

**POLITECHNIKA RZESZOWSKA im. I. Łukasiewicza**

**Wydział Zarządzania**

**Zakład Metod Ilościowych**

---

# **OPTYMALIZACJA PROCESÓW LOGISTYCZNYCH**

**Prowadzący:**

**dr Tomasz Pisula**

**e-mail: [tpisula@prz.edu.pl](mailto:tpisula@prz.edu.pl)**

# □ Treści kształcenia:

## Wykład – 12

## Laboratorium - 9

- Model matematyczny problemu decyzyjnego. Zagadnienia programowania matematycznego w problemach logistycznych (W01, L01)
- Programowanie liniowe. Przedstawianie wybranych problemów decyzyjnych z logistyki w postaci zadań programowania liniowego. Dualizm w programowaniu liniowym. Interpretacja graficzna zadań programowania liniowego. Istota algorytmu simpleks (W02, L02)
- Zagadnienia i problemy transportowe. Otwarte oraz zamknięte zagadnienie transportowe. Algorytm transportowy. Zagadnienie transportowo–produkcyjne oraz transportowo-magazynowe. Zagadnienia transportowe z ograniczoną przepustowością tras. Minimalizacja pustych przebiegów. Modele zagadnień transportowych z kryterium czasu (W03-W04, L02-L03)
- Programowanie nieliniowe. Wybrane problemy optymalizacji nieliniowej w zastosowaniach logistycznych (W05-W06, L04)
- Optymalizacja dyskretna. Zagadnienie optymalnego przydziału. Problem komiwojażera. Zagadnienie rozwózki (W07-W08, L05)

## □ Treści kształcenia:

- **Optymalizacja przepływów w sieciach transportowych. Maksymalny przepływ w sieci transportowej. Wyznaczanie najkrótszej drogi w sieci transportowej. Zagadnienie przepływu o minimalnym koszcie (W09-W10, L06-L07)**
- **Elementy wielokryterialnego wspomaganie decyzji logistycznych - budowa rankingów obiektów w świetle ocen wielokryterialnych (W11)**
- **Praktyczne zaliczenie laboratorium (L08-L09)**
- **Pisemne zaliczenie wykładów (W12)**

## **□ Efekty kształcenia – wiedza - umiejętności:**

---

- 1. Zdobycie wiedzy o sposobach modelowania matematycznego problemów decyzyjnych w procesach logistycznych.**
- 2. Zdobycie wiedzy o różnych metodach poszukiwania rozwiązań optymalnych w logistycznych zagadnieniach decyzyjnych.**
- 3. Umiejętność budowania właściwych modeli matematycznych dla logistycznych problemów decyzyjnych.**
- 4. Umiejętność rozwiązywania logistycznych problemów decyzyjnych z wykorzystaniem właściwych technik i metod optymalizacji.**
- 5. Umiejętność praktycznego poszukiwania optymalnych rozwiązań optymalizacyjnych problemów decyzyjnych z wykorzystaniem odpowiednich narzędzi analitycznych (np. arkusza kalkulacyjnego Excel i modułu Solver).**

## □ Warunki zaliczenia przedmiotu:

### 1. Zaliczenie pisemne wykładów:

- sprawdzenie umiejętności poprawnego formułowania modeli matematycznych omawianych logistycznych problemów decyzyjnych).

### 2. Praktyczne zaliczenie laboratoriów:

- sprawdzenie praktycznych umiejętności modelowania i rozwiązywania wybranych logistycznych problemów decyzyjnych z wykorzystaniem arkusza kalkulacyjnego „Excel” oraz modułu „Solver”.

3. Ocena końcowa jest średnią ocen z zaliczenia pisemnego wykładów (z wagą 0,4) oraz zaliczenia praktycznego laboratoriów (z wagą 0,6).  
Obie składowe oceny muszą być pozytywne.

## □ Literatura Podstawowa:

---

1. Kauf S., Tłuczak A., **Optymalizacja decyzji logistycznych**, Wydawnictwo Difin, Warszawa 2016.
2. Bendkowski J., Kramarz M., Kramarz W., **Metody i techniki ilościowe w logistyce stosowanej. Wybrane zagadnienia**, Wydawnictwo Politechniki Śląskiej, Gliwice 2010.
3. Sikora W. (red.), **Badania operacyjne**, Polskie Wydawnictwo Ekonomiczne, Warszawa 2008.
4. Jędrzejczyk Z., Kukuła K. (red.), Skrzypek J., Walkosz A., **Badania operacyjne w przykładach i zadaniach**, Państwowe Wydawnictwo Naukowe, Warszawa 2011.
5. Szymczak M., **Decyzje logistyczne z Excelem**, Wydawnictwo Difin, Warszawa 2011.

