# Algorytm simplex i dualność

Łukasz Kowalik, Marcin Pilipczuk

Instytut Informatyki, Uniwersytet Warszawski

March 23, 2020

# Dualność programowania liniowego

# Certyfikat optymalności

- Powiedzmy, że mamy pewne wartościowanie zmiennych $\mathbf{x}$ dla pewnego PL: $\max \mathbf{c}^T\mathbf{x}, A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$.
- Łatwo sprawdzić czy $\mathbf{x}$ jest dopuszczalny.
- Chcemy przekonać kolegę, że $\mathbf{x}$ jest **optymalny**. Albo bliski optymalności.
- Czy można to zrobić w **prosty** sposób?

# Poszukiwanie górnego ograniczenia

Rozważmy następujący PL **w postaci standardowej**:

$$\begin{aligned} \max \quad & 3x_1 - x_2 + 2x_3 \\ & x_1 - x_2 + \tfrac{1}{2}x_3 \le 4 \\ & 4x_1 + 2x_2 + 3x_3 \le 20 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

$(x_1 = 2, x_2 = 0, x_3 = 4)$ jest rozw. dopuszczalnym o wartości f-cji celu 14.

Ponieważ $x_1, x_2, x_3 \ge 0$, to

- $3x_1 \le 4x_1$
- $-x_2 \le 2x_2$
- $2x_3 \le 3x_3$.

Razem: $3x_1 - x_2 + 2x_3 \le 4x_1 + 2x_2 + 3x_3 \le 20$

Czyli $\mathbf{c}^T\mathbf{x} \le 20$ dla dowolnego dopuszczalnego $\mathbf{x}$.

# Poszukiwanie górnego ograniczenia: próba 2

$$\begin{aligned} \max \quad & 3x_1 - x_2 + 2x_3 \\ & x_1 - x_2 + \tfrac{1}{2}x_3 \le 4 \\ & 4x_1 + 2x_2 + 3x_3 \le 20 \\ & x_1, x_2, x_3 \ge 0 \end{aligned}$$

$(x_1 = 2, x_2 = 0, x_3 = 4)$ jest rozw. dopuszczalnym o wartości f-cji celu 14.

Ponieważ $x_1, x_2, x_3 \ge 0$, to

$$\begin{aligned} 3x_1 - x_2 + 2x_3 \le x_1 - x_2 + \tfrac{1}{2}x_3 + \\ \tfrac{1}{2} \cdot (4x_1 + 2x_2 + 3x_3) \le 4 + \tfrac{1}{2} \cdot 20 = 14. \end{aligned}$$

Czyli $\mathbf{c}^T\mathbf{x} \le 14$ dla dowolnego dopuszczalnego $\mathbf{x}$. ($\mathbf{x}$ jest optymalny!)

# Poszukiwanie górnego ograniczenia: co się stało?

$$\max \quad 3x_1 - x_2 + 2x_3$$

$$\begin{aligned} x_1 - x_2 + \tfrac{1}{2}x_3 &\leq 4 \\ 4x_1 + 2x_2 + 3x_3 &\leq 20 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

- Fartownie, pokazaliśmy:

$$3x_1 - x_2 + 2x_3 \leq x_1 - x_2 + \tfrac{1}{2}x_3 + \tfrac{1}{2} \cdot (4x_1 + 2x_2 + 3x_3) \leq 4 + \tfrac{1}{2} \cdot 20 = 14.$$

- Co tu się działo? Szukaliśmy kombinacji liniowej nierówności $y_1(x_1 - x_2 + \frac{1}{2}x_3 \leq 4) + y_2(4x_1 + 2x_2 + 3x_3 \leq 20)$ t.ż.
  - $y_1, y_2 \geq 0$,
  - $3 \leq y_1 \cdot 1 + y_2 \cdot 4$,
  - $-1 \leq y_1 \cdot (-1) + y_2 \cdot 2$,
  - $2 \leq y_1 \cdot \frac{1}{2} + y_2 \cdot 3$,
  - $y_1 \cdot 4 + y_2 \cdot 20$ możliwie najmniejsze.
- To jest program liniowy!

# Program dualny

program prymalny

$$\max \quad 3x_1 - x_2 + 2x_3$$

$$\begin{aligned} & x_1 - x_2 + \tfrac{1}{2}x_3 \leq 4 \\ & 4x_1 + 2x_2 + 3x_3 \leq 20 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

$\rightarrow$

program dualny

$$\min \quad 4y_1 + 20y_2$$

$$\begin{aligned} & y_1 + 4y_2 \geq 3 \\ & -y_1 + 2y_2 \geq -1 \\ & \tfrac{1}{2}y_1 + 3y_2 \geq 2 \\ & y_1, y_2 \geq 0. \end{aligned}$$

Dla dowolnego programu w postaci standardowej:

P1:

$$\begin{aligned} \max \quad & \mathbf{c}^T\mathbf{x} \\ & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

$\rightarrow$

P2:

$$\begin{aligned} \min \quad & \mathbf{b}^T\mathbf{y} \\ & \mathbf{A}^T\mathbf{y} \geq \mathbf{c} \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

## Obserwacja

Rozumując analogicznie, wychodząc od P2 dostaniemy P1.
P1 jest dualny do P2.

