

# Generarea unor variabile pa

*Curs 7*

# Generarea unor variabile discrete

## 1. Probe Bernoulli

Fie  $A$  un eveniment aleator observabil cu probabilitate constantă  $p = P(A) > 0$ . În urma unui experiment fie se realizează evenimentul contrar lui  $A$ ,  $\bar{A}$ . Un astfel de numește *probă Bernoulli*. Presupunem că atunci când are loc un *succes*, iar când se realizează  $\bar{A}$  are loc un *eșec*.

Asociem unei probe Bernoulli variabila aleatoare  $Z$  a succes și  $Z = 0$  în caz de eșec:

$$Z : \begin{pmatrix} 0 & 1 \\ q & p \end{pmatrix}$$

unde  $q = 1 - p$ . Observăm că

$$E[Z] = p, \quad Var[Z] = pq = p(1 - p)$$

Funcția de repartiție a variabilei  $Z$  este:

$$F(x) = \begin{cases} 0 & \text{adcă } x < 0 \\ q & \text{adcă } x \in [0, 1] \\ 1 & \text{adcă } x > 1 \end{cases}$$

De aici rezultă următorul algoritm de generare a variabilei  $Z$  prin metoda inversă, cazul discret.

### **Algoritm Bernoulli**

**Intrare:**  $p$

P1: Se generează  $U \sim U(0, 1)$ ;

P2: Dacă  $U \leq q$  atunci  $Z = 0$ , altfel  $Z = 1$ .

**Ieșire:** Variabila aleatoare  $Z$ .

## 2. Repartiția binomială

Fie  $X$  o variabilă aleatoare discretă,  $n \in \mathbb{N}^*$ ,  $p \in [0, 1]$  repartiție binomială  $Binom(n, p)$  dacă

$X =$  nr. de succese în  $n$  probe Bernoulli în

Observăm că:

$$X = \sum_{i=1}^n Z_i,$$

unde  $Z_i, i = 1, \dots, n$  sunt variabile Bernoulli independente

Observăm că:

$$P(X = x) = \binom{n}{x} p^x q^{n-x}$$

Adică  $P(X = x)$  este termenul general al dezvoltării de unde derivă și denumirea de repartiție binomială.

Funcția caracteristică a variabilei binomiale este:

$$\varphi_X(t) = E[e^{itX}] = E[e^{it \sum_{j=1}^n Z_j}] = (q + pe^{it})^n$$

de unde rezultă că:

$$E[X] = np, \quad Var[X] = npq$$

Din (2) rezultă următorul algoritm de generare a variabilei binomiale.

### **Algoritm Binom1**

**Intrare:**  $n, p$

P1:  $i = 1, X = 0;$

P2: Se generează  $Z_i \sim Bern(p), X := X + Z_i;$

P3: Dacă  $i = n$  stop, altfel  $i := i + 1,$

**Ieșire:** Variabila aleatoare  $X$ .

O altă metodă de generare a variabilei binomiale rezu  
*centrală*: pentru  $n \rightarrow \infty$  variabila:

$$W_n = \frac{X - np}{\sqrt{npq}}$$

este repartizată normal  $N(0, 1)$ . De aici se deduce urm  
pentru  $n$  mare:

### **Algoritm Binom2**

**Intrare:**  $n, p$

P1: Se generează  $W \sim N(0, 1)$ ;

P2: Se ia  $X =$  cel mai apropiat nr.  
 $np + W\sqrt{npq}$ ;

**Ieșire:** Variabila aleatoare  $X$ .

### 3. Repartiția Pascal

Fie  $X$  o variabilă aleatoare discretă,  $k \in \mathbb{N}^*$ ,  $p \in [0, 1]$  repartiția  $Pascal(k, p)$  dacă:

$X =$  nr. de eșecuri până la apariția a  $k$  succese dintr-o serie de probe Bernoulli independente.

#### Observație 1.

$$P(X = x) = \binom{x + k - 1}{k - 1} p^k q^x$$

*este termenul general al dezvoltării în serie al expresiei  $(1 - q)^{-k}$  (unde  $q = 1 - p$ ). Această repartiție Pascal se mai numește și repartiție binomială negativă).*

**Dem:**

$$\frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots + (-1)^k$$

$$\left(\frac{1}{1+x}\right)^{(r)} = \sum_{k=r}^{\infty} (-1)^k k(k-1)\dots(k-r)$$

$$= \sum_{j=0}^{\infty} (-1)^{r+j} \frac{(r+j)!}{j!} x^j$$

**Dar**

$$\left(\frac{1}{1+x}\right)^{(r)} = \frac{(-1)^r r!}{(1+x)^{r+1}}$$

Din (3) și (4) rezultă că:

$$\frac{(-1)^r r!}{(1+x)^{r+1}} = \sum_{j=0}^{\infty} (-1)^{r+j} \frac{(r+j)!}{j!}$$



Notând  $r + 1$  cu  $n$  avem:

$$(1 + x)^{-n} = \sum_{j=0}^{\infty} (-1)^j \binom{n + j - 1}{j}$$

Notând

$$(-1)^j \binom{n + j - 1}{j} = \binom{-n}{j}$$

avem următoarea dezvoltare în serie a binomului cu  $n$  negativ:

$$(1 + x)^{-n} = \sum_{j=0}^{\infty} \binom{-n}{j} x^j.$$

Acum aplicăm (5) pentru  $p^k(1 - q)^{-k}$ :

$$p^k(1 - q)^{-k} = p^k \sum_{j=0}^{\infty} (-1)^j \binom{k + j - 1}{j} (-q)^j = \sum_{j=0}^{\infty}$$

Observăm că termenul  $j$  al acestei dezvoltări în serie probabilitate a unei variabile  $Pascal(k, p)$ , calculată în

Repartiția *Pascal* este stabilă și:

$$E[X] = \frac{kq}{p}, \quad Var[X] = \frac{kq}{p^2}$$

Din definiție rezultă următorul algoritm de generare a *Pascal*:

### **Algoritm Pascal**

**Intrare:**  $k, p, j=0$

P1: Se generează  $Y \sim Bernoulli(p)$ ;

P2: Dacă  $Y = 0$  atunci  $X := X + 1$ , alt

P3: Dacă  $j=k$  stop, altfel mergi la

**Ieșire:** Variabila aleatoare  $X$ .

## 4. Repartiția geometrică

Este un caz particular de repartiție *Pascal*:  $k = 1$ . Atunci probabilitatea devine pentru  $k = 1$ :

$$P(X = x) = pq^x, \quad x = 0, 1, 2, \dots$$

care este termenul unei progresii geometrice (de aici și denumirea acestei distribuții).

Observăm că:

$$F(x) = P(X \leq x) = \sum_{i=0}^x pq^i = 1 - q^{x+1}, \quad x = 0, 1, 2, \dots$$

și

$$E[X] = \frac{q}{p}, \quad Var[X] = \frac{q}{p^2}.$$

Simularea variabilei geometrice se poate realiza fie prin  
algoritmului de generare a variabilei Pascal, fie prin m

### **Algoritm Geometrică**

**Intrare:**  $p, q = 1 - p;$

P1: Se generează  $U \sim U(0, 1);$

P2:  $X := \left\lceil \frac{\log U}{\log q} \right\rceil - 1.$

**Ieșire:** Variabila aleatoare  $X.$

unde  $[a]$  reprezintă partea întreagă a lui  $a.$

## 5. Repartiția hipergeometrică

Considerăm următorul experiment aleator: fie o urnă albă și  $B$  bile negre, cu  $A + B = N$ . Presupunem că se face o extracție, fără întoarcerea bilei extrase în urnă. Atunci

$X =$  numărul de bile albe extrase

este o variabilă *hipergeometrică*.

Fie  $E_A$  evenimentul care constă în extragerea unei bile albe și  $E_B$  evenimentul care constă în extragerea unei bile negre. Probabilitățile de extragere avem:

$$p = P(E_A) = \frac{A}{N}, \quad P(E_B) = \frac{B}{N}$$

Probabilitățile de extragere a unei bile albe sau negre sunt condiționate de rezultatele primei extrageri:

$$P(E_A|E_A) = \frac{A-1}{N-1}, \quad P(E_A|E_B) =$$

$$P(E_B|E_A) = \frac{B}{N-1}, \quad P(E_B|E_B) =$$

La fiecare extragere compoziția urnei se modifică și p  
extrage o bilă albă sau neagră depinde de extragerile a

Variabila hipergeometrică se notează cu  $H(N, p, n)$  cu  
Atunci  $A$  poate fi considerat cel mai apropiat nr. între  
Probabilitatea ca în extrageri succesive fără întoarcere  
albe este:

$$P(X = x) = \frac{\binom{A}{x} \binom{B}{n-x}}{\binom{N}{n}}, \quad 0 \leq a \leq n,$$

Media și dispersia variabilei hipergeometrice sunt:

$$E[X] = np, \quad Var[X] = np(1 - p) \frac{N}{N-1}$$

Din definiție rezultă următorul algoritm de generare a unei variabile aleatoare hipergeometrice:

### **Algoritm Hipergeometrică**

**Intrare:**  $A, B, n, N = A + B, p = A/N,$

P1: Se generează  $U \sim U(0,1)$ ;

P2: Dacă  $U < p$  (s-a extras o bilă albă) mergi la P3, altfel (s-a extras o bilă neagră) mergi la P4;

P3:  $X := X + 1, S := 1$ , mergi la P5;

P4:  $S = 0$ ;

P5:  $N := N - 1, A := A - S, p := \frac{A}{N}$ ;

P6: Dacă  $j = n$  stop, altfel mergi la P2;

**Ieșire:** Variabila aleatoare  $X$ .

## 6. Repartiția Poisson

Variabila aleatoare discretă  $X \in \mathbb{N}$  are repartiția *Poisson*

$$P(X = x) = \frac{\lambda^x}{x!} e^{-\lambda}$$

Funcția caracteristică a variabilei *Poisson*( $\lambda$ ) este:

$$\varphi_X(t) = e^{\lambda(e^{it} - 1)}$$

De aici rezultă că:

$$E[X] = \lambda, \quad Var[X] = \lambda.$$

Repartiția *Poisson*( $\lambda$ ) este folosită pentru a modela numărul de evenimente care apar într-un interval de timp. Parametrul  $\lambda$  reprezintă numărul mediu de evenimente dintr-un anumit interval de timp.



**Lemă 1.** Dacă  $E_1, E_2, \dots$  sunt variabile aleatoare rep este cel mai mic întreg astfel încât

$$\sum_{i=1}^{X+1} E_i > \lambda$$

atunci variabila  $X$  este repartizată Poisson( $\lambda$ ).

**Dem:**

Din condiția care se pune asupra lui  $X$  rezultă că dacă  
 loc  $\sum_{i=1}^{k+1} E_i > \lambda$  și nu are loc  $\sum_{i=1}^k E_i > \lambda$ . Prin urm

$$P(X = k) = P\left(\sum_{i=1}^{k+1} E_i > \lambda\right) - P\left(\sum_{i=1}^k E_i > \lambda\right) = \int_{\lambda}^{\infty} f_{k+1}(y) - f_k(y) dy$$

unde  $f_k(y)$  este densitatea unei variabile aleatoare  $E_i$

Rezultă că:

$$\begin{aligned} P(X = k) &= \int_{\lambda}^{\infty} \left[ \frac{1}{k!} y^k e^{-y} - \frac{1}{(k-1)!} y^{k-1} e^{-y} \right] dy \\ &= \frac{1}{k!} \left[ \int_{\lambda}^{\infty} y^k e^{-y} dy - k \int_{\lambda}^{\infty} y^{k-1} e^{-y} dy \right] \end{aligned}$$

Dacă integrăm prin părți prima integrală obținem:

$$\begin{aligned} P(X = k) &= \frac{1}{k!} \left[ \lambda^k e^{-\lambda} + k \int_{\lambda}^{\infty} y^{k-1} e^{-y} dy - k \int_{\lambda}^{\infty} y^{k-1} e^{-y} dy \right] \\ &= \frac{\lambda^k}{k!} e^{-\lambda} \end{aligned}$$

Ținând cont că variabilele  $E_i$  sunt exponențiale  $Exp()$  generate prin metoda inversă cu ajutorul relației  $E_i = -\ln U_i$ , atunci condiția (9) se scrie:

$$\prod_{i=1}^{X+1} U_i < e^{-\lambda}$$

și rezultă următorul algoritm de generare a unei variabile aleatoare Poisson.

### **Algoritm Poisson1**

**Intrare:**  $\lambda, i := 0, P = 1;$

P1: Se generează  $U \sim U(0,1), i := i + 1,$

P2: Dacă  $P \geq e^{-\lambda}$  atunci mergi la P1, la P3;

P3:  $X := i - 1.$

**Ieșire:** Variabila aleatoare  $X$ .

O altă metodă de generare pentru o variabilă *Poisson* următoarea proprietate a variabilei  $Poisson(\lambda)$ :

dacă  $Y \sim Poisson(\lambda)$  cu  $\lambda = np$  și  $n \rightarrow \infty, p \rightarrow 0$ , atunci  $Y \sim Binom(n, p)$ .

Acest lucru se poate demonstra folosind funcția caracteristică a repartiției.

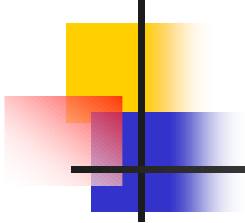
### **Algoritm Poisson2**

**Intrare:**  $\lambda$ , se alege  $p \approx 0$ , de exemplu

P1: Se determină  $n = \lceil \lambda/p \rceil$  cel mai apropiat număr întreg de  $\lambda/p$ ;

P2: Se generează  $X \sim Binom(n, p)$ ;

**Ieșire:** Variabila aleatoare  $X$ .



# Tehnici de simulare – Curs 1

[florentina.suter@yahoo.com](mailto:florentina.suter@yahoo.com)

[florentina.suter@g.unibuc.ro](mailto:florentina.suter@g.unibuc.ro)

Yahoo groups: [simulare2009](#), comentarii: nume,  
prenume, grupa



## Bibliografie:

---

- Ion Văduva (2004) *Modele de simulare*, Editura Universității din București;
- Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol (2005) *Discrete-Event System Simulation*, Pearson Education, Inc., Upper Saddle River, New Jersey.
- Sheldon M. Ross (1997) *Simulation*, Academic Press, San Diego.
- Christopher Chung (2004) *Simulation Modeling Handbok – A practical Approach*, CRC Press, Boca Raton.



# Introducere

---

- Exemple:

- Traficul într-o intersecție – timpul de staționare;
- Bancă – număr de ghișee;
- Mic magazin: farmacie
  - Un farmacist se gândește să-și deschidă o nouă farmacie care să funcționeze între 9:00 și 17:00. În medie știe că îi sosesc 32 de clienți în acest interval orar. Durata servirii unui client este aleatoare cu o medie de 10 minute și o deviație standard de 4 minute. El plănuiește să nu mai accepte clienți după ora 17:00, dar să-i servească pe cei care sunt deja în farmacie. Întrebări:
    - Care este timpul mediu zilnic petrecut în farmacie?
    - Care este timpul mediu de servire a unui client de la sosirea lui în farmacie?
    - Câți clienți sunt serviți pe oră?
    - Dacă își schimbă politica de servire, câți clienți ar fi pierduți?

