Wewnętrzna funkcja kary		
Dominik Wróbel	**14 V 2018**	**Pon 08:00, s. 111**
Numery zadań		A, B

Spis treści

1 Cel ćwiczenia

Celem ćwiczenia jest rozwiązanie zadań optymalizacyjnych z wykorzystaniem podejścia opartego o wewnętrzną 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. wewnętrznej funkcji kary.
W odróżnieniu od zewnętrznej funkcji kary, wewnętrzna funkcja kary osiąga rozwiązanie znajdując kolejno rozwiązania znajdujące się w zbiorze dopuszczalnym - działa od 'wnętrza' zbioru dopuszczalnego.

2 Przebieg ćwiczenia

2.1. Zadanie A - Minimalizacja funkcjonału z ograniczeniami liniowymi

2.1.1. Rozwiązania programowe bez metody wew. f. kary

W zadaniu rozważany jest funkcjonał postaci $(n = 16)$:

$$f(x_1, x_2) = \frac{1}{3}(x_1 + n + 1)^3 + x_2 + n = \frac{1}{3}(x_1 + 17)^3 + x_2 + 16 \rightarrow min$$

przy danych ograniczeniach nierównościowych

$$\begin{cases} g_1(x) = x_1 - 1 + n = x_1 + 15 \geq 0 \\ g_1(x) = x_2 + n = x_2 + 16 \geq 0 \end{cases}$$

Nie trudno zauważyć, że minimum tego funkcjonału przy przyjętych ograniczeniach jest osiągane dla $x = \begin{bmatrix} -15 & -16 \end{bmatrix}$. 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 :

$$f(x) \rightarrow min$$
$$x \in X \subset \mathbb{R}^n$$
$$X = \{x : g_i(x) \leq 0, \quad i = 1, \dots, m\}$$
$$f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^1, \quad g_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}^1 \quad \forall i = 1, \dots, m$$

W tym celu konieczne było przekształcenie ograniczeń do postaci odpowiadającej nierówności mniejszościowej słabej. Implementację powyższych obliczeń przedstawia Listing 1.

```
% znalezienie minimum z ograniczeniami
n = 16;
x0 = [ 5, 5];

fun = @(x)1/3*( x(1)+1+n )^3 + x(2) + n;

A = [-1, 0;
      0  -1];

b = [ 15;
       16];

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

Listing 1: Znalezienie minimum z ograniczeniami (fmincon)

Zgodnie z oczekiwaniami znalezione przez funkcję minimum to

$$x^* = [-15 \quad -16] \tag{1}$$

2.1.2. Rozwiązanie analityczne

Następnie zadanie rozwiązano przy użyciu kilku iteracji metodą wewnętrznej funkcji kary, która polega na generowaniu ciągu punktów będących minimum kolejnych wyznaczanych funkcji $P(x, k)$. W zadaniu rozważana jest funkcja

$$P(x, k) = f(x) + k \cdot \sum_{i=1}^{r} \frac{1}{g_i(x)} \rightarrow min$$
$$P(x, k) = \frac{1}{3}(x_1 + 17)^3 + x_2 + 16 + k\frac{1}{x_1 + 15} + k\frac{1}{x_2 + 16} \rightarrow min$$

z funkcją kary:

$$\phi(x, k) = k \cdot \sum_{i=1}^{r} \frac{1}{g_i(x)}$$
$$\phi(x, k) = k\frac{1}{x_1 + 15} + k\frac{1}{x_2 + 16}$$

Definiując dwa zbiory:

$$X = \{x : g_i(x) \geq 0; \quad i = 1, 2, \dots, r\}$$
$$X_0 = \{x : g_i(x) > 0; \quad i = 1, 2, \dots, r\}$$

$$X = \{x : x_1 \geq -15 \wedge x_2 \geq -16\}$$
$$X_0 = \{x : x_1 > -15 \wedge x_2 > -16\}$$

W przypadku z zadania spełnione są konieczne założenia:

- X_0 jest niepusty,
- $f(x)$ oraz $-g_i(x)$ są wypukłe i dwukrotnie różniczkowalne
- zbiory poziomicowe $\{x : f(x) \leq \alpha\}$ są ograniczone dla każdego skończonego α.
- Dla każdego $k > 0$, $P(x, k)$ jest ściśle wypukła

Spełnienie powyższych założeń daje gwarancję, że każda funkcja $P(x, k)$ niezależnie od k osiąga minimum na X_0 i gradient $P(x, k)$ po zmiennej x jest w tym punkcie równy 0 oraz, że ciąg punktów znajdywanych w kolejnych iteracjach jest zbieżny do minimum.

Rozwiązanie rozpoczęto od obliczenia pochodnych cząstkowych funkcji $P(x, k)$:

$$\frac{\partial P(x,k)}{\partial x_1} = (x_1 + 17)^2 - \frac{k}{(x_1 + 15)^2}$$

$$\frac{\partial P(x,k)}{\partial x_2} = 1 - \frac{k}{(x_2 + 16)^2}$$

$$\begin{cases} (x_1+17)^2 - \frac{k}{(x_1+15)^2} = 0 \\ \frac{\partial P(x,k)}{\partial x_2} = 1 - \frac{k}{(x_2+16)^2} = 0 \end{cases} \implies \begin{cases} x_1 = \sqrt{k^2+1} - 16 \\ x_2 = \sqrt{k} - 16 \vee -\sqrt{k} - 16 \end{cases}$$

Dokonując przejścia granicznego otrzymano, że:

$$\lim_{k \to 0} = \sqrt{k^2 + 1} - 16 = -15$$

$$\lim_{k \to 0} = \sqrt{k} - 16 = -16$$

Stąd wyznaczone analitycznie rozwiązania zgadza się z obliczonym programowo i wynosi :

$$x^* = [-15 \quad -16]$$

2.1.3. Rozwiązania programowe z metodą wew. f. kary

Implementacje powyższych obliczeń przedstawia Listing 2.

