

Raport Tema1

Eduard-Mihail Hamza

1 decembrie 2020

1 Introducere

În analiza matematică, maximul și minimul unei funcții sunt cele mai mari, respectiv cele mai mici valori, pe care funcția le poate lua, fie pe un anumit interval (caz în care poartă denumirea de maxim/minim local) sau pe întreg domeniul de definiție al funcției (caz în care se numește maxim/minim global).

În acest raport se va realiza o comparație între un **algoritm genetic de optimizare** și metodele **Hill Climbing** și **Simulated Annealing** și se vor prezenta câteva date experimentale ce caracterizează execuția acestor algoritmi de aflare a minimului global al unei funcții cu număr variabil de parametri.

1.1 Motivație

Problemele de optimizare a unei funcții presupun găsirea minimului sau maximumului unei funcții pe un interval. Astfel de probleme se abordează de obicei, atunci când instrumentele matematice clasice nu mai pot oferi soluții, cu ajutorul algoritmilor genetici. Una dintre cele mai mari provocări ale acestor algoritmi de optimizare este găsirea minimului/global fără a rămâne blocați în minime/maxime locale.

Acest raport își propune să observe care dintre cele trei metode amintite mai sus are rezultate mai bune în funcție de fiecare funcție de test în parte și în funcție de dimensiunea fiecărei funcții.

2 Metode

Implementarea **algoritmului genetic** s-a realizat după o schemă clasică în care se generează o populație inițială care apoi evoluează de lungul mai multor generații (prin crossovere și mutații) sub controlul unei funcții fitness care măsoară meritul individual.

Mai jos sunt amintite câteva detalii de implementare:

- Număr generații: 1000
- Probabilitate mutație: 0.001

- Probabilitate crossover: 0.4
- Monstră statistică: 30

În ceea ce privește metoda de selecție a fost folosită **roata norocului**, metodă prin care numărul estimat de copii pe care îl primește un individ este proporțional cu fitness-ul său împărțit la fitness-ul total al populației. Mai jos este prezentat codul funcției fitness folosită:

Celelalte 2 metode folosite au fost **Hill Climbing** și **Simulated Annealing**: **Hill Climbing**, o metodă iterativă ce realizează o căutare locală. Este utilizată varianta iterată (Iterated Hill Climbing) în care HC este restartat, pentru a mări gradul de explorare a spațiului de căutare.

De asemenea, se vor folosi 2 tipuri de HC: **Best Improvement Hill Climbing** și **First Improvement Hill Climbing**

Best Improvement Hill Climbing examinează fiecare vecin și îl alege pe cel care determină cea mai bună soluție.

First Improvement Hill Climbing nu examinează fiecare vecin înainte de a hotărî pe care îl alege. Pur și simplu alege un vecin la întâmplare până când găsește unul mai promițător decât cel curent.

Simulated Annealing, o meta-euristică de tip traiectorie care permite o mai bună explorare a spațiului și ieșirea din puncte de optim local, dând posibilitatea vizitării unor soluții de calitate mai slabă decât cea curentă.

Pentru funcția de modificare a temperaturii s-a folosit înmulțirea cu un număr subunitar (0.9), aceasta fiind inițializată la începutul algoritmului cu valoarea 1000.

În cazul tuturor metodelor pentru **reprezentarea soluțiilor** s-au folosit șiruri de biti, iar **precizia** utilizată a fost 10^{-2} (2 zecimale).

3 Experiment

Pentru testarea algoritmilor vom folosi următoarele funcții:

1. Funcția lui De Jong 1

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

Minim global: 0

2. Funcția lui Schwefel

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

Minim global: $-n \cdot 418.9829$

3. Functia lui Rastrigin

$$f(x) = A \cdot n + \sum_{i=1}^n [x_i^2 - A \cdot \cos(2\pi x_i)], A = 10, x_i \in [-5.12, 5.15]$$

Minim global: 0

4. Functia lui Michalewicz

$$f(x) = - \sum_{i=1}^n \sin(x_i) \cdot \left(\sin \left(\frac{i \cdot x_i^2}{\pi} \right) \right)^{2m}, m = 10, x_i \in [0, \pi]$$

Minim global: -4.687 (n=5), -9.66 (n=10)

4 Rezultate

teoretic = minimul teoretic al funcției (cel corect)

minim = cel mai bun minim returnat de algoritm

medie = media minimelor obținute

σ = deviație standard

timp = timpul mediu de execuție în secunde

4.1 5 dimensiuni

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 1: Rastrigin 5 dimensiuni - monștră statistică 30

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 2: DeJong1 5 dimensiuni - monștră statistică 30

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 3: Michalewicz 5 dimensiuni - monștră statistică 30

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 4: Schwefel 5 dimensiuni - monștră statistică 30