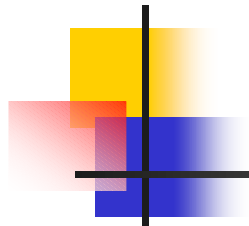


# Introducere

---

- Exemple:
  - Mic magazin: farmacie
    - Pentru a raspunde la intrebari trebuie stabilite anumite ipoteze si creat un model
    - Model probabilist:
      - cu ce frecventa vin cei 32 de clienti?
      - Care este distributia timpului de servire?
    - Teoretic acest model se poate rezolva analitic, dar in practica, datorita complexitatii lui, nu este asa
    - Raspunsurile vor fi aflate prin simulare: este programat mecanismul probabilist folosind "numerele aleatoare" si programul este rulat pentru un numar mare de zile astfel incat utilizand teoria statistica se pot da estimari ale raspunsurilor. De exemplu, pentru 1000 de zile simulate exista 150 in care farmacistul inca lucreaza la ora 17:30.





# Introducere

---

SIMULARE: imitare a lumii reale

- Domeniu interdisciplinar ce folosește Matematica, Statistica și Informatica;
- John von Neumann;
- Los Alamos, 1944: noi domenii de matematici aplicate care necesită utilizarea calculatorului:
  - Cercetări operaționale;
  - Teoria jocurilor;
  - Simulare (Metoda Monte Carlo): imitarea aleatorului.



# Introducere

---

**Definiție:** Un sistem este o colecție de componente care interacționează. Sistemul primește date de intrare și furnizează date de ieșire.

**Definiție:** Simularea este o tehnică de realizare a experimentelor cu calculatorul care implică utilizarea unor modele matematice și logice care descriu comportarea unui sistem real (sau a unor componente ale sale) de-a lungul unor perioade mari de timp.

## Simularea

- Completează teoria matematică;
- Completează studiul unor experimente fizice.
- Instrument de analiza si instrument de proiectare.



# Introducere

---

- Aplicații ale simulării:
  - Analizarea comportării utilajelor industriale
  - Transporturi;
  - Operații desfășurate în aeroporturi;
  - Spitale;
  - Construcții;
  - Ingineria spațială;



# Introducere

---

- Scopurile simulării:
  - Se pot obține informații despre ceea ce se petrece în interiorul sistemelor complexe;
  - Se pot dezvolta strategii de îmbunătățire a performanțelor sistemului;
  - Se pot testa noi concepte și sisteme înainte de implementare;
  - Se pot obține informații fără a interveni în evoluția sistemului fizic.



# Introducere

---

- Avantajele simulării
  - Scurtarea timpului de experimentare;
  - Studiul dinamic al sistemelor;
  - Evoluția modelelor este ușor de urmărit (folosirea animației)
- Dezavantajele simulării:
  - Simularea nu poate da rezultate precise dacă datele de intrare nu sunt precise;
  - Simularea nu poate furniza rezultate simple la probleme complexe;
  - Simularea furnizează soluții posibile, nu rezolvă problema.



# Introducere

---

- Tipuri de simulare:
  - **Metoda Monte Carlo:** o simulare statică (fără axă a timpului) care folosește numerele aleatoare pentru a modela fenomene probabiliste care nu își schimbă caracteristicile în timp sau pentru a evalua expresii matematice, al căror rezultat nu poate fi obținut prin metode analitice (integrale, ecuații sisteme de ecuații).
  - **Simularea bazată pe traiectorie:** traiectoria este o înregistrare ordonată în timp, a evenimentelor care au loc într-un sistem real.
    - Este folosită în analizarea funcționării sistemelor de calcul;
    - Analizarea și îmbunătățirea algoritmilor de gestionare a resurselor;
    - Algoritmi de organizare a operațiilor unui procesor;
    - Algoritmi de prevenire a blocajelor



# Introducere

---

- Folosește un model al sistemului bazat pe stări discrete (există și simulare bazată pe evenimente continue în care stările sistemului se modifică continuu în timp - folosită mai ales în chimie, biologie, medicină).
- Timpul este continuu sau discret.

Model de simulare: model matematic + algoritm;

Modelul de simulare: trebuie să descrie corect evoluția sistemului și să permită efectuarea de experiențe (prin rulări ale algoritmului pe calculator) care să înlocuiască experiențele pe sistemul real.



# Model matematic

---

Model matematic: reprezintă realitatea folosind elemente sau noțiuni abstracte.

Elemente constitutive ale unui model matematic:

- Variabile și parametri;
- Relații funcționale;
- Caracteristici operative;
- Tehnica de rezolvare;
- Scopul modelului matematic este de a exprima variabilele și parametrii de ieșire în funcție de variabilele și parametrii de intrare.





# Model matematic

---

- Tehnica de rezolvare:

- Tehnică matematică care realizează exprimarea elementelor de ieșire în funcție de elementele de intrare:

$$(VE, PE) = f(VI, PI);$$

- De cele mai multe ori sunt necesare ipoteze simplificatoare;
- Uneori problema nu poate fi rezolvată prin tehnici de rezolvare.



# Model matematic

---

Clasificări ale modelelor matematice:

- După tipul variabilelor: modele continue sau discrete, modele statice sau dinamice, modele deterministe sau stocastice;
- După structura determinată de părțile modelului: modele cu o componentă sau cu mai multe componente.



# Model matematic

---

## **Exemplu:**

Sistemul de așteptare: parte a lumii reale în care se produc aglomerări.

Componentă:

- Resurse: una sau mai multe stații de servire care servesc
- Entități: clienții care sosesc în sistem și care formează
- Cozi de așteptare

Scop: realizarea unui echilibru între pierderile datorate așteptării clienților și pierderile datorate lenevirii stațiilor de servire.



# Model matematic

---

Teoria matematică a cozilor sau teoria așteptării cu aplicații în:

- Economie;
- Comunicații și transport;
- Rețele de calculatoare.

Un model matematic de așteptare:

- Variabile de intrare cunoscute VI:

AT= timpul între sosiri succesive ale clienților

ST=timpul de servire a unui client

sau

NA=numărul de clienți sosiți în unitatea de timp

NS=numărul de clienți serviți în unitatea de timp;



# Model matematic

---

- Variabile de ieșire necunoscute VE:
  - WT=timp de așteptare
  - WL=lungimea cozii
  - sau
  - TID=timp de lenevire
  - NID=numărul de stații care lenevesc



# Model matematic

---

Scopul modelului este realizat astfel:

- cunoscând repartițiile de probabilitate ale AT (NA) și ST (NS) se determină informații despre WT (WL) sau TID (NID) și se stabilesc condițiile pe care trebuie să le îndeplinească ST pentru ca o anumită funcție de cost să fie optimă.
- se studiază procesul stocastic discret  $N(t)$  = numărul de clienți din sistemul de așteptare la momentul  $t$  - proces de naștere și deces



# Algoritmul modelului de simulare

---

Utilizarea calculatorului -- îmbunătățirea performanțelor modelelor matematice prin aplicarea metodelor numerice și ale simulării.

Componentele sistemului din punct de vedere al simulării:

- entitati
  - attribute
  - activitati
  - stari
  - Evenimente
- 
- Exemplu: activitatea intr-o banca: etitati – clientii, atribut – suma depusa in cont, activitate – crearea unui depozit, evenimente – sosirea, plecarea unui client, variabile de stare – numarul de functionari ocupati, numarul de clienti care asteapta.



# Algoritmul modelului de simulare

---

Structura algoritmică a unui model de simulare (care depinde de timp) are două concepte de baza:

- Ceasul simulării;
- Agenda simulării;





# Algoritmul modelului de simulare

---

Eveniment: modificarea valorilor uneia sau mai multor variabile care se calculează sau se generează prin instrucțiunile modelului.

Ceasul simulării: asigură eșalonarea corectă în timp a evenimentelor create de model și uneori determină terminarea simulării.

Ceas cu

- creștere constantă;
- creștere variabilă.



# Algoritmul modelului de simulare

---

Creșterea variabilă: se face cu valoarea care corespunde apariției primului eveniment următor.

Creșterea constantă: se face cu o cuantă de timp constantă.

Terminarea simulării: se impune condiția ca ceasul să ajungă la un  $T_{max}$  sau să se fi prelucrat un anumit număr de evenimente.

Agenda simulării: organizează prelucrarea evenimentelor.

- Agenda evenimentelor curente AEC: evenimentele cu timpul de apariție valoarea curentă a ceasului;
- Agenda evenimentelor viitoare AEV: evenimentele cu timpul de apariție mai mare decât valoarea curentă a ceasului



# Algoritmul modelului de simulare

---

Prelucrarea unui eveniment (în algoritmul simulării): determinarea apariției unui nou eveniment (care se memorează în AEC), modificarea unei stări sau distrugerea unui eveniment (ștergerea) din agendă.

Algoritmul simulării:

1. Se inițializează ceasul cu valoarea 0;
2. Se selectează din agendă evenimentele care fac parte din AEC;
3. Se prelucrează evenimentele din AEC până când aceasta devine vidă. Dacă este îndeplinită condiția de oprire algoritmul se termină, altfel
4. Se crește ceasul simulării și se reia pasul 2.



# Metodologia simulării

---

Etapele realizării unui experiment de simulare:

- Formularea problemei prin precizarea:
  - întrebărilor la care trebuie să răspundă modelul;
  - domeniului lumii reale ce trebuie analizat;
  - formei răspunsului la întrebări (grafice, tabele, rapoarte).
- Realizarea unor experimente preliminare (dacă sunt posibile): pe baza observațiilor și a datelor se stabilesc variabilele și parametri de intrare sau de ieșire;
- Prelucrarea (interpretarea) primară a datelor preliminare:
  - Se disting variabilele aleatoare;
  - Se estimează parametri;
  - Se testează ipotezele statistice;



# Metodologia simulării

---

- Formularea unui model matematic preliminar
  - Se precizează relații funcționale și ipoteze de lucru;
  - Se identifică relațiile care nu pot fi exprimate matematic și dificultățile care trebuie înlăturate;
- Evaluarea modelului:
  - Evaluarea complexității modelului (dacă poate răspunde în timp real și complet la întrebări);
  - Revizuirea răspunsurilor din etapele precedente prin simplificări sau completări.
- Construcția modelului de simulare
  - Scrierea unui algoritm detaliat care să cuprindă cazul cel mai general al problemei;



# Metodologia simulării

---

- Se va ține cont de limbajul în care se va programa algoritmul: limbaj specializat pentru simulare sau nu.
  - Folosirea unui limbaj de simulare (GPSS, SIMUB, SIMULA, Arena):
    - Modelele se construiesc rapid;
    - Experiențele se desfășoară repede;
    - Au implementate entități specifice simulării cum ar fi ceasul și agenda simulării;
    - Nu sunt foarte flexibile;
    - Nu se poate controla foarte bine ce se întâmplă în interiorul modelului.
  - Folosirea unui alt tip de limbaj:
    - Rezultate precise și controlabile;
    - Se construiesc mult mai greu.



# Metodologia simulării

---

- Verificarea si validarea modelului.
- Planificarea experiențelor de simulare (experimental design).  
Ce fel de experiente sunt facute: lungimea etapei de initializare, lungimea rularilor si numarul de repetari ale fiecărei rulari.
- Prelucrarea și interpretarea experiențelor de simulare prin rularea programului și determinarea valorilor statisticilor construite cu ajutorul valorilor de selecție obținute.
- Relizarea documentatiei si a rapoartelor.



# Elemente de implementare

---

## Implementarea agende de evenimente

- Necesitatea organizării evenimentelor: agenda evenimentelor. Cum este ea implementată?
- Agenda simulării: o listă ordonată înlănțuită a evenimentelor. Fiecare element al listei trebuie să memoreze timpul de apariție a evenimentului și să aibă un pointer către codul care trebuie executat la acel moment de timp.
- Operații frecvente: inserarea unui nou eveniment și găsirea și eliminarea evenimentului care a avut loc.
- Alegerea structurii de date folosite pentru memorarea acestei liste afectează timpul de rulare.





# Elemente de implementare

---

- Liste ordonate dublu înlănțuite (GPSS): primul element din listă este următorul cel mai recent eveniment. Inserția se face căutând locul potrivit pentru noul eveniment.
- Liste indexate: mulțimea evenimentelor viitoare este împărțită în mai multe submulțimi. Fiecare submulțime este asociată unui anumit interval de timp de lungime  $\Delta t$  din timpul total de simulare. Un vector de indici, asociază fiecărui indice  $i$  lista cu evenimentele programate în intervalul  $[(i-1)\Delta t, i\Delta t)$
- Structuri arborescente: arbori binari de sortare cu rădăcina fiind cel mai recent eveniment din listă.



# Verificarea și validarea modelului

---

- Implementare corectă: verificare;
- Ipoteze corecte: validare;

Modelul de simulare: program de dimensiuni mari.

- Verificarea modelului este facilitată de:
  - Proiectarea modulară “top-down”: modelul dezvoltat într-o structură ierarhică în care programul este format dintr-o serie de module care comunică prin interfețe bine stabilite;
  - Includerea de verificări pe parcursul rulării programului și determinarea de rezultate parțiale;
- Verificarea de face prin:
  - Rularea de cazuri simplificate;
  - Rularea programului pentru valori ale parametrilor care diferă foarte puțin (test de continuitate);



# Verificarea și validarea modelului

---

- Rularea programului pentru cazuri extreme (testul valorilor degenerate);
- Verificarea dacă modelul produce rezultate asemănătoare pentru aceleași date de intrare (test de consistență);
- Verificarea independenței de valoarea de plecare a generatorului de numere aleatoare;
- Validarea se face pentru:
  - ipoteze;
  - valori și distribuții ale parametrilor de intrare;
  - valorile de ieșire și concluziicu ajutorul
  - intuiției expertului;
  - măsurărilor asupra sistemului real;
  - rezultatelor teoretice.



## Terminarea simulării

---

- Criteriile de oprire trebuie să țină cont de faptul că:
  - o durată prea scurtă implică rezultate imprecise;
  - o durată prea lungă implică irosirea resurselor de calcul.
- Trebuie să ia în considerare și observațiile rezultate:
  - independente si cu o anumita repartitie.



## Erori care se pot face în simulare

---

- **Nivel de detaliere neadecvat:** simularea permite ca sistemul să fie studiat în detaliu, nivelul de detaliu fiind limitat doar de timpul alocat simulării. Detalierea nu determină neapărat calitatea modelului de simulare
  - Probabilitatea de eroare crește;
  - Lipsa de informație precisă despre parametrii de intrare;
  - Necesitatea unui timp prea îndelungat pentru a obține rezultate.
- **Limbaj de programare nepotrivit.**
- **Modele neverificate:** programele care implementează un model de simulare sunt în general mari și atunci probabilitatea de eroare crește.



## Erori care se pot face în simulare

---

- **Modele imprecise:** programul modelului de simulare poate să nu reprezinte în mod corect sistemul simulat din cauza unor ipoteze greșite asupra comportării sistemului. Un model de simulare trebuie confirmat de modele analitice, observații sau intuiție.
- **Prelucrarea incorectă a condițiilor inițiale:** de obicei rezultatele inițiale ale simulării nu sunt relevante pentru evoluția sistemului.
- **Durata prea scurtă a rulării modelelor de simulare:** dacă simularea nu se face un timp suficient de îndelungat, rezultatele pot depinde prea mult de condițiile inițiale și pot să fie irelevante pentru evoluția unui sistem real. Durata corectă a simulării este dată de precizia dorită și de dispersia mărimilor observate.