```
% znalezienie minimum przy uzyciu wewnetrznej f. kary

x0 = [ 5, 5];
max_iter = 20;
points = [ x0 ];
f_values = [ 0 ];
diff_table = [ 0 ];

c=20;
k=10;

k_table = [ k ];
fun_1 = @(x)1/3*( x(1)+1+n )^3 + x(2) + n + k*( 1/( x(1)+15 ) ) +...
 k*( 1/( x(2)+16 ) );
```

```

for n = 1:max_iter
    [x_opt_k, fval] = fmincon(fun_1,x0,A,b);
    %[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/c;
    k_table = [ k_table ; k ];
    fun_1 = @(x)1/3*( x(1)+1+n )^3 + x(2) + k*( 1/( x(1)+15 ) ) +...
     k*( 1/( x(2)+16 ) );

end
```

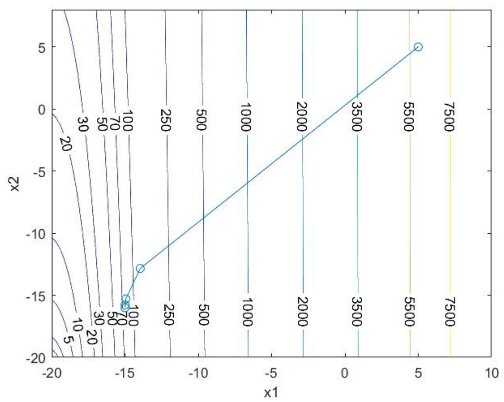
Listing 2: Znalezienie minimum przy użyciu wewnętrznej funkcji kary

Następnie znalezione przed metodę punkty zostały zaznaczone na wykresie przy pomocy kodu przedstawionego na Listingu 3

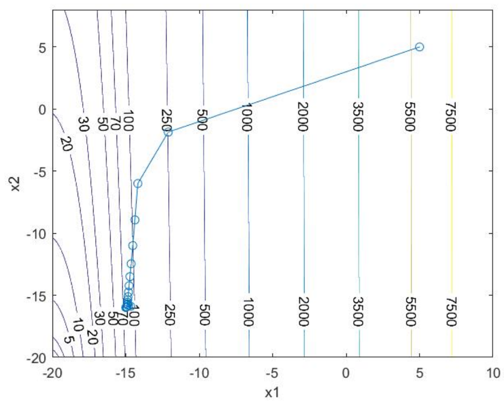
```
% Rysowanie punktow

