

# 4EK314 Diskrétní modely

Jan Fábry

Fakulta informatiky a statistiky  
Katedra ekonometrie

fabry@vse.cz  
<https://janfabry.cz>

Únor 2024, Praha

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskretních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskrétních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

# Úloha celočíselného programování

Obecná úloha lineárního programování (LP):

$$z_{\text{LP}} = \max\{c^T x : Ax \leq b, x \in \mathbb{R}_+^n\}. \quad (1)$$

Úloha celočíselného programování (IP):

$$z_{\text{IP}} = \max\{c^T x : Ax \leq b, x \in \mathbb{Z}_+^n\}. \quad (2)$$

$z_{\text{LP}} \geq z_{\text{IP}}$  neboť  $\mathbb{Z}_+^n \subset \mathbb{R}_+^n$

$P = \{x : Ax \leq b, x \in \mathbb{R}_+^n\}$ ,  $S = \{x : Ax \leq b, x \in \mathbb{Z}_+^n\}$ ,  $S \subset P$

Úloha smíšeně-celočíselného programování (MIP):

$$z_{\text{MIP}} = \max\{c^T x + h^T y : Ax + Gy \leq b, x \in \mathbb{R}_+^n, y \in \mathbb{Z}_+^p\}. \quad (3)$$

Úloha bivalentního programování (BIP):

$$z_{\text{BIP}} = \max\{c^T x : Ax \leq b, x \in \mathbb{B}^n\}, \mathbb{B} = \{0, 1\}. \quad (4)$$

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskrétních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

## 1. Úloha výrobního plánování

**Proměnné:**  $x_i$  je počet kusů  $i$ -tého typu produktu

**Omezující podmínky:**  $x_1, x_2, \dots, x_n$  - celé

## 2. Řezná úloha

**Proměnné:**  $x_i$  je počet kusů původního materiálu rozděleného podle  $i$ -tého řezného schématu

**Omezující podmínky:**  $x_1, x_2, \dots, x_n$  - celé

**Účelová funkce:**

- minimalizace počtu rozřezaných kusů původního materiálu
- minimalizace odpadu
- maximalizace počtu výrobků sestavených z nařezaných dílů (maximalizace zisku)

## 3. (0-1) Úloha batohu

**Formulace:** Je dán rozpočet  $b$  pro investice do  $n$  uvažovaných projektů, kde  $a_j$  je hodnota nákladů na projekt  $j$  a  $c_j$  je jeho očekávaný výnos. Cílem je vybrat takový soubor projektů, který maximalizuje celkový očekávaný výnos a nepřekročí daný rozpočet.

**Proměnné:**

$$x_j = \begin{cases} 1 & \text{je-li projekt } j \text{ vybrán} \\ 0 & \text{jinak} \end{cases} \quad (5)$$

**Model:**

$$\max \sum_{j=1}^n c_j x_j \quad (6)$$

$$\sum_{j=1}^n a_j x_j \leq b \quad (7)$$

$$x_j \in \{0, 1\} \quad \text{pro } j = 1, 2, \dots, n \quad (8)$$

## 4. Úloha perfektního párování

**Formulace:** Na školním výletě má být  $n$  (sudé číslo) studentů rozřazeno do dvoulužkových pokojů. Pro potenciální spolubydlící  $i$  a  $j$  je dán index spokojenosti  $c_{ij}$ . Cílem je umístit studenty do pokojů tak, abychom maximalizovali celkovou spokojenost třídy.

**Proměnné:**

$$x_{ij} = \begin{cases} 1 & \text{jestliže studenti } i \text{ a } j \text{ jsou spolubydlící} \\ 0 & \text{jinak} \end{cases} \quad i < j \quad (9)$$

**Model:**

$$\max \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} x_{ij} \quad (10)$$

$$\sum_{j < i} x_{ji} + \sum_{j > i} x_{ij} = 1 \quad \text{pro } i = 1, 2, \dots, n \quad (11)$$

$$x_{ij} \in \{0, 1\} \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n-1 \\ j = i+1, i+2, \dots, n \end{matrix} \quad (12)$$

## 5. Zobecněný přiřazovací problém

**Formulace:** Uvažujme  $m$  čerpacích stanic odebírajících naftu z  $n$  terminálů. Každá stanice  $i$  může odebírat naftu pouze z jednoho terminálu, její požadavek je  $a_i$ . Kapacita terminálu  $j$  je  $b_j$ . Pokud stanice  $i$  odebírá naftu z terminálu  $j$ , jsou účtovány fixní náklady  $c_{ij}$ . Cílem je minimalizovat celkové náklady.

**Proměnné:**

$$x_{ij} = \begin{cases} 1 & \text{jestliže stanice } i \text{ odebírá naftu z terminálu } j \\ 0 & \text{jinak} \end{cases} \quad (13)$$

## 5. Zobecněný přiřazovací problém

Model:

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (14)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{pro } i = 1, 2, \dots, m \quad (15)$$

$$\sum_{i=1}^m a_i x_{ij} \leq b_j \quad \text{pro } j = 1, 2, \dots, n \quad (16)$$

$$x_{ij} \in \{0, 1\} \quad \text{pro } \begin{matrix} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{matrix} \quad (17)$$

## 6. Lineární přiřazovací problém

**Formulace:** K dispozici máme  $n$  pracovníků schopných provést  $n$  úkolů. Každý pracovník má vykonávat právě jeden úkol. Realizace  $j$ -tého úkolu  $i$ -tým pracovníkem je ohodnocena náklady  $c_{ij}$ . Cílem je rozdělit mezi pracovníky všechny úkoly s minimální výší celkových nákladů.

**Variables:**

$$x_{ij} = \begin{cases} 1 & \text{jestliže pracovník } i \text{ vykonává úkol } j \\ 0 & \text{jinak} \end{cases} \quad (18)$$

## 6. Lineární přiřazovací problém

Model:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (19)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{pro } i = 1, 2, \dots, n \quad (20)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{pro } j = 1, 2, \dots, n \quad (21)$$

$$x_{ij} \in \{0, 1\} \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n \\ j = 1, 2, \dots, n \end{matrix} \quad (22)$$

## 7. Úzkoprofilový přiřazovací problém

**Formulace:** Je dáno  $n$  úkolů a  $n$  paralelních strojů k jejich realizaci. Koeficient  $c_{ij}$  představuje dobu nutnou pro dokončení úkolu  $i$  strojem  $j$ . Cílem je minimalizovat dobu, za kterou budou úkoly dokončeny (všechny stroje začnou na úkolech pracovat současně).

**Proměnné:**

$$x_{ij} = \begin{cases} 1 & \text{jestliže úkol } i \text{ je přiřazen na stroj } j \\ 0 & \text{jinak} \end{cases} \quad (23)$$

$$T = \text{čas dokončení posledního úkolu} \quad (24)$$

## 7. Úzkoprofilový přiřazovací problém

Model:

$$\min T \quad (25)$$

$$c_{ij}x_{ij} \leq T \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n \\ j = 1, 2, \dots, n \end{matrix} \quad (26)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{pro } i = 1, 2, \dots, n \quad (27)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{pro } j = 1, 2, \dots, n \quad (28)$$

$$x_{ij} \in \{0, 1\} \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n \\ j = 1, 2, \dots, n \end{matrix} \quad (29)$$

## 8. Kvadratický přiřazovací problém

**Formulace:** Množina  $n$  zařízení má být umístěna na  $n$  pracovišť (na každém pracovišti bude právě jedno zařízení). Koeficient  $c_{ij}$  představuje hodnotu toku produktů od zařízení  $i$  k zařízení  $j$  a  $d_{kl}$  je vzdálenost mezi pracovišti  $k$  a  $l$ . Cílem je rozmístit zařízení na pracoviště s minimálními náklady.

**Variables:**

$$x_{ik} = \begin{cases} 1 & \text{je-li zařízení } i \text{ umístěno na pracoviště } k \\ 0 & \text{jinak} \end{cases} \quad (30)$$

## 8. Kvadratický přiřazovací problém

Model:

$$\min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n c_{ij} d_{kl} x_{ik} x_{jl} \quad (31)$$

$$\sum_{k=1}^n x_{ik} = 1 \quad \text{pro } i = 1, 2, \dots, n \quad (32)$$

$$\sum_{i=1}^n x_{ik} = 1 \quad \text{pro } k = 1, 2, \dots, n \quad (33)$$

$$x_{ik} \in \{0, 1\} \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n \\ k = 1, 2, \dots, n \end{matrix} \quad (34)$$

## 8. Kvadratický přiřazovací problém

Linearizace účelové funkce:

$$y_{ijkl} = \begin{cases} 1 & \text{jestliže zařízení } i \text{ je umístěno na pracoviště } k \\ & \text{a zařízení } j \text{ je umístěno na pracoviště } l \\ 0 & \text{jinak} \end{cases} \quad (35)$$

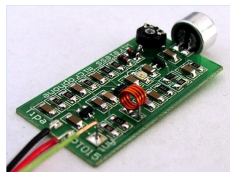
$$\min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n c_{ij} d_{kl} y_{ijkl} \quad (36)$$

$$y_{ijkl} \geq x_{ik} + x_{jl} - 1 \quad \text{pro } i, j, k, l = 1, 2, \dots, n \quad (37)$$

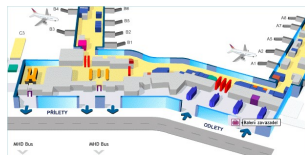
## 8. Kvadratický přiřazovací problém

### Aplikace:

- Letovací problém



- Přiřazení letadel k odletovým bránám



## 9. Pokrytí, dělení a rozklad množiny

**Formulace:** Necht'  $M = \{1, 2, \dots, m\}$  je konečná množina úkolů a  $N = \{1, 2, \dots, n\}$  konečná množina firem zajišťujících tyto úkoly. Je dána matice  $A$  s hodnotami  $a_{ij} = 1$ , jestliže firma  $j$  je schopna řešit úkol  $i$ ,  $a_{ij} = 0$  jinak. Jestliže  $j$ -tá firma bude vybrána, bude si účtovat náklady  $c_j$ . Cílem je pokrýt všechny úkoly s minimálními celkovými náklady.

Necht'  $M_j \subseteq M$  je množina úkolů, které je firma  $j \in N$  schopna řešit.

Říkáme, že

- $F \subseteq N$  je **pokrytím**  $M$ , jestliže  $\bigcup_{j \in F} M_j = M$
- $F \subseteq N$  je **dělením**  $M$ , jestliže  $M_j \cap M_k = \emptyset$  pro všechna  $j, k \in F, j \neq k$
- $F \subseteq N$  je **rozkladem**  $M$ , jestliže  $F$  je pokrytím a dělením  $M$

## 9. Pokrytí, dělení a rozklad množiny

Proměnné:

$$x_j = \begin{cases} 1 & \text{jestliže firma } j \text{ je vybrána, tj. } j \in F \\ 0 & \text{jinak} \end{cases} \quad (38)$$

Model:

$$\min \sum_{j=1}^n c_j x_j \quad (39)$$

$$\text{(pokrytí)} \quad \sum_{j=1}^n a_{ij} x_j \geq 1 \quad \text{pro } i = 1, 2, \dots, m$$

$$\text{(dělení)} \quad \sum_{j=1}^n a_{ij} x_j \leq 1 \quad \text{pro } i = 1, 2, \dots, m \quad (40)$$

$$\text{(rozklad)} \quad \sum_{j=1}^n a_{ij} x_j = 1 \quad \text{pro } i = 1, 2, \dots, m$$

$$x_j \in \{0, 1\} \quad \text{pro } j = 1, 2, \dots, n \quad (41)$$

## 9. Pokrytí, dělení a rozklad množiny

### Aplikace:

- Nechť  $N = \{1, 2, \dots, n\}$  je množina potenciálních lokalit pro umístění hasičských stanic. Náklady na umístění stanice v lokalitě  $j$  jsou  $c_j$ . Nechť  $M = \{1, 2, \dots, m\}$  je množina oblastí, které mají být zajištěny. Množina  $M_j$  obsahuje oblasti, které je možné zajistit z lokality  $j$  (např. jsou dosažitelné ze stanice v rámci 10 minut).
- Přiřazení posádek letům.
- Rozvržení pracovníků na směny.

## 10. Úloha optimálního rozmístění zařízení

**Formulace:** Je dána množina  $M = \{1, 2, \dots, m\}$  lokalit pro umístění zařízení a množina zákazníků  $N = \{1, 2, \dots, n\}$ . Nechť  $a_i$  je kapacita zařízení v lokalitě  $i$  a  $b_j$  je velikost poptávky  $j$ -tého zákazníka. Za využití lokality  $i$  se účtují fixní náklady  $f_i$ . Přeprava jedné jednotky z lokality  $i$  k zákazníkovi  $j$  stojí  $c_{ij}$ . Cílem je rozhodnout, kam umístit zařízení a jaké množství přepravit mezi jednotlivými lokalitami a zákazníky tak, aby celkové náklady byly minimální.

