Úloha: minimální vrcholové pokrytí

$G = (V, E)$

$V^* \subseteq V, \ \forall e = \{u, v\} : u \in V^* \vee v \in V^*$

1) proměnné: x_v, $\forall v \in V : x_v \in \{0, 1\}$

2) omezení: $\forall e = \{u, v\} : x_u + x_v \geq 1$

3) kritérium: $\min \sum_{v=1}^{|V|} x_v$

Formulace v Gurobi:

```
import gurobipy as g

n=6
edges = [(1,2),

model = g.Model()

x = model.addVars(n, vtype = g.GRB.BINARY)

for u,v in edges:
    model.addConstr(x[u] + x[v] ≥ 1)

expr = 0
for idx in range (n)
    expr += x[idx]
model.setObjective(expr, g.GRB.MAXIMIZE)

model.optimize()
```

Řešení splnitelnosti logických výroků

$\varphi = (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2 \vee x_3)$

1) proměnné: $x_1, x_2, x_3 \in \{0, 1\}$

2) omezení: $c_1 = x_1 + (1 - x_2) \geq 1$

$c_2 = (1 - x_1) + x_2 + x_3 \geq 1$

3) kritérium: není nutné, zajímá nás pouze splnitelnost

Přednáška 5.3.2019 Formulace úloh ILP

Př ①: Investice do nemovitostí: máme 14 miliard a chceme největší výnos

číslo budovy	1	2	3	4	5	6
cena [mil]	5	7	4	3	4	6
výnos [mil]	16	22	12	8	11	19

(každou budovu lze koupit jen jednou)

$\max(16x_1 + 22x_2 + 12x_3 + 8x_4 + 11x_5 + 19x_6)$ ← maximální zisk

s.t.: $5x_1 + 7x_2 + 4x_3 + 3x_3 + 4x_5 + 6x_6 \leq 14$ ← můžeme utratit jen 14 mil

$x_{1..6} \in \{0, 1\}$ ← $x_i = 1$ když kupujeme i

$x_i = 0$ když nekupujeme i

- pokud koupíme budovu 1, musíme koupit i budovu 2

  $x_2 \geq x_1$

- pokud koupíme budovu 3, nemůžeme koupit budovu 4

  $x_4 \geq x_3$ ✗ první podmínka je špatně, můžeme koupit 3 a nekoupit 4:

  $x_3 + x_4 \leq 1$

  opraveno:

  $x_3 + x_4 \leq 1$

- musíme koupit jednu z budov 5, 6, ale ne obě najednou

  $x_5 + x_6 = 1$

- musíme koupit budovu 1, nesmíme koupit budovu 2

  $x_1 = 1$

  $x_2 = 0$

- musíme koupit aspoň 3 budovy

  $\sum_{i=1 \dots 6} x_i \geq 3$

  (přesně tři budovy)

  $\sum x_i = 3$

- pokud jsme koupili budovy 1 a 2, musíme koupit i 3

  $x_1 + x_2 \leq x_3 + 1$

  ověření:

	✓	✓	✓	✓	✓	✓	✗	✓
x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1

- nesmíme koupit právě dvě budovy

  $x \geq 3$ nebo $x \leq 1 \approx -x \geq -1$

  $x \geq 3 + My$

  $-x \geq -1 + M(y-1)$

  zvolím $M=3$:

  $x \geq 3 + 3y$

  $-x \geq -1 + 3(y-1)$

domácí úkol

Př. 2 Výroba oblečení: máme 150 hodin a 160 materiálu, chceme max. zisk

produkt	triko $= t = x_1$	košile $= k = x_2$	kalhoty $= d = x_3$ (džíny)
doba výroby	3	2	6
materiál	4	3	4
zisk	6	4	7

$$\max \; 6t + 4k + 7d$$

$$\text{s.t.:} \quad 3t + 2k + 6d \leq 150$$

$$4t + 3k + 4d \leq 160$$

- na výrobu si pronajímáme stroje:

cena stroje	200	150	100

Zavedeme binární proměnnou $y_i \in \{0,1\}$ $\begin{cases} = 1 \text{ pokud máme stroj} \\ = 0 \text{ pokud nemáme stroj} \end{cases}$

Kriteriální fce se změní: $\max \; 6t + 4k + 7d - 200y_1 - 150y_2 - 100y_3$

Dál chceme $y_i = 0$ iff $x_i = 0$, $y_i = 1$ iff $x_i \geq 1$, přidáme omezení:

$x_i \leq x_{max} \cdot y_i$, x_{max} je nejvyšší možná hodnota, jaké x_i může dosáhnout:

$x_{max} = \min(150/2, 160/3) = \min(75, 54) = 54$, nemůžeme udělat víc než 54 výrobků, protože by nám došel materiál

$x_i \geq y_i$, to nám stlačí y_i na nulu, když je x_i nulové. V tomhle případě to můžeme vynechat, protože za $y_i = 1$ máme penalizaci ve kriteriální fci

Přidáme proto jedno nové omezení: $x_i \leq 54 y_i$

- chceme, aby práce trvala přesně 40, 80 nebo 120 hodin

Přidáme pomocné proměnné v_i:

$$3x_1 + 2x_2 + 6x_3 = 40v_1 + 80v_2 + 120v_3$$
$$v_1 + v_2 + v_3 = 1$$

Máme dvě omezení a chceme, aby platilo aspoň jedno z nich

$$2x_1 + 2x_2 \leq 8$$
$$2x_3 - 2x_4 \leq 2$$

použijeme "big M", kde $M \in \mathbb{N}$, $y \in \{0,1\}$

$$2x_1 + 2x_2 \leq 8 + \underbrace{M}_{M=15} \cdot y$$
$$2x_3 - 2x_4 \leq 2 + M(1-y)$$

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

pro $y = 0$:

$$2x_1 + 2x_2 \leq 8$$
$$2x_3 - 2x_4 \leq 2 + M$$

← M máme zvolené tak, aby tohle vždy platilo, takže se problém redukuje na jedno omezení

pro $y = 1$:

$$2x_1 + 2x_2 \leq 8 + M$$
$$2x_3 - 2x_4 \leq 2$$

DÚ: Nakreslete prostor řešení úlohy s omezeními:

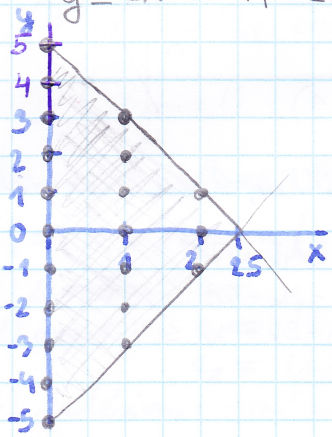
1) $2x + y \leq 5 + Mz$
$2x - y \leq 2 + M(1-z)$
$z \in \{0,1\}$

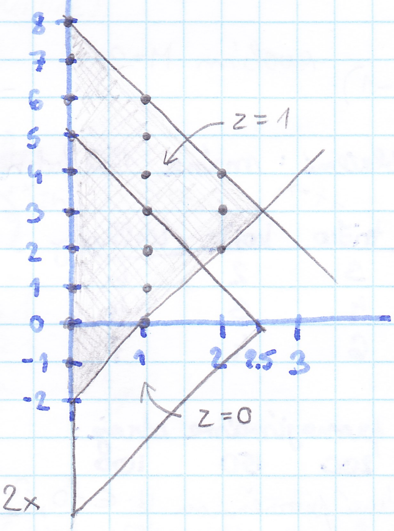
$2x + y \leq 5 + Mz \qquad 2x - y \leq 2 + M(1-z)$
$y \leq -2x + 5 + Mz \qquad y \geq 2x - 2 - M(1-z)$

Aby to šlo nakreslit, volím třeba $M = 3$

pro $z = 0$:
$y \leq -2x + 5$
$y \geq 2x - 2 - M = 2x - 5$

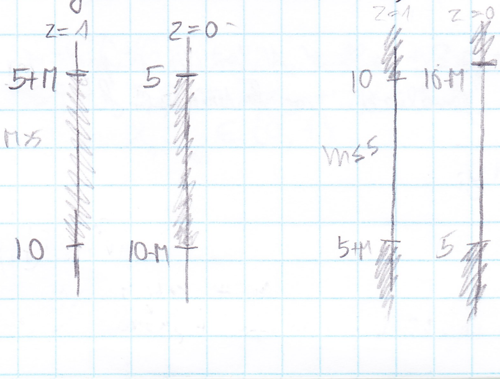
pro $z = 1$:
$y \leq -2x + 5 + M = -2x + 8$
$y \geq 2x - 2$



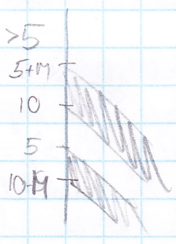


2) $2x + y \leq 5 + Mz \quad \approx \quad y \leq 5 + Mz - 2x$
$M(1-z) + 2x + y \geq 10$
$z \in \{0,1\}$

$$M(1-z) + 2x + y \geq 10$$
$$y \geq 10 - 2x - M(1-z)$$

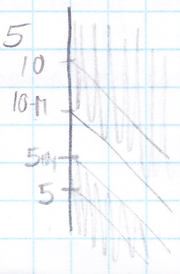


pro $M > 5$



Jednou nerovnicí to nahradit nejde, protože to má různé nerovnosti (≥, ≤).

pro $M \leq 5$



Je možné, aby v předchozím případě pro nějaké x,y platily obě nerovnice?

