Katedra matematiky a deskriptívnej geometrie

Porovnanie algoritmov na riešenie úlohy obchodného cestujúceho

Úvod

Problém obchodného cestujúceho, známy pod anglickým názvom traveling salesman problem, patrí medzi najznámejšie optimalizačné problémy. Podstata problému spočíva v tom, že obchodník chce precestovať n daných miest tak, aby prešiel každým mestom len raz, a vrátil sa do mesta, z ktorého vychádzal. Chce precestovať všetky mestá čo najefektívnejšie, čo znamená, že chce ušetriť náklady na cestovanie a čas. Problém zaujal a inšpiroval mnohých svetových vedcov z rôznych oblastí, pre jeho jednoduchosť a zároveň pre potrebu efektívneho vyriešenia daného problému.

Na riešenie úlohy sa často používajú dva algoritmy, a to Littleho metóda vetiev a hraníc a Christofidesov aproximačný algoritmus. Cieľom práce bude porovnať výpočtovú zložitosť a výkonnosť oboch algoritmov na príkladoch cestných prepojení okresných miest na Slovensku.

Algoritmy a ich zložitosť

Časová zložitosť, teda počet krokov na vyriešenie problému je dôvod, prečo ešte stále nebol nájdený efektívny algoritmus, ktorý pracuje v polynomiálnom čase. V 60-tych rokoch minulého storočia si matematici uvedomili fundamentálnu dôležitosť 2 tried algoritmov, ktorými sú:

- trieda P tzv. polynomiálnych algoritmov
- trieda NP tzv. nedeterministických polynomiálnych algoritmov

Problém patrí do triedy P, ak existuje algoritmus, ktorý dokáže vyriešiť tento problém pri dĺžke vstupu n v čase nanajvýš p(n) (v nejakých časových jednotkách), kde p je polynóm v premennej n.

Problém patrí do triedy NP, ak správnosť ľubovoľného jeho riešenia pri dĺžke vstupu n možno overiť v čase nanajvýš p(n), kde p je opäť nejaký polynóm v premennej n. Je zrejmé, že ak problém patrí do triedy P, tak patrí aj do triedy NP.

Jedna z najväčších matematických otázok súčasnosti je či P=NP. Vyriešenie NP-úplnej úlohy, do ktorej spadá problém obchodného cestujúceho alebo naopak vyvrátenie by odpovedalo na túto otázku.

Vybrané druhy algoritmov na riešenie problému obchodného cestujúceho

Existujú rôzne druhy algoritmov, pretože tzv. **metóda hrubej sily** a teda nájdenie všetkých možností pomocou vzťahu pri symetrickom probléme $\frac{(n-1)!}{2}$, nie je aplikovateľná na veľký vstupný parameter *n*(počet miest).

Algoritmy sa delia na:

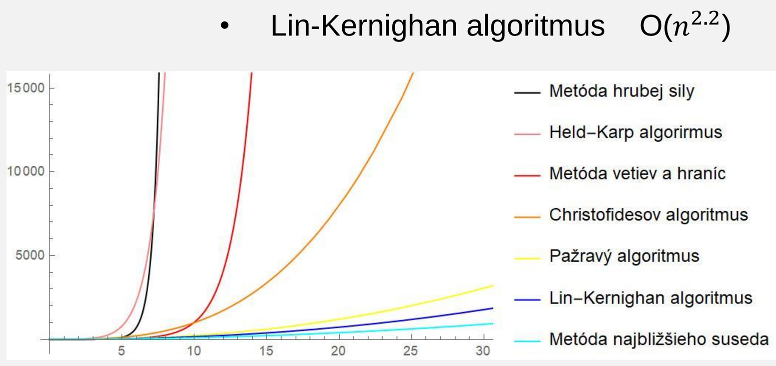
- **Presné** – výstup sa priblíži k optimálnemu riešeniu
- **Aproximačné** - umožňuje nájsť približné riešenie pre väčšiu množinu vstupných údajov v prijateľnom čase

