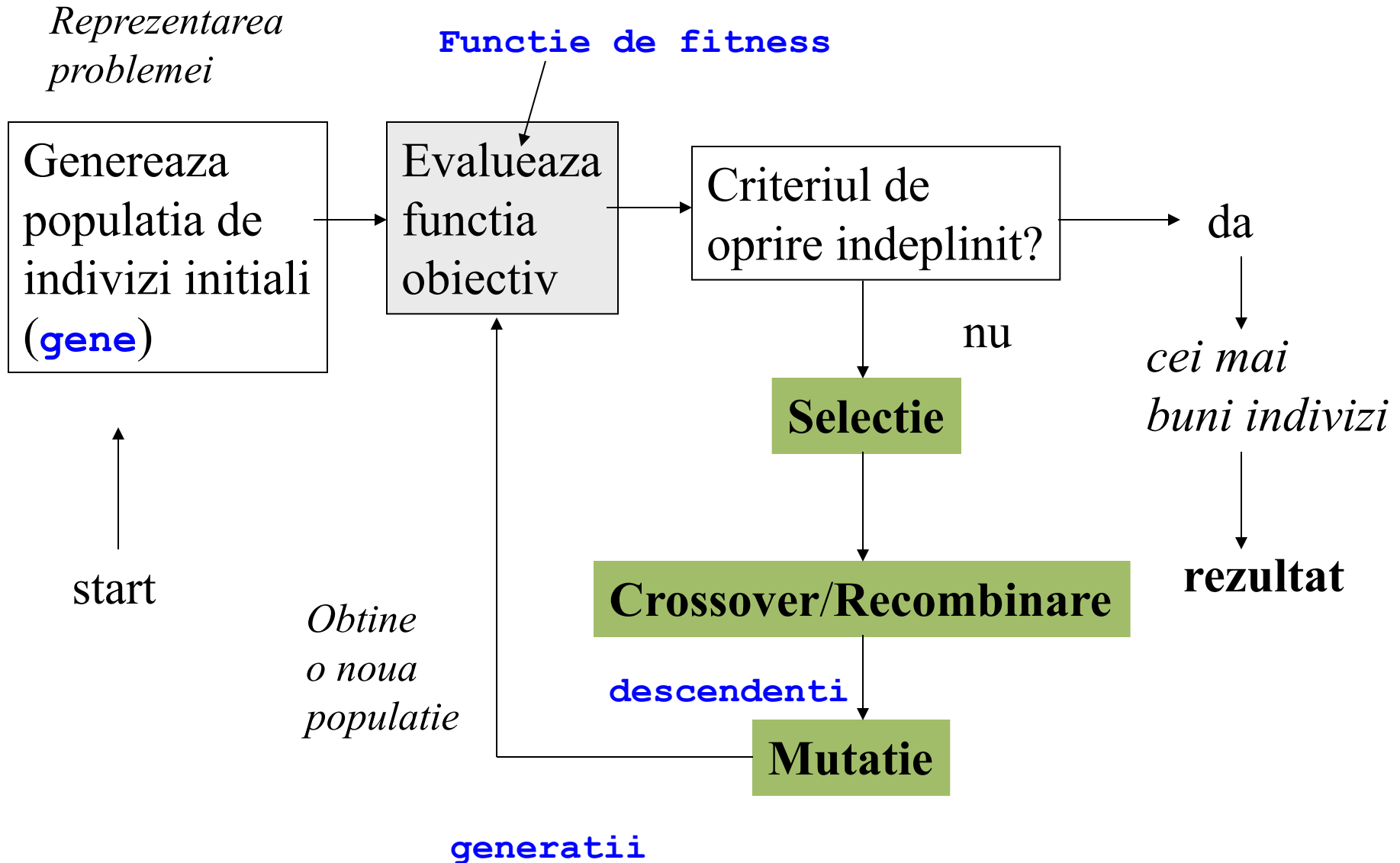


Algoritmi genetici

- **Optimizare**
 - Opereaza asupra unei populatii de indivizi = solutii potentiale – aplica principiul supravietuirii pe baza de adaptare
 - Fiecare generatie – o noua aproximatie a solutiei
 - Evolutia unei populatii de indivizi mai bine adaptati mediului
 - Modeleaza procese naturale: selectie, recombinare, mutatie, migrare, localizare
 - Populatie de indivizi – cautare paralela

