Programowanie matematyczne
Laboratorium 3

Piotr Widomski, grupa D

09.11.2022

1 Postać ZPL

Rozwiązywany problem jest następujący: mając dwa czworościany P, Q w przestrzeni $\mathbb{R}^3$ zdefiniowane przez wierzchołki $\mathbf{p}_i, \mathbf{q}_i \in \mathbb{R}^3, i = 1 \ldots 4$, należy sprawdzić, czy czworościany posiadają część wspólną.
Czworościany P, Q (z wnętrzem) definiujemy jako:

$$P = \{\mathbf{x} \in \mathbb{R}^3 : \ \mathbf{x} = \sum_{i=1}^{4} \mathbf{p}_i \lambda_i, \ \sum_{i=1}^{4} \lambda_i = 1, \ \forall i \ \lambda_i \geqslant 0\}$$
$$Q = \{\mathbf{x} \in \mathbb{R}^3 : \ \mathbf{x} = \sum_{i=1}^{4} \mathbf{q}_i \lambda_i, \ \sum_{i=1}^{4} \lambda_i = 1, \ \forall i \ \lambda_i \geqslant 0\}$$

Jeżeli problem ma rozwiązanie, to będzie nim dowolny punkt x należący jednocześnie do P i Q. Z definicji P i Q otrzymujemy następujące ograniczenia:

$$\Omega : \begin{cases} \mathbf{p}_1\lambda_1 + \mathbf{p}_2\lambda_2 + \mathbf{p}_3\lambda_3 + \mathbf{p}_4\lambda_4 = \mathbf{q}_1\lambda_5 + \mathbf{q}_2\lambda_6 + \mathbf{q}_3\lambda_7 + \mathbf{q}_4\lambda_8 \\ \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1 \\ \lambda_5 + \lambda_6 + \lambda_7 + \lambda_8 = 1 \\ \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7, \lambda_8 \geqslant 0 \end{cases}$$

Zadanie programowania liniowego w postaci standardowej rozwiązujące przedstawiony problem wygląda następująco:

$$\max_{x \in \Omega} 0$$
$$\Omega : \begin{cases} p_{1,1}x_1 + p_{2,1}x_2 + p_{3,1}x_3 + p_{4,1}x_4 - q_{1,1}x_5 - q_{2,1}x_6 - q_{3,1}x_7 - q_{4,1}x_8 = 0 \\ p_{1,2}x_1 + p_{2,2}x_2 + p_{3,2}x_3 + p_{4,2}x_4 - q_{1,2}x_5 - q_{2,2}x_6 - q_{3,2}x_7 - q_{4,2}x_8 = 0 \\ p_{1,3}x_1 + p_{2,3}x_2 + p_{3,3}x_3 + p_{4,3}x_4 - q_{1,3}x_5 - q_{2,3}x_6 - q_{3,3}x_7 - q_{4,3}x_8 = 0 \\ x_1 + x_2 + x_3 + x_4 = 1 \\ x_5 + x_6 + x_7 + x_8 = 1 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 \geqslant 0 \end{cases}$$

Ze względu na to, że interesuje nas jedynie istnienie dopuszczalnego rozwiązania bazowego, funkcja celu ma wszystkie współczynniki zerowe. Nie występują żadne ograniczenia nierównościowe, dlatego nie dołożone zostały żadne zmienne dopełniające. Taka postać ZPL nie jest kanoniczna, gdyż nie jesteśmy w stanie dobrać zmiennych bazowych w taki sposób, aby A_B była macierzą jednostkową (niezależnie od wartości **p** i **q**, gdyż ograniczenie 4 i 5 sprawia, że nie otrzymamy wiersza z tylko jedną jedynką).

