Dawid Brząkała, Marcel Garczyk, Mikołaj Wypych

ZADANIA MINIMAKSOWE I Z PARAMETRYZOWANYMI OGRANICZENIAMI

ZADANIA MINIMAKSOWE

Zadanie minimaksowe jest jednym z typów problemów optymalizacyjnych, które występują, gdy celem jest minimalizacja maksymalnej wartości funkcji celu w danym zbiorze możliwych rozwiązań.

Generalna postać zadania minimaksowego:

$$\min_{x} \max_{i} f_i(x) \qquad i \in \{1,2,\dots,m\}$$

Gdzie:

x – zmienne decyzyjne.

$f_i(x)$ – różne funkcje celu (można je traktować jako różne kryteria do spełnienia).

Celem jest znalezienie takiego x, które minimalizuje największą wartość funkcji $f_i(x)$.

ZASTOSOWANIA FUNKCJI MINIMAKSOWYCH

- **Teoria decyzji**: Optymalizacja strategii w grach, gdzie gracze chcą zminimalizować maksymalny zysk przeciwnika. Gracz stara się przewidzieć możliwe ruchy rywala i wybrać taki sposób, który ograniczy jego potencjalną przewagę.
- **Optymalizacja inżynierska**: Projektowanie systemów technicznych, do tworzenia rozwiązań odpornych na najgorsze możliwe scenariusze.
- **Uczenie maszynowe**: Minimalizacja błędów w obliczu maksymalnych zakłóceń w danych. Celem tego jest stworzenie modeli, które są odporne na celową manipulację danych, co zapewnia lepsze bezpieczeństwo i generalizację.
- **Optymalizacja portfeli inwestycyjnych**: Minimalizacja maksymalnego ryzyka strat.

FUNKCJA FMINIMAX

$$\min_{x} \max_{\{F_i\}} \{F_i(x)\}$$

Funkcja **fminimax** służy do rozwiązywania zadań minimaksowych w postaci:

$$przy\ obecności\ ograniczeń$$

$$c(x) \leq 0 \qquad c_{eq} = 0$$

$$A_x \leq b, \qquad A_{eq}x = b_{eq}$$

$$lb \leq x \leq ub$$

A: Macierz współczynników ograniczeń nierównościowych liniowych.

$\boldsymbol{A_{eq}}$: *Macierz współczynników równościowych liniowych.*

$\boldsymbol{c(x)}$: *Funckja opisująca nieliniowe ograniczenia nierównościowe.*

$\boldsymbol{c_{eq}}$: *Funckja opisująca nieliniowe ograniczenia równościowe.*

$\boldsymbol{x}$: *Wektor lub macierz zmiennych decyzyjnych, które optymalizujemy.*

lb *oraz* ***ub***: *wektorych dolnych i górnych ograniczeń dla zmiennych* $\boldsymbol{x}$.

PORÓWNANIE FUNKCJI

- **fminimax** minimalizuje maksymalną wartość spośród wielu funkcji celu. Jest przydatna, gdy mamy kilka funkcji, które chcemy optymalizować jednocześnie, minimalizując ich najgorszy wynik.
- **fseminf** minimalizuje jedną funkcję celu pod warunkiem spełnienia ograniczeń pół-nieskończonych, które muszą być spełnione dla nieskończonego zbioru punktów w przestrzeni ograniczeń.

$$\min_x \max_i F_i(x) \text{ such that } \begin{cases} c(x) \le 0 \\ ceq(x) = 0 \\ A \cdot x \le b \\ Aeq \cdot x = beq \\ lb \le x \le ub \end{cases}$$

where b and beq are vectors, A and Aeq are matrices, and $c(x)$, $ceq(x)$, and $F(x)$ are functions that return vectors.

$$\min_x f(x) \text{ such that } \begin{cases} A \cdot x \le b, \\ Aeq \cdot x = beq, \\ lb \le x \le ub, \\ c(x) \le 0, \\ ceq(x) = 0, \\ K_i(x, w_i) \le 0,\ 1 \le i \le n. \end{cases}$$

- b and beq are vectors.
- A and Aeq are matrices.
- $c(x)$, $ceq(x)$, and $K_i(x,w_i)$ are functions that return vectors.
- $f(x)$ is a function that returns a scalar.

