Optimalizace

11. Simplexová metoda

Tomáš Werner

FEL ČVUT

Báze a bázová řešení

Vstupní mnohostěn: $\{\, \mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \,\}$ kde $\mathbf{A} \in \mathbb{R}^{m \times n}$ má lin. nezávislé řádky.

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

Báze a bázová řešení

Vstupní mnohostěn: $\{\, \mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \,\}$ kde $\mathbf{A} \in \mathbb{R}^{m \times n}$ má lin. nezávislé řádky.

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

- **Báze** je množina $J \subseteq \{1, \ldots, n\}$ taková, že $|J| = m$ a sloupce $\mathbf{A}$ s indexy J jsou lin. nezávislé.

Báze a bázová řešení

Vstupní mnohostěn: $\{\, \mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \,\}$ kde $\mathbf{A} \in \mathbb{R}^{m \times n}$ má lin. nezávislé řádky.

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

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

- **Báze** je množina $J \subseteq \{1, \dots, n\}$ taková, že $|J| = m$ a sloupce $\mathbf{A}$ s indexy J jsou lin. nezávislé.
- **Bázové řešení** příslušné bázi J je (jediné) řešení $\mathbf{x} = (x_1, \dots, x_n)$ soustavy

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$x_j = 0 \quad \forall j \notin J$$

Báze a bázová řešení

Vstupní mnohostěn: $\{\, \mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \,\}$ kde $\mathbf{A} \in \mathbb{R}^{m \times n}$ má lin. nezávislé řádky.

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

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

- **Báze** je množina $J \subseteq \{1, \ldots, n\}$ taková, že $|J| = m$ a sloupce $\mathbf{A}$ s indexy J jsou lin. nezávislé.
- **Bázové řešení** příslušné bázi J je (jediné) řešení $\mathbf{x} = (x_1, \ldots, x_n)$ soustavy

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

$$x_j = 0 \quad \forall j \notin J$$

- Bázové řešení $\mathbf{x}$ je **přípustné**, jestliže $\mathbf{x} \geq \mathbf{0}$.
  Přípustná bázová řešení odpovídají extremálním bodům mnohostěnu.

Báze a bázová řešení

Vstupní mnohostěn: $\{\, \mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \,\}$ kde $\mathbf{A} \in \mathbb{R}^{m \times n}$ má lin. nezávislé řádky.

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

$$\mathbf{x} = \begin{array}{cccccc} 4 & -1 & 0 & 6 & 0 & 0 \end{array}$$

- **Báze** je množina $J \subseteq \{1, \ldots, n\}$ taková, že $|J| = m$ a sloupce $\mathbf{A}$ s indexy J jsou lin. nezávislé.
- **Bázové řešení** příslušné bázi J je (jediné) řešení $\mathbf{x} = (x_1, \ldots, x_n)$ soustavy

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
$$x_j = 0 \quad \forall j \notin J$$

- Bázové řešení $\mathbf{x}$ je **přípustné**, jestliže $\mathbf{x} \geq \mathbf{0}$.
  Přípustná bázová řešení odpovídají extremálním bodům mnohostěnu.

Báze a bázová řešení

Vstupní mnohostěn: $\{\, \mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0} \,\}$ kde $\mathbf{A} \in \mathbb{R}^{m \times n}$ má lin. nezávislé řádky.

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

- **Báze** je množina $J \subseteq \{1, \ldots, n\}$ taková, že $|J| = m$ a sloupce $\mathbf{A}$ s indexy J jsou lin. nezávislé.
- **Bázové řešení** příslušné bázi J je (jediné) řešení $\mathbf{x} = (x_1, \ldots, x_n)$ soustavy

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$
$$x_j = 0 \quad \forall j \notin J$$

- Bázové řešení $\mathbf{x}$ je **přípustné**, jestliže $\mathbf{x} \geq \mathbf{0}$.
  Přípustná bázová řešení odpovídají extremálním bodům mnohostěnu.
- Báze J a J' jsou **sousední**, jestliže $|J \cap J'| = m - 1$ (tj. liší se o jeden sloupec).
  Např. báze $\{1, 4, 5\}$ a $\{1, 2, 4\}$ jsou sousední.

