

# **Calcul Evolutiv**

## **Algoritmi genetici**

**Student: Dobrin Mihai**  
**Anul III - Informatica**

- **Ce inseamna Calcul Evolutiv?**
- **Ce inseamna Algoritmi Genetici?**
- **Etapele care trebuie parcurse in realizarea unui algoritm genetic**
- **Exemplu: Maximizarea unei functii**

- *Ce inseamna Calcul Evolutiv?*
- **Ce inseamna Algoritmi Genetici?**
- **Etapele care trebuie parcurse in realizarea unui algoritm genetic**
- **Exemplu: Maximizarea unei functii**

# *Ce inseamna Calcul Evolutiv?*

## **D. p d. v. al rezolvarii automate:**

- *Probleme “bine-puse”*: caracterizate prin faptul ca li se poate asocia un model formal (de exemplu, un model matematic) pe baza caruia se poate dezvolta o metoda de rezolvare cu caracter algoritmic.
- *Probleme “rau-puse”*: caracterizate prin faptul ca nu pot fi descrise complet printr-un model formal, ci cel mult se cunosc raspunsuri pentru cazuri particulare ale problemei

# *Ce inseamna Calcul Evolutiv?*

**D.p.d.v complexitatii rezolvarii si  
consistentei raspunsului:**

- *Probleme pentru care este esentiala obtinerea unui raspuns exact indiferent de resursele implicate. Acestea necesita utilizarea unor tehnici exacte.*
- *Probleme pentru care este preferabil sa se obtina un raspuns “aproximativ” folosind resurse “rezonabile”, decat un raspuns exact dar folosind resurse foarte costisitoare.*

# *Ce inseamna Calcul Evolutiv?*

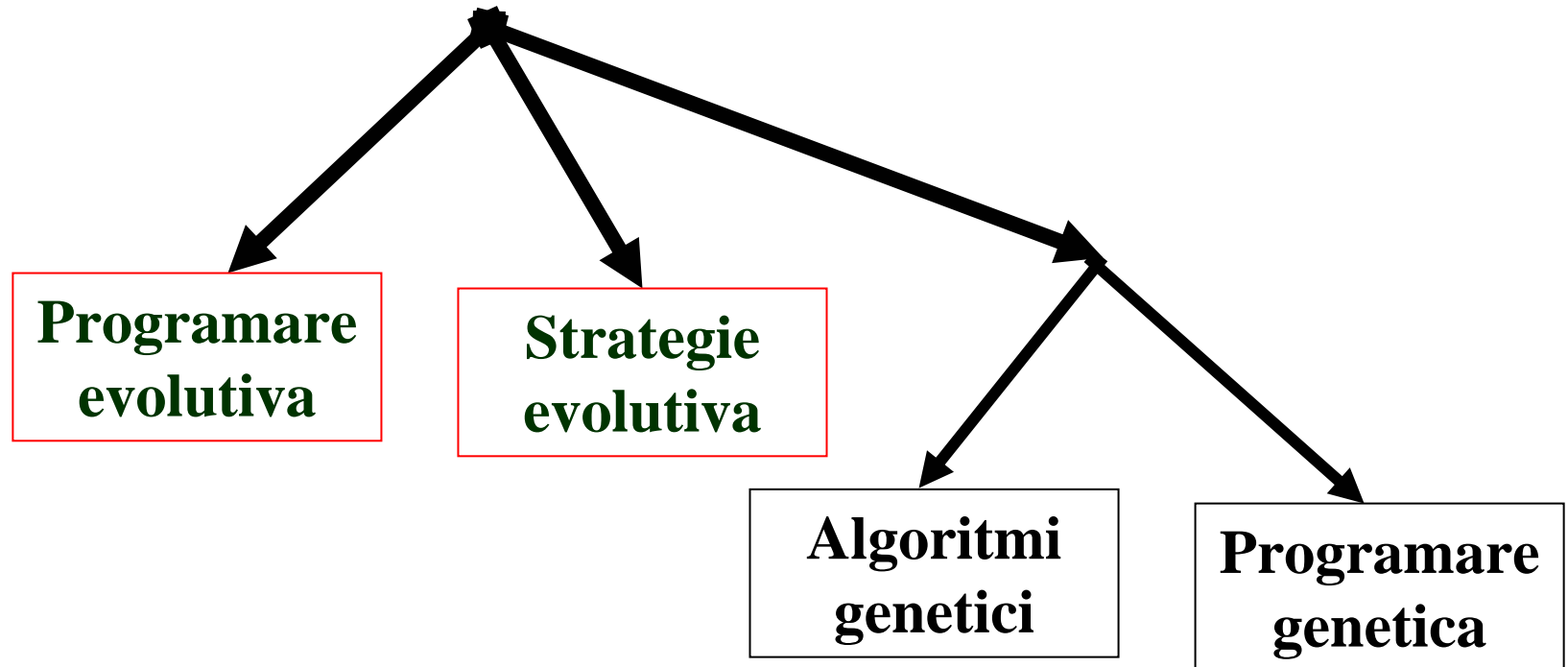
- **Calculul evolutiv - este folosit in principal in rezolvarea problemelor bazate pe cautarea solutiei intr-un spatiu mare de solutii potentiale. Sursa de inspiratie o reprezinta principiile evolutionismului darwinist.**
- **Principiul fundamental este de a dezvolta sisteme complexe pornind de la implementarea unor reguli simple.**