**Proměnné:**

$$x_i = \begin{cases} 1 & \text{jestliže zařízení bude umístěno v lokalitě } i \\ 0 & \text{jinak} \end{cases} \quad (42)$$

$$y_{ij} = \text{množství přepravené z lokality } i \text{ k zákazníkovi } j \quad (43)$$

## 10. Úloha optimálního rozmístění zařízení

Model:

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} y_{ij} + \sum_{i=1}^m f_i x_i \quad (44)$$

$$\sum_{j=1}^n y_{ij} \leq a_i x_i \quad \text{pro } i = 1, 2, \dots, m \quad (45)$$

$$\sum_{i=1}^m y_{ij} = b_j \quad \text{pro } j = 1, 2, \dots, n \quad (46)$$

$$x_i \in \{0, 1\} \quad \text{pro } i = 1, 2, \dots, m \quad (47)$$

$$y_{ij} \in \mathbb{R}_+ \quad \text{pro } \begin{matrix} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{matrix} \quad (48)$$

## 11. Úloha výrobního plánování s fixními náklady

**Formulace:** Lze vyrábět  $n$  produktů na  $n$  výrobních linkách (každý produkt na jedné VL). Pokud je VL  $i$  využita (tj. vyrábí se produkt  $i$ ), účtují se fixní náklady  $f_i$ . Jednotkový zisk z produktu  $i$  je  $c_i$ . Jsou definované standardní (kapacitní) omezující podmínky úlohy výrobního plánování. Cílem je maximalizovat celkový zisk z produkce snížený o fixní náklady.

**Proměnné:**

$$x_i = \begin{cases} 1 & \text{jestliže se produkt } i \text{ vyrábí (na VL } i) \\ 0 & \text{jinak} \end{cases} \quad (49)$$

$$y_i = \text{objem výroby produktu } i \quad (50)$$

## 11. Úloha výrobního plánování s fixními náklady

Model:

$$\max \sum_{i=1}^n c_i y_i - \sum_{i=1}^n f_i x_i \quad (51)$$

$$\text{(kapacitní omezení)} \quad \sum_{i=1}^n a_{li} y_i \leq b_l \quad \text{pro } l = 1, 2, \dots, m \quad (52)$$

$$y_i \leq M x_i \quad \text{pro } i = 1, 2, \dots, n \quad (53)$$

$$x_i \in \{0, 1\} \quad \text{pro } i = 1, 2, \dots, n \quad (54)$$

$$y_i \in \mathbb{R}_+ \quad \text{pro } i = 1, 2, \dots, n \quad (55)$$

$M$  = vysoká konstanta

## 12. Kontejnerový dopravní problém

**Formulace:** Homogenní zboží je přepravováno přímo od  $m$  dodavatelů k  $n$  odběratelům. Nechť  $a_i$  je kapacita dodavatele  $i$  a  $b_j$  je požadavek odběratele  $j$ . Pro přepravu jsou použity kontejnery o kapacitě  $K$ ; za přepravu jednoho kontejneru mezi dodavatelem  $i$  a odběratelem  $j$  se účtují náklady  $c_{ij}$ . Cílem je uspokojit požadavky všech odběratelů s minimálními přepravními náklady.

**Proměnné:**

$$y_{ij} = \begin{array}{l} \text{objem přepravy od dodavatele } i \\ \text{k odběrateli } j \end{array} \quad (56)$$

$$x_{ij} = \begin{array}{l} \text{počet kontejnerů použitých pro přepravu} \\ \text{zboží od dodavatele } i \text{ k odběrateli } j \end{array} \quad (57)$$

## 12. Kontejnerový dopravní problém

Model:

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (58)$$

$$\sum_{j=1}^n y_{ij} \leq a_i \quad \text{pro } i = 1, 2, \dots, m \quad (59)$$

$$\sum_{i=1}^m y_{ij} = b_j \quad \text{pro } j = 1, 2, \dots, n \quad (60)$$

$$y_{ij} \leq K x_{ij} \quad \text{pro } \begin{matrix} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{matrix} \quad (61)$$

$$y_{ij} \in \mathbb{R}_+ \quad \text{pro } \begin{matrix} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{matrix} \quad (62)$$

$$x_{ij} \in \mathbb{Z}_+ \quad \text{pro } \begin{matrix} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{matrix} \quad (63)$$

## 13. Rozšířená úloha batohu

**Formulace 1:** Je dána množina  $n$  předmětů, které mohou být umístěny do  $m$  kontejnerů. Je dána hmotnost předmětu  $j$ , kterou označíme  $w_j$  a jeho hodnota  $c_j$ . Kontejner  $i$  má nosnost  $K_i$ . Cílem je maximalizovat celkovou hodnotu všech vybraných položek.

**Proměnné:**

$$x_{ij} = \begin{cases} 1 & \text{jestliže předmět } j \text{ je umístěn do kontejneru } i \\ 0 & \text{jinak} \end{cases} \quad (64)$$

## 13. Rozšířená úloha batohu

Model:

$$\max \sum_{i=1}^m \sum_{j=1}^n c_j x_{ij} \quad (65)$$

$$\sum_{i=1}^m x_{ij} \leq 1 \quad \text{pro } j = 1, 2, \dots, n \quad (66)$$

$$\sum_{j=1}^n w_j x_{ij} \leq K_i \quad \text{pro } i = 1, 2, \dots, m \quad (67)$$

$$x_{ij} \in \{0, 1\} \quad \text{pro } \begin{matrix} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{matrix} \quad (68)$$

## 13. Rozšířená úloha batohu

**Formulace 2:** Je dána množina  $n$  typů předmětů, které mají být převezeny s využitím  $m$  kontejnerů s identickou nosností  $K$ . Pro typ předmětu  $j$  je dána jeho hmotnost  $w_j$  a počet  $r_j$ , který má být převezen. Cílem je minimalizovat počet použitých kontejnerů k přepravě všech předmětů.

**Proměnné:**

$$x_i = \begin{cases} 1 & \text{je-li kontejner } i \text{ použitý} \\ 0 & \text{jinak} \end{cases} \quad (69)$$

$$y_{ij} = \begin{array}{l} \text{počet položek typu } j \\ \text{umístěných v kontejneru } i \end{array} \quad (70)$$

## 13. Rozšířená úloha batohu

Model:

$$\min \sum_{i=1}^m x_i \quad (71)$$

$$\sum_{i=1}^m y_{ij} = r_j \quad \text{pro } j = 1, 2, \dots, n \quad (72)$$

$$\sum_{j=1}^n w_j y_{ij} \leq K x_i \quad \text{pro } i = 1, 2, \dots, m \quad (73)$$

$$x_i \in \{0, 1\} \quad \text{pro } i = 1, 2, \dots, m \quad (74)$$

$$y_{ij} \in \mathbb{Z}_+ \quad \text{pro } \begin{matrix} i = 1, 2, \dots, m \\ j = 1, 2, \dots, n \end{matrix} \quad (75)$$

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech**
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskrétních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

## Úvod to teorie grafů

**Graf** je množina  $G = \{V, E\}$ , kde  $V$  je množina uzlů (vrcholů) a  $E$  množina hran.

**Neorientovaná hrana** je množina dvou uzlů  $\{i, j\}$ .

**Orientovaná hrana** je uspořádaná dvojice uzlů  $(i, j)$ .

V **neorientovaném grafu** jsou všechny hrany neorientované.

V **orientovaném grafu (digrafu)** jsou všechny hrany orientované.

**Smíšený graf** obsahuje neorientované i orientované hrany.

Dva uzly spojené hranou se nazývají **sousední**.

Dvě hrany se společným uzlem se nazývají **sousední**.

Hrana a vrchol obsažený v této hraně se nazývají **incidentní**.

**Stupeň** uzlu (v neorientovaném grafu) je počet hran s ním incidentních.

**Vstupní polostupeň** uzlu (v orientovaném grafu) je počet incidentních hran, v nichž je tento uzel koncovým uzlem.

## Úvod to teorie grafů

**Výstupní polostupeň** uzlu (v orientovaném grafu) je počet incidentních hran, v nichž je tento uzel počátečním uzlem.

**Sled** z uzlu  $i$  do uzlu  $j$  je posloupnost uzlů a hran, která začíná v uzlu  $i$  a končí v uzlu  $j$  (uzly a hrany se mohou opakovat).

**Tah** je sled, v němž se neopakují žádné hrany.

**Cesta** je tah, v němž se neopakují žádné uzly.

**Cyklus** je uzavřený sled (začíná a končí ve stejném uzlu).

V **orientované cestě** (v orientovaném grafu) je respektována orientace všech hran.

V **neorientované cestě** (v orientovaném grafu) nemusí být orientace hran respektována.

Neorientovaný graf je **souvislý**, jestliže mezi každými dvěma uzly existuje cesta.

Orientovaný graf je **souvislý**, jestliže mezi každými dvěma uzly existuje orientovaná nebo neorientovaná cesta.

## Úvod to teorie grafů

Orientovaný graf je **silně souvislý**, jestliže mezi každými dvěma uzly existuje orientovaná cesta.

Neorientovaný graf je **úplný**, jestliže mezi každými dvěma uzly existuje hrana.

**Strom** je souvislý neorientovaný graf, v němž neexistuje cyklus.

**Podgraf** grafu  $G = \{V, E\}$  je graf  $G' = \{V', E'\}$ , kde  $V' \subseteq V$  a  $E' \subseteq E$ .

**Kostra** grafu  $G$  je podgraf  $G'$ , kde  $V' = V$ , a který je stromem.

V **ohodnoceném grafu** jsou uzlům a/nebo hranám přiřazena čísla.

**Hamiltonův cyklus** v grafu je cyklus, který obsahuje každý uzel grafu právě jednou.

**Eulerův cyklus** v grafu obsahuje každou jeho hranu právě jednou.

**Eulerův tah** v grafu je tah, který obsahuje každou jeho hranu.

**Eulerovský graf** je graf, v němž existuje Eulerův cyklus.

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 **Úlohy na grafech**
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskretních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

## 1. Úloha hledání maximálního toku

**Formulace:** Necht'  $G = \{V, E\}$  je digraf, v němž je pro každou hranu  $(i, j)$  definována její kapacita  $k_{ij}$ . Cílem je nalézt maximální hodnotu toku mezi zdrojem  $s$  a místem určení (stokem)  $d$ .

**Proměnné:**

$$x_{ij} = \text{hodnota toku z uzlu } i \text{ do uzlu } j \quad (76)$$

$$F = \text{hodnota celkového toku} \quad (77)$$

**Model:**

$$\max F \quad (78)$$

$$\sum_{\substack{j \in V \\ (i, j) \in E}} x_{ij} - \sum_{\substack{j \in V \\ (j, i) \in E}} x_{ji} = \begin{cases} F & \text{pro } i = s \\ 0 & \text{pro } i \in V \setminus \{s, d\} \\ -F & \text{pro } i = d \end{cases} \quad (79)$$

$$x_{ij} \leq k_{ij} \quad \text{pro } (i, j) \in E \quad (80)$$

$$x_{ij} \in \mathbb{R}_+ \quad \text{pro } (i, j) \in E \quad (81)$$

$$F \in \mathbb{R}_+ \quad (82)$$

## 1. Úloha hledání maximálního toku

**Alternativní model:** Přidáme fiktivní orientovanou hranu ze stoku  $d$  do zdroje  $s$ , která má kapacitu  $k_{ds} = M$  (vysoká konstanta).

**Proměnné:**

$$x_{ij} = \text{hodnota toku z uzlu } i \text{ do uzlu } j \quad (83)$$

**Model:**

$$\max x_{ds} \quad (84)$$

$$\sum_{\substack{j \in V \\ (i,j) \in E}} x_{ij} - \sum_{\substack{j \in V \\ (j,i) \in E}} x_{ji} = 0 \quad \text{pro } i \in V \quad (85)$$

$$x_{ij} \leq k_{ij} \quad \text{pro } (i,j) \in E \quad (86)$$

$$x_{ij} \in \mathbb{R}_+ \quad \text{pro } (i,j) \in E \quad (87)$$

## 2. Úloha hledání toku s minimálními náklady

**Formulace:** Necht'  $G = \{V, E\}$  je digraf, v němž pro každou hranu  $(i, j)$  je definována její kapacita  $k_{ij}$  a jednotkové náklady  $c_{ij}$ . Cílem je splnit požadovanou hodnotu celkového toku  $F_0$  (ze zdroje  $s$  do stoku  $d$ ) s minimálními celkovými náklady.

**Proměnné:**  $x_{ij}$  = hodnota toku z uzlu  $i$  do uzlu  $j$  (88)

**Model:** 
$$\min \sum_{i \in V} \sum_{\substack{j \in V \\ (i,j) \in E}} c_{ij} x_{ij} \quad (89)$$

$$\sum_{\substack{j \in V \\ (i,j) \in E}} x_{ij} - \sum_{\substack{j \in V \\ (j,i) \in E}} x_{ji} = \begin{cases} F_0 & \text{pro } i = s \\ 0 & \text{pro } i \in V \setminus \{s, d\} \\ -F_0 & \text{pro } i = d \end{cases} \quad (90)$$

$$x_{ij} \leq k_{ij} \quad \text{pro } (i, j) \in E \quad (91)$$

$$x_{ij} \in \mathbb{R}_+ \quad \text{pro } (i, j) \in E \quad (92)$$

## 3. Úloha hledání maximálního toku s limitovanými náklady

**Formulace:** Necht'  $G = \{V, E\}$  je digraf, v němž pro každou hranu  $(i, j)$  je definována její kapacita  $k_{ij}$  a jednotkové náklady  $c_{ij}$ . Cílem je nalézt maximální hodnotu toku mezi zdrojem  $s$  a místem určení (stokem)  $d$  při respektování omezených celkových nákladů  $C_0$ .