## □ Literatura Uzupełniająca:

---

1. **Dąbek A.**, Ćwiczenia i zadania z transportu, spedycji i logistyki z rozwiązaniami, **Wydawnictwo Difin, Warszawa 2014.**
2. **Krawczyk S.**, Metody ilościowe w logistyce przedsiębiorstwa, **Wydawnictwo C. H. Beck, Warszawa 2001.**
3. **Trzaskalik T.**, Wprowadzenie do badań operacyjnych z komputerem, **Polskie Wydawnictwo Ekonomiczne, Warszawa 2008.**

## □ Wprowadzenie do programowania matematycznego - Pojęcie **Problemu Decyzyjnego**

**Sytuacje decyzyjne** – to sytuacje, w których zmuszeni jesteśmy podejmować różnego rodzaju **decyzje**, przy uwzględnieniu różnego rodzaju **uwarunkowań (wewnętrznych i zewnętrznych)**:

- Zarząd przedsiębiorstwa musi podjąć decyzję jakimi środkami transportu i w jakich ilościach dokonać przewozu towarów z magazynów do hurtowni przy optymalnym koszcie transportu;
- Dyrektor firmy spedycyjnej musi podjąć decyzję o ustaleniu najlepszego programu inwestycyjnego firmy - przynoszącego największe korzyści firmie;
- Dyrekcja przedsiębiorstwa komunikacji miejskiej musi podjąć decyzję o ustaleniu optymalnego harmonogramu pracy kierowców;

**Decydent** – ktoś, kto podejmuje decyzje (jednoosobowe lub kolegialne ciało zarządcze)

Warunki w jakich działa decydent nie pozwalają na ogół na podjęcie dowolnej decyzji, lecz ograniczają je tylko do pewnego podzbioru – **decyzji dopuszczalnych**.

## □ Wprowadzenie do programowania matematycznego - Pojęcie **Problemu Decyzyjnego**

Nie każda **decyzja dopuszczalna** przynosi równie duże **korzyści** (jest równie **użyteczna**, **wartościowa**) w świetle stawianych przez **zarządzającego** (decydenta) **celów**, które chce on **zrealizować**.

Biorąc pod uwagę stawiane **cele** – zwane **kryteriami wyboru** lub **kryteriami oceny** jedne z podjętych decyzji będą lepsze, zaś inne gorsze.

Zachodzi zatem problem wyboru decyzji najlepszej (najbardziej użytecznej - o największych korzyściach), którą będziemy nazywać – **decyzją optymalną**.

Przykład: Która decyzja jest optymalna ?

Decyzje	I	II	III
Nakłady inwestycyjne (mln zł)	10	20	30
Zyski (mln zł)	3	5	9

# □ Wprowadzenie do programowania matematycznego - Zadanie Decyzyjne

**Model matematyczny** problemu decyzyjnego (**Zadanie Decyzyjne**) – to opis określonego **problemu decyzyjnego** w języku **matematycznym** (taki głównie nas interesuje) za pomocą określonego **modelu matematycznego**.

**Zmienne Decyzyjne** – to wielkości (czynniki) od których zależy wynik (ocena) podjętej decyzji:

$$x = [x_1, \dots, x_n] \in R^n; x_i \geq 0,_{i=1, \dots, n}$$

**Funkcja Celu** (funkcja kryterium) – to funkcja, która zależy od zmiennych decyzyjnych i mierzy cel, który chce osiągnąć zarządzający (decydent):

$$f(x) = f(x_1, \dots, x_n) \in R$$

Oznaczmy przez „**D**” – zbiór **decyzji dopuszczalnych**. Jest on najczęściej określany przez układ równań i nierówności postaci:

$$x \in D \Leftrightarrow \begin{cases} g_i(x_1, \dots, x_n) \geq b_i & i = 1, \dots, k \\ g_i(x_1, \dots, x_n) \leq b_i & i = k + 1, \dots, l \\ g_i(x_1, \dots, x_n) = b_i & i = l + 1, \dots, m \end{cases}$$

**Analityczna postać funkcji  $g_i$  może być dowolna;**  
 **$b_i$  – współczynniki liczbowe;**

# □ Wprowadzenie do programowania matematycznego - Zadania Decyzyjne

Wybór **decyzji optymalnej** – oznaczanej przez  $x^* = [x_1^*, \dots, x_n^*]$  polega na określeniu takiej **decyzji dopuszczalnej**  $x^* \in D$ , dla której wartość funkcji celu osiąga wartość **najkorzystniejszą - optymalną** (w zależności od sytuacji **minimalną** lub **maksymalną**).

**Programowanie Matematyczne** – to rozwiązywanie zadań decyzyjnych.

**Zadanie decyzyjne** w formie **programowania matematycznego** możemy zatem zapisać następująco. Znajdź taką decyzję dopuszczalną  $x^*$ , że:

(1)

$$f(x^*) = f(x_1^*, \dots, x_n^*) = \max(\min) \{f(x_1, \dots, x_n)\} \quad \text{- funkcja celu}$$

Przy warunkach ograniczających:

$$\begin{cases} x \in D \\ x \geq 0 \end{cases} \Leftrightarrow \begin{cases} g_i(x_1, \dots, x_n) \geq b_i & i = 1, \dots, k \\ g_i(x_1, \dots, x_n) \leq b_i & i = k + 1, \dots, l \\ g_i(x_1, \dots, x_n) = b_i & i = l + 1, \dots, m \\ x_1 \geq 0, \dots, x_n \geq 0 \end{cases}$$



# **PROGRAMOWANIE LINIOWE**

# □ Programowanie liniowe - matematyczny model liniowych problemów decyzyjnych

**Programowanie liniowe (PL)** – to specyficzny wariant programowania matematycznego, w którym **funkcja celu** jest **postaci liniowej** oraz wszystkie **warunki ograniczające** są również **postaci liniowej**.

