



# **Invatare automata**

**Universitatea Politehnica Bucuresti**  
**Anul universitar 2017-2018**

**Adina Magda Florea**

# Curs Nr. 7

---

## Algoritmi genetici

- **Introducere**
- **Schema de baza**
- **Modelare probleme**
- **Exemplu**
- **Selectie**
- **Recombinare**
- **Mutatie**
- **TSP cu algoritmi genetici**
- **Implementare paralela AG**

# 1. Introducere

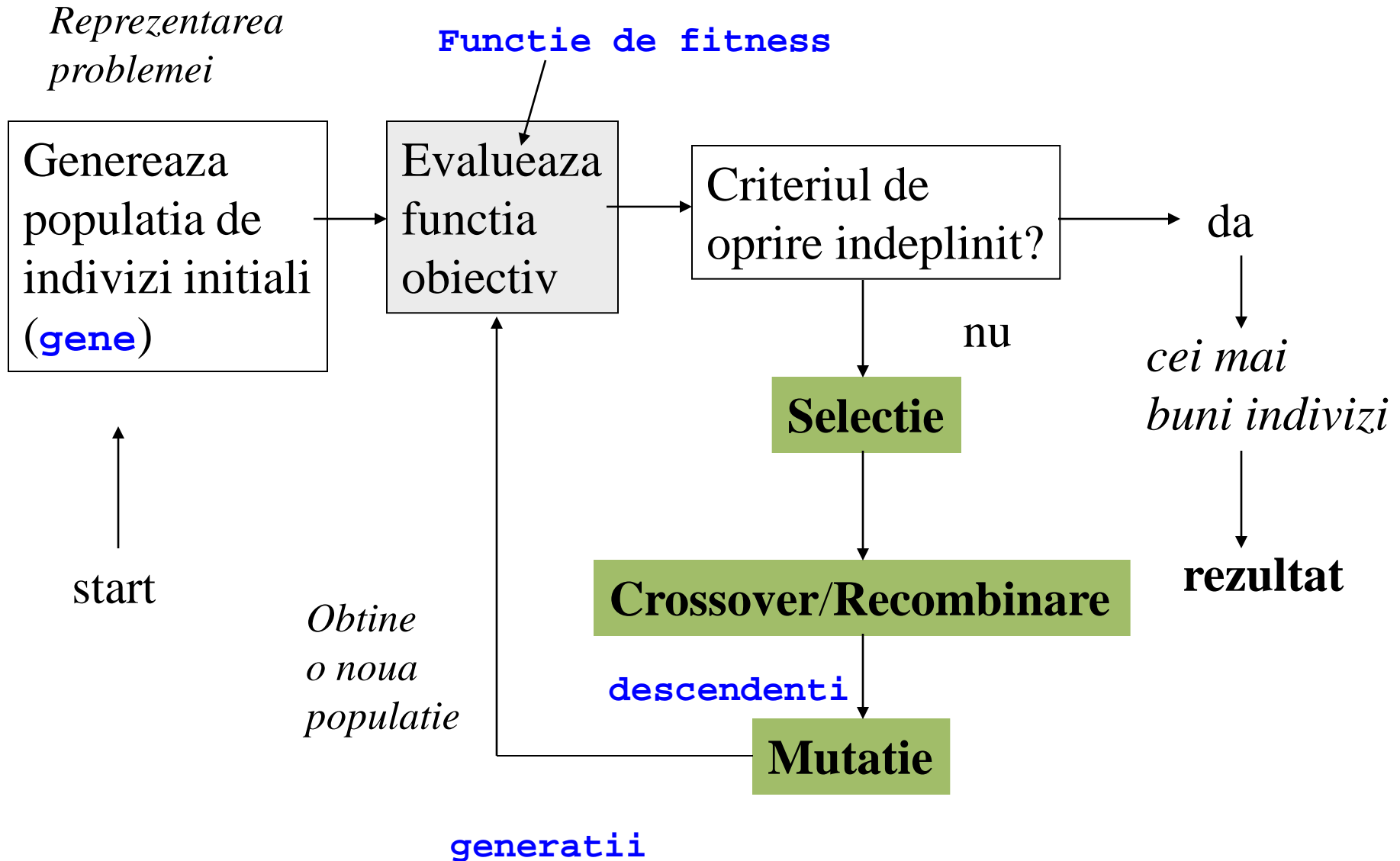
---

- GA - USA, J. H. Holland (1975)
- Algoritmi evolutivi – Germania, I. Rechenberg, H.-P. Schwefel (1975-1980)
- Programare genetica (1960-1980, 2000) J. Koza
  
- **Optimizare**
- **Modele economice**
- **Modele ecologice**
- **Modele ale sistemelor sociale**
- **Invatare**

# Introducere - cont

- Opereaza asupra unei populatii de indivizi = solutii potentiale – aplica principiul supravietuirii pe baza de adaptare (fitness)
- 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

## 2. Schema de baza



# 3. 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 existenta 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
- buget
- platou pt cel mai bun fitness

## Multipopulation GAs

- Rezultate mai bune - **subpopulatii**
- Fiecare populatie evolueaza separat
- Indivizi sunt schimbati dupa un numar de generatii

## 3.1 Selectie

### *(1) Primul pas: atribuire fitness*

- atribuire proportionala
- atribuire bazata pe rang

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

- **roulette-wheel selection** (selectie ruleta)
- **stochastic universal sampling** (esantionare universala stohastica)
- **local selection** (selectie locala)
- **tournament selection** (selectie turneu)
- **proportional selection** (selectie proportionala)



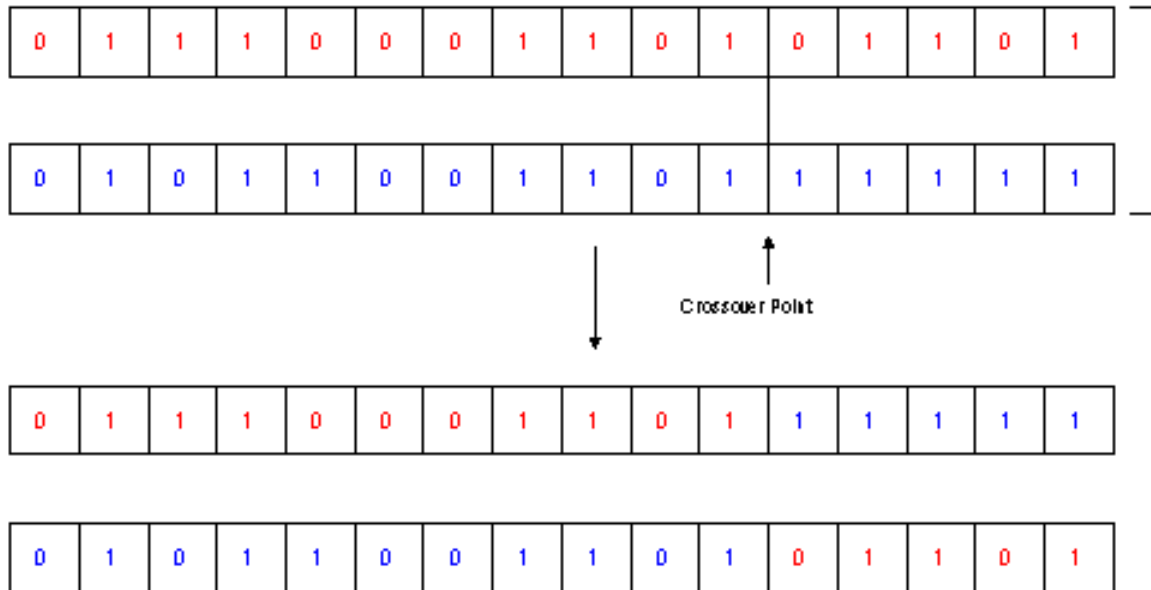
## 3.2 Reinserare

- Offspring – daca se produc mai putini indivizi sau mai putini copii atunci indivizi suplimentari trebuie reinserati in noua populatie
- Algoritmul de selectie determina schema de reinserare (in general)
- **reinserare globala** – in toata populatia pt. roulette-wheel selection, stochastic universal sampling, truncation selection
- **reinserare locala** pt selectie locala

### 3.3 Crossover/Recombination

- Recombinarea produce noi indivizi prin combinarea informatiei din parinti (parents - mating population).
- Diverse scheme de recombinare
- *O posibilitate* – imperechere aleatoare
- La fel cu *Crossing Over* din genetica
  1. Un procent  $P_M$  din indivizii noii populati sunt selectati si se imperecheaza aleator
  2. Un *crossover point* este selectat pentru fiecare pereche (acelasi sau diferit cu probabilitate)
  3. Informatia este schimbata intre cei doi indivizi pe baza pct de crossover

