LINIOWE MODELE OPTYMALIZACJI DYSKRETNEJ

❑ Liniowe Modele Optymalizacji Dyskretnej – wprowadzenie w tematykę zagadnień

Istnieje bardzo wiele sytuacji decyzyjnych, których nie możemy opisać używając tylko wyłącznie zmiennych ciągłych.

Wynika to z nieciągłości pewnych rozważanych procesów ekonomicznych:

- pracownika można przydzielić tylko do jednego z kilku dostępnych stanowisk pracy;
- projekt inwestycyjny będzie przyjmowany do realizacji lub nie;
- zakład produkcyjny będzie lokalizowany w jednym z możliwych punktów lokalizacji lub też nie;

We wszystkich przytoczonych sytuacjach decyzyjnych wymagamy, aby wszystkie (lub choć jedna zmienna decyzyjna), spośród tych które mamy wyznaczyć przyjmowały wartości tzw. ***dyskretne*** (np. ze zbioru liczb całkowitych: $x \in Z$, lub ze zbioru liczb binarnych: $x \in \{0,1\}$).

Zagadnienia decyzyjne, w których przynajmniej jedna zmienna decyzyjna przyjmuje wartości dyskretne nazywamy – ***dyskretnym zagadnieniem decyzyjnym***, a ich matematyczne modele – ***dyskretnym zadaniem decyzyjnym*** (DZD).

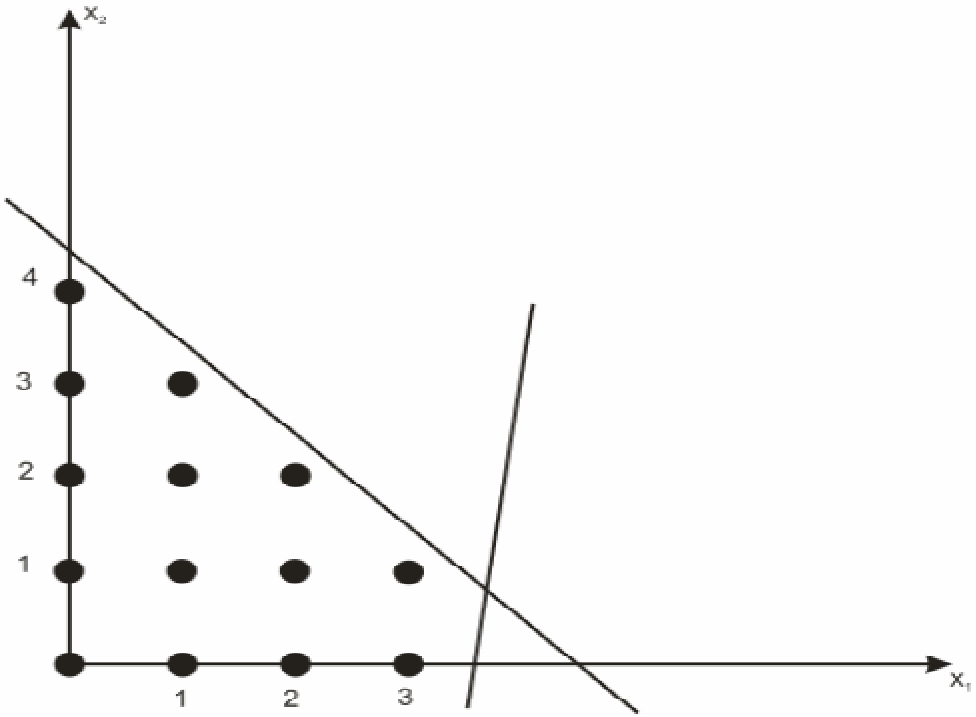
❑ Liniowe Modele Optymalizacji Dyskretnej – **wprowadzenie w tematykę zagadnień**

Interesować nas będą obecnie tylko takie dyskretne problemy decyzyjne, w których zarówno funkcja celu jak i warunki ograniczające są postaci liniowej – ***zadania programowania dyskretnego - liniowego*** (PDL). Wśród tego typu zadań wyróżnia się trzy podstawowe grupy:

- zadania ***programowania całkowitoliczbowego - liniowego*** (PCL)
- zadania ***programowania binarnego – liniowego*** (PBL)
- zadania ***programowania mieszanego – liniowego*** (PML)

❑ Liniowe Modele Optymalizacji Dyskretnej – wprowadzenie w tematykę zagadnień

Zbiór rozwiązań dopuszczalnych „*D*” zadania programowania dyskretnego – liniowego jest zawsze zbiorem niespójnym (np. dla zadania programowania całkowitoliczbowego z dwoma zmiennymi - będzie to zbiór punktów o współrzędnych całkowitych znajdujących się w pewnym wieloboku). Nieciągłość zmiennych decyzyjnych powoduje, że zadania tego typu są trudniejsze do rozwiązania, niż zwykłe zadania programowania liniowego.



❑ Liniowe Modele Optymalizacji Dyskretnej – przykład dyskretnego problemu decyzyjnego

- **Zagadnienie optymalnego przydziału**

Istnieje możliwość obsadzenia „n" – stanowisk roboczych (i = 1,2,...,n) przez „n" – osób (pracowników) (j = 1,2,...,n). Znane są efekty pracy j – tego robotnika na i – tym stanowisku pracy (macierz efektów pracy - $W_{i,j} = [w_{i,j}]_{i,j=1,2,\ldots,n}$).

Efekty te mogą być oceniane pozytywnie (wydajność pracy, wartość produkcji w przeliczeniu na jednostkę czasu) lub negatywnie (liczba braków, czas wykonania pracy, koszty związane z pracą).
Należy dokonać takiego przydziału pracowników do poszczególnych stanowisk pracy, tak aby zminimalizować negatywne lub zmaksymalizować pozytywne efekty pracy dla całego zespołu (zakładu pracy).
Zakłada się ponadto, że każde stanowisko pracy może być obsadzone tylko przez jednego pracownika, a tym samym każdy pracownik może pracować tylko na jednym stanowisku.

Oznaczenia:

Oznaczmy przez $X_{i,j} = [x_{i,j}]_{i,j} = 1,2,\ldots,n$ - macierz zmiennych decyzyjnych, która jest postaci:

$$x_{i,j} = \begin{cases} 1, & \text{gdy j - ty pracownik jest przydzielony do i - tego stanowiska} \\ 0, & \text{w przeciwnym wypadku} \end{cases}$$

❑ Liniowe Modele Optymalizacji Dyskretnej – przykład dyskretnego problemu decyzyjnego

Model matematyczny:

Problem ten można przedstawić za pomocą następującego liniowego zadania programowania binarnego (PBL):

(funkcja celu)

$$f(x_{i,j}) = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} \cdot x_{i,j} \rightarrow \max(\min)$$

(warunki ograniczające)

$$\sum_{j=1}^{n} x_{i,j} = 1, \quad i = 1,2,\ldots,n$$

(każde stanowisko jest obsadzone tylko przez 1 pracownika)

$$\sum_{i=1}^{n} x_{i,j} = 1, \quad \mathrm{j} = 1,2,\ldots,n$$

(każdy pracownik jest przydzielony tylko do 1 stanowiska)

$$x_{i,j} = \begin{cases} 1 \\ 0 \end{cases} \qquad \text{i, j=1,2,...,n.}$$

❑ Liniowe Modele Optymalizacji Dyskretnej – przykład dyskretnego problemu decyzyjnego

Każde rozwiązanie bazowe dopuszczalne (tym samym optymalne) to macierze postaci (dla n=5):

$$X = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

(w każdym wierszu oraz w każdej kolumnie jest tylko jedna jedynka).

