

POLITECHNIKA WARSZAWSKA
WYDZIAŁ MATEMATYKI I NAUK INFORMACYJNYCH



WSTĘP DO ALGORYTMÓW EWOLUCYJNYCH

Wielostartowy algorytm wspinaczkowy
dla ciągłych zadań optymalizacji o
różnych wymiarowościach

RAPORT

Autorzy:

Anna ZAWADZKA
Piotr WASZKIEWICZ

27 maja 2016

1 Opis problemu

Celem projektu było zbadanie zachowania wielostartowego algorytmu wspinaczkowego dla ciągłych zadań optymalizacji o różnych wymiarowościach z uwzględnieniem różnych strategii losowania punktów startowych: losowanie z rozkładem równomiernym, przeszukiwanie po hipersiatce, poisson-disc. Testy przeprowadzone zostały na benchmarku CEC 2013.

Podczas działania algorytmu wspinaczkowego można napotkać takie problemy jak: lokalne minima, *plateaux*, czyli równiny oraz wąskie grzbiety. W niektórych problemach mogą pomóc wielokrotne starty z przypadkowych punktów, stąd cel tego zadania - sprawdzenie które podejście może zminimalizować ryzyko nieodnalezienia poszukiwanego globalnego ekstremum.

2 Metoda realizacji zadania

2.1 Algorytm wspinaczkowy

W projekcie wykorzystany został algorytm wspinaczkowy w wersji z wyborem następnika na podstawie sąsiadów najlepszego znalezionej dotychczas punktu. Należy zaznaczyć, że wszystkie funkcje w pakiecie CEC 2013 realizują problem minimizacji.

```
x ← x0
while !stop
    y ← randomNeighbor(x, δ)
    if cec2013(i, y) < cec2013(i, x)
        x ← y
```

Dodatkowo zaimplementowany algorytm co 100 iteracji (czyli co 100 losowań nowego punktu sąsiadującego z najlepszym dotychczas znalezionym rozwiązaniem) zapisuje wartość funkcji celu dla najlepszego osobnika. Warunkiem stopu jest uzyskanie 50 obserwacji.

Funkcja *randomNeighbor(x, δ)* zwraca losowy punkt w sąsiedztwie x -a, znajdujący się w odległości nie większej niż δ od x -a. Do obliczania odległości stosowana jest metryka euklidesowa. W przypadku wygenerowania punktu, który

znajduje się poza dopuszczalnym obszarem, jest on korygowany przy pomocy metody zawijania.

Funkcja *cec2013(i, z)* dostępna jest w ramach pakietu CEC2013 i zwraca wartość funkcji *i* w punkcie *z*. Traktowana jest jako funkcja celu.

2.2 Strategie losowania punktów startowych

Punkty startowe wybierane są przy pomocy następujących metod:

2.2.1 Losowanie z rozkładem równomiernym

Rozkład równomierny gwarantuje wybranie punktu z określonej *n*-wymiarowej przestrzeni z jednakowym prawdopodobieństwem. Środowisko R dostarcza funkcję *runif(n, a, b)*, która zwraca *n* liczb z przedziału $[a, b]$ z rozkładu równomiernego, które traktowane są jako kolejne współrzędne generowanego punktu.

2.2.2 Przeszukiwanie po hipersiatce

Metoda ta zakłada istnienie *n*-wymiarowej siatki. Przecięcia linii tejże siatki wyznaczają kolejne punkty, będące punktami startowymi algorytmu.

2.2.3 Poisson Disc Sampling

Technika Poisson Disc generuje zbiór punktów o zadanej liczności w *n*-wymiarowej przestrzeni, ściśle upakowanych, lecz odległych od siebie co najmniej o podaną minimalną odległość. Algorytm zrealizowany został zgodnie z metodą podaną w poniższym dokumencie:

<http://www.cs.ubc.ca/~rbridson/docs/bridson-siggraph07-poissondisk.pdf>

3 Przebieg eksperymentów

Funkcja cec2013 dostarczona w ramach pakietu CEC2013 przyjmuje dwa argumenty: punkt oraz numer funkcji wykorzystanej do obliczenia wartości funkcji celu dla podanego punktu. Dostępne funkcje numerowane są od 1 do 28, a podany punkt musi być punktem z przestrzeni o wymiarze 2, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90 lub 100. Daje to 336 doświadczeń, które dodatkowo przeprowadzane są dla każdej z trzech strategii losowania punktów startowych, co daje 1008 eksperymentów. Liczba doświadczeń przez nas przeprowadzonych jest jednak mniejsza ze względu na dużą złożoność czasową obliczeń. Ponadto generowanie punktów metodą Poisson Disc zrealizowane jest tylko dla przestrzeni 2 i 5-wymiarowej ze względu na bardzo dużą złożoność pamięciową.

Do eksperymentów zostały wybrane wymiary: 2, 5, 10, 20 oraz funkcje: 1, 2, 7, 16, 23. Poniższa tabela przedstawia wartości minimalne wybranych funkcji (realizujemy problem minimaizacji).

numer funkcji	minimum funkcji celu
1	-1400
2	-1400
7	-800
16	200
23	900

Funkcje te zostały wybrane na podstawie analizy poniższego dokumentu, który zawiera opisy i wizualizacje wszystkich dostępnych w pakiecie funkcji:

http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/Definitions%20of%20CEC%2013%20benchmark%20suite%200117.pdf

Ogólna postać głównego algorytmu:

```
dimensions ← c(2,5,10,20);  
functions ← c(1,2,7,16,23);  
N ← 10;  
foreach d in dimensions  
  S1 ← starting points set – uniform distribution;  
  S2 ← starting points set – hyper mesh;  
  S3 ← starting points set – poisson disc;  
  foreach f in functions)  
    run hillClimbing for d, f, S1;  
    run hillClimbing for d, f, S2;  
    run hillClimbing for d, f, S3;
```

Algorytm wspinaczkowy wywoływany jest dla każdego punktu ze zbioru punktów startowych, po czym następuje wyliczenie wartości średnich na podstawie wyników otrzymanych dla każdego z punktów startowych.

Ponieważ testowana metoda jest niedeterministyczna, **każde doświadczenie jest powtarzane 20 razy**, a wyniki uśredniane.

4 Wyniki eksperymentów

Poniższe wykresy przedstawiają wartości funkcji celu dla kolejnych najlepszych osobników w populacji.

5 Wnioski

Na podstawie przeprowadzonych eksperymentów z pewnością możemy stwierdzić, że wielokrotne uruchomienie algorytmu wspinaczkowego z różnych punktów startowych przyczynia się do uzyskania lepszych wyników, niż rozpatrzenie jednego punktu startowego. Takie podejście zwiększa szanse na to, że globalne ekstremum zostanie osiągnięte i zmniejsza ryzyko napotkania lokalnych ekstremów.

Wyniki przeprowadzonych doświadczeń pokazują, że najbardziej efektywną metodą wyboru punktów startowych jest...