

ALHE

GREY WOLF OPTIMIZER

Specyfikacja wstępna

Andrzej Dackiewicz
Mateusz Jarzemski

1. Opis metody

- Ustalamy wielkość populacji. Ustalamy ilość iteracji algorytmu. Obliczamy parametry A, C.

$$A = 2a * r1 * a * D$$

$$C = 2 * r2 * E$$

Gdzie a zmniejsza się liniowo od 2 do 0 w kolejnych iteracjach a r1 i r2 to losowe wektory o wartościach od 0 do 1. D i E parametry podane przez użytkownika(domyślnie 1).

- Na początku losowane będą losowe punkty z dziedziny problemu.
- Wyliczamy wartości funkcji celu dla poszczególnych punktów.
- Wybieramy 3 najlepsze punkty, są to nasze wilki alfa, beta, delta (dla których wyliczone wartości funkcji celu są najlepsze). Te 3 wilki nie zmieniają swojego położenia dopóki inne wilki nie przejmą ich roli (alfa, beta, delta), a one nie staną się przez to wilkami omega.
- Pozostałe punkty stają się wilkami omega. Zapisujemy Najlepszy dotychczasowy wynik.
- Wyliczamy nowe położenie dla wilków omega. (na podstawie położenia 3 najlepszych punktów)

$$Dalfa = |C1 * Xalfa - X|$$

$$Dbeta = |C2 * Xbeta - X|$$

$$Ddelta = |C3 * Xdelta - X|$$

$$X1 = Xalfa - A1 * Dalfa$$

$$X2 = Xbeta - A2 * Dbeta$$

$$X3 = Xdelta - A3 * Ddelta$$

$$X(t+1) = (X1 + X2 + X3) / 3$$

Gdzie:

Xalfa - Pozycja wilka alfa

Xbeta - Pozycja wilka beta

Xdelta - pozycja wilka delta

A1,2,3 C1,2,3 - wartości parametrów A i C w kolejnych losowaniach r1, r2

X - aktualna pozycja przemieszczanego wilka

- Losujemy nowe wektory r1, r2. Wyliczamy parametry A, C.
- Dopóki nie spełniony został warunek zakończenia algorytmu powracamy do wyliczania wartości dla poszczególnych punktów. (Krok 3)
- Warunkiem zakończenia algorytmu jest wykonanie określonej ilości iteracji (n).

2. Parametry programu:

D - Parametr podawany wraz z uruchomieniem programu. Jego wartość będzie miała wpływ na czynność eksploracji.. Zwiększenie tego parametru spowoduje zwiększenie skłonności do eksploracji. Jego wpływ zmniejsza się z każdą kolejną iteracją.

E - Parametr jest podobny do parametru D ale jego wpływ jest niezależny od iteracji algorytmu.

maxIt - Jest to ilość iteracji po których algorytm ma zakończyć działanie.

vecMin - Jest to wektor wartości minimalnych dla poszczególnych pozycji wilków.

vecMax - Jest to wektor wartości maksymalnych dla poszczególnych pozycji wilków.

wolvesNumber - Jest to liczba wilków dla której algorytm ma działać.

3. Raport z przeprowadzania eksperymentów

Wybrane funkcje do testowania algorytmu:

- Sphere function - funkcja wielowymiarowa, n to liczba wymiarów

$$f(\mathbf{x}) = \sum_{i=1}^n (x_i^2)$$

- Rosenbrock function - wielowymiarowa, n to liczba wymiarów

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$$

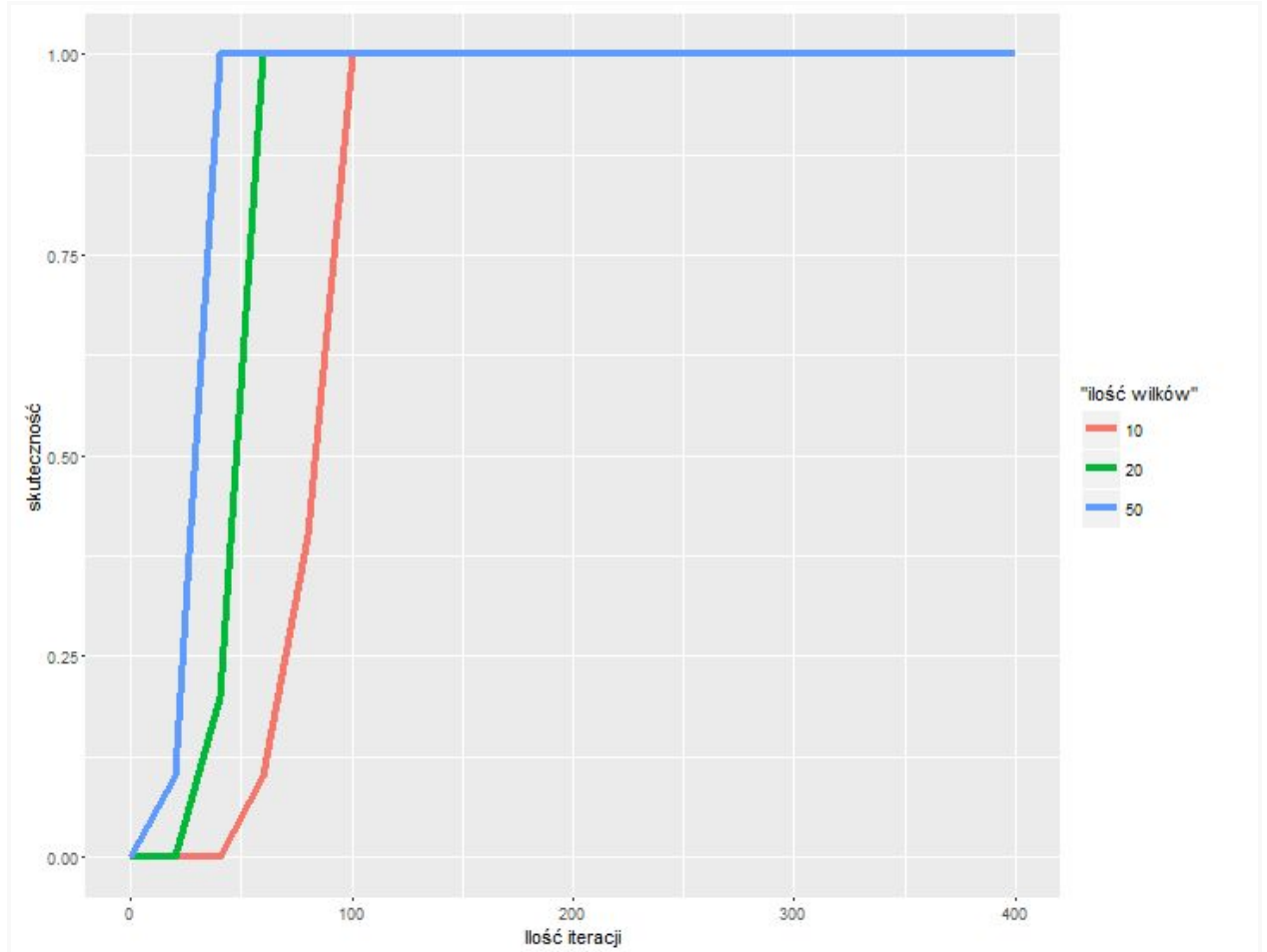
\mathbf{x} to wektor n wymiarowy