4.2 10 dimensiuni

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 5: Rastrigin 10 dimensiuni - monștră statistică 30

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 6: DeJong1 10 dimensiuni - monștră statistică 30

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 7: Michalewicz 10 dimensiuni - monștră statistică 30

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 8: Schwefel 10 dimensiuni - monștră statistică 30

4.3 30 dimensiuni

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 9: Rastrigin 30 dimensiuni - monștră statistică 30

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 10: DeJong1 30 dimensiuni - monștră statistică 30

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 11: Michalewicz 30 dimensiuni - monștră statistică 30

Rastrigin (teoretic min=0)				
Algoritm	minim	medie	σ	timp(s)
GA	0	0	0	0
BIHC	0	0	0	0
FIHC	0	0	0	0
SA	0	0	0	0

Figura 12: Schwefel 30 dimensiuni - monștră statistică 30

4.4 Interpretare

4.4.1 5 dimensiuni

Toate cele trei metode produc soluții asemănătoare. Simulated Annealing are cea mai slabă soluție în cazul funcției lui Rastrigin.

În ceea ce privește timpul de execuție HC Best Improvement este cea mai costisitoare, în timp ce HC First Improvement și SA au timpi asemănători. O observație interesantă este că în cazul funcției lui Schwefel timpul de execuție a fost de aproximativ 14 ori mai mare decât în cazul celorlalte 2 metode.

Din punct de vedere al deviației standard, SA are o valoare de aproape 4 ori mai mare în cazul funcției Schwefel.

4.4.2 10 dimensiuni

HC Best Improvement continua să ofere cele mai apropiate soluții de cele reale, însă timpul de execuție este substanțial mai mare (109.499 secunde în cazul funcției Schwefel).

HC First Improvement și SA ofera rezultate identice pentru minimul funcțiilor Rastrigin și Michalewicz (și foarte apropiate pentru DeJong) însă diferă observabil la rezultatul mediu în cazul funcției Rastrigin. Timpii de execuție sunt asemănători, însă deviația standard este mult mai mare în cazul SA la Schwefel (151.24).

4.4.3 30 dimensiuni

Toate cele 3 metode încep să produca rezultate departate față de cele reale. SA este pe primul loc, urmată de HC First Improvement. HC Best Improvement nu poate fi luat în calcul la aceasta categorie datorită monstre statistice foarte mici.

HC First Improvement și SA au în continuare timpii asemanatori de execuție, excepție făcând funcția Schwefel, pentru care HC are un timp dublu. HC Best Improvement are un timp de execuție uriaș, în comparație cu celelalte doua metode. În cazul funcției Schwefel o singura rulare a algoritmului a durat 2942.72 secunde, de aici și alegerea unei monstre statistice foarte mici.

Se observă ca SA are o deviație standard mult mai mare decât HC First Improvement.

5 Concluzii

Pentru dimensiunea 5, toate metodele au returnat o valoare apropiată de cea reală. Deci toate 3 sunt metode bune pentru determinarea minimului.

Pentru dimensiunea 10, HC First Improvement și SA încep să se îndepărteze ușor de rezultatele reale, în timp HC Best Improvement oferă cele mai bune răspunsuri, dar cu prețul unui timp mai mare.

Pentru dimensiunea 30, HC Best Improvement devine inutil de executat datorita timpului uriaș, în schimb SA reușește sa producă cele mai bune rezultate și

cei mai buni timpi. Asadar, SA devine metoda recomandată pentru dimensiuni mari ale funcției.

Bibliografie

- [1] Minim și maxim
https://en.wikipedia.org/wiki/Maxima_and_minima
- [2] Hill Climbing și Simulated Annealing
<https://profs.info.uaic.ro/~eugennc/teaching/ga/>
http://www.lia.deis.unibo.it/~aro/pubs/blum_rol_i_metaheuristics-preprint.pdf
- [3] Intocmirea unui raport
Formularea introducerii.
<https://www.monash.edu/rlo/assignment-samples/engineering/eng-writing-technical-reports/introduction>
- [4] Tipuri de Hill Climbing
<https://www.geeksforgeeks.org/introduction-hill-climbing-artificial-intelligence/>
- [5] Simulatin Annealing
<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/learn-43/lib/photoz/.g/web/glossary/anneal.html>
- [6] De Jong1
http://www.geatbx.com/docu/fcnindex-01.html#P89_3085
- [7] Schwefel
http://www.geatbx.com/docu/fcnindex-01.html#P150_6749
- [8] Rastrigin
http://www.geatbx.com/docu/fcnindex-01.html#P140_6155
- [9] Michalewicz
http://www.geatbx.com/docu/fcnindex-01.html#P204_10395
<http://www.alliot.mobi/papers/gecco2014b.pdf>