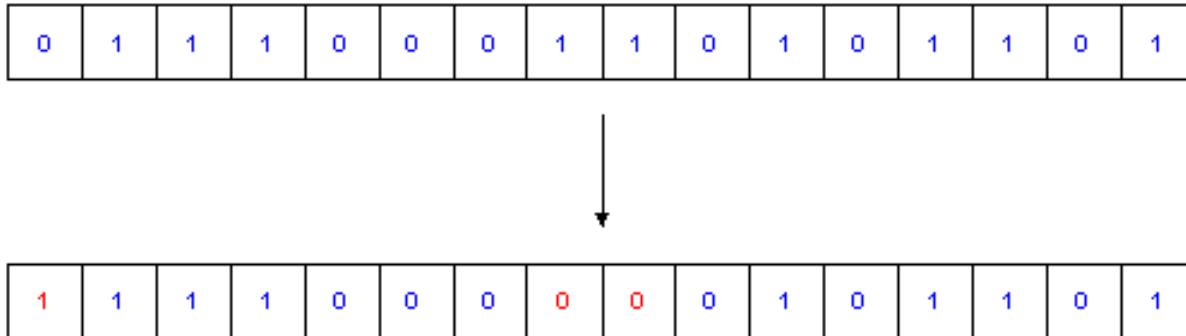
# Crossover



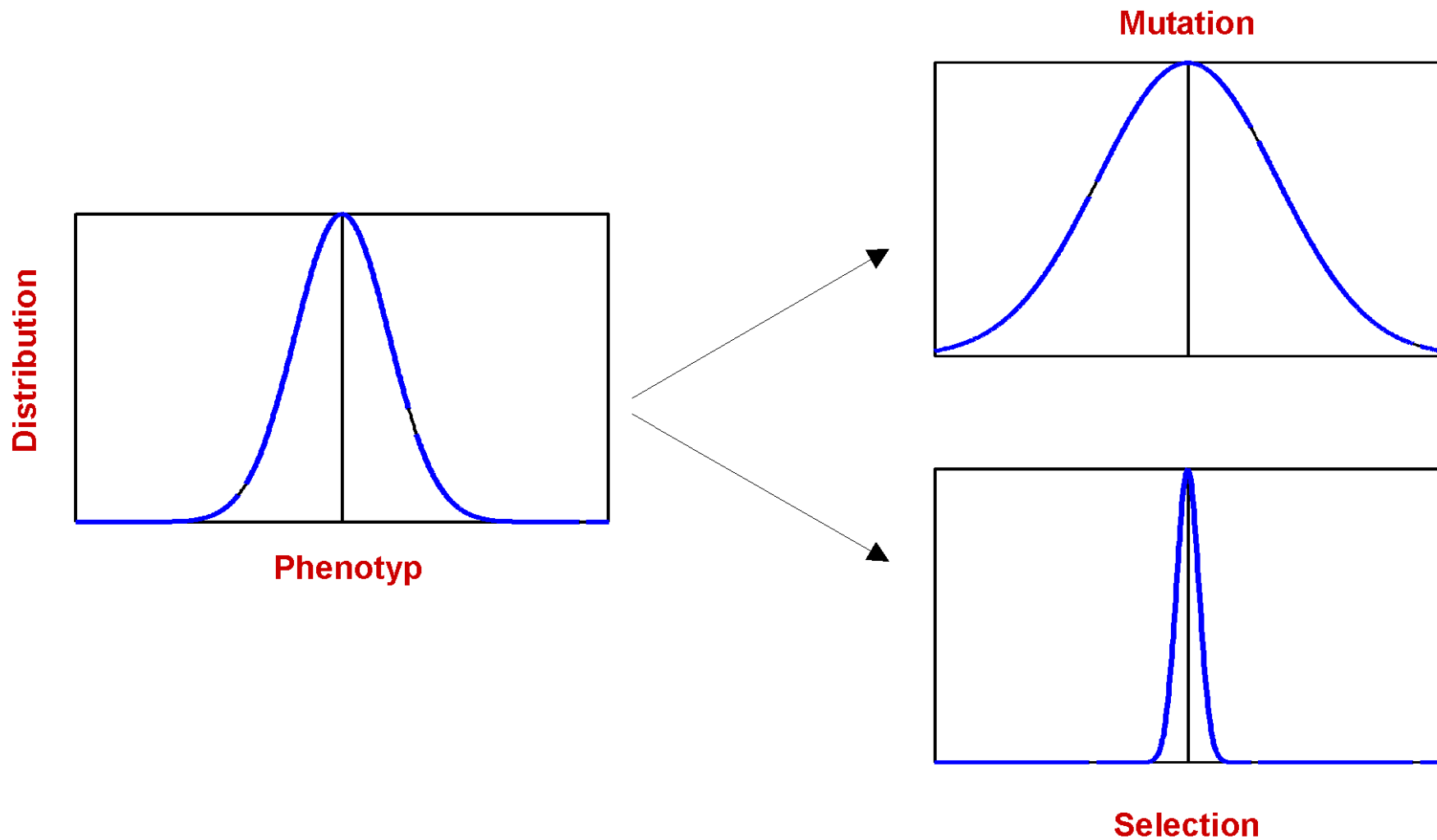
## 3.4 Mutatie

- Offspring - mutatie
- Mutatie cu perturbatii mici aleatoare
- Diverse forme de mutatie, depind de reprezentare
- **Mutatie** – *explorare* vs *exploatare*
- Schema simpla
  - *Fiecare bit are o probabilitate de mutatie*

# Mutatie



## Efectul mutatiei si a selectiei



## 4. Selectia in detaliu

---

- Primul pas este **atribuirea de fitness** (A)
  - Atribuire directa pe baza functiei obiectiv SAU
  - Atribuire pe baza unui mecanism (A1, A2)
- Fiecare individ din populatie primeste o valoare de fitness
- Pe baza valorii de fitness se realizeaza **selectia** (S) dupa o schema de selectie (S1, S2, S3, S4)

# Termeni

- **presiunea de selectie:** probabilitatea de selectie a celui mai bun individ comparata cu probabilitatea medie de selectie a tuturor indivizilor
- **raspandire:** domeniul de valori al numarului de descendenti al unui individ
- **pierderea diversitatii:** proportia de indivizi din populatie care nu este selectata
- **intensitatea de selectie:** valoare medie a fitness-ului populatiei dupa aplicarea unei metode de selectie
- **covarianta selectiei:** covarianta distributiei de fitness a populatiei dupa aplicarea unei metode de selectie



# A1. Atribuire fitness proportionala

- *Fiecare gena are un fitness asociat*
- Se calculeaza fitness mediu al populatiei
- Fiecare individ va fi copiat in noua populatie in fct de fitness comparat cu fitness mediu
- *fitness mediu 5.76, fitness individ 20.21 – se copiaza de 3 ori*
- Indivizii cu fitness egal sau sub medie se ignora
- Noua populatie – poate fi mai mica
- Noua populatie se completeaza cu indivizi selectati aleator din vechea populatie

## A2. 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

# Atribuire fitness bazata pe rang

- $Nind$  – numarul de indivizi din populatie
- $Pos$  – pozitia individului in populatie (cel mai prost  $Pos=1$ , cel mai bun  $Pos=Nind$ )
- $SP$  – presiunea de selectie (probabilitatea de selectie a celui mai bun individ relativ la probabilitatea medie de selectie a tuturor indivizilor)

## Rang liniar

$$\text{Fitness}(Pos) = 2 - SP + 2 * (SP - 1) * (Pos - 1) / (Nind - 1)$$

- Rangul liniar permite valori  $SP$  in  $(1.0, 2.0]$ .
- *Probabilitatea de selectie a unui individ pentru recombinare este fitness-ul (normalizat) raportat la fitness-ul total al populatiei*

# Atribuire fitness bazata pe rang

## Rang neliniar:

**Fitness(*Pos*) =**

$$Nind * X^{(Pos - 1)} / \sum_{i=1, Nind} (X^{(i - 1)})$$

- *X* se calculeaza ca radacina a polinomului:

$$0 = (SP - Nind) * X^{(Nind - 1)} + SP * X^{(Nind - 2)} + \dots + SP * X + SP$$

- Rang neliniar permite valori SP in

$$[1.0, Nind - 2.0]$$

- SP mai mari
- *Probabilitatea de selectie a unui individ pentru recombinare este fitness (normalizat) raportat la fitness total al populatiei*