# Słaba dualność

## Twierdzenie o słabej dualności

Niech $\mathbf{x}$ i $\mathbf{y}$ będą dowolnymi rozwiązaniami dopuszczalnymi odpowiednio programów $\max \mathbf{c}^T\mathbf{x}, A\mathbf{x} \leq b, \mathbf{x} \geq 0$ oraz $\min \mathbf{b}^T\mathbf{y}, A^T\mathbf{y} \geq c, \mathbf{y} \geq 0$. Wówczas $\mathbf{c}^T\mathbf{x} \leq \mathbf{b}^T\mathbf{y}$.

## Proof.

$$c_j x_j \leq \left( \sum_{i=1}^{m} a_{ij} y_i \right) x_j \quad \forall j \quad \text{oraz} \quad \left( \sum_{j=1}^{n} a_{ij} x_j \right) y_i \leq b_i y_i \quad \forall i$$

Stąd,

$$\mathbf{c}^T\mathbf{x} = \sum_{j=1}^{n} c_j x_j \leq \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} y_i x_j = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_j y_i \leq \sum_{i=1}^{m} b_i y_i = \mathbf{b}^T\mathbf{y}.$$

□

# Komplementarne warunki swobody

## Wniosek z dowodu

Niech $\mathbf{x}$ i $\mathbf{y}$ będą rozwiązaniami dopuszczalnymi odpowiednio dla zadania prymalnego i dualnego w postaci standardowej. Rozwiązania $\mathbf{x}$ i $\mathbf{y}$ są oba optymalne wtedy i tylko wtedy gdy

(i) **prymalne komplementarne warunki swobody**
dla każdego $j = 1, \ldots, n$ mamy $x_j = 0$ lub $\sum_{i=1}^{m} a_{ij} y_i = c_j$.
($x_j = 0$ lub $j$-ta nierówność programu dualnego jest spełniona z równością.)

(ii) **dualne komplementarne warunki swobody**
dla każdego $i = 1, \ldots, m$ mamy $y_i = 0$ lub $\sum_{j=1}^{n} a_{ij} x_j = b_i$.
($y_i = 0$ lub $i$-ta nierówność programu prymalnego jest spełniona z równością.)

## Gdy program nie jest standardowy: równości

Tak jak poprzednio, chcemy znaleźć jak najlepsze górne ograniczenie.

program prymalny

$$\max \quad 3x_1 - x_2 + 2x_3$$

$$\begin{aligned} x_1 - x_2 + \tfrac{1}{2}x_3 &\le 4 \\ 4x_1 + 2x_2 + 3x_3 &= 20 \\ x_1, x_2, x_3 &\ge 0 \end{aligned}$$

$\rightarrow$

program dualny

$$\min \quad 4y_1 + 20y_2$$

$$\begin{aligned} y_1 + 4y_2 &\ge 3 \\ -y_1 + 2y_2 &\ge -1 \\ \tfrac{1}{2}y_1 + 3y_2 &\ge 2 \\ y_1 &\ge 0. \end{aligned}$$

Teraz $y_2$ nie musi być nieujemny!

# Gdy program nie jest standardowy: zmienne bez znaku

Tak jak poprzednio, chcemy znaleźć jak najlepsze górne ograniczenie.

| program prymalny | | program dualny |
|---|---|---|
| $\max \quad 3x_1 - x_2 + 2x_3$ <br> $x_1 - x_2 + \frac{1}{2}x_3 \leq 4$ <br> $4x_1 + 2x_2 + 3x_3 = 20$ <br> $x_1, x_3 \geq 0$ | $\rightarrow$ | $\min \quad 4y_1 + 20y_2$ <br> $y_1 + 4y_2 \geq 3$ <br> $-y_1 + 2y_2 = -1$ <br> $\frac{1}{2}y_1 + 3y_2 \geq 2$ <br> $y_1 \geq 0.$ |

- Chcemy $-x_2 \leq (-y_1 + 2y_2)x_2$.
- Nierówność $-1 \leq -y_1 + 2y_2$ już nam tego nie implikuje.
- Zadowolimy się równością $-1 = -y_1 + 2y_2$.

# Ogólne reguły tworzenia programów dualnych

| | PRYMALNY | $\leftrightarrow$ | | DUALNY |
|---|---|---|---|---|
| f. celu | $\max \mathbf{c}^T \mathbf{x}$ | $\leftrightarrow$ | f.celu | $\min \mathbf{b}^T \mathbf{y}$ |
| macierz | $A$ | $\leftrightarrow$ | macierz | $A^T$ |
| $i$-ty warunek | $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$ | $\leftrightarrow$ | $i$-ta zmienna | $y_i \geq 0$ |
| $i$-ty warunek | $\sum_{j=1}^{n} a_{ij} x_j = b_i$ | $\leftrightarrow$ | $i$-ta zmienna | $y_i$ nieograniczone. |
| $j$-ta zmienna | $x_j \geq 0$ | $\leftrightarrow$ | $j$-ty warunek | $\sum_{i=1}^{m} a_{ij} y_i \geq c_j$ |
| $j$-ta zmienna | $x_j$ nieograniczone | $\leftrightarrow$ | $j$-ty warunek | $\sum_{i=1}^{m} a_{ij} y_i = c_j$ |

(Słaba dualność dalej zachodzi!)

# Silna dualność

**Wniosek ze słabej dualności**

Niech $\mathbf{x}^*$ będzie rozwiązaniem optymalnym programu $\min \mathbf{c}^T\mathbf{x}, \ldots$
Niech $\mathbf{y}^*$ będzie rozwiązaniem optymalnym dualnego $\max \mathbf{b}^T\mathbf{y}, \ldots$
Wtedy, $\mathbf{c}^T\mathbf{x}^* \geq \mathbf{b}^T\mathbf{y}^*$.

# Silna dualność

**Wniosek ze słabej dualności**

Niech $\mathbf{x}^*$ będzie rozwiązaniem optymalnym programu $\min \mathbf{c}^T\mathbf{x}, \ldots$
Niech $\mathbf{y}^*$ będzie rozwiązaniem optymalnym dualnego $\max \mathbf{b}^T\mathbf{y}, \ldots$
Wtedy, $\mathbf{c}^T\mathbf{x}^* \geq \mathbf{b}^T\mathbf{y}^*$.

Czyli, gdy $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$, to $\mathbf{y}^*$ jest certyfikatem optymalności $\mathbf{x}^*$.

# Silna dualność

## Wniosek ze słabej dualności

Niech $\mathbf{x}^*$ będzie rozwiązaniem optymalnym programu $\min \mathbf{c}^T\mathbf{x}, \ldots$
Niech $\mathbf{y}^*$ będzie rozwiązaniem optymalnym dualnego $\max \mathbf{b}^T\mathbf{y}, \ldots$
Wtedy, $\mathbf{c}^T\mathbf{x}^* \geq \mathbf{b}^T\mathbf{y}^*$.

Czyli, gdy $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$, to $\mathbf{y}^*$ jest certyfikatem optymalności $\mathbf{x}^*$.

Okazuje się, że tak jest zawsze!

## Twierdzenie o silnej dualności

Niech $\mathbf{x}^*$ będzie rozwiązaniem optymalnym programu $\min \mathbf{c}^T\mathbf{x}, \ldots$
Niech $\mathbf{y}^*$ będzie rozwiązaniem optymalnym dualnego $\max \mathbf{b}^T\mathbf{y}, \ldots$
Wtedy, $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$.

# Lemat Farkasa

**Theorem (lemat Farkasa)**

*Zachodzi dokładnie jedno z poniższych:*

1. *istnieje wektor* $\mathbf{x}$ *taki, że* $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$,
2. *istnieje wektor* $\mathbf{y}$ *taki, że* $A^T\mathbf{y} \geq \mathbf{0}$ *oraz* $\mathbf{b}^T\mathbf{y} < 0$.

Dowód: $\neg(1) \leftrightarrow (2)$

- $(2) \rightarrow \neg(1)$ łatwo
- $\neg(1) \rightarrow (2)$ z twierdzenia o oddzielaniu punktu od zbioru wypukłego

# Silna dualność jako wniosek z lematu Farkasa

**Lemat Farkasa**

Zachodzi dokładnie jedno z:

1. $\exists \mathbf{x}$ t.ż. $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$,
2. $\exists \mathbf{z}$ t.ż. $A^T\mathbf{z} \geq \mathbf{0}$, $\mathbf{b}^T\mathbf{z} < 0$.

$\rightarrow$

**Silna dualność**

Jeśli $\mathbf{x}^*$ i $\mathbf{y}^*$ to rozw. optymalne $\min \mathbf{c}^T\mathbf{x}, A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ oraz $\max \mathbf{b}^T\mathbf{y}, A^T\mathbf{y} \leq \mathbf{c}$, to $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$.

**Dowód:** $\mathbf{c}^T\mathbf{x}^* \geq \mathbf{b}^T\mathbf{y}^*$ ze słabej dualności. Oznaczmy $\gamma := \mathbf{b}^T\mathbf{y}^*$
Załóżmy że nie istnieje $\mathbf{x} \geq \mathbf{0}$ t.ż. $A\mathbf{x} = \mathbf{b}$ i $\mathbf{c}^T\mathbf{x} = \gamma$.
Czyli nie istnieje $\mathbf{x} \geq \mathbf{0}$ t.ż. $\begin{bmatrix} A \\ c \end{bmatrix} \mathbf{x} = \begin{bmatrix} b \\ \gamma \end{bmatrix}$.
Z lematu Farkasa istnieje $\mathbf{z}$ tż. $[A^T; c^T]\mathbf{z} \geq \mathbf{0}$ i $[b; \gamma]^T\mathbf{z} < 0$.
Czyli istnieje wektor $\mathbf{y}$ i $\lambda \in \mathbb{R}$ tż. $A^T\mathbf{y} \geq -\lambda c$ i $\mathbf{b}^T\mathbf{y} < -\gamma\lambda$.

1. $\lambda = 0$: $A^T\mathbf{y} \geq \mathbf{0}$ i $\mathbf{b}^T\mathbf{y} < 0$. Farkas $\rightarrow$ program prymalny sprzeczny.
2. $\lambda \neq 0$: $A^T(\frac{1}{-\lambda}\mathbf{y}) \leq c$ i $\mathbf{b}^T(\frac{1}{-\lambda}\mathbf{y}) > \gamma$. Czyli $\frac{1}{-\lambda}\mathbf{y}$ lepszy niż optimum.

# Silna dualność dla postaci standardowej

**Silna dualność**

Jeśli $\mathbf{x}^*$ i $\mathbf{y}^*$ to rozw. optymalne $\min \mathbf{c}^T\mathbf{x}, A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ oraz $\max \mathbf{b}^T\mathbf{y}, A^T\mathbf{y} \leq \mathbf{c}$, to $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$.

$\rightarrow$

**Silna dualność**

Jeśli $\mathbf{x}^*$ i $\mathbf{y}^*$ to rozw. optymalne $\min \mathbf{c}^T\mathbf{x}, A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ oraz $\max \mathbf{b}^T\mathbf{y}, A^T\mathbf{y} \leq \mathbf{c}, \mathbf{y} \geq \mathbf{0}$, to $\mathbf{c}^T\mathbf{x}^* = \mathbf{b}^T\mathbf{y}^*$.

**Dowód:**

$$\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \\ & \downarrow \end{array} \qquad \begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} \\ & \mathbf{A}^T\mathbf{y} \leq \mathbf{c} \\ & \mathbf{y} \geq \mathbf{0} \\ & \uparrow \end{array}$$

$$\begin{array}{ll} \min & \mathbf{c}^T\mathbf{x} \\ & [\mathbf{A}| - I]\left[\begin{array}{c} \mathbf{x} \\ \mathbf{s} \end{array}\right] = \mathbf{b} \\ & \mathbf{x}, \mathbf{s} \geq \mathbf{0} \end{array} \rightarrow \begin{array}{ll} \max & \mathbf{b}^T\mathbf{y} \\ & \left[\begin{array}{c} \mathbf{A}^T \\ -I \end{array}\right]\mathbf{y} \leq \left[\begin{array}{c} \mathbf{c} \\ \mathbf{0} \end{array}\right] \end{array}$$

# Pełne twierdznie o silnej dualności

**Twierdzenie o silnej dualności**

Są cztery możliwości:

1. prymalny i dualny są sprzeczne,
2. prymalny jest sprzeczny i dualny jest nieograniczony,
3. prymalny jest nieograniczony i dualny jest sprzeczny,
4. prymalny i dualny są ograniczone i wartości ich rozwiązań optymalnych są równe.

(dowód pomijamy)

# Efektywne algorytmy programowania liniowego

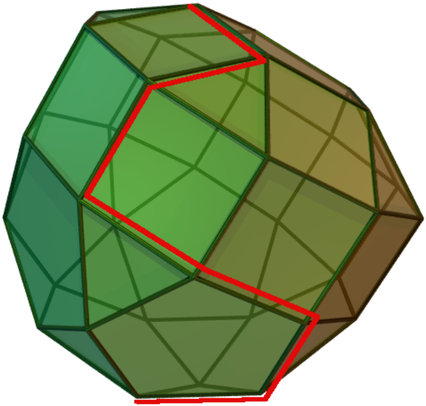
- Algorytm Simplex (George Dantzig 1947).
  - Złożoność pesymistyczna wykładnicza.
  - Istnieją implementacje o złożoności oczekiwanej $2^{O(\sqrt{n})}\mathrm{poly}(n)$.
  - Bardzo dobrze zachowuje się na „rzeczywistych” danych.
  - Wielomianowa ‘smoothed complexity’.
  - Powszechnie używany w praktyce
- Algorytm Elipsoidalny (Leonid Khachiyan 1979).
  - czas $O(n^4 L)$,
    gdzie $L =$ długość zapisu binarnego danych $(\mathbf{A}, \mathbf{b}, \mathbf{c})$
  - zaimplementowany ale niepraktyczny
- Algorytm punktu wewnętrznego (Narendra Karmarkar 1984).
  - $O(n^{3.5} L)$
  - w praktyce porównywalny z simplexem, ale rzadko stosowany.

# Algorytm simplex

# Algorytm simplex: idea

Podejście przeszukiwania lokalnego:

- wybierz dowolny wierzchołek
- dopóki jeden z sąsiednich wierzchołków jest lepszy (lub nie gorszy) przenieś się tam.



# Algorytm simplex: przejście do postaci dopełnieniowej

$$
\begin{array}{rrcll}
 & & & \max & z \\
\max & 3x_1 + x_2 + 2x_3 & & z & = 3x_1 + x_2 + 2x_3 \\
 & x_1 + x_2 + 3x_3 \leq 30 & \to & x_4 & = 30 - x_1 - x_2 - 3x_3 \\
 & 2x_1 + 2x_2 + 5x_3 \leq 24 & & x_5 & = 24 - 2x_1 - 2x_2 - 5x_3 \\
 & 4x_1 + x_2 + 2x_3 \leq 36 & & x_6 & = 36 - 4x_1 - x_2 - 2x_3 \\
 & x_1, x_2, x_3 \geq 0. & & & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.
\end{array}
$$

początkowe brd

$(0, 0, 0) \quad \to \quad (0, 0, 0, 30, 24, 36)$

# Algorytm simplex: niezmiennik

$$
\begin{aligned}
\max \quad & z \\
z \quad & = 3x_1 + x_2 + 2x_3 \\
x_4 \quad & = 30 - x_1 - x_2 - 3x_3 \\
x_5 \quad & = 24 - 2x_1 - 2x_2 - 5x_3 \\
x_6 \quad & = 36 - 4x_1 - x_2 - 2x_3 \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.
\end{aligned}
$$

$B = \{4, 5, 6\}$
$N = \{1, 2, 3\}$

## Niezmiennik

Zmienne dzielą się na bazowe $B = \{B_1, \ldots, B_m\}$ i niebazowe $N = \{N_1, \ldots, N_n\}$
Program zawiera:

- równanie postaci $z = v + \sum_{j=1}^{n} c_j x_{N_j}$;
- $\forall i = 1, \ldots, m$ równanie postaci $x_{B_i} = b_i + \sum_{j=1}^{n} a_{i,j} x_{N_j}$, gdzie $b_i \geq 0$;
- $\forall i = 1, \ldots, n + m$ nierówność $x_i \geq 0$,

gdzie $v$, $c_j$, $b_j$, $a_{i,j}$ są stałymi.

# Algorytm simplex: niezmiennik

$$
\begin{aligned}
\max \quad & z \\
z & = 3x_1 + x_2 + 2x_3 \\
x_4 & = 30 - x_1 - x_2 - 3x_3 \\
x_5 & = 24 - 2x_1 - 2x_2 - 5x_3 \\
x_6 & = 36 - 4x_1 - x_2 - 2x_3 \\
& x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.
\end{aligned}
$$

$B = \{4, 5, 6\}$
$N = \{1, 2, 3\}$

**Fakt**

Jeśli spełniony jest niezmiennik, to rozwiązanie $(x_1, \ldots, x_{n+m})$ postaci

$$x_i = \begin{cases} 0 & \text{gdy } i \in N \\ b_j & \text{gdy } i = B_j \text{ dla pewnego } j = 1, \ldots, n \end{cases}$$

jest bazowym rozwiązaniem dopuszczalnym o wartości funkcji celu $v$.

Tu: $\mathbf{x} = (0, 0, 0, 30, 24, 36)$.

# Algorytm simplex: powiększanie wartości funckji celu

$$\begin{aligned} \max \quad & z \\ z &= 3x_1 + x_2 + 2x_3 \\ x_4 &= 30 - x_1 - x_2 - 3x_3 \\ x_5 &= 24 - 2x_1 - 2x_2 - 5x_3 \\ x_6 &= 36 - 4x_1 - x_2 - 2x_3 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0. \end{aligned}$$

$$B = \{4, 5, 6\}$$
$$N = \{1, 2, 3\}$$

- powiększając $x_1$ powiększamy $z$
- jak bardzo możemy powiększyć $x_1$? $x_1 := \min\{\frac{30}{1}, \frac{24}{2}, \frac{36}{4}\} = \frac{36}{4} = 9.$
- wówczas jedna ze zmiennych bazowych (tu: $x_6$) przyjmuje wartość 0.
- $B := B - \{6\} \cup \{1\}$, $N := N - \{1\} \cup \{6\}$.
- zmieniamy bazę: $x_1$ jest zmienną **wchodzącą**, $x_6$ **wychodzącą**.

# Wymiana bazy (pivot)

Operacja wymiany bazy (ang. **pivot**) przebiega w dwóch krokach:

1. Rozwiąż równanie zawierające zmienną wychodzącą ze względu na zmienną wchodzącą. W tym przypadku otrzymujemy:

$$x_1 = 9 - \frac{1}{4}x_2 - \frac{1}{2}x_3 - \frac{1}{4}x_6.$$

2. wstaw wynik zamiast $x_1$ z prawej strony wszystkich równań (czyli uaktualnij współczynniki przy zmiennych niebazowych i wyrazy wolne). W tym przypadku otrzymujemy:

$$\begin{array}{rcrcrcrcr} z & = & 27 & + & \frac{1}{4}x_2 & + & \frac{1}{2}x_3 & - & \frac{3}{4}x_6 \\ x_1 & = & 9 & - & \frac{1}{4}x_2 & - & \frac{1}{2}x_3 & - & \frac{1}{4}x_6 \\ x_4 & = & 21 & - & \frac{3}{4}x_2 & - & \frac{5}{2}x_3 & + & \frac{1}{4}x_6 \\ x_5 & = & 6 & - & \frac{3}{2}x_2 & - & 4x_3 & + & \frac{1}{2}x_6. \end{array}$$

$B = \{1,4,5\}$ $\quad$ $N = \{2,3,6\}$ $\quad$ $\mathbf{x} = (9,0,0,21,6,0)$ $\quad$ $z = 27.$

# Druga wymiana bazy

$$\begin{array}{rcrcrcrcr} z & = & 27 & + & \frac{1}{4}x_2 & + & \frac{1}{2}x_3 & - & \frac{3}{4}x_6 \\ x_1 & = & 9 & - & \frac{1}{4}x_2 & - & \frac{1}{2}x_3 & - & \frac{1}{4}x_6 \\ x_4 & = & 21 & - & \frac{3}{4}x_2 & - & \frac{5}{2}x_3 & + & \frac{1}{4}x_6 \\ x_5 & = & 6 & - & \frac{3}{2}x_2 & - & 4x_3 & + & \frac{1}{2}x_6. \end{array}$$

- Do bazy mogą wejść $x_2$ lub $x_3$.
  (współczynnik w f-cji celu musi być dodatni!)
- Wybierzmy $x_3$. Wtedy $x_3 := \min\left\{\frac{9}{\frac{1}{2}}, \frac{21}{\frac{5}{2}}, \frac{6}{4}\right\} = \frac{6}{4} = \frac{3}{2}$.
- $x_5$ wychodzi z bazy.

$$\begin{array}{rcrcrcrcr} z & = & \frac{111}{4} & + & \frac{1}{16}x_2 & - & \frac{1}{8}x_5 & - & \frac{11}{16}x_6 \\ x_1 & = & \frac{33}{4} & - & \frac{1}{16}x_2 & + & \frac{1}{8}x_5 & - & \frac{5}{16}x_6 \\ x_3 & = & \frac{3}{2} & - & \frac{3}{8}x_2 & - & \frac{1}{4}x_5 & + & \frac{1}{8}x_6 \\ x_4 & = & \frac{69}{4} & + & \frac{3}{16}x_2 & + & \frac{5}{8}x_5 & - & \frac{1}{16}x_6. \end{array}$$

$B = \{1, 3, 4\}$ $\quad N = \{2, 5, 6\}$ $\quad \mathbf{x} = (\frac{33}{4}, 0, \frac{3}{2}, \frac{69}{4}, 0, 0)$ $\quad z = \frac{111}{4}$.

# Trzecia wymiana bazy

$$
\begin{array}{rcrcrcrcr}
z & = & \frac{111}{4} & + & \frac{1}{16}x_2 & - & \frac{1}{8}x_5 & - & \frac{11}{16}x_6 \\
x_1 & = & \frac{33}{4} & - & \frac{1}{16}x_2 & + & \frac{1}{8}x_5 & - & \frac{5}{16}x_6 \\
x_3 & = & \frac{3}{2} & - & \frac{3}{8}x_2 & - & \frac{1}{4}x_5 & + & \frac{1}{8}x_6 \\
x_4 & = & \frac{69}{4} & + & \frac{3}{16}x_2 & + & \frac{5}{8}x_5 & - & \frac{1}{16}x_6.
\end{array}
$$

- Do bazy może wejść tylko $x_2$.
- Uwaga: w ostatnim równaniu wsp. przed $x_2$ dodatni: $x_4$ nie będze wychodzący.

# Trzecia wymiana bazy

$$
\begin{array}{rclclclcl}
z & = & \frac{111}{4} & + & \frac{1}{16}x_2 & - & \frac{1}{8}x_5 & - & \frac{11}{16}x_6 \\
x_1 & = & \frac{33}{4} & - & \frac{1}{16}x_2 & + & \frac{1}{8}x_5 & - & \frac{5}{16}x_6 \\
x_3 & = & \frac{3}{2} & - & \frac{3}{8}x_2 & - & \frac{1}{4}x_5 & + & \frac{1}{8}x_6 \\
x_4 & = & \frac{69}{4} & + & \frac{3}{16}x_2 & + & \frac{5}{8}x_5 & - & \frac{1}{16}x_6.
\end{array}
$$

- Do bazy może wejść tylko $x_2$.
- Uwaga: w ostatnim równaniu wsp. przed $x_2$ dodatni: $x_4$ nie będze wychodzący.

**Co by było gdyby...**

... gdyby istniała zmienna z dodatnim współczynnikiem w funkcji celu i nieujemnymi współczynnikami w pozostałych równaniach?
Wówczas algorytm simplex zwraca komunikat “PROGRAM NIEOGRANICZONY” i kończy działanie.

# Trzecia wymiana bazy

$$\begin{array}{rcrcrcrcr}
z & = & \frac{111}{4} & + & \frac{1}{16}x_2 & - & \frac{1}{8}x_5 & - & \frac{11}{16}x_6 \\
x_1 & = & \frac{33}{4} & - & \frac{1}{16}x_2 & + & \frac{1}{8}x_5 & - & \frac{5}{16}x_6 \\
x_3 & = & \frac{3}{2} & - & \frac{3}{8}x_2 & - & \frac{1}{4}x_5 & + & \frac{1}{8}x_6 \\
x_4 & = & \frac{69}{4} & + & \frac{3}{16}x_2 & + & \frac{5}{8}x_5 & - & \frac{1}{16}x_6.
\end{array}$$

- Do bazy może wejść tylko $x_2$.
- Uwaga: w ostatnim równaniu wsp. przed $x_2$ dodatni: $x_4$ nie będze wychodzący.
- $x_2 := \min\left\{\frac{\frac{33}{4}}{\frac{1}{16}}, \frac{\frac{3}{2}}{\frac{3}{8}}\right\} = \frac{\frac{3}{2}}{\frac{3}{8}} = 4.$
- $x_3$ wychodzi z bazy.

$$\begin{array}{rcrcrcrcr}
z & = & 28 & - & \frac{1}{6}x_3 & - & \frac{1}{6}x_5 & - & \frac{2}{3}x_6 \\
x_1 & = & 8 & + & \frac{1}{6}x_3 & + & \frac{1}{6}x_5 & - & \frac{1}{3}x_6 \\
x_2 & = & 4 & - & \frac{8}{3}x_3 & - & \frac{2}{3}x_5 & + & \frac{1}{3}x_6 \\
x_4 & = & 18 & - & \frac{1}{2}x_3 & + & \frac{1}{2}x_5 & + & 0x_6.
\end{array}$$

$B = \{1, 2, 4\}$ $N = \{3, 5, 6\}$ $\mathbf{x} = (8, 4, 0, 18, 0, 0)$ $z = 28.$

# Rozwiązanie optymalne

$$\begin{array}{rcrcrcrcr} \max z & & & & & & & & \\ z & = & 28 & - & \frac{1}{6}x_3 & - & \frac{1}{6}x_5 & - & \frac{2}{3}x_6 \\ x_1 & = & 8 & + & \frac{1}{6}x_3 & + & \frac{1}{6}x_5 & - & \frac{1}{3}x_6 \\ x_2 & = & 4 & - & \frac{8}{3}x_3 & - & \frac{2}{3}x_5 & + & \frac{1}{3}x_6 \\ x_4 & = & 18 & - & \frac{1}{2}x_3 & + & \frac{1}{2}x_5 & + & 0x_6 \\ x_1, \ldots, x_6 & \geq & 0 & & & & & & \end{array}$$

$B = \{1, 2, 4\}$ $\quad N = \{3, 5, 6\}$ $\quad \mathbf{x} = (8, 4, 0, 18, 0, 0)$ $\quad z = 28.$

- Nie możemy wymienić bazy (wszystkie współczynniki ujemne!)
- Wszystkie współczynniki ujemne $\Rightarrow$ mamy rozwiązanie optymalne! (bo dla dow. $x_3, x_5, x_6 \geq 0$ będzie $z \leq 28$)
- Kolejne programy liniowe miały taki sam zbiór rozwiązań dopuszczalnych (bo są opisane równoważnymi układami równań)

# Rozwiązanie optymalne

## Wniosek

Jeśli w pewnym kroku algorytmu simplex wszystkie współczynniki (przy zmiennych niebazowych) w funkcji celu są ujemne, to znalezione bazowe rozwiązanie dopuszczalne jest optymalnym rozwiązaniem oryginalnego programu.

W naszym przypadku dostaliśmy rozwiązanie $(x_1, x_2, x_3) = (8, 4, 0)$ o wartości funkcji celu 28.

# simplex: pseudokod

1. Sprowadź PL do postaci dopełnieniowej.
2. Znajdź równoważny PL taki, żeby spełniony był niezmiennik.
3. Dopóki istnieje $j \in \{1, \ldots, n\}$ takie, że $c_j > 0$ ($c_j$ = współczynnik przed $x_{N_j}$ w aktualnej funkcji celu),
   - 3.1 Wybierz takie $j$ ($x_{N_j}$ jest zmienną wchodzącą).
   - 3.2 Jeśli dla każdego $i = 1, \ldots, m$, jest $a_{i,j} \geq 0$ (tzn. dla każdego równania współczynnik przed $x_{N_j}$ jest nieujemny) zwróć „PROGRAM NIEOGRANICZONY".
   - 3.3 wpp., wybierz $i$ takie, że $\frac{b_i}{-a_{i,j}} = \min\{\frac{b_i}{-a_{i,j}} \mid a_{i,j} < 0\}$ ($x_{B_i}$ jest zmienną wychodzącą).
   - 3.4 wykonaj operację Pivot ($j$,$i$)
4. Zwróć rozwiązanie postaci: dla każdego $i = 1, \ldots, n$,

$$x_i = \begin{cases} 0 & \text{gdy } i \in N \\ b_j & \text{gdy } i = B_j \text{ dla pewnego } j = 1, \ldots, m. \end{cases}$$

# simplex: własność stopu

1. Jeśli podczas każdej operacji zamiany bazy wartość funkcji celu rośnie, algorytm zatrzyma się.
2. Ale nie musi tak być:

$$
\begin{array}{rcl}
z &=& 4 + 2x_1 - x_2 - 4x_4 \\
x_3 &=& \frac{1}{2} \qquad\qquad\quad - \frac{1}{2}x_4 \\
x_5 &=& \quad - 2x_1 + 4x_2 + 3x_4 \\
x_6 &=& \quad + x_1 - 3x_2 + 2x_4.
\end{array}
\quad \rightarrow \quad
\begin{array}{rcl}
z &=& 4 + 3x_2 - x_4 - x_5 \\
x_1 &=& \quad + 2x_2 + \frac{3}{2}x_4 - \frac{1}{2}x_5 \\
x_3 &=& \frac{1}{2} \qquad\quad - \frac{1}{2}x_4 \\
x_6 &=& \quad - x_2 + \frac{7}{2}x_4 - \frac{1}{2}x_5.
\end{array}
$$

$$
\begin{array}{lcl}
B = \{3,5,6\},\ N = \{1,2,4\} & \rightarrow & B = \{1,3,6\},\ N = \{2,4,5\}, \\
\mathbf{x} = (0,0,\frac{1}{2},0,0,0),\ z = 4 & & \mathbf{x} = (0,0,\frac{1}{2},0,0,0),\ z = 4
\end{array}
$$

$x_1$ wychodzi z bazy, $x_5$ wchodzi (nie ma innego wyjścia)
baza się zmienia, $\mathbf{x}$ się nie zmienia.

# simplex: własność stopu

## Fakt

Jeśli zmienne wchodzące/wychodzące są wybierane dowolnie algorytm simplex nie ma własności stopu.

## Twierdzenie (Reguła Blanda, 1977)

Jeśli podczas wymiany bazy:

- spośród możliwych zmiennych wchodzących wybierana jest zmienna o najmniejszym indeksie oraz,
- spośród możliwych zmiennych wychodzących wybierana jest zmienna o najmniejszym indeksie

to algorytm simplex kończy swoje działanie.

(Nie będziemy tego dowodzić.)

**Fakt**

Istnieją przykłady programów liniowych, dla których algorytm simplex działa w czasie $\Omega(2^n)$.

Mimo to, następujący problem pozostaje otwarty.

**(Ważny!) problem otwarty**

Czy istnieją reguły wyboru zmiennej wchodzącej i wychodzącej, dla których algorytm simplex działa w czasie wielomianowym?