2 Zastosowany algorytm

Z uwagi na niekanoniczną postać ZPL, która utrudnia wybór początkowego BRD, do rozwiązania problemu użyty został dwufazowy wariant algorytmu sympleks. Wariant ten wymaga dołożenia zmiennych technicznych. Ograniczenia problemu bazowego reprezentujemy następująco:

$$\Omega : \begin{cases} Ax = b, b \geqslant 0 \\ x \geqslant 0 \end{cases}$$

W pierwszej fazie do zadania wprowadzone zostają zmienne sztuczne $S_i, i = 1, \ldots, 5$ oraz szukamy rozwiązania optymalnego następującego pomocniczego ZPL:

$$\min_{(x,s)} \sum_i s_i$$

$$\Omega : \begin{cases} Ax + s = b \\ x \geqslant 0, s \geqslant 0 \end{cases}$$

Rozwiązanie obliczane jest algorytmem sympleks, którego kroki opisane zostały w dalszej części. Jako zmienne bazowe użyte zostały zmienne sztuczne. Jest to umotywowane tym, że pomocnicze ZPL ma postać kanoniczną, ponieważ macierz A' zadania pomocniczego jest konkatenacją macierzy A zadania bazowego oraz macierzy jednostkowej. Macierz ta odpowiada dodanym zmiennym sztucznym. Z tego powodu dla $x_B = s$ $A_B = I$. Czyli $s = b$ jest BRD pomocniczego ZPL.

W fazie drugiej na podstawie RO problemu pomocniczego jesteśmy w stanie określić początkowe BRD bazowego problemu. Jeżeli RO problemu pomocniczego zawiera zmienne sztuczne z wartościami niezerowymi, to bazowe ZPL jest sprzeczne i kolejny krok nie jest wykonywany. Jeżeli rozwiązanie nie zawiera zmiennych sztucznych, lub zawiera je z wartością zerową, to usunięcie tych zmiennych z bazy daje nam BRD bazowego problemu. Wykorzystując otrzymane BRD obliczane jest RO bazowego problemu.

Przy użyciu algorytmu sympleks otrzymywany jest w następujących krokach:

1. Obliczenie wskaźników optymalności $z_j - c_j$. Jeżeli wszystkie są nieujemne to algorytm znalazł rozwiązanie optymalne.

2. Znalezienie indeksu k zmiennej dodawanej do bazy odpowiadającej najmniejszemu wskaźnikowi optymalności.

3. Znalezienie indeksu r zmiennej usuwanej z bazy odpowiadającej najmniejszej wartości $\frac{\bar{b}_i}{\bar{a}_{ik}}$ dla nieujemnej wartości $\bar{a}_{ik}$.

4. Obliczenie nowych wartości $\overline{A}$ i $\bar{b}$.

Dla przykładowej instancji problemu z punktami $\mathbf{p}_i$ i $\mathbf{q}_i$ przedstawionymi w formie macierzy

$$P = \begin{bmatrix} 9 & 3 & 9 & 2 \\ 10 & 6 & 9 & 6 \\ 8 & 1 & 8 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 4 & 1 & 6 & 7 \\ 3 & 4 & 5 & 7 \\ 1 & 1 & 7 & 0 \end{bmatrix}$$

startowa tabelka sympleksowa wygląda następująco:

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	b
c	0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	NaN
x9	9	3	9	2	-4	-1	-6	-7	1	0	0	0	0	0
x10	10	6	9	6	-3	-4	-5	-7	0	1	0	0	0	0
x11	8	1	8	0	-1	-1	-7	0	0	0	1	0	0	0
x12	1	1	1	1	0	0	0	0	0	0	0	1	0	1
x13	0	0	0	0	1	1	1	1	0	0	0	0	1	1
z	-28	-11	-27	-9	7	5	17	13	-1	-1	-1	-1	-1	NaN
z - c	-28	-11	-27	-9	7	5	17	13	0	0	0	0	0	NaN

3 Przykłady obliczeniowe

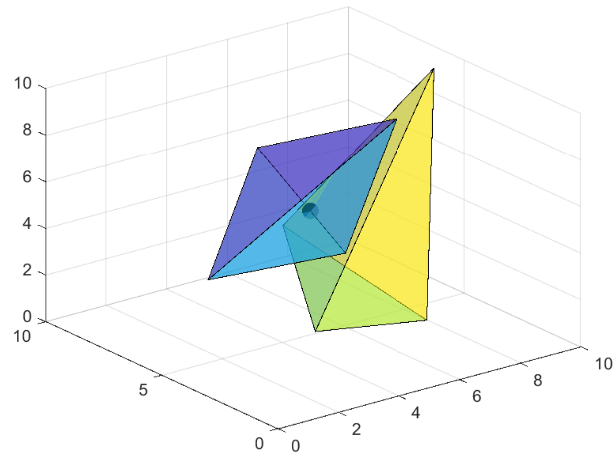
3.1 Kolizja

Dla następujących wierzchołków czworościanów:

$$P = \begin{bmatrix} 0 & 7 & 7 & 3 \\ 3 & 4 & 10 & 1 \\ 5 & 9 & 5 & 6 \end{bmatrix}$$

$$Q = \begin{bmatrix} 8 & 2 & 4 & 9 \\ 4 & 1 & 5 & 5 \\ 0 & 3 & 5 & 10 \end{bmatrix}$$

