



Facultatea de Electronică,
Telecomunicații și
Tehnologia Informației

SISTEME INTELIGENTE DE SUPORT DECIZIONAL

Ș.l.dr.ing. Laura-Nicoleta IVANCIU

Curs 10 – Calcul evoluționist. Algoritmi genetici.

Cuprins

- Calcul evoluționist
- Algoritmi genetici

Calcul evoluționist

- Bazele biologice
- Principii
- Operații specifice
- Domenii
- Structura generală a unui algoritm evolutiv

Să ne aducem aminte...



Logica fuzzy oferă posibilitatea **aproximării**.

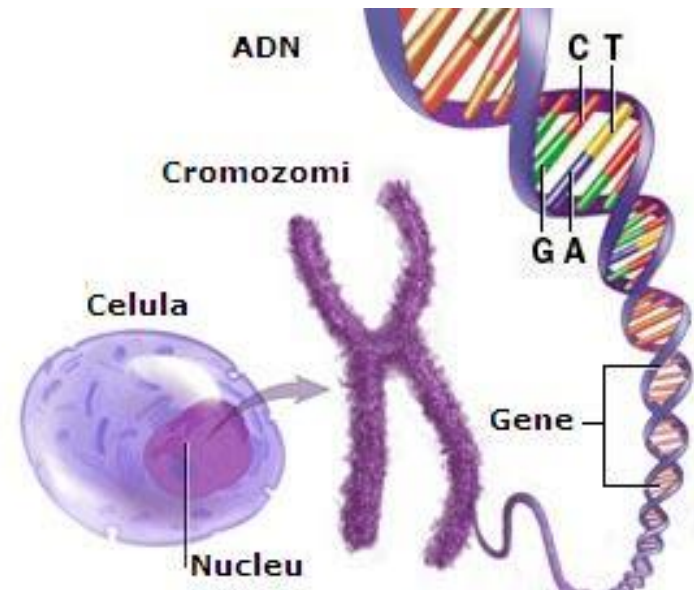
Rețelele neuronale au capacitatea de **a învăța și de a se adapta**.

Algoritmii genetici realizează o **căutare “sistematică”** a soluției.

Bazele biologice ale CE

Genetica = ramură a biologiei care studiază fenomenele și legile eredității și variabilității organismelor

Cromozom = structură ordonată de elemente, numite gene, ale căror valori determină caracteristicile unui individ, și care transmite informația genetică.

