

Hill Climbing - Demonstrație Pas cu Pas

Funcția de optimizat:

$$f(x) = x^3 - 60x^2 + 900x + 100$$

unde $x \in \{0, 1, 2, \dots, 31\}$.

Reprezentare și vecinătate

Fiecare număr este reprezentat în binar pe 5 biți:

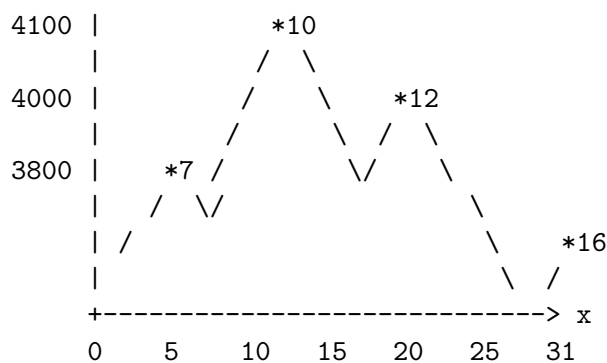
- $x = 0 \rightarrow 00000$
- $x = 10 \rightarrow 01010$
- $x = 31 \rightarrow 11111$

Vecinii unui punct se obțin prin schimbarea unui singur bit. Astfel, fiecare punct are exact 5 vecini.

Funcția are 4 maxime locale:

- $x = 10$: $f(10) = 4100$ (optimul global)
- $x = 12$: $f(12) = 3988$
- $x = 7$: $f(7) = 3803$
- $x = 16$: $f(16) = 3236$

Reprezentare grafică a funcției:



Exemplu 1: First Improvement din $x=0$

First Improvement acceptă prima îmbunătățire găsită, fără a evalua restul vecinilor.

Iterația 1

Poziția curentă: $x = 0$ (binar: 00000)

Fitness curent: $f(0) = 100$

Generare vecini prin schimbarea fiecărui bit:

Bit schimbat	Reprezentare	x	f(x)
bit 0	00001	1	841
bit 1	00010	2	1668
bit 2	00100	4	2916
bit 3	01000	8	2276
bit 4	10000	16	3236

Evaluare în ordine:

- Vecin 1: $x = 1$, $f(1) = 841 > f(0) = 100$
- Prima îmbunătățire găsită, evaluarea se oprește

Mișcare: $0 \rightarrow 1$

Îmbunătățire: $\Delta f = +741$

Iterația 2

Poziția curentă: $x = 1$ (binar: 00001)

Fitness curent: $f(1) = 841$

Vecini:

Bit schimbat	Reprezentare	x	f(x)
bit 0	00000	0	100
bit 1	00011	3	2548
bit 2	00101	5	3500
bit 3	01001	9	3700
bit 4	10001	17	3332

Evaluare:

- Vecin 1: $x = 0$, $f(0) = 100 < f(1) = 841$
- Vecin 2: $x = 3$, $f(3) = 2548 > f(1) = 841$
- Prima îmbunătățire găsită

Mișcare: $1 \rightarrow 3$

Îmbunătățire: $\Delta f = +1707$

Iterația 3

Poziția curentă: $x = 3$ (binar: 00011)

Fitness curent: $f(3) = 2548$

Vecini:

Bit schimbat	Reprezentare	x	f(x)
bit 0	00010	2	1668
bit 1	00001	1	841
bit 2	00111	7	3803
bit 3	01011	11	3971
bit 4	10011	19	3356

Evaluare:

- $x = 2$: $f(2) = 1668 < f(3) = 2548$
- $x = 1$: $f(1) = 841 < f(3) = 2548$
- $x = 7$: $f(7) = 3803 > f(3) = 2548$
- Prima îmbunătățire găsită

Mișcare: $3 \rightarrow 7$

Îmbunătățire: $\Delta f = +1255$

Iterația 4

Poziția curentă: $x = 7$ (binar: 00111)

Fitness curent: $f(7) = 3803$

Vecini:

Bit schimbat	Reprezentare	x	f(x)
bit 0	00110	6	3652
bit 1	00101	5	3500
bit 2	00011	3	2548
bit 3	01111	15	3500
bit 4	10111	23	3628

Evaluare tuturor vecinilor:

- Toți vecinii au fitness mai mic decât $f(7) = 3803$
- Nu există îmbunătățire posibilă

Algoritm oprit. Maxim local atins.

Rezultat First Improvement

Traseu: $0 \rightarrow 1 \rightarrow 3 \rightarrow 7$

Puncte vizitate: 4

Total evaluări: 11

Rezultat final: $x = 7$, $f(7) = 3803$

Calitate: 92.8% din optimul global

Exemplu 2: Best Improvement din $x=8$

Best Improvement evaluează toți vecinii și alege cel mai bun dintre ei.