- Easom function - funkcja 2 wymiarowa

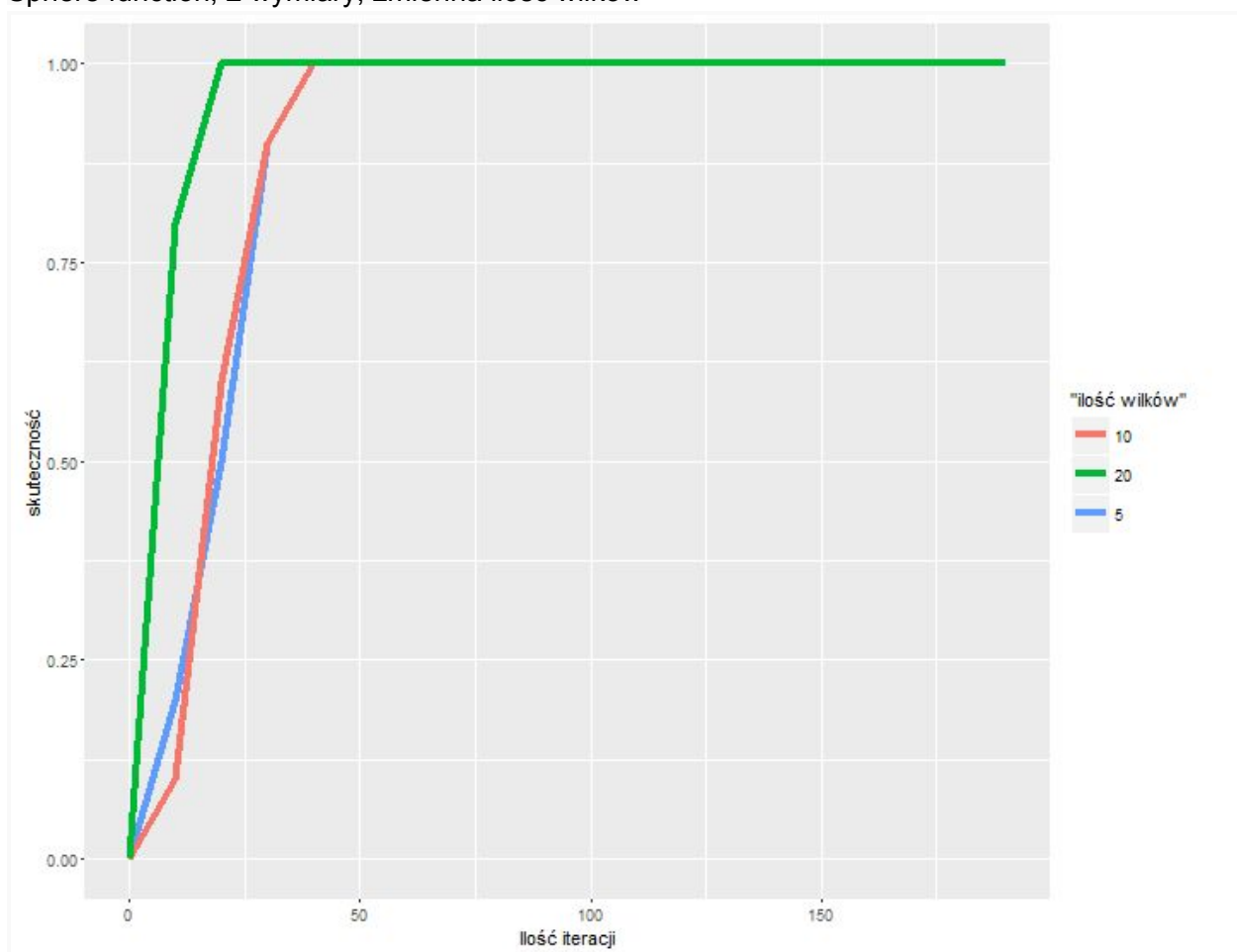
$$f(x,y) = -\cos(x)\cos(y)\exp(-((x-\pi)^2 + (y-\pi)^2))$$

Dla każdej mierzonej ilości iteracji test był powtarzany 10 razy. Uznawane były wartości których ostateczny wynik był odległy od oczekiwanego o mniej niż 10^{-7}

