SIMULACIÓ

Cristina Montañola, cristina.montanola@upc.edu

SIMULACION

- TECNICA PARA IMITAR EN UN COMPUTADOR
- LAS OPERACIONES DE LOS SISTEMAS DEL MUNDO REAL
- A MEDIDA QUE EVOLUCIONAN EN EL TIEMPO,
- MEDIANTE MODELOS QUE LOS REPRESENTAN DE FORMA REALISTA

La Naturaleza de la Simulación

- La simulación de un sistema es la operación con un ***modelo que es una representación del sistema***, es sensible a las manipulaciones, y a partir del cual se pueden inferir propiedades sobre el comportamiento del sistema real
- La simulación puede considerarse una alternativa a los modelos analíticos consistente en un técnica que ***imita en un computador*** las operaciones del sistema real a medida que ***evoluciona en el tiempo***

CONDICIONES GENERALES DE UTILIZACION DE LA SIMULACION (SHANNON)

1. No existe una formulación matemática completa del problema, o no se han desarrollado aún los métodos analíticos para resolver el modelo matemático.

2. Existen los métodos analíticos, pero las hipótesis simplificadoras, necesarias para su aplicación, desvirtúan las soluciones obtenidas y su interpretación.

3. Los métodos analíticos existen, y en teoría están disponibles, pero los procedimientos numéricos son tan arduos y complejos que la simulación constituye un método más sencillo para obtener una solución.

4. Es deseable observar una historia simulada del proceso dentro de un horizonte temporal dado para poder estimar ciertos parámetros.

5. La simulación constituye la mejor alternativa por la dificultad de realizar experiencias en el contexto real.

6. Es necesario realizar una *compresión temporal* para estudiar la evolución del sistema a largo plazo.

ETAPAS DE UN ESTUDIO DE SIMULACION

1. Definición del problema y planificación del estudio
2. Recogida de Datos
3. Formulación del modelo de simulación
4. Construcción y verificación del programa para computador del modelo
5. Ejecuciones de prueba del modelo
6. Validación del modelo
7. Diseño de los experimentos de simulación
8. Ejecución de los experimentos.
9. Análisis de los resultados

La parte de imagen con el identificador de relación rId1 no se encontró en el archivo.

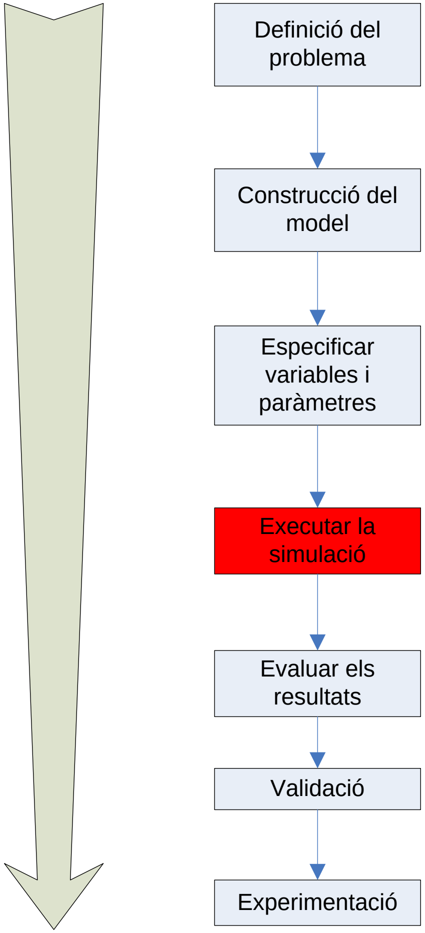
Aleatorietat

Aleatorietat: perquè?

- Definir els objectes i les variables del sistema.
- Especificar els paràmetres, regles de decisió, distribucions de probabilitat.
- Determinació de les condicions inicials.
- Generació dels GNA/GVA.
- Proves estadístiques.
- DOE.

Aleatorietat en simulació: perquè?

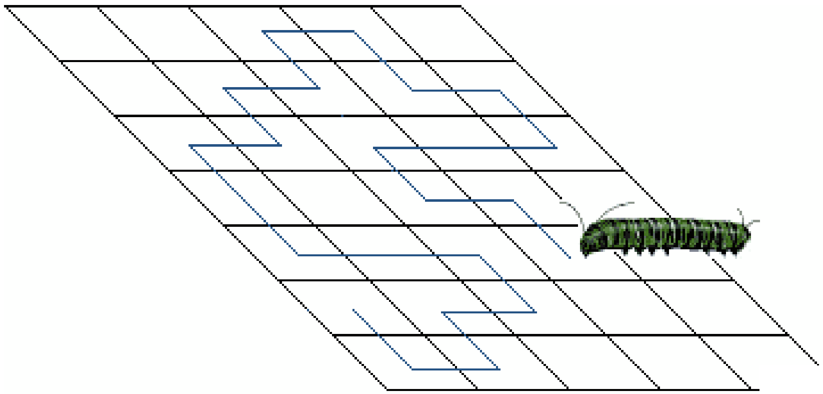
- Definir els objectes i les variables del sistema.
- Especificar els paràmetres, regles de decisió, distribucions de probabilitat.
- Determinació de les condicions inicials.
- Generació dels GNA/GVA.
- Proves estadístiques.
- Disseny d'experiments.

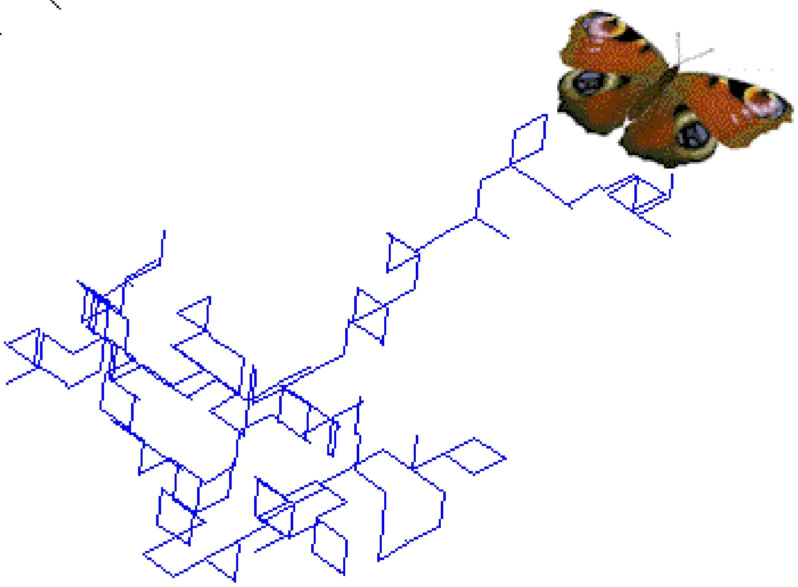


Fonts d’aleatorietat en aplicacions

Type of system	Sources of randomness
Manufacturing	Processing times, machine operating times before a downtime, machine repair times
Computer	Interarrival times of jobs, job types, processing requirements of jobs
Communication	Interarrival times of messages, message types, message lengths
Defense-related	Arrival times and payloads of missiles or airplanes, outcome of an engagement, miss distances for munitions

