Algoritm pentru generarea T_s

În toate modelele cu coadă de așteptare vom presupune că sosirea clienților se face conform cu un proces Poisson neomogen cu funcția de intensitate $\lambda(t)$ mărginită, $t>0$.

Fie λ o constantă a.î. $\lambda(t) \leq \lambda \ \forall t>0$.

$T_s \rightarrow$ momentul primei sosiri după momentul s.

Generarea lui T_s

1. $t = s$ (ne plasăm în timp la momentul s)
2. Generăm U_1, U_2
3. $t = t - \frac{1}{\lambda} \cdot \log U_1$ (actualizăm timpul curent)
4. Dacă $U_2 \leq \frac{\lambda(t)}{\lambda}$ atunci $T_s = t$ și STOP.
5. Mergi la 2.

Scanned with CamScanner

SISTEM DE TIP COADĂ CU DOUĂ SERVERE LEGATE ÎN SERIE

Variabila de timp: $\boxed{t}$

Variabilele de stare ~~ale~~ sistemului (SS): $\boxed{(n_1, n_2)}$

$\boxed{n_1}$ reprezintă nr. de clienți de la serverul 1 (cei din coadă și cel care este servit)

$\boxed{n_2}$ reprezintă nr. de clienți de la serverul 2.

Variabile contor: $\boxed{N_A} \to$ nr. de sosiri până la momentul t

$\boxed{N_D} \to$ nr. de plecări până la momentul t

Variabile output: $\boxed{A_1(n)} \to$ momentul sosirii clientului n (nu la serverul 1, ci în sistem)

$\boxed{A_2(n)} \to$ momentul sosirii clientului n la serverul 2

$\boxed{D(n)} \to$ momentul plecării clientului n din sistem

Lista de evenimente: $\boxed{t_A} \to$ momentul sosirii următorului client

$\boxed{t_i} \to$ momentul la care clientul curent va finaliza interacțiunea cu serverul i, $i \in \{1, 2\}$

OBS: 1. Dacă nu există client la serverul i facem convenția că $t_i = \infty$, $i \in \{1, 2\}$

2. Lista de evenimente va conține mereu cele 3 variabile: t_1, t_2 și t_A.

Scanned with CamScanner

Schema de simulare

Algoritm de generare pt. T_s

1. $t = s$
2. Generăm U_1, U_2
3. $t = t - \frac{1}{\lambda} \cdot \log U_1$
4. Dacă $U \leq \frac{\lambda(t)}{\lambda}$ $T_s = t$ și STOP
5. Mergi la 2.

$s=0$ | $s=t$

① Inițializare

$t = N_A = N_D = 0$

$SS = (0,0)$

Generăm T_0 și atribuim $t_A = T_0$

$t_1 = t_2 = \infty$

Pentru a actualiza sistemul ne deplasăm pe axa temporală până la următorul eveniment.

În continuare vom considera mai multe cazuri, în funcție de care element din lista de evenimente are valoarea mai mică.

$\boxed{\text{CONVENȚIE}}$: Y_i sunt v.a. cu funcția de repartiție G_i, $i = \overline{1,2}$.

$SS = (n_1, n_2) \to$ starea sistemului

$EL = t_A, t_1, t_2 \to$ lista de evenimente

② Cazul 1 : $\boxed{t_A = \min(t_A, t_1, t_2)}$

↳ sosește un client nou ~~...~~ și verificăm dacă serverul 1 este ~~...~~ liber (altfel trebuie pus în coada de așteptare)

$t = t_A$

$N_A = N_A + 1$

$n_1 = n_1 + 1$

Generăm T_t și apoi $t_A = T_t$

Dacă $n_1 = 1$ atunci generăm Y_1 și $t_1 = t + Y_1$

Output: $A_1(N_A) = t$.

clientul care tocmai a sosit va fi servit imediat de serverul 1

③ Cazul 2 : $\boxed{t_1 < t_A,\ t_1 \leq t_2}$

↳ serverul 1 se eliberează înainte de sosirea unui client nou

$t = t_1$

$n_1 = n_1 - 1$, $n_2 = n_2 + 1$

Dacă $n_1 = 0$ $t_1 = \infty$

Altfel, generăm Y_1 și $t_1 = t + Y_1$

Dacă $n_2 = 1$, generăm Y_2 și $t_2 = t + Y_2$

Output: $A_2(N_A - n_1) = t$

④ Cazul 3 : $\boxed{t_2 < t_A,\ t_2 < t_1}$

↳ serverul 2 se eliberează înainte de a sosi un client nou și înainte de finalizarea activității la serverul 2