Zadanie optymalnego przydziału, mimo że jest klasycznym problemem **programowania dyskretnego**, to może być rozwiązane metodami programowania liniowego – **algorytmem simpleks** (co jest bardzo pracochłonne). Istnieje jednak stosunkowo prosty i skuteczny algorytm postępowania – **algorytm węgierski** (oparty na twierdzeniu węgierskiego matematyka - **Denesa Königa**), który można zastosować do rozwiązywania zadań optymalnego przydziału.

❑ Liniowe Modele Optymalizacji Dyskretnej – przykład dyskretnego problemu decyzyjnego

Przykład: W pewnym magazynie pracuje 3 pracowników magazynowych: P1, P2, P3, którzy mogą wykonywać 4 rodzaje zadań: Z1, Z2, Z3, Z4, z różną wydajnością. W tabeli poniżej podana jest wydajność pracowników przy wykonywaniu poszczególnych zadań:

Pracownicy	Wydajność pracowników (szt./godz.) przy wykonywaniu zadań magazynowych			
	Z1	Z2	Z3	Z4
P1	15	4	5	2
P2	3	6	3	10
P3	12	4	6	3

Zakładając specjalizację w ciągu dnia pracowników przy wykonywaniu tylko jednego zadania, przydzielić zadania poszczególnym pracownikom, tak aby zmaksymalizować łączną wydajność ich pracy.

Ponieważ w problemie optymalnego przydziału zakłada się, że liczba stanowisk pracy jest taka sama jak liczba pracowników, to w naszym przykładzie musimy wprowadzić czwartego fikcyjnego pracownika. Oczywiście wydajność jego pracy dla poszczególnych zadań będzie równa 0.
Macierz wydajności pracy (współczynników funkcji celu) jest więc postaci:

$$W = \begin{bmatrix} 15 & 4 & 5 & 2 \\ 3 & 6 & 3 & 10 \\ 12 & 4 & 6 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

❑ Liniowe Modele Optymalizacji Dyskretnej – przykład dyskretnego problemu decyzyjnego

Matematyczny model zadania:

$$F(x_{i,j}) = 15x_{1,1} + 4x_{1,2} + 5x_{1,3} + 2x_{1,4} + 3x_{2,1} + 6x_{2,2} + 3x_{2,3} + 10x_{2,4} +$$
$$+ 12x_{3,1} + 4x_{3,2} + 6x_{3,3} + 3x_{3,4} \rightarrow \max$$

$$\begin{cases} x_{1,1} + x_{1,2} + x_{1,3} + x_{1,4} & = & 1 \\ x_{2,1} + x_{2,2} + x_{2,3} + x_{2,4} & = & 1 \\ x_{3,1} + x_{3,2} + x_{3,3} + x_{3,4} & = & 1 \\ x_{4,1} + x_{4,2} + x_{4,3} + x_{4,4} & = & 1 \\ x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} & = & 1 \\ x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} & = & 1 \\ x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} & = & 1 \\ x_{1,4} + x_{2,4} + x_{3,4} + x_{4,4} & = & 1 \end{cases} \qquad \begin{array}{l} x_{i,j} = 0 \ \text{ lub } \ x_{i,j} = 1; \\ i, j = 1,\ldots,4; \end{array}$$

Rozwiążemy zadanie korzystając z wersji algorytmu węgierskiego, która zakłada, że funkcja celu jest postaci - minimum. Dlatego w rozwiązaniu będziemy minimalizować funkcję przeciwną do funkcji celu: $-F(x_{i,j})$, dla której macierz współczynników jest postaci:

$$W = \begin{bmatrix} -15 & -4 & -5 & -2 \\ -3 & -6 & -3 & -10 \\ -12 & -4 & -6 & -3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

❑ Liniowe Modele Optymalizacji Dyskretnej – przykład dyskretnego problemu decyzyjnego

Krok 1: Przekształcenie macierzy: **W** – tak, aby w każdym wierszu i każdej kolumnie znalazło się co najmniej jedno zero;

$$W = \begin{bmatrix} \boxed{-15} & -4 & -5 & -2 \\ -3 & -6 & -3 & \boxed{-10} \\ \boxed{-12} & -4 & -6 & -3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

element najmniejszy

$$W = \begin{bmatrix} 0 & 11 & 10 & \boxed{13} \\ 7 & 4 & 7 & 0 \\ 0 & 8 & 6 & 9 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

-2 - (-15) = 13

Krok 2: Skreślenie w przekształconej macierzy współczynników funkcji celu wierszy oraz kolumn zawierających zero możliwie najmniejszą liczbą linii;

$$W = \begin{bmatrix} 0 & 11 & 10 & 13 \\ 7 & 4 & 7 & 0 \\ 0 & 8 & 6 & 9 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

trzy linie – zatem przechodzimy do kroku 4

Jeżeli najmniejsza liczba linii konieczna do pokrycia wszystkich zer jest równa wymiarowi macierzy (**czyli - n**), to rozwiązanie, które otrzymamy na podstawie tak przekształconej macierzy współczynników będzie optymalne – przechodzimy do **kroku 3**. Jeżeli jest ona mniejsza niż wymiar macierzy – **W**, to przechodzimy do **kroku 4**.

❑ Liniowe Modele Optymalizacji Dyskretnej – przykład dyskretnego problemu decyzyjnego

Krok 3: Ustalić tak rozwiązanie optymalne, aby w macierzy $\left[x^*_{i,j}\right]$ jedynki znalazły się tylko na tych miejscach, gdzie są zera w przekształconej macierzy – **W** (musimy dbać także, aby w każdym wierszu i każdej kolumnie była tylko jedna jedynka).

