1 Opis projektu

Celem projektu jest stworzenie metody simplexu Neldera i Meada . Metoda ta polega na utworzeniu w przestrzeni E^{n+1} *n*-wymiarowego simplexu o *n+1* wierzchołkach tak, aby można było go wpisać w powierzchnię reprezentującą badaną funkcję celu. Jednowymiarowym simplexem jest odcinek o dwóch wierzchołkach, simplexem dwuwymiarowym jest trójkąt i ogólnie simplexem *n*-wymiarowym o *n+1* wierzchołkach jest zbiór wszystkich punktów określonych przez wektory:

$$x = \sum_{j=1}^{n+1} x_j S_j \quad \text{przy czym} \quad \sum_{j=1}^{n+1} x_j = 1 \quad \text{oraz} \quad x_j \geq 0$$

czyli jest to wielościan o *n+1* wierzchołkach rozpiętych na *n+1* wektorach bazowych (S_j). Współrzędne punktów simplexu oznaczono jako x_j. Na początku procedury wylicza się współrzędne punktów wierzchołkowych simplexu P_j (dla $j = 1 .. n+1$) przy założeniu pewnej odległości między tymi wierzchołkami (czyli kroku). W następnych iteracjach dokonuje się przekształceń simplexu aż odległość pomiędzy jego wierzchołkami w pobliżu poszukiwanego ekstremum będzie mniejsza od założonej dokładności obliczeń *e*. To właśnie zostało przyjęte jako kryterium zbieżności dla tej metody.

2. Algorytm

1) Obliczenie wartości funkcji celu w punktach wierzchołkowych simplexu $F_j = f(P_j)$ dla $j = 1 .. n+1$
2) Wyznaczenie *h* i *L* takich, że $f(P_h)$ = max oraz $f(P_L)$ = min spośród zbioru F_j.
3) Obliczenie środka symetrii simplexu P'.
4) Odbicie P* punktu P_h względem P'.
5) Obliczenie wartości funkcji $F_s = f(P')$ oraz $F_o = f(P^*)$.

Jeśli F_o < min, to:
6) Obliczenie P** (ekspansja) i wartości funkcji $F_e = f(P^{**})$.
7) Jeśli F_e < max, to podstawiamy $P_h = P^{**}$, w przeciwnym przypadku podstawiamy $P_h = P^*$.
8) Powtórzenie procedury od kroku 1), o ile nie jest spełnione kryterium na minimum.

Jeśli F_o > min, to:
6) Jeśli $F_o >= f(P_j)$ dla $j = 1 .. n+1$ (z pominięciem $j = h$) i F_o >= max, przejście do następnego kroku. Jeśli zaś F_o < max, to podstawiamy $P_h = P^*$.
7) Wykonanie kontrakcji P*** punktu P_h względem P'.
8) Obliczenie $F_k = f(P^{***})$.
9) Jeżeli F_k >= max to wykonujemy redukcję simplexu wg wzoru: $P_j = 0{,}5*(P_j + P_L)$ dla $j = 1 .. n+1$. Jeżeli natomiast F_k < max, to podstawiamy $P_h = P^{***}$ i przechodzimy do kroku 11).
10) Jeśli $F_o < f(P_j)$ dla $j = 1 .. n+1$ (z pominięciem $j = h$), podstawiamy $P_h = P^*$.
11) Powtórzenie procedury od kroku 1), jeśli nie zostało spełnione kryterium na minimum.

Oznaczenia:

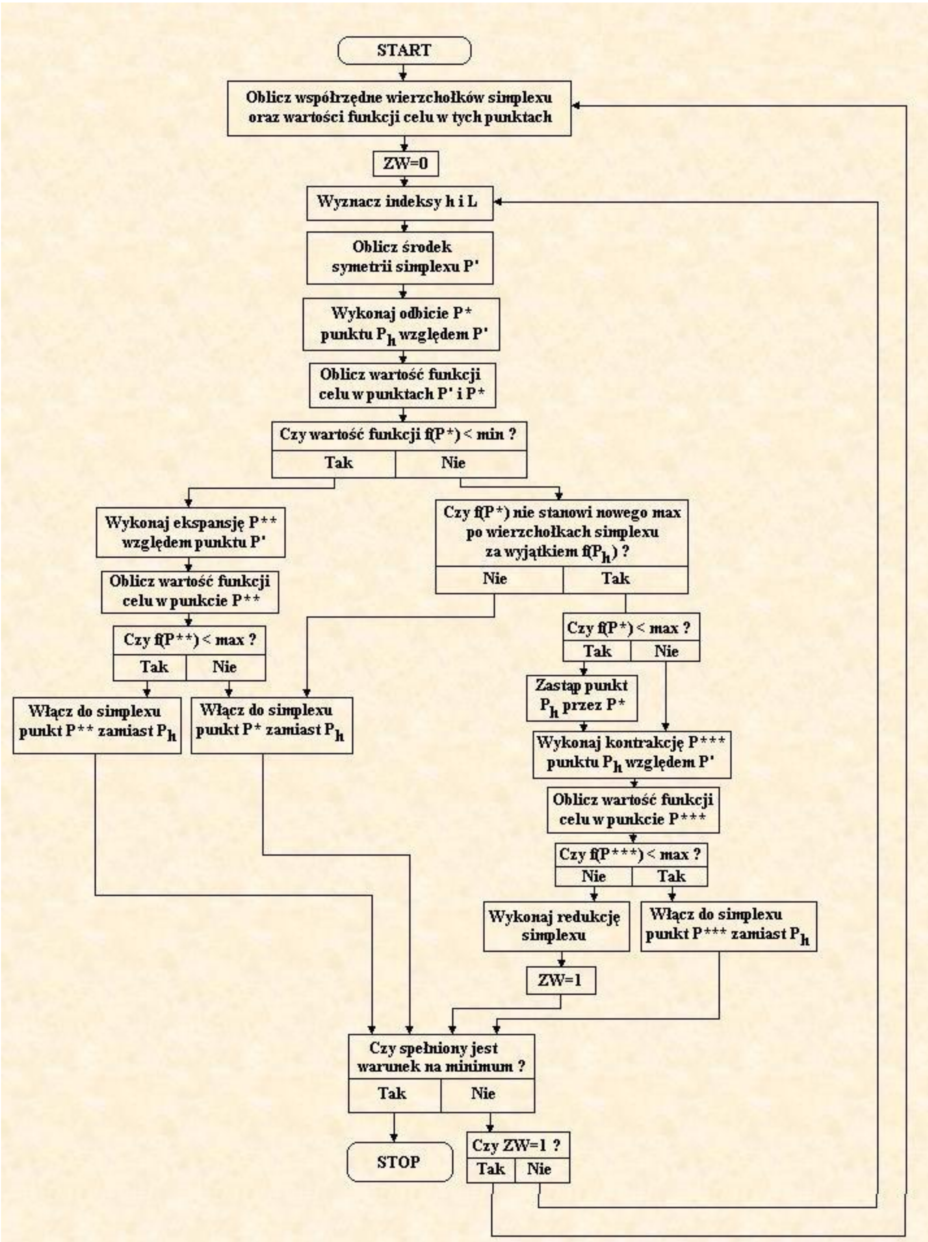
x_0 - punkt startowy
e - wymagana dokładność obliczeń
P_h - wybrany punkt wierzchołkowy simplexu spośród *n+1* wierzchołków P_i, w którym wartość badanej funkcji osiąga maksimum.
P_L - wybrany punkt wierzchołkowy simplexu spośród *n+1* wierzchołków P_i, w którym wartość badanej funkcji osiąga minimum.
P' - środek symetrii simplexu z wyłączeniem punktu P_h zdefiniowany jako:

P*- odbicie punktu

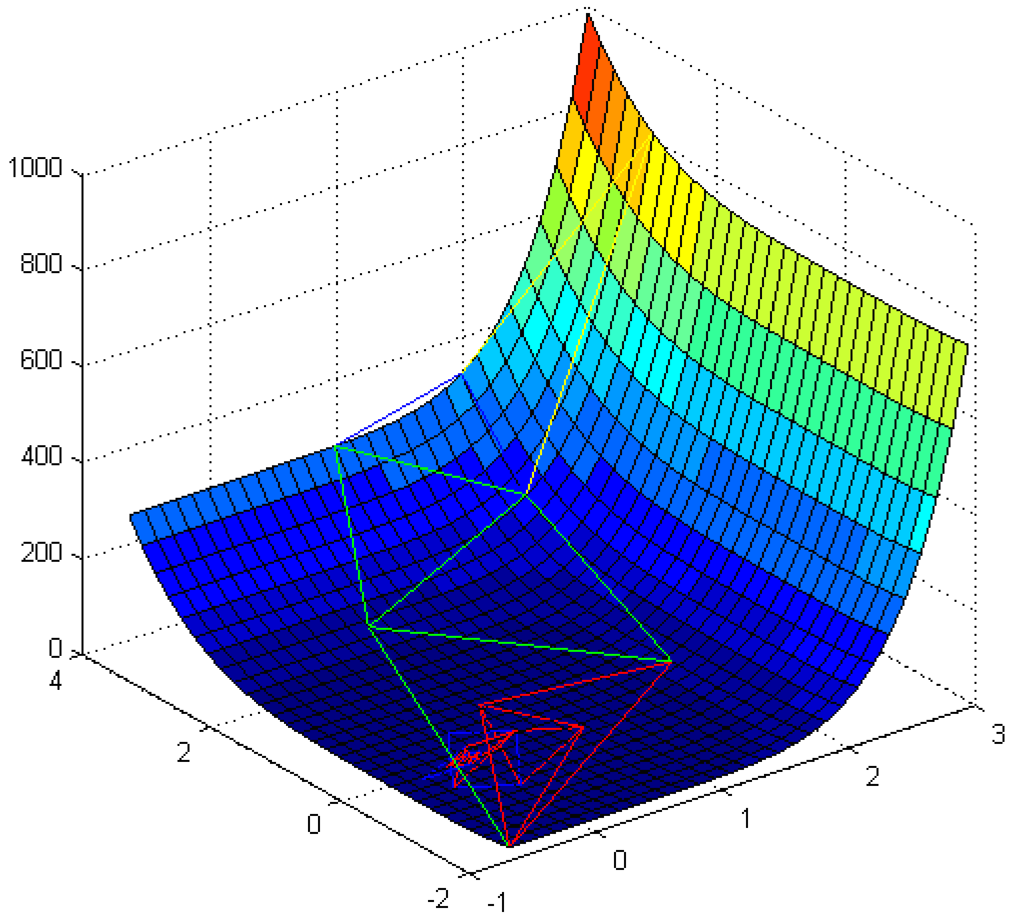
P**-ekspansja punktu

P***kontrakcja punktu

3. Schemat algorytmu



4. Przykład



$F(x, y) = x^2+y^2$, punkt początkowy (2, 3)

po 11 iteracjach znaleziono przybliżony pkt. minimalny: (-0,08, -0,07)

legenda: żółty -początkowy symplex, niebieski -odbicie, zielony -ekspansja, czerwony -kontrakcja

5. Harmonogram

1)przygotowanie funkcji odbicia, ekspansji i kontrakcji.

2)przygotowanie jednowymiarowego simpleksu

3)przygotowanie dwuwymiarowego simpleksu (połowa projektu)

4)przygotowanie n-wymiarowego simpleksu

5)testowanie funkcji

6.literatura i inne źródła

- http://www.scholarpedia.org/article/Nelder-Mead_algorithm
- https://www.researchgate.net/publication/324084428_Less_is_more_Simplified_Nelder-Mead_method_for_large_unconstrained_optimization
- https://en.wikipedia.org/wiki/Nelder%E2%80%93Mead_method
- http://www.kmg.zut.edu.pl/opt/wyklad/bezgrad/simplex.html?fbclid=IwAR0fo4rz1dYzl_7MzvuruBNySkqW5etm1KVE0XYFHo2eP5ao7hXgkg0POX8