

Žilinská univerzita v Žiline
Fakulta riadenia a informatiky

**Navrhovanie verejných obslužných systémov
s exaktným optimalizačným jadrom**

Dizertačná práca

Kód(28360020163003)

Študijný program: Aplikovaná informatika
Študijný odbor: 9.2.9 Aplikovaná informatika
Pracovisko: Katedra matematických metód a operačnej analýzy
Fakulta riadenia a informatiky, Žilinská univerzita v Žiline
Školiteľ: Prof. RNDr. Jaroslav Janáček, CSc.

Žilina, 2016

Ing. Ján Bendík

Vyhlásenie

Prehlasujem, že som dizertačnú prácu spracoval samostatne pod odborným dohľadom školiteľa a uviedol som všetky použité pramene, z ktorých som čerpal.

V Žiline,

dňa 22. apríla 2016

Podpis:.....

Pod'akovanie

Chcem sa úprimne poďakovať svojmu školiteľovi
prof. RNDr. Jaroslavovi Janáčkovi, CSc.
za jeho vedenie, neoceniteľnú pomoc, cenné rady, návrhy, pripomienky
a drahocenný čas, ktorý mi venoval pri vypracovaní tejto práce.

Za pomoc a cenné rady pri štatistických testoch sa chcem úprimne poďakovať
RNDr. Ide Stankovianskej, CSc.

Moja vďaka patrí kolegom z Katedry matematických metód a operačnej analýzy
za zaujímavé myšlienky, postrehy, podnety a cenné odborné diskusie.

Napokon sa chcem úprimne poďakovať mojej rodine a priateľom
za ich neustálu podporu, trpezlivosť a pochopenie.

Abstrakt

BENDÍK, JÁN: *Navrhovanie verejných obslužných systémov s exaktným optimalizačným jadrom* [dizertačná práca] - Žilinská univerzita v Žiline. Fakulta riadenia a informatiky; Katedra matematických metód a operačnej analýzy. - Školiteľ: Prof. RNDr. Jaroslav Janáček, CSc. - Stupeň odbornej kvalifikácie: Doktor filozofie v študijnom odbore 9.2.9 Aplikovaná informatika. Žilina: FRI ŽU v Žiline, 2016.

Cieľom tejto dizertačnej práce je výskum v oblasti aplikovanej informatiky zameraný na navrhovanie verejných obslužných systémov s exaktným optimalizačným jadrom. Návrh verejného obslužného systému predstavuje ťažko riešiteľný kombinačný problém spočívajúci v riešení umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk. Práca obsahuje niektoré súčasné prístupy k riešeniu úlohy návrhu verejného obslužného systému so špeciálnym zameraním na Erlenkotterov prístup k riešeniu kapacitne neobmedzenej umiestňovacej úlohy. Erlenkotterov prístup je založený na poznatkoch z teórie duality a využití duálneho riešenia k získaniu dolnej a hornej hranice riešenia. V predkladanej dizertačnej práci prezentujem dva prístupy k riešeniu úlohy návrhu verejného obslužného systému s využitím Erlenkotterovho prístupu. Semi-exaktný iteratívny prístup je založený na transformácii úlohy s obmedzeným počtom vybudovaných stredísk na kapacitne neobmedzenú umiestňovaciu úlohu pomocou Lagrangeovej relaxácie p-mediánovej podmienky. Zovšeobecnený exaktný prístup je založený na zovšeobecnení Erlenkotterovho prístupu na riešenie úlohy s obmedzeným počtom vybudovaných stredísk. Navrhnuté prístupy sú porovnané z hľadiska náročnosti na výpočtový čas a presnosti nájdeného riešenia.

Kľúčové slová: návrh verejného obslužného systému, umiestňovacia úloha, Erlenkotterov prístup, Lagrangeova relaxácia, metóda vetiev a hraníc, zovšeobecnený exaktný prístup, semi-exaktný iteratívny prístup, kompozičný prístup

Abstract

BENDÍK, JÁN: Designing the public service systems with an exact optimization core [dissertation thesis] - University of Žilina. Faculty of Management Science and Informatics; Department of Mathematical Methods and Operations Research. - Supervisor: Prof. RNDr. Jaroslav Janáček, CSc. - Qualification level: Philosophiae doctor in the study field 9.2.9 Applied Informatics. Žilina: FRI ŽU in Žilina, 2016.

The target of this dissertation thesis is to do research in field of the applied informatics based on designing the public service systems with an exact optimization core. Designing the public service system represents NP-hard problem consisting in solving the p-median location problem. The thesis contains some of the current approaches to the solving the public service system design with the special fixation on the Erlenkotter approach to the solving uncapacitated facility location problem. Erlenkotter approach is based on knowledges from the theory of duality and using dual solution to obtaining the lower and upper bound of solution. I present two approaches to solving the p-median location problem with using Erlenkotter approach in the presented dissertation thesis. Semi-exact iterative approach is based on the transformation of the p-median location problem to the uncapacitated facility location problem by Lagrangean relaxation of the p-median condition. Generalized exact approach is based on the generalization of Erlenkotter approach to the solving the p-median location problem. The proposed approaches are compared in terms of demands on the computational time and accuracy of the obtained solution.

Keywords: Public service system design, location problem, Erlenkotter approach, Lagrangean relaxation, branch and bound method, generalized exact approach, semi-exact iterative approach, compositional approach.

Obsah

Zoznam použitých skratiek a označení	8
1 Úvod do problematiky	9
2 Súčasný stav riešenej problematiky	11
2.1 Umiestňovacia úloha a jej aplikácie.....	11
2.1.1 Matematická formulácia umiestňovacej úlohy	11
2.1.2 Aplikácie umiestňovacej úlohy.....	12
2.2 Prehľad exaktných a semi-exaktných metód riešenia	20
2.2.1 Erlenkotterov prístup ku kapacitne neobmedzenej umiestňovacej úlohe	21
2.2.2 Vylepšenie Erlenkotterovho prístupu	33
2.2.3 Erlenkotterov prístup pre riešenie p-mediánu s Lagrangeovou relaxáciou ...	35
2.2.4 ZEBRA	38
2.2.5 Radiálny prístup s deliacimi bodmi	43
2.2.6 Lagrangeová heuristika pre riešenie úlohy váženého p-mediánu.....	48
3 Ciele a metodika práce.....	52
3.1 Ciele práce.....	52
3.2 Metodika práce.....	54
4 Výsledky a ich zhodnotenie	58
4.1 Zlepšenie riešenia získaného iteratívnym prístupom	58
4.1.1 Heuristické vylepšenia riešenia	59
4.1.2 Obmedzenie vetvenia v metóde vetiev a hraníc	62
4.1.3 Predbežné experimenty.....	63
4.1.4 Celkové zhodnotenie experimentov.....	67
4.2 Vylepšenie Erlenkotterovho prístupu s Lagrangeovou relaxáciou	68
4.2.1 Odhad nákladov distribučného systému	69
4.2.2 Iteratívny prístup s odhadom Lagrangeovho multiplikátora.....	71
4.2.3 Porovnanie navrhnutých iteratívnych algoritmov.....	76

4.3	Zovšeobecnený Erlenkotterov prístup.....	80
4.3.1	Matematický model úlohy návrhu verejného obslužného systému	80
4.3.2	Duálny model LP-relaxácie úlohy návrhu verejného obslužného systému ...	83
4.3.3	Výpočet dolnej hranice účelovej funkcie.....	85
4.3.4	Výpočet hornej hranice riešenia	87
4.3.5	Schéma prehľadávania stromu riešení v metóde vetiev a hraníc.....	90
4.3.6	Spôsob vetvenia exaktného algoritmu pMedBBDual.....	91
4.3.7	Úplná fixácia v metóde vetiev a hraníc	92
4.3.8	Predbežné experimenty	94
4.4	Vylepšenia exaktného optimalizačného jadra	95
4.4.1	Vylepšenie výpočtu dolnej hranice	97
4.4.2	Vplyv fixácie na získanie dobrej dolnej hranice.....	99
4.4.3	Výber kandidáta na vetvenie.....	100
4.4.4	Vylepšenie hornej hranice	104
4.4.5	Porovnanie algoritmov so zovšeobecneným Erlenkotterovým prístupom ..	109
4.5	Porovnanie iteratívneho a zovšeobecneného prístupu	112
4.6	Kompozičný prístup k riešeniu úlohy návrhu VOS	115
4.7	Zhrnutie	117
5	Záver	119
	Zoznam použitej literatúry.....	122
	Zoznam obrázkov a grafov	127
	Zoznam tabuliek	128
	Zoznam príloh.....	130

Zoznam použitých skratiek a označení

Benchmark	- Testovací příklad
BBDual	- Exaktný algoritmus na riešenie kapacitne neobmedzenej umiestňovacej úlohy
CDA	- Correction Dual Ascent (korekčný algoritmus duálneho vzrastu)
DA	- Dual Ascent (Algoritmus duálneho vzrastu)
DAD	- Dual Adjustment (Algoritmus duálnych úprav)
DH	- Dolná hranica
ES	- Exact solution (exaktné riešenie)
ICDA	- Increment Correction Dual Ascent (inkrementačný korekčný duálny algoritmus)
IP	- Integer programming (celočíselné programovanie)
XPRESS-IVE	- univerzálny IP solver
HH	- Horná hranica
LP	- Linear programming (Lineárne programovanie)
PC	- Personal computer (osobný počítač)
pMBBDual	- Semi-exaktný iteratívny algoritmus na riešenie úlohy návrhu verejného obslužného systému
pMedBBDual	- Zovšeobecnený exaktný algoritmus na riešenie úlohy návrhu verejného obslužného systému
RAM	- Random access memory (operačná pamäť)
SPR	- Systém pre podporu rozhodovania
SR	- Slovenská republika
UFLP	- Uncapacitated facility location problem (kapacitne neobmedzená umiestňovacia úloha)
VOS	- Verejný obslužný systém
ZEBRA	- Z-Erlange and BRanch Algoritmus

1 Úvod do problematiky

V reálnych situáciách sa stretávame s rôznymi obslužnými systémami, ktoré sa stali súčasťou nášho života. Podľa [27] sa obslužný systém chápe ako sústava na seba naväzujúcich dopravných, výrobných, manipulačných aj informačných činností slúžiacich na zabezpečenie potrieb zákazníkov v danom mieste a čase zo zdrojov obsluhy, pomocou ktorých možno ich potreby uspokojiť. Obslužné systémy môžeme rozdeliť na súkromné obslužné systémy a verejné obslužné systémy. Pri verejných obslužných systémoch platí, že služba musí byť dostupná pre všetkých zákazníkov. Pri súkromných obslužných systémoch je hlavným cieľom maximalizácia zisku majiteľa systému. To môže viesť k minimalizácii nákladov na uspokojenie požiadaviek časti zákazníkov, teda tých, z ktorých je nejaký zisk. Potreba návrhu rôznych obslužných systémov vyplýva zo vzniku nových potrieb alebo zo zmien vonkajších podmienok. V návrhu obslužného systému vystupuje množina zákazníkov a množina kandidátov na umiestnenie stredísk. Vyberajú sa strediská z množiny kandidátov a priradzujú sa im zákazníci, ktorí sú zväčša rozptýlení v rámci obsluhovaného územia. Obsluha zákazníka sa vždy uskutočňuje z najbližšieho strediska. Najčastejšou úlohou návrhu verejného obslužného systému je určenie optimálneho rozmiestnenia stredísk. Určenie optimálneho rozmiestnenia stredísk je rozsiahla kombinatorická úloha s využitím poznatkov z matematického programovania, ktorá patrí do kategórie NP-ťažkých úloh [25], [26]. V úlohe návrhu verejného obslužného systému je možné kvalitu riešenia vyhodnocovať pomocou rôznych kritérií. Systémové kritérium zohľadňuje minimalizáciu súčtu nákladov spojených s vybudovaním stredísk a obsluhou zákazníkov. Minimaxové kritérium zohľadňuje polohu najhoršieho zákazníka k najbližšiemu stredisku. Kritérium solidárnosti zohľadňuje férovosť zabezpečenia obsluhy zákazníka. V praxi to znamená, že ak nie je služba dostupná z najbližšieho umiestneného strediska, bude služba dostupná z druhého najbližšieho umiestneného strediska obsluhy. Návrh verejného obslužného systému môže obsahovať rôzne obmedzenia. Tieto obmedzenia závisia od špecifikácie obslužného systému. Z ekonomického alebo technologického hľadiska môže týmto obmedzením byť napríklad obmedzenie počtu vybudovaných stredísk obsluhy. Toto obmedzenie môžeme považovať za kapacitné obmedzenie, kedy nám finančné prostriedky neumožňujú vybudovať viac stredísk.

