Simularea variațiilor Hill Climbing și Simulated Annealing pentru optimizarea funcțiilor test (Sphere/De Jong, Rastrigin, Schwefel, Michalewicz)

Autor: Negură Teodor Alexandru

Rezumat (Abstract)

- Problema Studiata : Gasirea minimului global pe domeniul celor 4 functii matematice
- Metode: Algoritmul Hill Climbing(First Improvement/Best Improvement /Worst Improvement), Simulated Annealing hibridizat cu Best Improvement .
- Setări experimentale esențiale: D∈{5,10,30,100}, N=50 rulari/configuratie Functie-Variatie_Alg-Dimensiune, buget E={50,100,250}*10000*D evaluări, δ definit pentru precizie de 5 zecimale si răcire geometrică in cadrul SA.
- Rezultate cheie
-
- Concluzie si o direcție viitoare.

1. Introducere

1.1 Context & motivație: Optimizare cu peisaje multi-modale, cost evaluare,
1.2 Problema. Minimizare celor 4 functii matematice (f(x)) , pe domeniu ([a,b]^D).
1.3 Contribuții. (a) Protocol experimental echitabil (buget infinit D), (b) analiză parametrică δ, T0, δ, L, (c) hibrid SA→BI și comparație cu HC clasice.
1.4 Structura raportului.

> **Checklist secțiune:** clarifică obiectivul (minimizare), definește "fitness", spune pe ce dimensiuni lucrezi și de ce, precizează criteriul de comparație (buget fix sau până la ϵ).

2. Fundamente & Funcții test

2.1 Notări și termeni.

"**H.C.**" - HillClimbing

"**S.A.**" - Simulated Annealing

"**Fitness**" - valoarea functiei matematice f in punctul x de coorodnate [x1,x2,...,xn].Cu alte cuvinte , fitnesul functiei f , in punctul x cu coordonatele expuse mai devreme si variabile in functie de dimensiunea D = f(x1) + f(x2) + ... + f(xn);

"**Vecin**" - adunarea sau scaderea valorii delta dintr-o singura coordonata a unui punct aleatoriu , indidfrent de lungime.

"**Delta**" = valoare ajutatoare pt calculaul vecinilor , = 0.00001

"Dimensiune"= lungimea sirului de coordonate a unui punct aleatoriu x. avem x[x1,x2,...,xn], in Dimensiunea D , considera n ca fiind egal cu D.

"**Functie Unimodala**" = un singur optim global și niciun alt optim local.

"**Functie Multimodala**" = un optim global, dar și mai mulți optimi locali.

2.2 Domenii & formule:

De Jong1(continua,grafic convex si unimodala):

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \qquad -5.12 \le x_i \le 5.12$$

Schwefel(multimodala):

$$f_7(x) = \sum_{i=1}^{n} -x_i \cdot \sin\left(\sqrt{|x_i|}\right) \qquad -500 \le x_i \le 500$$

Rastrigin(multe minime locale):

$$f_6(x) = 10 \cdot n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i)\right) \qquad -5.12 \le x_i \le 5.12$$

Michalewicz(multimodala si cu cat paramaterul m creste cu atat si dificultatea):

$$f_{12}(x) = -\sum_{i=1}^{n} \sin(x_i) \cdot \left(\sin\left(\frac{i \cdot x_i^2}{\pi}\right)\right)^{2 \cdot m} \qquad i = 1:n, m = 10, 0 \le x_i \le \pi$$

3. Metode

3.1 Reprezentare & vecinătate. Puncte reale calculate astfel: prin modiicarea unei coorodonate $\pm\delta$ ($2 \cdot$ D vecini/iterație). δ este egal cu 0.00001 , pentru o precizie de 5 zecimale.