Krok 4: Gdy liczba linii pokrywających zera jest mniejsza od wymiaru macierzy współczynników, to w bieżącej (przekształconej) macierzy współczynników należy znaleźć element najmniejszy oraz:

- odjąć go od elementów nieskreślonych;
- dodać go do elementów podwójnie skreślonych;
- elementy skreślone jedną linią (raz) pozostawiamy bez zmian;

$$W = \begin{bmatrix} 0 & 11 & 10 & 13 \\ 7 & 4 & 7 & 0 \\ 0 & 8 & \boxed{6} & 9 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

element minimalny

$$W = \begin{bmatrix} 0 & 5 & 4 & 7 \\ 13 & 4 & 7 & 0 \\ 0 & 2 & 0 & 3 \\ 6 & 0 & 0 & 0 \end{bmatrix}$$

elementy odjęte

Powrót do **kroku 2** i powtórzenie procedury, aż do uzyskania rozwiązania optymalnego.

elementy dodane

$$W = \begin{bmatrix} \boxed{0} & 5 & 4 & 7 \\ 13 & 4 & 7 & \boxed{0} \\ 0 & 2 & \boxed{0} & 3 \\ 6 & \boxed{0} & 0 & 0 \end{bmatrix}$$

cztery linie
zatem rozwiązanie optymalne
(krok 3)

$$X = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$F(x_{i,j}) = 15 + 10 + 6 =$$
$$= 31 \text{ [szt./godz.]}$$

❑ Liniowe Modele Optymalizacji Dyskretnej – metoda podziału i ograniczeń

Dla dyskretnych zadań decyzyjnych uniwersalną metodą ich rozwiązywania jest metoda ***podziału i ograniczeń***. Nie jest to ściśle ustalony algorytm postępowania jak np. algorytm „simpleks", lecz raczej pewne podejście do rozwiązywania określonej klasy (***dyskretnych***) zadań optymalizacyjnych. Podejście to zostanie zaprezentowane na następującym przykładzie:

Przykład:
Rozwiązać następujące zadanie programowania całkowitoliczbowego – liniowego (PCL):

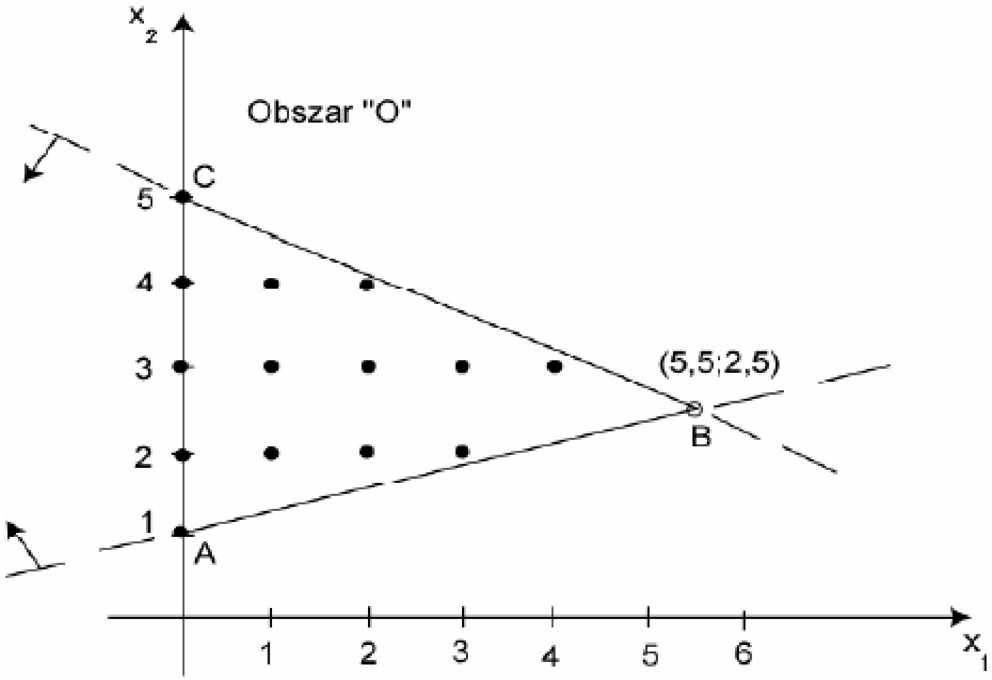
$$f(x_1, x_2) = 4x_1 + 2x_2 \to \max$$

$$\begin{cases} -3x_1 + 11x_2 \geq 11 \\ 5x_1 + 11x_2 \leq 55 \\ x_1, x_2 \geq 0 \\ x_1, x_2 \text{ - całkowite} \end{cases}$$

Jeżeli pominiemy warunek całkowitoliczbowości, to zadanie to jest zadaniem programowania liniowego (PL), które można rozwiązać (geometrycznie, algorytmem „simpleks").
Jeżeli zadanie (PL) nie ma rozwiązania optymalnego, to nie posiada go także zadanie PCL.
Jeżeli otrzymane rozwiązanie optymalne dla (PL) spełnia ponadto warunek całkowitoliczbowości, to jest to oczywiście poszukiwane rozwiązanie zadania PCL, jeśli nie, to stosuje się wtedy metodę ***podziału i ograniczeń***.

❑ Liniowe Modele Optymalizacji Dyskretnej – metoda podziału i ograniczeń

Ponieważ nasze zadanie posiada tylko dwie zmienne decyzyjne, to spróbujemy go rozwiązać metodą geometryczną.



$f(x_1, x_2) = 4x_1 + 2x_2$

warstwica: $x_2 = -2x_1$

$f(A) = 2$, $f(C) = 10$

$\max = f(B) = 27$

Ponieważ nasze rozwiązanie maksymalne nie jest całkowitoliczbowe, to należy zastosować metodę podziału i ograniczeń.

❑ Liniowe Modele Optymalizacji Dyskretnej – metoda podziału i ograniczeń

Zdefiniujmy tzw. funkcję ograniczającą z góry *„ozn. w”* (*kres górny*) określoną na rodzinie 2^O wszystkich podzbiorów obszaru „O”, która spełnia warunki:

(1) $x \in O_1 \subset O \Rightarrow f(x) \leq w(D_1)$ - kres górny zbioru nie może być mniejszy od maksymalnej wartości funkcji celu dla rozwiązań z tego zbioru.

(2) $O_2 \subset O_1 \subset O \Rightarrow w(O_2) \leq w(O_1)$ - kres górny zbioru zawierającego jakiś swój podzbiór nie może być mniejszy od kresu górnego tego jego podzbioru.

(3) $O_1 = \{x\} \subset D \Rightarrow w(O_1) = f(x)$ - jeżeli zbiór jest jednoelementowy, to jego kres górny jest równy wartości funkcji celu dla tego elementu.

Uwaga: W definicji kresu dolnego nierówności w warunkach (1) i (2) zmieniają się na przeciwne.