## Erori care se pot face în simulare

---

- **Generatoare de numere aleatoare neperformante.**
- **Alegerea unor valori nepotrivite** de plecare pentru generatorii de numere aleatoare.

# Generatori de numere aleatoare

## *Curs 2*



# Bibliografie suplimentară:

- Knuth, D. E.(1983) *Tratat de programare a calculatoarelor, Vol. 2 - Algoritmi seminumerici*, Editura Tehnică.
- Knuth D.E. (1974) *Tratat de programare a calculatoarelor, Vol 1 - Algoritmi fundamentali*, Editura Tehnică.
- Văduva, I (1977) *Modele de simulare cu calculatorul*, Ed. Tehnică.

# Recapitularea unor noțiuni probabiliste

**Experiment aleator**= un experiment al cărui rezultat nu este cunoscut înainte.

**Spațiu de selecție (S)**= spațiul tuturor rezultatelor posibile.

**Eveniment (A)**= orice submulțime a spațiului de selecție. Dacă rezultatul unui experiment aparține lui A, atunci se spune că a avut loc A.

**Probabilitate**= probabilitatea de apariție a evenimentului A  $P(A)$  este un număr care:

- $0 \leq P(A) \leq 1$
- $P(S) = 1$
- $P(\bigcup_{i=1}^n A_i) = \sum_{i=1}^n P(A_i), \forall n, A_i \cap A_j = \phi$

$$\Rightarrow 1 = P(S) = P(A \cup A^c) = P(A) + P(A^c)$$

**Probabilitate condiționată**=  $P(A|B) = \frac{P(A \cap B)}{P\{B\}}$

**Evenimente independente**=  $P(A \cap B) = P(A)P(B)$

# Recapitularea unor noțiuni probabiliste

**Variabilă aleatoare**  $X : S \rightarrow \mathbb{R}$  (discrete și continue)

**Funcție de repartiție**  $F(x) = P\{X \leq x\}$

**Funcție și densitate de probabilitate**  $p(x) = P\{X = x\}$ ,  $f(x) = F'(x)$ ,

$$P\{X \in C\} = \int_C f(x)dx, F(a) = \int_{-\infty}^a f(x)dx$$

**Două variabile aleatoare**  $F(x, y) = P\{X \leq x, Y \leq y\}$ ,

$$p(x, y) = P\{X = x, Y = y\},$$

$$P\{X \in C, Y \in D\} = \int \int_{x \in C, y \in D} f(x, y)dx dy$$

**Variabile aleatoare independente**

$$P\{X \in C, Y \in D\} = P\{X \in C\}P\{Y \in D\},$$

$$P\{X = x, Y = y\} = P\{X = x\}P\{Y = y\}, f(x, y) = f_X(x)f_Y(y)$$

# Numere aleatoare

**Șir de numere aleatoare:** (definiție intuitivă) un sir de numere alese la întâmplare astfel încât se cunoaște probabilitatea de apariție a fiecărui număr într-o succesiune de valori dată.

Șirurile de numere aleatoare au aplicații în: criptografie, simulare, etc.

În general pentru șirurile de numere aleatoare probabilitatea de apariție a unei valori corespunde **repartiției uniforme**.

**Repartiția uniformă:** (intuitiv) Toate valorile sunt egal probabile.

În **simulare**: Șiruri de numere aleatoare  $\Rightarrow$  Valori ale variabilelor aleatoare  $\Rightarrow$  Model de simulare

*Numere aleatoare:* Valori ale unor variabile aleatoare uniforme pe  $[0, 1]$ .

Șirurile de numere aleatoare pot fi obținute din:

- tabele - greu de implementat;
- fenomene fizice (cea mai bună sursă de aleator, de exemplu: intervalele de timp dintre apăsarea unei taste și mișcarea mouse-ului)  
- nu se pot refolosi;
- algoritmi.

# Algoritmi de numere aleatoare

Se bazează pe utilizarea unei **valori inițiale (sămânță)** și a unei **relații de recurență** cu ajutorul căreia se obțin celelalte valori ale șirului.

Numere **pseudo-aleatoare**: numerele obținute nu sunt chiar aleatoare pentru că se bazează pe o relație de recurență. Numerele produse trebuie să aibă două proprietăți statistice importante:

- uniformitate;
- independență.

Majoritatea algoritmilor generează  $X_n$  numere întregi între 0 și  $m - 1$  (de obicei  $m - 1$  este valoarea maximă a tipului întreg memorat în calculator) și apoi se iau:

$$U_n = \frac{X_n}{m}$$

uniforme pe  $[0, 1]$ .

# Algoritmi de numere aleatoare

Trebuie să aibă următoarele **proprietăți**:

- Rapiditate;
- Portabilitate de diverse calculatoare;
- Șirul de numere produs
  - să aibă o perioadă mare;
  - să fie cât mai aproape de independență și uniformitate;
  - să fie reproductibil.

## Metoda părții din mijloc a pătratului

- Prima metodă propusă pentru a fi implementată pe calculatoare. A fost descrisă de John von Neumann în 1946.
- Are doar interes istoric.
- Presupunem că vrem să generăm numere aleatoare cu cel mult  $i$  cifre. Atunci:

$$X_{n+1} = \text{cele } i \text{ cifre din mijloc ale lui } X_n^2$$

# Algoritmi de numere aleatoare

- Șirul tinde să se stabilizeze în scurte cicluri de elemente.
- De exemplu: 43, 84, 05, 02, 00, 00,... (se pun 0-uri în fața numerelor care nu au patru sau două cifre).

## Metoda Fibonacci

- Doar interes istoric;
- Se bazează pe relația

$$X_{n+1} = (X_n + X_{n-1}) \mod m;$$

- numerele produse nu sunt destul de aleatoare.

Alte metode:

- metoda regiștrilor de translație (shift register), metode combinate.

# Metoda congruențială liniară

- Este folosit cel mai frecvent.
- Se bazează pe relația:

$$X_{n+1} = (aX_n + c) \mod m \quad (1)$$

unde

- $m > 0$  = modulul;
  - $a$  = multiplicatorul;
  - $c$  = incrementul;
  - $X_0$  = termenul inițial.
- Fie  $b = a - 1$ . Se presupune  $a \geq 2$  și  $b \geq 1$ , pentru că pentru  $a = 0$  și  $a = 1$  nu se obțin șiruri aleatoare.
  - Din (1), pentru  $k \geq 0$  și  $n \geq 0$ , rezultă că:

$$X_{n+k} = (aX_{n+k-1} + c) \mod m$$

și prin urmare relația dintre termenii șirului aflați la distanța  $k$  este:



$$\begin{aligned}
X_{n+k} &= [a(aX_{n+k-2} + c) \bmod m + c] \bmod m \\
&= [a^2 X_{n+k-2} + (a+1)c] \bmod m \\
&= [a^3 X_{n+k-3} + (a^2 + a + 1)c] \bmod m \\
&= \dots \\
&= [a^k X_n + (a^{k-1} + a^{k-2} + \dots + a + 1)c] \bmod m \\
&= \left[ a^k X_n + \frac{a^k - 1}{a - 1} c \right] \bmod m
\end{aligned}$$

relație utilă pentru alegerea valorilor care caracterizează șirul.

# Alegerea modulului

- $m$ : trebuie să fie suficient de mare și să asigure o complexitate scăzută a calculului.
- o alegere convenabilă a lui  $m$  ar fi  $w =$  dimensiunea cuvântului calculatorului ( $= 2^x$  unde  $x$  este numărul de biti pe care sunt construiți regiștrii calculatorului).
- alte alegeri:  $m = w \pm 1$  sau  $m =$  cel mai mare număr prim mai mic decât  $w$ .

# Alegerea multiplicatorului

Se alege  $a$  pentru un  $m$  oarecare astfel încât pentru orice valoare a lui  $X_0$  să rezulte un generator de perioadă maximă.

**Teorema 1.** *Șirul congruențial liniar definit de  $m$ ,  $a$ ,  $c$ , și  $X_0$  are perioada de lungime maximă  $m$  dacă și numai dacă:*

- 1.  $c$  și  $m$  sunt două numere întregi prime între ele;*
- 2.  $b = a - 1$  este un multiplu de  $p$ , pentru orice număr prim  $p$  care-l divide pe  $m$ .*
- 3.  $b$  este multiplu de 4 dacă  $m$  este multiplu de 4.*

Datorită următoarei leme este suficientă demonstrarea teoremei pentru  $m$  putere a unui număr prim.

**Lemă 1.** *Fie descompunerea lui  $m$  în factori primi:*

$$m = p_1^{e_1} \dots p_t^{e_t}. \quad (2)$$

*Lungimea  $\lambda$  a perioadei șirului congruențial liniar definit de  $(X_0, a, c, m)$  este cel mai mic multiplu comun al lungimilor  $\lambda_j$  ale perioadelor șirurilor congruențiale liniare  $(X_0 \bmod p_j^{e_j}, a \bmod p_j^{e_j}, c \bmod p_j^{e_j}, p_j^{e_j})$ ,  $1 \leq j \leq t$ .*

De aici rezultă că este suficientă demonstrarea teoremei pentru  $m$ , putere a unui număr prim:

$$p_1^{e_1} \dots p_t^{e_t} = \lambda = \text{c.m.m.m.c}\{\lambda_1, \dots, \lambda_t\} \leq p_1^{e_1} \dots p_t^{e_t} \quad (3)$$

iar această relație poate avea loc dacă și numai dacă  $\lambda_j = p_j^{e_j}$  pentru  $\forall j$ ,  $1 \leq j \leq t$ . De aceea se poate presupune  $m = p^e$ , unde  $p$  este un număr prim iar  $e$  este un număr întreg pozitiv.

Perioada poate avea lungime  $m$  dacă și numai dacă orice număr întreg din  $[0, m)$  apare în cadrul perioadei o singură dată. Dacă luăm  $X_0 = 0$ , atunci:

$$X_n = \left( \frac{a^n - 1}{a - 1} \right) c \mod m$$

Dacă  $c$  și  $m$  nu sunt prime între ele, atunci în acest șir nu poate exista 1. Prin urmare condiția 1. din teoremă este necesară. Demonstrarea teoremei se reduce la demonstrarea următoarei leme:

**Lemă 2.** Presupunem că  $1 < a < p^e$ , cu  $p$  număr prim. Dacă  $\lambda$  este cel mai mic număr întreg pozitiv pentru care

$$\frac{a^\lambda - 1}{a - 1} \equiv 0 \pmod{p^e}$$

atunci

$$\lambda = p^e$$

dacă și numai dacă:

- pentru  $p = 2$   $a \equiv 1 \pmod{4}$ ;
- pentru  $p > 2$   $a \equiv 1 \pmod{p}$ .

Această leamnă se demonstrează aplicând de mai multe ori următoarea leamnă:

**Lemă 3.** *Fie  $p$  un număr prim și fie  $e$  un număr întreg pozitiv cu  $p^e > 2$ .  
Dacă*

$$x \equiv 1 \pmod{p^e}, \quad x \not\equiv 1 \pmod{p^{e+1}} \quad (4)$$

*atunci*

$$x^p \equiv 1 \pmod{p^{e+1}}, \quad x^p \not\equiv 1 \pmod{p^{e+2}} \quad (5)$$

# Generatorul multiplicativ congruențial

Este un generator liniar congruențial cu  $c = 0$ :

$$X_{n+1} = aX_n \mod m \quad (6)$$

Observăm că  $X_n$  și  $m$  trebuie să fie prime între ele, pentru că altfel generatorul ar deveni un șir de 0. Prin urmare lungimea perioadei poate fi maxim  $\varphi(m)$ , numărul numerelor întregi cuprinse între 0 și  $m$ , prime cu  $m$ .

Putem să presupunem din nou că  $m = p^e$  cu  $p = \text{nr. prim}$  și  $e$  întreg pozitiv. Avem:

$$X_n = a^n X_0 \mod p^e$$

Dacă  $a$  este multiplu de  $p$ , atunci perioada are lungime 1  $\Rightarrow a$  trebuie să fie prim cu  $p$ .



Testarea șirurilor de numere aleatoare se face cu **teste statistice**:

- Teste de frecvență (testează repartiția uniformă pe care trebuie să o aibă numerele): testul  $\chi^2$ , testul Kolmogorov-Smirnov.
- Teste de independență.

# Generarea variabilelor neuniforme

*Curs 3*

# Introducere

Fie  $X$  o variabilă aleatoare.

Generarea v.a  $X$  = găsirea unui număr  $n$  (mare) de valori pe care le poate lua  $X$ .

Cum se găsește o astfel de valoare?

Presupunem că  $S_1, S_2, \dots, S_n$  sunt v.a. pentru care se cunosc metode de generare (de exemplu variabilele uniforme pe  $[0, 1]$  se generează cu ajutorul generatorilor de numere aleatoare, generatori care sunt deja implementați în majoritatea limbajelor de programare).

Atunci o valoare a lui  $X$  se poate determina găsiind o relație între  $X$  și  $S_1, S_2, \dots, S_n$ .

Algoritm efectiv de generare a lui  $X$  = aplicarea de  $n$  ori a metodei furnizate de relația dintre  $X$  și  $S_1, S_2, \dots, S_n$ .

Fiecare dintre algoritmi prezentați în curs se referă la o astfel de metodă.

# Metoda inversă - Cazul continuu

Fie  $U$  o variabilă aleatoare uniformă pe  $[0, 1]$ , cu densitatea de repartiție  $f(x)$  și funcția de repartiție  $F(x)$ .

$$f(x) = \begin{cases} 1, & \text{dacă } x \in [0, 1] \\ 0, & \text{în rest} \end{cases}, \quad F(x) = \begin{cases} 0, & \text{dacă } x < 0 \\ x, & \text{dacă } x \in [0, 1] \\ 1, & \text{dacă } x > 1 \end{cases}. \quad (1)$$

**Propoziție 1.** Variabila aleatoare  $U$  este uniformă pe  $[0, 1]$  dacă și numai dacă variabila  $1 - U$  este uniformă pe  $[0, 1]$ .

Dem:

Fie  $x < 0$ . Atunci:

$$P\{1 - U \leq x\} = P\{U \geq 1 - x\} = 1 - P\{U < 1 - x\} = 1 - 1 = 0$$

Fie  $x \in [0, 1]$ . Atunci:

$$P\{1 - U \leq x\} = P\{U \geq 1 - x\} = 1 - P\{U < 1 - x\} = 1 - (1 - x) = x$$

Fie  $x > 1$ . Atunci:

$$P\{1 - U \leq x\} = P\{U \geq 1 - x\} = 1 - P\{U < 1 - x\} = 1 - 0 = 1$$

**Teorema 1.** (Hincin) Fie  $X$  o variabilă aleatoare continuă cu funcția de repartiție  $F$ . Atunci variabila aleatoare  $F(X)$  este uniformă pe intervalul  $[0, 1]$ , iar  $F^{-1}(U)$  are funcția de repartiție  $F$ .

