Informatyka, studia dzienne, mgr II st. semestr I

Metody obliczeniowe optymalizacji **2011/2012**

Prowadzący: mgr inż. Łukasz Chomątek czwartek, 16:00

Data oddania: ______________ Ocena: ______________

Paweł Musiał 178726

Łukasz Michalski 178724

Zadanie 5 - Optymalizacja wielowymiarowa z ograniczeniami*

1. Cel

Napisać program rozwiązujący problem optymalizacji z ograniczeniami za pomocą jednej z poniższych metod:

- **zewnętrzna lub wewnętrzna funkcja kary,**
- SLP,
- **optymalizacja wielokryterialna - metoda sumy ważonej,**
- Box.

2. Rozwiązanie zadania

2.1. Optymalizacja wielokryterialna - metoda sumy ważonej

Problemy wielokryterialne, to problemy w których optymalizowanym wyrażeniem, jest parę funkcji celu. Jednym z proponowanych rozwiązań w literaturze [2] do tego typu problemów jest metoda sumy ważonej. Metoda ta polega na zastąpieniu wielu funkcji jedną funkcją będącą liniową kombinacją składowych.
Dla przykładu, mając problem postaci:

$$minF(f_1, f_2, f_3, \dots, f_n)) \tag{1}$$

* SVN: `https://serce.ics.p.lodz.pl/svn/labs/moo/lc_cz1600/lmpm`

przy ograniczeniach :

$$g(x) \leq 0$$
$$h(x) = 0$$

zastępujemy grupę funkcji celu za pomocą :

$$U = \sum_{i=1}^{n} w_i \cdot f_i \tag{2}$$

gdzie :

$$\sum_{i=1}^{n} w_i = 1$$

oraz $w > 0$.

Rozwiązanie takiego problemu będzie leżało na tak zwanym froncie pareto. Jest to powierzchnia na której występują rozwiązania niedominujące, czyli rozwiązań w sensie pareto, co oznacz, że każde rozwiązanie w zbiorze jest optymalne i nie lepsze ani gorsze od pozostałych.

Metoda ta jedynie opisuje sposób w jaki można rozwiązać problem z wieloma funkcjami celu, nie precyzuje jednak sposobu dalszych rozwiązań wyznaczonego problemu zastępczego U. Wartości wag w tej metodzie, można dobierać według potrzeb, oczywiście pamiętając o nakreślonych restrykcjach. Z drugiej strony zostały zaproponowanie rozwiązania adaptacyjnego doboru wag.

Jako, że metoda ta jedynie mówi o podejściu jakie można zastosować do problemów wielokryterialnych, polegającym na złożeniu wielu funkcji bazowych w jedną funkcje będącą superpozycją składowych przemnożonych przez jakieś wartości wag, tak na prawdę każdy bardziej złożony problem możemy traktować jako takie właśnie rozwiązanie. Dlatego też w naszym programie skupiliśmy się na jednej z metod rozwiązywania problemu „zastępczego” czyli problemu wielowymiarowej optymalizacji z ograniczeniami.

2.2. Zewnętrzna funkcja kary

Jest to metoda rozwiązywania problemów wielokryterialnych z ograniczeniami. Metoda ta jest prosta i polega na rozwiązywaniu problemu zastępczego, w którym rozwiązujemy problem podstawowy z dodaną funkcją kary będącą wartością - karą, za niespełnienie ograniczeń. Niestety nie przedstawimy tutaj dowodu poprawności tej metody. Jednak w wielu przypadkach metoda ta pozwala na dobre wyniki. Przedstawiony poniżej opis jest zaczerpnięty z [3].
Minimalizacja $f(x)$, przy ograniczeniach:

$$[g(x)]_m \leq 0$$
$$[h(x)]_l = 0$$

Wprowadzamy problem zastępczy, postaci:

$$F(x, r_h, r_g) = f(x) + P(x, r_h, r_g) \tag{3}$$

gdzie :

$$P(x,r_h,r_g) = r_h\left[\sum_{j=1}^{l}(h_j(x))^2\right] + r_g\left[\sum_{j=1}^{m}\left(max\left(0,g_j(x)\right)\right)^2\right] \tag{4}$$

Gdzie parametry r_h i r_g mówią, jak „mocno” karać za naruszenie ograniczeń. Podstawiając wszystko do jednego równania otrzymujemy :

$$F(x,r_h,r_g) = f(x) + r_h\left[\sum_{j=1}^{l}(h_j(x))^2\right] + r_g\left[\sum_{j=1}^{m}\left(max\left(0,g_j(x)\right)\right)^2\right] \tag{5}$$

Jak widać niejako pozbyliśmy się ograniczeń włączając je do minimalizowanego wyrażenia. W sposób ten otrzymaliśmy problem wielowymiarowej optymalizacji bez ograniczeń który możemy rozwiązywać znanymi nam metodami czyli na przykład algorytmem pełzającego sympleksu, bądź jedną z metod quasi-newtonowskich.

Algorytm w opisie krokowym:

1. Wybierz punkt początkowy $x = x_0$ znajdujący się w obszarze dopuszczalnym, współczynniki kary r_h, r_g i odpowiadające im współczynniki skalujące C_{r_h} i C_{r_g}, oraz kryterium stopu, tutaj ilość iteracji k_{max}, zainicjuj licznik iteracji $k = 1$.
2. Rozwiąż problem zastępczy $x_q = min\left(F(x,r_g,r_h)\right)$
3. Sprawdź czy ograniczenia zostały naruszone $g_i(x_q) \leq 0$, $h_j(x_q) = 0, i \in (1\ldots m), j \in (1\ldots l)$, jeśli ograniczenia nie zostały naruszone, sprawdź kryterium stopu.
4. Skaluj współczynniki kary : $r_h = r_h \cdot C_{r_h}$, $r_g = r_g \cdot C_{r_g}$
5. Zwiększ licznik iteracji $k = k + 1$, przypisz $x = x_q$ przejdź do 2.

3. Opis programu

```
function fstr=equalConSum(f)
fstr='( ';
for i=1:length(f)-1
    fstr=strcat(fstr,'',f(i),')^2 + ');
end
fstr=strcat(fstr,'(',f(length(f)),')^2 )');
```

Ograniczenia równościowe

```
function fstr=inEqualConSum(f)
fstr='( ';
for i=1:length(f)-1
    fstr=strcat(fstr,'(max(0,',f(i),'))^2 + ');
end
fstr=strcat(fstr,'(max(0,',f(length(f)),'))^2 )');
```

Ograniczenia nierównościowe

```
function fout = penaltyFun( f, h, g, rh, rg )
hfun=0;
gfun=0;
if ~isempty(h)
    hfun=@(x,y)subs(h,{'x','y'},{x,y});
end
if ~isempty(g)
    gfun=@(x,y)subs(g,{'x','y'},{x,y});
end

ffun=@(x,y)subs(f,{'x','y'},{x,y});

fout = @(x) ffun(x(1),x(2)) + rh*hfun(x(1),x(2)) +
 rg*gfun(x(1),x(2));

end
```

Funkcja problemu zastępczego

```
function [xi, fval] = epf(f, h, g, rh, rg, x0, Crh, Crg, kMax )
hpenalty='';
gpenalty='';
if ~isempty(h)
    hpenalty=equalConSum(h);
end
if ~isempty(g)
    gpenalty=inEqualConSum(g);
end
xi=x0
for i=1:kMax

    Fp=penaltyFun(f,hpenalty,gpenalty,rh,rg);

    [xi, fval]=fminunc(Fp,xi)
    unsatisfied=0;
    for j=1:length(h)
        hc=subs(h(j),{'x','y'},{xi(1),xi(2)});
        if (hc~=0)
            unsatisfied=unsatisfied+1;
        end
    end
    for j=1:length(g)
        gc=subs(char(g(j)),{'x','y'},{xi(1),xi(2)});
        if (gc>0)
            unsatisfied=unsatisfied+1;
        end
    end
    if(unsatisfied==0 || i>=kMax)
        break;
    end
    rh=rh*Crh;
    rg=rg*Crg;
end
end
```

Algorytm zewnętrznej funkcji kary

4. Wyniki

Rozwiązanie problemu testowano na funkcji przykładowej zawartej w książce [3].

$$f(x_1, x_2) = x_1^4 - 2x_1^2 x_2 + x_1^2 + x_1 x_2^2 - 2x_1 + 4 \tag{6}$$

przy ograniczeniach:

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

$$g(x_1, x_2 : 0.25x_1^2 + 0.75x_2^2 - 1 \leq 0$$

przy dobranych parametrach:

1. $x_0 = [1, 2]$
2. $k_{max} = 3$
3. $r_h = 5,\ C_{r_h} = 0.95$
4. $r_g = 5,\ C_{r_g} = 0.95$

iter	x	f(x)
1	0.9464, 1.0364	2.9740
2	0.9454, 1.0369	2.9736
3	0.9445, 1.0373	2.9731

Wynik w [2] dla porównania wynosił $x = [0.9775, 1.0165]$, $f(x) = 2.9810$. Jak widać uzyskany wynik jest bardzo bliski wartości otrzymanej przez obliczonej przez naszą implementacje, tejże metody już po 3 iteracjach.

5. Wnioski

Przedstawiona powyżej implementacja optymalizacji wielowymiarowej z ograniczeniami przy pomocy zewnętrznej funkcji kary jest jedną z metod optymalizacji znajdujących rozwiązanie wykorzystując przybliżenie problemu do oryginalnego. Podejście takie skutkuje dobrymi wynikami i jest łatwe w implementacji ponieważ opiera się na algorytmach rozwiązywania standardowych problemów optymalizacji, mamy tutaj na uwadze takie algorytmy jak SLP, SQP, metody oparte o funkcję kary. Jedynym minusem ich działania jest nie działanie na prawdziwej funkcji celu a jedynie na jej przybliżeniu, co w konsekwencji niejako implikuje słabą jakość przy zadaniach bardziej złożonych. Zaimplementowana przez nad metoda była testowana na prostym przykładzie dlatego też szybko uzyskała zbieżność.

Literatura

[1] „An Introduction to Optimization" , Edwin Kah Pin Chong and Stanislaw H. Zak, Hoboken, EUA : Wiley-Interscience 2008

[2] „Introduction to Optimum Design", Arora, Jasbir S., Elsevier Academic Press 2004

[3] „Applied Optimization with MATLAB Programming", Venkatamaran P., Wiley 2001