Príklad presných algoritmov a ich časová náročnosť:

- Metóda hrubej sily O(n!)
- Bellman-Held-Karp algotitmus $O(n^2\, 2^n)$
- Metóda vetiev a hraníc $O(2^n)$

Príklad aproximačných algoritmov a ich časová náročnosť:

1. Konštrukčné- nájdu riešenie:
   - Christofidesov algoritmus $O(n^3)$
   - Pažravý algoritmus $O(n^2 \log n)$
   - Metóda najbližšieho suseda $O(n^2)$
2. Zlepšovacie- zoptimalizujú už nájdené riešenie:
   - Lin-Kernighan algoritmus $O(n^{2.2})$



Metóda vetiev a hraníc

Metóda vetiev a hraníc patrí k presným algoritmom na riešenie problému obchodného cestujúceho. Tento algoritmus navrhli v roku 1960 matematičky Ailsa Landová a Alison Doigová, pričom pod názvom Metóda vetiev a hraníc (angl. branch and bound) sa prvýkrát objavil v práci matematika Johna Littla. Táto metóda je založená na redukcii pôvodnej úlohy na stále menšie podúlohy pomocou vetvenia, čím nakoniec získame optimálnu trasu pre obchodného cestujúceho. Vstup tvorí matica vzdialeností $C = c_{i,j}$, ktorá môže byť symetrická, ale aj nesymetrická, čo znamená, že vzdialenosť medzi mestami nie je rovná spiatočnej vzdialenosti medzi danými mestami. V tejto práci počítame len so symetrickou maticou, preto samotný popis algoritmu bude ďalej zúžený len na tento prípad. Treba zadefinovať niektoré označenia, ktoré budeme ďalej pri tejto metóde využívať. Matica vzdialeností $C = c_{i,j}$ s veľkosťou *n x n*, je matica obsahujúca hodnoty c_{ij} reprezentujúce napríklad vzdialenosti jednotlivých miest, kde i={1,...,n} sú riadky matice a j={1,...,n} sú stĺpce matice. Symbolom *T* budeme označovať cestu tvorenú množinou usporiadaných dvojíc vrcholov *T*={(i_1 i_2),(i_2 , i_3),...,(i_n , i_1)}. Ohodnotenie cesty *T* si označíme symbolom *h(T),* a získame ho ako súčet hodnôt $c_{i,j}$ matice $C = c_{i,j}$,teda vzdialenosti medzi jednotlivými trasami.

Princíp algoritmu:

1. Spravíme riadkovú a stĺpcovú redukciu matice C výberom najmenšej hodnoty α_i kde i={1,..n} a β_j kde j={1,..n} .V každom riadku/ stĺpci následne odčítame tieto hodnoty a získame redukovanú maticu $\tilde{C} = \tilde{c}_{i,j}$
2. Určíme spodnú hranicu ohodnotenia najlacnejšej cesty spočítaním α_i a β_j , ktorými sme v predchádzajúcom kroku redukovali maticu C
$$h(\tilde{T}) \geq \sum_{i=1}^{n} \alpha_i + \sum_{j=1}^{n} \beta_j$$
3. Každej 0 nachádzajúcej sa v redukovaniej matici $\tilde{C} = \tilde{c}_{i,j}$ na pozícii (i,j) priradíme hodnotu $\delta_{i,j}$, ktorá je tvorená súčtom minimálnej hodnoty $\tilde{c}_i$ v i-tom riadku a minimálnej hodnoty $\tilde{c}_j$ v j-tom stĺpci
$$\delta_{i,j} = \tilde{c}_i + \tilde{c}_j$$
4. Vyberieme maximálnu hodnotu , pre na konkrétnej pozícii(r,s). Pozícia maximálnej hodnoty znamená konkrétne prepojenie dvoch vrcholov r→s
5. Odstránime r-tý riadok a s-tý stĺpec a získavame o stupeň menšiu maticu. Skontrolujeme, či sa v každom riadku a stĺpci nachádza „-“ práve raz. Ak nie doplnímeho na požadovanú pozíciu a zabránime tvoreniu slučiek
6. Opakujeme pokiaľ nemáme maticu veľkosti 1x1

