

Simulated Annealing

Simulated Annealing

Simulated Annealing reprezintă o metodă algoritmică dezvoltată de Kirkpatrick et Al. În 1983, iar metoda sa are la baza algoritmul dezvoltat de Metropolis et Al. în 1953 și reprezintă un algoritm pentru a simula răcirea materialului.

Deci procesul de răcire a unui metal se numește recoacere.

Dacă un material solid este încălzit peste punctul său de topire și apoi răcit înapoi la stare solidă, proprietățile structurale ale solidului rezultat după răcire depind de viteza de răcire.

Kirkpatrick a sugerat că acest tip de simulare ar putea fi folosit pentru a căuta soluții fezabile ale problemelor de optimizare cu scopul de a converge către o soluție optimă.

Simulated Annealing

Astfel, s-a demonstrat că algoritmul Metropolis poate fi aplicat problemelor de optimizare prin maparea elementelor procesului de răcire fizică pe elementele unei componente a problemelor combinatorii sau chiar de optimizare continuă.

Simulated Annealing (Recoacerea simulată) este o variantă de căutare locală sau de vecinătate, fiind de asemenea, o metodă metaeuristică probabilistică, deoarece transformă probabilitățile implicate.

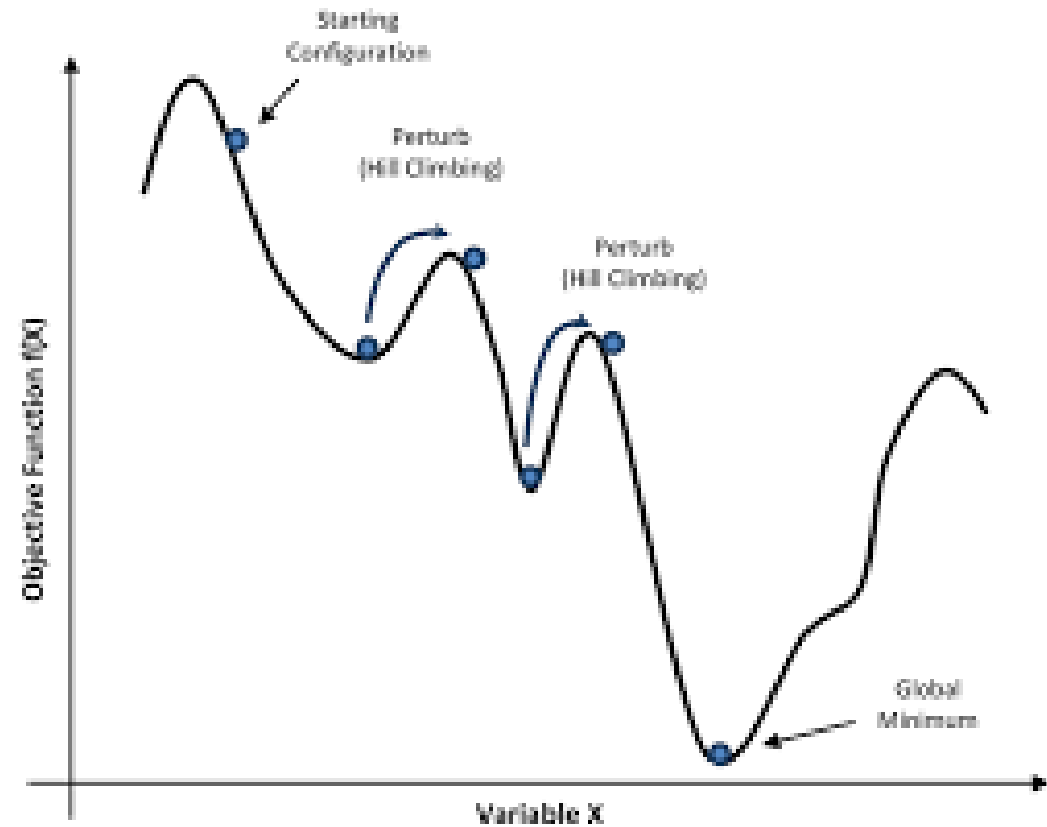
*“A **metaheuristic** is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space; learning strategies are used to structure information in order to find efficiently near-optimal solutions” [I.H. Osman, G. Laporte, Metaheuristics: a bibliography, Annals of Operations Research 63 (1996) 513–623]*