Najlepsze, co udało się uzyskać:

**Twierdzenie [Kalai 1992, Matousek, Sharir, Welzl 1996]**

Istnieją reguły dla których algorytm simplex działa w **oczekiwanym** czasie $2^{\tilde{O}(\sqrt{n})}$.

# Czego brakuje?

# Czego brakuje? Znajdowanie pierwszego brd

1. Chcemy znaleźć początkowe brd (inaczej: chcemy mieć PL równoważny oryginalnemu, który spełnia niezmiennik).
2. Jeśli program jest postaci

$$\max \mathbf{c}^T \mathbf{x}, \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq 0,$$

   oraz $\mathbf{b} \geq \mathbf{0}$ to **łatwo**: $(0, \ldots, 0, b_1, \ldots, b_m)$.
3. Idea: w ogólnym przypadku... użyjemy algorytmu simplex!

# Znajdowanie pierwszego brd

1. Sprowadź program do postaci dopełnieniowej (P1).
   (Niezmiennik nie zachodzi, bo niekoniecznie $b_i \geq 0$.)
2. Dodaj nową zmienną $x_0$ i zbuduj nowy program.

$$
\begin{array}{c}
(P1) \\
\begin{array}{rl}
\max & 0 + c_1x_1 + \ldots + c_nx_n \\
x_{n+1} = & b_1 + a_{11}x_1 + \ldots + a_{1n}x_n \\
x_{n+2} = & b_2 + a_{21}x_1 + \ldots + a_{2n}x_n \\
\vdots & \\
x_{n+m} = & b_m + a_{m1}x_1 + \ldots + a_{mn}x_n \\
\forall i \; x_i \geq & 0
\end{array}
\end{array}
\rightarrow
\begin{array}{c}
(P2) \\
\begin{array}{rl}
\min & x_0 \\
x_{n+1} = & b_1 + a_{11}x_1 + \ldots + a_{1n}x_n + x_0 \\
x_{n+2} = & b_2 + a_{21}x_1 + \ldots + a_{2n}x_n + x_0 \\
\vdots & \vdots \\
x_{n+m} = & b_m + a_{m1}x_1 + \ldots + a_{mn}x_n + x_0 \\
\forall i \; x_i \geq & 0
\end{array}
\end{array}
$$

**Fakt**

Program (P1) ma rozwiązanie dopuszczalne wtw gdy (P2) ma rozwiązanie optymalne o wartości funckcji celu 0.

# Znajdowanie pierwszego brd

$$
\begin{array}{rl}
\min & x_0 \\
x_{n+1} = & b_1 + a_{11}x_1 + \ldots + a_{1n}x_n + x_0 \\
x_{n+2} = & b_2 + a_{21}x_1 + \ldots + a_{2n}x_n + x_0 \\
\vdots & \vdots \\
x_{n+m} = & b_m + a_{m1}x_1 + \ldots + a_{mn}x_n + x_0 \\
\forall i\ x_i \geq & 0
\end{array}
$$

3. Wybierz $k$ takie, że $b_k = \min_i\{b_i\}$. Możemy założyć, że $b_k < 0$. Usuń z bazy $x_k$ i wprowadź do bazy $x_0$:

$$
\begin{array}{rl}
\min & -b_k - a_{k1}x_1 - \ldots - a_{kn}x_n + x_{n+k} \\
i \neq k \Rightarrow x_{n+i} = & b_i - b_k + (a_{i1} - a_{k1})x_1 + \ldots + (a_{in} - a_{kn})x_n + x_{n+k} \\
x_0 = & -b_k - a_{k1}x_1 - \ldots - a_{kn}x_n + x_{n+k} \\
x_i \geq & 0
\end{array}
$$

Wówczas niezmiennik zachodzi!

4. Za pomocą algorytmu simplex znajdujemy rozwiązanie optymalne $\mathbf{x}^*$.

# Znajdowanie pierwszego brd

Mamy rozwiązanie optymalne $(x_0^*, x_1^*, \ldots, x_n^*)$ programu (P3):

$$\begin{array}{rcccccccc}
 & \min & -b_k & - & a_{k1}x_1 & - \ldots - & a_{kn}x_n & + x_{n+k} \\
i \neq k \Rightarrow & x_{n+i} = & b_i - b_k & + & (a_{i1} - a_{k1})x_1 & + \ldots + & (a_{in} - a_{kn})x_n & + x_{n+k} \\
 & x_0 = & -b_k & - & a_{k1}x_1 & - \ldots - & a_{kn}x_n & + x_{n+k} \\
 & x_i \geq & 0 & & & & &
\end{array}$$

- Jeśli $x_0^* > 0$ zwracamy informację „PROGRAM SPRZECZNY”
- Jeśli $x_0^* = 0$ oraz $x_0$ jest niebazowa, wystarczy z (P3) usunąć $x_0$. Otrzymujemy wtedy program równoważny oryginalnemu programowi (P1), który spełnia niezmiennik
- Jeśli $x_0$ bazowa to mamy równość:

$$x_0 = 0 + \sum_{j \in N} a'_j x_{N_j}.$$

dla pewnego $j$ mamy $a'_j \neq 0$ (inaczej we wszystkich rozwiązaniach dopuszczalnych $x_0 = 0$, sprzeczność, bo w pierwszym brd tak nie było). Zamienamy $x_0$ z $x_{N_j}$.