[x1, x2] = meshgrid( -20:0.1:10, -20:0.1:8 );
rys_war = 1/3*( x1+1+n ).^3 + x2 + n;
contour(x1, x2, rys_war, [1, 2, 5, 10, 20, 30, 50, 70, 100, 250, 500, 1000, ...
 2000, 3500, 5500, 7500 ], 'ShowText', 'on');
hold on;
plot(points(:,1),points(:,2),'-o');
xlabel('x1') % x-axis label
ylabel('x2') % y-axis label
axis([-20 10 -20 8]);
```

Listing 3: Rysowanie punktów

Wyniki wraz z poziomicami przedstawia Rysuneki 1 i 2.



Rysunek 1: c = 20, k0 = 10



Rysunek 2: c = 2, k0 = 200

2.1.3 Rozwiązania programowe z metodą wew. f. kary

Znalezione przez metodą rozwiązania wraz z numerem iteracji dla c = 20 przedstawiono w poniższej tabeli.

k	x1	x2	f(x)
200	5	5	3570,3
100	-12,1087	-1,85786	136,4651
50	-14,1789	-6	-476,36
25	-14,3786	-8,92891	-412,466
12,5	-14,5249	-11	-342,006
6,25	-14,633	-12,4645	-272,833
3,125	-14,7131	-13,5	-209,709
1,5625	-14,7726	-14,2322	-155,245
0,78125	-14,8166	-14,75	-110,56
0,390625	-14,8489	-15,1161	-75,7581
0,195313	-14,8717	-15,375	-50,2461
0,097656	-14,8863	-15,5581	-32,9637
0,048828	-14,8919	-15,6875	-22,5333
0,024414	-14,8826	-15,779	-17,3662
0,012207	-14,8062	-15,8437	-15,7362
0,006104	-14,6676	-15,8895	-15,7301
0,003052	-14,9272	-15,9219	-15,3483
0,001526	-14,9727	-15,9448	-13,0004
0,000763	-15	-15,9895	-3443667
0,000381	-15	-15,9895	-1721825
0,000191	-15	-15,9895	-860889

Rysunek 3: Tabela z wynikami działania metody wewnętrznej funkcji kary dla c =20.

Porównując otrzymany w ostatniej iteracji wynik z analitycznie wyznaczonym minimum stwierdzono, że błąd jest niewielki, rzędu 2 miejsca po przecinku dla x_2, x_1 zostało natomiast obliczone poprawnie i wynosi -15.

Na rysunkach widać, że wewnętrzna funkcja kary osiąga minimum znajdując w kolejnych iteracjach rozwiązania z obszaru dopuszczalnego.

2.2. Zadanie B - Minimalizacja funkcjonału z ograniczeniami nieliniowymi

2.2.1. Rozwiązania analityczne

W zadaniu rozważany jest funkcjonał postaci $(n = 16)$:

$$f(x_1, x_2) = x_1 + 2x_2 + 3n = x_1 + 2x_2 + 48 \to min$$

przy danych ograniczeniach nierównościowych

$$\begin{cases} g_1(x) = -x_1^2 + x_2 + n(1 - 2x_1) - n^2 = -x_1^2 + x_2 - 32x_1 - 240 \geq 0 \\ g_1(x) = \frac{1}{4}x_1 + 1 + \frac{1}{4}n = \frac{1}{4}x_1 + 5 \geq 0 \end{cases}$$

Najpierw zadanie zostało rozwiązane analitycznie metodą Karusha-Kuhna-Tuckera.

$$f(x) = x_1 + 2x_2 + 48$$

$$\frac{\partial f}{\partial x_1} = 1 > 0$$
$$\frac{\partial f}{\partial x_2} = 2 > 0$$

Ograniczenia zamienione zostały na nierówności mniejszościowe słabe aby były zgodne z warunkami twierdzenia KKT.

$$F(x,\lambda) = x_1 + 2x_2 + 48 + \lambda_1(x_1^2 + 32x_1 + x_2 - 240) - \lambda_2(\frac{x_1}{4} + 5)$$
$$\frac{\partial F(x,\lambda)}{\partial x_1} = 1 - \lambda_1(-2x_1 - 32) - \frac{\lambda_2}{4}$$
$$\frac{\partial F(x,\lambda)}{\partial x_2} = -\lambda_1 + 2$$

Warunki konieczne twierdzenia KKT :

$$\begin{cases} 1 - \lambda_1(-2x_1 - 32) - \frac{\lambda_2}{4} = 0 \\ -\lambda_1 + 2 = 0 \\ \lambda_2(\frac{1}{4}x_1 + 5) = 0 \\ \lambda_1(-x_1^2 + x_2 - 32x_1 - 240) = 0 \end{cases} \implies \begin{cases} x_1 = \frac{\frac{\lambda_2}{4} - 1 - 32\lambda_1}{2\lambda_1} \\ \lambda_1 = 2 \\ x_1 = -20 \vee \lambda_2 = 0 \\ x_2 = x_1^2 + 32x_1 + 240 \end{cases}$$

Z równania na x_1 podstawiając $\lambda_1 = 2$ praz $\lambda_2 = 0$ otrzymujemy $x_1 = -16.25$. Następnie z równania ostatniego otrzymujemy $x_2 = -15,9375$. Dla $x_1 = -20$ otrzymujemy sprzeczność w równaniu pierwszym, ponieważ wówczas $\lambda_2 < 0$.

Ostatecznie pokazano, że minimum tego funkcjonału przy przyjętych ograniczeniach jest osiągane dla $x^* = \begin{bmatrix} -16,25 & -15,9375 \end{bmatrix}$.

2.2.2. Rozwiązania programowe bez metody wew. f. kary

W implementacji programowej zadanie najpierw rozwiązano przy użyciu funkcji matlaba *fmincon*. Funkcja ta znajduje rozwiązanie optymalne przy określonych ograniczeniach nierównościowych. W tym przypadku konieczne było napisanie dodatkowej funkcji, która określa ograniczenie nieliniowe. Konieczne było również przekształcenie ograniczeń do postaci odpowiadającej nierówności mniejszościowej słabej. Implementację i funkcję ograniczającą przedstawiają Listingi 4 i 5.

