

# Raport tema nr 1

## 1 Descrierea problemei

Să se determine minimul pentru următoarele funcții: De Jong 1, Schwefel 7, Rastigrin, Six-hump camel back implementând algoritmi Hill Climbing (variantele First Improvement și Best Improvement) și Simulated Annealing.

## 2 Algoritmul utilizat

Ideea algoritmului constă în faptul că, pentru o soluție curentă  $x = (x_1, \dots, x_n)$ , vecinii acestei soluții pot conduce la soluții mai bune. Deci evaluăm funcția în vecinii soluției curente, folosind algoritmi Hill Climbing și Simulated Annealing.

### 2.1 Hill Climbing, pseudocod:

Pentru Best Improvement:

Cât timp (mai există iterații de făcut)

1. Generează o soluție random și evaluează funcția pentru această soluție.
2. Generează vecinii acestei soluții.
3. Evaluează funcția pentru fiecare vecin.
4. Selectează cel mai bun vecin - bestNeighbour.
5. Dacă bestNeighbour este mai bun decât soluția curentă, soluția curentă devine bestNeighbour, și sare la punctul 2. Altfel, trece la următoarea iterație.
6. Returnează cea mai bună soluție.

Pentru First Improvement:

În loc să selecteze cel mai bun vecin, algoritmul selectează primul vecin care îmbunătățește soluția, chiar dacă nu este cel mai bun.

### Simulated Annealing, pseudocod:

Cât timp (mai există iterații de făcut)

1. Generează o soluție random și evaluează funcția pentru această soluție.
2. Cât timp (nu s-a ajuns la o răcire absolută)
  - a) Generează vecinii soluției curente și selectează un vecin random.
  - b) Evaluează funcția pentru acel vecin - neighbourValue.
  - c) Soluția curentă devine neighbourValue dacă neighbourValue este mai bun sau dacă temperatura curentă permite selectarea lui, chiar dacă nu este mai bun, și sare la punctul 2. Altfel, sare la punctul 1 (la următoarea iterație).
  - d) Temperatura se răcește.
3. Returnează cea mai bună soluție.

## 2.1 Detalii de implementare

Reprezentarea soluțiilor a fost făcută pe șiruri binare, astfel: “Spațiul de căutare se va discretiza până la o anumită precizie  $10^{-d}$ . Un interval  $[a, b]$  va fi împărțit în  $N = (b - a) * (10^d)$  subintervale egale. Pentru a putea reprezenta cele  $(b - a) * (10^d)$  valori, este nevoie de un număr  $n = \lceil \log_2 N \rceil$  de biți. Lungimea șirului de biți care reprezintă o soluție candidat va fi suma lungimilor reprezentărilor pentru fiecare parametru al funcției de optimizat. În momentul evaluării soluției (apelul funcției de optimizat) este necesară decodificarea fiecărui parametru reprezentat ca șir de biți în număr real, după formula:  $X_{real} = a + decimal(x_{biti}) * (b - a) / (2^n - 1)$ .”

Vecinii sunt construiți prin schimbarea câte unui bit, pe rând, în reprezentarea binară a soluției.

Inițializarea constă în crearea random a unei soluții binare.

Pentru Hill Climbing, condiția de oprire este negăsirea niciunui vecin care să îmbunătățească soluția curentă, semn că am ajuns la un minim. Pentru Simulated Annealing, pe lângă condiția de oprire de la Hill Climbing, mai avem și condiția ca temperatura să nu fi ajuns la o valoare foarte mică.

În funcție de punctul de plecare al temperaturii și de funcția de răcire, algoritmul Simulated Annealing se comportă foarte asemănător cu Hill Climbing, deoarece, dacă temperatura aleasă este foarte mică, șansa ca o soluție mai proastă să treacă mai departe este mult mai mică, iar Hill Climbing nu alege niciodată soluții proaste; astfel cele două metode sunt asemănătoare.

## 3 Rezultate experimentale

Algoritm	Funcție	Dimensiuni	Rulări	Minim	Medie	Maxim	Deviație	Minim	Medie	Maxim	Deviație
				Best Improvement				First Improvement			
Hill Climbing	De Jong	5	30	1.204	4.647	9.574	4.966	1.137	4.296	8.205	5.987
		10		10.44	23.24	34.67	12.28	14.69	25.40	41.40	9.950
		30		98.07	145.3	168.8	20.53	134.2	149.7	169.2	8.969
	Rastrigin	5	30	14.21	30.57	43.29	18.54	14.21	30.20	39.74	14.49
		10		68.71	93.05	114.2	18.50	66.86	96.10	116.2	21.05
		30		337.5	390.8	423.1	24.60	335.9	392.3	417.1	18.74
	Schwefel 7	5	30	-1801	-1260	0	505	-1810	-1188	0	457.5
		10		-2128	-1871	0	186.8	-2714	-1909	0	329.6
		30		-5525	-5489	0	51.47	-5529	-5499	0	38.92
	Six-hump camel back	2	30	-1.02	-0.95	0	0.38	-1.03	-0.96	0	0.235

Algoritm testat	Funcție testată	Nr dimensiuni	Nr rulări	Minim	Medie	Maxim	Deviație
Simulated Annealing	De Jong	5	30	1.49	4.01	7.117	4.2
		10		7.89	23.7	32.6	10
		30		98.07	145.9	162.1	15.68
	Rastrigin	5	30	13.7	25.7	38.08	16.96
		10		65.17	93.85	106.6	15.82
		30		360.9	392.6	423.7	16.25
	Schwefel 7	5	30	-1642	-1247	0	354.4
		10		-2361	-1820	0	440
		30		-5417	-3266	0	594.9
	Six-hump camel back	2	30	-1.03	-1	0	0.16

Rezultatele pentru Simulated Annealing au fost obținute cu temperatura inițială de 100 de grade, răcindu-se după funcția:  $g(T) = T/3$ .

Din tabel, se observă tendința algoritmilor Best Improvement de a da rezultate mai bune decât First Improvement. Algoritmul Hill Climbing are rezultate mai bune decât Simulated Annealing.