Schema de baza



Modelare probleme

Populatie initiala

- Stabileste reprezentare – gena – individ
- Stabileste numar de indivizi in populatie
- Stabileste functia de evaluare (obiectiv)
- Populatia initiala (genele) creata aleator

Selectie

- **Selectie** – extragerea unui subset de gene dintr-o populatie exsidenta in fct de o definitie a calitatii - **functia de evaluare**
- Determina indivizii selectati pt recombinare si cati **descendenti** (offsprings) produce fiecare individ

Modelare probleme

Criteriul de oprire

- solutie care satisface criteriul
- numar de generatii
- platou pt cel mai bun fitness

Selectie

(1) *Selectia efectiva:* parintii sunt selectati in fct de fitness pe baza unuia din algoritmi:

- **roulette-wheel selection** (selectie ruleta)
- **tournament selection** (selectie turneu)

(2) *Reinserare:*

- Offspring – se genereaza noi indivizi care sunt inserati in noua populatie
- Elitism – se pastreaza un procent din cei mai buni indivizi din generatia curenta (care se copiaza in noua generatie)

Exemplu

Calculez maximul unei functii

- $f(x_1, x_2, \dots, x_n)$
- Sa se gaseasca x_1, x_2, \dots, x_n pt care f este maxima
- Utilizare algoritmi genetici

Reprezentare

1. Scalare variabile prin inmultire cu 10^n , unde n este precizia dorita

$$\text{Variabila} = \text{integer}(\text{Var} \times 10^n)$$

2. Variabilele se reprezinta binar
3. Variabilele se concateneaza - individ
4. Daca dorim semn:
 - Adauga o valoare si transforma in pozitiv sau
 - Un bit pentru semn

Calcul

1. Populatie initiala aleator
2. Selectie
3. Crossover
4. Mutatie
5. *Transforma* fiecare individ *in* variabilele: $x_1, x_2, \dots x_n$
6. Testeaza calitatea fiecarui individ: $f(x_1, x_2, \dots x_n)$
7. Testeaza daca calitatea celui mai bun individ este suficient de buna
8. **daca** da stop **altfel** mergi la 2

Atribuire fitness bazata pe rang

- Fitness-ul atribuit fiecarui individ depinde numai de pozitia lui intre indivizii din populatie
- Pozitia unui individ in populatie depinde de functia obiectiv
- $Pos = 1$ – cel mai putin bun
- $Pos = N_{ind}$ – cel mai bun
- Populatia este ordonata in functie de fitness

