La inceput am zis ca pentru determinarea optimelor globale, Solutia determinista ia foarte mult timp si din cauza asta se folosesc algoritmi euristici, iar in acest raport voi compara rezultatele obtinute de 2 astfel de algoritmi si anume Hill Climbing si Simulated Annealing.

Am descris pe scurt cum functioneaza cei doi algoritmi. Hill climbing selecteaza aleator o reprezentare ca si solutie candidat si incearca sa o imbunatateasca intr-un mod iterativ prin cautare printre vecinii sai care se afla la distanta Hamming 1 de el. Daca nu a putut fi imbunatatita Solutia, atunci se afla intr-un punct de minim local.

Algoritmul are doua variatii care depind de cum imbunatateste soltuia candidat. Best Improvement care ia cea mai buna imbunatatire dintre toti vecinii si First Improvement care ia prima imbunatatire gasita. In practica e folosita varianta iterativa pentru a folosi o solutie candidat initiala diferita.

Simulated Annealing selecteaza aleator o reprezentare pentru Solutia candidat initiala. Spre deosebire de Hill Climbing, permite uphill jumps datorita unei componente de temperature. In Implementare, algortimul are 2 bucle. Cea interioara verifica vecini aleatorii la distanta Hamming 1 de Solutia candidat si incearca sa o imbunatateasca. Daca nu s-a putut imbunatatii, exista sansa de a sari la o solutie apparent mai rea printr-o probabilitate care devine din ce in ce mai putin probabila cu cat temperature scade. Cea exterioara este legata de temperature care se raceste spre o stare de echilibru unde va gasim un optim.

Pentru experiment am ales cateva functii pe care vom aplica algoritmii in incercarea de a le gasi minimul global. Am testat pe mai multe dimensiuni. Precizia este de 5 zecimale, 10-5, iar sample size-ul e de 30.

Valorile mai mici de 10-5 au fost aproximate la 0. Si au fost 10k iteratii. Functia lui De Jong are minimul global in 0. Din cauza formei functiei, s-au obtinut rezultate care au fost approximate la 0. La timp putem observa deja ca odata cu cresterea numarului de dimensiuni, creste si timpul de executie si variatia la cat dureaza o executie. Am observant o greseala de redactare. Media e 2404.555

La First Improvement am reusit sa obtinem 0, dar intr-un timp mai bun fata de Best Improvement, datorita formei functiei

La Simulated Annealing am obtinem tot 0 si obtinem si cei mai buni timpi de executie. Temperatura de start e de 150 si o inmultesc cu 0.995. Conditia de oprire a temperaturii e cand aceasta scade sub 10-7.La 5 dimensiuni a obtinut timp asemanator cu HCBI, dar in rest cei mai buni timpi in medie.

La Schwefel, minimul global depinde de numarul de dimensiuni.

Se observa o crestere exponentiala in timpul de executie odata cu cresterea numarului de dimeniuni.

First Improvement rezultat apropiat la 5 dimensiuni de HCB si din ce in ce mai departate cu cresterea numarului de dimesniuni, intr-un timp mai scurt.

SA are cel mai bun timp de executie in medie si din tabel se observa ca are si in medie cel mai apropiat rezultat de minimul global. La aceasta functie are si o deviatie standard mai mica decat HC.

La Rastrigin, Minimul global e in 0, dar functia are multe minime locale. Observam ca odata cu cresterea numarului de dimeniuni, algoritmii in medie incep sa se departeze de 0. La 5 dimensiuni, HCB a reusit sa gaseasca minimul, spre deosebire de HCF si SA. La 10 dimensiuni e cel mai apropiat de 0 si cu un timp rezonabil, dar la 30 de dimensiuni, SA e are in medie cea mai mica valoare si in cel mai bun timp.

Si la functia lui Michalewicz minimul global depinde de numarul de dimensiuni. Un Typo, am uitat – la medie. Timpul creste rapid cu numarul de dimensiuni la HCB, dar si la HCF care obtine rezultate putin mai rele decat HCB. SA pastreaza cel mai buni timpi de executie si observam din nou ca pentru 30 de dimensiuni obtine cel mai bun rezultat si in cel mai rezonabil timp dintre toti algoritmii.

In urmatoarele grafice am comparat timpi in medie raportat la numarul de dimensiuni. Eroare de redactare la DE Jong, trebuie inversate HCB si HCF. In general se observa ca HCB are o crestere rapida cu numarul de dimensiuni, iar SA are cel mai bun timp si mic timp, iar HCF are o crestere usor asemanatoare, insa nu la fel de rapida ca HCB.

Din rezultatele obtinute, observam ca SA obtine rezultatele cele mai bune in ce mai bun timp la numar ridicat de dimensiuni. Prin alegerea unei temperature potrivite si a unei rate de racier potrivite, SA poate fi optimizat pentru a obtine rezultate mai bune decat HC.

In Concluzie, Amandoi algoritmii ofera rezultate bune intr-un timp rezonabil de timp, insa HC tinde sa aiba o durata de timp mai mare decat SA. Dintre variantele HC, First are rezultate mai putin bune decat Best, insa intr-un timp mai bun. SA a avut cele mai bune rezultate la dimeniuni multe si a fost si cel mai rapid dintre toti.