Ekvivalentní úpravy úlohy

Uvažujeme úlohu ve tvaru

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

Reprezentujeme ji **simplexovou tabulkou**

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

Ekvivalentní úpravy úlohy

Uvažujeme úlohu ve tvaru

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

Reprezentujeme ji **simplexovou tabulkou**

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

Ekvivalentní řádkové úpravy tabulky:

- 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 násobek jiného řádku této matice.
- K účelovému řádku $\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix}$ přičti násobek řádku matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$.

Ekvivalentní úpravy úlohy

Uvažujeme úlohu ve tvaru

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

Reprezentujeme ji **simplexovou tabulkou**

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

Ekvivalentní řádkové úpravy tabulky:

- 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 násobek jiného řádku této matice.
- K účelovému řádku $\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix}$ přičti násobek řádku matice $\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}$.

Tyto úpravy zachovávají:

- množinu řešení soustavy $\mathbf{A}\mathbf{x} = \mathbf{b}$,
- hodnotu účelové funkce $\mathbf{c}^T\mathbf{x} - d$ pro všechna řešení soustavy $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Standardní tvar simplexové tabulky

$$\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} = \begin{array}{ccccccc} \color{red}{0} & -2 & 1 & \color{red}{0} & \color{red}{0} & -3 & -1 \end{array}$$

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

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

- Podmnožina J sloupců matice $\mathbf{A}$ tvoří **standardní bázi**.
  Ihned vidíme příslušné bázové řešení $\mathbf{x}$: jeho nenulové složky tvoří vektor $\mathbf{b}$.

Standardní tvar simplexové tabulky

$$
\begin{array}{rcrrrrrrr}
\begin{bmatrix} \mathbf{c}^T & d \end{bmatrix} & = & {\color{red}0} & -2 & 1 & {\color{red}0} & {\color{red}0} & -3 & -1 \\[1em]
 & & {\color{red}0} & 2 & 6 & {\color{red}1} & {\color{red}0} & 4 & 4 \\
\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix} & = & {\color{red}1} & 1 & 3 & {\color{red}0} & {\color{red}0} & 2 & 3 \\
 & & {\color{red}0} & -1 & 1 & {\color{red}0} & {\color{red}1} & 2 & 1 \\[1em]
\mathbf{x} & = & 3 & 0 & 0 & 4 & 1 & 0 &
\end{array}
$$

- Podmnožina J sloupců matice $\mathbf{A}$ tvoří **standardní bázi**.
  Ihned vidíme příslušné bázové řešení $\mathbf{x}$: jeho nenulové složky tvoří vektor $\mathbf{b}$.
- Bázové řešení je **přípustné** ($\mathbf{b} \geq \mathbf{0}$).

Standardní tvar simplexové tabulky

$$\begin{bmatrix}\mathbf{c}^T & d\end{bmatrix} = \begin{array}{ccccccc} 0 & -2 & 1 & 0 & 0 & -3 & -1 \end{array}$$

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

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

- Podmnožina J sloupců matice $\mathbf{A}$ tvoří **standardní bázi**.
  Ihned vidíme příslušné bázové řešení $\mathbf{x}$: jeho nenulové složky tvoří vektor $\mathbf{b}$.
- Bázové řešení je **přípustné** ($\mathbf{b} \geq \mathbf{0}$).
- Ceny c_j v bázových sloupcích jsou nulové (tzv. **redukované ceny**).
  Protože pro nebázové sloupce je $x_j = 0$, aktuální hodnota účelové funkce je $\mathbf{c}^T\mathbf{x} - d = -d$.

