1 Úvod do celočíselné lineární optimalizace

Martin Branda, verze 6. 5. 2021

1.1 Motivace

Reálné (smíšeně-)celočíselné úlohy se vyskytují v mnoha oblastech, kde pouze reálné (spojité) rozhodovací proměnné nestačí. Jako příklady uveďme

- **Optimalizace portfolia** – celočíselné počty akcií, modelování fixních transakčních nákladů,
- **Rozvrhování** – binární proměnné přiřazující úlohy ke strojům,
- **Úlohy rozvozu** – binární proměnné identifikující následníka na cestě vozidla.

My se budeme věnovat pouze lineárním úlohám s celočíselnými rozhodovacími proměnnými. Existuje však mnoho aplikací, které vedou i na úlohy nelineární. I pro ně je k dispozici mnoho teoretických výsledků a specializovaných výpočetních postupů.

Nejprve se podíváme na motivační úlohu umístění skladů (facility location problem), kde je cílem minimalizovat náklady na výstavbu skladů a zároveň na dopravu z vystavěných skladů k zákazníkům. Jedná se tedy o rozšíření dopravního problému. Parametry úlohy jsou

- $i = 1, \ldots, m$ potenciální sklady/pobočky (facilities),
- $j = 1, \ldots, n$ zákazníci,
- c_{ij} – náklady za dodanou jednotku zákazníkovi j ze skladu i,
- f_i – fixní náklady na postavení skladu i,
- K_i – kapacita skladu i,
- D_j – poptávka zákazníka j.

Rozhodovací proměnné jsou potom

- $x_{ij} \geq 0$ – převezené množství ze skladu i k zákazníkovi j,
- $y_i \in \{0, 1\}$ – binární rozhodovací proměnné, je-li sklad postaven $y_i = 1$, není-li postaven $y_i = 0$.

Úlohu můžeme matematicky formulovat takto

$$\begin{aligned}
\min_{x_{ij}, y_i} \ & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} f_i y_i \\
\text{s.t.} \ & \sum_{j=1}^{n} x_{ij} \leq K_i y_i, \ i = 1, \ldots, m, \\
& \sum_{i=1}^{m} x_{ij} = D_j, \ j = 1, \ldots, n, \\
& x_{ij} \geq 0, \ y_i \in \{0, 1\}.
\end{aligned}$$

Účelová funkce udává náklady na převoz a náklady na postavení skladů. Náklady jsou minimalizovány. První omezení říká, že je-li sklad i postaven, tj. $y_i = 1$, dodávky z něj nemohou překročit jeho kapacitu K_i. Naopak není-li postaven, tj. $y_i = 0$, nemůže z něj být nic dodáno. Druhá omezení zaručují, že je rozvozem uspokojena poptávka D_j u všech zákazníků. Poslední jsou omezení na rozhodovací proměnné: dodaná množství jsou nezáporná, sklad je buď postaven nebo ne.

Takováto úloha se nazývá smíšeně-celočíselná nebo také smíšeně-binární, protože obsahuje reálné (nezáporné) a celočíselné (binární) rozhodovací proměnné. V případě, že je rozvážené zboží nedělitelné, např. krabice, dostaneme přirozený požadavek na celočíselnost proměnných $x_{ij} \in \mathbb{Z}_+$, kde $\mathbb{Z}_+$ značí nezáporná celá čísla.

Do úlohy je možné doplnit další logická omezení, např.

- je postaven nejvýše jeden ze skladů $\{i_1, i_2, i_3\}$:

$$y_{i_1} + y_{i_2} + y_{i_3} \leq 1,$$

- buď je postavena dvojice skladů $\{i_4, i_5\}$ nebo žádný z nich

$$y_{i_4} = y_{i_5}.$$

Obecně binární proměnné umožňují modelovat logické relace.

