Kapitola 14

Lineární programování

Vraťme se k problému sestavení ideální štafety. V něm jsme měli úplný bipartitní graf, každou hranu ohodnocenou nějakým reálným číslem a hledali jsme perfektní párování s co nejmenším součtem hodnot hran. Tento problém můžeme obecně formulovat následovně.

Je dán úplný bipartitní graf $G = (V, E)$, to znamená, že $V = L\dot{\cup}R$ je disjuntkní sjednocení dvou množin téže mohutnosti n a množina n^2 neorientovaných hran $E = \{\{v, w\} : v \in L, w \in R\}$. Dále je dána funkce $\omega : E \to \mathbf{R}$. Hodnotu $\omega(e)$ budeme zapisovat také jako ω_e. *Minimální perfektní párování* v grafu G je nějaká množina n hran M taková, že

- žádné dvě hrany M neobsahují společný vrchol,
- součet $\sum_{e\in M} \omega_e$ je minimální.

Tuto úlohu přeformulujeme pomocí charakteristických vektorů. Každou podmnožinu $F \subseteq E$ můžme popsat pomocí reálného vektoru $\mathbf{x}_F = (x_e)_{e\in E}$ dimenze n^2 se všemi souřadnicemi $x_e \in \{0, 1\}$, pro které platí

$$x_e = 1 \text{ právě když } e \in F.$$

Množina $F \subseteq E$ je perfektní párování v grafu G právě když pro vektor $\mathbf{x}_F = (x_e)_{e\in E}$ platí

$$\sum_{v\in e} x_e = 1 \text{ pro každý vrchol } v \in V.$$

Jinak řečeno, každý vrchol $v \in V$ je obsažen v právě jedné hraně $e \in F$.

Součet hodnot hran v množině F se rovná

$$\sum_{e\in E} \omega_e x_e.$$

Úlohu nalezení perfektního párování s minimálním součtem hodnot tak může být formulován následovně jako úloha na nalezení n^2 reálných čísel x_e splňujících podmínky součet

- součet $\sum_{e\in E}\omega_e x_e$ je minimální,
- $\sum_{v\in e} x_e = 1$ pro každý vrchol $v \in V$,
- $x_e \in \{0,1\}$ pro každou hranu $e \in E$.

Lineární funkce $\sum_{e\in E}\omega_e x_e$ se nazývá *účelová funkce*, dalším podmínkám říkáme *omezující podmínky* nebo jenom *omezení.* Protože omezující podmínky říkají, že každá proměnná x_e musí nabývat celočíselných hodnot, říkáme že jde o úlohu *celočíselného lineárního programování.*

Tímto přeformulováním získáme *dolní odhad* pro naši úlohu. Získáme jej tak, že oslabíme třetí podmínku $x_e \in \{0,1\}$ pro každou hranu $e \in E$ na $0 \le x_e \le 1$ pro každou hranu $e \in E$. Dostaneme tak úlohu *lineárního programování*

- součet $\sum_{e\in E}\omega_e x_e$ je minimální,
- $\sum_{v\in e} x_e = 1$ pro každý vrchol $v \in V$,
- $0 \le x_e \le 1$ pro každou hranu $e \in E$.

Protože každý vektor $(x_e)_{e\in E}$ splňující omezení úlohy celočíselného lineárního programování rovněž splňuje všechny podmínky příslušné úlohy lineárního programování, platí, že minimální hodnota účelové funkce u úlohy lineárního programování je menší nebo rovná minimální hodnotě vytvořující funkce odpovídající úlohy celočíselného lineárního programování.

Věta 14.1 *Pro úlohu nalezení minimálního perfektního paárování v bipartitním grafu $G = (V,E)$ existuje optimální řešení $(x_e)_{e\in E}$ příslušné úlohy lineárního programování, jehož všechny souřadnice x_e jsou celočíselné.*

Důkaz. Na přednášce. □

Předcházející tvrzení je spíš výjimečné. V naprosté většině případů je optimální řešení úlohy celočíselného lineárního programování různé od optimálního řešení odpovídající úlohy lineárního programování.