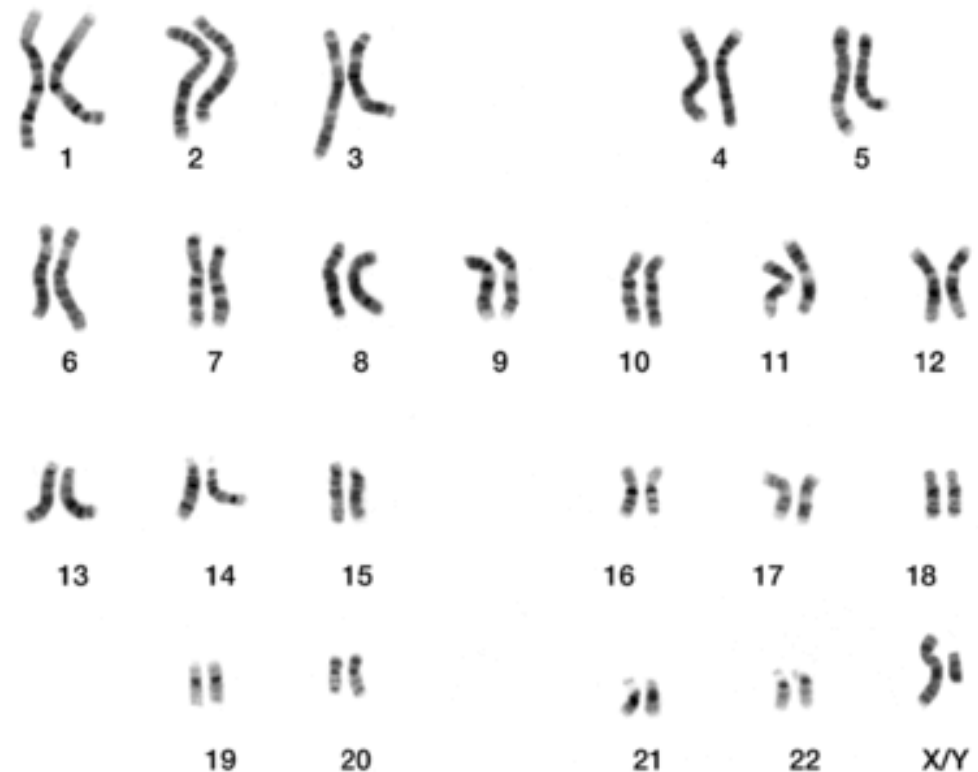


Bazele biologice ale CE

La om: 46 de cromozomi

22 perechi – autozomi

1 pereche – heterozomi



În fiecare pereche, un cromozom este derivat de la mamă și unul de la tată.

Principii ale CE

- căutare în spațiul soluțiilor, bazată pe **principiul evoluției naturale** (Darwin – teoria evoluționistă – supraviețuirea celui mai bun)
- pentru găsirea populației finale (soluție), se lucrează cu o populație de soluții potențiale, care **evoluează**

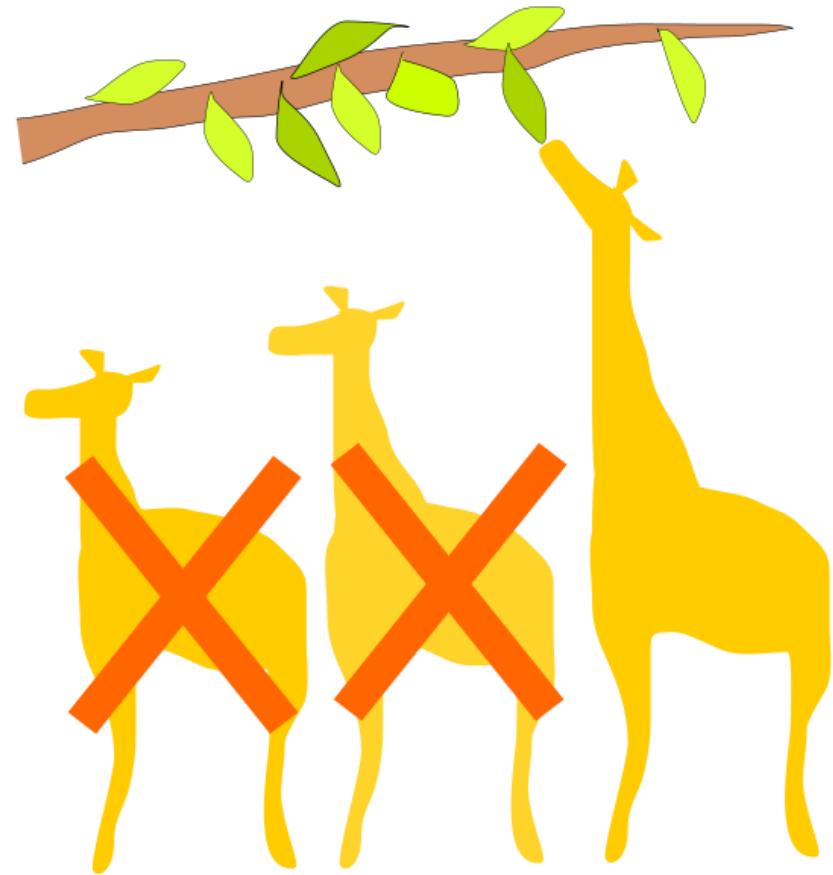
Evoluție = indivizii din noua generație sunt mai adaptați la mediu decât indivizii din care au fost creați

