

## HEURISTICKÉ ALGORITMY PRO ŘEŠENÍ ÚLOH OBCHODNÍHO CESTUJÍCÍHO

Heuristické algoritmy jsou speciálními algoritmy, které byly vyvinuty pro obtížné úlohy, jejichž řešení je obtížné získat v rozumném čase. Mezi takovéto obtížné úlohy, pro něž neexistuje polynomiální algoritmus, patří úloha obchodního cestujícího. Pro zopakování: v úloze obchodního cestujícího máme zadáno výchozí místo a uzly  $2,3,\dots,n$  představující zákazníky, které musíme navštívit. Vzdálenost mezi uzly  $i$  a  $j$  značíme  $c_{ij}$ . Matici vzdáleností  $C$  získáme tak, že určíme nejkratší možnou cestu mezi každou dvojicí uzlů, a to ať už jsou přímo spojeny hranou, nebo ne (v tom případě musíme zkrátka jet přes jiný uzel – jako bychom dodefinovali umělou hranu, jejíž délka se rovná nejkratší vzdálenosti mezi těmito uzly). Naším cílem je navštívit všechny zákazníky a vrátit se do výchozího místa tak, aby délka trasy byla minimální. Jinak řečeno, chceme najít Hamiltonův cyklus s minimálním součtem ohodnocení hran.

Heuristiky nezajišťují nalezení optimálního řešení, ale umožní najít dostatečně dobré řešení v rozumném čase. Lze je rozdělit na algoritmy, které vytvoří nové řešení (trasa je vytvářena postupně z výchozího uzlu – sem patří metoda nejbližšího souseda, metoda vkládací, metoda výhodnostech čísel a metody založené na algoritmu minimální kostry), a na metody, které nalezené řešení zlepšují (sem patří například metoda 2-opt).

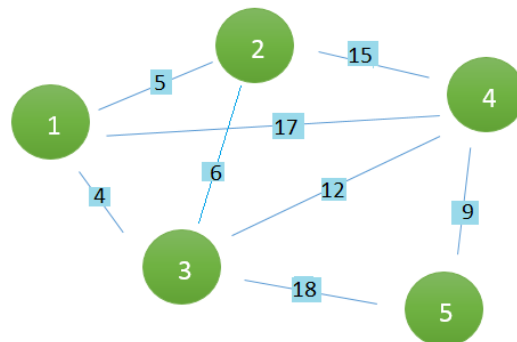
Ke všem algoritmům potřebujeme matici nejkratších vzdáleností. Nejkratší cestu v grafu lze najít pomocí tzv. Dijkstrova algoritmu.

Uvažujme následující graf. V Lingu najdeme nejkratší cestu z uzlu 1 do uzlu 4 takto:

```
model: !hledáme nejkratší cestu mezi uzlem
1 a 4;
sets:
uzel/1..5/;
tok(uzel,uzel):x,C;
endsets
```

```
data: !tam, kde neexistují hrany, dáme
1000;
C=
1000 5 4 1000 1000
5 1000 6 15 1000
4 6 1000 12 18
1000 15 12 1000 9
1000 1000 18 9 1000;
enddata
```

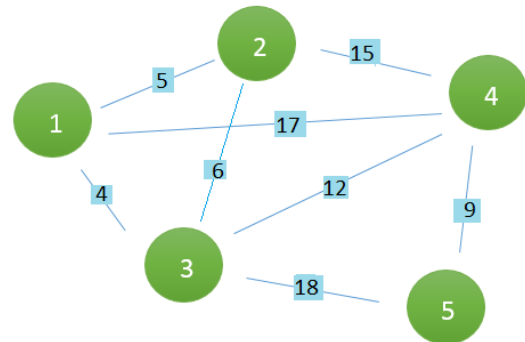
```
min = @sum(tok: x*C);
@for(uzel(i)|i#NE#1#AND#i#NE#4: @sum(uzel(j): x(i,j)) = @sum(uzel(k): x(k,i))); !z
každého uzlu kromě 1 a 4 musíme tolikrát vyjet, kolikrát do něj přijedeme;
@for(uzel(i)|i#EQ#1: @sum(uzel(j): x(i,j)) = 1);
@for(uzel(i)|i#EQ#1: @sum(uzel(j): x(j,i)) = 0);
@for(uzel(i)|i#EQ#4: @sum(uzel(j): x(i,j)) = 0);
@for(uzel(i)|i#EQ#4: @sum(uzel(j): x(j,i)) = 1);!do prvního uzlu nepřijedeme ani
jednou, ale musíme jednou vyjet, u čtvrtého uzlu je to naopak, do ostatních uzlu
nemusíme ani přijet ani z nich vyjet;
@for(tok: @bin(x));
end
```