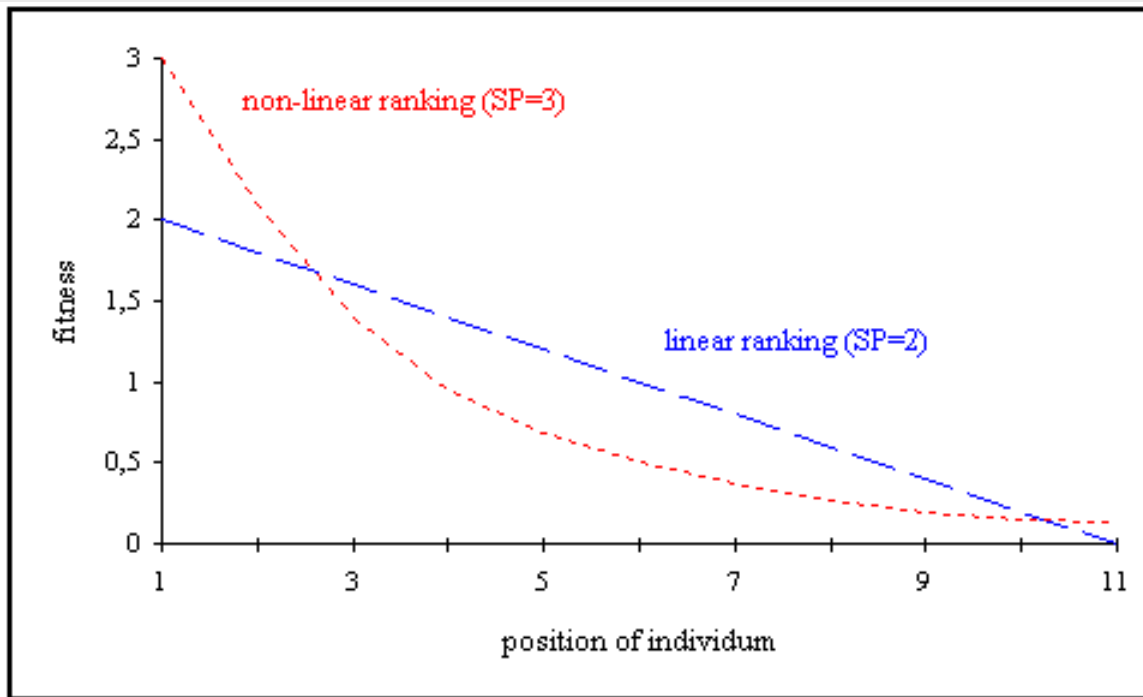
# Atribuire fitness bazata pe rang

Atribuire fitness rang liniar si rang neliniar

Liniar

$SP=2 \rightarrow 0..2$

$SP = 1.5 \rightarrow 0.5..1.5$



# Atribuire fitness bazata pe rang

- Fata de atribuirea proportionala:
  - Evita problema stagnarii in cazul in care presiunea de selectie este prea mica sau convergenta prematura genereaza o zona de cautare prea ingusta
  - Oferă un mod simplu de a controla presiunea de selectie
  - In general mai robusta

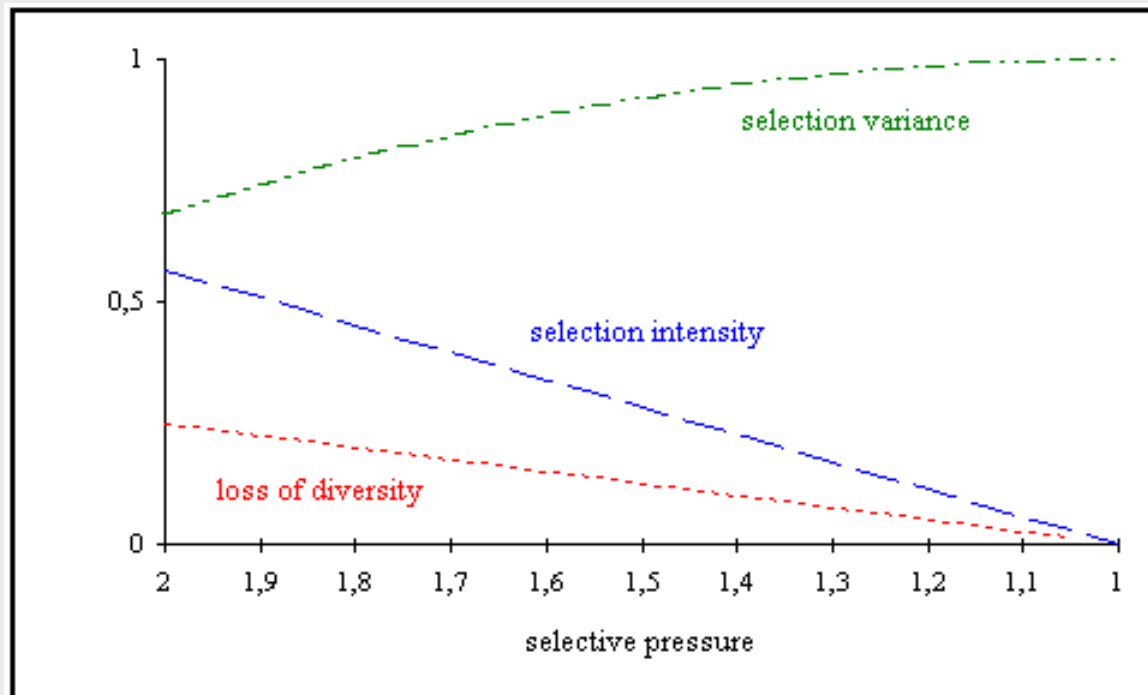
# Atribuire fitness bazata pe rang

## Proprietati

**Intensitatea de selectie:**  $\text{SelInt}_{\text{Rank}}(\text{SP}) = (\text{SP}-1) * (1/\sqrt{\pi})$ .

**Pierderea diversitatii:**  $\text{LossDiv}_{\text{Rank}}(\text{SP}) = (\text{SP}-1)/4$ .

**Covarianta selectiei:**  $\text{SelVar}_{\text{Rank}}(\text{SP}) = 1 - ((\text{SP}-1)^2 / \pi) = 1 - \text{SelInt}_{\text{Rank}}(\text{SP})^2$ .



# S1. Selectia bazata pe ruleta

"Roulette-wheel selection" sau "stochastic sampling with replacement"

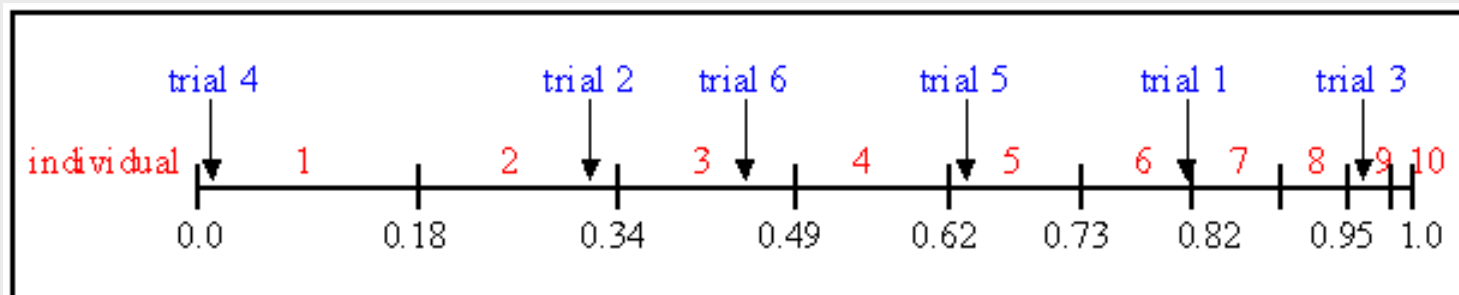
11 indivizi, rang linear si  $SP = 2$

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.

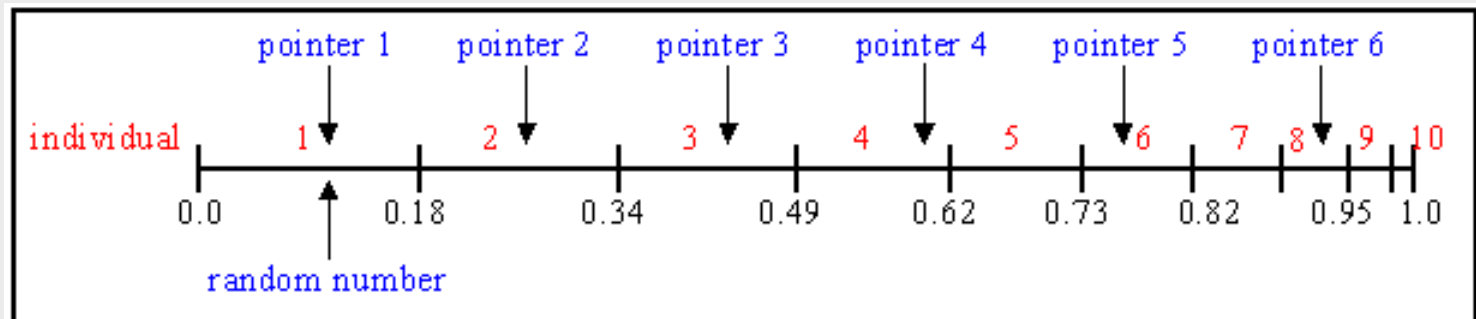
6, 2, 9, 1, 5, 3





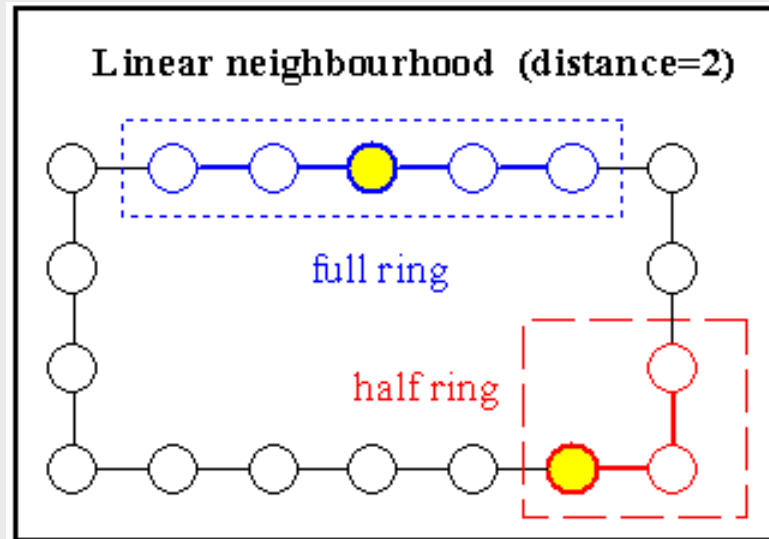
## S2. Esantionare universală stohastica