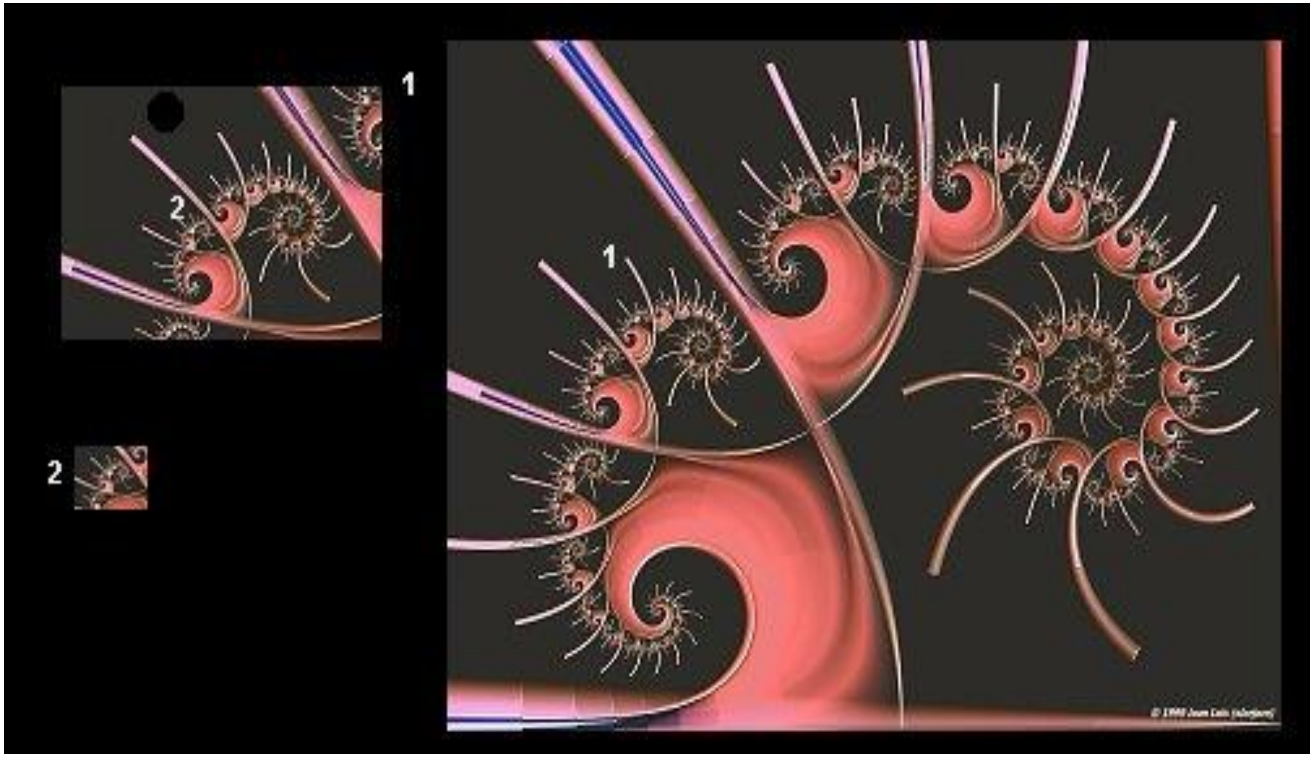
Representació de moviment





La natura

- Tenen els pulmons dimensió de Hausdorff 2½ ?



Ús en un simulador

- Els nombres aleatoris constitueixen un component fonamental d’un simulador.
- L’aleatorietat vindrà introduïda pels **generadors de nombres aleatoris** (GNA)
- Els GNA són variables aleatòries generades segons una distribució uniforme U[0,1).

Nombres pseudo-aleatoris

- Els nombres generats pels GNA no són aleatoris, donat que procedeixen d'un mètode <u>reproduïble</u> i <u>conegut</u>
- Són nombres pseudo-aleatoris, NO ALEATORIS.
- Es poden usar doncs encara que són dependents entre ells un conjunt de <u>tests</u> permeten assegurar que <u>semblen</u> aleatoris.

Característiques dels GNA

- Estructurals
  - Llarg període
  - Reproductibilitat
- Aspectes estadístics
  - Uniformitat de la distribució.
  - Densitat.
  - Independència estadística.
- Aspectes teòrics
  - Complexitat.
  - Estabilitat del procés.
- Aspectes computacionals
  - Facilitat de programació.
  - Requeriments de memòria.
  - Eficiència.

Estructurals

- **Llarg període**: Interessa GNA amb un llarg període.
  - **Període**: nombre màxim de valors diferents que el GNA es capaç de generar a partir d'uns paràmetres determinats. (*streams*)
  - Els GNA amb un període més gran son els *Mersenne Twister* de Matsumoto y Kurita [MATS94], produeixen sèries de l'ordre de 2^{19937}-1, un nombre de l'ordre de 6000 dígits decimals (algoritme MT19937B en http://www.math.keio.ac.jp/~matumoto).

Estructurals

- **Reproductibilitat**: ser capaços de reproduir de forma fàcil i sistemàtica qualsevol sèrie de nombres pseudo-aleatoris. Usant GNA.
  - El nombre base de la sèrie s’anomena llavor.
  - Aquesta característica és <u>fonamental</u> en la fase d'experimentació, anàlisi de resultats i tècniques de reducció de la variància.

Aspectes estadístics

- **Uniformitat de la distribució**: Els nombres aleatoris han de distribuir-se de forma uniforme a lo llarg de tot el rang de valors possibles que pugui prendre la sèrie, de forma que no apareguin trams no recoberts ni agrupacions no desitjades.

Aspectes estadístics

- **Densitat:** Aquesta propietat té molta relació amb la dimensió del període i del cicle de la sèrie de nombres generada.
  - A més gran densitat, més gran precisió i finor en la definició dels fenòmens.

Aspectes estadístics

- **Independència estadística:** Bàsicament una seqüència de nombres aleatoris és aquell conjunt de nombres sobre els que no es pot establir cap relació rellevant amb els restants nombres de la sèrie.
  - Molts dels mètodes de generació més habituals son recursius, per lo que no es pot afirmar-se el principi anterior ja que cada nombre és calculat mitjançant un algoritme determinista aplicat a un altre nombre que l'ha precedit en la sèrie.
  - És per aquest motiu que els nombres s'anomenen <u>pseudo-aleatoris</u>.
  - Cal assegurar que compleixen els requisits especificats mitjançant la superació de determinats <u>tests</u>.

Aspectes teòrics

- **Complexitat**: els procediments de generació han de ser necessàriament simples, poc costosos en termes computacionals i, en conseqüència, la complexitat algorítmica dels mateixos haurà d'estar convenientment acotada.

Aspectes teòrics

- **Estabilitat del procés**: Els algoritmes de generació (GNA) han de ser estables i no degenerar a lo llarg de la seva utilització en un experiment de simulació.

Aspectes computacionals

- **Facilitat de programació**: possibilitat de realitzar probes i depuració d'errors.
  - Implementació raonable, fiabilitat, mantenibilitat i capacitat de reutilització.

Aspectes computacionals

- **Requeriments de memòria:** Quantitat de memòria usada pel GNA.
  - Encara que aquest no és un requeriment tant crític com ho va estar en el passat, criteris de racionalitat en determinats problemes poden aconsellar l'ús d'algoritmes amb baixa ocupació de memòria.