Ilustračný príklad metódy vetiev a hraníc

	A	B	C	D	E	F
A	-	1	2	6	7	6
B	1	-	9	5	2	8
C	2	9	-	2	2	4
D	6	5	2	-	5	3
E	7	2	2	5	-	3
F	6	8	4	3	3	-

Vstupná matica vzdialeností

	A	B	C	D	E	F	
A	-	0	1	5	6	4	1
B	0	-	8	4	1	6	1
C	0	7	-	0	0	1	2
D	4	3	0	-	3	0	2
E	5	0	0	3	-	0	2
F	3	5	1	0	0	-	3
	0	0	0	0	0	1	**12**

Redukovanie matice po stĺpcoch a riadkoch

	A	B	C	D	E	F
A	-	0^1	1	5	6	4
B	0^1	-	8	4	1	6
C	0^0	7	-	0^0	0^0	1
D	4	3	0^0	-	3	0^0
E	5	0^0	0^0	3	-	0^0
F	3	5	1	0^0	0^0	-

Priradenie hodnoty $\delta_{i,j}$

	A	C	D	E	F	
B	-	8	4	1	6	-1
C	0	-	0	0	1	
D	4	0	-	3	0	
E	5	0	3	-	0	
F	3	1	0	0	-	

Nová matica po odstránení vybraného riadku a stĺpcu

	A	C	D	E	F
B	-	7	3	0^3	5
C	0^3	-	0^0	0^0	1
D	4	0^0	-	3	0^0
E	5	0^0	3	-	0^0
F	3	1	0^0	0^0	-

	A	C	D	F
C	0^3	-	0^0	1
D	4	0^0	-	0^0
E	-	0^0	3	0^0
F	3	1	0^1	-

	C	D	F
D	0^0	-	0^0
E	-	3	0^0
F	1	0^4	-

	C	F
D	0^0	-
E	-	0^0

Opakovanie krokov pokiaľ nezískame finálnu cestu, ktorou je v tomto prípade C→A→B→E→F→D→C

Christofidesov algoritmus

Autorom je matematik Nicos Christofides, ale procedúra je známa aj pod názvom Christofides–Serdyukov algoritmus, pretože nezávisle od jeho práce ho objavil v roku 1976 aj A.I.Serdyukov. Tento algoritmus patrí k najlepším polynomiálnym aproximačným algoritmom, čo je potvrdené príslušnou vedeckou komunitou zameranou na problém obchodného cestujúceho. Časová náročnosť tohoto algoritmu je $O(n^3)$ s aproximáciou $\frac{3}{2}$. V roku 2020 však matematici Karlin, Klein, a Gharan vydali prácu, v ktorej popísali nový aproximačný algoritmus s údajne lepšou aproximáciou $\frac{3}{2} - 10^{-36}$. Tento algoritmus je odvodený z Christofidesovho algoritmu, len s menšími odmenami.

V literatúre sa obvykle predpokladá, že Christofidesov algoritmus má na vstupe kompletný ohodnotený graf *G=(V,E,c),* kde funkcia ohodnotenia c nadobúda kladné hodnoty, ktoré navyše spĺňajú trojuholníkovú nerovnosť. Tá vyžaduje, aby pre každé tri rôzne vrcholy *u,v,w* ∈ *V* platila nerovnosť *c(u,v)+c(v,w)≥ c(u,w).*

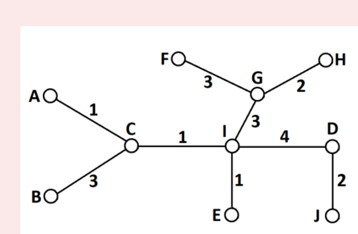
Princíp algoritmu:

1. Nájdeme v G **najlacnejšiu kostru** *T.*
2. Identifikujeme množinu vrcholov *W* **nepárneho stupňa** v *T.*
3. Nájdeme **najlacnejšie úplné párovanie** *M* v podgrafe grafu *G* indukovanom množinou *W.*
4. Hrany z párovania *M* pridáme do kostry *T*, čím získame podgraf *H* grafu *G*, ktorý má všetky vrcholy párnych stupňov.
5. V podgrafe *H* nájdeme **Eulerovský ťah**.
6. Nájdený Eulerovský ťah modifikujeme na výslednú **Hamiltonovskú kružnicu.**

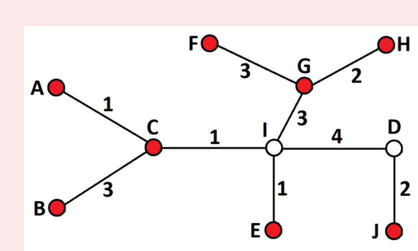
Ilustračný príklad Christofidesovho algoritmu

Vstupná matica vzdialeností

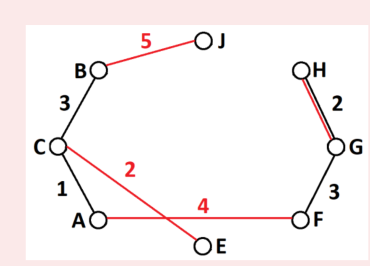
	A	B	C	D	E	F	G	H	I	J
A	-	5	1	6	7	4	6	7	5	5
B		-	3	5	7	5	5	5	8	5
C			-	6	2	8	7	6	1	4
D				-	8	7	6	5	4	2
E					-	6	5	8	1	6
F						-	3	7	9	7
G							-	2	3	5
H								-	7	9
I									-	8
J										-