Ogólną postać zadania programowania liniowego (**ZPL**) można przedstawić:

$$f(x^*) = f(x_1^*, \dots, x_n^*) = \max(\min) \{c_1 \cdot x_1 + \dots + c_n \cdot x_n\} = \max(\min) \{(c, x)\}$$

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

$(c, x)$  - iloczyn skalarny wektorów: „c” oraz „x”

$$\left\{ \begin{array}{l} x \in D \\ x \geq 0 \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} \sum_{j=1}^n (a_{i,j} \cdot x_j) \geq b_i \quad i = 1, \dots, k \\ \sum_{j=1}^n (a_{i,j} \cdot x_j) \leq b_i \quad i = k + 1, \dots, l \\ \sum_{j=1}^n (a_{i,j} \cdot x_j) = b_i \quad i = l + 1, \dots, m \\ x_1 \geq 0, \dots, x_n \geq 0 \end{array} \right.$$

## □ Programowanie liniowe - przykłady liniowych zadań decyzyjnych

### 1. Optymalny wybór asortymentu produkcji.

Firma może produkować  $n$  – wyrobów. Do ich produkcji zużywane są różne środki produkcji, z których kilka –  $m$  są limitowane.

Dane są:

$a_{i,j}$  - normy zużycia  $i$  - tego środka produkcji ( $i=1,\dots,m$ ) na wytworzenie jednostki  $j$  - tego wyrobu ( $j=1,\dots,n$ );

$b_i$  - posiadany zasób  $i$  - tego środka produkcji;

$c_j$  - cena jednostkowa ze sprzedaży  $j$  - tego wyrobu;

Należy określić, które wyroby i w jakich ilościach mają być produkowane, aby nie przekraczając posiadanych zasobów środków produkcji zmaksymalizować przychód z ich sprzedaży.

Zmiennymi decyzyjnymi w zadaniu są wielkości produkcji poszczególnych wyrobów:  $x_j \geq 0$



# □ Programowanie liniowe - przykłady liniowych zadań decyzyjnych

## 2. Optymalny wybór wielkości sprzedaży.

Firma ForkLift Service (FLS) jest jednym z dystrybutorów wózków widłowych firmy Clark na terenie zachodniej Polski.

W 2002 roku największą sprzedaż firmy FLS stanowiły 2 typy wózków widłowych:

- Clark CDP20S CSP 20S, oznaczmy go jako 20S, oraz
- Clark CDP45H CGP 45H, oznaczany jako 45H.

Wózek 20S u producenta kosztuje 19.000 € , natomiast model 45H kosztuje 33.000 € .

Pod koniec roku 2002 specjaliści do spraw sprzedaży w firmie FLS postanowili, przy uwzględnieniu danych historycznych, oszacować wielkość sprzedaży wózków widłowych typu 20S i 45H na rok 2003.

Na podstawie wstępnej analizy stwierdzono, że na zakup obu typów wózków w 2003 roku będą mogli przeznaczyć:

- maksymalnie 2,4 mln €.

Przy czym zysk ze sprzedaży wózka:

- 20S wynosi 15%,
- wózka 45H natomiast 19%.

## □ Programowanie liniowe - przykłady liniowych zadań decyzyjnych

Doświadczony sprzedawca orzekł, że czas poświęcony przez pracowników FLS na sprzedaż jednego wózka jest zróżnicowany i zależy od typu wózka.

Stopień zaangażowania pracownika wynika przede wszystkim z: konieczności przeprowadzenia prezentacji wózka, przygotowania dokumentacji sprzedaży oraz przeprowadzenia podstawowego instruktażu w obecności klienta.

Sprzedawca ten oszacował, że czas pracy poświęcony na sprzedaż jednego wózka typu 20S wynosi ok. 6 godz., podczas gdy sprzedaż jednego wózka 45H zajmuje około 4 godz.

•Jednocześnie obliczono, że łączny czas pracy sprzedawców w ciągu roku, jaki mogą oni poświęcić na sprzedaż obu typów wózka wynosi 520 godz.

Na podstawie wstępnych rozmów z przedstawicielem firmy Clark (producentem wózków widłowych) ustalono również, że w roku 2003 firma FLS może spodziewać się:

- dostarczenia maksymalnie 100 wózków typu 20S i nie więcej niż 75 wózków typu 45H.
- z drugiej zaś strony dla zapewnienia ciągłości sprzedaży firma FLS jednorazowo musi zamawiać w firmie Clark minimum 10 wózków typu 20S oraz 5 wózków typu 45H.

Wspomóż decydenta firmy ForkLift Service w zaplanowaniu liczby wózków typu 20S i 45H, która zapewni jego firmie maksymalny roczny zysk przy zasobach dostępnych do realizacji tej sprzedaży (najlepsze wykorzystanie posiadanych zasobów)

# □ Programowanie liniowe - przykłady liniowych zadań decyzyjnych

## Konstrukcja modelu matematycznego:

- Zmienne decyzyjne w analizowanym problemie:

S – liczba zakupionych przez FLS wózków widłowych typu 20S

H – liczba zakupionych przez FLS wózków widłowych typu 45H

- funkcja celu (cel postawiony przez firmę FLS)

– Maksymalizacja zysku ze sprzedaży wózków widłowych typu 20S i 45H

– zysk całkowity:

$$Z = Z_S + Z_H$$

–  $Z_S$  - jednostkowy zysk ze sprzedaży wózków typu 20S

$$Z_S = 15\% \cdot 19.000 \text{ €} \cdot S = 2.850 S$$

–  $Z_H$  – jednostkowy zysk ze sprzedaży wózków typu 45H

$$Z_H = 19\% \cdot 33.000 \text{ €} \cdot H = 6.270 H$$

– ostateczne sformułowanie funkcji celu

$$\text{Max } Z(S, H) = 2.850 S + 6.270 H$$

# □ Programowanie liniowe - przykłady liniowych zadań decyzyjnych

---

## Identyfikacja ograniczeń:

- zasoby finansowe firmy FLS: max. 2.400.000 zł / rok
- dostępny fundusz czasu pracy poświęcany przez pracowników FLS na sprzedaż wózków 20S i 45H - max. 520 rbh / rok
  