Selectia bazata pe ruleta

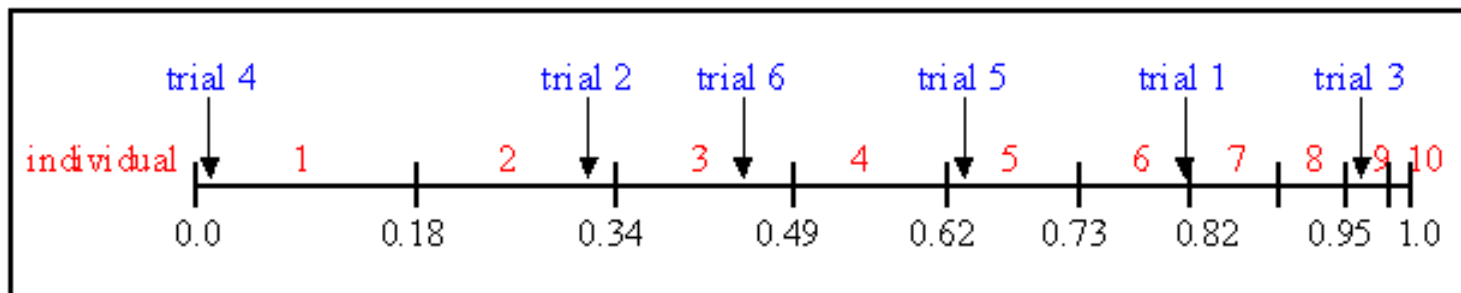
11 indivizi

Nr individ	1	2	3	4	5	6	7	8	9	10	11
Valoare fitness	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.0
Probabil. de selectie	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.0

6 numere aleatoare (distribuite uniform intre 0.0 si 1.0):

- 0.81, 0.32, 0.96, 0.01, 0.65, 0.42.

6, 2, 9, 1, 5, 3

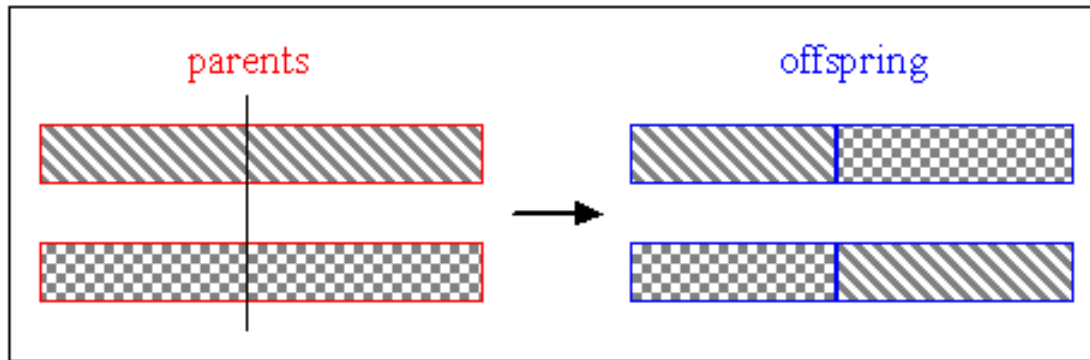


Selectie turneu

- Un numar *Tour* de indivizi din populatie este selectat aleator si cel mai bun individ dintre acestia este selectat ca parinte
- Procesul se repeta pt cati indivizi dorim sa selectam
- Parametrul pt dimensiunea turneului este *Tour*.
- *Tour* - valori intre 2 .. *Nind*
- Relatie intre *Tour* si intensitatea de selectie

Recombinare binara

- Pozitia de crossover selectata aleator → se produc 2 descendenti



Recombinare binara

- Exemplu

- individ 1: 0 1 1 1 0 | 0 1 1 0 1 0

- individ 2: 1 0 1 0 1 | 1 0 0 1 0 1

- pozitie crossover = 5

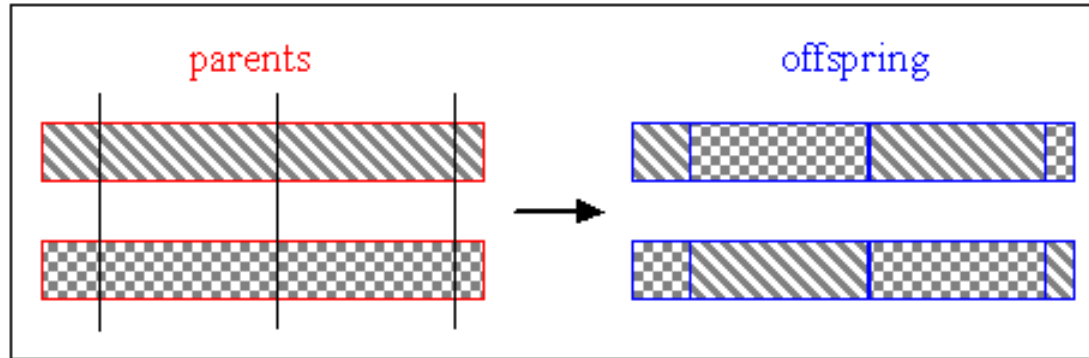
- Se creaza 2 indivizi noi:

- offspring 1: 0 1 1 1 0 | 1 0 0 1 0 1

- offspring 2: 1 0 1 0 1 | 0 1 1 0 1 0

Recombinaire multi-punct

- m pozitii de crossover k_i



- *Individ 1:* 0 1 1 1 0 0 1 1 0 1 0
- *Individ. 2:* 1 0 1 0 1 1 0 0 1 0 1
- pos. cross ($m=3$) 2 6 10
- offspring 1: 0 1 | 1 0 1 1 | 0 1 1 1 | 1
- offspring 2: 1 0 | 1 1 0 0 | 0 0 1 0 | 0

Mutatie

- Dupa recombinare – mutatia descendentilor
- Valori din descendenti sunt mutati
 - prin inversiune (binar)
 - adaugarea unor valori mici aleatoare, cu probabilitati mici

Mutatie binara

- Schimb valorile binar
- Pt fiecare individ, bitul de mutat este ales aleator
- 11 valori, bit 4

inainte de mutatie	0	1	1	1	0	0	1	1	0	1	0
dupa mutatie	0	1	1	0	0	0	1	1	0	1	0

Implementare paralela: Migrare

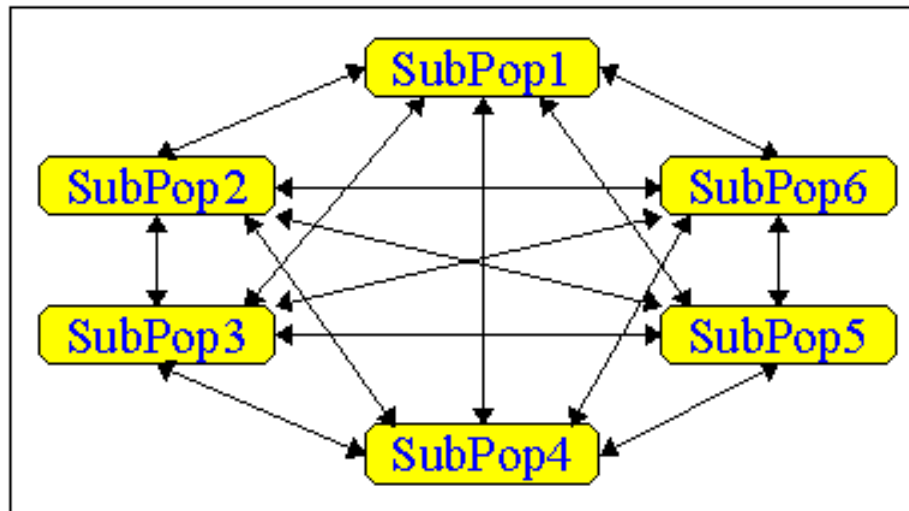
- Modelul migrator imparte populatia in **subpopulatii**.
- Aceste populatii evolueaza independent un numar de generatii (timp de izolare)
- Dupa timpul de izolare, un numar de indivizi este distribuit intre subpopulatii = **migrare**.
- **Diversitatea genetica** a subpopulatiilor si schimbul de informatii intre subpopulatii depinde de:
 - numarul de indivizi schimbati = **rata migrare**
 - metoda de **selectie** a indivizilor pentru migrare
 - **schema** de migrare

Migrare

- **Selectia** indivizilor pentru migrare:
 - aleator
 - bazata pe fitness (selectez pentru migrare cei mai buni indivizi).
- Schema de migrare intre subpopulatii:
 - intre toate subpopulatiile (topologie completa)
 - topologie inel
 - topologie vecinatate

Migrare

Topologie completa



Migrare

Schema de migrare intre subpopulatii

