PROGRAMOWANIE

LINIOWE

(istota algorytmu Simpleks)

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

Zadanie pierwotne - ZP PL

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

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

Zadanie dualne - ZD PL

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

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

❑ Programowanie liniowe – wzajemne zależności pomiędzy rozwiązaniami zadania pierwotnego oraz dualnego – twierdzenia o dualności

Dla zadań: ZP PL w postaci standardowej na maksimum i dualnego ZD PL

Twierdzenie 1 (o istnieniu obu rozwiązań):
Jeżeli ZP PL i ZD PL mają rozwiązania dopuszczalne (spełniające warunki ograniczające), to oba mają również rozwiązania optymalne. Gdy choć jedno z nich nie ma rozwiązania dopuszczalnego, to oba nie posiadają rozwiązań optymalnych.

Twierdzenie 2 (o wzajemnej relacji rozwiązań dopuszczalnych):
Jeżeli $x_1^0,\ldots,x_n^0$ **- jest rozwiązaniem dopuszczalnym ZP PL, zaś** $y_1^0,\ldots,y_m^0$ **- jest rozwiązaniem dopuszczalnym ZD PL, to pomiędzy nimi zachodzi zależność:** $\sum_{j=1}^{n} c_j x_j^0 \le \sum_{i=1}^{m} b_i y_i^0$

Twierdzenie 3 (o równości optymalnych rozwiązań ZP PL oraz ZD PL - John von Neumann):
Jeżeli istnieją takie dwa rozwiązania dopuszczalne: $x_1^*,\ldots,x_n^*$ **dla ZP PL oraz** $y_1^*,\ldots,y_m^*$ **dla ZD PL, dla których zachodzi zależność:**

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

to obydwa te rozwiązania są optymalne: $\max(\min) f(x^*)=\min(\max) g(y^*)$

❑ Programowanie liniowe – wzajemne zależności pomiędzy rozwiązaniami zadania pierwotnego oraz dualnego – twierdzenia o dualności

Twierdzenie 4 (o równowadze obu rozwiązań):
Jeżeli $x_1^0,\ldots,x_n^0$ - jest rozwiązaniem dopuszczalnym ZP PL, zaś $y_1^0,\ldots,y_m^0$ - jest rozwiązaniem dopuszczalnym ZD PL, to aby te rozwiązania były rozwiązaniami optymalnymi wystarcza, aby spełnione były warunki:

$$(1)\ \sum_{j=1}^{n} a_{i,j}x_j^0 < b_i \Rightarrow y_i^0 = 0;\quad (2)\ \sum_{i=1}^{m} a_{i,j}y_i^0 > c_j \Rightarrow x_j^0 = 0;$$

$$(3)\ y_i^0 > 0 \Rightarrow \sum_{j=1}^{n} a_{i,j}x_j^0 = b_i;\quad (4)\ x_j^0 > 0 \Rightarrow \sum_{i=1}^{m} a_{i,j}y_i^0 = c_j;$$

❑ Programowanie liniowe – metoda Simpleks - wprowadzenie

Metoda Simpleks – jest podstawową i uniwersalną metodą rozwiązywania zadań programowania liniowego, którą szczególnie łatwo daje się opracować algorytmicznie, a tym samym wykorzystać w procesie efektywnego poszukiwania rozwiązań ZPL nowoczesnych komputerowych narzędzi obliczeniowych.

Twórcą metody Simpleks jest B. Dantzing (1947 r.), a jej wprowadzenie zainicjowało burzliwe wykorzystanie metod matematycznych w praktyce rozwiązywania wielu sformułowanych, a dotąd nierozwiązanych problemów decyzyjnych.

Polega ona na sekwencyjnym (krokowym) i ściśle ukierunkowanym (efektywnym) przeglądzie tzw. rozwiązań bazowych.

Rozwiązania bazowe związane są z postacią kanoniczną ZPL (warunki ograniczające w postaci równości).

Dlatego punktem wyjścia w algorytmie Simpleks jest postać kanoniczna rozwiązywanego zadania programowania liniowego.