- Se plasează pointeri la distanțe egale pe un interval  $[0..1]$  – atâția pointeri câți indivizi se vor selecta
  - $NPointer$  – numărul de indivizi care va fi selectat
  - Distanța între pointeri  $1/Npointer$
  - Poziția primului pointer este dată de un număr aleator în intervalul  $[0..1/NPointer]$ .
  - Pentru 6 indivizi de selectat, distanța între pointeri este  $1/6=0.167$ .
  - **1 număr aleator în intervalul  $[0, 0.167]$ : 0.1.**
- 1, 2, 3, 4, 6, 8



## S3. Selectie locala

- Fiecare individ este intr-o vecinatate
- Structurarea populatiei
- Vecintatea – grup de indivizi care se pot recombina (potential)
- Vecintatea liniara: full si half ring

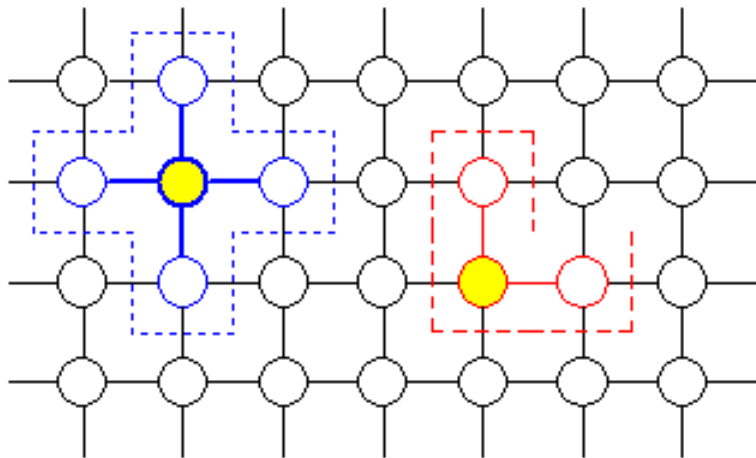


# Selectie locala

Twodimensional neighbourhood (distance=1)

full cross

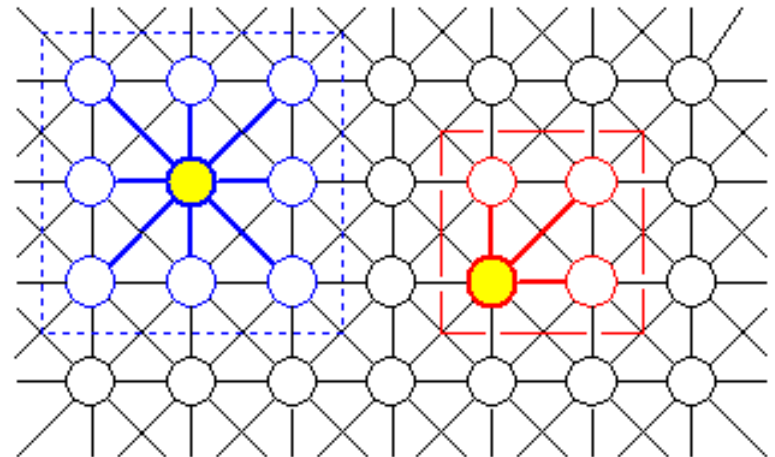
half cross



Twodimensional neighbourhood (distance=1)

full star

half star



## Selectie locala

- Se selecteaza prima jumatate a populatiei aleator (sau utilizand un algoritm de selectie – esantionare stohastica sau turneu).
- Se defineste apoi o vecintate pentru fiecare individ selectat.
- Se selecteaza un alt individ pt recombinare din vecintate (best, fitness proportional, sau aleator).

# Selectie locala

- Distanța de izolare între indivizi
- Cu cât mai mică vecintatea, cu atât mai mare izolarea
- Vecinătăți care se suprapun – apare transmisie de caracteristici
- Dimensiunea vecinătății determină viteza de propagare
- Propagare rapidă vs menținere diversitate mare
- Diversitate mare – previne convergența prematură

## S4. 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

Dimensiune turne	1	2	3	5	10	30
Intensitate selectie	0	0.56	0.85	1.15	1.53	2.04

*Valoare medie a fitness-ului populatiei  
dupa aplicarea unei metode de selectie*

# Selectie turneu

- **Intensitatea de selectie:**

$$\text{SelInt}_{\text{Tour}}(\text{Tour}) = \sqrt{2 * (\log(\text{Tour}) - \log(\sqrt{4.14 * \log(\text{Tour})}))}$$

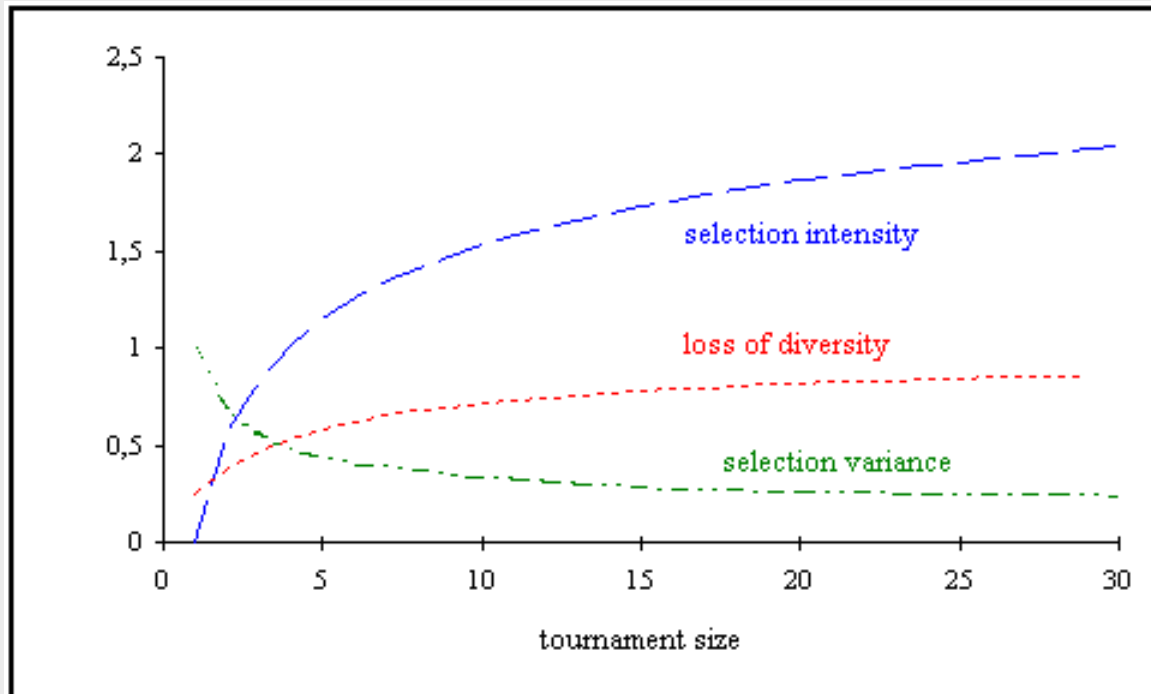
- **Pierderea diversitatii:**

$$\text{LossDiv}_{\text{Tour}}(\text{Tour}) = \text{Tour}^{-(1/(\text{Tour}-1))} - \text{Tour}^{-(\text{Tour}/(\text{Tour}-1))}$$

(Aprox 50% din populatie se pierde pt Tour=5).