# *Ce inseamna Calcul Evolutiv?*

- *Pentru gasirea solutiei se utilizeaza o populatie de “agenti” de rezolvare a problemei.*
- *Populatia este supusa unui proces de evolutie caracterizat prin: selectie, incrucisare, mutatie.*

# *Ce inseamna Calcul Evolutiv?*

**In functie de cum este construita populatia si cum este implementata evolutia calculul evolutiv este asociat cu:**





- **Ce inseamna Calcul Evolutiv?**
- *Ce inseamna Algoritmi Genetici?*
- **Etapele care trebuie parcurse in realizarea unui algoritm genetic**
- **Exemplu: Maximizarea unei functii**

# *Ce inseamna Algoritmi Genetici?*

- **Populatia este reprezentata de stari din spatiul problemei care reprezinta solutii potentiale.**
- **De regula elementele populatiei sunt codificate in forma binara.**
- **Operatorul de incrucisare este cel principal, cel de mutatie are un rol secundar.**
- **Sunt folositi in special in rezolvarea problemelor de optimizare combinatoriala**

# *Ce inseamna Algoritmi Genetici?*

```
graph TD; A["Ce inseamna Algoritmi Genetici?"] --> B["Algoritmi genetici standard"]; A --> C["Algoritmi genetici hibridi"];
```

## **Algoritmi genetici standard**

- codificare binara
- lungimea cromozomului fixa
- marimea populatie fixa

## **Algoritmi genetici hibridi**

- codificare reala
- lungimea cromozomului variabila
- marimea populatie variabila

- **Ce inseamna Calcul Evolutiv?**
- **Ce inseamna Algoritmi Genetici?**
- *Etapele care trebuie parcurse in realizarea unui algoritm genetic*
- **Exemplu: Maximizarea unei functii**

## *La inceput trebuie stabilit:*

- **Modul de configurare.** Se specifica modul in care fiecarei configuratii din spatiul de cautare i se asociaza un cromozom
- **Functia de adecvare.** Se construiește functia care exprima gradul de adecvare la mediu
- **Dimensiune si modul de initializare a populatiei.** Populatiile pot avea dimensiune fixa(standard) sau variabila(hibrid).
- **Mecanismul de selectie** a parintilor si supravietuitorilor.

## *La inceput trebuie stabilit:*

- **Mecanismul de incrucisare** a parintilor pentru a genera urmasi.
- **Mecanismul de mutatie** care asigura perturbarea elementelor.
- **Supravietuirea** determina in ce masura cromozomii unei generatii supravietuiesc in urmatoarea.
- **Criteriul de oprire.** Cand nu se cunoaste un criteriu se opteaza pentru numar maxim de iteratii. Se pot folosi informatii despre populatie (ex: gradul de diversitate).