Úloha 14.1 *Nalezení maximální nezávislé množiny vrcholů v neorientovaném grafu.*

Řešení. Na přednášce. □

Simplexová metoda

Lineární programování (LP) ve své nejobecnější podobě je úloha jak maximalizovat nějakou lineární funkci v n proměnných $x_1, \ldots, x_n$, které musí splňovat m lineárních nerovností. Pokud navíc požadujeme, aby $x_i \geq 0$, říkáme, že úloha LP je ve standardním tvaru. Ten můžeme zapsat následovně:

- maximalizujte funkci $\sum_{j=1}^{n} c_j x_j$
- za předpokladů $\sum_{j=1}^{n} a_{ij} x_j \leq b_j$ pro $i = 1, 2, \ldots, m$
- a dále $x_j \geq 0$ pro $j = 1, 2, \ldots, n$,

kde c_j, b_i a a_{ij} jsou reálná čísla.

Označíme-li

$$\begin{aligned} \mathbf{x} &= (x_1, \ldots, x_n)^T, \\ \mathbf{c} &= (c_1, \ldots, c_n)^T, \\ \mathbf{b} &= (b_1, \ldots, b_m)^T, \\ \mathbf{A} &= \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \end{aligned}$$

můžeme úlohu lineárního programování zapsat kompaktněji v podobě

- maximalizujte funkci $\mathbf{c}^T\mathbf{x}$
- vzhledem k omezením $\mathbf{A}\mathbf{x} \leq \mathbf{b}$
- a $\mathbf{x} \geq \mathbf{0}$.

Nerovnost $\mathbf{x} \leq \mathbf{y}$ mezi dvěma vektory téže délky platí právě když platí pro každou souřadnici obou vektorů.

Vektor $\mathbf{c}$ se nazývá *cenový vektor*, funkce $\mathbf{x} \mapsto \mathbf{c}^T\mathbf{x}$ se nazývá *účelová funkce*, nerovnosti $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ a $\mathbf{x} \geq \mathbf{0}$ se nazývají *omezující podmínky* nebo také *omezení.* Vektor $\mathbf{b}$ se nazývá *pravá strana* úlohy. Úloha LP se nazývá *přípustná*, pokud existuje nezáporný vektor $\mathbf{x}$ vyhovující omezujícím podmínkám $\mathbf{A}\mathbf{x} \leq \mathbf{b}$. Takovému vektoru $\mathbf{x}$ pak říkáme *přípustné řešení.* Pokud žádné přípustné řešení úlohy LP neexistuje, pak říkáme, že taková úloha je *nepřípustná.* Úloha je *neomezená*, pokud existuji přípustná řešení, pro která účelová funkce nabývá libovolně velkých hodnot. V opačném případě se nazývá *omezená.* Pro úlohu LP, která je současně přípustná a omezená, existuje jednoznačně určené maximum účelové funkce $\mathbf{c}_T\mathbf{x}$. Ta jej dosahuje v nějakém vektoru $\mathbf{x}$, kterému říkáme *optimální řešení.* Optimální řešení nemusí být určené jednoznačně.

Tabulky

Zavedení nových proměnných

$$x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij}x_j$$

spolu s omezeními $x_{n+i} \geq 0$ pro $i = 1, \ldots, m$.

Původní úlohu LP upravíme tímto do tvaru

- maximalizujte funkci $\sum_{j=1}^{n} c_j x_j$
- vzhledem k omezením $x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij}x_j$ pro $i = 1, \ldots, m$
- a dále $x_j \geq 0$ pro $j = 1, 2, \ldots, m + n$.

V kompaktním tvaru tutéž úlohu zapíšeme ve tvaru

- maximalizujte funkci $\mathbf{d}^T\mathbf{x}$
- vzhledem k omezením $\mathbf{Bx} = \mathbf{b}$
- a dále $\mathbf{x} \geq \mathbf{0}$.