**Proměnné:**

$$x_{ij} = \text{hodnota toku z uzlu } i \text{ do uzlu } j \quad (93)$$

$$F = \text{hodnota celkového toku} \quad (94)$$

**Model:**

$$\max F \quad (95)$$

$$\sum_{\substack{i \in V \\ (i,j) \in E}} \sum_{j \in V} c_{ij} x_{ij} \leq C_0 \quad (96)$$

a omezující podmínky (79) - (82)

## 4. Úloha hledání toku s minimálními fixními náklady

**Formulace:** Necht'  $G = \{V, E\}$  je digraf, v němž je pro každou hranu  $(i, j)$  definována její kapacita  $k_{ij}$ . Za využití hrany  $(i, j)$  je účtována částka  $c_{ij}$ . Cílem je splnit požadovanou hodnotu celkového toku  $F_0$  (ze zdroje  $s$  do stoku  $d$ ) s minimálními celkovými fixními náklady.

**Proměnné:**

$$x_{ij} = \text{hodnota toku z uzlu } i \text{ do uzlu } j \quad (97)$$

$$y_{ij} = \begin{cases} 1 & \text{je-li hrana } (i, j) \text{ využita} \\ 0 & \text{jinak} \end{cases} \quad (98)$$

## 4. Úloha hledání toku s minimálními fixními náklady

Model:

$$\min \sum_{i \in V} \sum_{\substack{j \in V \\ (i,j) \in E}} c_{ij} y_{ij} \quad (99)$$

$$\sum_{\substack{j \in V \\ (i,j) \in E}} x_{ij} - \sum_{\substack{j \in V \\ (j,i) \in E}} x_{ji} = \begin{cases} F_0 & \text{pro } i = s \\ 0 & \text{pro } i \in V \setminus \{s, d\} \\ -F_0 & \text{pro } i = d \end{cases} \quad (100)$$

$$x_{ij} \leq k_{ij} y_{ij} \quad \text{pro } (i, j) \in E \quad (101)$$

$$y_{ij} \in \{0, 1\} \quad \text{pro } (i, j) \in E \quad (102)$$

## 5. Přepravní úloha (Transshipment Problem)

**Formulace:** Nechť  $G = \{V, E\}$  je digraf se třemi množinami uzlů: množina zdrojů  $V_s$ , množina cílových uzlů  $V_d$  a množina průběžných uzlů  $V_t$ . Nechť  $a_i > 0$  je velikost nabídky homogenního produktu ve zdroji  $i \in V_s$  a  $a_i < 0$  je velikost poptávky po produktu v cílovém uzlu  $i \in V_d$ . Pro průběžný uzel  $i \in V_t$  platí  $a_i = 0$ . Pro každou hranu  $(i, j)$  je definována její kapacita  $k_{ij}$  a jednotkové náklady  $c_{ij}$ . Poptávka ve všech cílových uzlech musí být uspokojena, aniž by došlo v některém zdroji k překročení jeho nabídky. Cílem je minimalizovat celkové náklady. Předpokládejme, že velikost celkové poptávky je rovna velikosti celkové nabídky.

**Předpoklady:**

$$V = V_s \cup V_d \cup V_t \quad \text{a} \quad V_s \cap V_d \cap V_t = \emptyset \quad (103)$$

$$\sum_{i \in V_s} a_i + \sum_{i \in V_d} a_i = 0 \quad (104)$$

## 5. Přepravní úloha (Transshipment Problem)

Proměnné:

$$x_{ij} = \text{hodnota toku z uzlu } i \text{ do uzlu } j \quad (105)$$

Model:

$$\min \sum_{i \in V} \sum_{\substack{j \in V \\ (i,j) \in E}} c_{ij} x_{ij} \quad (106)$$

$$\sum_{\substack{j \in V \\ (i,j) \in E}} x_{ij} - \sum_{\substack{j \in V \\ (j,i) \in E}} x_{ji} = a_i \quad \text{pro } i \in V \quad (107)$$

$$x_{ij} \leq k_{ij} \quad \text{pro } (i,j) \in E \quad (108)$$

$$x_{ij} \in \mathbb{R}_+ \quad \text{pro } (i,j) \in E \quad (109)$$

## 6. Minimální kostra grafu

**Formulace:** Necht'  $G = \{V, E\}$  je neorientovaný graf, v němž jsou pro každou hranu  $\{i, j\}$  definovány náklady  $c_{ij}$ . Cílem je najít kostru grafu  $G$  s minimálními celkovými náklady odpovídajícími součtu nákladů vybraných hran.

**Úprava grafu:** Množina neorientovaných hran  $E$  je převedena na množinu orientovaných hran  $A$  takto:

Každá hrana  $\{i, j\} \in E$  je nahrazena dvojicí orientovaných hran  $(i, j) \in A$  a  $(j, i) \in A$ ,  $c_{ji} = c_{ij}$ .

**Proměnné:**

$$x_{ij} = \begin{cases} 1 & \text{je-li hrana } (i, j) \text{ vybrána} \\ 0 & \text{jinak} \end{cases} \quad (110)$$

$$y_{ij} = \text{hodnota toku z uzlu } i \text{ do uzlu } j \quad (111)$$

## 6. Minimální kostra grafu

Model:

$$\min \sum_{\substack{i \in V \\ (i,j) \in A}} \sum_{\substack{j \in V \\ (i,j) \in A}} c_{ij} x_{ij} \quad (112)$$

$$\sum_{\substack{j \in V \\ (1,j) \in A}} x_{1j} = 0 \quad (113)$$

$$\sum_{\substack{j \in V \\ (i,j) \in A}} x_{ij} = 1 \quad \text{pro } i \in V \setminus \{1\} \quad (114)$$

$$\sum_{\substack{j \in V \\ (i,j) \in A}} y_{ij} - \sum_{\substack{j \in V \\ (j,i) \in A}} y_{ji} = 1 \quad \text{pro } i \in V \setminus \{1\} \quad (115)$$

$$0 \leq y_{ij} \leq (|V| - 1)x_{ij} \quad \text{pro } (i,j) \in A \quad (116)$$

$$x_{ij} \in \{0, 1\} \quad \text{pro } (i,j) \in A \quad (117)$$

## 7. Minimální Steinerův strom

**Formulace:** Nechť  $G = \{V, E\}$  je digraf,  $s \in V$  je zdroj signálu (vysílač),  $D \subset V$  je množina uživatelů (přijímačů, cílových míst) a  $T \subset V$  je množina pomocných stanic (ústředny), přes které může být signál přenášen. Uživatelé mohou být připojeni na vysílač buď přímo nebo přes ústředny. Hodnota  $c_{ij}$  představuje náklady na spojení  $(i, j) \in E$ . Za použití ústředny  $i \in T$  je účtována fixní částka  $f_i$ . Cílem je spojit všechny uživatele s vysílačem s minimálními celkovými náklady.

**Proměnné:**

$$z_i = \begin{cases} 1 & \text{je-li uzel } i \text{ zařazen do stromu} \\ 0 & \text{jinak} \end{cases} \quad (118)$$

$$x_{ij} = \begin{cases} 1 & \text{je-li hrana } (i, j) \text{ zařazena do stromu} \\ 0 & \text{jinak} \end{cases} \quad (119)$$

$$y_{ij} = \text{hodnota toku z uzlu } i \text{ do uzlu } j \quad (120)$$

## 7. Minimální Steinerův strom

Model:

$$\min \sum_{\substack{i \in V \\ (i,j) \in E}} \sum_{j \in V} c_{ij} x_{ij} + \sum_{i \in T} f_i z_i \quad (121)$$

$$z_i = 1 \quad \text{pro } i \in D \quad (122)$$

$$\sum_{\substack{j \in V \\ (i,j) \in E}} x_{ij} = z_i \quad \text{pro } i \in V \setminus \{s\} \quad (123)$$

$$\sum_{\substack{j \in V \\ (i,j) \in E}} y_{ij} - \sum_{\substack{j \in V \\ (j,i) \in E}} y_{ji} = z_i \quad \text{pro } i \in V \setminus \{s\} \quad (124)$$

$$0 \leq y_{ij} \leq (|V| - 1)x_{ij} \quad \text{pro } (i,j) \in E \quad (125)$$

$$x_{ij} \in \{0, 1\} \quad \text{pro } (i,j) \in E \quad (126)$$

$$z_i \in \{0, 1\} \quad \text{pro } i \in T \quad (127)$$

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech**
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskrétních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

## Klasifikace úloh

- **Typ trasy v grafu**
  - ▶ Trasy jsou tvořeny uzly
    - ★ Úloha obchodního cestujícího (TSP) - nekonečně velká kapacita vozidel
    - ★ Rozvozní úloha (VRP) - omezená kapacita vozidel
  - ▶ Trasy jsou tvořeny hranami
    - ★ Úloha čínského listonoše (CPP)
- **Počet dep a vozidel**
  - ▶ Jedno depo (výchozí místo) s jedním nebo několika vozidly
  - ▶ Více dep
- **Znalost zákazníků**
  - ▶ Statické úlohy - všichni zákazníci jsou známí předem
  - ▶ Dynamické úlohy - zákazníci známí předem a zákazníci s on-line požadavky
- **Cíl**
  - ▶ Minimalizace celkové délky tras (celkových nákladů)
  - ▶ Minimalizace celkové doby jízdy všech vozidel
  - ▶ Minimalizace nejdelší doby jízdy ze všech vozidel

## 1. Nesymetrická úloha obchodního cestujícího

**Formulace:** Nechť  $G = \{U, E\}$  je úplný digraf se vzdáleností  $c_{ij}$  definovanou pro každou hranu  $(i, j)$  (matice  $C$  je obecně nesymetrická). Nechť uzel 1 je depo a  $|U| = n$ . Cílem je určit minimální Hamiltonův cyklus.

**Proměnné:**

$$x_{ij} = \begin{cases} 1 & \text{jestliže vozidlo jede přímo} \\ & \text{z uzlu } i \text{ do uzlu } j \\ 0 & \text{jinak} \end{cases} \quad (128)$$

$$u_i = \text{umělá proměnná zavedená v podmínkách} \\ \text{zabraňujících vytváření parciálních cyklů} \quad (129)$$

## 1. Nesymetrická úloha obchodního cestujícího

**Model** (Miller, Tucker, Zemlin):

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (130)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{pro } i = 1, 2, \dots, n \quad (131)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{pro } j = 1, 2, \dots, n \quad (132)$$

$$u_i + 1 - (n - 1)(1 - x_{ij}) \leq u_j \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n \\ j = 2, 3, \dots, n \end{matrix} \quad (133)$$

$$x_{ij} \in \{0, 1\} \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n \\ j = 1, 2, \dots, n \end{matrix} \quad (134)$$

$$u_i \in \mathbb{R}_+ \quad \text{pro } i = 1, 2, \dots, n \quad (135)$$

## 2. Úloha obchodního cestujícího s časovými okny (TSPTW)

**Formulace:** Nechť je definována nesymetrická úloha TSP. Každý uzel  $i$  musí být navštíven v rámci časového intervalu  $\langle e_i, l_i \rangle$ . Vozidlo stráví v uzlu  $i$  daný čas  $S_i$ . Hodnota  $d_{ij}$  představuje dobu přejezdu mezi uzly  $i$  a  $j$ . Cílem je určit minimální Hamiltonův cyklus (z hlediska vzdálenosti) při respektování všech časových oken.

**Proměnné:**

$$x_{ij} = \begin{cases} 1 & \text{jestliže vozidlo jede přímo} \\ & \text{z uzlu } i \text{ do uzlu } j \\ 0 & \text{jinak} \end{cases} \quad (136)$$

$$t_i = \text{čas příjezdu vozidla do uzlu } i \quad (137)$$

## 2. Úloha obchodního cestujícího s časovými okny (TSPTW)

Úprava modelu MTZ:

$$e_i \leq t_i \leq l_i \quad \text{pro } i = 2, 3, \dots, n \quad (138)$$

$$t_1 = 0 \quad (139)$$

$$t_i \in \mathbb{R}_+ \quad \text{pro } i = 2, 3, \dots, n \quad (140)$$

Proměnné  $u_i$  jsou eliminovány a omezující podmínky (133) jsou nahrazeny podmínkami

$$t_i + S_i + d_{ij} - M(1 - x_{ij}) \leq t_j \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n \\ j = 2, 3, \dots, n \end{matrix} \quad i \neq j \quad (141)$$

## 3. Metrická úloha obchodního cestujícího ( $\Delta$ - TSP)

Trojúhelníkové nerovnosti:

$$c_{ij} \leq c_{ik} + c_{kj} \quad \text{pro } i, j, k = 1, 2, \dots, n, i \neq j \neq k \quad (142)$$

## 4. Euklidovská úloha obchodního cestujícího

Euklidovské vzdálenosti:

$$c_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad \text{pro } i, j = 1, 2, \dots, n, i \neq j \quad (143)$$

## 5. Otevřená úloha obchodního cestujícího

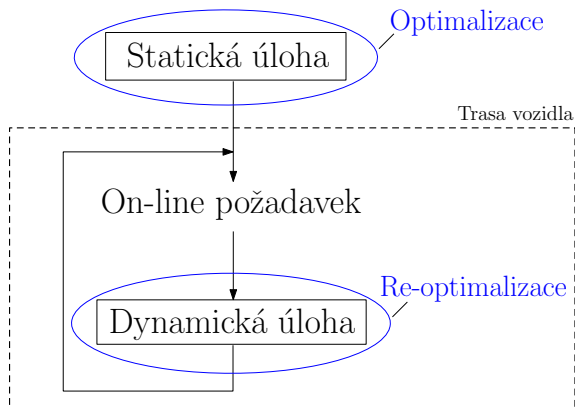
Namísto minimálního Hamiltonova cyklu se hledá minimální otevřená cesta vedoucí přes všechny uzly (trasa není zakončena v depu):

$$\text{nastavíme } c_{i1} = 0 \quad \text{pro } i = 2, 3, \dots, n \quad (144)$$

## 6. Dynamická úloha obchodního cestujícího

Ve statické verzi TSP jsou všichni zákazníci známi předem.

