Laboratori de Telemàtica 3

Memòria de la

Pràctica 3

David Guillen Fandos
Toni Jaume Mir

Exercicis previs de la pràctica 3

· Exercici 1

Les fórmules per a un sistema M/M/N són, tal com vam veure a XSSC, les següents:

Probabilitat de demora: $$P_d = \frac{\frac{(N \cdot \rho)^N}{n! \cdot (1-\rho)}}{\sum_{k=0}^{N-1} \frac{(N \cdot \rho)^k}{k!} + \frac{N^N}{N!} \cdot \frac{\rho^N}{1-\rho}}$$

Factor d'utilització: $$\rho = \frac{\lambda}{N \cdot \mu}$$

Temps d'espera: $$T_w = \frac{\lambda}{\mu \cdot (\mu - \lambda)}$$

Nombre d'unitats al sistema: $$N_w = \frac{\lambda^2}{\mu \cdot (\mu - \lambda)}$$

Per al cas de l'exercici 2 tenim que el temps entre arribades és 10s, que vol dir que el nombre d'arribades per segon serà de 0,1 paquets/s. La utilització és del 75%, així fent ús de la fórmula del factor d'utilització trobem la taxa de servei, que és 0,134 paquets/s.

Calculem el temps d'espera que és de 22,5 segons i el nombre mitjà de paquets al sistema que és de 2,25 unitats.

Finalment la probabilitat de demora es redueix a l'expressió (N=1) següent:

$$P_d = \rho = 0{,}75 \equiv 75$$

· Exercici 2

Per a guardar valors de la forma en que ho fa Omnet (en el fitxer de vectors amb el nom i el timestamp corresponent) cal fer ús de la classe *cOutVector* (manual, pàgina 167). Es crearà una instància d'aquest objecte a la classe Server (o allà on sigui necessari) com a variable privada. En el constructor de la classe server s'inicialitzarà l'objecte fent ús del mètode *initialize* que pren com a paràmetre el nom de la variable (per al fitxer de vectors). Es pot veure com a la classe Server es fa això amb la variable *queueOccupancy*.

Omnet dóna la possibilitat de triar un fitxer de vectors diferent per a cada conjunt de variables i també la possibilitat d'especificar un temps de transitori (warm-up) on els valors de les variables són descartats.

Exercici 3

Per a realitzar una tongada de simulacions només cal compilar el programa fent ús de *cmdenv* (Command Line Environment) i activar l'*express mode* per defecte. D'aquesta manera no s'obrirà l'interfície gràfica de Omnet++ i directament s'engegarà la simulació de l xarxa.

Fent ús del shell script *mrun.sh* podem automatitzar molt el procés. Aquest script fa el següent:

- Genera *seeds* aleatoris per a cada un dels experiments.
- Prepara el fitxer de seeds.ini on afegeix el *seed* a fer servir i especifica el fitxer de vectors a usar (un diferent per cada experiment).
- Executa cada una de les proves fent ús d'un *seed* i un fitxer de sortida diferents.
- Guarda una còpia de les dades que ha emprat (per a poder reproduir una simulació)

Finalment cal filtrar la variable que ens interessa dels fitxer de vectors (com fèiem la pràctica 2) i per a fer-ho de forma automatitzada per a tots els fitxer emprarem el *fvector.sh* que simplement realitza el filtre *awk* per a tots els fitxers que li especifiquem.

· Exercici 4

```
function [mean,std] = time_weighted_mean(W)
   TD =  W(2:size(W,1),1:1) – W(1:size(W,1)-1,1:1);
   raw_avg = TD'*W(1:size(W,1)-1,2:2);
   SM = TD'*power(W(1:size(W,1)-1,2:2),2);
   mean = raw_avg./W(size(W,1):size(W,1),1:1);
   std = sqrt(SM./W(size(W,1):size(W,1),1:1)- mean*mean);
end
```

· Exercici 5

```
function result = confidence_interval(path)
   % Afegir el / final
   if (path(size(path,2)) ~= '/')
            path = [path '/']
   end
   % Matriu de mitjes i desviacions
   result = zeros(size(fitxers),2);
   % Per cada fitxer calcular
   fitxers = dir(path);
   for i=1:size(fitxers);
            vector=load([path fitxers(i).name]);
            [mean,std] = time_weighted_mean(vector);
            % Guardar el resultat
            result(i,1) = mean;
            result(i,2) = std;
   end
end
```