Punkt wspólny znaleziony przez funkcję `linprog` ma współrzędne $x = [4{,}6;\ 4{,}6;\ 5{,}6]$.



Rysunek 1: Punkt wspólny czworościanów wyznaczony funkcją `linprog`.

Wynik dla zaimplementowanego algorytmu sympleks otrzymywany jest przez zastosowanie wyżej opisanych kroków. Ostatnia iteracja przeprowadzona została następująco:

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	b
0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	NaN
0	1.5	0	1	-0.35714	-0.14286	0	-1.4286	0.17857	-0.17857	0.10714	0	0.35714	0.35714
1	-0.35714	0	0	1.8469	0.22449	0	0.81633	-0.28061	0.13776	0.11735	0	0.15306	0.15306
0	-2.498e-16	0	0	1	1	1	1	-8.5249e-17	-6.0468e-17	3.7668e-17	0	1	1
0	-0.5	0	0	-1.0714	-0.42857	0	0.71429	0.035714	-0.035714	-0.17857	1	-0.92857	0.071429
0	0.35714	1	0	-0.41837	0.34694	0	-0.10204	0.066327	0.076531	-0.045918	0	0.41837	0.41837
0	0.5	0	0	1.0714	0.42857	0	-0.71429	-0.035714	0.035714	0.17857	-1	0.92857	NaN
0	0.5	0	0	1.0714	0.42857	0	-0.71429	0.96429	1.0357	1.1786	0	1.9286	NaN

Rysunek 2: Tabelka sympleksowa przed ostatnią iteracją algorytmu.

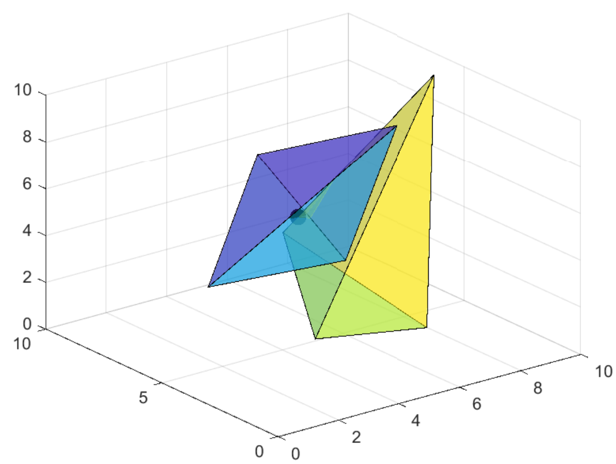
Najmniejsza wartość wskaźnika optymalności $z_j - c_j$ wynosi -0.7143 dla $k = j = 8$. Z tego powodu zmienną wchodzącą do bazy jest x_8. Szukając zmiennej wychodzącej z bazy rozważona została ósma kolumna macierzy $\overline{A}$. Niezerowe wartości znajdują się w wierszach $2-4$. Najmniejsza wartość $\frac{\bar{b}_i}{\bar{a}_{ik}}$ osiągnięta została dla $r = i = 4$. Zatem zmienna x_8 zastąpiła zmienną x_{12} jako zmienna bazowa. Po wyliczeniu tabelki sympleksowej dla nowej bazy otrzymujemy:

9:

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	b
0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	NaN
0	0.5	0	1	-2.5	-1	0	0	0.25	-0.25	-0.25	2	-1.5	0.5
1	0.21429	0	0	3.0714	0.71429	0	0	-0.32143	0.17857	0.32143	-1.1429	1.2143	0.071429
0	0.7	0	0	2.5	1.6	1	0	-0.05	0.05	0.25	-1.4	2.3	0.9
0	-0.7	0	0	-1.5	-0.6	0	1	0.05	-0.05	-0.25	1.4	-1.3	0.1
0	0.28571	1	0	-0.57143	0.28571	0	0	0.071429	0.071429	-0.071429	0.14286	0.28571	0.42857
0	0	0	0	0	0	0	0	0	0	0	0	0	NaN
0	0	0	0	0	0	0	0	1	1	1	1	1	NaN

Rysunek 3: Tabelka sympleksowa po ostatniej iteracją algorytmu.

