



## 380360394-PEAG - Summary Programare evolutiva si algoritmi genetici

Programare evolutiva si algoritmi genetici (Academia de Studii Economice din București)

# ALGORITHM EVOLUTIV

## I SCHEMA GENERALĂ

Pas 1. Fie  $f: D \rightarrow R$  o funcție ce trebuie max.

Pas 2. Generăm aleator un set inițial de candidați  $P_0$ .

Pas 3. Avem un  $x \in P_0$ :  $f(x) \Rightarrow$  calitatea / fitnessul candidatului  $x$ .

Pas 4: Determinăm populația următoare:

$\rightarrow$  selecționăm o mulțime  $B_{P_0} \subset P_0 \Rightarrow$  cei mai buni membri ai populației curente

$\rightarrow$  aplicăm operatorii de combinare / mutație pe  $B_{P_0} \Rightarrow$  progeniturile sale intră în competiție cu  $P_0$  pentru a forma noua populație

## II. Algoritmul genetic

$i \leftarrow 0$

Pas 1. inițializăm pop  $P_0$

Pas 2. evaluăm candidații:  $\forall x \text{ det. } f(x)$

Pas 3. Repetă

3.1. Selectează mulțimea de părinți  $B_{P_i}$

3.2. Recombina perechi (n-tepluri) de părinți

3.3 efectuează mutație pe progenitură

3.4. evaluează noii candidați

3.5. selectează individul pentru constituirea generației  $P_{i+1}$ ;

3.6  $i \leftarrow i+1$ ;

până când (condiție satisfăcută)

### III. DEFINIȚII

a) spațiul genotipului = mulțimea <sup>de nr al</sup> intervalului dat în problemă

ex:  $f: [-1, 1] \rightarrow \mathbb{R}$  → GENOTIP

b) COMPONENTE EA:

- REPRESENTAREA
- FUNCȚIA DE FITNESS
- POPULAȚIA
- MECANISMUL DE SELECTARE AL PĂRINȚILOR
- OPERATORII DE VARIATIE (RECOMB & MUTAȚIE)
- MECANISMUL DE SELECTARE A GENURII
- DEFINIREA MODULUI DE ÎNȘTARE (DET POP. ÎNȘTARE)
- DEFINIREA CONDIȚIEI TERMINALE

c) fenotip = soluție posibilă în contextul pb de rezolvat

d) genotip = cromozom; reprezentarea fenotipurilor în context EA; un genotip conține valori (alele) plasate în poziții numite gene.



## IV. REPRESENTAREA

→ Denumirea : codificare / decodificare,  
struct de date utilizată

### **I. REPRESENTARE BINARĂ**

Pentru un individ  $x$  de lungime / dim  $n$  se parcurge ~~la fiecare iterație~~ individul cu eu  
for de la 1 la  $n$  și se generează la fiecare  
iterație un număr  $r$  aleator. Dacă  $r$  este  
mai mare decât probabilitatea de mutație dată,

bitul de pe poziția  $i$  se menține:

ex :  $r = 10$  prob =  $0,7 \Rightarrow i = 2$   
 $r > \text{prob} \Rightarrow x(i) = 1010$   
 $\Downarrow$   
 $\sim x(i) = \text{1110}$

## V. OPERATORI DE VARIATIE

→ operatorul mutație : operator unar  
 - operator de recombinare : operator binar  
 ↳ recombinare stochastică (aleator)

↳ PT GA CEL MAI UTILIZAT OPERATOR DE  
VARIATIE !!

## VI. HILL CLIMBING ?

# ALGORITHM GENETIC

pag 4

I & II: respectă structura unui EA

## III. REPRESENTARE

- ① SECVENȚE BINARE, (~~VER. PAG 3~~)
- ② SECVENȚE NUMERE ÎNTREGI
- ③ — " — REALE
- ④ PERMUTĂRI

## IV. OPERATORUL MUTAȚIE

- ① REPRESENTARE BINARĂ (VERSO)
- ② MUTAȚIA ÎN REPR. NR. ÎNTREGI

### A) ALEATOARE

Se parcurge cu for de la 1 la  $n$  din pop  $x$ ,  
se generează la fiecare iterație un nr  $p$ .  
Dacă  $p >$  probabilitatea de mutație  $\rightarrow x(i)$   
și schimbă valoarea cu una generată  
aleator.

ex: ~~1234~~  $x = 1234$

$i = 1$ :  $p = 0 > \text{prob} = 1 \rightarrow 1234$

$i = 2$ :  $p = 3 > \text{prob} = 1 \rightarrow 1534$

$\vdots$

### B) DE TIP FLUJ

Parcurgem for de la 1 la  $n$  din pop  $x$ ,  
generăm la fiecare iterație un nr  $p$   
Dacă  $p > \text{prob} \Rightarrow x(i) = x(i) + \text{o valoare}$   
aleatoare

$x(i)$  trebuie să rămână în interval!



ex:  $x = 1234$   $[0, 10]$

$i = 1$  :  $p = 3$ ,  $prob = 0,5 \Rightarrow$  val gen aleator : 3  
 $\Rightarrow 4234$

:

### ③ NUMERE REALE

① DE TIP FLUJ : LATEL

② MUTAȚIE UNIFORMĂ (CA ALEATOARE)

### ④ PERMUTĂRI : ① INTERSCHIMBARE

Se generează aleator două poziții și se schimbă valorile între ele.