Exercicis de la pràctica 3

· Exercici 1

Els paràmetres del sistema a simular són:

Nombre de servidors: 10000
Temps de servei: 6s (exponencial)
Temps entre arribades: 0.001s (exponencial)

El resultat obtingut en la simulació dels paràmetres és:

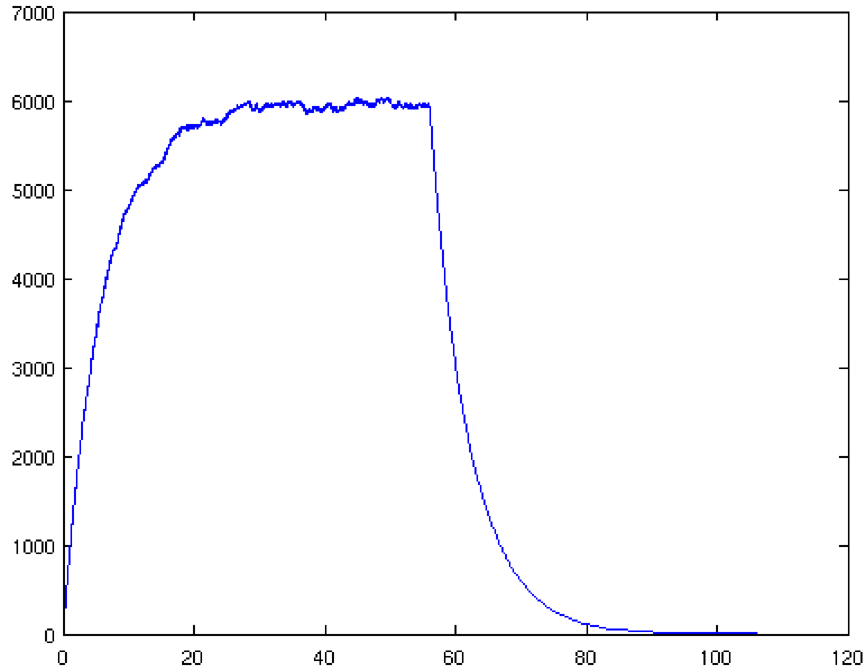
	Amb transitori	Sense transitori	Teòric
Temps entre arribades	0,001s	0,001s	0,001s
Nombre d'unitats en cua	0	0	~0
Nombre de servidors ocupats	3082	5930	6000
Probabilitat de demora	0	0	~0

Els resultats teòrics es dedueixen de les fórmules de l'exercici previ. La probabilitat de demora és massa petita com per a ser calculada (hi ha un factorial de 10^4), així que direm que és molt propera a zero.

Pel que fa a les variables de nombre de paquets en cua i temps d'espera en cua el seu valor teòric és molt petit i en la nostra simulació no tenim cap paquet que hagi esperat a la cua, així que no podem saber quin és el valor de les variables. Si féssim una simulació molt més llarga és possible que hi hagués algun paquet que s'aturés a la cua, però veient el gràfic ho veiem bastant difícil, ja que el nombre de servidors ocupats té una variància molt petita (es desvia molt poc de la mitjana).

Podem veure la importància de fer ús de les dades en tot l'interval o només en el transitori. Hem triat com a interval en règim permanent el que comprèn els instants 25s i 45s. Es pot veure al gràfics (del nombre de servidors ocupats) el perquè.

La variable que es veu més afectada pel transitori és l'ocupació dels servidors (veure gràfic). El temps entre arribades no es veu afectat pel fet que és independent del nombre de paquets al sistema (és de Poisson).



· Exercici 2

El nou sistema a simular és un M/M/1 amb paràmetres:

Temps de servei: 7,5s
Temps entre arribades: 10s
Factor d'utilització: 0,75
Cua infinita i un servidor.

Realitzem la simulació i prenem les dades per a les mostres indicades. La primera taula representa el valor mig de la variable:

	200	2000	20000	Teòric
T. arribades	9,17	10,02	10.062	10
Unitats cua	1,992	2,006	2,1611	2,25
Ocupació s.	0,7474	0,7199	0,7417	0,75
Temps cua	22,77	19,15	21,74	22,5
P. de demora	0,79	0,72	0,74	0,75

La següent taula representa la desviació típica de les variables.

