Algoritmi metaeuristici.

Lab 7:
Optimizare multi-modala
Antrenarea evolutivă a rețelelor neuronale

---

1. Optimizare multimodala

In functie de scopul urmarit problemele de optimizare pot fi grupate in urmatoarele categorii:

- *Optimizare globala:* se cauta configuratia pentru care valoarea functiei obiectiv este cea mai mica (in cazul problemelor de minimizare) sau cea mai (in cazul problemelor de maximizare) dintre toate valorile posibile pentru configuratiile din spatial solutiilor.
- *Optimizare locala:* se cauta, in vecinatatea unei configuratii curente, cea mai buna configuratie. Un optim local este o configuratie mai buna decat toate cele din vecinatate insa poate fi mai putin buna decat optimul global.
- *Optimizare multimodala:* se cauta toate optimele, atat cele locale cat si cele globale; este utila atunci cand exista mai multe optime globale sau optimele locale sunt acceptabile din punct de vedere al calitatii. Necesitatea de a identifica mai multe optime apare frecvent in inginerie cand se pune problema determinarii parametrilor structurali ai unor sisteme (de exemplu identificarea punctelor de rezonanta ale unui sistem mecanic sau electric).

Determinarea tuturor optimelor poate fi realizata in cel putin doua moduri:

- Aplicarea repetata a unui algoritm de optimizare locala (pornind din diferite configuratii initiala)
- Determinarea mai multor optime in cadrul unei singure rulari a unui algoritm bazat pe populatii astfel incat la sfarsitul executiei algoritmului elementele populatiei sa fie concentrate in vecinatatea fiecarui optim

Metaeuristicile destinate optimizarii globale au ca scop orientarea intregii populatii catre optim, astfel incat la finalul procesului iterativ populatia sa conveargа catre configuratia optima. Aceasta inseamna ca, treptat, populatia se restrange pierzandu-se din diversitate. In cazul optimizarii multimodale scopul urmarit este diferit: populatia trebuie sa ramana suficient de diversa pentru a putea acoperi fiecare dintre optime. Asigurarea diversitatii populatiei poate fi realizata prin diferite mecanisme care urmaresc constituirea (implicita sau explicita) a unor subpopulatii care au ca scop fiecare dintre ele identificarea unui optim. Aceasta se bazeaza pe stimularea elementelor populatiei de a „vizita” diferite zone din spatiul, ceea ce se poate implementa in mai multe moduri:

- Prin penalizarea elementelor aflate in zone aglomerate (scorul unui element se imparte la o valoare proportionala cu numarul de elemente aflate in vecinatatea, definita de o anumita raza, elementului evaluat). Strategia este cunoscuta sub numele de „sharing” si este folosita atat la optimizarea mult-modala cat si la optimizarea multi-criteriala.
- Prin aplicarea unor operatori (de exemplu, selectie) la nivel local, adica intre elemente apropiate. Una dintre cele mai utilizate strategii, denumita „crowding”,

Exemplu: Crowding DE este un exemplu in care o modificare simpla a regulii de selectie transforma un algoritm de optimizare globala intr-un algoritm de optimizare multi-modala. Crowding DE a fost propus initial de Thomsen (in „Multimodal optimization using crowding-based differential evolution”, CEC 2004). Algoritmul DE (vezi lab 5) se modifica doar in etapa

de selectie in sensul ca noul element z_i (construit prin mutatia bazata pe diferente si incrucisare) nu il inlocuieste pe x_i ci pe cel mai apropiat element din populatie (daca este mai bun decat el). Fig 1 ilustreaza efectul modificarii strategiei de selectie

