

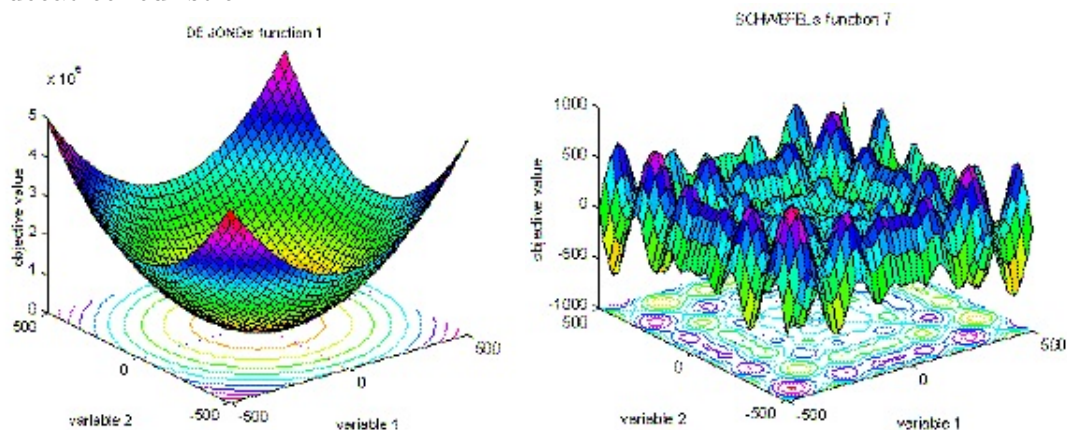
# Tema 2' - Optimizare algoritmi genetici si comparare cu rezultatele obtinute la Simulated Annealing, Hill Climbing și algoritmi genetici neoptimizat pentru aflarea minimului funcțiilor DeJong, Schwefel, Rastrigin și Michalewicz

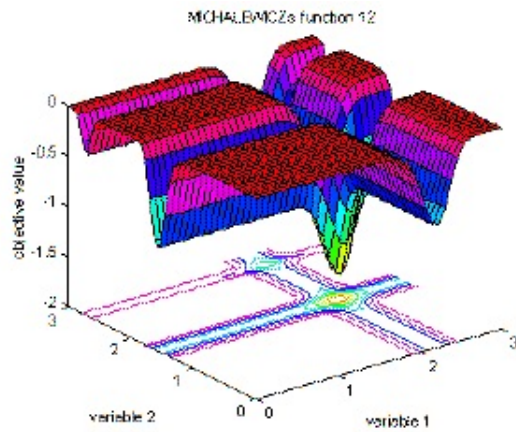
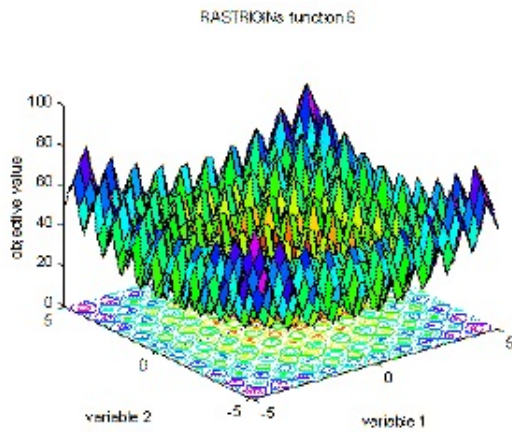
Renghiuc Bianca Elena si Culbece Rose-Marie 2A2

22.11.2022

## Abstract

Raportul nostru conține rezultate, comparații, concluzii și grafice care scot în evidență eficiența algoritmilor genetici. Problema presupune găsirea minimului global al unor funcții cu dimensiuni diferite. Scopul este observarea rezultatelor pentru imputuri mai mari sau mai mici. Am testat algoritmul folosind următoarele patru funcții: **De-Jong's function**, **Schwefel's function**, **Rastrigin's function** si **Michalewicz's function**. Vom compara rezultatele cu cele obținute anterior, la algoritmii Simulated Annealing, Hill Climbing și la algoritmul genetic neoptimizat și vom observa că în anumite situații algoritmi genetici, în special varianta optimizată, dau rezultate mai bune decât cei euristici.





## 1 Introducere

Algoritmii genetici sunt algoritmi evolutivi pentru căutare și optimizare care copiază operațiile genetice din organismul uman. Aceștia conțin o secvență de operații, printre care se numără: Selecția, Incrucișarea și Mutația.

