Algoritmi Genetici: Tema H_0

Adia Ioana Romanescu, George Grumăzescu

11 octombrie 2023

1 Abstract

Această lucrare urmărește utilizarea unor metode clasice pentru determinarea unor valori cât mai apropiate de minimul și maximul funcțiilor Rastrigin, Michalewicz, Sphere și Griewank. În acest sens, prin două metode, una euristică și una deterministică, lucrarea analizează diferențele generate atât din punct de vedere al punctelor extreme determinate, cât și din punct de vedere al timpului de execuție.

2 Introducere

Minimizarea unei funcții cu un număr arbitrar de variabile este o problemă centrală în domeniul optimizării matematice și a găsirii soluțiilor optime în diverse domenii științifice și tehnice. Această problemă impune identificarea punctelor în care funcția atinge valori minime. Acest raport are ca obiectiv principal explorarea a două metode, una deterministă și una euristică, pentru minimizarea funcțiilor cu un număr arbitrar de variabile. Încă de peste câteva decenii, odată cu dezvoltarea tehnologiei și, implicit, a resurselor de calcul, s-a încercat o altă abordare a problemelor pentru care nu s-a găsit o soluție deterministă, care să fie fezabilă. ”Metodele clasice, dezvoltate pentru utilizare umană, pun adesea accent pe minimizarea operațiilor aritmetice, având ca efect creșterea sofisticării logice, un obiectiv care poate să nu fie dorit atunci când se utilizează un calculator.” (”Direct Search” Solution of Numerical and Statistical Problems, Robert Hooke, T. A. Jeeves (01-5-1961)). Din acest motiv, am optat să urmăm două direcții de cercetare cu acești algoritmi fundamental diferiți.

Un algoritm determinist se bazează pe principiul că, pentru un anumit set de date de intrare, va produce întotdeauna același rezultat de ieșire. Aceasta se datorează faptului că stările prin care trece algoritmul sunt întotdeauna constante pentru un anumit set de date de intrare, asigurând astfel predictibilitatea rezultatului. Pe de altă parte, un algoritm euristic se bazează pe reguli empirice și experiență pentru a selecta o cale de execuție în funcție de cât de aproape este de rezultatul dorit. Euristicile reprezintă strategii care nu oferă o garanție pentru o soluție optimă, ci încearcă să furnizeze o soluție satisfăcătoare, cu o anumită toleranță de eroare acceptată, într-un interval de timp rezonabil. Acest tip de abordare este frecvent utilizat în probleme complexe sau în situații în care spațiul de căutare este extins, și obținerea unei soluții optime poate fi impracticabilă sau prea costisitoare în ceea ce privește resursele necesare.

3 Metode

Prin problemă analizăm următoarele funcții: Rastrigin, Michalewicz, Sphere și Griewank

.

1. Funcția Rastrigin

$$f(X) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right]$$
$$\text{unde } i = 1, 2, \cdots, n$$

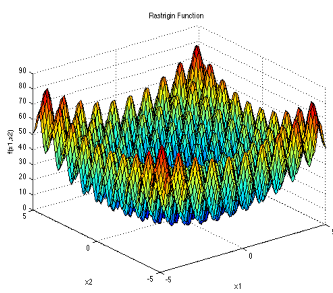

Figure 1: Funcția Rastrigin

2. Funcția Michalewicz

$$f(X) = -\sum_{j=1}^{n} \sin(x_j) \left[\sin\left(\frac{j x_j^2}{\pi} \right) \right]^{2m} \quad , \quad m = 10$$
unde $0 \leq x_i \leq pi$

