

Raport pentru tema numarul 0

October 14, 2018

Lipan Radu-Matei

1 Problema

Enunt: Sa se determine minimul folosind o metoda euristica pentru urmatoarele functii: De Jong 1, Schwefel 7, Rastrigin 6, Six-hump camel back.

2 Algoritm utilizat

Se va folosi un algoritm ce va calcula valoarea functiei in nrGen puncte alese aleatoriu conform unei distributi uniforme pe intervalul de definitie al functiei. Dintre aceste puncte se va alege valoarea minima ca rezultat. Acest algoritm se va rula de un nrRep de ori si se va alege iarasi valoarea minima dintre rezultate. Rezultatul final va fi o aproximare a minimului global.

2.1 Pseudocod

```
// o iteratie a algoritmului
def Test(...)
    min = +oo
    for (g = 0; g < nrGen; g = g + 1)
        x = valori_aleatoare_distribuite_uniform();
        if ( f(x) < min )
            min = f(x)
    return min
```

```
// rulam algoritmul de nrRep ori
def TestRunner (...)
    min_final = Test()
    for (i = 1; i < nrRep; i = i + 1)
        rez = Test()
        if ( rez < min_final)
            min_final = rez
    return min_final
```

2.2 Detalii de implementare

Intantele pentru functiile date sunt vectori de nrDim dimensiuni de numere reale (tip double in C++). Rezultatele functiilor sunt reprezentate tot ca valori reale.

Parametrii principali folositi sunt:

- nrDim : numarul de dimensiuni al datelor de intrare pentru fiecare functie sau dimensiunea vectorului x.
- nrGen : numarul de vectori x generati intr-o rulare a algoritmului.
- nrRep : numarul de rulari ale algoritmului.

Vecinii unui vector sunt reprezentati de ceilalti nrGen-1 vectori generati aleatoriu intr-o aceasi rulare a algoritmului. Initializarea variabilelor se face in mod aleatoriu ele fiind uniform distribuite pe intervalul de definitie al functiei. Minimul este initializat cu valoarea maxima pentru double(intuitiv: cu o valoare ce reprezinta $+\infty$).

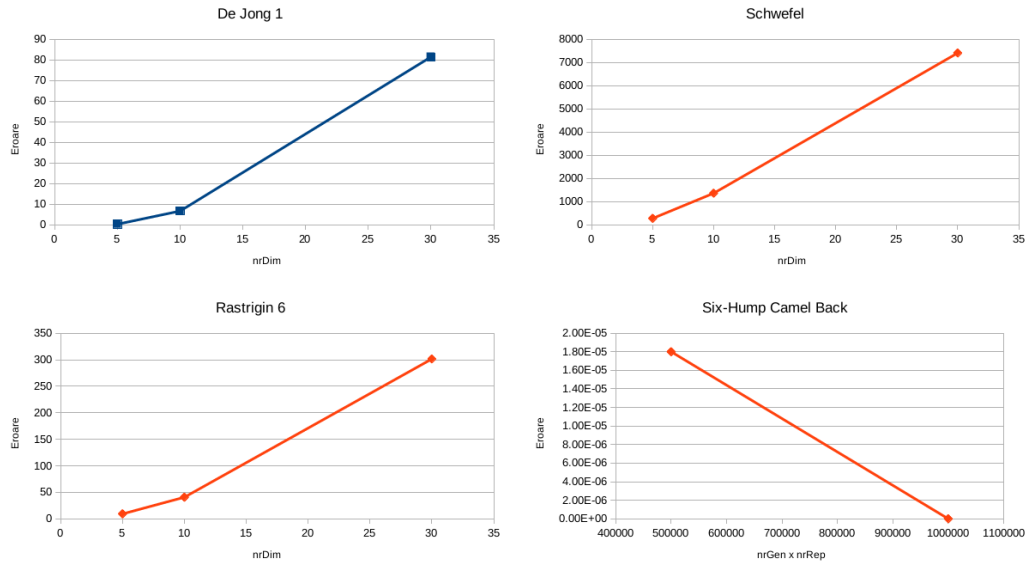
Algoritmul se termina dupa nrRep rulari.

3 Rezultate

3.1 Rezultate experimentale

	nrDim	nrGen	nrRep	Minim global	Minim obtinut	Maxim pe rep	Medie pe rep	Deviatie Standard	Timp mediu repetitie
De Jong 1	5	10000	50	0	0.298799	3.053265	1.23636	0.48827	0.003129
De Jong 1	10	10000	50	0	6.669661	19.158886	12.723105	2.980399	0.005873
De Jong 1	30	10000	50	0	81.420343	133.344081	111.333603	10.892122	0.017011
Schwefel 7	5	10000	50	-2094.9145	-1819.191097	-1394.925529	-1589.112758	98.230312	0.00454
Schwefel 7	10	10000	50	-4189.829	-2829.706785	-2037.712639	-2336.651766	168.957583	0.008652
Schwefel 7	30	10000	50	-12569.487	-5163.686087	-3451.60805	-4087.47515	380.952267	0.025187
Rastrigin 6	5	10000	50	0	9.210386	24.134191	15.537532	4.278077	0.004285
Rastrigin 6	10	10000	50	0	40.69892	85.270627	70.118466	7.678903	0.008234
Rastrigin 6	30	10000	50	0	301.426302	372.237388	349.02681	14.66686	0.023552
Six-hump camel back	2	10000	50	-1.0316	-1.031582	-1.024972	-1.029517	0.276763	0.002221
Six-hump camel back	2	10000	100	-1.0316	-1.031622	-1.019927	-1.029185	0.052652	0.00216
Six-hump camel back	2	5000	50	-1.0316	-1.031618	-1.013238	-1.027087	0.032674	0.001232

3.2 Grafice



3.3 Influenta Parametrilor

Observam ca pentru metoda aleasa principalul parametrul de care depinde puterea de aproximare a algoritmului este nrDim. Pentru numar mic de dimensiuni aproximarea rezultatului este buna chiar daca modificam usor ceilalti doi parametri (de exemplu Six-hump camel back care are doar 2

dimensiuni).

Odata cu marirea numarului de dimensiuni puterea de aproximare a algoritmului scade puternic deoarece creste domeniul in care este cautat punctul de minim si o simpla functie de alegere aleatoare nu mai face fata.

Modificarea celorlalti doi parametri (nrGen si nrRep) ajuta la obtinerea a unor aproximari ceva mai bune dar cu un impact semnificativ asupra timpului de executie.

4 Parametrii pentru rezultatele obtinute

Numarul de rulari, generari si executii cu care au fost obtinute rezultatele experimentale:

- 5,10 sau 30 dimensiuni
- 50 sau 100 rulari
- 10.000 sau 5.000 generari