3.2 Hill Climbing (HC).

First Improvement - calcularea liniara a vecinilor si alegerea primului care are fitnesul strcit mai bun decat punctul curent

Best Improvement - calcularea tuturor vecinilor si alegerea celui cu diferetta dintre fitnessul curent si cel al unui potential vecin cea mai mare

Worst Improvement - calcularea tuturor vecinilor si alegerea celui cu diferetta dintre fitnessul curent si cel al vecinului strict mai mare ca 0 , dar minimul posibil

3.3 Simulated Annealing (SA). temp = temp / (1 + beta * temp) - temperatura nouă e raportul dintre temperatura veche și $(1 + \beta \cdot \text{temp})$

3.4 Hibridizare SA+BI – Dupa rularea alg B.I. salvez minimul local , si aplica algorimtul S.A. din punctul respectiv.

3.5 Pseudocod (exemplu, varianta A)

```
function BuildBestReference(f, D, delta, num_runs):

    best_ref ← +∞

    repeat num_runs times:

        result ← RunBestImprovement(f, D, delta)   // hill climbing
BI

        if result < best_ref:

            best_ref ← result

    return best_ref

function HybridSAwithBI(f, D, delta, initial_temp, beta, best_ref):

    hc ← new HillClimbing(f, D, delta)     // ctor seed-uiește
RNG, punct random

    hc.current_fitness ← best_ref          // pornești cu fitnessul
BI

    T ← initial_temp

    while hc.evaluations < hc.maxEvaluations and T > ϵ:

        candidate ← hc.GenerateRandomPoint()

        candidate_fitness ← f(candidate)

        hc.evaluations++

        if candidate_fitness < hc.current_fitness:

            hc.Accept(candidate, candidate_fitness)     // actualizezi
punct și componente

        else:

            prob ← exp(-(candidate_fitness - hc.current_fitness)
/ T)

            if Uniform(0, 1) < prob:

                hc.Accept(candidate, candidate_fitness)
```

```
        T ← T / (1 + beta * T)          // schema ta de răcire
reciprocă
    return hc.current_fitness
// orchestrare
for each func f ∈ functii, dimensiune D:
    best_ref ← BuildBestReference(f, D, delta, num_runs)
    repeat num_runs times:
        sa_result ← HybridSAwithBI(f, D, delta, initial_temp,
beta, best_ref)
        colectează metricile (best, worst, mean, α, evaluări
medii)
```

3.6 Cost & echitate. Conditia de oprire a fost cate un numar standard constant de vecini verificati, indiferent de dimensiune. Am implementat acest lucru printr-o inmultire care are rolul de a creste numarul de iteratii in functie de parametrii de intrare. Am ales acest lucru pentru ca in Dimensiunile mari, 50/100 , rezultatele erau aprox nesemnificative , deoarece alg se oprea dupa foarte putini vecini verificati.Parametrii functiilor de cautare au fost Functia Matematica si Dimensiunea.

3.7 Implementare.

Limbaj : C++ 17

- Biblioteci:<algorithm>,<cmath>,<future>,<iomanip>,<iostream>,<limits>,<map>,<memory>,<num

seed fixat : Seed-ul e sursa de entropie pentru generatorul pseudo-aleator (mt19937).

Hardware : i7-13650HX si 24 GB DDR5

4. Protocol experimental (0.5–1.5 pagini)

4.1 Configurații. D∈{5,10,30,100}; N=50 rulari/config; precizie output 5 zecimale.
4.2 Domenii & inițializare. Uniform pe domeniul funcției; multi-start.
4.3 Buget & oprire. Buget E per run (50,100,250)*10000 v ecini viiztatati , obtinuti prin formula (50,100,250)*10000*maximum(1,Dimensiune).
4.4 Metrice. Best, Worst, Mean, Sigma , Evaluari medii ,Coeficientul de Variatie și numărul total de Rulari.

4.6 Reproductibilitate. Rezultatele le-am salvat intr-un fisier cu extensia.txt sub forma de pseudo-tabela.

5. Rezultate experimentale

5.1. Performanță finală (buget fix) În toate cazurile, Simulated Annealing obține o valoare medie a funcției obiectiv semnificativ mai bună. Cea Mai Bună Soluție (Best): SA nu doar că este mai bun în medie, dar identifică și soluții individuale (Best) mult superioare. Performanța între Strategiile Hill Climbing: Nu există un câștigător clar între First, Best și Worst Improvement. Performanța lor este foarte similară în majoritatea scenariilor, sugerând că toate trei eșuează în mod similar prin blocarea în optimi locale. De exemplu, pentru DeJong1 (D=30, Buget Mediu), mediile sunt 267.6 (First) , 273.3 (Best) și 263.8 (Worst), valori foarte apropiate.
5.3 Sensibilitate la parametri. Simulated Annealing este mult mai robust. Valorile Sigma pentru SA sunt, în mod constant, cu unul sau mai multe ordine de mărime mai mici decât cele ale strategiilor Hill Climbing.Pentru **Michalewicz (D=100, Buget Mediu)**, SA are o deviație standard de doar 0.88 , în timp ce strategiile Hill Climbing au valori între 8.9 și 13.2. Această discrepanță uriașă indică faptul că strategiile Hill Climbing sunt extrem de sensibile la punctul de pornire aleatoriu și se blochează în optimi locale de calități foarte variate.

5.4 Eficiența și Impactul Bugetului Computațional: Simulated Annealing (Randament Descrescător): Creșterea bugetului de evaluări aduce beneficii minore pentru SA, sugerând că algoritmul converge relativ repede către o soluție de înaltă calitate. Pentru Rastrigin (D=100), trecerea de la bugetul mediu (20M evaluări, Media 1312.37) la cel mare (40M evaluări, Media 1300.56) aduce o îmbunătățire foarte mică. **Hill Climbing (Ineficiență):** Pentru strategiile First, Best și Worst Improvement, creșterea bugetului computațional este ineficientă.Pentru **Rastrigin (D=30)**, strategia Best Improvement are o medie de 559.9 (Buget Mic) , 547.9 (Buget Mediu) și 561.4 (Buget Mare). Nu există nicio corelație între creșterea bugetului și îmbunătățirea soluției

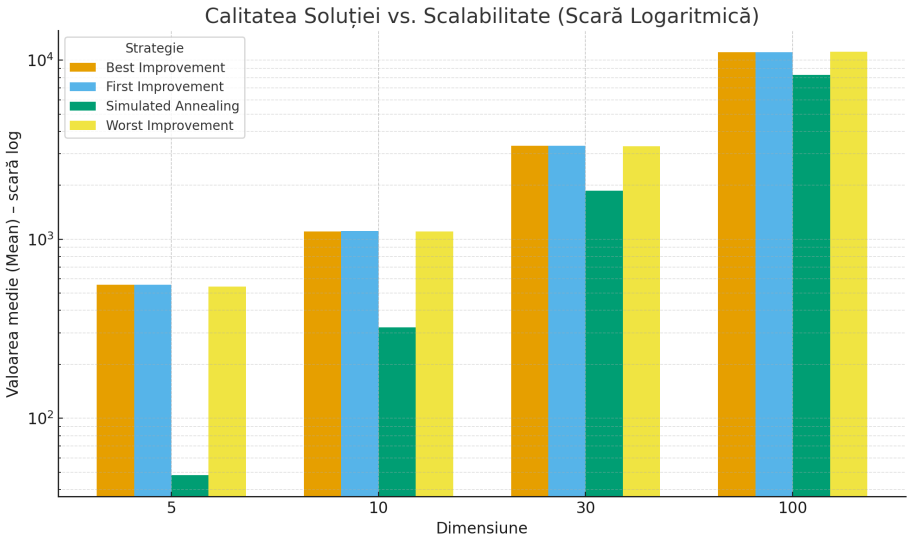

Figure 1: Calitatea Soluției vs. Scalabilitate (Scară Logaritmică)

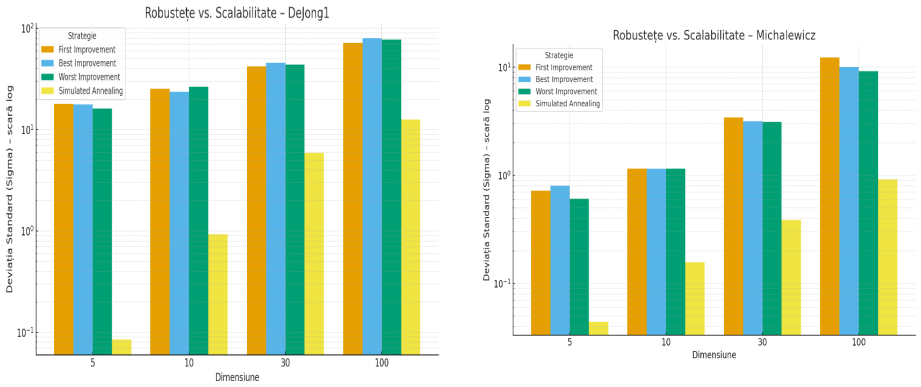

Figure 2: Comparația robusteții (Sigma) pentru DeJong1 (stânga) și Michalewicz (dreapta).

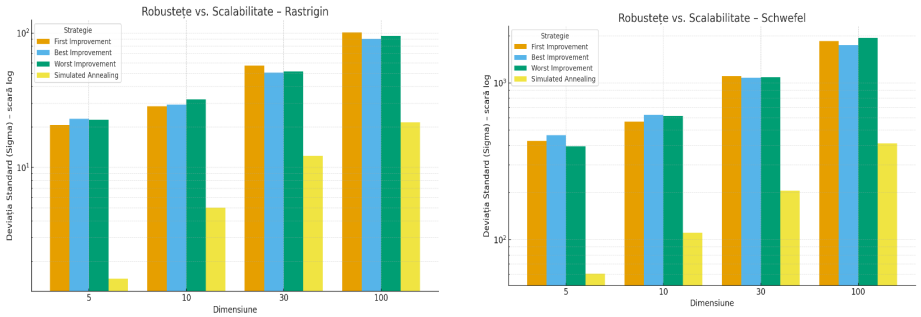

Figure 3: Comparația robusteții (Sigma) pentru Rastrigin (stânga) și Schwefel (dreapta).

Calitatea Soluției vs. Scalabilitate

Impactul Bugetului Computațional

6. Interpretarea diferențelor între metode pe peisaje unimodale vs multimodale.

Strategiile Hill Climbing (First, Best, Worst): Eșuează catastrofal. Fiind strategii "greedy", ele urcă pe primul "deal" (optim local) pe care îl găsesc și rămân blocate acolo. Nu au niciun mecanism de a "coborî" pentru a căuta un deal mai înalt (o soluție mai bună). Acest lucru este evidențiat de valorile medii (Mean) foarte slabe și deviațiile standard (Sigma) uriașe, care arată că se blochează în optimi locale diferite la fiecare rulare. Simulated Annealing (SA): Este superior cu ordine de mărime. Datorită mecanismului său de a accepta soluții mai proaste (bazat pe "temperatură"), SA poate scăpa din aceste optimi locale. Acest lucru îi permite să exploreze peisajul global mult mai eficient și să găsească regiuni de soluții net superioare, așa cum o demonstrează valorile Mean și Best dramatic mai bune.

7. Amenințări la validitate (Threats to validity)

- Primul punct.

Parametrizarea Algoritmilor: Aceasta este cea mai mare amenințare. Simulated Annealing (SA) are parametri critici (de ex., schema de răcire). Dacă