Môj výskum sa bude zaoberať metódami návrhov verejného obslužného systému s využitím informatických prostriedkov. Tento návrh verejného obslužného systému spočíva v riešení umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk.

Mojim vedeckým cieľom spojeným s riešením spomenutej úlohy je zistiť do akej miery je možné a výhodné využiť k riešeniu Erlenkotterov prístup, ktorý bol v minulosti úspešne použitý na riešenie kapacitne neobmedzenej umiestňovacej úlohy. Tento prístup realizujem pomocou metódy vetiev a hraníc so špeciálnym prístupom k výpočtu dolnej hranice. Budem sa snažiť zovšeobecniť Erlenkotterov prístup na prípad umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk, analyzovať možné prístupy k riešeniu príslušnej umiestňovacej úlohy, navrhnúť účinné algoritmy pre ich realizáciu, implementovať ich a vykonať výskum ich správania.

2 Súčasný stav riešenej problematiky

2.1 Umiestňovacia úloha a jej aplikácie

2.1.1 Matematická formulácia umiestňovacej úlohy

Na danom území je potrebné optimálne rozmiestniť strediská obsluhy pre obsluhu zákazníkov v uzloch j z množiny J , pričom tieto strediská obsluhy je možné umiestniť iba v miestach i z množiny I . Aj jedno stredisko umiestnené v ktoromkoľvek uzle z množiny I je schopné obslúžiť všetkých zákazníkov. Úloha spočíva v minimalizácii celkových nákladov, ktoré zahŕňajú fixné náklady f_i spojené s umiestnením strediska v mieste i a náklady c_{ij} na uspokojenie požiadaviek zákazníka j z miesta i . Podmienkou je, aby každý zákazník bol priradený niektorému z vybudovaných stredísk, z ktorého bude obslúžený. Počet vybudovaných stredísk je maximálne p .

Takto definovaná úloha je umiestňovacou úlohou s obmedzeným počtom vybudovaných stredísk. V danej úlohe predstavuje I množinu kandidátov, kde je možné umiestniť stredisko a J množinu všetkých zákazníkov. Pracuje sa tu s dvomi druhmi rozhodnutí. Rozhodnutia o umiestnení resp. neumiestnení strediska v mieste i z množiny I , ktoré sú v modeli modelované sadou bivalentných premenných y_i , ktoré nadobúdajú hodnotu y_i rovnú jednej, ak umiestnime stredisko v mieste i z množiny I . Premenné y_i sú rovné nule, ak neumiestnime stredisko v mieste i z množiny I . Rozhodnutia o priradení resp. nepriradení zákazníkov j z množiny J k stredisku i z množiny I sú modelované sadou premenných z_{ij} , kde z_{ij} sú rovné jednej, ak priradíme zákazníka $j \in J$ k stredisku i z množiny I . Premenné z_{ij} sú rovné nule, ak nepriradíme zákazníka j z množiny J k stredisku i z množiny I . Náklady spojené s uspokojením požiadavky zákazníka j z množiny J z miesta i z množiny I sú modelované konštantami c_{ij} . Konštanty f_i predstavujú fixné náklady na udržanie obslužného strediska v mieste i z množiny I . Konštantu p udáva horné ohraničenie počtu vybudovaných stredísk. Matematický model umiestňovacej úlohy s obmedzeným počtom umiestnených stredísk vyzerá potom nasledovne (2.1.01) – (2.1.06):

$$\text{Minimalizujte} \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \quad (2.1.01)$$

$$\text{Za podmienok: } \sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.1.02)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.1.03)$$

$$\sum_{i \in I} y_i \leq p \quad (2.1.04)$$

$$y_i \in \{0,1\} \quad \text{pre } i \in I \quad (2.1.05)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I, \quad j \in J \quad (2.1.06)$$

Účelová funkcia (2.1.01) predstavuje celkové náklady uspokojenia požiadaviek zákazníkov a vybudovania stredísk, ktoré minimalizujeme. Podmienky (2.1.02) zabezpečujú, aby každý zákazník j z množiny J bol priradený k práve jednému vybudovanému stredisku i z množiny I . Podmienky (2.1.03) zabezpečujú, aby nebolo možné priradiť zákazníka j z množiny J miestu (kandidátovi na umiestnenie strediska), kde stredisko nie je vybudované. Vybudovanie maximálne p stredísk zabezpečuje podmienka (2.1.04). Podmienky (2.1.05) a (2.1.06) určujú obor hodnôt rozhodovacích premenných.

2.1.2 Aplikácie umiestňovacej úlohy

Umiestňovaciú úlohu s obmedzeným počtom vybudovaných stredísk môžeme označiť ako kombináciu kapacitne neobmedzenej umiestňovacej úlohy a úlohy p -mediánu. V nižšie uvedených úlohách bude obslužný systém pozostávať z množiny J zákazníkov a z množiny I možných umiestnení stredísk. Zákazníci aj kandidáti na stredisko obsluhy sú nachádzajú v uzloch, ktoré sú rozmiestnené na danom území. Obsluhované územie má svoju infraštruktúru, kde uzly spolu s úsekmi spájajúcimi tieto uzly tvoria dopravnú sieť obsluhovanej oblasti. Rozhodnutia, ktoré je treba urobiť pre získanie riešenia daných úloh sú nasledovné:

- rozhodnutie o umiestnení alebo neumiestnení strediska v mieste i z množiny I
- rozhodnutie o priradení alebo nepriradení zákazníka j z množiny J k stredisku i z množiny I

Vo všetkých nižšie uvedených úlohách v tejto kapitole budú rozhodnutia o umiestnení resp. neumiestnení stredísk obsluhy v mieste i z množiny I modelované sadou bivalentných premenných y_i , ktoré nadobúdajú hodnotu y_i rovnú jednej, ak umiestnime stredisko v mieste i a hodnotu y_i rovnú nule, ak neumiestnime stredisko v mieste i . Rozhodnutia o priradení resp. nepriradení zákazníkov j k stredisku i budú modelované sadou bivalentných premenných z_{ij} , kde hodnoty z_{ij} sú rovné jednej, ak priradíme zákazníka j k stredisku i . V opačnom prípade sú hodnoty z_{ij} nulové.

Úloha p-mediánu

Riešme umiestňovaciu úlohu s obmedzeným počtom vybudovaných stredísk s nasledovnými údajmi. Fixné náklady f_i sú rovné nule pre všetky i z množiny I v účelovej funkcii (2.1.01), teda o nich neuvažujeme. Konštanty c_{ij} sú rovné d_{ij} a udávajú vzdialenosť zákazníka j od možného strediska i . Takto formulovaná úloha je úloha p-mediánu, ktorá je najčastejšou úlohou navrhovania štruktúry verejných obslužných systémov.

Úlohu p-mediánu [44] je možné všeobecne definovať ako výber najviac p stredísk z množiny I kandidátov na umiestnenie strediska tak, aby celkové náklady na obsluhu zákazníkov boli čo najnižšie. Náklady na poskytovanie služby zákazníkovi j zo strediska i bývajú priamo úmerné vzdialenosti d_{ij} , teda minimalizuje sa súčet vzdialeností medzi zákazníkmi a ich obsluhujúcimi strediskami. Matematický model úlohy p-mediánu je potom formulovaný nasledovne (2.1.07) – (2.1.12):

$$\text{Minimalizujte} \quad \sum_{i \in I} \sum_{j \in J} d_{ij} z_{ij} \quad (2.1.07)$$

$$\text{Za podmienok:} \quad \sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.1.08)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.1.09)$$

$$\sum_{i \in I} y_i \leq p \quad (2.1.10)$$

$$y_i \in \{0,1\} \quad \text{pre } i \in I \quad (2.1.11)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I, \quad j \in J \quad (2.1.12)$$

Účelová funkcia (2.1.07) udáva celkové náklady medzi zákazníkmi a ich obsluhujúcimi strediskami, ktorú minimalizujeme. Podmienky (2.1.08) zabezpečujú, aby každý zákazník j z množiny J bol priradený k práve jednému vybudovanému stredisku i z množiny I . Podmienky (2.1.09) zabezpečujú, aby nebolo možné priradiť zákazníka j z množiny J k nevybudovanému stredisku i z množiny I . Vybudovanie maximálne p stredísk zabezpečuje podmienka (2.1.10). Podmienky (2.1.11) a (2.1.12) určujú obor hodnôt rozhodovacích premenných. Riešiť úlohu p -mediánu je možné rôznymi prístupmi. Tieto prístupy sú popísané v zdrojoch [41] a [45].

Úloha váženého p -mediánu

Majme nasledovný príklad. Je zadaná množina I možných umiestnení požiarnych jednotiek a množina J obsluhovaných obcí, kde každá obec j má b_j stavieb, v ktorých môže vzniknúť požiar. Nech konštanty t_{ij} označujú čas potrebný na prekonanie vzdialenosti medzi možným umiestnením i a obcou j . Určte takú štruktúru obslužného systému zloženého z najviac p požiarnych staníc, aby priemerná doba zásahu bola čo najmenšia.

Ak vychádzame z umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk, potom to znamená zanedbanie fixných nákladov, podobne ako pri úlohe p -mediánu. Náklady c_{ij} by boli priamo úmerné súčinu b_j a t_{ij} , ako časové náklady na obsluhu zákazníkov. Takto formulovaná úloha sa označuje ako úloha váženého p -mediánu [44] a jej matematický model vyzerá nasledovne (2.1.13) – (2.1.18):

$$\text{Minimalizujte} \quad \sum_{i \in I} \sum_{j \in J} t_{ij} b_j z_{ij} \quad (2.1.13)$$

$$\text{Za podmienok:} \quad \sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.1.14)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.1.15)$$

$$\sum_{i \in I} y_i \leq p \quad (2.1.16)$$

$$y_i \in \{0,1\} \quad \text{pre } i \in I \quad (2.1.17)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I, \quad j \in J \quad (2.1.18)$$

Účelová funkcia (2.1.13) udáva celkový súčet všetkých časov, čo je priamo úmerné priemernej vzdialenosti medzi zákazníkmi a ich obsluhujúcimi strediskami, ktorú minimalizujeme. Podmienky (2.1.14) zabezpečujú, aby každý zákazník j z množiny J bol priradený k práve jednému vybudovanému stredisku i z množiny I . Podmienky (2.1.15) zabezpečujú, aby nebolo možné priradiť zákazníka j z množiny J k nevybudovanému stredisku i z množiny I . Vybudovanie maximálne p stredísk zabezpečuje podmienka (2.1.16). Podmienky (2.1.17) a (2.1.18) určujú obor hodnôt rozhodovacích premenných.

Úloha p-centra

Majme úlohu p-mediánu, ktorá minimalizuje súčet vzdialenosti medzi zákazníkmi a ich obsluhujúcimi strediskami. Modifikujme túto úlohu (2.1.07) – (2.1.12), tak aby sme minimalizovali vzdialenosť najhoršie postaveného zákazníka k najbližšiemu obslužnému stredisku. Takto formulovaná úloha je úloha p-centra [44]. Modelujme horný odhad vzdialenosti najhoršie postaveného zákazníka od najbližšieho obslužného strediska rozhodovacou premennou h . Matematický model takejto úlohy (2.1.19) – (2.1.26) vyzerá nasledovne:

$$\text{Minimalizujte} \quad h \quad (2.1.19)$$

$$\text{Za podmienok:} \quad d_{ij}z_{ij} \leq h \quad \text{pre } i \in I, \quad j \in J \quad (2.1.20)$$

$$\sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.1.21)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.1.22)$$

$$\sum_{i \in I} y_i \leq p \quad (2.1.23)$$

$$y_i \in \{0,1\} \quad \text{pre } i \in I \quad (2.1.24)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I, \quad j \in J \quad (2.1.25)$$

$$h \geq 0 \quad (2.1.26)$$

Účelová funkcia (2.1.19) zhora obmedzuje všetky vzdialenosti od zákazníka k jemu priradenému stredisku. Podmienky (2.1.20) zabezpečujú, aby minimálna vzdialenosť priradeného zákazníka k najbližšiemu stredisku bola maximálne h . Podmienky (2.1.21) zabezpečujú, aby každý zákazník j z množiny J bol priradený k práve jednému vybudovanému stredisku i z množiny I . Podmienky (2.1.22) zabezpečujú, aby nebolo možné priradiť zákazníka j z množiny J k nevybudovanému stredisku i z množiny I . Vybudovanie maximálne p stredísk zabezpečuje podmienka (2.1.23). Podmienky (2.1.24) – (2.1.26) určujú obor hodnôt rozhodovacích premenných. Metódy riešenia úlohy p -centra sú bližšie popísané v zdroji [21].