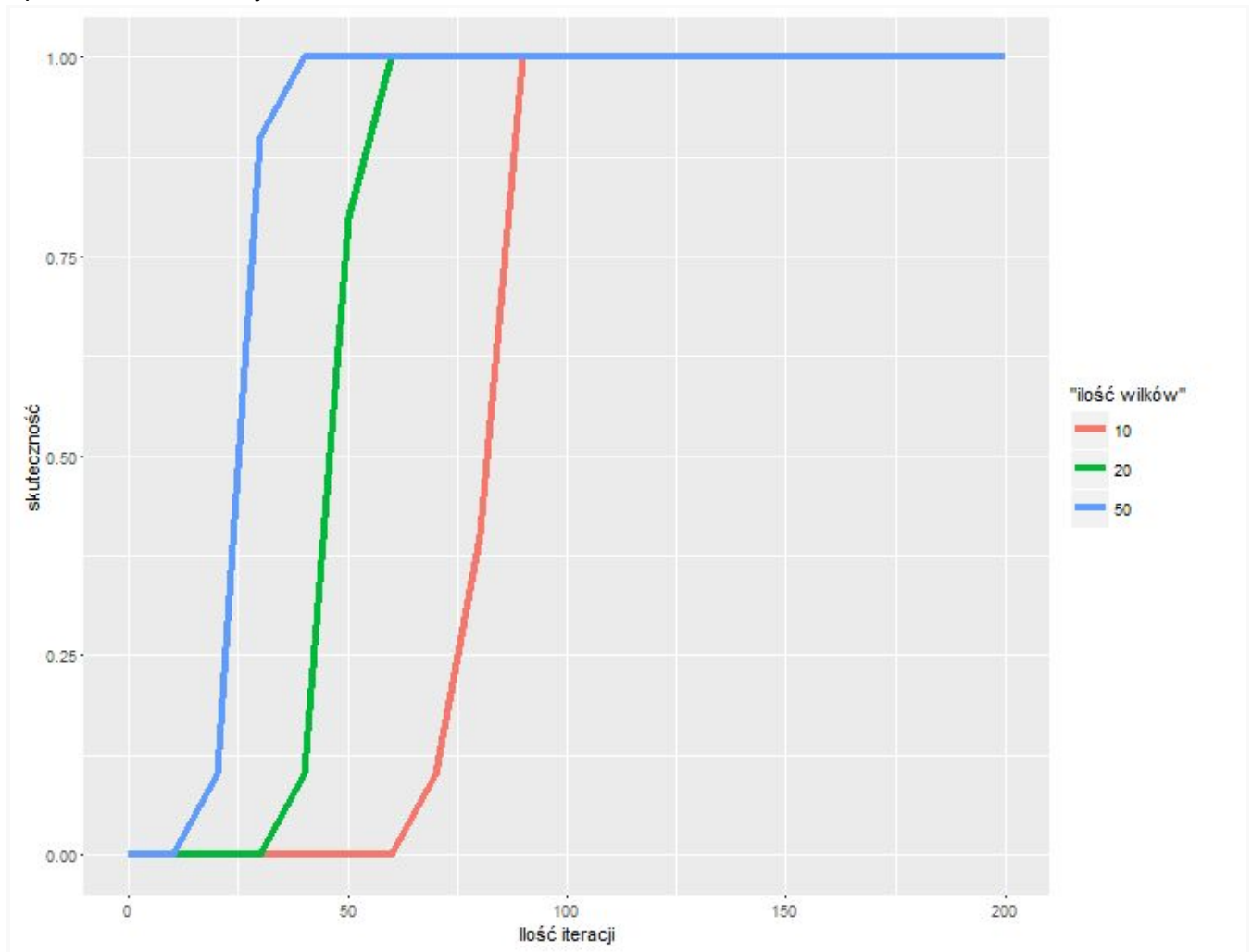
Rosenbrock function, 5 wymiarów, zmienna ilość wilków