```
% znalezienie minimum z ograniczeniami
n = 16;

x0 = [ -17, 10];

fun = @(x) x(1) + 2 * x(2) + 3*n ;

A = [-1, 0 ];

b =  20 ;

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

Listing 4: Znalezienie minimum z ograniczeniami (fmincon)

```
function [c, ceq] = non_lin(x)
% Nonlinear inequality constraints
% Pamietac, ze tu musi byc mniejsze lub rowne
c = x(1)^2 - x(2) - 16*( 1-2*x(1) ) + 256;
% Nonlinear equality constraints
ceq = [];
```

Listing 5: Ograniczenie nieliniowe

Zgodnie z rozwiązaniem analitycznym znalezione przez program minimum to

$$x^* = [-16,25 \quad -15,9375] \tag{2}$$

2.2.3. Rozwiązania programowe z metodą wew. f. kary

Następnie zadanie rozwiązano przy użyciu kilku iteracji metodą wewnętrznej funkcji kary, która polega na generowaniu ciągu punktów będących minimum kolejnych wyznaczanych funkcji $P(x,k)$. W zadaniu rozważana jest funkcja

$$P(x,k) = f(x) + k \cdot \sum_{i=1}^{r} \frac{1}{g_i(x)} \rightarrow min$$

$$P(x,k) = x_1 + 2x_2 + 48 + k\frac{1}{-x_1^2 + x_2 - 32x_1 - 240} + k\frac{1}{\frac{1}{4}x_1 + 5} \rightarrow min$$

z funkcją kary:

$$\phi(x,k) = k \cdot \sum_{i=1}^{r} \frac{1}{g_i(x)}$$

$$\phi(x,k) = k\frac{1}{x_1 + 15 - x_1^2 + x_2 - 32x_1 - 240} + k\frac{1}{\frac{1}{4}x_1 + 5}$$

Definiując dwa zbiory:

$$X = \{x : g_i(x) \geq 0; \quad i = 1, 2, \ldots, r\}$$
$$X_0 = \{x : g_i(x) > 0; \quad i = 1, 2, \ldots, r\}$$

które określają fragment paraboli opisanej ograniczeniem $g_1(x)$ i 'uciętej' przez prostą $x_1 = -20$ można stwierdzić, że zachodzą założenia:

- X_0 jest niepusty,
- $f(x)$ oraz $-g_i(x)$ są wypukłe i dwukrotnie różniczkowalne
- zbiory poziomicowe $\{x : f(x) \leq \alpha\}$ są ograniczone dla każdego skończonego α.
- Dla każdego $k > 0$, $P(x,k)$ jest ściśle wypukła

Spełnienie powyższych założeń daje gwarancję, że każda funkcja $P(x,k)$ niezależnie od k osiąga minimum na X_0 i gradient $P(x,k)$ po zmiennej x jest w tym punkcie równy 0 oraz, że ciąg punktów znajdywanych w kolejnych iteracjach jest zbieżny do minimum.

Implementacje powyższych obliczeń przedstawia Listing 6.