Aspectes computacionals

- **Eficiència**: La velocitat d'execució del GNA.
  - Fa referència al temps de posada a punt i el temps d'execució.
  - En alguns projectes, com l'aplicació de mètodes de Montecarlo en l'àmbit de la física o de la biologia, pot arribar a ser un factor fonamental.

Història dels GNA

- Les primeres famílies de generadors es van basar en mètodes físics o mecànics
  - Taules, nombres de loteries, boles d'urna, emissions de raig gamma, discs perforats, raigs còsmics, soroll tèrmic, fenòmens radioactius

- Aquests mètodes no són útils en simulació:
  - No permeten reproduir fàcilment les sèries. Introdueixen una gran complexitat en la depuració d'errors i en la gestió d'experiments.

Història

- *El primer generador aritmètic va ser proposat per Von Neumann*
- *i Metropolis al 1940 : el "Mètode del quadrat mig"*

i	Zi	Ui	Zi2
0	2684	–	07203856
1	2038	0.2038	04153444
2	1534	0.1534	02353156
3	3531	0.3531	12446161
4	4631	0.4631	21446161
5	4461	0.4461	19900521
6	9005	0.9005	81090025
:	:	:	:

Història

- *"Mètode del quadrat mig"*
  - Partir de un número cualquiera (es recomendable un número par) de n dígitos x_0
  - Elevarlo al cuadrado y extraer los n dígitos del medio $\rightarrow$ un nuevo número x_1
  - Repertir el proceso
  - Ejemplo: $x_0=5729$

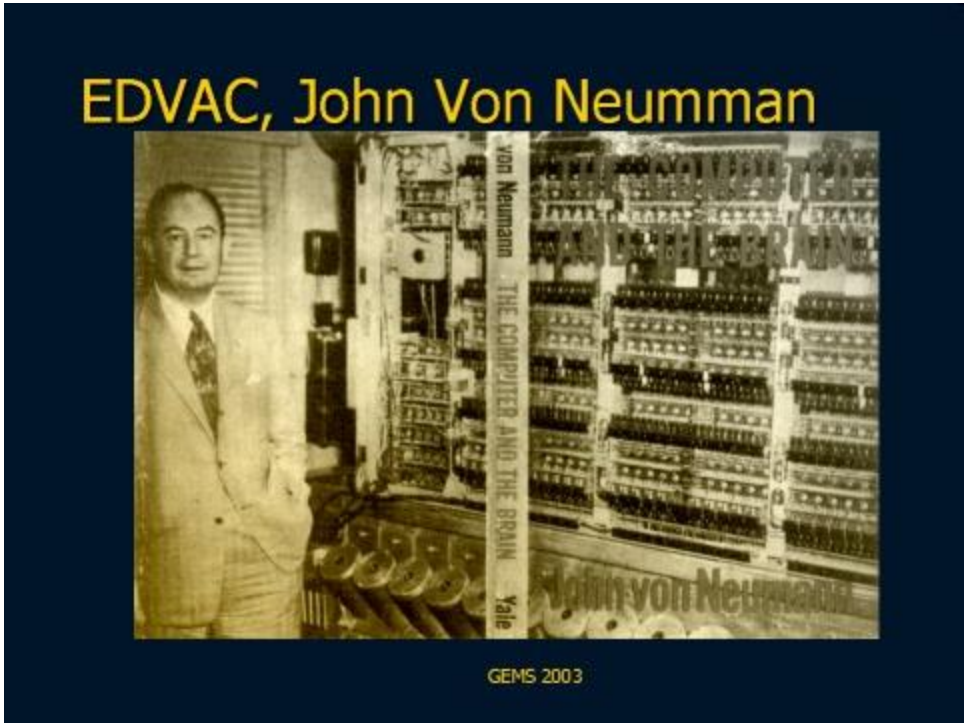
    $x_0^2=32/8214/41$

    $x_1=8214$
  - Generación de 60 números de cuatro dígitos partiendo del 5729:

8214	2842	0415	1456	4250	1405
9140	0769	1722	1199	0625	9740
5396	5913	9652	4376	3906	8676
1168	9635	1611	1493	2568	2729
3642	8332	5953	2290	5946	4474
2641	4222	4382	2441	3549	0166
9748	8252	2019	9584	5954	0275
0235	0955	0763	8530	4501	0756
0552	9120	5821	7609	2590	5715
3047	1744	8840	8968	7081	6612

Història

- Von Neumann amb el EDVAC (Electronic Discrete Variable Automatic Computer) una de les primeres computadores electròniques. A diferència del ENIAC, no era decimal, sinó binaria i va tenir el primer programa dissenyat per ser emmagatzemat. Aquest disseny es va convertir en estàndard d'arquitectura per la majoria dels ordinadors moderns.



Tipus de GNA

- Marsaglia y Zaman (1991) classifiquen els GNA en tres classes principals:
  1. **Generadors Congruencials Lineals** (CGL).
  2. Generadors Shift – Register.
  3. Generadors *lagged* – Fibonacci.

Generadors congruencials

- Los GCL van ser introduïts per Lehmer al 1951.
- Molts dels generadors que existeixen en l'actualitat són d'aquest tipus.
- Aquests generadors estan definits per la següent fórmula recursiva:

$$Z_i = (a Z_{i-1} + c) \pmod m$$

  - **m** és el mòdul.
  - **a** és el multiplicador
  - **c** és el factor additiu.
  - $\mathbf{Z_o}$ és la llavor
- Es compleix, evidentment, que $0 \leq Z_i \leq m-1$

Generadors congruencials

- Per obtenir els nombres que es desitja, conforme una U[0,1) cal fer la divisió

$$U_i = Z_i / m$$

- Per evitar la negativitat els enters m, c, a y Z_0 han de complir:
  - $0 < m$,
  - $a < m$,
  - $c < m$,
  - $Z_0 < m$.

Generadors congruencials

- Exemple de GNA de cicle complert.
- Donat que m=16 el màxim nombre de números que pot generar es 16.
- És convenient que el valor de ***m*** sigui molt gran, tant com puguem, per exemple de l'ordre de10^9.

i	*Ui*
0	-
1	0.375
2	0.063
3	0.500
4	0.688
5	0.625
6	0.313
7	0.750
8	0.938
9	0.875
10	0.563
11	0.000
12	0.188
13	0.125
14	0.813
15	0.250
16	0.438
17	0.375
18	0.063
19	0.500

$Z_0 = 7$

m = 16
a=5
c=3

Generadors congruencials

- **Cicles**: quan Z_i pren un valor igual a un altre Z_j amb $j<i$ apareix un cicle, els valors es repeteixen eternament.
- **Període**: la longitud d'aquest cicle, "i".
  - Donat que $0 \leq Z_i \leq m-1$, el valor més gran del període és ***m***.
- **Període complert** (full period): quan i=m

Generadors congruencials

- Selecció dels paràmetres:
  - ***c*** y ***m*** han de ser primers entre si.
  - Si ***q*** és un nombre primer que divideix a ***m***, llavors ***q*** també divideix a **a-1**
  - Si ***m*** és múltiple de 4, ***a-1*** es múltiple de 4 (***a*** no pot ser múltiple de 4).
- Aquestes condicions es compleixen si:
  - $m= 2^b$
  - $a=4k+1$
  - c senar (amb a, c i k enters positius)
- D'aquesta forma obtenim període complert.

