Raport tema1

Stratianu Bianca-Ionela

Octombrie 2020

1 Introducere

1.1 Introducere generala

Acest raport analizeaza diferenta dintre Hill Climbing in cele doua variante:First Improvement si Best Improvement, si Simulated Annealing, diferente care conta in minimul gasit si timpul in care algoritmii dau solutia pentru patru functii specificate, iar numarul de dimensiuni fiind 5, 10 si 30, pentru a compara cat mai bine cele 3 metode.

1.2 Motivatie

Multe probleme din viata reala se regasesc ca fiind probleme de baza in domeniul informaticii.De exemplu ajungi dintr-un loc in altul in cel mai scurt timp, ajungandu-se astfel la problema optimizarii globale. Motivatia pentru acest raport este posibilitatea de a gasi cea mai buna si mai rapida metoda de aflare a minimului unei functii.

1.3 Descrierea problemei

Se dă: o funcție

$$f : A \to R, f : A \to R$$

pentru o mulțime A de numere reale Se cere: un element

$$x_0 \in A, x_0 \in A pentru care f(x_0) \leq f(x), \forall x \in A f(x_0) \leq f(x), \forall x \in A$$

Astfel vom folosi functiile:
De Jong:

$$f(x) = \sum_{i=1}^{n} x_i^2 x_i, \in [-5.12, 5.12], i = 1, 2, ..., n \tag{1}$$

Minimul global este: f(x)=0, x(i)=0, i=1:n.

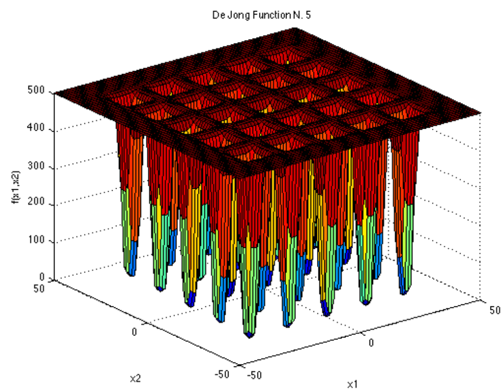


Figure 1: Functia De Jong pentru cinci variabile

Schwefel:

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i sin(\sqrt{|x_i|}) x_i \in [-500, 500]\,, i = 1, 2, ..., n$$

Minimul global este: f(x)=-n·418.9829; x(i)=-n*420.9687, i=1:n, unde n este numarul de dimensiuni.

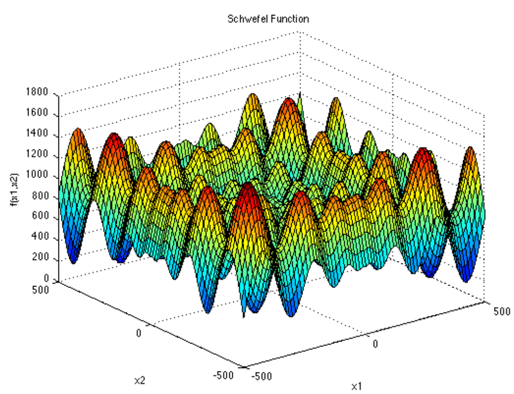


Figure 2: Functia Schwefel pentru doua variabile

Rastrigin:

$$f(x) = 10n + \sum_{i=1}^{n} (x_i^2 - 10cos(2\pi x_i)), x_i \in [-5.12, 5.12]\,, i = 1, 2, ..., n$$

Minimul global este: f(x)=0; x(i)=0, i=1:n.

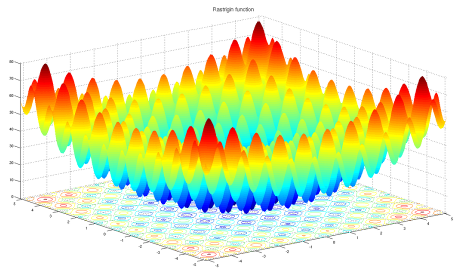


Figure 3: Functia Rastrigin pentru doua variabile.