Wskaźnik optymalności jest nieujemny dla wszystkich zmiennych. Rozwiązanie optymalne pomocniczego ZPL otrzymujemy poprzez przypisanie zmiennym bazowym odpowiadających wartości $\bar{b}$. Żadna zmienna sztuczna nie znajduje się w bazie, więc jest to BRD problemu bazowego. Jak ustalono wcześniej, problem przecinania się czworościanów rozwiązuje optymalnie dowolne BRD stworzonego ZPL. Zatem otrzymany wynik jest rozwiązaniem optymalnym i pewnym punktem należącym do obu czworościanów, co oznacza, że czworościany kolidują. Współrzędne tego punktu wynoszą $x = [4{,}5;\ 5;\ 5{,}5]$. Na podstawie zaimplementowanego algorytmu sympleks w wariancie dwuetapowym otrzymano taki sami wniosek jak przy użyciu funkcji `linprog`.



Rysunek 4: Punkt wspólny czworościanów wyznaczony własną implementacją algorytmu sympleks.

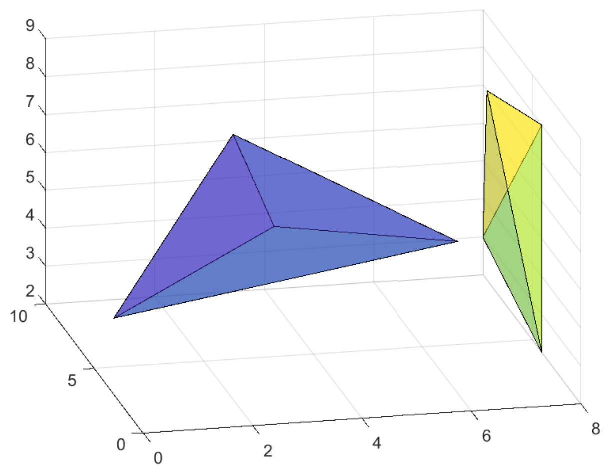
3.2 Brak kolizji

Dla następujących wierzchołków czworościanów:

$$P = \begin{bmatrix} 7 & 0 & 2 & 4 \\ 7 & 3 & 2 & 9 \\ 4 & 4 & 9 & 4 \end{bmatrix}$$

$$Q = \begin{bmatrix} 8 & 8 & 8 & 7 \\ 4 & 4 & 10 & 4 \\ 8 & 2 & 3 & 9 \end{bmatrix}$$

Wynikiem funkcji `linprog` jest brak kolizji między czworościanami.



Rysunek 5: Czworościany P i Q.

Wynik dla zaimplementowanego algorytmu sympleks otrzymywany jest przez zastosowanie wyżej opisanych kroków. Tabelka sympleksowa po ostatniej iteracji wygląda następująco:

:

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	b
0	0	0	0	0	0	0	0	-1	-1	-1	-1	-1	NaN
1	1	1	1	4.4409e-16	0	0	0	0	0	0	1	0	1
0	0.99419	0.94477	0.97674	0.33721	1	0	0	-0.22674	0.14826	0.11047	0.10756	0	0.10756
0	0.10465	0.49419	-0.5814	-0.069767	0	1	0	0.081395	-0.1686	0.011628	0.56395	0	0.56395
0	-0.25581	-0.93023	-0.023256	0.83721	0	0	1	0.023256	0.023256	-0.13953	0.23256	0	0.23256
0	-0.84302	-0.50872	-0.37209	-0.10465	0	0	0	0.12209	-0.002907	0.017442	-0.90407	1	0.09593
0	0.84302	0.50872	0.37209	0.10465	0	0	0	-0.12209	0.002907	-0.017442	0.90407	-1	NaN
0	0.84302	0.50872	0.37209	0.10465	0	0	0	0.87791	1.0029	0.98256	1.9041	0	NaN