Ano, zvolím $M = 6$ a $x = 0$. Pak:

$y \le 5 + Mz = 5 + 6z$

$y \ge 10 - M(1-z) = 4 + 6z$

Zkusím $x = 0$, $y = 11$, $z = 1$:

$11 \le 5 + 6$ ✓

$11 \ge 4 + 6$ ✓

Aplikace ILP: rozvrhování: máme sadu úkolů, dobu jejich trvání p_i, čas odkdy je můžeme řešit r_i, dokdy musí být hotové d_i

$\max \; s_i + p_i$

s.t. $s_i \ge r_i$ — začneme, teprve když můžeme začít

$d_i \ge s_i + p_i$ — dokončíme to včas

$s_j + M(1 - x_{ij}) \ge s_i + p_i \;,\; j<i$ — T_i je před T_j

$s_i + M(x_{ij}) \ge s_j + p_j \;,\; j<i$ — T_j je před T_i

Když chceme, aby aspoň K z našich N omezení platilo:

$f_1(x) \le b_1$

$f_2(x) \le b_2$

$\vdots$

$f_N(x) \le b_N$

přidáme $y_i \in \{0,1\}$

$f_1(x) \le b_1 + y_1 M$

$f_2(x) \le b_2 + y_2 M$

$\vdots$

$f_N(x) \le b_N + y_N M$

$\sum_{i=1} y_i = N - K$

Přednáška 12.3.2019

Nejkratší cesty

Varianty nejkratších cest:

- ze zdroje do cíle
- ze zdroje do všech vrcholů (Shortest Path Tree – SPT)
- mezi všemi dvojicemi (All pairs shortest path)
- ze všech vrcholů do jednoho cíle

Nejdelší cesta: invertujeme ceny hran a najdeme nejkratší cestu

Nejkratší cesta v grafu s oceněnými vrcholy: vrchol nahradíme za v_1 v_2 tak, že hrany vstupují do v_1 a vystupují z v_2. Mezi v_1 a v_2 je hrana s cenou vrcholu.

V následujících algoritmech předpokládáme, že v grafu nejsou cykly záporné délky, v tom případě je totiž hledání nejkratší cesty NP úplný problém.

Věta: Pokud v grafu existuje cesta mezi u a v, pak mezi nimi existuje i nejkratší cesta. V grafu ale nesmí být záporné cykly.

Věta: Pokud v grafu není cyklus záporné délky, pak nejkratší sled z s do t obsahuje nejkratší cestu z s do t.

Věta: Pokud v grafu není cyklus záporné délky, pak $\forall i,j,k: l(i,j) \le l(i,k) + l(k,j)$

Dijkstra algorithm

- vrcholy si udržuje seřazené podle toho, za kolik se do nich dostane
- vybere nejlevnější vrchol a pro odchozí hrany spočítá, za kolik se dostane do jejich sousedů.
- jakmile dojde do cíle nebo nejsou ve frontě vrcholy, končí

