

Badania Operacyjne

Programowanie liniowe

□ Zadanie programowania liniowego – istota programowania liniowego

2. Istota programowania liniowego:

- Jedna z najpopularniejszych i najbardziej użytecznych technik optymalizacyjnych stosowanych w badaniach operacyjnych
- Programowanie liniowe znajduje powszechne zastosowanie przy rozwiązywaniu problemów alokacji (przydziału) ograniczonych zasobów do konkurencyjnych operacji (zadań), np.:
 - wybór portfela oferowanych usług transportowych przy ustalonych kosztach przewozu i posiadanym taborze
 - określenie rodzaju magazynowanych wyrobów przy uwzględnieniu ich zyskowności oraz możliwościach magazynowych
- Problemy sformułowane w kategoriach programowania liniowego polegają głównie na:
 - maksymalizacji zysku
 - minimalizacji kosztów
 - osiągnięciu zakładanej wartości

□ Zadania programowania liniowego – postacie zadań programowania liniowego

3. Postacie zadań programowania liniowego:

Ogólna postać ZPL:

$x = (x_1, \dots, x_n)$ - wektor zmiennych decyzyjnych

$c = (c_1, \dots, c_n)$ - wektor współczynników funkcji celu

$$\text{Max(Min)} F(x_1, \dots, x_n) = \sum_{j=1}^n c_j x_j$$

Przy ograniczeniach:

$$\begin{cases} & \leq \\ \sum_{j=1}^n a_{i,j} x_j & = b_i; (i = 1, \dots, m) \\ & \geq \\ x_1 \geq 0, \dots, x_n \geq 0 \end{cases}$$

$b = (b_1, \dots, b_m)$ - wektor ograniczeń warunków ograniczających (występujących po prawej stronie równań i nierówności)

$A = [a_{i,j}]_{m \times n}$ - macierz współczynników warunków ograniczających (występujących po lewej stronie równań i nierówności)

□ Zadania programowania liniowego – postacie zadań programowania liniowego

Postać standardowa ZPL:

$$\text{Max} \sum_{j=1}^n c_j x_j$$

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j \leq b_i; & (i = 1, \dots, m) \\ x_j \geq 0; & (j = 1, \dots, n) \end{cases}$$

$$\text{Min} \sum_{j=1}^n c_j x_j$$

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j \geq b_i; & (i = 1, \dots, m) \\ x_j \geq 0; & (j = 1, \dots, n) \end{cases}$$

Postać kanoniczna ZPL:

$$\text{Max}(\text{Min}) \sum_{j=1}^n c_j x_j$$

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j = b_i; & (i = 1, \dots, m) \\ x_j \geq 0; & (j = 1, \dots, n) \end{cases}$$

w zapisie macierzowym:

$$\text{Max}(\text{Min}) \sum_{j=1}^n c_j x_j = (c, x)$$

$$\begin{cases} A \cdot x^T = b^T \\ x^T \geq 0 \end{cases}$$

□ Zadania programowania liniowego – postacie zadań programowania liniowego

Każdą postać ZPL możemy sprowadzić do postaci kanonicznej poprzez wprowadzenie zmiennych swobodnych.

Dla warunków postaci \leq , jeżeli i-ty warunek jest taką nierównością, wprowadzamy zmienną swobodną: $x_{n+i} \geq 0$; $x_{n+i} = b_i - \sum_{j=1}^n a_{i,j}x_j$ ze współczynnikiem (+1)

np. $2x_1 + 3x_2 \leq 6$ zamieniamy na warunek $2x_1 + 3x_2 + x_3 = 6$

Dla warunków postaci \geq , jeżeli i-ty warunek jest taką nierównością, wprowadzamy zmienną swobodną: $x_{n+i} \geq 0$; $x_{n+i} = \sum_{j=1}^n a_{i,j}x_j - b_i$ ze współczynnikiem (-1)

np. $7x_1 - x_2 \geq 2$ zamieniamy na warunek $7x_1 - x_2 - x_4 = 2$

Uwaga: Do funkcji celu zmienne swobodne wchodzi z współczynnikami równymi zero, w przykładzie $c_3 = 0, c_4 = 0$.

□ Zadania programowania liniowego – dualność w programowaniu liniowym

4. Dualność w programowaniu liniowym:

Z każdym zadaniem PL sprzężone jest pewne inne zadanie PL, zwane **zadaniem dualnym** ZD PL.

Dla zadania pierwotnego (ZP) na maksimum w postaci standardowej:

$$\begin{aligned} & \text{Max } \sum_{j=1}^n c_j x_j \\ & \begin{cases} \sum_{j=1}^n a_{ij} x_j \leq b_i; & (i = 1, \dots, m) \\ x_j \geq 0; & (j = 1, \dots, n) \end{cases} \end{aligned}$$

Zadaniem dualnym ZD PL będzie zadanie:

$$\begin{aligned} & \text{Min } \sum_{i=1}^m b_i y_i \\ & \begin{cases} \sum_{i=1}^m a_{ij} y_i \geq c_j; & (j = 1, \dots, n) \\ y_i \geq 0; & (i = 1, \dots, m) \end{cases} \end{aligned}$$

□ Zadania programowania liniowego – dualność w programowaniu liniowym

Z relacji zachodzących pomiędzy zadaniem ZP (pierwotnym) i ZD (zadaniem dualnym) wynika, że:

- w zadaniu dualnym jest tyle zmiennych, ile warunków w zadaniu pierwotnym (każdemu warunkowi odpowiada jedna zmienna)
- w zadaniu dualnym jest tyle warunków, ile zmiennych w zadaniu pierwotnym
- wagi funkcji celu zadania pierwotnego są wyrazami wolnymi (prawymi ograniczeniami warunków) w zadaniu dualnym
- wyrazy wolne zadania pierwotnego są wagami funkcji celu w zadaniu dualnym
- macierz współczynników zadania dualnego jest transpozycją macierzy współczynników zadania pierwotnego
- jeżeli zadanie pierwotne jest ma maksimum, to zadanie dualne jest na minimum (i odwrotnie)

