

# Algoritmi genetici pentru rezolvarea problemelor prin evolutie si co-evolutie

Coordonator proiect:

Prof. dr. ing. **Adina Magda Florea**

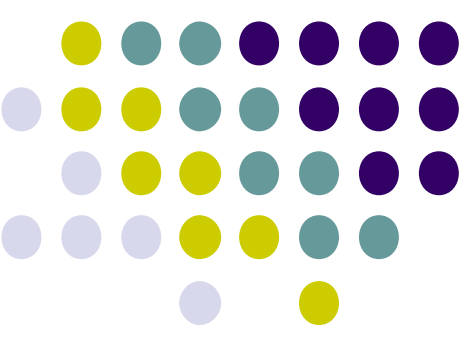
Absolvent:

**Sorin OSTAFIEV**

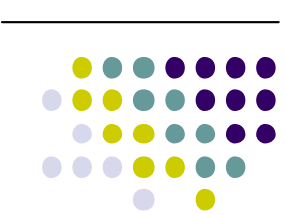
UNIVERSITATEA POLITEHNICA BUCURESTI

FACULTATEA DE AUTOMATICA SI CALCULATORARE

2003

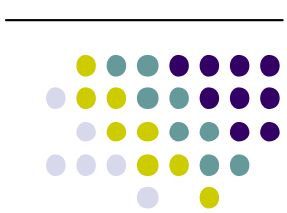


# Scopul si motivatia



- Utilizarea algoritmilor evolutivi pentru rezolvarea problemelor din ce in ce mai complexe necesita oportunitati pentru ca sub-problemele rezolvate individual sa interactioneze
- Este prezentata o arhitectura pentru co-evolutia unor astfel de componente ca o colectie de specii cooperante

## De ce algoritmi evolutivi clasici nu sunt suficient de buni?



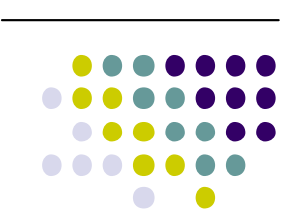
- Populatiile de indivizi au o puternica convergenta
  - se inlatura persistenta pe termen lung a sub-componentelor diverse
- Indivizii reprezinta solutii complete evaluate individual, in izolare
  - nu exista ocazii pentru aparitia co-adaptarii
- Cum ar trebui sa fie sub-componentele? Cate ar trebui sa fie?

# Pentru extinderea modelului clasic trebuie sa avem in vedere urmatoarele aspecte



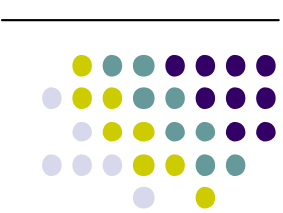
- **Descompunerea problemei**
  - de la inceput sau ca urmare a evolutiei algoritmului
- **Dependentele dintre sub-componente**
  - co-adaptarea poate aparea doar daca interactiunile sunt modelate
- **Asocierea rolurilor**
  - Determinarea contributiei fiecărei sub-componente la intreaga solutie
- **Mentineră diversitatii**
  - Diversitatea trebuie pastrata mai mult decat in cazul algoritmilor evolutivi clasici

# Funcția de fitness



- Fiecare specie evoluează în propria populație și se adaptează mediului prin intermediul algoritmului evolutiv
- Indivizii dintr-o specie sunt evaluați între ei prin colaborarea cu indivizii “reprezentativi” din alte specii

# Exemple



- **Maximizarea unei functii  $f(x)$  cu  $n$  variabile independente ( $n$  specii, cate una pentru fiecare variabila).**
  - indivizii sunt selectati in functie de cat de bine maximizeaza functia
- **Dezvoltarea unui sistem de control bazat pe reguli pentru simularea unui robot autonom**
  - fiecare populatie reprezinta un set de reguli pentru diferite comportamente
  - indivizii sunt selectati in functie de cat de bine completeaza seturile de reguli ale altor specii

# Acoperirea sirurilor



- Gasirea unui set potrivit de  $N$  vectori binari care se potrivesc (cat mai bine cu putinta) pe un set sursa de  $K$  vectori binari ( $K \gg N$ )
- In acest context, problema descompunerii consta in:
  - determinarea dimensiunii setului de potrivit
  - determinare acoperirii fiecarui element din set
- Calcularea fitness-ului unui set potrivit  $M$  se face folosind formula:

$$S(M) = \frac{1}{K} \sum_{i=1}^K \max(S(\vec{m}_1, \vec{t}_i), \dots, S(\vec{m}_N, \vec{t}_i))$$

# Acoperirea sirurilor - Exemplu



Fie setul sursa format din trei siruri de lungime 32

```
11111111111111111111111111111111
11111111111000000000000000000000
00000000000000000000000011111111
```

- Acoperirea optima folosind un sir singur:  
1111111111100000000000001111111111
- Una din acoperiri cu 2 siruri:  
11111111111111111111111111111111  
100101101100000000000001111110101
- Acoperirea cu 3 siruri este formata exact din sirurile setului sursa

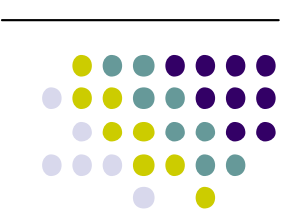


# Algoritmul de baza

```
gen = 0
for each species s do begin
     $Pop_s(gen)$  = initializeaza populatia cu valori aleatorii
    evalueaza fitness-ul fiecarui individ  $Pop_s(gen)$ 
end
while conditia de terminare = false do begin
    gen = gen +1
    for each species s do begin
        selecteaza  $Pop_s(gen)$  din  $Pop_s(gen-1)$  pe baza fitness-ului
        aplica operatorii genetici pe  $Pop_s(gen)$ 
        evalueaza fitness-ul fiecarui individ  $Pop_s(gen)$ 
    end
end
```



# Concluzii



- Arhitectura cooperativa co-evolutiva poate fi privita ca o extensie generala a oricarei paradigme evolutive si nu doar a algoritmilor genetici
  - de exemplu: software de control al robotilor in medi multi-agent
- Furnizeaza stimuli pentru evolutia spre un anumit numar interdependent de sub-componente care acopera nise multiple
- Evolueaza spre un nivel de generalitate potrivit
- Deoarece domeniile devin tot mai complexe, evolutia speciilor ar putea fi condusa de mai mult decat fitness-ul sistemului pentru a putea produce o descompunere optima