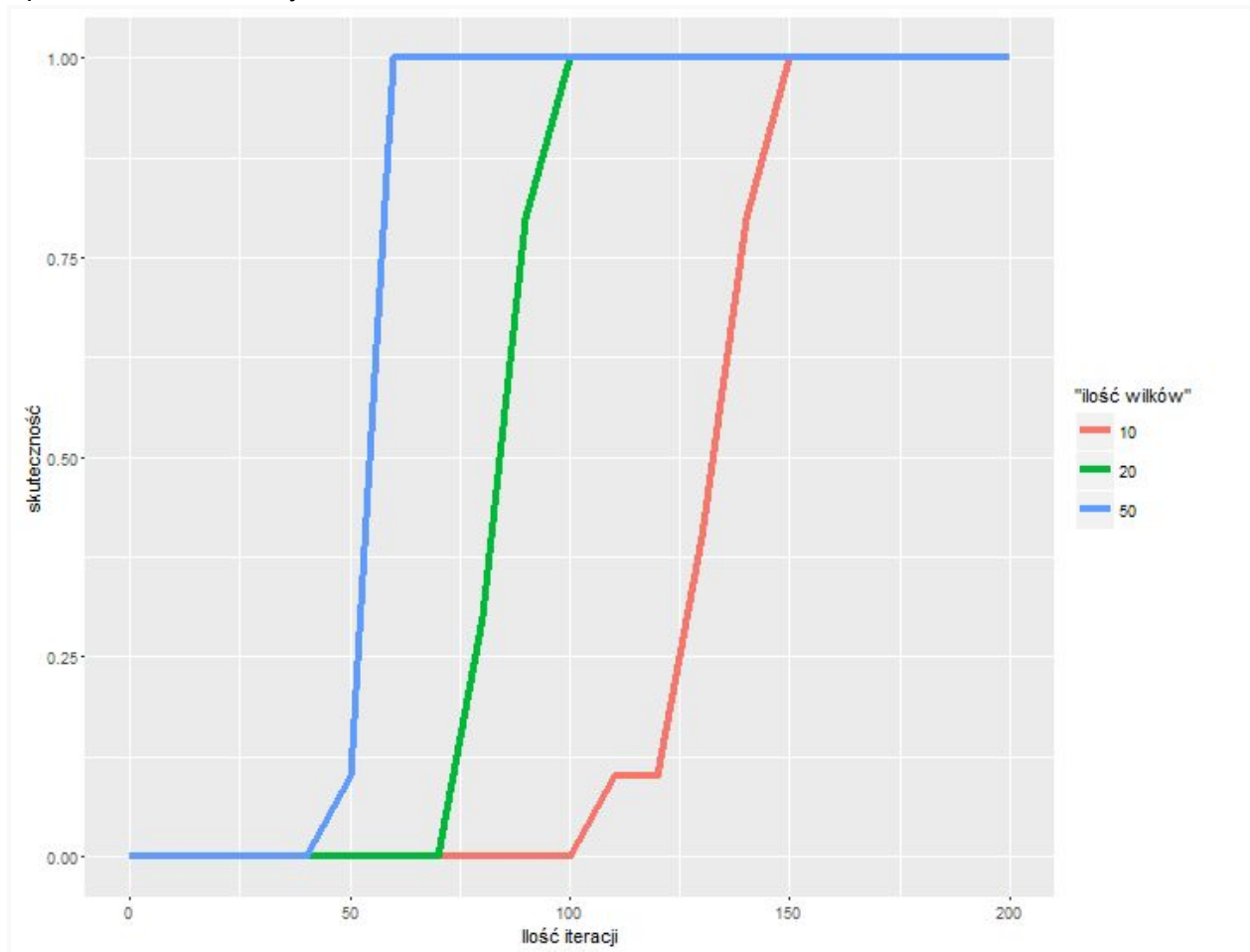
Sphere function, 2 wymiary, zmienna ilość wilków



