = q_{00}(1)p_{00}(1) + q_{01}(1)p_{01}(1) + q_{02}(1)p_{02}(1) = 30$$

$$C_{02} = q_{00}(2)p_{00}(2) + q_{01}(2)p_{01}(2) + q_{02}(2)p_{02}(2) = 60$$

$$C_{10} = q_{10}(0)p_{10}(0) + q_{11}(0)p_{11}(0) + q_{12}(0)p_{12}(0) = -20$$

$$C_{11} = q_{10}(1)p_{10}(1) + q_{11}(1)p_{11}(1) + q_{12}(1)p_{12}(1) = 30$$

$$C_{12} = q_{10}(2)p_{10}(2) + q_{11}(2)p_{11}(2) + q_{12}(2)p_{12}(2) = 60$$

$$C_{20} = q_{20}(0)p_{20}(0) + q_{21}(0)p_{21}(0) + q_{22}(0)p_{22}(0) = -20$$

$$C_{21} = q_{20}(1)p_{20}(1) + q_{21}(1)p_{21}(1) + q_{22}(1)p_{22}(1) = 30$$

$$C_{22} = q_{20}(2)p_{20}(2) + q_{21}(2)p_{21}(2) + q_{22}(2)p_{22}(2) = 60$$

Let $V_k(s)$ denote the optimal cost from day k if the state at that day is s.

$$V_3(0) = \min\{C_{00}, C_{01}, C_{02}\} = C_{00} = 0$$

$$V_3(1) = \min\{C_{10}, C_{11}, C_{12}\} = C_{10} = -20$$

$$V_3(2) = \min\{C_{20}, C_{21}, C_{22}\} = C_{20} = -20$$

Recursion

$$V_k(s) = \min_x \left\{ C_{sx} + \sum_{j=0}^{2} p_{sj}(x) V_{k+1}(j) \right\}$$

$$V_2(0) = \min\{0 + 1 * 0, 30 + 0.2 * 0 + 0.8 * (-20), 60 + 0.2 * (-20) + 0.8 * (-20)\} = 0$$

$$V_2(1) = \min\{-20 + 0.2 * 0 + 0.8 * (-20), 30 + 0.2 * (-20) + 0.8 * (-20), 60 + 1 * (-20)\} = -36$$

$$V_2(2) = \min\{-20 + 0.2 * (-20) + 0.8 * (-20), 30 + 1 * (-20), 60 + 1 * (-20)\} = -40$$

$$V_1(0) = \min\{0 + 1 * 0, 30 + 0.2 * 0 + 0.8 * (-36), 60 + 0.2 * (-36) + 0.8 * (-40)\} = 0$$

So it is optimal not to buy any cakes.

(b) Starting policy:
If $s_k = 0$, make decision $x_k = 0$, for the expected cost 0.
If $s_k = 1$, make decision $x_k = 0$, for the expected cost -20.
If $s_k = 2$, make decision $x_k = 0$, for the expected cost -20.
Use the policy iteration algorithm. Let $v_2 = 0$, then the value determination equations

$$g + v_0 = 0 + 1v_0 + 0v_1 + 0v_2$$

$$g + v_1 = -20 + 0.2v_0 + 0.8v_1 + 0v_2$$

$$g + v_2 = -20 + 0v_0 + 0.2v_1 + 0.8v_2$$

gives $g = 0$, $v_0 = 200$, $v_1 = 100$, and $v_2 = 0$.
To find out if it is optimal we do one step of the policy iteration.
For $i = 0$

$$\min_{k=0,1,2} \{C_{0k} + (p_{00}(k)v_0 + p_{01}(k)v_1 + p_{02}(k)v_2)\} =$$

$$= \min\{C_{00}+(p_{00}(0)v_0+p_{01}(0)v_1+p_{02}(0)v_2), C_{01}+(p_{00}(1)v_0+p_{01}(1)v_1+p_{02}(1)v_2)C_{02}+(p_{00}(2)v_0+p_{01}(2)v_1+p_{02}(2)$$

$$= \min\{\underbrace{0 + (1 * 200)}_{200}, \underbrace{-20 + (0.2 * 200 + 0.8 * 100)}_{100}, \underbrace{-20 + (0.2 * 100)}_{0}, \} = 200 \text{ for } k = 2.$$