- **Covarianta selectiei:**

$$\text{SelVar}_{\text{Tour}}(\text{Tour}) = 1 - 0.096 * \log(1 + 7.11 * (\text{Tour} - 1)), \text{SelVar}_{\text{Tour}}(2) = 1 - 1/\pi$$



## 6. Crossover / recombination

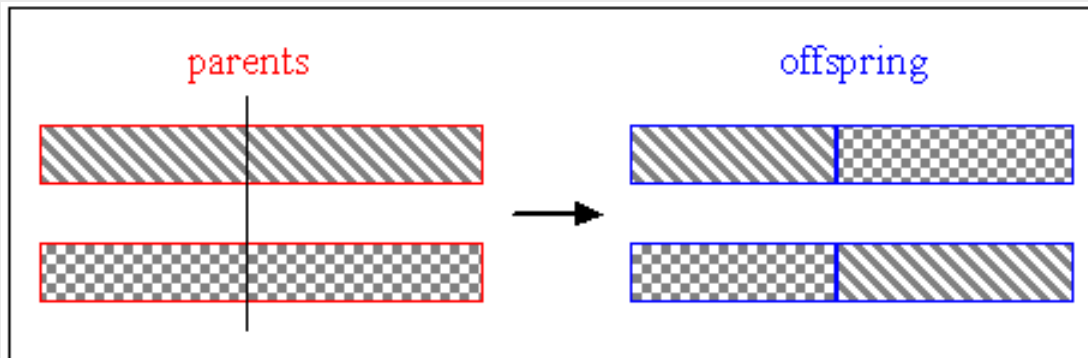
---

- Produce noi indivizi prin recombinaarea informatei din parinti
- Reprezentari binare
  - binara
  - multipunct
  - uniforma
- Reprezentari intregi/reale
  - discreta
  - reala intermediara
  - liniara



# 6.1 Recombinare binara

- Pozitia de crossover selectata aleator → se produc 2 descendenti



# Recombinare binara

- Exemplu

- individual 1: 0 1 1 1 0 | 0 1 1 0 1 0
- individual 2: 1 0 1 0 1 | 1 0 0 1 0 1

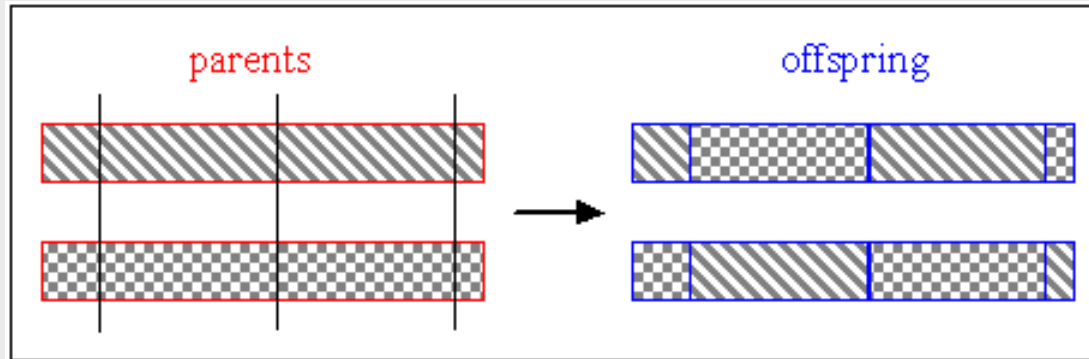
- pozitie crossover = 5

- 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

## 6.2 Recombinare multi-punct

- $m$  pozitii de crossover  $k_i$



- *individual 1:*      0 1 1 1 0 0 1 1 0 1 0
- *individual 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

## 6.3 Recombinare uniforma

- Generalizeaza binara simpla si multipunct
- **Crossover mask** – aceeași dimensiune cu a individului;
- creata aleator si paritatea bitilor din masca indica ce parinte va oferi descendentilor care bit
- *individual 1:*           0 1 1 1 0 0 1 1 0 1 0
- *individual 2:*           1 0 1 0 1 1 0 0 1 0 1
- mask 1: 0 1 1 0 0 0 1 1 0 1 0
- mask 2: 1 0 0 1 1 1 0 0 1 0 1           (inversa a mask 1)
- *offspring 1:*           1 1 1 0 1 1 1 1 1 1 1
- *offspring 2:*                   0 0 1 1 0 0 0 0 0 0 0
- Spears and De Jong (1991) – parametrizare prin asocierea unei probabilitati

## 6.4 Recombinare reala discreta

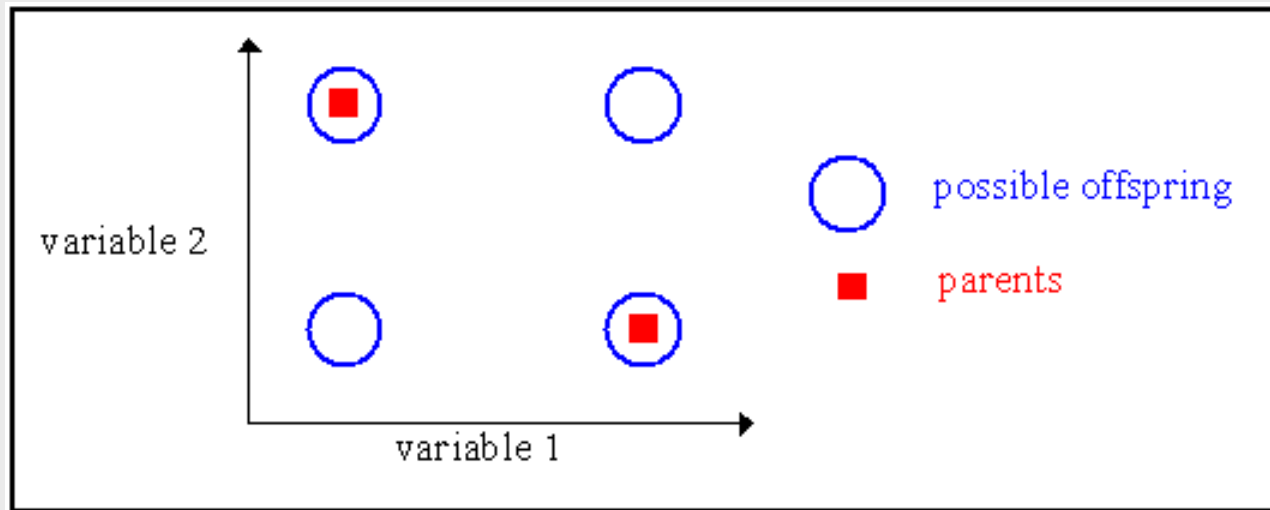
### Recombinare discreta

- Schimb de valori reale intre indivizi.
- individual 1:    12   25   5
- individual 2:    123   4   34
- Pt fiecare valoare, parintele care contribuie este ales aleator cu probabilitati egale
- sample 1:            2    2    1
- sample 2:            1    2    1
- Dupa recombinare:
- offspring 1:        123   4   5
- offspring 2:        12   4   5

# Recombinare reala discreta

## Recombinare discreta

- Pozitiile posibile ale descendentilor



- Poate fi utilizata cu orice valori (binare, reale or simboluri).

# Recombinare reala intermediara

## Recombinare reala intermediara

- Numai pt valori reale
- Valorile din descendenti alese in jurul valorilor din parinti
- Regula:  
$$\text{offspring} = \text{parent 1} + \text{Alpha} (\text{parent 2} - \text{parent 1})$$

unde Alpha este un factor de scalare ales aleator in intervalul  $[-d, 1 + d]$ .
- $d = 0$  sau  $d > 0$ . O valoare buna  $d = 0.25$ .
- Fiecare valoare din descendenti este rezultatul combinarii parintilor cu o noua Alpha pt fiecare variabila

# Recombinare reala intermediara

## Recombinare reala intermediara

- individual 1: 12 25 5
- individual 2: 123 4 34

Valorile Alpha sunt:

- sample 1: 0.5 1.1 -0.1
- sample 2: 0.1 0.8 0.5
- Indivizi noi

(offspring = parent 1 + Alpha (parent 2 - parent 1))

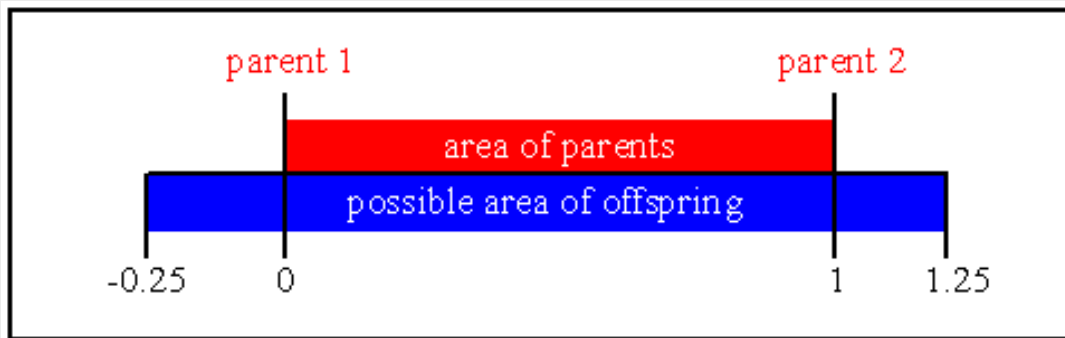
- offspring 1: 67.5 1.9 2.1
- offspring 2: 23.1 8.2 19.5