Umiestňovacia úloha s lexikografickým min-max kritériom

Majme nasledovný príklad: Je zadaná množina možných umiestnení jednotiek záchranej služby I a množina obsluhovaných obcí J , kde každá obec j má b_j zákazníkov, z ktorých každý môže požiadať o lekársku pomoc. Nech konštanty d_{ij} označujú vzdialenosti medzi možným umiestnením i a obcou j . Určte štruktúru obslužného systému zloženého z najviac p jednotiek záchranej služby, aby dostupnosť všetkých zákazníkov mala najväčšiu mieru férovosti podľa zdroja [42].

Takto definovaná úloha predstavuje umiestňovaciu úlohu s lexikografickým min-max kritériom ako úlohu návrhu obslužného systému s férovým prístupom. Základy lexikografického prístupu položili Ogryczak a Śliwiński v zdroji [42]. Lexikografický prístup predstavuje minimalizovanie vzdialenosti všetkých zákazníkov pri rešpektovaní najmenej vzdialenosti najhoršie položeného zákazníka. Lexikografický prístup sa zakladá na vytvorení vektora $D = (D_1, D_2, \dots, D_{max})$, ktorý tvorí usporiadanú postupnosť vzdialeností zákazníka j k možným umiestneniam stredísk i udávaných v matici vzdialenosti $C = (d_{ij})$, pričom sú odstránené všetky možné duplicity. Získaná postupnosť (2.1.27) je použitá pri konštrukcii množiny kritických umiestnení I_{ju} (2.1.28) v iteračnom kroku u pre zákazníka j .

$$0 = D_0 < D_1 < \dots < D_k = \max\{d_{ij}; i \in I\} \quad (2.1.27)$$

$$I_{ju} = \{i \in I, d_{ij} = D_{k-u}\} \quad u = 0, \dots, k-1, \quad j \in J \quad (2.1.28)$$

Matematický model pre u -ty krok v riešení umiestňovacej úlohy s lexikografickým min-max kritériom je formulovaný nasledovne (2.1.29) – (2.1.35):

$$\text{Minimalizujte} \quad \sum_{j \in J} \sum_{i \in I_{ju}} b_j z_{ij} \quad (2.1.29)$$

$$\text{Za podmienok:} \quad \sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.1.30)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.1.31)$$

$$\sum_{i \in I} y_i \leq p \quad (2.1.32)$$

$$\sum_{j \in J} \sum_{i \in I_{jv}} b_j z_{ij} \leq F_v \quad \text{pre } v = 0, \dots, u-1 \quad (2.1.33)$$

$$y_i \in \{0,1\} \quad \text{pre } i \in I \quad (2.1.34)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I, \quad j \in J \quad (2.1.35)$$

Účelová funkcia (2.1.29) udáva minimalizáciu požiadaviek hraničných zákazníkov. Podmienky (2.1.30) zabezpečujú, aby každý zákazník j z množiny J bol priradený k práve jednému vybudovanému stredisku i z množiny I . Podmienky (2.1.31) zabezpečujú, aby nebolo možné priradiť zákazníka j z množiny J k nevybudovanému stredisku i z množiny I . Vybudovanie maximálne p stredísk zabezpečuje podmienka (2.1.32). V podmienkach (2.1.33) hodnota F_v predstavuje hodnotu účelovej funkcie (2.1.29) získanú v predchádzajúcom kroku. Podmienky (2.1.33) udávajú počet zákazníkov, ktorí budú mať vzdialenosť D_{k-v} od priradeného strediska. Podmienky (2.1.34) a (2.1.35) určujú obor hodnôt rozhodovacích premenných.

Tento iteratívny prístup po konečnom počte krokov zabezpečí možnosť oddelenia najhoršie postavených zákazníkov od ostatných zákazníkov, ktorých postavenie je možné zlepšiť. Nájdenie zlepšenia predstavuje posledný krok lexikografického prístupu.

Kapacitne neobmedzená umiestňovacia úloha (UFLP)

Úloha spočíva v minimalizácii celkových nákladov, ktoré zahrňujú fixné náklady f_i spojené s umiestnením strediska v mieste i a náklady c_{ij} na uspokojenie požiadaviek zákazníka j z miesta i . Podmienkou je, aby každý zákazník bol obslužený, t.j. musí byť priradený niektorému z vybudovaných stredísk. Náklady spojené s uspokojením požiadavky zákazníka j z množiny J z miesta i z množiny I sú modelované konštantami c_{ij} . Konštanty f_i predstavujú fixné náklady na udržanie obslužného strediska v mieste i z množiny I .

Táto úloha je modifikáciou umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk, pričom táto modifikácia spočíva v zanedbaní p -mediánovej podmienky (2.1.04). Zanedbaním tejto podmienky sme formulovali kapacitne neobmedzenú umiestňovaciu úlohu s nasledovným matematickým modelom (2.1.36) – (2.1.40):

$$\text{Minimalizujte} \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \quad (2.1.36)$$

$$\text{Za podmienok:} \quad \sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.1.37)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.1.38)$$

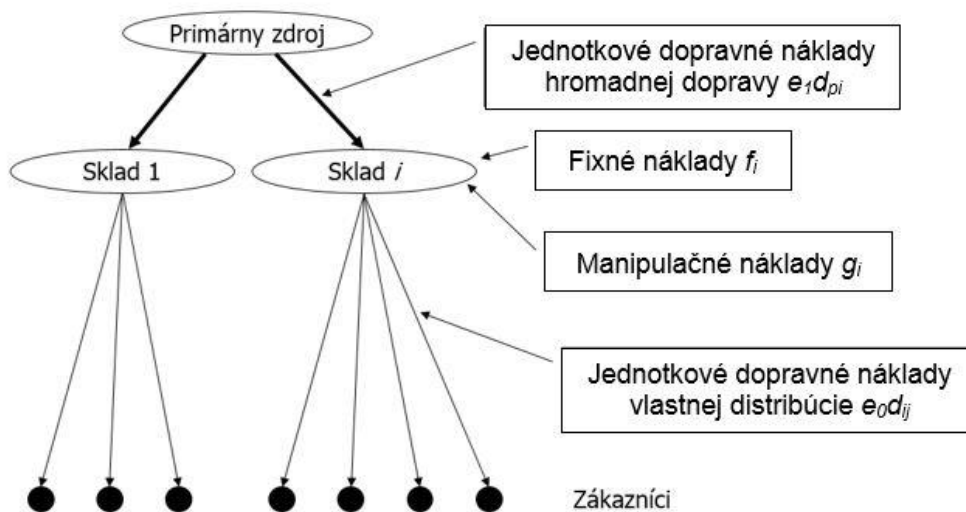
$$y_i \in \{0,1\} \quad \text{pre } i \in I \quad (2.1.39)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I, \quad j \in J \quad (2.1.40)$$

Účelová funkcia (2.1.36) vyjadruje celkové náklady uspokojenia požiadaviek zákazníkov a vybudovania stredísk, ktoré minimalizujeme. Podmienky (2.1.37) zabezpečujú, aby každý zákazník j z množiny J bol priradený k práve jednému vybudovanému stredisku i z množiny I . Podmienky (2.1.38) zabezpečujú, aby nebolo možné priradiť zákazníka j z množiny J k stredisku i z množiny I , ktoré nie je vybudované. Podmienky (2.1.39) a (2.1.40) určujú obor hodnôt rozhodovacích premenných.

Úloha návrhu distribučného systému

Pod pojmom distribučný systém podľa zdroja [32] chápeme druh dopravného systému zabezpečujúceho prepravu tovaru z jedného alebo niekoľko primárnych zdrojov k zákazníkom. Táto preprava tovaru môže byť priama alebo sa môže jednať o prepravu s prekládkou. Štruktúru distribučného systému určuje množina zákazníkov, množina primárnych zdrojov, množina prekládok, ktoré sú označované ako terminály, a taktiež aj toky tovaru medzi nimi. Prekládkové miesto môže byť realizované aj vo forme skladu. Jednotkové náklady medzi primárnym zdrojom a skladmi na prepravu jednotky tovaru na jeden kilometer sú podstatne menšie ako jednotkové náklady na prepravu jednotky tovaru na jeden kilometer medzi skladmi a zákazníkmi. To je aj predpoklad toho, aby malo zmysel tieto prekládkové miesta resp. sklady vôbec budovať. Na základe tohto predpokladu je prevoz tovaru z primárneho zdroja ku skladom realizovaný hromadnou prepravou, keď sa preváža veľké množstvo tovaru naraz. Táto preprava je zvyčajne realizovaná kyvadlovými jzdami. Prevoz tovaru zo skladov k zákazníkom sa uskutočňuje vlastnou distribúciou. Táto distribúcia je realizovaná buď kyvadlovými alebo okružnými jzdami.



Obr. 1 Distribučný systém s relevantnými nákladmi

V distribučnom systéme sa stretávame s rôznymi relevantnými nákladmi (viď Obr. 1). Fixné náklady f_i súvisia s vybudovaním skladu v mieste i , manipulačné náklady g_i súvisia s manipuláciou s tovarom v sklade, t.j. s vykládkou, uskladnením, nakládkou, atď. Dopravné náklady hromadnej prepravy na prevoz tovaru z primárneho zdroja do skladov sú priamo

úmerné súčinu jednotkových nákladov e_1 na jeden kilometer prevezeného tovaru z primárneho zdroja k skladom a vzdialenosti d_{pi} primárneho zdroja p k jednotlivým skladom i . Dopravné náklady vlastnej distribúcie na prevoz tovaru zo skladov k zákazníkom sú priamo úmerné súčinu jednotkových nákladov e_0 na jeden kilometer prevezeného tovaru zo skladu k zákazníkom a vzdialenosti d_{ij} skladu i od zákazníka j . Úlohu návrhu distribučného systému je možné popísať umiestňovacou úlohou, kde matematický model úlohy vyzerá nasledovne (2.1.41) – (2.1.45):

$$\text{Minimalizujte} \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} (e_1 d_{pi} + e_0 d_{ij} + g_i) b_j z_{ij} \quad (2.1.41)$$

$$\text{Za podmienok:} \quad \sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.1.42)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.1.43)$$

$$y_i \in \{0,1\} \quad \text{pre } i \in I \quad (2.1.44)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I, \quad j \in J \quad (2.1.45)$$

Účelová funkcia (2.1.41) určuje celkové náklady na prevádzku distribučného systému, ktoré minimalizujeme. Podmienky (2.1.42) zabezpečujú, aby každý zákazník j z množiny J bol priradený k práve jednému miestu, kde je vybudované stredisko i z množiny I . Podmienky (2.1.43) zabezpečujú, aby nebolo možné priradiť zákazníka j z množiny J k stredisku i z množiny I , ktoré nie je vybudované. Podmienky (2.1.44) a (2.1.45) určujú obor hodnôt rozhodovacích premenných.

2.2 Prehľad exaktných a semi-exaktných metód riešenia

Riešiť umiestňovacie úlohy alebo vyššie spomínané úlohy je možné pomocou exaktných metód, semi-exaktných metód, heuristik alebo metaheuristik. Exaktné metódy riešenia umožňujú získať optimálne riešenie zvolenej úlohy. Semi-exaktné metódy poskytujú zväčša optimálne riešenie. Ak semi-exaktná metóda nezíska optimálne riešenie,

získané riešenie je veľmi dobré suboptimálne riešenie, blízke k optimálnemu riešeniu. Semi-exaktná metóda nevie zaručiť získanie optimálneho riešenia vo všetkých prípadoch, ale poskytuje dolnú hranicu riešenej úlohy a tým možnosť zníženia gapu. Pod gapom rozumieme rozdiel medzi hodnotou získaného suboptimálneho riešenia a dolnou hranicou riešenej úlohy. Heuristické metódy umožňujú získať dobré riešenie, ale ich najväčšou nevýhodou je možnosť uviaznutia v lokálnom minime. Túto nevýhodu uviaznutia odstraňujú do istej miery metaheuristické metódy založené na rôznych princípoch. Ako heuristické tak aj metaheuristické prístupy však poskytujú výsledné riešenie, ktoré môže byť vzdialené od optimálneho riešenia a navyše neposkytujú odhad tejto vzdialenosti. V tejto kapitole by sme chceli bližšie popísať exaktné a semi-exaktné metódy riešenia, ktoré sa používajú na riešenie úloh návrhu verejného obslužného systému.