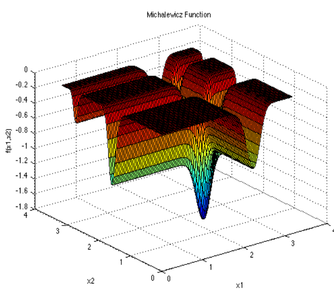

Figure 2: Funcția Michalewicz

3. Funcția Sphere

$$f(X) = \sum_{i=1}^{n} x_i^2$$
unde $-5.12 \leq x_i \leq 5.12$

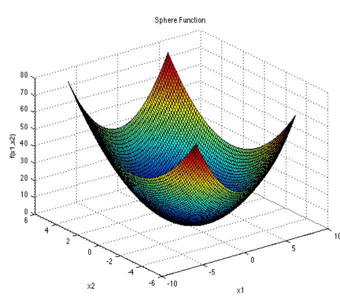

Figure 3: Funcția Sphere

4. Funcția Griewank

$$f(X) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

unde $-600 \leq x_i \leq 600$

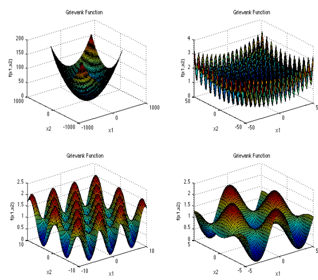

Figure 4: Funcția Griewank

Pentru a aborda problema discutată, am implementat două metode distincte: una deterministă și una euristică. Am ales explorarea ambelor metode pentru a evidenția diferențele dintre ele, precum și avantajele și dezavantajele fiecăreia. În esență, un algoritm determinist implică un număr mare de iterații și constrângeri, necesitând astfel resurse considerabile și, implicit, timp de execuție extins. Pe de altă parte, algoritmul euristic nu analizează toate stările posibile ale problemei, rezultând într-o soluție care nu este exactă, dar cu un timp de execuție semnificativ mai scurt. Acest aspect se traduce printr-un compromis între precizie și rezolvarea problemei într-un interval de timp rezonabil. Algoritmul clasic determinist pentru determinarea punctului de minim global (pe întregul domeniu al funcției) presupune o iterație uniformă a domeniului, începând de la un capăt și ajungând la celălalt capăt. Dat fiind că într-un interval fixat $[a, b]$ există un număr infinit de numere, se utilizează un pas foarte mic (de ordinul lui $c \cdot 10^{-5}$, unde c este de obicei o cifră), care generează, în general, o valoare foarte apropiată de cea căutată, diferită printr-un epsilon (o valoare foarte mică și nesemnificativă). Cu toate acestea, această soluție nu este practicabilă. Să presupunem că folosim un pas egal cu 1 pentru a itera prin valorile posibile ale variabilelor. Astfel, pentru un interval $[a, b]$, avem $b - a + 1$ posibilități. Un pas egal cu 1 nu acoperă nici pe departe toate valorile din domeniu, motiv pentru care trebuie să alegem un pas mai mic. Cu cât deplasăm punctul mai mult către stanga, numărul de posibilități devine de 10^d ori mai mic (unde d reprezintă numărul de deplasări).

