Algoritmi evolutivi

1 CERINȚE

1. Să se implementeze **un algoritm evolutiv pentru problema rucsacului**.
    a. Codificare binară
    b. Operatori specifici (încrucişare, mutaţie)
    c. Algoritm și parametrizare
    d. Experimente pe cele două instanțe primite la Tema 1

2. Să se implementeze **un algoritm evolutiv pentru problema comis-voiajorului**.
    a. Codificare prin permutări
    b. Operatori specifici (încrucişare, mutaţie)
    c. Algoritm și parametrizare
    d. Experimente pe instanța primită la Tema 2

Observaţii (neîndeplinirea cerinţelor din observaţii conduce la scăderea punctajului acordat):

- ✓ Aplicația trebuie să fie modularizată, să permită parametrizarea algoritmului și afișarea soluției.
- ✓ Testaţi algoritmul în diverse variante pentru comparaţii.
- ✓ Rezultatele experimentelor trebuie salvate (cu indicarea setărilor folosite: algoritmul, valori parametri, număr rulări, calitatea soluției – best/avg).

2 TERMEN DE PREDARE

- **Lab 4**

Total Punctaj Tema 3 = 150p

3 PREDAREA TEMEI PRIN MS TEAMS

Incarcati urmatoarele fisiere **INAINTE** de a incepe lab-ul in care este setat termenul de predare:

1. O arhiva cu codul sursa
2. Un document (Word/PDF) care sa contina:
    - ✓ Descrierea pe scurt a algoritmului implementat (pseudocod) si principalelor componente (reprezentare solutie, functie de fitness, operatori, etc)

- ✓ Indicarea parametrilor algoritmului
- ✓ Tabele/grafice cu rezultatele obtinute (comparatii pentru cel putin 3 seturi de valori ale parametrilor pentru fiecare instanta de problema)
- ✓ Grafic cu evolutia best/avg individ din populatie pe parcursul generatiilor (considerand o rulare oarecare)
- ✓ Analiza rezultatelor

4 ALGORITM EVOLUTIV

Algoritmul evolutiv pentru problema rucsacului va fi implementat pe baza unei reprezentări binare, repararea indivizilor care reprezintă soluții nevalide (pentru a respecta capacitatea rucsacului) și o funcție de fitness ce calculeaza valoarea totală pusă în rucsac (ce trebuie maximizată).

Algoritmul evolutiv pentru problema TSP va fi implementat pe baza unei reprezentări prin permutări, folosirea unor operatori de încrucișare și mutaţie care generează soluţii valide și o funcţie de fitness ce calculează distanţa totală parcursă (ce trebuie minimizată).

Componentele algoritmului sunt:

- Generare populație inițială aleator (populație P).
- Selecție părinți.
- Generare descendenți: încrucișare și mutație (populație O).
- Selecție supraviețuitori: cei mai buni size(P) din P+O.
- Parametri algoritm: mărimea populației, numărul de generații, probabilitatea de încrucișare, probabilitatea de mutație.

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```