WYTŁUMACZENIE FMINIMAX

Załóżmy, że mamy trzy funkcje celu f1(x), f2(x), f3(x), a celem jest minimalizacja największej z tych funkcji. Przykładowe zadanie może wyglądać tak:

$$\min_{x} \max(x^2, (x-2)^2, (x+1)^2)$$

To oznacza, że chcemy znaleźć wartość x, która minimalizuje maksymalną wartość z trzech funkcji kwadratowych.

PRZYKŁAD 1

Rozwiąż zadanie minimaksowe dla zestawu składającego się z 5 funkcji:

$$
\begin{aligned}
f_1(x) &= 2 \cdot x_1^2 + x_2^2 - 48 \cdot x_1 - 40 \cdot x_2 + 304, \\
f_2(x) &= -x_1^2 - 3 \cdot x_2^2, \\
f_3(x) &= x_1 + 3 \cdot x_2 - 18, \\
f_4(x) &= -x_1 - x_2, \\
f_5(x) &= x_1 + x_2 - 8
\end{aligned}
$$

cw9_1.m

/MATLAB Drive/cw9_1.m

```matlab
function f=cw9_1(x)
f(1)=2*x(1)^2+x(2)^2-48*x(1)-40*x(2)+304;
f(2)=-x(1)^2-3*x(2)^2;
f(3)=x(1)+3*x(2)-18;
f(4)=-x(1)-x(2);
f(5)=x(1)+x(2)-8;
```

```
>> x0=[0.1;0.1];
>> [x,fval]=fminimax('cw9_1',x0)

Local minimum possible. Constraints satisfied.

fminimax stopped because the size of the current search direction is less than
twice the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

    4.0000
    4.0000


fval =

    0.0000  -64.0000   -2.0000   -8.0000   -0.0000
```

W wyniku realizacji funkcji **fminimax** uzyskano rozwiązanie x = [4 4], czyli minimalizację maksymalnej wartości spośród zadanych 5 funkcji.

PRZYKŁAD 2

- Znajdź minimum funkcji max($-x_1$, x_1, $-x_2$, x_2)

cw9_2.m

/MATLAB Drive/cw9_2.m

```
function f=cw9_2(x)
f(1)=-x(1);
f(2)=x(1);
f(3)=x(2);
f(4)=-x(2);
```

```
>> x0=[1;1];
>> x=fminimax(@cw9_2,x0)

Local minimum possible. Constraints satisfied.

fminimax stopped because the size of the current search direction is less than
twice the value of the step size tolerance and constraints are
satisfied to within the value of the constraint tolerance.

<stopping criteria details>

x =

   1.0e-08 *

    0.3725
    0.3725
```

Znalezione minimum jest osiągnięte w punkcie bardzo bliskim początku współrzędnych.

ZADANIE 1

Rozwiąż zadanie minimaksowe dla zestawu składającego się z 6 funkcji:

$$f(1) = 3x_1^2 + 2x_2^2 - 30x_1 - 50x_2 + 200$$

$$f(2) = -2x_1^2 - 4x_2^2 + 10$$

$$f(3) = 2x_1 + 5x_2 - 50$$

$$f(4) = -5x_1 - 7x_2 + 20$$

$$f(5) = x_1 + 2x_2 - 10$$

$$f(6) = x_1 - 4$$

ZADANIE 1 - ROZWIĄZANIE

```
function f=zad1(x)
f(1) = 3*x(1)^2 + 2*x(2)^2 - 30*x(1) - 50*x(2) + 200;
f(2) = -2*x(1)^2 - 4*x(2)^2 + 10;
f(3) = 2*x(1) + 5*x(2) - 50;
f(4) = -5*x(1) - 7*x(2) + 20;
f(5) = x(1) + 2*x(2) - 10;
f(6) = x(1) - 4;
end

% Warunki początkowe
x0 = [1; 1]; % Początkowe wartości

% Rozwiązanie problemu optymalizacji
[x, fval] = fminimax(@zad1, x0);

disp('Optymalne rozwiązanie:');
disp(['x(1): ', num2str(x(1))]);
disp(['x(2): ', num2str(x(2))]);

disp(['fval: ', num2str(fval)]);
```

```
Optymalne rozwiązanie:
x(1): 2.0189
x(2): 3.5566
fval: -0.868028      -48.748      -28.1795      -14.9902      -0.868028      -1.98115
>>
```