	200	2000	20000	Teòric
T. arribades	10,02	9,88	9,98	10
Unitats cua	2,8402	2,9315	3,0582	?
Ocupació s.	0,4345	0,4419	0,4377	0,56
Temps cua	20,32	24,49	27,25	?
P. de demora	0,41	0,45	0,44	0,56

Cal notar que no sabem calcular el valor teòric de la majoria de les variables. Per a calcular la mitjana i la desviació típica hem fet ús de la funció de matlab dels exercicis previs (en el cas de les variables que depenen del temps) i de *mean* i *std* en el cas de les variables que no requereixen ponderació en el temps.

· Exercici 3

En l'exercici 1 tenim 10000 servidors i l'ocupació teòrica és de 0,6. Per a que això es pugui complir necessitem 6000 paquets al sistema i això requereix un cert temps. Podem plantejar-ho com una equació on busquem trobar el nombre de paquets al sistema coneixenet el fluxe d'entrada i sortida:

$$\frac{\partial N(t)}{\partial t}=1000-\frac{1}{6}\cdot N(t)$$

És a dir, el ritme d'entrada són 1000paq/s i el de sortida és 1/6paq/s per cada servidor. Suposant que no arribem mai al nombre de servidors màxim (que es compleix en el nostre cas, com veiem al gràfic) podem dir que el fluxe de sortida és el fluxe de sortida d'un servidor pel nombre de paquets al sistema, ja que el nombre de paquets al sistema i als servidors coincideix.

Aquesta equació diferencial té una solució senzilla que és:

$$N(t)=6000\cdot(1-e^{\frac{-t}{6}})$$

Que no és res més el que obtenim al gràfic. Si aproximem el règim permanent com a $t=3\tau$ (com es fa en altres disciplines de la carrera) obtenim un transitori de 18 segons, és a dir uns 18000 paquets.

El nombre màxim de paquets a la cua del exercici 2 és 28. Ho podem justifica calculant les probabilitats dels estats. La probabilitat d'estar a l'estat k-èssim és

$$P_k = (1-\rho) \cdot \rho^k$$

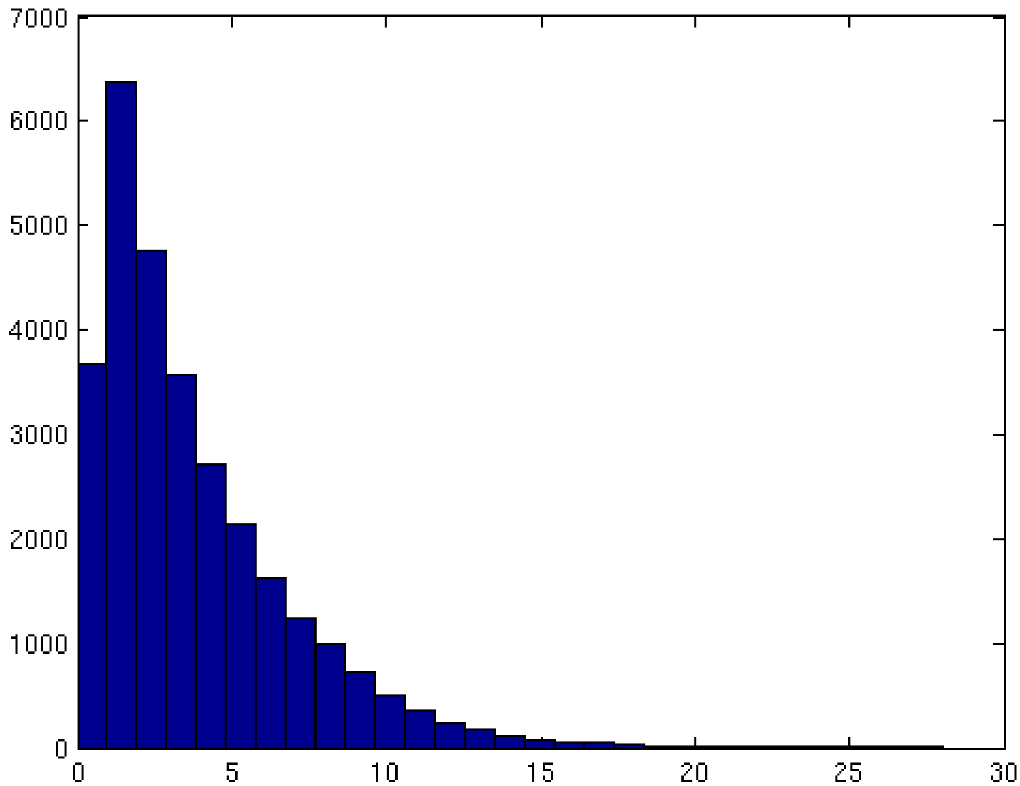
Així el nombre de paquets que trobaran el sistema a l'estat k-èssim és el nombre de paquets totals que arriben per la probabilitat de l'estat. Calculem aquests valors i veiem que teòricament s'espera que hi hagi 1,5 paquets a l'estat k=28, mentre que per a 29 el valor és quasi 1.

