**Report Tema 2 Algoritmi Genetici**

**Descrierea algoritmului** (reprezentarea, selectia, operatorii utilizati):

Reprezentarea numereleor a fost facuta pe 32 de biti inclusive bitul de semn, numerele reale au fost reprezentate ca numere naturale: au fost inmultite cu 1000 si partea intreaga a fost convertita, la conversia inversa se imparte cu 1000 astfel se pot pastra 3 zecimale.Precizia poate fi modificata printr-un parametru in functiile de conversie.

Selectia a fost realizata cu un algoritm de selectie de tip turneu: se aleg aleatoriu POP\_SIZE/10 indivizi din populatie si se salveaza prima jumatate, procesul se repeta pana cand noua populatie este la fel de mare ca cea initiala.  
Am implementat si algoritmul „Roata norocului” dar rezultatele erau mult mai slabe decat la algoritmul turneu, populatia convergea mult prea repede intr-un rezultat complet gresit indiferent de ceilalti parametrii.

Crossover-ul a fost realizat in *n* puncte –cate dimensiuni are parametrul atatea puncte de taiere, iar mutatia s-a realizat pe un singur bit. Punctele in care s-au realizat cele doua au fost alese aleatoriu, dar astfel incat numerele sa ramana in intervalul functiei.

Probabilitatea de cross este de 50% pentru fiecare individ si de mutatie 30% pentru fiecare individ, prin rulari repetate acestea au aduc rezultatele cele mai bune in cazul meu.

Conditia de oprire a algoritmului este:

1. Un numar fixat de iteratii de 1000
2. La fiecare 10 iteratii se calculeaza minimul populatiei, daca acesta nu se modifica pentru 5 verificari consecutive algoritmul se opreste. (Nu se modifica = Diferenta dintre cele doua este mai mica decat 10-5 )

Am incercat si cu distanta Hamming medie a rezultatelor si deviata standard dar in primul caz nu functiona bine datorita reprezentarii folosite, iar deviatia era prea diferita pentru fiecare functie pentru a fi practica si de multe ori oprea algoritmul mult prea devreme.

**Influenta valorilor parametrilor**

public static double GetMinimum(IFunction function, ISelection selector, int dimensions, int size, double crossChance = 0.5, double mutationChance = 0.3)

Acesta este declaratia functiei de minim

* function - functia care trebuie minimizata
* selector – selectia folosita
* dimensions – numarul de dimensiuni pe care se va face minimizarea
* size – marimea populatiei: cu cat e mai mare cu atat mai greu va converge catre un singur rezultat si va dura mai mult rularea, dar cu cat e mai mare cu atat rezultatul va fi mai bun, pentru o functie simpla (ex DeJong) recomand ~100, pentru o functie cu multe puncte de minim (ex Rastrigin) sunt necesare peste 1000.
* crossChance – sansa unui individ de a intra in „pool”-ul pentru cross, cu cat e mai mare cu atat mai greu va converge populatia si cu atat mai rapid se va face explorare. O valoare prea mica dupa la convergere prematura, o valoare prea mare la pierdere de indivizi cu fitness bun. Defalut: 0.5
* mutationChance – sansa unui individ de a suferi o mutatie, la fel ca la crossChance, prea mare duce la pierdere, prea mica la convergere prematura. Default: 0.3

Pe langa acestea exista si maxIterations care este setat constant 1000 si distanta la care doua numere reale sunt considerate egale (10-5)

Pentru detalii numerice accesati fisierul excel atasat.