Simulated Annealing

Kirkpatrick, a utilizat două concepte din procesul de recoacere propriu-zis: temperatura și rata de răcire.

Conf. Imaginii alăturate observăm că mergând mai departe din minimul local, vom obține soluții mai rele, dar după aceea putem obține minimul căutat.

Această metodă ne permite să depășim un minim local acceptând mutări mai rele, adică un minim mai slab decât cel mai bun obținut până atunci.



Simulated Annealing

Mutările mai rele sunt acceptate pentru a putea trece de un minim local.

Acceptarea unei astfel de mutări se realizează prin alegerea unui număr aleatoriu între 0 și 1 care este comparat cu rezultatul formulei din această pagină.

În cazul în care valoarea aleatorie aleasă este mai mică decât rezultatul funcției se acceptă mutarea.

$f(x_{tmp})$ – reprezintă valoarea obiectiv calculată pentru mutarea curentă

$f(x_m)$ – reprezintă valoarea minimă obținută la pasul m

T_m – este temperatura la pasul m

$$\frac{1}{e^{\left(\frac{f(x_{tmp}) - f(x_m)}{T_m}\right)}}$$

Simulated Annealing

Pe măsură ce numărul de pași realizați (m) crește, șansele pentru a alege o soluția mai slabă decât cea mai bună soluție scad, întrucât rezultatul funcției va fi tot mai aproape de 0.

Operatorul de mutare este diferit în funcție de problema abordată:

- pentru o funcție matematică cu un parametru este alegerea noii valori care să reprezinte posibila soluție
- pentru problema comis-voiajorului, poate fi dată de inversarea ordinii a două locații

Simulated Annealing

M – numărul de pași de răcire

x_0 – soluția inițială

x_m – soluția la pasul m , unde $m \in \{0, \dots, M\}$

$f(x_m)$ – funcția obiectiv pentru soluția x_m

T_0 – temperatura inițială

T_m – temperatura la pasul m

α – rata de răcire la fiecare pas

N – numărul de pași realizați la fiecare temperatură

δ – operatorul de mutare

Simulated Annealing

Algoritmul este următorul:

- *stabilim inițial: temperatura inițială, rata de răcire, numărul de pași de răcire, numărul de pași realizați la fiecare temperatură*
- *alegem aleatoriu soluția inițială și calculăm valoarea funcției obiectiv*
- *iterăm pentru fiecare pas de răcire:*
 - *iterăm pentru numărul de căutări locale:*
 - *alegem o posibilă soluție din vecinătate*
 - *dacă este mai mică valoarea funcției obiectiv o considerăm valoare actuală*

Simulated Annealing

--- altfel:

---- alegem un număr aleatoriu între 0 și 1

---- calculăm valoarea conform formulei

---- dacă numărul este mai mic decât valoarea formulei alegem soluția mai puțin bună

-- modificăm temperatura aplicând coeficientul stabilit

- afisăm soluția finală și valoarea funcției obiectiv

Simulated Annealing

Exemple de funcții continue:

- *Funcția Himmelblau*
- *Funcția Ackley*
- *Funcția Rastrigin*

O suită de funcții care pot fi testate cu acest algoritm se găsesc la adresa:

https://en.wikipedia.org/wiki/Test_functions_for_optimization

Simulated Annealing

Exemplificare în Python a modului de rezolvare cu această metodă.

Simulated Annealing

Cosmin Sabo

VĂ MULȚUMESC!