Dem (a doua parte):

Fie  $x \in \mathbb{R}$ . Atunci:

$$P\{F^{-1}(U) \leq x\} = P\{F(F^{-1}(U)) \leq F(x)\} = P\{U \leq F(x)\} = F(x)$$

### Algoritm M-inversă

**Intrare:** Inversa funcției de repartiție  $F$ :  $F^{-1}$ .

P1: Se generează  $U$  variabilă uniformă pe  $[0, 1]$ ;

P2:  $X = F^{-1}(U)$ ;

**Ieșire:**  $X$  cu funcția de repartiție  $F(x)$

Observăm că dacă în expresia care definește funcția  $F^{-1}$  din algoritmul M-Inversă, apare  $1 - U$ , atunci, datorită Propoziției 1,  $1 - U$  poate fi înlocuită direct cu  $U$ .

**Exemplu 1.** Fie  $X$  o variabilă  $\text{Exp}(\lambda)$  cu densitatea și funcția de repartiție:

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & \text{dacă } x \geq 0; \\ 0, & \text{în rest} \end{cases}, \quad F(x) = \begin{cases} 1 - e^{-\lambda x}, & \text{dacă } x \geq 0; \\ 0, & \text{în rest} \end{cases}$$

Atunci:

$$F^{-1}(U) = -\frac{1}{\lambda} \ln(1 - U)$$

iar algoritmul de generare prin metoda inversă este:

### **Algoritm M-inversă-Exp**

**Intrare:** Parametrul  $\lambda$ .

P1: Se generează  $U$  variabilă uniformă pe  $[0,1]$ ;

P2:  $X = -\frac{1}{\lambda} \ln(U)$ ;

**Ieșire:**  $X$  cu funcția de repartiție  $F(x)$

# Cazul discret

Fie  $X$  o variabilă aleatoare discretă cu repartiția:

$$X : \begin{pmatrix} a_1, & a_2, & \dots & a_m \\ p_1, & p_2, & \dots & p_m \end{pmatrix} \quad \text{cu} \quad \sum_{i=1}^m p_i = 1$$

Funcția de repartiție a lui  $X$  va lua valorile:

$$F(x) = \begin{cases} 0, & \text{dacă } x < a_1 \\ p_1, & \text{dacă } a_1 \leq x < a_2 \\ p_1 + p_2, & \text{dacă } a_2 \leq x < a_3 \\ \dots & \\ p_1 + p_2 + \dots + p_k, & \text{dacă } a_k \leq x < a_{k+1} \\ \dots & \\ 1, & \text{dacă } a_m \leq x \end{cases}$$



Algoritmul constă în găsirea valorii  $a_i$  astfel încât  $F(a_i) = U$ , unde  $U$  este o variabilă uniformă pe  $[0, 1]$ .

Fie  $s_i = \sum_{j=1}^i p_j$ . Observăm că:

$$P\{s_{i-1} < U \leq s_i\} = F_U(s_i) - F_U(s_{i-1}) = p_i = P\{X = a_i\}$$

### Algoritm M-inversă-Discret

**Intrare:**  $s_i = \sum_{j=1}^i p_j$  și  $a_i$ ,  $i = 1, 2, \dots, m$ .

P1: Se generează  $U$  variabilă uniformă pe  $[0, 1]$ ;

P2:  $i = 1$ ;

P3: Dacă  $U \leq s_i$   $X = a_i$  STOP. Altfel mergi la P4.

P4:  $i := i + 1$ , mergi la P3;

**Ieșire:**  $X$  cu funcția de repartiție  $F(x)$

**Exemplu 2.** *Simularea unei variabile aleatoare Bernoulli Z:*

$$Z : \begin{pmatrix} 0, & 1 \\ q, & p \end{pmatrix} \quad \text{cu } p + q = 1$$

*Z are funcția de repartiție:*

$$F(x) = P(Z \leq x) = \begin{cases} 0, & \text{dacă } x < 0 \\ q, & \text{dacă } x \in [0, 1) \\ 1 & \text{dacă } x \geq 1 \end{cases}$$

*Un algoritm de generare a unei variabile Bernoulli este:*

**Intrare:** Parametrul  $p$ ,  $q = 1 - p$ .

P1: Se generează  $U$  variabilă uniformă pe  $[0, 1]$ ;

P2: Dacă  $U \leq q$   $Z = 0$ . Altfel  $Z = 1$ .

**Ieșire:**  $Z$  cu funcția de repartiție  $F(x)$ .

# Metoda compunerii sau a amestecării

Cazul discret

**Definiție 1.** *Funcția de repartiție este o amestecare (sau compunere sau mixtură) discretă a mulțimii de funcții de repartiție  $\{F_i(x)\}_{1 \leq i \leq m}$  cu repartiția discretă*

$$J : \begin{pmatrix} 1, & 2, & \dots & m \\ p_1, & p_2, & \dots & p_m \end{pmatrix} \quad \text{cu} \quad \sum_{i=1}^m p_i = 1 \quad (2)$$

dacă

$$F(x) = \sum_{i=1}^m p_i F_i(x). \quad (3)$$

Relația (3) poate fi scrisă și în funcție de densitățile de repartiție:

$$f(x) = \sum_{i=1}^m p_i f_i(x). \quad (4)$$

Fie  $X$  variabila aleatoare cu funcția de repartiție  $F(x)$  și  $X_i$  variabila aleatoare cu funcția de repartiție  $F_i(x)$ .

### **Algoritm compunere discretă**

**Intrare:** Repartiția (2), familia de funcții  $\{F_i(x)\}_{1 \leq i \leq m}$

P1: Generează  $J$  cu repartiția (2);

P2: Generează  $X_J$  cu funcția de repartiție  $F_J(x)$ ;

P3:  $X = X_J$ .

**Ieșire:**  $X$  cu funcția de repartiție  $F(x)$

**Exemplu 3.** Presupunem că la o stație de benzină sosesc  $m$  tipuri de mașini și se cunoaște  $p_i$  probabilitatea să sosească un automobil de tipul  $i$ ,  $1 \leq i \leq m$ . Presupunem că timpul  $X_i$  între sosirile autoturismelor de tipul  $i$  este distribuit exponențial de parametru  $\lambda_i$ . Atunci timpul dintre două sosiri oarecare are o repartiție mixt exponențială, adică este o amestecare discretă cu densitatea:

$$f(x) = \begin{cases} 0, & \text{dacă } x < 0 \\ \sum_{i=1}^m p_i \lambda_i e^{-\lambda_i x}, & \text{dacă } x \geq 0 \end{cases}$$

**Exemplu 4.** Fie  $X$  variabila aleatoare cu repartiția Laplace( $\lambda$ ) a cărei densitate este:

$$f(x) = \frac{\lambda}{2} e^{-\lambda|x|}; \quad x \in \mathbb{R}, \quad \lambda > 0$$

Atunci, putem să scriem

$$f(x) = p_1 f_1(x) + p_2 f_2(x)$$

cu

$$p_1 = p_2 = \frac{1}{2}$$

și

$$f_1(x) = \begin{cases} \lambda e^{\lambda x} & \text{dacă } x \leq 0 \\ 0 & \text{dacă } x > 0 \end{cases} \quad f_2(x) = \begin{cases} 0 & \text{dacă } x \leq 0 \\ \lambda e^{-\lambda x} & \text{dacă } x > 0 \end{cases}$$

Prin urmare un algoritm de generare al variabilei Laplace( $\lambda$ ) poate fi:

**Algoritm LAPLACE**

**Intrare:** parametrul  $\lambda$ .

P1: Se generează  $U$  variabilă uniformă pe  $[0, 1]$ ;

P2: Dacă  $U \leq 0.5$  atunci  $s := -1$ , altfel  $s = 1$ ;

P3: Generează  $Y \sim \text{Exp}(\lambda)$ ;

P4:  $X := sY$ .

**Ieșire:**  $X$  cu funcția de repartiție  $F(x)$

$S$  se numește semn aleator.

Variabila Laplace se poate simula ușor și cu metoda inversă.

Metoda compunerii discrete se poate aplica pentru orice densitate de repartiție:

**Teorema 2.** *Fie  $X$  o variabilă aleatoare cu densitatea de repartiție  $f(x)$ ,  $x \in \Delta \subseteq \mathbb{R}$ . Fie o diviziune a lui  $\Delta$  de forma  $\Delta = \cup_{i=1}^m \Delta_i$ , cu  $\Delta_i \cap \Delta_j = \emptyset$ ,  $\forall i \neq j$ . Notând cu  $p_i = P(X \in \Delta_i) > 0$ , există densitățile  $f_i(x)$ , care iau valoarea 0 pentru  $x \notin \Delta_i$  astfel încât*

$$f(x) = \sum_{i=1}^m p_i f_i(x). \quad (5)$$

Dem:

$p_i = \int_{\Delta_i} f(x) dx \Rightarrow$  funcțiile definite astfel:

$$f_i(x) = \begin{cases} \frac{f(x)}{p_i} & \text{dacă } x \in \Delta_i \\ 0, & \text{dacă } x \notin \Delta_i \end{cases}$$

sunt densități de repartiție.



Fie un  $x \in \Delta$ , oarecare, cu  $f(x) \neq 0$ . Atunci există un  $i$ ,  $1 \leq i \leq m$  astfel încât  $x \in \Delta_i$ . Atunci avem:

$$f(x) = \frac{f(x)}{p_i} p_i = p_i f_i(x) = \sum_{j=1}^m p_j f_j(x)$$

pentru că  $f_j(x) = 0, \forall j \neq i$ .

# Cazul continuu

**Definiție 2.** *Funcția de repartiție  $F(x)$  este o amestecare continuă a familiei de funcții de repartiție  $\{G(x, Y)\}_{Y \in \mathbb{R}}$  cu funcția de repartiție continuă  $H(y)$  a lui  $Y$  dacă ea este de forma:*

$$F(x) = \int_{\mathbb{R}} G(x, y) dH(y) \quad (6)$$

*unde ultima integrală este integrala Stieltjes.*

Relația (5) poate fi scrisă și în funcție de densitățile de repartiție:

$$f(x) = \int_{\mathbb{R}} g(x, y) h(y) dy. \quad (7)$$

### **Algoritm COMP-CONT**

**Intrare:** Funcțiile de repartiție  $H$  și  $G$ .

P1: Se generează  $Y$  cu funcția de repartiție  $H(y)$ ;

P2: Se generează  $Z_Y$  cu funcția de repartiție  $G(x, Y)$ ;

P3:  $X = Z_Y$

**Ieșire:**  $X$  cu funcția de repartiție  $F(x)$

**Exemplu 5.** Fie  $X > 0$  o v.a. care reprezintă durata în funcționare a unui aparat. Presupunem că  $X$  este o variabilă exponențială de parametru  $\eta\lambda$ , unde  $\lambda > 0$  este un parametru care reprezintă o caracteristică aparatului, iar  $\eta$  este un parametru aleator care indică influența mediului în care lucrează aparatul. Presupunem că  $\eta$  este la rândul ei o variabilă aleatoare și că are densitatea de repartiție:

$$h(\eta) = \begin{cases} \frac{b^a}{\Gamma(a)} \eta^{a-1} e^{-b\eta}, & \text{dacă } \eta \geq 0; \\ 0 & \text{dacă } \eta < 0 \end{cases} . \quad (8)$$

Observăm că  $X$  se obține ca o amestecare continuă a unei familii de variabile exponențiale după o distribuție Gama. Densitatea de repartiție a variabilei  $X$  are forma:

$$f(x) = \int_0^\infty \eta \lambda e^{-\lambda \eta x} \frac{b^a}{\Gamma(a)} \eta^{a-1} e^{-b\eta} d\eta = \frac{\lambda b^a}{\Gamma(a)} \int_0^\infty \eta^a e^{-\eta(\lambda x + b)} d\eta =$$

$$= \frac{\lambda b^a \Gamma(a+1)}{\Gamma(a)(\lambda x + b)^{a+1}} = \frac{\lambda a}{b} \frac{b^{a+1}}{(\lambda x + b)^{a+1}} = \frac{a\theta}{(\theta x + 1)^{a+1}}$$

cu

$$\theta = \frac{\lambda}{b}$$

Deci densitatea lui  $X$  este:

$$f(x) = \begin{cases} \frac{a\theta}{(\theta x + 1)^{a+1}}, & \text{dacă } x \geq 0 \\ 0 & \text{dacă } x < 0 \end{cases} \quad (9)$$

Variabila cu densitatea (8) se numește variabilă Lomax, iar algoritmul ei de generare (presupunând ca se cunoaște o metodă de generare a unei variabile Gama) se poate scrie astfel:

## Algoritm Lomax

**Intrare:** Parametrii  $\lambda$ ,  $a$  și  $b$ .

P1: Se generează  $Y$  cu repartiția  $\text{Gama}(a, b)$ ;

P2: Se generează  $Z_Y$  cu funcția de repartiție  $\text{Exp}(x, Y)$ ;

P3:  $X = Z_Y$

**Ieșire:**  $X$  cu densitatea de repartiție (8).

# Metoda respingerii

Mai poate fi numită metoda acceptării-respingerii.

Fie  $X$  o variabilă aleatoare pe care vrem să o generăm cu metoda respingerii și fie următoarele elemente cunoscute:

- Un procedeu de generare a unei variabile aleatoare  $N$  cu valori întregi pozitive;
- Procedee de generare a unor variabile aleatoare  $S_i \in \mathcal{S}$ ,  $i \geq 1$ , unde  $\mathcal{S}$  este o familie de variabile aleatoare dată;
- Un predicat  $\mathcal{P}(S_1, S_2, \dots, S_n)$  care se poate calcula simplu;
- Funcția  $\Psi$ , astfel încât
$$X = \Psi(\{S_1, S_2, \dots, S_n\}, \mathcal{P}(S_1, S_2, \dots, S_n) = \text{true})$$

Atunci forma generală a unui algoritm de respingere este:

## Algoritm RESPINGERE

**Intrare:**  $N, S, \mathcal{P}(S_1, S_2, \dots, S_n), \Psi$ .

P1: Se generează  $N$ ;

P2: Se generează  $S_1, S_2, \dots, S_n$  din  $S$ ;

P3: Dacă  $\mathcal{P}(S_1, S_2, \dots, S_n) = \text{true}$  atunci  $X = \Psi(S_1, S_2, \dots, S_n)$  și STOP, altfel mergi la P1;

**Ieșire:** Variabila aleatoare  $X$ .

Observăm că

- Dacă  $\mathcal{P}(S_1, S_2, \dots, S_n) = \text{false}$  atunci mulțimea de variabile aleatoare  $\{S_1, S_2, \dots, S_n\}$  se respinge, de aici provenind numele de “metoda respingerii”.
- Dacă  $p_a = P(\mathcal{P}(S_1, S_2, \dots, S_n) = \text{true})$ , numită și probabilitate de acceptare, este mare, atunci algoritmul este “bun”, altfel algoritmul este prea lent

Trei algoritmi de respingere bazați pe trei teoreme:



# Prima teoremă de respingere

**Teorema 3.** *Fie  $X$  o variabilă aleatoare cu densitatea de repartiție  $f(x)$  pentru  $x \in \mathbb{R}$ . Fie  $Y$  o altă variabilă aleatoare pentru care este cunoscută o metodă de generare și a cărei densitate de repartiție este  $h(x)$ , astfel încât densitățile  $f$  și  $h$  iau valori diferite de 0 pe aceeași submulțime  $A \subseteq \mathbb{R}$ . Presupunem că există o constantă  $\alpha$ , cu  $0 < \alpha < \infty$  astfel încât  $f(x) \leq \alpha h(x)$  pentru  $\forall x \in A$ . Atunci dacă  $U$  este o variabilă aleatoare  $U(0, 1)$ , independentă de  $Y$ , densitatea de repartiție a variabilei  $Y$ , condiționată de*

$$0 \leq U \leq \frac{f(Y)}{\alpha h(Y)}$$

*este  $f$ .*

Dem:

Trebuie să arătăm că:

$$P\left(Y < x | 0 \leq U \leq \frac{f(Y)}{\alpha h(Y)}\right) = F(x) = \int_{-\infty}^x f(v) dv$$

Fie evenimentele  $A$  și  $B$  definite astfel:

$$A = \{Y < x\}, \quad B = \left\{0 \leq U \leq \frac{f(Y)}{\alpha h(Y)}\right\}$$

Atunci trebuie să arătăm că:

$$P(A|B) = F(x)$$

Avem că:

$$\begin{aligned} P(B) &= P\left(0 \leq U \leq \frac{f(Y)}{\alpha h(Y)}\right) = \int_{-\infty}^{+\infty} \left[ \int_0^{\frac{f(v)}{\alpha h(v)}} du \right] h(v) dv = \\ &= \int_{-\infty}^{+\infty} \frac{f(v)}{\alpha h(v)} h(v) dv = \frac{1}{\alpha} \end{aligned}$$

deci

$$\begin{aligned} P(A|B) &= \frac{P(A \cap B)}{P(B)} = \alpha \int_{-\infty}^x \left[ \int_0^{\frac{f(v)}{\alpha h(v)}} du \right] h(v) dv = \\ &= \alpha \int_{-\infty}^x \frac{f(v)}{\alpha h(v)} h(v) dv = \int_{-\infty}^x f(v) dv = F(x) \end{aligned}$$

Observăm că:

- Această teoremă este cunoscută ca “teorema înfășurătoarei” pentru că graficul densității  $f(x)$  se poate “înfășura” cu  $\alpha h(x)$ .
- Din demonstrație rezultă că probabilitatea de acceptare este  $p_a = 1/\alpha$ . De aici rezultă că pentru a avea o metodă a înfășurătoarei nebanală, trebuie ca  $\alpha > 1$ .
- Procedura de respingere este formată din următoarele elemente:
  - $N = 2$  variabilă aleatoare constantă;
  - $\mathcal{S} = \{U, Y\}$ ;
  - $\mathcal{P}(U, Y) = \text{true}$  dacă  $0 \leq U \leq \frac{f(Y)}{\alpha h(Y)}$ ;
  - $\Psi(U, Y) = Y$ .

**Exemplu 6.** Fie  $X$  o variabilă Gama( $0, 1, \nu$ ) (adică Gama standard) cu  $0 < \nu < 1$ . Vom aplica metoda înfășurătoarei, folosind o densitate Weibull( $0, 1, \nu$ ).

*Avem densitățile de repartiție*

$$f(x) = \begin{cases} \frac{1}{\Gamma(\nu)} x^{\nu-1} e^{-x}, & \text{dacă } x \geq 0 \\ 0 & \text{dacă } x < 0 \end{cases} ;$$

$$h(x) = \begin{cases} \nu x^{\nu-1} e^{-x^\nu}, & \text{dacă } x \geq 0 \\ 0 & \text{dacă } x < 0 \end{cases}$$

*Pentru a determina constanta  $\alpha$  de înfășurare analizăm raportul:*

$$r(x) = \frac{f(x)}{h(x)} = \frac{1}{\nu \Gamma(\nu)} e^{-x+x^\nu}.$$

*Punctul de maxim al funcției  $r(x)$  este  $x_{\max} = \nu^{-\frac{1}{\nu-1}}$  de unde rezultă:*

$$\alpha = \frac{e^{\zeta(1-\nu)}}{\Gamma(\nu+1)} \text{ cu } \zeta = \nu^{\frac{\nu}{1-\nu}}$$

*Algoritmul pentru generarea variabilei  $X$  prin metoda respingerii este:*

**Algoritm Gama-Resp**

**Intrare:**  $\nu$ ,  $c := 1/\nu$ ,  $\zeta = \nu^{\frac{\nu}{1-\nu}}$ ,  $a = e^{\zeta(\nu-1)}$ .

P1: Se generează  $Y \sim Weib(0, 1, \nu)$  (metoda inversă);

P1.1:  $U \sim U(0, 1)$ ;

P1.2:  $Y := [-\ln(U)]^c$

P2: Se generează  $U \sim U(0, 1)$ ;

P3: Dacă  $U \leq ae^{Y^\nu - Y}$ ,  $X := Y$ , STOP. Altfel, mergi la P1;

**Ieșire:** Variabila aleatoare  $X$ .

# A doua teoremă de respingere

**Teorema 4.** *Fie  $X$  o variabilă aleatoare cu funcția de repartiție de forma:*

$$F(x) = c \int_{-\infty}^x Q(\phi(x)) dR(x) \quad (10)$$

*unde  $Q(z)$  este funcția de repartiție a unei variabile aleatoare  $Z$ ,  $Z \in [0, M]$ ,  $\phi(x)$  este o funcție care ia valori în  $[0, M]$  (cu  $M$  putând lua și valoarea  $\infty$ ), iar  $R(y)$  este funcția de repartiție a unei variabile aleatoare  $Y \in \mathbb{R}$ , independente de  $Z$ . În aceste condiții funcția de repartiție a variabilei  $Y$  condiționată de  $Z \leq \phi(Y)$  este  $F(x)$ .*

Dem:

Fie evenimentele  $A$  și  $B$  definite astfel:

$$A = \{Y < x\}; \quad B = \{Z \leq \phi(Y)\}$$

pentru a demonstra teorema trebuie să arătăm că:

$$P(A|B) = F(x).$$

Observăm că  $c$  din (10) este o constantă de normare, adică:

$$c = \left[ \int_{-\infty}^{+\infty} Q(\phi(x)) dR(x) \right]^{-1}.$$

Probabilitatea de realizare a evenimentului  $B$  este:

$$P(B) = \int_{-\infty}^{+\infty} \left( \int_0^{\phi(x)} dQ(y) \right) dR(x) = \int_{-\infty}^{+\infty} Q(\phi(x)) dR(x) = \frac{1}{c}.$$



Prin urmare:

$$\begin{aligned} P(A|B) &= \frac{P(A \cap B)}{P(B)} = cP(A \cap B) = c \int_{-\infty}^x \left( \int_0^{\phi(y)} dQ(z) \right) dR(y) = \\ &= c \int_{-\infty}^x Q(\phi(y)) dR(y) = F(x). \end{aligned}$$

Observăm că:

- Probabilitatea de acceptare este  $p_a = P(B) = \frac{1}{c}$ ;
- Elementele algoritmului de respingere sunt
  - $N = 2$  variabilă aleatoare constantă;
  - $\mathcal{S} = \{Z, Y\}$ ;
  - $\mathcal{P}(Z, Y) = \text{true}$  dacă  $Z \leq \phi(Y)$ ;
  - $\Psi(Z, Y) = Y$ .
- teorema se verifică și dacă relația (10) se scrie în termeni de densități de repartiție:

$$f(x) = cQ(\phi(x))r(x), \text{ cu } r(x) = R'(x).$$

- O formă duală a teoremei se obține dacă  $F(x)$  este de forma:

$$F(x) = c \int_{-\infty}^x (1 - Q(\phi(x))) dR(x) \quad (11)$$

cu

$$c = \left[ \int_{-\infty}^{+\infty} (1 - Q(\phi(x))) dR(x) \right]^{-1}$$

în acest caz evenimentul  $B$  este:  $B = \{Z \geq \phi(Y)\}$ .

- Relația (11) se poate scrie în funcție de densități astfel:

$$f(x) = c(1 - Q(\phi(x)))r(x)$$

iar probabilitatea de acceptare pentru varianta duală este:

$$p_a = P(Z \geq \phi(Y)) = \frac{1}{c}$$

**Exemplu 7.** Fie  $X$  o variabilă aleatoare cu densitatea:

$$f(x) = c(1 - e^{-\lambda x})\mu e^{-\mu x}, \quad x \geq 0 \quad (12)$$

unde  $c$  este o constantă de normare. Atunci un algoritm de generare a variabilei aleatoare  $X$  se poate scrie folosind a doua teoremă de respingere. Avem:

$$\phi(x) = x, \quad Q(z) = 1 - e^{-\lambda z}, \quad z > 0, \quad r(x) = \mu e^{-\mu x}.$$

Atunci:

$$c = \left[ \int_0^{+\infty} (1 - e^{-\lambda x})\mu e^{-\mu x} dx \right]^{-1} = \left[ \frac{\lambda}{\lambda + \mu} \right]^{-1}$$

iar un algoritm pentru generarea lui  $X$  este următorul:

## Algoritm Resp2

**Intrare:** Parametrii  $\lambda, \mu$ .

P1: Se generează  $Z \sim \text{Exp}(\lambda)$ ;

P2: Se generează  $Y \sim \text{Exp}(\mu)$ ;

P3: Dacă  $Z \leq Y$ ,  $X := Y$ , STOP. Altfel, mergi la P1;

**Ieșire:** Variabila aleatoare  $X$ .

- *Algoritmul Resp2 este rapid dacă  $\mu \ll \lambda$ .*
- *Metoda inversă nu este recomandabilă pentru că determinarea inversei funcției  $F$  nu este imediată.*

# A treia teoremă de respingere

## Teorema șirului descendent

**Teorema 5.** *Fie variabilele  $Z_i \sim G(x)$ ,  $i = 1, 2, \dots$ ,  $Z_0 \sim G_0(z)$  independente. Atunci următoarele afirmații sunt adevărate:*

1. *Dacă  $x$  și  $k$  sunt fixate atunci:*

$$P(x \geq Z_1 \geq Z_2 \geq \dots \geq Z_{k-1} < Z_k) = \frac{[G(x)]^{k-1}}{(k-1)!} - \frac{[G(x)]^k}{k!}. \quad (13)$$

2. *Dacă  $x$  este fixat și  $K$  este indicele aleator la care se “rupe” șirul descendent (ca la punctul 1), atunci*

$$P(K = \text{nr.impar}) = P(K \bmod 2 = 1) = e^{-G(x)}. \quad (14)$$

3. *Dacă subșirul descendent este  $Z_0 \geq Z_1 \geq \dots \geq Z_{K-1} < Z_K$  (adică se rupe la  $K$  aleator și începe cu  $Z_0 \sim G_0(x)$ ), atunci:*

$$P(Z_0 < x | K \bmod 2 = 1) = \frac{1}{p_a} \int_{-\infty}^x e^{-G(t)} dG_0(t), \quad (15)$$

unde  $p_a$  este constanta de normare:

$$p_a = \left[ \int_{-\infty}^{+\infty} e^{-G(x)} dG_0(x) \right] \quad (16)$$

Dem:

1. Fie evenimentele:

$$A = \{x \geq Z_1 \geq Z_2 \geq \dots \geq Z_{k-1}\}; \quad B = \{x \geq Z_1 \geq Z_2 \geq \dots \geq Z_k\}$$

Observăm că  $P(Z_i \leq x) = G(x)$  și

$$P(Z_1 \leq x, Z_2 \leq x, \dots, Z_{k-1} \leq x) = [G(x)]^{k-1}$$

Deoarece subșirul care definește evenimentul  $A$  conține numai una din cele  $(k-1)!$  ordini în care se pot afla cele  $k-1$  variabile aleatoare  $Z_i$ ,

$1 \leq i \leq k-1$ , rezultă că:

$$P(A) = \frac{[G(x)]^{k-1}}{(k-1)!}$$

pentru a demonstra (13) observăm că probabilitatea din membrul stâng se scrie  $P(A \setminus B)$  și pentru că  $B \subseteq A$  avem:

$$P(A \setminus B) = P(A) - P(B) = \frac{[G(x)]^{k-1}}{(k-1)!} - \frac{[G(x)]^k}{k!}$$

iar afirmația 1. este adevărată.

2. Au loc următoarele relații:

$$\begin{aligned} P(K = \text{nr. impar}) &= P(K = 1) + P(K = 3) + \dots = \\ &= 1 - \frac{G(x)}{1!} + \frac{[G(x)]^2}{2!} - \frac{[G(x)]^3}{3!} + \dots = e^{-G(x)} \end{aligned}$$



3. Observăm că atunci când  $Z_0$  este aleator avem:

$$P(K = \text{nr. impar}) = P(K \bmod 2 = 1) = \int_{-\infty}^{+\infty} e^{-G(t)} dG_0(t)$$

și ținând cont de forma probabilității  $p_a$  din (16) obținem

$$P(Z_0 < x | K \bmod 2 = 1) = \frac{1}{p_a} \int_{-\infty}^x e^{-G(t)} dG_0(t)$$

Algoritmul rezultat din a treia teoremă de respingere este următorul:

### Algoritm Resp3

#### Intrare:

P1: Se generează  $Z_0 \sim G_0(x)$ ;

P2:  $Z^* := Z_0, K = 1$ ;

P3: Se generează  $Z_1 \sim G(x)$ ;

P4: Dacă  $Z_0 \geq Z_1$  mergi la P5, altfel mergi P6;

P5:  $K := K + 1, Z_0 := Z_1$ , mergi pa P3;

P6: Dacă  $K \bmod 2 = 1$   $X = Z^*$ , STOP.

Altfel mergi la P1.

**Ieșire:** Variabila aleatoare  $X$ .

# Generarea variabilelor neuniforme

## *Curs 4*

# Algoritmul 3 de respingere

## Analiza performanței

Probabilitatea de acceptare  $p_a$  dă informații asupra vitezei algoritmului. O probabilitate  $p_a$  mare înseamnă acceptarea mai rapidă a lui  $Z_0$  (când  $K$  este impar). Dar probabilitatea  $p_a$  nu este suficientă pentru a caracteriza pe deplin performanța algoritmului. Trebuie verificat câte  $Z_i$  sunt necesare pentru acceptarea unui  $Z_0$ .