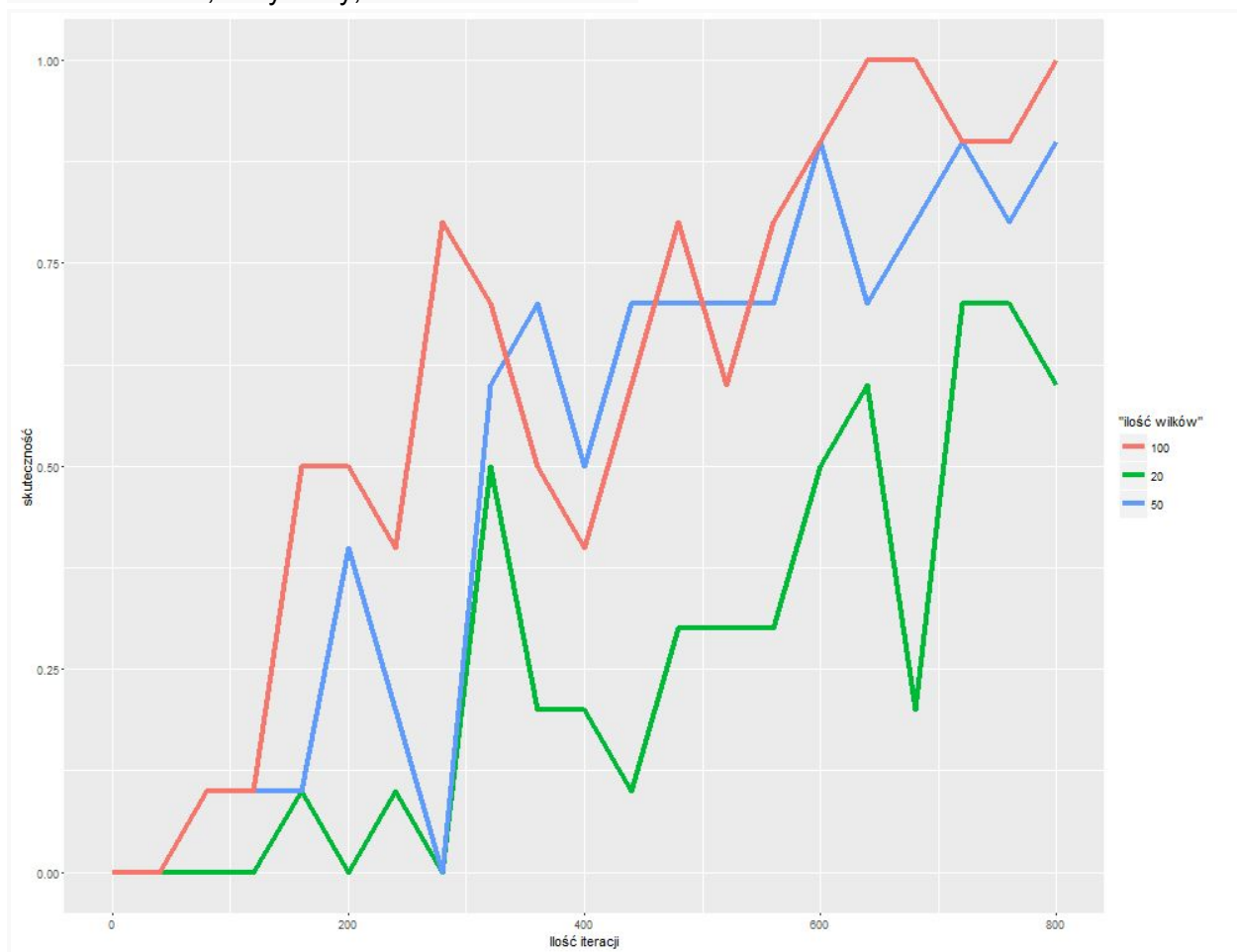
Sphere function, 5 wymiarów, zmienna ilość wilków