- dostępność wózków u producenta
  - max 100 szt. / rok 20S
  - max 75 szt. / rok 45H
  
- rynkowy popyt na wózki widłowe
  - min 10 szt. 20S
  - min 5 szt. 45H

# □ Programowanie liniowe - przykłady liniowych zadań decyzyjnych

## Matematyczny zapis ograniczeń:

• zasoby finansowe firmy FLS ograniczają na przestrzeni roku możliwość zakupu wózków widłowych u producenta

$$19.000 S + 33.000 H \leq 2.400.000$$

• czas pracy ludzi zatrudnionych w FLS i zajmujących się sprzedażą wózków 20S i 45H jest ograniczony

$$6 S + 4 H \leq 520$$

• możliwości produkcyjne firmy Clark w zakresie dostarczenia firmie FLS wózków widłowych typu 20S i 45H

- dostępność wózków 20S

$$S \leq 100$$

- dostępność wózków 45H

$$H \leq 75$$

• minimalna liczba wózków w jednorazowym zamówieniu, zapewniająca ciągłość sprzedaży przy jednoczesnym zachowaniu satysfakcji klientów firmy FLS

- zapotrzebowanie firmy FLS na wózki typu 20S

$$S \geq 10$$

- zapotrzebowanie firmy FLS na wózki typu 45H

$$H \geq 5$$

• formalnie poszukiwane rozwiązanie (S, H) nie powinno przyjmować wartości ujemnych

$$S \geq 0$$

$$H \geq 0$$

# □ Programowanie liniowe - przykłady liniowych zadań decyzyjnych

Ostateczna postać modelu matematycznego problemu decyzyjnego sformułowanego w postaci zadania programowania liniowego:

•funkcja celu

$$\text{Max } Z(S, H) = 2.850 S + 6.270 H$$

przy ograniczeniach:

$$(1) 19 S + 33 H \leq 2.400$$

$$(2) 6 S + 4 H \leq 520$$

$$(3) S \leq 100$$

$$(4) H \leq 75$$

$$(5) S \geq 10$$

$$(6) H \geq 5$$

$$(7) S \geq 0$$

$$(8) H \geq 0$$

## □ Programowanie liniowe – dwie szczególne formy (postacie) zadań programowania liniowego

Bardzo ważną rolę przy formułowaniu zadań programowania liniowego odgrywają dwie jego szczególne postacie:

1. **Postać standardowa**: występuje wówczas, gdy **wszystkie nierówności** w warunkach ograniczających są postaci ( $\leq$ ) dla funkcji celu postaci (**maksimum**), zaś ( $\geq$ ) dla funkcji celu postaci (**minimum**) oraz wszystkie zmienne decyzyjne **są nieujemne**. Tego typu nierówności nazywają się **nierównościami typowymi** dla zadania na **maksimum i minimum** funkcji celu.

Każdą **postać standardową** (ZPL) można przedstawić za pomocą zapisu macierzowego następująco (wersja **z minimum** funkcji celu):

$$(3) \quad f(x) = (c, x) = \sum_{j=1}^n c_j \cdot x_j \rightarrow \min$$
$$\begin{cases} A \cdot x^T \geq b^T & A = [a_{i,j}]_{i=1,\dots,m; j=1,\dots,n} & \text{- macierz współczynników} \\ & & \text{w warunkach ograniczających} \\ x \geq 0 & b = [b_1, \dots, b_m] & \text{- wektor tzw. wyrazów wolnych} \end{cases}$$

**Uwaga:** Stosując odpowiednie **przekształcenie funkcji celu**, a także przekształcając **nierówności** w warunkach ograniczających **na typowe** możliwe jest sprowadzanie zadań PL do swej **postaci standardowej**.<sup>22</sup>

## □ Programowanie liniowe – dwie szczególne formy (postacie) zadań programowania liniowego

2. **Postać kanoniczna:** występuje wówczas, gdy wszystkie warunki ograniczające są podane w formie równań oraz wszystkie zmienne decyzyjne są nieujemne.

Postać kanoniczną zadania PL w zapisie macierzowym można przedstawić następująco: (4)

$$f(x) = (c, x) = \sum_{j=1}^n c_j \cdot x_j \rightarrow \max(\min)$$
$$\begin{cases} A \cdot x^T = b^T \\ x \geq 0 \end{cases}$$

**Uwaga:** Każde zadanie PL można sprowadzić do równoważnej postaci kanonicznej stosując następujące postępowanie:

Każdy warunek ograniczający w postaci nierówności daje się sprowadzić do równości wprowadzając dodatkowe zmienne

swobodne (pozorne). Np. warunek:  $2 \cdot x_1 + 3 \cdot x_2 \leq 100$  sprowadzamy do równości:  $2 \cdot x_1 + 3 \cdot x_2 + x_3 = 100$  - wprowadzając nową zmienną  $x_3 \geq 0$ .

Natomiast warunek postaci:  $1,5x_1 + x_2 \geq 50$  sprowadzamy do równości:  $1,5 \cdot x_1 + x_2 - x_4 = 50$  - wprowadzając nową zmienną  $x_4 \geq 0$ .