**METODA NEJBLIŽŠÍHO SOUSEDA:**

1. Vybereme libovolný uzel.
2. Nalezneme nejbližšího souseda k vybranému uzlu.
3. Nalezneme nejbližšího souseda k uzlu z bodu 2 atd., dokud nevznikne Hamiltonův cyklus.

Tato metoda je rychlá a jednoduchá, ale nemusí dávat dobré výsledky, i proto, že na konci se musíme vrátit do výchozího uzlu, který může být dost daleko od posledního navštíveného uzlu. Je možné zvolit jako výchozí uzel postupně všechny uzly a vybrat nejlepší řešení.



Příklad: máme stejnou úlohu jako dříve. Pomocí metody nejbližšího souseda ji řešíme takto.

- (1) Sestavíme matici nejkratších vzdáleností.
- (2) Vyjedeme z uzlu 1. První sloupec v matici nejkratších vzdáleností můžeme vyškrtnout, protože do uzlu jedna už nepojedeme, dokud neobjedeme všechny uzly. Nejbližším sousedem uzlu 1 je uzel 3, pojedeme tedy tam.
- (3) Vyškrtneme třetí sloupec, protože z jiného uzlu už znovu do uzlu 3 nepojedeme. Nejbližším potenciálním sousedem, kam můžeme jet z uzlu 3, je uzel 2. Pojedeme tedy do uzlu 2.
- (4) Vyškrtneme druhý sloupec, protože z jiného uzlu už znovu do uzlu 2 nepojedeme. Nyní jsme tedy v uzlu 2 a nejbližším sousedem, kam můžeme jet, je uzel 4. Pojedeme tedy do uzlu 4.
- (5) Vyškrtneme čtvrtý sloupec, protože z jiného uzlu už do něj nepojedeme. Z uzlu 4 jedeme do uzlu 5.
- (6) Uzel 5 je posledním navštíveným uzlem. Z něj se tedy vrátíme do uzlu 1. Celá trasa je tudíž  $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 1$ . Celková ujetá vzdálenost je 56. Skutečně optimální řešení je přitom 51.

(1)

C	1	2	3	4	5
1	0	5	4	16	22
2	5	0	6	15	24
3	4	6	0	12	18
4	16	15	12	0	9
5	22	24	18	9	0

(2)

C	1	2	3	4	5
<b>1</b>	0	5	<u>4</u>	16	22
2	5	0	6	15	24
3	4	6	0	12	18
4	16	15	12	0	9
5	22	24	18	9	0

(3)

C	1	2	3	4	5
<b>1</b>	0	5	<u>4</u>	16	22
2	5	0	6	15	24
<b>3</b>	4	<u>6</u>	0	12	18
4	16	15	12	0	9
5	22	24	18	9	0

(4)

C	1	2	3	4	5
<b>1</b>	0	5	<u>4</u>	16	22
<b>2</b>	5	0	6	<u>15</u>	24
3	4	<u>6</u>	0	12	18
4	16	15	12	0	9
5	22	24	18	9	0

(5)

C	1	2	3	4	5
<b>1</b>	0	5	<u>4</u>	16	22
<b>2</b>	5	0	6	<u>15</u>	24
<b>3</b>	4	<u>6</u>	0	12	18
<b>4</b>	16	15	12	0	<u>9</u>
5	22	24	18	9	0

