

Wyznaczenie optymalnego rozwiązania zadania ZP na podstawie zadania ZD

Lab4c

Wojciech Klusek 305943 grupa C

30 listopada 2023

Spis treści

1	Zadanie	2
2	Opis algorytmu	2
3	Rozwiązanie	3
4	Przykłady	4
4.1	Rozwiązanie optymalne	4
4.2	Brak rozwiązania	7
5	Testy	8
6	Wnioski	8
7	Oświadczenie o samodzielności	8

1 Zadanie

Zadaniem była modyfikacja algorytmu simplex do rozwiązania problemu ZP na podstawie zadania ZD. Dodatkowo, celem było porównanie efektywności tej implementacji z działaniem wbudowanej funkcji `linprog` z pakietu Optimization Toolbox.

$$\max_{x \in \Omega} c^t x$$

$$Ax \leq b, \quad b > 0$$

$$x \leq g, \quad g > 0$$

$$[x_1, \dots, x_n] - \text{nieograniczone}$$

$$c, x, g \in R^n, b \in R^m, A \in R^{m \times n}, n = 5, m = 10$$

Do testów wygenerowane zostały losowe wektory o wartościach całkowitoliczbowych, dla c oraz A z zakresu $[-2, 2]$, natomiast dla b oraz g z $[1, 5]$.

2 Opis algorytmu

1. **Inicjalizacja:** Wprowadź problem optymalizacyjny do postaci dualnej.

Dla powyższego zadania aby sprowadzić problem do postaci dualnej należało wykonać transpozycję macierzy współczynników. Jako, że x w zadaniu ZP były nieograniczone nierówności przechodzą na równości, a jako współczynniki b przyjmujemy współczynniki wartości funkcji celu c , a jako funkcję celu wektor $[b; g]$. Ostatecznie otrzymujemy poniższe zadanie dualne:

$$\min_{y \in \Omega} [b; g]^t y$$

$$[A' I] y = c, c \in [-2, 2]$$

$$[y_1, \dots, y_m] \geq 0$$

$$c, g \in R^n, y \in R^{m+n}, b \in R^m, A' \in R^{n \times m}, I \in R^{n \times n}, n = 5, m = 10$$

2. **Określenie punktu startowego:** Przekształć do postaci kanonicznej, wybierz punkt początkowy i oblicz wartość funkcji celu.

Aby przekształcić powyższe zadanie do postaci kanonicznej należy zmienić minimalizację funkcji celu na maksymalizację funkcji o przeciwnym znaku czyli zamienić $\min_{y \in \Omega} [b; g]^t y$ na $\max_{y \in \Omega} -[b; g]^t y$.

Jeśli wszystkie prawe strony równości są nieujemne jako bazę wybieramy zmienne odpowiadające macierzy jednostkowej I , liczymy wartość funkcji celu i możemy przejść do 3 kroku.

W przeciwnym przypadku problemem są ujemne zmienne w wektorze c , gdyż w postaci kanonicznej prawe strony równości powinny być nieujemne. Aby pozbyć się ujemnych prawych stron, mnożymy wiersze, które mają ujemną wartość po prawej stronie $\cdot -1$. Przemnożenie wybranych wierszy przez -1 powoduje kolejny problem, a mianowicie, tracimy zmienne tworzące początkową bazę gdyż teraz w części macierzy zawierającej macierz jednostkową mogą się pojawić wartości -1 . Aby utworzyć nową bazę dodajemy tyle sztucznych zmiennych ile jest -1 w macierzy jednostkowej dzięki czemu możemy stworzyć bazę ze zmiennych sztucznych oraz zmiennych, z poprzedniej macierzy jednostkowej, które mają dodatnie wartości. Jako nową funkcję celu przyjmujemy maksymalizację wektora zerowego, który ma -1 w miejscach, gdzie znajdują się sztuczne zmienne (intuicyjnie: chcemy usunąć sztuczne zmienne z bazy). Liczymy wartość funkcji celu i możemy przejść do 3 kroku.