Un algoritm genetic simulează evoluția printr-o succesiune de generații ale unei populații. O soluție candidat este reprezentată ca un șir de gene (șir de biți). Populația menținută de algoritmul genetic evoluează prin aplicarea operatorilor genetici menționați anterior, soluția fiind îmbunătățită pas cu pas de-a lungul mai multor generații. Astfel, folosind algoritmii genetici, vom determina valoarea minimă a funcțiilor prezentate anterior.

## 2 Metode

Am folosit un algoritm genetic, ce pleacă de la ideea de a avea inițial o populație random cu 140 indivizi, ce evoluează pe parcurs. Astfel, putem selecta un număr random de candidați și să îi multiplicăm doi câte doi, creând doi copii care sunt adăuți la populație.

La fiecare generație, candidații sunt aleși astfel încât cei mai buni dintre ei să aibă șansă mai mare de a fi selectați. Selecția am făcut-o prin metoda Selecției Turneu.

Un membru al populației este reprezentat printr-o structură ce are următoarele date: cromozomul său (generat random, și codificat cu ajutorul codului Gray), valoarea fitness-ului și o probabilitate random. Cromozomul este reprezentat printr-un șir de biți ce conține coordonatele punctului curent. Fitness-ul fiecărui candidat este calculat în modul următor:  $P(i).fitness = 1/(-min(P) * ok + f(P(i).gene) + 100)$ , unde  $ok$  este 1 dacă  $min(P)$  este negativ și 0 altfel.  $min(P)$  reprezintă minimul curent,  $P(i)$  populația, iar  $f(gene)$  este valoarea funcției căreia trebuie să îi calculăm minimul. Astfel, fitness-ul va avea mereu o valoare pozitivă. Condiția de oprire este atunci când vom avea 1400 de generații.

Probabilitatea de mutație folosită este 0.009, iar cea de crossover de 0.9.

Mutația se realizează în modul următor: pentru fiecare genă a fiecărui membru al

populației, generăm o probabilitate, dacă aceasta e mai mică decât probabilitatea de mutație (0.009), atunci vom modifica gena de pe poziția respectivă în urmatorul mod:  $gena = gena^{(ok < i)}$ , unde  $ok=1$  dacă probabilitatea generată anterior e mai mică decât probabilitatea de mutație și 0 altfel, iar  $i$  este poziția genei curente.

Încrucișarea se realizează astfel: ordonăm membrii populației după probabilitate. Între cei care au probabilitatea mai mică decât probabilitatea de crossover (0.9) se va realiza o încrucișare (pe o poziție generată random).

### 3 Rezultate

Mai jos am realizat un tabel pentru fiecare dintre dimensiunile: 5, 10, 30. Fiecare celulă conține următoarele valori în această ordine: valoarea minimă, timpul de execuție și media valorilor. Precizia valorilor este egală cu 5, iar algoritmul a folosit un număr de 30 de iterații.

Algorithm Result (2)				
functie	Aloritm genetici optimizati	Medie	Timp	Algoritmi genetici
De Jong	0.006	0.32	12s	0.00018
Schwefel	-837.961	-837.715	14s	-837.913
Rastrigin	0	4.869	18s	0.002
Michalewicz	-0.97	-0.10	14s	-0.80

Algorithm Result (5)				
functie	Aloritm genetici optimizati	Medie	Timp	Algoritmi genetici
De Jong	0.099	3.85	26s	0.069
Schwefel	-2094.73	-2089.55	31s	-2094.77
Rastrigin	0.108	5.346	26s	0.07
Michalewicz	-3.64	-1.7581	24s	-3.61

Algorithm Result (5)				
functie	HCFI	HCBI	HCWI	SA
De Jong	$1.13687e - 10$	$1.13687e - 10$	0.21914	$1.15801e - 08$
Schwefel	-2094.91	-2094.91	-1913.34	-2094.91
Rastrigin	$4.51065e - 09$	$4.51065e - 09$	10.5475	$4.51065e - 09$
Michalewicz	-3.69888	-3.69888	-3.2954	-3.69888

Algorithm Result (10)				
functie	Algoritmi genetici optimizati	Medie	Timp	Algoritmi genetici
De Jong	0.1091	8.2716	42s	0.2952
Schwefel	-4189.29	-4171.05	53s	-4188.34
Rastrigin	2.99	17.324	40s	2.61
Michalewicz	-7.81358	-4.6742	48s	-7.9337

Algorithm Result (10)				
functie	HCFI	HCBI	HCWI	SA
De Jong	$2.27374e - 10$	$2.27374e - 10$	3.99246	$1.33266e - 07$
Schwefel	-4130.74	-4189.51	-3630.58	-4160.03
Rastrigin	1.99004	0.995017	32.247	3.23129
Michalewicz	-8.61317	-8.6432	-5.06779	-8.59603