Michalewicz:

$$f(x) = -\sum_{i=1}^{n}(sin(x_i)(sin(ix_i^2/pi))^{2^m}, i = 1:n, m = 10, x_i \in [0, pi], i = 1, 2, ..., n, pi = 3.141$$

Minimul global: f(x)=-4.687 (n=5), f(x)=-9.66 (n=10)

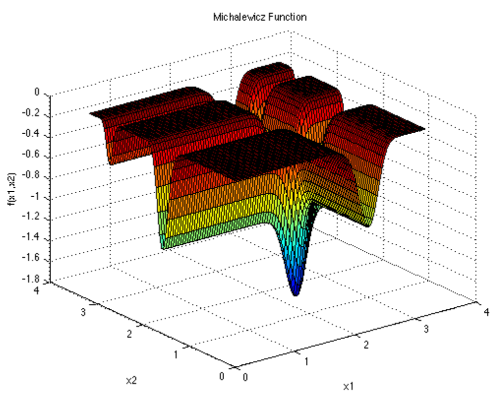


Figure 4: Functia Michalewicz pentru doua variabile.

2 Metode

Ca metode am folosit: Hill Climbing(First si Best Improvement) si Simulated Annealing si am analizat algoritmii pe cele patru functii. Pentru reprezentare am folosit siruri de biti transformati in valori de numere reale. Fiecare dintre algoritmi cauta un vecin care este mai bun decat valoarea valoarea noastra. Pentru eficienta, cautarea unui vecin se face prin schimbarea unui singur bit. Pentru cei trei algoritmi putem modifica numarul de dimensiuni. Vecin inseamna orice sirul initial de biti la care se modifica un singur bit. Spatiul de cautare este discretizat pana la precizia de 10^{-5} pentru Hill Climbing, iar pentru Simulated Annealing este de 10^{-3}. Un interval [a,b] va fi impartit in l=$(b-a)*10^d$ subintervale egale. Pentru a putea reprezenta cele l valori avem nevoie de un numar L=parteaintreagasuperioara($log2(l)$) de biti. In momentul evaluarii solutiei este necesara decodificarea fiecarui parametru reprezentat ca sir de biti in numar real.

2.1 Hill Climbing(First Improvement)

Prima valoare este initializata aleatorie. Algoritmul alege dintre vecini prima valoare care este mai buna decat valoarea noastra. Conditia de oprire este ca se gaseste minimul local sau se depaseste numarul de rulari de 100. Precizia este de 5.

2.2 Hill Climbing(Best Improvement)

Prima valoare este initializata aleatorie. Algoritmul alege dintre toti vecinii pe cea mai buna valoarea decat valoarea noastra. Conditia de oprire este ca se gaseste minimul local sau se depaseste numarul de rulari de 100. Precizia este de 5.

2.3 Simulated Annealing

Repetam de 150 de ori alegerea la intamplare a unui vecin, indiferent daca este sau nu este mai bun decat valoarea noastra. Vedem daca vecinul ales este mai mic decat valoarea initiala sau daca este o solutie candidat pentru a iesi dintr-un minim local. Temperatura (T) este initializata initial cu 1000 si pentru fiecare iteratie este inmultita cu 0.999. Conditia de oprire este: temperatura mai mica decat $10e^{-8}$. Precizia este de 3.

3 Experimente

Pentru efectuarea experimentului s-au utilizat: n=5, n=10, n=30 (numarul de variabile), precizia de 5, iar numarul de repetari pentru Hill Climbing fiind de 100 de ori, iar pentru Simulated Annealing este de 150 de ori..Pentru a avea rezultate cat mai bune si apropiate de realitate, am repetat fiecare algoritm de cate 30 de ori.