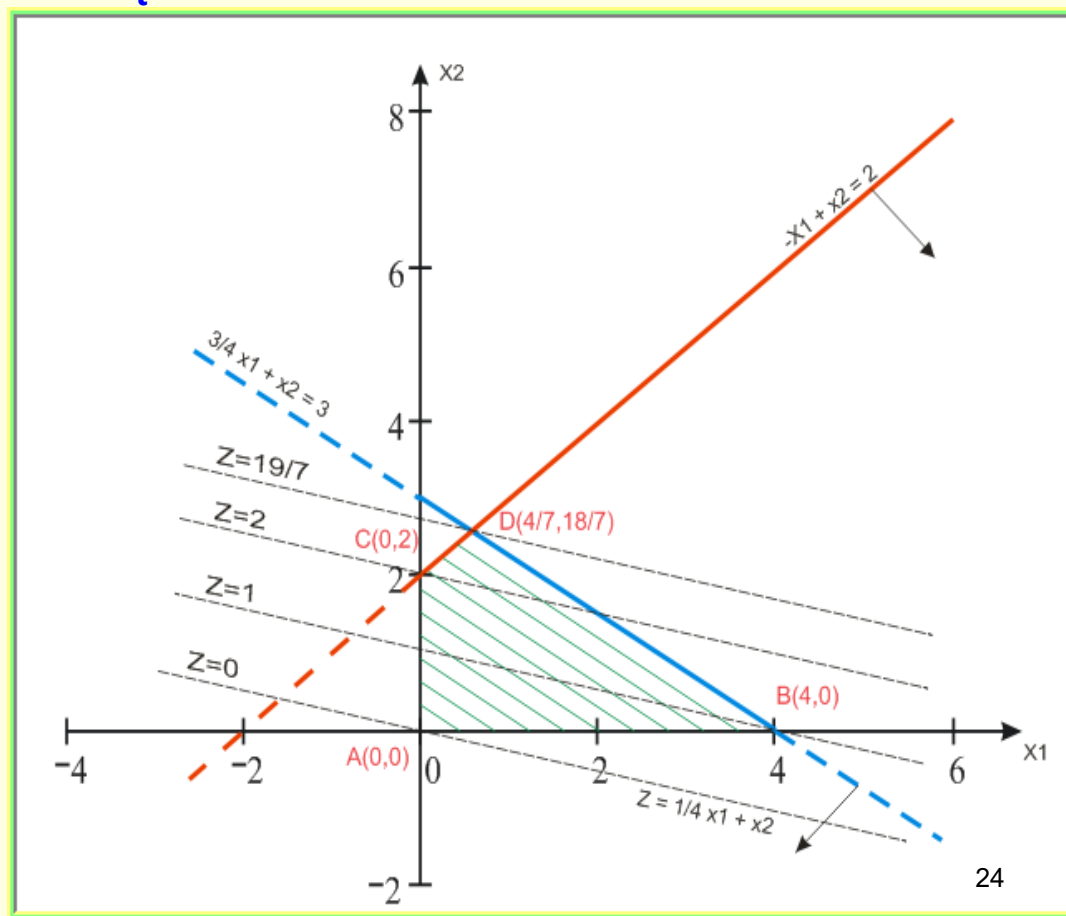
W funkcji celu pozorne zmienne decyzyjne:  $x_3, x_4$  pojawią się ze współczynnikami (wagami) równymi zero  $c_3 = 0$ ;  $c_4 = 0$ .