V dynamické verzi se během realizace optimální trasy objevují on-line požadavky dalších zákazníků.



## 7. Rozvozní úloha

**Formulace:** Necht'  $G = \{U, E\}$  je úplný digraf se vzdáleností  $c_{ij}$  definovanou pro každou hranu  $(i, j)$ . Necht' uzel 1 je depo, v němž je k dispozici vozidlo s kapacitou  $V$ ,  $|U| = n$ . Každý zákazník  $i$  má požadavek o velikosti  $q_i$ . Cílem je uspokojit požadavky všech zákazníků a minimalizovat celkovou délku všech tras.

**Proměnné:**

$$x_{ij} = \begin{cases} 1 & \text{jestliže vozidlo jede přímo} \\ & \text{u uzlu } i \text{ do uzlu } j \\ 0 & \text{jinak} \end{cases} \quad (145)$$

$$u_i = \text{umělá proměnná zavedená pro bilanci nákladu vozidla} \quad (146)$$

**Předpoklady:**

$$\sum_{i=2}^n q_i > V \quad (147)$$

$$q_i \leq V \quad \text{pro } i = 2, 3, \dots, n \quad (148)$$

## 7. Rozvozní úloha

Model:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (149)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{pro } i = 2, 3, \dots, n \quad (150)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{pro } j = 2, 3, \dots, n \quad (151)$$

$$u_i + q_j - V(1 - x_{ij}) \leq u_j \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n \\ j = 2, 3, \dots, n \end{matrix} \quad (152)$$

$$u_i \leq V \quad \text{pro } i = 2, 3, \dots, n \quad (153)$$

$$u_1 = 0 \quad (154)$$

$$x_{ij} \in \{0, 1\} \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n \\ j = 1, 2, \dots, n \end{matrix} \quad (155)$$

$$u_i \in \mathbb{R}_+ \quad \text{pro } i = 2, 3, \dots, n \quad (156)$$

## 8. Rozvozní úloha s heterogenním vozovým parkem

**Formulace:** Nechť je definována úloha VRP s  $K$  typy vozidel, která jsou k dispozici v jednom depu. Pro každý typ  $k$  je dána kapacita vozidla  $V_k$ , počet vozidel  $p_k$  a nákladový koeficient  $d_k$ , odvozený od spotřeby vozidla. Cílem je uspokojit požadavky všech zákazníků a minimalizovat celkové náklady.

**Proměnné:**

$$x_{ij}^k = \begin{cases} 1 & \text{jestliže vozidlo typu } k \text{ jede přímo} \\ & \text{z uzlu } i \text{ do uzlu } j \\ 0 & \text{jinak} \end{cases} \quad (157)$$

$$u_i = \text{umělá proměnná zavedená pro bilanci nákladu vozidla} \quad (158)$$

**Předpoklad:**

$$\sum_{i=2}^n q_i \leq \sum_{k=1}^K p_k V_k \quad (159)$$

**Značení:**

$$\bar{V} = \max_{k=1,2,\dots,K} V_k \quad (160)$$

## 8. Rozvozní úloha s heterogenním vozovým parkem

Model:

$$\min \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^n d_k c_{ij} x_{ij}^k \quad (161)$$

$$\sum_{k=1}^K \sum_{j=1}^n x_{ij}^k = 1 \quad \text{pro } i = 2, 3, \dots, n \quad (162)$$

$$\sum_{i=1}^n x_{ij}^k = \sum_{i=1}^n x_{ji}^k \quad \text{pro } \begin{matrix} j = 1, 2, \dots, n \\ k = 1, 2, \dots, K \end{matrix} \quad (163)$$

$$\sum_{j=2}^n x_{1j}^k \leq p_k \quad \text{pro } k = 1, 2, \dots, K \quad (164)$$

$$u_i + q_j - \bar{V}(1 - x_{ij}^k) \leq u_j \quad \text{pro } \begin{matrix} i = 1, 2, \dots, n \\ j = 2, 3, \dots, n \\ k = 1, 2, \dots, K \end{matrix} \quad (165)$$

## 8. Rozvozní úloha s heterogenním vozovým parkem

Model (pokrač.):

$$u_i \leq \sum_{j=1}^n \sum_{k=1}^K x_{ij}^k V_k \quad \text{pro } i = 1, 2, \dots, n \quad (166)$$

$$u_1 = 0 \quad (167)$$

$$x_{ij}^k \in \{0, 1\} \quad \text{pro } \begin{array}{l} i = 1, 2, \dots, n \\ j = 1, 2, \dots, n \\ k = 1, 2, \dots, K \end{array} \quad (168)$$

$$u_i \in \mathbb{R}_+ \quad \text{pro } i = 2, 3, \dots, n \quad (169)$$

## 9. Rozvozní úloha s časovými okny (VRPTW)

Jestliže jsou v úloze definována časová okna, je zavedena proměnná  $t_i$  a všechny odpovídající omezující podmínky podobně jako v úloze TSPTW.

## 10. Rozvozní úloha s dělenou dodávkou (SDVRP)

Standardní model úlohy VRP nelze použít, pokud  $\exists i, q_i > V$ .

Takový požadavek musí být rozdělen do více tras. I v případě, že  $q_i \leq V, \forall i$ , může být výhodné dělit dodávky. V modelu proměnná  $Q_i^k$  značí část požadavku  $q_i$  zahrnutou do trasy  $k$ :

$$\sum_{k=1}^K Q_i^k = q_i \quad \text{pro } i = 2, 3, \dots, n \quad (170)$$

## 11. Převravní problémy - úlohy vyzvednutí a doručení (PDP)

- One-to-One PDP

(úloha přepravy hendikepovaných osob, úloha kurýrní služby)  
Každý požadavek je dán místem vyzvednutí a místem doručení. Trasy vozidel začínají a končí ve společném depu.

- Many-to-Many PDP

Zboží může být vyzvednuto v jednom z několika míst a doručeno do jednoho z několika míst.

- One-to-Many-to-One PDP

Každý zákazník obdrží zásilku z depa a zašle do něj jinou zásilku.

## 12. Neorientovaná úloha čínského listonoše

**Formulace:** Necht'  $G = \{U, E\}$  je neorientovaný souvislý graf. Pro každou hranu  $\{i, j\}$  jsou dány náklady  $c_{ij}$ . Cílem je najít cyklus s minimálními celkovými náklady takový, že v něm bude obsažena každá hrana alespoň jednou.

**Věta:** Neorientovaný graf  $G$  je eulerovský právě tehdy, když  $G$  je souvislý a všechny uzly mají sudý stupeň.

Pokud graf  $G$  není eulerovský, sestrojíme supergraf  $G^*$  grafu  $G$  takový, že  $G^*$  je eulerovský a obsahuje Eulerův cyklus, který je kratší než Eulerův cyklus v jakémkoli jiném supergrafu grafu  $G$ .

## 12. Neorientovaná úloha čínského listonoše

Proměnné v modelu I:

$$x_{ij} = \begin{cases} 1 & \text{jestliže hrana } \{i, j\} \text{ je zdvojena v } G^* \\ 0 & \text{jinak} \end{cases} \quad i < j \quad (171)$$

$$y_i = \text{umělá proměnná pro vyjádření} \\ \text{sudého/lického čísla} \quad (172)$$

Značení v modelu I:

$U_0 \subset U$  je množina uzlů sudého stupně

$U_1 \subset U$  je množina uzlů lichého stupně

## 12. Neorientovaná úloha čínského listonoše

Model I:

$$\min \sum_{\substack{\{i,j\} \in E \\ i < j}} c_{ij} x_{ij} \quad (173)$$

$$\sum_{\substack{\{j,i\} \in E \\ j < i}} x_{ji} + \sum_{\substack{\{i,j\} \in E \\ j > i}} x_{ij} = 2y_i \quad \text{pro } i \in U_0 \quad (174)$$

$$\sum_{\substack{\{j,i\} \in E \\ j < i}} x_{ji} + \sum_{\substack{\{i,j\} \in E \\ j > i}} x_{ij} = 2y_i + 1 \quad \text{pro } i \in U_1 \quad (175)$$

$$x_{ij} \in \{0, 1\} \quad \text{pro } \{i, j\} \in E, i < j \quad (176)$$

$$y_i \in \mathbb{Z}_+ \quad \text{pro } i \in U \quad (177)$$

## 12. Neorientovaná úloha čínského listonoše

Proměnné v modelu II:

$$x_{ij} = \text{počet hran } \{i, j\} \text{ v grafu } G^* \quad (178)$$

Model II:

$$\min \sum_{\{i,j\} \in E} c_{ij} x_{ij} \quad (179)$$

$$x_{ij} + x_{ji} \geq 1 \quad \text{pro } \{i, j\}, \{j, i\} \in E \quad (180)$$

$$\sum_{\{j,i\} \in E} x_{ji} = \sum_{\{i,j\} \in E} x_{ij} \quad \text{pro } i \in U \quad (181)$$

$$x_{ij} \in \mathbb{Z}_+ \quad \text{pro } \{i, j\} \in E \quad (182)$$

## 13. Orientovaná úloha čínského listonoše

**Formulace:** Necht'  $G = \{U, E\}$  je silně souvislý digraf. Pro každou hranu  $(i, j)$  jsou dány náklady  $c_{ij}$ . Cílem je najít cyklus s minimálními celkovými náklady takový, že v něm bude obsažena každá hrana alespoň jednou.

**Věta:** Orientovaný graf  $G$  je eulerovský právě tehdy, když  $G$  je silně souvislý a vstupní polostupeň každého uzlu je roven jeho výstupnímu polostupni.

Pokud  $G$  není eulerovský, sestrojíme supergraf  $G^*$  grafu  $G$  takový, že  $G^*$  je eulerovský a obsahuje Eulerův cyklus, který je kratší než Eulerův cyklus v jakémkoli jiném supergrafu grafu  $G$ .

## 13. Orientovaná úloha čínského listonoše

### Značení:

$\text{deg}_i^-$  = vstupní polostupeň uzlu  $i$

$\text{deg}_i^+$  = výstupní polostupeň uzlu  $i$

$I$  = množina uzlů  $i$  pro které  $\text{deg}_i^- > \text{deg}_i^+$

$J$  = množina uzlů  $j$  pro které  $\text{deg}_j^- < \text{deg}_j^+$

$a_i = \text{deg}_i^- - \text{deg}_i^+$  pro  $i \in I$

$b_j = \text{deg}_j^+ - \text{deg}_j^-$  pro  $j \in J$

$d_{ij}$  = délka nejkratší orientované cesty z  $i \in I$  do  $j \in J$

### Proměnné:

$x_{ij}$  = počet nejkratších orientovaných cest mezi uzly  $i$  a  $j$ ,  
které přidáme do grafu  $G^*$

(183)

## 13. Orientovaná úloha čínského listonoše

Model:

$$\min \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \quad (184)$$

$$\sum_{j \in J} x_{ij} = a_i \quad \text{pro } i \in I \quad (185)$$

$$\sum_{i \in I} x_{ij} = b_j \quad \text{pro } j \in J \quad (186)$$

$$x_{ij} \in \mathbb{R}_+ \quad \text{pro } \begin{matrix} i \in I \\ j \in J \end{matrix} \quad (187)$$

## 14. Smíšená úloha čínského listonoše (čištění ulic)

CPP na smíšeném grafu  $G = \{U, E\}$ .

## 15. „Rural“ verze úlohy čínského listonoše (doručování pošty)

Nechť  $G = \{U, E\}$  je souvislý s množinou  $R \subset E$  povinných hran, kterými je nutné projet alespoň jednou. Zbývající hrany v množině  $E \setminus R$  jsou volitelné a mohou být využity v optimální trase.

## 16. Kapacitní úloha čínského listonoše (svoz odpadu)

Nechť  $G = \{U, E\}$  je souvislý graf s požadavky o velikosti  $q_{ij}$  pro každou hranu  $\{i, j\}$  (pro každou povinnou hranu ve verzi „Rural CPP“). Pro uspokojení požadavků je k dispozici vozidlo s omezenou kapacitou. Je nutné najít cykly, aniž by na některém z nich došlo k překročení kapacity vozidla.

## 17. Hierarchická úloha čínského listonoše (odklizení sněhu pluhem)

U každé hrany je dána její priorita z hlediska důležitosti příslušné komunikace.