3.1 Hill Climbing(First Improvement)

Functia De Jong	n=5	n=10	n=30
Minimul	0	0	0
Maximul	0	0	0
Media	0	0	0
Timpul	120.346	371.096	670.567

Functia Schwefel	n=5	n=10	n=30
Minimul	-2067.82	-3893.14	-10039.9
Maximul	-1915.25	-3611.27	-8398.18
Media	-1998.36	-3720.47	-9089.47
Timpul	58.1466	474.599	1094.81

Functia Rastrigin	n=5	n=10	n=30
Minimul	0.9959	6.2315	32.1376
Maximul	3.730	11.9498	47.8378
Media	2.240	8.8733	38.8054
Timpul	26.49	171.012	4713.43

Functia Michalewicz	n=5	n=10	n=30
Minimul	-4.683	-9.3713	?
Maximul	-4.5146	-8.6956	?
Medie	-4.6148	-8.81774	-26.3187
Timpul	29.995	194.994	5029.47

3.2 Hill Climbing(Best Improvement)

Functia De Jong	n=5	n=10	n=30
Minimul	0	0	0
Maximul	0	0	0
Medie	0	0	0
Timpul	150.37	235.787	350.67

Functia Schwefel	n=5	n=10	n=30
Minimul	-2094.6	-4002.01	-7888.44
Maximul	-1976.47	-3798.66	-5245.3
Medie	-2060.78	-3908.48	-6635.18
Timpul	113.606	825.602	308.601

Functia Rastrigin	n=5	n=10	n=30
Minimul	0.9959	4.2355	24.8999
Maximul	3.23544	8.4873	37.1067
Medie	1.5978	6.90409	32.6737
Timpul	50.497	299.855	8010.83

Functia Michalewicz	n=5	n=10	n=30
Minimul	-4.68702	-9.4479	?
Maximul	-4.56206	-9.000	?
Medie	4.66613	-9.1431	-26.2254
Timpul	50.9975	336.688	7997.55

3.3 Simulated Annealing

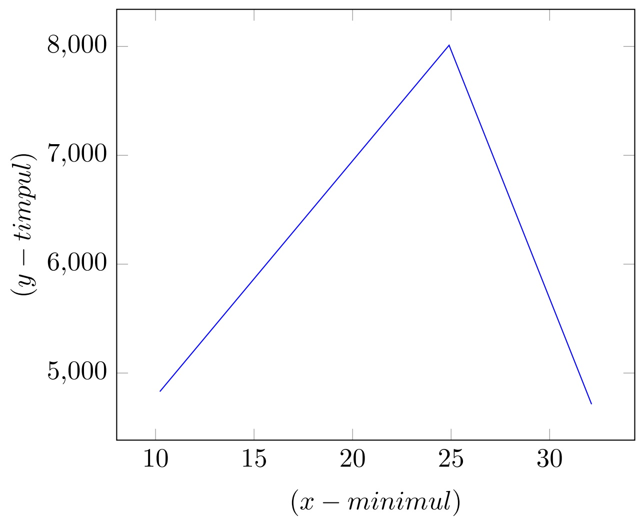
Functia De Jong	n=5	n=10	n=30
Minimul	0	0	0
Maximul	0	0	0
Medie	0	0	0
Timpul	1509.568	3062.47	6590.754

Functia Schwefel	n=5	n=10	n=30
Minimul	-2094.81	-4189.62	-7027.32
Maximul	-2060.37	-3934.2	-6398.88
Medie	-2082.77	-4159.49	-66.27
Timpul	1353.77	2136.55	6097.76

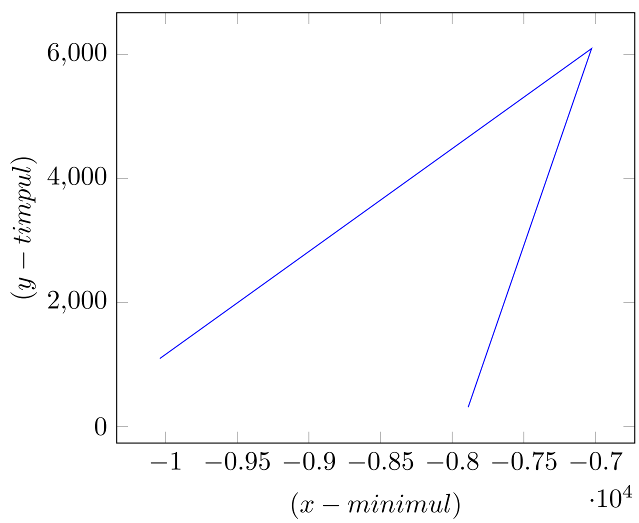
Functia Rastrigin	n=5	n=10	n=30
Minimul	0.99	2.9879	10.2196
Maximul	6.243	11.2073	30.83
Medie	2.506	6.2387	23.0606
Timpul	1155.07	1837.13	4829.84

