

1. (25 points)

- a. X_t = état de la machine à la fin de la journée t

$$X_t = \begin{cases} 0, & \text{si la machine est fonctionnelle à la fin de la journée } t \\ 1, & \text{si la machine est défectueuse à la fin de la journée } t \end{cases}$$

- b. Ce processus stochastique est une chaîne de Markov, car l'état de la machine aujourd'hui dépend uniquement de son état la veille, et non de son état avant. Sa matrice de transition est :

$$\begin{pmatrix} 0,9 & 0,1 \\ 1 & 0 \end{pmatrix}$$

- c. X_t = état des deux machines, principale (P) et de réserve (R), à la fin de la journée t

$$X_t = \begin{cases} 0, & \text{si les deux machines sont fonctionnelles à la fin de la journée } t \\ 1, & \text{si la machine P est défectueuse et la machine R fonctionnelle à la fin de la journée } t \\ 2, & \text{si la machine P est fonctionnelle et la machine R défectueuse à la fin de la journée } t \end{cases}$$

Note : il est impossible que les deux machines soient défectueuses à la fin d'une journée, car si R devient défectueuse, cela se produit nécessairement un jour où P est en réparation, donc fonctionnelle à la fin de la journée

- d. Ce processus stochastique est une chaîne de Markov, car l'état de la machine aujourd'hui dépend uniquement de son état la veille, et non de son état avant. Sa matrice de transition est :

$$\begin{pmatrix} 0,9 & 0,1 & 0 \\ 0,9 & 0 & 0,1 \\ 0,9 & 0,1 & 0 \end{pmatrix}$$

2. (25 points)

a. $L = \frac{\lambda}{\mu - \lambda} = 1$

$$W = \frac{1}{\mu - \lambda} = 1/5$$

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)} = 1/2$$

$$W_q = \frac{\lambda}{\mu(\mu - \lambda)} = 1/10$$

- b. $P(\text{arrivée en moins de 6 minutes}) = 1 - e^{5/10} = 0,393$
 $P(\text{service en moins de 6 minutes}) = 1 - e^{10/10} = 0,632$

Arrivée : nombre aléatoire $\leq 0,392$

Départ : nombre aléatoire $\leq 0,631$

t	N(t)	Nombre(a)	Nombre(d)	Arrivée	Départ
0	0				
6	1	0,096	-	oui	-
12	1	0,569	0,665	non	non
18	1	0,764	0,842	non	non
24	0	0,492	0,224	non	oui
30	0	0,950	-	non	-
36	0	0,610	-	non	-
42	1	0,145	-	oui	-
48	0	0,484	0,552	non	oui
54	1	0,350	-	oui	-

Deux clients sont entrés dans le système : le premier y est demeuré 18 minutes (0,3 heures), le deuxième y est demeuré 6 minutes (0,1 heures). On a donc

$$W = (0,3 + 0,1)/2 = 0,2$$

Chacun de ces deux clients a été directement servi : leur temps passé dans la file est donc 0, ce qui implique $W_q = 0$.

Le nombre moyen de clients dans le système peut être estimé à $(1+1+1+0+0+0+1+0+1)/9 = 5/9$.

Le nombre moyen de clients dans la file peut être estimé à 0 (aucun client n'est demeuré en file).

- c. Voici le tableau présentant les itérations (plusieurs solutions sont possibles) :

Temps	Clients (file)	Service?	Proch. arrivée	Proch. départ
0	1	oui	0,042	0,041
0,041	0	non	0,042	-
0,042	0	oui	0,963	0,118
0,118	0	non	0,963	-
0,963	0	oui	1,136	0,964
0,964	0	non	1,136	-
1,136	0	oui	1,201	1,159
1,159	0	non	1,201	-
1,201	0	oui	1,324	1,336
1,324	1	oui	1,507	1,336
1,336	0	oui	1,507	1,340
1,340	0	non	1,507	-
1,507	0	oui	1,810	1,530
1,530	0	non	1,810	-
1,810	0	oui	1,875	1,940