Iterace simplexového algoritmu

Cíl iterace: Přejít k sousední bázi tak, aby bázové řešení zůstalo přípustné a účelová funkce se nezhoršila.

Iterace simplexového algoritmu

Cíl iterace: Přejít k sousední bázi tak, aby bázové řešení zůstalo přípustné a účelová funkce se nezhoršila.

Vstup i výstup iterace je simplexová tabulka ve standardním tvaru.

$$
\begin{array}{cccccc|c}
0 & -2 & 1 & 0 & 0 & -3 & -1 \\
\hline
0 & 2 & 6 & 1 & 0 & 4 & 4 \\
1 & 1 & 3 & 0 & 0 & 2 & 3 \\
0 & -1 & 1 & 0 & 1 & 2 & 1
\end{array}
$$

Postup iterace:

Iterace simplexového algoritmu

Cíl iterace: Přejít k sousední bázi tak, aby bázové řešení zůstalo přípustné a účelová funkce se nezhoršila.

Vstup i výstup iterace je simplexová tabulka ve standardním tvaru.

$$
\begin{array}{rrrrrr|r}
0 & -2 & 1 & 0 & 0 & -3 & -1 \\
\hline
0 & 2 & 6 & 1 & 0 & 4 & 4 \\
1 & 1 & 3 & 0 & 0 & 2 & 3 \\
0 & -1 & 1 & 0 & 1 & 2 & 1
\end{array}
$$

Postup iterace:

1. Zvol sloupec j tak, že $c_j < 0$.
   Sloupec j vstoupí do báze. Podmínka zaručuje, že účelová funkce se nezvětší (ideálně zmenší).

Iterace simplexového algoritmu

Cíl iterace: Přejít k sousední bázi tak, aby bázové řešení zůstalo přípustné a účelová funkce se nezhoršila.

Vstup i výstup iterace je simplexová tabulka ve standardním tvaru.

$$\begin{array}{rrrrrr|r} 0 & -2 & 1 & 0 & 0 & -3 & -1 \\ \hline 0 & 2 & 6 & 1 & 0 & 4 & 4 \\ 1 & 1 & 3 & 0 & 0 & 2 & 3 \\ 0 & -1 & 1 & 0 & 1 & 2 & 1 \end{array}$$

Postup iterace:

1. Zvol sloupec j tak, že $c_j < 0$.
   Sloupec j vstoupí do báze. Podmínka zaručuje, že účelová funkce se nezvětší (ideálně zmenší).
2. Zvol řádek $i \in \operatorname*{argmin}_{i' : a_{i'j} > 0} \dfrac{b_{i'}}{a_{i'j}}$. Prvek (i, j) se nazývá **pivot**.
   Bázový sloupec, který má v pivotovém řádku jedničku, opustí bázi.
   Podmínka zaručuje, že nové bázové řešení bude přípustné.

Iterace simplexového algoritmu

Cíl iterace: Přejít k sousední bázi tak, aby bázové řešení zůstalo přípustné a účelová funkce se nezhoršila.

Vstup i výstup iterace je simplexová tabulka ve standardním tvaru.

$$
\begin{array}{cccccc|c}
0 & -2 & 1 & 0 & 0 & -3 & -1 \\
\hline
0 & 2 & 6 & 1 & 0 & 4 & 4 \\
1 & 1 & 3 & 0 & 0 & 2 & 3 \\
0 & -1 & 1 & 0 & 1 & 2 & 1
\end{array}
$$

Postup iterace:

1. Zvol sloupec j tak, že $c_j < 0$.
   Sloupec j vstoupí do báze. Podmínka zaručuje, že účelová funkce se nezvětší (ideálně zmenší).

2. Zvol řádek $i \in \operatorname*{argmin}_{i' :\, a_{i'j} > 0} \dfrac{b_{i'}}{a_{i'j}}$. Prvek (i, j) se nazývá **pivot**.
   Bázový sloupec, který má v pivotovém řádku jedničku, opustí bázi.
   Podmínka zaručuje, že nové bázové řešení bude přípustné.

3. Ekvivalentními úpravami nastav pivot na jedničku a prvky nad a pod pivotem na nulu.
   Bázový sloupec, který se tím 'zničí', bázi opouští.

Iterace simplexového algoritmu

Cíl iterace: Přejít k sousední bázi tak, aby bázové řešení zůstalo přípustné a účelová funkce se nezhoršila.

Vstup i výstup iterace je simplexová tabulka ve standardním tvaru.

$$
\begin{array}{cccccc|c}
0 & -2 & 1 & 0 & 0 & -3 & -1 \\
\hline
0 & 2 & 6 & 1 & 0 & 4 & 4 \\
1 & 1 & 3 & 0 & 0 & 2 & 3 \\
0 & -0.5 & 0.5 & 0 & 0.5 & 1 & 0.5
\end{array}
$$

Postup iterace:

1. Zvol sloupec j tak, že $c_j < 0$.
   Sloupec j vstoupí do báze. Podmínka zaručuje, že účelová funkce se nezvětší (ideálně zmenší).

2. Zvol řádek $i \in \operatorname*{argmin}_{i' \,:\, a_{i'j} > 0} \dfrac{b_{i'}}{a_{i'j}}$. Prvek (i,j) se nazývá **pivot**.
   Bázový sloupec, který má v pivotovém řádku jedničku, opustí bázi.
   Podmínka zaručuje, že nové bázové řešení bude přípustné.

3. Ekvivalentními úpravami nastav pivot na jedničku a prvky nad a pod pivotem na nulu.
   Bázový sloupec, který se tím ‘zničí’, bázi opouští.

Iterace simplexového algoritmu

Cíl iterace: Přejít k sousední bázi tak, aby bázové řešení zůstalo přípustné a účelová funkce se nezhoršila.

Vstup i výstup iterace je simplexová tabulka ve standardním tvaru.

$$
\begin{array}{cccccc|c}
0 & -2 & 1 & 0 & 0 & -3 & -1 \\
\hline
0 & 2 & 6 & 1 & 0 & 4 & 4 \\
1 & 2 & 2 & 0 & -1 & 0 & 2 \\
0 & -0.5 & 0.5 & 0 & 0.5 & 1 & 0.5
\end{array}
$$

Postup iterace:

1. Zvol sloupec j tak, že $c_j < 0$.
   Sloupec j vstoupí do báze. Podmínka zaručuje, že účelová funkce se nezvětší (ideálně zmenší).

2. Zvol řádek $i \in \underset{i' \,:\, a_{i'j} > 0}{\operatorname{argmin}} \dfrac{b_{i'}}{a_{i'j}}$. Prvek (i, j) se nazývá **pivot**.
   Bázový sloupec, který má v pivotovém řádku jedničku, opustí bázi.
   Podmínka zaručuje, že nové bázové řešení bude přípustné.

3. Ekvivalentními úpravami nastav pivot na jedničku a prvky nad a pod pivotem na nulu.
   Bázový sloupec, který se tím ‘zničí’, bázi opouští.

Iterace simplexového algoritmu

Cíl iterace: Přejít k sousední bázi tak, aby bázové řešení zůstalo přípustné a účelová funkce se nezhoršila.

Vstup i výstup iterace je simplexová tabulka ve standardním tvaru.

$$
\begin{array}{rrrrrr|r}
0 & -2 & 1 & 0 & 0 & -3 & -1 \\
\hline
0 & 4 & 4 & 1 & -2 & 0 & 2 \\
1 & 2 & 2 & 0 & -1 & 0 & 2 \\
0 & -0.5 & 0.5 & 0 & 0.5 & 1 & 0.5
\end{array}
$$

Postup iterace:

1. Zvol sloupec j tak, že $c_j < 0$.
   Sloupec j vstoupí do báze. Podmínka zaručuje, že účelová funkce se nezvětší (ideálně zmenší).

2. Zvol řádek $i \in \operatorname*{argmin}_{i' \,:\, a_{i'j} > 0} \dfrac{b_{i'}}{a_{i'j}}$. Prvek (i, j) se nazývá **pivot**.
   Bázový sloupec, který má v pivotovém řádku jedničku, opustí bázi.
   Podmínka zaručuje, že nové bázové řešení bude přípustné.

3. Ekvivalentními úpravami nastav pivot na jedničku a prvky nad a pod pivotem na nulu.
   Bázový sloupec, který se tím 'zničí', bázi opouští.

Iterace simplexového algoritmu

Cíl iterace: Přejít k sousední bázi tak, aby bázové řešení zůstalo přípustné a účelová funkce se nezhoršila.

Vstup i výstup iterace je simplexová tabulka ve standardním tvaru.

$$
\begin{array}{rrrrrr|r}
0 & -3.5 & 2.5 & 0 & 1.5 & 0 & 0.5 \\
\hline
0 & 4 & 4 & 1 & -2 & 0 & 2 \\
1 & 2 & 2 & 0 & -1 & 0 & 2 \\
0 & -0.5 & 0.5 & 0 & 0.5 & 1 & 0.5
\end{array}
$$

Postup iterace:

1. Zvol sloupec j tak, že $c_j < 0$.
   Sloupec j vstoupí do báze. Podmínka zaručuje, že účelová funkce se nezvětší (ideálně zmenší).

2. Zvol řádek $i \in \operatorname*{argmin}_{i' : \, a_{i'j} > 0} \dfrac{b_{i'}}{a_{i'j}}$. Prvek (i, j) se nazývá **pivot**.
   Bázový sloupec, který má v pivotovém řádku jedničku, opustí bázi.
   Podmínka zaručuje, že nové bázové řešení bude přípustné.

3. Ekvivalentními úpravami nastav pivot na jedničku a prvky nad a pod pivotem na nulu.
   Bázový sloupec, který se tím 'zničí', bázi opouští.

Ukončení algoritmu

Algoritmus končí, když je splněna jedna z podmínek:

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

$$
\begin{array}{cccccc|c}
0 & 0 & 7 & 1 & 0 & 1 & 4 \\
\hline
0 & 1 & 3 & 0.5 & 0 & 2 & 2 \\
1 & 0 & 0 & -0.5 & 0 & 0 & 1 \\
0 & 0 & 4 & 0.5 & 1 & 4 & 3
\end{array}
$$

Ukončení algoritmu

Algoritmus končí, když je splněna jedna z podmínek:

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

$$\begin{array}{rrrrrr|r}
0 & 0 & 7 & 1 & 0 & 1 & 4 \\
\hline
0 & 1 & 3 & 0.5 & 0 & 2 & 2 \\
1 & 0 & 0 & -0.5 & 0 & 0 & 1 \\
0 & 0 & 4 & 0.5 & 1 & 4 & 3
\end{array}$$

- Existuje sloupec, který je celý záporný (úloha je neomezená).

$$\begin{array}{rrrrrr|r}
0 & 0 & 7 & -1 & 0 & 1 & 4 \\
\hline
0 & 1 & 3 & -0.5 & 0 & 2 & 2 \\
1 & 0 & 0 & -0.5 & 0 & 0 & 1 \\
0 & 0 & 4 & -0.5 & 1 & 4 & 3
\end{array}$$

Příklad

0	−2	1	0	0	−3	−1
0	2	6	1	0	4	4
1	1	3	0	0	2	3
0	−1	1	0	1	$\boxed{2}$	1

Příklad

$$
\begin{array}{cccccc|c}
0 & -2 & 1 & 0 & 0 & -3 & -1 \\
\hline
0 & 2 & 6 & 1 & 0 & 4 & 4 \\
1 & 1 & 3 & 0 & 0 & 2 & 3 \\
0 & -1 & 1 & 0 & 1 & \boxed{2} & 1
\end{array}
$$

$$
\begin{array}{cccccc|c}
0 & -3.5 & 2.5 & 0 & 1.5 & 0 & 0.5 \\
\hline
0 & \boxed{4} & 4 & 1 & -2 & 0 & 2 \\
1 & 2 & 2 & 0 & -1 & 0 & 2 \\
0 & -0.5 & 0.5 & 0 & 0.5 & 1 & 0.5
\end{array}
$$

Příklad

0	−2	1	0	0	−3	−1
0	2	6	1	0	4	4
1	1	3	0	0	2	3
0	−1	1	0	1	$\boxed{2}$	1

0	−3.5	2.5	0	1.5	0	0.5
0	$\boxed{4}$	4	1	−2	0	2
1	2	2	0	−1	0	2
0	−0.5	0.5	0	0.5	1	0.5

0	0	6	0.875	−0.25	0	2.25
0	1	1	0.25	−0.5	0	0.5
1	0	0	−0.5	0	0	1
0	0	1	0.125	$\boxed{0.25}$	1	0.75

0	−2	1	0	0	−3	−1
0	2	6	1	0	4	4
1	1	3	0	0	2	3
0	−1	1	0	1	$\boxed{2}$	1

0	−3.5	2.5	0	1.5	0	0.5
0	$\boxed{4}$	4	1	−2	0	2
1	2	2	0	−1	0	2
0	−0.5	0.5	0	0.5	1	0.5

0	0	6	0.875	−0.25	0	2.25
0	1	1	0.25	−0.5	0	0.5
1	0	0	−0.5	0	0	1
0	0	1	0.125	$\boxed{0.25}$	1	0.75

0	0	7	1	0	1	3
0	1	3	0.5	0	2	2
1	0	0	−0.5	0	0	1
0	0	4	0.5	1	4	3

Degenerace a cyklení

Bázové řešení $\mathbf{x}$ je **degenerované**, jestliže odpovídá více než jedné bázi.

To nastane právě tehdy, když nějaká bázová složka vektoru $\mathbf{x}$ je nulová.

V tom případě se simplexový algoritmus může **zacyklit**.

Degenerace a cyklení

Bázové řešení **x** je **degenerované**, jestliže odpovídá více než jedné bázi.

To nastane právě tehdy, když nějaká bázová složka vektoru **x** je nulová.

V tom případě se simplexový algoritmus může **zacyklit**.

$$
\begin{array}{rrrrrr|r}
-2.3 & -2.15 & 13.55 & 0.4 & 0 & 0 & 0 \\
\hline
\boxed{0.4} & 0.2 & -1.4 & -0.2 & 1 & 0 & 0 \\
-7.8 & -1.4 & 7.8 & 0.4 & 0 & 1 & 0
\end{array}
$$

$$
\begin{array}{rrrrrr|r}
0 & -1 & 5.5 & -0.75 & 5.75 & 0 & 0 \\
\hline
1 & 0.5 & -3.5 & -0.5 & 2.5 & 0 & 0 \\
0 & \boxed{2.5} & -19.5 & -3.5 & 19.5 & 1 & 0
\end{array}
$$

$$
\begin{array}{rrrrrr|r}
0 & 0 & -2.3 & -2.15 & 13.55 & 0.4 & 0 \\
\hline
1 & 0 & \boxed{0.4} & 0.2 & -1.4 & -0.2 & 0 \\
0 & 1 & -7.8 & -1.4 & 7.8 & 0.4 & 0
\end{array}
$$

Sloupce tabulky jen zrotovaly o dvě doprava $\Longrightarrow$ další 4 iterace dospějí do počátečního stavu!

Degenerace a cyklení

Bázové řešení **x** je **degenerované**, jestliže odpovídá více než jedné bázi.

To nastane právě tehdy, když nějaká bázová složka vektoru **x** je nulová.