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými**
- 5 Vlastnosti diskrétních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

## 1. Nekonvexní množina přípustných řešení

**Příklad:** Jsou dány následující omezující podmínky. Použijte diskrétní proměnné tak, aby bylo možné úlohu s podmínkami řešit jako model MIP.

$$\begin{aligned}y_1 + y_2 &\leq 40 \\y_1 &\geq 20 \text{ nebo } y_2 \geq 10 \\y_1, y_2 &\in \mathbb{R}_+\end{aligned}$$

**Proměnné:**

$$x_1 = \begin{cases} 1 & \text{jestliže } y_1 \geq 20 \\ 0 & \text{jinak} \end{cases} \quad (188)$$

$$x_2 = \begin{cases} 1 & \text{jestliže } y_2 \geq 10 \\ 0 & \text{jinak} \end{cases} \quad (189)$$

## 1. Nekonvexní množina přípustných řešení

Model:

$$y_1 + y_2 \leq 40 \quad (190)$$

$$y_1 \geq 20x_1 \quad (191)$$

$$y_2 \geq 10x_2 \quad (192)$$

$$x_1 + x_2 \geq 1 \quad (193)$$

$$y_1, y_2 \in \mathbb{R}_+ \quad (194)$$

$$x_1, x_2 \in \{0, 1\} \quad (195)$$

## 2. Platnost různých soustav omezení (typ buď-nebo)

**Příklad:** Tři různé produkty mohou být na stroji vyráběny buď v pořadí  $P_1 \rightarrow P_2 \rightarrow P_3$  nebo  $P_3 \rightarrow P_2 \rightarrow P_1$ . Předpokládejme, že výroba produktu  $P_i$  trvá  $t_i$ . Formulujte omezující podmínky modelující přípustnou produkci.

**Proměnné:**

$$y_i = \text{okamžik začátku výroby produktu } P_i \quad (196)$$

$$x = \begin{cases} 1 & \text{vyrábí-li se produkty v pořadí } P_1 \rightarrow P_2 \rightarrow P_3 \\ 0 & \text{vyrábí-li se produkty v pořadí } P_3 \rightarrow P_2 \rightarrow P_1 \end{cases} \quad (197)$$

## 2. Platnost různých soustav omezení (typ buď-nebo)

Model:

$$y_1 + t_1 \leq y_2 + M(1 - x) \quad (198)$$

$$y_2 + t_2 \leq y_3 + M(1 - x) \quad (199)$$

$$y_3 + t_3 \leq y_2 + Mx \quad (200)$$

$$y_2 + t_2 \leq y_1 + Mx \quad (201)$$

$$y_i \in \mathbb{R}_+ \quad \text{pro } i = 1, 2, 3 \quad (202)$$

$$x \in \{0, 1\} \quad (203)$$

## 3. Platnost různých soustav omezení (splnění $k$ z $m$ podmínek)

**Příklad:** Model obsahuje soustavu  $m$  omezujících podmínek. Nechť  $i$ -tá podmínka je definována jako  $a_i^T y \leq b_i$ . Zajistěte, aby bylo splněno přesně  $k$  ze všech  $m$  podmínek ( $k < m$ ).

**Umělé proměnné:**

$$x_i = \begin{cases} 1 & \text{je-li } i\text{-tá omezující podmínka zahrnuta v modelu} \\ 0 & \text{jinak} \end{cases} \quad (204)$$

**Omezující podmínky:**

$$a_i^T y \leq b_i + M(1 - x_i) \quad \text{pro } i = 1, 2, \dots, m \quad (205)$$

$$\sum_{i=1}^m x_i = k \quad (206)$$

## 4. Speciální omezení pro úroveň výroby

**Příklad:** Firma zvažuje výrobu nového produktu. V případě výroby, úroveň musí být alespoň 500 ks a nesmí překročit 1000 ks.

Naformulujte odpovídající omezující podmínky.

**Proměnné:**

$$y = \text{úroveň výroby} \quad (207)$$

$$x = \begin{cases} 1 & \text{jestliže se firma rozhodne produkt vyrábět} \\ 0 & \text{jinak} \end{cases} \quad (208)$$

**Model:**

$$500x \leq y \leq 1000x \quad (209)$$

$$y \in \mathbb{Z}_+ \quad (210)$$

$$x \in \{0, 1\} \quad (211)$$

## 5. Plánování diskretní úrovně výroby

**Příklad:** Firma zvažuje, zda vyrobit 500, 1000 nebo 2000 ks daného produktu. Naformulujte odpovídající omezující podmínky.

**Proměnné:**

$$y = \text{úroveň výroby} \quad (212)$$

$$x_i = \begin{cases} 1 & \text{je-li výroba na } i\text{-té úrovni} \\ 0 & \text{jinak} \end{cases} \quad i = 1, 2, 3 \quad (213)$$

**Model:**

$$y = 500x_1 + 1000x_2 + 2000x_3 \quad (214)$$

$$x_1 + x_2 + x_3 = 1 \quad (215)$$

$$y \in \mathbb{Z}_+ \quad (216)$$

$$x_i \in \{0, 1\} \quad \text{pro } i = 1, 2, 3 \quad (217)$$

## 6. Výroba určitého počtu druhů produktů

**Příklad:** Firma může teoreticky vyrábět  $n$  druhů produktů.

Rozhodla se z nich ale vyrábět jen  $k$  druhů. Pro každý produkt  $i$  je dána maximální úroveň výroby  $q_i$ . Naformulujte odpovídající omezující podmínky.

**Proměnné:**

$$y_i = \text{úroveň výroby produktu } i \quad (218)$$

$$x_i = \begin{cases} 1 & \text{jestliže se produkt } i \text{ vyrábí} \\ 0 & \text{jinak} \end{cases} \quad (219)$$

**Omezující podmínky:**

$$\frac{1}{M} x_i \leq y_i \leq q_i x_i \quad \text{pro } i = 1, 2, \dots, n \quad (220)$$

$$\sum_{i=1}^n x_i = k \quad (221)$$

## 6. Výroba určitého počtu druhů produktů

**Příklad** (rovnající se množství vyráběných druhů produktů):

V předchozím příkladu je navíc určeno, že se úrovně výroby u vyráběných druhů produktů musí rovnat (podmínka komplementarity).

**Další proměnné:**

$$w = \text{úroveň výroby vyráběných druhů produktů} \quad (222)$$

**Další omezující podmínky:**

$$w - M(1 - x_i) \leq y_i \leq w + M(1 - x_i) \quad \text{pro } i = 1, 2, \dots, n \quad (223)$$

## 7. Definice proměnné rovnající se minimu dalších proměnných

**Příklad:** Zapište proměnnou jako minimum  $n$  proměnných.

**Proměnné:**

$$w = \min(y_1, y_2, \dots, y_n) \quad (224)$$

$$x_i = \begin{cases} 1 & \text{jestliže } w = y_i \\ 0 & \text{jinak} \end{cases} \quad (225)$$

**Omezující podmínky:**

$$w \leq y_i \leq w + M(1 - x_i) \quad \text{pro } i = 1, 2, \dots, n \quad (226)$$

$$\sum_{i=1}^n x_i \geq 1 \quad (227)$$

## 8. Linearizace součinu binárních proměnných

**Příklad:** Linearizujte maximalizační účelovou funkci  $x_1x_2x_3$  (všechny proměnné jsou binární).

**Umělá proměnná:**

$$w = x_1x_2x_3 \quad (228)$$

**Model:**

$$\max w \quad (229)$$

$$x_1 + x_2 + x_3 - 2 \leq w \quad (230)$$

$$w \leq x_i \quad \text{pro } i = 1, 2, 3 \quad (231)$$

$$x_i \in \{0, 1\} \quad \text{pro } i = 1, 2, 3 \quad (232)$$

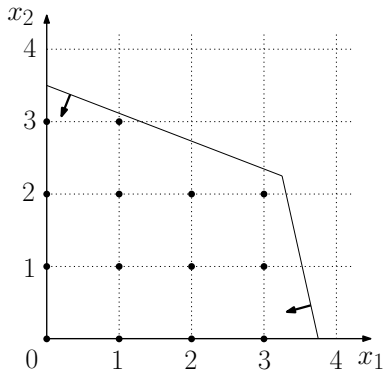
$$w \in \mathbb{R}_+ \quad (233)$$

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskrétních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

Množina přípustných bodů v úlohách lineárního a celočíselného programování

$$(LP) \quad P = \{x \in \mathbb{R}_+^n : Ax \leq b\} \quad (234)$$

$$(IP) \quad S = \{x \in \mathbb{Z}_+^n : Ax \leq b\} \quad (235)$$

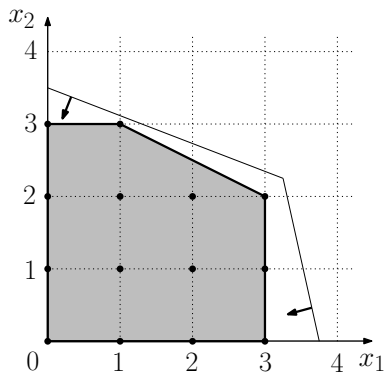


**Definice:** *Polyedr*  $P \subseteq \mathbb{R}^n$  je množina bodů, které splňují konečný počet lineárních nerovností:  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ . Polyedr je *omezený*, jestliže existuje hodnota  $\delta \in \mathbb{R}_+$  taková, že  $P \subseteq \{x \in \mathbb{R}^n : -\delta \leq x_j \leq \delta \text{ pro } j = 1, 2, \dots, n\}$ . Omezený polyedr se nazývá *polytop*.

**Definice:** Nechť je dána množina  $S \subseteq \mathbb{R}^n$ . Bod  $x \in \mathbb{R}^n$  je *konvexní kombinací* bodů z  $S$ , jestliže existuje konečná množina bodů  $\{x^1, x^2, \dots, x^t\}$  v  $S$  a vektor  $\lambda \in \mathbb{R}_+^t$  takový, že  $\sum_{i=1}^t \lambda_i = 1$  a  $x = \sum_{i=1}^t \lambda_i x^i$ .

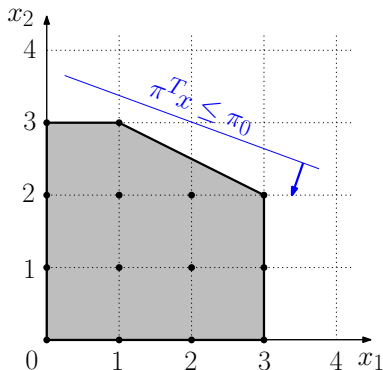
**Definice:**  $T \subseteq \mathbb{R}^n$  je *konvexní množina*, pokud platí: jestliže  $x^1, x^2 \in T$ , pak  $\lambda x^1 + (1 - \lambda)x^2 \in T$  pro všechna  $\lambda \in \langle 0, 1 \rangle$ .

**Definice:** *Konvexní obal*  $\text{conv}(S)$  množiny  $S$  je množina všech bodů, které jsou konvexní kombinací bodů z  $S$ .

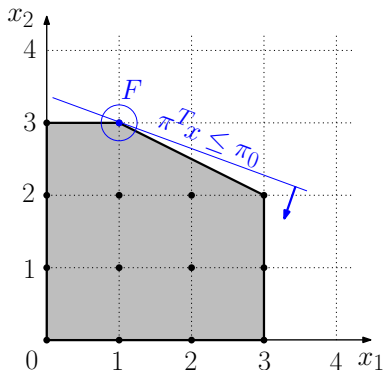


$$S \subset \text{conv}(S) \subset P$$

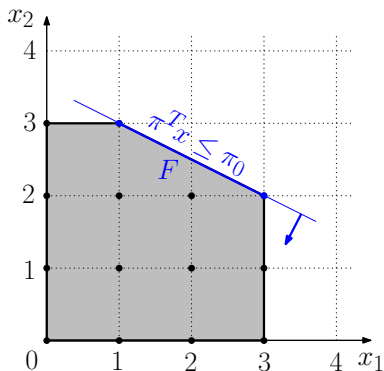
**Definice:** Nerovnost  $\pi^T x \leq \pi_0$  se nazývá *platná nerovnost* vzhledem k  $S$ , jestliže ji splňují všechny body z  $S$ .



**Definice:** Jestliže  $\pi^T x \leq \pi_0$  je platná nerovnost vzhledem k  $S$  a  $\exists x^0 \in S$  takový, že  $\pi^T x^0 = \pi_0$ , pak se nerovnost nazývá *opěrná nerovnost* vzhledem k  $S$ . Množina  $F = \{x \in \text{conv}(S) : \pi^T x = \pi_0\}$  se nazývá *opěrná stěna* množiny  $\text{conv}(S)$ . Říkáme, že opěrná nerovnost  $\pi^T x \leq \pi_0$  *reprezentuje* opěrnou stěnu  $F$ .



**Definice:** Opěrná stěna  $F$  množiny  $\text{conv}(S)$  se nazývá *fazeta* množiny  $\text{conv}(S)$ , jestliže  $\dim F = \dim \text{conv}(S) - 1$ .



**Definice:** Platné nerovnosti  $\pi^T x \leq \pi_0$  a  $\gamma^T x \leq \gamma_0$  se nazývají *ekvivalentní*, jestliže  $\gamma = \lambda\pi$  a  $\gamma_0 = \lambda\pi_0$  pro některé  $\lambda > 0$ .

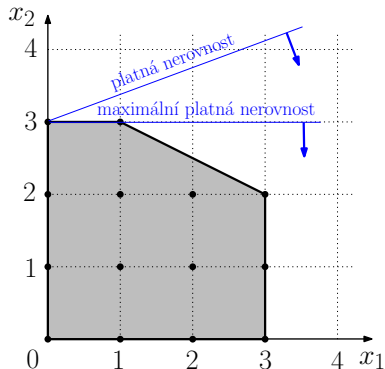
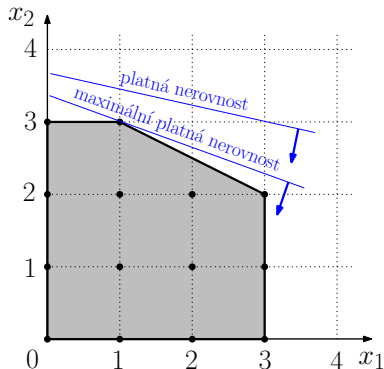
**Definice:** Necht'  $\pi^T x \leq \pi_0$  a  $\gamma^T x \leq \gamma_0$  jsou dvě platné nerovnosti vzhledem k  $\text{conv}(S)$ , které nejsou ekvivalentní. Jestliže existuje hodnota  $\lambda > 0$  taková, že  $\gamma \geq \lambda\pi$  a  $\gamma_0 \leq \lambda\pi_0$ , pak říkáme, že  $\gamma^T x \leq \gamma_0$  *dominuje* (je silnější než)  $\pi^T x \leq \pi_0$ . Říkáme také, že  $\pi^T x \leq \pi_0$  *je dominována* (je slabší než)  $\gamma^T x \leq \gamma_0$ .

Pokud  $\gamma^T x \leq \gamma_0$  *dominuje*  $\pi^T x \leq \pi_0$ , pak  $\{x \in \mathbb{R}_+^n : \gamma^T x \leq \gamma_0\} \subset \{x \in \mathbb{R}_+^n : \pi^T x \leq \pi_0\}$ .

**Definice:** *Maximální* platná nerovnost je taková, která není dominována žádnou jinou platnou nerovností.

# Vlastnosti diskrétních úloh

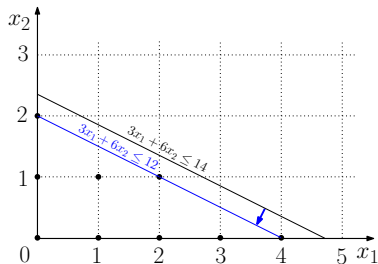
Každá maximální platná nerovnost vzhledem k  $S$  reprezentuje neprázdnou opěrnou stěnu množiny  $\text{conv}(S)$  a množina všech maximálních platných nerovností obsahuje nerovnosti, které reprezentují všechny fazety množiny  $\text{conv}(S)$ .



## Zesilování nerovnosti

**Věta** (Diophantos): Lineární rovnice  $\sum_{j=1}^n \pi_j x_j = \pi_0$ , v níž  $\pi_j \in \mathbb{Z}$  ( $j = 0, 1, \dots, n$ ), má řešení  $x \in \mathbb{Z}^n$ , právě tehdy, když největší společný dělitel koeficientů  $\pi_j$  ( $j = 1, 2, \dots, n$ ) dělí celočíselně hodnotu  $\pi_0$ .

**Příklad:** Nechť  $3x_1 + 6x_2 \leq 14$  je platná nerovnost vzhledem k  $S$ . Cílem je najít silnější platnou nerovnost, která je opěrnou nerovností vzhledem k  $S$ .



## Zesilování nerovností - lifting

**Definice:** Nechť je dána nerovnost  $\sum_{j=1}^n \pi_j x_j \leq \pi_0$ , v níž  $\pi_j \in \mathbb{R}_+$  ( $j = 0, 1, \dots, n$ ) a  $x \in \mathbb{B}^n$ . Jestliže pro nějakou hodnotu  $\Delta_k > 0$  je nerovnost  $\sum_{j=1}^n \pi_j x_j + \Delta_k x_k \leq \pi_0$  platná, pak říkáme, že je *liftovaná* z původní nerovnosti vzhledem k proměnné  $x_k$ .

### Algoritmus:

Opakuj pro  $k = 1, 2, \dots, n$ :

- 1 Nastav  $x_k = 1$  a vypočti  $\alpha_k = \max_{x_k \in \mathbb{B}} \sum_{j=1}^n \pi_j x_j \leq \pi_0$
- 2 Vypočti  $\Delta_k = \pi_0 - \alpha_k$
- 3 Nahraď  $\pi_k$  součtem  $\pi_k + \Delta_k$
- 4 Nerovnost  $\sum_{j=1}^n \pi_j x_j \leq \pi_0$  je liftovaná vzhledem k proměnné  $x_k$

Nerovnost  $\sum_{j=1}^n \pi_j x_j \leq \pi_0$  je liftovaná vzhledem ke všem proměnným.

## Zesilování nerovností - lifting

**Příklad:** Liftujte nerovnost  $4x_1 + 5x_2 + 6x_3 + 8x_4 \leq 13$ , kde  $x_j \in \mathbb{B}$  ( $j = 1, 2, 3, 4$ ).

$$x_1 = 1 \rightarrow \alpha_1 = 12 \rightarrow \Delta_1 = 1 \rightarrow \pi_1 = 5 \rightarrow 5x_1 + 5x_2 + 6x_3 + 8x_4 \leq 13$$

$$x_2 = 1 \rightarrow \alpha_2 = 13 \rightarrow \Delta_2 = 0 \rightarrow \pi_2 = 5 \rightarrow 5x_1 + 5x_2 + 6x_3 + 8x_4 \leq 13$$

$$x_3 = 1 \rightarrow \alpha_3 = 11 \rightarrow \Delta_3 = 2 \rightarrow \pi_3 = 8 \rightarrow 5x_1 + 5x_2 + 8x_3 + 8x_4 \leq 13$$

$$x_4 = 1 \rightarrow \alpha_4 = 13 \rightarrow \Delta_4 = 0 \rightarrow \pi_4 = 8 \rightarrow 5x_1 + 5x_2 + 8x_3 + 8x_4 \leq 13$$

## Zesilování nerovností - fixing

**Příklad:** Nechť je dána nerovnost  $2x_1 + 3x_2 + 4x_3 - 15x_4 \geq 2$ , v níž  $x_j \in \mathbb{B}$  ( $j = 1, 2, 3, 4$ ).

Pokud  $x_4 = 1$ , nerovnost nemůže být splněna  $\rightarrow$  přípustné řešení existuje jen tehdy, jestliže proměnná je zafixována  $x_4 = 0$ .

**Příklad:** Nechť je dána nerovnost  $20x_1 + 5x_2 + 1x_3 - 8x_4 \geq 7$ , v níž  $x_j \in \mathbb{B}$  ( $j = 1, 2, 3, 4$ ).

Zafixování  $x_1 = 1$ .

**Příklad:** Nechť je dána rovnice  $x_1 + x_2 + 3x_3 = 4$ , v níž  $x_j \in \mathbb{B}$  ( $j = 1, 2, 3$ ).

Protože  $x_3 = 0$  je nepřípustné, proměnná je zafixována  $x_3 = 1$ .  
Potom je rovnice redukována na rovnici  $x_1 + x_2 = 1$ .

## Zesilování nerovností - Gomoryho řez

Nechť  $S = \{x \in \mathbb{Z}_+^n : \pi^T x = \pi_0\}$ , kde  $\pi_j \in \mathbb{R}$  ( $j = 0, 1, \dots, n$ ).

Zvolíme-li libovolně  $d \in \mathbb{N}$ , pak každý koeficient  $\pi_j$  lze vyjádřit jako

$$\pi_j = \alpha_j d + \pi'_j, \quad (236)$$

kde  $\alpha_j = \left\lfloor \frac{\pi_j}{d} \right\rfloor$  a  $\pi'_j = \pi_j \bmod d$ . Tedy  $\alpha_j \in \mathbb{Z}$  a  $\pi'_j \in \langle 0, d \rangle$ .

Pak rovnici

$$\sum_{j=1}^n \pi_j x_j = \pi_0 \quad (237)$$

lze psát jako

$$\sum_{j=1}^n (\alpha_j d + \pi'_j) x_j = \alpha_0 d + \pi'_0 \quad (238)$$

nebo

$$d \left( \sum_{j=1}^n \alpha_j x_j - \alpha_0 \right) = \pi'_0 - \sum_{j=1}^n \pi'_j x_j. \quad (239)$$

## Zesilování nerovností - Gomoryho řez

Protože hodnota na levé straně nerovnosti (239) je celočíselným násobkem hodnoty  $d$ , hodnota na pravé straně musí být také celočíselná. Vzhledem k  $\pi'_0 \in \langle 0, d \rangle$  a  $\sum_{j=1}^n \pi'_j x_j \geq 0$ , hodnota na pravé straně nemůže být kladným násobkem  $d$ , tj. musí být nekladná. Proto také hodnota na levé straně musí být nekladná.

*Zlomkový Gomoryho řez:*

$$\pi'_0 - \sum_{j=1}^n \pi'_j x_j \leq 0 \quad \rightarrow \quad \sum_{j=1}^n \pi'_j x_j \geq \pi'_0 \quad (240)$$

*Celočíselný Gomoryho řez:*

$$\sum_{j=1}^n \alpha_j x_j - \alpha_0 \leq 0 \quad \rightarrow \quad \sum_{j=1}^n \alpha_j x_j \leq \alpha_0 \quad (241)$$

## Zesilování nerovností - Gomoryho řez

**Příklad:** Necht'  $S = \{x \in \mathbb{Z}_+^3 : 37x_1 - 68x_2 + 78x_3 \leq 141\}$ .

Nalezněte silnější platnou nerovnost vzhledem k  $S$ .

Transformace na rovnici  $37x_1 - 68x_2 + 78x_3 + x_4 = 141$

Zvolíme  $d = 12$

$$37 = 3 \cdot 12 + 1$$

$$-68 = -6 \cdot 12 + 4$$

$$78 = 6 \cdot 12 + 6$$

$$1 = 0 \cdot 12 + 1$$

$$141 = 11 \cdot 12 + 9$$

Zlomkový Gomoryho řez:  $x_1 + 4x_2 + 6x_3 + x_4 \geq 9$

Celočíselný Gomoryho řez:  $3x_1 - 6x_2 + 6x_3 \leq 11$

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskretních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

## Unimodularita

**Definice:** Čtvercová celočíselná matice  $A$  se nazývá *unimodulární*, jestliže její determinant  $\det(A) = \pm 1$ . Celočíselná matice  $A$  se nazývá *totálně unimodulární*, jestliže každá čtvercová nesingulární submatice  $A$  je unimodulární.

**Pozn.:** Je-li matice  $A$  totálně unimodulární, pak  $a_{ij} \in \{+1, -1, 0\}$  pro všechna  $i, j$ .

**Věta** (postačující podmínka): Matice  $A$  je totálně unimodulární, jestliže

- $a_{ij} \in \{+1, -1, 0\}$  pro všechna  $i, j$
- každý sloupec obsahuje nejvýše dva nenulové koeficienty
- řádky matice  $A$  mohou být rozděleny do dvou množin takových, že
  - ▶ jestliže sloupec obsahuje dva koeficienty se stejným znaménkem, jejich řádky leží v různých množinách
  - ▶ jestliže sloupec obsahuje dva koeficienty s různými znaménky, jejich řádky leží ve stejné množině

## Unimodularita

Nechť je dána následující úloha LP s celočíselnými  $A, b$ :

$$z_{LP} = \max\{c^T x : Ax \leq b, x \in \mathbb{R}_+^n\}. \quad (242)$$

Vektor základních proměnných lze vyjádřit jako

$$x_B = B^{-1}b = \frac{B^{\text{adj}}}{\det(B)}b, \quad (243)$$

kde  $B$  je matice báze,  $B^{-1}$  její inverze a  $B^{\text{adj}}$  je adjungovaná matice  $B$  (transponovaná matice algebraických doplňků prvků matice  $B$ ).

**Pozn.:** Je-li optimální matice báze  $B$  unimodulární, pak optimální řešení je celočíselné.

**Tvrzení:** Je-li matice  $A$  totálně unimodulární, pak optimální řešení je celočíselné.

**Příklady:** dopravní problém, úloha hledání maximálního toku, ...

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskrétních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

**Definice:** Nechť je dána následující úloha celočíselného programování (IP):

$$z_{\text{IP}} = \max\{c^T x : x \in S \subseteq \mathbb{Z}_+^n\}. \quad (244)$$

Úloha (R)

$$z_{\text{R}} = \max\{d^T x : x \in X \subseteq \mathbb{R}_+^n\} \quad (245)$$

je *relaxace* úlohy (IP), jestliže

- $S \subseteq X$
- $d^T x \geq c^T x$  pro všechna  $x \in X$ .

**Tvrzení:** Je-li úloha (R) relaxace úlohy (IP), pak  $z_{\text{R}} \geq z_{\text{IP}}$ , tj.  $z_{\text{R}}$  je horní mez pro  $z_{\text{IP}}$ .

## Lineární relaxace

**Definice:** Nechť je dána následující úloha celočíselného programování (IP):

$$z_{\text{IP}} = \max\{c^T x : Ax \leq b, x \in \mathbb{Z}_+^n\}. \quad (246)$$

Úloha (LP)

$$z_{\text{LP}} = \max\{c^T x : Ax \leq b, x \in \mathbb{R}_+^n\} \quad (247)$$

je *lineární relaxace* úlohy (IP).

