# 9 Úloha celočíselné optimalizace

Úlohy lineárního programování již ponecháme za námi a podíváme se na novou zajímavou třídu optimalizačních úloh:
V praxi se totiž často vyskytují okolnosti formulace úloh, ve kterých některé (či všechny) vystupující proměnné mohou nabývat pouze celočíselných hodnot. (Například nemůžeme jednoho pracovníka "přepůlit" na dva úkoly, započatý pracovní úkon třeba nelze přerušit, nemůžeme poslat na trasu linky pouze čtvrt autobusu, a podobně.)

Zmíněné okolnosti pak vedou na úlohy celočíselné lineární optimalizace, neboli *celočíselné programování* IP. V zásadě se dá říci, že úloha IP je úlohou LP s dodatečnými podmínkami celočíselnosti proměnných. Tato analogie je však zavádějící, neboť úlohy IP jsou nesrovnatelně komplikovanější při formulaci i obtížnější při řešení.

## Stručný přehled lekce

- Ukázat matematické formulace úloh celočíselné optimalizace.
- Formální zápis úloh IP a MIP pomocí matic a vektorů.
- Základní popis metody *větvení a mezí* pro řešení úloh IP.

## 9.1 Úvodní příklady IP

Opět pro názornost výkladu začneme hned s jednoduchými příklady úloh celočíselné optimalizace. Poznamenáváme, že pro tuto třídu úloh se používají zkratky IP nebo obecněji MIP. (Z anglického *Integer Programming*.)

**Příklad 9.1.** *Opět se vraťme ke starému známému Příkladu 3.1 o lupíncích a hranolcích s dodatečným požadavkem, že musíme vyrábět (kvůli balení produktů) lupínky i hranolky v celočíselných násobcích množství* $15$*kg.*

**Řešení:** Použijeme původní LP formulaci úlohy

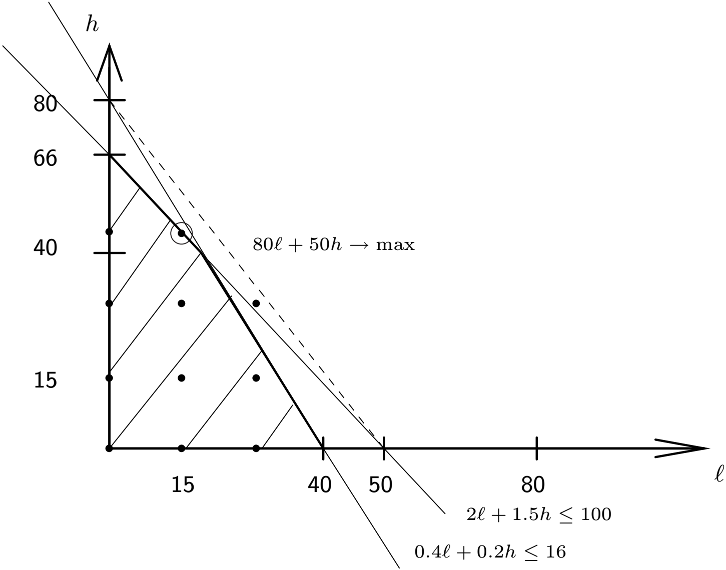
$$\begin{aligned} 2\ell + 1.5h &\leq 100 \\ 0.4\ell + 0.2h &\leq 16 \\ \ell, h &\geq 0\,, \end{aligned}$$

ale přidáme dodatečnou podmínku

$$\ell = 15z_1,\ h = 15z_2,\ \text{ kde } z_1, z_2 \in \boldsymbol{N}\,. \tag{1}$$

Graficky si tuto formulaci vyznačíme na Obrázku 9.1.

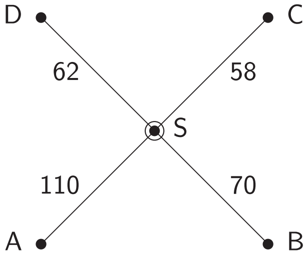
Všimněme si, že je nově nalezené řešení $\ell = 15$kg, $h = 45$kg mírně horší než v Příkladě 3.1 bez podmínky celočíselnosti $\ell = 20$kg, $h = 40$kg. Je přirozené, že přidáním dalších podmínek se kvalita řešení může zhoršit.




Obrázek 9.1: Grafický význam formulace a řešení úlohy IP (Příklad 9.1): Množina řešení původní úlohy LP je šrafovaná, možná řešení v celočíselných násobcích $15$kg jsou značená tečkami a celočíselné optimum je vyznačeno kroužkem.

**Příklad 9.2.** *Letecká společnost přepravuje cestující z města S do čtyř sousedních měst A,B,C,D. Na dnešní den jsou požadavky na přepravu do města A* $110$ *cestujících, do B* $70$*, do C* $58$ *a do D* $62$ *cestujících. Naše společnost přitom má k dispozici dva typy letadel: Prvního typu má 6 letadel s kapacitou po* $33$ *místech a s fixní cenou letu* $120$*. Druhého typu má 4 letadel s kapacitou po* $60$ *místech a s fixní cenou letu* $190$*. Navrhněte, kolik letadel kterého typu má dnes společnost vypravit do kterého města, aby pokryla požadavky přepravy a zároveň minimalizovala cenu letů.*

Schematickým obrázkem si zadání zakreslíme takto:




| typ let. | počet | kapacita | cena |
|---|---|---|---|
| 1 | 6 | 33 | 120 |
| 2 | 4 | 60 | 190 |

**Řešení:** Nechť $n_{1X}$ značí počet letadel prvního typu letících do města $X = \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}$. Podmínky celkového počtu letadel jednotlivých typů zformulujeme:

$$\begin{aligned} n_{1A} + n_{1B} + n_{1C} + n_{1D} &\leq 6 \\ n_{2A} + n_{2B} + n_{2C} + n_{2D} &\leq 4 \end{aligned}$$

Podmínky přepravy cestujících zní:

$$\begin{aligned} 33n_{1A} + 60n_{2A} &\geq 110 \\ 33n_{1B} + 60n_{2B} &\geq 70 \\ 33n_{1C} + 60n_{2C} &\geq 58 \\ 33n_{1D} + 60n_{2D} &\geq 62 \end{aligned}$$

Účelová funkce je taktéž zřejmá

$$\min \ 120(n_{1A} + n_{1B} + n_{1C} + n_{1D}) + 190(n_{2A} + n_{2B} + n_{2C} + n_{2D}).$$

Co nám ještě ve formulaci úlohy chybí? – neuvedli jsme podmínky celočíselnosti

$$n_{1A}, n_{1B}, n_{1C}, n_{1D}, n_{2A}, n_{2B}, n_{2C}, n_{2D} \in \boldsymbol{N}\,.$$

S dodatečnými celočíselnými podmínkami již optimální řešení vyjde

$$n_{1B} = 1,\ n_{1D} = 2,\ n_{2A} = 2,\ n_{2B} = 1,\ n_{2C} = 1,\ n_{2D} = 0\,,$$

neboli do města A poletí dvě velká letadla, do B malé a velké, do C jedno velké a do D dvě malá. O způsobech obecného řešení úloh IP si řekneme za chvíli. □

## 9.2 Formulace úlohy (M)IP

Při matematické formulaci úloh celočíselné optimalizace postupujeme podobně jako při úlohách LP.

**Definice 9.3. Úloha celočíselné (lineární) optimalizace**
je úlohou najít $\max\ \vec{c}\cdot(\vec{x},\vec{z})$ za podmínek

$$\begin{aligned} \boldsymbol{A}\cdot(\vec{x},\vec{z})^T &\leq \vec{b} \\ \vec{x},\vec{z} &\geq 0 \\ \vec{z} &\in \boldsymbol{Z}^* \end{aligned}$$

Složky vektoru $\vec{x}$ jsou *reálné proměnné*, složky $\vec{z}$ jsou *celočíselné proměnné*, oba typy se volně mísí v lineárních nerovnostech vyjařujících podmínky úlohy.
Hodnoty složek $\vec{z}$ obvykle mohou nabývat jen konečně mnoha hodnot.

**Značení**: Takto zadané úloze se obvykle říká *smíšená* se zkratkou *MIP* (mixed IP), vyjadřujíce fakt, že ve formulaci se mísí celočíselné i reálné proměnné. Zkratka *IP* pak označuje úlohu celočíselné optimalizace bez reálných proměnných, což je poměrně častý praktický případ.

**Definice**: Úloha MIP je v *základním tvaru*, pokud je formulována jako

$$\begin{aligned} \max\ \vec{c}\cdot(\vec{x},\vec{z}) \quad &: \\ \boldsymbol{A}\cdot(\vec{x},\vec{z})^T \quad &\leq \quad \vec{b} \\ \vec{x} \quad &\geq \quad 0 \\ \vec{z} \quad &\in \quad \{0,1\}^* \end{aligned}$$

Proměnným $z_i \in \{0,1\}$ se říká *binární proměnné*.

V praxi se ukazuje, že obvykle mohou celočíselné proměnné nabývat jen konečně mnoha hodnot (například $0, 1, \ldots, k$) a v jiných případech omezení hodnot celočíselných proměnných lze snadno do úlohy doplnit. Proto se nadále budeme zabývat jen úlohami MIP s omezenými celočíselnými proměnnými.

**Věta 9.4.** *Každou úlohu MIP s omezenými celočíselnými proměnnými lze převést na základní tvar.*

**Důkaz**: Každou celočíselnou proměnnou $z_i \in \{0, 1, ..., k_i\}$ (dle předpokladů omezená) nahradíme $l = \lceil \log_2(k_i + 1) \rceil$ proměnnými $z_i^0, z_i^1, ..., z_i^{l-1}$ a píšeme $z_i = z_i^0 + 2z_i^1 + ... + 2^{l-1}z_1^{l-1}$ (jako v binárním zápise hodnoty $z_i$). Je zřejmé, že každá přípustná hodnota $z_i$ jednoznačně odpovídá jisté přípustné posloupnosti hodnot $z_i^0, z_i^1, ..., z_i^{l-1} \in \{0,1\}^l$, která vyjadřuje číslice binárního zápisu čísla $z_i$. Nakonec případně rovnosti a nerovnosti podmínek úlohy převedeme do základního LP tvaru podle Věty 3.5. $\square$

## Zobecněné formulace úloh MIP

**Lema 9.5.** *Do základního tvaru úlohy MIP lze převést také úlohy diskrétní optimalizace, ve kterých se vyskytují diskrétní proměnné $\vec{t}$ následujících typů:*

- *Proměnné s více diskrétními reálnými hodnotami, tj. proměnná $t_i \in \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$, kde $\alpha_1, \ldots, \alpha_k$ jsou libovolná reálná čísla.*
- *Semikontinuální proměnné, tj. proměnná $t_i \in \{0\} \cup [\alpha, \beta]$ nabývající nulové hodnoty nebo hodnoty z intervalu $[\alpha, \beta]$ (neobsahujícího nulu).*
- *Proměnné s více intervalovými hodnotami, tj. proměnná $t_i \in [\alpha_1, \beta_1] \cup [\alpha_2, \beta_2] \cup \ldots$ s hodnotami ze sjednocení několika disjunktních intervalů.*

**Důkaz**: Je vidět, že stačí dokázat třetí bod, jelikož první dva jsou jeho speciálními případy. Nechť se množina přípustných hodnot pro $t_i$ skládá z $l$ intervalů $[\alpha_1, \beta_1], [\alpha_2, \beta_2], \ldots, [\alpha_l, \beta_l]$. Použijeme $l-1$ binárních proměnných $z_{i2}, \ldots, z_{il}$ a dodatečné podmínky

$$z_{i,2} + \ldots + z_{i,l} \leq 1$$

$$\alpha_1 + \sum_{j=2}^{l} z_{ij}(\alpha_j - \alpha_1) \ \leq\ t_i \ \leq\ \beta_1 + \sum_{j=2}^{l} z_{ij}(\beta_j - \beta_1)\,.$$

Proměnné $z_{ij}$ nám tak "vyberou", ve kterém z intervalů bude ležet hodnota $t_i$. (Všimněte si, že první nerovnost zaručuje, že bude vybrán jen jeden interval.) □

## 9.3 Řešení MIP relaxací a větvením

Nyní si zjednodušeně uvedeme jednu ze základních metod pro řešení úloh MIP.

**Definice 9.6.** **LP - relaxací** úlohy MIP základního tvaru

$$\begin{aligned} \max\ \vec{c}\cdot(\vec{x},\vec{z}) & \quad : \\ \boldsymbol{A}\cdot(\vec{x},\vec{z})^T & \leq \vec{b} \\ \vec{x} & \geq 0 \\ \vec{z} & \in \{0,1\}^* \end{aligned}$$

je úloha LP ve tvaru:

$$\begin{aligned} \max\ \vec{c}\cdot(\vec{x},\vec{z}) & \quad : \\ \boldsymbol{A}\cdot(\vec{x},\vec{z})^T & \leq \vec{b} \\ \vec{z} & \leq 1 \\ \vec{x},\vec{z} & \geq 0 \end{aligned}$$

LP relaxaci základní úlohy MIP vlastně získáme rozšířením přípustného oboru pro proměnné $\vec{z}$ z hodnot $0,1$ na celý interval $[0,1]$. Geometricky si lze představit polyedr všech přípustných řešení LP-relaxace a v něm obsažené "mřížové" body odpovídající přípustným celým hodnotám $\vec{z}\in\{0,1\}^*$, podobně Obrázku 9.1. Podrobněji viz příští lekce.

**Fakt**: Množina přípustných řešení úlohy MIP je právě průnikem množiny přípustných řešení její LP-relaxace s mřížovými body $\{\vec{z} \in \{0,1\}^*\}$.

**Definice**: Hodnotu optimálního řešení LP-relaxace úlohy MIP nazýváme *(LP-)relaxační mezí* pro původní úlohu MIP.

**Fakt**: Hodnota optimálního řešení úlohy MIP není nikdy lepší než hodnota její LP-relaxační meze. Proto pokud najdeme přípustné řešení úlohy MIP dosahující hodnoty její relaxační meze, máme optimální řešení.

## Algoritmus 9.7. Jednoduchá metoda větvení a mezí s lineární relaxací

*Mějme danou úlohu MIP v základním tvaru, tj. s binárními proměnnými $\vec{z}$.*
Nechť $f$ je globální proměnná inicializovaná $f = -\infty$.

Procedura `branch&bound`($\mathcal{U}$: úloha MIP)

1. $\mathcal{L}$ = LP-relaxace $\mathcal{U}$, $\vec{s}^o$ = optimální řešení $\mathcal{L}$, $r^o$ = relaxační mez.
2. Pokud $\mathcal{L}$ nemá přípustné řešení nebo $r^o \leq f$, vrátíme se bez řešení `return` $\emptyset$.
3. Pokud $\mathcal{L}$ nemá optimální řešení z důvodu neomezenosti a zároveň v některém přípustném řešení $\mathcal{L}$ jsou složky $\vec{z}$ celé, položíme $f = \infty$ a vrátíme `return` $\emptyset$.
4. Pokud $\vec{s}^o$ je přípustné (celoč.) řešení $\mathcal{U}$, položíme $f = r^o$ a vrátíme `return` $\vec{s}^o$.
5. Zvolíme binární proměnnou $z_i$ (*nedeterministicky*) v $\mathcal{U}$ a vytvoříme jejím dosazením podúlohy

$$\mathcal{U}_0 := (\mathcal{U} \restriction z_i = 0) , \qquad \mathcal{U}_1 := (\mathcal{U} \restriction z_i = 1) .$$

Zavoláme rekurzivně

$$\vec{s}^1 = \texttt{branch\&bound}(\mathcal{U}_1) \text{ a } \qquad \vec{s}^2 = \texttt{branch\&bound}(\mathcal{U}_2)$$

zároveň *nedeterministicky*. Vracíme lepší z obou řešení `return` $\max\ (\vec{s}^1, \ \vec{s}^2)$.

Návratová hodnota procedury je optimálním řešením úlohy $\mathcal{U}$ s účelovou funkcí $f$. Pokud $f = -\infty$, úloha nemá přípustné řešení, pokud $f = \infty$, úloha nemá optimální řešení z důvodu neomezenosti.

**Tvrzení 9.8.** *Algoritmus 9.7 vždy (pro jakoukoliv volbu vykonání a pořadí nedeterministických kroků) skončí a správně nalezne optimální řešení.*

**Důkaz**: Nechť $d$ je počet binárních proměnných – složek $\vec{z}$. Pak zřejmě žádná větev rekurze není hlubší než $d$, neboť každým zanořením se sníží počet binárních proměnných v úloze. Nakonec pokud úloha $\mathcal{U}$ neobsahuje binární proměnné ($\vec{z}$ nemá složky), tak se vždy aplikuje jeden z kroků 2,3,4 před 5 a rekurze se ukončí. Takže algoritmus skončí po $O(2^d)$ iteracích procedury `branch&bound`.

Předpokládejme, že optimální řešení úlohy $\mathcal{U}$ má binární složky rovny $\vec{z}^o$. Pak ve větvi rekurze, která odpovídá volbám hodnot $\vec{z}$ jako v $\vec{z}^o$ bude toto řešení nalezeno jako přípustná relaxační mez úlohy $\mathcal{L}'$. (Úloha $\mathcal{U} \restriction \vec{z} = \vec{z}^o$ je již úlohou LP, kterou umíme přesně vyřešit.) Toto přípustné řešení pak jako nejlepší možné (případně jedno z několika stejné hodnoty) bude vráceno procedurou `branch&bound`. □

Různými implementačními způsoby volby proměnné $z_i$ pro "větvení" se budeme zabývat v příští kapitole. (Algoritmus korektně funguje pro jakoukoliv volbu, ale vhodnou volbou jej lze výrazně urychlit.)

Podívejme se blíže na analýzu složitosti v důkaze 9.8 – časový odhad $O(2^d)$ je velmi "špatný" již pro poměrně malé hodnoty $d$. Co však způsobuje tento exponenciální nárust složitosti algoritmu? Je to prohledávání mnoha větví až do úplného konce, do hloubky $d$. Naši snahou při implementaci algoritmu tedy musí být co nejrychleji "zabít" všechny neperspektivní větve rekurze. V praxi se s největším potenciálem k vyloučení větví rekurze ukazuje krok 2, když relaxační mez vyjde horší než nejlepší dosud nalezené řešení. K aplikaci tohoto kroku však již nějaké přípustné řešení potřebujeme mít nalezené. . .

## 9.4 Jednoduché ukázky řešení IP

**Příklad 9.9.** *Letecká společnost má přepravit* $141$ *lidí z města A do B. K dispozici má čtyři letadla:*

| | *kapacita* | *cena letu* | *posádka* |
|---|---|---|---|
| *L1* | *90* | *300* | *7* |
| *L2* | *60* | *210* | *4* |
| *L3* | *50* | *200* | *3* |
| *L4* | *33* | *130* | *2* |

*Která letadla má zvolit, aby dosáhla nejlevnějšího řešení?*

**Řešení:**

$$\max \; -300z_1 - 210z_2 - 200z_3 - 130z_4$$

$U$: pro $90z_1 + 60z_2 + 50z_3 + 33z_4$

$$z_i \in \{0,1\}$$

$f = -\infty$ $U$ LP relaxace

$$\max \; -300z_1 - 210z_2 - 200z_3 - 130z_4$$

$$90z_1 + 60z_2 + 50z_3 + 33z_4 \geq 141$$

$$0 \leq z_i \leq 1$$

Řešení...

Výsledek: poletí L1 a L2. □