# Modul de configurare

- **Structuri de date folosite.** In algoritmi genetici cromozomii sunt reprezentati prin structuri liniare cu numar fix (tablouri) sau cu numar variabil (liste inlantuite).
- **Reguli de codificare:** 1. Codificare binara (cromozomii sunt vectori cu valori  $\{0,1\}$ ) 2. Codificare reala (vectori cu elemente reale).
- **Decodificarea** este necesara pentru (1).  
Pregateste evaluarea configurarii

# Funcția de adecvare

- *In procesul de evoluție naturală se urmărește maximizarea gradului de adecvare a indivizilor la mediu.*
- *Pentru a ne folosi de analogia dintre procese de căutare și cele de evoluție este util să reformulăm problemele de optimizare ca probleme de maximizare.*



# Selectie

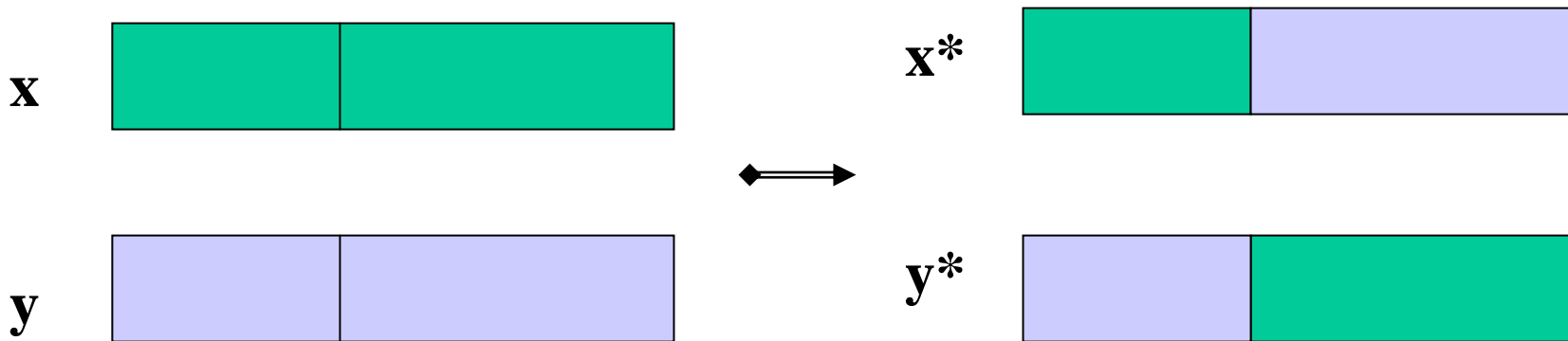
- **Are ca scop determinarea populatiei intermediare ce contine parintii care vor fi supusi operatorilor genetici de incrucisare si mutatie si determina elementele ce vor face parte din generatia urmatoare.**
- **Este legata de functia de adecvare.**
- **Nu este obligatoriu ca atat parintii cat si supravietuitorii sa fie determinati prin selectie, fiind posibil ca aceasta sa fie folosita doar intr-o singura etapa.**

# Incrucisarea

- **Permite combinarea informatiilor provenite de la doi sau mai multi parinti pentru generarea unuia sau mai multor urmasi.**
- **Incrucisarea in cazul algoritmilor standard parintele se inlocuieste cu copilul**
- **Incrucisarea in cazul algoritmilor hibrizi se lupta pentru supravietuire.**

# Incrucisarea cu un punct de taietura

- Avem  $x=\{x_1,\dots,x_n\}$  si  $y=\{y_1,\dots,y_n\}$
- Se alege aleator  $k$  din  $\{1,\dots,n-1\}$  numit punct de taietura
- $x^*=\{x_1,\dots,x_k,y_{k+1},\dots,y_n\}$   $y^*=\{y_1,\dots,y_k,x_{k+1},\dots,x_n\}$



# Mutatia

- **Asigura alterarea valorii unor gene pentru a evita situatiile in care o anumita gena nu apare in populatie fiindca nu a fost generata de la inceput. In acest fel se asigura diversitatea populatiei.**
- **In cazul codificarii binare cea mai simpla si frecventa metoda este alegerea aleatoare a unui cromozom in cadrul caruia se selecteaza o gena (daca este 0 devine 1, daca este 1 devine 0).**

# Supravietuirea

- **In general e formata din indivizi care nu au intrat in reproducere si populatia intermediara rezulta in urma reproducerii.**
- **Se pot selecta un numar din cei mai buni indivizi care sa apara in populatia urmatoare, apoi intra ei in reproducere iar restul de completat e formata din populatia rezultata in urma reproducerii**

- **Ce inseamna Calcul Evolutiv?**
- **Ce inseamna Algoritmi Genetici?**
- **Etapele care trebuie parcurse in realizarea unui algoritm genetic**
- *Exemplu: Maximizarea unei functii*

