

Optimalizace

Simplexová metoda

Tomáš Kroupa Tomáš Werner

2023 LS

Fakulta elektrotechnická
ČVUT v Praze

Co zatím víme o úloze LP?

- Každý lineární program lze převést do *rovnicevého tvaru*

$$\min\{ \mathbf{c}^T \mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \},$$

kde $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$.

- Je-li úloha přípustná, konvexní polyedr

$$\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \}$$

neobsahuje přímku a tedy má alespoň jeden vrchol.

- Má-li lineární funkce polyedru minimum, leží ve vrcholu.

Naučíme se efektivně procházet vrcholy.

Předpoklad: $\text{rank } \mathbf{A} = m$

Definice

Báze je m -prvková množina $J \subseteq \{1, \dots, n\}$ taková, že sloupce matice \mathbf{A} indexované množinou J jsou lineárně nezávislé.

Bázevá řešení

Pro každou bázi J má soustava $\mathbf{Ax} = \mathbf{b}$ jediné řešení \mathbf{x} splňující $x_j = 0$ pro všechna $j \notin J$. Takové řešení se nazývá

- **přípustné**, pokud $\mathbf{x} \geq \mathbf{0}$,
- **degenerované**, pokud $x_j = 0$ pro nějaké $j \in J$.

Příklad

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} = \begin{array}{ccccccc} -1 & 1 & 3 & 1 & 0 & 2 & 1 \\ 1 & 0 & 4 & 0 & 1 & 4 & 4 \\ -1 & 0 & 4 & 1 & 1 & 4 & 2 \end{array}$$

•

Příklad

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} = \begin{array}{ccccccc} -1 & 1 & 3 & 1 & 0 & 2 & 1 \\ 1 & 0 & 4 & 0 & 1 & 4 & 4 \\ -1 & 0 & 4 & 1 & 1 & 4 & 2 \end{array}$$

- $\{2, 3, 5\}$ není báze (sloupce jsou lineárně závislé)

Příklad

$$\begin{array}{r} [\mathbf{A} \quad \mathbf{b}] = \\ \mathbf{x} = \end{array} \begin{array}{r} -1 \quad 1 \quad 3 \quad 1 \quad 0 \quad 2 \quad 1 \\ 1 \quad 0 \quad 4 \quad 0 \quad 1 \quad 4 \quad 4 \\ -1 \quad 0 \quad 4 \quad 1 \quad 1 \quad 4 \quad 2 \\ 3 \quad 0 \quad 0 \quad 4 \quad 1 \quad 0 \end{array}$$

- $\{1, 4, 5\}$ je báze. Bázové řešení \mathbf{x} je řešením soustavy

$$\begin{bmatrix} -1 & 1 & 0 \\ 1 & 0 & 1 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix}, \quad x_2 = x_3 = x_6 = 0.$$

Je přípustné, není degenerované.

Příklad

$$\begin{array}{r} \left[\mathbf{A} \quad \mathbf{b} \right] = \\ \begin{array}{cccccccc} -1 & 1 & 3 & 1 & 0 & 2 & 1 \\ 1 & 0 & 4 & 0 & 1 & 4 & 4 \\ -1 & 0 & 4 & 1 & 1 & 4 & 2 \end{array} \\ \mathbf{x} = \begin{array}{cccccc} 4 & -1 & 0 & 6 & 0 & 0 \end{array} \end{array}$$

- $\{1, 2, 4\}$ je báze. Bázové řešení je nepřípustné, není degenerované.

Příklad

$$\begin{array}{r} \mathbf{[A \quad b]} = \\ \mathbf{x} = \end{array} \begin{array}{cccccc} -1 & 1 & 3 & 1 & 0 & 2 & 1 \\ 1 & 0 & 4 & 0 & 1 & 4 & 4 \\ -1 & 0 & 4 & 1 & 1 & 4 & 2 \\ 0 & 0 & 1 & -2 & 0 & 0 & 0 \end{array}$$

- $\{3, 4, 5\}$ je báze. Bázové řešení je nepřípustné a degenerované.

Příklad

$$\begin{array}{r} \left[\mathbf{A} \quad \mathbf{b} \right] = \\ \mathbf{x} = \end{array} \begin{array}{cccccccc} -1 & 1 & 3 & 1 & 0 & 2 & 1 \\ 1 & 0 & 4 & 0 & 1 & 4 & 4 \\ -1 & 0 & 4 & 1 & 1 & 4 & 2 \\ 0 & 0 & 1 & -2 & 0 & 0 & 0 \end{array}$$

- Stejné bázové řešení odpovídá bázi $\{3, 4, 6\}$.
Degenerované bázové řešení odpovídá více než jedné bázi!

Tvrzení

Bod x konvexního polyedru je přípustným bázevým řešením právě tehdy, pokud je x vrchol.

Báze jsou **sousední**, pokud mají společných $m - 1$ prvků.

- Sousední báze odpovídají dvěma vrcholům spojených hranou anebo jedinému degenerovanému vrcholu
- Simplexová metoda přechází mezi sousedními bázemi, přičemž zachovává přípustnost řešení a účelová funkce neroste

Standardní báze

Simplexová metoda používá jen **standardní báze**:

- Nenulové složky bazového řešení \mathbf{x} jsou jednoduše složky \mathbf{b}
- Tedy bazové řešení \mathbf{x} je přípustné, právě když $\mathbf{b} \geq \mathbf{0}$

Standardní báze

Simplexová metoda používá jen **standardní báze**:

- Nenulové složky bazového řešení \mathbf{x} jsou jednoduše složky \mathbf{b}
- Tedy bazové řešení \mathbf{x} je přípustné, právě když $\mathbf{b} \geq \mathbf{0}$

$$\begin{aligned} [\mathbf{A} \quad \mathbf{b}] &= \begin{array}{cccccc} 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array} \\ \mathbf{x} &= \begin{array}{cccccc} 3 & 0 & 0 & 4 & 1 & 0 \end{array} \end{aligned}$$

Bázové řešení příslušné standardní bázi $\{1, 4, 5\}$ je řešením soustavy

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}, \quad x_2 = x_3 = x_6 = 0$$

Stavební kameny algoritmu

Přechod k sousední standardní bázi

- Řádek matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$ vynásob nenulovým číslem
- K řádku matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$ přičti lineární kombinaci ostatních řádků

Chceme bázový sloupec $j' \in J$ nahradit nebázovým sloupcem $j \notin J$:

- **Pivot** je prvek (i, j) , kde i je řádek splňující $a_{ij'} = 1$
- Proveď *ekvivalentní úpravu kolem pivotu* (i, j) , tj. nastav $a_{ij} = 1$ a $a_{i'j} = 0$ pro každé $i' \neq i$:

Přechod k sousední standardní bázi

- Řádek matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$ vynásob nenulovým číslem
- K řádku matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$ přičti lineární kombinaci ostatních řádků

Chceme bázevý sloupec $j' \in J$ nahradit nebázevým sloupcem $j \notin J$:

- **Pivot** je prvek (i, j) , kde i je řádek splňující $a_{ij'} = 1$
- Proveď *ekvivalentní úpravu kolem pivotu* (i, j) , tj. nastav $a_{ij} = 1$ a $a_{i'j} = 0$ pro každé $i' \neq i$:

$$\begin{array}{ccccccc} & 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ & 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ i & 0 & -1 & 1 & 0 & 1 & 2 & 1 \\ & & j & & & j' & & \end{array}$$

Přechod k sousední standardní bázi

- Řádek matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$ vynásob nenulovým číslem
- K řádku matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$ přičti lineární kombinaci ostatních řádků

Chceme bázový sloupec $j' \in J$ nahradit nebázovým sloupcem $j \notin J$:

- **Pivot** je prvek (i, j) , kde i je řádek splňující $a_{ij'} = 1$
- Proveď *ekvivalentní úpravu kolem pivotu* (i, j) , tj. nastav $a_{ij} = 1$ a $a_{i'j} = 0$ pro každé $i' \neq i$:
 1. Vyděl řádek i pivotem a_{ij} .

$$\begin{array}{cccccccc} & & 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ & & 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ i & & 0 & \mathbf{1} & -1 & 0 & -1 & -2 & -1 \\ & & & j & & & j' & & \end{array}$$

Přechod k sousední standardní bázi

- Řádek matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$ vynásob nenulovým číslem
- K řádku matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$ přičti lineární kombinaci ostatních řádků

Chceme bázevý sloupec $j' \in J$ nahradit nebázevým sloupcem $j \notin J$:

- **Pivot** je prvek (i, j) , kde i je řádek splňující $a_{ij'} = 1$
- Proveď *ekvivalentní úpravu kolem pivotu* (i, j) , tj. nastav $a_{ij} = 1$ a $a_{i'j} = 0$ pro každé $i' \neq i$:
 1. Vyděl řádek i pivotem a_{ij} .
 2. Pro každé $i' \neq i$ odečti $a_{i'j}$ -násobek řádku i od řádku i' .

$$\begin{array}{cccccccc} & 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ & 1 & \mathbf{0} & 4 & 0 & 1 & 0 & 4 \\ i & 0 & \mathbf{1} & -1 & 0 & -1 & -2 & -1 \\ & & & & j & & & j' \end{array}$$

Přechod k sousední standardní bázi

- Řádek matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$ vynásob nenulovým číslem
- K řádku matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$ přičti lineární kombinaci ostatních řádků

Chceme bázový sloupec $j' \in J$ nahradit nebázovým sloupcem $j \notin J$:

- **Pivot** je prvek (i, j) , kde i je řádek splňující $a_{ij'} = 1$
- Proveď *ekvivalentní úpravu kolem pivotu* (i, j) , tj. nastav $a_{ij} = 1$ a $a_{i'j} = 0$ pro každé $i' \neq i$:
 1. Vyděl řádek i pivotem a_{ij} .
 2. Pro každé $i' \neq i$ odečti $a_{i'j}$ -násobek řádku i od řádku i' .

$$\begin{array}{cccccccc} 0 & \mathbf{0} & 8 & 1 & 2 & 8 & 6 & \\ 1 & \mathbf{0} & 4 & 0 & 1 & 0 & 4 & \\ i & 0 & \mathbf{1} & -1 & 0 & -1 & -2 & -1 \\ & & & j & & & j' & \end{array}$$

Bude nové báze řešení přípustné?

Jak se změní \mathbf{b} po ekvivalentní úpravě kolem pivotu (i, j) ?

- b_i se změní na b_i/a_{ij}
- Pro každé $i' \neq i$ se $b_{i'}$ změní na $b_{i'} - a_{i'j} \frac{b_i}{a_{ij}}$

Tato čísla musejí zůstat nezáporná, což nastane právě když

$$a_{ij} > 0 \quad \text{a} \quad \forall i' \neq i \text{ platí } a_{i'j} \leq 0 \text{ nebo } \frac{b_i}{a_{ij}} \leq \frac{b_{i'}}{a_{i'j}}$$

$$\begin{array}{ccccccc} 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array}$$

Bude nové báze řešení přípustné?

Jak se změní \mathbf{b} po ekvivalentní úpravě kolem pivotu (i, j) ?

- b_i se změní na b_i/a_{ij}
- Pro každé $i' \neq i$ se $b_{i'}$ změní na $b_{i'} - a_{i'j} \frac{b_i}{a_{ij}}$

Tato čísla musejí zůstat nezáporná, což nastane právě když

$$a_{ij} > 0 \quad \text{a} \quad \forall i' \neq i \text{ platí } a_{i'j} \leq 0 \text{ nebo } \frac{b_i}{a_{ij}} \leq \frac{b_{i'}}{a_{i'j}}$$

$$\begin{array}{ccccccc} 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array}$$

- Po úpravě kolem $(3, 2)$ **nebude** $\mathbf{b} \geq \mathbf{0}$, neboť $-1 \not\geq 0$

Bude nové bázové řešení přípustné?

Jak se změní \mathbf{b} po ekvivalentní úpravě kolem pivotu (i, j) ?

- b_i se změní na b_i/a_{ij}
- Pro každé $i' \neq i$ se $b_{i'}$ změní na $b_{i'} - a_{i'j} \frac{b_i}{a_{ij}}$

Tato čísla musejí zůstat nezáporná, což nastane právě když

$$a_{ij} > 0 \quad \text{a} \quad \forall i' \neq i \text{ platí } a_{i'j} \leq 0 \text{ nebo } \frac{b_i}{a_{ij}} \leq \frac{b_{i'}}{a_{i'j}}$$

$$\begin{array}{ccccccc} 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & \mathbf{1} & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array}$$

- Po úpravě kolem $(2, 2)$ **nebude** $\mathbf{b} \geq \mathbf{0}$, neboť $\frac{3}{1} \not\leq \frac{4}{2}$

Bude nové bázové řešení přípustné?

Jak se změní \mathbf{b} po ekvivalentní úpravě kolem pivotu (i, j) ?

- b_i se změní na b_i/a_{ij}
- Pro každé $i' \neq i$ se $b_{i'}$ změní na $b_{i'} - a_{i'j} \frac{b_i}{a_{ij}}$

Tato čísla musejí zůstat nezáporná, což nastane právě když

$$a_{ij} > 0 \quad \text{a} \quad \forall i' \neq i \text{ platí } a_{i'j} \leq 0 \text{ nebo } \frac{b_i}{a_{ij}} \leq \frac{b_{i'}}{a_{i'j}}$$

$$\begin{array}{ccccccc} 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array}$$

- Po úpravě kolem $(3, 6)$ bude $\mathbf{b} \geq \mathbf{0}$, neboť $2 > 0$, $\frac{1}{2} \leq \frac{3}{2}$, $\frac{1}{2} \leq \frac{4}{4}$

Nekladný sloupec

Když jsou všechny prvky v nebázovém sloupci $j \notin J$ nekladné:

- Sloupec j se nemůže stát bázovým (nelze v něm vybrat pivot)
- Existuje směr \mathbf{v} tak, že $\mathbf{x} + \alpha\mathbf{v} \in X$ pro každé $\alpha \geq 0$

$$\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} = \begin{array}{cccccc} 0 & -2 & 6 & 1 & 0 & 4 & 4 \\ 1 & -1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array}$$

$$\mathbf{x} = \begin{array}{cccccc} 3 & 0 & 0 & 4 & 1 & 0 \end{array}$$

$$\mathbf{v} = \begin{array}{cccccc} 1 & 1 & 0 & 2 & 1 & 0 \end{array}$$

Ekvivalentní úpravy účelového řádku

Lineární program

$$\min\{\mathbf{c}^T \mathbf{x} - d \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

reprezentujeme **simplexovou tabulkou**

$$\begin{bmatrix} \mathbf{c}^T & d \\ \mathbf{A} & \mathbf{b} \end{bmatrix}.$$

Přičti k prvnímu řádku tabulky lineární kombinaci ostatních řádků:

$$\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} + \mathbf{y}^T \begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} = \begin{bmatrix} \mathbf{c}^T + \mathbf{y}^T \mathbf{A} & d + \mathbf{y}^T \mathbf{b} \end{bmatrix}$$