Przykład:

Kresem górnym może być funkcja:

$$w(O) = 15,\ \ O_1 \subset O = \{x^1, x^2\}, f(x^1) = 8, f(x^2) = 9,\ \ w(O_1) = 10,$$

$$O_2 \subset O = \{x^3\}, f(x^3) = 14, w(O_2) = 14.$$

Nie może być to natomiast funkcja zdefiniowana następująco:

$$w(O) = 15,\ \ O_1 \subset O = \{x^1, x^2\}, f(x^1) = 8, f(x^2) = 9,\ \ w(O_1) = 20,$$

$$O_2 \subset O = \{x^3\}, f(x^3) = 14, w(O_2) = 16$$

(nie są spełnione warunki (2) i (3))

❑ Liniowe Modele Optymalizacji Dyskretnej – metoda podziału i ograniczeń

Dokonujemy podziału zbioru (obszaru) „O" na coraz mniejsze podzbiory. W wyniku „r" podziałów zbioru „O" uzyskujemy „2r" jego podzbiorów. Podziałów kolejnych podzbiorów dokonujemy dla tzw. ***podzbiorów perspektywicznych***.

Podzbiorem perspektywicznym O_p w „r - tym" kroku obliczeń (dla zadań na maksimum) jest taki zbiór, dla którego $w(O_p) = \max\{w(D_l) : D_l \in G^r\}$, gdzie G^r - rodzina podzbiorów ***aktywnych*** (które nie zostały jeszcze podzielone). Dla zadań na minimum w warunku tym pojawia się (minimum oraz kres dolny).

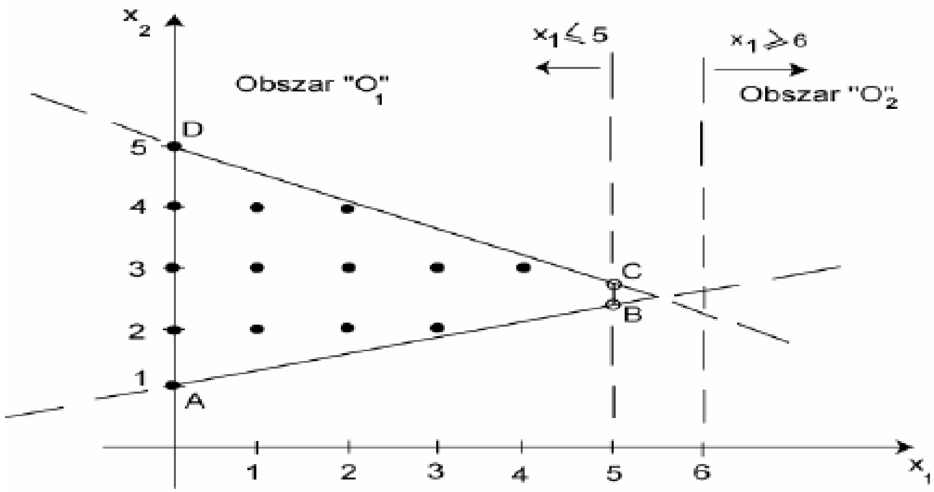
Podstawę podziału zbiorów aktywnych – perspektywicznych stanowi pierwsza zmienna w rozwiązaniu optymalnym (dla zbioru perspektywicznego) - x^p, która nie spełnia warunku całkowitoliczbowości. Jeżeli będzie to zmienna x_k^p - wtedy dzielimy go na dwa podzbiory:

(4)

$$O_{2r+1} = \{x : x \in O_p \wedge x_k \leq N(x_k^p)\}, O_{2r+2} = \{x : x \in O_p \wedge x_k \geq N(x_k^p) + 1\}$$

❑ Liniowe Modele Optymalizacji Dyskretnej – metoda podziału i ograniczeń

Dla naszego przykładu początkowym zbiorem aktywnym perspektywicznym jest zbiór „O". Kres górny tego zbioru wynosi: $w(O) = f(B) = 27$. Obie zmienne decyzyjne w stowarzyszonym rozwiązaniu optymalnym (PL) nie są całkowite, więc dokonujemy podziału „O" na dwa podzbiory: O_1, O_2 względem zmiennej x_1 zgodnie ze wzorem (4).

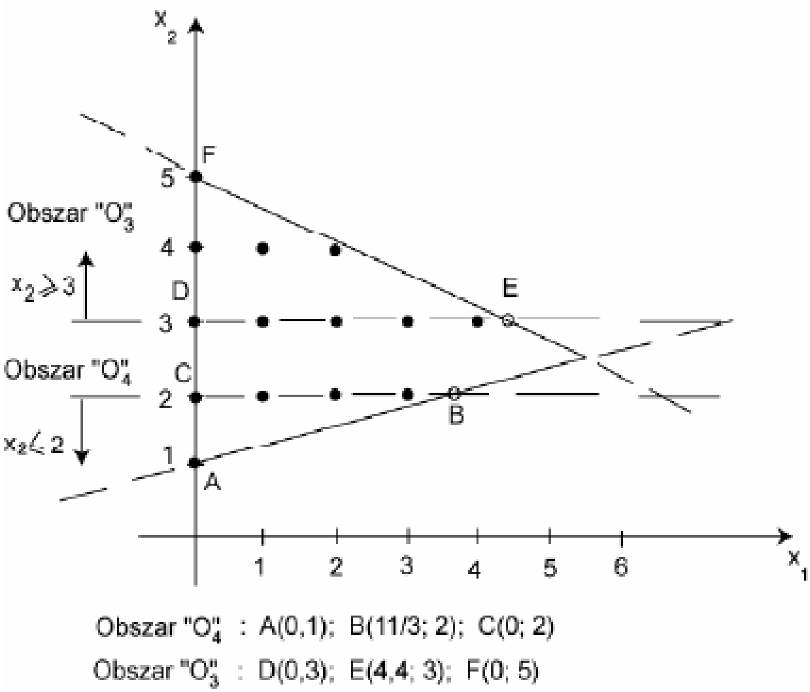


Obszar "O_1" : A(0,1); B(5; 26/11); C(5; 30/11); D(0,5)

Obszar O_2 - jest zbiorem pustym dlatego ***zamykamy go*** przypisując mu symboliczny kres górny: $w(O_2) = -\infty$. Obszar O_1 jest zatem obszarem perspektywicznym – aktywnym. Optymalna wartość funkcji celu dla stowarzyszonego zadania PL jest spełniona dla punktu „C" tego obszaru i wynosi: 280/11=25,45. Jako kres górny możemy przyjąć: $w(O_1) = 26$.