Functia Michalewicz	n=5	n=10	n=30
Minimul	-4.687	-9.533	?
Maximul	-4.223	-9.0004	?
Media	-4.555	-9.28766	-28.2091
Timpul	1362.38	2144.07	6056.68

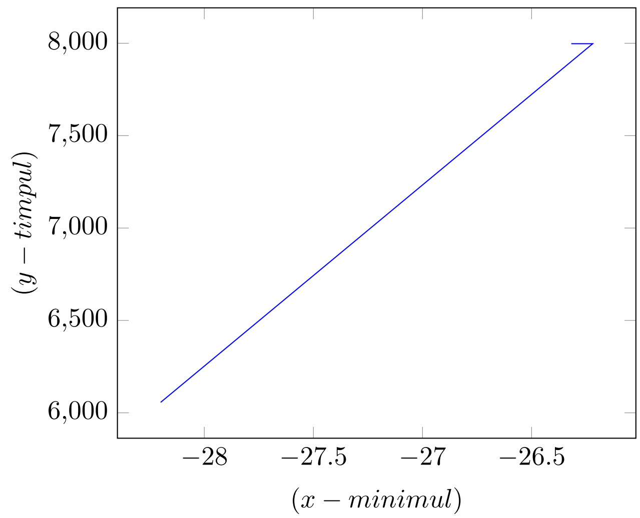
Pentru n=30 comparatii intre Simulated Annealing, Best Improvement si First Improvement pentru functia Rastrigin.



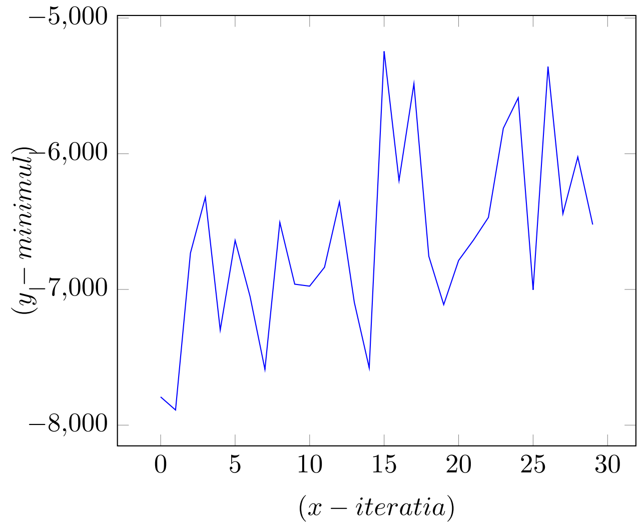
Pentru n=30 comparatii intre Simulated Annealing, Best Improvement si First Improvement pentru functia Schwefel.



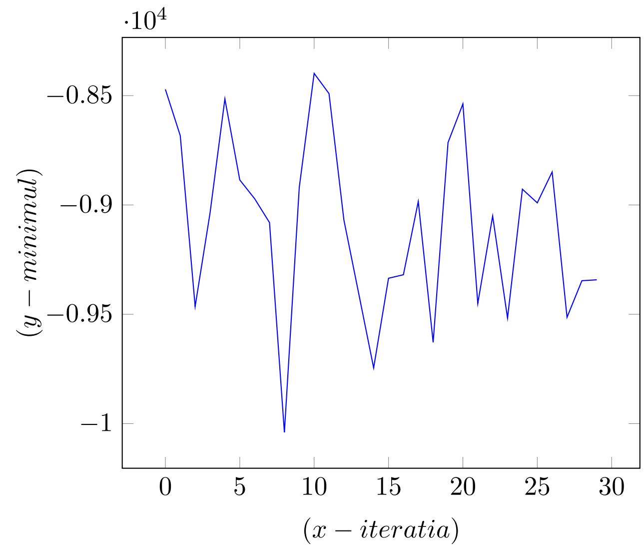
Pentru n=30 comparatii intre Simulated Annealing, Best Improvement si First Improvement pentru functia Michalewicz.