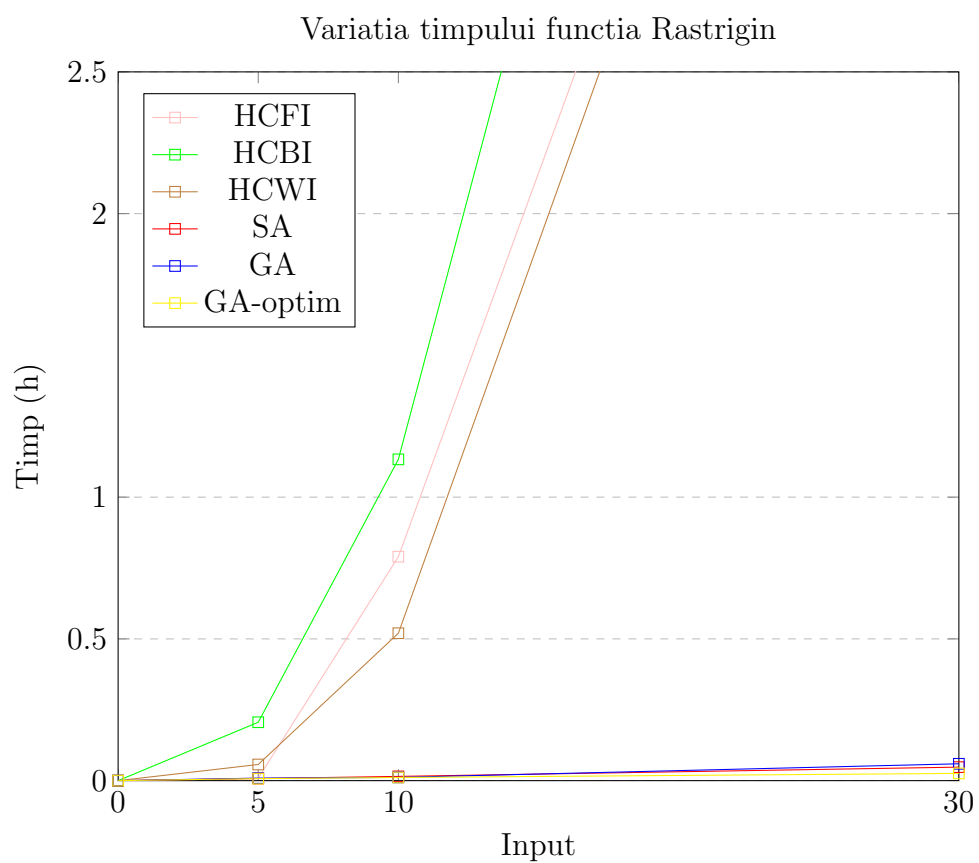
Algorithm Result (30)				
functie	Algoritmi genetici optimizati	Medie	Timp	Algoritmi genetici
De Jong	1.7276	9.253	1m45s	1.6392
Schwefel	-12563.8	-12384.3	2m15s	-12560.6
Rastrigin	23.064	31.2746	1m32s	33.452
Michalewicz	-25.943	-23.7673	1m47s	-25.8139

Algorithm Result (30)				
functie	HCFI	HCBI	HCWI	SA
De Jong	$6.82121e - 10$	$6.82121e - 10$	51.3112	0.00135782
Schwefel	-11122.1	-11807.9	-7148.69	-11656.1
Rastrigin	33.2081	23.0617	84.493	31.296
Michalewicz	-25.8959	-26.4804	-8.46187	-24.9721