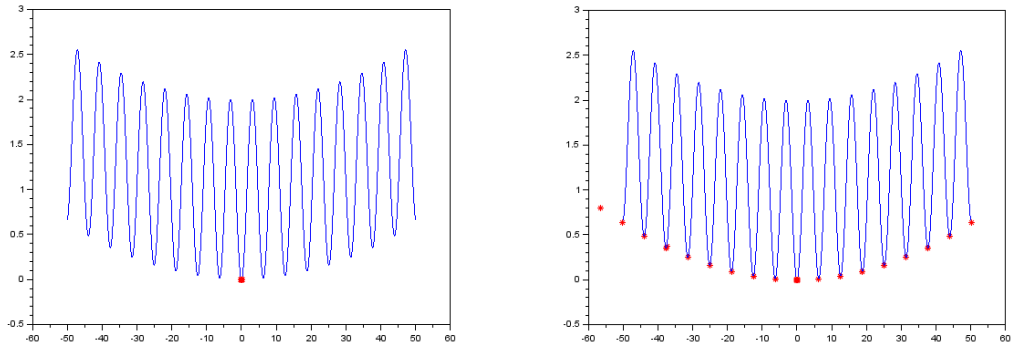

Fig. 1. Distributia unei populatii cu 30 de elemente dupa 500 de generatii dupa aplicarea algoritmului DE (stanga) respectiv a algoritmului DE combinat cu crowding (dreapta)

Aplicatie 1. Modificati implementarea algoritmului DE (din lab 5) pentru a incorpora selectia de tip crowding si testati algoritmul pentru functii multi-modale uni si bi-dimensionale. Indicatie: vezi Decrowding.sci

2. Proiectarea evolutivă a rețelelor neuronale artificiale

O retea neuronala artificiala este un *graf etichetat* ce poate fi utilizat pentru a modela dependenta dintre date de iesire si date de intrare (astfel de dependente pot interveni in probleme de clasificare, estimare, predictie). Retelele neuronale sunt modele de tip "black-box" ceea ce inseamna ca nu se construies reguli explicite care sa descrie dependenta dintre datele de iesire si cele de intrare.

Proiectarea unei retele neuronale presupune:

- Stabilirea arhitecturii (numar de unitati, functii de activare, mod de interconectare unitati)
- Alegerea algoritmului de antrenare, a setului de antrenare si antrenarea propriu-zisa a retelei (estimarea valorilor ponderilor)
- Validarea retelei (analiza comportarii pentru exemple care nu fac parte din setul de antrenare).

Algoritmii evolutivi pot fi utilizați în contextul proiectării rețelelor neuronale pentru:

- Determinarea ponderilor (în locul algoritmilor de antrenare bazați pe metode de optimizare locală se folosește o strategie evolutivă). Aceasta abordare este utila cand functiile de transfer sunt nederivabile sau arhitectura retelei este recurenta (nu se poate calcula valoarea erorii explicit in functie de ponderile retelei).
- Stabilirea arhitecturii retelei (de exemplu

Aplicatie 2. Determinarea evolutivă a ponderilor unei rețele neuronale

Considerăm problema antrenării unei rețele neuronale pentru reprezentarea disjuncției exclusive. Se consideră o rețea constituită din:

- 3 unități de intrare (două efective și una corespunzătoare pragului)
- K unități ascunse (K este un parametru de intrare) cu funcție de transfer de tip Heaviside

- o unitate de ieşire cu funcţie de transfer de tip Heaviside

Fiecare element din populaţie conţine 4*K+1 elemente corespunzătoare tuturor ponderilor şi valorilor prag. Criteriul de optimizat este reprezentat de eroarea medie pătratică pe setul de antrenare iar algoritmul utilizat poate fi o strategie evolutivă.

Indicaţie. Se adaugă la aplicaţia de la strategii evolutive o funcţie pentru evaluarea erorii medii pătratice în cazul unei reţele cu arhitectură specificată (vezi SEretea.sci).

Exercitiu. Adaptati unul dintre algoritmii Differential Evolution sau Particle Swarm Optimization pentru estimarea parametrilor unei retele neuronale feed-forward.

Temă.

1. Extindeţi aplicaţia cu antrenarea evolutivă a unei reţele neuronale (pentru reprezentare XOR) pentru cazul în care numărul de unităţi ascunse este variabil (elementele populaţiei vor conţine atât numărul de unităţi ascunse cât şi valorile ponderilor – prin urmare elementele populaţiei vor avea număr variabil de componente). Observaţie: e necesară adaptarea operatorilor de recombinare şi mutaţie.