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

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

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

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

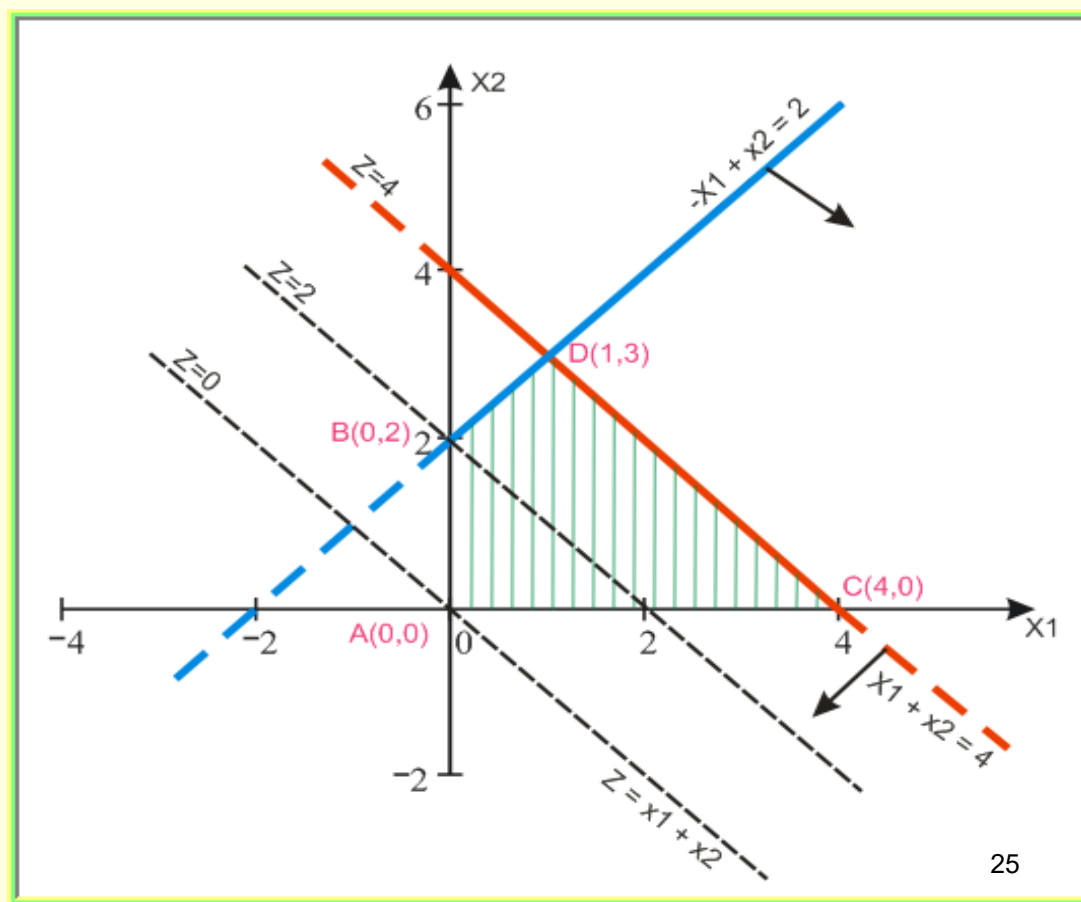


# □ Programowanie liniowe – graficzna interpretacja zadań programowania liniowego - metoda geometryczna poszukiwania rozwiązań

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

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

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

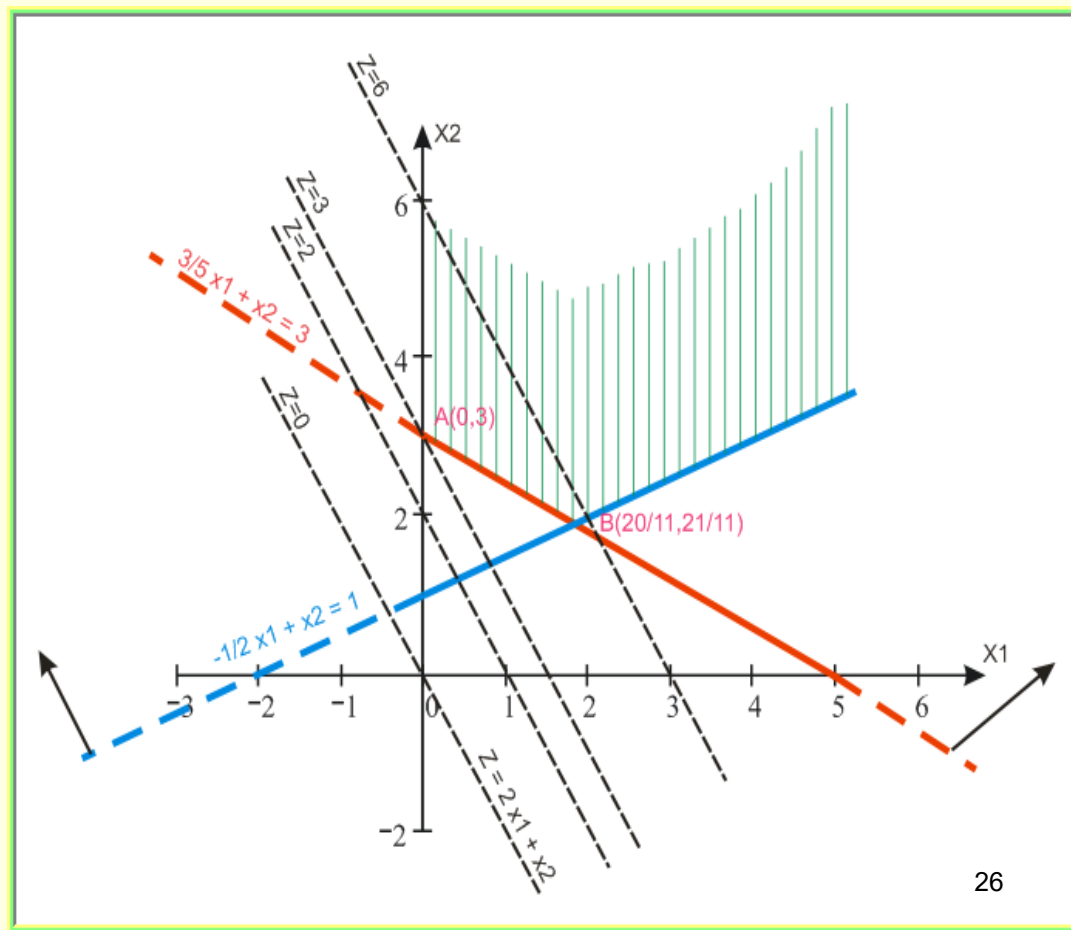


# □ Programowanie liniowe – graficzna interpretacja zadań programowania liniowego - metoda geometryczna poszukiwania rozwiązań

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

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

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



## □ Programowanie liniowe – dualność w programowaniu liniowym

Z **każdym** zadaniem **pierwotnym** (ZP PL) związane jest **inne zadanie**, które nazywać będziemy – zadaniem **dualnym** (ZD PL).

**Reguły** tworzenia **zadania dualnego** można scharakteryzować następująco:

- W zadaniu **dualnym** jest tyle **zmiennych** ile **warunków** ograniczających w zadaniu **pierwotnym**;
- W zadaniu **dualnym** jest tyle **warunków** ile **zmiennych** w zadaniu **pierwotnym**;
- **Współczynniki** w funkcji celu **zadania pierwotnego** są **wyrazami wolnymi** w warunkach ograniczających **zadania dualnego**;
- **Wyrazy wolne** zadania **pierwotnego** stają się **wagami** (współczynnikami) w funkcji celu **zadania dualnego**.
- **Macierz** współczynników **zadania dualnego** jest **transpozycją** zadania **pierwotnego**;
- Gdy zadanie **pierwotne** jest na **minimum**, to zadanie **dualne** jest **maksimum** i na odwrót.