Účelová funkce se tím nezmění:

$$(\mathbf{c}^T + \mathbf{y}^T \mathbf{A})\mathbf{x} - (d + \mathbf{y}^T \mathbf{b}) = \mathbf{c}^T \mathbf{x} - d$$

Redukované ceny

Vynulujeme bázové složky vektoru \mathbf{c} , tj. $c_j = 0$ pro $j \in J$:

$$\begin{array}{r} \left[\mathbf{c}^T \quad d \right] = \\ \left[\mathbf{A} \quad \mathbf{b} \right] = \\ \mathbf{x} = \end{array} \begin{array}{ccccccc} 1 & -2 & -3 & -1 & 2 & 1 & 4 \\ 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \\ 3 & 0 & 0 & 4 & 1 & 0 & \end{array}$$

Redukované ceny

Vynulujeme bázové složky vektoru \mathbf{c} , tj. $c_j = 0$ pro $j \in J$:

$$\begin{array}{r} \left[\mathbf{c}^T \quad d \right] = \\ \left[\mathbf{A} \quad \mathbf{b} \right] = \\ \mathbf{x} = \end{array} \begin{array}{r} 0 \quad -3 \quad -6 \quad -1 \quad 2 \quad -1 \quad 1 \\ 0 \quad 2 \quad 6 \quad 1 \quad 0 \quad 4 \quad 4 \\ 1 \quad 1 \quad 3 \quad 0 \quad 0 \quad 2 \quad 3 \\ 0 \quad -1 \quad 1 \quad 0 \quad 1 \quad 2 \quad 1 \\ 3 \quad 0 \quad 0 \quad 4 \quad 1 \quad 0 \end{array}$$

Redukované ceny

Vynulujeme bázové složky vektoru \mathbf{c} , tj. $c_j = 0$ pro $j \in J$:

$$\begin{array}{rcl} \left[\mathbf{c}^T & d \right] & = \quad 0 \quad -1 \quad 0 \quad 0 \quad 2 \quad 3 \quad 5 \\ & & 0 \quad 2 \quad 6 \quad 1 \quad 0 \quad 4 \quad 4 \\ \left[\mathbf{A} & \mathbf{b} \right] & = \quad 1 \quad 1 \quad 3 \quad 0 \quad 0 \quad 2 \quad 3 \\ & & 0 \quad -1 \quad 1 \quad 0 \quad 1 \quad 2 \quad 1 \\ & & \mathbf{x} = \quad 3 \quad 0 \quad 0 \quad 4 \quad 1 \quad 0 \end{array}$$

Redukované ceny

Vynulujeme bázové složky vektoru \mathbf{c} , tj. $c_j = 0$ pro $j \in J$:

$$\begin{array}{r} \left[\mathbf{c}^T \quad d \right] = \\ \left[\mathbf{A} \quad \mathbf{b} \right] = \\ \mathbf{x} = \end{array} \begin{array}{ccccccc} 0 & 1 & -2 & 0 & 0 & -1 & 3 \\ 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \\ 3 & 0 & 0 & 4 & 1 & 0 & \end{array}$$

Redukované ceny

Vynulujeme bázové složky vektoru \mathbf{c} , tj. $c_j = 0$ pro $j \in J$:

$$\begin{array}{r} \left[\mathbf{c}^T \quad d \right] = \\ \left[\mathbf{A} \quad \mathbf{b} \right] = \\ \mathbf{x} = \end{array} \begin{array}{ccccccc} 0 & 1 & -2 & 0 & 0 & -1 & 3 \\ 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \\ 3 & 0 & 0 & 4 & 1 & 0 & \end{array}$$

- Protože $x_j = 0$ pro $j \notin J$, je $\mathbf{c}^T \mathbf{x} = 0$ a tedy $\mathbf{c}^T \mathbf{x} - d = -d$.
 - Ihned vidíme, co by udělal vstup sloupce $j \notin J$ do báze:
 - když $c_j > 0$, účelová hodnota by stoupla
 - když $c_j < 0$, účelová hodnota by klesla
- (*předpoklad*: nové bázové řešení nebude degenerované, tj. $x_j > 0$).
- Pokud v některém sloupci j je $c_j < 0$ a $a_{ij} \leq 0$ pro všechna i , účelovou funkci lze libovolně zmenšovat \Rightarrow úloha je neomezená.

Základní algoritmus

Tvar simplexové tabulky

$$\begin{bmatrix} \mathbf{c}^T & d \\ \mathbf{A} & \mathbf{b} \end{bmatrix}$$

- Podmnožina sloupců \mathbf{A} tvoří standardní bázi
- $\mathbf{b} \geq \mathbf{0}$
- ceny c_j v bazových sloupcích jsou nulové

Iterace simplexového algoritmu

$$\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} = \left[\begin{array}{cccccc|c} 0 & -2 & 1 & 0 & 0 & -3 & 0 \\ 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ \mathbf{A} & \mathbf{b} & & & & & \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array} \right]$$

1. Vyber sloupec j pivotu tak, aby $c_j < 0$.

Iterace simplexového algoritmu

$$\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} = \left[\begin{array}{cccccc|c} 0 & -2 & 1 & 0 & 0 & -3 & 0 \\ \hline 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array} \right]$$

1. Vyber sloupec j pivotu tak, aby $c_j < 0$.
2. Vyber řádek i pivotu tak, aby nové bázevé řešení bylo přípustné:

$$i \in \arg \min_{i' \mid a_{i'j} > 0} \frac{b_{i'}}{a_{i'j}}$$

Iterace simplexového algoritmu

$$\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} = \left[\begin{array}{cccccc|c} 0 & -2 & 1 & 0 & 0 & -3 & 0 \\ 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array} \right]$$

1. Vyber sloupec j pivotu tak, aby $c_j < 0$.
2. Vyber řádek i pivotu tak, aby nové bázové řešení bylo přípustné:

$$i \in \arg \min_{i' \mid a_{i'j} > 0} \frac{b_{i'}}{a_{i'j}}$$

3. Udělej ekvivalentní úpravu okolo pivotu (i, j) a redukuj cenu c_j , tj.

Iterace simplexového algoritmu

$$\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} = \left[\begin{array}{cccccc|c} 0 & -2 & 1 & 0 & 0 & -3 & 0 \\ 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array} \right]$$

1. Vyber sloupec j pivotu tak, aby $c_j < 0$.
2. Vyber řádek i pivotu tak, aby nové bázové řešení bylo přípustné:

$$i \in \arg \min_{i' \mid a_{i'j} > 0} \frac{b_{i'}}{a_{i'j}}$$

3. Udělej ekvivalentní úpravu okolo pivotu (i, j) a redukuj cenu c_j , tj.
 - nastav pivot na jedničku

Iterace simplexového algoritmu

$$\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} = \left[\begin{array}{cccccc|c} 0 & -2 & 1 & 0 & 0 & -3 & 0 \\ 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -0.5 & 0.5 & 0 & 0.5 & \mathbf{1} & 0.5 \end{array} \right]$$

1. Vyber sloupec j pivotu tak, aby $c_j < 0$.
2. Vyber řádek i pivotu tak, aby nové bázové řešení bylo přípustné:

$$i \in \arg \min_{i' \mid a_{i'j} > 0} \frac{b_{i'}}{a_{i'j}}$$

3. Udělej ekvivalentní úpravu okolo pivotu (i, j) a redukuj cenu c_j , tj.
 - nastav pivot na jedničku
 - vynuluj prvky nad a pod pivotem (vč. účelového řádku)