2.2.1 Erlenkotterov prístup ku kapacitne neobmedzenej umiestňovacej úlohe

Erlenkotterov prístup je exaktným prístupom na riešenie kapacitne neobmedzenej umiestňovacej úlohy (UFLP). Tento prístup je založený na metóde vetiev a hraníc a bol navrhnutý v roku 1978 a publikovaný v zdroji [22] pod názvom DualLoc. Modifikáciou tohto prístupu bola vytvorená na Katedre dopravných sietí Fakulty riadenia a informatiky verzia BBDual popísaná v zdroji [33]. Tento prístup vychádza z teórie duality [31], z ktorej vyplýva, že duálny model úlohy môžeme definovať pre každú lineárnu úlohu matematického programovania bez celočíselných premenných.

Metóda vetiev a hraníc

Metóda vetiev a hraníc je jednou z metód riešenia úloh celočíselného lineárneho programovania. Princíp tejto metódy spočíva v prehľadávaní a delení množiny možných riešení. Rozdeľovaním tejto množiny vzniká čiastočne usporiadaný súbor, ktorého prvky tvoria podmnožiny množiny prípustných riešení. Tento súbor je označovaný ako strom riešení, prípadne strom prehľadávania. Koreň stromu je tvorený množinou všetkých prípustných riešení a jednotlivé vrcholy stromu predstavujú jej podmnožiny. Uvedený strom riešení sa vytvára postupne, kde v každom kroku je vybraný vrchol stromu riešení, ktorý bol vytvorený, ale doposiaľ nespracovaný a tento vrchol je spracovávaný. Na počiatku prehľadávania je jediným vytvoreným, ale nespracovaným vrcholom stromu riešenia koreň stromu. Spracovanie vrcholu stromu riešenia vedie k jednému z troch prípadov:

1. V spracovávanom vrchole stromu riešení sme z riešení obsiahnutých v danom vrchole našli najlepšie riešenie.
2. Zistili sme, že vrchol stromu riešení neobsahuje optimálne riešenie úlohy a môže byť z prehľadávania vyradený.
3. Došlo k vytvoreniu aspoň dvoch nových nespracovaných vrcholov.

Pri spracovávaní stromu riešení sa uchováva hodnota účelovej funkcie doposiaľ najlepšieho nájdeného celočíselného riešenia. Táto hodnota udáva hornú hranicu účelovej funkcie optimálneho riešenia celej úlohy. Následne je pre každý novovytvorený vrchol získaná dolná hranica hodnoty účelovej funkcie najlepšieho riešenia z riešení obsiahnutých v danom vrchole. Spracovávaný vrchol nemôže obsahovať lepšie riešenie, ak dolná hranica riešenia spracovávaného vrcholu je vyššia ako hodnota účelovej funkcie doposiaľ nájdeného celočíselného riešenia, teda je vrchol vylúčený z procesu prehľadávania. Ak je dolná hranica vrcholu rovnaká ako hodnota účelovej funkcie riešenia, ide o najlepšie riešenie zo spracovávaného vrcholu. Daným riešením sa aktualizuje doposiaľ najlepšie nájdené riešenie a vrchol je vylúčený z ďalšieho prehľadávania. Ak sa vrchol stromu riešení nedá vylúčiť, v takomto prípade spracovávaný vrchol obsahuje lepšie riešenie ako to, ktoré sme doposiaľ získali a vetvením je spracovaný vrchol rozdelený aspoň na dve nové, doposiaľ nespracované, vrcholy stromu riešení. Proces vytvárania a prehľadávania stromu riešení sa neustále opakuje, dokiaľ sa nespracujú všetky nespracované vrcholy stromu riešení. Optimálnym riešením úlohy je po spracovaní všetkých vrcholov stromu doposiaľ najlepšie nájdené riešenie.

Podľa [27] a [31] je realizácia princípu metódy vetiev a hraníc možná viacerými spôsobmi. Ak chceme skonštruovať konkrétny algoritmus, je potrebné špecifikovať nasledovné vlastnosti:

1. Spôsob vetvenia
2. Výpočet dolnej hranice hodnôt účelovej funkcie riešenia zo spracovávaného vrcholu
3. Metódu nájdenia celočíselného prípustného riešenia vrcholu pre aktualizáciu hornej hranice

4. Schému prehrádavania stromu riešení - poradie spracovania vytvorených dopsial' nespracovaných vrcholov

Matematický model kapacitne neobmedzenej umiestňovacej úlohy

Modelovaná úloha spočíva v minimalizácii celkových nákladov, ktoré zahŕňajú fixné náklady f_i spojené s umiestnením strediska v mieste i a náklady c_{ij} na uspokojenie požiadaviek zákazníka j z miesta i . Podmienkou je, aby každý zákazník bol obslužený, t.j. musí byť priradený niektorému z vybudovaných stredísk. Náklady spojené s uspokojením požiadavky zákazníka j z množiny J z miesta i z množiny I sú modelované konštantami c_{ij} , ktoré sú nezáporné. Konštanty f_i predstavujú fixné náklady na udržanie obslužného strediska v mieste i z množiny I , ktorých hodnoty sú nezáporné.

V danej úlohe predstavuje I množinu kandidátov, kde je možné umiestniť stredisko i a J množinu všetkých zákazníkov j . Stretávame sa tu s dvomi rozhodnutiami. Rozhodnutie o umiestnení resp. neumiestnení strediska v mieste i z množiny I , ktoré je v modeli modelované bivalentnou premennou y_i , ktorá nadobúda hodnotu y_i rovnú jednej, ak umiestnime stredisko v mieste i z množiny I a y_i rovnú nule, ak neumiestnime stredisko v mieste i z množiny I . Rozhodnutie o priradení resp. nepriradení zákazníka j z množiny J k stredisku i z množiny I je modelované premennou z_{ij} , kde z_{ij} je rovné jednej, ak priradíme zákazníka j z množiny J k stredisku i z množiny I a z_{ij} je rovné nule, ak nepriradíme zákazníka j z množiny J k stredisku i z množiny I . Matematický model vyzerá nasledovne (2.2.01) – (2.2.05):

$$\text{Minimalizujte} \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \quad (2.2.01)$$

$$\text{Za podmienok:} \quad \sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.2.02)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.2.03)$$

$$y_i \in \{0,1\} \quad \text{pre } i \in I \quad (2.2.04)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I, \quad j \in J \quad (2.2.05)$$

Účelová funkcia (2.2.01) vyjadruje celkové náklady uspokojenia požiadaviek zákazníkov a vybudovania stredísk, ktoré minimalizujeme. Podmienky (2.2.02) zabezpečujú, že každý zákazník j z množiny J bude priradený k práve jednému vybudovanému stredisku i z množiny I . Podmienky (2.2.03) zabezpečujú, aby nebolo možné priradiť zákazníka j z množiny J k stredisku i z množiny I , ktoré nie je vybudované. Podmienky (2.2.04) a (2.2.05) určujú obor hodnôt rozhodovacích premenných.

Keďže rozhodovacie premenné majú celočíselný obor hodnôt, tento model nazývame diskretným modelom kapacitne neobmedzenej umiestňovacej úlohy. Na základe podmienky (2.2.03), ktorá hodnotou premennej y_i zhora obmedzuje hodnotu premennej z_{ij} , je možné obor hodnôt premennej z_{ij} (2.2.05) zapísať v tvare (2.2.06) :

$$z_{ij} \geq 0 \quad \text{pre } i \in I, \quad j \in J \quad (2.2.06)$$

Uskutočnenie LP-relaxácie rozhodovacej premennej y_i kapacitne neobmedzenej umiestňovacej úlohy [27] v modeli (2.2.01) – (2.2.04) , (2.2.06) a úprava primárneho matematického modelu kapacitne neobmedzenej umiestňovacej úlohy zmenia model na nasledovný tvar:

$$\text{Minimalizujte} \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \quad (2.2.07)$$

$$\text{Za podmienok:} \quad \sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.2.08)$$

$$y_i - z_{ij} \geq 0 \quad \text{pre } i \in I, \quad j \in J \quad (2.2.09)$$

$$y_i \geq 0 \quad \text{pre } i \in I \quad (2.2.10)$$

$$z_{ij} \geq 0 \quad \text{pre } i \in I, \quad j \in J \quad (2.2.11)$$

Účelová funkcia (2.2.07) vyjadruje celkové náklady uspokojenia požiadaviek zákazníkov a vybudovania stredísk, ktoré minimalizujeme. Podmienky (2.2.08) zabezpečujú, že každý zákazník j z množiny J bude priradený k práve jednému vybudovanému stredisku i z množiny I . Podmienky (2.2.09) zabezpečujú, aby nebolo možné

priradiť zákazníka j z množiny J k stredisku i z množiny I , ktoré nie je vybudované. Podmienky (2.2.10) a (2.2.11) určujú obor hodnôt rozhodovacích premenných.

Duálny model LP-relaxácie kapacitne neobmedzenej umiestňovacej úlohy

Na základe teórie duality [31] je možné skonštruovať k primárnemu modelu (2.2.07) – (2.2.11) aj duálny model (2.2.12) – (2.2.16), ktorý vyzerá nasledovne:

$$\text{Maximalizujte} \quad \sum_{j \in J} v_j \quad (2.2.12)$$

$$\text{Za podmienok:} \quad \sum_{j \in J} w_{ij} \leq f_i \quad \text{pre } i \in I \quad (2.2.13)$$

$$v_j - w_{ij} \leq c_{ij} \quad \text{pre } i \in I, \quad j \in J \quad (2.2.14)$$

$$v_j \in R \quad \text{pre } j \in J \quad (2.2.15)$$

$$w_{ij} \geq 0 \quad \text{pre } i \in I, \quad j \in J \quad (2.2.16)$$

V duálnom modeli premenná v_j odpovedá podmienke (2.2.08) z primárneho modelu, ktorá zabezpečuje, aby zákazníkovi j z množiny J bolo priradené práve jedno stredisko i z množiny I . Premenná w_{ij} odpovedá štruktúrálnej podmienke (2.2.09) z primárneho modelu, ktorá zabezpečí priradenie zákazníka len vybudovaným strediskám, teda kde y_i je rovné jednej. Premenné y_i v primárnom modeli odpovedajú podmienkam (2.2.13). Premenné z_{ij} odpovedajú podmienkam (2.2.14). Premenné v_j sú neobmedzené duálne premenné s oborom hodnôt R . Ak si vyjadríme premennú v_j z podmienky (2.2.14) podmienka bude vyzerat' nasledovne (2.2.17):

$$v_j \leq c_{ij} + w_{ij} \quad \text{pre } i \in I, \quad j \in J \quad (2.2.17)$$

Premenná w_{ij} podľa vzťahu (2.2.16) a konštanta c_{ij} nadobúdajú nezáporné hodnoty, teda ich súčet nemôže byť záporný. Aby táto podmienka vždy platila, postačí, aby hodnota premennej v_j bola rovná nule. Ak sa pozrieme na účelovú funkciu (2.2.12), kde maximalizujeme premenné v_j , tak žiadna záporná hodnota premennej v_j nám nezlepší riešenie, ale vychádzame z toho, že ak hodnota premennej v_j je rovná nule, síce riešenie nám to nezlepší, no na druhej strane ani nezhorší. Na základe dodržania podmienok (2.2.17) a snahe maximalizovať účelovú funkciu (2.2.12) nám postačia nezáporné hodnoty

premenných v_j , teda obmedziť premenné v_j na obor \mathbb{R}^+ . Premenné w_{ij} podľa (2.2.16) nadobúdajú nezáporné hodnoty. Ak si vyjadríme premennú w_{ij} z podmienky (2.2.14), podmienka bude vyzerat' nasledovne (2.2.18):

$$w_{ij} \geq v_j - c_{ij} \quad \text{pre } i \in I, \quad j \in J \quad (2.2.18)$$