Nr. mediu de șiruri (care au  $K + 1$  variabile) generate până la acceptarea unui șir (și implicit al unei variabile  $Z_0$ ) este  $\frac{1}{p_a}$ . Fie  $N^*$  variabila aleatoare care reprezintă numărul total de variabile  $\{Z_i\}_{i \geq 0}$  generate până la acceptarea unui  $Z_0$ . Atunci:

$$E[N^*] = \frac{1}{p_a} E[K + 1].$$

Observăm că:

$$E[K + 1] = E[K] + 1$$

$$\begin{aligned}
E[K] &= \sum_{k=1}^{\infty} k \int_{-\infty}^{+\infty} \left[ \frac{(G(x))^{k-1}}{(k-1)!} - \frac{(G(x))^k}{k!} \right] dG_0(x) = \\
&= \int_{-\infty}^{+\infty} \left\{ \sum_{k=1}^{\infty} k \left[ \frac{(G(x))^{k-1}}{(k-1)!} - \frac{(G(x))^k}{k!} \right] \right\} dG_0(x) = \\
&= \int_{-\infty}^{+\infty} \left\{ 1 + \sum_{k=2}^{\infty} k \frac{(G(x))^{k-1}}{(k-1)!} - \sum_{k=1}^{\infty} k \frac{(G(x))^k}{k!} \right\} dG_0(x) = \\
&= \int_{-\infty}^{+\infty} \left\{ 1 + \sum_{k=1}^{\infty} (k+1) \frac{(G(x))^k}{k!} - \sum_{k=1}^{\infty} k \frac{(G(x))^k}{k!} \right\} dG_0(x) = \\
&= \int_{-\infty}^{+\infty} \left\{ 1 + \sum_{k=1}^{\infty} \frac{(G(x))^k}{k!} \right\} dG_0(x) = \int_{-\infty}^{+\infty} e^{G(x)} dG_0(x)
\end{aligned}$$

În concluzie

$$E[N^*] = \frac{1}{p_a} \left( 1 + \int_{-\infty}^{+\infty} e^{G(x)} dG_0(x) \right). \quad (1)$$

**Exemplu 1.** Presupunem că variabilele  $Z_i, i \geq 0$  sunt uniforme pe intervalul  $[0, 1]$ ,  $Z_i = U_i, i \geq 0$ . Atunci conform celei de-a treia teoreme de respingere avem:

$$P(U_0 \leq x | K = nr.impar) = \frac{1}{p_a} \int_0^x e^{-t} dt$$

cu

$$p_a = \int_0^1 e^{-x} dx = 1 - e^{-1}.$$

*Cu alte cuvinte  $U_0$  acceptat are funcția de repartiție*

$$F(x) = \begin{cases} 0, & \text{dacă } x < 0 \\ \frac{1-e^{-x}}{1-e^{-1}}, & \text{dacă } 0 \leq x \leq 1 \\ 1, & \text{dacă } x > 1 \end{cases}$$

*care reprezintă funcția de repartiție a unei variabile  $\text{Exp}(1)$  trunchiată pe intervalul  $[0, 1]$ .*

*Numărul mediu  $E[N^*]$  de variabile  $\{Z_i\}_{i \geq 0}$  care trebuie generate conform (1) este:*

$$E[N^*] = \frac{1}{1-e^{-1}} \left( 1 + \int_0^1 e^x dx \right) = \frac{e}{1-e^{-1}} = \frac{e^2}{e-1}.$$

# Alte metode de generare

În această secțiune vom prezenta câteva metode de generare ale unor variabile aleatoare, metode care nu se înscriu în cazurile teoremelor de mai sus.

## Variabila modul

Variabila aleatoare  $X$  are distribuția modul dacă densitatea ei de repartiție este:

$$f(x) = \begin{cases} 1 - |x|, & \text{dacă } x \in [-1, 1] \\ 0, & \text{altfel.} \end{cases}$$

Atunci funcția de repartiție a variabilei  $X$  este:

$$F(x) = \begin{cases} 0, & \text{dacă } x < -1 \\ x - |x|\frac{x}{2} + \frac{1}{2} & \text{dacă } x \in [-1, 1] \\ 1 & \text{dacă } x > 1. \end{cases}$$



Fie  $U_1, U_2$  două variabile aleatoare uniforme pe  $[0, 1]$ . Atunci variabila aleatoare  $Y = U_1 - U_2$  are aceeași distribuție ca și  $X$ .

Fie  $y \in [-1, 0]$ , atunci funcția de repartiție a lui  $Y$ , calculată în  $y$  este:

$$P(Y < y) = P(U_1 - U_2 < y) = \int_D du_1 du_2 = y + \frac{y^2}{2} + \frac{1}{2}.$$

Fie  $y \in [0, 1]$ , atunci funcția de repartiție a lui  $Y$ , calculată în  $y$  este:

$$P(Y < y) = P(U_1 - U_2 < y) = \int_D du_1 du_2 = y - \frac{y^2}{2} + \frac{1}{2}.$$

## Repartiția maximului

O variabilă aleatoare  $X$  are repartiția maximului dacă are densitatea de repartiție:

$$f(x) = \begin{cases} nx^{n-1}, & \text{dacă } x \in [0, 1] \\ 0, & \text{altfel.} \end{cases}$$

Fie  $U_1, U_2, \dots, U_n$  variabile aleatoare uniforme pe  $[0, 1]$ . Atunci variabila aleatoare  $Y = \max\{U_1, U_2, \dots, U_n\}$  are aceeași repartiție ca și variabila aleatoare  $X$ .

Funcția de repartiție a variabilei  $X$  este:

$$F(x) = \begin{cases} 0 & \text{dacă } x < 0 \\ x^n & \text{dacă } x \in [0, 1] \\ 1 & \text{dacă } x > 1. \end{cases}$$

Calculăm funcția de repartiție a variabilei  $Y$  în punctul  $y \in [0, 1]$ :

$$\begin{aligned} P(Y < y) &= P(\max\{U_1, U_2, \dots, U_n\} < y) = \\ &= P(U_1 < y, U_2 < y, \dots, U_n < y) = \prod_{i=1}^n P(U_i < y) = y^n \end{aligned}$$

### **Repartiția minimului**

O variabilă aleatoare  $X$  are repartiția minimului dacă are densitatea de repartiție:

$$f(x) = \begin{cases} n(1-x)^{n-1}, & \text{dacă } x \in [0, 1] \\ 0, & \text{altfel.} \end{cases}$$

Fie  $U_1, U_2, \dots, U_n$  variabile aleatoare uniforme pe  $[0, 1]$ . Atunci variabila aleatoare  $Y = \min\{U_1, U_2, \dots, U_n\}$  are aceeași repartiție ca și variabila aleatoare  $X$ .

Funcția de repartiție a variabilei  $X$  este:

$$F(x) = \begin{cases} 0 & \text{dacă } x < 0 \\ 1 - (1 - x)^n & \text{dacă } x \in [0, 1] \\ 1 & \text{dacă } x > 1. \end{cases}$$

Calculăm funcția de repartiție a variabilei  $Y$  în punctul  $y \in [0, 1]$ :

$$\begin{aligned} P(Y < y) &= P(\min\{U_1, U_2, \dots, U_n\} < y) = \\ &= 1 - P(\min\{U_1, U_2, \dots, U_n\} \geq y) = 1 - P(U_1 \geq y, U_2 \geq y, \dots, U_n \geq y) \end{aligned}$$

$$\begin{aligned}
&= 1 - \prod_{i=1}^n P(U_i \geq y) = \\
&= 1 - \prod_{i=1}^n (1 - P(U_i < y)) = 1 - (1 - y)^n
\end{aligned}$$

## Repartiția Erlang

Fie  $X$  o variabilă aleatoare  $Erlang(k)$ ,  $k \in \mathbb{N}^*$ , cu densitatea de repartiție:

$$f(x) = \begin{cases} 0, & \text{dacă } x < 0; \\ \frac{1}{\Gamma(k)} x^{k-1} e^{-x} & \text{dacă } x \geq 0. \end{cases}$$

Fie  $Z_1, Z_2, \dots, Z_k$  variabile distribuite  $Exp(1)$  independente. Atunci variabila  $Y = \sum_{j=1}^k Z_j$  are aceeași distribuție ca și variabila  $X$ .

Variabilele aleatoare  $Z_j$  pot fi generate cu metoda inversă și prin urmare putem să scriem:

$$X = -\ln \left\{ \prod_{j=1}^k U_j \right\}.$$

## Repartiția Normală

Teorema limită centrală (formă simplificată): Dacă  $\{V_n\}_{n \in \mathbb{N}}$  este un șir de variabile aleatoare independente și identic distribuite care au momente de ordinul 1 și 2 și dacă  $S_n = \sum_{i=1}^n V_i$ , atunci

$$\lim_{n \rightarrow \infty} P \left( \frac{S_n - E[S_n]}{\sqrt{Var[S_n]}} < x \right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{t^2}{2}} dt. \quad (2)$$

În membrul al doilea este funcția de repartiție normală  $N(0,1)$ . Dacă considerăm ca variabile  $V_i$  variabilele uniforme  $U_i$ , atunci:

$$E[U_i] = \frac{1}{2} \text{ și } Var[U_i] = \frac{1}{12}.$$

Viteza de convergență depinde de viteza de generare a variabilelor  $V_i$ . În cazul  $V_i = U_i$  membrul stâng din (2) se apropie suficient de mult de funcția de repartiție normală  $N(0, 1)$  din membrul drept, dacă  $n \geq 10$ . Pentru  $n = 12$  se obține:

$$E[S_n] = 6, \quad Var[S_n] = 1 \Rightarrow \sum_{i=1}^{12} U_i - 6$$

care are (aproximativ) repartiția normală  $N(0, 1)$ .

Fie  $Y$  o variabilă normală  $N(m, \sigma)$ , atunci

$$Y = m + \sigma X$$

unde  $X$  este o variabilă normală  $N(0, 1)$ .

## Simularea unor repartiții înrudite cu repartiția normală

### Repartiția $\chi^2$

Fie  $Z_1, Z_2, \dots, Z_f$  variabile normale  $N(0, 1)$ , independente. Atunci:

$$\chi_f^2 = \sum_{i=1}^f Z_i^2 \quad (3)$$

se numește variabilă  $\chi^2$  centrată, cu  $f$  grade de libertate.

Dacă  $Z_i \sim N(m_i, 1)$  atunci variabila (3) se notează cu  $\chi_{f,\delta}^2$  și se numește variabilă  $\chi^2$  necentrată, cu  $f$  grade de libertate și cu parametrul de excentricitate  $\delta$ , unde

$$\delta^2 = \sum_{i=1}^f m_i^2.$$



Se poate arăta că  $\chi_f^2$  centrată este o variabilă  $\text{Gamma}(0, \frac{1}{2}, \frac{f}{2})$ .

Din formula (3) variabilele  $\chi_f^2$  și  $\chi_{f,\delta}^2$  se pot simula direct folosind definiția lor.

### **Variabila t Student**

Dacă  $Z \sim N(0, 1)$  este o variabilă independentă de variabila  $\chi_f^2$ , atunci variabila

$$t_f = \frac{Z}{\sqrt{\frac{\chi_f^2}{f}}} \quad (4)$$

se numește variabila *t Student* cu  $f$  grade de libertate. Dacă în (4) în loc de  $\chi_f^2$  se folosește  $\chi_{f,\delta}^2$ , atunci se obține o variabilă  $t_{f,\delta}$  numită t Student necentrată, cu  $f$  grade de libertate și cu parametrul de excentricitate  $\delta$ .

Variabilele t Student se simulează folosind (4).

## Variabila F Snedecor

Dacă  $\chi_{f_1}^2, \chi_{f_2}^2$  sunt independente, atunci variabila:

$$F_{f_1, f_2} = \frac{f_2 \chi_{f_1}^2}{f_1 \chi_{f_2}^2} \quad (5)$$

se numește variabila  $F$  a lui *Snedecor* centrată, cu  $f_1, f_2$  grade de libertate. Dacă în (5) se folosește câte una din  $\chi_{f_1, \delta_1}^2, \chi_{f_2, \delta_2}^2$ , sau ambele, atunci se obțin variabilele  $F$  simplu necentrate  $F_{f_1, f_2, \delta_1, 0}, F_{f_1, f_2, 0, \delta_2}$  cu parametrii corespunzători de excentricitate, sau variabila  $F$  dublu necentrată  $F_{f_1, f_2, \delta_1, \delta_2}$ .

Variabilele  $F$  se pot simula direct din (5).

## Variabila log-normală

Variabila aleatoare  $Y$  se numește log-normală  $LN(\mu, \sigma)$  de parametrii  $\mu$  și  $\sigma$  dacă variabila  $X = \log(Y)$  este normală  $N(\mu, \sigma)$ .

Prin urmare, dacă densitatea variabilei  $X$  este:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad x \in \mathbb{R}$$

atunci densitatea variabilei  $Y$  este:

$$g(y) = \frac{1}{\sqrt{2\pi}y\sigma} e^{-\frac{(\log(y)-\mu)^2}{2\sigma^2}} \quad y \in \mathbb{R}_+ \quad (6)$$

cu media  $m$  și dispresia  $s^2$  date de:

$$m = E[Y] = e^{\mu+\sigma^2/2}, \quad s^2 = Var[Y] = m^2(e^{\sigma^2} - 1).$$

Dacă se cunosc  $m$  și  $s^2$  atunci  $\mu$  și  $\sigma$  se pot determina astfel:

$$\mu = \log(m) - \frac{1}{2} \log \left[ \frac{s^2}{m^2} + 1 \right], \quad \sigma^2 = \log \left[ \frac{s^2}{m^2} + 1 \right]$$

Prin urmare simularea unei variabile aleatoare  $Y$  log-normale de medie  $m$  și dispersie  $\sigma^2$  se face prin următorul algoritm:

### **Algoritm Lnorm**

**Intrare:**  $m, s^2$ . Se calculează  $\mu, \sigma$ .

P1: Se generează  $Z \sim N(0,1)$ ;

P2: Calculează  $X = \mu + \sigma Z$ ;

P3:  $Y = e^X$

**Ieșire:** Variabila aleatoare  $Y$  cu densitatea (6).

# Generarea variabilelor neuniforme

*Curs 5*

# Simularea unor variabile particulare

În continuare se vor prezenta metode de generare a anumitor tipuri de variabile aleatoare. Aceste metode se bazează fie pe teoremele de simulare, fie pe proprietăți ale acestor tipuri de variabile aleatoare.

# Variabila Exponențială

Fie  $X$  o variabilă exponențială de parametru 1 cu densitatea de repartiție

$$f(x) = \begin{cases} e^{-x}, & \text{dacă } x \geq 0 \\ 0, & \text{dacă } x < 0. \end{cases} \quad (1)$$

Fie  $Y$  o variabilă exponențială de parametru  $\lambda$  cu densitatea de repartiție

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & \text{dacă } x \geq 0 \\ 0, & \text{dacă } x < 0. \end{cases} \quad (2)$$

Atunci are loc relația:

$$Y = \frac{X}{\lambda}$$

Prin urmare pentru a simula  $Y$  este suficient să descriem o metodă de simulare pentru  $X$ .

## Metode de simulare pentru $X \sim \text{Exp}(1)$

### - Metoda inversă:

$$X = -\log(U)$$

unde  $U$  este o variabilă uniformă pe  $[0, 1]$ .

Are dezavantajul că pentru valori ale lui  $U$  apropiate de 0 nu se poate calcula logaritmul.

### - Metodă de generare cu a treia teoremă de respingere

**Teorema 1.** *În teorema subșirului descendent considerăm  $Z_0 = U_0$ ,  $Z_i = U_i$ ,  $i \geq 1$ , unde  $U_0, U_1, \dots$  sunt uniforme pe  $[0, 1]$ . Dacă notăm cu  $N$  numărul aleator de subșiruri descendente respinse până când se acceptă un subșir, atunci  $X = N + Z_0$  este o variabilă  $\text{Exp}(1)$ , unde  $Z_0$  este variabila acceptată (din ultimul subșir descendent).*



Dem:

Din exemplul de la a treia teoremă de respingere, pentru  $x \in [0, 1]$  avem:

$$P(Z_0 \leq x | K = nr.impar) = \frac{1}{p_a} \int_0^x e^{-x} dx = F(x) = \frac{1 - e^{-x}}{p_a}$$

cu

$$p_a = 1 - e^{-1}.$$

unde  $F(x)$  este funcția de repartiție pentru variabila exponențială trunchiată pe  $[0, 1]$ .