□ Zadania programowania liniowego – dualność w programowaniu liniowym

Stosuje się także następujące dodatkowe reguły tworzenia zadania dualnego:

- jeżeli z ZP i-ty warunek jest równością, to odpowiadająca mu zmienna w zadaniu dualnym nie ma ograniczeń (przyjmuje dowolne wartości)
- jeżeli z ZP i-ty warunek jest typową nierównością, to odpowiadająca mu zmienna w zadaniu dualnym jest nieujemna
- jeżeli z ZP i-ty warunek jest nietypową nierównością, to odpowiadająca mu zmienna w zadaniu dualnym jest niedodatnia ($y_i \leq 0$)
- jeżeli w ZP na zmienną x_j nie nałożono ograniczeń, to j-ty warunek w ZD jest równością
- jeżeli w ZP zmienna $x_j \geq 0$ (jest nieujemna), to j-ty warunek w ZD jest typową nierównością
- jeżeli w ZP zmienna $x_j \leq 0$ (jest niedodatnia), to j-ty warunek w ZD jest nietypową nierównością

□ Programowanie liniowe – dualność w programowaniu liniowym – przykład konstrukcji ZD PL

Zadanie pierwotne - ZP PL

$$f(x_1, x_2, x_3, x_4) = 2x_1 - x_2 + 5x_3 - 2x_4 \rightarrow \min$$

$$\begin{cases} x_1 + x_2 + 2x_3 + 5x_4 \leq 20 \\ 2x_1 - x_2 - x_4 \geq 15 \\ 3x_2 + x_3 - 2x_4 = 14 \\ x_1, x_3, x_4 \geq 0, x_2 - \text{dowolne} \end{cases}$$

Zadanie dualne - ZD PL

$$g(y_1, y_2, y_3) = 20y_1 + 15y_2 + 14y_3 \rightarrow \max$$

$$\begin{cases} y_1 + 2y_2 \leq 2 \\ y_1 - y_2 + 3y_3 = -1 \\ 2y_1 + y_3 \leq 5 \\ 5y_1 - y_2 - 2y_3 \leq -2 \\ y_1 \leq 0, y_2 \geq 0, y_3 - \text{dowolne} \end{cases}$$

□ Zadania programowania liniowego – dualność w programowaniu liniowym

5. Twierdzenia o Dualności:

Twierdzenie 1 (o istnieniu)

Jeżeli ZP i ZD mają rozwiązania dopuszczalne, to oba mają rozwiązania optymalne. Jeżeli natomiast chociaż jedno z nich nie ma rozwiązania dopuszczalnego, to obydwa nie mają rozwiązań dopuszczalnych.

Twierdzenie 2

Jeżeli x_1, \dots, x_n jest rozwiązaniem dopuszczalnym zadania pierwotnego, a y_1, \dots, y_m rozwiązaniem dopuszczalnym zadania dualnego, to pomiędzy wartościami funkcji celu zachodzi nierówność:

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i$$

Dla rozwiązań dopuszczalnych wartość funkcji celu ZP nie może być większa od wartości funkcji celu ZD

□ Zadania programowania liniowego – dualność w programowaniu liniowym

Twierdzenie 3 (o optymalności)

Jeżeli istnieją dwa takie rozwiązania dopuszczalne x_1^*, \dots, x_n^* (ZP) i y_1^*, \dots, y_m^* (ZD), że

$$\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*,$$

to obydwa rozwiązania są rozwiązaniami optymalnymi.

Twierdzenie 4 (o równowadze)

Jeżeli x_1, \dots, x_n jest rozwiązaniem dopuszczalnym zadania pierwotnego oraz y_1, \dots, y_m rozwiązaniem dopuszczalnym zadania dualnego, to aby te rozwiązania były rozwiązaniami optymalnymi wystarczy, że spełniane będą następujące warunki:

$$(1) \sum_{j=1}^n a_{ij} x_j < b_i \Rightarrow y_i = 0$$

$$(2) \sum_{i=1}^m a_{ij} y_i > c_j \Rightarrow x_j = 0$$

$$(3) y_i > 0 \Rightarrow \sum_{j=1}^n a_{ij} x_j = b_i$$

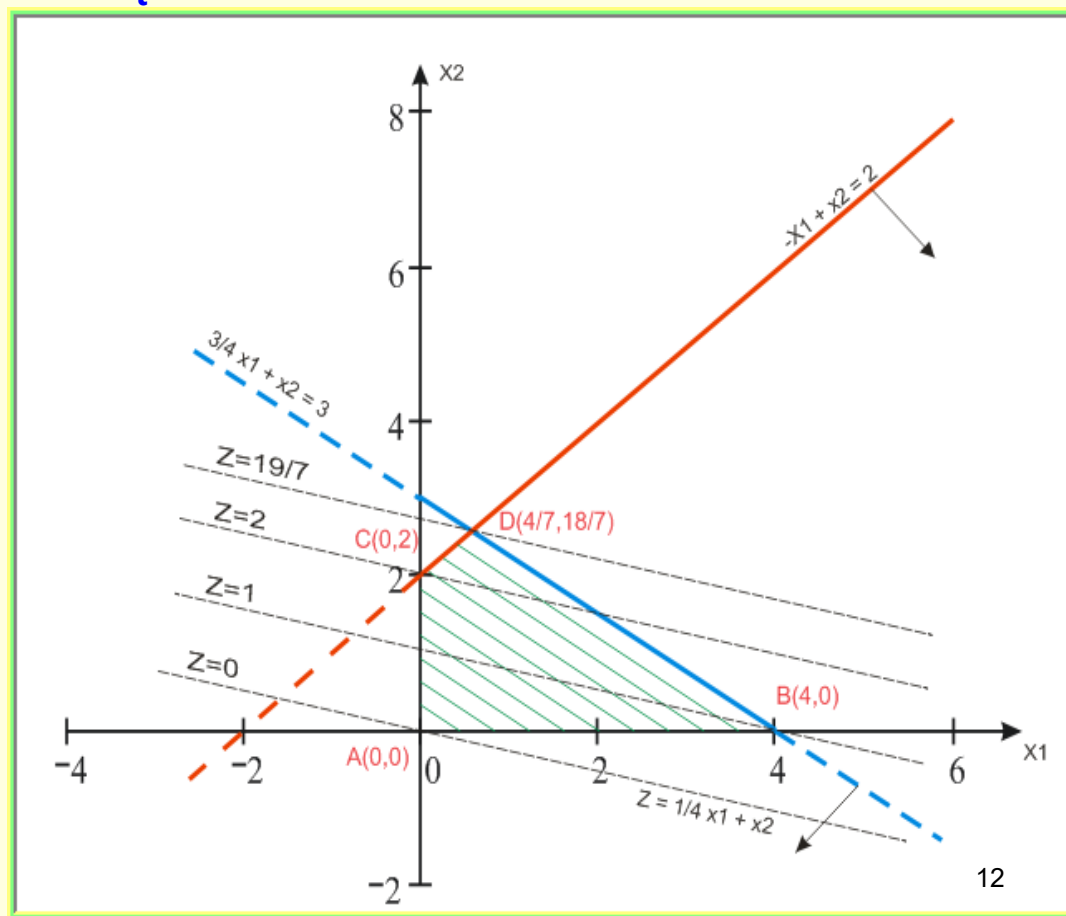
$$(4) x_j > 0 \Rightarrow \sum_{i=1}^m a_{ij} y_i = c_j$$

□ Programowanie liniowe – graficzna interpretacja zadań programowania liniowego - metoda geometryczna poszukiwania rozwiązań

1. Przykład zadania programowania liniowego ZPL - mającego jednoznaczne tylko jedno rozwiązanie.

$$f(x_1, x_2) = z = \frac{1}{4}x_1 + x_2 \rightarrow \max$$

$$\begin{cases} -x_1 + x_2 \leq 2 \\ \frac{3}{4}x_1 + x_2 \leq 3 \\ x_1, x_2 \geq 0 \end{cases}$$

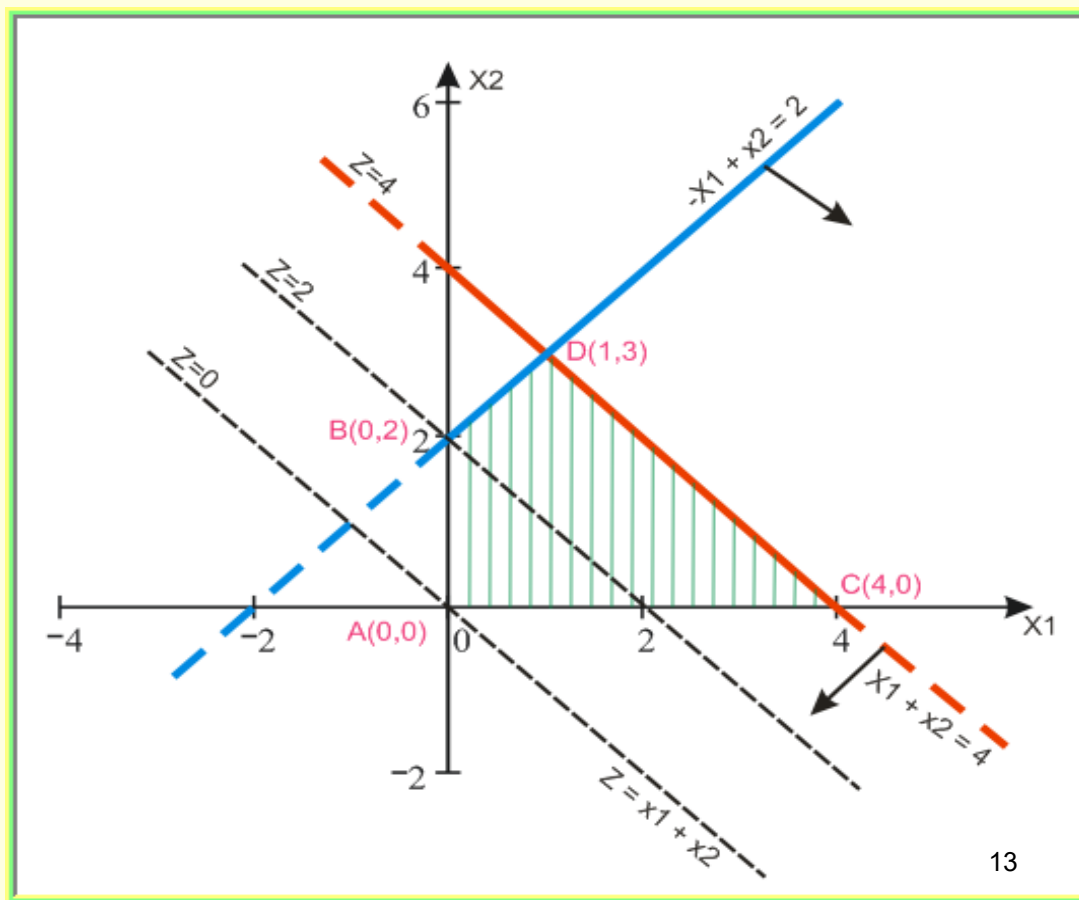


□ Programowanie liniowe – graficzna interpretacja zadań programowania liniowego - metoda geometryczna poszukiwania rozwiązań