Ak podmienky (2.2.16) a (2.2.18) zlúčime, dokážeme vyjadriť hodnotu premenných w_{ij} nasledovným vzťahom (2.2.17):

$$w_{ij} \geq \max\{0, v_j - c_{ij}\} \quad \text{pre } i \in I, \quad j \in J \quad (2.2.19)$$

Aby sme čo najlepšie dokázali dodržať podmienky (2.2.13), postačuje nám minimálna hodnota premennej w_{ij} daná nasledovným vzťahom:

$$w_{ij} = \max\{0, v_j - c_{ij}\} \quad \text{pre } i \in I, \quad j \in J \quad (2.2.20)$$

Zavedením tejto substitúcie do duálneho modelu dostaneme redukovaný duálny model, kde v modeli nájdeme iba duálne premenné v_j pre j z množiny J . Aby sme mali podmienky v tvare rovnosti zavedieme doplnkové premenné u_i pre i z množiny I . Redukovaný model (2.2.21) – (2.2.24) potom vyzerá nasledovne:

$$\text{Maximalizujte} \quad \sum_{j \in J} v_j \quad (2.2.21)$$

$$\text{Za podmienok:} \quad \sum_{j \in J} \max\{0, v_j - c_{ij}\} + u_i = f_i \quad \text{pre } i \in I \quad (2.2.22)$$

$$v_j \geq 0 \quad \text{pre } j \in J \quad (2.2.23)$$

$$u_i \geq 0 \quad \text{pre } i \in I \quad (2.2.24)$$

Redukovaný duálny model pozostáva z účelovej funkcie (2.2.21). V podmienke (2.2.22) je zahrnutá spomínaná substitúcia (2.2.20), kde sme nahradili premennú w_{ij} v modeli (2.2.12) – (2.2.16) a upravili podmienku (2.2.13) na tvar rovnosti vložením doplnkovej premennej u_i . Podmienky (2.2.23) a (2.2.24) sú obligatórne, teda zabezpečujú obor hodnôt rozhodovacích duálnych premenných v_j a u_i .

Výpočet dolnej hranice účelovej funkcie

Podľa [27] je lineárna relaxácia úlohy a nájdenie jej optimálneho riešenia najčastejším spôsobom určenia dolnej hranice. Lineárna relaxácia spočíva v zanedbaní podmienok celočíselnosti premenných a ich nahradenie oveľa slabšími podmienkami nezápornosti. Množina prípustných riešení relaxovanej úlohy obsahuje všetky riešenia pôvodnej úlohy. Z toho vyplýva, že hodnota účelovej funkcie optimálneho riešenia minimalizačnej relaxovanej úlohy je dolnou hranicou hodnôt účelovej funkcie riešení v spracovávanej vetve. Získanie optimálneho riešenia však vyžaduje riešenie rozsiahlej úlohy lineárneho programovania, čo kladie veľké nároky na čas výpočtu, a preto je táto dolná hranica iba odhadovaná zdola heuristickou metódou. Hodnota účelovej funkcie každého prípustného riešenia duálneho modelu (2.2.21) – (2.2.24) poskytuje dolnú hranicu (DH) hodnoty účelovej funkcie ľubovoľného prípustného riešenia primárnej úlohy (2.2.07) – (2.2.11) v nasledovnom tvare (2.2.25):

$$DH = \sum_{j \in J} v_j \quad (2.2.25)$$

Ako východiskové riešenie na určenie dolnej hranice môžeme použiť nasledovné nastavenie duálnych premenných:

$$v_j = \min \{c_{ij} : i \in I\} \quad \text{pre } j \in J \quad (2.2.26)$$

$$u_i = f_i \quad \text{pre } i \in I \quad (2.2.27)$$

Na zlepšenie tejto hranice bol navrhnutý heuristický postup, nazvaný ako vzostupný duálny algoritmus DA (**Dual Ascent Algorithm**). Ten vychádza z počiatočného prípustného riešenia duálnej úlohy, ktoré je dané vzťahmi (2.2.26) – (2.2.27) a hodnotu dolnej hranice získava vzťahom (2.2.25). V duálnom algoritme sa spracováva postupne vždy hodnota v_j prislúchajúca jednému zákazníkovi j z množiny J . Zvyšuje sa hodnota príslušného v_j po určitú hranicu a tým aj hodnota dolnej hranice. Hranicou obmedzujúcou príslušné v_j je na základe podmienky (2.2.22), buď hodnota premennej u_i alebo konštanta c_{ij} . Celý duálny vzostupný algoritmus DA (**Dual Ascent Algorithm**) je možné podľa [22] a [27] zapísať pomocou nasledujúcich krokov:

0. {Inicializácia} Určte počiatočné prípustné duálne riešenie \mathbf{u} a \mathbf{v} : $u_i = f_i$ pre $i \in I$ a $v_j = \min\{c_{ij} : i \in I\}$ pre $j \in J$. Podľa vzťahu (2.2.25) určte hodnotu DH aktuálnej

dolnej hranice.

1. Nastavte hodnotu DH^* rovnú hodnote DH .
2. Pre každé $j \in J$ vykonajte nasledujúci postup: Určte $\delta = \min\{\min\{u_i: i \in I, v_j \geq c_{ij}\}, \min\{c_{ij} - v_j: i \in I, v_j < c_{ij}\}\}$. Ak je $\delta > 0$, tak pre každé $i \in I$, pre ktoré platí $v_j \geq c_{ij}$, položte $u_i = u_i - \delta$, upravte každé v_j podľa vzťahu $v_j = v_j + \delta$ a zvýšte $DH = DH + \delta$.
3. Ak $DH^* = DH$ skončte, inak choďte na krok č. 1.

Touto metódou je získané riešenie, v ktorom pre každého zákazníka j z množiny J existuje aspoň jedno umiestnenie i z množiny I , pre ktoré platí $v_j \geq c_{ij}$ a $u_i = 0$, teda zákazník j sa „opiera“ o umiestnenie i alebo naopak, umiestnenie i sa „opiera“ o zákazníka j . Konkrétny zákazník j sa musí opierať aspoň o jedno umiestnenie i , ale môže sa opierať aj o viacero umiestnení. Taktiež každé umiestnenie i sa môže opierať o jedného alebo viacerých zákazníkov j alebo sa nemusí opierať o žiadneho zákazníka.

Výpočet hornej hranice riešenia

Duálne riešenie získané pri výpočte dolnej hranice daného vrcholu môžeme využiť pri heuristickej konštrukcii prípustného primárneho riešenia, ktorého hodnota účelovej funkcie môže byť použitá ako horná hranica prislúchajúca najlepšiemu celočíselnému riešeniu vo vrchole. Počas výpočtu aktualizujeme hodnotu hornej hranice prislúchajúcej najlepšiemu celočíselnému riešeniu podľa hodnôt účelových funkcií nájdených prípustných riešení primárnej úlohy. Pri získaní hornej hranice riešenia sa podľa zdroja [31] budeme opierať o vetu o komplementárnosti, ktorá hovorí, že pre optimálne riešenie dvojice primárnej a duálnej úlohy platí, že buď je nerovnosť jednej zo sústav splnená ako rovnosť alebo je premenná druhej sústavy rovná nule. Prípustné primárne riešenie budeme konštruovať tak, aby podmienky pre primárnu a duálnu úlohu boli splnené čo najtesnejšie. Pre všetky dvojice prípustných riešení oboch sústav platia nasledujúce podmienky (2.2.28) – (2.2.30):

$$(y_i - z_{ij}) \max\{0, v_j - c_{ij}\} \geq 0 \quad \text{pre } i \in I, \quad j \in J \quad (2.2.28)$$

$$u_i y_i \geq 0 \quad \text{pre } i \in I \quad (2.2.29)$$

$$(c_{ij} - v_j + \max\{0, v_j - c_{ij}\}) z_{ij} \geq 0 \quad \text{pre } i \in I, \quad j \in J \quad (2.2.30)$$

Ľavú stranu podmienky (2.2.30) môžeme zapísať v zjednodušených tvaroch (2.2.31) a to nasledovne:

$$(c_{ij} - v_j + \max\{0, v_j - c_{ij}\}) z_{ij} = -\min\{0, v_j - c_{ij}\} z_{ij} = \max\{0, c_{ij} - v_j\} z_{ij} \quad (2.2.31)$$

Označme $F_P(y, z)$ ako hodnotu účelovej funkcie primárneho riešenia a $F_D(u, v)$ ako hodnotu účelovej funkcie duálneho riešenia, potom pre tieto prípustné riešenia platí (2.2.32):

$$\begin{aligned} F_D(u, v) &\leq F_D(u, v) + \sum_{i \in I} u_i y_i + \sum_{i \in I} \sum_{j \in J} \max\{0, c_{ij} - v_j\} z_{ij} + \\ &+ \sum_{i \in I} \sum_{j \in J} (y_i - z_{ij}) \max\{0, v_j - c_{ij}\} = F_P(y, z) \end{aligned} \quad (2.2.32)$$

Hodnoty účelových funkcií vypočítame nasledovnými vzťahmi:

$$F_D(u, v) = \sum_{j \in J} v_j \quad (2.2.33)$$

$$F_P(y, z) = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \quad (2.2.34)$$

Ak nahradíme zápisy účelových funkcií vzťahmi (2.2.33) – (2.2.34), potom vzťah (2.2.32) bude vyzerat' nasledovne:

$$\begin{aligned} \sum_{j \in J} v_j &\leq \sum_{j \in J} v_j + \sum_{i \in I} u_i y_i + \sum_{i \in I} \sum_{j \in J} \max\{0, c_{ij} - v_j\} z_{ij} + \\ &+ \sum_{i \in I} \sum_{j \in J} (y_i - z_{ij}) \max\{0, v_j - c_{ij}\} = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \end{aligned} \quad (2.2.35)$$

Platnosť nerovnice zo vzťahu (2.2.35) je zrejmá, keďže súčet podmienok (2.2.28 – 2.2.30) je vždy nezáporný. Rovnica zo vzťahu (2.2.35), ktorej platnosť teraz dokážeme má nasledovný tvar:

$$\begin{aligned} & \sum_{j \in J} v_j + \sum_{i \in I} u_i y_i + \sum_{i \in I} \sum_{j \in J} \max\{0, c_{ij} - v_j\} z_{ij} + \\ & + \sum_{i \in I} \sum_{j \in J} (y_i - z_{ij}) \max\{0, v_j - c_{ij}\} = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \end{aligned} \quad (2.2.36)$$

Upravme rovnicu nahradením tretieho člena ľavej strany jeho pôvodným zápisom na základe vzťahu (2.2.31). Premenné u_i substituujeme podľa vzťahu (2.2.37) do druhého členu výrazu (2.2.33), ktorý sme vyjadrili z podmienky (2.2.22).

$$u_i = f_i - \sum_{j \in J} \max\{0, v_j - c_{ij}\} \quad \text{pre } i \in I \quad (2.2.37)$$

Zavedením týchto úprav bude rovnica (2.2.36) vyzerat' nasledovne:

$$\begin{aligned} & \sum_{j \in J} v_j + \sum_{i \in I} (f_i - \sum_{j \in J} \max\{0, v_j - c_{ij}\}) y_i + \sum_{i \in I} \sum_{j \in J} (c_{ij} - v_j + \max\{0, v_j - c_{ij}\}) z_{ij} + \\ & + \sum_{i \in I} \sum_{j \in J} (y_i - z_{ij}) \max\{0, v_j - c_{ij}\} = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \end{aligned} \quad (2.2.38)$$

Podrobne rozpíšme rovnicu (2.2.38).

$$\begin{aligned} & \sum_{j \in J} v_j + \sum_{i \in I} f_i y_i - \sum_{i \in I} \sum_{j \in J} \max\{0, v_j - c_{ij}\} y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} - \sum_{j \in J} (v_j \sum_{i \in I} z_{ij}) + \\ & + \sum_{i \in I} \sum_{j \in J} \max\{0, v_j - c_{ij}\} z_{ij} + \sum_{i \in I} \sum_{j \in J} \max\{0, v_j - c_{ij}\} y_i - \sum_{i \in I} \sum_{j \in J} \max\{0, v_j - c_{ij}\} z_{ij} = \\ & = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \end{aligned} \quad (2.2.39)$$

Po zjednodušení (2.2.39) bude mať rovnica tvar (2.2.40). Substituujeme sumu premenných z_{ij} hodnotou jedna pre každého zákazníka na základe podmienky (2.2.08) a opäť zjednodušíme rovnicu.