Generadors congruencials

TABLE 7.2
The LCG $Z_i = (5Z_{i-1} + 3) \pmod{16}$ with $Z_0 = 7$

i	Z_i	U_i	i	Z_i	U_i	i	Z_i	U_i	i	Z_i	U_i
0	7	—	5	10	0.625	10	9	0.563	15	4	0.250
1	6	0.375	6	5	0.313	11	0	0.000	16	7	0.438
2	1	0.063	7	12	0.750	12	3	0.188	17	6	0.375
3	8	0.500	8	15	0.938	13	2	0.125	18	1	0.063
4	11	0.688	9	14	0.875	14	13	0.813	19	8	0.500

The "looping" behavior of the LCG in Example 7.2 in inevitable. By the definition in Eq. (7.1), whenever Z_i takes on a value it has had previously, exactly the same sequence of values is generated, and this cycle repeats itself endlessly. The length of a cycle is called the *period* of a generator. For LCGs, Z_i depends *only* on the previous integer Z_{i-1}, and since $0 \le Z_i \le m-1$, it is clear that the period is at most m; if it is in fact m, the LCG is said to have *full period*. (The LCG in Example 7.2 has full period.) Clearly, if a generator is

Generadors congruencials

- Els generadors congruencials es poden classificar en funció dels valors que prenguin la seva variable c:
  - si $c > 0$ GCL Mixt
  - si $c = 0$ **GCL Multiplicatiu**
  - si $c \neq 0$ y $a=1$ GCL Additiu

LCG Multiplicatius c=0

- Impedeix obtenir períodes complerts.
- Es poden obtenir períodes de mida ***m***-1, si els valors de ***m*** i ***a*** són seleccionats adequadament.
- Al igual que per c>0 és convenient usar $m=2^b$ per eliminar explícitament la divisió.

La parte de imagen con el identificador de relación rId1 no se encontró en el archivo.

GVA

Generació de variables aleatòries

GVA Mètode de la transformada inversa

- Considerem una variable aleatòria X, amb una distribució acumulada $F_X(x)$. És senzill mostrar que la variable aleatòria:

  $$U = F_X(x)$$

  té una distribució U(0,1), ja que U: R → [0,1).
- Si $F_X(X)$ és estrictament creixent, es pot reescriure l'equació en la seva forma equivalent

  $$X = F_X^{-1}(U)$$

  I veiem que F^{-1}: [0,1) → R
- Aquesta transformació s'anomena **la transformada inversa.**
- Presenta un bon mètode quan la inversa de $F_X(x)$ és fàcil de calcular.

GNA Mètode de la transformada inversa (Exemple 1)

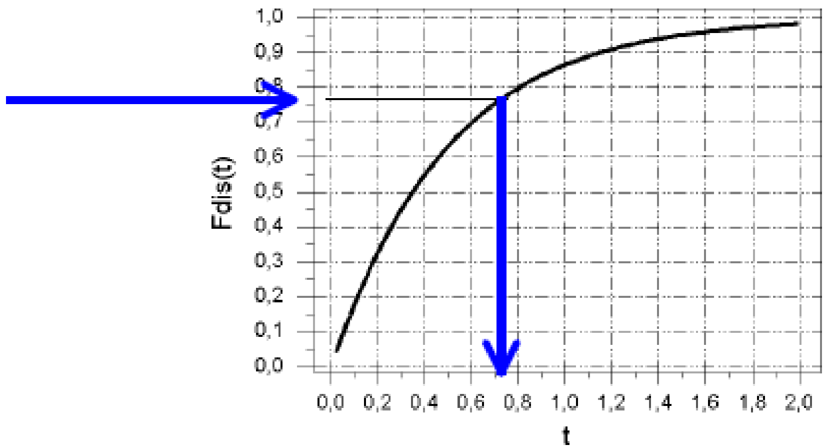
- **Exemple**: Sigui X una variable aleatòria que segueix una distribució exponencial amb paràmetre λ. Llavors:
- $F_X(x) = 1 - exp(-\lambda x)$ si $x > 0$ and $F_X(x) = 0$ altrament.
- A partir d'aquí: $F_X^{-1}(u) = - \lambda^{-1} ln(1-u)$
- El mètode de la transformada inversa es pot aplicar també a distribucions discretes, però no es pot calcular la inversa de la distribució acumulada. No obstant es pot agafar la inversa generalitzada definida per:

$$F_X^{-1}(u) = \min\{x \mid u \leq F_X(x)\}$$

Transformada inversa: Exponencial

- Mètode de la transformada inversa per a la distribució Exponencial.

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



DISTRIBUCIÓN K-Erlang (K etapas exponenciales, de parámetro α

$$x = \frac{\ln\left(\Pi_{i=1}^{k} r_i\right)}{-\alpha}$$

Transformada inversa: Exponencial

- Exemple:

$$f(x) = \frac{1}{9.8} e^{-x/9.8}$$

$$y = F(x) = \int_{-\infty}^{x} \frac{1}{9.8} e^{-x/9.8} dx = 1 - e^{-x/9.8}$$

$$X = F^{-1}(Y) = -9.8 \times \ln(1 - Y)$$

y=Unif(0,1)	x=-9.8 ln(1-y)
0.298	3.472
0.461	6.058
0.593	8.808
0.140	1.481
0.539	7.593
0.762	14.079
0.422	5.365
0.625	9.609
0.395	4.930
0.159	1.702
0.840	17.964
0.784	15.038

Transformada inversa: Bernouilli

- Sigui X una variable aleatòria Bernoulli amb paràmetre *p*. La seva funció de distribució acumulada ve donada per
  - $F_X(x) = 0$ si $x < 0$
  - $F_X(x) = 1 - p$ si $0 \leq x < 1$
  - $F_X(x) = 1$ si $1 \leq x$
- Per tant
  - Si $1 - p < u$ llavors $F_X^{-1}(u) = 1$
  - Sinó $F_X^{-1}(u) = 0$.
- Substituint *u* per $1 - u$ es pot observar que la variable aleatòria de Bernoulli amb paràmetre *p* pot ser generada a partir de una variable aleatòria U(0,1) amb el següent algoritme:
  - Si U < p , llavors retorna 1.
  - Sinó retorna 0.

GNA Mètode d'acceptació - rebuig

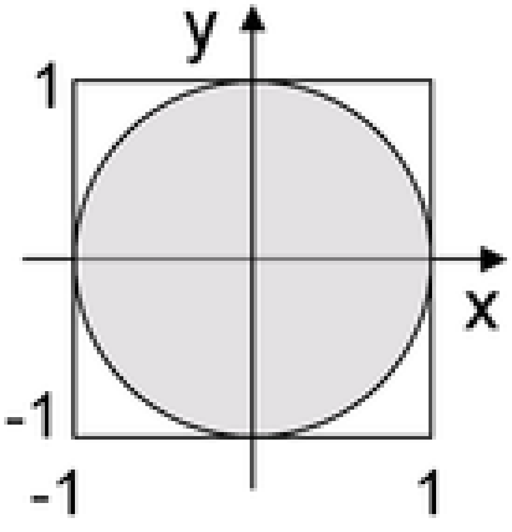
- S'explica el mètode d'acceptació – rebuig per a distribucions contínues.
- Suposem que es vol simular la variable aleatòria X amb densitat f(x).
- És difícil amb el mètode de la transformada inversa.
- Suposem que coneixem un mètode per simular una variable aleatòria amb densitat g(y) tal que:
- $f(y) \leq c \cdot g(y)$
- Per tot y i una constant positiva c, el mètode d'acceptació – rebuig es descriu amb el següent procediment.
  1. Generar Y a partir de la distribució de densitat $g(y)$.
  2. Generar U de U(0,1).
  3. Si $U \cdot c \cdot g(y) \leq f(y)$, retornar X=Y sinó retornar al pas 1

GNA Mètode d'acceptació - rebuig

- El següent teorema mostra que la variable aleatòria X té efectivament densitat f(x).
- **Teorema**: La variable X generada a partir del mètode d'acceptació rebuig té densitat f(x).
- Demostració:
- Cal usar el fet que a partir del procediment descrit
  - $P(X \leq x) = P\{Y \leq x \,|\, U \leq f(y)/(c \cdot g(y))\}$
- Amb *U* i *Y* sent independents.