2. Przykład zadania programowania liniowego ZPL - mającego niejednoznacznie nieskończenie wiele rozwiązań.

$$f(x_1, x_2) = z = x_1 + x_2 \rightarrow \max$$

$$\begin{cases} x_1 + x_2 \leq 4 \\ -x_1 + x_2 \leq 2 \\ x_1, x_2 \geq 0 \end{cases}$$

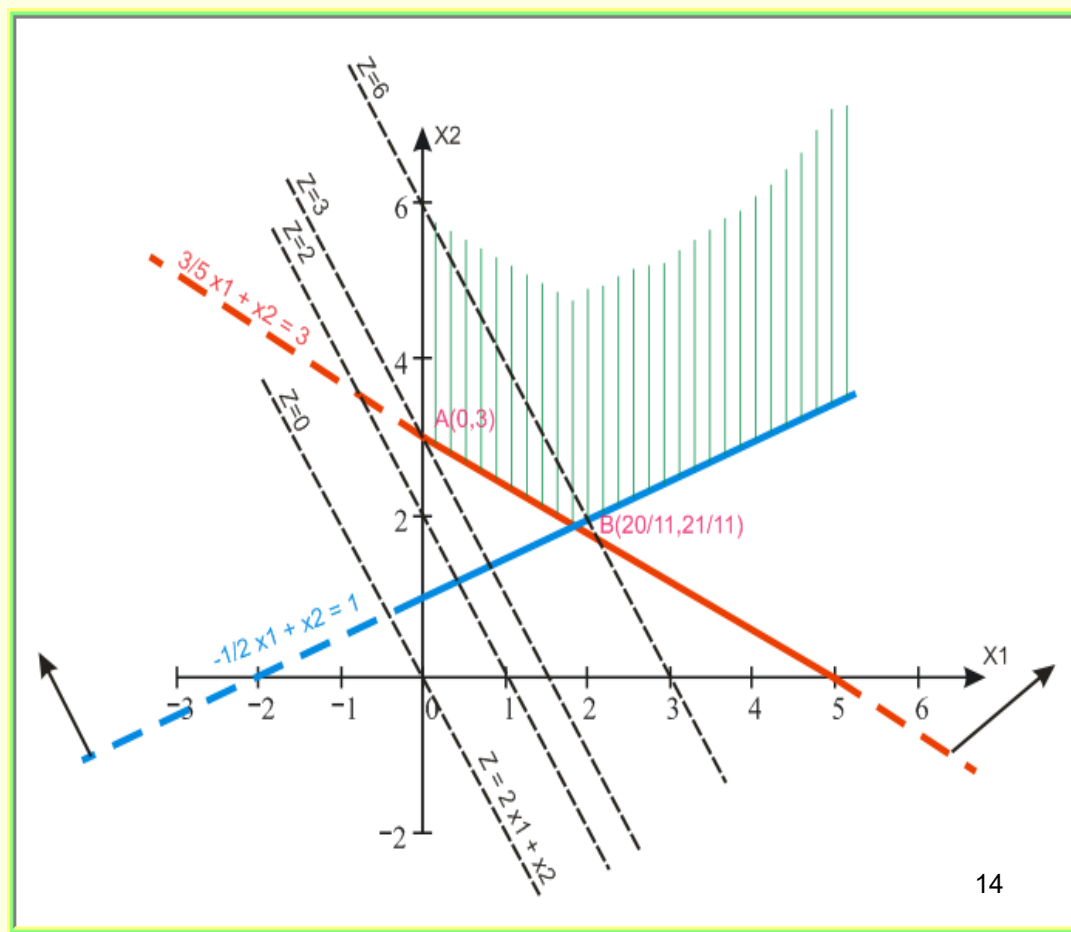


□ Programowanie liniowe – graficzna interpretacja zadań programowania liniowego - metoda geometryczna poszukiwania rozwiązań

3. Przykład zadania programowania liniowego ZPL – nie mającego żadnych rozwiązań.

$$f(x_1, x_2) = z = 2x_1 + x_2 \rightarrow \max$$

$$\begin{cases} \frac{3}{5}x_1 + x_2 \geq 3 \\ -\frac{1}{2}x_1 + x_2 \geq 1 \\ x_1, x_2 \geq 0 \end{cases}$$



□ Zadania programowania liniowego – idea metody simpleks

6. Algorytm metody Simpleks:

Metoda simpleks jest podstawową metodą znajdowania optymalnych rozwiązań zadań programowania liniowego. Jest to metoda ogólna, pozwalająca rozwiązać każde zadanie PL.

Polega ona na sekwencyjnym (ściśle określonym - ukierunkowanym przeglądzie tzw. rozwiązań bazowych)

Rozwiązanie bazowe jest związane z postacią kanoniczną zadania PL.

$$\text{Max}(\text{Min}) \sum_{j=1}^n c_j x_j$$

$$\begin{cases} A \cdot x = b \\ x \geq 0 \end{cases}$$

gdzie:

$$x = \begin{bmatrix} x_1 \\ \dots \\ x_n \end{bmatrix} - \text{wektor zmiennych decyzyjnych (w zapisie kolumnowym),}$$

$$b = \begin{bmatrix} b_1 \\ \dots \\ b_n \end{bmatrix} - \text{wektor prawych ograniczeń w warunkach}$$

$$c = [c_1 \quad \dots \quad c_n] - \text{wektor współczynników funkcji celu (w zapisie wierszowym)}$$

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} - \text{macierz współczynników warunków ograniczających po prawej stronie}$$

równości

□ Zadania programowania liniowego – dualność w programowaniu liniowym

Niech B oznacza **bazę**, czyli macierz kwadratową m -tego stopnia składającą się z m - liniowo niezależnych kolumn macierzy A ($\det(B) \neq 0$).