$$\sum_{j \in J} v_j + \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} - \sum_{j \in J} (v_j \sum_{i \in I} z_{ij}) \stackrel{=1}{=} \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \quad (2.2.40)$$

Výsledný výraz (2.2.41) po vyššie popísaných úpravách a zjednodušeníach ľavej strany rovnice je rovný pravej strane rovnice, čím sme dokázali platnosť rovnice (2.2.36).

$$\sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \quad (2.2.41)$$

Ak by boli podmienky (2.2.28 - 2.2.30) splnené na rovnosť, tak dané riešenia sú optimálne pre obe úlohy. Vtedy hovoríme o tzv. komplementárnych podmienkach, ktoré sú konštruované pre optimálne riešenie primárnej a duálnej úlohy. V prípade, že niektoré podmienky nie sú splnené na rovnosť, bude hodnota účelovej funkcie primárneho riešenia vyššia ako hodnota duálneho riešenia, a to práve o súčet ľavých strán podmienok (2.2.28 - 2.2.30). Čím viac sa budú ľavé strany podmienok (2.2.28 - 2.2.30) blížiť k nule, tým kvalitnejšiu hornú hranicu pre daný vrchol získame. Teda rozdiel medzi riešeniami primárnej a duálnej úlohy sa snažíme minimalizovať a môžeme ho vyjadriť vzťahom (2.2.42):

$$F_p(y, z) - F_D(u, v) = \sum_{i \in I} u_i y_i + \sum_{i \in I} \sum_{j \in J} \max\{0, c_{ij} - v_j\} z_{ij} + \sum_{i \in I} \sum_{j \in J} (y_i - z_{ij}) \max\{0, v_j - c_{ij}\} \quad (2.2.42)$$

Pri konštrukcii primárneho prípustného riešenia budeme postupovať takto: Ak umiestnime strediská obsluhy z množiny umiestnení, o ktoré sa opiera aspoň jeden zákazník ($u_i=0$) a inde nie, tak prvý člen na pravej strane rovnice (2.2.42) bude nulový. Ak budeme priradovať zákazníka j len k tým umiestneniam i , o ktoré sa zákazník opiera ($v_j \geq c_{ij}$), tak bude nulový aj druhý člen pravej strany rovnice (2.2.42). Rozdiel medzi riešeniami nám bude udávať tretí člen pravej strany rovnice (2.2.42). Aby sme minimalizovali počet takýchto výskytov budeme sa snažiť pomocou heuristického postupu nájsť čo najmenšiu množinu umiestnení, o ktorú sa opierajú zákazníci, tak, aby každý sa opieral aspoň o jedno umiestnenie z hľadanej podmnožiny. Podľa [27] je výsledkom získanie minimálnej množiny umiestnení I , o ktoré sa opiera každý zákazník z množiny J . Algoritmus pre získanie minimálnej množiny umiestnení I má nasledovné kroky:

0. Položte $I^* = \emptyset$
1. Postupne do I^* vložte všetky $i \in I$ pre ktoré $u_i = 0$ a existuje aspoň jeden zákazník j , ktorý sa opiera iba o i (t.j. $v_j \geq c_{ij}$ a pre každé $k \neq i, u_k = 0$ platí $v_j < c_{kj}$).
2. Postupne preberte všetky $j \in J$ a pokiaľ pre všetky $i \in I^*$ platí $v_j < c_{ij}$, nájdite $i^* = \text{solmin}\{c_{ij} : i \in I - I^*, u_i = 0\}$ a položte $I^* = I^* \cup \{i^*\}$.

Ak sa v metóde vetiev a hraníc objaví rozdiel medzi hornou a dolnou hranicou, Erlenkotter sa ho pokúša znížiť algoritmom duálnych úprav DAD (**Dual Adjustment Algorithm**), čo je istý typ výmennej heuristiky. Tento algoritmus je založený na zmenšení jednej hodnoty duálnej premennej v_j , ktoré vytvorí voľnú „kapacitu“ aspoň u dvoch umiestnení i_1, i_2 a následné zvýšenie aspoň dvoch hodnôt iných dvoch duálnych premenných v_j . Je teda potrebné, aby v súčasnom maximálnom duálnom prípustnom riešení u, v bola identifikovaná situácia, kedy sa zákazník j opiera aspoň o dve umiestnenia i_1, i_2 a pre všetky umiestnenia i , o ktoré sa opiera, platí $v_j > c_{ij}$. Ďalej ku každému z umiestnení i_1, i_2 musia existovať zákazníci $j(i_1), j(i_2)$, kde $j(i_1)$ sa opiera len o i_1 a $j(i_2)$ sa opiera len o i_2 . Podľa [27] je možné vytvoriť rezervu $u_i = \delta$ u všetkých umiestnení i , o ktoré sa zákazník j opieral. Teda aj u i_1 a i_2 , ak sa podarí znížiť v_j o δ tak, aby podmienky $v_j \geq c_{ij}$ zostali v platnosti pre tie i , pre ktoré platilo $v_j > c_{ij}$. Takýmto spôsobom sa zníži duálna účelová funkcia o δ , o ktoré sa zníži aj v_j . Po tejto úprave je možné zvyšovať hodnoty $v_{j(i_1)}$ ako aj $v_{j(i_2)}$ a v umiestneniach i_1 a i_2 , o ktoré sa zákazníci $j(i_1)$ a $j(i_2)$ postupne opierali vznikne kladná rezerva. Pokiaľ zvyšovanie $v_{j(i_1)}$ a $v_{j(i_2)}$ o δ nenaruší podmienky $v_{j(i_1)} \leq c_{(ij(i_1))}$ a $v_{j(i_2)} \leq c_{(ij(i_2))}$ pre tie i , o ktoré sa $j(i_1)$ a $j(i_2)$ neopierali, zvýši sa týmto spôsobom účelová funkcia minimálne o dvojnásobok δ , a teda výsledok duálnej úpravy je daný vzrastom hodnoty duálnej účelovej funkcie minimálne o δ . Uvedené duálne úpravy je možné realizovať nasledujúcim algoritmom [27] vychádzajúcim z minimálnej množiny I^* zodpovedajúcej vstupnému prípustnému riešeniu u, v nasledovne:

0. Položte $j^* = 0$ a položte $DH^* = DH = F_D(u, v)$.
1. Preskúmajte množinu zákazníkov $j \in J, j > j^*$ a pre najmenší index j zákazníka, ktorý sa opiera aspoň o dve umiestnenia z I^* , z ktorých sa každé opiera aspoň o jedného zákazníka, ktorý sa neopiera o iné umiestnenie z I , určte: $i_l(j) = \text{solmin}\{c_{ij} :$

$i \in I^*$, o i sa opiera aspoň jeden zákazník, ktorý sa neopiera o iné umiestnenie z I }, $i_2(j) = \text{solmin } \{c_{ij} : i \in I^* - \{i_1(j)\}\}$, o i sa opiera aspoň jeden zákazník, ktorý sa neopiera o iné umiestnenie I }. Ďalej určte zoznam zákazníkov j_1, j_2, \dots, j_p , ktorí sa opierajú buď len o $i_1(j)$, alebo len o $i_2(j)$ a pre každého zákazníka z tohto zoznamu špecifikujte $i(j_1), i(j_2), \dots, i(j_p)$ ako index toho jediného umiestnenia z $i_1(j)$ a $i_2(j)$, o ktoré sa opiera. Ak je zákazník j nájdený a uvedené údaje sú špecifikované, choďte na krok 2, inak choďte na krok 3.

2. Položte j^* , definujte $\delta = \min\{v_{j^*} - c_{ij^*} : i \in I, v_{j^*} - c_{ij^*} > 0\}$, položte pre každé $i \in I$, pre ktoré $v_{j^*} - c_{ij^*} > 0$ doplnkovú premennú $u_i = u_i + \delta$, upravte $v_{j^*} = v_{j^*} - \delta$. Vykonaťte algoritmus duálneho vzostupu len s množinou zákazníkov j_1, j_2, \dots, j_p . S výsledným u, v vykonaťte algoritmus duálneho vzostupu len pre zákazníka j . S výsledným u, v vykonaťte algoritmus duálneho vzostupu pre množinu J všetkých zákazníkov. Aktualizujte minimálnu množinu I^* . Aktualizujte $DH = F_D(u, v)$.
3. Ak je $DH > DH^*$ položte $DH^* = DH$ a choďte na krok 1, inak končite.

2.2.2 Vylepšenie Erlenkotterovho prístupu

Erlenkotterov prístup k riešeniu kapacitne neobmedzenej umiestňovacej úlohy, publikovaný v [22] ako algoritmus DualLoc, sa v roku 1989 pokúsil modifikovať Körkel, ktorého výsledkom modifikácie bolo skonštruovanie algoritmu nazývaného PDLoc a publikovaného v zdroji [37]. Na nich nadviazali Janáček a Buzna v zdroji [33], ktorí sa snažili vylepšiť Erlenkotterov prístup a skonštruovali algoritmus BBDual, ktorý realizuje metódu vetiev a hraníc pre riešenie kapacitne neobmedzenej umiestňovacej úlohy. Toto vylepšenie si v tejto kapitole popíšeme.

Spôsob vetvenia v algoritme BBDual

Vytváranie stromu prehľadávania je dané fixovaním premenných, ktoré rozhodujú o umiestnení strediska. V modeli (2.2.07) – (2.2.11) je rozhodovacia premenná y_i , ktorá nadobúda hodnoty y_i rovné jednej alebo y_i rovné nule v závislosti na tom, či je alebo nie je v danom mieste i postavené stredisko. Vektor y , ktorého zložky korešpondujú s hodnotami premenných y_i pre každé i z množiny I , udáva jedno riešenie úlohy. Množinu všetkých riešení úlohy predstavuje množina všetkých vektorov y , ktoré boli vytvorené vetvením. Vetvenie vykonáme fixovaním jednotlivých premenných y_i . Poradie fixácií jednotlivých

premenných y_i závisí od vyhodnotenia narušenia komplementárnych podmienok (2.2.28) – (2.2.30) pre jednotlivé strediská i . Ak by sme postupne vykonávali vetvenie podľa všetkých y_i a žiadnu vetvu nevyklúčovali, dostaneme úplný strom prehľadávania. Táto fixácia v algoritme BBDual zredukuje množiny možných riešení vypustením premenných, ktoré sú jednoznačne určené a vhodným použitím prohibívnej konštanty v účelovej funkcii pre vynútenie minimalizácie. Pri fixácii premennej na hodnotu nula, zvýšime hodnotu príslušných nákladov f_i o hodnotu prohibívnej konštanty. V prípade fixácie premennej na hodnotu jedna, nastavíme príslušné fixné náklady f_i na hodnotu nula.

Výpočet dolnej a hornej hranice riešenia

Vylepšenia Erlenkotterovho prístupu k získaniu dolnej hranice riešenia v algoritme BBDual spočívajú v určení poradia spracovania duálnych premenných v_j . Tu sú uprednostňovaní zákazníci, ktorých výber sľubuje väčšie zvýšenie dolnej hranice (2.2.25) v nasledujúcich krokoch. Toto usporiadanie všetkých zákazníkov je uskutočnené v duálnom vzostupnom algoritme (Dual-Ascent Algorithm). Tento algoritmus má nasledovné kroky:

0. {Inicializácia} Určte počiatočné prípustné duálne riešenie \mathbf{u} a \mathbf{v} : $u_i = f_i$ pre $i \in I$ a $v_j = \min\{c_{ij} : i \in I\}$ pre $j \in J$. Podľa vzťahu (2.2.25) určte hodnotu DH aktuálnej dolnej hranice.
1. Nastavte hodnotu DH^* rovnú hodnote DH .
2. Usporiadajte množinu všetkých relevantných zákazníkov J do postupnosti $j_1, \dots, j_k, \dots, j_n$ vzostupne podľa kardinality K_j , kde K_j je mohutnosť množiny $\{i \in I : c_{ij} \leq v_j\}$ pre zákazníka j .
3. Pre každé $j \in J$ vykonajte nasledujúci postup: nastavte $j_k = j$ a určte $\delta = \min\{\min\{u_i : i \in I, v_j \geq c_{ij}\}, \min\{c_{ij} - v_j : i \in I, v_j < c_{ij}\}\}$. Ak je $\delta > 0$, tak pre každé $i \in I$, pre ktoré platí $v_j \geq c_{ij}$, položte $u_i = u_i - \delta$ a upravte v_j a DH podľa vzťahov $v_j = v_j + \delta$, $DH = DH + \delta$. Ak sa mohutnosť K_j zvýšila reorganizujte postupnosť j_k, \dots, j_n .
4. Ak $DH^* = DH$ skončíte, inak choďte na krok č. 1.