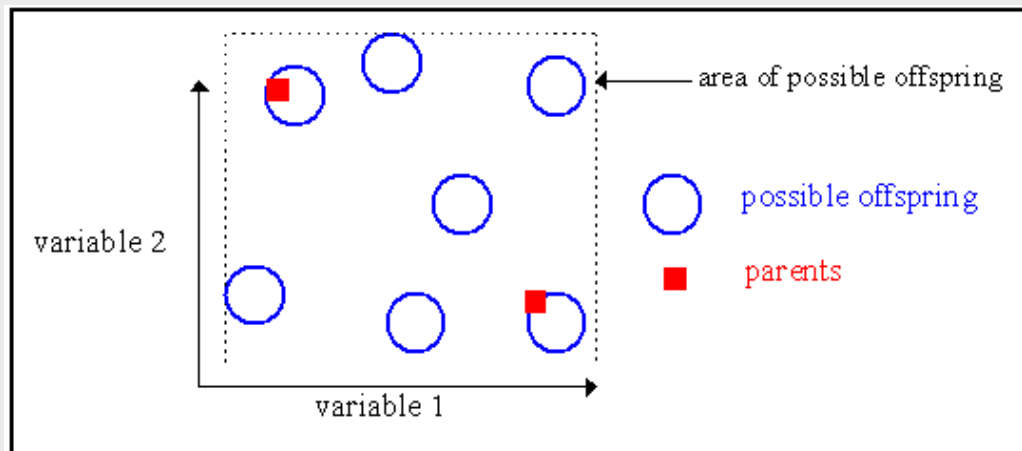
# Recombinare reala intermediara

## Recombinare reala intermediara

- Domeniul de valori ale descendentilor fata de cel al parintilor



- Repartizare descendenti dupa recombinare intermediara



# Recombinare reala liniara

## Recombinare liniara

- Similara cu cea intermediara dar se foloseste un singur Alpha.

- individual 1:           12   25   5

- individual 2:           123   4   34

Valorile Alpha sunt:

- sample 1: 0.5

- sample 2: 0.1

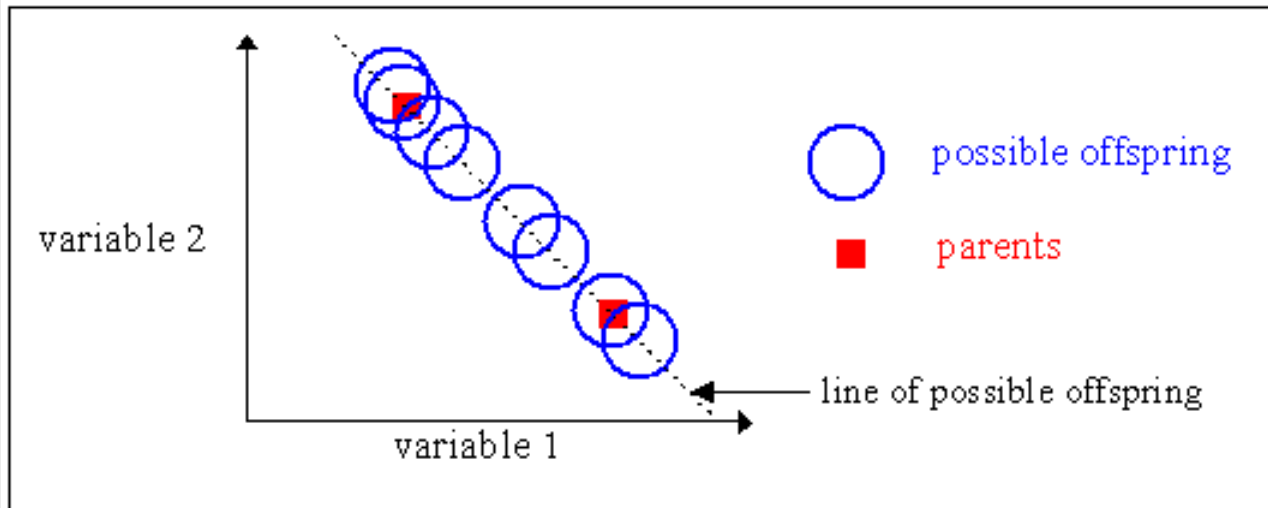
Indivizii noi:

- offspring 1:           67.5   14.5   19.5

- offspring 2:           23.1   22.9   7.9

# Recombinare reala liniara

## Recombinare liniara



# 7. Mutatie

---

- Dupa recombinare – mutatia descendentilor
- Valori din descendenti sunt mutati prin inversiune (binar) sau adaugarea unor valori mici aleatoare (pasul de mutatie), cu probabilitati mici
- Probabilitatea de mutatie este invers proportionala cu dimensiunea indivizilor
- Cu cat avem indivizi mai lungi cu atat este mai mica probabilitatea de mutatie

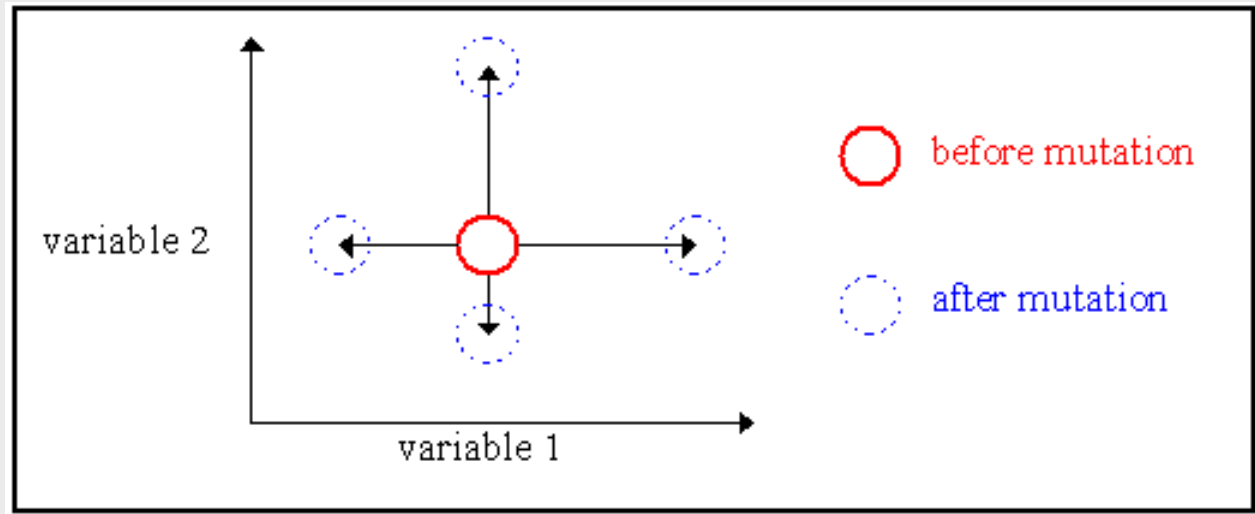
## 7.1 Mutatie binara

- 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

## 7.2 Mutatie cu valori reale

- **Efect**
- **mutatie**



- Dimensiune pas – dificil; poate varia in timpul evolutiei
- Mici – bine, lent; mari – mai repede
- Operator mutatie :

$\text{mutated variable} = \text{variable} \pm \text{range} \cdot \text{delta}$  (+ sau – cu probabilitate egala)

$\text{range} = 0.5 \cdot \text{domeniu variabila}$

$\text{delta} = \sum(a(i) 2^{-i})$ ,  $a(i) = 1$  cu probabilitate  $1/m$ , altfel  $a(i) = 0$ .

## 8. Utilizarea GA pt:

---

- **Problema 0/1 Knapsack**
  - **TSP**

## 8.1 0/1 Knapsack Problem

- Se da o multime de obiecte, fiecare cu o greutate/pondere  $w(i)$  si valoare/profit  $p(i)$ .
- Sa se determine numarul de obiecte din fiecare tip care sa se includa intr-o colectie a.i. greutatea sa fie mai mica decat o valoare data  $W$  si valoarea totala sa fie maxima.
- Problema 0/1 knapsack - 0 sau 1 obiecte din fiecare tip.
- *Multiobjective optimization problem*: maximizeaza profit si minimizeaza greutate
- Nu exista o (singura) solutie optima ci un set de solutii cu "trade-off" optim = multimea de solutii pt care nu se poate imbunatati un criteriu fara a se inrautati altul



# 0/1 Knapsack Problem

- Maximizeaza  $\text{sum}(\mathbf{x(i)} * \mathbf{p(i)})$
- Restrictie  $\text{sum}(\mathbf{x(i)} * \mathbf{w(i)}) \leq \mathbf{W}$
- $\mathbf{x(i)} = 0$  or  $1$

Sir binar - lungime = numarul de obiecte.

Fiecare obiect are asociata o pozitie in sirul binar

0 – obiectul nu este in solutie

1 – obiectul este in solutie

Operatori genetici:

- selectie turneu
- one-point crossover
- bit-flip mutation.