❑ Programowanie liniowe – metoda Simpleks - postać kanoniczna i jej Baza

- **Postać kanoniczna ZPL:**

$$f(x_1,...,x_n)=(c,x)=c_1x_1+...+c_nx_n \rightarrow \max(\min)$$

$$\begin{cases} a_{1,1}x_1+...+a_{1,n}x_n=b_1 \\ ..... \\ a_{m,1}x_1+...+a_{m,n}x_n=b_m \\ x_1,...,x_n \geq 0 \end{cases} \Leftrightarrow \begin{cases} A\cdot x^T=b^T \\ x \geq 0 \end{cases}$$

- **Baza postaci kanonicznej ZPL:**

Macierz B – kwadratową m – tego stopnia nazywamy bazą, jeżeli składa się z m – liniowo niezależnych kolumn macierzy A. Jej kolumny nazywamy kolumnami bazowymi, zaś pozostałe kolumny macierzy A kolumnami niebazowymi. Zmienne związane z kolumnami bazowymi nazywamy zmiennymi bazowymi, zaś pozostałe niebazowymi $rz(B)=m;\ \det(B)\neq 0$

gdzie:

A - macierz o wymiarach (m,n) współczynników w warunkach ograniczających;

b – (m – wymiarowy) wektor wyrazów wolnych;

c – (n – wymiarowy) wektor wag funkcji celu;

x – (n – wymiarowy) wektor zmiennych decyzyjnych – rozszerzony o ewentualne zmienne swobodne oraz sztuczne;

$$A=[a_{i,j}]_{i=1,...,m;\ j=1,...,n} \quad b=(b_1,...,b_m)$$
$$c=(c_1,...,c_n) \quad x=(x_1,...,x_n)$$

❑ Programowanie liniowe – metoda Simpleks - rozwiązania bazowe

- **Przykład.** Znaleźć rozwiązania bazowe następującego zadania ZPL

Postać standardowa ZPL:

$$f(x_1, x_2) = 2x_1 + 3x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + 2x_2 \le 14 \\ x_1 + 2x_2 \le 8 \\ x_1, x_2 \ge 0 \end{cases}$$

Postać kanoniczna ZPL:

$$f(x_1, x_2, x_3, x_4) = 2x_1 + 3x_2 + 0x_3 + 0x_4 \rightarrow \max$$

$$\begin{cases} 2x_1 + 2x_2 + x_3 = 14 \\ x_1 + 2x_2 + x_4 = 8 \\ x_1, x_2, x_3, x_4 \ge 0 \end{cases}$$

x_3, x_4 – wprowadzone zmienne swobodne

Przykładowa Baza dla powyższej postaci kanonicznej:

Kolumny niebazowe

$$A = \begin{bmatrix} 2 & 2 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}$$

Kolumny bazowe

- Macierz współczynników

$$B = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}$$

- Baza

x_1, x_3 **- zmienne bazowe**

x_2, x_4 **- zmienne niebazowe**

$$x = (x_B, x_N)$$

bazowe **niebazowe**

$$A \cdot x^T = b^T \Leftrightarrow B \cdot x_B^T + N \cdot x_N^T = b^T$$

N – kolumny niebazowe

❑ Programowanie liniowe – metoda Simpleks - rozwiązania bazowe

Z każdą bazą B układu równań w postaci kanonicznej ($Ax^T=b^T$) związane jest jego rozwiązanie bazowe. Jeżeli układ ten jest niesprzeczny (posiada rozwiązania) oraz (n>m), to posiada skończoną liczbę rozwiązań bazowych.

A mianowicie co najwyżej:

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

Rozwiązania bazowe tego układu można uzyskać następująco:

- Wyznaczyć kolejne bazy B tego układu;
- Zmiennym niebazowym - x_N przypisać wartość zero: $x_N=0$;
- Wartości zmiennych bazowych x_B – wyznaczamy rozwiązując układ m – równań z m – niewiadomymi: $Bx_B^T=b^T$ – wynika to z zależności:

$$A \cdot x^T = b^T \Leftrightarrow B \cdot x_B^T + N \cdot x_N^T = b^T$$

Jeżeli wartości każdej zmiennej bazowej w rozwiązaniu układu równań $Bx_B^T=b^T$ jest różne od zera, to takie rozwiązanie bazowe nazywamy niezdegenerowanym. Jeżeli wartość choć jednej zmiennej bazowej jest równa zero, to takie rozwiązanie nazywamy zdegenerowanym.