Hodnota hornej hranice riešenia ako aj zníženie rozdielu medzi hornou a dolnou hranicou, ktorý je získaný pôvodným algoritmom DAD bez akýchkoľvek vylepšení sú

popísané v kapitole 2.2.1 a [27].

Schéma prehľadávania stromu riešení v metóde BBDual

Pri metóde vetiev a hraníc je dôležitým aspektom spôsob, akým sa bude postupovať pri prehľadávaní stromu riešení. Schéma prehľadania je určená elementárnym pravidlom. Elementárne pravidlo hovorí o tom, ktorý z dvoch novovytvorených vrcholov so spoločným predchodcom bude vybraný a spracovaný ako prvý. Dvoma novovytvorenými vrcholmi chápeme vrchol získaný príkazom a vrchol získaný zákazom. Vrchol, ktorý sme vytvorili fixovaním získaného kandidáta na vetvenie na hodnotu jedna je vrchol získaný príkazom. Vrchol, ktorý sme vytvorili fixáciou získaného kandidáta na vetvenie na hodnotu nula je vrchol získaný zákazom. V algoritme BBDual sa strom riešení prehľadáva do hĺbky. Pri prehľadaní do hĺbky v algoritme BBDual je elementárnym pravidlom výber vrcholu s menšou hodnotou dolnej hranice. To znamená, že v prípade vytvorenia dvoch nových vrcholov so spoločným predchodcom, ak dolná hranica vrcholu získaného zákazom je väčšia ako dolná hranica vrcholu získaného príkazom, potom ako prvý spracujeme vrchol získaný príkazom a vrchol získaný zákazom si odložíme do zásobníka nespracovaných vrcholov. V opačnom prípade ako prvý spracujeme vrchol získaný zákazom a vrchol získaný príkazom odložíme do zásobníka pre neskoršie spracovanie pri návrate (backtracking).

2.2.3 Erlenkotterov prístup pre riešenie p-mediánu s Lagrangeovou relaxáciou

Pri návrhu verejného obslužného systému sa stretávame s obmedzením nákladov na vybudovanie stredísk obsluhy. Ak sú známe náklady f_i na vybudovanie strediska v mieste i a celkové obmedzenie nákladov N na vybudovanie celého systému, potom podmienka obmedzenia nákladov na umiestnenie stredísk má tvar:

$$\sum_{i \in I} f_i y_i \leq N \quad (2.2.43)$$

Túto podmienku je možné zastúpiť podmienkou obmedzujúcou počty vybudovaných stredísk (2.1.04) a tým úlohu návrhu verejného obslužného systému previesť na úlohu o p-mediánu. Riešením úlohy p-mediánu s využitím teórie duality sa zaoberal Galvao v článku [23]. Dodanie podmienky, ktorá obmedzuje počet stredísk v úlohe návrhu verejného

obslužného systému znamená formulovať umiestňovaciu úlohu s obmedzeným počtom vybudovaných stredísk. Riešenie takejto úlohy je možné získať pomocou Lagrangeovej relaxácie [27]. Lagrangeovou relaxáciou je možné umiestňovaciu úlohu s obmedzeným počtom vybudovaných stredísk previesť na kapacitne neobmedzenú umiestňovaciu úlohu. Vychádzajme z modelu umiestňovacej úlohy (2.1.01) – (2.1.06). Pomocou Lagrangeovej relaxácie zavedením Lagrangeovho multiplikátora Lg prevedieme úlohu (2.1.01) – (2.1.06) na kapacitne neobmedzenú umiestňovaciu úlohu. Stačí relaxovať podmienku (2.1.04) a účelovú funkciu (2.1.01) doplniť o relaxovanú podmienku (2.1.04). Model relaxovanej úlohy potom vyzerá nasledovne:

$$\text{Minimalizujte} \quad \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} + Lg \left(\sum_{i \in I} y_i - p \right) \quad (2.2.44)$$

$$\text{Za podmienok:} \quad \sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.2.45)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.2.46)$$

$$y_i \geq 0 \quad \text{pre } i \in I \quad (2.2.47)$$

$$z_{ij} \geq 0 \quad \text{pre } i \in I, \quad j \in J \quad (2.2.48)$$

Opakovaným riešením úlohy (2.2.44) – (2.2.48) s meniacou sa hodnotou Lagrangeovho multiplikátora Lg dokážeme získať riešenie pôvodnej nerelaxovanej úlohy (2.1.01 – 2.1.06), Pre optimálne riešenie relaxovanej úlohy (2.2.44) – (2.2.48) platí, že ak posledný člen výrazu v účelovej funkcii (2.2.44) je rovný nule, t.j hodnota Lagrangeovho multiplikátora Lg je rovná nule, alebo podmienka (2.1.04) je splnená na rovnosť, potom je toto riešenie aj optimálnym riešením úlohy (2.1.01) – (2.1.06). Hodnota Lg , s ktorou bolo toto riešenie získané má význam veľkosti úbytku účelovej funkcie (2.1.01) optimálneho riešenia pôvodnej úlohy (2.1.01) – (2.1.06) pri zvýšení p na pravej strane podmienky (2.1.04) o jednotku. Zoberme do úvahy, že fixné náklady na vybudovanie strediska sú nulové, t.j. prvý člen účelovej funkcie je nulový. Potom získaná hodnota Lg udáva, o koľko sa hodnota optimálneho riešenia zníži, ak sa pridá jedno obslužené stredisko. S určitou toleranciou tak hodnota Lg udáva hodnotu jedného strediska v jednotkách účelovej funkcie. Ak podmienka

(2.1.04) je pre optimálne riešenie z, y úlohy (2.2.44) – (2.2.48) splnená ako ostrá nerovnosť, nemôžeme tvrdiť, že sme našli optimálne riešenie pôvodnej nerelaxovanej úlohy, ale hodnota (2.2.44) udáva dolnú hranicu optimálneho riešenia pôvodnej úlohy (2.1.01) – (2.1.06). Tretí člen účelovej funkcie (2.2.44) udáva maximálny rozdiel medzi hodnotou účelovej funkcie (2.1.04) pre získané riešenie z, y a pre neznáme optimálne riešenie nerelaxovanej úlohy (2.1.01) – (2.1.06).

V algoritme pMBBDual [7], ktorý je rozšírením algoritmu BBDual na riešenie umiestňovacej úlohy s obmedzením počtom stredísk, je realizované iteratívne vyhľadávanie Lagrangeovho multiplikátora Lg pomocou metódy bisekcie. Bisekciou sa pokúšame nájsť najlepšie nastavenie Lagrangeovho multiplikátora, ktorého hodnota odpovedá navýšeniu fixných nákladov všetkých kandidátov na stredisko. Takýmto prístupom môžeme dospieť k získaniu riešenia umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk. Získanie riešenia úlohy návrhu verejného obslužného iteratívnym prístupom je časovo náročné, čo považujeme za nevýhodu tohto prístupu. Nevýhodou je aj nezaručiteľnosť získania optimálneho riešenia úlohy návrhu verejného obslužného systému. Lagrangeovou relaxáciou síce získame vo väčšine prípadov optimálne riešenie, avšak ojedinele sú prípady, kedy optimálne riešenie nezískame [27]. Výhodou je však získanie dolnej hranice riešenia, ktorú nám určuje hodnota účelovej funkcie optimálne riešenie relaxovanej úlohy (OF_{RP}). Hodnota účelovej funkcie neznámeho optimálneho riešenia sa nachádza niekde medzi hodnotou účelovej funkcie optimálneho riešenia relaxovanej úlohy a hodnotou optimálneho riešenia nerelaxovanej úlohy. Hodnota účelovej funkcie optimálneho riešenia relaxovanej úlohy (OF_{RP}) je daná vzťahom:

$$OF_{RP} = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} + Lg\left(\sum_{i \in I} y_i - p\right) \quad (2.2.49)$$

Ak posledný člen na pravej strane vzťahu (2.2.49) je nulový, potom sme získali optimálne riešenie pôvodnej úlohy, inak nemôžeme tvrdiť, že sme našli optimálne riešenie pôvodnej nerelaxovanej úlohy, ale hodnota (OF_{RP}) udáva dolnú hranicu optimálneho riešenia pôvodnej úlohy. Hodnota účelovej funkcie optimálneho riešenia pôvodnej nerelaxovanej úlohy (OF_{UP}) je daná vzťahom:

$$OF_{UP} = \sum_{i \in I} f_i y_i + \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} \quad (2.2.50)$$

Spomínané výhody a nevýhody sú demonštrované v Tab. 1, kde porovnávame hodnoty získané algoritmom pMBBDual s hodnotami optimálneho riešenia úlohy získanými komerčným nástrojom XPRESS-IVE [51]. Výsledky experimentov v Tab. 1 sú prevzaté zo zdroja [8] pre úlohu BB100x100 o rozsahu 100 kandidátov na umiestnenie a 100 zákazníkov. Súbory, ktoré prislúchajú testovacej úlohe BB100x100 sú uvedené v prílohe 1.

Tab. 1 Porovnanie získaných riešení nástrojom XPRESS-IVE a algoritmom pMBBDual

p	XPRESS-IVE		pMBBDual		
	OF	NoF	OF _{UP}	NoF	OF _{RP}
10	1275	10	1275	10	1275
20	745	20	745	20	745
30	521	30	521	30	521
40	373	40	385	39	373
50	263	50	315	45	260
60	176	60	198	57	174
70	112	70	142	65	112
80	64	80	97	73	62
90	25	90	28	89	25

¹ stĺpec p udáva hodnotu maximálneho počtu stredísk

² stĺpec OF udáva hodnotu účelovej funkcie optimálneho riešenia získaného solverom XPRESS-IVE

³ stĺpec NoF udáva počet vybudovaných stredísk

⁴ stĺpec OF_{UP} – hodnotu riešenia nerelaxovanej (pôvodnej) úlohy

⁵ stĺpec OF_{RP} – hodnotu riešenia LP-relaxácie – dolná hranica optimálneho riešenia pôvodnej úlohy

⁶ zelená farba – nájdená hodnota optimálneho riešenia

⁷ oranžová farba – počet vybudovaných stredísk v optimálnom riešení

2.2.4 ZEBRA

Z-Erlange and BRanch Algoritmus, nazývaný aj ako ZEBRA, je exaktný algoritmus pre riešenie úlohy p-mediánu v návrhu obslužného systému. Princíp algoritmu ZEBRA popísaný v zdroji [24] spočíva v reformulácii pôvodnej úlohy p-mediánu na jej pokrývaciu úlohu. Základy tohto princípu položil Cornuéjols a kol., ktorí popísali myšlienku redukcie pre umiestňovaciu úlohu nazvanú ako kanonická reprezentácia (CR) v zdroji [16] a [17]. Neskôr na neho nadviazali Garcia, Labbé a Marín, ktorý tento princíp použili na riešenie úlohy p-mediánu s využitím poznatkov [26] a skonštruovali algoritmus ZEBRA opísaný v zdroji [24]. Spomínaný princíp sa zakladá na tom, že pre každého zákazníka j z množiny J je vytvorený vektor $D_j = (D_{j1}, D_{j2}, \dots, D_{jG_j})$, ktorý tvorí usporiadanú postupnosť

vzdialeností zákazníka j k možným umiestneniam stredísk i udávaných v matici vzdialenosti $C = (d_{ij})$, pričom sú odstránené všetky možné duplicity. Získaná postupnosť pre zákazníka j má potom nasledovný tvar (2.2.51):

$$0 = D_{j_1} < D_{j_2} < \dots < D_{j_{G_j}} = \max\{d_{ij}: i \in I\} \quad (2.2.51)$$