- direcționarea căutării se face prin transformări specifice asupra populației, similare cu **procesele naturale**: selecție, recombinare, mutație

Operații specifice CE

Selecție

mai apropiat \equiv mai adecvat



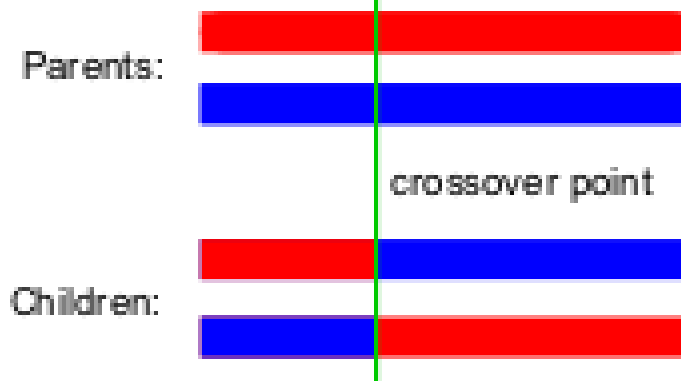
- șanse **mai mari** de a fi ales pentru a contribui la generarea noii populații

Operații specifice CE

Recombinare (încrucișare)

crossover

single point

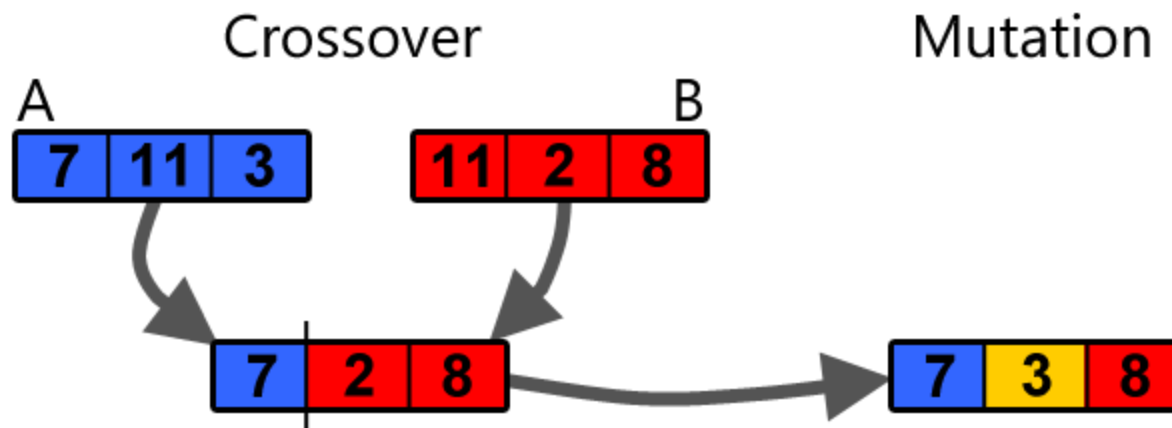


double point



Operații specifice CE

Mutație – permite apariția unor trăsături și gene care nu ar fi putut să apară exclusiv prin selecție sau încrucișare.



- se asigură diversitatea populației

Domenii ale CE

- **algoritmi genetici (genetic algorithms)**
- programare evoluționistă (evolutionary programming)
- strategii evolutive (evolution strategies)
- programare genetică (genetic programming)
- optimizarea roiurilor de particule (particle swarm optimization)