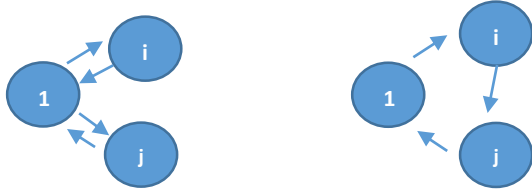
(6)

C	1	2	3	4	5
<b>1</b>	0	5	<u>4</u>	16	22
<b>2</b>	5	0	6	<u>15</u>	24
<b>3</b>	4	<u>6</u>	0	12	18
<b>4</b>	16	15	12	0	<u>9</u>
<b>5</b>	<u>22</u>	24	18	9	0

### METODA VÝHODNOSTNÍCH ČÍSEL (CLARK, WRIGHT)

1. Sestavíme matici výhodnostních čísel:  $s_{ij} = (C_{1i} + C_{i1} + C_{1j} + C_{j1}) - (C_{1i} + C_{ij} + C_{1j}) = C_{i1} + C_{1j} - C_{ij}$ , pro  $i, j = 2, 3, \dots, n, i \neq j$ .
2. Setřídíme čísla  $s_{ij}$  sestupně.
3. Podle setříděných čísel  $s_{ij}$ , postupně spojujeme uzly  $i$  a  $j$  tak, aby vznikl výsledný cyklus (čísla  $s_{ij}$ , která by spojením vrcholu  $i$  a  $j$  vytvořila cyklus, přeskočíme).

Výhodnostní čísla říkají, o kolik je výhodnější jet jako na obrázku B oproti způsobu na obrázku A.



A

B

Uvažujme stejný příklad jako výše.

- (1) Z matice nejkratších vzdáleností sestavíme matici výhodnostních čísel. Například prvek  $s_{23}$  spočítáme jako  $s_{23} = s_{21} + s_{13} - s_{23} = 5 + 9 - 6 = 3$ . Hledáme největší výhodnostní číslo, což je  $s_{45} = 29$ . Pojedeme tedy z uzlu 4 do uzlu 5.
- (2) Vyškrtneme  $s_{54}$  (aby nevznikl předčasně cyklus), 4. řádek (do jiného uzlu už ze 4. uzlu nepojedeme) a 5. sloupec (do 5. uzlu už z jiného uzlu nepojedeme). Pak hledáme největší výhodnostní číslo ve čtvrtém sloupci a pátém řádku. Chceme totiž prodloužit dosavadní trasu, takže musíme tedy buď vyjet z 5. uzlu nebo přijet do 4 uzlu.
- (3) Největší výhodnostní číslo je buď  $s_{53} = 8$ , nebo  $s_{34} = 8$ . Lepší bude vybrat  $s_{53}$ , protože po vyškrtnutí 5. řádku a 3. sloupce nám zůstane v matici větší použitelné výhodnostní číslo než po vyškrtnutí 3. řádku a 4. sloupce. Dosavadní trasa je tedy 4-5-3.
- (4) Vyškrtneme 5. řádek, 3. sloupec a  $s_{34}$ , aby nevznikl předčasně cyklus. Hledáme největší výhodnostní číslo ve třetím řádku a čtvrtém sloupci.
- (5) Tímto největším výhodnostním číslem  $s_{24}$ . Dosavadní trasa je tedy 2-4-5-3.
- (6) Trasu získanou v posledním kroku musíme spojit s výchozím bodem. Výsledná trasa bude 1-2-4-5-3-1 s délkou 51, což je optimální řešení.

(1)		1	2	3	4	5
1						
2				3	6	3
3			3		8	8
4			6	8		<b>29</b>
5			3	8	29	

(2)		1	2	3	4	5
1						
2				3	6	3
3			3		8	8
4			6	8		<b>29</b>
5			3	8	29	

(3)		1	2	3	4	5
1						
2				3	6	3
3			3		8	8
4			6	8		<b>29</b>
5			3	<b>8</b>	29	

(4)		1	2	3	4	5
1						
2				3	6	3
3			3		8	8
4			6	8		<b>29</b>
5			3	<b>8</b>	29	

(5)		1	2	3	4	5
1						
2				3	<b>6</b>	3
3			3		8	8
4			6	8		<b>29</b>
5			3	<b>8</b>	29	

