# ZAGADNIENIA OPTYMALIZACJI NIELINIOWEJ

## w aspekcie zadań

# PROGRAMOWANIA LINIOWEGO

# ❑ Zadania optymalizacji nieliniowej – zagadnienia ogólne

**Ogólny problemem optymalizacji nieliniowej (programowaniem nieliniowym)** nazywamy zadanie decyzyjne postaci:

$$f(x_1,...,x_n) \to \max(\min)$$

$$(1)\quad \begin{cases} g^j(x_1,...,x_n) - c_j \le 0 \ \text{ dla } \ (j=1,...,r); \\ g^j(x_1,...,x_n) - c_j = 0 \ \text{ dla } \ (j=r+1,...,m); \\ x_1 \ge 0, x_2 \ge 0, ...x_n \ge 0. \end{cases}$$

gdy funkcja celu $f(x) = f(x_1,...,x_n)$, lub chociaż jeden z warunków ograniczających: $g^j(x) = g^j(x_1,...,x_n)$ jest **funkcją nieliniową**.

Jeżeli w zadaniu (1) warunki ograniczające nie występują, a funkcja celu jest postaci nieliniowej, to zadanie takie nosi nazwę optymalizacji bezwarunkowej (problemu bez ograniczeń).

Zakładamy, że funkcje: $f, g^j$ - są funkcjami ciągłymi.

Niech $D \subset R^n$ będzie zbiorem wypukłym, funkcję o wartościach rzeczywistych nazywamy funkcją wypukłą w zbiorze D, jeżeli dla dowolnych $x^1, x^2 \in D$ oraz dowolnego $\alpha \in [0,1]$ zachodzi nierówność:

$$f\left[\alpha x^1 + (1-\alpha)x^2\right] \le \alpha f(x^1) + (1-\alpha) f(x^2)$$

Jeżeli $x^1 \ne x^2$ i $\alpha \in (0,1)$ oraz spełniona jest nierówność:

$f\left[\alpha x^1 + (1-\alpha)x^2\right] < \alpha f(x^1) + (1-\alpha) f(x^2)$, to funkcję f – nazywamy ściśle wypukłą

Funkcja $f(x)$ jest funkcją wklęsłą jeśli funkcja $-f(x)$ - jest wypukła (i odwrotnie).

Funkcja $f(x)$ jest funkcją ściśle wklęsłą jeśli funkcja $-f(x)$ - jest ściśle wypukła (i na odwrót).

# ❑ Zadania optymalizacji nieliniowej – zagadnienia ogólne

Załóżmy, że funkcja f – jest różniczkowalna w $R^n$, wektor $\nabla f(x_0) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right]$ - nazywa się gradientem funkcji f w punkcie $x_0$ - wskazuje kierunek, w którym przyrost wartości funkcji jest największy.

Załóżmy, że funkcja f – jest dwukrotnie różniczkowalna w $R^n$, to macierz:

$$\nabla^2 f(x_0) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \ldots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \ldots & \ldots & \ldots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \ldots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$ - nazywa się Hesjanem funkcji

Funkcja jest wypukła w otwartym (niezawierającym punktów brzegowych) zbiorze D wtedy i tylko wtedy, gdy dla każdego $x \in D$ - Hesjan jest nieujemnie określony.

Funkcja jest ściśle wypukła w otwartym (niezawierającym punktów brzegowych) zbiorze D wtedy i tylko wtedy, gdy dla każdego $x \in D$ - Hesjan jest dodatnio określony.

Macierz kwadratową $A = \left[a_{i,j}\right]_{n \times n}$, nazywamy dodatnio określoną, gdy: $M_1 = a_{11} > 0$,

$$M_2 = \det\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} > 0, \ldots, \; M_n = \det\begin{bmatrix} a_{11} & \ldots & a_{12} \\ \ldots & \ldots & \ldots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix} > 0$$

Macierz nazywamy ujemnie określoną, gdy: $M_1 < 0, M_2 > 0, M_3 < 0, \ldots$

# ❑ Zadania optymalizacji nieliniowej – zagadnienia ogólne

Problemy optymalizacji nieliniowej dzielimy ogólnie na:

- problemy programowania wypukłego
  - Minimalizacja funkcji celu wypukłej, lub maksymalizacja funkcji celu wklęsłej
  - zbiór warunków ograniczających jest zbiorem wypukłym

  Niech funkcje: $f, g^j; j=1,..,m_1, h^j; j=1,...,m_2$ - są funkcjami wypukłymi, wtedy można udowodnić, że zbiory ograniczone warunkami: $g^j(x_1,...,x_n)-c_j \le 0$ oraz $h^j(x_1,...,x_n)-c_j=0$ są zbiorami wypukłymi. Ponieważ część wspólna zbiorów wypukłych jest zbiorem wypukłym, więc zadania programowania wypukłego przyjmują postać:

$$f(x_1,...,x_n) \to \min$$

lub

$$-f(x_1,...,x_n) \to \max$$

Przy warunkach:

$$\begin{cases} g^j(x_1,...,x_n) \le 0 \text{ dla } (j=1,...,m_1); \\ h^j(x_1,...,x_n)=0 \text{ dla } (j=1,...,m_2); \\ x_1 \ge 0, x_2 \ge 0, ...x_n \ge 0. \end{cases}$$

- problemy programowania niewypukłego

  problemy decyzyjne nieliniowe, które niespełnianą warunków programowania wypukłego nazywamy zadaniami programowania niewypukłego

# ❑ Zadania optymalizacji nieliniowej – zagadnienia ogólne

Szczególnym przypadkiem zadań programowania wypukłego jest programowanie kwadratowe.

Zakłada on, że funkcja celu jest wypukłą funkcją kwadratową, zaś funkcje $g^j(x_1,...,x_n)$ z warunków ograniczających są funkcjami liniowymi (a więc tworzą obszar wypukły).