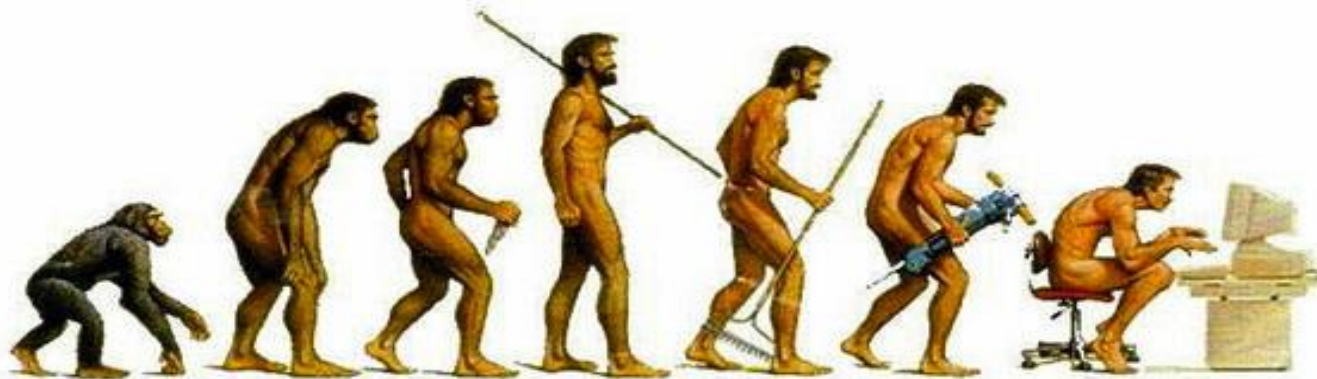
Algoritmi genetici

- Principii
- Structura AG
- Reprezentarea variabilelor
- Generarea populației inițiale
- Funcția de adecvare (fitness)
- Selecție/recombinare/mutație

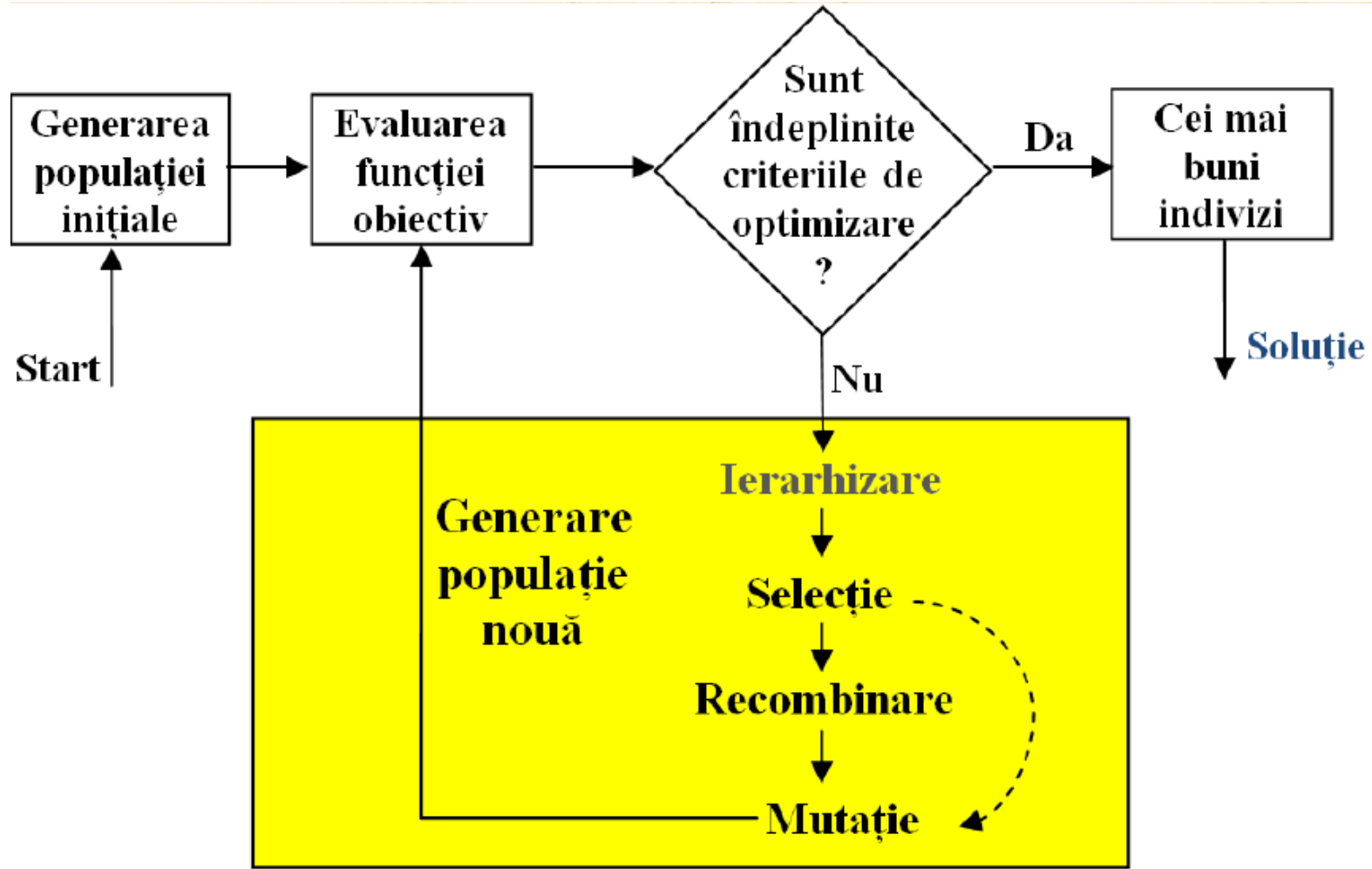
- tehnici de căutare și optimizare, având ca punct de pornire metafora biologică a **moștenirii genetice** și **evoluției naturale**