Jej kolumny nazywa się kolumnami bazowymi, zaś pozostałe kolumny macierzy A nie bazowymi. Zmienne związane z kolumnami bazowymi nazywamy zmiennymi bazowymi, zaś pozostałe nie bazowymi.

Oznaczmy przez Z_B - zbiór zmiennych bazowych, zaś przez Z_N - zbiór zmiennych nie bazowych.

Z każdą bazą B układu równań $Ax = b$ jest związane rozwiązanie bazowe. Jeżeli układ $Ax = b$ jest niesprzeczny oraz $n > m$, to układ ten ma nieskończenie wiele rozwiązań, ale skończoną liczbę rozwiązań bazowych. Dla m - równań z n - niewiadomymi ma co najwyżej

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

Wektor zmiennych decyzyjnych oraz macierz współczynników można przedstawić teraz przy zadanej bazie B następująco: $x = (x_B, x_N)$, $A = (B, N)$

Wówczas układ równań $Ax = b$ zapiszemy w postaci: $B \cdot x_B + N \cdot x_N = b$

Mnożąc lewostronnie przez macierz B^{-1} , otrzymujemy postać bazową: $I \cdot x_B + W \cdot x_N = b^*$

gdzie: $I = B^{-1} \cdot B$, $W = B^{-1} \cdot N$, $b^* = B^{-1} \cdot b$

Z postaci bazowej łatwo wyznaczyć rozwiązanie bazowe: $x_N = 0$, $x_B = b^* = B^{-1} \cdot b$.

Jeżeli dla danej bazy B : $x_B = b^* = B^{-1} \cdot b \geq 0$, to rozwiązanie bazowe jest rozwiązaniem dopuszczalnym.

□ Zadania programowania liniowego – dualność w programowaniu liniowym

Idea metody simpleks:

Zastosowanie podejścia pełnego przeglądu zbioru rozwiązań bazowych jest nieefektywne ze względu na liczbę tych rozwiązań oraz ze względu na wielkość układu równań jaki należy przekształcać.

Jeżeli $n=20$ i $m=10$, to rozwiązań bazowych może być:
$$\binom{20}{10} = \frac{20!}{10! \cdot 10!} = 184756$$

W metodzie simpleks stosujemy przegląd ukierunkowany zbioru rozwiązań bazowych. Przechodzimy od jednego rozwiązania bazowego dopuszczalnego do drugiego, o którym wiemy, że jest nie gorsze od poprzedniego (pomijając niedopuszczalne i te gorsze od aktualnie rozpatrywanego).

Przegląd ukierunkowany sprowadza się do realizacji następujących kroków (jeżeli zadanie ma rozwiązanie optymalne)

Krok A: Wyznaczyć rozwiązanie wejściowe – dopuszczalne i bazowe

Krok B: Sprawdzić, czy aktualne rozwiązanie bazowe jest optymalne. Jeżeli tak, to koniec obliczeń, jeżeli nie, to przejście do kroku C

Krok C: Przejść do sąsiedniego rozwiązania bazowego, o którym wiadomo, że jest nie gorsze od poprzedniego i powrót do kroku B.

Dwie bazy nazywamy sąsiednimi jeżeli różnią się tylko jedną kolumną macierzy **A**. Podobnie dwa rozwiązania bazowe nazywamy sąsiednimi, jeżeli różnią się tylko jedną zmienną bazową.

W sensie rachunkowym przechodzenie od jednego rozwiązania bazowego do drugiego sąsiedniego polega na przekształcaniu układu równań: $I \cdot x_B + W \cdot x_N = b^*$ od jednej postaci bazowej do drugiej. Najlepiej, gdy początkowa postać kanoniczna jest także postacią bazową.

□ Zadania programowania liniowego – dualność w programowaniu liniowym

Początkowe rozwiązanie bazowe:

Punktem wyjścia metody simpleks jest uzyskanie początkowego rozwiązania bazowego. W tym celu należy przekształcić je do postaci bazowej z nieujemnym wektorem wyrazów wolnych \mathbf{b} . Przyjmując, że zmienne bazowe są związane z wektorami jednostkowymi uzyskamy bardzo łatwo początkowe rozwiązanie bazowe: $x_B = \mathbf{b}^* = \mathbf{B}^{-1} \cdot \mathbf{b} = \mathbf{I}^{-1} \cdot \mathbf{b} = \mathbf{b}$, $x_N = 0$.

Jeżeli i -te prawe ograniczenie jest ujemne, to można przemnożyć obustronnie to równanie (warunek) przez (-1) .

Gdy macierz współczynników postaci kanonicznej nie zawiera macierzy jednostkowej, którą można uznać jako bazę, to możemy:

- (1) tak przekształcić układ równań (stosując reguły eliminacji Gaussa), aby zawierał on macierz jednostkową
- (2) albo sztucznie utworzyć taką macierz, uzupełniając macierz współczynników \mathbf{A} o odpowiednią liczbę brakujących wektorów jednostkowych – co często jest praktyczniejsze – jest to tzw. metoda sztucznej bazy (rozszerzamy wtedy listę zmiennych decyzyjnych o tzw. zmienne sztuczne).

Uwaga:

- Waga przy zmiennej sztucznej w funkcji celu jest taka, że nieopłacalne jest pozostawienie jej w rozwiązaniu optymalnym (duża liczba dodatnia dla minimum lub ujemna i duża co do modułu dla zadania na maksimum)
- Optymalne rozwiązanie zadania ze zmiennymi sztucznymi (pomocniczego) wyznacza optymalne rozwiązanie zadania początkowego, jeśli tylko wszystkie zmienne sztuczne są w rozwiązaniu optymalnym zerowe.
- jeżeli choć jedna zmienna sztuczna jest dodatnia, to początkowe rozwiązanie jest sprzeczne.