❑ Liniowe Modele Optymalizacji Dyskretnej – metoda podziału i ograniczeń

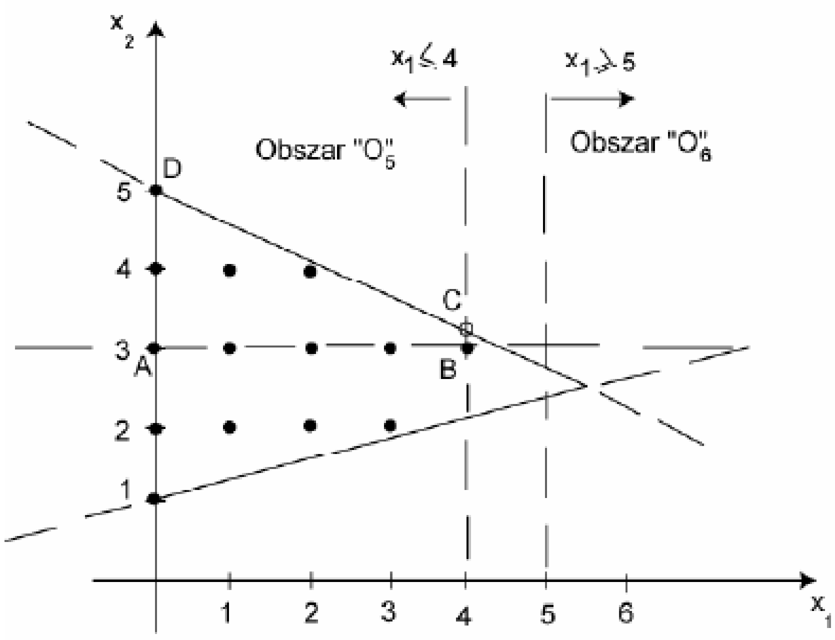
Ponieważ druga współrzędna optymalnego rozwiązania (PL) nie jest całkowita, więc zgodnie z (4) dzielimy O_1 na dwa podzbiory aktywne: O_3, O_4 względem zmiennej x_2.



Opt. wart. funkcji celu dla zadania PL w obszarze O_3 jest spełniona dla punktu „E" tego obszaru oraz wynosi: 23,6.
Jako kres górny możemy przyjąć: $w(O_3) = 24$.
Opt. wart. funkcji celu dla zadania PL w obszarze O_4 jest spełniona dla punktu „B" tego obszaru oraz wynosi: 56/3=18,67.
Za kres górny możemy przyjąć: $w(O_4) = 19$.

❑ Liniowe Modele Optymalizacji Dyskretnej – metoda podziału i ograniczeń

Podzbiorem perspektywicznym jest zatem zbiór O_3. Ponieważ pierwsza współrzędna optymalnego rozwiązania (PL) nie jest całkowita, więc zgodnie z (4) dzielimy O_3 na dwa podzbiory aktywne: O_5, O_6 względem zmiennej x_1.



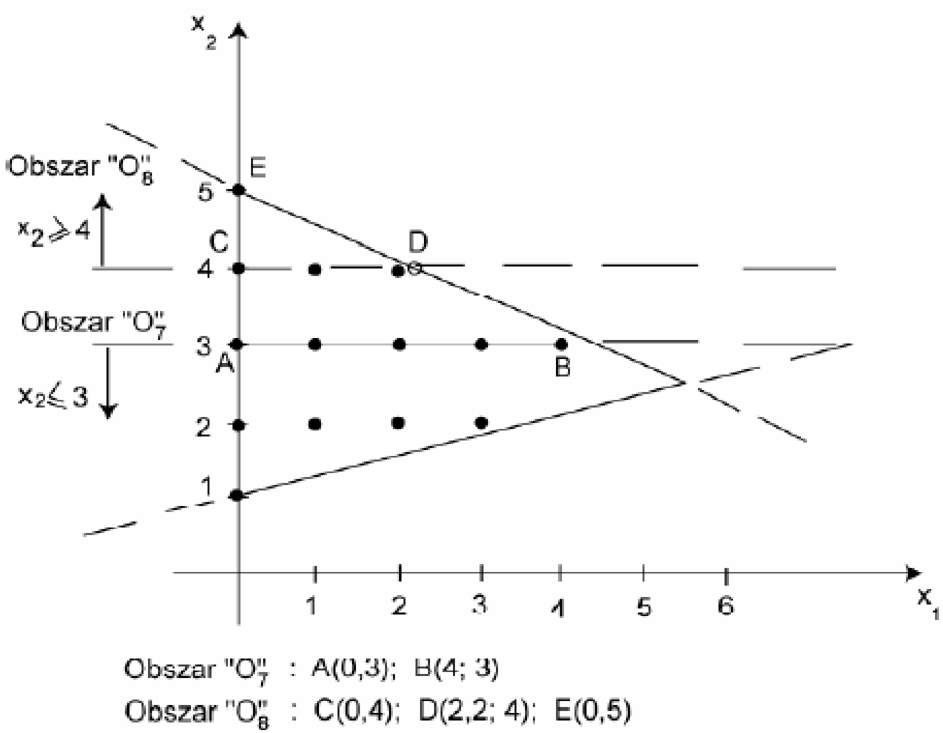
Obszar "O5" : A(0,3); B(4; 3); C(4; 35/11); D(0,5)

Opt. wart. funkcji celu dla zadania PL w obszarze O_5 jest spełniona dla punktu „C" tego obszaru oraz wynosi: 246/11=22,36. Jako kres górny możemy przyjąć: $w(O_5) = 23$.

Obszar O_6 jest zbiorem pustym zatem zamykamy go przyjmując kres górny $w(O_6) = -\infty$.

❑ Liniowe Modele Optymalizacji Dyskretnej – metoda podziału i ograniczeń

Zbiór O_5 jest perspektywiczny, więc znowu go dzielimy na dwa podzbiory O_7, O_8 względem zmiennej posiadającej niecałkowite rozwiązanie optymalne dla zadania stowarzyszonego (PL) w tym obszarze. Będzie to zmienna - x_2.



Opt. wart. funkcji celu dla zadania PL w obszarze O_7 jest spełniona dla punktu „B" tego obszaru oraz wynosi: 22.

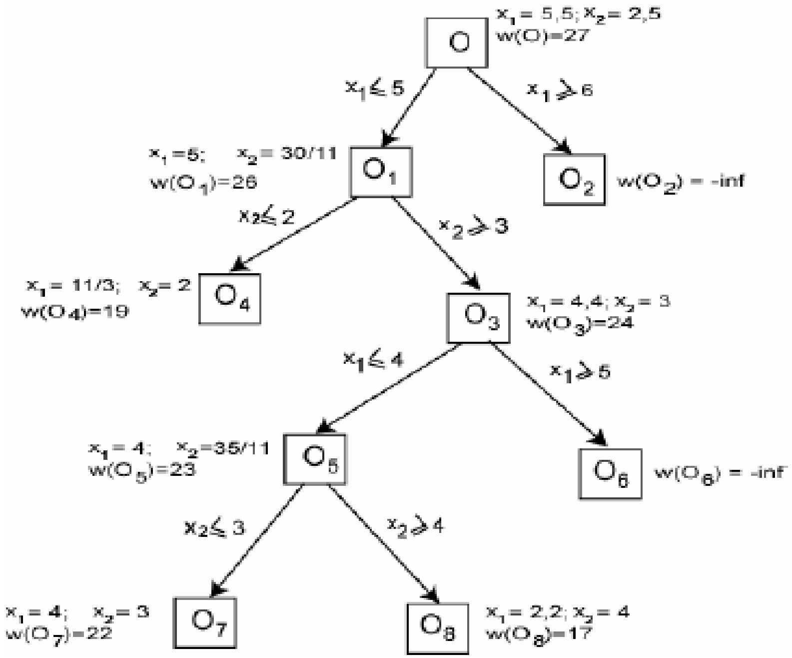
Ponieważ optymalne rozwiązanie PL dla obszaru O_7 jest całkowitoliczbowe, to:
jest to również rozwiązanie optymalne wyjściowego zadania ***PCL***.