3. **Test optymalności:** Sprawdź warunki zakończenia algorytmu (wszystkie współczynniki funkcji celu są ujemne lub brak dopuszczalnych kierunków poprawy).

Jeśli warunki zakończenia algorytmu są spełnione i jesteśmy w pierwszej fazie (usuwanie zmiennych sztucznych z bazy) usuwamy columny, w których znajdują się zmienne sztuczne, przyjmujemy jako nową funkcję celu $\max_{y \in \Omega} -[bg]^t y$, liczymy wartość funkcji celu i ponownie przechodzimy do kroku 3.

Jeśli warunki zakończenia algorytmu są spełnione i jesteśmy w drugiej fazie odczytujemy z tabeli optymalną wartość wektora y . Aby otrzymać optymalną wartość wektora x wybieramy z końcowej tabeli ostatnią macierz kwadratową pomnożoną przez -1 (z części tabeli zawierającej współczynniki przy zmiennych) o rozmiarze bazy i mnożymy współczynniki przy zmiennych bazowych w końcowej tabeli przez powyższą macierz otrzymując optymalną wartość x .

4. **Wybór kierunku:** Wybierz indeks zmiennej, która wejdzie do bazy (w przypadku wielu kandydatów zastosuj regułę wyboru, np. największy dodatni współczynnik w funkcji celu).
5. **Określenie kroku:** Wybierz zmienną, która opuści bazę (minimalny dodatni iloraz wyrazów wolnych do współczynników w kolumnie wybranej zmiennej).
6. **Aktualizacja bazy:** Dokonaj operacji elementarnej, aby zaktualizować bazę.
7. **Powrót do kroku 3.**

3 Rozwiązanie

Rozwiązanie składa się z 4 plików:

1. `simplex.m` - Implementacja modyfikacji algorytmu `simplex` dla powyższego zadania. Przyjmuje jako wejście zmienne:

- c - wektor współczynników funkcji celu,
- A - macierz współczynników ograniczeń,
- b - wektor prawych stron ograniczeń,
- g - wektor prawych stron ograniczeń,
- `debug` - opcjonalna flaga określająca, czy wyświetlać kolejne tabele w czasie iteracji.

Funkcja zwraca wektor ROx będący rozwiązaniem zadania, wektor ROy będący rozwiązaniem zadania dualnego oraz *exitflag* określającą czy rozwiązanie zostało znalezione (1 - tak, 0 - nie).

2. `simplex_test.m` - Porównanie algorytmu wraz z algorytmem `linprog`.
3. `generator.m` - Generuje macierze oraz wektory: A , b , c zgodnie z zadaniem.
4. `get_value_with_precision.m` - Wyznacza wartość funkcji i zaokrągla ją do określonej liczby miejsc po przecinku. Przyjmuje x - rozwiązanie, c - zadanie, oraz `precision` - liczbę miejsc po przecinku.

4 Przykłady

4.1 Rozwiązanie optymalne

$$A = \begin{bmatrix} 0 & 0 & 2 & -2 & 2 \\ 1 & 1 & 2 & 0 & 1 \\ -2 & -1 & -1 & 2 & -1 \\ -1 & 2 & 1 & 0 & 1 \\ -2 & -2 & 2 & 1 & -2 \\ -2 & 1 & 2 & -1 & 0 \\ -2 & 0 & -2 & 1 & 2 \\ -1 & 0 & -2 & 2 & -1 \\ -1 & -2 & -2 & -2 & -1 \\ 0 & -2 & 2 & 1 & -2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 4 \\ 2 \\ 2 \\ 3 \\ 1 \\ 3 \\ 1 \\ 3 \\ 4 \\ 3 \\ 4 \\ 3 \\ 5 \\ 3 \end{bmatrix}$$

$$c = [-2 \quad 0 \quad 1 \quad 0 \quad -2]$$

Pierwszym krokiem w rozwiązaniu powyższego ZP przy pomocy zadania ZD jest przekształcenie powyższego zadania do formy dualnej. Aby przeprowadzić przekształcenie należy przeprowadzić transpozycję macierzy A i zamienić b z c .