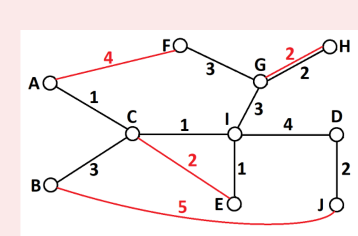
Nájdená najlacnejšia kostra



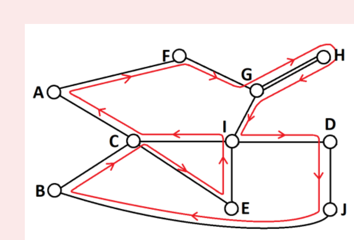
Nepárne vrcholy nájdenej kostry



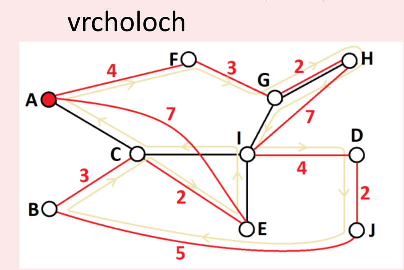
Párovanie na nepárnych vrcholoch



Pridané párovanie do kostry



Nájdený Eulerov ťah

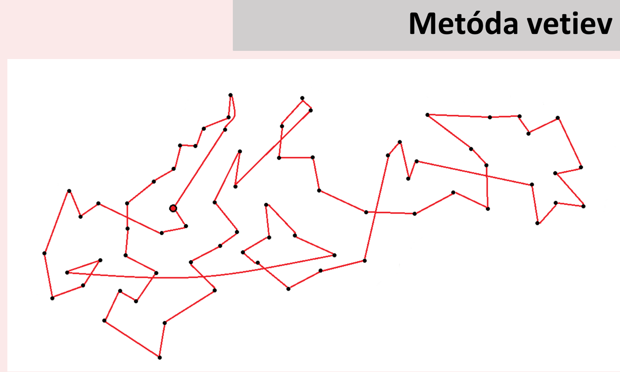


Hamiltonovská kružnica a finálna trasa

Aplikovanie algoritmov

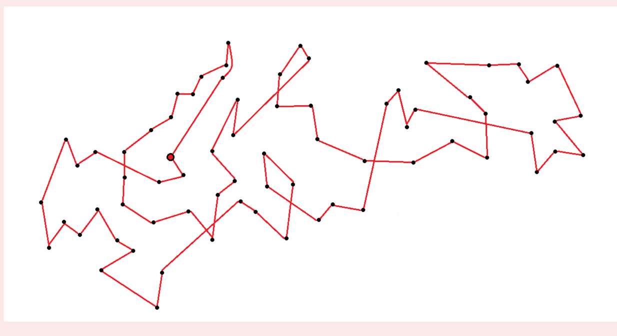
Zvolené algoritmy boli aplikované na príklad cestného prepojenia okresných miest na Slovensku. Aplikovaním metódy vetiev a hraníc sme získali viacero výsledkov, všetky s vychádzajúcim mestom Bánovce nad Bebravou. Prepojenia ktoré nám algoritmus vyhodnotil sú zobrazené na nasledujúcich obrázkoch. Druhý algoritmus nám dal 2 konkrétne výsledky, z ktorých jeden výsledok je s počiatočným mestom Bánovce nad Bebravou a druhým počiatočným mestom je Skalica.

Metóda vetiev a hraníc



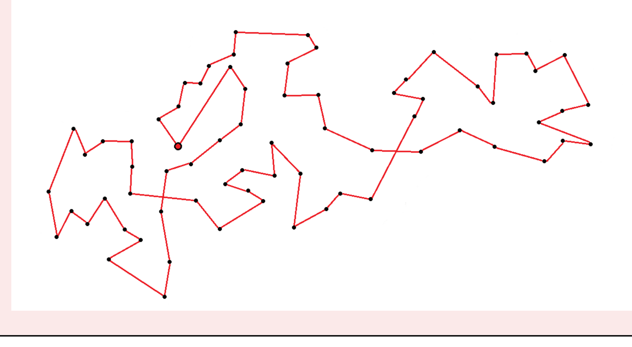
BN → PE → TO → MY → SE → SI → MA → BA → SC → TT → PK → PT → DT → BB → ZV → BS → KA → VK → LC → RS → PP → KK → SN → LE → VT → TV → MI → SO → HE → SV → ML → SP → SK → BJ → SL → SB → PO → KE → GL → RV → RA → BR → LM → RK → DK → NO → TS → TR → MT → PD → ZH → ZC → ZM → LV → NZ → KN → DS → GA → SA → NR → HC → PN → NM → TN → IL → PU → PB → BY → KM → CA → ZA → BN

Výsledná trasa s celkovým ohodnotením 2701 km a spodnou hranicou 2616km



BN → PE → TO → MY → SE → SI → MA → BA → PK → SC → TT → GA → SA → DS → KN → NZ → BS → KA → VK → DT → BB → ZV → LC → PT → RS → PP → KK → SN → LE → VT → TV → MI → SO → HE → SV → ML → SP → SK → BJ → SL → SB → PO → KE → GL → RV → RA → BR → LM → RK → DK → NO → TS → TR → MT → PD → ZH → ZC → LV → ZM → NR → HC → PN → NM → TN → IL → PU → PB → BY → KM → CA → ZA → BN

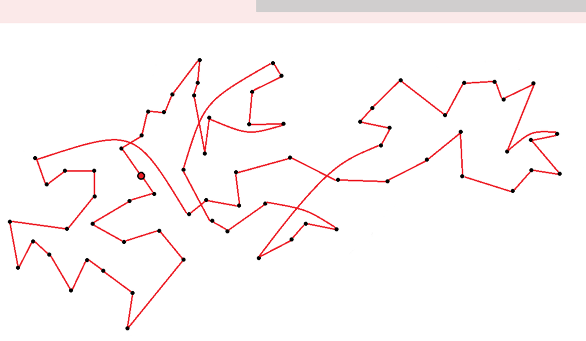
Výsledná trasa s celkovým ohodnotením 2534 km a spodnou hranicou 2449km