Zadanie programowania kwadratowego przyjmuje postać:

$$cx + \frac{1}{2}x^T Ex \rightarrow \min$$

$$Ax \leq b$$

$$x \geq 0$$

gdzie:

$x = [x_1,...,x_n]^T$ - wektor kolumnowy zmiennych decyzyjnych

$c = [c_1,...,c_n]$ - wektor wierszowy współczynników funkcji celu składnika liniowego

$b = [b_1,...,b_m]^T$ - wektor kolumnowy wyrazów wolnych

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

$E = [e_{i,j}]_{n \times n}$ - macierz ujemnie określona – współczynników składnika kwadratowego (co gwarantuje wypukłość składnika kwadratowego funkcji celu)

# ❑ Zadania optymalizacji nieliniowej – zagadnienia ogólne

Dla zadań optymalizacji nieliniowej w postaci kanonicznej (warunki ograniczające w postaci równości) możemy w ogólnym przypadku zastosować metodę tzw. czynników nieoznaczonych Lagrange'a.

W metodzie tej zamiast poszukiwać ekstremum warunkowego postaci:

$$f(x_1,\ldots,x_n) \rightarrow \max(\min)$$

$$(2)\ \begin{cases} g^j(x_1,\ldots,x_n) - c_j = 0 \quad \text{dla} \quad (j = 1,\ldots,m); \\ x_1 \geq 0, x_2 \geq 0, \ldots x_n \geq 0. \end{cases}$$

poszukujemy ekstremum bezwarunkowego dla tzw. funkcji Lagrange'a utworzonej w oparciu o wyjściową funkcje celu f(X) poprzez włączenie do tej funkcji warunków ograniczających z odpowiednimi (sztucznie wprowadzonymi) czynnikami nieoznaczonymi (zakładamy, że m<n).

Dla zadania programowania nieliniowego w postaci kanonicznej (2) funkcja Lagrange'a ma postać:

$$L(X;\lambda) = L(x_1,\ldots,x_n;\lambda_1,\ldots,\lambda_m) = f(x_1,\ldots,x_n) + \sum_{j=1}^{m} \lambda_j \left[ c_j - g^j(x_1,\ldots,x_n) \right]$$

**Warunek konieczny istnienia ekstremum warunkowego zadania jest następujący:** zerowanie się pochodnych cząstkowych funkcji Lagrange'a:

$$(*)\ \begin{cases} \dfrac{\partial L}{\partial \lambda_j} = c_j - g^j(x_1,\ldots,x_n) = 0; \quad \text{dla} \quad (j = 1,\ldots,m); \\ \dfrac{\partial L}{\partial x_i} = \dfrac{\partial f}{\partial x_i} - \sum_{j=1}^{m} \lambda_j \dfrac{\partial g^j}{\partial x_i} = 0; \quad \text{dla} \quad (i = 1,2,\ldots,n); \end{cases}$$

# ❑ Zadania optymalizacji nieliniowej – zagadnienia ogólne

**Warunek dostateczny istnienia ekstremum warunkowego zadania (2) w postaci kanonicznej jest następujący:**

$$L = f(x_1,...,x_n) + \sum_{j=1}^{m} \lambda_j \left[ c_j - g^j(x_1,...,x_n) \right]$$ **- Funkcja Lagrange'a (m<n)**

**Hesjan obrzeżony**

$$|\overline{H}| = \begin{vmatrix} 0 & 0 & \dots & 0 & g_1^1 & g_2^1 & \dots & g_n^1 \\ 0 & 0 & \dots & 0 & g_1^2 & g_2^2 & \dots & g_n^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & g_1^m & g_2^m & \dots & g_n^m \\ g_1^1 & g_1^2 & \dots & g_1^m & L_{11} & L_{12} & \dots & L_{1n} \\ g_2^1 & g_2^2 & \dots & g_2^m & L_{21} & L_{22} & \dots & L_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_n^1 & g_n^2 & \dots & g_n^m & L_{n1} & L_{n2} & \dots & L_{nn} \end{vmatrix}$$

**gdzie:** $g_i^j = \frac{\partial g^j}{\partial x_i}; L_{pq} = \frac{\partial^2 L}{\partial x_q \partial x_p}$

$|\overline{H}_2|$

**Dla maksimum funkcji f – warunkiem dostatecznym jest, aby** $|\overline{H}_{m+1}|, |\overline{H}_{m+2}|, ..., |\overline{H}_n| = |\overline{H}|$ **zmieniały znak (dla** $|\overline{H}_{m+1}|$ **znak taki jak** $(-1)^{m+1}$ **)**

**Dla minimum funkcji f – warunkiem dostatecznym jest, aby miały one ten sam znak (i to taki jak dla** $(-1)^m$ **)**

# PROGRAMOWANIE DYNAMICZNE

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

Twórcą teorii **programowania dynamicznego** jest **Richard Bellman**, który opracował jej podstawy teoretyczne.

Wyczerpujący opis teoretyczny oraz metodologię wykorzystania programowania dynamicznego do zagadnień podejmowaniu optymalnych decyzji można znaleźć między innymi w pracy monograficznej:

**[1] Bellman R., Dreyfus S. F., Programowanie Dynamiczne, PWE, Warszawa 1967.**

**Metodologia programowania dynamicznego:**

Formalnie rzecz biorąc, metody programowania dynamicznego polegają na **zamianie** zadania optymalizacyjnego z **N** zmiennymi decyzyjnymi (znalezienia ekstremum warunkowego funkcji **N** – zmiennych) na **N** zadań z **jedną zmienną decyzyjną**, przy czym zadania te są powiązane ze sobą określoną **zależnością rekurencyjną** (na każdym etapie zadania składowego wyznacza się ekstremum warunkowe uwzględniając rezultat osiągnięty na etapie poprzednim).

Postępując w ten sposób upraszczamy proces rachunkowy (zamiast rozwiązywać zadanie złożone rozwiązujemy zadania prostsze).

Operacja rozbicia zadania optymalizacyjnego na zadania składowe jest możliwa tylko wtedy, gdy **funkcja celu** zadania jest tzw. **funkcją separowalną**.

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

**Określenie:**

Funkcja **N** zmiennych $F(x_1, x_2, ..., x_N)$ będzie funkcją **separowalną**, jeśli $F(x_1, x_2, ..., x_N) = f_1(x_1) \oplus f_2(x_2) \oplus ... \oplus f_N(x_N)$.

Operację: $x \oplus y$ należy rozumieć jako:

1) $x \oplus y := x + y$, albo
2) $x \oplus y := x \cdot y$, albo
3) $x \oplus y := \min\{x, y\}$, albo
4) $x \oplus y := \max\{x, y\}$.

**Uwaga:** Następujące funkcje celu:

$F_1(x_1, x_2, x_3) = 5g_1(x_1) + 8g_3(x_3)x_3 + 4g_2(x_2)x_2$

$F_2(x_1, x_2, x_3) = \ln[9g_1(x_1) + 2g_2(x_2) + 4g_3(x_3)]$ - **mogą być** funkcjami celu w programowaniu dynamicznym.

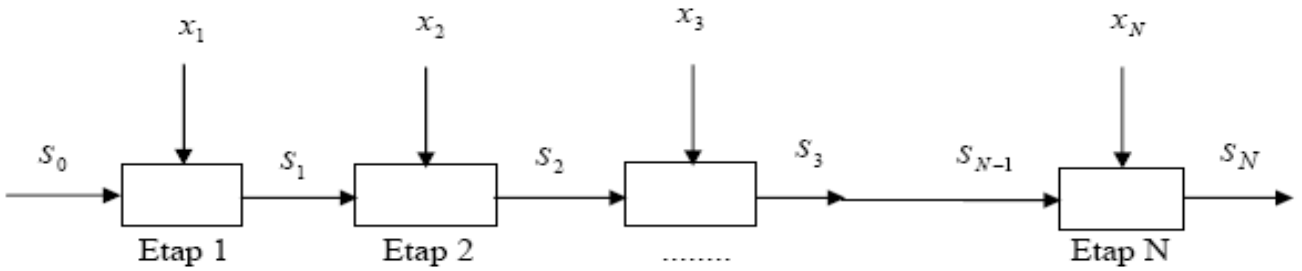
**Nie może** być to natomiast funkcja celu postaci:

$F_3(x_1, x_2, x_3) = g_1(x_1)x_2 + g_2(x_2)x_3 + g_3(x_3)$

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

Metody programowania dynamicznego są w głównej mierze wykorzystywane do rozwiązywania zadań optymalizacyjnych dla tzw. **wieloetapowych procesów decyzyjnych**.

Ogólny schemat wieloetapowego procesu decyzyjnego przedstawia następujący rysunek:




Schemat ten przedstawia dowolny proces (np. realizację jakiejkolwiek działalności), którego przebieg można podzielić na **N - etapów**. Na dowolnym etapie tego procesu możemy wyróżnić następujące elementy:

1) Stan wejściowy procesu do danego etapu (na schemacie - $s_{i-1}, (i = 1,..., N)$) – jest to stan jaki osiągnął proces w wyniku realizacji etapu poprzedniego.
2) Decyzję podejmowaną na danym etapie (na schemacie - $x_i, (i = 1,..., N)$).
3) Stan wyjściowy procesu z danego etapu (na schemacie - $s_i, (i = 1,..., N)$) – stan ten zależy od stanu wejściowego oraz od podjętej decyzji na danym etapie.

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

Stan procesu można opisywać za pomocą jednego lub kilku parametrów – zwanych: **zmiennymi stanu**. W podanym schemacie proces decyzyjny jest opisywany za pomocą jednej charakterystyki – jednej **zmiennej stanu**.

Oznaczmy przez: $S_i \ (i = 1,...,N)$ - zbiór możliwych w **i - tym** etapie wartości zmiennej stanu - $s_i$ (**zbiór możliwych stanów**). Natomiast przez: $D_i$ $(i = 1,...,N)$ - zbiór możliwych decyzji w **i - tym** etapie. Oznacza to, że zmienna decyzyjna może przyjmować wartości z tego zbioru $(x_i \in D_i)$.

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

Formalnie wieloetapowy proces decyzyjny możemy wyrazić następującymi zależnościami rekurencyjnymi:

$$s_i = g_i(s_{i-1}, x_i), i = 1,2,...,N, s_i \in S_i, x_i \in D_i$$

$s_0$ - jest ustalonym stanem początkowym procesu

**Uwaga:**

Zależności te przedstawiają ważną cechę wieloetapowych procesów decyzyjnych, a mianowicie, że stan procesu $s_i$ - osiągnięty w i - tym etapie zależy od stanu wejściowego $s_{i-1}$ - do i - tego etapu oraz od decyzji $x_i$ - podjętej na tym etapie.

Problem, który należy rozwiązać w każdym wieloetapowym procesie decyzyjnym polega na określeniu ciągu decyzji: $x_1^*, x_2^*, ..., x_N^*$, $(x_i^* \in D_i)$, przy których ustalona funkcja celu dla całości procesu przebiegającego w N - etapach osiąga wartość optymalną (min lub max).

Ciąg decyzji optymalnych wyznaczonych dla każdego etapu: $x_1^*, x_2^*, ..., x_N^*$ nazywa się: **polityką optymalną** wieloetapowego procesu decyzyjnego.

