

Strategia Ewolucyjna 1+1

Karol Korszun

March 2024

1 Opis algorytmu

Algorytm (1+1)-ES (Evolution strategy) jest algorytmem optymalizacyjnym służącym do znajdowania ekstremum funkcji celu w problemach optymalizacji ciągłej. Należy do szerszej klasy algorytmów ewolucyjnych, które opierają się na zasadach inspirowanych przez teorie ewolucji. Warto zaznaczyć, że ta metoda nie wykorzystuje różniczkowania, co może być jej zaletą. Na początku wybieramy losowy wektor jako punkt startowy. Następnie nowy osobnik jest tworzony poprzez dodanie losowego wektora, który powstaje przez pomnożenie parametru σ przez zmienną losową generowaną z rozkładem normalnym, do aktualnego rozwiązania. Matematycznie można to zapisać jako:

$$x'_j = x_j + \sigma \cdot \mathcal{N}(0, 1)$$

Gdzie:

- x'_j oznacza nowego osobnika,
- x_j oznacza aktualne najlepsze rozwiązanie,
- σ jest parametrem określającym wielkość kroku ewolucji,
- $\mathcal{N}(0, 1)$ to losowa zmienna z rozkładem normalnym o średniej 0 i odchyleniu standardowym 1.

Jeśli wartość funkcji dla nowego osobnika jest mniejsza niż dla jego rodzica wtedy to on staje się obecnym rodzicem. W algorytmie została też zaimplementowana reguła 1/5 sukcesu. Reguła 1/5 sukcesu implementuje koncepcję, według której wielkość kroku powinna być zwiększana, jeśli "zbyt wiele" kroków kończy się sukcesem, co wskazuje na zbyt lokalne poszukiwanie, oraz zmniejszana, jeśli "zbyt mało" kroków kończy się sukcesem, co sugeruje, że długość kroku używana do próbkowania rozwiązań jest "zbyt duża". Optymalnie prawdopodobieństwo udanych kroków powinno być bliskie jednej piątej.

2 Testy algorytmu

2.1 Wizualizacja funkcji

W zadaniu należało przeprowadzić eksperymenty na funkcjach f3 oraz f19 z biblioteki cec2017, która służy do testowania algorytmów optymalizacyjnych.

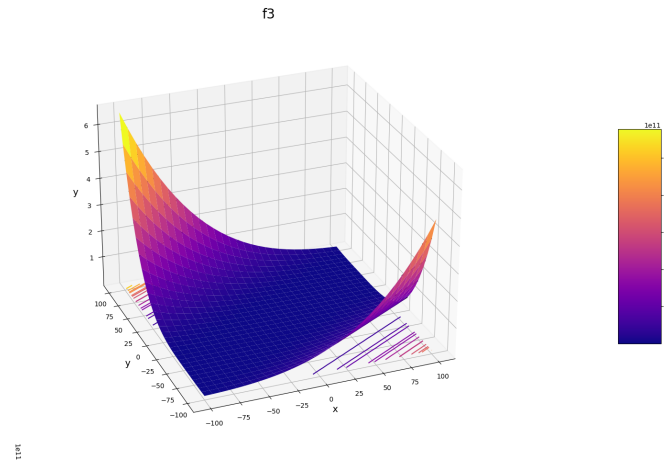


Figure 1: Funkcja f3

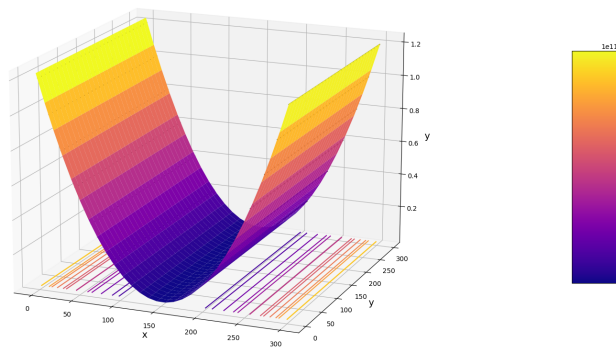
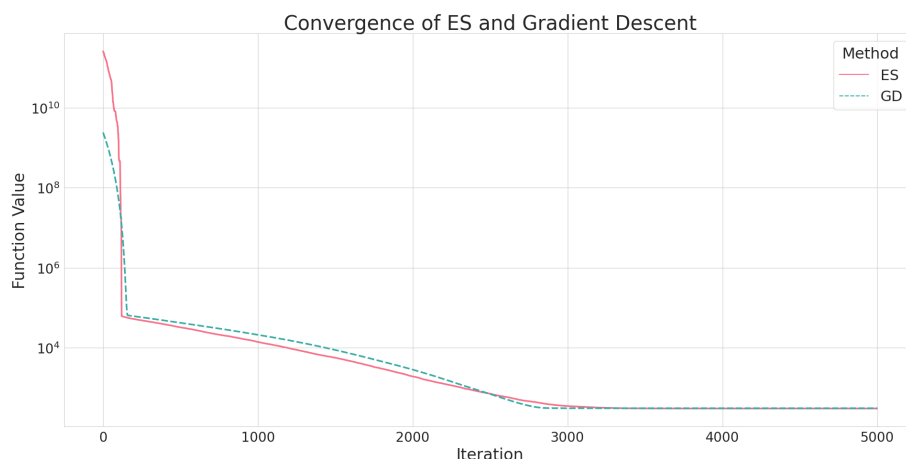


