

# Analiza metodelor Hill Climbing și Simulated Annealing în calcularea minimului unei funcții

Haivas Daniel-Andrei

4 Noiembrie 2020

## Abstract

Atât variantele de Hill Climbing cât și metoda Simulated Annealing se apropie foarte mult de soluția corectă. Diferența majoră ar fi faptul că algoritmul de tipul Simulated Annealing ajunge să dea un răspuns bun într-un timp mai scurt, comparativ cu algoritmul Hill Climbing. Totuși, are și un minus - faptul că oferă răspunsuri destul de depărtate unele de altele.

## 1 Introducere

### 1.1 Introducere generală

Raportul prezintă în prima secțiune, "Introducere", informații introductive referitoare atât la motivarea studierii problemei aflării minimului global cât și la descrierea acesteia. În capitolul "Metode" sunt prezentați amănunțit cei 2 algoritmi, cât și micile retușuri asupra lor.

Următoarele două secvențe, "Experimente" și "Comparații", au ca scop observarea rezultatelor, dar și a modului în care rulează cei 2 algoritmi. Totodată, se măsoară diferențele între rezultatele obținute, că apoi, în "Concluzii", să se poată trage niste concluzii pertinente asupra eficienței fiecărei metode.

### 1.2 Motivație

În zilele noastre tehnologia a ajuns să capete un rol extrem de important în viața oamenilor, rezolvând tot felul de probleme din jurul nostru, facându-ne viața mai ușoară. Un rol fundamental în dezvoltarea acesteia îl reprezintă matematica și implicit studiul funcțiilor. Prin intermediul matematicii se pot rezolva o sumedenie de probleme de optim.

Algoritmii Hill Climbing și Simulated Annealing sunt 2 algoritmi ce au ca scop aflarea unor rezultate optime, cum ar fi minimul/maximul unei funcții. Astfel, în acest raport, vă fi pus în prim plan modul în care acești 2 algoritmi iterativi funcționează pe o anumită problemă din viața reală. Totodată, în urma rezultatelor obținute, vă fi ilustrată importanța practică a acestora.

### 1.3 Descriere problemă

Se cere implementarea algoritmilor: Simulated Annealing și Hill Climbing (cu first și best improvement) cu scopul determinării minimului global al unei funcții.

### 1.4 Funcții folosite

Drept studiu, vom lua următoarele funcții:

#### 1.4.1 Funcția De Jong

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

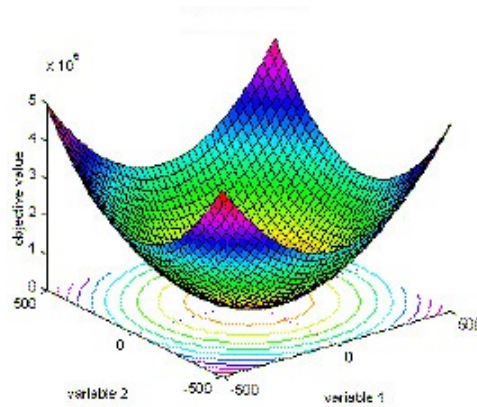


Figure 1: Funcția De Jong

Minimul global al funcției este:  $f(0, 0, \dots, 0) = 0$ .

#### 1.4.2 Funcția lui Schwefel

$$f(x) = \sum_{i=1}^b -x_i \cdot \sin\left(\sqrt{|x_i|}\right), x_i \in [-500, 500], \forall i \in [1, b]$$

Minimul global al funcției este:

$$f(420.9687, 420.9687 \dots, 420.9687) = d \cdot 418.9829$$

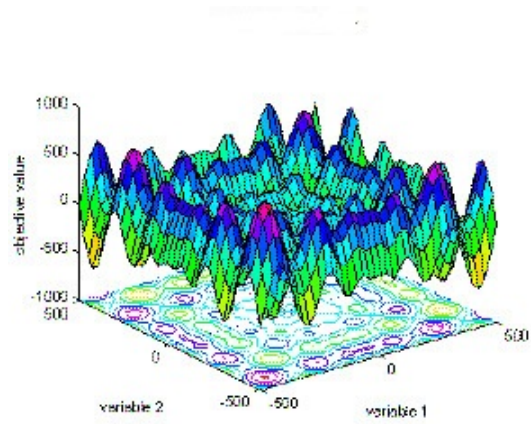


Figure 2: Funcția Schwefel

### 1.4.3 Funcția Rastrigin

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

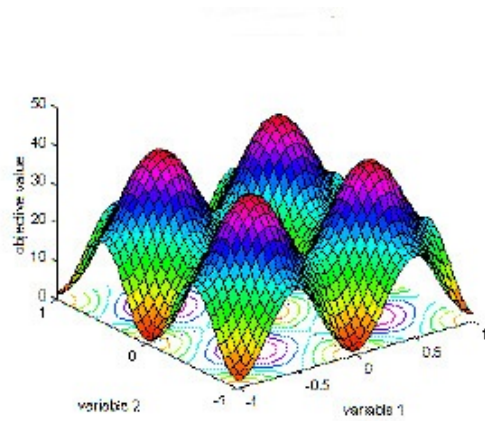


Figure 3: Funcția Rastrigin

Minimul global al funcției este:  $f(0, 0, \dots, 0) = 0$ .

#### 1.4.4 Funcția Michalewicz

$$f(x) = - \sum_{i=1}^n \sin(x_i) \cdot \left(\sin\left(\frac{i \cdot x_i^2}{\pi}\right)\right)^{2 \cdot m}, x_i \in [-5.12, 5.12], \forall i \in [1, n]$$

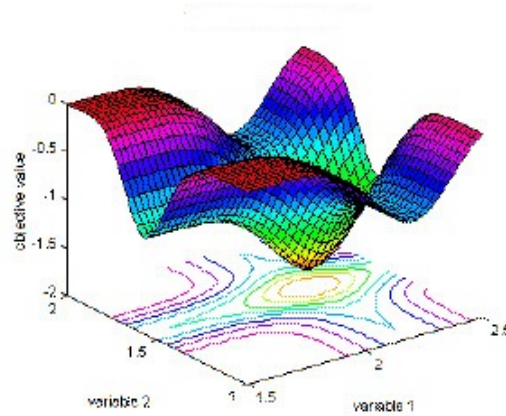


Figure 4: Funcția Michalewicz

Minimul global al funcției este:  
 $f(x) = -4.687$  ( $n=5$ );  $f(x) = -9.66$  ( $n=10$ ).

## 2 Metode

Pentru rezolvarea problemei vom folosi algoritmi: Hill Climbing (ambele versiuni) și Simulated Annealing. Ambii algoritmi presupun reprezentarea numerelor din intervalul funcțiilor, ca un sir de biți, modificarea vecinilor făcându-se începând cu biții cei mai neesențiali și continuând cu cei mai semnificativi.

Reprezentarea se face conform următoarelor notații matematice:

Fie un interval  $[a, b]$  și o precizie  $p$ . Acesta va fi împărțit în  $X$  subintervale egale ca dimensiune.

$$X = (b - a) \cdot 10^p$$

Pentru reprezentarea unei dimensiuni avem nevoie de  $l$  biți.

$$l = \lceil \log_2 X \rceil$$

Fie  $n$  dimensiunea aleasa. Șirul final de biți va fi de lungime  $L$ .

$$L = l \cdot n$$

Presupunem  $k = \text{decode}(\text{sir\_biti})$ , unde  $\text{decode}$  este o funcție care transformă un sir de biți într-un număr pozitiv. Transformarea într-un număr real  $x$ , din intervalul  $[a, b]$ , se face conform formulei:

$$x = a + k \cdot (b - a) / (2^l - 1)$$

În studiul funcțiilor enumerate am stabilit:  $[a, b]$  este intervalul fiecărei funcții,  $p = 5$ , iar  $n \in \{5, 10, 30\}$ .

## 2.1 Hill Climbing

Hill Climbing este o căutare euristica utilizată pentru probleme de optimizare matematică și în domenii precum inteligența artificială. Algoritmul este unul iterativ și se stabilește de la început un număr natural  $t$  ( $t = 100$ ), ce va spune de câte ori el va căuta un minim local.

O astfel de iterație arată în felul următor: se stabilește la început un sir aleatoriu de biți. Acest sir tot va fi îmbunătățit (prin verificarea vecinilor) până când reușește să ajungă într-un punct de minim local. Dintre cele  $t$  puncte de minim local găsite, se alege rezultatul cel mai bun.