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

**Schemat ogólny programowania dynamicznego:**

Rozpatrzmy model decyzyjny wieloetapowego procesu decyzyjnego. Oznaczmy:

$X = (x_1,...,x_N)$ - wektor zmiennych decyzyjnych ustalanych na każdym etapie;

$s_0$ - zadany stan początkowy procesu;

$s_1, s_2,..., s_N$ - stany wyjściowe procesu dla poszczególnych etapów;

$Z_1(s_0, x_1)$ - wartość funkcji celu uzyskana w pierwszym etapie przy zadanym stanie początkowym;

$Z_2(s_1, x_2), Z_3(s_2, x_3),..., Z_{N-1}(s_{N-2}, x_{N-1}), Z_N(s_{N-1}, x_N)$ - odpowiednio wartości funkcji celu w kolejnych etapach: 2,3,…,N.

Oczywiste jest, że zachodzi: $Z(s_0, X) = Z_1(s_0, x_1) + ... + Z_N(s_{N-1}, x_N)$

Należy ustalić **optymalną strategię** – ciąg decyzji $X^* = (x_1^*,..., x_N^*)$, taką aby $Z(s_0, X^*) \to \max(\min)$, **przy ograniczeniach**: $X \subset \Omega$, gdzie $\Omega$ - obszar **określenia zadania wyjściowego**.

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

W celu rozwiązania tego zadania dokonujemy dekompozycji na N – zadań (etapów) otrzymując rodzinę zadań:

Niech $\Omega_N, \Omega_{N-1,N}, \ldots, \Omega_{1,2,\ldots,N} \equiv \Omega$ - oznacza rozbicie obszaru dla zadania wyjściowego na obszary ograniczające zmienne decyzyjne dla poszczególnych etapów.

Oznaczmy przez:

$F_1(s_{N-1}) = \max_{x_N \in \Omega_N}(\min) Z_N(s_{N-1}, x_N)$ - optymalną wartość funkcji celu uzyskaną na 1 – rozpatrywanym etapie.

Dalej uzyskujemy, że:

$F_2(s_{N-2}) = \max_{x_{N-1} \in \Omega_{N-1,N}}(\min)\left[Z_{N-1}(s_{N-2}, x_{N-1}) + F_1(s_{N-1})\right]$ - optymalna wartość funkcji celu w 2 – rozpatrywanym etapie.

Analogicznie dla 3 – rozpatrywanego etapu mamy:

$F_3(s_{N-3}) = \max_{x_{N-2} \in \Omega_{N-2,N-1,N}}(\min)\left[Z_{N-2}(s_{N-3}, x_{N-2}) + F_2(s_{N-2})\right]$

..... i dla kolejnych

Wreszcie dla N – tego rozpatrywanego etapu:

$$F_N(s_0) = \max_{x_1 \in \Omega_{1,2,\ldots,N} \equiv \Omega}(\min)\left[Z_1(s_0, x_1) + F_{N-1}(s_1)\right]$$

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

**Uwaga:**

Z równania tego wynika, że optymalna wartość funkcji celu dla N – etapowego procesu decyzyjnego jest równa optymalnej wartości funkcji celu ze względu na pierwszą decyzję, przy założeniu stanu początkowego $s_0$ - procesu oraz maksymalnej wartości funkcji celu dla procesu (N-1) – etapowego.

Powyższy ciąg równań funkcyjnych wyraża tzw. **zasadę optymalności** – sformułowaną przez **R. Bellmana**.

*„Niezależnie od tego jakie były decyzje początkowe, każda następna decyzja w ciągu sekwencyjnym jest decyzją optymalną z punktu widzenia stanu wynikłego z decyzji poprzednich. W efekcie końcowym otrzymamy zawsze strategię optymalną”*

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

Stosując metodologię programowania dynamicznego oraz ideę algorytmu sekwencyjnego można rozwiązać bardzo wiele różnorodnych problemów decyzyjnych.

**Przytoczymy tutaj tylko niektóre z tych problemów:**

- Problem optymalnego wyboru przedsięwzięć inwestycyjnych.
- Problem wyznaczenia najkrótszej trasy przejazdu pomiędzy dwoma miejscowościami w wieloetapowej sieci drogowej (transportowej).
- Problem optymalnego wyznaczenia wielkości odnawianych zasobów magazynowych w wieloetapowym (np. co kwartał) procesie dostaw magazynowych itp.

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

**Przykład 1 - Problem optymalnego rozdziału inwestycji**

Międzynarodowe przedsiębiorstwo transportowe planuje zainwestować 10 000 000 $ w rozwój sieci swoich placówek na nowych potencjalnych rynkach świadczenia usług. Pod uwagę bierze 4 strefy (rynki), a mianowicie: (I) – rosyjski, (II) – ukraiński, (III) – białoruski, (IV) – polski.

Firma konsultingowa przeprowadziła wstępne badania opracowując tabelę oraz krzywe potencjalnych zysków dla poszczególnych krajów lokalizacji sieci swoich placówek, przy zainwestowaniu „x" - jednostek pieniężnych (jednostka to 1 000 000 $).

