

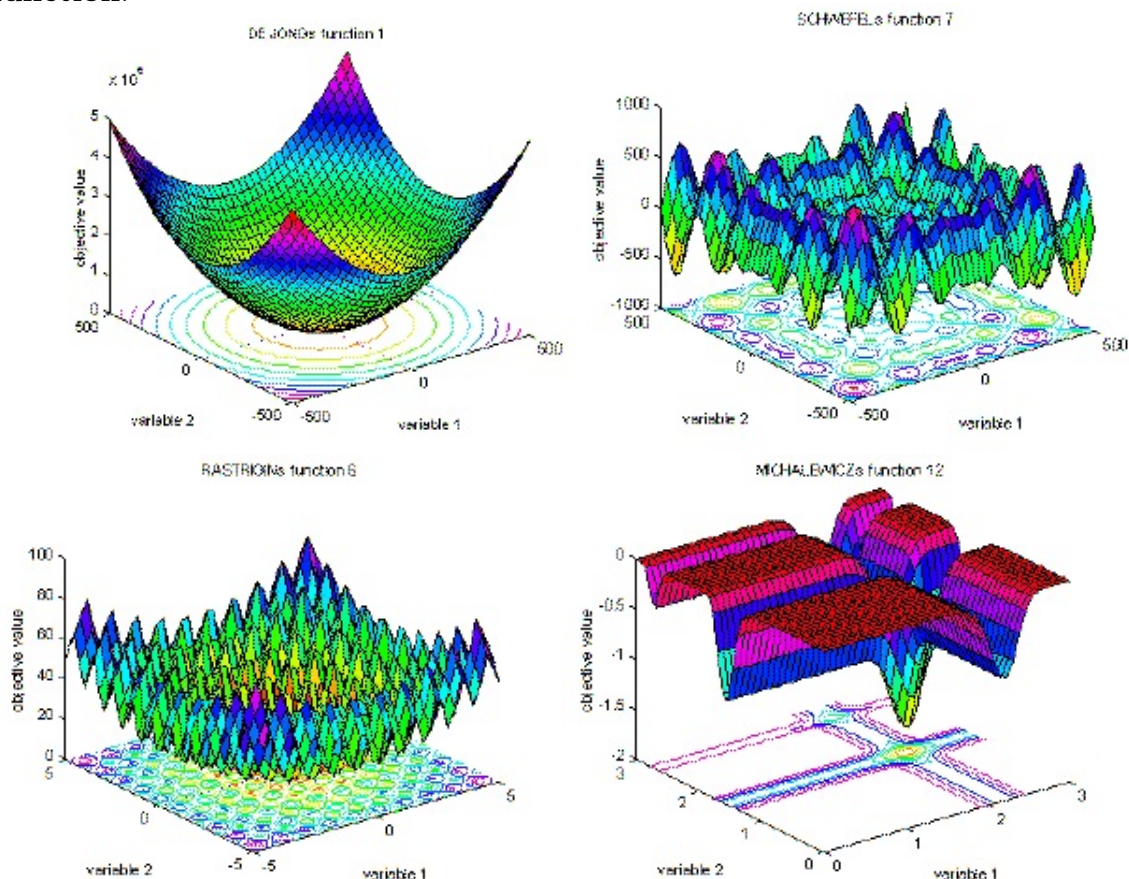
Tema 2 - Implementare algoritmi genetici pentru aflarea minimului unei functii

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**.



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 Mutarea.

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.

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, 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.

3 Rezultate

Mai jos am realizat un tabel pentru fiecare dintre dimensiunile: 5, 10, 30. Fiecare celulă conține următoarele valori în această ordine: valoarea minima, 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	Rezultate Algoritm genetici	Medie	Timp
De Jong	0.00018	0.21	22s
Schwefel	-837.913	-837.522	16s
Rastrigin	0.002	1.699	15s
Michalewicz	-0.80	-0.67	14s

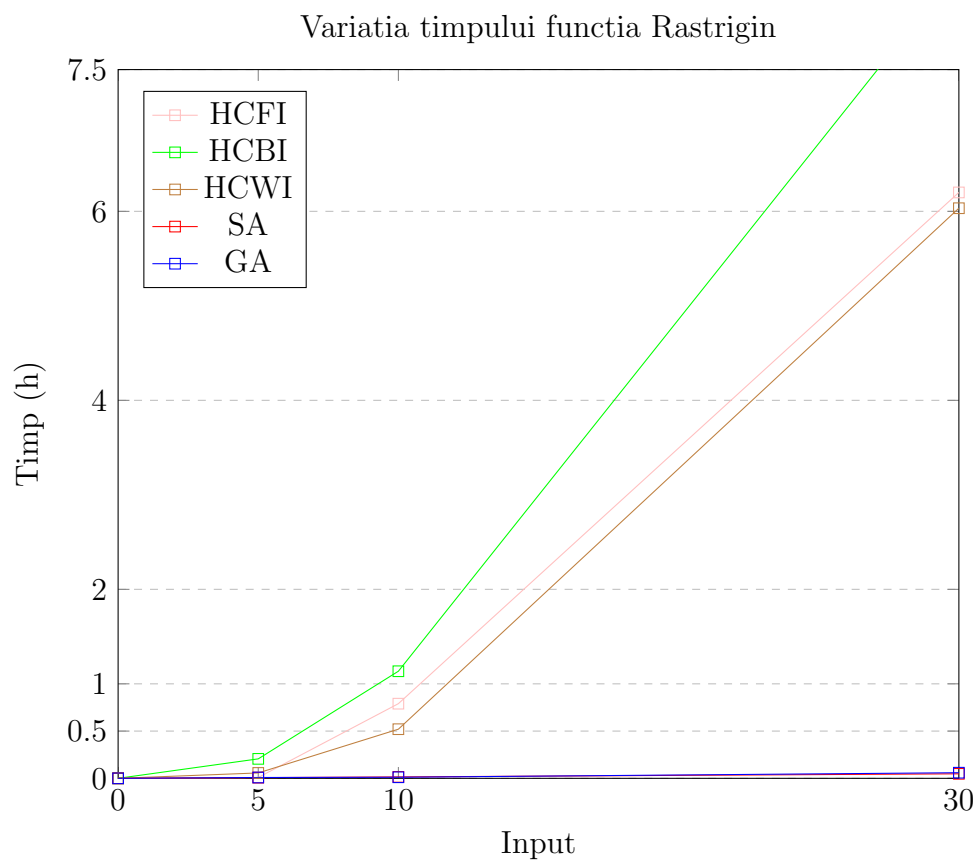
Algorithm Result (5)			
functie	Rezultate Algoritm genetici	Medie	Timp
De Jong	0.069	2.35	27s
Schwefel	-2094.77	-2081.83	32s
Rastrigin	0.07	4.866	26s
Michalewicz	-3.61	-3.0226	25s