Algoritmul are două variante iar diferență este la modul în care sunt aleși vecinii. Varianta First Improvement oprește îmbunătățirea șirului de biți în momentul în care găsește primul vecin mai bun, pe când varianta Best Improvement caută prin toți vecinii și la final îl alege pe cel mai bun. Evident, cea de-a doua variantă va da rezultate mai bune, dar va lua mai mult timp la rulare.

## 2.2 Simulated Annealing

Simulated Annealing este o metoda probabilistică de rezolvare a problemelor de optimizare ce a apărut ca o îmbunătățire pentru varianta Hill Climbing, care uneori rămâne blocată în puncte de minim local.

Acest algoritm este similar cu procedeul de încălzirea al unui material, urmat de răcirea sa lentă astfel încât să i se reducă defectele. Din acest motiv, putem observa variabila  $T$  inițializată cu o valoare mare, care ulterior scade încet ( $T \cdot \alpha = 0.99$ ) până ajunge aproape de 0.

Spre deosebire de Hill Climbing cu variantele sale First/Best Improvement, acest algoritm alege vecini aleatoriu (și nu îi ia în ordine), lucru care se repetă de 150 de ori (și nu până când ajungem într-un punct de minim local).

## 3 Experimente

Metodele au fost testate pentru fiecare din funcțiile prezentate mai sus, pe fiecare din următoarele dimensiuni  $\{5, 10, 30\}$ . Fiecare din cei 3 algoritmi (Hill Climbing First Improvement, Hill Climbing Best Improvement, Simulated Annealing) a fost rulat de 30 de ori, pentru a oferi valori relevante, din punct de vedere statistic.

### 3.1 Hill Climbing First Improvement

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	1.1921e-10	1.1921e-10	1.1921e-10	0	61.97445
10	2.38419e-10	2.38419e-10	2.38419e-10	0	555
30	7.15257e-10	7.15257e-10	7.15257e-10	0	7876

Table 1: De Jong

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	-2094.81	-1895.42	-1976.159	47.65022	122.1466
10	-3997.17	-3549.19	-3732.425	107.9329	1280
30	-10801.7	-10245.6	-10525.45	197.818	14271

Table 2: Schwefel

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	0.994959	4.46156	2.276433	0.8503629	23.539
10	3.46661	11.1741	8.245509	1.937947	455.6541
30	33.7079	51.93	42.39509	5.539547	6817.777

Table 3: Rastrigin

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	-3.69886	-3.68145	-3.693657	0.006790391	52.94284
10	-8.47857	-7.78534	-8.115823	0.1566827	461.8379
30	-25.7113	-23.6385	-24.77363	0.4696328	6961.977

Table 4: Michalewicz

### 3.2 Hill Climbing Best Improvement

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	1.1921e-10	1.1921e-10	1.1921e-10	0	130.9994
10	2.38419e-10	2.38419e-10	2.38419e-10	0	798.512
30	7.15257e-10	7.15257e-10	7.15257e-10	0	13244

Table 5: De Jong

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	-2094.91	-1975.68	-2064.315	41.46196	182.2199
10	-4155.28	-3755.76	-3940.264	88.2907	1402.118
30	-11572.4	-11009.2	-11386.99	138.7506	32529.27

Table 6: Schwefel

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	2.36502e-08	2.23078	1.409466	0.6300265	113.3666
10	2.23583	8.21064	5.683328	1.568602	677.8695
30	19.3645	39.3659	33.42555	4.092921	12844.37

Table 7: Rastrigin

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	-3.69886	-3.69149	-3.697127	0.002097675	100.3404
10	-8.58399	-8.02368	-8.259936	0.1377044	474.7674
30	-26.4406	-25.0391	-25.51295	0.3690223	12453.65

Table 8: Michalewicz

Observăm ca cele două variante de Hill Climbing dau rezultate bune, doar că timpul este destul de mare.