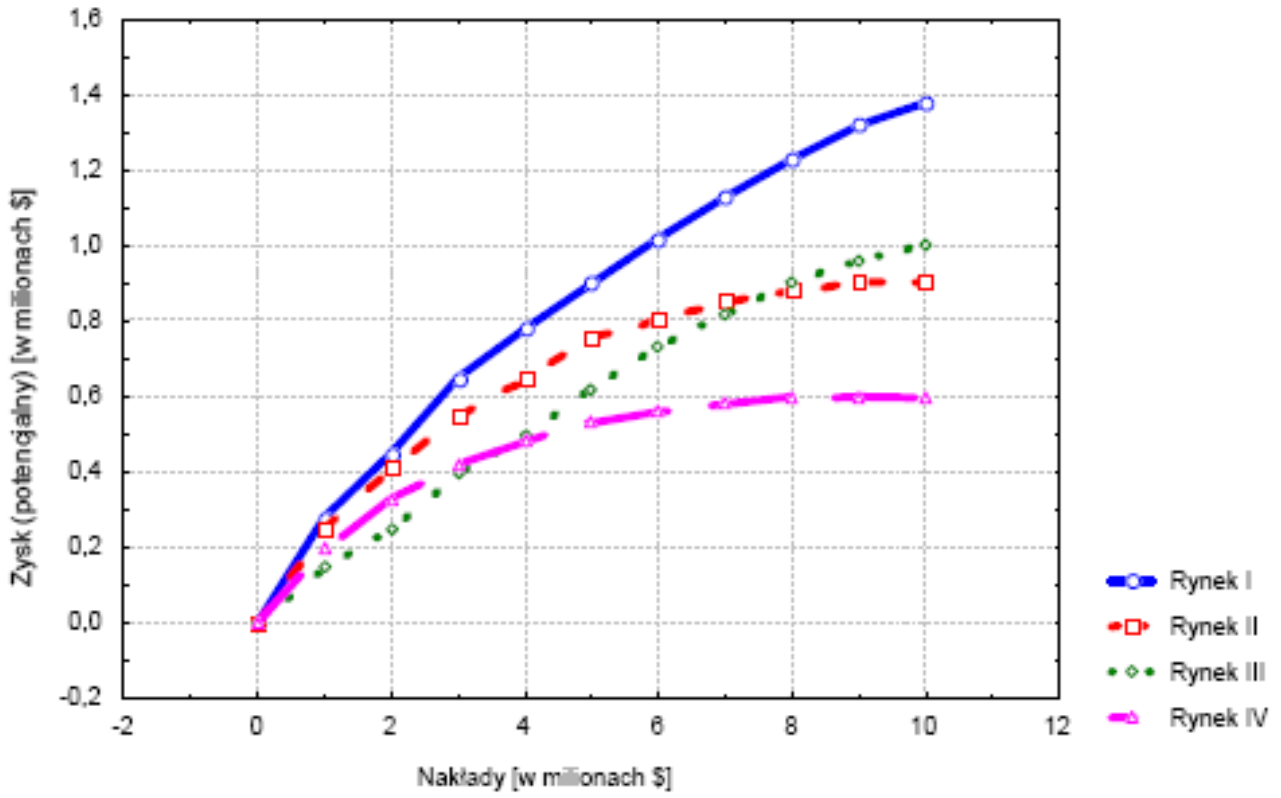
# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

**Badania firmy konsultingowej przedstawia tabela:**

| Suma zainwestowana (w milionach $) | Przewidywany zysk (w milionach $) przy inwestycji w danej strefie rynku I – IV | | | |
|---|---|---|---|---|
| | I etap $f_1$ | II etap $f_2$ | III etap $f_3$ | IV etap $f_4$ |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0,28 | 0,25 | 0,15 | 0,20 |
| 2 | 0,45 | 0,41 | 0,25 | 0,33 |
| 3 | 0,65 | 0,55 | 0,40 | 0,42 |
| 4 | 0,78 | 0,65 | 0,50 | 0,48 |
| 5 | 0,90 | 0,75 | 0,62 | 0,53 |
| 6 | 1,02 | 0,80 | 0,73 | 0,56 |
| 7 | 1,13 | 0,85 | 0,82 | 0,58 |
| 8 | 1,23 | 0,88 | 0,90 | 0,60 |
| 9 | 1,32 | 0,90 | 0,96 | 0,60 |
| 10 | 1,38 | 0,90 | 1,00 | 0,60 |

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

**Krzywe potencjalnego zysku prezentują poniższe wykresy:**



# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

**Zadanie decyzyjne jest następujące:** Jak rozłożyć kwotę inwestycyjną nie przekraczającą 10 jednostek, aby sumaryczny zysk (potencjalnie) był jak największy ?

Jest to zagadnienie kombinatoryczne, ale mające bardzo dużo kombinacji, dlatego zastosujemy algorytm sekwencyjny **R. Bellmana**.

**Przyjmiemy następujące oznaczenia:**

$f_1(x_1)$ - funkcja zysku z rynku I, przy inwestycji kwoty $x_1$,

$f_2(x_2)$ - funkcja zysku z rynku II, przy inwestycji kwoty $x_2$,

$f_3(x_3)$ - funkcja zysku z rynku III, przy inwestycji kwoty $x_3$,

$f_4(x_4)$ - funkcja zysku z rynku IV, przy inwestycji kwoty $x_4$,

$x_i \in \{0,1,...,10\}, i = 1,2,3,4.$

**Ponadto oznaczymy dla potrzeb algorytmu sekwencyjnego:**

$F_{12}(A)$ - maksymalny zysk przy optymalnym podziale środków inwestycyjnych w strefie I i II, tzn. $x_1 + x_2 = A$, $A \in \{0,1,...,10\}$

$F_{123}(A)$ - maksymalny zysk przy optymalnym podziale środków inwestycyjnych w strefie I, II i III, tzn. $x_1 + x_2 + x_3 = A$, $A \in \{0,1,...,10\}$

$F_{1234}(A)$ - zysk przy optymalnym podziale kwoty inwestycyjnej wielkości **A** w czterech strefach: I - IV, tzn. $x_1 + x_2 + x_3 + x_4 = A$, $A \in \{0,1,...,10\}$,

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

Stosując opisany wcześniej ogólnie algorytm sekwencyjny **R. Bellmana** możemy uzyskać optymalne decyzje podziału środków inwestycyjnych.
Dla przykładu pokażemy jak optymalnie zainwestować kwotę A=2 000000 $ (2 jednostki) w rozpatrywane 4 rynki:

Na pierwszym etapie bierzemy pod uwagę tylko rynek rosyjski – w jeden rynek optymalnie inwestujemy zgodnie z wartościami funkcji celu: $F_1(x) = f_1(x)$ - podanymi w tabeli.

W drugim etapie dołączamy drugi rynek – ukraiński:

$$(*)\ F_{12}(A) = \max_{\substack{x_2=0,1,\ldots,A \\ x_1+x_2=A}} \{f_2(x_2) + f_1(A-x_2)\}$$

$$F_{12}(2) = \max\{f_2(0)+f_1(2-0), f_2(1)+f_1(2-1), f_2(2)+f_1(2-2)\} =$$
$$= \max\{0+0.45; 0.25+0.28; 0.41+0\} = 0.53$$

**Strategia optymalna:** $\langle 1,1 \rangle$

**Ponadto otrzymujemy:**

$$F_{12}(0) = \max\{f_2(0)+f_1(0)\} = 0$$
$$F_{12}(1) = \max\{f_2(0)+f_1(1-0), f_2(1)+f_1(1-1)\} = \max\{0.28; 0.25\} = 0.28$$

**Badania firmy konsultingowej przedstawia tabela:**

| Suma zainwestowana (w milionach $) | Przewidywany zysk (w milionach $) przy inwestycji w danej strefie rynku I – IV | | | |
|---|---|---|---|---|
| | I etap $f_1$ | II etap $f_2$ | III etap $f_3$ | IV etap $f_4$ |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0,28 | 0,25 | 0,15 | 0,20 |
| 2 | 0,45 | 0,41 | 0,25 | 0,33 |
| 3 | 0,65 | 0,55 | 0,40 | 0,42 |
| 4 | 0,78 | 0,65 | 0,50 | 0,48 |
| 5 | 0,90 | 0,75 | 0,62 | 0,53 |
| 6 | 1,02 | 0,80 | 0,73 | 0,56 |
| 7 | 1,13 | 0,85 | 0,82 | 0,58 |
| 8 | 1,23 | 0,88 | 0,90 | 0,60 |
| 9 | 1,32 | 0,90 | 0,96 | 0,60 |
| 10 | 1,38 | 0,90 | 1,00 | 0,60 |

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

W 3 - etapie wyznaczymy optymalną wartość funkcji celu biorąc pod uwagę trzy rynki: rosyjski, ukraiński, białoruski:

$$(**)\ F_{123}(A) = \max_{\substack{x_3=0,1,2,\ldots,A \\ x_1+x_2+x_3=A}} \{f_3(x_3) + F_{12}(A-x_3)\}$$

$$F_{123}(2) = \max\{f_3(0)+F_{12}(2-0); f_3(1)+F_{12}(2-1); f_3(2)+F_{12}(2-2)\} =$$
$$= \max\{0+0.53; 0.15+0.28; 0.25+0\} = 0.53$$

**Strategia optymalna:** $\langle 1,1,0 \rangle$

**Ponadto otrzymujemy:**

$$F_{123}(0) = \max\{f_3(0)+F_{12}(0)\} = 0$$

$$F_{123}(1) = \max\{f_3(0)+F_{12}(1-0); f_3(1)+F_{12}(1-1)\} = \max\{0.28; 0.15\} = 0.28$$

**Badania firmy konsultingowej przedstawia tabela:**

| Suma zainwestowana (w milionach $) | Przewidywany zysk (w milionach $) przy inwestycji w danej strefie rynku I – IV | | | |
|---|---|---|---|---|
| | I etap $f_1$ | II etap $f_2$ | III etap $f_3$ | IV etap $f_4$ |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0,28 | 0,25 | 0,15 | 0,20 |
| 2 | 0,45 | 0,41 | 0,25 | 0,33 |
| 3 | 0,65 | 0,55 | 0,40 | 0,42 |
| 4 | 0,78 | 0,65 | 0,50 | 0,48 |
| 5 | 0,90 | 0,75 | 0,62 | 0,53 |
| 6 | 1,02 | 0,80 | 0,73 | 0,56 |
| 7 | 1,13 | 0,85 | 0,82 | 0,58 |
| 8 | 1,23 | 0,88 | 0,90 | 0,60 |
| 9 | 1,32 | 0,90 | 0,96 | 0,60 |
| 10 | 1,38 | 0,90 | 1,00 | 0,60 |

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

Wreszcie w 4 etapie wyznaczymy wartość optymalną funkcji celu biorąc pod uwagę wszystkie cztery rynki.

$$(***)\ F_{1234}(A) = \max_{\substack{x_4 = 0,1,2,\ldots,A \\ x_1 + x_2 + x_3 + x_4 = A}} \{f_4(x_4) + F_{123}(A - x_4)\}$$

$$F_{1234}(2) = \max\{f_4(0) + F_{123}(2-0); f_4(1) + F_{123}(2-1); f_4(2) + F_{123}(2-2)\} =$$
$$= \max\{0 + 0.53; 0.20 + 0.28; 0.33 + 0\} = 0.53$$

**Strategia optymalna:** $\langle 1,1,0,0 \rangle$

Otrzymany wynik jest wynikiem końcowym, gdy inwestor decyduje się na wydanie 2 - jednostek pieniężnych – optymalny rozdział inwestycji: przeznaczyć po jednostce na rynki I i II.

**Badania firmy konsultingowej przedstawia tabela:**

| Suma zainwestowana (w milionach $) | Przewidywany zysk (w milionach $) przy inwestycji w danej strefie rynku I – IV | | | |
|---|---|---|---|---|
| | I etap $f_1$ | II etap $f_2$ | III etap $f_3$ | IV etap $f_4$ |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0,28 | 0,25 | 0,15 | 0,20 |
| 2 | 0,45 | 0,41 | 0,25 | 0,33 |
| 3 | 0,65 | 0,55 | 0,40 | 0,42 |
| 4 | 0,78 | 0,65 | 0,50 | 0,48 |
| 5 | 0,90 | 0,75 | 0,62 | 0,53 |
| 6 | 1,02 | 0,80 | 0,73 | 0,56 |
| 7 | 1,13 | 0,85 | 0,82 | 0,58 |
| 8 | 1,23 | 0,88 | 0,90 | 0,60 |
| 9 | 1,32 | 0,90 | 0,96 | 0,60 |
| 10 | 1,38 | 0,90 | 1,00 | 0,60 |

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

Dalej należy przeprowadzić analogiczne rozważania rachunkowe dla $A = 3,4,...,10$. Ostateczne wyniki podaje tabela poniżej:

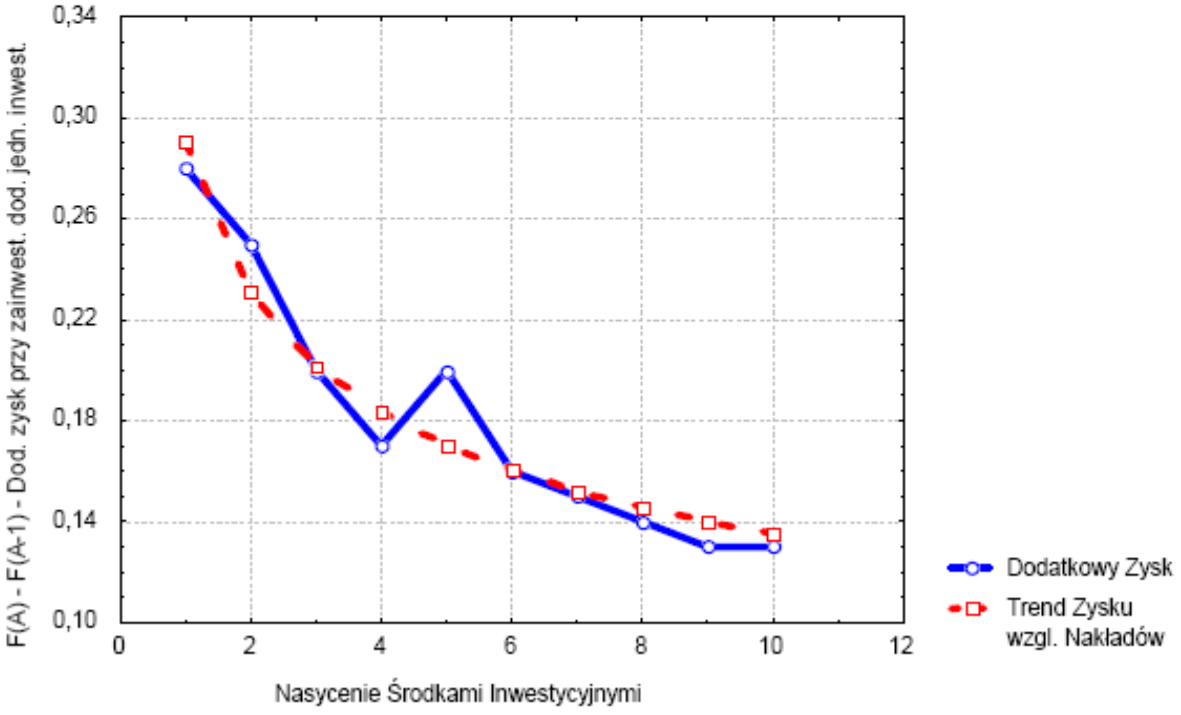
| Nakł. Inwest. (A) | Zysk pot. ze strefy $f_1$ | $f_2$ | $f_3$ | $f_4$ | $F_{12}(A)$ | Strat. Opt. | $F_{123}(A)$ | Strat. Opt. | $F_{1234}(A)$ | Strat. Opt. | $\Delta F_{1234} = F_{1234}(A) - F_{1234}(A-1)$ (Przyrosty Zysku) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | ⟨0,0⟩ | 0 | ⟨0,0,0⟩ | 0 | ⟨0,0,0,0⟩ | --- |
| 1 | 0,28 | 0,25 | 0,15 | 0,20 | 0,28 | ⟨1,0⟩ | 0,28 | ⟨1,0,0⟩ | 0,28 | ⟨1,0,0,0⟩ | 0,28 |
| 2 | 0,45 | 0,41 | 0,25 | 0,33 | 0,35 | ⟨1,1⟩ | 0,53 | ⟨1,1,0⟩ | 0,53 | ⟨1,1,0,0⟩ | 0,25 |
| 3 | 0,65 | 0,55 | 0,40 | 0,42 | 0,70 | ⟨2,1⟩ | 0,70 | ⟨2,1,0⟩ | 0,73 | ⟨1,1,0,1⟩ | 0,20 |
| 4 | 0,78 | 0,65 | 0,50 | 0,48 | 0,90 | ⟨3,1⟩ | 0,90 | ⟨3,1,0⟩ | 0,90 | ⟨3,1,0,0⟩<br>⟨2,1,0,1⟩ | 0,17 |
| 5 | 0,90 | 0,75 | 0,62 | 0,53 | 1,06 | ⟨3,2⟩ | 1,06 | ⟨3,2,0⟩ | 1,10 | ⟨3,1,0,1⟩ | 0,20 |
| 6 | 1,02 | 0,80 | 0,73 | 0,56 | 1,20 | ⟨3,3⟩ | 1,21 | ⟨3,2,1⟩ | 1,26 | ⟨3,2,0,1⟩ | 0,16 |
| 7 | 1,13 | 0,85 | 0,82 | 0,58 | 1,33 | ⟨4,3⟩ | 1,35 | ⟨3,3,1⟩ | 1,41 | ⟨3,2,1,1⟩ | 0,15 |
| 8 | 1,23 | 0,88 | 0,90 | 0,60 | 1,44 | ⟨5,3⟩ | 1,48 | ⟨4,3,1⟩ | 1,55 | ⟨3,3,1,1⟩ | 0,14 |
| 9 | 1,32 | 0,90 | 0,96 | 0,60 | 1,57 | ⟨6,3⟩ | 1,60 | ⟨5,3,1⟩<br>⟨3,3,3⟩ | 1,68 | ⟨4,3,1,1⟩<br>⟨3,3,1,2⟩ | 0,13 |
| 10 | 1,38 | 0,90 | 1,00 | 0,60 | 1,68 | ⟨7,3⟩ | 1,73 | ⟨4,3,3⟩ | 1,81 | ⟨4,3,1,2⟩ | 0,13 |