❑ Programowanie liniowe – metoda Simpleks - przegląd zupełny rozwiązań bazowych

Twierdzenie. Jeżeli zadanie ZPL ma rozwiązanie optymalne, to ma także rozwiązanie optymalne bazowe.

Stąd wniosek, że rozwiązania optymalnego wystarczy szukać wśród rozwiązań bazowych. Można je znaleźć dokonując zupełnego przeglądu wszystkich rozwiązań bazowych. Dla naszego przykładu mamy:

$$f(x_1,x_2,x_3,x_4) = 2x_1 + 3x_2 + 0x_3 + 0x_4 \to \max$$

$$\begin{cases} 2x_1 + 2x_2 + x_3 = 14 \\ x_1 + 2x_2 + x_4 = 8 \\ x_1,x_2,x_3,x_4 \geq 0 \end{cases} \qquad A = \begin{bmatrix} 2 & 2 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}$$

A^1 A^2 A^3 A^4

Wszystkich rozwiązań bazowych może być co najwyżej: $\binom{4}{2} = \frac{4!}{2!\cdot 2!} = 3 \cdot 2 = 6$

$\{x_1,x_2\};\{x_1,x_3\};\{x_1,x_4\};\{x_2,x_3\};\{x_2,x_4\};\{x_3,x_4\}$

Każda z kombinacji może być bazą, bo każda z kolumn A jest liniowo niezależna od pozostałych

Będziemy mieć zatem 6 rozwiązań bazowych. Niektóre z nich mogą być jednak niedopuszczalne (a takie nas nie interesują). Rozwiązanie bazowe nazywamy dopuszczalnym, gdy dla danej bazy B, zachodzi: $x_B \geq 0$

❑ Programowanie liniowe – metoda Simpleks - przegląd zupełny rozwiązań bazowych

- **Rozwiązania bazowe dla naszego przykładu:**

Zmienne decyzyjne	Baza (B1) $\{x_1,x_2\}$	Baza (B2) $\{x_1,x_3\}$	Baza (B3) $\{x_1,x_4\}$	Baza (B4) $\{x_2,x_3\}$	Baza (B5) $\{x_2,x_4\}$	Baza (B6) $\{x_3,x_4\}$
x_1	6	8	7	0	0	0
x_2	1	0	0	4	7	0
x_3	0	- 2	0	6	0	14
x_4	0	0	1	0	- 6	8
Funkcja celu $f(x_1,x_2,x_3,x_4)$	15 (Max)	Niedop.	14	12	Niedop.	0

