Zewnętrzna funkcja kary		
Dominik Wróbel	**07 V 2018**	**Pon 08:00, s. 111**
Numery zadań		1, 2

Spis treści

1 Cel ćwiczenia

Celem ćwiczenia jest rozwiązanie zadań optymalizacyjnych z wykorzystaniem podejścia opartego o zewnętrzną funkcje kary. Metoda ta pozwala na rozwiązywanie zadań programowania nieliniowego, polega na generowaniu rozwiązań zadania bez ograniczeń i wprowadzaniu kary za przekroczenie tych graniczeń przy pomocy tzw. zewnętrznej funkcji kary.
Działanie funkcji kary można w uproszczeniu opisać jako funkcję, która ma wartość 0 jeśli dana zmienna $x \in D$, daje natomiast wartość większą od zera gdy znalezione optimum nie zawiera się w rozwiązaniach dopuszczalnych.

2 Przebieg ćwiczenia

2.1. Zadanie 1 - Minimalizacja funkcjonału z ograniczeniem równościowym

W zadaniu rozważany jest funkcjonał postaci

$$f(x_1, x_2) = x_1^2 + x_2^2 \to min$$

przy danym ograniczeniu równościowym (numer na liście n = 16)

$$x_2 = n = 16$$

Nie trudno zauważyć, że minimum tego funkcjonału przy przyjętym ograniczeniu jest osiągane dla $x = \begin{bmatrix} 0 & 16 \end{bmatrix}$. Zadanie najpierw rozwiązano przy użyciu funkcji matlaba *fmincon*. Funkcja ta znajduje rozwiązanie optymalne przy określonych ograniczeniach nierównościowych. Rozwiązanie zadania przy pomocy tej funkcji odpowiada rozwiązaniu Zadania Programowania Nieliniowego (ZPN), które opisuje model matematyczny :

$$f(x) \to min$$

$$x \in X \subset \mathbb{R}^n$$

$$X = \{x : g_i(x) \leq 0, \quad i = 1, \dots, m\}$$

$$f(x) : \mathbb{R}^n \to \mathbb{R}^1$$

$$g_i(x) : \mathbb{R}^n \to \mathbb{R}^1 \quad \forall i = 1, \dots, m$$

Następnie zadanie rozwiązano przy użyciu kilku iteracji funkcją *fminunc*. Podejście to odpowiada metodzie zewnętrznej funkcji kary, polega na generowaniu ciągu punktów będących rozwiązaniami zadania minimalizacji bez ograniczeń (ZMB). W zadaniu rozważana jest funkcja

$$P(x,k) = f(x) + k^j \cdot \sum_{i=1}^{m} max[0, g_i(x)]^2$$

z kwadratową funkcją kary:

$$\phi(x,k) = k^j \cdot \sum_{i=1}^{m} max[0, g_i(x)]^2$$

W ZPN występują ograniczenia nierównościowe z nierównością słabą, na takich ograniczeniach działa też funkcja matlaba *fmincon*, dlatego konieczna jest zamiana ograniczenia równościowego na dwa ograniczenia nierównościowe w następując sposób:

$$x_2 = 1 \iff \begin{cases} x_2 \leq 1 \\ x_2 \geq 1 \end{cases}$$

W implementacji umieszczono następujące wzory : Funkcja celu:

$$f(x) = x_1^2 + x_2^2 \rightarrow min$$

Ograniczenia :

$$g_1(x) = 1 - x_2 \leq 0$$
$$g_2(x) = x_2 - 1 \leq 0$$

Dla powyższych ograniczeń funkcja kary z zadania spełnia założenia konieczne do spełnienia przez funkcję kary :

- $\phi(x, k^j) = 0 \quad \forall x \in X$
- $\phi(x, k^j) > 0 \quad \forall x \notin X$
- $\phi(x, k^j) \rightarrow \infty \quad dla \quad k \rightarrow \infty \quad \forall x \in X$
- $\phi(x, k^j) > \phi(x, k^{j+1}) \quad \forall x \notin X$

gdzie

$$\phi(x,k) = k^j \cdot max[0, (1 - x_2)]^2 + k^j \cdot max[0, (x_2 - 1)]^2$$
$$X = \{(x_1, x_2) : x_2 = 16\}$$