D'una manera intuïtiva podem dir que sabem la probabilitat de cada estat del sistema però que amb poques mostres si la probabilitat és petita no arribarem a l'estat desitjat. Per altra banda, si tenim moltíssimes mostres la probabilitat d'arribar a un estat poc probable creix.

Finalment realitzem el test de Pearson. Ens va costar bastant ajustar-ho però vam descobrir quin era el problema.

Donat que Omnet++ registra els successos en variables tenim una variable per al nombre de paquets en cua i una altra per al nombre de servidors ocupats. Com que cada cop que hi ha un esdeveniment que fa canviar una variable aquesta es guarda al registre si la variable no canvia, no es registra. És per això que la variable "Queue GLOBAL occupancy" no registra els canvis d'estat de k=0 a k=1 i viceversa, ja que no hi ha canvis en el nombre de paquets en cua. Per aquest motiu el gràfic no és una exponencial perfecte, sinó que per al valor n=0 té un comportament diferent.

Fent ús de la funció del test de Pearson de la pràctica 2 calculem el valor de K^2 que és de 17,28, cosa que ens assegura la hipòstesi per a una p=0,05.



· Exercici 4

Calculem els paràmetres del sistema ja que ara és un M/G/1.

Arribades: 0,1paq/s (exponencial)
Servei: 7,5s (3 erlang)
Factor d'utilització: 0,75
Temps d'espera a la cua: 15s
Nombre mig de paquets en cua: 1,5 ($N_Q = T_w \cdot \lambda = 1{,}5$)

$$T_w = \frac{\rho}{1-\rho} \cdot \frac{E(Ts^2)}{2 \cdot E(Ts)} = \frac{\rho}{1-\rho} \cdot \frac{4/3 \cdot E(Ts)}{2} = 15\text{s}$$ (Tw per a 3-erlang)

Apartat a:

Realitzem la simulació amb 500 mostres.

	5 simulacions			40 simulacions		
	Mitjana	Var. (biax)	Var. (no b)	Mitjana	Var. (biax)	Var. (no b)
T. espera	15,25	18,590	23,237	15,04	22,91	23,50
Arribades	9,89	0,346	0,433	9,98	0,293	0,301
Ocupació	0,761	0,00293	0,0036	0,749	0,00194	0,00199
N. espera	1,56	0,239	0,299	1,52	0,283	0,290

	5 simulacions			
	A (biaix)	A (no biaix)	B	T
T. espera	±3,779 (24,78%)	±4,225 (24,78%)	±65,362 (281,28%)	±5,541 (36,34%)
Arribades	±0,516 (5,22%)	±0,577 (5,83%)	±1,219 (281,28%)	±0,757 (7,65%)
Ocupació	±0,047 (6,23%)	±0,053 (6,97%)	±0,010 (281,28%)	±0,069 (9,14%)
N. espera	±0,428 (27,44%)	±0,479 (30,68%)	±0,842 (281,28%)	±0,629 (40,24%)

	40 simulacions			
	A (biaix)	A (no biaix)	B	T
T. espera	±1,438 (9,86%)	±1,502 (9,99%)	±11,31 (48,15%)	±1,549 (10,30%)
Arribades	±0,167 (1,68%)	±0,170 (1,70%)	±0,145 (48,15%)	±0,175 (1,76%)
Ocupació	±0,013 (1,82%)	±0,013 (1,84%)	±0,0009 (48,15%)	±0,014 (1,90%)
N. espera	±0,165 (10,86%)	±0,167 (10,99%)	±0,140 (48,15%)	±0,172 (11,34%)

Podem comprovar tal com diu la teoria que l'estimador T és quasi igual a l'estimador 4, especialment quan tenim un nombre de mostres gran (es veu que per a 40 són quasi iguals, per a 5 no massa). L'estimador T és però més conservador en la seva estimació. Hem triat com a marge de confiança el 95% per a tots els càlculs.