❑ Liniowe Modele Optymalizacji Dyskretnej – metoda podziału i ograniczeń

Podział zbioru „O" na coraz mniejsze podzbiory można przedstawić w postaci tzw. ***„drzewa podziału"***. Z każdym jego wierzchołkiem (o numerze „k - tym") jest związany:

- podzbiór O_k
- zadanie programowania liniowego PL^k (dla tego podzbioru)
- jego rozwiązanie optymalne: x_k^*
- kres górny (dolny) tego zbioru $w(O_k)$

Dla naszego zadania drzewo jest postaci:



Przykłady – problemów optymalizacji dyskretnej

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Zagadnienie rozwózki:

Często mamy do czynienia z sytuacją, gdy pewien jednorodny produkt musi zostać przewieziony od producenta do wielu jego odbiorców. Dla przykładu z cukrowni rozwozi sie wyprodukowany cukier, z mleczarni np. masło i mleko, z browaru piwo itd.

Niekiedy mamy sytuację odwrotną, zwłaszcza w przemyśle spożywczym zakupiony surowiec od wielu jego producentów (np. mleko) należy przewieźć do zakładu (np. zakładów mleczarskich) w którym odbywa się dalsza jego przeróbka.

Tego typu zagadnienia nazywają się ogólnie zagadnieniami rozwózkowo-przywozowymi. Dla uproszczenia w dalszym ciągu będziemy rozważać tylko zagadnienie rozwózki.

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Zakładamy, że dane są:

- baza będąca miejscem produkcji jednorodnego towaru oraz postoju parku transportowego;

- dana jest liczba pojazdów o jednakowej ładowności;

- znamy popyt każdego odbiorcy;

- oraz macierz odległości (lub kosztu przewozu, czasu przewozu) pomiędzy wszystkimi punktami odbioru;

- popyt każdego odbiorcy jest mniejszy od ładowności pojazdów, a łączne zapotrzebowanie wszystkich punktów odbioru jest mniejsze od ładowności całego parku transportowego;

- towar jest dostarczany do odbiorcy w okresie planowanym (w dniu, tygodniu) przez jeden pojazd.

Należy ustalić taki zbiór marszrut (trasę dostaw towaru) aby:

1. popyt każdego odbiorcy był zrealizowany przez jeden pojazd.

2. ładowność każdego pojazdu nie była przekroczona

3. długość wszystkich marszrut była minimalna

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Wprowadzamy oznaczenia:

n - liczba odbiorców towaru (punktów odbioru)

P - zbiór indeksów wszystkich punktów odbioru $P = \{1, 2, ..., n\}$

$\overline{P}$ - zbiór indeksów wszystkich punktów odbioru i dostawy $\overline{P} = \{0, 1, 2, ..., n\}$

m - liczba pojazdów

V - zbiór wszystkich połączeń pomiędzy punktami (możliwych marszrut)

$V = \{\langle i, j \rangle : i, j \in \overline{P} \wedge i \neq j\}$

w - jednakowa ładowność wszystkich pojazdów

b_j - popyt j-tego odbiorcy

c_{ij} - odległość od punktu i do punktu j (długość trasy $\langle i, j \rangle$)

Zgodnie z przyjętymi założeniami dane te muszą spełniać warunki:

$$\sum_{j=1}^{n} b_j \leq m \cdot w \text{ oraz } b_j < w, j \in P$$

Dla każdego pojazdu wyznaczamy jedną marszrutę (łącznie będzie ich m).

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Przyjmiemy następujące zmienne decyzyjne:

x_{ij} - ilość towaru (dobra) przewożona na trasie $\langle i, j \rangle$

$$y_{ij} = \begin{cases} 1, & \text{gdy pojazd pokonuje trasę } \langle i, j \rangle \\ 0, & \text{w przypadku przeciwnym} \end{cases}$$

Zadanie decyzyjne (PML) będzie miało postać: Znaleźć takie wartości zmiennych x_{ij} oraz y_{ij}, aby:

Funkcja celu: $\sum_{\langle i,j \rangle \in V} c_{ij} x_{ij} \to \min$

przy warunkach ograniczających:

(1) $\sum_{i \in P} y_{ij} = 1$, dla $(j \in P)$

(2) $\sum_{i \in P} y_{ji} = 1$, dla $(j \in P)$

warunki (1) i (2) oznaczają, że dla każdego odbiorcy wjeżdża i z każdego wyjeżdża jeden pojazd

Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

(3) $\sum_{i \in P} y_{i0} = m$

(4) $\sum_{i \in P} y_{0i} = m$

warunki (3) i (4) wymuszają, aby z bazy wyjechało i do niej wróciło dokładnie m - pojazdów

(5) $\sum_{i \in P} x_{ij} - \sum_{i \in P} x_{ji} = b_j, \ (j \in P)$

warunek (5) oznacza, że w każdym punkcie zostawiamy tyle ile wynosi jego popyt

(6) $\sum_{j \in P} x_{0j} = \sum_{j \in P} b_j$

warunek (6) pozwala wywieźć z bazy tyle towaru ile wynosi łączny popyt odbiorców

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

(7) $x_{ij} \le w \cdot y_{ij}, \ \left(\langle i,j \rangle \in V\right)$

warunek (7) zapewnia, że na każdej trasie przewieziemy towaru nie więcej niż wynosi ładowność pojazdu. Jeśli danej trasy pojazd nie pokonuje, to przewóz towaru na tej trasie jest zerowy.

(8) $x_{ij} \ge 0, \ \left(\langle i,j \rangle \in V\right)$

(9) $y_{ij} \in \{0,1\}, \ \left(\langle i,j \rangle \in V\right)$

Zadanie rozwózki jest zadaniem o dużych wymiarach.

liczba zmiennych, to: $L(z) = 2n(n+1)$

liczba warunków: $L(w) = 3n + n(n+1) + 3$

Dla n=30 odbiorców mamy 8450 zmiennych i 483 warunki.

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Zagadnienie komiwojażera - klasyczny problem optymalizacji dyskretnej

Komiwojażer (dawny sprzedawca objeżdżający domy i oferujący produkty) wyrusza z pewnego miasta (z bazy), ma odwiedzić kilka miejscowości i wrócić do punktu startu. każde z miast może być odwiedzone tylko raz i w dowolnej kolejności.