Având nevoie de un pas foarte mic pentru a asigura veridicitatea soluției, de exemplu 10^{-5}, o singură iterare prin interval presupune$(b-a+1)\cdot 10^5$ posibilități.
Pentru versiunea unidimensională și bidimensională a funcției, o soluție de acest gen poate fi viabilă (în funcție și de lungimea domeniului). Totuși, odată ce trecem la varianta tridimensională, numărul de posibilități devine$((b-a+1)\cdot 10^5)^3$, mai precis $(b-a+1)^3\cdot 10^{15}$, ceea ce face acest algoritm impracticabil pentru domenii extinse.
Pe un sistem de calcul octa-core echipat cu un procesor AMD RYZEN 7 5000 și 16GB de RAM, s-a constatat că aproximativ 10^9 operații elementare pot fi executate în circa 17 secunde. Prin urmare, se estimează că 10^{15} operații ar dura aproximativ $17\cdot 10^6$ secunde. Cum 10^6 secunde inseamna aproximativ 11 zile, rezultă astfel un total de $17\cdot 11 = 187$ de zile. În cazul unui domeniu suficient de restrâns, o astfel de soluție ar putea rămâne fezabilă, în funcție și de cerințele impuse de problemă. Cu toate acestea, pe măsură ce domeniul devine mai extins, timpul necesar devine din ce în ce mai mare și nu mai este acceptabil.
Algoritmul euristic se bazează pe generarea cât mai multor puncte de minim local, încercând să se apropie cât mai mult de punctul de minim global. Mai precis, euristica pe care am aplicat-o a presupus selectarea repetată a unor valori arbitrar alese din domeniul funcției pentru variabilele respective. Pentru fiecare alegere, am încercat să determinăm cea mai bună opțiune pe care o poate lua fiecare variabilă pentru a minimiza rezultatul.
Am implementat acest proces luând în considerare cele două intervale în variabila împarte domeniul: $[a, x_i]$ și $[x_i, b]$, unde $i = 1, n$. Am căutat un punct de minim local în fiecare interval, începând cu potentialul punct de minim local ca fiind x_i.
Pentru fiecare interval, punctul de minim local a fost calculat tot pe baza unei euristici. Astfel, s-a generat o valoare aleatorie pentru x_i până la găsirea uneia care produce o valoare a funcției mai mică decât cea găsită până atunci. Dupa aceea, ne-am uitat la valorile din vecinătatea acelui punct pentru a vedea evoluția functiei în acea zonă. În cazul în care urmează o direcție descrescătoare, am setat acel capăt ca fiind noul capăt stâng. În celălalt caz, când urmează o direcție crescătoare, am setat acel capăt ca fiind noul capăt drept. Am repetat acești pași fie până când cele 2 capete devin egale, fie până când diferența dintre valoarea minimă de până atunci și valoarea produsă de valoarea generata pentru x_i diferă printr-un epsilon.
Totuși, în cazul unui domeniu suficient de mare, cum este în cazul funcției Griewank, probabilitatea de a alege aleatoriu valori pentru x_i care să producă o valoare mai mică decat cea anterioara scade semnificativ. Astfel, în cazul în care am găsit o valoare mai mare, pentru intervalul stâng sau drept, setăm capătul stâng, respectiv drept ca fiind acea variabila x_i generată, pentru care s-a atins valoarea respectivă.

În final, după ce am ales de un număr suficient de ori fiecare set de variabile inițiale, care au generat fiecare în parte o anumită valoare, în urma algoritmului descris, am ales minimul dintre acestea.

4 Descriere experimentale

Dimensiunile pe care le-am folosit pentru derularea experimentelor pe funcțiile Rastrigin, Michalewicz, Sphere si Griewank au fost cea 2-dimensională, cea 5-dimensională si cea 10-dimensională. Atât valorile generate, cât și reprezentarea rezultatelor au o precizie de 5 zecimale după 0.
Am utilizat trei dimensiuni pentru experimentele efectuate asupra funcțiilor Rastrigin, Michalewicz, Sphere și Griewank: unul bidimensional, unul pentadimensional și unul decadimensional. Atât valorile generate, cât și reprezentarea rezultatelor, au fost exprimate cu o precizie de 5 zecimale.

5 Rezultate experimentale

5.1 Rezultatele experimentelor pe funcția Rastrigin

D	minim	medie	maxim	timp minim	timp mediu	timp maxim	stdev
2	0.00000	0.09954	0.99497	0.04401	2.07818	12.98760	0.29847
5	0.00006	2.62018	5.97009	1.51635	6.13615	11.40911	1.57280
10	1.99024	9.78388	18.90430	12.63785	23.19017	44.35022	3.67891

Figure 5: Tabel cu rezultatele funcției Rastrigin pe 30 de teste realizate pe metoda euristică prezentată - unde am notat D reprezintă dimensiunea funcției și stdev deviația standard