John Henry Holland, 1960

- teoria evoluționistă a lui Darwin (1896) – “survival of the fittest”
- populația evoluează, prin mecanisme de inspirație biologică: selecție, încrucișare, mutație



Structura unui AG



Formularea problemei de optimizare

Găsește vectorul soluțiilor x
 care optimizează (minimizează) funcția $f(x)$
 supus la constrângeri de inegalitate, de egalitate și de mărginire

➤ Formularea problemei de optimizare este de tipul
minimizează sau maximizează (se poate trece de la una la cealaltă utilizând
 semnul minus în fața funcției sau inversa funcției)

Găsește vectorul soluțiilor $x = [x_1, x_2, \dots, x_N]^T$

care minimizează funcția $f(x)$

supusă la constrângeri

de inegalitate: $g_i(x) \leq 0, \quad i = 1, \dots, M$

de egalitate: $h_j(x) = 0, \quad j = 1, \dots, P$

de mărginire: $lb_k \leq x_k \leq ub_k, \quad k = 1, \dots, N$

Reprezentarea variabilelor

individ al populației \equiv cromozom

cromozom \equiv colecție de gene

genă \equiv variabilă



Reprezentarea variabilelor:

binară – fiecare individ este un șir de biți

reală – fiecare individ este un șir de numere reale

Generarea populației inițiale

- generare stocastică (aleatoare)
- câțiva indivizi promițători + indivizi aleatori

De respectat:

- varietate mare a indivizilor
- mărime moderată a populației (50-500 indivizi)
- mărimea populației proporțională cu dimensiunea individului

Generarea populației inițiale

Exemplu:

Funcția De Jong – 2 variabile

$$x = [x_1, x_2]$$

$$F(x_1, x_2) = x_1^2 + x_2^2$$

$$D = [-10;10] * [-10;10]$$

x1	x2
6.2945	5.1548
8.1158	4.8626
-7.4603	-2.1555
8.2675	3.1096
2.6472	-6.5763
-8.0492	4.1209
-4.4300	-9.3633
0.9376	-4.4615
9.1501	-9.0766
...	...

25 de indivizi generați aleator

Funcția de adecvare (fitness) – în concordanță cu valoarea funcției obiectiv, definită pentru problema de optimizare de rezolvat.

În locul funcției de adecvare, se poate utiliza funcția obiectiv (funcția de optimizat).

Ordonarea (ierarhizarea) indivizilor se face după valorile funcției obiectiv.