Iterace simplexového algoritmu

$$\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} = \left[\begin{array}{cccccc|c} 0 & -2 & 1 & 0 & 0 & -3 & 0 \\ 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 2 & 2 & 0 & -1 & 0 & 2 \\ 0 & -0.5 & 0.5 & 0 & 0.5 & 1 & 0.5 \end{array} \right]$$

1. Vyber sloupec j pivotu tak, aby $c_j < 0$.
2. Vyber řádek i pivotu tak, aby nové bázové řešení bylo přípustné:

$$i \in \arg \min_{i' \mid a_{i'j} > 0} \frac{b_{i'}}{a_{i'j}}$$

3. Udělej ekvivalentní úpravu okolo pivotu (i, j) a redukuj cenu c_j , tj.
 - nastav pivot na jedničku
 - vynuluj prvky nad a pod pivotem (vč. účelového řádku)

Iterace simplexového algoritmu

$$\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} = \left[\begin{array}{cccccc|c} 0 & -2 & 1 & 0 & 0 & -3 & 0 \\ 0 & 4 & 4 & 1 & -2 & 0 & 2 \\ 1 & 2 & 2 & 0 & -1 & 0 & 2 \\ 0 & -0.5 & 0.5 & 0 & 0.5 & 1 & 0.5 \end{array} \right]$$

1. Vyber sloupec j pivotu tak, aby $c_j < 0$.
2. Vyber řádek i pivotu tak, aby nové bázevé řešení bylo přípustné:

$$i \in \arg \min_{i' \mid a_{i'j} > 0} \frac{b_{i'}}{a_{i'j}}$$

3. Udělej ekvivalentní úpravu okolo pivotu (i, j) a redukuj cenu c_j , tj.
 - nastav pivot na jedničku
 - vynuluj prvky nad a pod pivotem (vč. účelového řádku)

Iterace simplexového algoritmu

$$\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} = \left[\begin{array}{cccccc|c} 0 & -3.5 & 2.5 & 0 & 1.5 & 0 & 1.5 \\ 0 & 4 & 4 & 1 & -2 & 0 & 2 \\ 1 & 2 & 2 & 0 & -1 & 0 & 2 \\ 0 & -0.5 & 0.5 & 0 & 0.5 & 1 & 0.5 \end{array} \right]$$

1. Vyber sloupec j pivotu tak, aby $c_j < 0$.
2. Vyber řádek i pivotu tak, aby nové bázové řešení bylo přípustné:

$$i \in \arg \min_{i' \mid a_{i'j} > 0} \frac{b_{i'}}{a_{i'j}}$$

3. Udělej ekvivalentní úpravu okolo pivotu (i, j) a redukuj cenu c_j , tj.
 - nastav pivot na jedničku
 - vynuluj prvky nad a pod pivotem (vč. účelového řádku)

Ukončení algoritmu

Algoritmus končí, když je splněna jedna z podmínek:

- Všechny ceny c_j jsou nezáporné (jsme v minimu).

0	0	7	1	0	1	4
0	1	3	0.5	0	2	2
1	0	0	-0.5	0	0	1
0	0	4	0.5	1	4	3

Ukončení algoritmu

Algoritmus končí, když je splněna jedna z podmínek:

- Všechny ceny c_j jsou nezáporné (jsme v minimu).

0	0	7	1	0	1	4
0	1	3	0.5	0	2	2
1	0	0	-0.5	0	0	1
0	0	4	0.5	1	4	3

- V každém sloupci j , ve kterém $c_j < 0$, je $a_{ij} \leq 0$ pro všechna i (úloha je neomezená).

0	0	7	-1	0	1	4
0	1	3	-0.5	0	2	2
1	0	0	-0.5	0	0	1
0	0	4	-0.5	1	4	3

Cyklení

Algoritmus nemusí vždy skončit kvůli *cyklení*.

-2.3	-2.15	13.55	0.4	0	0		0
0.4	0.2	-1.4	-0.2	1	0		0
-7.8	-1.4	7.8	0.4	0	1		0

Cyklení

Algoritmus nemusí vždy skončit kvůli *cyklení*.