Zde platí

$$\mathbf{B} = (\mathbf{A}|\mathbf{I}_m),$$

dále $\mathbf{d}$ je vektor dimenze $n + m$ tvaru $(\mathbf{c}|\mathbf{0})^T$ a $\mathbf{x}$ je sloupcový vektor $m + n$ neznámých

$$\mathbf{x} = (\mathbf{x}_O|\mathbf{x}_S)^T,$$

kde $\mathbf{x}_O = (x_1, \ldots, x_n)$ a $\mathbf{x}_S = (x_{n+1}, \ldots, x_{n+m})$.

Vyjádření nových proměnných $x_{n+i} = b_i - \sum_{j=1}^{n} a_{ij}x_j$ pro $i = 1, \ldots, m$ spolu s účelovou funkcí obsahuje všechny informace o úloze. Tradičně se tyto informace zapisují do tabulky.

Úloha 14.2 *Maximalizujte funkci $x_1 + x_2$ vzhledem k omezením $-x_1 + x_2 \leq 1$, $x_1 \leq 3$, $x_2 \leq 2$, $x_1, x_2 \geq 0$.*

Řešení. Úvodní tabulka k této úloze je

$$
\begin{array}{ccccccc}
x_3 & = & 1 & + & x_1 & - & x_2 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 1 & & & - & x_2 \\
\hline
z & = & & & x_1 & + & x_2
\end{array}
$$

Položíme-li proměnné na pravých stranách rovné 0, hodnoty proměnných na levé straně nabývají hodnot $x_3 = 1$, $x_4 = 3$, $x_5 = 2$. Vektor $(0, 0, 1, 3, 2)$ je proto přípustné řešení. Tomuto řešení říkáme *základní přípustné řešení* a jeho tabulka je *přípustná tabulka*. Účelová funkce z nabývá hodnoty $z = 0$. Proměnným na levé straně říkáme *bázové proměnné*, proměnné na pravé straně jsou *nebázové proměnné*.

Nyní budeme postupně tabulku upravovat tak, abychom dostali nové baázové přípustné řešení s vyšší hodnotou účelové funkce z nebo abychom dokázali, že neexistuje žádné přípustné řešení s vyšší hodnotou z.

V první tabulce si všimneme, že zvětšení hodnoty x_1 nebo x_2 zvýší hodnotu účelové funkce z. Obě proměnné mají kladné znaménko ve formulce pro účelovou funkci. Jak moc ale můžeme zvětšit např. x_2 ? Chceme-li zachovat přípustnost řešení, můžeme kvůli první rovnici zvětšit x_2 nejvýše na hodnotu 1. Druhá rovnice nijak hodnotu x_2 neomezuje. Třetí rovnice omezuje x_2 nejvýše na hodnotu 2. Největší omezení tak pochází z první rovnice. Z ní tedy vypočítáme hodnotu

$$x_2 = 1 + x_1 - x_3$$

a dosadíme ji do ostatních rovnic a do účelové funkce. Dostaneme tak následující tabulku

$$
\begin{array}{ccccccc}
x_2 & = & 1 & + & x_1 & - & x_3 \\
x_4 & = & 3 & - & x_1 & & \\
x_5 & = & 1 & - & x_1 & + & x_3 \\
\hline
z & = & 1 & + & 2x_1 & - & x_3
\end{array}
$$

Je zřejmé, že oba systémy rovnic mají stejné množiny řešení. Původně nebázová proměnná x_2 byla přesunuta do báze, ze které naopak byla odsunuta proměnná x_3. Položíme-li opět hodnoty nebázových proměnných x_1, x_3 rovné nule, dostaneme $x_2 = 1$, $x_4 = 3$, $x_5 = 2$. Dostáváme tak nové bázové přípustné řešení $(0, 1, 0, 3, 1)$ a účelová funkce z nyní nabývá hodnoty $z = 1$.