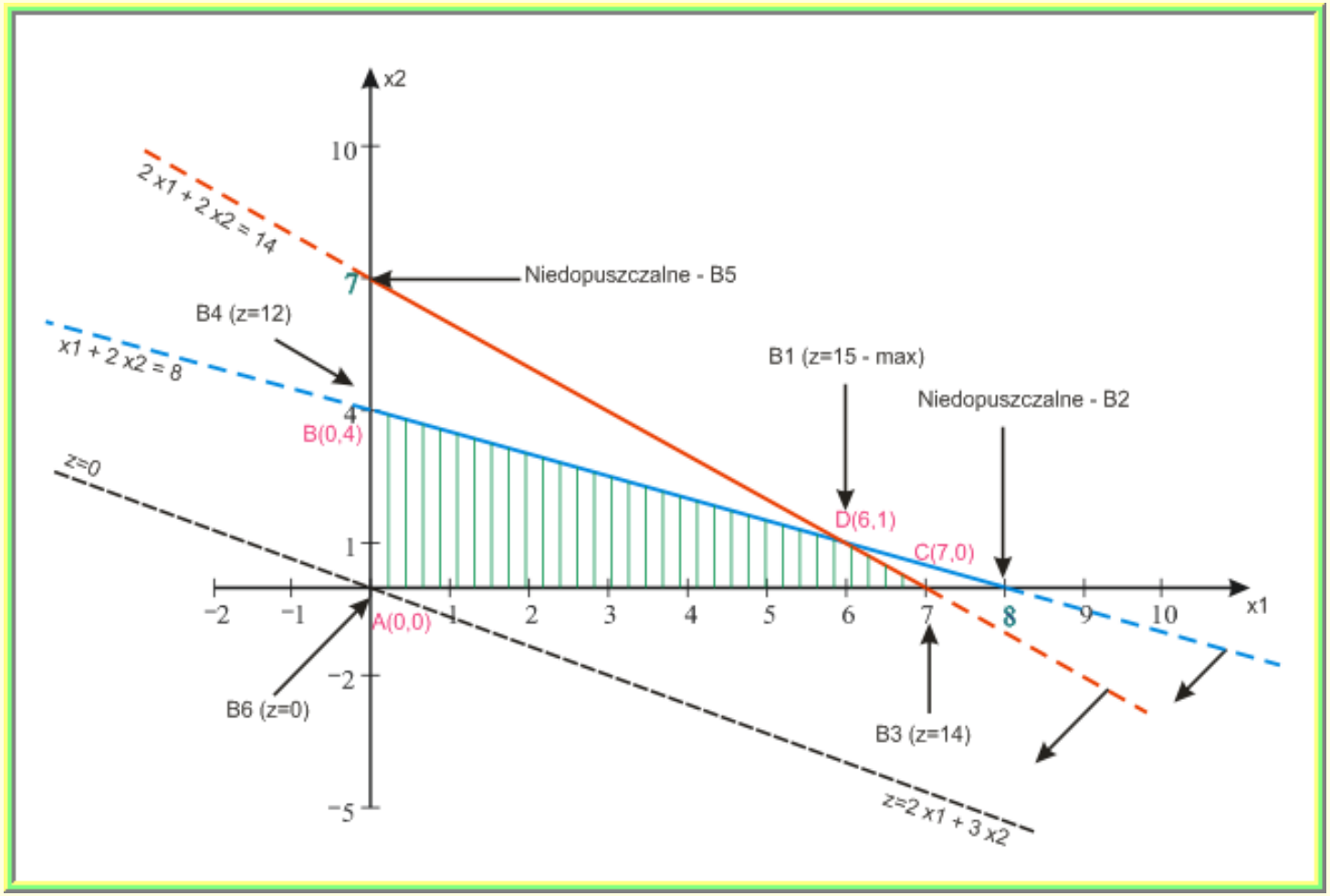
Układ równań w postaci kanonicznej:

$$\begin{cases} 2x_1 + 2x_2 + x_3 = 14 \\ x_1 + 2x_2 + x_4 = 8 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases}$$

Funkcja celu:

$$f(x_1, x_2, x_3, x_4) = 2x_1 + 3x_2 + 0x_3 + 0x_4$$

❑ Programowanie liniowe – metoda Simpleks - przegląd zupełny rozwiązań bazowych

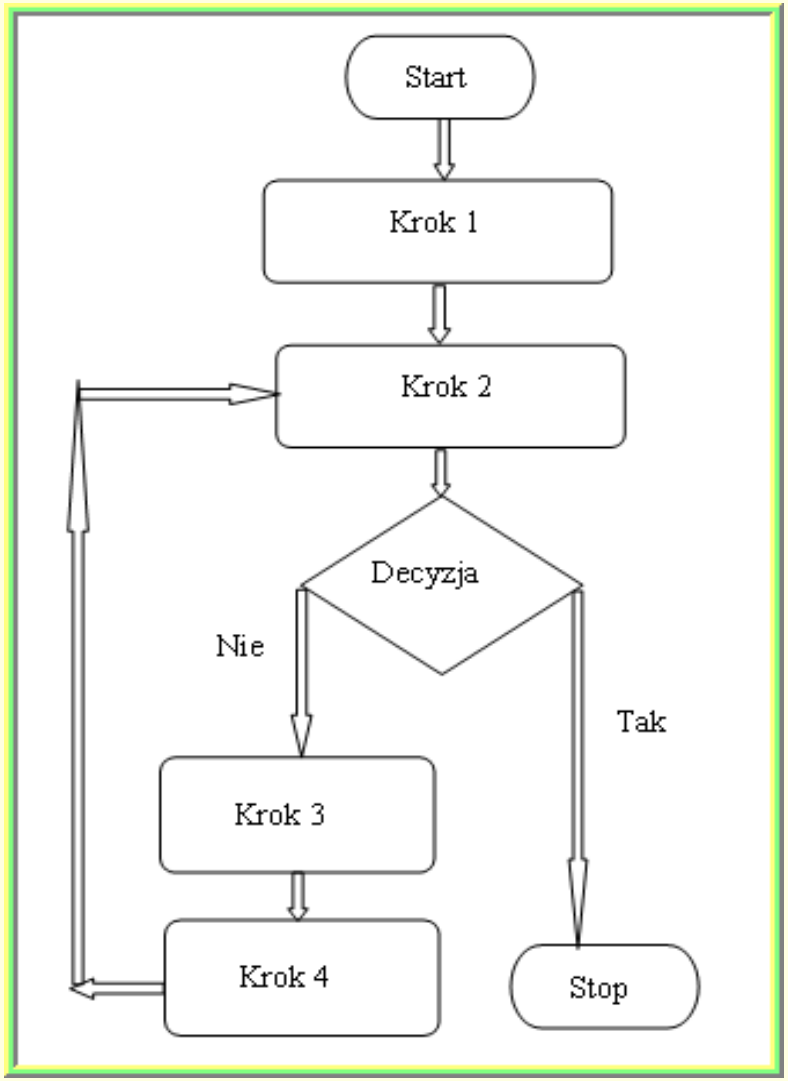


❑ Programowanie liniowe – metoda Simpleks – ujęcie algorytmiczne

W ogólnym przypadku metoda przeglądu zupełnego rozwiązań bazowych jest nieefektywna (duża złożoność obliczeniowa). Dla n=10 i m=5 musimy dokonać przeglądu już 252 ewentualnych rozwiązań bazowych oraz rozwiązywać układy 5 równań z 5 niewiadomymi. Dlatego w praktyce stosuje się przegląd ukierunkowany (tak jak w metodzie - Simpleks)

W metodzie Simpleks wykorzystuje się metodę ukierunkowanego przeglądu rozwiązań bazowych, od pierwszego znanego rozwiązania bazowego dopuszczalnego, do następnego, o którym wiadomo, że nie jest gorsze od poprzedniego. Pomijamy rozwiązania niedopuszczalne oraz te, które są gorsze od aktualnie rozważanego.

❑ Programowanie liniowe – metoda Simpleks – ujęcie algorytmiczne



ALGORYTM SIMPLEX

Krok 1:
Przedstawienie zadania - ZPL w postaci **kanonicznej - bazowej**. Zapisanie zadania w **tablicy simpleksowej**. Znalezienie **pierwszego** rozwiązania **bazowego dopuszczalnego**.

Krok 2:
W oparciu o **simpleksowe kryterium optymalności** badamy, czy aktualne rozwiązanie bazowe dopuszczalne jest optymalne.

Decyzja 1:
Jeżeli – **Tak**, to koniec algorytmu;
Jeżeli **Nie**, to następny **krok 3**;

Krok 3:
Znaleźć numer wektora wprowadzanego do bazy - zastosowanie **kryterium wejścia** oraz numer wektora wyprowadzanego z aktualnej bazy - zastosowanie **kryterium wyjścia**. Określić tzw. **element centralny** tablicy simpleksowej.

Krok 4:
Wymienić wektory w bazie i utworzyć nową postać kanoniczną - bazową (z nową bazą). Wyznaczyć **nową postać** tablicy simpleksowej. **Powrót do kroku 2**.

Zagadnienia

Transportowe

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

ZAGADNIENIA TRANSPORTOWE

Zagadnienie transportowe (ZT) - jest szczególnym przypadkiem liniowego problemu decyzyjnego. Zadanie transportowe posiada bardzo wiele praktycznych zastosowań, cechuje się również tym, że posiada prawie kompletną teorię obejmująca: własności tego typu zadań oraz metody ich rozwiązywania. W teorii tej wykorzystuje się nie tylko elementy ***teorii programowania liniowego***, ale także elementy ***teorii grafów***, w szczególności zagadnienia związane z ***sieciami transportowymi***.

Zadanie transportowe sformułowane zostało po raz pierwszy przez Kantorowicza (1934) i było jednym z pierwszych rozwiązanych problemów programowania liniowego. Zostało ono później zmodyfikowane i opublikowane przez Hitchcoca (1941), który podał także pewną wersję algorytmu jego rozwiązania. Zadanie sformułowane przez Hitchcoca nosi nazwę klasycznego zadania transportowego.