BN → TN → IL → PU → PB → BY → KM → CA → NO → TS → DK → RK → LM → BR → RA → RV → GL → KE → TV → MI → SO → VT → HE → SV → ML → SP → SK → BJ → PO → SB → SL → KK → PP → LE → SN → RS → PT → LC → VK → DT → BB → ZV → ZH → ZC → BS → KA → LV → ZM → HC → PN → NM → MY → SE → SI → MA → BA → PK → SC → TT → GA → SA → DS → KN → NZ → NR → TO → PE → PD → TR → MT → ZA → BN

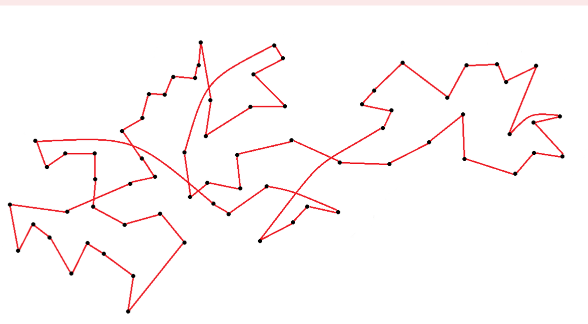
Výsledná trasa s celkovým ohodnotením 2339km a spodnou hranicou 2254km

Christofidesov algoritmus



BN → PE → TO → HC → NR → ZM → LV → KN → NZ → SA → GA → DS → SC → PK → BA → MA → TT → PN → NM → MY → SE → SI → ZC → ZH → ZV → BB → BR → RA → RV → GL → PO → KE → TV → MI → SO → HE → SV → VT → ML → SP → SK → BJ → SB → SL → KK → PP → LE → SN → VK → LC → PT → RS → DT → KA → BS → PD → NO → TS → DK → RK → LM → MT → TR → ZA → KM → CA → BY → PB → PU → IL → TN → BN

Výsledná trasa s celkovým ohodnotením 2662km



SI → SE → MY → NM → PN → HC → NR → ZM → LV → KN → NZ → SA → GA → DS → SC → PK → BA → MA → TT → TO → PE → BN → TN → IL → PU → PB → BY → ZA → KM → CA → MT → TR → RK → LM → DK → TS → NO → PD → ZC → ZH → ZV → BB → BR → RA → RV → GL → PO → KE → TV → MI → SO → HE → SV → VT → ML → SP → SK → BJ → SB → SL → KK → PP → LE → SN → VK → LC → PT → RS → DT → KA → BS → SI

Výsledná trasa s celkovým ohodnotením 2525km

Záver

Problém obchodného cestujúceho napriek jednoduchosti zadania úlohy patrí medzi NP-úplné úlohy, pretože neexistuje algoritmus, ktorý by za prijateľný čas vedel nájsť zaručene najlepší výsledok. Na základe výstupov na náš zadaný problém sme mohli vidieť že príklady získané metódou vetiev a hraníc sú lepšie ako výsledky získané Christofidesovým algoritmom. Na základe spodných hraníc pri metóde vetiev a hraníc si však môžeme všimnúť že aj keď metóda patrí k presným algoritmom stále je rozdiel medzi nájdenou trasou a spodnou hranicou čo značí priestor na zlepšenie. Časová náročnosť oboch algoritmov bola približne rovnaká, pretože vstupný počet vrcholov nebol príliš veľký avšak rozdiel medzi metódou vetiev a hraní a Christofidesovým algoritmom bola ≈0,06s. Vieme preto povedať, že je lepšie aplikovať aproximačný algoritmus.

Pedagóg: doc. Ing. Alexandra Šipošová, PhD.
Poslucháč: Gabriela Cvečková

Akademický rok: 2020/2021
Študijný program: Matematicko-počítačové modelovanie

BAKALÁRSKA PRÁCA
Porovnanie algoritmov na riešenie úlohy obchodného cestujúceho