For $i = 1$

$$\min_{k=0,1,2} \{C_{1k} + (p_{10}(k)v_0 + p_{11}(k)v_1 + p_{12}(k)v_2)\} =$$

$$= \max\{C_{10}+(p_{10}(0)v_0+p_{11}(0)v_1+p_{12}(0)v_2), C_{11}+(p_{10}(1)v_0+p_{11}(1)v_1+p_{12}(1)v_2), C_{12}+(p_{10}(2)v_0+p_{11}(2)v_1+p_{12}($$

$$= \max\{\underbrace{-20 + (0.2 * 200 + 0.8 * 100)}_{100}, \underbrace{30 + (0.2 * 100)}_{50}, \underbrace{60 + (1 * 0)}_{60}, \} = 50 \text{ for } k = 1.$$

For $i = 2$

$$\max_{k=0,1,2} \{C_{2k} + (p_{20}(k)v_0 + p_{21}(k)v_1 + p_{22}(k)v_2)\} =$$

$$= \max\{C_{20}+(p_{20}(0)v_0+p_{21}(0)v_1+p_{22}(0)v_2), C_{21}+(p_{20}(1)v_0+p_{21}(1)v_1+p_{22}(1)v_2), C_{22}+(p_{20}(2)v_0+p_{21}(2)v_1+p_{22}($$

$$= \max\{\underbrace{-20 + (0.2 * 100)}_{0}, \underbrace{30 + 1 * 0}_{30}, \underbrace{60 + 1 * 0}_{60}, \} = 0 \text{ for } k = 0.$$

The starting policy is not optimal, it is better to use the updated policy
Updated policy:
If $s_k = 0$, make decision $x_k = 2$, for the expected cost 60.
If $s_k = 1$, make decision $x_k = 1$, for the expected cost 30.
If $s_k = 2$, make decision $x_k = 0$, for the expected cost -20.
It was enough to get this far. For completeness the rest of the iterations are described below. Notice how the mean expected cost per time step g decreases at each iteration.

Use the policy iteration algorithm. Let $v_2 = 0$, then the value determination equations

$$g + v_0 = 60 + 0v_0 + 0.2v_1 + 0.8v_2$$

$$g + v_1 = 30 + 0v_0 + 0.2v_1 + 0.8v_2$$

$$g + v_2 = -20 + 0v_0 + 0.2v_1 + 0.8v_2$$

gives $g = -10$, $v_0 = 80$, $v_1 = 50$, and $v_2 = 0$.
To find out if it is optimal we do one step of the policy iteration.
For $i = 0$

$$\min_{k=0,1,2} \{C_{0k} + (p_{00}(k)v_0 + p_{01}(k)v_1 + p_{02}(k)v_2)\} =$$

$$= \min\{C_{00}+(p_{00}(0)v_0+p_{01}(0)v_1+p_{02}(0)v_2), C_{01}+(p_{00}(1)v_0+p_{01}(1)v_1+p_{02}(1)v_2), C_{02}+(p_{00}(2)v_0+p_{01}(2)v_1+p_{02}(2$$

$$= \min\{\underbrace{0 + (1 * 80)}_{80}, \underbrace{-20 + (0.2 * 80 + 0.8 * 50)}_{11}, \underbrace{-20 + (0.2 * 50)}_{5}, \} = 5 \text{ for } k = 2.$$

For $i = 1$

$$\min_{k=0,1,2} \{C_{1k} + (p_{10}(k)v_0 + p_{11}(k)v_1 + p_{12}(k)v_2)\} =$$

$$= \max\{C_{10}+(p_{10}(0)v_0+p_{11}(0)v_1+p_{12}(0)v_2), C_{11}+(p_{10}(1)v_0+p_{11}(1)v_1+p_{12}(1)v_2), C_{12}+(p_{10}(2)v_0+p_{11}(2)v_1+p_{12}($$

$$= \max\{\underbrace{-20 + (0.2 * 80 + 0.8 * 50)}_{36}, \underbrace{30 + (0.2 * 50)}_{40}, \underbrace{60 + (1 * 0)}_{60}, \} = 36 \text{ for } k = 0.$$