(6)	C	1	2	3	4	5
1		0	<b>5</b>	4	16	22
2		5	0	6	<b>15</b>	24
3		<b>4</b>	6	0	12	18
4		16	15	12	0	<b>9</b>
5		22	24	<b>18</b>	9	0

## METODA VKLÁDACÍ

1. Najdeme nejvzdálenější uzel od výchozího místa  $s$ , označíme jej  $s$ . Vytvoříme uzavřenou cestu 1- $s$ -1.
2. Vložíme do této cesty některý ze zbylých uzlů tak, aby prodloužení bylo minimální.
3. To opakujeme, dokud nenajdeme Hamiltonův cyklus.

Uvažujme stejný příklad jako výše.

Nejvzdálenějším uzlem od prvního uzlu je uzel 5.

Nyní se musíme rozhodnout, který uzel vložit mezi ně. Zjistíme, o kolik by se tím prodloužila trasa.

$$\begin{aligned}(1,2),(2,5) & 5 + 24 - 22 = 7 \\ (1,3),(3,5) & 4 + 18 - 22 = 0 \\ (1,4),(4,5) & 16 + 9 - 22 = 3\end{aligned}$$

Nejmenší prodloužení je při vložení uzlu 3 mezi uzly 1 a 5. Trasa bude 1-3-5.

Dále musíme vložit některý ze zbylých uzlů do stávající trasy. Možnosti jsou následující.

$$\begin{aligned}(1,2),(2,3) & 5 + 6 - 4 = 7 \\ (1,4),(4,3) & 16 + 12 - 4 = 24 \\ (3,2),(2,5) & 6 + 24 - 18 = 12 \\ (3,4),(4,5) & 12 + 9 - 18 = 3 \\ (5,2),(2,1) & 24 + 5 - 22 = 7 \\ (5,4),(4,1) & 9 + 16 - 22 = 3\end{aligned}$$

Můžeme vložit uzel 4 mezi uzly 5 a 1. Trasa bude 1-3-5-4-1. Možnosti, kam vložit zbylý uzel 2, jsou tyto:

$$\begin{aligned}(1,2),(2,3) & 5 + 6 - 4 = 7 \\ (3,2),(2,5) & 6 + 24 - 18 = 12 \\ (5,2),(2,4) & 24 + 15 - 9 = 30 \\ (4,2),(2,1) & 15 + 5 - 16 = 4\end{aligned}$$

Vložíme jej mezi uzly 4 a 1. Výsledná trasa bude 1-3-5-4-2-1 s délkou 51.

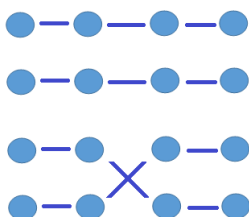
U velkých matic rychle roste počet míst, kam můžeme vložit další uzly.

## METODA VÝMĚN (LIN. KERNIGHEN, METODA 2X2)

Tato metoda zlepšuje již nalezené řešení.

1. Vynecháme dvě hrany, které nemají společný uzel.
2. Vyměníme tyto dvě hrany za jiné dvě tak, aby opět vznikl Hamiltonův cyklus.
3. Pokud takováto výměna přináší úsporu, necháme ji tam.

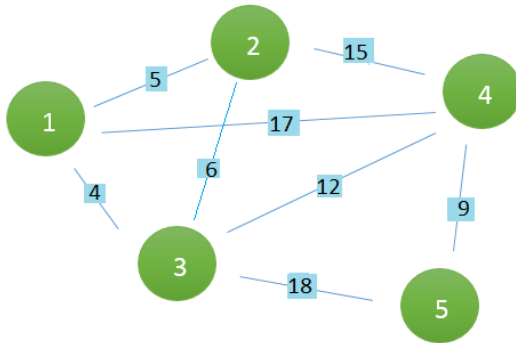
Příklad: zlepšuje tato výměna optimální řešení?