## □ Programowanie liniowe – dualność w programowaniu liniowym

- Gdy w zadaniu **pierwotnym** „i-ta” zmienna jest **nieujemna** ( $\geq 0$ ), to w zadaniu **dualnym** odpowiadający jej **warunek** jest **typową** nierównością;
- Gdy w zadaniu **pierwotnym** na „i-tą” zmienną nie nałożono **żadnych ograniczeń** (może przyjmować dowolne wartości), to w zadaniu **dualnym** odpowiadający jej **warunek** jest **równością**;
- Gdy w zadaniu **pierwotnym** „i-ty” warunek jest **typową** nierównością, to odpowiadająca mu **zmienna** w zadaniu **dualnym** jest **nieujemna** ( $\geq 0$ );
- Gdy w zadaniu **pierwotnym** „i-ty” **warunek** jest **równością**, to odpowiadająca mu **zmienna** w zadaniu **dualnym** nie ma ograniczeń (może przyjmować **dowolne wartości**);
- Gdy w zadaniu **pierwotnym** „i-ty” **warunek** jest **nietypową** nierównością, to odpowiadająca mu **zmienna** w zadaniu **dualnym** jest **niedodatnia** ( $\leq 0$ );

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

## Zadanie pierwotne - ZP PL

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

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

## Zadanie dualne - ZD PL

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

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

# □ 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  $x_1^0, \dots, x_n^0$  - jest rozwiązaniem dopuszczalnym ZP PL, zaś  $y_1^0, \dots, y_m^0$  - jest rozwiązaniem dopuszczalnym ZD PL, to pomiędzy nimi zachodzi zależność: 
$$\sum_{j=1}^n c_j x_j^0 \leq \sum_{i=1}^m b_i y_i^0$$

**Twierdzenie 3** (o równości optymalnych rozwiązań ZP PL oraz ZD PL - John von Neumann):

Jeżeli istnieją takie dwa rozwiązania dopuszczalne:  $x_1^*, \dots, x_n^*$  dla ZP PL oraz  $y_1^*, \dots, y_m^*$  dla ZD PL, dla których zachodzi zależność:

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

to obydwa te rozwiązania są optymalne:  $\max(\min) f(x^*) = \min(\max) g(y^*)$ <sup>30</sup>

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

**Twierdzenie 4** (o równowadze obu rozwiązań):

Jeżeli  $x_1^0, \dots, x_n^0$  - jest rozwiązaniem dopuszczalnym ZP PL, zaś  $y_1^0, \dots, y_m^0$  - 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 - rozwiązania bazowe

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

**Postać standardowa ZPL:**

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

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

**Postać kanoniczna ZPL:**

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

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

$x_3, x_4$  –  
wprowadzone  
zmienne **swobodne**

Przykładowa **Baza** dla powyższej postaci kanonicznej:

**Kolumny niebazowe**

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

- 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**

$$A \cdot x^T = b^T \Leftrightarrow B \cdot x_B^T + N \cdot x_N^T = b^T \quad \mathbf{N} - \text{kolumny niebazowe}$$

# □ 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:**

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

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

$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow$   
 $A^1 \quad A^2 \quad A^3 \quad A^4$

**Wszystkich rozwiązań bazowych**

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

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

**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 <sub>1</sub> , x <sub>2</sub> }	Baza (B2) {x <sub>1</sub> , x <sub>3</sub> }	Baza (B3) {x <sub>1</sub> , x <sub>4</sub> }	Baza (B4) {x <sub>2</sub> , x <sub>3</sub> }	Baza (B5) {x <sub>2</sub> , x <sub>4</sub> }	Baza (B6) {x <sub>3</sub> , x <sub>4</sub> }
x <sub>1</sub>	6	8	7	0	0	0
x <sub>2</sub>	1	0	0	4	7	0
x <sub>3</sub>	0	-2	0	6	0	14
x <sub>4</sub>	0	0	1	0	-6	8
Funkcja celu f(x <sub>1</sub> , x <sub>2</sub> , x <sub>3</sub> , x <sub>4</sub> )	15 (Max)	Niedop.	14	12	Niedop.	0