Jak widać powyżej wektor c ma 2 wartości ujemne więc aby otrzymać postać kanoniczą, w której wszystkie prawe strony równości są nieujemne musimy pomnożyć pierwszy i ostatni wiersz w macierzy współczynników $*$ -1. Przemnożenie to

sprawiło, że współczynniki w kolumnach A_{11} oraz A_{15} stały się ujemne więc nie mogą być wzięte do bazy. W celu utworzenia bazy początkowej dodajemy 2 sztuczne zmienne: A_{16} oraz A_{17}). Dodanie sztucznych zmiennych sprawiło, że musimy wykorzystać algorytm simplex dwufazowy, gdzie w pierwszej fazie należy pozbyć się zmiennych sztucznych z bazy, a w drugiej wyznaczyć rozwiązanie ZD, na podstawie tabeli wyznaczonej w pierwszej fazie. W celu wyrzucenia zmiennych sztucznych z bazy jako funkcję celu przyjmujemy maksymalizację funkcji celu ze współczynnikami: $c = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, -1]$. Następnie wyznaczamy wartość funkcji celu i w ten sposób otrzymujemy poniższą tabelę:

```
>> [ROx, ROy, exitflag] = simplex(c, A, b, 1)
```

c =

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_10	A_11	A_12	A_13	A_14	A_15	A_16	A_17	b
x_16	0	-1	2	1	2	2	2	1	1	0	-1	0	0	0	0	1	0	2
x_12	0	1	-1	2	-2	1	0	0	-2	-2	0	1	0	0	0	0	0	0
x_13	2	2	-1	1	2	2	-2	-2	-2	2	0	0	1	0	0	0	0	1
x_14	-2	0	2	0	1	-1	1	2	-2	1	0	0	0	1	0	0	0	0
x_17	-2	-1	1	-1	2	0	-2	1	1	2	0	0	0	0	-1	0	1	2
oJ - zJ	-2	-2	3	0	4	2	0	2	2	2	-1	0	0	0	-1	0	0	0

Rysunek 1: Tabela początkowa pierwszej fazy.

Po wykonaniu algorytmu simplex dla powyższej tabeli kończymy z optymalnym rozwiązaniem przedstawionym poniżej. Jak widać usuneliśmy zmienne sztuczne (A_{16} , A_{17}) z bazy:

c =

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_10	A_11	A_12	A_13	A_14	A_15	A_16	A_17	b
x_9	0.4	-0.2	-0.6	-0.2	0	0.4	-0.8	-0.6	1	0	0	0	0	-0.4	-0.2	0	0.2	0.4
x_12	-1.6	-0.2	-0.6	0.8	0	1.4	-2.8	0.4	0	0	0	1	0	-0.4	-1.2	0	1.2	2.4
x_13	5.2	2.4	-3.8	1.4	0	3.2	-2.4	-4.8	0	0	0	0	1	-1.2	0.4	0	-0.4	0.2
x_5	-0.2	-0.4	1.3	0.6	1	0.8	1.4	0.8	0	0	-0.5	0	0	0.2	0.1	0.5	-0.1	0.8
x_10	-1	0	-0.5	-1	0	-1	-2	0	0	1	0.5	0	0	0	-0.5	-0.5	0.5	0
oJ - zJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	0

Rysunek 2: Tabela końcowa pierwszej fazy.

Po uzyskaniu rozwiązania 1 fazy możemy przejść do rozwiązywania 2 fazy algorytmu dwufazowego simplex. W tym celu usuwamy z naszej tabeli kolumny zmiennych sztucznych i zmieniamy funkcję celu na $c = -b$ z zadania ZP. Minus przy b wynika z tego, że zadanie w ZD przechodzi na minimalizację funkcji celu, a algorytm simplex maksymalizuje funkcję celu. Po powyższych przekształceniach otrzymujemy poniższą tabelę początkową 2 fazy algorytmu simplex:

c =																	
	-1	-4	-2	-2	-3	-1	-3	-1	-3	-4	-3	-4	-3	-5	-3		
		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_10	A_11	A_12	A_13	A_14	A_15	b
x_9		0.4	-0.2	-0.6	-0.2	0	0.4	-0.8	-0.6	1	0	0	0	0	-0.4	-0.2	0.4
x_12		-1.6	-0.2	-0.6	0.8	0	1.4	-2.8	0.4	0	0	0	1	0	-0.4	-1.2	2.4
x_13		5.2	2.4	-3.8	1.4	0	3.2	-2.4	-4.8	0	0	0	0	1	-1.2	0.4	0.2
x_5		-0.2	-0.4	1.3	0.6	1	0.8	1.4	0.8	0	0	-0.5	0	0	0.2	0.1	0.8
x_10		-1	0	-0.5	-1	0	-1	-2	0	0	1	0.5	0	0	0	-0.5	0
o_j - z_j		4.8	0.6	-15.7	2.6	0	13.8	-27.6	-13.2	0	0	-2.5	0	0	-10.8	-8.9	0

Rysunek 3: Tabela początkowa drugiej fazy.

Po kilku iteracjach algorytmu simplex otrzymujemy poniższe optymalne rozwiązanie:

c =																	
	-1	-4	-2	-2	-3	-1	-3	-1	-3	-4	-3	-4	-3	-5	-3		
		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_10	A_11	A_12	A_13	A_14	A_15	b
x_9		-0.25	-0.5	-0.125	-0.375	0	0	-0.5	0	1	0	0	0	-0.125	-0.25	-0.25	0.375
x_12		-2	0	-1.75	-0.125	-1.25	0	-4.25	0	0	0	0.625	1	-0.125	-0.5	-1.375	1.375
x_6		0.5	0	0.5	0.625	0.75	1	0.75	0	0	0	-0.375	0	0.125	-5.5511e-17	0.125	0.625
x_8		-0.75	-0.5	1.125	0.125	0.5	0	1	1	0	0	-0.125	0	-0.125	0.25	-1.3978e-17	0.375
x_10		-0.5	0	-2.2204e-16	-0.375	0.75	0	-1.25	0	0	1	0.125	0	0.125	-5.5511e-17	-0.375	0.625
o_j - z_j		-12	-6	-7.75	-4.375	-3.75	0	-24.75	0	0	0	-0.625	0	-3.375	-7.5	-10.625	0

Rysunek 4: Tabela końcowa drugiej fazy.

Po znalezieniu optymalnego rozwiązania możemy odczytać z kolumny b rozwiązanie optymalne ZD:

$$RO_y = [0, 0, 0, 0, 0, 0, 0.6250, 0, 0.3750, 0.3750, 0.6250, 0, 1.3750, 0, 0, 0]$$

Na podstawie powyższej tabeli jesteśmy również w stanie wyznaczyć rozwiązanie zadania ZP:

$$RO_x = c(\text{zmienne_bazowe}) * -\text{tabela}(1 : 5, 11 : 15)$$

$$RO_x = [2.3750, 4.0000, -0.3750, -2.5000, -7.6250]$$

4.2 Brak rozwiązania

$$A = \begin{bmatrix} 2 & 2 & -2 & -1 & -2 \\ -2 & 1 & 2 & 0 & -2 \\ -2 & 1 & 0 & 2 & 2 \\ 2 & -1 & 0 & -1 & 1 \\ -1 & -1 & 0 & 2 & -2 \\ -2 & 2 & -1 & 1 & 1 \\ 2 & 0 & 2 & -2 & 1 \\ -1 & 2 & 0 & 2 & 2 \\ 1 & 1 & -2 & 1 & 1 \\ 1 & 1 & 1 & 2 & -2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 5 \\ 3 \\ 3 \\ 5 \\ 1 \\ 2 \\ 2 \\ 5 \\ 2 \\ 5 \\ 4 \end{bmatrix}$$