V případě úlohy bivalentního programování (BIP)

$$z_{\text{BIP}} = \max\{c^T x : Ax \leq b, x \in \mathbb{B}^n, \mathbb{B} = \{0, 1\}\} \quad (248)$$

je lineární relaxace definována jako

$$z_{\text{LP}} = \max\{c^T x : Ax \leq b, 0 \leq x_j \leq 1, j = 1, 2, \dots, n\}. \quad (249)$$

## Lineární relaxace

**Definice:** Hodnota *absolutní gap* je definována jako rozdíl

$$\text{Gap} = z_{\text{LP}} - z_{\text{IP}} \quad (250)$$

a pro  $z_{\text{IP}} \neq 0$  je hodnota *relativní gap* definována jako

$$\text{Gap}\% = \frac{z_{\text{LP}} - z_{\text{IP}}}{|z_{\text{IP}}|} 100\%. \quad (251)$$

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskrétních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - **Exaktní metody**
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

## Metoda řezných nadrovin

Algoritmus je založen na simplexové metodě.

*Řezná nadrovina* (nebo jednoduše *řez*) je lineární omezující podmínka, která nevylučuje žádné přípustné celočíselné řešení.

- 1 Vyřeš lineární relaxaci původní úlohy.
- 2 Je-li optimální řešení celočíselné, jdi na krok 5.
- 3 Vyber proměnnou, jejíž optimální hodnota není celočíselná a sestav Gomoryho řez podle (240). Ten je přidán do simplexové tabulky jako

$$-\sum_{j=1}^n \pi'_j x_j + x_{n+1} = -\pi'_0, \quad (252)$$

kde  $x_{n+1}$  je přídatná proměnná.

- 4 Pokračuj duálním simplexovým algoritmem pro získání optimálního řešení. Jdi na krok 2.
- 5 Konec

## Metoda větvení a mezí

**Tvrzení:** Je dána úloha  $z_{\text{IP}} = \max\{c^T x : x \in S\}$ . Necht'  $S = S_1 \cup S_2 \cup \dots \cup S_K$  je dekompozice množiny  $S$  na menší množiny a necht'  $z^k = \max\{c^T x : x \in S_k\}$  pro  $k = 1, 2, \dots, K$ . Pak  $z_{\text{IP}} = \max_k z^k$ .

## Značení:

$M$  je posloupnost úloh řešených ve větvích výpočetního stromu,  $x^*$  je dosud nejlepší nalezené celočíselné řešení,  $z^* = c^T x^*$  je dosud nejlepší hodnota účelové funkce pro celočíselné řešení.

## Algoritmus:

### 1 Počáteční nastavení

$M = (LP)$ ,  $LP$  je lineární relaxace původní úlohy,  
 $x^*$  není definován,  
 $z^* = -\infty$ .

## Metoda větvení a mezí

### ② *Výběr řešené úlohy*

Pokud  $M = ()$ , jdi na krok 5,  
jinak vyber poslední úlohu v posloupnosti  $M$ .

### ③ *Řešení vybrané úlohy*

- (a) Jestliže neexistuje přípustné řešení, odstraň úlohu z  $M$  a jdi na krok 2.
- (b) Je-li nalezeno optimální řešení  $x^0$  s hodnotou účelové funkce  $z^0$ , pak
  - (b1) pokud  $z^0 \leq z^*$ , odstraň úlohu z  $M$  a jdi na krok 2,
  - (b2) pokud  $z^0 > z^*$  a řešení  $x^0$  je celočíselné, nahraď  $x^* = x^0, z^* = z^0$ , odstraň úlohu z  $M$  a jdi na krok 2,
  - (b3) pokud  $z^0 > z^*$  a řešení  $x^0$  není celočíselné, jdi na krok 4.

## Metoda větvení a mezí

### 4 *Větvení*

Vyber proměnnou  $x_k$ , jejíž optimální hodnota není celočíselná. Přidej kopii poslední řešené úlohy na konec posloupnosti  $M$  spolu s omezující podmínkou

$$x_k \leq \lfloor x_k^0 \rfloor. \quad (253)$$

K předposlední úloze v  $M$  přidej omezující podmínku

$$x_k \geq \lfloor x_k^0 \rfloor + 1. \quad (254)$$

Jdi na krok 2.

### 5 *Konec*

Optimální celočíselné řešení je  $x^*$ , optimální hodnota účelové funkce je  $z^*$ .

## Metoda větvení a mezí

**Tvrzení:** Větev výpočetního stromu je uzavřena, pokud v jejím uzlu nastane jedna ze tří možností:

- úloha nemá řešení,
- optimální řešení  $x^0$  je celočíselné,
- platí  $z^0 \leq z^*$ .

V případě bivalentní úlohy (v níž je  $n$  binárních proměnných), je  $n$  maximální hloubka výpočetního stromu a  $2^n$  maximální počet jeho listů.

## Metoda větvení a mezí

### Výběr uzlu

- Apriorní pravidla
  - ▶ uzly jsou vybírány v opačném pořadí, než jsou vytvářeny (LIFO)
  - ▶ uzly se zpracovávají po vrstvách stromu
- Adaptivní pravidla
  - ▶ výběr uzlu s nejvyšší horní mezí

### Výběr větvící proměnné

- Výběr proměnné, jejíž hodnota nejvíce porušuje celočíselnost
- Výběr proměnné na základě řešení lineárních relaxací s dolní a horní mezí dané proměnné (provede se pouze 1 iterace simplexové tabulky, jedná se tedy o odhad poklesu hodnoty účelové funkce)

## Metoda větvení a mezí

### Vylepšení metody

- Metoda větvení a řezů  
Jedná se o kombinaci metody větvení a mezí s metodou řezných nadrovin za účelem zesílit lineární relaxace v uzlech výpočetního stromu.
- Metoda větvení a oceňování  
Metoda se používá pro řešení úloh IP a MIP s velkým počtem proměnných. Jedná se o hybridní spojení metody větvení a mezí s algoritmem generování sloupců.

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskrétních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

## Složitost algoritmů

Velikost (rozsah) úlohy:

- lineární programování  
 $m$  ... počet omezujících podmínek  
 $n$  ... počet proměnných
- úlohy na grafech  
 $|U|$  ... počet uzlů  
 $|E|$  ... počet hran

Výpočetní složitost algoritmu je funkcí velikosti úlohy, kterou algoritmus řeší, např.  $f(n)$ .

## Složitost algoritmů

Nechť elementární operace v počítači trvá 1 ns. V následující tabulce je uvedena závislost výpočetního času pro různé funkce a velikost úlohy.

$f(n)$	$n$ (velikost úlohy)				
	10	20	50	100	1000
$n$	10 ns	20 ns	50 ns	100 ns	1 $\mu$ s
$n \log n$	10 ns	26 ns	85 ns	200 ns	3 $\mu$ s
$n^2$	100 ns	400 ns	2.5 $\mu$ s	10 $\mu$ s	1 ms
$n^3$	1 $\mu$ s	8 $\mu$ s	125 $\mu$ s	1 ms	1 s
$2^n$	1 $\mu$ s	1 ms	13 dnů	$10^{13}$ let	-
$3^n$	59 $\mu$ s	4 s	$10^7$ let	-	-
$n!$	4 ms	77 let	-	-	-

## Složitost algoritmů

Důležitá je asymptotická míra růstu složitosti algoritmu.

**Definice:** Necht'  $f(n)$  a  $g(n)$  jsou reálné funkce přirozených čísel.

- Píšeme  $f(n) = O(g(n))$ , pokud existuje konstanta  $c > 0$  taková, že pro dostatečně velká  $n$  je  $f(n) \leq cg(n)$  (notace velké  $O$ ).
- Píšeme  $f(n) = \Omega(g(n))$ , pokud existuje konstanta  $c > 0$  taková, že pro dostatečně velká  $n$  je  $f(n) \geq cg(n)$  (notace velké omega).
- Píšeme  $f(n) = \Theta(g(n))$ , pokud existují konstanty  $c, c' > 0$  takové, že pro dostatečně velká  $n$  je  $cg(n) \leq f(n) \leq c'g(n)$  (notace velké theta).

Polynomiální algoritmy:  $n, n^2, n^3, \log n, n \log n$

Nepolynomiální algoritmy:  $2^n, e^n, n!$

## Třídy úloh

### Třída $\mathcal{P}$

Jedná se o třídu rozhodovacích úloh, které lze řešit v polynomiálním čase. Rozhodovací úloha velikosti  $n$  leží ve třídě  $\mathcal{P}$ , jestliže pro ni existuje algoritmus s  $f(n) = O(n^p)$  pro nějaké pevně dané  $p$ .

### Třída $\mathcal{PO}$

Je to třída optimalizačních úloh řešitelných v polynomiálním čase. Optimalizační úloha rozměru  $n$  leží ve třídě  $\mathcal{PO}$ , jestliže pro ni existuje algoritmus s  $f(n) = O(n^p)$  pro nějaké pevně dané  $p$ .

Vybrané úlohy řešitelné v polynomiálním čase.

- Minimální kostra grafu.
- Úloha hledání nejkratší cesty v grafu.
- Úloha hledání maximálního toku.
- Lineární přiřazovací problém.
- Úloha lineárního programování.

## Třídy úloh

### Třída $\mathcal{NP}$

Jedná se o třídu rozhodovacích úloh řešitelných *nedeterministickým polynomiálním algoritmem* sestávajícím ze dvou fází:

- 1 Navrhovací (nedeterministická) fáze  
Je vygenerováno řešení.
- 2 Ověřující fáze  
Pomocí polynomiálního algoritmu je dokazováno, zda řešení je přípustné.

Rozhodovací úloha leží ve třídě  $\mathcal{NP}$ , jestliže kladné rozhodnutí je ověřeno v polynomiálním čase.

Leží-li úloha ve třídě  $\mathcal{P}$ , pak leží také ve třídě  $\mathcal{NP}$ , tj.  $\mathcal{P} \subset \mathcal{NP}$ .

Příklady  $\mathcal{NP}$ -řešitelných úloh: hledání Hamiltonova cyklu v daném grafu, hledání přípustného řešení úlohy MIP, ...

## Třídy úloh

### Třída $\mathcal{NPO}$

Optimalizační úloha leží ve třídě  $\mathcal{NPO}$ , pokud její rozhodovací verze leží ve třídě  $\mathcal{NP}$ .

Leží-li úloha ve třídě  $\mathcal{PO}$ , pak leží ve třídě  $\mathcal{NPO}$ , tj.  $\mathcal{PO} \subset \mathcal{NPO}$ .

**Definice:** Rozhodovací úloha  $A$  je *polynomiálně redukovatelná* na rozhodovací úlohu  $B$ , jestliže existuje polynomiální funkce transformující zadání úlohy  $A$  na zadání úlohy  $B$  taková, že z řešení úlohy  $B$  lze odvodit řešení úlohy  $A$ .

Jestliže je rozhodovací úloha  $A$  redukovatelná na rozhodovací úlohu  $B$ , pak pokud existuje algoritmus pro řešení úlohy  $B$ , lze jej použít pro řešení úlohy  $A$ .

Rozhodovací úloha  $A$  je speciální případ rozhodovací úlohy  $B$ , tj.  $B$  je obecnější, a tudíž složitější.

## Třídy úloh

**Tvrzení:** Je-li úloha  $A$  polynomiálně redukovatelná na úlohu  $B$  a  $B \in \mathcal{P}$ , pak  $A \in \mathcal{P}$ .

**Tvrzení:** Je-li úloha  $A$  polynomiálně redukovatelná na úlohu  $B$  a  $B \in \mathcal{NP}$ , pak  $A \in \mathcal{NP}$ .

## Třída $\mathcal{NPC}$

Říkáme, že rozhodovací úloha  $A \in \mathcal{NP}$  je  $\mathcal{NP}$ -úplná, pokud všechny problémy ve třídě  $\mathcal{NP}$  lze polynomiálně redukovat na úlohu  $A$ .

**Tvrzení:** Je-li úloha  $A \in \mathcal{NPC}$  polynomiálně redukovatelná na úlohu  $B \in \mathcal{NP}$ , pak  $B \in \mathcal{NPC}$ .

**Důsledek:** Abychom dokázali, že daná úloha je  $\mathcal{NP}$ -úplná, je nutné:

- prokázat, že úloha leží ve třídě  $\mathcal{NP}$  a
- najít nějakou známou  $\mathcal{NP}$ -úplnou úlohu, která je na danou úlohu redukovatelná.

## Třídy úloh

**Tvrzení:** Jestliže  $\mathcal{P} \cap \mathcal{NPC} \neq \emptyset$ , pak  $\mathcal{P} = \mathcal{NP}$ .

**Důsledek:** Kdyby existoval polynomiální algoritmus, který by řešil nějakou  $\mathcal{NP}$ -úplnou úlohu, pak, s použitím redukovatelnosti, bychom všechny úlohy v  $\mathcal{NP}$  dokázali řešit v polynomiálním čase.