Figure 2: Funkcja f19

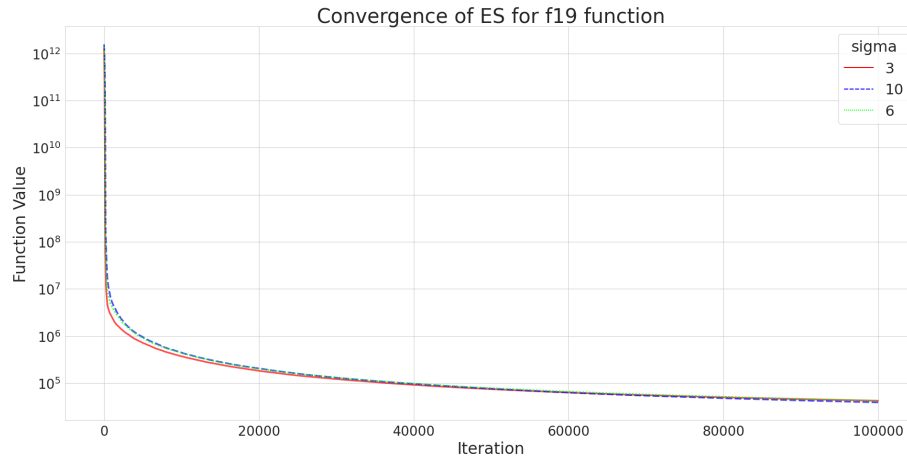
2.2 Analizacja zbieżności

Spójrzmy na porównanie zbieżności algorytmu testowanego na funkcji f3. Wektor w tym, jak i w pozostałych przypadkach, miał 10 wymiarów. Algorytm działał przez 5000 generacji w dziedzinie od -100 do 100. Parametr sigma był równy 3. Na wykresie ukazano również trajektorie zbieżności algorytmu gradientu prostego. Jak możemy zauważyć oba algorytmy osiągnęły minimum funkcji w dosyć podobną ilość iteracji. Na wykresie została przedstawiona skala logarytmiczna. Rozmiar kroku dla metody gradientu prostego wynosił 0.001. Warto wspomnieć, iż w metodzie gradientu prostego została zaimplementowany tak zwany "gradient clipping" ponieważ w początkowej fazie działania gradienty stawały się zbyt duże co prowadziło do nieprawidłowego działania programu.



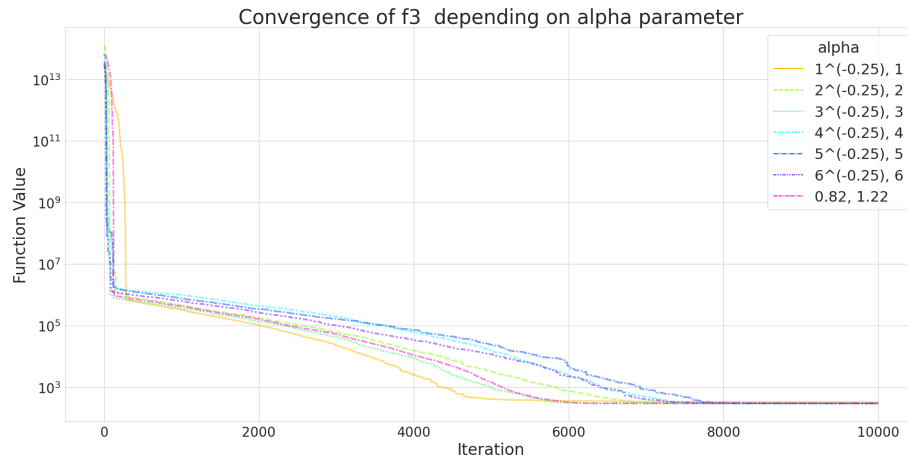
2.3 Wpływ parametru σ na zbieżność

Przeprowadzono również eksperyment na funkcji f19 w celu określenia wpływu parametru sigma na zbieżność algorytmu. Wybrano 3 parametry $\sigma = 3$, $\sigma = 6$, $\sigma = 10$. Algorytm działał przez 100000 generacji. Wyniki, gdzie oś y jest w skali logarytmicznej, zostały przedstawione poniżej. Wynika z nich, że dla różnych wartości parametru σ algorytm zbiegał do tych samych wartości. Wartość funkcji dla każdej iteracji została uśredniona z 50 niezależnych eksperymentów.



2.4 Wpływ parametru α na zbieżność

W tym eksperymencie porównano wpływ parametru α na zbieżność. Definiujemy parametr α jako dwie liczby. W regule 1/5 sukcesu jeśli na przestrzeni n ostatnich generacji odsetek sukcesu jest mniejszy niż 1/5 to mnożymy nasz parametr σ przez pierwszą liczbę parametru α podniesioną do potęgi -0.25 , a gdy jest większy to mnożymy przez drugą liczbę tego parametru.



Jak możemy zauważyć wartości parametru α mają dość duży wpływ na szybkość zbieżności algorytmu, jednak dla wszystkich parametrów algorytm finalnie znajduje minimum.

3 Wnioski i obserwacje

Analizy działania algorytmu strategii ewolucyjnej w wariancie 1+1 pokazały, że algorytm ten radzi sobie z optymalizacją funkcji na podobnym poziomie do algorytmu gradientu prostego. Niewatpliwa zaleta strategii ewolucyjnej jest fakt, że nie wymaga ona tego, aby funkcja minimalizowana była różniczkowalna czego nie możemy powiedzieć o wspomnianym poprzednio gradientie prostym. Z doświadczeń wynika, że dobranie odpowiednich parametrów jak α i σ może mieć znaczny wpływ na szybkość konwergencji naszego algorytmu.