Implementacje powyższych obliczeń przedstawiają Listing 1 i Listing 2.

```
% znalezienie minimum z ograniczeniami

x0 = [ 2, 1];

fun = @(x)x(1)^2 + x(2)^2;
A = [0,-1;
     0  1];

b = [ -1;
       1];

[x_opt, f_opt] = fmincon(fun,x0,A,b);
```

Listing 1: Znalezienie minimum z ograniczeniami (fmincon)

```

% znalezienie minimum przy uzyciu zewnetrznej f. kary

x0 = [ 2, 1];
max_iter = 10;
points = [ 0 0 ];
f_values = [ 0 ];
diff_table = [ 0 ];

k=1;
k_table = [ k ];
fun_1 = @(x)x(1)^2 + x(2)^2 + k*( ( max( 0, 1-x(2) ) ) ) +
k*( ( max( 0, x(2)-1 ) ) );

for n = 1:max_iter
    [x_opt_k, fval] = fminunc(fun_1,x0);

    diff = abs( f_opt - fval );
    diff_table = [ diff_table; diff ];
    f_values = [ f_values ; fval ];
    points = [ points ; x_opt_k ];

    x0 = x_opt_k;
    k= k*1.15;
    k_table = [ k_table ; k ];
    fun_1 = @(x)x(1)^2 + x(2)^2 + k*( ( max( 0, 1-x(2) ) ) ) +
k*( ( max( 0, x(2)-1 ) ) );

end
```

Listing 2: Znalezienie minimum przy użyciu zewnętrznej funkcji kary (fminunc)

Po znalezieniu wartości wyniki zostały przedstawione na wykresie, realizujący to kod przedstawia Listing 3

```

% Rysowanie punktow

[x1, x2] = meshgrid( -3:0.1:13, -1:0.1:17 );
rys_war = x1 .^ 2 + x2 .^ 2;
contour(x1, x2, rys_war, ...
[0.1, 1, 2, 10, 20, 30, 50, 100, 200, 300 ], 'ShowText', 'on');
hold on;
plot(points(:,1),points(:,2),'-o');
xlabel('x1') % x-axis label
ylabel('x2') % y-axis label
```

Listing 3: Stworzenie wykresu z wynikami

Wyniki wraz z poziomicami przedstawia Rysunek 1 zkara_1.jpg

Rysunek 1: Znalezione przez metodę zewnętrznej funkcji kary punkty w kolejnych iteracjach.

Znalezione przez metodą rozwiązania wraz z numerem iteracji przedstawiono w poniższej tabeli.

zkara_2.jpg

Rysunek 2: Tabela z wynikami działania metody zewnętrznej funkcji kary.

Porównując otrzymany w ostatniej iteracji wynik z analitycznie wyznaczonym minimum stwierdzono, że błąd jest niewielki, rzędu 6 miejsca po przecinku dla x_1, narzucone ograniczenie $x_2 = 16$ została natomiast osiągnięte w procesie szukania minimum.

Z wyników można wyciągnąć wniosek, że im 'dalej' od obszaru dopuszczalnego znajduje się rozwiązanie tym większa wartość funkcji kary, a tym samym większa zmiana rozwiązania w kolejnej iteracji, im bliżej wartości optymalnej tym zmiany i wartość funkcji kary są mniejsze.

2.2. Zadanie 2 - Minimalizacja odległości od punktu

W zadaniu należy zminimalizować odległość od punktu $(0, 0)$ przy danych ograniczeniach. W pierwszej kolejności napisano funkcję celu odpowiadającą zagadnieniu. W tym celu wykorzystano metrykę euklidesową:

$$f_1(x_1, x_2) = \sqrt{(x_1 - 0)^2 + (x_2 - 0)^2}$$

Wyrażenie z pierwiastkiem będzie mieć wartość minimalną gdy wyrażenie pod pierwiastkiem będzie mieć wartość minimalną dlatego w zadaniu rozważana jest funkcja celu :

$$f(x_1, x_2) = (x_1 - 0)^2 + (x_2 - 0)^2$$

Następnie sprawdzono poprawność postawionego zadania. Zadanie ma dwa ograniczenia równościowe :

$$g_1(x) = x_1 + x_2 - 1 = 0$$

$$g_2(x) = x_1 + x_2 - 2 = 0$$

Wyznaczając z pierwszego z tych ograniczeń wartość x_1: $x_1 = 1 - x_2$ i wstawiając do drugiego równania otrzymamy sprzeczność, dlatego **zadanie jest źle postawione**.

