APUNTS DE TEORIA DE CUES I SIMULACIÓ

(Grau InterUniversitari d'Estadística)

Simulació i Nºs aleatoris

FACULTAT DE MATEMÀTIQUES I ESTADÍSTICA

Esteve Codina Sancho
Lídia Montero Mercadé
Departament Estadística i Investigació Operativa

desembre del 2.011

TAULA DE CONTINGUTS

1. INTRODUCCIÓ A LA SIMULACIÓ

La simulació és una tècnica de la Investigació Operativa adreçada a l'anàlisi de sistemes complexes.

Els models analítics de sistemes reals poden resultar totalment immanejables per la seva complexitat, alternativament, si el model intenta simplificar massa la realitat, per guanyar en manejabilitat, pot proporcionar resultats invàlids. La simulació reprodueix el comportament d'un sistema real al llarg del temps.

La construcció i explotació d'un model de simulació està constituït de les següents etapes:

1. Formulació del Problema
2. Recollida de Dades i Desenvolupament del Model
3. Construcció del Programa d'Ordinador
4. Iterar fins la verificació de la correctesa del programa
5. Iterar fins la validació de la correctesa del model: el model desenvolupat ha de reproduir adequadament les dades recollides a la fase 2.
6. Disseny d'Experiments
7. Realitzar les execucions del programa de simulació amb els jocs d'experiments dissenyats a la fase 6. Recollida de resultats estadístics.
8. Anàlisi estadístic dels resultats de la simulació.
9. Documentació, discussió, presentació i si s'escau implementació dels resultats.

Les avantatges de la Simulació enfront dels models analítics són:

- Una teoria molt més assequible.
- La capacitat de tractar models més complexes amb menys hipòtesis simplificadores.
- L'adaptabilitat dels models de simulació construïts a situacions similars que impliquin variants en els paràmetres, opcions de disseny o política de gestió d'alguna de les components implicades.

- Permet reproduir el comportament d'un sistema sense haver de construir-lo, característica especialment valuosa si l'experiència té un caire destructiu (comportament de reactors nuclears) o un impacte social negatiu (assajar un pla de coordinació semafòrica per la campanya de Nadal a l'Eixample o per l'operació de sortida en un pont de Sant Joan).

Els inconvenients de la Simulació enfront dels models analítics són:

- Cost de desenvolupament i explotació dels models de simulació, doncs requereixen d'execucions repetides per facilitar resultats estadístics. El cost de desenvolupament dels models s'ha reduït cada cop més amb l'aparició en el mercat primer dels llenguatges de simulació i després dels entorns de simulació. Per una altra banda, el temps d'execució dels models és cada cop menys important amb la millora de les prestacions de les ordinadors.

- La simulació facilita ***estimacions estadístiques*** de les variables d'interès.

- No és un eina d'optimització, malgrat que pot combinar-se amb les tècniques d'optimització, a un cost avui en dia gens menyspreable.

Alguns sistemes reals que han requerit de la Simulació per la seva anàlisi són:

- El trànsit, tant a nivell de corredors urbans o suburbans d'autopistes com a nivell de xarxa urbana. La simulació permet avaluar polítiques de gestió, sistemes de control semafòric, avaluar dissenys de noves infrastructures, etc.

- Automatització de magatzems.

- Processos de fabricació: equilibri en les línies de producció de vehicles.

- Simuladors de vol, de túnels de vent, etc.

- Anàlisi de configuracions de xarxes d'ordinadors, etc.

1.1 Conceptes bàsics

L'objecte d'un estudi de simulació s'anomena sistema. Un ***sistema*** és un conjunt d'entitats que actuen i interaccionen entre elles per tal d'assolir algun objectiu.

L'estat del sistema és el conjunt de variables necessàries per la descripció de la situació del sistema en un moment donat. Les variables s'anomenen ***variables d'estat***.

Una ***entitat*** és cadascun dels elements en que es descomposa el sistema objecte de l'estudi. Les entitats es caracteritzen per les seves propietats o atributs.

<u>Classificació dels Sistemes:</u>

- Sistemes discrets: les variables d'estat canvien en punts discrets del temps. Per exemple, en la simulació d'una terminal de *check-in* d'equipatges, l'estat del sistema canvia cada cop que arriba o se'n va un client.

- Sistemes continus: les variables d'estat canvien de manera continua al llarg del temps. Per exemple, en la simulació de la reacció en cadena en un reactor nuclear el temps ha de tractar-se de manera contínua i les variables d'estat caracteritzar-se *contínuament*.

- Sistemes combinats: algunes variables d'estat canvien d'estat en punts concret del temps, mentre que d'altres evolucionen de manera contínua. La simulació del trànsit urbà constitueix un exemple d'aquest darrer tipus de sistemes, els semàfors canvien en instants concrets, mentre que la posició dels vehicles a la xarxa evoluciona de manera contínua.

<u>Classificació dels Models de Simulació</u>:

A continuació es proposen tres possibles classificacions, que no són incompatibles entre elles.

Una primera classificació obeeix al caire estàtic o dinàmic de l'objecte d'estudi:

- Models de simulació estàtics: donen representacions del sistema en un moment donat, no la seva evolució al llarg del temps. La simulació de Montecarlo és una tècnica de simulació estàtica i pot emprar-se per exemple pel càlcul d'integrals definides.

- Models de simulació dinàmics: representen l'evolució de les variables d'estat del sistema al llarg del temps.

Una segona classificació obeeix al caire determinista o estocàstic de les variables d'estat involucrades en la modelització del sistema objecte de l'estudi:

- Models de Simulació Deterministes: no contenen variables d'estat aleatòries.

- Models de Simulació Estocàstics: alguna de les variables d'estat és aleatòria.

Una tercera classificació obeeix a les característiques de l'evolució de les variables d'estat, és a dir del mateix sistema, al llarg del temps:

- Models de simulació discrets: les variables d'estat canvien en punts concrets del temps.

- Models de simulació continus: les variables d'estat evolucionen de manera contínua al llarg del temps.

- Models de simulació combinats: amb variables d'evolució discreta i variables d'evolució contínua.

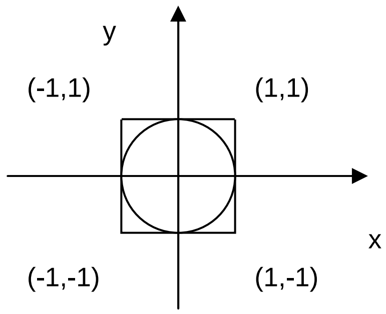
1.2 Simulació de Monte Carlo

Un mètode de Monte Carlo és aquell que usa per resoldre un problema números aleatoris, independents i uniformement distribuïts a l'interval entre 0 i 1 i s'usen mètodes numèrics no analítics. A continuació es presenta un exemple de simulació de Monte Carlo.