Veiem que a més simulacions millor és l'estimació del paràmetre i també que el marge de confiança es fa més petit. Això és bo i ens indica que els estimadors són consistents (que disminueix la seva variància amb el nombre de mostres). En l'únic cas on hi ha un cert dubte és en el temps d'espera, ja que la variable aleatòria 3-erlang té una variància molt gran, que amb 5 simulacions no és possible veure.

Finalment comentar que tal com indica la teoria i la pròpia lògica les variables amb un interval de confiança més gran són aquelles que tenen una variància més gran, és el cas del temps d'espera i el nombre de paquets en cua, que depenen d'una variable 3-erlang.

Apartat b

Repetim però ara amb 20000 mostres.

	5 simulacions			40 simulacions		
	Mitjana	Var. (biax)	Var. (no b)	Mitjana	Var. (biax)	Var. (no b)
T. espera	15,48	0,448	0,560	14,92	0,569	0,584
Arribades	9,98	0,005	0,007	10,00	0,004	0,005
Ocupació	0,75	0,00002	0,00002	0,749	0,00003	0,00003
N. espera	1,55	0,005	0,006	1,491	0,0065	0,0067

	5 simulacions			
	A (biaix)	A (no biaix)	B	T
T. espera	±0,586 (3,79%)	±0,656 (4,24%)	±1,576 (281,28%)	±1,576 (5,56%)
Arribades	±0,066 (0,66%)	±0,073 (0,74%)	±0,019 (281,28%)	±0,096 (0,97%)
Ocupació	±0,003 (0,50%)	±0,004 (0,56%)	±0,00006 (281,28%)	±0,005 (0,73%)
N. espera	±0,062 (4,00%)	±0,069 (4,47%)	±0,017 (281,28%)	±0,091 (5,87%)

	40 simulacions			
	A (biaix)	A (no biaix)	B	T
T. espera	±0,233 (1,57%)	±0,236 (1,59%)	±0,2814 (48,15%)	±0,244 (1,64%)
Arribades	±0,021 (0,21%)	±0,0021(0,21%)	±0,002 (48,15%)	±0,022 (0,22%)
Ocupació	±0,001 (0,23%)	±0,001 (0,23%)	±0,00002 (48,15%)	±0,001 (0,24%)
N. espera	±0,025 (1,69%)	±0,025 (1,71%)	±0,003 (48,150%)	±0,026 (1,76%)

Veiem com clarament els valors mitjos són molt millors en aquest cas. Donat que tenim més mostres, cada variable proporcionada per la simulació és més precisa. Així doncs al calcular la desviació de totes les simulacions ens trobem amb que té un valor prou baix, que ens porta a un interval de confiança molt petit. Aquí ja no es veu una diferència tan gran entre fer 5 o 40 simulacions, ja que cada simulació proporciona dades acurades.

Apartat c

Tornem a repetir però ara canviem el generador de nombres aleatoris.

	5 simulacions			40 simulacions		
	Mitjana	Var. (biax)	Var. (no b)	Mitjana	Var. (biax)	Var. (no b)
T. espera	15,34	3,82	4,77	15,6	4,12	4,22
Arribades	9,94	0,06	0,08	9,93	0,065	0,066
Ocupació	0,756	0,00023	0,00029	0,755	0,00023	0,00023
N. espera	1,54	0,057	0,059	1,57	0,059	0,061

	5 simulacions			
	A (biaix)	A (no biaix)	B	T
T. espera	±1,713 (11,17%)	±1,915 (12,49%)	±13,43 (281,28%)	±2,512 (16,38%)
Arribades	±0,229 (2,31%)	±0,256 (2,58%)	±0,241 (281,28%)	±0,336 (3,39%)
Ocupació	±0,013 (1,75%)	±0,014 (1,96%)	±0,0008 (281,28%)	±0,019 (2,57%)
N. espera	±0,210 (13,57%)	±0,234 (15,17%)	±0,202 (281,28%)	±0,308 (19,89%

	40 simulacions			
	A (biaix)	A (no biaix)	B	T
T. espera	±0,629 (4,03%)	±0,637 (4,08%)	±2,035 (48,15%)	±0,657 (4,21%)
Arribades	±0,079 (0,80%)	±0,080 (0,81%)	±0,032 (48.15%)	±0,082 (0,83%)
Ocupació	±0,004 (0,62%)	±0,004 (0,63%)	±0,00011(48,15%)	±0,005 (0,65%)
N. espera	±0,075 (4,80%)	±0,076 (4,86%)	±0,029 (48,15%)	±0,079 (5,01%)

Com es pot veure els intervals de confiança son molt més grans que en el cas anterior. Podem destacar que les mitjanes no són gaire bones (suposem que el generador aleatori té un cert biaix) però les desviacions no són massa grans (com a mínim no pitjor que el cas de 500 mostres).