GNA Mètode d'acceptació - rebuig

- **Exemple:** Es volen generar nombres que pertanyin a una circumferència:
  - *Es generen punts (x,y) que pertanyen a (-1,1).*
    - *En aquest cas c=1, g(y)=* 2u-1, on $u \in [0,1)$.
  - *S'avalua si compleixen la condició:* $x^2+y^2 \leq 1$
  - En cas que la compleixin s'accepten, si no es rebutgen.



GNA Mètode d'acceptació - rebuig

- **Exemple:** Sigui Z una variable aleatòria normal. Suposem que es vol simular el valor absolut $X=|Z|$ que descriu la desviació del seu valor mitjà. La seva densitat de probabilitat ve donada per:
  - $f(x) = (2/\pi)^{1/2} exp(-x^2/2)$
- Es pot agafar *Y* una variable aleatòria exponencial amb paràmetre 1, per tant $g(y) = exp(-y)$. Es fàcil mostrar que
  - $f(y)/g(y) = (2e/\pi)^{1/2} exp(-(y-1)^2/2)$
- Per tant $c = (2e/\pi)^{1/2}$.

GNA Mètode d'acceptació - rebuig

- Òbviament hom vol seleccionar $c{\cdot}g(y)$ tant a prop com sigui possible de $f(y)$ per tal de mantenir la proporció de punts rebutjats petita, però per una altra banda, $g(y)$ es selecciona de forma que tingui una densitat simple per evitar llargs càlculs per Y.

Exemples de GVA

Alguns mètodes per generar les variables aleatòries més comunes.

Uniforme U(a,b)

- Mètode de la transformada inversa
  - $U \in [0,1)$
  - $X = a + (b - a)u$

Exponencial Exp(λ)

- Mètode de la transformada inversa.
- u$\in$[0,1)

$$x = \frac{\ln(1-u)}{-\lambda}$$

Normal N(μ,σ)

- x segueix una normal N(μ,σ)
  - $x = \mu + \sigma\cdot(u_1 + \ldots + u_{12} - 6)$
- On u_k , $k = 1,\ldots,12$ son variables independents U(0,1).

Normal N(μ,σ): Box-Muller

- Un mètode simple és el de Box-Muller basat en una transformació polar:
- Si u_1 i u_2 són variables aleatòries U(0,1) independents, llavors
  - $x_1 = (-2*\ln(u_1))^{½}*\cos(2\pi*u_2)$
  - $x_2 = (-2*\ln(u_2))^{½}*\cos(2\pi*u_1)$
- *Llavors x_1 i x_2* són variables aleatòries N(0,1) independents.

Bernoulli B(1,p) i Binomial B(N,p)

- Com s'ha vist abans una Bernoulli de paràmetre *p* es pot generar d'una U(0,1) seguint el següent algoritme:
  - Si $U < p$, retornar 1.
  - sinó retornar 0.
- Per generar una Binomial B(N,p) s'usa el fet que és la suma de N variables Bernoulli B(1,p) independents
  - $X = 0$
  - Per ($i = 1$ to N) fer
    - $\{X = X + B(1,p)\}$
  - Retornar X

Geomètrica GEOM(1,p)

- Una distribució geomètrica proporciona el temps fins que s'aconsegueix el primer èxit en una seqüència d'experiments de Bernoulli.

- Generar GEOM(1,p):
  - $a = 1/\ln(1-p)$
  - Retornar $X = 1 + \text{floor}[a \times \ln(U)]$

Binomial negativa GEOM (N,p)

- Una variable aleatòria binomial negativa pot ser calculada com la suma de variables geomètriques independents.

- Generar GEOM(N,p):
  - X = 0
  - For(i = 1 to N) {Set X = X +GEOM(1,p)}
  - Retornar X

Poisson (λ)

- El mètode usat es basa en el següent fet: Si els esdeveniments ocorren de forma aleatòria en el temps amb una taxa r i X és el nombre d'esdeveniments que tenen lloc en un període t, llavors X és una distribució de Poisson amb paràmetre $\lambda = r \cdot t$.
- Per tant:

$$X = \min\{n: U1 \cdot U2 \cdot \ldots \cdot Un < exp(-\lambda)\} - 1$$

- o equivalentment

$$X = \max\{n: U1 \cdot U2 \cdot \ldots \cdot Un \geq exp(-\lambda)\}$$

- Generador:
  - Set a = exp(-λ), r = 1, X = -1
  - Mentre > a) {Set r = r·U, X = X + 1)}
  - Retornar X

La parte de imagen con el identificador de relación rId1 no se encontró en el archivo.

Tests de GNA

Validesa dels GNA

Tests o proves de GNA

- En l’actualitat, tots els generadors usats en aplicacions professionals es basen en un **mecanisme determinista** (no son pròpiament aleatoris per natura, per aquest motiu els denominen pseudo-aleatoris).
- No existeix garantia d'aleatorietat en funció d’un mètode determinat **d’elecció de paràmetres** (per exemple els paràmetres a, c i m de un GCL).
- Usant GNA únicament garantim:
  - Un llarg període.
  - L’eficiència aritmètica del procediment.

Tests o proves de GNA

- Las probes sobre la qualitat dels generadores de nombres aleatoris es classifiquen en: (segons Law y Kelton, 1999):
- ***Empíriques***: Són proves de tipus estàtic.
- ***Teòriques***: No són proves en el sentit estadístics. Es basen en els paràmetres numèrics del generador per provar las seva qualitat.

Proves empíriques

- Aquest tipus de proves examina de forma estadística els nombres obtinguts mitjançant un generador per esbrinar quan s'assemblen a uns nombres IID de una U[0,1)
- Es busca si es compleixen les propietats que exigiríem a una tira realment aleatòria.
  - **Uniformitat** de la distribució dels valors de la seqüència.
  - **No correlació** en la seqüència.