Exemple d'Estimació del número Π*:*

Suposem que es vol calcular l'àrea del cercle de radi unitat, és a dir $\pi \cdot R^2 = \pi$, resolent la següent integral definida:

$$\iint_{x^2+y^2\le 1} dx \cdot dy = 4 \cdot \iint_{\substack{x^2+y^2\le 1 \\ x\ge 0 \\ y\ge 0}} dx \cdot dy = 4 \cdot \frac{\pi}{4} = \pi$$



És a dir, l'àrea del cercle de radi unitat, és 4 vegades l'àrea del quadrant positiu del cercle unitat. El procediment basat en un mètode de Monte Carlo podria consistir en generar N=*1000* parelles de valors $(x_n,\, y_n)$ tal que x i y són números aleatoris independents entre 0 i 1.

Sigui $C_n = \begin{cases} 1 & x_n^2 + y_n^2 \le 1 \\ 0 & altrament \end{cases}$. Llavors $\frac{\pi}{4}$ es pot estimar a partir dels N parells de punts generats aleatòriament, com la proporció dels punts que s'escauen dins del quadrant positiu del cercle unitat, és a dir:

$$\frac{\hat{\pi}}{4} = \frac{\sum_{n=1}^{N} C_n}{N}$$

■

En Estadística Matemàtica, la simulació per Monte Carlo sol emprar-se per determinar les conseqüències de la violació de les hipòtesis de models estadístics determinats, per exemple, quin és l'efecte de la no normalitat de les variables de resposta en una regressió lineal simple.

1.3 Simulació de Successos Discrets

La simulació de successos discrets modelitza l'evolució d'un sistema mitjançant una representació a on les variables d'estat canvien en instants determinats del temps. Els instants del temps que provoquen canvis en algunes (o totes) les variables d'estats venen provocats pels esdeveniments o successos. En síntesi, requereix de determinar:

- Quines són les variables d'estat.

- Identificar quins esdeveniments provoquen un canvi en alguna/es variable/s d'estat i en quin instant de temps es produeixen. La implementació de la simulació de sistemes discrets requereix per cada esdeveniment o succés definir un procediment associat que actuï sobre les variables d'estat afectades.

- Mantenir un rellotge de la simulació (*clock simulation*).

La simulació consisteix en un seguiment o traça de l'evolució dinàmica dels canvis d'estat del sistema.

1.3.1 Exemple de simulació de successos discrets: sistema d'espera X/Y/s

Un sistema d'espera X/Y/s suposa:

- Una població infinita.

- Un espai d'espera a la cua únic i il·limitat.

- Una política de servei FIFO.

- Temps entre arribades i temps de servei per servidor distribuïts segons lleis de distribucions respectives $F_X(x)$ i $F_Y(y)$.

- L'existència de s servidors, tots ells de característiques idèntiques.

La funció de distribució de probabilitat de la variable discreta temps entre arribades de clients al sistema d'espera és:

$[X \le i]$	*1*	*2*	*3*	*4*
$F_X(x)$	0.2	0.5	0.85	1

La funció de distribució de la variable discreta temps de servei és:

$[Y \le i]$	*1*	*2*	*3*
$F_Y(y)$	0.35	0.75	1.0

L'estat del sistema ve definit per les variables d'estat:

- Número de clients al sistema d'espera.

- Estat de cada servidor.

- Instant de la propera arribada al sistema.

- Instant de la propera sortida del sistema.

Successos que provoquen canvis en les variables d'estat, en terminologia de la simulació de sistemes discrets *events*:

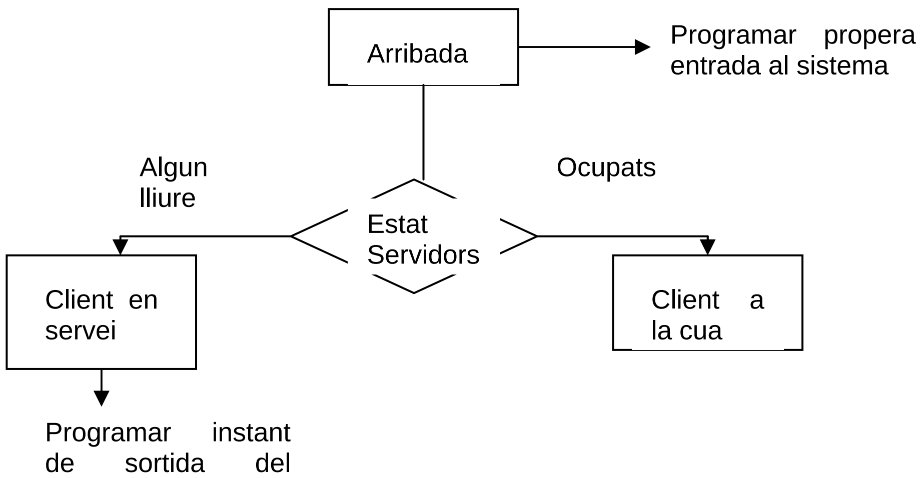
1. Arribada d'un nou client al sistema.

2. Sortida d'un client del sistema, per finalització del seu servei.

La simulació consisteix en la planificació dels diferents tipus de successos o *events* en l'eix de temps controlat per rellotge de la simulació. Per cada tipus de succés sol haver-hi una llista ordenada de successos, segons el temps de simulació, la *event list*.

Les condicions inicials del sistema a simular és la de sistema buit, per tant tots els servidors estan lliures, la cua està buida i l'inici del temps de simulació a 0 pot venir determinat per l'arribada del primer client al sistema: *time=0*.

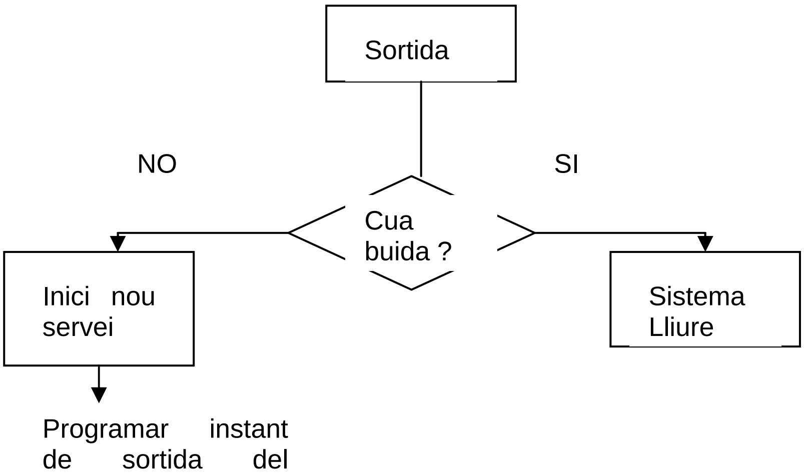
Event ARRIBADA:



L'instant de sortida del sistema d'un client que entra en servei és l'instant actual més el temps de servei generat aleatòriament segons la distribució $F_Y(y)$, és a dir: *simulation time + durada del servei*.

L'instant de la propera arribada d'un client al sistema és l'instant actual més el temps entre arribades generat aleatòriament segons la distribució $F_X(x)$, és a dir: *simulation time + interval fins la propera arribada.*

Event SORTIDA:



Els mecanismes d'actualització del temps de simulació més corrents són:

- Incrementar el temps fins el proper succés. El rellotge de la simulació avança fins l'instant de l'*event* més proper programat, requereix que els procediments vinculats a cada *event* provoquin la planificació de nous *events*.

- Increment fix del temps de simulació: Δt. El rellotge de la simulació avança de Δt en Δt i si existeix algun *event* a ser tractar aleshores procedeix a activar el procediment vinculat a l'*event*.

1.3.1.1 Algorisme de simulació discreta per la cua X/Y/s:

```
Programa X_Y_s;
       {Inicialització de les variables d'estat};
       {Seleccionar el primer succés};

       Temps = Temps del primer succés;

       Mentre temps < Durada de la Simulació fer
              Si  Event = 'Arribada' llavors
              {Procediment de l'event arribada}
              sinó
              {Procediment de l'event sortida}
              fsi;

              {Recollir estadístiques};
              {Seleccionar el proper event};
       Fmentre;
       {Imprimir resultats}
Fiprograma
```

{Procediment de l'event ARRIBADA}

```
       Temps = Temps Propera Arribada;
       Si Algun servidor està lliure llavors
              Triar servidor lliure-> s;
              Servidor s ocupat;
              {Generar sortida}
       sinó
              Afegir a la cua
       Fsi;
       {Generar Arribada}
```

{Procediment de l'event SORTIDA}

```
Temps = Temps Propera Sortida;
Si Cua buida llavors
       Servidor Lliure;
Sinó
       Triar servidor lliure-> s;
       Servidor s ocupat;
       Decrementar la cua;
       {Generar sortida}
Fsi;
```

{Generar Arribada}

```
Generar un número aleatori distribuït segons llei entre arribades -> Interval;
Temps propera arribada = Temps + Interval;
Afegir a la llista de l'event ARRIBADA;
```

{Generar Sortida}

```
Generar un número aleatori distribuït segons llei de temps de servei ->
Interval;
Temps propera sortida = Temps + Interval;
Afegir a la llista de l'event SORTIDA;
```

1.3.1.2 *Traça de la Simulació*

Suposem l'arribada al sistema X/Y/2 de 9 clients i un cronograma d'arribades i temps de servei generat *a priori* amb les següents característiques:

ID. Client	***Interval des de l'arribada anterior***	***Durada del servei***
1	-	3
2	2	3
3	2	2
4	3	1
5	4	1
6	2	2
7	1	1
8	3	2
9	3	1

Traça de la Simulació:

Event	Tipus Event	Id. Client	Time	Servidors lliures	Nº en cua	Propera Arribada	Propera Sortida
0	Inicialitz.	-	0	2	0	0	-
1	Arribada	1	0	1	0	2	3

2	Arribada	2	2	0	0	4	5
3	Sortida	1	3	1	0	4	5
4	Arribada	3	4	0	0	7	6
5	Sortida	2	5	1	0	7	6
6	Sortida	3	6	2	0	7	-
7	Arribada	4	7	1	0	11	8
8	Sortida	4	8	2	0	11	-
9	Arribada	5	11	1	0	13	12
10	Sortida	5	12	2	0	13	-
11	Arribada	6	13	1	0	14	15
12	Arribada	7	14	0	0	17	15
13	Sortida	6	15	1	0	17	15
14	Sortida	7	15	2	0	17	-
15	Arribada	8	17	1	0	20	19
							…..

1.3.1.3 Recollida d'estadístiques

Les fórmules per les estadístiques dels valors mitjos de la longitud de cua i el número de clients en el sistema són més entenedores si es defineixen les següents variables:

- Δt_i : Interval de temps entre el succés *i* i *i+1*.
- I: Número total de successos durant la simulació.
- l_{q_i} : Longitud de la cua durant l'interval Δt_i .
- l_i : Número de clients en el sistema durant l'interval Δt_i .

De manera que el número mig de clients en la cua del sistema és:

$$L_q = \frac{\sum_{i=1}^{I} l_{q_i} \cdot \Delta t_i}{\sum_{i=1}^{I} \Delta t_i}$$

I el número mig de clients en el sistema és:

$$L = \frac{\sum_{i=1}^{I} l_i \cdot \Delta t_i}{\sum_{i=1}^{I} \Delta t_i}$$

1.4 Generació de Números Aleatoris

Formalment un número aleatori és una variable aleatòria que compleix les següents propietats:

1. La variable està uniformement distribuïda en l'interval $[0,1)$.

2. La seqüència d'aquestes variables mostra independència estadística. És a dir, el coneixement d'un valor de la seqüència, x_i, no aporta coneixement sobre un altre valor x_j, $i \neq j$

1.4.1 Tècniques per la generació de números aleatoris

Una tècnica conceptualment molt simple fa ús d'una moneda equilibrada que es llença repetidament *n* vegades, suposem que es vincula el valor 1 a l'aparició de cara en un llançament i el valor 0 a l'aparició de creu. El llançament *n* vegades de la moneda configura un número binari de *n bits*. Si es divideix el número resultant per $2^n - 1$, aleshores s'obté un valor decimal entre 0 i 1 . La tècnica proporciona com a màxim 2^n valors diferents, és a dir, números aleatoris diferents. Per exemple, per *n=3* el conjunt de números binaris de 3 dígits és:

$$B = \{000 \quad 001 \quad 010 \quad 100 \quad 011 \quad 101 \quad 110 \quad 111\}$$

Si es divideixen els valors de *B* per $2^3 - 1 = 7$, aleshores es transformen en 8 números aleatoris entre 0 i 1. Tots els valors són equiprobables si la moneda és equilibrada.

Es poden dissenyar rodes de la fortuna o dispositius físics complexes per la generació de números aleatoris (per exemple, basats en els ratjos gamma emesos per un nucli radioactiu).

També es pot fer ús de les taules amb un mil.lió de dígits aleatoris publicada per la Rand Corporation (1955). Per formar un número de diguem 10 decimals a partir de la taula, s'agafarien 10 dígits seguits de la taula. La simulació manual fa us de les taules de números aleatoris, però els requeriments de números aleatoris pot ser molt gran en simulació. És pràcticament inconcebible a l'actualitat no usar la simulació per ordinador i per tant, els números *quasi aleatoris* generats pels propis ordinadors, doncs la lectura de taules de dígits aleatoris de dispositius externs (disquets, etc.) pot ser farragos i enlentir el procés de simulació. Ara bé, les seqüències de números generats pels ordinadors no solen ser totalment aleatoris, en general són números pseudoaleatoris de més o menys qualitat segons l'algorisme de generació implementat. Parlarem de la generació de números pseudoaleatoris en la propera Secció.

1.4.2 Generació de Números Pseudoaleatoris

Els números veritablement aleatoris estan distribuït uniformement a l'interval $[0,1)$ i són estadísticament independents. Els números pseudoaleatoris han d'aproximar-se a aquest ideal. Els números pseudoaleatoris han de ser reproduïbles. El generador ha de ser ràpid, perquè el programa de simulació crida el generador en moltes ocasions i els requeriments de memòria per implementar el generador han de ser reduïts i independents del tamany de la mostra a generar.

Hi ha virtualment infinites tècniques per la generació per ordinador de seqüències de números pseudoaleatoris i contínuament n'apareixen de noves, es presentaran per tant, algunes tècniques il·lustratives:

- El mètode dels quadrats mitjos. Tècnica inventada per John von Neumann als anys 40. Pressuposa que es parteix de un número x_o de *m* dígits. Quan es fa el quadrat de x_o s'obté un nou número de com a màxim *2m* dígits (s'afegeixen zeros pel davant si el nombre de dígits és inferior a *2m*). El proper número de la seqüència, x_1, s'obté retenint els *m* dígits centrals de x_0^2. Per obtenir un número en l'interval 0 a 1, es divideix x_1 per 10^m. Repetint el procediment sobre x_1, s'obté el següent valor pseudoaleatori de la seqüència. Desafortunadament, no tots el 10^m números poden aparèixer. Per exemple, si *m=2* i $x_o = 23$ s'obté:

x	x^2	r
23	0529	0.52
52	2704	0.70
70	4900	0.90
90	8100	0.10
10	0100	0.10 sempre

- El mètode congruencial. Es considera la relació $n_{i+1} = \left(a \cdot n_i + b\right) \operatorname{mod} m$ on n_0, a, b, m són números naturals. El número inicial n_0 s'anomena la *llavor* (*seed*). L'operació de mòdul (residu en la divisió entera) indica que com a màxim poden obtenir-se *m* números diferents, de *0* a *m-1*. Però en realitat solen obtenir-se *h<m* números diferents, *h* es denomina el període o longitud de cicle del generador.

  Si la constant additiva *b* és zero, aleshores el generador s'anomena *multiplicatiu*, altrament s'anomena *mixt*. Un generador congruencial genera *m* números diferents només si n_0, a, b, m compleixen uns certs requeriments: $a = 4d + 1$ on *d*, *b* són senars i $m = 2^k$ on *k* sol ser el nombre de bits per paraula en que opera l'ordinador (en binari). Les constants a, b, m tenen efectes importants sobre la independència de la seqüència de números pseudoaleatoris generada.

  Per exemple, si $n_0 = 13, a = 17, b = 0, m = 100$ aleshores s'obté un període de 20:

i	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*
n_i	$\boxed{13}$	21	57	69	73	41	97	49	33	61	37	29	93	81	77	9	53	1	17	89	$\boxed{13}$

En la literatura tècnica sobre el tema, s'han desenvolupat testos per determinar si una seqüència de números està uniformement distribuïda i són mútuament independents.

Un exemple de test per la uniformitat de la distribució d'una seqüència de *N* números pertanyents a l'interval $[0,1)$ és ***el test assimptòtic de freqüències***.

L'interval $[0,1)$ es particiona en *k* subintervals de la mateixa longitud $1/k$, cal que $N \geq 5k$. Es postula la hipòtesi nul·la que els *N* números pseudoaleatoris estan uniformement distribuïts, aleshores el nombre esperat de valors a cada subinterval ha de ser N/k. Es pot demostrar que l'estadístic indicat a continuació està distribuït aproximadament com una llei χ^2_{k-1} amb *k-1* graus de llibertat per $N \geq 5k$:

$$\frac{k}{N}\sum_{i=1}^{k}\left(f_i - \frac{N}{k}\right) \approx \chi^2_{k-1}$$

on f_i indica el total dels N números de la seqüència que s'escauen a l'interval i-èssim.

Un exemple de test per contrastar la correlació d'una seqüència de N números $r_1,\ldots,r_N$ es basa en l'estadístic $\hat{\gamma}_j$, estimador de la covariança a distància *j* de la seqüència de números, és a dir,

$$\hat{\gamma}_j = \frac{1}{N-j}\sum_{i=1}^{N-j}(r_i - 0.5)(r_{i+j} - 0.5)$$

$$\text{doncs } \gamma_j = E\left[(r_i - 0.5)(r_{i+j} - 0.5)\right] \quad j > 0$$

Ara bé, com el coeficient de correlació ρ és, $\rho = \dfrac{Cov(X,Y)}{\sigma_X\sigma_Y} = \dfrac{E[(X - E[X])(Y - E[Y])]}{\sigma_X\sigma_Y}$, si els números pseudoaleatoris d'una seqüència estan distribuït independentment llavors $\gamma_j = E\left[(r_i - 0.5)(r_{i+j} - 0.5)\right] = 0 \quad j > 0$ on $E[r_i] = 0.5,\ E[r_{i+j}] = 0.5 \ \ j > 0$. Es pot demostrar que $\hat{\gamma}_j$ és assimptòticament normal amb variança $Var[\hat{\gamma}_j] = 1/(144(n-j))$ i per tant, es pot emprar la distribució normal per contrastar la hipòtesi nul·la $H_0 : \rho_j = 0 \quad j = 1,2,\ldots,J \quad N >> J$. Malauradament per alguns valors de *j* la hipòtesi nul·la pot ser acceptada i per altres valors de *j* no.

Els generadors de números pseudoaleatoris han de passar bateries de tests i fins i tot així no hi ha una garantia absoluta de la seva bondat, construir bons generadors de números aleatoris és molt difícil, com il·lustren alguns autors. L'ús de generadors de mala qualitat pot ser nefast en simulació, doncs la validesa dels resultats d'aquesta queda automàticament invalidada. La llibreria NAG (*Numerical Algorithm Group*) conté subrutines Fortran per la generació de números pseudoaleatoris, els noms de les quals comencen per GO5 (GO5CAF, GO5CBF, GO5CCGF).

1.4.3 Generació de Números Pseudoaleatoris de distribucions discretes

En la Secció 1.4.2 es tracta la generació de seqüències de números pseudoaleatoris en l'interval $[0, 1)$, generadors que resulten imprescindibles per la generació de seqüències pertanyents a altres distribucions, siguin discretes o contínues. En aquesta secció es presenta el mètode de la taula de cerca per la generació de seqüències pseudoaleatòries de distribucions discretes qualsevol.

Sigui una funció de probabilitat d'una variable aleatòria discreta X definida amb probabilitat no nul·la per un conjunt de k valors $\chi = \{x_1, \ldots, x_k\},\ x_1 < \ldots < x_k$, és a dir $P(\{X = x_i\}) = p_i > 0 \ \ i = 1, \ldots, k$, el que dona lloc a la funció de distribució de probabilitat $F_X(x_i) = \sum_{j=1}^{i} P(\{X = x_j\}) = \sum_{j=1}^{i} p_j$.

El procediment de generació d'un número pseudoaleatori pel ***mètode de la taula de cerca*** parteix de la generació d'un valor pseudoaleatori r pertanyent a l'interval $[0, 1)$ i facilita com a resultat un valor aleatori de la distribució discreta $x = x_i$ si,

$$x = x_i \quad si \quad \sum_{j=1}^{i-1} p_j \le r_k < \sum_{j=1}^{i} p_j$$

1.4.3.1 Generació de seqüències de la llei geomètrica

Si X és una variable aleatòria geomètrica de paràmetre p aleshores la seva funció de probabilitat és $p_X(i) = P(\{X = i\}) = p \cdot (1-p)^{i-1} \quad i = 1, 2, \ldots$ i la funció de distribució de probabilitat és $F_X(i) = \sum_{j=1}^{i} p_X(j) = \sum_{j=1}^{i} P(\{X = j\}) = \sum_{j=1}^{i} p \cdot (1-p)^{j-1} = 1 - (1-p)^{i+1} \quad i = 1, 2, \ldots$. Aplicant el mètode de la taula de cerca sobre un valor aleatori r pertanyent a l'interval $[0, 1)$ s'obté,

$$x = i \qquad si \quad \sum_{j=1}^{i-1} p_j = 1 - (1-p)^i \le r < \sum_{j=1}^{i} p_j = 1 - (1-p)^{i+1}$$

1.4.3.2 Generació de seqüències de la llei binomial

Si X és una variable aleatòria binomial de paràmetres n i p aleshores la seva funció de probabilitat és $p_X(i) = P(\{X = i\}) = \binom{n}{i} \cdot p^i \cdot (1-p)^{n-i} \quad i = 0, 1, \ldots, n$ i la funció de distribució de probabilitat és $F_X(i) = \sum_{j=1}^{i} p_X(j) = \sum_{j=1}^{i} P(\{X = j\}) = \sum_{j=1}^{i} \binom{n}{j} \cdot p^j \cdot (1-p)^{n-j} \quad i = 0, 1, 2, \ldots, n$. Per la generació d'un valor pseudoaleatori x de la distribució s'emprarà una simulació estàtica que requereix de n valors aleatoris en l'interval $[0, 1)$, el procediment pot descriure's com,

```
Llegir n i p;
x=0;
Per i=1 fins n fer
Generar un valor aleatori r a l'interval [0, 1);
Si r ≤ p llavors x=x+1;
Fper
```

1.4.4 Generació de Números Pseudoaleatoris de distribucions contínues

En la Secció 1.4.2 es tracta la generació de seqüències de números pseudoaleatoris en l'interval $[0, 1)$, generadors que resulten imprescindibles per la generació de seqüències pertanyents a altres distribucions, siguin discretes o contínues. En aquesta secció es presenta el mètode de la transformació inversa per la generació de seqüències pseudoaleatòries de distribucions contínues, el qual és similar en concepció al mètode de la taula de cerca per la generació de seqüències pseudoaleatòries de distribucions discretes.

Sigui *X* una variable aleatòria contínua amb funció de distribució de probabilitat $F_X(x)$. Sigui r un valor pseudoaleatori pertanyent a l'interval $[0, 1)$, aleshores el generador facilita el valor x de la seqüència pseudoaleatòria distribuïda segons *X* tal que $x = F_X^{-1}(r) = min\{x \mid F_X(x) \geq r\}$, és a dir *x* és l'antiimatge del valor r i això té sentit, *probabilísticament parlant*, perquè la funció distribució de qualsevol variable pren valors entre 0 i 1. No sempre es requereix disposar de la forma analítica de la funció inversa de la funció de distribució $F_X(x)$, $F_X^{-1}(.)$.

1.4.4.1 Generació de Números Pseudoaleatoris de distribució exponencial

Sigui *X* una variable aleatòria exponencial de paràmetre α, llavors la funció de distribució de probabilitat $F_X(x) = 1 - e^{-\alpha \cdot x}$ $x \geq 0$. La inversa de la funció de distribució $F_X(x)$ és $F_X^{-1}(r) = \dfrac{\ln(r)}{-\alpha}$ doncs,

$$r = 1 - e^{-\alpha \cdot x} \quad \Rightarrow x = \frac{\ln(1-r)}{-\alpha} = \frac{\ln(r)}{-\alpha}$$

Així donat el valor pseudoaleatori *r* pertanyent a l'interval $[0, 1)$ es determina un valor pseudoaleatori *x* de la distribució de X.

1.4.4.2 Generació de Números Pseudoaleatoris de distribució Erlang

Sigui *X* una variable aleatòria Erlang de paràmetres α i k. La inversa de la funció de distribució $F_X(x)$ no és immediata de trobar en aquest cas i per tant cal recórrer a les propietats conegudes d'una variable d'aquestes característiques: una variable Erlang de paràmetres α i k és la suma de *k* variables exponencials independents de paràmetre α.