Deci probabilitatea de a respinge un șir descendent (de forma  $Z_0 \geq Z_1 \geq \dots \geq Z_{K-1} < Z_K$ ) este  $p_r = 1 - p_a = e^{-1}$ . Prin urmare

$$P(N = n) = e^{-n}(1 - e^{-1}).$$

Trebuie să mai arătăm că:

$$P(N + Z_0 \leq x) = \begin{cases} 0 & \text{dacă } x < 0; \\ 1 - e^{-x}, & \text{dacă } x \geq 0 \end{cases}$$

Fie  $x > 0$  oarecare. Notăm cu  $k = [x]$  și cu  $z = x - k$ ,  $z \in [0, 1)$ . Atunci avem:

$$P(N + Z_0 \leq x) = P(N + Z_0 \leq k + z) = P(N < k) + P(N = k, Z_0 \leq z) =$$

$$= \sum_{j=0}^{k-1} (1 - e^{-1})e^{-j} + (1 - e^{-1}) \frac{e^{-k}}{1 - e^{-1}} \int_0^z e^{-u} du =$$

$$= 1 - e^{-k} + e^{-k}(1 - e^{-z}) = 1 - e^{-(k+z)} = 1 - e^{-x}$$

## Algoritm Resp3-Exp

### Intrare:

P1:  $N = 0$ ;  
P2: Se generează  $U_0, U_1 \sim U(0, 1)$  independente;  
P3:  $U^* = U_0, K = 1$ ;  
P4: Dacă  $U_0 \geq U_1$  mergi la P5, altfel mergi la P7;  
P5:  $K := K + 1, U_0 := U_1$ ;  
P6: Se generează  $U_1 \sim U(0, 1)$ , mergi la P4;  
P7: Dacă  $K \bmod 2 = 1$   $X = N + U^*$ , STOP. Altfel  
 $N := N + 1$ , mergi la pasul 2.

**Ieșire:** Variabila aleatoare  $X$ .

Pe lângă probabilitatea de acceptare  $p_a = 1 - e^{-1}$ , performanța algoritmului este caracterizată și de numărul  $N^*$  al variabilelor uniforme  $\{U_i\}_{i \geq 0}$  generate, necesare pentru a obține o valoare de selecție exponențială  $X$ . Din evaluarea performanțelor algoritmului celei de-a treia metode de respingere:

$$E[N^*] = \frac{1}{p_a} \left( 1 + \int_0^1 e^x dx \right) = \frac{1}{1 - e^{-1}} (1 - e^{-1}) = \frac{e^2}{e - 1} \approx 3,8$$

# Variabila Gama

Fie  $Y$  o variabilă aleatoare distribuită  $Gama(\alpha, \lambda, \nu)$ . Atunci  $Y$  are următoarea densitate de repartiție:

$$f(x) = \begin{cases} \frac{\lambda^\nu}{\Gamma(\nu)} (x - \alpha)^{\nu-1} e^{-\lambda(x-\alpha)}, & \text{dacă } x \geq \alpha \\ 0, & \text{altfel} \end{cases} . \quad (3)$$

Cu funcția  $\Gamma(\nu)$  definită astfel:

$$\Gamma(\nu) = \int_0^{+\infty} x^{\nu-1} e^{-x} dx.$$

Fie  $X$  o variabilă aleatoare  $Gama(0, 1, \nu)$ . Atunci  $X$  se mai numește variabilă *Gama standard* și are densitatea de repartiție:

$$f(x) = \begin{cases} \frac{1}{\Gamma(\nu)} x^{\nu-1} e^{-x}, & \text{dacă } x \geq 0 \\ 0, & \text{altfel} \end{cases} . \quad (4)$$

Atunci  $Y$  se poate scrie

$$Y = \alpha + \frac{X}{\lambda}$$

Prin urmare pentru a genera variabila  $Y$  este suficientă găsirea unei metode de generare a variabilei  $X$ .

Algoritmii de simulare pentru variabila  $X$  depind de intervalul în care se află parametrul  $\nu$ .

## 1. Pentru $0 < \nu < 1$

1.1. Generarea variabilei  $X$  prin metoda respingerii cu ajutorul unei variabile Weibull (exemplu în **cursul 3**).

1.2. Generarea variabilei  $X$  printr-o metodă de compunere-respingere.

Vom scrie densitatea (4) sub forma:

$$f(x) = p_1 f_1(x) + p_2 f_2(x)$$

cu

$$f_1(x) = \begin{cases} \frac{f(x)}{p_1}, & \text{dacă } x \in [0, 1] \\ 0, & \text{altfel} \end{cases}, \quad f_2(x) = \begin{cases} \frac{f(x)}{p_2}, & \text{dacă } x \in (1, +\infty) \\ 0, & \text{altfel} \end{cases}$$

unde

$$p_1 = \frac{\Gamma(1; \nu)}{\Gamma(\nu)}, \quad p_2 = 1 - p_1 = \frac{\Gamma(\nu) - \Gamma(1; \nu)}{\Gamma(\nu)}.$$

Funcția  $\Gamma(1; \nu)$  este o funcție gama incompletă:

$$\Gamma(1; \nu) = \int_0^1 x^{\nu-1} e^{-x} dx$$

Presupunem că variabila  $X_1$  are densitatea de repartiție  $f_1$  și că variabila  $X_2$  are densitatea de repartiție  $f_2$ . Atunci are loc următoarea teoremă.

**Teorema 2.** Variabila  $X_1$  se simulează folosind a treia teoremă de respingere (a subșirului descendent) cu  $Z_0 = U_0^{1/\nu}$ ,  $Z_i = U_i$ , cu  $\{U_i\}_{i \geq 0}$  variabile uniforme pe  $[0, 1]$ . Variabila  $X_2$  se simulează cu a doua teoremă de respingere, forma duală, unde densitatea  $f_2(x)$  este de forma

$$f_2(x) = c(1 - Q(x))r(x), \quad x > 0$$

cu

$$c = \frac{1}{e(\Gamma(\nu) - \Gamma(1; \nu))}$$

și

$$r(x) = \begin{cases} e^{-x+1}, & \text{dacă } x \geq 1 \\ 0 & \text{dacă } x < 1 \end{cases}, \quad Q(x) = \begin{cases} 1 - x^{\nu-1}, & \text{dacă } x \geq 1 \\ 0 & \text{dacă } x < 1 \end{cases}$$

Dem:

Fie  $x \in [0, 1]$ . Atunci aplicând a treia teoremă de respingere avem:

$$G_0(x) = P(U_0^{1/\nu} < x) = P(U_0 < x^\nu) = x^\nu$$

$$H(x) = P(U_0 < x | K = nr.impar) = \frac{1}{p_a} \int_0^x e^{-t} dG_0(t) = \frac{1}{p_a} \int_0^x e^{-t} \nu t^{\nu-1} dt$$

cu

$$p_a = \int_0^1 e^{-t} \nu t^{\nu-1} dt = \nu \Gamma(1; \nu).$$

Dacă derivăm  $H(x)$  obținem:

$$H'(x) = h(x) = \frac{e^{-x} x^{\nu-1}}{\Gamma(1; \nu)} = f_1(x), \quad x \in [0, 1]$$

ceea ce demonstrează prima parte a teoremei.



Pentru a demonstra a doua parte îl scriem pe  $f_2(x)$  astfel:

$$f_2(x) = \begin{cases} \frac{x^{\nu-1}e^{-x}}{\Gamma(\nu)-\Gamma(1;\nu)}, & \text{dacă } x \geq 1 \\ 0, & \text{dacă } x < 1 \end{cases}$$

adică  $f_2(x)$  are forma din enunțul teoremei.

Din această teoremă rezultă următorul algoritm de generare a variabilei  $X$ .

### **Algoritm Gama2**

**Intrare:**  $p_1, p_2, g = \Gamma(\nu), g_1 = \Gamma(1;\nu), p_1 = \frac{g_1}{g}, p_2 = \frac{g-g_1}{g},$

$c = \frac{1}{e(\Gamma(\nu)-\Gamma(1;\nu))}, a = \frac{1}{\nu}, b = -\frac{1}{1-\nu}$

P1: Se generează  $U \sim U(0,1);$

P2: Dacă  $U \leq p_1$  mergi la P3, altfel mergi la P4;

P3: Se generează  $X_1 \sim f_1(x), X := X_1.$

P4: Se generează  $X_2 \sim f_2(x), X := X_2.$

**Ieșire:** Variabila aleatoare  $X$ .

Variabilele  $X_1$  și  $X_2$  se generează cu următorii algoritmi:

### **Algoritm Gama2- $X_1$**

#### **Intrare:**

P1: Se generează  $U \sim U(0, 1)$ ;

P2:  $Z_0 := U^a$ ;

P3: Se generează  $Z_1 \sim U(0, 1)$ ;

P4:  $K := 1, Z^* := Z_0$ ;

P5: Dacă  $Z_0 \geq Z_1$  mergi la P6, altfel, mergi la P7;

P6:  $Z_0 := Z_1$ , se generează  $Z_1 \sim U(0, 1)$ ,  $K := K + 1$ ,  
mergi la P5;

P7: Dacă  $K \bmod 2 = 1$   $X_1 = Z^*$ . Altfel, mergi la P1.

**Ieșire:** Variabila aleatoare  $X_1$ .

Pentru algoritmul Gama2- $X_1$  avem:

$$p_a = \nu \Gamma(1; \nu); \quad E(N_1^*) = \frac{1}{p_a} \left( 1 + \int_0^1 \nu t^{\nu-1} e^t dt \right)$$

## Algoritm Gama2- $X_2$

### Intrare:

P1: Se generează  $U \sim U(0, 1)$ ;

P2:  $Z := U^b$ ;

P3: Se generează  $X_0 \sim \text{Exp}(1)$ ;

P4:  $Y := X_0 + 1$  P5: Dacă  $Y > Z$  mergi la P1, altfel mergi la P6;

P6:  $X_2 := Y$ ;

Ieșire: Variabila aleatoare  $X_2$ .

Pentru algoritmul Gama- $X_2$  avem:

$$p_2 = 1 - p_1; \quad E(N_2^*) = \frac{2}{p_2}$$

și prin urmare numărul mediu de variabile necesare pentru a genera în final un  $X$  este:

$$E(N^*) = p_1 E(N_1^*) + p_2 E(N_2^*) = p_1 E(N_1^*) + 2$$

## 2. Pentru $\nu > 1$

Doi algoritmi de respingere bazați pe metoda înfășurătoarei (prima teoremă de respingere).

### 2.1 Primul algoritm de respingere pentru variabila $X \sim \text{Gama}(0, 1, \nu)$ cu $\nu > 1$ .

Considerăm ca înfășurătoare densitatea  $h(x) \sim \text{Exp}(\frac{1}{\nu})$ , adică:

$$h(x) = \begin{cases} \frac{1}{\nu} e^{-\frac{x}{\nu}} & \text{dacă } x \geq 0 \\ 0 & \text{dacă } x < 0 \end{cases} . \quad (5)$$

Notăm cu  $r(x)$  raportul:

$$r(x) = \frac{f(x)}{h(x)} = \frac{\nu x^{\nu-1} e^{-x}}{\Gamma(\nu) e^{-\frac{x}{\nu}}}, \quad \nu > 1$$

Atunci funcția  $r(x)$  are un punct de maxim pentru  $x = \nu$  și constanta  $\alpha$  din teorema înfășurătoarei este:

$$\alpha = r(\nu) = \frac{\nu^\nu e^{1-\nu}}{\Gamma(\nu)}$$

Algoritmul de respingere va avea probabilitatea de acceptare:

$$p_a = \frac{\Gamma(\nu)}{\nu^\nu e^{1-\nu}} \approx \sqrt{\frac{e^2 2\pi}{\sqrt{\nu - 1}}}$$

ultima relație rezultând din aproximarea lui Stirling pentru  $\nu \rightarrow \infty$ :

$$\Gamma(\nu) \approx (\nu - 1)^{\nu-1} e^{-(\nu-1)} \sqrt{2\pi(\nu - 1)}$$

## 2.2 Al doilea algoritm de respingere pentru variabila

$X \sim \text{Gama}(0, 1, \nu)$  cu  $\nu > 1$ .

Algoritmul de mai sus este lent pentru  $\nu$  foarte mare, de aceea vom prezenta un alt algoritm de respingere, de data aceasta bazat pe o înfășurătoare dată de o densitate Cauchy nestandard trunchiată pe  $[0, \infty)$ :

$$h(x) = \frac{k}{1 + \frac{(x - (\nu - 1))^2}{c}}, \quad x \geq 0 \quad (6)$$

unde  $k$  este o constantă de normare. Atunci are loc următoarea teoremă:

**Teorema 3.** *Dacă se înfășoară densitatea  $\text{Gama}(0, 1, \nu)$ ,  $\nu > 1$  cu densitatea  $h(x)$  dată de (6), atunci pentru  $c \geq 2\nu - 1$  avem:*

$$r(x) = \frac{f(x)}{h(x)} \leq \alpha = \frac{1}{k\Gamma(\nu)} (\nu - 1)^{\nu-1} e^{-(\nu-1)} \quad (7)$$

Dem:  
Avem

$$r(x) = \frac{1}{k\Gamma(\nu)}\varphi(x)$$

unde

$$\varphi(x) = x^{\nu-1}e^{-x} \left[ 1 + \frac{(x - (\nu - 1))^2}{c} \right].$$

Atunci:

$$\varphi'(x) = -\frac{e^{-x}x^{\nu-1}}{c}[x - (\nu - 1)][(x - \nu)^2 + c - (2\nu - 1)]$$

de unde rezultă că ecuația  $\varphi'(x) = 0$  are soluția  $x_0 = \nu - 1 > 0$  iar dacă  $c \geq 2\nu - 1$  atunci  $x_0$  este punct de maxim. Dacă luăm  $c = 2\nu - 1$  atunci avem:

$$\alpha = \frac{1}{k\Gamma(\nu)}(\nu - 1)^{\nu-1}e^{-(\nu-1)}$$

Din această teoremă rezultă următorul algoritm de generare a unei variabile  $Gama(0, 1, \nu)$  cu  $\nu > 1$ :

### Algoritm Gama3

**Intrare:**  $\nu, b = \nu - 1, c = \nu + b, s = \sqrt{2\nu - 1}$

P1: Se generează  $U \sim U(0, 1)$ ,  $T := s \cdot tg[\pi(U - 0.5)]$  ( $T$  este o variabilă Cauchy standard generată cu metoda inversă);

P2:  $Y = b + T$  ( $Y$  este o variabilă Cauchy nestandard);

P3: Dacă  $Y > 0$  mergi la P4, altfel mergi la P1;

P4: Se generează  $U_1 \sim U(0, 1)$ ;

P5: Dacă  $U_1 \leq e^{b \log(Y/b) - T + \log(1 + T^2/c)}$  mergi la P6, altfel mergi la P1;

P6:  $X = Y$ .