## 4 Comparatii

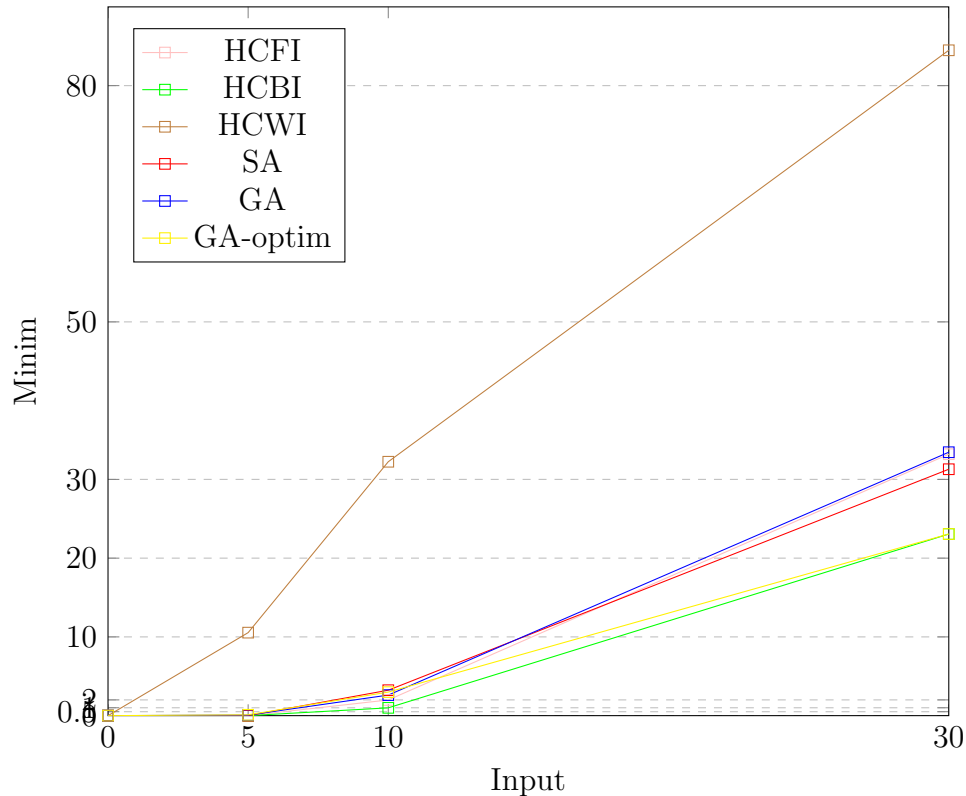
În continuare vom vedea cum variază valoarea minimului aflat pentru funcția Rastrigin, dar și a timpului de execuție, în funcție de dimensiunile inputului.

În acest grafic se poate observa variația timpului de execuție pe măsură ce valoarea inputului (numarul de componente) crește pentru funcția Rastrigin



În acest grafic se poate observa variația minimului obținut pe măsură ce valoarea inputului (numarul de componente) crește pentru funcția Rastrigin

Variatia minimului functia Rastrigin



Pentru funcția Rastrigin, se poate observa că timpul de execuție al algoritmului genetic (atât optimizat, cât și neoptimizat) este aproximativ la fel de bun ca cel al algoritmului SA. În schimb, Best Improvement a reușit să găsească o valoare mai mică pentru minimul global.

Vom compara rezultatele obținute la algoritmul genetic optimizat cu cele obținute la algoritmul genetic neoptimizat pentru fiecare funcție:

De Jong: Pentru fiecare din cele 4 dimensiuni (2, 5, 10, 30) varianta neoptimizată pare să dea rezultate puțin mai bune decât cea optimizată.

Schwefel: Obținem valori mai bune la varianta optimizată decât la cea neoptimizată.

Rastrigin: Pe această funcție se observă cel mai bine eficiența variantei optimizate a algoritmului genetic, dând rezultate mult mai bune decât varianta neoptimizată (și la fel de bune la Hill Climbing - Best Improvement, dar într-un timp mult mai bun) Michalewicz: Obținem valori puțin mai bune la varianta optimizată decât la cea neoptimizată.

## 5 Concluzie

În concluzie, algoritmi genetici dau rezultate destul de bune într-un timp foarte scurt, ceea ce ne ajută în rezolvarea unor probleme, printre care și cea a găsirii minimului unei funcții. Testând algoritmul (atât optimizat, cât și neoptimizat) pe cele 4 funcții

menționate la început am putut observa cum variază gasirea optimului, în funcție de numărul de minime locale.

## References

- [1] <https://profs.info.uaic.ro/~eugennc/teaching/ga/>
- [2] [http://www.geatbx.com/docu/fcnindex-01.html#P150\\_6749](http://www.geatbx.com/docu/fcnindex-01.html#P150_6749)
- [3] [http://www.geatbx.com/docu/fcnindex-01.html#P140\\_6155](http://www.geatbx.com/docu/fcnindex-01.html#P140_6155)
- [4] [http://www.geatbx.com/docu/fcnindex-01.html#P204\\_10395](http://www.geatbx.com/docu/fcnindex-01.html#P204_10395)
- [5] [https://www.youtube.com/watch?v=InVJWW\\_NzFY&list=LL&index=10&t=240s](https://www.youtube.com/watch?v=InVJWW_NzFY&list=LL&index=10&t=240s)
- [6] [http://www.geatbx.com/docu/fcnindex-01.html#P89\\_3085](http://www.geatbx.com/docu/fcnindex-01.html#P89_3085)
- [7] <http://www.optiwater.com/optiga/ga.html?fbclid=IwAR0hjKRmp0jFvSetqI4Wz-jpNtSrxjLGZaP4wjpfANwguvmAcbYl7ND93GE>
- [8] <https://www.infoarena.ro/coduri-gray>