Així donat *k* valors pseudoaleatoris $r_1, \ldots, r_k$ pertanyents a l'interval $[0, 1)$ es determina un valor pseudoaleatori *x* de la distribució Erlang de X segons el següent procediment:

```
Llegir α i k;
x=0;
Per i=1 fins k fer
      Generar un valor aleatori r_i a l'interval [0, 1);
      {Determinar el valor aleatori corresponent de la distribució exponencial
      de              paràmetre α}
      x_i = ln(r_i) / (-α);
      x = x + x_i
Fper
```

1.4.4.3 Generació de Números Pseudoaleatoris de la distribució discreta de Poisson

Sigui *X* una variable aleatòria discreta de Poisson de paràmetre α. En aquesta ocasió també es sol recórrer a les propietats conegudes d'una variable d'aquestes característiques: una variable de Poisson de paràmetres α caracteritza el nombre d'ocurrències d'un succés en un període de temps, té un valor mig per període de α i el temps entre 2 successos està distribuït segons una llei exponencial de paràmetre α.

Així donats *k* valors pseudoaleatoris $r_1,\ldots,r_k$ pertanyents a l'interval $[0, 1)$ es determinen *k* valors pseudoaleatoris $y_1,\ldots,y_k$ pertanyents a la distribució exponencial de paràmetre α, $y_i = \frac{\ln(r_i)}{-\alpha}$, i es determina un valor pseudoaleatori *x* de la distribució de Poisson de paràmetre α tal que:

$$x = i \quad si \quad \sum_{j=1}^{i} y_j \le 1 < \sum_{j=1}^{i+1} y_j$$

x=i és el valor més gran pel qual la suma dels *i* temps inter arribades és inferior a un període.

1.4.4.4 Generació de Números Pseudoaleatoris de distribució uniforme U[a,b]

Sigui *X* una variable aleatòria uniforme en l'interval real [a,b] amb $a < b$, llavors la funció de distribució de probabilitat és $F_X(x) = \frac{x-a}{b-a} \quad a \le x \le b$. La inversa de la funció de distribució $F_X(x)$ és $F_X^{-1}(r) = a + (b-a)\cdot r$ doncs,

$$r = \frac{x-a}{b-a} \quad \Rightarrow x = a + (b-a)\cdot r$$

Així donat el valor pseudoaleatori *r* pertanyent a l'interval $[0, 1)$ es determina un valor pseudoaleatori *x* de la distribució de *X*.

1.4.4.5 Generació de Números Pseudoaleatoris de distribució Normal

Sigui *Z* una variable aleatòria Normal de paràmetres $\mu = 0 \;\; i \;\; \sigma^2 = 1$ (normal estàndard), llavors la funció de distribució de probabilitat és $F_Z(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \cdot e^{-x^2/2} \cdot dx \quad -\infty \le z \le +\infty$. La inversa de la funció de distribució $F_Z^{-1}(z)$ és indeterminable analíticament, tanmateix com l'expressió per la pròpia funció de distribució $F_Z(z)$. El mètode més simplista i obsolet rau en el Teorema Central del Límit, que diu que la suma de variables aleatòries independents tendeix a distribuir-se segons una llei normal. El teorema té diverses formulacions, però la més útil per les finalitats de la present secció és la següent:

Siguin $Y_1, \ldots, Y_n$ variables aleatòries independents i idènticament distribuïdes cadascuna d'elles amb esperança matemàtica μ_Y i variança σ_Y^2, llavors la variable $Z = \left(\overline{Y}_n - \mu_Y\right) \Big/ \frac{\sigma_y}{\sqrt{n}}$ convergeix a la distribució Normal de paràmetres $\mu = 0 \;\; i \;\; \sigma^2 = 1$ on $\overline{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$ és la mitjana de les variables $Y_1, \ldots, Y_n$.

Aleshores per generar cadascun dels valors d'una seqüència pseudoaleatòria distribuïda normalment amb paràmetres $\mu = 0 \;\; i \;\; \sigma^2 = 1$, cal generar $r_1, \ldots, r_n$, *n* valors pseudoaleatoris a l'interval $[0, 1)$ (*n* gran, més gran de 100 per anar bé), per tant d'esperança 0.5 i variança 1/12 i per resultat anterior,

$$z = \left(\frac{1}{n}\sum_{i=1}^{n} r_i - 0.5\right) \Big/ \frac{1}{\sqrt{12 \cdot n}}$$

és un valor assimptòticament distribuït segons una normal estàndard.

Si es vol generar una seqüència de valors distribuïda normalment amb paràmetres $\mu \;\; i \;\; \sigma^2$ qualsevol, llavors s'ha de fer ús de la propietat de les lleis normal que diu que *una transformació lineal d'una variable normal és també normal*. Sigui *X* una variable distribuïda normalment amb paràmetres $\mu \;\; i \;\; \sigma^2$, aleshores després d'emprar el procediment anterior per obtenir un valor *z* distribuït segons una llei normal estàndard $\mu = 0 \;\; i \;\; \sigma^2 = 1$, cal afegir-li un pas més per obtenir un valor *x* distribuït segons *X*:

$$x = \mu + \sigma \cdot z$$

El procediment descrit no és el més emprat ni el més eficient (veure la Secció 1.4.4.6), però és el més fàcilment justificable.

A partir dels procediments de generació de seqüències de números pseudoaleatoris d'una llei normal de paràmetres $\mu \;\; i \;\; \sigma^2$ qualsevol es poden desenvolupar procediments per la generació de seqüències de números pseudoaleatoris de distribucions de probabilitat relacionades amb la llei normal: llei lognormal, llei de χ^2, *t* d'Student i *F* de Fisher.

1.4.4.6 Mètode de Box-Muller

El mètode de Box-Muller és el que es recomana d'emprar en les implementacions de generadors pseudoaleatoris requerits en les pràctiques. Box-Muller és un mètode exacte per la generació de valors pseudoaleatoris de distribució normal estàndard (paràmetres $\mu = 0 \;\; i \;\; \sigma^2 = 1$) que fa ús de 2 valors pseudoaleatoris independents en el interval $[0, 1)$, r_1 i r_2, per obtenir 2 valors pseudoaleatoris normals x_1 i x_2 de la següent manera:

$$\begin{cases} x_1 = \cos(2\pi\, r_2)(-2\ln(r_1))^{0.5} \\ x_2 = \sin(2\pi\, r_2)(-2\ln(r_1))^{0.5} \end{cases}$$

Una desavantatge del mètode rau en l'alt cost computacional necessari degut a les crides de les funcions sinus i cossinus.