Iterația 1

Poziția curentă: $x = 8$ (binar: 01000)

Fitness curent: $f(8) = 2276$

Vecini:

Bit schimbat	Reprezentare	x	f(x)
bit 0	01001	9	3700
bit 1	01010	10	4100
bit 2	01100	12	3988
bit 3	00000	0	100
bit 4	11000	24	3572

Evaluare completă:

- $f(9) = 3700$
- $f(10) = 4100$ (cel mai bun vecin)
- $f(12) = 3988$
- $f(0) = 100$
- $f(24) = 3572$

Cel mai bun vecin: $x = 10$ cu $f(10) = 4100$

Condiție îmbunătățire: $f(10) = 4100 > f(8) = 2276$ (îndeplinită)

Mișcare: $8 \rightarrow 10$

Îmbunătățire: $\Delta f = +1824$

Observație: În cazul First Improvement, algoritmul ar fi acceptat $x = 9$ (primul vecin mai bun). Best Improvement evaluează toți vecinii și selectează optimul local.

Iterația 2

Poziția curentă: $x = 10$ (binar: 01010)

Fitness curent: $f(10) = 4100$

Vecini:

Bit schimbat	Reprezentare	x	f(x)
bit 0	01011	11	3971
bit 1	01000	8	2276
bit 2	01110	14	3996
bit 3	00010	2	1668
bit 4	11010	26	676

Evaluare completă:

- $f(11) = 3971$ (cel mai bun vecin)
- $f(8) = 2276$
- $f(14) = 3996$
- $f(2) = 1668$
- $f(26) = 676$

Cel mai bun vecin: $x = 11$ cu $f(11) = 3971$

Condiție îmbunătățire: $f(11) = 3971 < f(10) = 4100$ (neîndeplinită)

Algoritm oprit. Maxim local atins.

Rezultat Best Improvement

Traseu: $8 \rightarrow 10$

Puncte vizitate: 2

Total evaluări: 10

Rezultat final: $x = 10$, $f(10) = 4100$

Calitate: 100% (optim global găsit)

Comparație

Rezultate numerice

Metrică	First (start 0)	Best (start 8)
Traseu	$0 \rightarrow 1 \rightarrow 3 \rightarrow 7$	$8 \rightarrow 10$
Puncte vizitate	4	2
Total evaluări	11	10
Evaluări/iterație	2.75 (variabil)	5.0 (fix)
Fitness final	3803	4100
Calitate	92.8%	100%
Optim găsit	Nu	Da

Caracteristici First Improvement

Algoritm:

1. Generare vecini
2. Evaluare în ordine fixă
3. Acceptare prima îmbunătățire
4. Oprește evaluare
5. Repetare până la convergență

Avantaje:

- Eficient computațional (1-5 evaluări/iterație)
- Convergență rapidă

Dezavantaje:

- Alegere greedy (acceptă prima opțiune mai bună)
- Risc crescut de convergență la maxime locale suboptimale
- Comportament non-deterministic (depinde de ordinea evaluării)

Caracteristici Best Improvement

Algoritm:

1. Generare vecini
2. Evaluare completă (toți vecinii)
3. Selectare cel mai bun vecin
4. Verificare condiție îmbunătățire
5. Repetare până la convergență

Avantaje:

- Direcție optimă de căutare (local)
- Probabilitate mai mare de găsire a optimului global
- Comportament deterministic

Dezavantaje:

- Cost computațional mai mare (evaluare completă obligatorie)
- Convergență mai lentă

Problema maximelor locale

Ambele metode pot converge către maxime locale în funcție de punctul de start.

Exemple pentru funcția dată:

Start	First converge	Best converge
0	$x = 7$ ($f = 3803$)	$x = 16$ ($f = 3236$)
8	$x = 12$ ($f = 3988$)	$x = 10$ ($f = 4100$)
15	$x = 7$ ($f = 3803$)	$x = 10$ ($f = 4100$)
20	$x = 12$ ($f = 3988$)	$x = 16$ ($f = 3236$)

Punctul de start influențează semnificativ rezultatul final.

Concluzii

Observații principale

1. Hill Climbing este o metodă de căutare locală. Rezultatul depinde de punctul inițial.
2. Best Improvement prezintă performanțe generale superioare:
 - Probabilitate mai mare de găsire a optimului global
 - Comportament reproductibil
 - Cost suplimentar moderat (aproximativ +40% evaluări)
3. Nicio metodă nu garantează găsirea optimului global.
4. Soluții pentru maximele locale:
 - Random restart (multiple puncte de start)
 - Simulated annealing (acceptare probabilistică a soluțiilor mai slabe)
 - Algoritmi genetici (evoluție de populații)

Domenii de aplicabilitate

Hill Climbing este potrivit pentru:

- Funcții cu puține maxime locale
- Vecinătăți mici
- Cerințe de viteză (soluții aproximative acceptabile)
- Posibilitatea rulării din multiple puncte de start

Hill Climbing nu este recomandat pentru:

- Funcții cu multe maxime locale
- Optima globale izolate
- Cerințe de garanție a optimului

Exemplu demonstrație pentru $f(x) = x^3 - 60x^2 + 900x + 100$ pe domeniul $[0, 31]$