$t = t_2$

$N_D = N_D + 1$

$n_2 = n_2 - 1$

Dacă $n_2 = 0$ $t_2 = \infty$

Dacă $n_2 = 1$ generăm Y_2 și $t_2 = t + Y_2$

Output: $\cancel{A_2(N_A - n_1) = t}$ $D(N_D) = t$.

Scanned with CamScanner

SISTEM DE TIP COADĂ CU DOUĂ SERVERE LEGATE ÎN PARALEL

Variabila de timp : $\boxed{t}$

nr. de clienți din sistem

clientul curent la serverul 1

Variabile de stare a sistemului (SS) : $\boxed{(n, i_1, i_2)}$ — clientul curent la serverul 2

Variabile contor : $\boxed{N_A}$ → nr. de sosiri până la momentul de timp t

$\boxed{C_j}$ → nr. de clienți serviți de serverul j, $j \in \{1, 2\}$ până la momentul de timp t

Variabile output : $\boxed{A(n)}$ → timpul la care sosește clientul n

$\boxed{D(n)}$ → timpul la care clientul n părăsește sistemul

Lista de evenimente : $\boxed{t_A, t_1, t_2}$

$\boxed{t_A}$ → timpul la care va sosi următorul client

$\boxed{t_i}$ → timpul la care clientul curent își încheie activitatea la serverul i, $i = \overline{1,2}$

OBS : ① Dacă nu există client la serverul curent (i) atunci $t_i = \infty$, $i = \overline{1,2}$

② Lista de evenimente va conține mereu cele 3 variabile: t_A, t_1, t_2.

Scanned with CamScanner

Schema de simulare

① Inițializare

$t = N_A = C_1 = C_2 = 0$

$SS = (0)$

Generăm T_0 și $t_A = T_0$, $t_1 = t_2 = \infty$

Pentru a actualiza sistemul ne deplasăm pe axa temporală până la momentul apariției următorului eveniment.

În continuare vom considera mai multe cazuri, în funcție de care element din lista de evenimente are valoarea mai mică.

$\boxed{\text{CONVENȚIE}}$: Y_i sunt v.a. cu funcția de repartiție G_i, $i = \overline{1,2}$

$SS = (n, i_1, i_2)$ dacă în sistem există n clienți și la serverul 1 clientul curent este i_1, iar la serverul 2 clientul curent este i_2

ex. $SS = (1, 0, j)$ → singurul client din sistem este clientul j și este servit de serverul 2

$SS = (0)$ → nu avem clienți în sistem

② Cazul 1: $\boxed{SS = (n, i_1, i_2) \text{ și } t_A = \min(t_A, t_1, t_2)}$

↳ sosește un client, verificăm dacă poate fi servit imediat sau intră în coada de așteptare

$t = t_A$

$N_A = N_A + 1$

folosim algoritmul ♡

Generăm T_t și $t_A = T_t$

Output: $A(N_A) = t$

Dacă $SS = (0)$: $SS = (1, N_A, 0)$
Generăm Y_1 și $t_1 = t + Y_1$

Dacă $SS = (1, j, 0)$: $SS = (2, j, N_A)$
Generăm Y_2 și $t_2 = t + Y_2$

Dacă $SS = (1, 0, j)$: $SS = (2, N_A, j)$
Generăm Y_1 și $t_1 = t + Y_1$

Dacă $n > 1$: $SS = (n+1, i_1, i_2)$

③ Cazul 2: $\boxed{SS = (n, i_1, i_2) \text{ și } t_1 < t_A,\ t_1 \le t_2}$

↳ serverul 1 se eliberează înainte de sosirea unui client nou

$t = t_1$

$C_1 = C_1 + 1$

Output: $D(i_1) = t$

Dacă $n = 1$: $SS = (0)$
$t_1 = \infty$

Dacă $n = 2$: $SS = (1, 0, i_2)$
$t_1 = \infty$

Dacă $n > 2$: $m = \max(i_1, i_2)$ Generăm Y_1
$SS = (n-1, m+1, i_2)$ și $t_1 = t + Y_1$

