

Celočíselné lineární programování (ILP)

Zdeněk Hanzálek, Přemysl Šůcha
hanzalek@fel.cvut.cz

ČVUT FEL Katedra řídicí techniky

23. června 2010

1 Úvod

- Formulace problému

2 Vlastnosti a řešení

- Srovnání ILP s LP
- Příklady
- Metody řešení
- Speciální případy ILP
- Vyjadřovací schopnosti ILP

3 Závěr

Celočíselné lineární programování (ILP)

Úloha celočíselného lineárního programování je zadána maticí $\mathbf{A} \in \mathbb{R}^{m \times n}$ a vektory $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c} \in \mathbb{R}^n$. Cílem je najít takový vektor $\mathbf{x} \in \mathbb{Z}^n$, že platí $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$ a $\mathbf{c}^T \cdot \mathbf{x}$ je maximální.

Obvykle se celočíselné lineární programování zapisuje ve tvaru:

$$\max \{ \mathbf{c}^T \cdot \mathbf{x} : \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}, \mathbf{x} \in \mathbb{Z}^n \}.$$

- Celá řada praktických problémů týkajících se optimalizace může být modelována a řešena pomocí celočíselného lineárního programování (Integer Linear Programming - ILP).

- Tato úloha se od úlohy běžného lineárního programování LP liší v tom, že proměnné jsou omezeny na celá čísla. Pokud některé proměnné mohou být i reálná čísla, potom se úloha nazývá smíšené celočíselné programování MIP (Mixed Integer Programming), ale častěji se i tomuto případu říká ILP.
- Pokud bychom takovou úlohu řešili pomocí lineárního programování s tím, že bychom výsledek zaokrouhlili, nejenom že bychom neměli zaručeno že výsledné řešení bude optimální ale ani to, zda bude přípustné.
- Zatímco úloha LP je řešitelná v polynomiálním čase, úloha ILP je tzv. NP-těžká (NP-hard), neboli není znám algoritmus, který by vyřešil libovolnou instanci této úlohy v polynomiálním čase.
- Protože prostor řešení ILP není konvexní množina, nelze přímo aplikovat metody konvexní optimalizace.

Dělení kořisti (2 partition problem)

- **Instance:** Počet bankovek $n \in \mathbb{Z}^+$ a hodnoty bankovek p_1, \dots, p_n , kde $p_{i \in 1..n} \in \mathbb{Z}^+$.
- **Rozhodnutí:** Existuje podmnožina $S \subseteq \{1, \dots, n\}$ taková, že
$$\sum_{i \in S} p_i = \sum_{i \notin S} p_i?$$

Rozhodovací problém, jež lze pomocí ILP zapsat jako omezující podmínku danou výše uvedenou rovnicí (my ji zapíšeme trochu jinak, abychom později snáze formulovali optimalizační problém).

- $x_i = 1$ iff $i \in S$

Toto je jeden z “nejsnazších” NP-úplných problémů.

min 0

subject to:

$$\sum_{i \in 1..n} x_i * p_i = 0.5 * \sum_{i \in 1..n} p_i$$

parameters: $n \in \mathbb{Z}_0^+$, $p_{i \in 1..n} \in \mathbb{Z}_0^+$

variables: $\mathbf{x}_{i \in 1..n} \in \{0, 1\}$

Dělení kořisti s relaxací na nedělitelnost bankovek

Pokud povolíme dělení bankovek, pak $x_{i \in 1..n} \in \langle 0, 1 \rangle$. Prostor řešení je konvexní množina - problém lze formulovat pomocí LP:

$$\begin{array}{ll}\min & 0 \\ \text{subject to:} & \\ & \sum_{i \in 1..n} x_i * p_i = 0.5 * \sum_{i \in 1..n} p_i \\ & x_i \leq 1 \quad i \in 1..n \\ \text{parameters:} & n \in \mathbb{Z}_0^+, p_{i \in 1..n} \in \mathbb{Z}_0^+ \\ \text{variables:} & x_{i \in 1..n} \in \mathbb{R}_0^+\end{array}$$

- Příklad: $p = [100, 50, 50, 50, 20, 20, 10, 10]$ s relaxací umožňuje dělit kořist ($x = [0, 0, 0.9, 1, 1, 1, 1, 1]$) na stejně velké poloviny $100 + 50 + 5 = 45 + 50 + 20 + 20 + 10 + 10$, ale v případě nedělitelných bankovek tato instance nemá řešení.
- U některých instancí snadno určíme, že je nelze rozdělit (např. součet všech cen podělený největším společným dělitelem není sudé číslo), ale není znám algoritmus, který by to uměl udělat v polynomiálním čase pro všechny možné instance problému s nedělitelnými bankovkami.