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

1. Matematyczny model zagadnienia transportowego

Od „n" – dostawców: $A_1,...,A_n$ należy dokonać przewozu ładunków do „m" – odbiorców: $B_1,...,B_m$. Wiadomo, że każdy z dostawców dysponuje odpowiednio: $a_1,...,a_n$ jednostkami towaru (***podaż***), natomiast każdy z odbiorców wymaga dostaw w wysokości odpowiednio: $b_1,...,b_m$ jednostek towaru (***popyt***).

Zakłada się, że każdy dostawca może zaopatrywać dowolnego odbiorcę oraz każdy odbiorca może otrzymać towar od dowolnego nadawcy. Suma dostaw od każdego nadawcy do „j-tego" odbiorcy równa się jego zapotrzebowaniu (popytowi), zaś suma dostaw wysłanych od „i-tego" dostawcy do wszystkich odbiorców nie przekracza wielkości jaką dysponuje (podaż).

Zakłada się również, że znane są jednostkowe koszty transportu o każdego dostawcy do odbiorcy: $C = \left\lfloor c_{i,j} \right\rfloor_{i=1,...,n;\, j=1...,m}$ (macierz kosztów transportu). Sumaryczny koszt transportu jest sumą kosztów transportu na poszczególnych trasach i na każdej z tras jest on proporcjonalny do wielkości dostaw.

Jeżeli wielkości $x_{i,j} \geq 0$ oznaczają wielkość dostaw (od „i" – tego dostawcy do „j" – tego odbiorcy), to matematyczny model zagadnienia transportowego można przedstawić w postaci:

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

Należy określić (zaprogramować) macierz $X = \lfloor x_{i,j} \rfloor_{i=1,\ldots,n;\, j=1\ldots,m}$ - wielkości przewozów, aby:

(1)

$$f(x_{i,j}) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{i,j}x_{i,j} \rightarrow \min \quad \text{(funkcja celu)}$$

przy warunkach ograniczających:

$$\begin{cases} \sum_{i=1}^{n} x_{i,j} = b_j ; j = 1,\ldots,m \\ \sum_{j=1}^{m} x_{i,j} \le a_i ; i = 1,\ldots,n \\ x_{i,j} \ge 0 \end{cases}$$

Zadanie sformułowane za pomocą (1) nazywane jest ***klasycznym zadaniem transportowym***. Można zauważyć, że będzie ono posiadać rozwiązanie, gdy między sumaryczną podażą a popytem spełniony jest warunek:

(2)

$$\sum_{i=1}^{n} a_i \ge \sum_{j=1}^{m} b_j$$

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

- Jeżeli w warunku (2) jest równość (równowaga między sumaryczną podażą a popytem), to zadanie (1) nazywamy ***zamkniętym zadaniem transportowym*** (ZZT) i posiada postać:

$$f(x_{i,j}) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{i,j}x_{i,j} \rightarrow \min$$

$$\begin{cases} \sum_{i=1}^{n} x_{i,j} = b_j; j = 1,\ldots,m \\ \sum_{j=1}^{m} x_{i,j} = a_i; i = 1,\ldots,n \\ x_{i,j} \geq 0 \end{cases}$$

- Jeżeli natomiast w warunku (2) jest ostra nierówność, to zadanie (1) nazywamy ***otwartym zadaniem transportowym*** (OZT) i posiada postać:

$$f(x_{i,j}) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{i,j}x_{i,j} \rightarrow \min$$

$$\begin{cases} \sum_{i=1}^{n} x_{i,j} = b_j; j = 1,\ldots,m \\ \sum_{j=1}^{m} x_{i,j} \leq a_i; i = 1,\ldots,n \\ x_{i,j} \geq 0 \end{cases}$$

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

2. Praktyczny przykład problemu decyzyjnego – sformułowanego za pomocą zamkniętego zadania transportowego ZZT.

Cztery piekarnie zlokalizowane na terenie miasta są zaopatrywane w mąkę z dwóch magazynów znajdujących się na peryferiach miasta. Zapasy mąki w magazynach wynoszą odpowiednio: I magazyn – 130 t, II – magazyn – 200 t. Natomiast zapotrzebowanie piekarń jest równe odpowiednio: I piekarnia – 80 t, II – piekarnia – 120 t, III – piekarnia – 70 t, III – piekarnia – 60 t. Koszty dostawy mąki do piekarń zależą tylko od odległości dostaw i są podane w tabeli kosztów:

Tabela kosztów (macierz kosztów)

Magazyny (dostawcy)	Piekarnie (odbiorcy)			
	1	2	3	4
1	25	24	28	13
2	17	30	15	26

Należy wyznaczyć, taki plan dostaw mąki z magazynów do piekarń, aby dostarczyć piekarniom wymagane ilości, przy minimalnych sumarycznych kosztach jej transportu.