1.2 Obecné vlastnosti celočíselných úloh

Uvažujme obecnou celočíselnou úlohu

$$\min c^T x \tag{1}$$
$$Ax \geq b, \tag{2}$$
$$x \in \mathbb{Z}_+^n. \tag{3}$$

Předpokládáme, že všechny koeficienty v úloze jsou celočíselné, resp. racionální. Označíme množinu přípustných řešení a její relaxaci

$$S = \{x \in \mathbb{Z}_+^n : \ Ax \geq b\}, \tag{4}$$
$$P = \{x \in \mathbb{R}_+^n : \ Ax \geq b\}. \tag{5}$$

Zřejmě platí $S \subseteq P$. Netriviální je však následující vztah mezi množinami

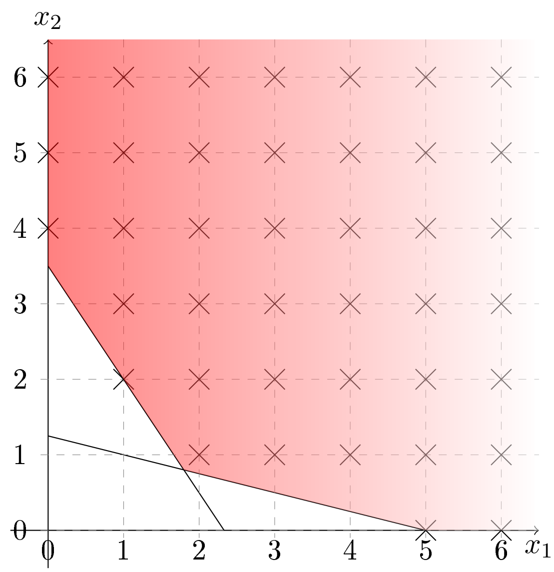
$$S \subseteq \text{conv}(S) \subseteq P.$$

Nebudeme zde uvádět důkaz, ale situaci si ukážeme na obrázku 1 následující množiny

$$x_1 + 4x_2 \geq 5, \tag{6}$$
$$3x_1 + 2x_2 \geq 7, \tag{7}$$
$$x_1, x_2 \in \mathbb{Z}_+^n. \tag{8}$$

Celočíselná množina S je na obrázcích označená křížky. Červeně je pak zobrazena její relaxace P, která je vymezena lineárními omezeními včetně nezápornosti. Všimněte si, že

Obrázek 1: Celočíselná množina S (kříže), relaxace P (červená) a konvexní obal conv(S) (modrá)



některé krajní body relaxované množiny jsou neceločíselné. Zajímavý je pak konvexní obal celočíselných bodů conv(S) zobrazený modře. Pochopitelně všechny krajní body obalu jsou celočíselné a patří do celočíselné množiny S. To nás vede k následující ekvivalenci.

Původní celočíselná úloha

$$\min c^T x : \; x \in S \tag{9}$$

je ekvivalentní následující úloze lineární programování (bez celočíselnosti)

$$\min c^T x : \; x \in \mathrm{conv}(S), \tag{10}$$

viz obrázek 1. Avšak zkonstruovat explicitní vyjádření conv(S) je prakticky velice náročné. Je nutné odvodit velké množství tzv. „silných řezů“, které definují conv(S), což je obvykle téměř nemožné. Pokud $S \neq \emptyset$, je možné ukázat následující vztahy mezi krajními body a směry

$$\mathrm{ext}(\mathrm{conv}(S)) \subseteq P, \quad \mathrm{extd}(\mathrm{conv}(S)) = \mathrm{extd}(P).$$

V praktických úlohách, viz motivace, se obvykle zároveň vyskytují celočíselné a reálné rozhodovací proměnné. Dostáváme pak úlohu smíšeného-celočíselného programování (mixed-integer programming)

$$\begin{aligned} \min\ & c^T x + d^T y \\ \text{s.t. } & Ax + By \geq b, \\ & x \in \mathbb{Z}^n_+, \; y \in \mathbb{R}^{n'}_+. \end{aligned}$$

Více se jí zabývat nebudeme. Jen poznamenejme, že její vlastností jsou kombinací vlastností úloh s reálnými a celočíselnými proměnnými.

1.3 Metoda větvení a mezí (Branch–and–Bound)

Základní metodou prořešení celočíselných úloh (nejen lineárních) je metoda větvení a mezí (Branch–and–Bound). Metoda funguje na principu „rozděl a panuj“. Nechť $M = M_1 \cup M_2 \cup \cdots \cup M_r$ je rozklad množiny přípustných řešení. Spočtěme minima přes jednotlivé podmnožiny

$$f_j = \min_{x \in M_j} f(x).$$

Potom pro celkové minimum platí

$$\min_{x \in M} f(x) = \min_{j=1,\dots,r} f_j.$$

Cílem je tedy rozdělit původní úlohu na sérii menších, které umíme snáze řešit.

Postup metody větvení a mezí reprezentujeme pomocí stromu, viz Obrázek 2, kde v každém kroku (uzlu stromu) řešíme reálnou lineární úlohu (bez celočíselnosti). Nalezení celočíselného optimálního řešení původní úlohy se snažíme zaručit pomocí následujících iterativních kroků:

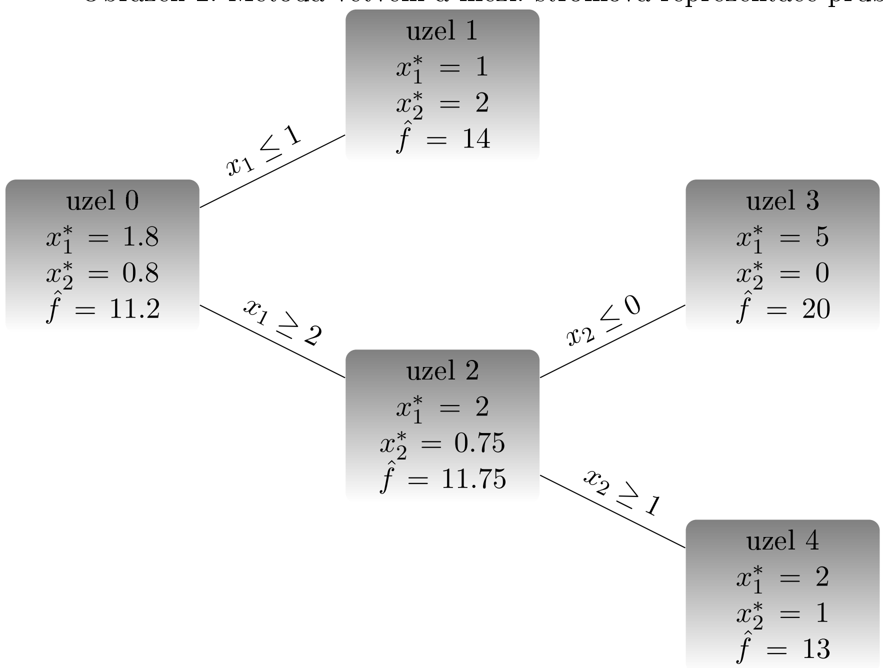
- Vyřešte úlohu LP z fronty (na počátku fronta obsahuje púvodní relaxovanou úlohu).
- **Větvení** (branching): obsahuje-li optimální řešení neceločíselnou složku, např. $\hat{x}_i$, vytvořte dvě nové úlohy tak, že přidáme do formulace vyřešené úlohy právě jedno z následujících omezení $x_i \leq \lfloor \hat{x}_i \rfloor$, $x_i \geq \lceil \hat{x}_i \rceil$. Tím dojde k rozvětvení vyřešené úlohy. Neceločíselnou složku můžeme vybrat libovolně.
- **Omezení** (bounding): pamatuj si dosud nejlepší nalezené celočíselné řešení, tj. to s nejmenší hodnotou účelové funkce. Je-li optimální hodnota LP úlohy horší (vyšší) než nejlepší dosud nalezená pro celočíselné řešení, nové větve nevytvářej (zapomeň), optimální hodnota by se totiž přidáním omezení nezlepšila. Nevětvi ani LP úlohu s celočíselným řešením (ani to z principu není nutné).

Příklad. Využijeme množinu přípustných řešení, se kterou jsme se již seznámili výše. Přidáme lineární účelovou funkci a řešíme metodou vetvění a mezí. Poznamenejme, že jednotlivé úlohy, které v průběhu algoritmu řešíme, jsou úlohy lineárního programování bez celočíselnosti.

