**Analiza Hill Climbing pentru Maximizarea Funcției**

**f(x) = x³ - 60x² + 900x + 100**

**1. CONTEXT ȘI CONFIGURARE**

**Funcția Obiectiv**

* **Funcție**: f(x) = x³ - 60x² + 900x + 100
* **Domeniu**: x ∈ [0, 31] (întregi)
* **Reprezentare**: 5 biți (32 de valori posibile)
* **Maximul global pe [0,31] continuu**: x = 10

**Configurare Hill Climbing**

* **Reprezentare**: Binară pe 5 biți
* **Vecinătate**: Distanță Hamming = 1 (flip 1 bit)
* **Fiecare punct are exact 5 vecini**
* **Două variante**: First Improvement și Best Improvement

**2. MAXIME LOCALE IDENTIFICATE**

Funcția are **4 maxime locale** în spațiul discret [0, 31]:

| **x** | **Reprezentare Binară** | **f(x)** | **Calitate** |
| --- | --- | --- | --- |
| **10** | 01010 | **4100.00** | ★ **Optim Global** |
| 12 | 01100 | 3988.00 | Optim local bun |
| 7 | 00111 | 3803.00 | Optim local mediu |
| 16 | 10000 | 3236.00 | Optim local slab |

**Observații despre Maxime Locale**

* **x = 10** este optimul global și are cel mai mare fitness
* Diferența între optimul global și cel mai apropiat optim local (x=12) este de 112 unități
* Optimul local x = 16 este cel mai slab (cu 864 unități sub optim)

**3. ANALIZA FIRST IMPROVEMENT**

**3.1 Caracteristici**

* Algoritm **greedy rapid**: acceptă prima îmbunătățire găsită
* Ordinea de verificare a vecinilor influențează rezultatul
* Mai rapid dar potențial mai puțin precis

**3.2 Bazine de Atracție - First Improvement**

**Bazin pentru x = 10 (Optim Global) ★**

* **Dimensiune**: 14 puncte (43.75% din spațiu)
* **Puncte**: [5, 6, 9, 10, 11, 13, 14, 21, 22, 25, 26, 27, 29, 30]
* **Binare**: ['00101', '00110', '01001', '01010', '01011', '01101', '01110', '10101', '10110', '11001', '11010', '11011', '11101', '11110']

**Observații:**

* Cel mai mare bazin, dar nu dominant
* Include vecinii direcți ai lui x=10
* Acoperire răspândită în spațiu

**Bazin pentru x = 16**

* **Dimensiune**: 8 puncte (25% din spațiu)
* **Puncte**: [0, 1, 2, 3, 16, 17, 18, 19]
* **Binare**: ['00000', '00001', '00010', '00011', '10000', '10001', '10010', '10011']

**Observații:**

* Al doilea cel mai mare bazin
* Captează multe puncte de la începutul intervalului
* Problematic: mulți algoritmi pornesc de la x=0!

**Bazin pentru x = 12**

* **Dimensiune**: 6 puncte (18.75% din spațiu)
* **Puncte**: [4, 8, 12, 20, 24, 28]
* **Binare**: ['00100', '01000', '01100', '10100', '11000', '11100']

**Observații:**

* Pattern interesant: toate au bit 2 = 1, bit 0 = 0
* Concentrat în jurul lui x=12

**Bazin pentru x = 7**

* **Dimensiune**: 4 puncte (12.5% din spațiu)
* **Puncte**: [7, 15, 23, 31]
* **Binare**: ['00111', '01111', '10111', '11111']

**Observații:**

* Cel mai mic bazin
* Pattern clar: ultimii 3 biți = 111
* Include toate numerele de forma \*\*\*\*\*111

**3.3 Probleme cu First Improvement**

**Puncte critice:**

1. Pornind de la **x = 0**, ajunge la **x = 16** (optim local slab!)
2. **25%** din spațiu converg către optim local slab
3. Doar **43.75%** găsesc optimul global

**4. ANALIZA BEST IMPROVEMENT**

**4.1 Caracteristici**

* Algoritm **mai exhaustiv**: evaluează toți vecinii
* Alege cea mai bună îmbunătățire disponibilă
* Mai lent dar mai precis

**4.2 Bazine de Atracție - Best Improvement**

**Bazin pentru x = 10 (Optim Global) ★**

* **Dimensiune**: 20 puncte (62.5% din spațiu!)
* **Puncte**: [0, 1, 2, 3, 5, 8, 9, 10, 11, 13, 14, 15, 21, 24, 25, 26, 27, 29, 30, 31]
* **Binare**: ['00000', '00001', '00010', '00011', '00101', '01000', '01001', '01010', '01011', '01101', '01110', '01111', '10101', '11000', '11001', '11010', '11011', '11101', '11110', '11111']

**Îmbunătățire majoră:**

* A crescut de la 14 → 20 puncte (+42.9%)
* Acum domină cu 62.5% din spațiu!
* A capturat punctele 0, 1, 2, 3 de la bazinul lui x=16
* A capturat punctele 8, 15, 24, 31 de la alte bazine

**Bazin pentru x = 16**

* **Dimensiune**: 5 puncte (15.6% din spațiu)
* **Puncte**: [16, 17, 18, 19, 20]
* **Binare**: ['10000', '10001', '10010', '10011', '10100']

**Scădere semnificativă:**

* A scăzut de la 8 → 5 puncte (-37.5%)
* A pierdut punctele 0, 1, 2, 3 către optimul global

**Bazin pentru x = 7**

* **Dimensiune**: 4 puncte (12.5% din spațiu)
* **Puncte**: [6, 7, 22, 23]
* **Binare**: ['00110', '00111', '10110', '10111']

**Redistribuire:**

* Dimensiune constantă dar puncte diferite
* A pierdut [15, 31], a câștigat [6, 22]

**Bazin pentru x = 12**

* **Dimensiune**: 3 puncte (9.4% din spațiu)
* **Puncte**: [4, 12, 28]
* **Binare**: ['00100', '01100', '11100']

**Scădere:**

* A scăzut de la 6 → 3 puncte (-50%)
* A pierdut [8, 20, 24] către alte bazine

**4.3 Avantaje Best Improvement**

**Îmbunătățiri majore:**

1. **62.5%** din puncte găsesc optimul global (vs 43.75%)
2. Pornind de la **x = 0**, acum găsește **x = 10**!
3. Bazinul optimului local slab (x=16) s-a redus dramatic

**5. COMPARAȚIE DIRECTĂ**

**5.1 Diferențe între Metode**

**11 puncte de start** duc la optim diferit:

| **Start (x)** | **Binar** | **First →** | **Best →** | **Observații** |
| --- | --- | --- | --- | --- |
| 0 | 00000 | 16 | **10** | ✓ Best corectează eroarea |
| 1 | 00001 | 16 | **10** | ✓ Best corectează eroarea |
| 2 | 00010 | 16 | **10** | ✓ Best corectează eroarea |
| 3 | 00011 | 16 | **10** | ✓ Best corectează eroarea |
| 6 | 00110 | 10 | 7 | ✗ Best pierde |
| 8 | 01000 | 12 | **10** | ✓ Best găsește optim |
| 15 | 01111 | 7 | **10** | ✓ Best găsește optim |
| 20 | 10100 | 12 | 16 | ✗ Ambele suboptimale |
| 22 | 10110 | 10 | 7 | ✗ Best pierde |
| 24 | 11000 | 12 | **10** | ✓ Best găsește optim |
| 31 | 11111 | 7 | **10** | ✓ Best găsește optim |

**5.2 Statistici Comparative**

| **Metrică** | **First Improvement** | **Best Improvement** | **Câștigător** |
| --- | --- | --- | --- |
| Puncte → optim global | 14 (43.75%) | **20 (62.5%)** | **Best** |
| Puncte → x=16 (slab) | 8 (25%) | **5 (15.6%)** | **Best** |
| Puncte → x=12 | 6 (18.75%) | 3 (9.4%) | First |
| Puncte → x=7 | 4 (12.5%) | 4 (12.5%) | Egal |
| Viteză execuție | **Mai rapid** | Mai lent | **First** |
| Calitate soluții | Inferioară | **Superioară** | **Best** |

**5.3 Exemple de Căutări Instructive**

**Exemplu 1: Start x = 0**

First Improvement: 0 → 16 (f: 100 → 3236) ✗ Optim local slab

Best Improvement: 0 → 8 → 10 (f: 100 → 2276 → 4100) ✓ Optim global

**Exemplu 2: Start x = 15**

First Improvement: 15 → 7 (f: 3675 → 3803) ✗ Optim local

Best Improvement: 15 → 11 → 10 (f: 3675 → 4059 → 4100) ✓ Optim global

**Exemplu 3: Start x = 6**

First Improvement: 6 → 10 (f: 3796 → 4100) ✓ Optim global

Best Improvement: 6 → 7 (f: 3796 → 3803) ✗ Optim local

*Caz interesant unde First este mai bun!*

**6. ANALIZA RELIEFULUI FITNESS**

**6.1 Structura Spațiului de Căutare**

**Caracteristici reliefului:**

1. **4 "văi" separate** (bazine de atracție)
2. Funcția nu este monotonă
3. Există **platouri** locale
4. **Distanța Hamming** nu corelează perfect cu distanța în fitness

**6.2 Proximitatea în Spațiul de Căutare**

**Exemple de vecini surprinzători:**

| **x₁** | **x₂** | **Distanță Hamming** | **Δf(x)** |
| --- | --- | --- | --- |
| 10 | 11 | 1 | -41 |
| 10 | 14 | 1 | -220 |
| 7 | 15 | 1 | -128 |
| 0 | 16 | 1 | 3136 |

**Observație**: Un singur bit schimbat poate produce diferențe mari în fitness!

**6.3 Peisajul Adaptiv**

★ Optim Global (x=10, f=4100)

/|\

/ | \

Optim x=7 / | \ Optim x=12

(f=3803) / | \ (f=3988)

/ | \

/ | \

Optim x=16 ... ...

(f=3236)

**7. PATTERN-URI BINARE INTERESANTE**

**7.1 Observații despre Reprezentarea Binară**

**Bazinul x=7 (00111):**

* Toate punctele au pattern **\*\*\*111** (ultimii 3 biți = 1)
* Sugerează că biții mai puțin semnificativi sunt critici pentru acest bazin

**Bazinul x=12 (01100):**

* În First Improvement: pattern **\*\*1\*0** (bit 2 = 1, bit 0 = 0)
* Biții din mijloc definesc apartenența

**Bazinul x=16 (10000):**

* Grup compact: 10000 → 10100
* Primul bit (cel mai semnificativ) este întotdeauna 1

**7.2 Distanța Hamming între Maxime**

| **Pereche** | **Hamming** | **Observații** |
| --- | --- | --- |
| 10 ↔ 12 | 2 | 01010 ↔ 01100 (aproape!) |
| 10 ↔ 7 | 3 | 01010 ↔ 00111 |
| 10 ↔ 16 | 4 | 01010 ↔ 10000 |
| 7 ↔ 16 | 4 | 00111 ↔ 10000 |

**Concluzie**: Optimele locale sunt relativ apropiate în spațiul Hamming, ceea ce face căutarea dificilă!

**8. CONCLUZII ȘI RECOMANDĂRI**

**8.1 Avantaje Best Improvement**

✅ **Câștiguri semnificative:**

* +42.9% mai multe puncte găsesc optimul global
* Reduce captarea în optim local slab (x=16)
* Mai robust pentru inițializări proaste (ex: x=0)

✅ **Calitate:**

* 62.5% vs 43.75% șanse de succes
* Evită "curse locale" mai eficient

**8.2 Avantaje First Improvement**

✅ **Eficiență:**

* Mai rapid (evaluează mai puțini vecini)
* Pentru unele puncte de start specific, găsește mai repede soluția

✅ **Surprize pozitive:**

* Uneori evită atracția unor bazine (ex: x=6 → 10 vs 7)

**8.3 Dezavantaje Comune**

❌ **Ambele metode:**

* Nu garantează găsirea optimului global
* Sensibile la punctul de start
* 37.5% (Best) sau 56.25% (First) ajung în optim local suboptimal

**8.4 Recomandări Practice**

**Pentru această problemă specifică:**

1. **Utilizați Best Improvement** dacă:
   * Doriți calitate maximă
   * Aveți resurse computaționale
   * Nu știți punctul de start
2. **Utilizați First Improvement** dacă:
   * Viteza este critică
   * Puteți alege puncte de start inteligent
   * Resurse limitate
3. **Strategii de îmbunătățire:**
   * **Multi-start**: Rulați de la mai multe puncte random
   * **Perturbații**: Dacă ajungeți în optim local, randomizați și relansați
   * **Memorie**: Țineți evidența optimelor vizitate
   * **Combinație**: First pentru explorare rapidă, Best pentru rafinare

**Puncte de start recomandate:**

**Cele mai bune puncte de start (ambele metode → x=10):**

* 5, 9, 10, 11, 13, 14, 25, 26, 27, 29, 30

**Cele mai proaste puncte de start:**

* First: 0, 1, 2, 3, 4, 7, 12, 16, 20, 23, 28, 31 (37.5% → suboptim)
* Best: 4, 12, 16, 17, 18, 19, 20, 22, 23, 28 (31.25% → suboptim)

**9. TABEL COMPLET - TRASEELE DE CĂUTARE**

| **Start** | **Binar** | **First: Traseu** | **Optim** | **Best: Traseu** | **Optim** |
| --- | --- | --- | --- | --- | --- |
| 0 | 00000 | 0→16 | 16 | 0→8→10 | **10** |
| 1 | 00001 | 1→17→16 | 16 | 1→9→11→10 | **10** |
| 2 | 00010 | 2→18→16 | 16 | 2→10 | **10** |
| 3 | 00011 | 3→19→16 | 16 | 3→11→10 | **10** |
| 4 | 00100 | 4→12 | 12 | 4→12 | 12 |
| 5 | 00101 | 5→13→9→11→10 | **10** | 5→13→9→11→10 | **10** |
| 6 | 00110 | 6→10 | **10** | 6→7 | 7 |
| 7 | 00111 | 7 | 7 | 7 | 7 |
| 8 | 01000 | 8→12 | 12 | 8→10 | **10** |
| 9 | 01001 | 9→11→10 | **10** | 9→11→10 | **10** |
| 10 | 01010 | 10 | **10** | 10 | **10** |
| 11 | 01011 | 11→10 | **10** | 11→10 | **10** |
| 12 | 01100 | 12 | 12 | 12 | 12 |
| 13 | 01101 | 13→9→11→10 | **10** | 13→9→11→10 | **10** |
| 14 | 01110 | 14→10 | **10** | 14→10 | **10** |
| 15 | 01111 | 15→7 | 7 | 15→11→10 | **10** |
| 16 | 10000 | 16 | 16 | 16 | 16 |
| 17 | 10001 | 17→16 | 16 | 17→16 | 16 |
| 18 | 10010 | 18→16 | 16 | 18→16 | 16 |
| 19 | 10011 | 19→16 | 16 | 19→16 | 16 |
| 20 | 10100 | 20→4→12 | 12 | 20→16 | 16 |
| 21 | 10101 | 21→5→13→9→11→10 | **10** | 21→5→13→9→11→10 | **10** |
| 22 | 10110 | 22→6→10 | **10** | 22→23→7 | 7 |
| 23 | 10111 | 23→7 | 7 | 23→7 | 7 |
| 24 | 11000 | 24→28→12 | 12 | 24→8→10 | **10** |
| 25 | 11001 | 25→9→11→10 | **10** | 25→9→11→10 | **10** |
| 26 | 11010 | 26→10 | **10** | 26→10 | **10** |
| 27 | 11011 | 27→11→10 | **10** | 27→11→10 | **10** |
| 28 | 11100 | 28→12 | 12 | 28→12 | 12 |
| 29 | 11101 | 29→13→9→11→10 | **10** | 29→13→9→11→10 | **10** |
| 30 | 11110 | 30→14→10 | **10** | 30→14→10 | **10** |
| 31 | 11111 | 31→15→7 | 7 | 31→15→11→10 | **10** |

**10. REZUMAT EXECUTIV**

**Răspuns la Întrebarea Principală**

**"Cum devine relieful fitness când se utilizează First vs Best Improvement?"**

1. **Structura reliefului** rămâne aceeași (4 maxime locale)
2. **Bazinele de atracție** se reorganizează semnificativ
3. **Best Improvement** creează un bazin dominant pentru optimul global (62.5%)
4. **First Improvement** distribuie mai uniform punctele între bazine

**Cifre Cheie**

| **Aspect** | **First** | **Best** |
| --- | --- | --- |
| Succes (→ x=10) | 43.75% | **62.5%** |
| Cel mai mare bazin | 14 puncte | **20 puncte** |
| Puncte → optim slab | 25% | **15.6%** |

**Concluzie Finală**

**Best Improvement este clar superior** pentru această problemă, cu aproape **50% mai multe șanse** de a găsi optimul global. Cu toate acestea, **nici o metodă nu garantează succesul**, fiind necesare tehnici suplimentare (multi-start, perturbări) pentru robustețe maximă.

*Analiză realizată: 26 Octombrie 2025*