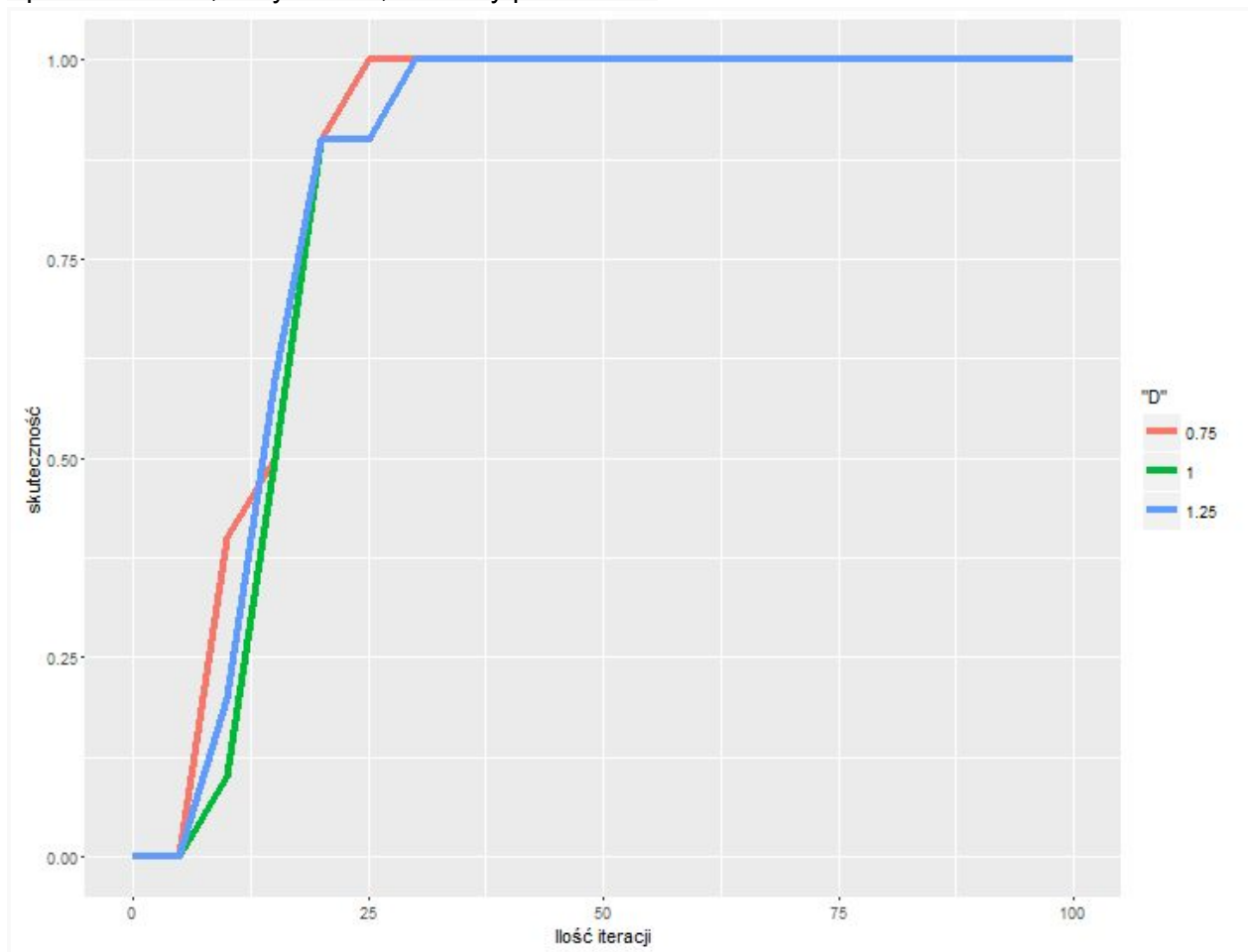
Sphere function, 10 wymiarów, zmienna ilość wilków