### ② INSERARE

Se gen aleator 2 poziții  $p_1$  și  $p_2$ . Elementul de pe poziția  $p_2$  se mută pe poziția  $p_1 + 1$ , iar toate elementele de la  $p_1 + 1$  se mută cu unul mai la dreapta.

ex:  $1234567$

$p_1 = 2$   
 $p_2 = 5 \} \rightarrow 1253467$

### ③ INVERSARE

din Se gen aleator 2 poziții  $p_1$  și  $p_2$  și elementele  $[p_1, p_2]$  se inversează

$1234567$

$p_1 = 2$   $p_2 = 5 \Rightarrow 1543267$

# ① AMESTEC

Se dau  $p_1, p_2$  și elementele din  $[p_1, p_2]$  se amestecă între ele.

1 2 3 4 5 6 7

$p_1 = 2$   $p_2 = 5 \Rightarrow$  1 4 2 5 3 6 7

## V. RECOMBINAREA

### 1. RB

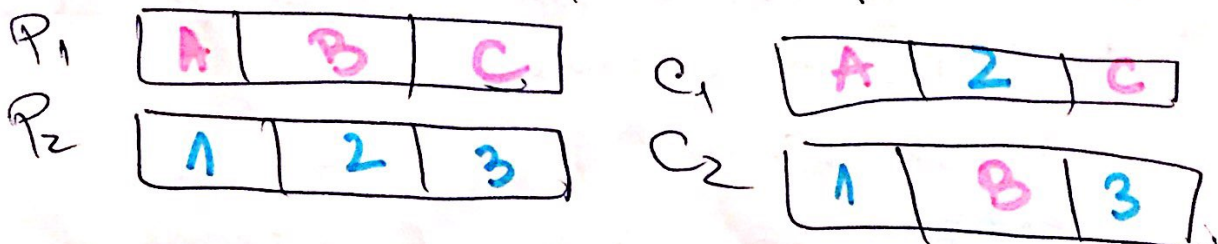
#### ① UNIPUNCT

- 2 părinți de dim  $m$  și 2 copii
- generăm aleator o poziție  $p$
- primarele  $p$  elemente din  $P_1$  se copiază în  $C_1$  (de la 1 la  $p \rightarrow$  poziție) și de la  $p+1$  la  $m$   $P_2$ . În  $C_2$ :  $P_2$  1- $p$ ,  $P_1$ :



#### ② MULTIPUNCT

- la fel doar că se generează aleator  $k$  poziții și se pun în copii aleator





2. RNT① INCRUCISARE UNIFORMĂ

- 2 părinți de dinu 2 și 2 copii
- 0 probab de recombinare

- for ( $i = 1 \rightarrow n$ )
- Generăm un  $r$ , nr aleator
- Dacă  $r < \text{prob de recombinare}$

$$C_1(i) = P_1(i)$$

$$C_2(i) = P_2(i)$$

altfel

$$C_1(i) = P_2(i)$$

$$C_2(i) = P_1(i)$$

③ RNR① DE TIP DISCRET

- 2 părinți : copii rezultă amestecând el. părinților

② ARITMETICĂ SIMPLĂ - AVEM  $\alpha = [0, 1]$  dat

- se gen aleator o poziție  $i$

- for  $j = 1, i-1$ :

$$C_1(j) = P_1(j)$$

$$C_2(j) = P_2(j)$$

- for  $j = i, n$ :

$$C_1(j) = \alpha \cdot P_2(j) + (1 - \alpha) \cdot P_1(j)$$

$$C_2(j) = \alpha \cdot P_1(j) + (1 - \alpha) \cdot P_2(j)$$



C) ARITMETICĂ (SINGULARĂ) : - - -

for  $j = 1, i-1$

$$Q_1(j) = P_1(j)$$

$$Q_2(j) = P_2(j)$$

for  $j = i$

$$Q_1(j) = \alpha \cdot P_2(j) + (1-\alpha) \cdot P_1(j)$$

$$Q_2(j) = \alpha \cdot P_1(j) + (1-\alpha) \cdot P_2(j)$$

for  $j = i+1, n$

$$Q_1(j) = P_1(j)$$

$$Q_2(j) = P_2(j)$$

D) ARITMETICĂ (COMPLETĂ) : - - -

for  $i = 1, n$

$$Q_1(j) = \alpha \cdot P_2(j) + (1-\alpha) \cdot P_1(j)$$

$$Q_2(j) = \alpha \cdot P_1(j) + (1-\alpha) \cdot P_2(j)$$

④ RP : to be continued

## ④ REPREZENTARE PERMUTĂRI

### A) CICLURI (C X crossover)

Se folosează cicluri cu cele 2 permutări. Apoi sub fiecare se trece numărul ciclului. În copiu, se copiază din  $x_1$  în  $x_2$  cele din ciclurile impare (și din  $y_1$  în  $y_2$  tot cele din ciclurile impare) iar din  $y_1$  în  $x_2$  cele din ciclurile pare și din  $x_1$  în  $y_2$  cele din ciclurile pare.

ex :

1	2	3	4	5	6	7
(1) ↓	↓ (3)		↓ (4)	↓ (2)		↓ (5)
5	4	6	7	2	3	1

Pas 1 : Găsim începând cu părintele 1 ciclurile definite de părinți

→ 1 → 5 → 2 → 4 → 7 ciclul aferent  $P_1$ .