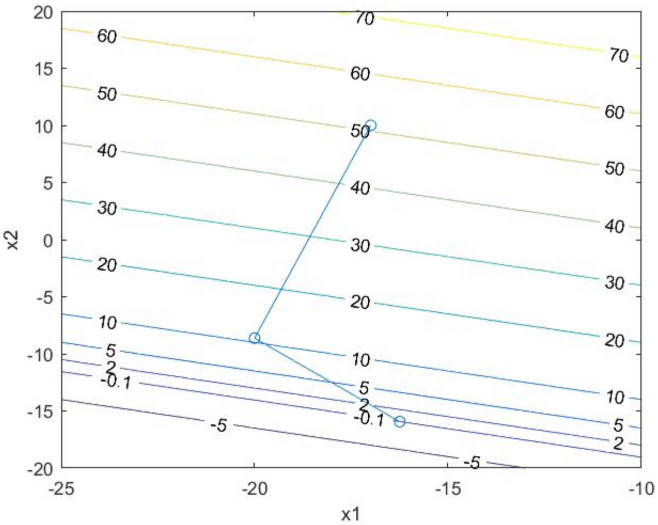
```
% znalezienie minimum przy uzyciu wewnetrznej f. kary

x0 = [ -17, 10 ];
max_iter = 20;
points = [ x0 ];
f_values = [ 0 ];
diff_table = [ 0 ];

c=20;
k=10;

k_table = [ k ];
fun = @(x) x(1) + 2 * x(2) + 3*n + k/(-x(1)^2+x(2)+n*(1-2*x(1))-n^2) + ...
    +k/( 1/4*x(1) +1 + 1/4*n );

for i = 1:max_iter
    [x_opt_k, fval] = fmincon(fun,x0,A,b,[],[],[],[],@non_lin);


    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/c;
    k_table = [ k_table ; k ];
    fun = @(x) x(1) + 2 * x(2) + 3*n + k*1/(-x(1)^2+x(2)+n*(1-2*x(1))-n^2) + ...
    +k*(1/( 1/4*x(1) +1 + 1/4*n ));

end
```

Listing 6: Znalezienie minimum przy użyciu wewnętrznej funkcji kary

Następnie znalezione przed metodę punkty zostały zaznaczone na wykresie przy pomocy kodu analogicznego do użytego w zadaniu A.

Wyniki wraz z poziomicami przedstawia Rysunek 4

2.2.3 Rozwiązania programowe z metodą wew. f. kary



Rysunek 4: c = 20, k0 = 10

Znalezione przez metodą rozwiązania wraz z numerem iteracji dla c = 20 przedstawiono w poniższej tabeli.

k	x1	x2	f(x)
200	-17	10	51
100	-20	-8,62352	-5,283E+10
50	-20	-8,62352	-2,642E+09
25	-20	-8,62352	-132077176
12,5	-20	-8,62352	-6603848,6
6,25	-20	-8,62352	-330182,22
3,125	-20	-8,62352	-16498,895
1,5625	-20	-8,62352	-814,72946
0,78125	-20	-8,62352	-30,521162
0,390625	-16,2503	-15,9373	-0,1249313
0,195313	-16,2515	-15,9368	-0,1253911
0,097656	-16,2515	-15,9368	-0,1290417
0,048828	-16,2515	-15,9368	-0,1251979
0,024414	-16,2515	-15,9368	-0,1250057
0,012207	-16,2515	-15,9368	-0,1249961
0,006104	-16,25	-15,9375	-0,1249999
0,003052	-16,25	-15,9375	-0,1249999
0,001526	-16,25	-15,9375	-0,1249999
0,000763	-16,25	-15,9375	-0,1249999
0,000381	-16,25	-15,9375	-0,1249999
0,000191	-16,25	-15,9375	-0,1249999

Rysunek 5: Tabela z wynikami działania metody wewnętrznej funkcji kary dla c =20.

Porównując otrzymany w ostatniej iteracji wynik z analitycznie wyznaczonym minimum stwierdzono, że program dał dokładnie ten sam wynik.

3 Wnioski końcowe

Idea działania metoda wewnętrznej funkcji kary polega na generowaniu ciągu rozwiązań dopuszczalnych. Metoda ta daje szybko dobre rezultaty, należy pamiętać, że ma też swoje ograniczenia, ponieważ we wnętrze obszaru dopuszczalnego musi być możliwe wpisanie kuli, co nie jest konieczne dla zewnętrznej funkcji kary, co oznacza, że metoda wewnętrznej funkcji kary nie może być stosowana dla ograniczeń równościowych.

Podobnie jak przy metodzie zewnętrznej funkcji kary na jakość działania algorytmu mają wpływ parametry algorytmu, najbardziej znaczące wydają się być parametry k oraz c, które decyduje o tym jak duża kara jest nałożona na daną funkcję za wyjście oraz jak szybko zmienia się wartość k. Prawdziwość tego stwierdzenia potwierdzają rysunki 1 i 2 .