Dělení kořisti - optimalizační verze

- Rozhodovací problém lze řešit optimalizačním algoritmem tak, že zavedeme práh (zde $0.5 * \sum_{i \in 1..n} p_i$) a ptáme se, zda je nalezená optimální hodnota nad prahem nebo rovna prahu.
- Navíc získáme hodnotu, která se prahu nejvíce blíží pokud rozhodovací problém nemá řešení.

min C_{max}

subject to:

$$\sum_{i \in 1..n} x_i * p_i \leq C_{max}$$

$$\sum_{i \in 1..n} (1 - x_i) * p_i \leq C_{max}$$

parameters: $n \in \mathbb{Z}_0^+$, $p_{i \in 1..n} \in \mathbb{Z}_0^+$

variables: $x_{i \in 1..n} \in \{0, 1\}$, $C_{max} \in \mathbb{R}_0^+$

Aplikace: rozvrhování množiny n nepreemptivních úloh $\{T_1, T_2, \dots, T_n\}$ s výpočetním časem $[p_1, p_2, \dots, p_n]$ na dvou paralelních identických procesorech a minimalizací dokončení poslední z nich (maximum Completion time) neboli $P2 \parallel C_{max}$ (při preempci $P2 |pmtn| C_{max}$)

Nejkratší cesta v grafu (Shortest Path)

- **Instance:** Orientovaný graf G s n vrcholy, matice vzdáleností $c : V \times V \rightarrow \mathbb{R}_0^+$ a dva vrcholy $s, t \in V$.
- **Cíl:** Nalézt nejkratší cestu z vrcholu s do vrcholu t nebo rozhodnout, že t není dosažitelný z s .

LP formulace podle fyz. analogie

- vrchol = kulička
- hrana (pro symetrickou matici c) = provázek
- vrchol 1 je uchycen, ostatní vrcholy taženy gravitací
- napnuté provázky = nejkratší cesta

$$\begin{array}{ll}\max & l_t \\ \text{subject to:} & \\ & l_s = 0 \\ & l_j \leq l_i + c_{i,j} \quad i \in 1..n, j \in 1..n \\ \text{parameters:} & n \in \mathbb{Z}_0^+, c_{i \in 1..n, j \in 1..n} \in \mathbb{R}_0^+ \\ \text{variables:} & l_{i \in 1..n} \in \mathbb{R}_0^+\end{array}$$

Úloha obchodního cestujícího

Problém asymetrického obchodního cestujícího (Asymmetric Traveling Salesman Problem)

- **Instance:** Úplný orientovaný graf K_n ($n \geq 3$), matice vzdáleností $c : V \times V \rightarrow \mathbb{Q}^+$.
- **Cíl:** Nalézt nejkratší cyklus (t.j. uzavřenou orientovanou cestu) procházející všemi vrcholy.

$x_{i,j} = 1$ iff vrchol i se nachází v cyklu bezprostředně po vrcholu j
 s_i “možný čas” návštěvy ve vrcholu i , zajišťuje nedělitelnost cyklu

$$\min \quad \sum_{i \in 1..n} \sum_{j \in 1..n} c_{i,j} * x_{i,j}$$

subject to:

$$\sum_{i \in 1..n} x_{i,j} = 1 \quad j \in 1..n$$

vstup jednou

$$\sum_{j \in 1..n} x_{i,j} = 1 \quad i \in 1..n$$

vystup jednou

$$s_i + c_{i,j} - (1 - x_{i,j}) * M \leq s_j \quad i \in 1..n, j \in 2..n$$

nedělitelnost cyklu

parameters: $M \in \mathbb{Z}_0^+$, $n \in \mathbb{Z}_0^+$, $c_{i \in 1..n, j \in 1..n} \in \mathbb{Q}^+$

variables: $x_{i \in 1..n, j \in 1..n} \in \{0, 1\}$, $s_{i \in 1..n} \in \mathbb{R}_0^+$

Mezi nejznámější metody řešení obecné úlohy ILP patří:

- Výčtové metody (Enumerative Methods)
- Metoda větví a mezí (Branch and Bound)
- Metody sečných nadrovin (Cutting Planes Methods)

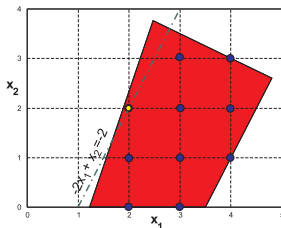


Za významné osobnosti v oblasti ILP jsou považováni Ralph Gomory a Vašek Chvátal. Dnes se po nich jmenují některé metody řešení této úlohy: Gomoryho řezy resp. Chvátal-Gomoryho řezy.

- Výpočet je založen na prohledávání oblasti zahrnující všechna přípustná řešení.
- Vzhledem k celočíselnému omezení proměnných je počet těchto řešení konečný, ale jejich počet je extrémně vysoký. Proto je tato metoda vhodná pouze pro malé problémy s omezeným počtem diskrétních proměnných.
- Tuto myšlenku je možno použít při řešení ILP tak, že ke každé kombinaci diskrétních proměnných je vyřešena úloha LP, kde jsou diskrétní proměnné považovány za konstanty.

$$\begin{array}{lll} \max & -2x_1 & + x_2 \\ \text{s.t.} & 9x_1 & - 3x_2 \geq 11 \\ & x_1 & + 2x_2 \leq 10 \\ & 2x_1 & - x_2 \leq 7 \\ & x_1, x_2 \geq 0, & x_1, x_2 \in \mathbb{Z}_0^+ \end{array}$$

Z obrázku dole je patrné, že v úvahu připadá 10 přípustných řešení s tím, že optimální řešení je $x_1 = 2, x_2 = 2$ s hodnotou kritéria -2 .

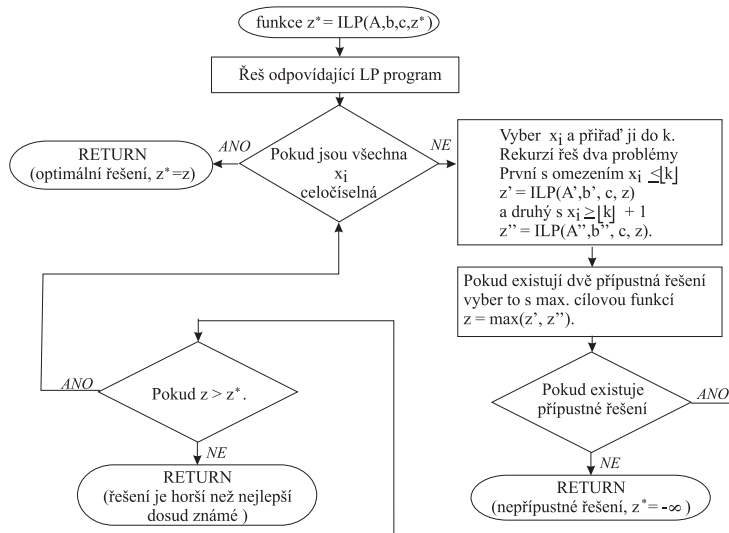


- Principem této metody je rozklad množiny přípustných řešení na disjunktí podmnožiny.
- Algoritmus začíná výpočtem úlohy tak, že zanedbává požadavek na celočíselnot a úloha je vyřešena klasickými metodami LP.
- Pokud jsou všechny proměnné x_i celočíselné, potom výpočet končí. Pokud nejsou, vybere se jedna proměnná $x_i \notin \mathbb{Z}$ a její hodnota je přiřazena do k .
- Následně se vyšetřovaná oblast rozdělí na dvě podmnožiny tak, že v první uvažujeme $x_i \leq \lfloor k \rfloor$ a v druhé $x_i \geq \lfloor k \rfloor + 1$.
- Výpočet je rekurzivně opakován pro obě nově vzniklé oblasti, dokud není nalezeno přípustné řešení, kde všechna x_i jsou celočíselná.

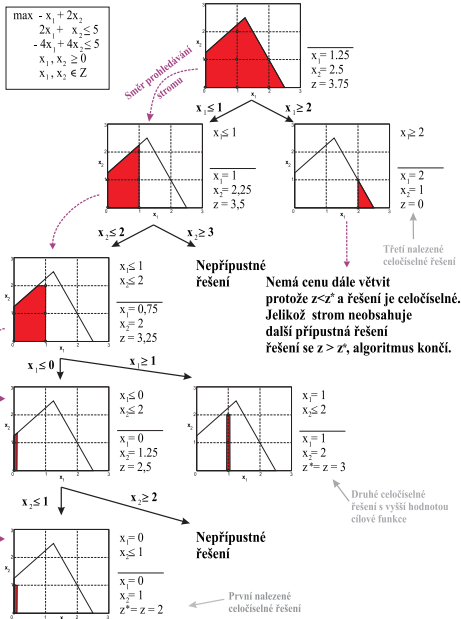
- pomocí větvení (*branching*) vytváří algoritmus stavový prostor řešení, který lze grafově znázornit stromem
- uzly představují nějaká částečná řešení problému
- listy odpovídají buď nalezeným celočíselným řešením nebo řešením, která jsou odříznuta (nepřípustná řešení a řešení která nevedou k lepšímu výsledku)
- jakmile algoritmus nalezne nějaké celočíselné řešení, může být hodnota odpovídající cílové funkce použita k prořezávání stromu (*bounding*)
- uzel je odříznut, pokud cílová funkce tohoto částečného (i neceločíselného) řešení z není lepší než z^* , hodnota cílové funkce nejlepšího doposud známého celočíselného řešení

Algoritmus ILP nejčastěji využívá k řešení LP simplexovou metodu protože po přidání nové omezující podmínky není nutné spouštět simplexový algoritmus znova od začátku ale umožňuje navázat na předchozí výpočet LP řešením duální simplexové metody.

Metoda větví a mezí



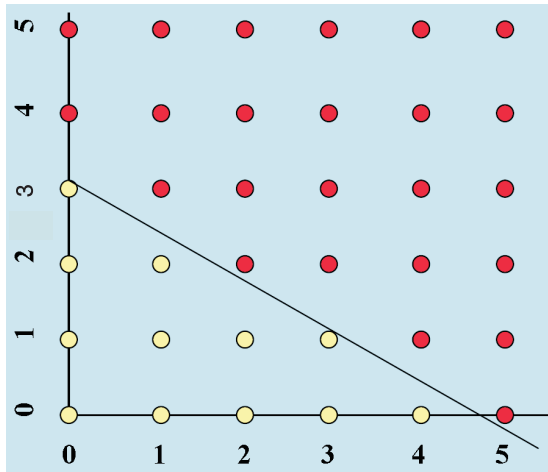
Metoda větví a mezí - příklad



Množina řešení ILP

$$\begin{aligned}\max z = & 3x_1 + 4x_2 \\ \text{s.t. } & 5x_1 + 8x_2 \leq 24 \\ & x_1, x_2 \in \mathbb{Z}_0^+\end{aligned}$$

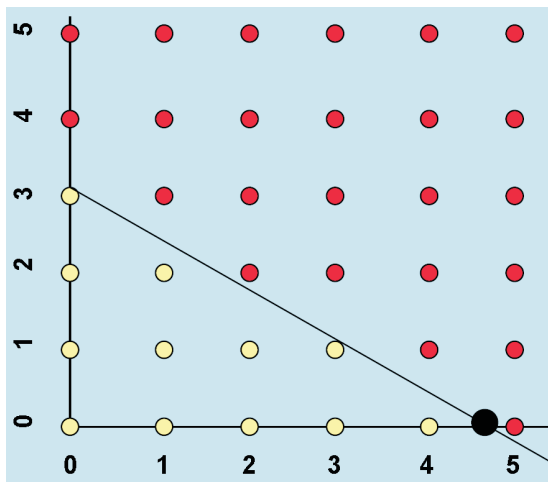
- Co je optimální řešení?
- Můžeme použít LP k řešení ILP problému?



Zaokrouhlení není vždy úspěšné

$$\max z = 3x_1 + 4x_2$$

- LP řešení $z = 14.4$ pro $x_1 = 4.8, x_2 = 0$
- zaokrouhlením získáme neproveditelné řešení $x_1 = 5, x_2 = 0$
- odtržením neceločíselné části získáme řešení $z = 12$ pro $x_1 = 4, x_2 = 0$
- optimální řešení $z = 13$ pro $x_1 = 3, x_2 = 1$



Výhody plynoucí z omezení na celočíselné proměnné

- realističtější (nelze vyrábět 4.3 auta)
- pružnější - do binárních proměnných často zakódujeme nějaká rozhodnutí (logické výrazy ...)
- schopnost formulovat NP-obtížné problémy

Nevýhody

- obtížnější tvorba modelu
- zpravidla lze řešit pouze problémy o přibližně 1000 celočíselných proměnných

Nejkratší cesta v grafu (Shortest Path)

- **Instance:** Orient. graf G , incidenční matice $w : V \times E \rightarrow \{-1, 0, 1\}$, vektor délek hran $c \in \mathbb{R}_0^+$ a dva vrcholy $s, t \in V$.
- **Cíl:** Nalézt nejkratší z vrcholu s do vrcholu t nebo rozhodnout, že t není dosažitelný z s .

LP formulace:

- $x_j = 1$ iff hrana j je vybrána
- vyjma zdroje a cíle platí pro každý vrchol: do vrcholu vstoupíme tolikrát, kolikrát z něho vystoupíme

$$\begin{array}{ll}\min & \sum_{j \in 1..m} c_j * x_j \\ \text{subject to:} & \\ & \sum_{j \in 1..m} w_{s,j} * x_j = 1 \quad \text{zdroj toku} \\ & \sum_{j \in 1..m} w_{t,j} * x_j = -1 \quad \text{cíl toku} \\ & \sum_{j \in 1..m} w_{i,j} * x_j = 0 \quad i \in V \setminus \{s, t\} \\ \text{pars:} & w_{i \in 1..n, j \in 1..m} \in \{-1, 0, 1\}, c_{j \in 1..m} \in \mathbb{R}_0^+ \\ \text{vars:} & x_{j \in 1..m} \in \mathbb{R}_0^+\end{array}$$

Výsledné hodnoty x_i jsou celočíselné(binární) přestože jde o LP. Proč ?

Obyčejná úloha ILP není řešitelná v polynomiálním čase. Existují však speciální případy, které lze řešit v polynomiálním čase.

Definice - Totálně unimodulární matice

Matice $\mathbf{A} = [a_{ij}]$ typu m/n je totálně unimodulární, jestliže

- 1. determinant každé čtvercové podmatice matice \mathbf{A} je roven 0, +1 nebo -1.

Pozn: $a_{ij} \in \{0, 1, -1\}$ je nezbytnou podmínkou pro to, aby \mathbf{A} byla totálně unimodulární.

Totálně unimodulární matice

Věta

Úlohu ILP s totálně unimodulární maticí \mathbf{A} a celočíselným vektorem b lze řešit simplexovým algoritmem a výsledné řešení je celočíselné.

Věta

Úloha ILP s totálně unimodulární maticí \mathbf{A} a celočíselným vektorem b je řešitelná v polynomiálním čase.

Věta

Nechť \mathbf{A} je matice typu m/n taková, že

- 1 $a_{ij} \in \{0, 1, -1\}$, $i = 1, \dots, m$, $j = 1, \dots, n$
- 2 každý sloupec matice \mathbf{A} obsahuje buďto nejvýše jeden nenulový prvek nebo právě dva nenulové prvky, a to $+1$ a -1 ,

pak je matice \mathbf{A} totálně unimodulární.

Vyjadřovací schopnosti ILP - př. investice do nemovitostí

Zvažujeme investice do 6 nemovitostí.

Cena a příjem z nájmu u každé z nich jsou uvedeny v tabulce.

nemovitost	1	2	3	4	5	6
cena[mil Kč]	5	7	4	3	4	6
nájem[tis Kč]	16	22	12	8	11	19

Cíl:

- maximalizovat příjem z nájmu

Omezení:

- investiční rozpočet je 14 mil Kč
- každou z budov lze koupit maximálně jednou

Vyjadřovací schopnosti ILP - př. investice do nemovitostí

Zvažujeme investice do 6 nemovitostí.

Cena a příjem z nájmu u každé z nich jsou uvedeny v tabulce.

nemovitost	1	2	3	4	5	6
cena[mil Kč]	5	7	4	3	4	6
nájem[tis Kč]	16	22	12	8	11	19

Cíl:

- maximalizovat příjem z nájmu

Omezení:

- investiční rozpočet je 14 mil Kč
- každou z budov lze koupit maximálně jednou

Formulace

- $x_i = 1$ právě když koupíme nemovitost i

$$\begin{aligned} \max \quad & z = 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6 \\ \text{s.t.} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14 \\ & x_{i \in 1 \dots 6} \in \{0, 1\} \end{aligned}$$

Přidání logických formulí $x_1 \Rightarrow \overline{x_3}$

Další omezení:

- jestliže je dům 1 vybrán, potom není vybrán dům 3

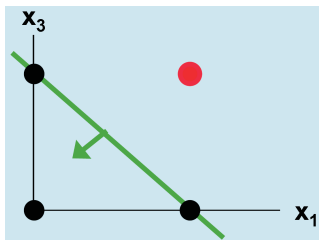
x_1	x_3	$x_1 \Rightarrow \overline{x_3}$
0	0	1
0	1	1
1	0	1
1	1	0

Přidání logických formulí $x_1 \Rightarrow \overline{x_3}$

Další omezení:

- jestliže je dům 1 vybrán, potom není vybrán dům 3

x_1	x_3	$x_1 \Rightarrow \overline{x_3}$
0	0	1
0	1	1
1	0	1
1	1	0

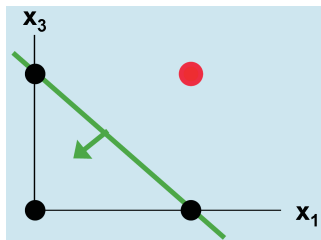


Přidání logických formulí $x_1 \Rightarrow \overline{x_3}$

Další omezení:

- jestliže je dům 1 vybrán, potom není vybrán dům 3

x_1	x_3	$x_1 \Rightarrow \overline{x_3}$
0	0	1
0	1	1
1	0	1
1	1	0



$$\begin{aligned} \max \quad & z = 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6 \\ \text{s.t.} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14 \\ & x_1 + x_3 \leq 1 \\ & x_{i \in 1 \dots 6} \in \{0, 1\} \end{aligned}$$

Přidání logických formulí $x_2 \Rightarrow x_1$

Další omezení:

- jestliže je dům 2 vybrán, potom musí být vybrán i dům 1

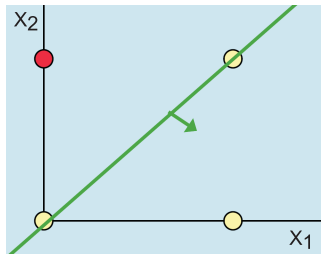
x_1	x_2	$x_2 \Rightarrow x_1$
0	0	1
0	1	0
1	0	1
1	1	1

Přidání logických formulí $x_2 \Rightarrow x_1$

Další omezení:

- jestliže je dům 2 vybrán, potom musí být vybrán i dům 1

x_1	x_2	$x_2 \Rightarrow x_1$
0	0	1
0	1	0
1	0	1
1	1	1

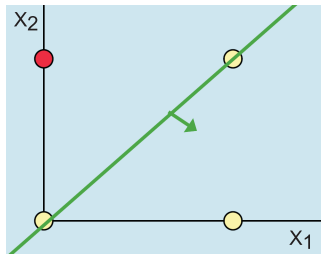


Přidání logických formulí $x_2 \Rightarrow x_1$

Další omezení:

- jestliže je dům 2 vybrán, potom musí být vybrán i dům 1

x_1	x_2	$x_2 \Rightarrow x_1$
0	0	1
0	1	0
1	0	1
1	1	1



$$\begin{array}{ll} \max & z = 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6 \\ \text{s.t.} & 5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14 \\ & x_2 \leq x_1 \\ & x_{i \in 1 \dots 6} \in \{0, 1\} \end{array}$$

Přidání logických formulí x_2 XOR x_1

Další omezení:

- buď je vybrán dům 4 nebo dům 5, ale ne oba

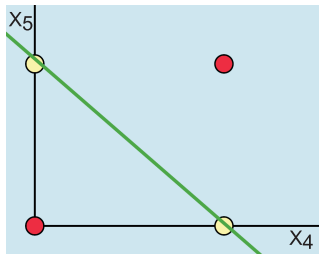
x_4	x_5	x_4 XOR x_5
0	0	0
0	1	1
1	0	1
1	1	0

Přidání logických formulí x_2 XOR x_1

Další omezení:

- buď je vybrán dům 4 nebo dům 5, ale ne oba

x_4	x_5	$x_4 \text{ XOR } x_5$
0	0	0
0	1	1
1	0	1
1	1	0

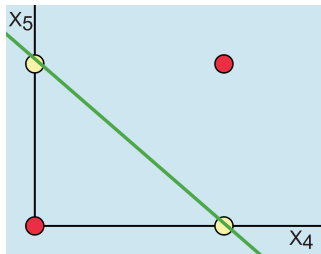


Přidání logických formulí x_2 XOR x_1

Další omezení:

- buď je vybrán dům 4 nebo dům 5, ale ne oba

x_4	x_5	$x_4 \text{ XOR } x_5$
0	0	0
0	1	1
1	0	1
1	1	0



$$\begin{aligned} \max \quad & z = 16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6 \\ \text{s.t.} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 + 4x_5 + 6x_6 \leq 14 \\ & x_4 + x_5 = 1 \\ & x_{i \in 1 \dots 6} \in \{0, 1\} \end{aligned}$$

Promyslete jak formulovat:

- musí být vybrán dům 1 a nesmí být vybrán dům 2
- musí být vybrány alespoň 3 domy
- musí být vybrány právě 3 domy
- pokud jsou vybrány domy 1 a 2 zároveň, pak musí být vybrán i dům 3
- neboli $(x_1 \text{ AND } x_2) \Rightarrow x_3$
- nesmí být vybrány právě dva domy

Vyjadřovací schopnosti ILP - př. plánování výroby oděvů

Objem práce, množství materiálu a zisk jsou uvedeny v tabulce.

produkt	tričko	košile	kalhoty	kapacita
objem práce	3	2	6	150
materiál	4	3	4	160
zisk	6	4	7	

Cíl:

- maximalizovat zisk

Omezení:

- celková pracovní kapacita je maximálně 150 hodin
- k dispozici je maximálně 160m látky

Vyjadřovací schopnosti ILP - př. plánování výroby oděvů

Objem práce, množství materiálu a zisk jsou uvedeny v tabulce.

produkt	tričko	košile	kalhoty	kapacita
objem práce	3	2	6	150
materiál	4	3	4	160
zisk	6	4	7	

Cíl:

- maximalizovat zisk

Omezení:

- celková pracovní kapacita je maximálně 150 hodin
- k dispozici je maximálně 160m látky

Formulace

- x_i počet výrobků produktu i

$$\begin{array}{ll}\max & z = 6x_1 + 4x_2 + 7x_3 \\ \text{s.t.} & 3x_1 + 2x_2 + 6x_3 \leq 150 \\ & 4x_1 + 3x_2 + 4x_3 \leq 160 \\ & x_{i \in 1 \dots 3} \in \mathbb{Z}_0^+\end{array}$$

Další omezení:

- zaplatit fixní cenu za pronájem stroje podle typu produktu

produkt	tričko	košile	kalhoty
cena stroje	200	150	100

Další omezení:

- zaplatit fixní cenu za pronájem stroje podle typu produktu

produkt	tričko	košile	kalhoty
cena stroje	200	150	100

Formulace

- zavedeme binární proměnnou y_i , tak aby $y_i = 1$ právě když je vypůjčen stroj na výrobu produktu i
- nové kritérium $\max z = 6x_1 + 4x_2 + 7x_3 - 200y_1 - 150y_2 - 100y_3$
- provážeme binární y_i s celočíselnou x_i

Stroj i je vypůjčen právě když je vyráběn produkt i , neboli provádíme binární y_i s celočíselnou x_i tak, aby platilo

- $y_i = 0$ právě když $x_i = 0$
- $y_i = 1$ právě když $x_i \geq 1$

Stroj i je vypůjčen právě když je vyráběn produkt i , neboli provádíme binární y_i s celočíselnou x_i tak, aby platilo

- $y_i = 0$ právě když $x_i = 0$
- $y_i = 1$ právě když $x_i \geq 1$

Pro obor hodnot $x_i \in \langle 0, 100 \rangle$ lze tento vztah zajistit pomocí nerovnic

- $x_i \leq 100y_i$ a
- $x_i \geq y_i$... zde lze vypustit díky kritériu (bod $x_i = 0, y_i = 1$ nebude vybrán jelikož chceme x_i co největší a y_i co nejmenší)

Další omezení:

- za účelem využití pracovní doby je celkový objem práce 40, 80 nebo 120 hodin

Požadujeme, aby nějaká funkce proměnných x nabývala pouze omezeného počtu hodnot

$$3x_1 + 2x_2 + 6x_3 = 40 \text{ nebo } 80 \text{ nebo } 120$$

Lze vyjádřit pomocí množiny pomocných proměnných $v_{i \in 1 \dots 3} \in \{0, 1\}$ následovně:

$$\begin{aligned} 3x_1 + 2x_2 + 6x_3 &= 40v_1 + 80v_2 + 120v_3 \\ \sum_{i=1}^3 v_i &= 1 \end{aligned}$$

Musí platit alespoň jedna ze dvou podmínek

- př. soustava nerovnic

Při modelování problémů pomocí ILP se často dostáváme do situace, kdy je potřeba vyjádřit skutečnost, že platí jedno omezení nebo druhé omezení nebo obě zároveň. Např. pro $x_{i \in 1 \dots 4} \in \langle 0, 5 \rangle$, $x_{i \in 1 \dots 4} \in \mathcal{R}$

$$\begin{array}{ll} \text{platí} & 2x_1 + x_2 \leq 5 \\ \text{nebo} & 2x_3 - x_4 \leq 2 \\ & \text{nebo obě} \end{array}$$

Tento případ lze modelovat pomocí konstanty M (velkého kladného čísla - big M , zde například 15) a pomocné proměnné $y \in \{0, 1\}$, tak aby byla “vypnuta” účinnost jedné z nerovnic

$$\begin{array}{l} 2x_1 + x_2 \leq 5 + M \cdot y \\ 2x_3 - x_4 \leq 2 + M \cdot (1 - y) \end{array}$$

Musí platit alespoň jedna ze dvou podmínek

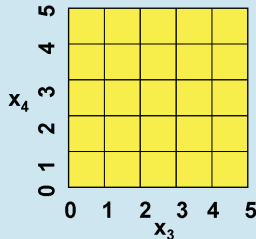
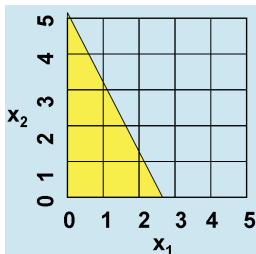
pro $y = 0$ se soustava

$$2x_1 + x_2 \leq 5 + M \cdot y$$

$$2x_3 - x_4 \leq 2 + M \cdot (1 - y)$$

redukuje na

$$2x_1 + x_2 \leq 5$$



Musí platit alespoň jedna ze dvou podmínek

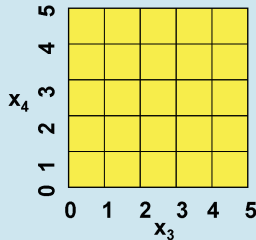
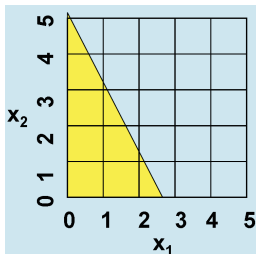
pro $y = 0$ se soustava

$$2x_1 + x_2 \leq 5 + M \cdot y$$

$$2x_3 - x_4 \leq 2 + M \cdot (1 - y)$$

redukuje na

$$2x_1 + x_2 \leq 5$$



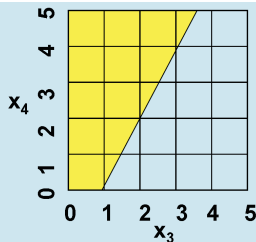
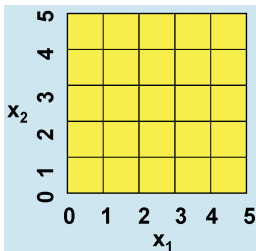
pro $y = 1$ se soustava

$$2x_1 + x_2 \leq 5 + M \cdot y$$

$$2x_3 - x_4 \leq 2 + M \cdot (1 - y)$$

redukuje na

$$2x_3 - x_4 \leq 2$$



Musí platit alespoň jedna ze dvou podmínek - domácí úkol

- Promyslete jaký prostor řešení je dán soustavou nerovnic:

$$\begin{aligned}2x_1 + x_2 &\leq 5 + M \cdot y \\ 2x_1 - x_2 &\leq 2 + M \cdot (1 - y) \\ y &\in \{0, 1\}\end{aligned}$$

- Promyslete jaký prostor řešení je dán soustavou nerovnic (všimněte si, že rovnice odpovídají rovnoběžným přímkám; mohou pro nějaké x_1, x_2 obě nerovnice platit naráz):

$$\begin{aligned}2x_1 + x_2 &\leq 5 + M \cdot y \\ 2x_1 + x_2 &\geq 10 + M \cdot (1 - y) \\ y &\in \{0, 1\}\end{aligned}$$

Platí alespoň 1 ze 2 podm. - př. nepreemptivní rozvrhování

1 $|r_j, \tilde{d}_j| C_{max} \dots$ NP-obtížný problém

- **Instance:** Množina nepreemptivních úloh $\mathcal{T} = \{T_1, \dots, T_i, \dots, T_n\}$ vykonávaná na jednom procesoru s omezeními na nejdřívější začátek r a deadline \tilde{d} každé úlohy. Délka úloh je dána vektorem p .
- **Cíl:** Nalézt proveditelný rozvrh daný začátky vykonávání úloh s tak, aby celková doba rozvrhu byla co nejkratší ($C_{max} = \max_{i \in \langle 1, n \rangle} s_i + p_i$), nebo rozhodnout, že neexistuje.

Například:

- procesor - truhlář
- T_i - výroba židle
- r_i - okamžik, kdy je k dispozici materiál na výrobu židle
- \tilde{d}_i - okamžik, kdy musí být židle vyrobena
- s_i - začátek výroby židle
- $s_i + p_i$ - konec výroby židle

Rozvrhování nepreemptivních úloh - platí alespoň jedna ze dvou podmínek

Jelikož v danou chvíli může být vykonávána maximálně jedna úloha, tak pro každou neuspořádanou dvojici úloh T_i, T_j musí platit:

- ❶ buď T_i předchází T_j (zapíšeme $s_j \geq s_i + p_i$)
- ❷ nebo T_j předchází T_i (zapíšeme $s_i \geq s_j + p_j$)

Povšimněme si, že současná platnost obou nerovnic je (pro $p_i > 0$) vyloučena.

Potřebujeme zformulovat, že platí alespoň jedna ze dvou nerovnic.

Zavedeme pomocnou proměnnou $x_{ij} \in \{0, 1\}$ tak, že $x_{ij} = 1$ právě když T_i předchází T_j .

Pro každou neuspořádanou dvojici úloh T_i, T_j zavedeme dvě nerovnice:

$$\begin{aligned} s_j + M \cdot (1 - x_{ij}) &\geq s_i + p_i \\ s_i + M \cdot x_{ij} &\geq s_j + p_j \end{aligned}$$

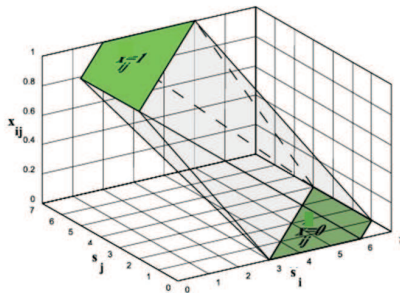
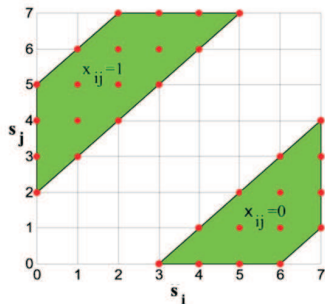
nerovnice je “vypnutá” když $x_{ij} = 0$

nerovnice je “vypnutá” když $x_{ij} = 1$

Rozvrhování - reprezentace nekonvexního prostoru

$$\begin{array}{llll}
 s_i \geq r_i & i \in 1..n & \text{release date} \\
 \tilde{d}_i \geq s_i + p_i & i \in 1..n & \text{deadline} \\
 s_j + M \cdot (1 - x_{ij}) \geq s_i + p_i & i \in 1..n, j < i & T_i \text{ předchází } T_j \\
 s_i + M \cdot x_{ij} \geq s_j + p_j & i \in 1..n, j < i & T_j \text{ předchází } T_i
 \end{array}$$

Například: $p_i = 2, p_j = 3, r_i = r_j = 0, \tilde{d}_i = 9, \tilde{d}_j = 10$



Nekonvexní dvojrozměrný prostor je průmětem dvou řezů v rovině $x = 0$ a $x = 1$ do trojrozměrného polytopu daného soustavou nerovnic.

Musí platit alespoň K z N podmínek

V případě že v ILP modelu uvažujeme případ, kdy z množiny N podmínek musí platit alespoň K podmínek ve tvaru:

$$\begin{aligned}f(x_1, x_2, \dots, x_n) &\leq b_1 \\f(x_1, x_2, \dots, x_n) &\leq b_2 \\&\vdots \\f(x_1, x_2, \dots, x_n) &\leq b_N\end{aligned}$$

Lze opět vyjádřit založením N pomocných proměnných $y_{i \in 1 \dots N} \in \{0, 1\}$

$$\begin{aligned}f(x_1, x_2, \dots, x_n) &\leq b_1 + M \cdot y_1 \\f(x_1, x_2, \dots, x_n) &\leq b_2 + M \cdot y_2 \\&\vdots \\f(x_1, x_2, \dots, x_n) &\leq b_N + M \cdot y_N \\ \sum_{i=1}^N y_i &= N - K\end{aligned}$$

Pro $K = 1$ a $N = 2$ lze zjednodušit na jedinou proměnnou y_i a její negaci vyjádřit jako $(1 - y_i)$ viz “alespoň jedna ze dvou podmínek”.

- CPLEX - komerční <http://www.ilog.com/products/cplex/>
- MOSEK - komerční <http://www.mosek.com/>
- GLPK - nekomerční <http://www.gnu.org/software/glpk/>
- LP_SOLVE - nekomerční
http://groups.yahoo.com/group/lp_solve/

YALMIP - nástroj určený k modelování ILP problémů v Matlabu
<http://control.ee.ethz.ch/~joloef/wiki/pmwiki.php>

Další skupinou algoritmů jsou metody sečných nadrovin (cutting plane method). Jsou založeny podobně jako metoda větví a mezí na opakovaném řešení úlohy LP. Výpočet je prováděn iterativně tak, že v každém kroku je přidána další omezující podmínka zužující oblast přípustných řešení. Každá nová omezující podmínka musí splňovat tyto vlastnosti:

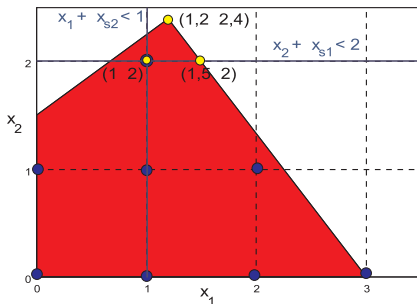
- Optimální řešení nalezené pomocí LP se stane nepřipustným.
- Žádné celočíselné řešení přípustné v předchozím kroku se nesmí stát nepřipustným.

Mezi nejznámější metody patří Dantzigovy řezy (*Dantzig cuts*), Gomoryho řezy (*Gomory cuts*) a Chvátal-Gomoryho řezy.

Algoritmus

- 1 (Inicializace) Vyřeš úlohu jako úlohu LP pomocí simplexového algoritmu.
- 2 (Test optimality) Pokud je nalezené řešení celočíselné, výpočet končí.
- 3 (Redukce) Do simplexové tabulky přidej nové omezení (Gomoryho řez). Přeoptimalizuj úlohu pomocí duálního LP a jdi na krok 2.

$$\begin{aligned} \min \quad & x_1 + 2x_2 \\ \text{s.t.} \quad & -3x_1 + 4x_2 \leq 6 \\ & 4x_1 + 3x_2 \leq 12 \\ & x_1, x_2 \geq 0, x_1, x_2 \in \mathbb{Z} \end{aligned}$$



- Úloha je NP-obtížná.
- Lze ji použít pro modelování většiny kombinatorických problémů.
- Nejčastěji je řešena metodou větví a mezí.