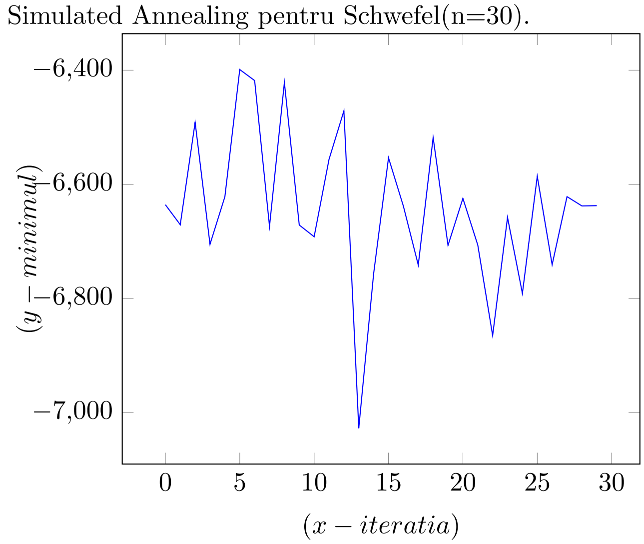
Best Improvement pentru Schwefel(n=30).



First Improvement pentru Schwefel(n=30).



Simulated Annealing pentru Schwefel(n=30).



4 Comparatii

Putem observa din grafice si din tabele ca Simulated Annealing da rezultate mai apropiate de minimul funtiilor, insa timpul este destul de mare. Intre First si Best Improvement putem observa ca dau rezultate apropiate, dar comparand timpul observam ca Best Improvement dureaza mai mult, ceea ce se justifica deoarece fata de Fist Improvement ia cel mai bun vecin ceea ce inseamna mai mute comparatii fata de First Improvement care ia primul vecin mai bun decat valoarea noastra. Observam ca pestru functia Rastrigin Simulated Annealing da cel mai bun minim, dar si cel prost timp. Pentru functia Schwefel(n=30) First Improvement da cel mai bun minim, dar are si cel mai mult timp consumat.

5 Concluzii

Acest raport se bazeaza pe comparatiile intre cele trei metode: Hill Cimbing , cu cele doua variante, First si Best Improvement, si Simulated Annealing. Astfel, putem observa ca pentru dimensiuni mici(n¡=10) Best Improvement are cele mai bune rezultate ale timpului de asteptare si ale minimului. Simulated Annealing are cele mai bune minime, dar cel mai mult timp de rulare, iar First Imrovement da un minim aproximativ, dar cu un timp putin mai bun decat la Best Improvement. Simulated Annealing depinde foarte mult de temperatura de start, cu cat o scadem, dar si conditia de acceptare a unei solutii mai rele, pentru a putea iesi dintr-un minim local. Astfel, daca ne dorim o aproximare cat mai buna a minimului putem folosi Simulated Annealing, iar daca ne dorim cat mai rapid o solutie folosi Hill Climbing. Deci, Hill Climbing este bun atunci cand avem doar un minim local minimele sunt apropiate de minimul global, iar Simulated Annealing este bun atunci cand minimul global este greu de gasit sau avem foarte multe minime locale.

6 Bibliografie

References

[1] https://profs.info.uaic.ro/ eugennc/teaching/ga/

[2] https://gitlab.com/eugennc/teaching/-/blob/master/GA/texample.pdf

[3] https://gitlab.com/eugennc/teaching/-/blob/master/GA/texample.tex

[4] https://www.youtube.com/watch?v=bSIAUkCD948

[5] https://www.youtube.com/watch?v=2BkED0TqKZY

[6] https://en.wikipedia.org/wiki/Rastrigin_function

[7] https://www.sfu.ca/~ssurjano/michal.html

[8] https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence/

[9] //www.cplusplus.com/reference/random/mersenne_twister_engine/

[10] https://en.wikipedia.org/wiki/Hill_climbing

[11] https://en.wikipedia.org/wiki/Simulated_annealing

[12] https://academicjournals.org/journal/AJMCSR/
article-full-text-pdf/EC495AE4687

[13] https://www.quora.com/What-is-main-difference-between-hill-climbing-and-simulated-anne