Apartat d

Realitzem aquest apartat per a la variable temps de cua. Si ens basem en el resultat de l'estimador A amb variància tenim un interval de confiança del 10%. Recuperant la fórmula d'aquest marge veiem que és proporcional a:

$$\frac{c_1 \cdot \sigma}{\sqrt{n}}$$

Així dons suposant una variància constant trobem quina és la nova N que ens redueix la variància de 10% a 7%.

$$n' = n \cdot \left(\frac{10}{7}\right)^2 = \frac{100}{49} \simeq 2$$

Així doncs amb 80 tests hauríem d'obtenir un 7%. Si revisem bé la hipòtesi veiem que si doblem el nombre de mostres les variàncies no seran iguals, ja que a més mostres tenim menys variància. Així doncs diem que aquest valor es el que ens garanteix el que volem però ho aconseguirem amb menys mostres de ben segur. Ho comprovem i els resultats són:

n	Interval
40	10%
63	6,96%
70	6,78%
80	6,12%

Així es veu clarament com amb aproximadament 1,5 vegades més mostres tenim un 7% de marge i amb el doble de mostres tenim quasi un 6%. Donat que és molt difícil aproximar la variància d'una forma precisa (justament aquest és el nostre objectiu a la pràctica) és una bona idea quedar-se amb la cota que hem trobat, que ens garanteix que assolim l'objectiu amb un marge “extra”.

Apartat e

Si la diferència és de 1 segon i aquesta variable segueix la distribució T ens porta a dir que la probabilitat que la diferència sigui de 1 segon o més és de $2.96 \cdot 10^{-6}$, cosa que és molt estrany. Així doncs podem dir que no és acceptable una diferència de més d'un segon. Per veure-ho d'una altra forma podem dir que si acceptem un valor de 1 segon amb un cert percentatge (suposem 95% o 99%, valors habituals) la variància que obtindrem serà molt gran.

Apartat f

Resumim en una taula les dades obtingudes:

Experiment	Mitjana	Interval	Dins?
1	15,25	3,73 (24,52%)	Sí
2	16,74	4,21 (25,15%)	Sí
3	17,01	4,08 (24,00%)	Sí
4	12,97	1,26 (9,78%)	No
5	12,80	2,40 (18,76%)	Sí
6	19,31	5,16 (26,75%)	Sí
7	15,66	3,59 (22,95%)	Sí
8	14,29	2,12 (14,85%)	Sí
9	14,51	3,80 (26,21%)	Sí
10	10,58	1,16 (11,05%)	No

En principi de forma “teòrica” i donat que calculem els marges de confiança en un interval del 95% això ens hauria de garantir que el marge inclou el 95% de les vegades el valor “real”. Tot i això veiem que ha fallat 2 vegades de 10, el que ens porta a pensar que és més aviat un 80% de les vegades. El que està clar és que amb 10 experiments no n'hi ha prou per a donar una conclusió amb fonament. Caldria realitzar uns 40 o 50 experiments i veure si és veritat o no que el 95% o més dels experiments fallen en el valor de la mitja.

Apartat g

Realitzem un ajust per a la p de les exponencials. Mostrem a continuació el valor de p òptim (que minimitza la K^2). Hem triat els experiments que millor s'adaptaven, ja que n'hi havia molts que no superaven el test de Pearson degut al reduït nombre de mostres (que donaven lloc a un histograma que no era una exponencial ben formada).

Num experiment	K^2	p
1	12,52	0,7072
2	7,14	0,6435
3	3,13	0,655
4	8,41	0,618
5	5,81	0,717

Un fet apreciable és que la p és molt més petita del que esperem. Això es deu a que per a un nombre petit de mostres el sistema encara no ha arribat al seu règim permanent. Al principi hi ha pocs paquets i la cua que es forma és més petita, a mesura que avança el temps la cua creix fins arribar al règim estable. És per això que p és petita, perquè és com si el sistema tingués un factor d'ocupació menor.