For $i = 2$

$$\max_{k=0,1,2} \{C_{2k} + (p_{20}(k)v_0 + p_{21}(k)v_1 + p_{22}(k)v_2)\} =$$

$$= \max\{C_{20}+(p_{20}(0)v_0+p_{21}(0)v_1+p_{22}(0)v_2), C_{21}+(p_{20}(1)v_0+p_{21}(1)v_1+p_{22}(1)v_2), C_{22}+(p_{20}(2)v_0+p_{21}(2)v_1+p_{22}($$

$$= \max\{\underbrace{-20 + (0.2 * 80)}_{-4}, \underbrace{30 + 1 * 0}_{30}, \underbrace{60 + 1 * 0}_{60}, \} = -4 \text{ for } k = 0.$$

Updated policy:
If $s_k = 0$, make decision $x_k = 2$, for the expected cost 60.
If $s_k = 1$, make decision $x_k = 0$, for the expected cost -20.
If $s_k = 2$, make decision $x_k = 0$, for the expected cost -20.
Use the policy iteration algorithm. Let $v_2 = 0$, then the value determination equations

$$g + v_0 = 60 + 0v_0 + 0.2v_1 + 0.8v_2$$

$$g + v_1 = 30 + 0.2v_0 + 0.8v_1 + 0v_2$$

$$g + v_2 = -20 + 0v_0 + 0.2v_1 + 0.8v_2$$

gives $g = -12$, $v_0 = 80$, $v_1 = 40$, and $v_2 = 0$.
To find out if it is optimal we do one step of the policy iteration.
For $i = 0$

$$\min_{k=0,1,2} \{C_{0k} + (p_{00}(k)v_0 + p_{01}(k)v_1 + p_{02}(k)v_2)\} =$$

$$= \min\{C_{00}+(p_{00}(0)v_0+p_{01}(0)v_1+p_{02}(0)v_2), C_{01}+(p_{00}(1)v_0+p_{01}(1)v_1+p_{02}(1)v_2)C_{02}+(p_{00}(2)v_0+p_{01}(2)v_1+p_{02}(2)$$

$$= \min\{\underbrace{0+(1*80)}_{80}, \underbrace{-20+(0.2*80+0.8*40)}_{28}, \underbrace{-20+(0.2*40)}_{-12}, \} = -12 \text{ for } k=2.$$

For $i = 1$

$$\min_{k=0,1,2}\{C_{1k}+(p_{10}(k)v_0+p_{11}(k)v_1+p_{12}(k)v_2)\} =$$

$$= \max\{C_{10}+(p_{10}(0)v_0+p_{11}(0)v_1+p_{12}(0)v_2), C_{11}+(p_{10}(1)v_0+p_{11}(1)v_1+p_{12}(1)v_2), C_{12}+(p_{10}(2)v_0+p_{11}(2)v_1+p_{12}($$

$$= \max\{\underbrace{-20+(0.2*80+0.8*40)}_{28}, \underbrace{30+(0.2*40)}_{38}, \underbrace{60+(1*0)}_{60}, \} = 28 \text{ for } k=0.$$

For $i = 2$

$$\max_{k=0,1,2}\{C_{2k}+(p_{20}(k)v_0+p_{21}(k)v_1+p_{22}(k)v_2)\} =$$

$$= \max\{C_{20}+(p_{20}(0)v_0+p_{21}(0)v_1+p_{22}(0)v_2), C_{21}+(p_{20}(1)v_0+p_{21}(1)v_1+p_{22}(1)v_2), C_{22}+(p_{20}(2)v_0+p_{21}(2)v_1+p_{22}($$

$$= \max\{\underbrace{-20+(0.2*80)}_{-4}, \underbrace{30+1*0}_{30}, \underbrace{60+1*0}_{60}, \} = -4 \text{ for } k=0.$$

So the updated policy is optimal.

(c) The selling value of the cakes are decreasing over time, it is not the income in the future that is discounted back to todays value, therefore it is not a discounted cost criterion problem.

If we want to model the problem as a Markov chain we have to have a state that holds all of the information necessary to calculate the value. Therefore, the state has to have one variable for each cake that tells us if they have bought that cake and how old it is. The size of the problem then becomes much larger. The transition matrix will be very large but rather sparse.