# ❑ PODSTAWY PROGRAMOWANIA DYNAMICZNEGO

**Wniosek:**

Dodatkowy zysk: $\Delta F_{1234} = F_{1234}(A) - F_{1234}(A-1)$ przy zainwestowaniu na tych czterech rynkach dodatkowej jednostki (1 000000 $) maleje wraz ze wzrostem nasycenia rynków w środki inwestycyjne - A (jest to zjawisko naturalne i prawo ekonomiczne rynku).

Sens praktyczny tego wniosku ilustruje następujący wykres:



# 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).

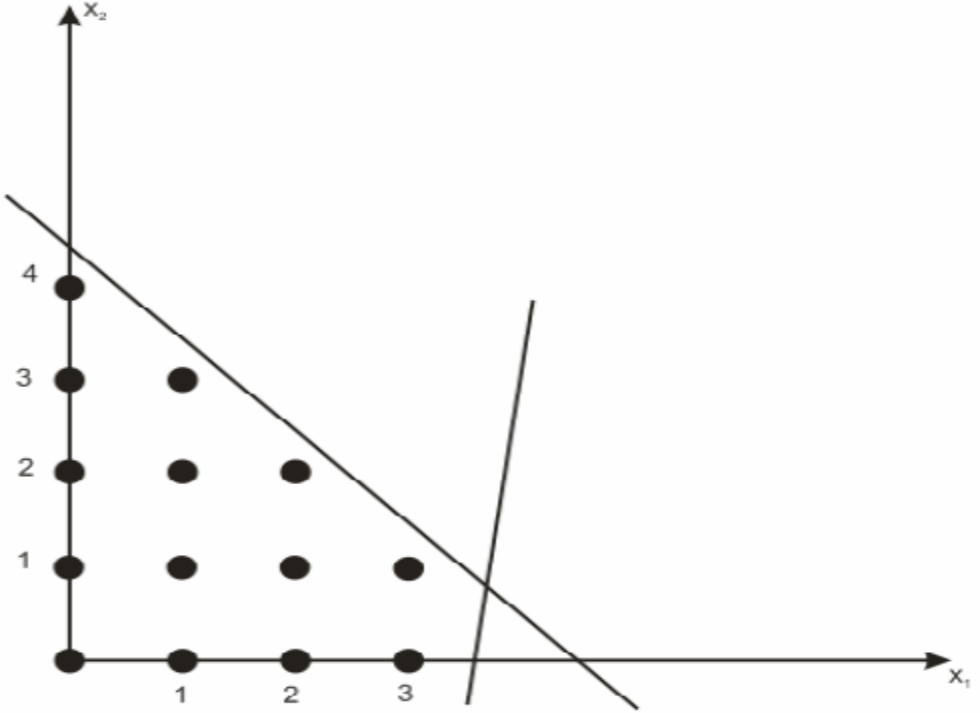
# ❑ Liniowe Modele Optymalizacji Dyskretnej – wprowadzenie w tematykę zagadnień

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 - $W_{i,j} = [w_{i,j}]_{i,j=1,2,...,n}$).

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 $X_{i,j} = [x_{i,j}]_{i,j} = 1,2,...,n$ - 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)**

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

(każde stanowisko jest obsadzone tylko przez 1 pracownika)

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

(każdy pracownik jest przydzielony tylko do 1 stanowiska)

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

# ❑ 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:

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

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

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;

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

element najmniejszy

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

-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 $[x^*_{i,j}]$ 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;

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

**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**

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

**cztery linie**
**zatem rozwiązanie optymalne**
**(krok 3)**

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

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

# 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: $\sum_{\langle i,j \rangle \in V} c_{ij} \cdot y_{ij} \rightarrow min$

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

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

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

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) $C = \left[ c_{ij} \right] i = 1,...,n; j = 1,...,n$. 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:**

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

**Zmienne decyzyjne:**

$$x_{ij} = \begin{cases} 1, & \text{gdy marszruta zawiera trasę } \langle i, j\rangle \\ 0, & \text{w przypadku przeciwnym} \end{cases}$$ - 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: $\sum_{\langle i,j\rangle \in V} c_{ij} x_{ij} \rightarrow \min$

przy warunkach ograniczających:

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

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

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 $(i=1,2,...,n;\ j=2,3,...,n;\ i \ne j)$

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.

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

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

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:
$F = c_{5,2} + c_{2,5} = 8 + 7 = 15$

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

# ❑ 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 $d = (\min\{c_{2,1}; c_{5,1}\} = 2, -, \min\{c_{2,3}; c_{5,3}\} = 1, \min\{c_{2,4}; c_{5,4}\} = 4, -)$.

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

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;