1.5 Simulació d'un Sistema d'Inventari: exemple d'una pràctica

L'informe corresponent a la pràctica podria contenir (criteri del professor de pràctiques corresponent):

1. Identificació personal.
2. L'enunciat de la pràctica.
3. Una descripció de les variables d'estat i dels successos dels subsistemes corresponents a la situació que descrigui l'enunciat.
4. Els mètodes emprats per la generació dels números aleatoris que requereixi la simulació.
5. Una descripció de la implementació (programa) efectuada en forma de les variables i estructures de dades més importants i el pseudo-codi.
6. Una traça de la simulació en forma de llistat de les variables més importants corresponents a un període de temps limitat que ocupi com màxim dos fulls i que reflecteixi l'actualització correcte de les variables d'estat i l'aparició del tots els tipus de successos que intervinguin en el model que s'està simulant.
7. Els resultats finals de un número (mínim 4) d'execucions del programa, fixat per vosaltres mateixos. (en el exemple que es presenta *beneficis mensuals al final de un període de 1 any*).

A continuació, es descriu un exemple de la gestió d'un inventari seguint les pautes indicades en els punts anteriors. Apareixen coberts els punts 2, 3, 5, 6 de l'informe corresponent. Aquest pot ser un enunciat de característiques molt semblants al que us pot haver estat assignat.

1.5.1 Enunciat

En una ciutat dividida en K districtes, una companyia de venda de recanvis de peces de vehicles ha instal·lat un magatzem en cada districte per cobrir les peticions diàries de la respectiva població. Quan les existències d'un dels magatzems necessiten de reposició, el magatzem efectua una comanda a un centre distribuïdor de la companyia que atén centralitzadament les peticions en forma de una cua del tipus **. / Y / s=1**. (El temps de servei de l'únic servidor del centre distribuïdor és una variable aleatòria discreta que segueix la llei de probabilitats Y). En efectuar una comanda, un magatzem no torna a efectuar una nova comanda fins que s'ha satisfet l'anterior i, d'aquesta forma el màxim número de comandes pendents de servir que tindrà el centre distribuïdor serà K. La política de cada magatzem es una ***(Q,r)*** sense retenció de la demanda: per un dia qualsevol, en quedar el nivell d'inventari per sota de les r unitats s'envia una comanda de Q unitats al centre distribuïdor.

A cada un dels K districtes s'efectua un número aleatori de peticions de recanvis al respectiu magatzem que es distribueix segons la llei de Poisson amb una mitjana de 25 unitats diàries. El temps de servei al centre de distribució segueix la llei Y = uniforme entre [1, 4] dies. El preu de venda al públic de cada unitat és de 10 ∈ en tots els districtes. Els costs de magatzematge venen associats al nivell d'inventari a l'inici del dia, ja que les vendes es serveixen al final del dia. Aquests costs són de 1 ∈ per unitat en magatzem. Si el magatzem no disposa de suficients unitats per servir totes les peticions d'un dia, llavors serveix només les que disposa en existència, però, per un acord amb la clientela, ofereix un descompte de 1 ∈ per unitat no satisfeta al client en una compra posterior, la qual cosa, per simplificar, pot tractar-se com una pèrdua immediata de 1 ∈ .

Establir mitjançant simulació quins són els beneficis mensuals per la companyia. Suposar K = 2.

1.5.2 Paràmetres

K = Número de magatzems
T = Longitud de l'horitzó de temps a simular
S = Longitud de temps en el que es reben demandes pels magatzems per part de la població (comú a totes les poblacions. Típicament *T/S* = enter)

Paràmetres específics de la població de clients l-èssima:

- Llei de probabilitats de la v.a. $X(l)$ = Nº de unitats sol·licitades durant les S unitats de temps al magatzem l. *(P.ex: v.a. aleatòria Poisson de mitjana λ)*

Paràmetres específics del magatzem l-èssim:

- Política (Q,r) sense retenció de la demanda.
- Preus de manteniment de l'inventari pel magatzem l : $P_M(l)$
- " de penalització " " " : $P_P(l)$
- " de venda del producte " : $P_V(l)$
- Quantitat d'unitats demanada " : $Q(l)$
- Nivell mínim d'inventari " : $r(l)$

Paràmetres específics del centre de distribució:

- Sistema d'espera . / Y / 1. (1 servidor amb temps de servei *Ts* , v.a. distribuïda segons una llei de probabilitats determinada. P.ex: v.a. discreta i uniforme distribuïda entre [*n.S*, *(n+m)S*], *n* > 0).

1.5.3 Variables d'estat

Tck = instant de simulació.

N(l) = Existències del magatzem *l* .

C(l) = 1 si el magatzem *l* ha fet una comanda al distribuïdor no satisfeta.

0 altrament.

Np = Número de peticions en el centre de distribució per satisfer.

1.5.4 Successos

1. Demanda de la població *l* : (*X(l) > 0)*.
2. Petició del magatzem *l* al distribuïdor.
3. Envio de *Q(l)* unitats al magatzem *l* per part del distribuïdor.

Típicament els successos *futurs* es guarden en una *estructura de dades* en forma de llista encadenada d'elements, la qual s'emplena pel final de llista i és buida pel inici de llista. Els elements de la llista consisteixen dels següents camps:

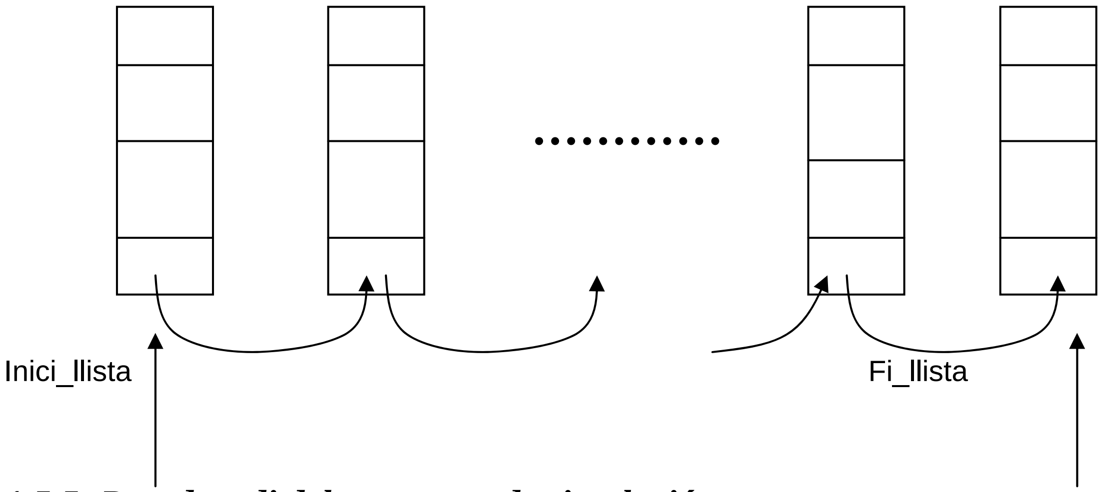
τ - Instant en el que es dona el succés.