Otrzymane wartości wskazują, że:

- Optymalna wartość dla x(1) wynosi **2.0189**.
- Optymalna wartość dla x(2) wynosi **3.5566**.

Te wartości to punkty, w których maksymalna wartość funkcji celu (spośród funkcji f(n)) jest najmniejsza.

ZADANIE 2

- Rozwiąż zadanie minimaksowe dla zestawu składającego się z 3 funkcji:
- $F_1(x) = 2\mathrm{sinc}(x_1)$
- $F_2(x) = 3\sin(7x_1) + 6\cos(8x_2)$
- $F_2(x) = 8\tan(3x_1) + \tan(2x_2)$

ZADANIE 2 - ROZWIĄZANIE

```
function f=zadanie2(x)
    f(1)= 2*sinc(x(1))
    f(2)= 3*sin(7*x(1)) + 6* cos(8*x(2))
    f(3)= 8*tan(3*x(1))+ tan(2*x(2))
end

% Warunki początkowe
x0 = [10; 1]; % Początkowe wartości

% Rozwiązanie problemu optymalizacji
[x, fval] = fminimax(@zadanie3, x0);

disp('Optymalne rozwiązanie:');
disp(['x(1): ', num2str(x(1))]);
disp(['x(2): ', num2str(x(2))]);

disp(['fval: ', num2str(fval)]);
```

```
Optymalne rozwiązanie:
x(1): 1.1092
x(2): 1.0516
fval: -0.19299    -0.19299    -0.19299
>>
```

ZADANIE 3

- Znajdź minimum funkcji max($-2x_1$, x_1+x_2, $-x_2$, x_2)

ZADANIE 3 - ROZWIĄZANIE

```
function f=zadanie3(x)
    f(1)= -2*x(1)
    f(2)= x(1)+x(2)
    f(3)= -x(2)
    f(4)= x(2)
end

% Warunki początkowe
x0 = [1; 1]; % Początkowe wartości

% Rozwiązanie problemu optymalizacji
[x, fval] = fminimax(@zadanie3, x0);

disp('Optymalne rozwiązanie:');
disp(['x(1): ', num2str(x(1))]);
disp(['x(2): ', num2str(x(2))]);

disp(['fval: ', num2str(fval)]);
```

```
Optymalne rozwiązanie:
x(1): 4.4409e-16
x(2): 2.2204e-16
fval: -8.8818e-16  6.6613e-16 -2.2204e-16  2.2204e-16
>>
```

ZADANIE 4

- Zaprojektuj zbiornik o minimalnej maksymalnej różnicy ciśnień wewnętrznych, uwzględniając ograniczenia na objętość i koszty materiałowe.
- Objętość zbiornika powinna być 500 m^3
- Powierzchnia materiału poniżej 4000 m^2
- Promień dodatni i powierzchnia dodatnia