1,875	1	oui	1,876	1,940
1,876	2	oui	2,087	1,940
1,940	1	oui	2,087	2,108
2,087	2	oui	2,131	2,108
2,108	1	oui	2,131	2,268
2,131	2	oui	2,262	2,268
2,262	3	oui	2,738	2,268
2,268	2	oui	2,738	2,300
2,300	1	oui	2,738	2,352
2,352	0	oui	2,738	2,382
2,382	0	non	2,738	-
2,738	0	oui	2,920	2,827
2,827	0	non	2,920	-
2,920	0	oui	2,995	3,852
2,995	1	oui	3,289	3,852
3,289	2	oui	3,517	3,852
3,517	3	oui	3,571	3,852
3,571	4	oui	3,620	3,852
3,620	5	oui	3,681	3,852
3,681	6	oui	3,906	3,852
3,852	5	oui	3,906	3,895
3,895	4	oui	3,906	4,027
3,906	5	oui	4,134	4,027
4,027	4	oui	4,134	4,159
4,134	5	oui	4,161	4,159
4,159	4	oui	4,161	4,187
4,161	5	oui	4,652	4,187
4,187	4	oui	4,652	4,238
4,238	3	oui	4,652	4,362

Les résultats finaux sont :

Results from the simulation run are as follows:

Average waiting time (excluding service)	0.19359855
Average waiting time (including service)	0.30770808
Average number waiting to begin service	1.166406
Average number waiting or in service	1.7048542

Continue Simulation

d. Voir le fichier *Queueing Simulator.xls* (plusieurs solutions sont possibles) :

Queueing Simulator

Données		Résultats		
		Estimé	Intervalle de confiance 95%	
			Valeur inférieure	Valeur supérieure
Nombre de serveurs =	1	L =	1,0176178	0,924695993 1,110539606
Temps interarrivée		L _q =	0,51899932	0,435985367 0,602013275
Loi =	Exponential	W =	0,20468411	0,187680035 0,221688189
Espérance =	0,2	W _q =	0,10439176	0,088497477 0,120286046
Temps de service		P ₀ =	0,50138152	0,487114273 0,51564877
Loi =	Exponential	P ₁ =	0,25139558	0,244202927 0,258588225
Espérance =	0,1	P ₂ =	0,12327761	0,117377792 0,129177429
		P ₃ =	0,06076612	0,055624158 0,065908089
		P ₄ =	0,02844012	0,02469865 0,032181586
Longueur de la simulation		P ₅ =	0,01636786	0,012907523 0,019828193
Nombre d'arrivées =	10 000	P ₆ =	0,00918318	0,006044617 0,012321737
		P ₇ =	0,00363525	0,002194291 0,005076203
		P ₈ =	0,00176598	0,000781557 0,002750408
		P ₉ =	0,00085645	0,000299473 0,001413424
		P ₁₀ =	0,00086804	-1,54083E-05 0,001751489

3. (25 points)

Étape n = composant n , $n = 1, 2, 3, 4$

x_n = nombre d'unités parallèles installées pour le composant n

s_n = quantité de la ressource non encore affectée = nombre de dollars non encore dépensés (en centaines de dollars)

$p_n(x_n)$ = probabilité que le composant fonctionne si on installe x_n unités parallèles

$c_n(x_n)$ = coût d'installation de x_n unités parallèles pour le composant n

$$f_n^*(s_n) = \max\{p_n(x_n) \cdot f_{n+1}^*(s_n - c_n(x_n)) \mid x_n = 0, 1, \dots, \min(3, M(s_n))\},$$

$$M(s_n) = \max\{z \mid c_n(z) \leq s_n, z \text{ entier}\}$$

Tableau pour $n = 4$

s_4	$f_4^*(s_4)$	x_4^*
0 ou 1	0	0
2	0,5	1
3	0,7	2
4,5,...,10	0,9	3

Tableau pour $n = 3$

s_3	$x_3 = 0$	$x_3 = 1$	$x_3 = 2$	$x_3 = 3$	$f_3^*(s_3)$	x_3^*
0	0	-	-	-	0	0
1 ou 2	0	0	-	-	0	0 ou 1
3	0	0,35	0	-	0,35	1
4	0	0,49	0	0	0,49	1
5	0	0,63	0,4	0	0,63	1
6	0	0,63	0,56	0,45	0,63	1
7	0	0,63	0,72	0,63	0,72	2
8,9,10	0	0,63	0,72	0,81	0,81	3