I - Tipus de succés.

Paràmetres o valors numèrics propis del succés

\- Apuntador al següent succés.

En aquest cas únicament els successos tipus 2 es guardaran en forma de llista ja que, per simplificar es considerarà que la demanda per part de la població i l'envio de unitats per part del centre de distribució són successos "presents" (es generen dins d'un període de *S* unitats de temps)



1.5.5 Pseudocodi del programa de simulació.

Declaracions de les variables més importants.

Tck - Enter. Instant de simulació

N(l) – Array d'enters. Existències dels magatzems

R(l) - Array de flags. Indica recepció de unitats provinents del distribuïdor

r(l) – Array d'enters. Nivell mínim d'inventari per sota del qual s'envia una comanda al distribuïdor.

X(l) – Array d'enters. Demandes de les poblacions

C(l) – Array de flags . Indica petició al centre distribuïdor.

Pen(l) - Array de flotants. Import total de les penalitzacions per demanda no satisfeta.

Ing(l) - Array de flotants. Import total dels ingressos per vendes als clients.

Man(l) - Array de flotants. Import total de les despeses pel manteniment d'inventari.

Succes – Estructura. Registra una petició de unitats per un magatzem al proveïdor.

Succes.TS = Enter. Instant de simulació en que es produirà el succés.

Succes.MAG = Enter. Magatzem que ha efectuat la petició

Succes.PUN = Apuntador

Inici_llista – Apuntador

Fi_llista – Apuntador

Np – Enter. Longitut de la llista *Succes*.

Inicialitzacions:

```
N(l) = valor inicial de l'inventari, l = 1, 2, ..., K

Fixar valor per T ;  Tck = 0

R(l) = 0,   l =  1, 2, ..., K

Q(l) = 100,  l = 1, 2, ..., K

r(l) = 20,  l = 1, 2, ..., K

Inici_llista = NULL;   Fi_llista = NULL
```

```
Mentre   Tck < T  fer
    Per  l = 1, ..., K
             Si   ( R(l) = 1 )  llavors
                      N(l) = N(l) + Q(l);   R(l) = 0;   C(l) = 0
             FiSi
             X(l) = { Generar_Demanda(l) };
             Man(l) = Man(l) + Manteniment( N(l), l )

             Si  ( X(l) <= N(l)  )  llavors
                         N(l) = N(l) - X(l);
                         Ing(l) = Ing(l) + PVP( X(l), l )
                                                Altrament
                         Ing(l) = Ing(l) + PVP( N(l) );
                         Pen(l) = Pen(l) + Import_Pen( N(l) - X(l), l );
                         N(l) = 0;
             FiSi

             Si ( N(l) < r(l) & C(l) = 0 ) llavors
                   { Efectuar_Comanda(l) };  C(l) = 1
             FiSi
    FiPer

    { Proveidor(Tck) };
     S'escriuen variables X, N, Man,, Ing, Pen, R, C, Estat de la cua;
    Tck = Tck + S

Fi Mentre
```

```
Proveidor(Tck)

    Variable local:  Pun = Apuntador

    Si ( Fi_llista ≠ NULL ) llavors
          Si ( Tck =  (↑Inici_llista). TS ) llavors
               R( (↑Inici_llista). MAG ) = 1; Np = Np -1;
               Pun = Inici_llista;
               Inici_llista = (↑Inici_llista).PUN; Alliberar( Pun );
               Si ( Inici_llista = NULL ) llavors
                          Fi_llista = NULL
               FiSi
          FiSi
    FiSi
Return.
```

```
Efectuar_Comanda(l)
      Variable local:  tau = enter;   Pun = Apuntador

      tau = { Generar_Temps_de_servei };

      Pun = Crear( Success );  Np = Np + 1;
     (↑Pun).PUN = NULL;
     (↑Pun).MAG = l;
      Si ( Inici_llista = NULL ) llavors
                    Fi_llista = Inici_llista = Pun;
                  (↑Pun).TS = Tck + tau
                                                Altrament
                  (↑Pun).TS = (↑Fi_llista).TS + tau
                  (↑Fi_llista).PUN = Pun;    Fi_llista = Pun
      FiSi
 Return

  Manteniment(a,l) = a
  Pvp(a, l) = 10 a
 Import_pen(a, l) = - min{a, 0}

 Retard en servir una comanda:   v.a. uniforme entre [1,4] dies.
 Demanda X(l):  v.a aleatòria Poisson de mitjana 25 unitats.
```

Tck	X(1)	N(1)	M(1)	Pvp(1)	Pen(1)	R(1)	C(1)	X(2)	N(2)	M(2)	Pvp(2)	Pen(2)	R(2)	C(2)	T1	T2	Np	1.1.1.1.1.1.1 Cua
1	28	72	100	280	0	0	0	21	79	100	210	0	0	0	-	-	0	-
2	25	47	172	530	0	0	0	22	57	179	430	0	0	0	-	-	0	-
3	15	32	219	680	0	0	0	20	37	236	630	0	0	0	-	-	0	-
4	18	**14**	251	860	0	0	1	25	**12**	273	880	0	0	1	2	1	2	1(6), 2(7)
5	15	0	265	1000	1	0	1	11	1	285	990	0	0	1	-	-	2	1(6), 2(7)
6	20	0	265	1000	21	1	1	21	0	286	1000	20	0	1	-	-	1	2(7)
7	14	**86**	365	1140	21	0	0	20	0	286	1000	40	1	1	-	-	0	-
8	24	62	451	1380	21	0	0	34	**66**	386	1340	40	0	0	-	-	0	-
9	30	32	513	1680	21	0	0	30	36	452	1640	40	0	0	-	-	0	-
10	15	**17**	545	1830	21	0	1	18	**12**	488	1820	40	0	1	3	2	2	1(13), 2(15)
11	2	15	562	1850	21	0	1	20	0	500	1940	48	0	1	-	-	2	1(13), 2(15)
12	20	0	571	2000	26	0	1	12	0	500	1940	60	0	1	-	-	2	1(13), 2(15)
13	1	0	571	2000	27	1	1	26	0	500	1940	86	0	1	-	-	1	2(15)
14	10	**90**	671	2100	27	0	0	14	0	500	1940	100	0	1	-	-	1	2(15)
15	7	83	761	2170	27	0	0	27	0	500	1940	127	1	1	-	-	0	-
16	11	72	844	2280	27	0	0	26	**74**	600	2200	127	0	0	-	-	0	-

Q=100, r=20

Cua:

Peticio_Magatzem (instant d’entrega)