V nové tabulce můžeme stále ještě zvýšit hodnotu x_1. Z první rovnice žádné omezení pro hodnotu x_1 nevyplývá, z druhé rovnice dostáváme $x_1 \leq 3$ a ze tgřetí rovnice vyplývá omezení $x_1 \leq 1$. Největší omezení tak vyplývá ze třetí rovnice, ze které vyjádříme x_1 a dosadíme je do zbývajících rovnic. Dostaneme tak další tabulku

$$\begin{array}{rcrcrcr}
x_2 & = & 2 & & & - & x_5 \\
x_4 & = & 2 & - & x_3 & + & x_5 \\
x_1 & = & 1 & + & x_3 & - & x_5 \\
\hline
z & = & 3 & + & x_3 & - & 2x_5
\end{array}$$

s odpovídajícím bázovým přípustným řešením $(1, 2, 0, 2, 0)$ a hodnotou $z = 3$. Provedeme další úpravu tabulky, při které se proměnná x_3 přesune mezi bázové proměnné a x_4 naopak bázi opustí. Dostaneme tak tabulku

$$\begin{array}{rcrcrcr}
x_2 & = & 2 & & & - & x_5 \\
x_3 & = & 2 & - & x_4 & + & x_5 \\
x_1 & = & 3 & - & x_4 & & \\
\hline
z & = & 5 & - & x_4 & - & x_5
\end{array}$$

s odpovídajícím bázovým přípustným řešením $(3, 2, 2, 0, 0)$ hodnotou účelové funkce $z = 5$. V této tabulce nemůžeme žádnou nebázovou proměnnou zvětšit aniž bychom nesnížili hodnotu účelové funkce z. Žádnou další analogickou úpravu tabulky proto nemůžeme udělat. Naštěstí to znamená, že už jsme našli optimální řešení. To si můžeme snadno ověřit. Je-li $(x'_1 x'_2 x'_3 x'_4 x'_5)$ libovolné přípustné řešení první tabulky s hodnotou účelové funkce z', pak je rovněž přípustným řešením poslední tabulky, ze které vyplývá

$$z' = 5 - x'_4 - x'_5.$$

Protože navíc musí platit $x'_4 \geq 0$ a $x'_5 \geq 0$, dostáváme tak $z' \leq 5$. □

Geometrická interpretace

Omezující podmínky úlohy 14.2 můžeme znázornit geometricky. Každá podmínka (včetně podmínek $x_i \geq 0$ pro $i = 1, 2$) určuje uzavřenou polorovinu v rovině. Jejich průnik je uzavřený konvexní mnohoúhelník P všech přípustných řešení. Kvůli podmínkám $x_i \geq 0$ pro $i = 1, 2$ je tento mnohoúhelník obsažený v prvním kvadrantu. Maximalizovat účelovou funkci $z = x_1 + x_2$ znamená najít ten bod (x_1, x_2) mnohoúhelníku přípustných řešení, pro který má fukce $z = x_1 + x_2$ maximální možnou hodnotu.

Obecná úloha lineárního programování je ve tvaru

- maximalizujte funkci $\sum_{j=1}^{n} c_j x_j$
- za předpokladů $\sum_{j=1}^{n} a_{ij} x_j \leq b_j$ pro $i = 1, 2, \ldots, m$
- a dále $x_j \geq 0$ pro $j = 1, 2, \ldots, n$,

kde c_j, b_i a a_{ij} jsou reálná čísla.

Každá nerovnost

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_j \text{ pro } \ i = 1, 2, \ldots, m$$

popisuje uzavřený poloprostor v n-dimezionálním aritmetickém prostoru $\mathbf{R}^n$. Množina P všech přípustných řešení je tak průnikem $m+n$ uzavřených poloprostorů. Průnik konečně mnoha uzavřených poloprostorů v $\mathbf{R}^n$ nazýváme *konvexní polytop*.

Co s výjimkami?