$m+1$ este nr. de ordine al unui client nou

Scanned with CamScanner

UN MODEL DE ASIGURARE DE RISC

Variabila de timp : $\boxed{t}$

Variabile de stare a sistemului (SS) : $\boxed{(n, a)}$ — nr. de clienți ; capitalul curent al firmei

Evenimente : 1) Apariția unui client nou $P = \frac{\nu}{\nu + n\mu + n\lambda}$

2) Pierderea unui client $P = \frac{n\mu}{\nu + n\mu + n\lambda}$

3) Înregistrarea unei cereri de despăgubire $P = \frac{n\lambda}{\nu + n\mu + n\lambda}$

Lista de evenimente : $\boxed{t_E}$ → momentul de timp la care va apărea următorul eveniment (E)

$\boxed{\text{OBS}}$: Lista de evenimente conține doar valorile ce reprezintă t_E!

Comentariu: Dacă (n, a) reprezintă starea sistemului la momentul t, cum minimul unor v.a. independente repartizate exponențial este tot o v.a. repartizată exponențial, atunci rezultă că momentul de timp la care se produce următorul eveniment va fi egal cu $t + X$, unde $X \sim Exp(\nu + n\mu + n\lambda)$

$\boxed{\text{Obs}}$: În dreptul celor 3 tipuri de evenimente am notat probabilitățile cu care se produc respectivele evenimente!

Variabila output: $I = \begin{cases} 1, & \text{dacă firma are capital pozitiv în intervalul } [0, t] \\ 0, & \text{altfel} \end{cases}$

Scanned with CamScanner

Schema de simulare

(1) Inițializare

$t = 0$

$a = a_0$

$n = n_0$

Generăm X și $t_E = X$

Pentru a actualiza sistemul ne deplasăm de-a lungul axei temporale până la apariția următorului eveniment, verificând în prealabil dacă acesta are loc înaintea momentul T.

(2) Cazul 1 : $\boxed{t_E > T}$

$I = 1$ și STOP

(firma nu se ruinează)

(3) Cazul 2 : $\boxed{t_E \leq T}$

$a = a + n \cdot c(t_E - t)$

$t = t_E$

– continuare (3) –

avem 3 tipuri de evenimente

Generăm Y :

- Dacă $Y = 1$ $\quad n = n+1$
- Dacă $Y = 2$ $\quad n = n-1$
- Dacă $Y = 3$:
  - Generăm Y
  - Dacă $Y > a$ $\quad I = 0$ STOP
  - Altfel $a = a - Y$

folosim metoda inversă pentru v.a. discrete

Generăm X și $t_E = t + X$

Obs. Actualizarea sistemului se reia până la finalul intervalului de timp pentru care vrem să facem simularea

Comentarii: După determinarea momentului la care are loc următorul eveniment generăm o valoare aleatoare (Y) pentru a vedea tipul de eveniment ce va avea loc.

$$Y : \begin{pmatrix} 1 & 2 & 3 \\ \frac{\nu}{\nu + n\mu + n\lambda} & \frac{n\mu}{\nu + n\mu + n\lambda} & \frac{n\lambda}{\nu + n\mu + n\lambda} \end{pmatrix}$$

$Y \to$ v.a. cu funcția de repartiție F ce reprezintă valoarea despăgubirii solicitate.

Scanned with CamScanner

④ Cazul 3: $\boxed{SS = (n, i_1, i_2) \text{ și } t_2 < t_A,\ t_2 < t_1}$

↳ serverul 2 se eliberează înaintea serverului 1 și înainte de sosirea unui client nou

$t = t_2$

$C_2 = C_2 + 1$

Output: $D(i_2) = t$

Dacă $n = 1$: $SS = (0)$

$t_2 = \infty$

Dacă $n = 2$: $SS = (1, i_1, 0)$

$t_2 = \infty$

Dacă $n > 2$: $m = \max(i_1, i_2)$

$SS = (n-1, i_1, m+1)$

Generăm Y_2 și $t_2 = t + Y_2$

Scanned with CamScanner