Atribuirea adecvării – ordonare

- funcție de adecvare proporțională
- funcție de adecvare bazată pe rang (ranking) – liniară/nelinară
- funcție de adecvare multiobiectiv

Ierarhizarea indivizilor - ranking

- populația este ordonată în concordanță cu valoarea funcției obiectiv
- fiecare individ primește o **probabilitate de selecție** în vederea reproducerii, dependentă de **propria valoare** a funcției obiectiv și de valorile funcțiilor obiectiv a **celorlalți indivizi**

$$Adecv(Pos) = 2 - PS + 2(PS - 1) \frac{Pos - 1}{Nind - 1}$$

Pos – poziție; Nind – dimensiunea populației; PS – presiunea de selecție, $1 \leq PS \leq 2$

Cel mai **adecvat** individ are **$Pos = Nind$** și **cea mai mare valoare a funcției de adecvare**.
Cel mai **inadecvat** individ are **$Pos = 1$** și **cea mai mică valoare a funcției de adecvare**.

Selecție

Determinarea populației intermediare ce conține părinții care vor fi supuși operatorilor genetici de **recombinare și mutație** - selectarea indivizilor care vor produce **urmași**.

Metode:

➤ **aleatoare** – indivizii sunt introduși aleator în procesul de selecție, prin utilizarea unor probabilități de selecție, depedente de gradul de adecvare.

grad de adecvare mare -> șanse mari de fi selectat

selecție de tip **ruletă, turneu, proporțională**

➤ **deterministe** – indivizii cu cel mai mare grad de adecvare sunt întotdeauna selectați
selecție **prin trunchiere**

Elitism – supraviețuirea celui mai bun dintre indivizii generației, până la un moment dat.

Ex.: transferul direct al celui mai bun individ al generației curente în generația următoare

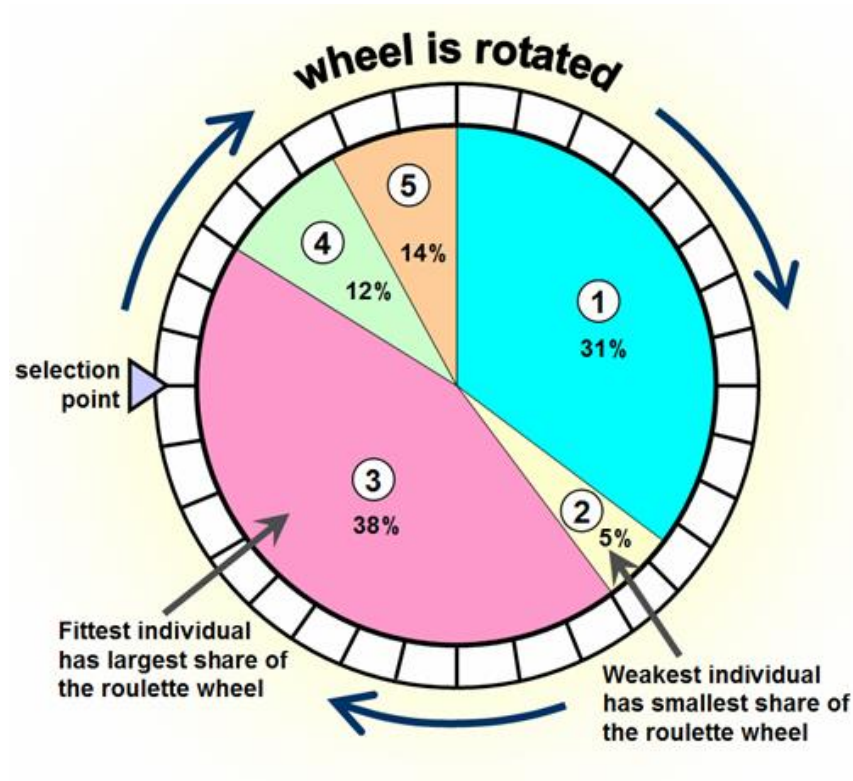
Selecție – selecția de tip ruletă

- indivizii sunt atribuiți la segmente contigue ale unei linii, astfel ca **lungimea** fiecărui segment să fie **proporțională** cu **gradul său de adecvare**
- se generează un număr aleator și este selectat individul al cărui segment corespunde valorii aleatoare
- pentru fiecare individ, se calculează o probabilitate de selecție:

