

Tema0 Algoritmi Genetici

Balteanu Andrei

October 6, 2021

1 Introducere

Subiectul acestei teme este gasirea punctelor de minim si de maxim folosind doua metode: determinista si nedeterminista.

O metoda este determinista daca pentru rulara de mai multe ori a unui input-ul al problemei, programul va trece mereu prin aceleasi set de instructiuni si va furniza mereu acelasi output.

O metoda este nedeterminista daca pentru rulara de mai multe ori a unui input, programul nu va trece mereu prin aceleasi instructiuni si output-ul difera.

2 Metoda determinista

2.1 Datele de intrare

De la tastatura utilizatorul va introduce initial numarul de variabile pe care functia o va avea.

Dupa introducerea acestei date, utilizatorul va trebui sa introduca pentru fiecare variabila a functiei urmatoarele date:

- 1)puterea variabilei
- 2)coeficientul variabilei
- 3)semnul variabilei

Se poate observa mai jos un exemplu de input al acestor date:

```
INTRODUCETI NUMARUL DE VARIABLE:2
INTRODUCETI PUTEREA VARIABLEI CU NUMARUL 1:2
INTRODUCETI COEFICIENTUL VARIABLEI CU NUMARUL 1:2
SEMNUL DIN FATA VARIABLEI CU NUMARUL 1:+
INTRODUCETI PUTEREA VARIABLEI CU NUMARUL 2:1
INTRODUCETI COEFICIENTUL VARIABLEI CU NUMARUL 2:3
SEMNUL DIN FATA VARIABLEI CU NUMARUL 2:+
```

2.2 Algoritm

Algoritmul este unu simplu.Se vor lua toate combinatiile posibile de valori pentru fiecare variabila din cele "n" variabile introude de utilizator, in cazul acestui cod am luat un domeniu in care fiecare variabila poate avea maxim valoarea 15 si minim valoare 0.

Minimul functiei va fi salvat intr-o variabila cu numele minim si va pleca cu o valoare foarte mare.

Maximul functiei va fi salvat intr-o variabila cu numele maxim si va pleca cu o valoare foarte mica.

Apoi folosind datele introduse de utilizator(puterea variabiliea, coeficient, semn) se va calcula valoarea functiei pt respectiva combinatie.

Daca acesta valoarea a functiei este mai mare decat variabila maxim, maxim se va actualiza, iar daca valoarea functiei este mai mica decat minim, variabila minim se va actualiza.

La final dupa parcurgerea tuturor posibilitatilor, se vor afisa variabilele minim si maxim.

2.3 Exemplu de rulare a algoritmului

Am luat urmatoarele date:

O functie cu 2 variabile de forma : $2*x*x + 3*y$ Inca o data tin sa precizez ca am luat predefinit ca domeniul variabilelor este $[0,15]$, doar numerele naturale. Output-ul este de forma urmatoare:

```
INTRODUCETI NUMARUL DE VARIABLE:2
INTRODUCETI PUTEREA VARIABLEI CU NUMARUL 1:2
INTRODUCETI COEFICIENTUL VARIABLEI CU NUMARUL 1:2
SEMNUL DIN FATA VARIABLEI CU NUMARUL 1:+
INTRODUCETI PUTEREA VARIABLEI CU NUMARUL 2:1
INTRODUCETI COEFICIENTUL VARIABLEI CU NUMARUL 2:3
SEMNUL DIN FATA VARIABLEI CU NUMARUL 2:+
minimul functiei este egal cu: 0
maximul functiei este egal cu: 495
C:\Users\Andrei\source\repos\Tema9_Alg_determinist\Debu
```

3 Metoda nedeterminista

3.1 Datele de intrare

Datele de intrare raman aceleasi ca si la metoda determinista. Un singur lucru difera in ceea ce priveste:

Utilizatorul va introduce numarul de incercari pe care algoritmul il va face pentru a gasi minimul si maxim.

3.2 Algoritm

Fiind vorba de un algoritm nedeterminist am ales ca in loc sa actualizez maxim si minim (unde e cazul) in timp ce parcurg valorile codomeniului, sa le salvez intr-un vector numit codomeniu.

```
while (vector[numar_variabile - 1] != 16)
{
    int valoare_functie = 0;
    for (int i = 0; i < numar_variabile; i++)
    {
        if (parametru[i].operatie == '+')
        {
            valoare_functie = valoare_functie + pow(vector[i], parametru[i].putere_variabila) * parametru[i].coeficient_variabila;
        }
        else if (parametru[i].operatie == '-')
        {
            valoare_functie = valoare_functie - pow(vector[i], parametru[i].putere_variabila) * parametru[i].coeficient_variabila;
        }
    }
    codomeniu[numar_elemente_codomeniu++] = valoare_functie;

    vector[0]++;
    for (int i = 0; i < numar_variabile - 1; i++)
    {
        if (vector[i] == 16)
        {
            vector[i] = 0;
            vector[i + 1]++;
        }
    }
}
```

Apoi selectez cu o functie de random cate un element din vectorul codomeniu.

Acolo unde va fi cazul, se va actualiza variabilele "minim" si "maxim". Dupa ce se vor termina numarul de incercari pe care l-a introdus utilizatorul, se vor afisa rezultatele cautarii random.

3.3 Exemplu de rulare a algoritmului

Funcția are aceeași ca la algoritmul determinist : $2*x*x + 3*y$.

După câteva rări am ajuns chiar la un rezultat destul de aproape de cel real:

```
INTRODUCETI NUMARUL DE VARIABLE:2
INTRODUCETI PUTEREA VARIABLEI CU NUMARUL 1:2
INTRODUCETI COEFICIENTUL VARIABLEI CU NUMARUL 1:2
SEMNUL DIN FATA VARIABLEI CU NUMARUL 1:+
INTRODUCETI PUTEREA VARIABLEI CU NUMARUL 2:1
INTRODUCETI COEFICIENTUL VARIABLEI CU NUMARUL 2:3
SEMNUL DIN FATA VARIABLEI CU NUMARUL 2:+
INTRODUCETI NUMARUL DE INCERCARI:100
0 465
```

References

<https://ro.theastrologypage.com/non-deterministic-algorithm>
Grafic Calculator