## *Exemplu: Maximizarea unei functii*

- **Dorim sa optimizam o functie simpla:  
 $f(x)=x*\sin(10\pi*x)+1.0$  unde  $x\in[-1,2]$**
- **Este usor sa descoperim maximul folosind prima derivata. Valoare maxima va fi  $f(1.85) = 2.85$**
- **Vom construi in continuare un algoritm genetic care sa maximize functia f.**

# *Exemplu: Maximizarea unei functii*

- **Reprezentare:** Vom folosi  $v$  un vector binar.
  1. Lungimea vectorului depinde de precizie, care in acest exemplu va fi 6 zecimale.
  2. Domeniul lui  $x$  are lungime 3([-1,2]).
- (1) + (2)  $\Rightarrow$  valoare lui  $v$  in baza 10 trebuie sa fie cel putin  $3*1000000 \Rightarrow$  avem nevoie de 22 de biti:

$$2097152=2^{21}<3000000\leq 2^{22}=4194304$$



# *Exemplu: Maximizarea unei functii*

- Vom transforma valoarea lui  $v=(v_{21}...v_0)$  intr-un numar real  $x$  din intervalul  $[-1,2]$  in doi pasi:
  1. Convertim vectorul binar  $(v_{21}...v_0)$  din baza 2 in baza 10 si il notam cu  $v_1$ .
  2.  $x$  corespunzator este:

$$x = -1.0 + v_1 \frac{3}{2^{22}-1}$$

# *Exemplu: Maximizarea unei functii*

- **Initializarea populatiei.** Este un proces foarte simplu vom crea o populatie de cromozomi unde fiecare cromozom este un vector binar alcatuit din 22 biti. Fiecare cromozom este initializat aleator.
- **Functia de adecvare.** In acest caz o vom nota cu  $eval(v)=f(x)$ . Cel mai bun cromozom va fi acela care are cea mai mare valoare.

# *Exemplu: Maximizarea unei functii*

- **Selectia**

1. Calculam  $\text{eval}(v_i)$  pentru fiecare cromozom  $i=(1\dots\text{pop\_size})$ .
2. Calculam  $F=\text{suma}(\text{eval}(v_i))$ .
3. Calculam probabilitatea  $p_i=\text{eval}(v_i)/F$ .
4. Calculam o probabilitate cumulata:  
$$q_i=\text{suma}(p_j) , j=(1,\dots,i)$$
5. Generam un numar  $r$  in intervalul  $[0,1]$ .
6. Daca  $r < q_1$  selectam pe  $q_1$  daca nu selectam pe  $q_i$  astfel incat  $q_i < r < q_{i+1}$

# *Exemplu: Maximizarea unei functii*

- **Incrucisare**

1. Se alege un  $p_c$  acesta reprezinta probabilitatea incrucisarii.

2. Pentru fiecare cromozom se genereaza un  $r$  din  $[0,1]$ .

3. Daca  $p_c > r$  atunci se selecteaza pentru incrucisare.

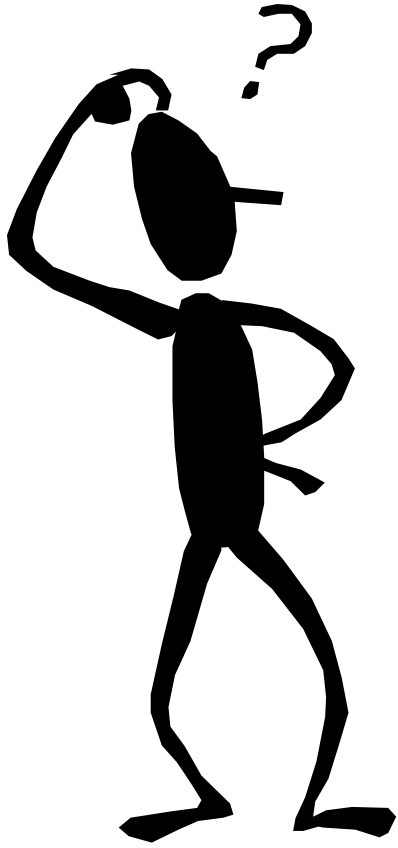
# *Exemplu: Maximizarea unei functii*

- **Mutatie**

1. Se alege un  $p_m$  acesta reprezinta probabilitatea incrucisarii.

2. Pentru fiecare bit din cromozom se genereaza un  $r$  din  $[0,1]$ .

3. Daca  $p_m > r$  atunci se aplica mutatia (daca este 0  $\Rightarrow$  1, daca este 1  $\Rightarrow$  0).



Q & A