Si extrapolem ens surt un valor de probabilitat proper a 0. Suposant la p mitjana (0,668) i un Nmax = 28 (exercici 3) la probabilitat és de 0,0003%, que és quasi zero, enfront al 0,005% que en el cas p=0,75.

Apartat h

Per a calcular un possible interval de confiança podem fer ús del estadístic T de student. Tenint std=0,424, mean=0,668 l'interval de confiança és de ±0,48 (per a una confiança del 95%).

Aquest resultat ens fa entreveure que no tenim unes bones mostres, ja que la p es mou molt al voltant de la mitjana. Fins i tot la mitjana està molt lluny del valor teòric.

Annexes

· Scripts bash per a l'exercici 4

treu_vectors.sh

```
for i in `ls *.vec`; do
  awk '{ if ($1 == 1) print $3 }' < $i > $i.tempscua
  awk '{ if ($1 == 5) print $3 }' < $i > $i.tempsarribades
  awk '{ if ($1 == 0) print $2 " " $3 }' < $i > $i.ocupaciocua
  awk '{ if ($1 == 6) print $2 " " $3 }' < $i > $i.ocupacioservidor
done
```

Aquest script extreu els vectors necessaris i els guarda amb el nom correcte per a que matlab els trobi (vegeu a continuació).

· Programa per al càlcul dels màrgens de confiança

```
function calcul_parametres(path,numtests,offset)
    % Parametres ideals calculats van aqui:
    temps_cua_ideal = 15;
    temps_arribades_ideal = 10;
    ocupacio_servidor_ideal = 0.75;
    ocupacio_cua_ideal = 1.5;

    % Lectura dels temps de cua
    llista = llista_directori(path,'\.tempscua');
    temps_cua = zeros(min(size(llista,2),numtests),1);
    for i=1:min(size(llista,2),numtests)
       dades=load(llista{i+offset});
       temps_cua(i) = mean(dades);
    end

    % Lectura dels temps d'arriaba
    llista = llista_directori(path,'\.tempsarribades');
    temps_arribades = zeros(min(size(llista,2),numtests),1);
    for i=1:min(size(llista,2),numtests)
       dades=load(llista{i+offset});
       temps_arribades(i) = mean(dades);
    end

    % Lectura de l'ocupacio del servidor
```

```
    llista = llista_directori(path,'\.ocupacioservidor');
    ocupacio_servidor = zeros(min(size(llista,2),numtests),1);
    for i=1:min(size(llista,2),numtests)
       dades=load(llista{i+offset});
       ocupacio_servidor(i) = time_weighted_mean(dades);
    end

    % Lectura de l'ocupacio de la cua
    llista = llista_directori(path,'\.ocupaciocua');
    ocupacio_cua = zeros(min(size(llista,2),numtests),1);
    for i=1:min(size(llista,2),numtests)
       dades=load(llista{i+offset});
       ocupacio_cua(i) = time_weighted_mean(dades);
    end

    % Calcul dels moments i estadistics
    [temps_cua_var_biax,temps_cua_var_no_biax,temps_cua_Am,temps_cua_Am2,temps_cua
_Bm,temps_cua_Tm] = ...
        calcul_estad(temps_cua,temps_cua_ideal);
    [temps_arribades_var_biax,temps_arribades_var_no_biax,temps_arribades_Am,temps
_arribades_Am2,temps_arribades_Bm,temps_arribades_Tm] = ...
        calcul_estad(temps_arribades,temps_arribades_ideal);
    [ocupacio_servidor_var_biax,ocupacio_servidor_var_no_biax,ocupacio_servidor_Am
,ocupacio_servidor_Am2,ocupacio_servidor_Bm,ocupacio_servidor_Tm] = ...
        calcul_estad(ocupacio_servidor,ocupacio_servidor_ideal);
    [ocupacio_cua_var_biax,ocupacio_cua_var_no_biax,ocupacio_cua_Am,ocupacio_cua_A
m2,ocupacio_cua_Bm,ocupacio_cua_Tm] = ...
        calcul_estad(ocupacio_cua,ocupacio_cua_ideal);

    % Resultats per pantalla
    fprintf('Execució per a %d simulacions\n\n',min(size(llista,2),numtests));

    print_results('Temps de cua', mean(temps_cua), temps_cua_ideal,
