Wrocław
University
of Science
and Technology

Metody Systemowe i Decyzyjne L

Optymalizacja numeryczna

Piotr Kawa
W4N, K46
sem. letni 2023/24

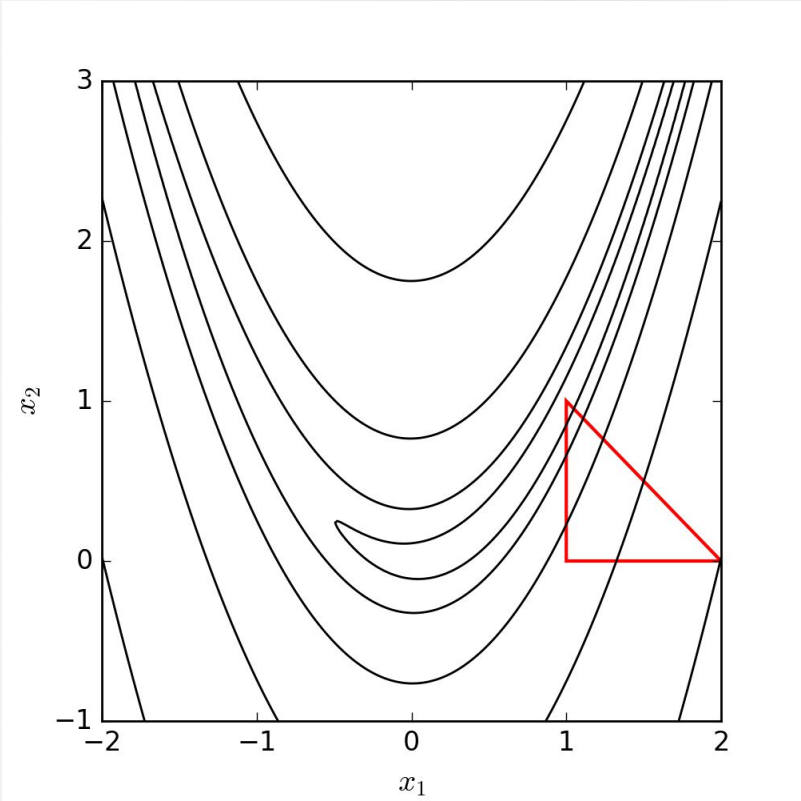
HR EXCELLENCE IN RESEARCH

Algorytm Nelder-Mead

Algorytm optymalizacji Nelder-Mead [1] to popularna metoda bezgradientowa służąca do znajdowania minimum funkcji wielu zmiennych. Jest to algorytm iteracyjny, który operuje na zbiorze punktów nazywanym "simpleksem" (wielościanem w przestrzeni parametrów funkcji).

[1] John A. Nelder and Roger Mead *A simplex method for function minimization.*, Computer Journal, 7:308–313, 1965.

Algorytm Nelder-Mead



Algorytm Nelder-Mead zastosowany na funkcji Rosenbrocka [2].

[2] https://en.wikipedia.org/wiki/Nelder%E2%80%93Mead_method#/media/File:Nelder-Mead_Rosenbrock.gif

Algorytm Nelder-Mead

Parametry algorytmu:

- **Alpha** - parametr kontrolujący odbicie punktu. Określa "długość" odbicia punktu. Przyjmuje wartości większe od 0. Domyślnie jest równy 1.
- **Beta** - parametr kontrolujący skurcz. Określa "długość" skurczenia punktu. Mieści się w przedziale (0, 1). Domyślnie jest równy 0.5.
- **Gamma** - parametr kontrolujący rozszerzenie. Określa "długość" rozszerzenia odbitego punktu. Przyjmuje wartości większe od 1. Domyślnie jest większy równy 2.
- **Sigma** - parametr kontrolujący zwężenie. Określa "długość" zmniejszenia simpleksu. Mieści się w przedziale (0, 1). Domyślnie jest równy 0.5.

Wrocław University of Science and Technology

Zadanie

- Zadaniem jest implementacja brakujących funkcji dotyczących algorytmu Nelder-Mead. Zadanie zostało rozwiązane częściowo w pliku *optimization_nelder_mead.py*.
- Jako pomoc wykorzystaj skrypt *test_optimization_nelder_mead.py*, który zawiera testy jednostkowe do funkcji do zaimplementowania.
- Spełnienie wszystkich testów jednostkowych jest wymogiem dla pełnej liczby punktów.

Algorytm Nelder-Mead

Funkcje do zaimplementowania to:

- *reflect* - dla zadanego *alpha* zwraca odbicie punktu względem centroidu,
- *expand* - dla zadanego *gamma* zwraca rozszerzony punkt względem centroidu,
- *contract* - dla zadanego *beta* zwraca skurczoną wersję punktu względem centroidu,
- *shrink* - dla zadanego *sigma* redukuje rozmiar elementów simplexu z pominięciem najlepszego elementu.

Pliki do wysłania

Rozwiązane zadanie zawierać powinno następujące pliki:

1) *optimization_nelder_mead.py*,
2) *test_optimization_nelder_mead.py*.

Wrocław University of Science and Technology

Powodzenia!