□ Zadania programowania liniowego – algorytm metody simpleks

Zadanie ZPL (przykład)

Przedsiębiorstwo transportowe dysponuje 10 ciężarówkami o ładowności 10[t]. Klient zlecił przedsiębiorstwu przewóz 66 ton ładunków w opakowaniach po 3[t] oraz 20 ton ładunków w opakowaniach po 2[t].

W jaki sposób należy załadować towar na ciężarówki, aby zrealizować zamówienie klienta, minimalizując łączną niewykorzystaną ładowność ciężarówek niezbędnych do przewozu towarów ?

Sposoby załadunku	(1)	(2)	(3)	(4)
ładunek – opakowania 3[t]	3	2	1	0
ładunek – opakowania 2[t]	0	2	3	5
Niewykorzystana ładowność [t]	1	0	1	0

Model matematyczny – problemu decyzyjnego:

$$F(x_1, x_2, x_3, x_4) = 1 \cdot x_1 + 0 \cdot x_2 + 1 \cdot x_3 + 0 \cdot x_4 \rightarrow \min$$

$$\begin{cases} x_1 + x_2 + x_3 + x_4 \leq 10 \\ 3 \cdot (3 \cdot x_1 + 2 \cdot x_2 + 1 \cdot x_3 + 0 \cdot x_4) = 66 \\ 2 \cdot (0 \cdot x_1 + 2 \cdot x_2 + 3 \cdot x_3 + 5 \cdot x_4) = 20 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{cases} \Leftrightarrow \begin{cases} x_1 + x_2 + x_3 + x_4 \leq 10 \\ 9 \cdot x_1 + 6 \cdot x_2 + 3 \cdot x_3 + 0 \cdot x_4 = 66 \\ 0 \cdot x_1 + 4 \cdot x_2 + 6 \cdot x_3 + 10 \cdot x_4 = 20 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{cases}$$

□ Zadania programowania liniowego – algorytm metody simpleks

Postać kanoniczna:

$$F(x_1, x_2, x_3, x_4) = 1 \cdot x_1 + 0 \cdot x_2 + 1 \cdot x_3 + 0 \cdot x_4 + 0 \cdot x_5 \rightarrow \min$$

$$\begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 = 10 \\ 9 \cdot x_1 + 6 \cdot x_2 + 3 \cdot x_3 + 0 \cdot x_4 = 66 \\ 0 \cdot x_1 + 4 \cdot x_2 + 6 \cdot x_3 + 10 \cdot x_4 = 20 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

Postać kanoniczna – bazowa, po wprowadzeniu zmiennych sztucznych:

Wprowadzamy 2 zmienne sztuczne (dla których oczekujemy wartości zerowej) $x_6 = 0$ oraz $x_7 = 0$ ze współczynnikami $c_6 = M$ oraz $c_7 = M$, $M > 0$ ($M \rightarrow \infty$) - dla zadania na minimum.

Gdyby zadanie ZPL było na maksimum, to $c_6 = -M, c_7 = -M$.

$$F(x_1, x_2, x_3, x_4) = 1 \cdot x_1 + 0 \cdot x_2 + 1 \cdot x_3 + 0 \cdot x_4 + 0 \cdot x_5 + M \cdot x_6 + M \cdot x_7 \rightarrow \min$$

$$\begin{cases} 1 \cdot x_1 + 1 \cdot x_2 + 1 \cdot x_3 + 1 \cdot x_4 + 1 \cdot x_5 + 0 \cdot x_6 + 0 \cdot x_7 = 10 \\ 9 \cdot x_1 + 6 \cdot x_2 + 3 \cdot x_3 + 0 \cdot x_4 + 0 \cdot x_5 + 1 \cdot x_6 + 0 \cdot x_7 = 66 \\ 0 \cdot x_1 + 4 \cdot x_2 + 6 \cdot x_3 + 10 \cdot x_4 + 0 \cdot x_5 + 0 \cdot x_6 + 1 \cdot x_7 = 20 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0 \end{cases}$$

□ Zadania programowania liniowego – algorytm metody simpleks

Algorytm metody Simpleks:

Krok A – znaleźć początkowe rozwiązanie bazowe dopuszczalne

Krok B – sprawdzić optymalność aktualnego rozwiązania bazowego

- Jeżeli aktualne bazowe jest optymalne, to **koniec obliczeń**
- Jeżeli nie to przejść do **kroku C**

Krok C – Wyznaczyć sąsiednie nie gorsze od poprzedniego rozwiązanie bazowe

Powrót do - **Kroku B**

□ Zadania programowania liniowego – algorytm metody simpleks

Krok A

Pierwsze – początkowe rozwiązanie bazowe:

Macierz współczynników w warunkach ograniczających:

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

$$\text{Pierwsza baza: } B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Zmienne bazowe: } x_B = \begin{bmatrix} x_5 \\ x_6 \\ x_7 \end{bmatrix}, \text{ Zmienne niebazowe: } x_N = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Pierwsze rozwiązanie bazowe:

$$x_N = 0, x_B = \begin{bmatrix} x_5 \\ x_6 \\ x_7 \end{bmatrix} = b^* = b = \begin{bmatrix} 10 \\ 66 \\ 20 \end{bmatrix}$$

Krok B

Sprawdzenie optymalności aktualnego rozwiązania bazowego:

Sprawdzanie optymalności aktualnego rozwiązania bazowego oraz tworzenie kolejnych rozwiązań bazowych dopuszczalnych wygodnie jest zilustrować tzw. Tablicą simpleksową.