**Ieșire:** Variabila aleatoare  $X$ .



Constanta de normare  $k$  nu intervine în construcția algoritmului, dar ea este necesară pentru a calcula probabilitatea de acceptare  $p_a$ . Se verifică ușor că:

$$k = \left[ \frac{\pi}{2} + \arctg \left( -\frac{\nu - 1}{\sqrt{2\nu - 1}} \right) \right]^{-1}$$

Un **alt algoritm de generare** pentru o variabilă  $Gama(0, 1, \nu)$  cu  $\nu > 1$  se poate obține folosind următorul raționament: fie

$$\nu = k + p$$

unde  $k = [\nu]$  este partea întreagă și  $p = \nu - k \in [0, 1)$ . Atunci

$$X = E_k + Y$$

unde  $E_k$  este variabila Erlang de parametru  $k$ ,  $Y$  este o variabilă  $Gama(0, 1, \nu)$  cu  $\nu < 1$ , iar  $E_k$  și  $Y$  sunt independente.

# Generarea unor variabile particulare

*Curs 6*

# Repartiția Beta

Fie  $X$  o variabilă aleatoare cu densitatea de repartiție:

$$f(x) = \begin{cases} \frac{1}{B(a,b)} x^{a-1} (1-x)^{b-1}, & \text{dacă } x \in [0, 1] \\ 0, & \text{altfel} \end{cases} \quad (1)$$

unde

$$B(a, b) = \int_0^1 x^{a-1} (1-x)^{b-1} dx, \quad B(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

este funcția beta.

Atunci variabila  $X$  are o distribuție  $Beta(a, b)$  și se poate genera folosind următoarea teoremă:

**Teorema 1.** *Dacă  $X_1 \sim \text{Gama}(0, 1, a)$ ,  $X_2 \sim \text{Gama}(0, 1, b)$  sunt două variabile independente, atunci variabila:*

$$X = \frac{X_1}{X_1 + X_2} \quad (2)$$

*este o variabilă Beta( $a, b$ ).*

Dem:

Densitatea comună de repartiție a variabilelor  $X_1, X_2$  este:

$$f(x_1, x_2) = \frac{1}{\Gamma(a)\Gamma(b)} x_1^{a-1} x_2^{b-1} e^{-(x_1+x_2)}$$

Făcând transformarea de variabile:

$$U = \frac{X_1}{X_1 + X_2}, \quad V = X_2,$$

atunci densitatea comună a variabilelor  $U$  și  $V$  este

$$g(u, v) = f(x_1(u, v), x_2(u, v)) \cdot J$$

cu

$$J = \det \left( \frac{\partial(x_1, x_2)}{\partial u \partial v} \right) = \frac{v}{(1-u)^2}, \quad 0 < u < 1.$$

Avem

$$g(u, v) = \frac{1}{\Gamma(a)\Gamma(b)} \frac{u^{a-1}v^{a+b-1}}{(1-u)^{a+1}} e^{-\frac{v}{1-u}}$$

cu  $0 < v < \infty$ . Densitatea de repartiție a variabilei  $U$  este

$$h(u) = \int_0^\infty g(u, v) dv = \frac{1}{\Gamma(a)\Gamma(b)} \int_0^\infty \frac{u^{a-1}v^{a+b-1}}{(1-u)^{a+1}} e^{-\frac{v}{1-u}} dv$$

adică

$$h(u) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} u^{a-1} (1-u)^{b-1}$$

ceea ce demonstrează teorema.

Prin urmare, un algoritm de generare a unei variabile *Beta* este dat de relația (2). Acest algoritm ar presupune generarea a două variabile *Gama* și acest lucru implică o complexitate mare. De aceea în cazuri particulare se aplică algoritmi rezultați din următoarele teoreme.

**Teorema 2.** *Fie  $a, b \in \mathbb{N}_+$ ,  $n = a + b - 1$  și fie  $U_1, U_2, \dots, U_n$  variabile aleatoare uniforme pe  $[0, 1]$ , independente. Fie  $U_{(1)} < U_{(2)} < \dots < U_{(n)}$  statisticile de ordine obținute prin ordonarea valorilor  $U_1, U_2, \dots, U_n$ . Atunci  $U_{(a)} \sim \text{Beta}(a, b)$ .*

**Teorema 3.** *Fie  $0 < a < 1$ ,  $0 < b < 1$  și  $U_1, U_2$  variabile aleatoare uniforme pe  $[0, 1]$  independente. Dacă  $V = U_1^{\frac{1}{a}}$ ,  $T = U_2^{\frac{1}{b}}$ , atunci repartiția variabilei  $X = \frac{V}{V+T}$  condiționată de  $V + T < 1$  este  $\text{Beta}(a, b)$ .*

**Teorema 4.** Fie  $0 < a < 1$ ,  $b > 1$  și  $U_1, U_2$  variabile aleatoare uniforme pe  $[0, 1]$  independente. Dacă  $V = U_1^{\frac{1}{a}}$ ,  $T = U_2^{\frac{1}{b-1}}$ , atunci repartiția variabilei  $V$  condiționată de  $V + T < 1$  este  $\text{Beta}(a, b)$ .

Dem:

Observăm că pentru  $x \in [0, 1]$  avem:

$$F(x) = P(V < x) = P(U_1^{\frac{1}{a}} < x) = P(U_1 < x^a) = x^a$$

De unde rezultă că densitatea de repartiție a lui  $V$  este

$$f(x) = ax^{a-1}, \quad x \in [0, 1].$$

Asemănător, densitatea de repartiție a lui  $T$  este:

$$h(y) = (b-1)y^{b-2}, \quad y \in [0, 1].$$

Prin urmare, densitatea comună de repartiție a variabilelor  $V, T$  independente este:

$$g(x, y) = a(b - 1)x^{a-1}y^{b-2}.$$

De unde rezultă că:

$$P(V + T < 1) = a(b - 1) \int_0^1 \left( \int_0^{1-x} y^{b-2} dy \right) x^{a-1} dx = aB(a, b).$$

Deci densitatea comună a variabilelor  $V, T$  condiționată de  $V + T < 1$  este:

$$p(x, y) = \frac{b - 1}{B(a, b)} x^{a-1} y^{b-2}, \quad x \in [0, 1], \quad y \in [0, 1].$$

Atunci densitatea lui  $V$  condiționată de  $V + T < 1$  este

$$q(x) = \int_0^{1-x} p(x, y) dy = \frac{1}{B(a, b)} x^{a-1} (1 - x)^{b-1}$$

ceea ce demonstrează teorema.



Algoritmul rezultat din Teorema 3 este următorul:

### **Algoritm Beta3**

**Intrare:**  $0 < a, b < 1$

P1: Se generează  $U_1, U_2 \sim U(0, 1)$  independente;

P2:  $V = U_1^{\frac{1}{a}}$ ,  $T = U_2^{\frac{1}{b}}$ ;

P3: Dacă  $V + T < 1$  mergi la P4, altfel mergi la P1;

P4:  $X := \frac{V}{V+T}$ ;

**Ieșire:** Variabila aleatoare  $X$ .

Probabilitatea de acceptare a acestui algoritm de respingere este:

$$p_a = P(V + T < 1) = \frac{ab}{a+b} B(a, b).$$

Probabilitatea de acceptare a algoritmului bazat pe Teorema 4 este:

$$p_a = P(V + T < 1) = aB(a, b)$$

# Repartiția normală

Vom prezenta algoritmi de simulare pentru  $X \sim N(0, 1)$ .

**1. Metoda bazată pe teorema limită centrală**  
(a fost prezentată).

**2. O metodă de compunere-respingere**

Fie  $X_1$  variabila aleatoare cu densitatea:

$$f_1(x) = \begin{cases} \sqrt{\frac{2}{\pi}} e^{-\frac{x^2}{2}}, & \text{dacă } x \geq 0 \\ 0, & \text{dacă } x < 0 \end{cases}$$

Fie  $X_2 = -X_1$ , atunci densitatea de repartiție a variabilei  $X_2$  este:

$$f_2(x) = \begin{cases} \sqrt{\frac{2}{\pi}} e^{-\frac{x^2}{2}}, & \text{dacă } x < 0 \\ 0, & \text{dacă } x \geq 0 \end{cases}$$

Prin urmare densitatea variabilei aleatoare  $X \sim N(0, 1)$  se poate scrie:

$$f(x) = \frac{1}{2} f_1(x) + \frac{1}{2} f_2(x)$$

adică  $f(x)$  este o compunere discretă a densităților  $f_1(x)$  și  $f_2(x)$ .

Pentru generarea variabilei aleatoare  $X_1$  folosim următoarea teoremă:

**Teorema 5.** *Fie  $h(x)$  densitatea de repartiție a unei variabile aleatoare  $Exp(1)$ . Atunci dacă înfășurăm  $f_1(x)$  cu  $h(x)$  avem*

$$\frac{f_1(x)}{h(x)} \leq \alpha = \sqrt{\frac{2e}{\pi}}$$

Dem:

Observăm că:

$$r(x) = \frac{f_1(x)}{h(x)} = \sqrt{\frac{2}{\pi}} e^{-\frac{x^2}{2} + x}$$

iar ecuația  $r'(x) = 0$  are soluția  $x_0 = 1$  care este un punct de maxim pentru  $r(x)$ , adică

$$r(x) \leq r(x_0) = \sqrt{\frac{2e}{\pi}}$$

ceea ce demonstrează teorema.

Deci un algoritm pentru generarea variabilei  $X \sim N(0, 1)$  este următorul:

### **Algoritm Norm2**

#### **Intrare:**

P1: Se generează  $U \sim U(0, 1)$ ;

P2: Se generează  $Y \sim Exp(1)$ ;

P3: Dacă  $U \leq e^{-\frac{Y^2}{2} + Y - 0.5}$  mergi la P4, altfel mergi la P1;

P4:  $X_1 := Y$ ;

P5: Se generează  $U \sim U(0, 1)$ ;

P6: Dacă  $U \leq 0.5$  atunci  $s = 1$ , altfel  $s := -1$ ;

P7:  $X := sX_1$ .

**Ieșire:** Variabila aleatoare  $X$ .

Se observă că probabilitatea de acceptare este:

$$p_a = \sqrt{\frac{\pi}{2e}} \approx 0.72$$

adică în medie, din patru perechi  $(U, Y)$  trei sunt acceptate pentru a genera un  $X_1$ .

### 3. Metoda polară

**Teorema 6.** *Dacă variabilele  $U_1, U_2$  sunt uniforme pe  $[0, 1]$  și independente, atunci variabilele aleatoare*

$$Z_1 = V_1 \sqrt{\frac{-2 \log S}{S}}, \quad Z_2 = V_2 \sqrt{\frac{-2 \log S}{S}} \quad (3)$$

cu

$$V_1 = 2U_1 - 1, \quad V_2 = 2U_2 - 1, \quad S = V_1^2 + V_2^2, \quad S < 1$$

sunt variabile  $N(0, 1)$  independente.

Dem:

Trebuie să arătăm că repartiția bidimensională comună a variabilelor  $Z_1$  și  $Z_2$  condiționată de  $\{S < 1\}$  este repartiția comună a două variabile normale independente.

Observăm că  $(V_1, V_2)$  este un vector aleator uniform pe suprafața mărginită de pătratul  $I^2 = [-1, 1] \times [-1, 1]$  iar  $V_1, V_2$  sunt uniforme pe  $[-1, 1]$  și independente. Condiția  $S < 1$  face ca repartiția vectorului  $(V_1, V_2)$  condiționată de  $\{S < 1\}$  să fie vector uniform pe suprafața mărginită de cercul unitate. De aceea putem să-i scriem pe  $V_1$  și  $V_2$  în funcție de coordonatele polare:

$$V_1 = R \cos \theta, \quad V_2 = R \sin \theta$$

cu  $R$  și  $\theta$  variabile aleatoare  $0 \leq R \leq 1$ ,  $0 \leq \theta \leq 2\pi$ . Identificând aceste relații cu (3) rezultă că:

$$S = R^2, \quad Z_1 = \sqrt{-2 \log S} \cos \theta, \quad Z_2 = \sqrt{-2 \log S} \sin \theta \quad (4)$$

Dar și  $Z_1, Z_2$  se pot exprima în coordonate polare:

$$Z_1 = R' \cos \theta', \quad Z_2 = R' \sin \theta' \quad (5)$$

cu  $R', \theta'$  variabile aleatoare,  $R' \in \mathbb{R}_+, 0 \leq \theta \leq 2\pi$ . Din (4) și (5) rezultă că:

$$\theta' = \theta, \quad R' = \sqrt{-2 \log S}$$

și pentru că  $V_1, V_2$  sunt independente pe  $I^2$ , atunci și perechile  $R, \theta$  și  $R', \theta'$  sunt independente.

Deoarece  $(V_1, V_2)$  are o repartiție uniformă pe cercul unitate rezultă că  $\theta = \theta'$  are o repartiție uniformă pe  $[0, 2\pi]$ , adică are densitatea:

$$\varphi(\theta) = \begin{cases} \frac{1}{2\pi} & \text{dacă } \theta \in [0, 2\pi] \\ 0, & \text{altfel.} \end{cases}$$

Să determinăm acum repartiția lui  $R'$ :

$$F(r) = P(R' \leq r) = P(\sqrt{-2 \log S} \leq r).$$

Dar  $S = R^2$  este uniformă pe  $[0, 1]$ . Atunci rezultă că:

$$F(r) = P(S \geq e^{-\frac{r^2}{2}}) = 1 - e^{-\frac{r^2}{2}}.$$

Prin urmare densitatea de repartiție a variabilei  $R'$  este:

$$\psi(r) = re^{-\frac{r^2}{2}}, \quad r \in [0, 1]$$

Vom determina acum funcția comună de repartiție a variabilelor  $Z_1, Z_2$  condiționată de  $S < 1$ . Pentru aceasta considerăm domeniile:

$$D_{(r,\theta)} = \{(r, \theta) | r \cos \theta \leq z_1, r \sin \theta \leq z_2\}$$

$$D_{(x,y)} = \{(x, y) | x \leq z_1, y \leq z_2\}.$$

Atunci

$$F(z_1, z_2) = P(Z_1 \leq z_1, Z_2 \leq z_2) = \int \int_{D_{(r,\theta)}} \frac{1}{2\pi} e^{-\frac{r^2}{2}} r dr d\theta$$



Facem schimbările de variabile:

$$\theta = \arctg \left( \frac{x}{y} \right), \quad r = \sqrt{x^2 + y^2}$$

și rezultă că:

$$\begin{aligned} F(z_1, z_2) &= \frac{1}{2\pi} \int \int_{D_{(x,y)}} e^{-\frac{x^2+y^2}{2}} dx dy = \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x_1} e^{-\frac{x^2}{2}} dx \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x_2} e^{-\frac{y^2}{2}} dy \end{aligned}$$

ceea ce demonstrează teorema.

Algoritmul de generare se deduce ușor din teoremă și el produce simultan două valori de selecție ale unor variabile  $N(0, 1)$  independente.

Observăm că se resping valorile pentru care  $S \geq 1$  iar probabilitatea de acceptare este:

$$p_a = \frac{\text{aria cercului } C(0, 1)}{\text{aria pătratului } [-1, 1] \times [-1, 1]} = \frac{\pi}{4}$$

Din demonstrația teoremei rezultă că variabilele  $Z_1, Z_2$  pot fi simulate și cu formulele:

$$Z_1 = \sqrt{-2 \log U_1} \cos(2\pi U_2), \quad Z_2 = \sqrt{-2 \log U_1} \sin(2\pi U_2) \quad (6)$$

În acest caz nu se fac respingeri, dar complexitatea algoritmului poate fi mai mare decât în cazul (3) din cauza funcțiilor trigonometrice și a funcției logaritm.