$$c = [0 \quad 2 \quad 0 \quad -2 \quad 2]$$

Tak jak w powyższym zadaniu z RO przekształcamy ZP do ZD. W tym wypadku tylko w jednej równości mamy po prawej stronie ujemną wartość (-2). Aby znaleźć początkową bazę dodajemy jedną zmienną sztuczną A_{16} , która wejdzie do bazy zamiast zmiennej A_{14} , która w swojej kolumnie ma wartość -1. Po wykonaniu powyższych operacji i wybraniu nowej funkcji celu, która ma za zadanie wyrzucić zmienną sztuczną z bazy otrzymujemy poniższą tabelę początkową dla pierwszej fazy algorytmu:

```
>> [ROx, ROy, exitflag] = simplex(c, A, b, 1)
```

c =

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	
		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_10	A_11	A_12	A_13	A_14	A_15	A_16	b
x_11	2	-2	-2	2	-1	-2	2	-1	1	1	1	1	0	0	0	0	0	0
x_12	2	1	1	-1	-1	2	0	2	1	1	1	0	1	0	0	0	0	2
x_13	-2	2	0	0	0	-1	2	0	-2	1	0	0	1	0	0	0	0	0
x_16	1	0	-2	1	-2	-1	2	-2	-1	-2	0	0	0	0	-1	0	1	2
x_15	-2	-2	2	1	-2	1	1	2	1	-2	0	0	0	0	0	1	0	2
oJ - zJ	1	0	-2	1	-2	-1	2	-2	-1	-2	0	0	0	0	-1	0	0	0

Rysunek 5: Tabela początkowa dla pierwszej fazy.

Po wykonaniu kilku iteracji 1 fazy algorytmu otrzymujemy poniższą tabelę:

c =

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	
		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_10	A_11	A_12	A_13	A_14	A_15	A_16	b
x_4	0	0	0.58824	1	-2.3529	0	3.2353	1.8824	0.29412	1	0.58824	0.70588	0.76471	0	0	0.52941	0	2.4706
x_1	1	0	-0.35294	0	-0.088235	0	0.058824	-0.029412	0.32353	0.5	0.14706	0.17647	-0.058824	0	0	-0.11765	0	0.11765
x_2	0	1	0.17647	0	-0.70589	0	1.4706	0.76471	-0.41176	1	0.17647	0.41176	0.52941	0	0	0.058824	0	0.94118
x_16	0	0	-1.1765	0	-0.79412	0	-0.47059	-2.2647	-1.0882	-3.5	-0.67647	-0.41176	-0.52941	-1	0	-0.058824	1	1.0588
x_6	0	0	1.0588	0	-1.2353	1	0.82353	1.5882	0.52941	-7.8369e-17	0.058824	0.47059	0.17647	0	0	0.35294	0	1.6471
oJ - zJ	0	0	-1.1765	0	-0.79412	0	-0.47059	-2.2647	-1.0882	-3.5	-0.67647	-0.41176	-0.52941	-1	0	-0.058824	0	0

Rysunek 6: Tabela końcowa dla pierwszej fazy.

Jak można zauważyć powyżej, pomimo otrzymania optymalnej wartości funkcji celu zmienna sztuczna A_{16} nie została wyrzucona z bazy więc początkowe ZP nie ma rozwiązania.

5 Testy

Do testów wykorzystane zostały następujące zmienne:

- Liczba różnych wejść - 100.
- Ziarno generatora liczb losowych - 1.
- Liczba miejsc po przecinku do porównania wartości funkcji - 10.

Poniżej przedstawione zostały wyniki dla powyższych parametrów:

- Dokładność dla RO: 100% (określa jak często algorytm zwrócił tę samą wartość dla ROx i ROy co funkcja `linprog`, kiedy `linprog` znalazł RO).
- Dokładność dla braku RO: 100% (określa jak często algorytm nie znalazł RO, kiedy `linprog` nie znalazł RO).
- Liczba znalezionych rozwiązań: 91.
- Liczba braków rozwiązań: 9.

6 Wnioski

Podsumowując powyższa implementacja algorytmu dwufazowego simplex działała dobrze osiągając skuteczność na poziomie 100%. Algorytm potrafił zawsze znaleźć optymalne rozwiązania dla ziarna generatora liczb losowych = 1.

7 Oświadczenie o samodzielności

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

Wojciech Klusek 305934