5.2 Rezultatele experimentelor pe funcția Michalewicz

D	minim	medie	maxim	timp minim	timp mediu	timp maxim	stdev
2	-1.80130	-1.80130	-1.80129	1.28329	7.55200	20.17412	0.00000
5	-4.68765	-4.56038	-4.02718	24.77260	64.59962	116.73243	0.16482
10	-9.53959	-8.71291	-7.67389	54.02853	152.93496	274.26584	0.44299

Figure 6: Tabel cu rezultatele funcției Michalewicz pe 30 de teste realizate pe metoda euristică prezentată- unde am notat D reprezintă dimensiunea funcției șistdev deviația standard

5.3 Rezultatele experimentelor pe funcția Sphere

D	minim	medie	maxim	timp minim	timp mediu	timp maxim	stdev
2	0.00000	0.00000	0.00001	4.78157	9.14328	14.12699	0.00000
5	0.00000	0.00001	0.00003	13.78811	21.77792	29.40772	0.00001
10	0.00002	0.00004	0.00009	45.45941	56.74467	78.66980	0.00002

Figure 7: Tabel cu rezultatele funcției Sphere pe 30 de teste realizate pe metoda euristică prezentată - unde am notat D reprezintă dimensiunea funcției și stdev deviația standard

5.4 Rezultatele experimentelor pe funcția Griewank

D	minim	medie	maxim	timp minim	timp mediu	timp maxim	stdev
2	0.00000	0.00312	0.00795	1.38708	1.40213	1.43532	0.00329
5	0.07875	0.31443	0.64229	4.84410	4.91131	5.29821	0.13894
10	0.96732	1.17600	1.56256	14.61671	14.83228	15.04238	0.15650

Figure 8: Tabel cu rezultatele funcției Griewank pe 30 de teste realizate pe metoda euristică prezentată - unde am notat D reprezintă dimensiunea funcției și stdev deviația standard

6 Comparații și discuție între metodele implementate

S-a constatat că algortimul euristic funcționează destul de bine pe teste, cu o marjă de eroare acceptată, în timp ce algoritmul determinist nu reușește să producă niște rezultate cu o acuratețe la fel de bună, întrucât subdomeniul pe care generăm soluții devine limitat pentru a avea un timp de rulare mediu cât mai mic. În testele luate în considerare, am urmărit rezultatele a aproximativ 30 de teste, pentru care algoritmul deterministic a fost aplicat pe submulțimi ale domeniilor funcțiilor. Astfel, pentru teste am considerat dimensiunea 2 și am analizat rezultatele pe submulțimi cu elemente la distanță de aproximativ 0.005 pentru a face posibilă evaluarea într-un timp mai scurt.

7 Concluzii

În concluzie, atât algoritmul deterministic cât și cel euristic reprezintă soluții viabile pentru problema determinării de maxime și minime pentru funcțiile Rastrigin, Michalewicz, Sphere și Griewank. De altfel, am diferențiat necesitatea soluțiilor prin timpul mediu mult mai scăzut pentru soluția euristică și prin diferența de acuratețe dintre soluții pentru un același număr de valori. Pe viitor, pentru a optimiza aceste soluții în vederea găsirii unor rezultate mai bune, am dori:

- Să modificăm soluția euristică prin algoritmul Hill Climbing astfel încât să obținem o precizie mai bună în determinarea minimului sau a maximului

8 Bibliografie

(a) https://profs.info.uaic.ro/ eugennc/teaching/ga/

(b) https://www.sfu.ca/ ssurjano/rastr.html

(c) https://www.sfu.ca/ ssurjano/michal.html

(d) https://www.sfu.ca/ ssurjano/spheref.html

(e) https://www.sfu.ca/ ssurjano/griewank.html

(f) https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-heuristics/

(g) https://en.wikipedia.org/wiki/Deterministic_algorithm

(h) https://typeset.io/papers/direct-search-solution-of-numerical-and-statistical-problems-2qi30kmlxk