Příklady  $\mathcal{NP}$ -úplných úloh (některé z nich jsou binárními verzemi optimalizačních úloh):

- Přípustné řešení úlohy BIP.
- Přípustné řešení rozkladu množiny.
- Přípustné řešení úlohy batohu s dolní mezí účelové funkce.
- Hamiltonův cyklus.
- Přípustné řešení úlohy TSP s horní mezí účelové funkce.
- Přípustné řešení kvadratického přiřazovacího problému s horní mezí účelové funkce.
- Rozdělení dané množiny přirozených čísel do dvou podmnožin se stejným součtem (Partition problem).

## Třídy úloh

### Třída $\mathcal{NP}\mathcal{H}$

Optimalizační úloha je  $\mathcal{NP}$ -obtížná, těžká, jestliže její rozhodovací verze leží ve třídě  $\mathcal{NPC}$ .

Příklady  $\mathcal{NP}$ -obtížných úloh:

- Úloha IP.
- Úloha batohu.
- TSP.
- Minimální Steinerův strom.
- Kvadratický přiřazovací problém.
- Kontejnerový dopravní problém.

- 1 Úloha celočíselného programování
- 2 Modelování základních úloh IP a MIP
- 3 Úlohy na grafech
  - Tokové úlohy
  - Okružní a rozvozní úlohy
- 4 Speciální omezení s logickými proměnnými
- 5 Vlastnosti diskrétních úloh
- 6 Řešení úloh - metody a algoritmy
  - Relaxace
  - Exaktní metody
  - Výpočetní složitost
  - Heuristiky a metaheuristiky

Přibližné metody:

- Heuristika

Je to postup, který vede k získání dobrého řešení (blízkého optimálnímu řešení) konkrétní optimalizační úlohy.

- Metaheuristika

Je to přístup, který lze přizpůsobit k řešení širokého okruhu úloh.

Základní principy použití přibližné metody:

- používá se k řešení úloh  $\mathcal{NPC}$  a  $\mathcal{NPH}$ ,
- nezaručuje nalezení optimálního řešení (poskytuje tzv. suboptimální řešení),
- je to polynomiální algoritmus,
- lze jej snadno upravit pro řešení konkrétní úlohy.

## Heuristiky pro TSP

Klasifikace metod:

- Konstruktivní heuristiky.
- Zatřídovací heuristiky.
- Zlepšující heuristiky.

**Předpoklad:** je dána matice vzdáleností mezi všemi dvojicemi uzlů (je definován úplný graf)

### Metoda nejbližšího souseda

- 1 Vyber libovolný uzel jako výchozí pro vytvářenou trasu.
- 2 Najdi nejbližší uzel (dosud nevybraný) k poslednímu uzlu na trase a přidej ho na konec trasy. Pokud takový uzel neexistuje (všechny uzly již byly vybrány), pak přidej na konec trasy výchozí uzel (je vytvořen Hamiltonův cyklus) a jdi na krok 4.
- 3 Jdi na krok 2.
- 4 Konec.

## Metoda výhodnostních čísel (Clarke a Wright)

- 1 Vypočti výhodnostní čísla

$$s_{ij} = c_{i1} + c_{1j} - c_{ij} \quad \text{pro} \quad \begin{matrix} i = 2, 3, \dots, n \\ j = 2, 3, \dots, n \end{matrix} \quad i \neq j. \quad (255)$$

- 2 Vytvoř  $(n - 1)$  tras  $(1, i, 1)$  pro  $i = 2, 3, \dots, n$  a seříd' výhodnostní čísla sestupně.

### Paralelní verze

- 3 (Nejlepší přípustné zatřídění)

Podle seříděných čísel postupujeme následujícím způsobem.

Pro dané výhodnostní číslo  $s_{ij}$  zjistíme, zda existují dvě trasy, z nichž první obsahuje hranu  $(1, j)$  a druhá hranu  $(i, 1)$ , které mohou být přípustným způsobem zatříděny. V takovém případě spojíme tyto dvě trasy v jednu odstraněním hran  $(1, j)$ ,  $(i, 1)$  a vložení hranou  $(i, j)$ .

## Metoda výhodnostních čísel (Clarke a Wright)

### Sekvenční verze

- 3 (Nastavení trasy)  
Uvažujme trasu  $(1, i, \dots, j, 1)$ . Najdi první výhodnostní číslo  $s_{ki}$  nebo  $s_{jl}$  takové, že uzly  $k$  a  $l$  jsou zahrnuty v jiných trasách obsahujících hranu  $(k, 1)$  nebo hranu  $(1, l)$ .
- 4 Proveď zatřídění a opakuj tento postup dokud není vytvořen Hamiltonův cyklus.

### Vkládací algoritmus

- 1 Vyber libovolný uzel jako výchozí pro vytvářenou trasu, např. uzel 1.
- 2 Najdi nejvzdálenější uzel  $s$  k výchozímu uzlu a vytvoř trasu  $(1, s, 1)$ .
- 3 Do stávající trasy postupně vkládej další dosud nezahrnuté uzly (na základě minimalizace prodloužení trasy), dokud nevznikne Hamiltonův cyklus.

## Metoda zatřídování cyklů

Nechť je daný úplný graf  $G = \{U, E\}$ .

- 1 Najdi výchozí systém cyklů  $\mathcal{F}$  (např. použitím metody perfektního párování; jestliže počet uzlů v  $U$  je lichý, pak jeden z cyklů bude obsahovat 3 uzly).
- 2 Proveď zatřídění dvou cyklů  $\alpha^*$  a  $\beta^*$  s použitím následující metriky:

$$D_{\alpha^*\beta^*} = \min_{\alpha, \beta \in \mathcal{F}} D_{\alpha\beta} = \min_{\substack{i, k \in \alpha \\ j, l \in \beta}} (c_{ij} + c_{kl} - c_{ik} - c_{jl}). \quad (256)$$

Nechť tímto zatříděním vznikne cyklus  $\gamma$ .

- 3 Odstraň cykly  $\alpha^*$  a  $\beta^*$  z  $\mathcal{F}$ , přidej do  $\mathcal{F}$  cyklus  $\gamma$ .
- 4 Jestliže  $\gamma$  není Hamiltonův cyklus, jdi na krok 2.

## Metoda výměn (Lin a Kernighen)

Nechť je daný úplný graf  $G = \{U, E\}$ .

- 1 Předpokládejme nalezení Hamiltonova cyklu některou konstruktivní nebo zatřídňovací heuristikou.
- 2 Vyměň dvě nesousední hrany z trasy za jiné dvě nesousední hrany tak, aby byl opět vytvořen Hamiltonův cyklus.
- 3 Pokud tato výměna přináší zlepšení řešení, proved' ji.
- 4 Opakuj proces hledání výměn dokud je možné zlepšovat řešení. Metoda končí, jestliže již není možné řešení zlepšit.
- 5 Získaný Hamiltonův cyklus je lokálně optimální (2-opt) trasa.

## Metaheuristiky

### Značení v algoritmech:

$x$  je přípustné řešení dané úlohy

$X$  je množina přípustných řešení, tj. množina všech  $x$

$N(x)$  je okolí řešení  $x$  (množina blízkých řešení)

$f(x)$  je minimalizační účelová funkce

$x^*$  je dosud nejlepší nalezené řešení

### Metoda lokálního hledání

- 1 Vyber výchozí řešení  $x \in X$  a přiřaď  $x^* = x$ .
- 2 Definuj okolí  $N(x) \subseteq X$  a ohodnoť všechna řešení.
- 3 Nechť  $x'$  je nejlepší řešení z  $N(x)$ .  
Jestliže  $f(x') < f(x^*)$ , pak nahraď  $x^* = x'$  a  $x = x'$ , jinak ukonči hledání.
- 4 Pokud není splněno ukončovací pravidlo, jdi na krok 2.
- 5 Řešení  $x^*$  odpovídá lokálnímu minimu.

## Metoda tabu prohledávání

$TL$  je posloupnost zakázaných řešení (tabu seznam)

$MaxSize$  je maximálně povolený počet řešení v tabu seznamu

- 1 Vyber výchozí řešení  $x \in X$  a přiřaď  $x^* = x$ . Nastav  $TL = \{x\}$ .
- 2 Definuj okolí  $N(x) \subseteq X \setminus TL$  a ohodnoť všechna řešení.
- 3 Nechť  $x'$  je nejlepší řešení z  $N(x)$ . Nahaď  $x = x'$ .  
Jestliže  $f(x') < f(x^*)$ , pak nahaď  $x^* = x'$ .
- 4  $TL = TL \cup \{x\}$ . Jestliže  $|TL| > MaxSize$ , pak z  $TL$  odstraň první řešení.
- 5 Pokud není splněno ukončovací pravidlo, jdi na krok 2.
- 6 Řešení  $x^*$  je nejlepší nalezené řešení.

## Metoda prahové akceptace

$T$  je hodnota prahu akceptace horších řešení

$T_0$  je počáteční hodnota prahu

$r \in (0, 1)$  je míra redukce prahu

- 1 Vyber výchozí řešení  $x \in X$  a přiřaď  $x^* = x$ . Nastav  $T = T_0$ .
- 2 Opakuj  $n$ -krát:
  - ▶ vyber  $x' \in N(x)$ ,
  - ▶ jestliže  $f(x') - T < f(x)$ , pak nahraď  $x = x'$ ,
  - ▶ jestliže  $f(x') < f(x^*)$ , pak nahraď  $x^* = x'$ .
- 3 Pokud není splněno ukončovací pravidlo, proved' redukci prahu  $T = rT$  a jdi na krok 2.
- 4 Řešení  $x^*$  je nejlepší nalezené řešení.

## Metoda simulovaného žíhání (SIAM)

$T$  je teplota

$T_0$  je počáteční teplota

$r \in (0, 1)$  je míra redukce teploty (míra ochlazování)

- 1 Vyber výchozí řešení  $x \in X$  a přiřaď  $x^* = x$ . Nastav  $T = T_0$ .
- 2 Opakuj  $n$ -krát:
  - ▶ vyber  $x' \in N(x)$ ,
  - ▶ jestliže  $f(x') < f(x)$ , pak nahraď  $x = x'$ ,
  - ▶ jestliže  $f(x') \geq f(x)$ , pak nahraď  $x = x'$  s pravděpodobností  $e^{-\frac{\Delta}{T}}$ , kde  $\Delta = f(x') - f(x)$ ,
  - ▶ jestliže  $f(x') < f(x^*)$ , pak nahraď  $x^* = x'$ .
- 3 Pokud není splněno ukončovací pravidlo (nebo nenastalo ztuhnutí procesu žíhání), pak proved' redukci teploty  $T = rT$  a jdi na krok 2.
- 4 Řešení  $x^*$  je nejlepší nalezené řešení.

## Genetický algoritmus

### Definice:

*Populace*  $R \subseteq X$  je konečná množina přípustných řešení.

*Fitness value*  $f(x)$  je ohodnocení řešení  $x \in R$ .

*Selekce* je výběr určitého páru řešení (rodičů)  $x, y \in R$  založený na jejich fitness value.

*Křížení* je operace spojení rodičů za účelem získání jednoho nebo dvou nových řešení (*potomků*).

*Mutace* je náhodná změna potomka.

Cílem selekce rodičů je vybrat lepší řešení s vyšší pravděpodobností. Předpokládejme, že fitness  $f(x)$  je maximalizačního typu. Pak pravděpodobnost výběru rodiče  $x$  je dána jako

$$\frac{f(x)}{\sum_{\forall y \in R} f(y)} \quad (257)$$

## Genetický algoritmus

Nechť rodič  $x$  je dán kódem  $x_1x_2 \dots x_r$ , rodič  $y$  kódem  $y_1y_2 \dots y_r$   
a potomek  $z$  kódem  $z_1z_2 \dots z_r$ .

- Křížení rodičů

- ▶ Jednobodové křížení

- Potomek #1:  $x_1 \dots x_p y_{p+1} \dots y_r$ ,

- Potomek #2:  $y_1 \dots y_p x_{p+1} \dots x_r$ .

- ▶ Dvoubodové křížení

- Potomek #1:  $x_1 \dots x_p y_{p+1} \dots y_q x_{q+1} \dots x_r$ ,

- Potomek #2:  $y_1 \dots y_p x_{p+1} \dots x_q y_{q+1} \dots y_r$ .

- ▶ Rovnoměrné křížení

- Potomek:  $z_1z_2 \dots z_r$ , kde  $z_i \in \{x_i, y_i\}, i = 1, 2, \dots, r$ .

- Mutace potomka

- ▶ Jednobodová mutace

- Změněný potomek:  $z_1 \dots z_{p-1} z_p^* z_{p+1} \dots z_r$ , kde  $z_p^* \neq z_p$ .

- ▶ Dvoubodová mutace

- Změněný potomek:  $z_1 \dots z_{p-1} z_q z_{p+1} \dots z_{q-1} z_p z_{q+1} \dots z_r$ .

## Mravenčí (feromonový) algoritmus

- Metoda je založena na *inteligenci mravenčích kolonií*.
- Každý mravenec se snaží najít cestu mezi mraveništěm a potravou.
- Mravenci zanechávají na cestě *feromonovou* stopu.
- Mravenci preferují cesty obsahující větší množství feromonu.
- Feromonové stopy na delších cestách se *vypařují* rychleji než na kratších cestách.
- Vypařování feromonu má velký význam, když zamezuje konvergenci k lokálnímu optimu.
- Idea algoritmu je napodobovat toto chování pomocí „simulovaných mravenců“ (umělých agentů) procházejících grafem reprezentujícím řešenou úlohu.