Neomezenost Při jednom kroku simplexového algoritmu hledáne nebázovou proměnnou, kterou můžeme zvětšit tak, abychom nějakou bázovou proměnnou zmenšili na 0. Může se ale stát, že žádná taková bázová proměnná neexistuje. Uvažujme úlohu maximalizovat účelovou funkci $z = x_1$ za podmínek $x_1 - x_2 \leq 1$ a $-x_1 + x_2 \leq 2$ (a samozřejmě $x_1, x_2 \geq 0$). Počáteční tabulka této úlohy je

$$\begin{array}{ccccccc} x_3 & = & 1 & - & x_1 & + & x_2 \\ x_4 & = & 2 & + & x_1 & - & x_2 \\ \hline z & = & & & x_1 & & \end{array}$$

V prvním kroku simplexového algoritmu přesuneme do báze proměnnou x_1. Dostaneme tak tabulku

$$\begin{array}{ccccccc} x_1 & = & 1 & + & x_2 & - & x_3 \\ x_4 & = & 3 & & & - & x_3 \\ \hline z & = & 1 & + & x_2 & - & x_3 \end{array}$$

Pokusíme-li se nyní přesunout do báze proměnnou x_2 tak zjistíme, že že žádná z rovnic nedává žádné omezení na její velikost. Jak x_2 tak i účelová funkce z mohou nabývat libovolně velkých hodnot, úloha je neomezenná. Podud zvětšujeme x_2 do nekonečna, dostaneme celou polopřímku přípustných řešení s počátkem v současném bázovém přípustném řešení $(1, 0, 0, 3)$. Je to množina

$$\{(1, 0, 0, 3) + x_2(1, 1, 0, 0) : x_2 \geq 0\}.$$

Příčinnou je neomezenost polytopu přípustných řešení P.

Degenerovanost V degenerovaném případě nastává opačný extrémní případ – některé rovnice tabulky způsobují, že nemůžeme žádnou nebázovou proměnnou zvětšit a žádný nárůst účelové funkce tak není možný. Ukážeme si to opět na příkladu. Tentokrát se budeme snažit maximalizovat účelovou funkci $z = x_2$ za podmínek $-x_1 + x_2 \leq 0$ a $x_1 \leq 2$ a $x_1, x_2 \geq 2$. Počáteční tabulka je

$$\begin{array}{rclcccc}
x_3 & = & & & x_1 & - & x_2 \\
x_4 & = & 2 & - & x_1 & & \\
\hline
z & = & & & & & x_2
\end{array}$$

Bázové přípustné řešení se rovná $(0, 0, 0, 2)$. Jediným kandidátem na přesunutí do báze je proměnná x_2. První rovnice ale říká, že ji nemůžeme zvětšit aniž bychom učinili proměnnou x_3 zápornou. Tato situace nastane vždy kdy v bázovém přípustném řešení se některá z bázových proměnných rovná 0. Takový případ nazýváme *degenerovaný*. Nemožnost zvýšení hodnoty účelové funkce ale neznamená optimalitu bázového přípustného řešení. V našem případě přesunutím proměnné x_2 do báze dostaneme další degenerovanou tabulku se stejným bázovým přípustným řešením:

$$\begin{array}{rclcccc}
x_2 & = & & & x_1 & - & x_3 \\
x_4 & = & 2 & - & x_1 & & \\
\hline
z & = & & & x_1 & - & x_3
\end{array}$$

Situace se přesto zlepšila. Nebázovou proměnnou x_1 llze nyní zvětšit a jejím přesunutím do báze dostaneme závěrečnou tabulku

$$\begin{array}{rclcccc}
x_1 & = & 2 & & & - & x_4 \\
x_2 & = & 2 & - & x_3 & - & x_4 \\
\hline
z & = & 2 & - & x_3 & - & x_4
\end{array}$$

s optimálním bázovým přípustným řešením $(2, 2, 0, 0)$.

V tomto případě bylo možné po jednom kroku, kdy zůstala hodnota účelové funkce nezměněná, udělat další úpravu tabulky, která již hodnotu účelové funkce zvětšila. V obecnosti takových degenerovaných kroků může být více, může dokonce nastat i případ, kdy se celý postup *zacyklí* a po několika degenerovaných krocích dostaneme opět původní tabulku. Jak ukazuje následující lemma, zacyklení je jediná možnost, při které algoritmus neskončí. Nemůže se stát, že bychom dostali nekonečnou posloupnost různých tabulek k dané úloze.