Pas 2 : Copiem în copilul 1 elem din părintele 1 din ciclul

1 2 6 4 5 3 7

Pas 3 : Umplem spațiile respective cu elementele de pe poz respective din părintele 2.

Pas 4 : Repetăm pentru  $C_2$  începând ciclul de la elem 1 al  $P_2$ .



## ⓑ ORDER



pag 10

Se generează 2 poziții  $x_1$  și  $x_2$ . Se copiază în  $C_1$  pe pozițiile de la  $p_1$  la  $p_2$  elementele de pe pozițiile  $p_1$  la  $p_2$  din  $P_1$  și în  $C_2$  pe pozițiile de la  $p_1$  la  $p_2$  elementele de la  $p_1$  la  $p_2$  din  $P_2$ .  
Apoi în  $C_1$  de la pozițiile  $p_2+1$  se copiază din  $P_2$  începând de la poziția  $p_2$  elementele care nu se repetă. Analog la  $C_2$ .

## ⓒ MUCHII (măș idea)

## ⓒ PMX (Partial - Mapped Crossover)



## (VI) SELECȚIA

### (A) DE TIP TURNEU

Sinteza - o populație cu  $n$  indivizi se va genera o populație de părinți tot cu  $n$  indivizi. Pentru a genera părinții parcurgem cu un forț de la 1 la  $n$  și la fiecare iterație se generează aleator un nr  $k$ . Selecționăm aleator  $k$  indivizi din populație și pentru cei  $k$  indivizi calculăm funcția obiectivă. Indiciul cu funcția obiectivă cu cea mai mare valoare devine părinte. Procedul se repetă de  $n$  ori (adică se face for-ul).

! Pentru ruletă și SUS se folosesc ranguri și FPS.

Pentru calculat FPS:

Pentru fiecare individ  $i$  se calculează funcția obiectivă, apoi se calculează suma funcțiilor obiective. Valoarea FPS-ului unui individ este egală cu funcția lui obiectivă / suma forț ob. (raportul)

Deoarece calculăm FPS-ul, se calculează probabilitatea cumulată: primul individ are ca probabilitate cumulată FPS-ul lui, următorul are FPSul celui de dinainte sau.

Pentru calculat ranguri:

Formula  $2 - S / \text{dim pop} * (2 * i (S - 1) / \text{dim} (\text{dim} - 1))$   
Apoi calculăm prob cumulată.



## B) DE TIP RULETĂ

Cu un  $for$   $i$  de la 1 la  $m$  se parcurge populația  $J$ . La fiecare iterație se generează un nr random  $r$ , apoi se compară valoarea lui  $r$  cu proba cumulată a elem  $k$ . Dacă  $r < g(k)$  părintele =  $g(k)$  astfel se merge la următorul individ din populație și se verifică dacă proba cumulată a urm. e mai mică decât  $r$ .

## VII Schimbarea generației

- A) Elitistă: îi luăm pe cei mai buni
- B) vârstă: îi luăm pe cei mai tineri

## TIPURI DE PROBLEME

- ① ONE - MAX - funcția e suma elem vectorului
  - reprezentare binară pt indivizi
  - pop inițială generată aleator
  - recombinația: încrucișare unipunct
  - mutația de la rept binară
  - selectia: SUS / Turnir cu  $k=2$ .
  - Schimb de generații: vârstă
  - cond de oprire: un anumit nr de iterații

- ② COMIS VOIAJOR - are funcția lui cu distanță și cost

- indivizi: permutări
- p.i = generată aleator
- CX
- inversare
- SUS cu ranguri / Turnir
- elitist / vârstă
- oprire: un anumit nr de iterații

### ③ PLANIFICAREA ACTIVITĂȚILOR

- permutări
- generație aleatoare
- PMX
- interschimbare
- sus cu FPS
- elitist
- după planificarea tuturor act.

### ④ PORTOFOLIU

- sir de nr reale (shortselling egal între -1 și 1 și la fără shortselling sunt între 0 și 1).
- gen aleator dar se ține cont de shortselling / fără shortselling
- aritmetică (,) completă
- mutație de tip flip
- sus
- elitist
- val maximă a portofoliului / nr de iterații

### ⑤ CELE 8 REGINE

- permutări
- gen aleator
- ORDER CROSSOVER
- INTERSCHIMBARE
- turneu  $k=2$
- elitist
- soluție sau nr de iterații