Descriere\_Tema\_Partea\_1

Pb. Rucsacului Tabu Search

* Programul primeste ca parametrii numele fisierului in care se afla datele de intrare (numarul de obiecte, obiectele, greutatea ghiozdanului), numarul de iteratii si numarul de iteratii tabu. Programul afiseaza in 2 fisiere diferite, unul pentru afisarea solutiei folosind Tabu Search varianta standard iar celalalt pentru afisarea solutiei folosind Tabu Search varianta imbunatatita.
* 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(calculvaloare) calculeaza valoarea unei solutii. Se procedeaza la fel ca si pentru calcularea greutatii. Aceasta este functia de fitness.
    - Functia calcularevecininontabu calculeaza vecinii unei solutii. Primeste ca parametru solutia si vectorul tabu, genereaza toti vecinii nontabu si il returneaza pe cel cu valoarea cea mai mare.
    - Functia UpdateMemory primeste ca parametrii lista tabu, pozitia unde se va updata si iteratiile tabu. Functia scade de pe toate pozitiile o iteratie tabu si, in vector pe pozitia poz se va adauga numarul de iteratii tabu.
    - Functia maximsolutie primeste ca parametru 2 solutii si in functie de valoarea fiecareia o returneaza pe cea cu valoarea mai mare.
    - Functia celmaibunvecinnontabu1 calculeaza vecinii nontabu si, in functie de un vector unde se retine de cate ori s-a schimbat un bit, se afiseaza solutia cea mai buna care este supusa uni penalty(din valoarea solutiei se scade penalty-ul).

Mai avem un modul cu calculul solutiilor dupa k iteratii si n iteratii tabu. Aici avem 2 functii:

1. Functia de generare care primeste ca parametru numarul de iteratii si numarul de iteratii tabu. Pornim cu o solutie valida. Intr-un while calculam cel mai bun vecin, facem un update la memoria si intr-o variabila retinem solutia cu fitnessul cel mai mare dintre vecin si solutia initiala. Apoi reluam while-ul cu solutia initiala ca fiind solutia optima din cele doua.
2. Functia de generare1 este un update la functia de generare care foloseste un vector unde se retine de cate ori s-a schimbat un bit.

Ultimul modul este main-ul care colecteaza de la tastatura parametrii(numarul de iteratii si iteratiile tabu)

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algoritm | Numar rulari | Valoare iteratii | Valoare iteratii tabu | Best Solutie | Avg solutie |
| Tabu  Search | 10 | 10 | 5 | 716 | 651.3 |
| 10 | 707 | 654.5 |
|  | 15 | 710 | 659.6 |
|  | 50 | 5 | 726 | 722.5 |
|  | 10 | 726 | 704 |
|  | 15 | 726 | 695.4 |
|  | 100 | 5 | 726 | 721.4 |
|  | 10 | 726 | 704.1 |
|  | 15 | 726 | 697.5 |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Algoritm | Numar rulari | Valoare iteratii | Valoare iteratii tabu | Best Solutie | Avg solutie |
| Tabu  Search | 10 | 10 | 5 | 132243 | 131,394.7 |
| 10 | 132836 | 131,687.2 |
|  | 15 | 132145 | 131,730 |
|  | 50 | 5 | 133003 | 131878.3 |
|  | 10 | 132448 | 131,713.5 |
|  | 15 | 133047 | 132,149.4 |
|  | 100 | 5 | 132838 | 132,125.9 |
|  | 10 | 132647 | 132,052.8 |
|  | 15 | 132340 | 131,706.7 |

* 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
* Se observa din tabel ca pentru un numar de iteratii tabu mai mic se obtine, in general, media solutiilor mai buna decat pentru un numar de iteratii tabu mai mare .Solutia optima este 726/133047 care in comparative cu cautarea locala si cea de Hill-Climbing este una mult mai buna analog si pentru media solutiilor.In consecinta TabuSearch este o tehnica mai buna de cautare a solutiei optime pentru problema Rucascului
* Algoritmul se bazeaza pe explorare pe cand algoritmii de cautare Hill-Climbing se bazeaza pe exploatare..putand sa se blocheze in maxime locale pe cand Tabu Seach, in general, nu se poate bloca pe maxime locale, generand, in cele mai multe din cazuri, solutii mai optime.