Dany jest zbiór miast (i=1,2,....,n) oraz nieujemna, kwadratowa macierz odległości (kosztu lub czasu przejazdu) $C=\left[c_{ij}\right] i=1,\ldots,n; j=1,\ldots,n$. Należy znaleźć taką drogę zamkniętą, przechodzącą przez wszystkie miejscowości, która jest minimalna.

Droga zamknięta jest zwana dalej marszrutą i składa się z n odcinków, które będziemy nazywać trasami. Ponieważ marszruta nie może zawierać trasy $\langle i,i \rangle$, więc przyjmujemy, że $c_{ii}=\infty$ dla i=1,2,...,n. Łączna liczba marszrut w problemie komiwojażera jest równa $(n-1)!$. Dla n=10 mamy $9!=362800$ różnych rozwiązań. Przegląd zupełny zbioru rozwiązań w celu znalezienia optymalnego jest efektywny tylko dla małych $n (n \le 8)$.

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Oznaczmy przez:

$V=\{\langle i,j\rangle : i\neq j; i,j=1,2,\ldots,n\}$ - niech będzie zbiorem wszystkich możliwych tras

Zmienne decyzyjne:

$$x_{ij}=\begin{cases}1, & \text{gdy marszruta zawiera trasę } \langle i,j\rangle \\ 0, & \text{w przypadku przeciwnym}\end{cases} \text{ - zmienna binarna}$$

z_j - zmienna całkowita, która każdemu miastu j-temu przyporządkowuje cechę - kolejność odwiedzenia tego miasta (przy założeni że dla punktu startu - bazy $z_{j_B}=0$)

Jeżeli n=5 miast oraz $j_B=1$ (miasto o numerze 1-baza), to przykładowa marszruta może być postaci: $(1,4,5,3,2,1)$, a zmienne kolejności odwiedzeń: $z_1=0, z_4=1, z_5=2, z_3=3, z_2=4$ (oczywiście miasto startu posiadające cechę $z_1=0$ nie może mieć drugiej cechy równej 5)

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

Problem komiwojażera sprowadza się do następującego zadania decyzyjnego (PML):

Wyznaczyć takie wartości zmiennych: x_{ij} oraz z_j, aby:

funkcja celu: $\sum_{\langle i,j \rangle \in V} c_{ij} x_{ij} \to \min$

przy warunkach ograniczających:

(1) $\sum_{i=1}^{n} x_{ij} = 1$, dla $(j = 1, 2, \ldots, n)$

(2) $\sum_{j=1}^{n} x_{ij} = 1$, dla $(i = 1, 2, \ldots, n)$

warunki (1) i (2) oznaczają, że komiwojażer przez każdy punkt przejeżdża tylko jeden raz

❑ Liniowe Modele Optymalizacji Dyskretnej – przykłady dyskretnych problemów decyzyjnych

(3) $z_i - z_j + nx_{ij} \le n-1$, dla $(i=1,2,\ldots,n;\ j=2,3,\ldots,n;\ i \ne j)$

niestety (1) i (2) nie gwarantują, że z wybranych n tras tworzymy tylko jedną marszrutę zamkniętą. Warunek (3) wyklucza możliwość powstawania tzw. podcykli w tworzonej marszrucie.

$z_j \ge 0, z_j \in C,\ (j=1,2,\ldots,n)$

$x_{ij} \in \{0,1\},\ (\langle i,j\rangle \in V)$

Zadanie komiwojażera jest zadaniem o dużych rozmiarach.

Liczba zmiennych to: $L(z) = n + n(n-1) = n^2$

Liczba warunków to: $L(w) = 2n + n(n-1) - (n-1) = 2n + (n-1)^2$

Dla n=10 mamy 100 zmiennych oraz 101 warunków.

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

Algorytm włączania dla zagadnienia komiwojażera:
Oznaczmy N – zbiór wszystkich miast, zaś przez V – zbiór miast włączonych do marszruty.
Formalnie algorytm ten można opisać w kilku krokach:

1. Podstawić $V := \varnothing$;
2. Wybrać dowolne miasto startowe (bazę, z której wyrusza zaopatrzeniowiec): $i_1 \in N; N := N - \{i_1\}; V := \{i_1\}$;
3. Wybrać zgodnie z pewnym kryterium drugie miasto $i_2 \in N$ tworząc marszrutę: $(i_1, i_2, i_1); N := N - \{i_2\}; V := V + \{i_2\}$;
4. Dla kolejnych miast o numerach k = 3 ,…, n – 1 przeprowadzić operacje:
   - wybrać miasto $i_k \in N$ korzystając z pewnego kryterium i włączyć je do V, czyli: $V := V + \{i_k\}$ - (jest to krok selekcyjny algorytmu),
   - dołączyć miasto i_k do marszruty $(i_1, i_2, ..., i_{k-1}, i_1)$ wstawiając je pomiędzy takie miasta, aby długość powstałej marszruty była największa (jest to krok wstawiania w algorytmie),
5. Dołączyć do marszruty ostatnie n - te miasto stosując to samo postępowanie co dla wcześniejszych miast w kroku 4. Po wykonaniu tych 5 – kroków otrzymuje się marszrutę pełną (V – składa się z n – miast, zaś $N = \varnothing$;

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

Przykład praktyczny:

Rozważmy zadanie komiwojażera z **n = 5** miastami. Macierz – **C** (asymetryczna) określa odległości między tymi miastami:

$$C = \begin{bmatrix} \infty & 2 & 4 & 7 & 3 \\ 2 & \infty & 4 & 5 & 7 \\ 8 & 4 & \infty & 7 & 3 \\ 5 & 9 & 2 & \infty & 7 \\ 4 & 8 & 1 & 4 & \infty \end{bmatrix}$$

- Wybieramy jako miasto początkowe miasto o numerze 5. $V = \{5\}; N = N - \{5\} = \{1,2,3,4\}$.
- W celu wyboru drugiego miasta w marszrucie i każdego kolejnego posłużymy się w kryterium selekcji strategią, która nakazuje wybór miasta położonego **najdalej od aktualnej marszruty niepełnej**.
  Liczne eksperymenty numeryczne potwierdzają dużą efektywność tej strategii.

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

Odległość miasta o **numerze - j** od niepełnej marszruty definiowana jest jako **minimalna odległość** między **j - tym miastem** a **wszystkimi miastami** należącymi do tej marszruty.
Określa to wektor odległości:
$d = (d_1, d_2, ..., d_n)$, gdzie: $d_j = \min\{c_{i,j}\};\ \ i \in V; j \in N;$

Dla $j \in V$ (czyli miast należących do marszruty) na **j – tej** pozycji wektora **d** umieszczany jest znak (**minus**).

Dla zadania $d = (4, 8, 1, 4, -)$ - najdalej oddalonym miastem od marszruty jest miasto 2, które dołączamy do marszruty.