Proves empíriques

- La uniformitat pot venir garantida per l'ús d'un generador apropiat amb el que es pugui obtenir un "període complert".
- Els tests processen:
  - n.a.'s $\{u_i\}$ $0 \leq u_i \leq 1$ $i=0,1,2,\ldots$
  - enters $\{y_i\}$ $0 \leq y_i \leq d$ $i=0,1,2,\ldots$

Test de Chi quadrat

- Es divideix el rang [0,1] en ***k intervals iguals***, amb ($k>100$, y $n/k \geq 5$).
- Sigui f_j ($j=1,2,\ldots,k$) la quantitat de nombres aleatoris que cauen en cada un dels intervals j. Si estan ben distribuïts, la freqüència trobada coincidirà amb l'esperada n/k.
- Es calcula el valor de Chi:

$$\chi^2 = k/n \sum_{j=1}^{k} (f_j - n/k)^2$$

- Hipòtesis:
  - H_0 = Independència entre els nombres
  - H_1 = No independència entre els nombres

Exemple test de Chi^2

- Valors obtinguts amb:
- Z_i = 630,360,016 Z_{i-1} (mod 231-1)
- k=2^{12} = 4096, n=2^{15} = 32,768
- S'obté

$$\chi^2 = 4141$$

- Mentre que

$$\chi^2_{4096,0.90} = 4211.4; \alpha = 0.10$$

- Per tant es pot considerar que els nombres obtinguts amb aquest generador són IID U(0,1)

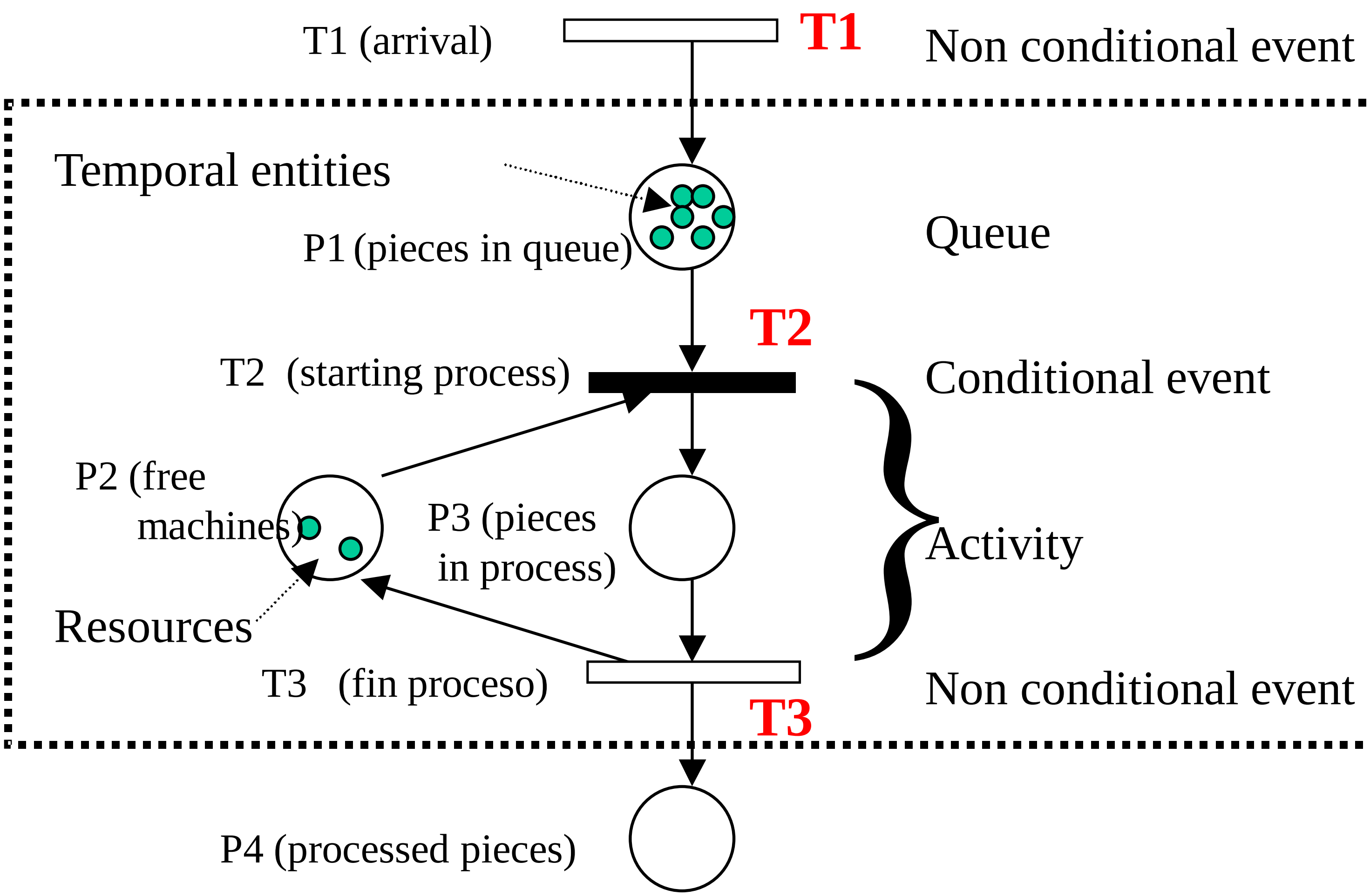
La parte de imagen con el identificador de relación rId1 no se encontró en el archivo.

Simulació discreta per events

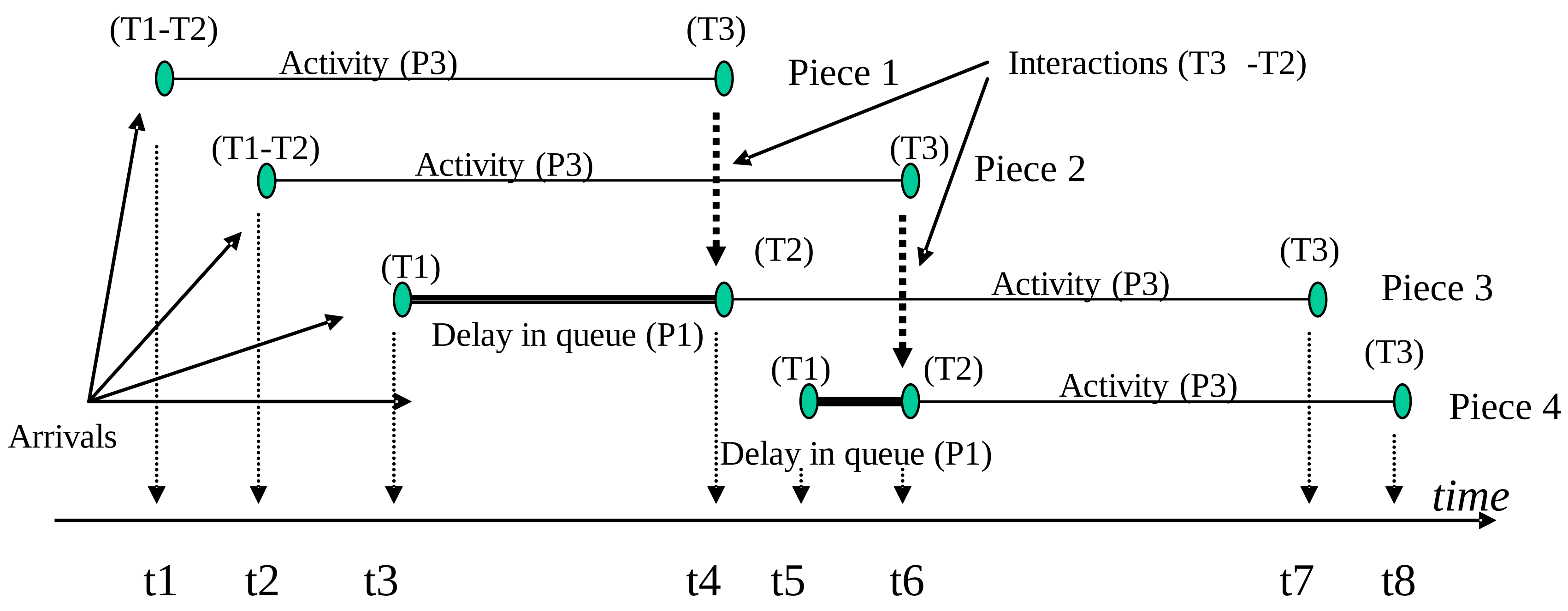
Elements of a simulation model

- Entities
  - Temporal entities
  - Resources or permanent entities
- Events
  - Conditional or not contidional
  - Internal or external
- Attributes
- Activities
- Queues

