

Studiu al variatiilor algoritmului Hill Climbing

Cristian Bozomitu

November 7, 2022

1 Ce este hill climb?

În analiza numerică, hill climbing este o tehnică de optimizare matematică. Este un algoritm iterativ care porneste cu o soluție aleasă arbitrar pentru a rezolva o problemă, ca după să încerce să găsească o soluție mai bună printr-o schimbare incrementală a soluției alese inițial. Acest algoritm poate fi implementat sub mai multe forme, iar în acest studiu ne vom concentra pe variațiile de first improvement, best improvement, worst improvement și simulated annealing.

(a) First improvement va genera la întâmplare o soluție candidat dintr-un anumit interval, care adesea este domeniul unei funcții peste care se aplică algoritmul, și o va considera ca fiind un candidat sau stare curentă. Algoritmul se va duce în vecinătatea stării curente pentru a selecta o nouă soluție candidat, care va fi denumită ca stare următoare. Prima soluție optimă care va fi găsită de variația first improvement va fi selectată de algoritm, aceasta în cele din urmă să devină noua stare curentă. Algoritmul va repeta acești pași până când va da de o soluție care nu este optimă, moment în care ultima soluție optimă găsită va deveni punct de maxim/minim local.

(b) Best improvement va face aceiași pași ca și variația first improvement, singura diferență fiind faptul că această variație va selecta din vecinătate cea mai optimă soluție posibilă. Spre deosebire de first improvement, best improvement va consuma mai mult timp de execuție, deoarece algoritmul va trece prin toate posibilitățile din vecinătatea curentă, alegând-o pe cea mai optimă. În ciuda faptului că first improvement este adesea mai rapid decât best improvement, acesta de regulă tinde să ofere rezultate mult mai bune.

(c) Worst improvement se diferențiază de celelalte două variații discutate prin faptul că acesta va alege cea mai nefavorabilă soluție din vecinătatea curentă, oferind astfel și cele mai nefavorabile soluții. În ciuda acestui fapt, există implementări a acestei variații foarte bine structurate și analizate, care pot aduce la găsirea unor soluții optime.

2 Detalii tehnice despre proiect

În codul nostru am folosit o clasă numită HillClimbing care conține 4 metode, fiecare metodă implementând o variație a algoritmului hill climbing. Clasa are

un atribut privat `IFunctions*function` care va fi initializat prin constructorul sau cu adresa unui obiect derivat din interfata `IFunctions`,obiectele derivate fiind instantieri ale claselor `Dejongf`,`Rastriginf`,`Schwefelf`.Prin aceasta initializare in constructor,se va aplica polimorfism(principiu al P.O.O),lucru ce ne va permite sa folosim cate o singura metoda pentru fiecare functie implementata de interfata `IFunctions`.Fiecare metoda din `HillClimbing` are 3 parametri,aceia fiind dimensiunea si cele doua capete de interval.Exceptie face metoda care implementeaza variatia `simulated annealing` la care,pe langa cei trei parametri mentionati mai sus,se adauga si o temperatura.Algoritmul nostru va avea doua bucle `while`.Prima bucla va itera dimensiunile date ca parametru in metoda,iar a doua va itera schimbarile de stare pe care algoritmul le produce pentru a gasi o solutie candidat optima.Deoarece in studiul nostru ne intereseaza punctul de minim global,ne intereseaza ca solutiile noastre sa reprezinte valorile pentru care functiile alese sa returneze cele mai mici valori posibile.(cat mai aproape de punctul minim global)Asadar,ne intereseaza ca evaluarea starii viitoare sa fie mai mica decat evaluarea starii curente,ca apoi starea viitoare sa devina noua stare curenta.In caz contrar,se va indeplini conditia de iesire din al doilea `while`,anume gasirea unui punct de minim local,din care nu mai putem avansa pentru dimensiunea curenta(iteratia din primul `while`).Rezultatele gasite reprezinta punctele de minim local gasite de fiecare metoda, reprezentate intr-un vector din STL.Fiecare metoda precizeaza care a fost cea mai buna solutie si care a fost cea mai slaba solutie dupa traversarea tuturor dimensiunilor.

3 Rezultate obtinute de variatiile hill climb

- (1) Variatiile hill climb pentru functia `DeJong1`
 - (a)Variatia `First Improvement`
 - (b)Variatia `Best Improvement`
 - (c)Variatia `worst improvement`

4 Comparatii intre metode

Se poate observa ca rezultatele obtinute de variatia `best improvement` vor fi mereu mai bune decat cele obtinute de celelalte variatii.Pe langa cele trei variatii discutate,o alta variatie a `hill climbing` este `simulated annealing`.`Simulated annealing` este o tehnica probabilistica pentru a aproxima optimul global a unei functii date.Mai specific,este o metaheuristica pentru a aproxima optimizarea globala intr-un spatiu mare pentru o problema de optimizare.Notiunea de racire incetinita implementata in `simulated annealing` este interpretata ca o coborare lenta in probabilitatea de a accepta solutii mai slabe in timp ce spatiul de cautare este explorat.Temperatura scade progresiv de la o valoare initiala la zero(aceasta scadere este implementata in algoritmul nostru printr-o inmultire a variabilei `Temperature` cu o constanta subunitara).La fiecare pas,algoritmul selecteaza la intamplare o solutie apropiata de starea curenta si o evalueaza,mai

apoi interpretand-o cu probabilitatile de a selecta solutii mai bune sau mai slabe, in functie de probabilitatile dependente de temperatura curenta.

1	0.592308
2	3.45738
3	0.925285
4	4.15668
5	0.355435
6	-0.854321
7	2.01874
8	-3.7569
9	-4.00944
10	1.31244
11	-0.917237
12	-1.25863
13	3.15212
14	0.560144
15	1.58421
16	2.22764
17	3.11085
18	-0.16707
19	-4.6179
20	1.43043
21	-0.136835
22	-1.24387
23	-2.86204
24	0.228886
25	-4.01842
26	-0.703474
27	1.22072
28	-0.426557
29	0.269807
30	3.15743

References

- [1] wikipedia
Definitii tehnice pentru simulated annealing https://en.wikipedia.org/wiki/Simulated_annealing
- [2] Youtube
Explicatia algoritmului simulated annealing. <https://www.youtube.com/watch?v=7Ch8P0TPVrU>

1	0.198829
2	-0.108793
3	1.50977
4	1.21698
5	-0.0521841
6	0.135347
7	0.298517
8	-0.00341618
9	0.00530377
10	-1.49679
11	0.556295
12	-1.95834
13	1.87262
14	0.0359109
15	-0.00171305
16	-1.03432
17	-0.0227606
18	-0.471735
19	0.0330798
20	1.49637
21	0.613409
22	1.76326
23	2.37439
24	-0.281863
25	-0.088366
26	-0.153304
27	0.699428
28	2.10687
29	2.17837
30	0.899117

Explicatia celor 3 variatii de hill climbing(first/best/worst improvement)
<https://www.youtube.com/watch?v=xtAnYm6hnz0>

- [3] Site profesor seminar algoritmi genetici
 Detalii despre implementarea algoritmilor folositi in cod <https://profs.info.uaic.ro/~eugennc/teaching/ga/>
- [4] GEATbx
 Detalii despre functiile implementate de algoritmii folositi in cod http://www.geatbx.com/docu/fcnindex-01.html#P89_3085