Lemma 14.2 *Úloha lineárního programování*

- *maximalizujte funkci* $\sum_{j=1}^{n} c_j x_j$
- *za předpokladů* $\sum_{j=1}^{n} a_{ij} x_j \leq b_j$ *pro* $i = 1, 2, \ldots, m$
- *a dále* $x_j \geq 0$ *pro* $j = 1, 2, \ldots, n$,

kde c_j, b_i *a* a_{ij} *jsou reálná čísla, má nejvýše* $\binom{m+n}{m}$ *různých tabulek.*

Důkaz. Na přednášce, vychází z toho, že výběr m bázových proměnných určuje tabulku jednoznačně. □

Perturbace

Standardní metoda jak vyloučit zacyklení spočívá v použití mocnin *symbolické proměnné* ϵ, o které předpokládáme, že je kladná a nekonečně malá. S její pomocí upravíme původní úlohu do tvaru

- maximalizujte funkci $\sum_{j=1}^{n} c_j x_j$
- za předpokladů $\sum_{j=1}^{n} a_{ij} x_j \leq b_j + \epsilon^j$ pro $i = 1, 2, \ldots, m$
- a dále $x_j \geq 0$ pro $j = 1, 2, \ldots, n$,

kde c_j, b_i a a_{ij} jsou reálná čísla. Má-li původní úloha přípustné řešení, má je i nová úloha. A naopak, z libovolného řešení nové úlohy dostaneme řešení původní úlohy tak, že položíme $\epsilon = 0$.

S pomocnou symbolickou proměnnou ϵ vypadá počáteční tabulka poslední úlohy následovně.

$$\begin{array}{ccccccccc} x_3 & = & & & \epsilon & + & x_1 & - & x_2 \\ x_4 & = & 2 & + & \epsilon^2 & - & x_1 & & \\ \hline z & = & & & & & & & x_2 \end{array}$$

Po přesunutí proměnné x_2 do báze dostaneme

$$\begin{array}{ccccccccc} x_2 & = & & & \epsilon & + & x_1 & - & x_3 \\ x_4 & = & 2 & + & \epsilon^2 & - & x_1 & & \\ \hline z & = & & & \epsilon & + & x_1 & - & x_3 \end{array}$$

přičemž hodnota účelové funkce se zvýší na ϵ. A po přesunutí x_1 do báze dostaneme tabulku

$$\begin{array}{ccccccc} x_2 & = & 2+\epsilon^2 & + & x_1 & - & x_3 \\ x_4 & = & 2+\epsilon+\epsilon^2 & - & x_3 & - & x_4 \\ \hline z & = & 2+\epsilon+\epsilon^2 & - & x_3 & - & x_4 \end{array}$$

s optimálním bázovým přípustným řešením $(2+\epsilon^2, 2+\epsilon+\epsilon^2, 0, 0)$. Po dosazení $\epsilon = 0$ dostaneme optimální řešení původní úlohy.

Pro výběr proměnné, kterou přesuneme do báze, potřebujeme umět porovnávat hodnoty polynomů v proměnné ϵ. To uděláme pomocí *lexikografického uspořádání.*

Definice 14.3 *Předpokládáme, že* $(a_0, a_1, \ldots, a_n)$ *a* $(b_0, b_1, \ldots, b_n)$ *jsou dvě posloupnosti reálných čísel. Říkáme, že posloupnost* $(a_0, a_1, \ldots, a_n)$ *je* lexikograficky menší *než posloupnost* $(b_0, b_1, \ldots, b_n)$ *jestliže existuje přirozené číslo* $j \in \{0, 1, \ldots, n-1\}$ *takové, že* $a_i = b_i$ *pro všechna* $i = 0, \ldots, j$ *a* $a_{j+1} < b_{j+1}$.