Elements of a simulation model



Simulation: process oriented



Manual simulation and statistical measures

Arrivals: **expo(10)**

$$Y = F(x) = \begin{cases} 1 - e^{-x/\beta} & x \geq 0 \\ 0 & x < 0 \end{cases}$$

$$x = F^{-1}(Y) = -\beta * \ln(1-Y) = -10 * \ln(1-Y)$$

U[0,1]		0.14	0.76	0.29	0.31	0.86	0.25	0.32	0.14	0.24	0.92
Time between arrivals -10*ln(1-Y)		1.50	14.20	3.40	3.70	19.50	2.90	3.90	1.50	2.70	25.80
Time of arrival	0	1.50	15.70	19.10	22.80	42.30	45.20	49.10	50.60	53.30	79.10

Process time: **unif(15,25)**

$$Y = F(x) = \begin{cases} 0 & x < a \\ \dfrac{x-a}{b-a} & a \leq x \leq b \\ 1 & b < x \end{cases}$$

$$x = F^{-1}(Y) = a + Y(b-a) = 15 + Y * 10$$

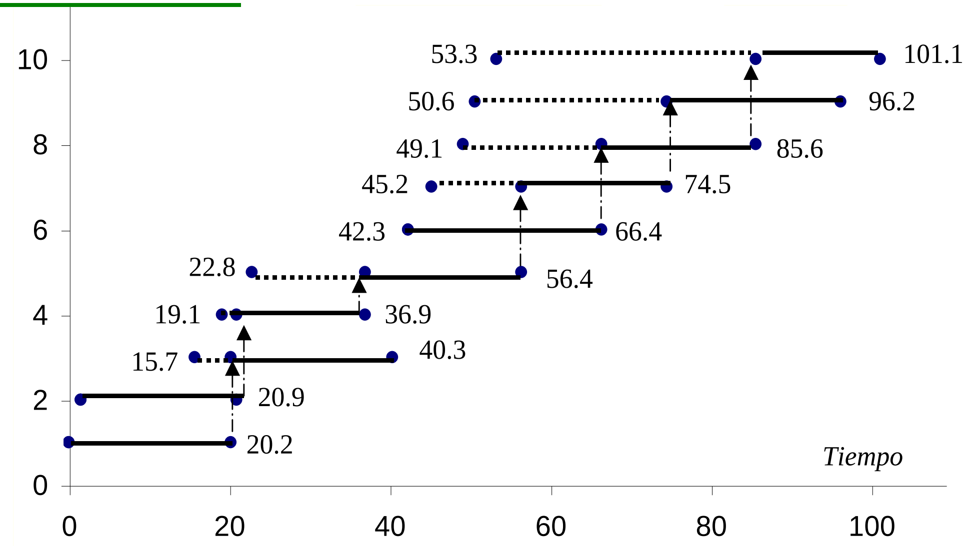
U[0,1]	0.52	0.44	0.51	0.10	0.45	0.91	0.31	0.42	0.67	0.05
Process time o	20.2	19.4	20.1	16	19.5	24.1	18.1	19.2	21.7	15.5

Manual simulation: event oriented

Clock	Events	P1	P2	P3	P4	Time P3 starts (T2)	Duration P3	Time P3 ends (T3)
0	T1-T2	0	1	1	0	0	20.2	20.2
1.5	T1-T2	0	0	2	0	1.5	19.4	20.9
15.7	T1	1	0	2	0			
19.1	T1	2	0	2	0			
20.2	T3-T2	1	0	2	1	20.2	20.1	40.3
20.9	T3-T2	0	0	2	2	20.9	16	36.9
22.8	T1	1	0	2	2			
36.9	T3-T2	0	0	2	3	36.9	19.5	56.4
40.3	T3	0	1	1	4			
42.3	T1-T2	0	0	2	4	42.3	24.1	66.4
45.2	T1	1	0	2	4			
49.1	T1	2	0	2	4			
50.6	T1	3	0	2	4			
53.3	T1	4	0	2	4			
56.4	T3-T2	3	0	2	5	56.4	18.1	74.5
66.4	T3-T2	2	0	2	6	66.4	19.2	85.6

Manual simulation: process oriented

Piece number	T1	T2	Duración P3	T3	Delay in P1 = T2 -T1
1	0	0	20.2	20.2	0
2	1.5	1.5	19.4	20.9	0
3	15.7	20.2	20.1	40.3	4.5
4	19.1	20.9	16	36.9	1.8
5	22.8	36.9	19.5	56.4	14.1
6	42.3	42.3	24.1	66.4	0
7	45.2	56.4	18.1	74.5	11.2
8	49.1	66.4	19.2	85.6	17.3
9	50.6	74.5	21.7	96.2	23.9
10	53.3	85.6	15.5	101.1	32.3



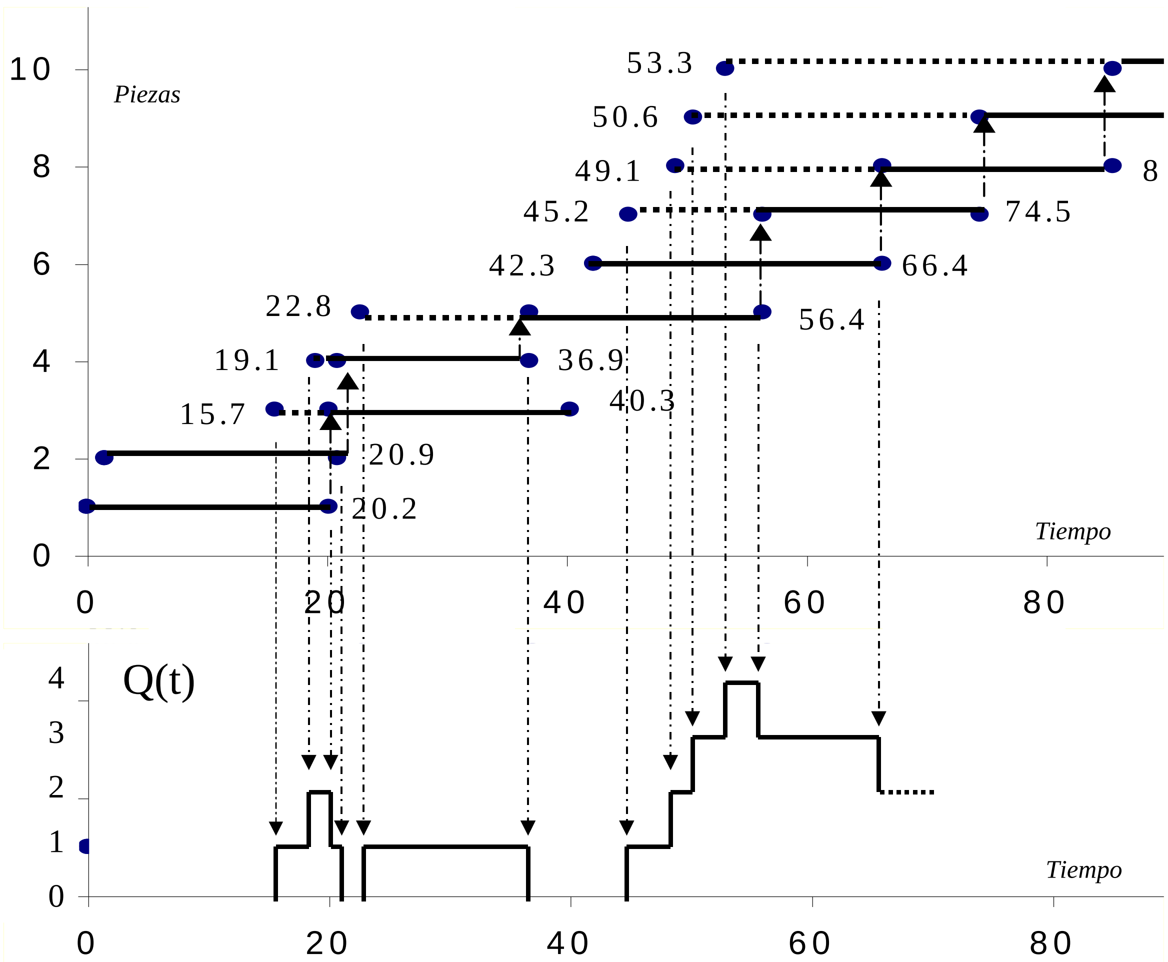
Manual simulation: average delay at queue