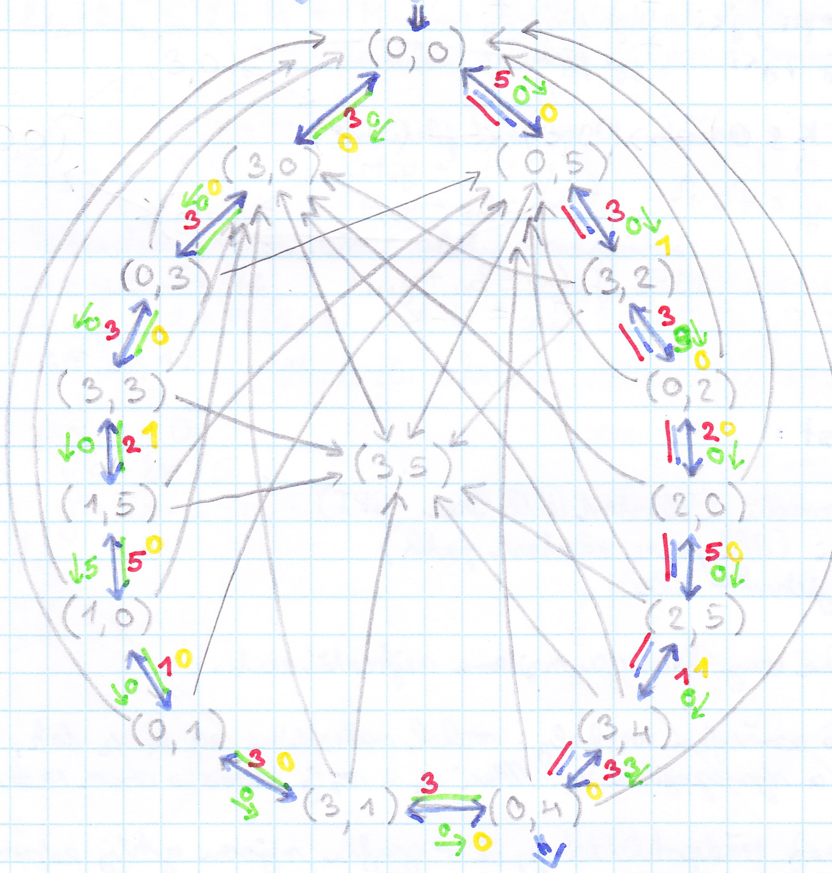
DÚ: Jaký je rozdíl mezi stromem SPT od Dijkstry a MST od Prima? Prim bere nejprve nejlevnější hrany, dijkstra nejlevnější vrcholy. Stromy budou stejné, pokud bude dijkstra spuštěn z kořenu MST. Liší se např. pro : Prim , Dijkstra: , i když záleží na implementaci.

Dijkstra má časovou náročnost $O(n^2)$, případně $O(m + n\log n)$ s prioritní frontou

Pro specifické grafy existují i algoritmy s lineární složitostí
- planární grafy
- neorientované grafy s cenami z $\mathbb{N}$
- orientované grafy bez cyklů (DAGy)

Vylepšení Dijkstry: A^*
- máme heuristiku $h(v, t)$, která dává minimální odhad ceny z v do t
- musí platit $c(v_1, v_2) \geq h(v_1, t) - h(v_2, t)$
- vrchol vybíráme podle $\ell(v) = \min\{\ell(i) + h(i, t)\}$
- je rychlejší než Dijkstra, ale ve špatných případech naopak musí projít vše.

Př: Máme dvě nádoby o objemu 3 a 5 litrů. Chceme získat 4 l vody.



a) jaká bude nejkratší cesta, pokud chceme dělat co nejméně operací?
b) ——— " ——— přelévat co nejmenší objem vody?
c) ——— " ——— vylít co nejméně vody zpět do řeky?

a) jdeme po pravé straně, provedeme 7 operací (vlevo 8)
b) jdeme ——— " ——— přelijeme 22 l vody (vlevo 23)
c) jdeme po levé straně, vylijeme 5 l vody (vpravo 6)

DÚ: Minimalizujte počet operací, kde se přelévá se slitím
zleva: 1 přelití, zprava 2 přelití ⇒ půjdu zleva

DÚ: Lze s nádobami 6 l a 4 l získat 5 l?
Ne. Během přelévání vlastně neustále lejeme jednu nádobu do druhé (zleva lijeme pořád 3 do 5, zprava obráceně). To je stejné jako počítat modulo. S nádobami o objemech m, n a cílovém objemu v:

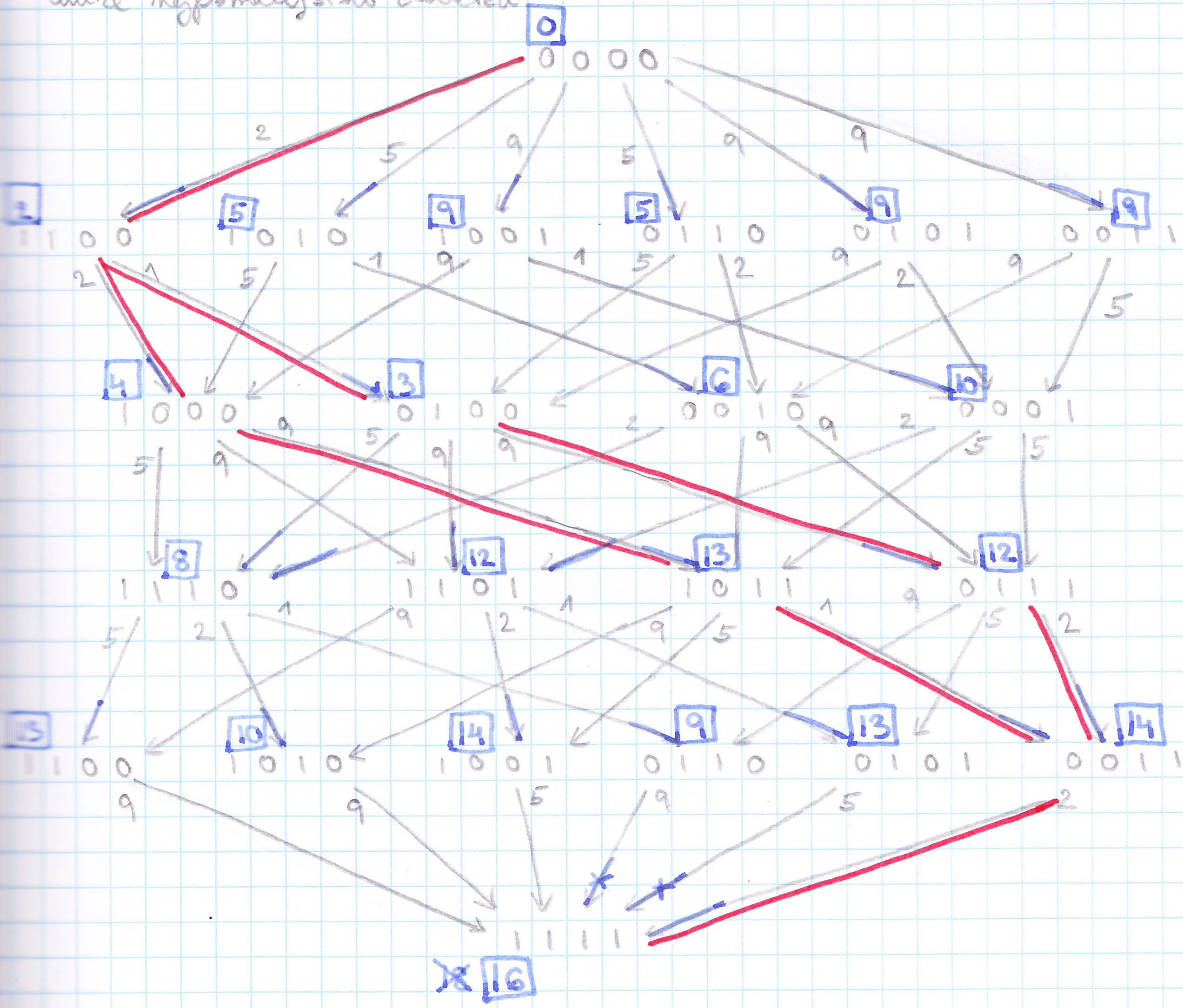
$x \cdot m = v \bmod n$

$xm + yn = v$

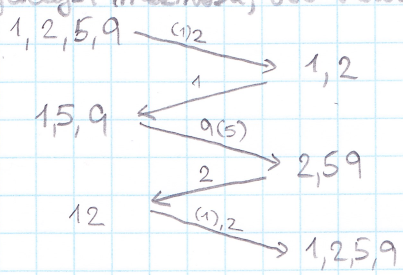
Podle Bezoutovy věty má soustava řešení, když $v / \gcd(m,n)$, což se o $5/\gcd(4,6) = 5/2$ říct nedá.

DÚ: Formulujte problém jako nejkratší cestu: 4 lidé přijdou k mostu v noci. Přejít ho mohou nejvýše ve dvojici, a vždy musí mít s sebou svítilnu. Svítilnu mají jen jednu, a přechod mostu trvá $L = \{1, 2, 5, 9\}$ minut.

Vrcholy označím podle toho, jací lidé už most přešli. Cena hrany je doba chůze nejpomalejšího člověka.



Jsou dvě nejrychlejší možnosti, obě budou trvat 16 minut



DÚ: Každou hranu v komunikační síti můžeme ohodnotit její spolehlivostí $p(i,j)$. Chceme najít spojení z s do t, aby bylo co nejspolehlivější.

a) Jak lze max. reliability převést na nejkratší cestu?

$P = p(s,b) \cdot p(b,c) \cdot \ldots \cdot p(x,t)$. Protože ceny nesčítáme, ale násobíme, budou inverzní ceny $1/p(i,j)$ a ne $0 - p(i,j)$. Použijeme Dijkstrův alg.

b) Jaký by byl algoritmus pro řešení úlohy $O(n^2)$ bez převodu na SPT?

Upravím Dijkstrův alg tak, aby se nejprve braly vrcholy s nejvyšší dosaženou spolehlivostí

c) Budou algoritmy v a), b) fungovat, pokud budou některé $p(i,j) > 1$?

Nejsem si jistá, jestli chápu úlohu správně, spolehlivost se podle mě vyjadřuje jako hodnota mezi 0 a 1 včetně.