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

Problem decyzyjny w postaci zamkniętego zadania transportowego jest postaci:

$$f(x_{i,j}) = 25x_{1,1} + 24x_{1,2} + 28x_{1,3} + 13x_{1,4} + 17x_{2,1} + 30x_{2,2} + 15x_{2,3} + 26x_{2,4} \rightarrow \min$$

$$\begin{cases} \sum_{i=1}^{2} x_{i,1} = 80, & \sum_{i=1}^{2} x_{i,2} = 120, & \sum_{i=1}^{2} x_{i,3} = 70, & \sum_{i=1}^{2} x_{i,4} = 60, \\ \sum_{j=1}^{4} x_{1,j} = 130, & \sum_{j=1}^{4} x_{2,j} = 200, & x_{i,j} \geq 0, _{i=1,2;j=1,2,3,4} \end{cases}$$

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

Zadanie to jako liniowy problem optymalizacyjny można zapisać w postaci zadania – ZPL:

$$f(x) = (c, x) = \sum_{k=1}^{2\cdot 4=8} c_k x_k \rightarrow \min$$

gdzie:

$c = [25 \quad 24 \quad 28 \quad 13 \quad 17 \quad 30 \quad 15 \quad 26]$

$x = [x_{1,1} \quad x_{1,2} \quad x_{1,3} \quad x_{1,4} \quad x_{2,1} \quad x_{2,2} \quad x_{2,3} \quad x_{2,4}]$

Przy warunkach ograniczających (w postaci macierzowej):

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

czyli:

$$A \cdot x^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{1,4} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{2,4} \end{bmatrix} = b^T = \begin{bmatrix} a_1 = 130 \\ a_2 = 200 \\ b_1 = 80 \\ b_2 = 120 \\ b_3 = 70 \\ b_4 = 60 \end{bmatrix}$$

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

Niektóre warianty zagadnień transportowych sprowadzalne do zamkniętego zagadnienia transportowego.

- Otwarte zagadnienie transportowe – OZT:
  Algorytm transportowy zakłada, że zadanie jest zbilansowane (zamknięte). Każde otwarte zadanie transportowe (OZT) można sprowadzić do (ZZT) wprowadzając dodatkowego: m+1 – fikcyjnego odbiorcę (w praktyce jest to najczęściej magazyn znajdujący się u każdego z dostawców, w którym będą magazynowane nadwyżki podaży: $b_{m+1} = \sum_{i=1}^{n} a_i - \sum_{j=1}^{m} b_j$. W zadaniu tym koszty magazynowania można pominąć.

Macierz kosztów dla tego zadania:

i \ j	1	2	…	m	m+1 (magazyn)	a_i
1	$c_{1,1}$	$c_{1,2}$	…	$c_{1,m}$	0	a_1
2	$c_{2,1}$	$c_{2,2}$	…	$c_{2,m}$	0	a_2
…	…	…	…	…	…	…
n	$c_{n,1}$	$c_{n,2}$	…	$c_{n,m}$	0	a_n
b_j	b_1	b_2	…	b_m	b_{m+1}	$\sum_{i=1}^{n} a_i = \sum_{j=1}^{m+1} b_j$

Model matematyczny zadania:

$$f(x_{i,j}) = \sum_{i=1}^{n}\sum_{j=1}^{m+1} c_{i,j}x_{i,j} \rightarrow \min$$

$$\begin{cases} \sum_{i=1}^{n} x_{i,j} = b_j; j = 1,\ldots,m+1 \\ \sum_{j=1}^{m+1} x_{i,j} = a_i; i = 1,\ldots,n \\ x_{i,j} \geq 0 \end{cases}$$

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

- Zadanie transportowo – magazynowe (ZT – M):
  Jeżeli w otwartym zadaniu transportowym – OZT uwzględnimy także koszty magazynowania, to zadanie to staje się zadaniem transportowo-magazynowym.
  Oznaczmy: h_i - jednostkowe koszty magazynowania w magazynach nadawców; $x_{i,m+1}; i = 1,...,n$ - nadwyżki podaży magazynowane u nadawców.