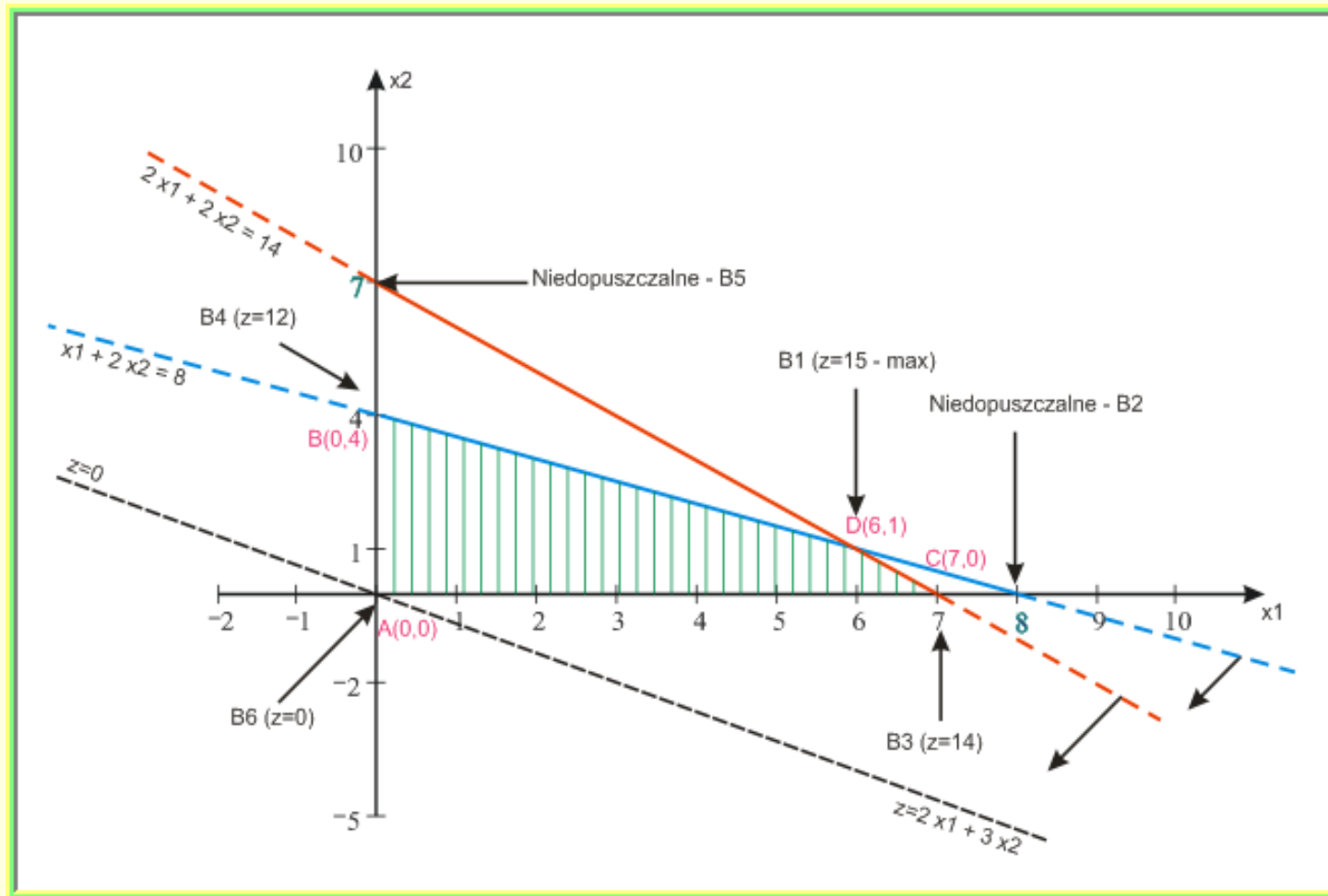
**Układ równań  
w postaci kanonicznej:**

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

**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



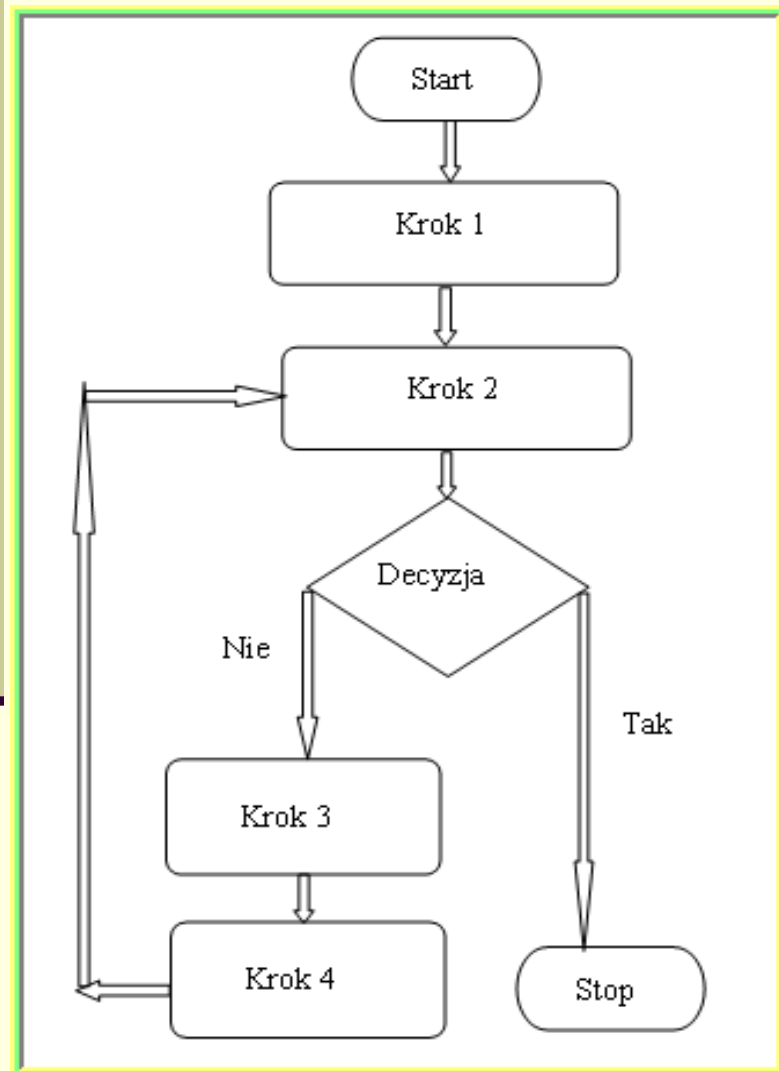
## □ Programowanie liniowe – metoda **Simpleks** – ujęcie algorytmiczne

---

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 **przeгляд 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**.