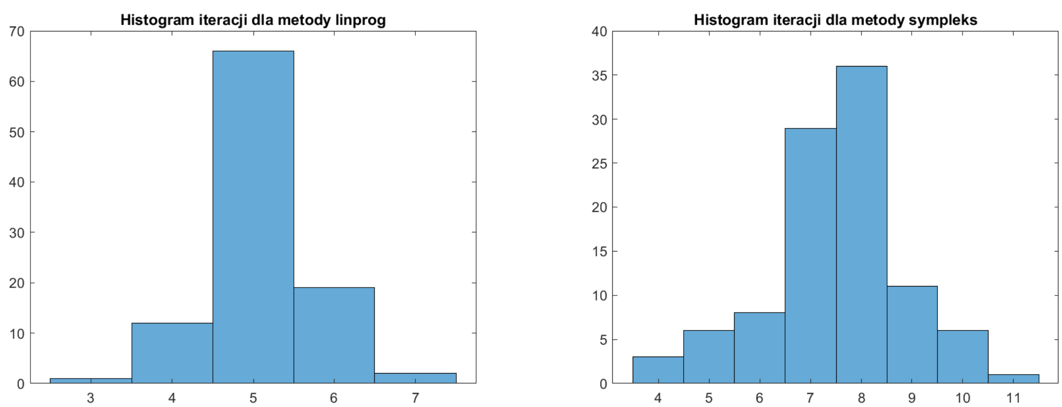
Rysunek 6: Tabelka sympleksowa po ostatniej iteracji algorytmu.

Wskaźnik optymalności jest nieujemny dla wszystkich zmiennych. Rozwiązanie optymalne pomocniczego ZPL otrzymujemy poprzez przypisanie zmiennym bazowym odpowiadających wartości $\bar{b}$. Zmienna sztuczna $x_1 3$ znajduje się w bazie rozwiązania z odpowiadającą wartością w $\bar{b}$ niezerową. Oznacza to, że bazowe ZPL nie posiada rozwiązania. Wynik otrzymany przez implementację algorytmu sympleks w wariancie dwuetapowym zgadza się z wynikiem otrzymanym przez funkcję `linprog`.

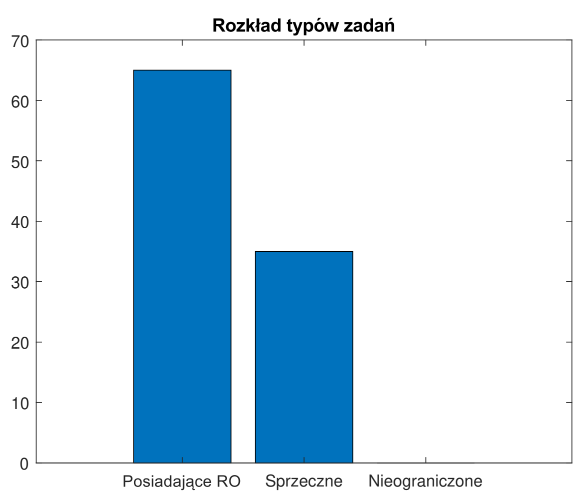
4 Analiza wyników

W celu sprawdzenia poprawności wyników zaimplementowanego algorytmu przeprowadzony został test dla $N = 100$ instancji problemu. Dla każdej instancji wynik, czyli informacja o występowaniu lub braku kolizji i powód jej braku, został porównany z wynikami funkcji `linprog`.

Dla wszystkich przypadków testowych otrzymany wynik zgadzał się z wynikiem funcki `linprog`. Zaimplementowany algorytm wykonywał większą ilość iteracji, ze średnią wynoszącą 7,51 iteracji. Funkcja `linprog` posiadała średnią ilość iteracji równą 5,09.



Rysunek 7: Histogram ilości iteracji dla algorytmu sympleks oraz funkcji `linprog`.



Rysunek 8: Rozkład typów wygenerowanych problemów.

5 Oświadczenie

Oświadczam, że niniejsza praca stanowiąca podstawę do uzyskania osiągnięcia efektów uczenia się z przedmiotu "Programowanie Matematyczne" została wykonana przeze mnie samodzielnie.

Piotr Widomski
298919