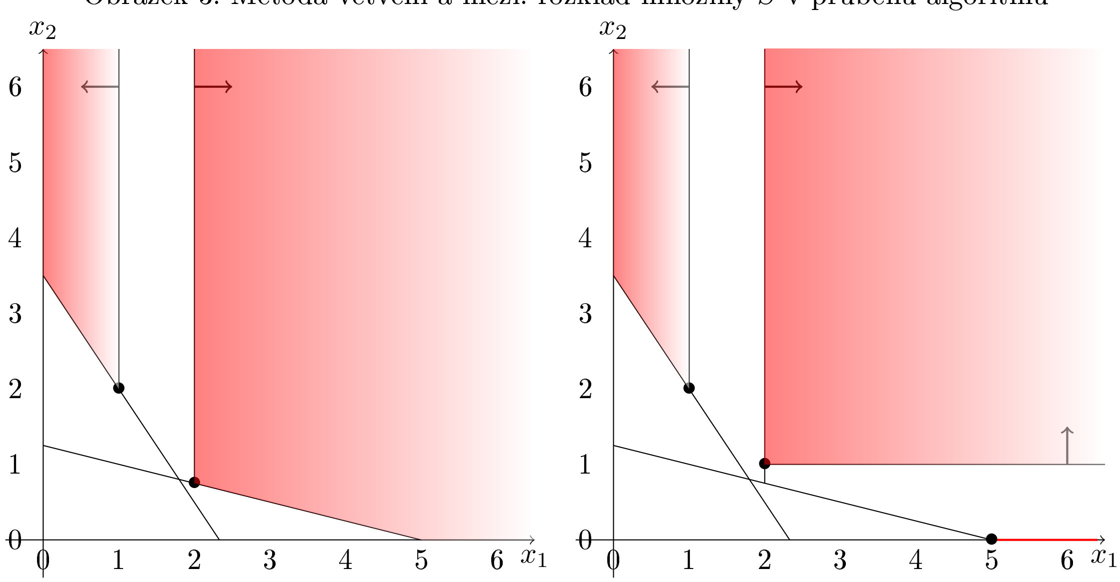
$$\begin{aligned} \min 4x_1 + 5x_2 & & & \quad (11)\\ x_1 + 4x_2 &\geq 5, & & \quad (12)\\ 3x_1 + 2x_2 &\geq 7, & & \quad (13)\\ x_1, x_2 &\in \mathbb{Z}^n_+. & & \quad (14) \end{aligned}$$

Strom reprezentující průběh algoritmu najdeme na obrázku 2. Obrázek 3 poté reprezentuje postupný rozklad relaxované množiny přípustných řešení. Například uzel 0 odpovídá řešení původní relaxované úlohy. Dostali jsem neceločíselné obě složky. Vybrali jsme si x_1 a přidali omezení $x_1 \leq \lfloor 1.8 \rfloor = 1$, resp. $x_1 \geq \lceil 1.8 \rceil = 2$. V uzlu 1 doateneme celočíselné řešení, tedy uplatníme *bounding* a uzel dále nevětvíme. Navíc si zapamatujeme řešení jako dosud nejlepší celočíselné. Uzel 2 má neceločíselné řešení, přidáme tedy dvě větve vedoucí na uzly

Obrázek 2: Metoda větvení a mezí: stromová reprezentace průběhu algoritmu



Obrázek 3: Metoda větvení a mezí: rozklad množiny S v průběhu algoritmu



3 a 4. Oba již mají celočíselná řešení, která porovnáme s dosud nejlepším nalezeným. Vidíme, že v uzlu 4 jsem našli optimální řešení původní celočíselné úlohy. Tím výpočet končí.

Metoda větvení a mezí obvykle na reálných větších úlohách nekončí vyprázdněním fronty, ale buď vyčerpáním časového limitu, například 1 hodina, nebo dosažením požadovaného rozdílu mezi dolní a horní mezí pro optimální hodnotu. Tyto meze postupně aktualizujeme v průběhu výpočtu, kde

- dolní mez získáme pomocí LP relaxace nebo duality,
- horní mez odpovídá přípustnému celočíselné řešení s nejmenší hodnotou účelové funkce.

1.4 Další metody řešení celočíselných úloh

Existuje velké množství dalších metod vhodných pro řešení celočíselných úloh. Uvádíme pouze přehled[1]:

- metoda sečných nadrovin (cutting plane method) – generování řezů vedoucí k aproximaci conv(S), řezy odřezávají pouze prvky z P, které nejsou v S[2],
- dynamické programování,
- metoda generování sloupců (columnn generation),
- Bendersova dekompozice,
- heuristiky (konstrukční heuristiky, tabu search, genetické algoritmy),
- ...

Předešlé algoritmy jsou obvykle kombinovány s branch-and-bound. Například kombinace s metodou sečných nadrovin vede na tzv. branch-and-cut, kdy je metoda větvení a mezí doplněna o přidávání řezů v uzlech stromu.

1.5 Totální unimodularita

Můžeme se setkat s úlohami se speciálním tvarem množiny přípustných řešení, které je možné řešit přímo pomocí algoritmů lineárního programování a dostaneme celočíselné optimání řešení. Matice A definující omezení musí mít speciální strukturu a vektor pravých stran b musí být celočíselný.

Definice. Řekneme, že matice A je totálně unimodulární (TU), jestliže je determinant každé její čtvercové podmatice roven +1, -1, nebo 0.

Je-li vektor pravých stran b celočíselný, potom je každé bazické přípustné řešení celočíselné. Má-li tedy úloha optimální řešení, potom existuje celočíselné optimální bazické

[1] Více se můžete dozvědět na přednášce Výpočetní aspekty optimalizace

[2] Existují silné řezy, které zároveň definují opěrné nadroviny conv(S), a slabé řezy, která mají pouze uvedenou vlastnost. Konstrukce silných řezů je obvykle velice náročná.

řešení. Pro každou regulární čtvercovou podmatici B matice omezení $(A|I)$ můžeme odvodit inverzní matici tak, že pro její prvky platí

$$(B^{-1})_{ij} = (-1)^{i+j} \frac{|B_{ji}|}{|B|},$$

kde matice B_{ji} vznikne z B vynecháním řádku j a sloupce i. Díky předpokladu jsou všechny prvky inverzní matice celočíselné, proto je i každé bazické řešení $B^{-1}b$ celočíselné.

Dá se snadno ukázat, že je-li A totálně unimodulární, potom i A^T, $-A$ a $(A|I)$ jsou totálně unimodulární.

Postačující podmínkou pro totální unimodularitu je například, obsahuje-li každý sloupec matice právě jeden prvek +1, právě jeden -1 a ostaní jsou nulové. Není to však nutná podmínka. Následující matice ji nesplňuje, avšak je totálně unimodulární:

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

Dá se ověřit, že matice omezení dopravního problému, se kterým jsme se seznámili, je totálně unimodulární. Omezení nevyváženého problému jsou

$$-\sum_{j=1}^{n} x_{ij} \geq -a_i, \ i = 1, \dots, m,$$
$$\sum_{i=1}^{m} x_{ij} \geq b_j, \ j = 1, \dots, n,$$

pro něž můžeme matici omezení zapsat jako

$$\begin{array}{ccccccccccc}
x_{11} & x_{12} & \dots & x_{1n} & x_{21} & \dots & x_{2n} & \dots & x_{m1} & \dots & x_{mn} \\
-1 & -1 & \dots & -1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\
0 & 0 & \dots & 0 & -1 & \dots & -1 & \dots & 0 & \dots & 0 \\
 & & \vdots & & & \vdots & & \vdots & & \vdots & \\
0 & 0 & \dots & 0 & 0 & \dots & 0 & \dots & -1 & \dots & -1 \\
1 & 0 & \dots & 0 & 1 & \dots & 0 & \dots & 1 & \dots & 0 \\
 & & \vdots & & & \vdots & & \vdots & & \vdots & \\
0 & 0 & \dots & 1 & 0 & \dots & 1 & \dots & 0 & \dots & 1
\end{array}$$

V každém sloupci najdeme právě jeden prvek rovný -1 a 1. Ostatní jsou rovny nule. Platí tedy, že pro celočíselné kapacity a poptávky existuje optimální rozvoz, který je též celočíselný.

Reference

[1] Wolsey LA (1998). **Integer programming**. Wiley-Interscience series in discrete mathematics and optimization, J. Wiley & sons, New York (N.Y.), Chichester, Weinheim.