ZADANIE 4 - ROZWIĄZANIE

```
function f = tank_design(x)
% x(1): promień zbiornika (r)
% x(2): wysokość zbiornika (h)

% Funkcje celu do minimalizacji maksymalnej różnicy ciśnień i kosztów
f(1) = pi * x(1)^2 * x(2) - 500; % Objętość zbiornika powinna być 500 m^3
f(2) = 2*pi*x(1)*x(2) + 2*pi*x(1)^2 - 4000; % Powierzchnia materiału poniżej 4000 m^2
f(3) = -x(1); % Promień musi być dodatni
f(4) = -x(2); % Wysokość musi być dodatnia
end

% Warunki początkowe
x0 = [1; 1]; % Początkowe wartości promienia i wysokości
lb = [0.1; 0.1]; % Dolne ograniczenia (promień i wysokość > 0)
ub = []; % Brak górnych ograniczeń


[x, fval] = fminimax(@tank_design, x0, [], [], [], [], lb, ub, [], options);
% Rozwiązanie problemu optymalizacji


disp('Optymalne wymiary zbiornika:');
disp(['Promień: ', num2str(x(1))]);
disp(['Wysokość: ', num2str(x(2))]);
```

```
Optymalne wymiary zbiornika:
Promień: 5.3997
Wysokość: 5.3997
>>
```

ZADANIE 5

- Optymalizacja diety
- Cel – minimalizacja maksymalnego kosztu diety, aby spełnić minimalne wymagania kaloryczne i białkowe.
- Owsianka koszt - **2zł** za porcję. Dostarcza **100kcal** i **5g** białka.
- Mleko koszt – **3zł** za porcję. Dostarcza **50kcal** i **3g** białka.
- Minimalne wymagania to **500kcal** i **30g** białka.

```
calories = [];  % Kalorie w porcji owsianki i mleka
protein = [];      % Białko w porcji owsianki i mleka
cost = [];         % Koszt owsianki i mleka

% Minimalne wymagania
min_calories = ;    % Minimalna liczba kalorii
min_protein = ;      % Minimalna liczba gramów białka

% Funkcja celu: minimalizacja maksymalnego kosztu diety
objective = @(x) ;  % Całkowity koszt owsianki i mleka

% Ograniczenia nierówności: zaspokojenie wymagań kalorycznych i białkowych
A = [];
b = [];


lb = [];  % Dolne ograniczenie

% Rozwiązanie za pomocą fminimax
x0 = [1, 1];  % Punkt początkowy
[x, fval] = fminimax(@(x) [cost(1)*x(1), cost(2)*x(2)], x0, A, b, [], [], lb, []);

% Wyświetlenie wyników
fprintf('Optymalna liczba porcji:\n');
fprintf('Owsianka: %.2f porcji\n', x(1));
fprintf('Mleko: %.2f porcji\n', x(2));
fprintf('Minimalny maksymalny koszt: %.2f zł\n', max(fval));
```

ZADANIE 5 - ROZWIĄZANIE

```
% Parametry problemu
calories = [100, 50];  % Kalorie w porcji owsianki i mleka
protein = [5, 3];      % Białko w porcji owsianki i mleka
cost = [2, 3];         % Koszt owsianki i mleka

% Minimalne wymagania
min_calories = 500;    % Minimalna liczba kalorii
min_protein = 30;      % Minimalna liczba gramów białka

% Funkcja celu: minimalizacja maksymalnego kosztu diety
objective = @(x) cost * x;  % Całkowity koszt owsianki i mleka

% Ograniczenia nierówności: zaspokojenie wymagań kalorycznych i białkowych
A = [-calories; -protein];  % Ujemne, ponieważ ograniczenia są w formie <=
b = [-min_calories; -min_protein];

% Dodatkowe ograniczenie: proporcje muszą być nieujemne
lb = [0, 0];  % Dolne ograniczenie (brak negatywnych porcji)

% Rozwiązanie za pomocą fminimax
x0 = [1, 1];  % Punkt początkowy
[x, fval] = fminimax(@(x) [cost(1)*x(1), cost(2)*x(2)], x0, A, b, [], [], lb, []);

% Wyświetlenie wyników
fprintf('Optymalna liczba porcji:\n');
fprintf('Owsianka: %.2f porcji\n', x(1));
fprintf('Mleko: %.2f porcji\n', x(2));
fprintf('Minimalny maksymalny koszt: %.2f zł\n', max(fval));
```

Optymalna liczba porcji:
Owsianka: 4.29 porcji
Mleko: 2.86 porcji
Minimalny maksymalny koszt: 8.57 zł

DZIĘKUJEMY ZA UWAGĘ