$$Prob_sel(i) = \frac{Adecv(i)}{\sum_{j=1}^{N_{ind}} Adecv(j)}$$

Selecție – selecția de tip ruletă

- procesul este asemănător **roții de ruletă**, în care mărimea fiecărui “sector” este proporțională cu gradul de adecvare



Recombinare (încrucișare - crossover)

Produce noi indivizi (urmași) prin combinarea informațiilor conținute de doi sau mai mulți părinți.

Tipuri:

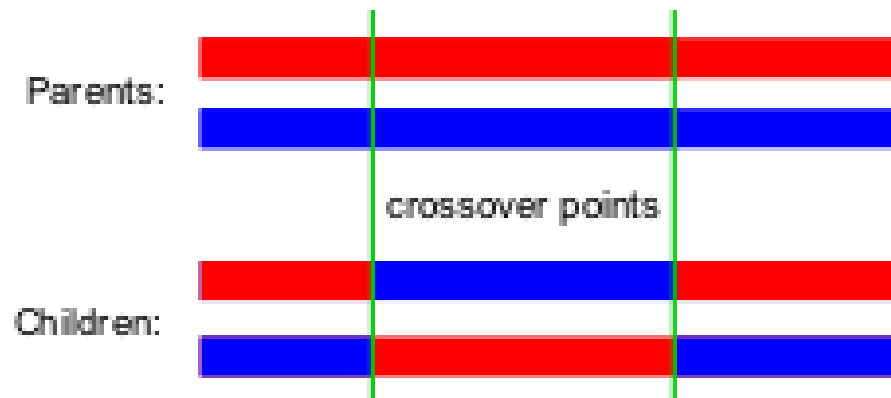
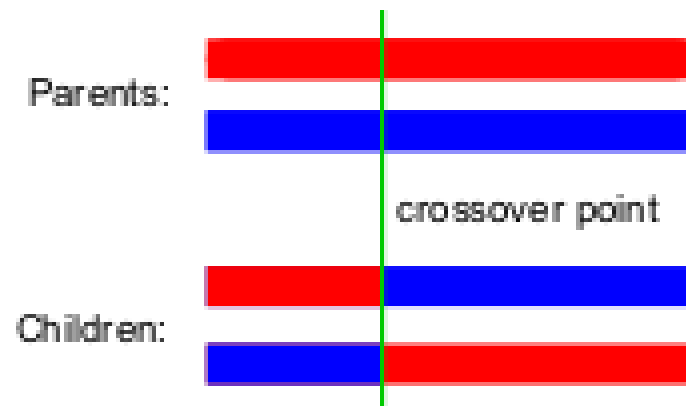
➤ **discretă**

➤ **pentru valori reale**

- recombinare intermediară
- recombinare liniară
- recombinare liniară extinsă

➤ **pentru valori binare**

- single point
- double point
- multiple point
- uniformă



Recombinare (încrucișare - crossover)

Recombinare intermediară

$$Var_j^U = a_j * Var_j^{P1} + (1 - a_j) * Var_j^{P2}, j = 1, 2, \dots, N \text{ var}$$

Variabila j a urmașului U este o combinație liniară între variabilele j ale celor 2 părinți, $P1$ și $P2$.

a – factor de scalare, generat aleator în intervalul $[-d; 1+d]$, uzual $d = 0.25$

Recombinare (încrucișare - crossover)

Recombinare intermediară

$$Var_j^U = a_j * Var_j^{P1} + (1 - a_j) * Var_j^{P2}, j = 1, 2, \dots, N \text{ var}$$

Exemplu:

Fie următorii doi indivizi, cu 3 variabile fiecare:

P1: 123 4 34

P2: 12 25 5

Fie valorile lui a după cum urmează:

Eșantion 1: 0.5 1.1 -0.1

Eșantion 2: 0.1 0.8 0.5

Să se calculeze urmașii U1 și U2.