Při výběru proměnné pro přesunutí do báze pak používáme nerovnost

$$\sum_{k=0}^{n} a_k \epsilon^k < \sum_{k=0}^{n} b_k \epsilon^k$$

právě když posloupnost $(a_0, a_1, \ldots, a_n)$ je lexikograficky menší než posloupnost $(b_0, b_1, \ldots, b_n)$.

Nepřípustnost

Aby mohl simplexový algoritmus začít, potřebuje nějakou tabulku, ze které po dosazení $x_i = 0$ za libovolnou nebázovou proměnnou x_i dostaneme přípustné řešení. Říkáme, že tabulka má *přípustný počátek*. Počáteční tabulka úlohy lineárního programování ale přípsutný počátek mít nemusí, jak ukazuje následující úloha.

Maximalizujte funkci $z = -x_2$ vzhledem k podmínkám $-x_1 - x_2 \leq -2$ a $x_1 - x_2 \leq -1$ a $x_1, x_2 \geq 0$. Počáteční tabulka této úlohy je

$$\begin{array}{ccccccc} x_3 & = & -2 & + & x_1 & + & x_2 \\ x_4 & = & -1 & - & x_1 & + & x_2 \\ \hline z & = & & & & - & x_2 \end{array}$$

a po dosazení $x_1 = x_2 = 0$ dostáváme $x_3 = -2$ a $x_4 = -1$, což není přípustné řešení. Počáteční tabulka nemá přípustný počátek.

Tento problém vyřešíme zavedením nové pomocné proměnné x_0 a budeme maximalizovat novou účelovou funkci $w = -x_0$ vzhledem k podmínkám $-x_1 - x_2 - x_0 \leq -2$ a $x_1 - x_2 - x_0 \leq -1$ a $x_0, x_1, x_2 \geq 0$. Tato úloha má přípustné řešení pro dostatečně velké x_0. Modifikovaná úloha má počáteční tabulku

$$\begin{array}{ccccccccc} x_3 & = & -2 & + & x_1 & + & x_2 & + & x_0 \\ x_4 & = & -1 & - & x_1 & + & x_2 & + & x_0 \\ \hline w & = & & & & & & - & x_0 \end{array}$$

Tato tabulka rovněž nemá přípustný počátek, můžeme z ní ale dostat tabulku s přípustným počátkem tak, že zvětšíme dostatečně hodnotu pomocné proměnné x_0. Cílem zvětšování proměnné x_0 tentokrát není zvýšení hodnoty účelové funkce, ale nalezení tabulky s přípustným počátkem. Abychom dostali $x_3 \geq 0$, musí být $x_0 \geq 2$. V tom případě je rovněž $x_4 \geq 1$. Přesuneme pomocnou proměnnou x_0 do báze místo proměnné x_3 a dostaneme novou tabulku pro pomocnou úlohu, tentokrát ale již s přípustným počátkem.

$$\begin{array}{ccccccccc} x_0 & = & 2 & - & x_1 & - & x_2 & + & x_3 \\ x_4 & = & 1 & - & 2x_1 & & & + & x_3 \\ \hline w & = & \text{-2} & + & x_1 & + & x_2 & - & x_3 \end{array}$$

Nyní lze již simplexový algoritmus použít, do báze přesuneme proměnnou x_2 a dostaneme závěrečnou tabulku

$$\begin{array}{rcrcrcrcr} x_2 & = & 2 & - & x_1 & + & x_3 & - & x_0 \\ x_4 & = & 1 & - & 2x_1 & + & x_3 & & \\ \hline w & = & & & & & & - & x_0 \end{array}$$

Bázové přípustné řešení je $(x_0, x_1, x_2, x_3, x_4) = (0, 0, 2, 0, 1)$ a účelová funkce nabývá maximální hodnoty $w = -x_0 = 0$.