  Macierz łącznych kosztów dla tego zadania:

i \ j	1	2	…	m	m+1 (magazyn)	a_i
1	$c_{1,1}$	$c_{1,2}$	…	$c_{1,m}$	h_1	a_1
2	$c_{2,1}$	$c_{2,2}$	…	$c_{2,m}$	h_2	a_2
…	…	…	…	…	…	…
n	$c_{n,1}$	$c_{n,2}$	…	$c_{n,m}$	h_n	a_n
b_j	b_1	b_2	…	b_m	b_{m+1}	$\sum_{i=1}^{n} a_i = \sum_{j=1}^{m+1} b_j$

Model matematyczny zadania:

$$f(x_{i,j}) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{i,j}x_{i,j} + \sum_{i=1}^{n} h_i x_{i,m+1} \rightarrow \min$$

$$\begin{cases} \sum_{i=1}^{n} x_{i,j} = b_j; j = 1,...,m+1 \\ \sum_{j=1}^{m+1} x_{i,j} = a_i; i = 1,...,n \\ x_{i,j} \geq 0 \end{cases}$$

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

- Zadanie transportowo – produkcyjne (ZT - P)
  W zagadnieniu transportowo-magazynowym zakładaliśmy, że towary są już wyprodukowane (znajdują się w magazynach). Jeżeli towary przed transportem należy wyprodukować to musimy uwzględnić w takim zagadnieniu także koszty produkcji. Tego typu zadania nazywają się transportowo-produkcyjnymi.

  Zakładamy, że $\sum_{i=1}^{n} a_i > \sum_{j=1}^{m} b_j$ (zdolności produkcyjne nadawców przewyższają zapotrzebowania odbiorców), a więc zadanie musimy zbilansować wprowadzając fikcyjnego odbiorcę – magazyn, w którym będą magazynowane nadwyżki mocy produkcyjnych w ilościach: $b_{m+1} = \sum_{i=1}^{n} a_i - \sum_{j=1}^{m} b_j$.

❑ Zagadnienia i Problemy Transportowe – Algorytm Transportowy

Oznaczmy: k_i - jednostkowe koszty produkcji u „i – tego" producenta (dostawcy); h_i - jednostkowe koszty magazynowania w magazynach nadawców nadwyżki produkcji; $x_{i,j}(i=1,...,n; j=1,...,m)$ - wielkości towarów wyprodukowane i dostarczone od i – tego nadawcy do j - tego odbiorcy; $x_{i,m+1}; i=1,...,n$ - nadwyżki produkcji magazynowane u nadawców.

Uwaga: gdy założymy, że zdolności produkcyjne nie będą w pełni wykorzystane, to w zadaniu przyjmujemy: $h_i + k_i = 0$.

Macierz łącznych kosztów dla tego zadania:

j / i	1	2	…	m	m+1 (magazyn)	a_i
1	$c_{1,1}+k_1$	$c_{1,2}+k_1$	…	$c_{1,m}+k_1$	h_1+k_1	a_1
2	$c_{2,1}+k_2$	$c_{2,2}+k_2$	…	$c_{2,m}+k_2$	h_2+k_2	a_2
…	…	…	…	…	…	…
n	$c_{n,1}+k_n$	$c_{n,2}+k_n$	…	$c_{n,m}+k_n$	h_n+k_n	a_n
b_j	b_1	b_2	…	b_m	b_{m+1}	$\sum_{i=1}^{n} a_i = \sum_{j=1}^{m+1} b_j$

Model matematyczny zadania:

$$f(x_{i,j}) = \sum_{i=1}^{n}\sum_{j=1}^{m}(c_{i,j}+k_i)x_{i,j} + \sum_{i=1}^{n}(h_i+k_i)x_{i,m+1} \to \min$$

$$\begin{cases} \sum_{i=1}^{n} x_{i,j} = b_j; j = 1,...,m+1 \\ \sum_{j=1}^{m+1} x_{i,j} = a_i; i = 1,...,n \\ x_{i,j} \geq 0 \end{cases}$$