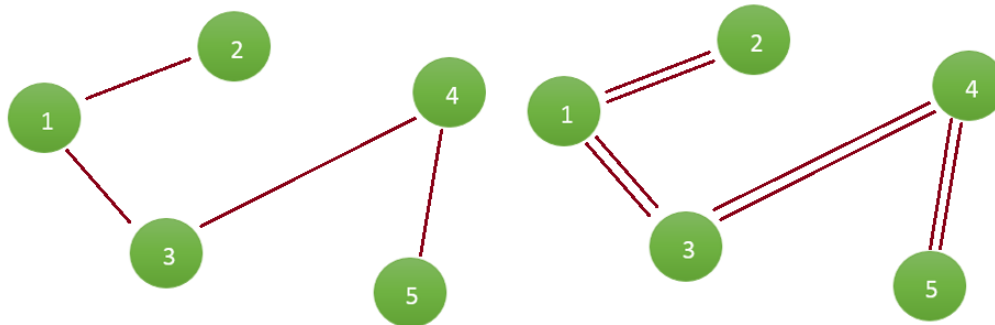
## METODA MINIMÁLNÍ KOSTRY

1. Najdeme minimální kostru grafu.
2. Zdvojíme hrany, čímž získáme Eulerův cyklus.
3. Transformujeme Eulerův cyklus na Hamiltonův.

Jde o polynomiální algoritmus. Uvažujme stejný příklad jako výše.



Najdeme minimální kostru. Tu transformujeme na Eulerův cyklus: 1-2-1-3-4-5-4-3-1.



Ten transformujeme na Hamiltonův cyklus tak, že vynecháme uzly, které již byly navštíveny. Výsledná trasa tedy bude 1-2-3-4-5-1. Její délka je 54.

## CHRISTOFIDOVA METODA

Tato metoda je podobná předcházející metodě minimální kostry, od níž se liší ve druhém kroku.

1. Najdeme minimální kostru grafu.
2. Uzly lichého stupně spojíme hranami metodou **perfektního párování s minimálními náklady**.
3. Získaný Eulerův cyklus transformujeme na Hamiltonův cyklus.

Ve druhém kroku se spojují hranou dvojice uzlů lichého stupně. Hledá se optimální párování v rámci jedné množiny, takže se řeší **obecný přiřazovací problém**.

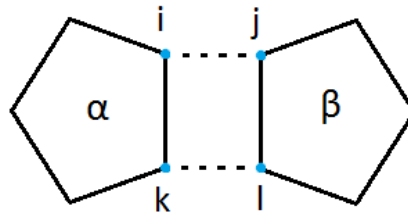
Našli bychom tedy minimální kostru jako na obrázku nahoře nalevo. Uzly lichého stupně jsou uzly 2 a 5. Kdyby jich bylo více, hledáme jejich optimální párování pomocí přiřazovacího problému, takhle je zřejmým řešením párování (2,5). Přidáme do grafu hranu (2,5), čímž vznikne Eulerův cyklus, který se pak transformuje na Hamiltonův (v tomto případě už bude zároveň Hamiltonovým cyklem). Délka trasy je 54.

## METODA ZATŘÍĐOVÁNÍ

1. Vytvoříme výchozí systém minicyklů.
2. Shlukujeme cykly na základě minimální vzdálenosti, dokud nevznikne Hamiltonův cyklus.

$$D_{\alpha,\beta} = \min_{\substack{i,k \in \alpha \\ j,l \in \beta}} (c_{ij} + c_{kl} - c_{ik} - c_{jl})$$

Příklad: máme minicykly alfa a beta. Zkusíme nahradit hrany (i,k) a (j,l) hranami (i,j) a (k,l). Takto zjišťujeme, jak nám sloučit cykly, aby byla vzdálenost co nejmenší.



## ZDROJE:

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

Burdová, J.: Heuristické a metaheuristické metody řešení úlohy obchodního cestujícího. Diplomová práce, VŠE, 2010. Dostupné z [http://www.vse.cz/vskp/23543\\_heuristicke\\_a%2%A0metaheuristicke\\_metody\\_reseni\\_ulohy\\_obcho\\_dniho\\_cestujiciho](http://www.vse.cz/vskp/23543_heuristicke_a%2%A0metaheuristicke_metody_reseni_ulohy_obcho_dniho_cestujiciho)