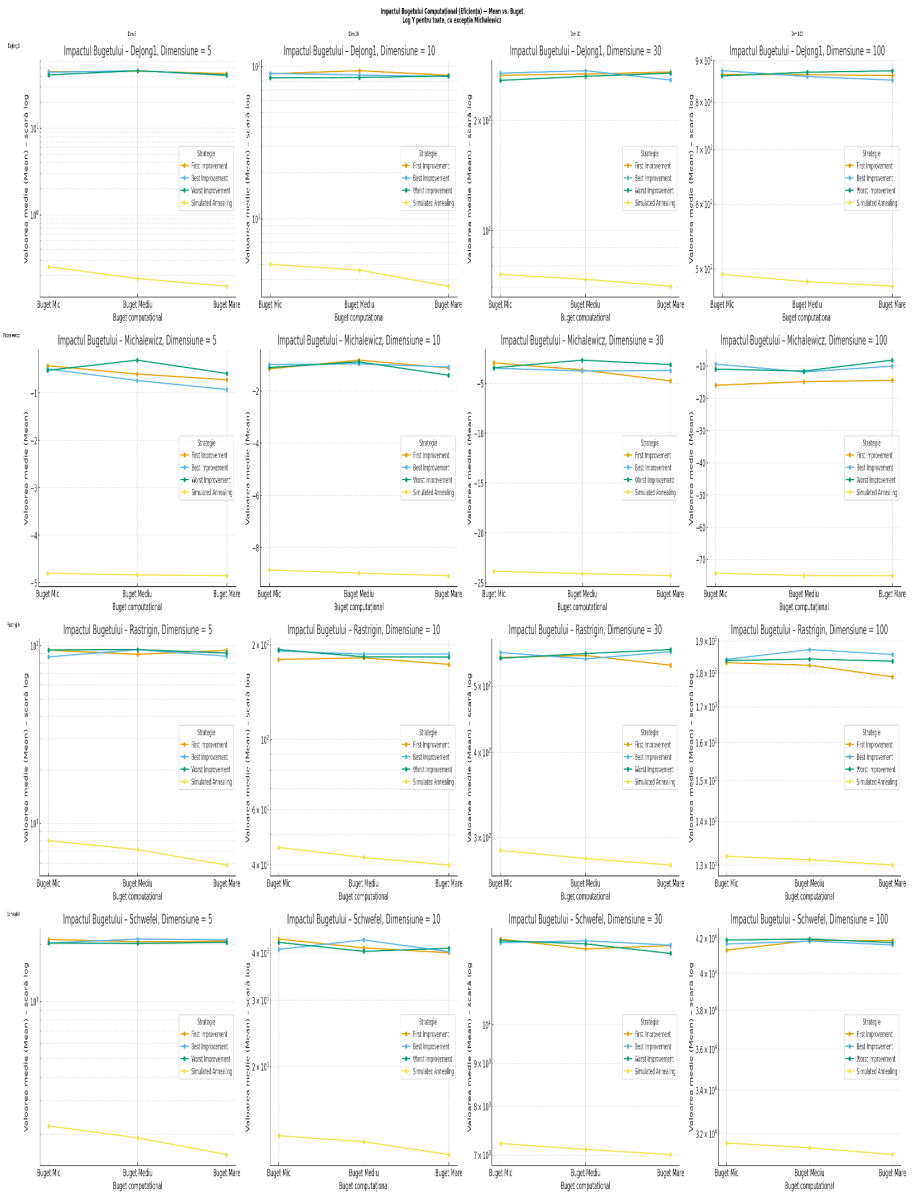

Figure 4: Impactul Bugetului Computațional: Performanța medie (Mean) vs. Buget (Mic, Mediu, Mare) pentru toate funcțiile și dimensiunile. Axa Y este logaritmică, cu excepția Michalewicz.

acești parametri au fost reglați (optimizat) extensiv pentru aceste funcții, în timp ce strategiile Hill Climbing (care au mai puțini parametri) au fost rulate cu setări implicite, comparația este nedreaptă. Practic, am compara o strategie SA "reglată" cu una Hill Climbing "nereglată".

- Primul punct.

Implementarea: Erori de implementare (bug-uri) într-una dintre strategiile Hill Climbing ar putea să le facă să pară mai slabe decât sunt în realitate.

- Primul punct.

Definiția Costului: Costul a fost măsurat exclusiv în număr de Evaluari medii. Această metrică ignoră faptul că o "evaluare" SA ar putea fi mai scumpă (în timp de procesor) decât o evaluare Hill Climbing, deoarece SA implică operații suplimentare (generare aleatorie, calculul probabilității). O comparație bazată pe timpul total de rulare ar putea arăta diferit.

- Primul punct.

Definiția Performanței: Performanța este măsurată ca valoarea finală (Mean, Best) după un buget fix de evaluări. Acest lucru nu ne spune nimic despre viteza de convergență. Este posibil ca o strategie Hill Climbing să găsească o soluție "suficient de bună" mult mai repede, chiar dacă soluția finală a SA este mai bună.

8. Concluzii și lucru viitor (mai mic sau egal ca 1 pagină)

Performanța actuală a SA se bazează pe un singur set de parametri (ex. temperatură inițială, rată de răcire) . Nu avem garanția că aceasta este configurația optimă.**Optimizarea Parametrilor (Meta-optimizare):** Se propune rularea unui studiu de meta-optimizare pentru a găsi setările optime ale SA pentru fiecare clasă de funcții.

Strategiile SA și Hill Climbing sunt, în forma lor de bază, "fără memorie" (decizia se ia doar pe baza stării curente și a celei noi). Adăugarea memoriei le transformă în algoritmi hibrizi, mai puternici.**Hibridizare SA + Hill Climbing (Exploatare):** Se propune un algoritm hibrid unde SA este folosit pentru **explorare globală** (pentru a găsi o regiune promițătoare a spațiului), urmat de o strategie Best Improvement pentru **exploatare locală** rapidă (pentru a găsi rapid optimul exact din acea regiune).

9. Bibliografie

- Manual curs AG

https://uaiciasi-my.sharepoint.com/:x:/g/personal/teodor_negura_student_uaic_ro/EWPZ1bTbTOJKoTZ1WFdOK7cBsGrZux6y1JyCD3iyeotO1g?e=5So9WF