0	-1	5.5	-0.75	5.75	0	0
1	0.5	-3.5	-0.5	2.5	0	0
0	2.5	-19.5	-3.5	19.5	1	0

Cyklení

Algoritmus nemusí vždy skončit kvůli *cyklení*.

0	0	-2.3	-2.15	13.55	0.4		0
1	0	0.4	0.2	-1.4	-0.2		0
0	1	-7.8	-1.4	7.8	0.4		0

Tohle je počáteční tabulka se sloupci rotovanými o dva doprava.
Další 4 iterace dospějí do počáteční tabulky!

Algoritmus nemusí vždy skončit kvůli *cyklení*.

0	0	-2.3	-2.15	13.55	0.4		0
1	0	0.4	0.2	-1.4	-0.2		0
0	1	-7.8	-1.4	7.8	0.4		0

Tohle je počáteční tabulka se sloupci rotovanými o dva doprava.
Další 4 iterace dospějí do počáteční tabulky!

Blandovo anticyklické pravidlo:

- Při výběru pivotového sloupce vyber sloupec s nejnižším indexem
- Při výběru pivotového řádku vždy vyber řádek s nejnižším indexem

Inicializace algoritmu – speciální případ

Pro zahájení simplexového algoritmu musíme úlohu převést na tvar

$$\min\{\mathbf{c}^T \mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

kde podmnožina sloupců \mathbf{A} tvoří standardní bázi a $\mathbf{b} \geq \mathbf{0}$.

- Někdy je to snadné. Např. když má vstupní úloha tvar

$$\min\{\mathbf{c}^T \mathbf{x} \mid \mathbf{Ax} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}, \quad \mathbf{b} \geq \mathbf{0}$$

- Přidáním slackových proměnných \mathbf{u} ji převedeme na úlohu

$$\min\{\mathbf{c}^T \mathbf{x} \mid \mathbf{Ax} + \mathbf{u} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{u} \geq \mathbf{0}\}$$

se simplexovou tabulkou

$$\begin{bmatrix} \mathbf{c}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{A} & \mathbf{I} & \mathbf{b} \end{bmatrix}$$

ve které sloupce příslušné proměnným \mathbf{u} tvoří standardní bázi \mathbf{I}

- Vstupní lineární program lze vždy efektivně převést na tvar

$$\min\{ \mathbf{c}^T \mathbf{x} \mid \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq \mathbf{0} \}$$

kde $\mathbf{b} \geq \mathbf{0}$

- Ale \mathbf{A} nemusí obsahovat standardní bázi!

Dvoufázová simplexová metoda

Dvoufázová simplexová metoda

Vyřeš pomocnou úlohu

$$\min\{ \mathbf{1}^T \mathbf{u} \mid \mathbf{Ax} + \mathbf{u} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{u} \geq \mathbf{0} \} \quad (1)$$

se simplexovou tabulkou $\begin{bmatrix} \mathbf{0} & \mathbf{1}^T & \mathbf{0} \\ \mathbf{A} & \mathbf{I} & \mathbf{b} \end{bmatrix}$.

- Vstupní úloha **nepřípustná** \Leftrightarrow (1) má kladnou optimální hodnotu
- Vstupní úloha je **přípustná** \Leftrightarrow (1) má nulovou optimální hodnotu
 - Je-li opt. řešení (\mathbf{x}, \mathbf{u}) úlohy (1) **nedegenerované**, pak všechny sloupce příslušné proměnným \mathbf{u} jsou nebázové, proto matice \mathbf{A} obsahuje standardní bázi.
 - Je-li je opt. řešení (\mathbf{x}, \mathbf{u}) úlohy (1) **degenerované**, některé sloupce příslušné proměnným \mathbf{u} mohou být bázové.
Dalším pivotováním je možno bázi z těchto sloupců 'vyhnat'.