Algorithm Result (10)			
functie	Rezultate Algoritm genetici	Medie	Timp
De Jong	0.2952	7.092	44s
Schwefel	-4188.34	-4172.18	55s
Rastrigin	2.61	14.91	41s
Michalewicz	-7.9337	-6.562	48s

Algorithm Result (30)			
functie	Rezultate Algoritm genetici	Medie	Timp
De Jong	1.6392	11.2269	2m22s
Schwefel	-12560.6	-12363	2m46s
Rastrigin	33.452	58.6656	3m34s
Michalewicz	-25.8139	-23.7673	2m52s

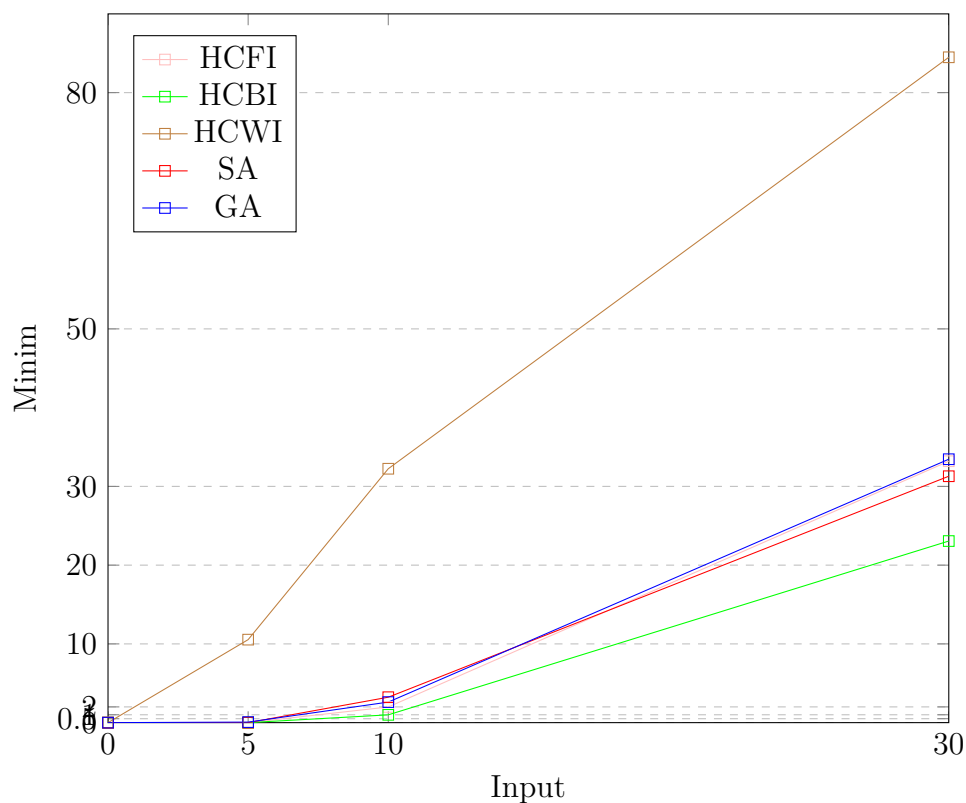
În continuare vom vedea cum variază valoarea minimului aflat pentru funcția Rastrigin, dar și a timpului de execuție, în funcție de dimensiunile inputului.

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 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.

4 Comparatii

Mai departe vom compara valorile minime obținute de algoritm, prin modificarea probabilității de mutație și a celei de crossover.

Rastrigin (30)			
Mutație	Crossover	Rezultate Algoritm genetici	Medie
0.009	0.9	33.45	58.665
0.001	0.2	42.7588	53.9822
0.009	0.6	29.42	63.91
0.001	0.7	38.6076	67.9042
0.1	0.9	315.81	363.036

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 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
- [3] http://www.geatbx.com/docu/fcnindex-01.html#P140_6155
- [4] 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
- [6] http://www.geatbx.com/docu/fcnindex-01.html#P89_3085
- [7] <http://www.optiwater.com/optiga/ga.html?fbclid=IwAR0hjKRmp0jFvSetqI4Wz-jpNtSrxjLGZaP4wjpfANwguvmAcbYl7ND93GE>