Descriere\_Problema1

* Programul primeste ca parametrii numele fisierului in care se afla datele de intrare (numarul de obiecte, obiectele, greutatea ghiozdanului), populatia(adica numarul de indivizi ai unei populatii), numarul de generatii si probabilitatea de incrucisare
* Programul contine 2 clase (una pentru Obiect cu parametrii: index, greutate, valoare iar cealalta pentru Solutie cu parametrii: solutie sub forma de vecori de biti si valoarea totala), un modul unde se citesc din fisier si se afiseaza datele in fisier, un modul cu functiile utile rezolvarii problemei:
  + - Prima functie genereaza o solutie random sub forma unui vector de biti de lungimea numarului de obiecte.
    - A doua functie calculeaza greutatea unei solutii generate random. Se parcurge vectorul de biti si se inmuteste greutatea obiectului de pe indicele i cu valoarea bitului.
    - A treia functie verifica daca este o solutie. Verifica daca greutatea obiectelor este mai mica egala cu capacitatea ghiozdanului. In caz afirmativ returneaza True.
    - A patra functie(generaresolutiirandom) genereaza o solutie random.
    - Functia generarepopulatie genereaza n solutii random, unde n semnifica numarul de indivizi din populatie.
    - Functia Calculfitnessindivid primeste o solutie si returneaza fitnessul ei.
    - Functia GenereazaFitnessPopulatie calculeaza pentru fiecare individ din populatie fitnessul si returneaza un vector cu fitnessul fiecaruia.
    - Functia SelectieTorunament allege aleator 2 indivizi din populatie si-l returneaza pe cel mai bun.
    - Functia SelectieRuleta aplica algoritmul ruletei pe vectorul de indivizi in functie de fitnessul fiecaruia.
    - Functai generare2parinti allege ce algoritm sa aplice pentru generarea parintiilor.
    - Functia generarecopii genereaza 2 copii din cei 2 parinti primiti ca parametru prin metoda punctului de taietura.
    - Functia GenerareParintiSiCopii genereaza 2\*n parinti si 2\*n copii unde n este numarul de indivizi din populatie. Functia se foloseste de functiile de mai sus.
    - Functia de mutatie interschimba un bit random din 1-0 respectiv 0-1.
    - Functia de GenerareMutatie aplica cu o anumita probabilitate mutatii declor 2\*n copii generate.
    - Functia de GenerareSupravieturitori calculeaza cei mai buni n indivizi din mutatii si parinti pe care ii returneaza.

Mai avem un modul cu algoritmul:

1. Se genereaza automat n indivizi random pentru prima generatie. Pentru fiecare generatie se calculeaza: parintii, copii in umar de 2\*n, se fac mutatii asupra copiilor cu o anumita probabilitate si apoi se aleg urmatorii supravieturitori. Pentru urmatoarea generatie se va repeta algoritmul cu populatia fiind supravietuitorii din generatia precedenta.

Ultimul modul este main-ul care colecteaza de la tastatura parametrii.

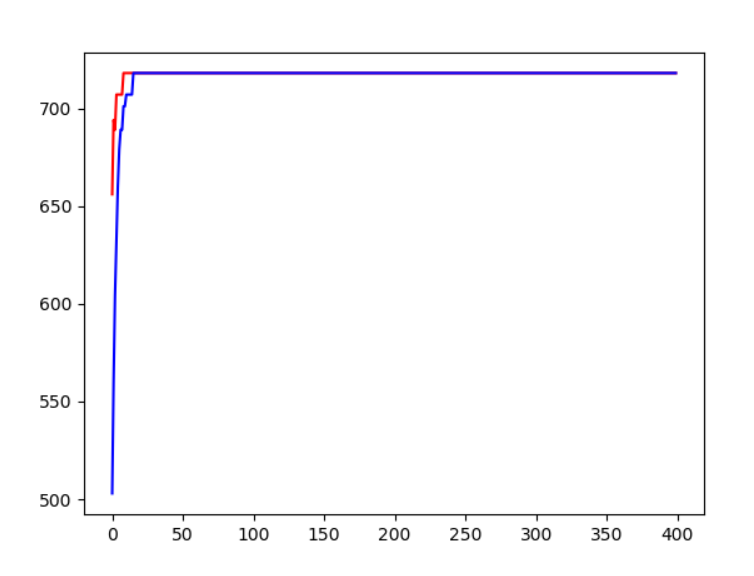
* Pentru setul de date care se afla in fisierul **Date2.txt** avem urmatorul tabel:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Numar rulari | Marime  Populatie | Numar  Generatii | Probabilitate  Mutatie | Best Solutie | Avg solutie |
| 10 | 100 | 10 | 0.4 | 726 | 705.4 |
| 20 | 0.8 | 726 | 703.7 |
| 50 | 0.6 | 726 | 703.6 |
| 150 | 50 | 0.8 | 726 | 705.8 |
| 100 | 0.6 | 726 | 706.9 |
| 150 | 0.5 | 726 | 712.3 |
| 200 | 50 | 0.8 | 726 | 710.3 |
| 150 | 0.6 | 726 | 720.5 |
| 200 | 0.5 | 726 | 710.2 |

* Pentru setul de date care se afla in fisierul **Date3.txt** avem urmatorul tabel:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Numar rulari | Marime  Populatie | Numar  Generatii | Probabilitate  Mutatie | Best Solutie | Avg solutie |
| 10 | 100 | 10 | 0.4 | 134255 | 133238.5 |
| 20 | 0.8 | 134255 | 133561.5 |
| 150 | 50 | 0.8 | 134613 | 133750.1 |
| 100 | 0.6 | 134441 | 133786.2 |
| 150 | 0.5 | 134144 | 133830.1 |
| 200 | 50 | 0.8 | 134200 | 133833.6 |
| 150 | 0.6 | 134347 | 133896.7 |
| 200 | 0.5 | 134446 | 134154.7 |

* Datele pentru care s-au facut observatiile din tabel se gasesc in fisierul Date2date.txt
* Datele pentru care s-au facut observatiile din tabel se gasesc in fisierul Date3date.txt
* Cu cat crestem populatia si numarul de generatii cu atat se obtin solutii mai optime dar si timpul este mai lung. Pentru populatii si numar de generari mari se obtin solutii mai bune decat maximele obtinute cu ceilalti algoritmiHC/SA/TS. Coeficientul de mutatie este optim sa fie in jur de 0.5 ca sa existe un echilibru intre exporare si exploatare. Din tabel se observa ca generatiile crescute conduc la un rezultat mai bun decat populatia crescuta.
* O imbunatatire a algoritmului este de a nu pelca cu solutii random, de a pleca cu solutii obtinute de unul din algoritmii de HC, TS, SN.



Se oberva ca dupa 25 de generatii se ajunge la solutia maxima. Best-ul este notat cu rosu iar low este notat cu albastru.