Recombinare (încrucișare - crossover)

Recombinare intermediară

$$Var_j^U = a_j * Var_j^{P1} + (1 - a_j) * Var_j^{P2}, j = 1, 2, \dots, N \text{ var}$$

P1: 123	4	34	
P2: 12	25	5	
Eșantion 1: 0.5		1.1	-0.1
Eșantion 2: 0.1		0.8	0.5

U1: 67.5	1.9	2.1	$Var_1^{U1} = 0.5 * 123 + (1 - 0.5) * 12 = 67.5$
U2: 23.1	8.2	19.5	$Var_2^{U1} = 1.1 * 4 + (1 - 1.1) * 25 = 1.9$
			$Var_3^{U1} = -0.1 * 34 + (1 - (-0.1)) * 5 = 2.1$
			$Var_1^{U2} = 0.1 * 123 + (1 - 0.1) * 12 = 23.1$
			$Var_2^{U2} = 0.8 * 4 + (1 - 0.8) * 25 = 8.2$
			$Var_3^{U2} = 0.5 * 34 + (1 - 0.5) * 5 = 19.5$

Mutație

Produce noi indivizi prin modificarea aleatoare a indivizilor din populația curentă.

Tipuri:

- **pentru variabile binare**
- **pentru variabile reale** – se adaugă valori aleatoare variabilelor reale

$$\begin{aligned}Var_j^{Mut} &= Var_j + s_j * r_j * a_j, j = 1, 2, \dots, Nvar \\s_j &\in \{-1, +1\} \\r_j &= r * domain_i, r - mutation_range \\a_j &= 2^{-uk}, u \in [0, 1], k - mutation_precision\end{aligned}$$

Reinserție

- mod de generare a noii populații

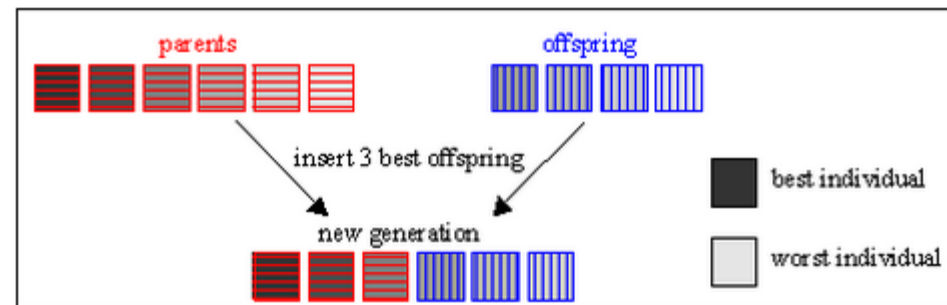
Tipuri:

➤ reinserție globală

Pură: număr urmași = număr părinți, toți părinții sunt înlocuiți de urmași

Uniformă: număr urmași < număr părinți, se înlocuiesc părinți în mod aleator


Elitistă: număr urmași < număr părinți, se înlocuiesc părinții cu cel mai scăzut grad de adecvare




Bazată pe fitness: număr urmași > număr părinți, doar cei mai buni urmași vor înlocui părinți

➤ reinserție locală – subpopulații

➤ Calcul evoluționist

- Bazele biologice
 - Principii
 - Operații specifice
 - Domenii
 - Structura generală a unui algoritm evolutiv
- 

➤ Algoritmi genetici

- Principii
 - Structura AG
 - Reprezentarea variabilelor
 - Generarea populației inițiale
 - Funcția de adecvare (fitness)
 - Selecție/recombinare/mutație
- 

În episodul următor: **Optimizare multiobiectiv. SISD bazate pe AG.**