Piece number	T1	T2	Duración P3	T3	Delay in P1 = T2 -T1
1	0	0	20.2	20.2	0
2	1.5	1.5	19.4	20.9	0
3	15.7	20.2	20.1	40.3	4.5
4	19.1	20.9	16	36.9	1.8
5	22.8	36.9	19.5	56.4	14.1
6	42.3	42.3	24.1	66.4	0
7	45.2	56.4	18.1	74.5	11.2
8	49.1	66.4	19.2	85.6	17.3
9	50.6	74.5	21.7	96.2	23.9
10	53.3	85.6	15.5	101.1	32.3

$$r(n) = \frac{\sum_{i=1}^{n} R_i}{n}$$

$$r(10) = \frac{\sum_{i=1}^{10} R_i}{n} = \frac{0+0+4.5+1.8+14.1+0+11.2+17.3+23.9+32.3}{10} = 10.5 \text{ minutes}$$

Manual simulation: average size delay



$$q(n) = \frac{\int_0^{T(n)} Q(t)dt}{T(n)}$$

$$q(n) = \frac{\int_0^{T(n)} Q(t)dt}{T(n)} = \frac{\sum_{i=0}^{\infty} iT_i}{T(n)}$$

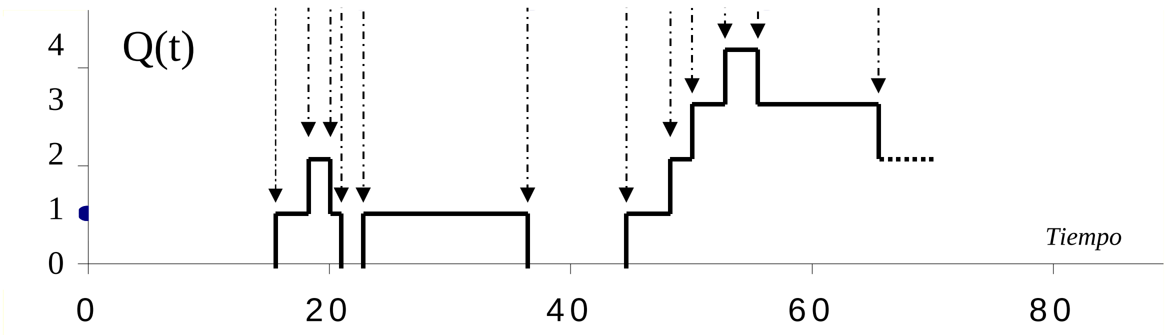
Manual simulation: average queue size delay

$$T_0 = (15.7-0)+(45.2-36.9)=24$$
$$T_1 = (19.1-15.7)+(20.9-20.2)+(36.9-22.8)+(49.1-45.2)=22.1$$
$$T_2 = (20.2-19.1)+(50.6-49.1)=2.6$$
$$T_3 = (53.3-50.6)+(66.4-56.4)=12.7$$
$$T_4 = (56.4-53.3)=3.1$$

$$q(n)=\frac{\sum_{i=0}^{4} iT_i}{T(n)}=\frac{0\times 24+1\times 22.1+2\times 2.6+3\times 12.7+4\times 3.1}{66.4}=1.17$$

$$q(n)=\frac{\int_0^{T(n)} Q(t)dt}{T(n)}$$

$$q(n)=\frac{\int_0^{T(n)} Q(t)dt}{T(n)}=\frac{\sum_{i=0}^{\infty} iT_i}{T(n)}$$



SIMULACIONES CON HORIZONTE FINITO (I)

- La simulación se inicia en un estado específico (por ejemplo, el sistema vacío y el servidor desocupado, en una cola con un único servidor), y se ejecuta hasta que se produce un suceso determinado que identifica el fin de la simulación (por ejemplo el fin de la jornada de trabajo)
- El output no puede alcanzar el estado estacionario $\Rightarrow$ la estimación de cualquier parámetro a partir del output dependerá del transitorio, y por tanto de las condiciones iniciales.
- Sean X_1, X_2,...., X_n los n valores observados durante la simulación:

$$\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i \quad \rightarrow E\left(\overline{X}_n\right) = \mu$$

es por definición un estimador insesgado de μ, pero como las X_i son, en general, variables aleatorias dependientes (por ejemplo, en muchos sistemas de colas las X_i están correlacionadas positivamente), el estimador de la variancia $VAR\left(\overline{X}_n\right)$ está fuertemente sesgado:

$$S_n^2(X) = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \overline{X}_n\right)^2$$

SIMULACIONES CON HORIZONTE FINITO (II)

- Alternativa: ejecutar k replicaciones independientes de la simulación del sistema
- Cada replicación parte del mismo estado y utiliza muestras independientes de números pseudoaleatorios
- Suponiendo que la replicación i-ésima produce los datos de output
- $X_{i1}, X_{i2}, \ldots, X_{in}$
- Entonces las medias muestrales $Y_i = \frac{1}{n}\sum_{j=1}^{n} X_{ij}$
- Son variableas aleatorias IID y $\overline{Y}_k = \frac{1}{k}\sum_{i=1}^{k} Y_i$
- Es también un estimador insesgado de μ, y la variancia muestral de las Yi

$$S_k^2(Y) = \frac{1}{k-1}\sum_{i=1}^{k}\left(Y_i - \overline{Y}_k\right)^2$$

- Es un estimador insesgado de $VAR\left(\overline{X}_n\right)$. Si además n y k son suficientemente grandes, el intervalo deconfianza 1-α para μ es:

$$\overline{Y}_k \pm t_{k-1,1-\alpha/2}\frac{S_k(Y)}{\sqrt{k}}$$