Dimeniune populatie 100

Tour 2

Probabilitate mutatie: 0.01

**greutate (w)** 3 3 3 3 3 4 4 4 7 7 8 8 9

**valoare (p)** 4 4 4 **4** 4 5 5 5 **10 10** 11 11 13

Generation 74:	08 23	0110010010000
No Fit Cromozom	09 23	1010101011100
00 24 000 <b>1</b> 0000 <b>11</b> 000	10 22	0000010011000
01 23 01101000000100	11 22	1010101100000
02 23 0010100101000	12 22	0110101100000
03 23 0110010001000	13 22	1010101100000
04 23 0110000110000	14 22	1010101100000
05 23 0101010001000	15 22	1010101011100
06 23 10101000000100	16 15	0000010001000
07 23 0110010001000	17 12	0110010011000
	18 10	0010100101010
	19 -18	0110011110011

## 8.2 TSP – Reprezentare problema

### Funcție de evaluare

- Funcția de evaluare pentru  $N$  orase este suma distanței euclidiene

$$Fitness = \sum_{i=1}^N \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$$

### Reprezentare

- individ = reprezentare a căii, în ordinea de parcurgere a orașelor

3	0	1	4	2	5
0	5	1	4	2	3
5	1	0	3	4	2

# TSP Operatori genetici

## Crossover

- Nu se potrivesc operatorii traditionali la TSPs

- Inainte de crossover

1	2	3		4	5	0	(parent 1)
2	0	5		3	1	4	(parent 2)

- Dupa crossover

1	2	3		3	1	4	(child 1)
2	0	5		4	5	0	(child 2)

- Greedy Crossover - Grefenstette in 1985

# TSP Operatori genetici - recombinare

Parinti **1 2 3 4 5 0** si **4 1 3 2 0 5**

- Se genereaza un descendent utilizand cel de al doilea parinte ca sablon: selectez orasul **4** ca primul oras al copilului **4 x x x x x**
- Gasesc legaturi de la oras 4 in ambii parinti: (4, 5) si (4, 1). Daca distanta (4,1) mai mica decat (4,5), selectez **1** ca urmatorul oras din copil: **4 1 x x x x**
- Gasesc legaturi de la oras 1 in ambii parinti: (1, 2) si (1, 3).
- $(1,2) < (1,3)$  – selectez **2** ca oras urmator: **4 1 2 x x x**
- $(2, 3) > (2, 0)$  – selectez **0**: **4 1 2 0 x x**
- $(0, 1) < (0, 5)$ . Deoarece 1 apare deja in copil, selectez **5** - **4 1 2 0 5 x**
- Legaturi 5 sunt (5, 0) si (5, 4), dar atat 0 cat si 4 apar in copil. Alegem un oras neselectat 3 - copil **4 1 2 0 5 3**

Acceasi metoda pt a genera celalat descendent **1 2 0 5 4 3**

# TSP Operatori genetici

## Mutatie

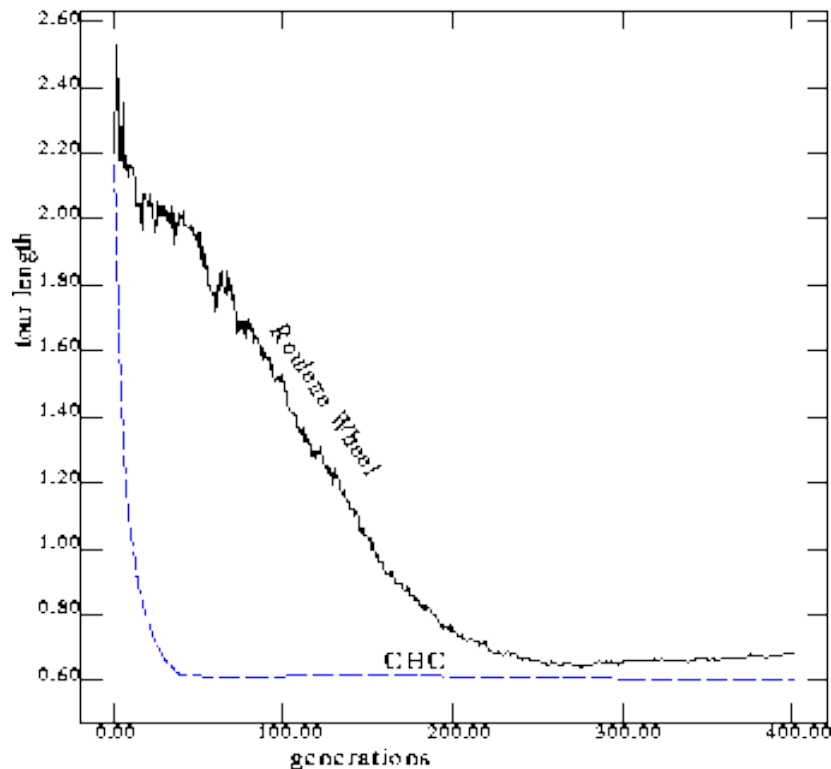
- Nu putem folosi mutatia clasica. De ex: **1 2 3 4 5 0**, mutam 3, schimbam aleator 3 la 5 - **1 2 5 4 5 0** - gresit
- Selectam aleator 2 valori si le interschimbam.
- **Swap mutation:**            **1 2 3 4 5 0**    **1 5 3 4 2 0**

## Selectie

- Roulette wheel selection – cel mai bun individ are probabilitatea cea mai mare de selectie dar nu este sigur selectat
- Utilizam **selectia CHC** pt a garanta ca cel mai bun individ supravietuieste (Eshelman 1991).

# TSP Comportare

- CHC selection – populatie de dimensiune  $N$
- Genereaza  $N$  copii cu roulette wheel selection
- Combina  $N$  parinti cu  $N$  copii
- Ordoneaza  $2N$  indivizi in functie de fitness
- Alege cei mai buni  $N$  indivizi pt generatia urmatoare

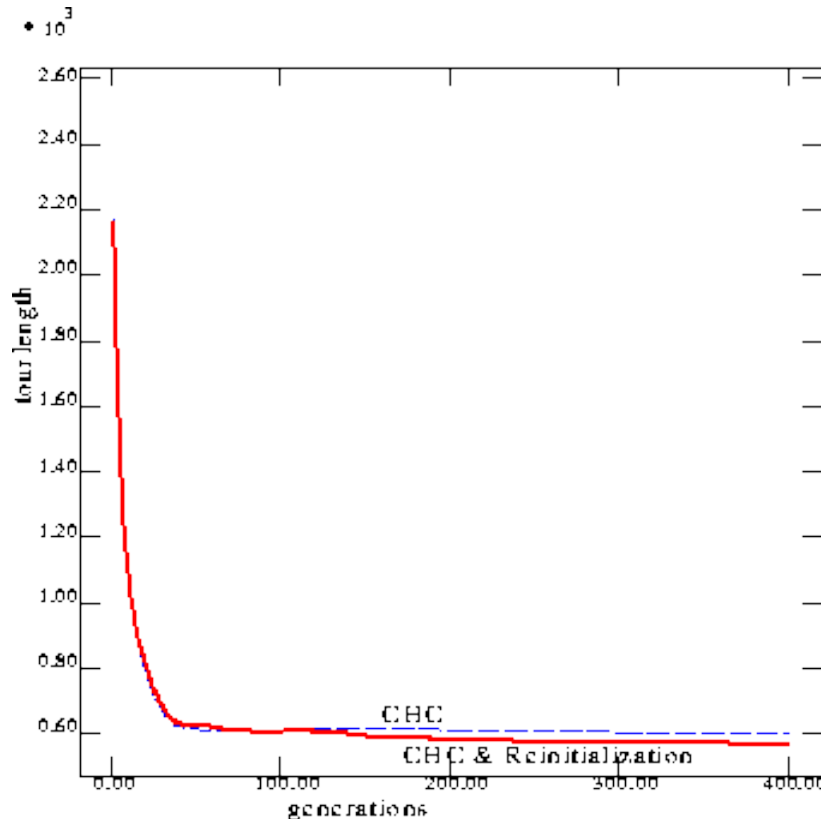


## Comparatie Roulette si CHC selection

Cu selectia CHC populatia converge mai repede decat cu roulette wheel selection si performantele sunt mai bune

# TSP Comportare

- Dar convergenta rapida = mai putina explorare
- Pt a preveni convergenta la un minim local, cand am obtinut convergenta populatiei, salvam cei mai buni individi si re-initializam restul populatiei aleator.
- Selectie CHC astfel modificata = selectie **R-CHC**.