Co z toho vyplývá pro řešení původní úlohy maximalizovat funkci $z = -x_2$ vzhledem k podmínkám $-x_1 - x_2 \le -2$ a $x_1 - x_2 \le -1$ a $x_1, x_2 \ge 0$? Protože účelová funkce pomocné úlohy $w = -z_0$ nabývá maximální hodnotu 0, splňují proměnné x_1, x_2, x_3, x_4 z bázového přípustného řešení pomocné úlohy podmínky (nerovnosti) úlohy původní. Ze závěrečné tabulky modifikované úlohy tak dostaneme přípustnou tabulku původní úlohy tak, že vynecháme pomocnou proměnnou x_0 a vyjádříme původní účelovou funkci $z = -x_2$ pomocí nebázových proměnných $z = -x_2 = -2 + x_1 - x_3$. Dostaneme tak tabulku

$$\begin{array}{rcrcrcr} x_2 & = & 2 & - & x_1 & + & x_3 \\ x_4 & = & 1 & - & 2x_1 & + & x_3 \\ \hline w & = & -2 & + & x_1 & - & x_3 \end{array}$$

s bázovým přípustným řešením $(x_1, x_2, x_3, x_4) = (0, 2, 0, 1)$. Od této tabulky můžeme dále postupovat obvyklým způsobem. Abzchom mohli takto získat přípustnou tabulku z konečné tabulky modifikované úlohy, musí být pomocná proměnná x_0 v této tabulce nebázová. To zajistíme tak, že kdykoliv je možné pomocnou proměnnou x_0 nahradit v bázi nějakou jinou proměnnou, tak to uděláme.

Je-li naopak maximální hodnota účelové funkce modifikované úlohy záporná, pak původní úloha nemá žádné přípustné řešení, je neřešitelná.

Podobně se můžme vyrovnat s obecnou úlohou

- maximalizujte funkci $\sum_{j=1}^{n} c_j x_j$
- za předpokladů $\sum_{j=1}^{n} a_{ij} x_j \le b_j$ pro $i = 1, 2, \dots, m$
- a dále $x_j \ge 0$ pro $j = 1, 2, \dots, n$,

kde c_j, b_i a a_{ij} jsou reálná čísla, pokud nemá přípustný počátek. Tuto úlohu modifikujeme zavedením pomocné proměnné x_0 na úlohu

- maximalizujte funkci $-x_0$
- za předpokladů $\sum_{j=1}^{n} a_{ij} x_j - x_0 \le b_j$ pro $i = 1, 2, \dots, m$

- a dále $x_j \geq 0$ pro $j = 0, 1, 2, \dots, n$,

kde c_j, b_i a a_{ij} jsou reálná čísla. Řešením této modifikované úlohy najdeme přípustnou tabulku původní úlohy, pokud nová účelová funkce $w = -x_0$ má maximální hodnotu 0. Pokud je maximální hodnota nové účelové funkce $w = -x_0$ nenulová (záporná), pak původní úloha nemá žádné přípustné řešení.

Simplexový algoritmus si uvedeme pro případ, že už máme nějakou přípustnou tabulku

Algoritmus SIMPLEX(úloha lineárního programování)

sestav počáteční tabulku úlohy $\{\mathbf{x}_B = \mathbf{b} + \mathbf{A}\mathbf{x}_N, z = \mathbf{c}^T\mathbf{x}_N\}$
opt:=false;
neom:=false;
`REPEAT`
    `IF` $\mathbf{c} \leq \mathbf{0}$ `THEN` opt:=true `ELSE`
        `BEGIN`
            s:=min$\{j : c_j > 0\}$;
            `IF` $\mathbf{A}_{*s} \geq \mathbf{0}$ `THEN` neom:=true `ELSE`
                `BEGIN`
                    r:=min$\{k : \frac{b_k}{-a_{ks}} = \min\{\frac{b_j}{-a_{js}} : a_{js} < 0\} : a_{ks} < 0\}$
                    přesuň proměnnou x_{N_s} do báze místo x_{B_r}
                `END`
        `END`
`UNTIL` opt `OR` neom;
`IF` opt `THEN` $\mathbf{x}_B$ je optimální řešení `ELSE` úloha je neomezená

Různá pravidla pro výběr prvku a_{rs}

Největší koeficient c_j, největší nárůst z, Blandovo pravidlo (použito v algoritmu), náhodně.