temps_cua_var_biax, temps_cua_var_no_biax, ...
                                  temps_cua_Am, temps_cua_Am2, temps_cua_Bm,
temps_cua_Tm)
    print_results('Temps entre arribades', mean(temps_arribades),
temps_arribades_ideal, temps_arribades_var_biax, temps_arribades_var_no_biax, ...
                                  temps_arribades_Am, temps_arribades_Am2,
temps_arribades_Bm, temps_arribades_Tm)
    print_results('Ocupació del servidor', mean(ocupacio_servidor),
ocupacio_servidor_ideal, ocupacio_servidor_var_biax,
ocupacio_servidor_var_no_biax, ...
                                  ocupacio_servidor_Am, ocupacio_servidor_Am2,
ocupacio_servidor_Bm, ocupacio_servidor_Tm)
    print_results('Ocupació de la cua', mean(ocupacio_cua), ocupacio_cua_ideal,
ocupacio_cua_var_biax, ocupacio_cua_var_no_biax, ...
                                  ocupacio_cua_Am, ocupacio_cua_Am2,
ocupacio_cua_Bm, ocupacio_cua_Tm)

end

function print_results(concepte, m, m_ideal, var_b, var_nb, Am, Am2, Bm, Tm)
    fprintf('  %s\n',concepte);
    fprintf('    Mitjana: %.5f, ideal(%f)\n',m,m_ideal);
    fprintf('    Variança (biaxada): %.5f\n',var_b);
    fprintf('    Variança (esbiaxada): %.5f\n',var_nb);
    fprintf('    Interval confiança (Estadístic A + variança amb biax): +/-%.5f
```

```
(%.2f %%)\n',Am,100*Am./m)
    fprintf('    Interval confiança (Estadístic A + variança sense biax): +/-%.5f
(%.2f %%)\n',Am2,100*Am2./m)
    fprintf('    Interval confiança (Estadístic B + variança sense biax): +/-%.5f
(%.2f %%)\n',Bm,100*Bm./var_nb)
    fprintf('    Interval confiança (Estadístic T): +/-%d (%.2f %
%)\n',Tm,100*Tm./m)
end

% La funció pren la variança (sigma cuadrat)!
function [var_biax,var_no_biax,mA,mA2,mB,mT] = calcul_estad(dades,mitjana_ideal)
    [var_biax,var_no_biax] = calcul_varianca(dades);

    interval_confianca = 0.95;
    interval_centrat = interval_confianca*(1+(1-interval_confianca)./2);

    % Intervals de confiança
    marge = norminv([interval_centrat],0,1);  % 95% gaussia, entre 2,5% i 97,5%
    mA = marge*sqrt(var_biax./size(dades,1));
    mA2= marge*sqrt(var_no_biax./size(dades,1));

    marge = (-1./chi2inv([interval_centrat],size(dades,1))
+1./chi2inv([interval_centrat-interval_confianca],size(dades,1)))./2; % La chi2 no
esta centrada a zero!
    marge
    mB = size(dades,1)*marge*var_no_biax;

    marge = tinv([interval_centrat],size(dades,1));
    mT = marge*sqrt(var_no_biax./size(dades,1));
end

function [var_biax,var_no_biax] = calcul_varianca(dades)
    mitjana = mean(dades);
    acum = 0;
    for i=1:size(dades,1)
        acum = acum + power(dades(i)-mitjana,2);
    end
    var_biax = acum./size(dades,1);
    var_no_biax = acum./(size(dades,1)-1);
end

% Calcula la mitjana ponderada per l'interval de temps com a pes
function mean = time_weighted_mean(W)
    TD = W(2:size(W,1),1:1) - W(1:size(W,1)-1,1:1);
    raw_avg = TD'*W(1:size(W,1)-1,2:2);
    mean = raw_avg./W(size(W,1):size(W,1),1:1);
end

% Retorna la llista de fitxers que es desitja amb el filtre
function llista = llista_directori(path,tipus)
    % Afegir el / final
    if (path(size(path,2)) ~= '/')
        path = [path '/'];
    end

    fitxers = dir(path);
    c = 1;
    llista = {};
```

```
    for i=1:size(fitxers);
        if (size(regexp(fitxers(i).name,['(~[])*' tipus]),1) == 1)
           llista{c} = [path fitxers(i).name];
           c = c+1;
        end
    end

end
```