Tworzymy marszrutę postaci: $(5, 2, 5)$, której długość wynosi: $F = c_{5,2} + c_{2,5} = 8 + 7 = 15$

$$C = \begin{bmatrix} \infty & 2 & 4 & 7 & 3 \\ 2 & \infty & 4 & 5 & 7 \\ 8 & 4 & \infty & 7 & 3 \\ 5 & 9 & 2 & \infty & 7 \\ \boxed{4 \quad 8 \quad 1 \quad 4} & & & & \infty \end{bmatrix}$$

PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

- **Dla kolejnych wybieranych miast (krok 4 algorytmu) mamy:**

a) $V = \{5,2\}; N = N - \{2\} = \{1,3,4\}$

Wektor $d = \left(\min\{c_{2,1}; c_{5,1}\} = 2, -, \min\{c_{2,3}; c_{5,3}\} = 1, \min\{c_{2,4}; c_{5,4}\} = 4, -\right)$.

Jako kolejne miasto do marszruty włączamy zatem miasto 4.

$V = \{5,2,4\}; N = N - \{4\} = \{1,3\}$. Możemy go wstawić pomiędzy miasta (5,2) lub pomiędzy miasta (2,5).

Jeżeli włączymy między miasta (5,2) utworzymy marszrutę: $(5,4,2,5)$, a tzw. koszt związany z dołączeniem tego miasta wynosi:

$c_{5,4} + c_{4,2} - c_{5,2} = 4 + 9 - 8 = 5$;

Jeżeli włączymy między miasta (2,5) utworzymy marszrutę: $(5,2,4,5)$, a tzw. koszt związany z dołączeniem tego miasta wynosi:

$c_{2,4} + c_{4,5} - c_{2,5} = 5 + 7 - 7 = 5$;

Włączamy zawsze między takie wierzchołki, dla których koszt włączenia jest najmniejszy. W naszym przypadku koszty są równe, zatem włączamy nowe miasto w miejsce jak najbliższe początkowi marszruty. Marszruta jest więc postaci: $(5,4,2,5)$. Oznacza to, że długość aktualnej marszruty wyniesie $F = F + 5 = 15 + 5 = 20$.

$$C = \begin{bmatrix} \infty & 2 & 4 & 7 & 3 \\ 2 & \infty & 4 & 5 & 7 \\ 8 & 4 & \infty & 7 & 3 \\ 5 & 9 & 2 & \infty & 7 \\ 4 & 8 & 1 & 4 & \infty \end{bmatrix}$$

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

b) $V=\{5,2,4\};N=\{1,3\}$

Wektor $d=\left(\min\{c_{2,1};c_{4,1};c_{5,1}\}=2,-,\min\{c_{2,3};c_{4,3};c_{5,3}\}=1,-,-\right)$.

Jako kolejne miasto do marszruty włączamy zatem miasto 1.

$V=\{5,2,4,1\};N=N-\{1\}=\{3\}$. Możemy go wstawić pomiędzy miasta (5,4) (4,2) lub (2,5).

Jeżeli włączymy między miasta (5,4) utworzymy marszrutę: $(5,1,4,2,5)$, a koszt związany z dołączeniem tego miasta wynosi:

$c_{5,1}+c_{1,4}-c_{5,4}=4+7-4=7$;

Jeżeli włączymy między miasta (4,2) utworzymy marszrutę: $(5,4,1,2,5)$, a koszt związany z dołączeniem tego miasta wynosi:

$c_{4,1}+c_{1,2}-c_{4,2}=5+2-9=-2$;

Jeżeli włączymy między miasta (2,5) utworzymy marszrutę: $(5,4,2,1,5)$, a koszt związany z dołączeniem tego miasta wynosi:

$c_{2,1}+c_{1,5}-c_{2,5}=2+3-7=-2$;

W naszym przypadku koszty włączenia są najmniejsze w dwóch przypadkach, włączamy miasto 1 pomiędzy miasta (2,5) – bliżej punktu 5 (początek marszruty). Marszruta jest zatem postaci: $(5,4,2,1,5)$. Oznacza to, że długość aktualnej marszruty wyniesie $F=F-2=20-2=18$.

$$C=\begin{bmatrix}\infty & 2 & 4 & 7 & 3\\ 2 & \infty & 4 & 5 & 7\\ 8 & 4 & \infty & 7 & 3\\ 5 & 9 & 2 & \infty & 7\\ 4 & 8 & 1 & 4 & \infty\end{bmatrix}$$

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

- **Dla ostatniego miasta w marszrucie (krok 5 algorytmu)**

  $V=\{5,2,4,1\};N=\{3\}$

  Wektor $d=\left(-,-,\min\{c_{1,3};c_{2,3};c_{4,3};c_{5,3}\}=1,-,-\right)$.

  Do marszruty możemy włączyć tylko miasto 3.

  $V=\{5,2,4,1,3\};N=N-\{3\}=\varnothing$. Możemy go wstawić pomiędzy miasta (5,4) (4,2), (2,1) lub (1,5).

  Dokonujemy porównań kosztów wstawień:

  - marszruta: (5,3,4,2,1,5) - koszt związany z dołączeniem tego miasta wynosi: $c_{5,3}+c_{3,4}-c_{5,4}=1+7-4=4$;

  - marszruta: (5,4,3,2,1,5) - koszt związany z dołączeniem tego miasta wynosi: $c_{4,3}+c_{3,2}-c_{4,2}=2+4-9=-3$;

  - marszruta: (5,4,2,3,1,5) - koszt związany z dołączeniem tego miasta wynosi: $c_{2,3}+c_{3,1}-c_{2,1}=4+8-2=10$;

  - marszruta: (5,4,2,1,3,5) - koszt związany z dołączeniem tego miasta wynosi: $c_{1,3}+c_{3,5}-c_{1,5}=4+3-3=4$;

  W naszym przypadku koszty włączenia są najmniejsze, gdy włączymy miasto 3 pomiędzy miasta (4,2). Pełna marszruta jest zatem postaci: (5,4,3,2,1,5). Oznacza to, że długość tej pełnej marszruty wynosi: $F=F-3=18-3=15$.

$$C=\begin{bmatrix}\infty & 2 & 4 & 7 & 3\\ 2 & \infty & 4 & 5 & 7\\ 8 & 4 & \infty & 7 & 3\\ 5 & 9 & 2 & \infty & 7\\ 4 & 8 & 1 & 4 & \infty\end{bmatrix}$$

❑ PROGRAMOWANIE SIECIOWE – zadanie komiwojażera (przybliżony algorytm rozwiązania)

Uwagi:

- wybierając jako miasto początkowe inne miasto możemy otrzymać inne rozwiązanie. Zaleca się w tym przypadku wyznaczyć algorytmem przybliżonym rozwiązania dla każdego wierzchołka jako początkowego i wybrać spośród nich najlepsze;
- gdy w danej iteracji w wektorze – d pojawi się więcej niż jeden element maksymalny (można dołączyć więcej niż jedno miasto), to dołączamy miasto o mniejszym numerze;