**Comparatie selectie CHC  
cu si fara re-initializare**



## **9 Implementarea paralela a AG**

- **Modelul migrator**
- **Model global - worker/farmer**

## 9.1 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

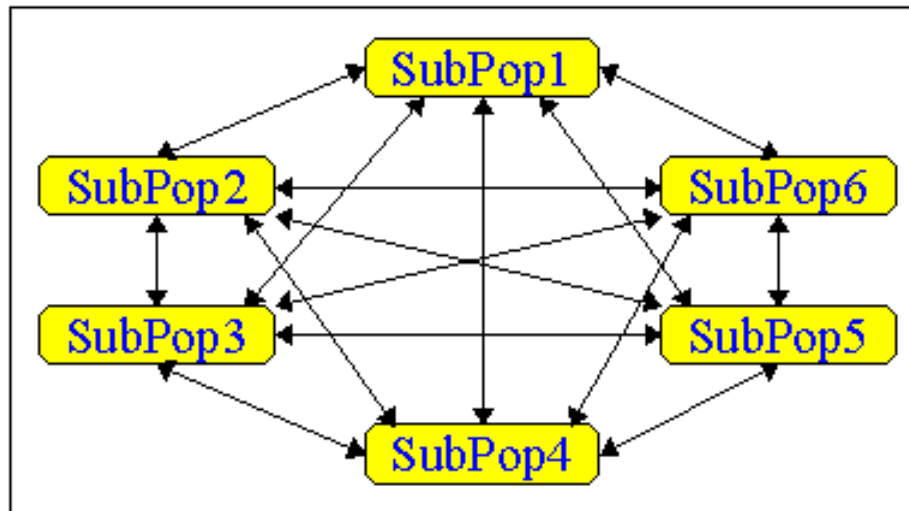
- Implementarea a modelului migrator:
  - scade timpul de prelucrare
  - necesita mai putine evaluari de functii obiectiv fata de un model cu o singura populatie
- Implementarea migrarii (paralel) pe un singur procesor (pseudo-paralel) este buna
- In anumite cazuri se obtin rezultate mai bune

# 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

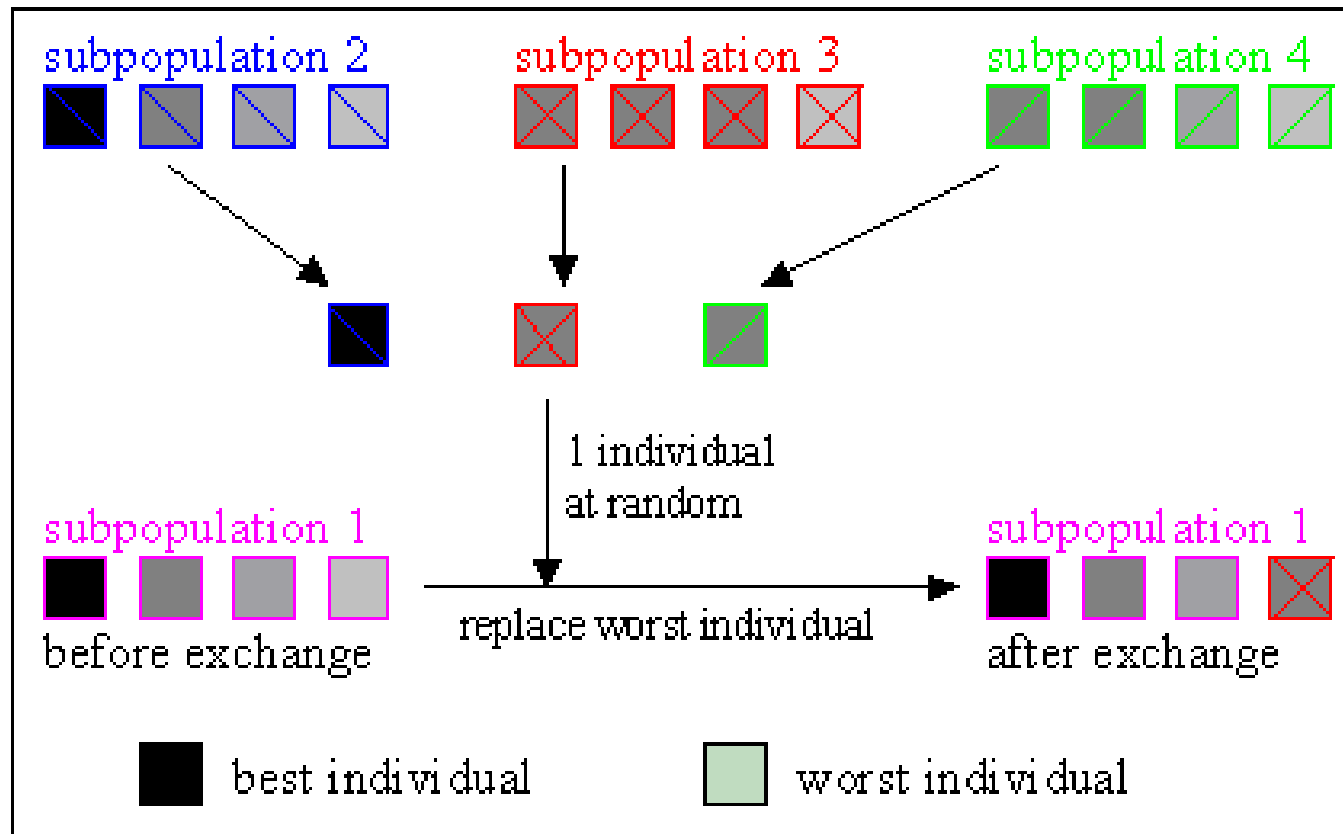
# Migrare

Topologie completa



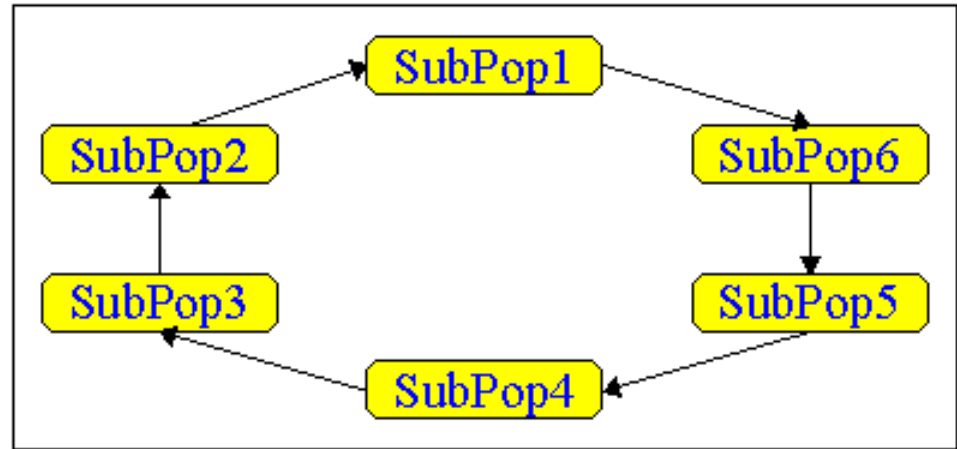
# Migrare

## Schema de migrare intre subpopulatii

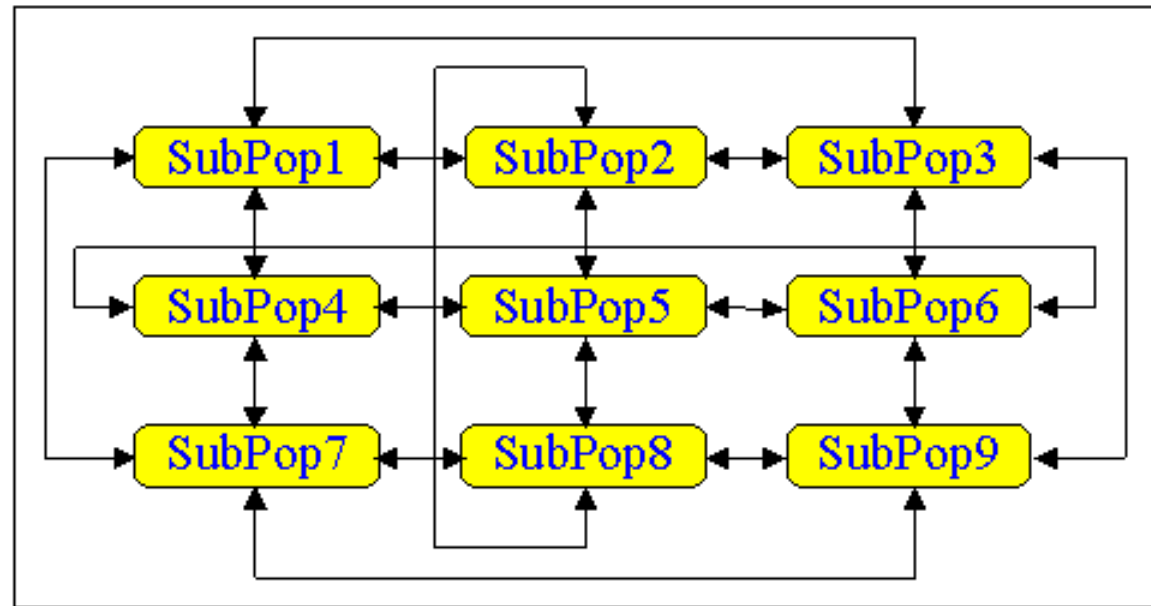


# Migrare

Topologie inel



Topologie  
vecintate  
(implemantare 2-  
D)



## 9.2 Modelul global - worker/farmer

- Exploateaza paralelismul inherent al GA
- Worker/Farmer – schema de migrare

