

# SISTEME CU IA

## CALCUL DE INSPIRATIE NATURALA

### Algoritmul furnicii

---

---

---

---

---

---

---

---

### Algoritmul furnicii

#### Inspiratia biologica

Urmărind o colonie de furnici, se constată că fiecare din ele pare să se miște la întâmplare, fără un scop precis. Totuși, deodată, grupuri din ce în ce mai mari de furnici se concentrează pe direcții comune. Această comportare pare să indice formarea unei **inteligente colective**, ce are la bază comportarea fiecărui individ.

---

---

---

---

---

---

---

---

### Algoritmul furnicii

#### Principii

Furnicile se orienteaza dupa urmele de feromoni cu densitati diferite.

#### Exemplu

Se consideră cazul a două grupuri de furnici, având același număr de membri (de exemplu, câte 3 furnici de fiecare grup), care trebuie să se deplaseze între punctele A și B, unde A reprezintă adăpostul sau locul de depozitare a hranei, iar B – sursa de hrană.

---

---

---

---

---

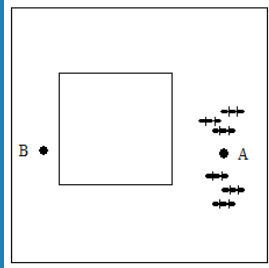
---

---

---

## Algoritmul furnicii

Exemplu



---

---

---

---

---

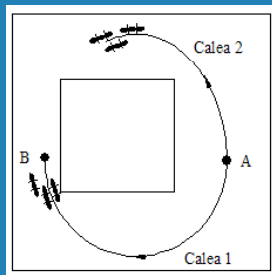
---

---

---

## Algoritmul furnicii

Exemplu



---

---

---

---

---

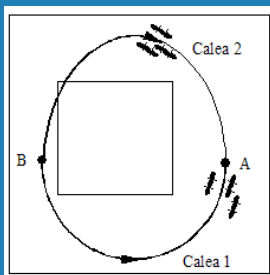
---

---

---

## Algoritmul furnicii

Exemplu



---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Implementare

Copierea comportamentului natural în cadrul unui algoritm artificial se face, de regulă, diferit, în funcție de tipul de problemă de optimizare ce se dorește a fi rezolvată. De asemenea, implementarea AF introduce o serie de modificări în raport cu modelul natural, modificări care au, în general, un caracter simplificator.

Exemplu → **Problema comis voiajorului.**

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Problema comis voiajorului

Un comis voiajor trebuie să viziteze o serie de clienți ce locuiesc într-un anumit număr de orașe.

Se dorește să se stabilească ordinea în care trebuie vizitate orașele astfel încât lungimea totală a itinerarului parcurs de comis voiajor să fie minimă.

---

---

---

---

---

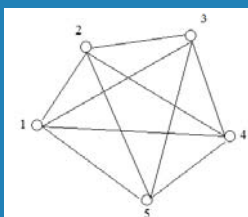
---

---

---

## Algoritmul furnicii

### Problema comis voiajorului



Soluția prezentată folosește o reprezentare de forma unui graf complet ce conține  $N$  noduri (orașele vizitate de comis voiajor) și toate laturile ce pot fi definite între aceste noduri.

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Problema comis voiajorului

Algoritmul – atasat in forma tiparita.

1. Fiecărei laturi  $(i, j)$  din graful complet  $i$  se asociază o concentrație de feromoni  $\tau_{ij}$ , care va fi folosită pentru selectarea traseului urmat de colonia de furnici, care în continuare vor fi denumite *agenți*. Inițial, concentrațiile  $\tau_{ij}$  se stabilesc la valori pozitive mici (de exemplu,  $c = 0.01$ ).

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Problema comis voiajorului

2. Cei  $NA$  agenți se distribuie uniform (dacă este posibil) între nodurile din graf. Dacă se admite că numărul de noduri și numărul de agenți se aleg astfel încât  $NA = m \cdot N$  (unde  $m$  este un întreg oarecare), în fiecare nod se vor distribui  $m = NA / N$  agenți. Nodul în care se află la un moment dat fiecare agent  $k$  este notat  $Oras_k$ .

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Problema comis voiajorului

3. Deoarece fiecare agent trebuie să treacă prin fiecare oraș fără a forma cicluri (un oraș se vizitează o singură dată, cu excepția primului oraș, în care se încheie turul), memorarea traseului fiecărui agent se face într-o listă TABU, care va conține  $N + 1$  elemente, care descriu succesiunea în care sunt vizitate orașele. De exemplu, dacă 5 orașe sunt vizitate de un agent în ordinea 3, 1, 4, 5 și 2, lista tabu a acestuia va avea forma (3, 1, 4, 5, 2, 3).

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Problema comis voiajorului

4. **Initializari:** Deoarece problema urmărește determinarea unei valori minime, lungimea traseului minim se inițializează cu o valoare foarte mare, notată simbolic  $LMin = \infty$ .

După distribuirea agenților în nodurile grafului, lista TABU asociată fiecărui agent este inițializată în prima poziție cu numărul de ordine al orașului în care a fost distribuit acel agent.

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Problema comis voiajorului

5. **Deplasarea agentilor in noduri pana la completarea listelor TABU:** pentru fiecare agent  $k$ , se consideră orașul de origine  $i = Oras_k$  și se alege orașul destinație  $j^*$ , care nu trebuie să figureze în lista TABU a agentului  $k$ , pe baza unor probabilități calculate în funcție de concentrația de feromoni de pe laturile  $(i, j)$  și vizibilitatea nodurilor  $j$  „privite” din nodul  $i$ , apreciată prin inversul distanței dintre noduri,  $d_{ij}$ .

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Problema comis voiajorului

Probabilitatea de deplasare a agentului  $k$  din orasul  $i$  in orasul  $j$ :

$$P_{ij} = \begin{cases} \frac{(\tau_{ij})^\alpha \cdot (1/d_{ij})^\beta}{\sum_{p \notin Tabu_k} (\tau_{ip})^\alpha \cdot (1/d_{ip})^\beta} & j \notin Tabu_k \\ 0 & j \in Tabu_k \end{cases}$$

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Problema comis voiajorului

6. **Completarea listelor TABU:** după parcurgerea a  $N$  pași, când toți agenții au trecut prin toate nodurile grafului, traseul fiecărui agent se încheie prin revenirea în nodul de origine. În continuare, se calculează lungimile traseelor pentru toți agenții și, dintre acestea, se memorează lungimea minimă. În acest moment se încheie un pas / o iterație a algoritmului.

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Problema comis voiajorului

7. **Actualizare concentrații feromoni:** Înainte de trecerea la pasul următor, se procedează la actualizarea concentrației de feromoni de pe fiecare latură a grafului. Modificarea cantității de feromoni de pe laturile grafului se face o singură dată, la încheierea fiecărei iterații. Cantitatea de feromoni adăugată pe fiecare latură este invers proporțională cu lungimea întregului traseu parcurs de agentul respectiv.

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Problema comis voiajorului

Actualizarea concentrațiilor de feromoni:

$$\tau_{ij} = \rho \cdot \tau_{ij} + \Delta \tau_{ij}$$

Corecția concentrațiilor de feromoni:

$$\Delta \tau_{ij} = \sum_{k=1}^{M_k} \Delta \tau_{ij}^k$$

Corecții elementare:

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{daca } i, j \in \text{Tabu}_k \text{ si } i = \text{Tabu}_k(p); j = \text{Tabu}_k(p+1) \\ 0 & \text{in caz contrar} \end{cases}$$

---

---

---

---

---

---

---

---

## Algoritmul furnicii

### Caracteristici

- este un algoritm de inspirație naturală.
- este un algoritm paralel și distribuit.
- este un algoritm de tip cooperant.
- este un algoritm versatil.
- este un algoritm robust.

---

---

---

---

---

---

---

---

**SFARSIT**

---

---

---

---

---

---

---

---