Tableau pour $n = 2$

s_2	$x_2 = 0$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$f_2^*(s_2)$	x_2^*
0 ou 1	0	-	-	-	0	0
2 ou 3	0	0	-	-	0	0 ou 1
4	0	0	0	-	0	1,2,3
5	0	0,21	0	0	0,21	1
6	0	0,294	0	0	0,294	1
7	0	0,378	0,245	0	0,378	1
8	0	0,378	0,343	0,28	0,378	1
9	0	0,432	0,441	0,392	0,441	2
10	0	0,486	0,441	0,504	0,504	3

Tableau pour $n = 1$

s_l	$x_l = 0$	$x_l = 1$	$x_l = 2$	$x_l = 3$	$f_1^*(s_l)$	x_1^*
10	0	0,22	0,227	0,302	0,302	3

La valeur optimale est donc 0,302. La solution optimale est $x_1^* = 3$ (d'où $s_2 = 7$), $x_2^* = 1$ (d'où $s_3 = 5$), $x_3^* = 1$ (d'où $s_4 = 4$), $x_4^* = 3$.

4. (25 points)

a. f s'exprime comme la somme de deux fonctions :

$f_1(x_1) = 8x_1 - x_1^2$ et $f_2(x_2, x_3) = 2x_2 + x_3$. Cette dernière est linéaire, donc concave, alors que la première est aussi concave, puisque sa dérivée seconde est $-2 < 0$. f étant la somme de deux fonctions concaves est aussi concave.

Par ailleurs, le domaine réalisable est convexe, puisqu'il est l'intersection de contraintes linéaires.

Puisque l'objectif est concave et que le domaine réalisable est convexe, il s'agit d'un modèle de programmation convexe.

b. Les conditions de KKT pour ce modèle sont (comme il a été vu au cours, on peut éliminer les multiplicateurs associés aux contraintes de non négativité, car on peut fixer ces derniers à 0) :

$$8 - 2x_1 - u \leq 0$$

$$2 - 3u \leq 0$$

$$1 - 2u \leq 0$$

$$x_1(8 - 2x_1 - u) = 0$$

$$x_2(2 - 3u) = 0$$

$$x_3(1 - 2u) = 0$$

$$x_1 + 3x_2 + 2x_3 - 12 \leq 0$$

$$u(x_1 + 3x_2 + 2x_3 - 12) = 0$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

$$u \geq 0$$

c. En raison de la contrainte $2 - 3u \leq 0$, on doit nécessairement avoir $u \geq \frac{2}{3}$. À cause de la contrainte $x_3(1 - 2u) = 0$, cela implique que $x_3 = 0$. On ne peut pas avoir $x_2 = 0$, car alors $x_1 = 12$ et on ne peut satisfaire la contrainte $x_1(8 - 2x_1 - u) = 0$. On doit donc $x_2 > 0$ et alors $u = \frac{2}{3}$. On doit donc trouver x_1 et x_2 qui satisfont les équations $8 - 2x_1 - \frac{2}{3} = 0$ et $x_1 + 3x_2 - 12 = 0$. D'où $x_1 = \frac{11}{3}$ et $x_2 = \frac{25}{9}$. Donc, $(\frac{11}{3}, \frac{25}{9}, 0)$ peut être une solution optimale. Puisqu'il s'agit des conditions KKT pour un modèle de programmation convexe (pour lequel toute solution optimale locale est aussi globale), cette solution est donc optimale.