Od razu można więc stwierdzić, że zbiór rozwiązań dopuszczalnych jest zbiorem pustym, co będzie pociągać za sobą błąd w działaniu algorytmu, ponieważ funkcja kary zawsze będzie dawać wartość niezerową. Aby to udowodnić przetestowano działanie algorytmu dla zadania źle postawionego.

Ograniczenia równościowe zamieniono na ograniczenia nierównościowe analogicznie jak w zadaniu 1. Funkcja celu :

$$f(x) = (x_1 - 0)^2 + (x_2 - 0)^2 \to min$$

Ograniczenia :

$$g_1(x) = x_1 + x_2 - 1 \leq 0$$

$$g_2(x) = -x_1 - x_2 + 1 \leq 0$$
$$g_3(x) = x_1 + x_2 - 2 \leq 0$$
$$g_4(x) = -x_1 - x_2 + 2 \leq 0$$

Implementację metod przeprowadzono tak jak w Zadaniu 1, zmieniona została jedynie funkcja kary i celu, co przedstawia Listing 4.

```

fun_1 = @(x)x(1)^2 + x(2)^2 + k*( ( max( 0, x(1)+x(2)-1 )^2 )  ) + ...
k*(  ( max( 0, -x(1)-x(2)+1 )  )^2 ) + ...
k*( ( max( 0, x(1)+x(2)-2 )^2 )  ) +  k*(  ( max( 0, -x(1)-x(2)+2 )^2 ) ) ;
```

Listing 4: Funkcja celu dla zadania 2

Wyniki wraz z poziomicami przedstawia Rysunek 3

zkara_3.jpg

Rysunek 3: Znalezione przez metodę zewnętrznej funkcji kary punkty w kolejnych iteracjach.

Znalezione przez metodą rozwiązania wraz z numerem iteracji przedstawiono w poniższej tabeli.

zkara_4.jpg

Rysunek 4: Tabela z wynikami działania metody zewnętrznej funkcji kary.

Metoda znajduje pewne optimum, ale nie spełnia ono założeń, co łatwo sprawdzić poprzez podstawienie otrzymanego minimum do ograniczeń. Metoda nie jest więc odporna na zadania w których obszar dopuszczalny jest zbiorem pustym.

Analityczne rozwiązanie tego zadania nie dałoby również wyniku zgodnego z ograniczeniami więc trudno w tym przypadku o porównanie z rzeczywistą wartością minimum.

3 Wnioski końcowe

Metoda zewnętrznej funkcji kary pozwala na rozwiązywanie zadań optymalizacji z ograniczeniami poprzez zastosowanie funkcji, która daje wartość niezerową gdy znalezione rozwiązanie nie znajduje się w obszarze dopuszczalnym. Pomimo to, otrzymany wynik należy zawsze zweryfikować, ponieważ nie zawsze metoda daje wynik zgodny z ograniczeniami, co można zauważyć w zadaniu 1 (potrzebna jest odpowiednia liczba iteracji aby osiągnąć wynik zwierający się w zbiorze dopuszczalnym) oraz w zadaniu 2 (metoda znajduje wynik, który nie zawiera się w zbiorze dopuszczalnym dla dowolnej liczby iteracji).

Nie bez znaczenia dla jakości wykonywania metody są parametry algorytmu, najbardziej znaczącym wydaje się być parametr k, który decyduje o tym jak duża kara jest nałożona na daną funkcję za wyjście poza obszar dopuszczalny. Z moich obserwacji wynika, że wartość tą najlepiej dobierać eksperymentalnie, ponieważ zasada, że im większa wartość tym lepiej nie zawsze działa dla wszystkich funkcji.