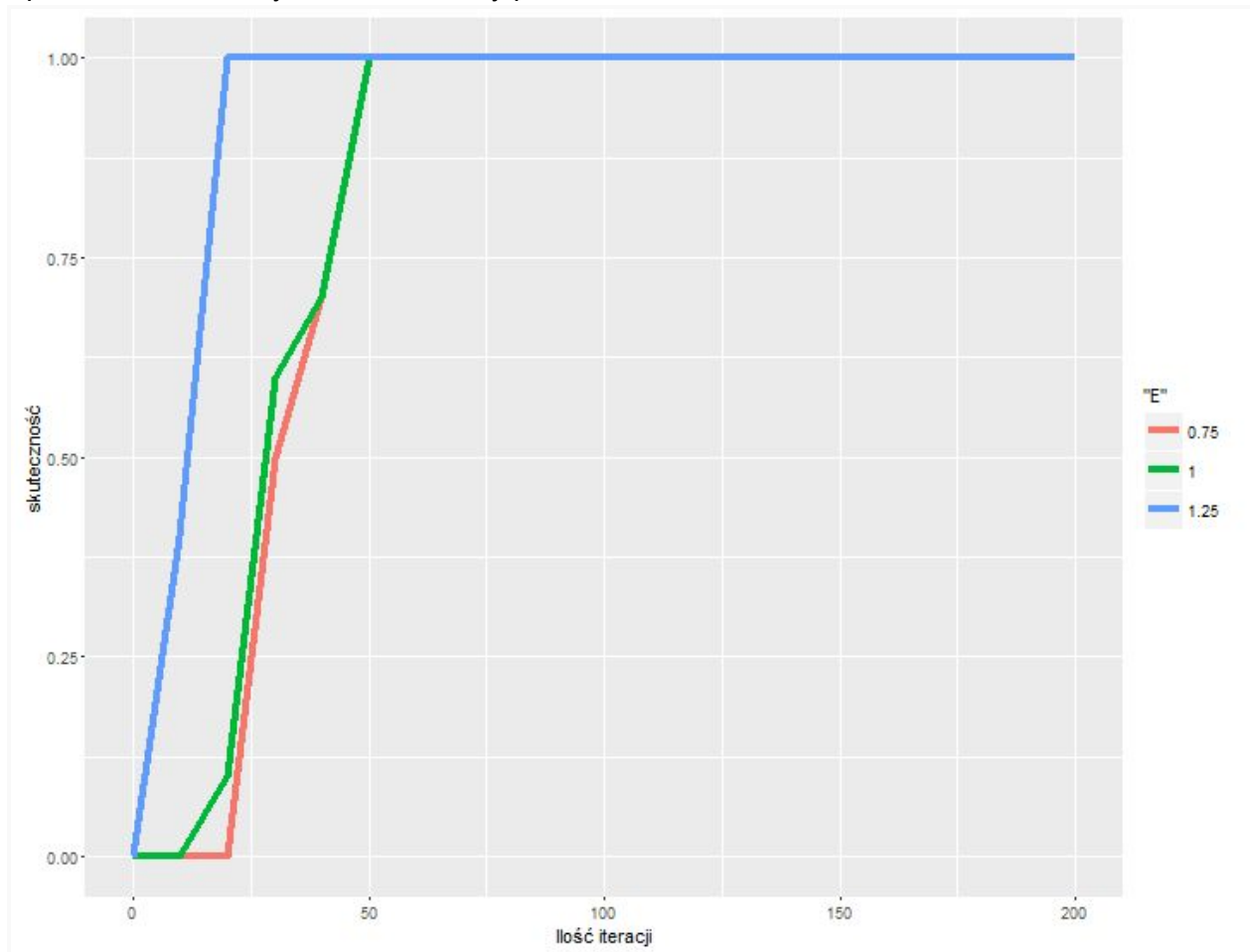
Easom function, 2 wymiary, zmienna ilość wilków



Sphere function, 2 wymiarów, zmienny parametr D



Sphere function, 2 wymiarów, zmienny parametr E



Analiza wyników:

Dla wszystkich przeprowadzonych testów działania programu możemy zauważyć, że dla pewnej ilości iteracji skuteczność funkcji osiąga 1. Czyli Grey Wolf Optimizer dla odpowiednio ustawionych paramterów poprawnie wykonuje swoją rolę co oznacza, że program działa prawidłowo.