V tom případě se simplexový algoritmus může **zacyklit**.

$$\begin{array}{rrrrrr|r}
-2.3 & -2.15 & 13.55 & 0.4 & 0 & 0 & 0 \\ \hline
\boxed{0.4} & 0.2 & -1.4 & -0.2 & 1 & 0 & 0 \\
-7.8 & -1.4 & 7.8 & 0.4 & 0 & 1 & 0
\end{array}$$

$$\begin{array}{rrrrrr|r}
0 & -1 & 5.5 & -0.75 & 5.75 & 0 & 0 \\ \hline
1 & 0.5 & -3.5 & -0.5 & 2.5 & 0 & 0 \\
0 & \boxed{2.5} & -19.5 & -3.5 & 19.5 & 1 & 0
\end{array}$$

$$\begin{array}{rrrrrr|r}
0 & 0 & -2.3 & -2.15 & 13.55 & 0.4 & 0 \\ \hline
1 & 0 & \boxed{0.4} & 0.2 & -1.4 & -0.2 & 0 \\
0 & 1 & -7.8 & -1.4 & 7.8 & 0.4 & 0
\end{array}$$

Sloupce tabulky jen zrotovaly o dvě doprava $\Longrightarrow$ další 4 iterace dospějí do počátečního stavu!

Blandovo anticyklící pravidlo:

- Při výběru pivotového sloupce: z přípustných sloupců vyber ten s nejnižším indexem.
- Při výběru pivotového řádku: z přípustných řádků vyber ten s nejnižším indexem.

Inicializace algoritmu (dvoufázová simplexová metoda)

Jak uvést libovolnou simplexovou tabulku do standardního tvaru?

Každou úlohu LP snadno převedeme na tvar

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

kde $\mathbf{A}$ má lin. nezávislé řádky a $\mathbf{b} \geq \mathbf{0}$. Ovšem $\mathbf{A}$ nemusí obsahovat standardní bázi!

Inicializace algoritmu (dvoufázová simplexová metoda)

Jak uvést libovolnou simplexovou tabulku do standardního tvaru?

Každou úlohu LP snadno převedeme na tvar

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

kde $\mathbf{A}$ má lin. nezávislé řádky a $\mathbf{b} \geq \mathbf{0}$. Ovšem $\mathbf{A}$ nemusí obsahovat standardní bázi!

Řešení: Vyřeš nejprve pomocnou úlohu

$$\min\{\, \mathbf{1}^T\mathbf{u} \mid \mathbf{A}\mathbf{x} + \mathbf{u} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0},\ \mathbf{u} \geq \mathbf{0} \,\}$$

se simplexovou tabulkou

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

Inicializace algoritmu (dvoufázová simplexová metoda)

Jak uvést libovolnou simplexovou tabulku do standardního tvaru?

Každou úlohu LP snadno převedeme na tvar

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

kde $\mathbf{A}$ má lin. nezávislé řádky a $\mathbf{b} \geq \mathbf{0}$. Ovšem $\mathbf{A}$ nemusí obsahovat standardní bázi!

Řešení: Vyřeš nejprve pomocnou úlohu

$$\min\{\, \mathbf{1}^T\mathbf{u} \mid \mathbf{A}\mathbf{x} + \mathbf{u} = \mathbf{b},\ \mathbf{x} \geq \mathbf{0},\ \mathbf{u} \geq \mathbf{0} \,\}$$

se simplexovou tabulkou

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

Platí $(\mathbf{1}^T\mathbf{u} = 0) \iff (\mathbf{u} = \mathbf{0}) \iff (\mathbf{A}\mathbf{x} = \mathbf{b})$.

- Jestliže je vstupní úloha nepřípustná, pak pomocná úloha má opt. hodnotu kladnou.
- Jestliže je vstupní úloha přípustná, pak pomocná úloha má opt. hodnotu nulovou.
  Pak, jestliže je opt. řešení pomocné úlohy nedegenerované, matice $\mathbf{A}$ obsahuje standardní bázi.