

Hill Climbing pentru problema rucsacului

Având exemplul cu problema rucsacului de la algoritmi evolutivi (pe care o adaug și mai jos în document), să se implementeze un algoritm Hill Climbing pentru rezolvarea unei instanțe a rucsacului.

Noutăți față de problema aranjării damelor și față de comis-voiajor:

- Avem reprezentare binară (doar zero și unu), deci inițializarea va fi mult mai simplă. Lungimea tabloului este egală cu numărul de obiecte existente.
- Evaluarea va fi diferită. Calculăm modul de capacitatea rucsacului minus suma obiectelor selectate în tablou (cele care corespund valorilor de 1).
- Perturbarea este diferită, dar mai simplă. Poziția aleasă se modifică din 1 în 0 sau din 0 în 1. Altfel spus, din i în $1 - i$.

Programul ar trebui să poată să se aplice pentru orice capacitate a rucsacului și pentru orice număr de obiecte. Evident, se dau valorile obiectelor.

Aveți două exemple de intrare pentru programul mai jos:

1. Capacitate = 15
Număr obiecte = 5
Obiecte = [10, 2, 1, 4, 3]
2. Capacitate = 368
Număr obiecte = 30
Obiecte = [45, 40, 60, 88, 12, 69, 73, 95, 65, 46, 71, 37, 10, 66, 38, 83, 91, 27, 24, 22, 51, 35, 63, 51, 76, 95, 21, 11, 90, 28]

Soluție

Soluție
<p>Observații: Programul de mai jos tratează doar instanța a doua, pentru a putea fi ușor urmărit. El se poate însă modifica ușor să rezolve și prima problemă.</p> <pre>#include <stdio.h> #include <stdlib.h> #include <time.h> #include <math.h> #include <iostream> #include <chrono> #define MILION 1000000 #define nr_obiecte 30 #define capacitate 368 using namespace std;</pre>

```

int nrEvaluari = 0;
void initializare(int d[]) {
    for (int i = 0; i < nr_obiecte; i++)
        d[i] = rand() % 2;
}

int evaluare(int d[], int greutati[], int cap) {
    nrEvaluari++;
    int interm = 0, cost = 0;
    for (int i = 0; i < nr_obiecte; i++)
        interm = d[i] * greutati[i] + interm;

    cost = abs(capacitate - interm);

    return cost;
}

void perturbare(int d[]) {

    int x = rand() % (nr_obiecte);
    d[x] = 1 - d[x];

}

int main() {
    int d[nr_obiecte], d1[nr_obiecte], obiecte_in_rucsac[nr_obiecte], xm = 0;

    int greutati[nr_obiecte] = { 45, 40, 60, 88, 12, 69, 73, 95, 65, 46, 71, 37,
        10, 66, 38, 83, 91, 27, 24, 22, 51, 35, 63, 51, 76, 95, 21, 11, 90, 28
    };

    srand(time(NULL));
    int k = 50;
    clock_t startTimp = clock();
    double timp = 0;

    int cost;
    do
    {
        initializare(d);
        for (int i = 0; i < nr_obiecte; i++)
            d1[i] = d[i];
        cost = evaluare(d, greutati, capacitate);
        for (int i = 0; i < k; i++) {
            perturbare(d);
            int evalPerturb = evaluare(d, greutati, capacitate);
            if (cost > evalPerturb) {
                cost = evalPerturb;
                for (int j = 0; j < nr_obiecte; j++) {
                    d1[j] = d[j];
                }
            }
        }
        clock_t curent = clock();
        timp = (double)(curent - startTimp) / CLOCKS_PER_SEC;
    } while ((cost != 0) && (timp < 3) && (nrEvaluari < MILION));
    cout << "Obiectele date sunt:\n";
    for (int i = 0; i < nr_obiecte; i++) {
        cout << greutati[i] << " ";
    }

    cout << "\n\nSolutia gasita: " << endl;
    for (int i = 0; i < nr_obiecte; i++) {

```

```

        cout << d1[i] << " ";
    }

    cout << endl << "\nCostul solutiei este " << cost << "." << endl;
    cout << endl << "Obiectele introduse in rucsac sunt:\n";
    for (int i = 0; i < nr_obiecte; i++) {
        if (d1[i] == 1) {
            cout << greutate[i] << " ";
            xm += greutate[i];
        }
    }
    cout << "\n\nValoarea obiectelor introduse in rucsac este de " << xm << ".";
    cout << "\n\nTimpul de rulare este de " << timp << " secunde." << endl;
    cout << "\nFunctia de evaluare a fost apelata de " << nrEvaluari << " ori."
<< endl;

    return 0;
}

```

Programul rulat

```

Obiectele date sunt:
45 40 60 88 12 69 73 95 65 46 71 37 10 66 38 83 91 27 24 22 51 35 63 51 76 95 2:
11 90 28

Solutia gasita:
0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 0 1

Costul solutiei este 0.

Obiectele introduse in rucsac sunt:
60 88 12 37 24 22 51 35 11 28

Valoarea obiectelor introduse in rucsac este de 368.

Timpul de rulare este de 0.003 secunde.

Functia de evaluare a fost apelata de 10353 ori.

```