### 3.3 Simulated Annealing

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	3.53361e-07	6.81356e-06	2.652877e-06	1.833866e-06	110.23
10	2.20209e-06	1.44894e-05	6.175147e-06	2.691621e-06	151.96196
30	1.12774e-05	4.8037e-05	2.462637e-05	7.484605e-06	248.8656

Table 9: De Jong

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	-2094.71	-1929.56	-1915.71	129.3848	130.87
10	-4189.2	-3430.76	-3651.738	171.9662	164.26685
30	-12056.24	-10704.94	-11339.19	502.8929	319.0692

Table 10: Schwefel

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	1.23582	26.0673	6.29206	3.031955	97.8766
10	2.23078	14.3548	15.39104	4.912838	152.45052
30	17.6158	49.0996	34.60136	10.0505	251.4886

Table 11: Rastrigin

n	Val. min.	Val. max.	Medie	Dev. st.	Timp (sec.)
5	-3.69878	-3.16539	-3.567933	0.1394911	137.51
10	-8.22281	-7.1581	-7.727867	0.2787746	173.95847
30	-27.4166	-24.8042	-26.54859	0.9267138	305.9644

Table 12: Michalewicz

Observăm că algoritmul Simulated Annealing dă rezultate foarte bune iar timpul este rezonabil.

## 4 Comparații

Pentru început, o să luăm drept comparație cele doua variante ale algoritmului Hill Climbing. Diferențele sunt destul de evidente: varianta First Improvement desi funcioneaza mai rapid (nu cu mult mai rapid), da rezultate mai proaste decât varianta Best Improvement.

Ambele tipuri de algoritmi au reușit să ajungă la răspunsuri foarte bune, dar cel de tip Simulated Annealing este mai bun decât oricare din variantele de Hill Climbing. Datorita faptului că acesta nu rămâne blocat în punctele de minim local, reușește să dea răspunsuri mai bune într-un timp mai scurt.

Algoritmul de tip Hill Climbing (varianta Best Improvement) reușește să egaleze pe cel de tip Simulated Annealing doar pentru dimensiuni mici ( $n = 5$ ), timpul și soluțiile fiind aproximativ egale pentru majoritatea funcțiilor. Totuși, în momentul în care se crește dimensiunea, Hill Climbing-ul ajunge la un timp de execuție mult prea mare, dând, totodată, răspunsuri mai îndepărtate.

Pe de alta parte, desi metoda Simulated Annealing învinge și la timp de execuție și la soluție, are și aceasta o mare problema: exista o diferență destul de mare între răspunsurile pe care le oferă. Putem observa că deviația standard este mai mare în cazul acestei metodei decât la Hill Climbing, lucru care ar putea face că soluția să oscileze și să dea un rezultat neașteptat.

## 5 Concluzii

În cazul algoritmului Hill Climbing, daca ar trebui să alegem una din cele doua variante, cea Best Improvement ar fi de preferat, ea oferind, în ciuda unui timp puțin mai lung, răspunsuri mai bune comparativ cu implementarea First Improvement.



Totuși, dacă aducem în discuție algoritmul Simulated Annealing, nici-una din variantele de mai sus ale algoritmului Hill Climbing nu pot ajunge să îl egaleze la performanță, cu excepția cazurilor în care avem dimensiuni mici. Așadar, pentru aflarea minimului global al unei funcții de mai multe dimensiuni este de preferat algoritmul Simulated Annealing cu o mențiune foarte importantă.

Acest algoritm prezintă un comportament destul de periculos (lucru dovedit de valorile deviației standard în tabelele corespunzătoare), din cauza răspunsurilor care sunt destul de depărtate unul de celălalt. Astfel, în cazul unor dimensiuni suficient de mari, temperatura trebuie scăzută și mai încet (de la  $T^* = 0.99$  la  $T^* = 0.999$  sau chiar mai mult) pentru a evita riscul că algoritmul să ofere răspunsuri greșite.

## References

- [1] <https://profs.info.uaic.ro/~eugennc/teaching/ga/>
- [2] <https://gitlab.com/eugennc/teaching/-/tree/master/GA>
- [3] <https://www.sfu.ca/~ssurjano/sumsq.html>
- [4] <http://www.cplusplus.com/reference/vector/vector/>
- [5] [https://en.wikipedia.org/wiki/Hill\\_climbing](https://en.wikipedia.org/wiki/Hill_climbing)
- [6] <https://www.mathworks.com/help/gads/what-is-simulated-annealing.html>
- [7] [https://en.wikipedia.org/wiki/Simulated\\_annealing](https://en.wikipedia.org/wiki/Simulated_annealing)<https://link.springer.com/chapter/10.1007>
- [8] <https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence>