VLASTNOSTI ÚLOH CELOČÍSELNÉHO PROGRAMOVÁNÍ

PRINCIP ZESILOVÁNÍ NEROVNOSTÍ A ZÁKLADNÍ METODY. METODA VĚTVENÍ A HRANIC.

TYPY ÚLOH

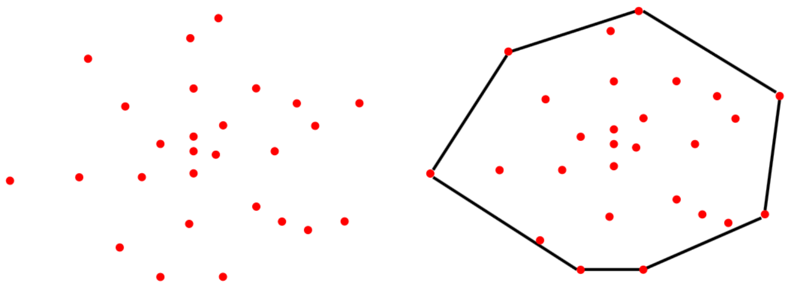
1. Úloha lineárního programování: max{$c^T x$ za podmínek $Ax \le b$, $x \in R_+^n$}
2. Úloha smíšeně-celočíselného programování: max{$c^T x + d^T y$ za podmínek $Ax + Dy \le b$, $x \in Z_+^n, y \in R_+^p$}
3. Úloha celočíselného programování: max{$c^T x$ za podmínek $Ax \le b$, $x \in Z_+^n$}
4. Úloha bivalentního programování: max{$c^T x$ za podmínek $Ax \le b$, $x \in B^n$}
5. Úloha smíšeně-celočíselného bivalentního programování: max{$c^T x + d^T y$ za podmínek $Ax + Dy \le b$, $x \in B^n, y \in R_+^p$} , B = {0,1}

VLASTNOSTI ÚLOH

Množina přípustných řešení úlohy IP: S = {$x \in Z_+^n$, $Ax \le b$}
Množina přípustných řešení úlohy LP: P = {$x \in R_+^n$, $Ax \le b$}
Množinou přípustných řešení je konvexní polyedr (pokud tato množina omezená, nazývá se polytop).
Množina přípustných řešení IP je průnikem množiny přípustných řešení LP a Z_+^n: $S = P \cap Z_+^n$. (tzv. **mřížové body**).

Bod $x \in R^n$ je **konvexní kombinací bodů** z množiny S, pokud existuje množina bodů {$x^1, x^2, \dots, x^t$} z S a parametry $\lambda^1, \lambda^2, \dots, \lambda^t \in R$ takové, že $\sum_{i=1}^{t} \lambda_i = 1$: $x = \lambda_1 x^1 + \lambda_2 x^2 + \dots + \lambda_t x^t$

Konvexní obal množiny S je množina všech bodů, které jsou konvexní kombinací bodů z S: $S \subset \mathrm{conv}(S) \subset P$. Je to nejmenší množina **S,** která celou množinu **S** obsahuje. Následující obrázek zachycuje množinu bodů a její konvexní obal.



Zdroj: http://www.surynkova.info/dokumenty/mff/PG/Prednasky/prednaska_8.pdf

Platná nerovnost k množině S je taková nerovnost, která obsahuje konvexní obal z množiny S a také body z množiny S: $\pi x \le \pi^0$, $x \in S$. Platných nerovností je nekonečně mnoho. Každá platná nerovnost je charakterizována koeficienty (π, π^0). Jestliže existuje $\lambda > 0$ takové, že $\pi^{0\prime} \le \lambda\pi^0$ a $\pi' \ge \lambda\pi$, pak je platná nerovnost $\pi' x \le \pi^{0\prime}$ silnější než platná nerovnost $\pi x \le \pi^0$. Maximální platná nerovnost je taková platná nerovnost, ke které neexistuje silnější platná nerovnost. $\{x \in R_+^n, \pi' x \le \pi^{0\prime}\} \subset \{x \in R_+^n, \pi x \le \pi^0\}$.

Opěrná nerovnost je taková nerovnost, pro kterou platí: $\exists x_0 \in S, \pi x_0 = \pi_0$, tzn. existuje alespoň jeden bod z množiny S, pro který je to splněno jako rovnost.

Fazeta (opěrná stěna) leží v konvexním obalu *S* a prochází krajem konvexního polyedru.

ZESILOVÁNÍ NEROVNOSTÍ

Chceme formulovat platnou nerovnost vzhledem k množině S, která není platná vzhledem k množině P. Existuje několik možností, jak to udělat, například zesilování nerovností s využitím dělitelnosti, lifting, fixing, přidávání dalších platných nerovností či Gomoryho nerovnosti.

ZESILOVÁNÍ NEROVNOSTÍ S VYUŽITÍM DĚLITELNOSTI

Diofantova věta říká: nechť $\pi_1 x_1 + \pi_2 x_2 + \ldots + \pi_n x_n = \pi_0, \pi_i - \text{celé}, i = 0,1,\ldots,n.$ Pak existuje celočíselné řešení právě tehdy, když největší společný dělitel koeficientů π_1, $\pi_{2\ldots}$ π_n dělí π_0. Máme-li nerovnost $\pi_1 x_1 + \pi_2 x_{2\ldots} \pi_n x_n \leq \pi^0$ a označíme-li *k* největšího dělitele π_1, $\pi_{2\ldots}$ π_n, můžeme zpřísnit nerovnost tak, že ji vydělíme *k* a pravou stranu pak zaokrouhlíme dolů (Chvátal-Gomoryho metoda). Například v rovnici $3x_1 + 6x_2 \leq 14$ je největším společným dělitelem 3. Vydělíme-li celou rovnici třemi a pravou stranu zaokrouhlíme dolů, dostaneme zpřísněnou nerovnost $x_1 + 2x_2 \leq 4$.

LIFTING spočívá v zesilování nerovností. Například:

$$4x_1 + 5x_2 + 6x_3 + 8x_4 \leq 13, \quad x_1, x_2, x_3, x_4 \in \{0,1\}.$$

$$x_1 = 1: \quad 5x_2 + 6x_3 + 8x_4 \leq 9 \longrightarrow 5x_1 + 5x_2 + 6x_3 + 8x_4 \leq 13$$

$$x_2 = 1: \quad 5x_1 + 6x_3 + 8x_4 \leq 8 \longrightarrow 5x_1 + 5x_2 + 6x_3 + 8x_4 \leq 13$$

$$x_3 = 1: \quad 5x_1 + 5x_2 + 8x_4 \leq 7 \longrightarrow 5x_1 + 5x_2 + 8x_3 + 8x_4 \leq 13$$

$$x_4 = 1: \quad 5x_1 + 5x_2 + 8x_3 \leq 5 \longrightarrow \boxed{5x_1 + 5x_2 + 8x_3 + 8x_4 \leq 13}$$

Zdroj: Ing. Jan Fábry, Ph.D.: 4EK314 Diskrétní modely

Pro *i* = 1, 2,..., *n* postupně nastavujeme $x_i = 1$ a zjišťujeme, o kolik (Δb) lze ještě snížit pravou strany nerovnice, aby nedošlo ke změně množiny přípustných řešení. Potom koeficient u x_i zvýšíme o Δb. Když nastavíme $x_1 = 1$, pak lze úvodní nerovnost přepsat jako $5x_2 + 6x_3 + 8x_4 \leq 9$. Můžeme snížit pravou stranu o 1, aniž by se změnila množina přípustných řešení. Zvýšíme tedy koeficient u x_1 o 1. Totéž uděláme pro zbylá x_i. Výsledkem je liftovaná (zesílená) nerovnost.

V dalším příkladu můžeme zesílit nerovnost tak, že snížíme hodnoty koeficientů celočíselných proměnných na hodnotu pravé strany.

$$5x_1 + 7x_2 + 10x_3 + 6x_4 \geq 4, \quad x_1, x_2 - \text{celé}, \; x_3, x_4 \geq 0.$$

$$x_1 \geq 1: \quad \longrightarrow 4x_1 + 7x_2 + 10x_3 + 6x_4 \geq 4,$$

$$x_2 \geq 1: \quad \longrightarrow \boxed{4x_1 + 4x_2 + 10x_3 + 6x_4 \geq 4.}$$

Zdroj: Ing. Jan Fábry, Ph.D.: 4EK314 Diskrétní modely

FIXING spočívá také v úpravě existujících nerovností, a to pomocí zafixování hodnot určitých proměnných. Například v úloze $2x_1 + 3x_2 + 4x_3 - 15x_4 \geq 2$, kde jsou všechny proměnné binární, nemůže být nikdy x_4 rovno 1 (nerovnost by nikdy nemohla být splněna), takže jej zafixujeme na nule. Naopak v úloze $20x_1 + 5x_2 + x_3 - 8x_4 \geq 7$ s binárními proměnnými musí být $x_1 = 1$, jinak by nemohla být nerovnost nikdy splněna, takže jej zafixujeme na 1.

PŘIDÁVÁNÍ DALŠÍCH PLATNÝCH NEROVNOSTÍ

Gomoryho řez:

Máme rovnici ve tvaru: $\sum_{j=1}^{n} \pi_j x_j = \pi_0$

Zvolíme si libovolné celé kladné číslo *d*.

Koeficienty π_j lze vyjádřit ve tvaru: $\pi_j = \alpha_j d + \pi'_j$, kde $\alpha_j = \left\lfloor \frac{\pi_j}{d} \right\rfloor$, a π_j' je celočíselný zbytek po dělení čísla π_j' číslem *d* z intervalu <0; *d*).

Rovnici lze tedy přepsat jako : $\sum_{j=1}^{n} (\alpha_j d + \pi'_j) x_j = \alpha_0 d + \pi'_0$

$$d\left(\alpha_0 - \sum_{j=1}^{n} \alpha_j x_j\right) = \sum_{j=1}^{n} \pi'_j x_j - \pi'_0$$

Levá strana je celé číslo (celočíselný násobek *d*). Proto i pravá strana musí být celé číslo, a to kladné: vpravo nemůže být $-d$, $-2d$… jelikož záporné je zde pouze π_0', které je z definice menší než *d*, neboť jde o zbytek po celočíselném dělení pravé strany číselm *d*. Takže výraz $\sum_{j=1}^{n} \pi'_j x_j - \pi'_0 \geq 0$ a stejně tak výraz $\alpha_0 - \sum_{j=1}^{n} \alpha_j x_j \geq 0$. Platí tedy:

Gomoryho zlomková nerovnost: $\sum_{j=1}^{n} \pi'_j x_j \geq \pi_0'$

Gomoryho celočíselná nerovnost: $\sum_{j=1}^{n} \alpha_j x_j \leq \alpha_0$

Příklad: máme rovnici $37x_1 - 68x_2 + 78x_3 + 1x_4 = 141$, kde jsou všechny proměnné celočíselné. Zvolíme $d = 12$. Pak $37 = 3\cdot 12 + 1$, $-68 = -6\cdot 12 + 4$, $78 = 6\cdot 12 + 6$, $1 = 0\cdot 12+1$, $141 = 11\cdot 12 + 9$.
Gomoryho zlomková nerovnost: $x_1 + 4x_2 + 6x_3 + x_4 \geq 9$
Gomoryho celočíselná nerovnost: $3x_1 - 6x_2 + 6x_3 \leq 11$

METODY ŘEŠENÍ ÚLOH IP A MIP

V určitých případech lze použít simplexovou metodu. V ní získáme vektor hodnot proměnných jako $B^{-1}b$. Pokud je *b* celočíselný vektor, pak celočíselnost tohoto výrazu bude zajištěna tehdy, pokud je B^{-1} celočíselné. Platí, že:

$$B^{-1} = \frac{B^{adj}}{\det B}$$

B^{adj} je adjungovaná matice báze, detB lze získat jako součin všech klíčových prvků od začátku výpočtu simplexovou metodou. Z tohoto vztahu je patrné, že celočíselnost bude zaručena pouze tehdy, pokud detB = ± 1. V opačném případě nelze simplex použít.

Předpokládejme, že strukturní koeficienty a hodnoty pravých stran jsou celočíselné. Základní řešení úlohy LP jsou celočíselná, jestliže pro všechny regulární matice *B* platí, že detB = ± 1 (tzn. matice B je unimodulární). To platí například pro tokové úlohy či dopravní problém. Pak lze použít simplexovou metodu.

RELAXACE

Máme úlohu celočíselného programování: $z_{IP} = \max\{c^T x : x \in S\}$, $S = \{x \in Z_+^n : Ax \le b\}$

Relaxace úlohy celočíselného programování je úloha: $z_R = \max\{z_R(x) : x \in S_R\}$, $S \subset S_R$, $c^T x \le z_R(x)$ pro $x \in S$. Hodnota z_R je horním odhadem z_{IP}.

Lineární relaxace je úloha LP bez podmínek celočíselnosti:

1) $z_{LP} = \max\{c^T x : x \in P\}$, $P = \{x \in R_+^n : Ax \le b\}$

2) $z_Q = \max\{c^T x : x \in Q\}$, $S = \{x \in Z_+^n : Ax \le b\}$, $conv(S) \subset Q$

Lagrangeova relaxace je úloha, kterou získáme z úlohy IP tak, že soustavu omezení *Ax* ≤ *b* rozdělíme na dvě soustavy,

$Ax \le b:\ A^1 x \le b^1$ m_1 omezujících podmínek

$A^2 x \le b^2$ m_2 omezujících podmínek,

a omezující podmínky $A^1 x \le b^1$ přesuneme do účelové funkce:

$z(\lambda, x) = c^T x + \lambda^T (b^1 - A^1 x)$, $x \in R_+^{m_1}$, kde λ je vektor Lagrangeových multiplikátorů.

Lagrangeova relaxace je tedy úloha ve tvaru *LR*(λ): $z_{LR}(\lambda) = \max\{z(\lambda, x) : A^2 x \le b^2, x \in Z_+^n\}$.

Význam Lagrangeovy relaxace je ten, že když odstraníme nerovnosti $A^1 x \le b^1$ z omezení úlohy při vhodné volbě těchto omezení, je úloha *LR*(λ) snadno řešitelná. Pro každé $\lambda \in R_+^{m_1}$ je $z_{LR}(\lambda)$ horním odhadem optimální hodnoty účelové funkce. Chtěli bychom přirozeně najít co nejmenší číslo $z_{LR}(\lambda)$ pro λ ≥ 0. Formulujeme tedy Lagrangeovu duální úlohu $z_{LD} = \min_{\lambda \ge 0} z_{LR}(\lambda)$.

METODY ŘEZNÝCH NADROVIN

Tyto metody se hodí pro IP a MIP (ne pro bivalentní úlohy). Patří sem Gomoryho metoda. Uvažujme úlohu s množinou přípustných řešení $S = \{x \in Z_+^n : Ax \le b\}$. Množina přípustných řešení této úlohy bez podmínek celočíselnosti je $P = \{x \in R_+^n : Ax \le b\}$. Všechny krajní body množiny *conv*(*S*) jsou celočíselné, ale většinou různé od krajních bodů množiny *P*. Platí: S ⊂ conv(S) ⊂ P. Chtěli bychom odstranit z množiny *P* body, ležící v *P – conv(S)* tak, aby krajní body vzniklé množiny byly již celočíselné. To se dělá pomocí přidávání platných nerovností vzhledem k *S* a *conv(S)*.

1. Vyřešíme úlohu IP bez podmínek celočíselnosti (lineární relaxací), běžně simplexovou metodou. Pokud tak najdeme celočíselné řešení, výpočet končí.
2. Pokud ne, přidáme omezení, které odřízne z množiny přípustných řešení nějakou podmnožinu. Nesmíme však přitom přijít o žádné přípustné celočíselné řešení. Vybereme proměnnou, jejíž hodnota v optimálním řešení nesplňuje podmínku celočíselnosti. Všechny hodnoty pravých stran simplexové tabulky lze vyjádřit jako $\beta_i = [\beta_i] + \pi_{i0}$, kde $[\beta_i]$ je nejbližší nižší celé číslo k hodnotě β_i, a π_{i0} je tedy z intervalu <0; 1). Vezmeme řádek simplexové tabulky, kde je hodnota π_{i0} největší, a z tohoto řádku vytvoříme Gomoryho nerovnost $\sum_{j=1}^{n+m} (-\pi'_j) x_j \le -\pi_0'$.

   Nerovnost vyrovnáme na rovnici $\sum_{j=1}^{n+m} (-\pi'_j) x_j + x_{n+m+1} = -\pi_0'$ a přidáme ji do simplexové tabulky jako další řádek, přičemž proměnná x_{n+m+1} bude základní proměnnou v tomto nově přidaném řádku.

3. Protože hodnota $-\pi_0'$ je záporná, řešíme úlohu duálně simplexovou metodou do optima. Pak se opět podíváme, jestli je řešení celočíselné. Pokud ne, postup přidáme další nerovnost atd. až do nalezení celočíselného řešení.

   Například mějme následující optimální neceločíselné řešení úlohy

zákl. proměnné	x_1	x_2	x_3	x_4	b_i
x_1	1	0	2/5	-3/5	8/5
x_2	0	1	-1/5	4/5	11/5
z_j	0	0	1/5	6/5	49/5

Zdroj: Jablonský, J., Lagová, M.: Lineární modely (2009).

r_{10} = 3/5 (8/5 = 1 + 3/5) a r_{20} = 1/5, takže vezmeme první řádek a vytvoříme omezení ve tvaru $-2/5x_3 - 2/5x_4 + x_5 = -3/5$, které přidáme do tabulky. Ta se tudíž rozroste o jeden řádek a jeden sloupec. Dále řešíme duálně simplexovou metodou.

Jde o nepolynomiální algoritmus a není příliš efektivní, protože počet přidaných řádků a proměnných rychle narůstá. Software většinou používá metodu větvení a mezí.

METODA VĚTVENÍ A MEZÍ

Tato metoda patří mezi kombinatorické metody. Nejprve se vypočte optimální řešení bez uvažování podmínek celočíselnosti (simplexem). Pokud je celočíselné, výpočet končí. Jinak se z množiny přípustných řešení vytvoří dvě podmnožiny, a to tak, že se vezme některá neceločíselná proměnná x_k, a první podmnožina se rozšíří o podmínku $x_k \le \lfloor x_k^0 \rfloor$, zatímco druhá o podmínku $x_k \ge \lfloor x_k^0 \rfloor + 1$. V každé z těchto podmnožin se vypočte optimální řešení a případně se tyto podmnožiny dále rozvětví atd. Zároveň se v každé vzniklé větvi odvozuje horní mez (při maximalizaci) pro hodnotu účelové funkce celočíselného řešení. Každá větev musí být uzavřena jedním ze tří způsobů:

1. je v ní nalezeno řešení vyhovující podmínkám celočíselnosti,

2. neexistuje v ní přípustné řešení,

3. je v ní nalezeno neceločíselné řešení, ale horní mez pro hodnotu účelové funkce odvozená z tohoto řešení je nižší než hodnota účelové funkce celočíselného řešení z dříve prohledaných větví.

Algoritmus (uvažujeme maximalizační úlohu)

M – posloupnost, v níž se nacházejí úlohy, které řešíme v jednotlivých větvích
x, z** - nejlepší nalezené celočíselné řešení, nejlepší hodnota účelové funkce

1. **krok: počáteční nastavení**
   M = {původní úloha bez podmínek celočíselnosti}
   $z^* = -\infty$
   *x** - není definováno

2. **krok: výběr úlohy**
   M je prázdná → konec, tisk *x*, z**
   M není prázdná → vybereme z posloupnosti *M* poslední úlohu

3. **krok: řešení vybrané úlohy**

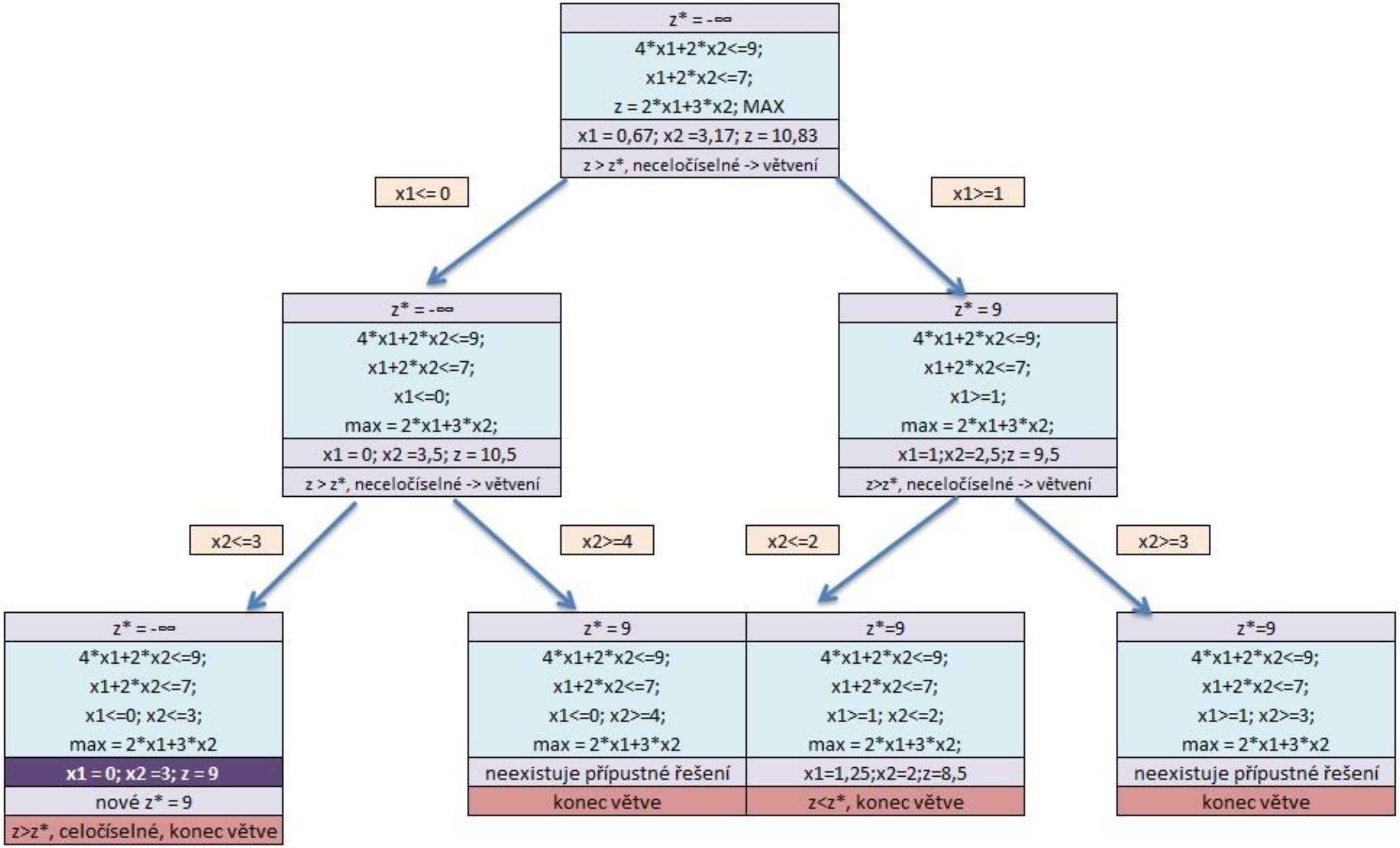
a) neexistuje přípustné řešení → odstraníme úlohu z posloupnosti *M* → 2. krok
b) optimální řešení x^0, z^0:
   B1) $z^0 \le z^* \rightarrow$ odstraníme úlohu z posloupnosti *M* → 2. krok
   B2) $z^0 > z^*, x^0$ celé → $x^*=x^0$, $z^*=z^0$ → odstraníme úlohu z posloupnosti *M* → 2. krok
   B3) $z^0 > z^*, x^0$ není celé → 4. krok (větvení)

4. **krok: větvení**
   Zvolíme větvící proměnnou x_k, jejíž hodnota x_k^0 není celé číslo
   →do *M* přidáme kopii úlohy řešené ve 3. kroku, ke které přidáme omezení I: $x_k \le \lfloor x_k^0 \rfloor$
   → k předposlední úloze v *M*, řešené ve 3. kroku přidáme omezení II: $x_k \ge \lfloor x_k^0 \rfloor + 1$
   → 2. krok

Existují různé doplňující strategie ohledně toho, jakým způsobem procházet uzly stromu řešení (v jaké posloupnosti úlohy řešit). Mohou se vybrat uzly v pořadí, v jakém byly vytvářeny (zleva doprava), nebo podle nejvyšší horní hranice účelové funkce, nebo se zpracovávají po vrstvách stromu (1. vrstva, pak 2. vrstva, …), výběr uzlů podle … Existují i doplňující postupy pro výběr větvící proměnné, např. výběr proměnné, která je nejblíže polovině dvou celých čísel.

Kombinací metody větvení a mezí a řezných nadrovin je metoda Branch and cut, metoda větvení a řezů. V ní se přidávají řezy, dokud lze nalézt platné omezení, nebo do okamžiku, kdy přidávání dalších řezů již nemá výrazný vliv na hodnotu účelové funkce, a pak následuje větvení. Díky tomu nevznikne tolik větví.

Pro úlohy IP s velkým počtem proměnných, kde je v optimálním řešení je jen malý podíl proměnných nenulový, se hodí metoda Branch and Price, větvení a oceňování.


z* = -∞
4*x1+2*x2<=9;
x1+2*x2<=7;
z = 2*x1+3*x2; MAX
x1 = 0,67; x2 =3,17; z = 10,83
z > z*, neceločíselné -> větvení
x1<= 0
x1>=1
z* = -∞
4*x1+2*x2<=9;
x1+2*x2<=7;
x1<=0;
max = 2*x1+3*x2;
x1 = 0; x2 =3,5; z = 10,5
z > z*, neceločíselné -> větvení
z* = 9
4*x1+2*x2<=9;
x1+2*x2<=7;
x1>=1;
max = 2*x1+3*x2;
x1=1;x2=2,5;z = 9,5
z>z*, neceločíselné -> větvení
x2<=3
x2>=4
x2<=2
x2>=3
z* = -∞
4*x1+2*x2<=9;
x1+2*x2<=7;
x1<=0; x2<=3;
max = 2*x1+3*x2
x1 = 0; x2 =3; z = 9
nové z* = 9
z>z*, celočíselné, konec větve
z* = 9
4*x1+2*x2<=9;
x1+2*x2<=7;
x1<=0; x2>=4;
max = 2*x1+3*x2
neexistuje přípustné řešení
konec větve
z*=9
4*x1+2*x2<=9;
x1+2*x2<=7;
x1>=1; x2<=2;
max = 2*x1+3*x2;
x1=1,25;x2=2;z=8,5
z<z*, konec větve
z*=9
4*x1+2*x2<=9;
x1+2*x2<=7;
x1>=1; x2>=3;
max = 2*x1+3*x2
neexistuje přípustné řešení
konec větve

VÝPOČETNÍ SLOŽITOST ÚLOH

Mnoho úloh většího rozsahu neumíme řešit v rozumném výpočetním čase. Rozsahem úlohy se rozumí počet proměnných a počet omezení, v teorii grafů pak počet uzlů a hran. Efektivní algoritmus je takový algoritmus, kterým dokážeme úlohu vyřešit v rozumném výpočetním čase. Výpočetní čas se měří počtem elementárních operací fiktivního počítače. Výpočetní složitostí úlohy *f(n)* se myslí maximální výpočetní čas pro všechna číselná zadání úlohy o rozměru *n* (čas se liší i podle konkrétních čísel v úloze). Přesný tvar funkce *f(n)* není třeba znát, většinou zjišťujeme asymptotický charakter růstu funkce *n*. Hledáme funkci $g(n) = O(f(n))$. To znamená, že existuje konstanta *c* taková, že pro všechna přirozená *n* platí $|f(n)| \leq c|g(n)|$. Pokud je *g(n)* např. n^2, n^3, $\log(n)$, je růst výpočetního času polynomiální, jde o polynomiální algoritmus. Pokud je *g(n)* např. e^n, 2^n, $n!$, jde o nepolynomiální algoritmus. Například simplex nepatří mezi polynomiální algoritmy. $g(n) = (O(f(n))$ je tzv. Omikron notace, horní odhad. Dále se někdy používá Theta notace $g(n) = (\Theta(f(n))$, což je průměrný odhad, či Omega notace $g(n) = (\Omega(f(n))$, což je dolní odhad.

g(*n*)	*n* = 10	*n* = 30	*n* = 50	*n* = 70
n^2	0,000001 s	0,00003 s	0,00005 s	0,00007 s
n^3	0,001 s	0,027 s	0,125 s	0,343 s
n^5	0,1 s	24,3 s	5,2 min	28 min
2^n	0,001 s	17,9 min	35,7 roků	37 mil. roků
3^n	0,059 s	6,5 roku	$2{,}3 \times 10^{10}$ roků	10^{13} mil. roků

Výpočetní složitost. Zdroj: Ing. Jan Fábry, Ph.D.: 4EK314 Diskrétní modely

Existují 4 základní třídy úloh: P, NP, NPC a NPH.

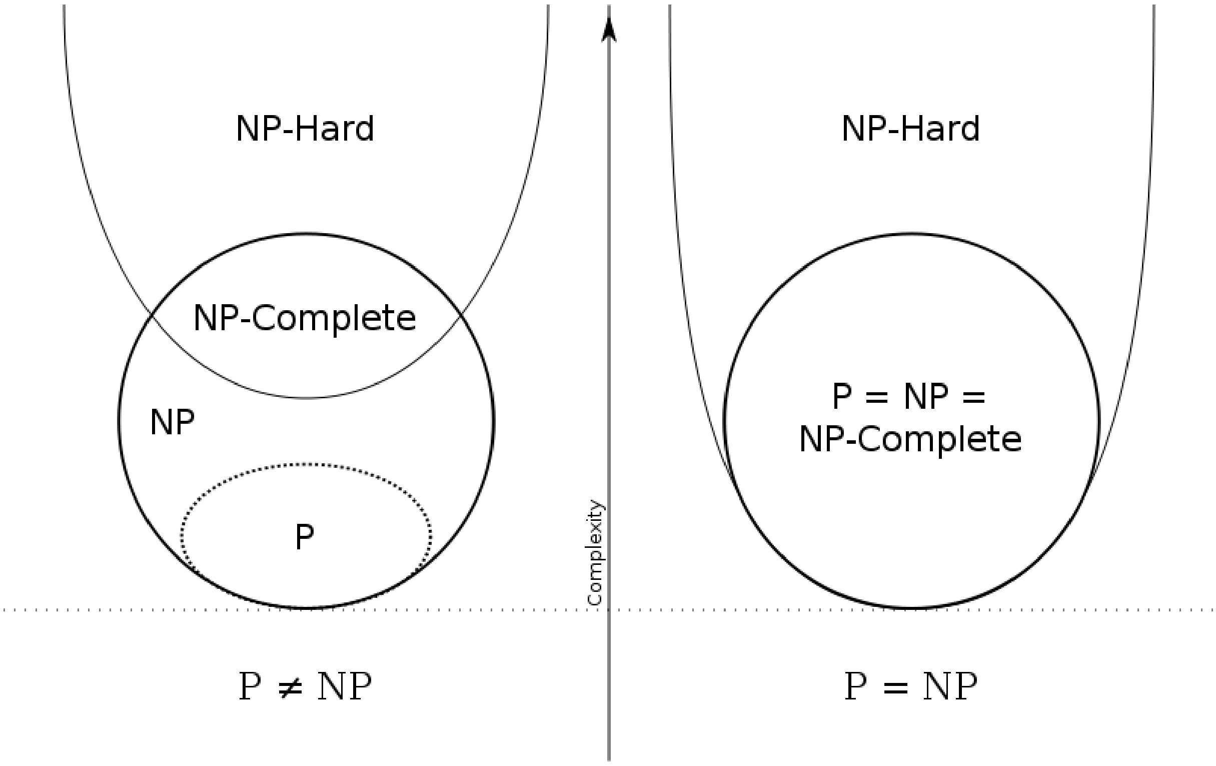
Třídu P tvoří úlohy, pro které existuje polynomiální algoritmus. Sem patří úlohy lineárního programování, hledání minimální kostry grafu, maximálního toku, nejkratší cesty, přiřazovací problém či hledání odpovědi na otázku, jestli v grafu existuje Eulerův cyklus.

Třídu NP tvoří rozhodovací úlohy typu ANO/NE. Jsou to úlohy, pro které existuje nedeterministický polynomiální algoritmus, který tuto úlohu řeší ve dvou fázích: 1. vygeneruje se náhodně nějaké řešení, na jehož existenci se ptáme v dané úloze. 2. ověří se, jestli je toto řešení hledaným řešením dané úlohy, a to polynomiálním algoritmem. Tímto polynomiálním algoritmem tedy jen ověříme, jestli vygenerované řešení je právě tím řešením, o jehož existenci se má rozhodnout. Jde spíše o teoretickou konstrukci k vymezení úloh NP. Patří sem například otázka, jestli existuje Hamiltonův cyklus. Máme seznam hran a uzlů grafu. Vybereme nějaký seznam hran a ptáme se, jestli tvoří Hamiltonův cyklus. To lze ověřit polynomiálním algoritmem.

K definici úloh **třídy NPC** je třeba zavést pojem polynomiální redukovatelnost. Úloha A je polynomiálně redukovatelná na úlohu B, jestliže existuje zadání úlohy B takové, že z výsledku řešení úlohy B lze odvodit řešení úlohy A. Toto odvození musí být realizováno polynomiálním algoritmem. Úloha A je speciálním případem úlohy B. Úloha B je obecnější, a tedy i obtížnější. Třída NPC obsahuje takové úlohy L z NP, pro kterou platí, že každá úloha z NP se dá redukovat na úlohu L. Je-li úloha *A* NP-úplná a úlohu *A* lze redukovat na úlohu *B z* NP, pak i *B* je NP-úplná úloha. Chceme-li dokázat, že úloha je NP-úplná, musíme ukázat, že tato úloha patří do třídy NP, a nalézt NP-úplnou úlohu, která se dá na danou úlohu redukovat (která je méně obtížná než daná úloha). Patří sem například úloha hledání Hamiltonova cyklu, binární forma úlohy batohu, set covering problem…

Třída NPH je třída nejobtížnějších úloh. Existuje-li pro úlohu ze třídy NPH polynomiální algoritmus, pak je možné pomocí něj polynomiálně vyřešit všechny úlohy z NP. Patří sem například úloha obchodního cestujícího, batohu, Steinerova stromu, celočíselné programování.

Zatím se neví, jestli NP = P nebo ne (spíš asi ne). Tahle otázka patří mezi Millennium Prize Problems. Kdo to vyřeší, dostane 1 000 000 USD. Pro zájemce viz http://www.claymath.org/millennium-problems.



Zdroj: Wikipedia.org

Tedy: NP jsou všechny úlohy typu ANO/NE, u nichž můžeme v polynomiálním čase **ověřit**, zda náhodně vygenerované řešení je hledaným řešením úlohy. P jsou všechny úlohy, které mohou být v polynomiálním čase **vyřešeny**. P jsou tedy podmnožinou NP. Úloha z NP patří do NPC pouze tehdy, pokud se na ni dá každý problém ze třídy NP redukovat (tzn. pokud lze z řešení této úlohy polynomiálně odvodit řešení ostatních úloh v NP. To zároveň znamená, že pokud by některou z úloh NPC bylo možné vyřešit polynomiálním algoritmem, pak by bylo možné vyřešit všechny úlohy v NP). NPH je skupina nejobtížnějších úloh. Všechny NPC úlohy jsou zároveň NPH, ale ne všechny NPH úlohy jsou NPC.

ZDROJE:

Ing. J. Fábry, Ph.D.: přednášky 4EK314 Diskrétní modely, 2011.

Jablonský, J., Lagová, M.: Lineární modely. Nakladatelství Oeconomica, Praha 2009.