Pierwsza tablica simpleksowa: $T^{(1)} = [t_{ij}]_{i=0, \dots, m; j=0, \dots, n}$



c_j		1	0	1	0	0	M	M	Kolumna tablicy simpleksowej dla wyrazów wolnych (zerowa) $b_i^* = t_{i,0}$
$c_s = c_j, j \in Z_s$	Zmienne Bazowe	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
				Elementy tablicy simpleksowej $t_{i,j}, i \in 1, \dots, m, j = 1, \dots, n$					
0	x_5	1	1	1	1	1	0	0	10
M	x_6	9	6	3	0	0	1	0	66
M	x_7	0	4	6	10	0	0	1	20
$z_j = \sum_{i \in Z_s} c_i \cdot t_{i,j}$		$z_1 = 0 \cdot 1 + M \cdot 9 + M \cdot 0 = 9M$	10M	9M	10M	0	M	M	$F(x) =$
Wiersz (zerowy) tablicy simpleksowej kryterium optymalności $t_{0,j} = c_j - z_j, j = 1, \dots, n$		$-9M + 1$	$-10M$	$-9M + 1$	$-10M$	0	0	0	$x_5 = 0 \cdot 10 + M \cdot 66 + M \cdot 20 = 86M$

Simpleksowe kryterium optymalności:

1. Dla zadania z funkcją celu postaci **maksimum**:

dla każdego $j \in Z_N, t_{0,j} \leq 0$

2. Dla zadania z funkcją celu postaci **minimum**:

dla każdego $j \in Z_N, t_{0,j} \geq 0$

Dla naszego zadania wszystkie współczynniki optymalności są ujemne, więc początkowe rozwiązanie bazowe nie jest optymalne.

Należy zatem przejść do **kroku (C)** - ustalić którą zmienną z aktualnych niebazowych należy do bazy wprowadzić, aby poprawić aktualne rozwiązanie, a którą z aktualnej bazy usunąć.

O tym mówi tzw. simpleksowe **kryterium wejścia** do bazy i **kryterium wyjścia** z bazy.

□ Zadania programowania liniowego – algorytm metody simpleks

Poprawa aktualnego rozwiązania bazowego – wyznaczenie sąsiedniego rozwiązania bazowego dopuszczalnego, które jest nie gorsze od aktualnego

Kryterium wejścia:

Do bazy należy wprowadzić, taką zmienną z niebazowych, dla której:

1. Dla zadania na **maksimum** zachodzi warunek: $t_{0,k} = \max \{t_{0,j}, j \in Z_N\}$
2. Dla zadania **na minimum** zachodzi warunek: $t_{0,k} = \min \{t_{0,j}, j \in Z_N\}$

Dla naszego zadania należy wprowadzić albo zmienną x_2 albo x_4 (wprowadzamy x_2 , gdyż jest to zmienna właściwa, a nie bilansująca zmienna swobodna jak x_4).

$$t_{0,2} = t_{0,4} = \min \{-9M + 1, -10M, -9M + 1, -10M\} = -10M$$

$k = 2$ - kolumna 2 staje się tzw. kolumną centralną tablicy simpleksowej

□ Zadania programowania liniowego – algorytm metody simpleks

Kryterium wyjścia:

Z aktualnej bazy należy usunąć tę zmienną, dla której:

$$\frac{t_{r,0}}{t_{r,k}} = \min \left\{ \frac{t_{i,0}}{t_{i,k}}; t_{i,k} > 0 \right\}$$

Dla naszego zadania $\min \left\{ \frac{t_{i,0}}{t_{i,k}}; t_{i,k} > 0 \right\} = \min \left\{ \frac{10}{1}, \frac{66}{6}, \frac{20}{4} \right\} = 5$, dla $r = 3$, a więc dla

trzeciej aktualnie zmiennej bazowej: x_7 , którą z bazy należy usunąć

$t_{r,k} = t_{3,2}$ - staje się tzw. elementem centralnym (ważnym dla dalszych przekształceń tablicy simpleksowej) przy wyznaczaniu nowego poprawionego rozwiązania bazowego.

□ Zadania programowania liniowego – algorytm metody simpleks

Przekształcenia tablicy simpleksowej w celu uzyskania nowego - lepszego rozwiązania bazowego.

$T'_{i,j} = [t'_{i,j}]_{i=1,\dots,m; j=0,\dots,n}$ - jest nową tablicą simpleksową

- $t'_{r,j} = \frac{t_{r,j}}{t_{r,k}}$; $j = 0, \dots, n$ - dla wiersza odpowiadającego wierszowi centralnemu, czyli dla zmiennej wyprowadzanej z aktualnej bazy
- $t'_{i,j} = t_{i,j} - t_{i,k} \cdot t'_{r,j}$; $i \neq r$; $j = 0, \dots, n$ - dla wierszy pozostałych

Dla naszego zadania: element centralny $t_{3,2} = 4$

$$t'_{3,j} = \left[0 \quad 1 \quad \frac{3}{2} \quad \frac{5}{2} \quad 0 \quad 0 \quad 1 \quad | \quad 5 \right]$$

$$\begin{aligned} t'_{1,j} &= \left[1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad | \quad 10 \right] - 1 \cdot \left[0 \quad 1 \quad \frac{3}{2} \quad \frac{5}{2} \quad 0 \quad 0 \quad 1 \quad | \quad 5 \right] = \\ &= \left[1 \quad 0 \quad -\frac{1}{2} \quad -\frac{3}{2} \quad 1 \quad 0 \quad -1 \quad | \quad 5 \right] \end{aligned}$$

$$\begin{aligned} t'_{2,j} &= \left[9 \quad 6 \quad 3 \quad 0 \quad 0 \quad 1 \quad 0 \quad | \quad 66 \right] - 6 \cdot \left[0 \quad 1 \quad \frac{3}{2} \quad \frac{5}{2} \quad 0 \quad 0 \quad 1 \quad | \quad 5 \right] = \\ &= \left[9 \quad 0 \quad -6 \quad -15 \quad 0 \quad 1 \quad -6 \quad | \quad 36 \right] \end{aligned}$$

Druga tablica simpleksowa: $T^{(2)}$

c_j		1	0	1	0	0	M	M	$b_i^* = t_{i,0}$
$c_B = c_j, j \in Z_B$	Zmienne Bazowe	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
		Elementy tablicy simpleksowej $t_{i,j}, i \in 1, \dots, m, j = 1, \dots, n$							
0	x_5	1	0	$-\frac{1}{2}$	$-\frac{3}{2}$	1	0	-1	5
M	x_6	9	0	-6	-15	0	1	-6	36
0	x_2	0	1	$\frac{3}{2}$	$\frac{5}{2}$	0	0	1	5
$Z_j = \sum_{i \in Z_B} c_i \cdot t_{i,j}$		9M	0	-6M	-15M	0	M	-6M	$F(x) =$
$t_{0,j} = c_j - Z_j$		-9M+1	0	6M+1	15M	0	0	7M	$x_0 = 36M$

Drugie rozwiązanie bazowe:

$$x_N = 0, x_B = \begin{bmatrix} x_5 \\ x_6 \\ x_2 \end{bmatrix} = b^* = b = \begin{bmatrix} 5 \\ 36 \\ 5 \end{bmatrix} \quad f(x) = 0 \cdot 5 + M \cdot 36 + M \cdot 5 = 41M$$

Kryterium optymalności: istnieje jeszcze jeden współczynnik $t_{0,1}$, dla zmiennej niebazowej x_1 , który jest ujemny, zatem aktualne rozwiązanie dalej nie jest optymalne.

Zastosowanie kryterium wejścia:

Tę zmienną: x_1 - należy zatem wprowadzić do bazy – zgodnie z kryterium wejścia

Zastosowanie kryterium wyjścia:

$$\frac{t_{r,0}}{t_{r,1}} = \min \left\{ \frac{5}{1}, \frac{36}{9} \right\} = 4 - \text{z bazy usuwamy } r=2 \text{ drugą zmienną bazową: } x_6$$

□ Zadania programowania liniowego – algorytm metody simpleks

Nowe – trzecie rozwiązanie bazowe

dla naszego zadania - element centralny: $t_{2,1} = 9$

$$t'_{2,j} = \left[1 \quad 0 \quad -\frac{2}{3} \quad -\frac{5}{3} \quad 0 \quad \frac{1}{9} \quad -\frac{2}{3} \quad | \quad 4 \right]$$

$$\begin{aligned} t'_{1,j} &= \left[1 \quad 0 \quad -\frac{1}{2} \quad -\frac{3}{2} \quad 1 \quad 0 \quad -1 \quad | \quad 5 \right] - 1 \cdot \left[1 \quad 0 \quad -\frac{2}{3} \quad -\frac{5}{3} \quad 0 \quad \frac{1}{9} \quad -\frac{2}{3} \quad | \quad 4 \right] = \\ &= \left[0 \quad 0 \quad \frac{1}{6} \quad \frac{1}{6} \quad 1 \quad -\frac{1}{9} \quad -\frac{1}{3} \quad | \quad 1 \right] \end{aligned}$$

$$\begin{aligned} t'_{3,j} &= \left[1 \quad 0 \quad \frac{3}{2} \quad \frac{5}{2} \quad 0 \quad 0 \quad 1 \quad | \quad 5 \right] - 0 \cdot \left[1 \quad 0 \quad -\frac{2}{3} \quad -\frac{5}{3} \quad 0 \quad \frac{1}{9} \quad -\frac{2}{3} \quad | \quad 4 \right] = \\ &= \left[1 \quad 0 \quad \frac{3}{2} \quad \frac{5}{2} \quad 0 \quad 0 \quad 1 \quad | \quad 5 \right] \end{aligned}$$

Trzecia tablica simpleksowa: $T^{(3)}$

c_j		1	0	1	0	0	M	M	$b_i^* = t_{i,0}$
$c_B = c_j, j \in Z_B$	Zmienne Bazowe	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
		Elementy tablicy simpleksowej $t_{i,j}, i \in 1, \dots, m, j = 1, \dots, n$							
0	x_5	0	0	$\frac{1}{6}$	$\frac{1}{6}$	1	$-\frac{1}{9}$	$-\frac{1}{3}$	1
1	x_1	1	0	$-\frac{2}{3}$	$-\frac{5}{3}$	0	$\frac{1}{9}$	$-\frac{2}{3}$	4
0	x_2	0	1	$\frac{3}{2}$	$\frac{5}{2}$	0	0	1	5
$Z_j = \sum_{i \in Z_B} c_i \cdot t_{i,j}$		1	0	$-\frac{2}{3}$	$-\frac{5}{3}$	0	$\frac{1}{9}$	$-\frac{2}{3}$	$F(x) =$
$t_{0,j} = c_j - Z_j$		0	0	$\frac{5}{3}$	$\frac{5}{3}$	0	$M - \frac{1}{9}$	$M + \frac{2}{3}$	$x_0 = 4$

Trzecie rozwiązanie bazowe:

$$x_N = 0, x_B = \begin{bmatrix} x_5 \\ x_1 \\ x_2 \end{bmatrix} = b^* = b = \begin{bmatrix} 1 \\ 4 \\ 5 \end{bmatrix} \quad f(x) = 0 \cdot 1 + 1 \cdot 4 + 0 \cdot 5 = 4$$

Sprawdzenie kryterium optymalności:

wszystkie współczynniki $t_{0,j}, j \in Z_N$, dla zmiennych niebazowych, zatem aktualne rozwiązanie jest **optymalne**.

Należy zatem wysłać 4 samochody załadowane sposobem pierwszym oraz 5 załadowanych sposobem drugim, aby zrealizować zamówienie klienta, przy minimalnej niewykorzystanej ładowności środków transportu wynoszącej 4 [tony].