Konštanta G_j predstavuje počet prvkov postupnosti (2.2.51) pre zákazníka j , ktorá odpovedá počtu rôznych vzdialeností v príslušnom stĺpci matice C . V postupnosti (2.2.51) je hodnota D_{j_l} rovná nule pre všetky j , čo vyplýva z návrhu algoritmu ZEBRA, ktorý rieši úlohy so štvorcovou maticou vzdialenosti o veľkosti $n \times n$. Z toho vyplýva, že množina umiestnených stredísk I a množina zákazníkov J je totožná. Pre zápis matematického modelu pokrývacej úlohy pre p -medián je potrebné si zaviesť nasledovné rozhodnutia. Rozhodnutie o vybudovaní strediska v mieste i z množiny I je modelované bivalentnou premennou y_i , ktorá nadobúda hodnotu y_i rovnú jednej, ak v mieste i bude vybudované obslužné stredisko a hodnotu y_i rovnú nule, ak v mieste i nebude vybudované obslužné stredisko. Rozhodnutie o existencii aspoň jedného vybudovaného strediska do vzdialenosti D_{jk} pre zákazníka j je modelované premennou z_{jk} pre $k = \{1, 2, \dots, G_j\}$. Premenná z_{jk} je rovná nule, ak existuje aspoň jedno vybudované stredisko, ktorého vzdialenosť od zákazníka j je menšia ako hodnota D_{jk} . Premenná z_{jk} je rovná jednej, ak neexistuje vybudované stredisko, ktorého vzdialenosť od zákazníka j je menšia ako hodnota D_{jk} . Model pokrývacej úlohy pre p -medián je možné zapísať nasledovne (2.2.52) – (2.2.56):

$$\text{Minimalizujte} \quad \sum_{j \in J} \sum_{k=2}^{G_j} (D_{jk} - D_{j_{k-1}}) z_{jk} \quad (2.2.52)$$

$$\text{Za podmienok:} \quad \sum_{i \in I} y_i = p \quad \text{pre } j \in J \quad (2.2.53)$$

$$z_{jk} + \sum_{\substack{i \in I \\ d_{ij} < D_{jk}}} y_i \geq 1 \quad \text{pre } j \in J, \quad k = \{2, 3, \dots, G_j\} \quad (2.2.54)$$

$$y_i \in \{0, 1\} \quad \text{pre } i \in I \quad (2.2.55)$$

$$z_{jk} \geq 0 \quad \text{pre } j \in J, \quad k = \{2, 3, \dots, G_j\} \quad (2.2.56)$$

Účelová funkcia (2.2.52) je redukovaným vyjadrením účelovej funkcie úlohy p -mediánu (2.1.07), ktorá udáva súčet vzdialeností medzi zákazníkmi a ich najbližšími vybudovanými strediskami. Podmienka (2.2.53) zabezpečuje vybudovanie presne p stredísk. Podmienky (2.2.54) hovoria, že premenná z_{jk} môže nadobúdať hodnotu nula, ak je v dosahu D_k od zákazníka j vybudované aspoň jedno stredisko. V opačnom prípade bude z_{jk} rovné jednej. Podmienky (2.2.55) a (2.2.56) určujú obor hodnôt rozhodovacích premenných. V redukovanom modeli (2.2.52) – (2.2.56) je index k inicializovaný od hodnoty dva, keďže premenná z_{j1} bude nadobúdať pre všetkých zákazníkov j hodnoty jedna, keďže $d_{j1} = 0$, čo znamená nevybudovanie strediska bližšie ako je d_{j1} . Keďže algoritmus ZEBRA je implementovaný na štvorcové matice vzdialeností, z čoho vyplýva aj totožnosť množín I a J , potom je možné vykonať úpravu podmienok (2.2.54), ktoré obsahujú premennú z_{j2} . Úprava spočíva v redukování podmienky na tvar (2.2.57), vďaka čomu nie je nutné premenné z_{j2} vôbec definovať a postačí nám substitúcia:

$$y_j = 1 - z_{j2} \quad (2.2.57)$$

Zadefinujme si pre každého zákazníka vektor Z_j s nasledovnou štruktúrou :

$$Z_j = (1, 1, 1, \dots, 1, 0, 0, 0, \dots, 0) \quad (2.2.58)$$

Z tejto štruktúry vyplýva, že vektor Z_j nadobúda z počiatku hodnoty jedna, no neskôr sa jeho hodnota mení na hodnotu nula. Táto zmena je spôsobená pokrytím zákazníka j aspoň jedným vybudovaným strediskom $i \in I$. Táto vlastnosť štruktúry vektorov Z_j tvorí podstatu princípu algoritmu ZEBRA. Berme do úvahy lineárnu relaxáciu LP redukovaného modelu (2.2.52) – (2.2.56). Odstránením všetkých štruktúrálnych podmienok (2.2.54) pre hodnotu $k \geq 3$ získame lineárnu relaxáciu LP_0 pôvodnej úlohy LP . Keďže všetky koeficienty účelovej funkcie sú nezáporné a samotná účelová funkcia je v tvare minimalizácie, môžu byť všetky premenné z_{jk} pre $k \geq 3$ fixované na nula. Získaním optimálneho riešenia LP_0 v kroku k je možné získať aj optimálne riešenie redukovanej úlohy LP . Pre získanie optimálneho riešenia je dôležité preskúmať hodnoty premenných z_{jk} . Teda ak v kroku $k = 2$ sú pre všetky j hodnoty $z_{j2} = 0$, potom sme získali optimálne riešenie úlohy LP . V opačnom prípade dochádza k dodaniu premennej z_{j3} do modelu a vytvoreniu novej relaxácie LP_0 . Tento proces sa

opakuje, pokiaľ nie sú všetky hodnoty $z_{jk} = 0$ pre všetkých zákazníkov j v danom kroku k , kedy by sme získali optimálne riešenie úlohy LP . Podľa [24] je možno tento princíp rozpisat' do týchto krokov:

1. Nech LP_0 je relaxáciou LP , ktorá je lineárnou relaxáciou problému (2.2.52) – (2.2.56). Nech LP_0 obsahuje premenné y_i pre $i \in I$ a podmienky s premennými z_j pre $j \in J$. Rovnako nech je redukovaná aj účelová funkcia. Položte $t_j = 2$ pre každé $j \in J$.
2. Riešte úlohu LP_0 a získajte optimálne riešenie (y^*, z^*) .
3. Nech $\Gamma = \{j: z_{jt_j}^* > 0\}$. Ak Γ je prázdna množina, získané riešenie úlohy LP_0 je optimálnym riešením úlohy LP a algoritmus končí. Inak pokračujte krokom 4.
4. Pre každý index $t_j \in \Gamma$ vykonajte nasledovné: vytvorte bivalentnú premennú $z_{jt_{j+1}}$, do modelu úlohy LP_0 pridajte podmienku (2.2.54) pre $k = t_{j+1}$, upravte účelovú funkciu doplnením výrazu $(D_{jt_{j+2}} - D_{jt_{j+1}}) z_{jt_{j+1}}$, zvýšte hodnotu t_j o jedna a vráťte sa na krok 2.

Tento algoritmus dokáže poskytnúť optimálne riešenie LP relaxácie a je včlenený do metódy vetiev a hraníc, kde východisková formulácia riešeného podproblému každého vrcholu potomka, ktorý vznikol fixáciou premennej y_i rodičovského vrcholu, buď na hodnotu nula alebo jedna, je konečnou formuláciou jeho rodičovského vrcholu. Táto exaktná metóda umožňuje riešiť p -medián problém s redukovanou formuláciou, čo predstavuje použitie menej podmienok a premenných ako klasická formulácia. V každej iterácii je získané optimálne riešenie LP relaxácie, ktoré udáva dolnú hranicu optimálneho riešenia ľubovoľnej podúlohy. Algoritmus ZEBRA realizuje metódu vetiev a hraníc s využitím vyššie uvedeného algoritmu pre získanie dolnej hranice spracovávaného vrcholu. Majme prípustné riešenie (y, z) s hodnotou účelovej funkcie f . Vykonaním kroku dva vyššie uvedeného algoritmu získame optimálne riešenie (y^*, z^*) a hodnotu účelovej funkcie f^* . Ak $f^* \geq f$ potom je v metóde vetiev a hraníc spracovávaný vrchol vylúčený z procesu prehľadania stromu, keďže žiadne lepšie riešenie vo vrchole nebude nájdené. Prehľadanie stromu vetvenia sa uskutočňuje fixáciou premenných y_i na hodnotu jedna alebo nula, kde stratégiou najlepší vhodný volíme vrchol ďalšieho spracovania, teda vrchol s menšou dolnou hranicou. Na výber fixovanej premennej y_i je podľa zdroja [24] použité silné vetvenie

a hybridná dynamická stratégia pseudonákladov. Na získanie hornej hranice predstavujúcej hodnotu prípustného riešenia ako aj samotného prípustného riešenia je použitá heuristická metóda nazývaná Popstar, ktorá je bližšie popísaná v zdroji [47]. Podľa zdroja [24] je možné zabezpečiť získanie hornej hranice v konkrétnej vetve s využitím iných heuristických metód, nakoľko je metóda Popstar pamäťovo náročná. Pre algoritmus ZEBRA je dôležité nastavenie začiatkovej hĺbky pre z -premenných, ktorá je v článku [24] označovaná ako sdz . Parameter sdz udáva maximálny index k premenných z_{jk} . Ak konštanta p , ktorá udáva maximálny počet stredísk obsluhy, je veľká, tak sa v optimálnom riešení očakávajú malé hodnoty vzdialeností, teda stačí malá hodnota sdz . Všeobecne je však známe, že pri začiatkovej malej hodnote parametra sdz sa vykonáva veľké množstvo iterácií, čo spôsobuje dlhý výpočtový čas. Veľká hodnota sdz si vyžaduje rozsiahle modely, dlhší výpočtový čas a vysoké nároky na pamäť. Táto počiatková hodnota sdz je v zdroji [24] odhadovaná rôznymi stratégiami. Určenie hodnoty sdz je závislé od riešenej úlohy.

Tab. 2 Výsledky experimentov algoritmu ZEBRA

File	I	p	ZEBRA			
			Time [s]	BLB	BUB	sdz
v1748	1748	5	3870*	4178344	4479421	146
v1748	1748	10	4245	2983645	2983645	77
v1748	1748	200	20	390350	390350	5
v1748	1748	300	24	286039	286039	4
v1748	1748	400	22	221526	221526	3
v1748	1748	500	22	176986	176986	3
d2103	2103	5	2872*	943629	1005136	106
d2103	2103	10	3143	687321	687321	53
d2103	2103	20	1759	482926	482926	28
d2103	2103	200	55	117753	117753	4
d2103	2103	300	305	90471	90471	3
d2103	2103	400	8917	75356	75356	3
pcb3038	3038	5	109*	0	1777835	213
pcb3038	3038	10	64*	0	1211704	118
pcb3038	3038	200	564	237399	237399	7
pcb3038	3038	300	274	186833	186833	5
pcb3038	3038	400	106	156276	156276	4
pcb3038	3038	500	115	134798	134798	4

¹ stĺpec *File* udáva testovaný benchmark

² stĺpec *|I|* udáva veľkosť riešenej úlohy

³ stĺpec *p* udáva hodnotu maximálneho počtu stredísk

⁴ stĺpec *Time* [s] udáva čas trvania výpočtu

⁵ stĺpec *BLB* udáva dolnú hranicu riešenia

⁶ stĺpec *BUB* udáva hornú hranicu riešenia

⁷ stĺpec *sdz* udáva hodnotu nastaveného parametra algoritmu ZEBRA

⁸ zelená farba – hodnota optimálneho riešenia

Výhodou algoritmu ZEBRA je získanie exaktného riešenia rozsiahlych úloh za pomerne krátky výpočtový čas pomocou redukovanej formulácie. Týmto algoritmom bola podľa zdroja [24] riešená úloha o 24978 zákazníkoch s prijateľným výpočtovým časom. Medzi nevýhody algoritmu ZEBRA patrí nutnosť implementácie príslušných metód ako dodatok k metóde vetiev a hraníc, ktoré môžu byť časovo náročné, ako aj nastavenia počiatočných parametrov pre efektívnejší výpočet. Nie vždy však algoritmom ZEBRA môžeme získať optimálne riešenie za prijateľný čas. Všetky tieto výhody a nevýhody sú demonštrované výsledkami experimentov v Tab. 2. Výsledky experimentov algoritmu ZEBRA uvedené v Tab. 2 pre riešenie úlohy p-mediánu sú prevzaté zo zdroja [24].

2.2.5 Radiálny prístup s deliacimi bodmi

Podľa zdroja [36] je radiálny prístup s deliacimi bodmi pre riešenie úlohy návrhu verejného systému založený na tom, že úlohu p-mediánu zapísanú modelom (2.1.07) – (2.1.12) je možné preformulovať na pokrývaciú úlohu a riešiť ju pomocou univerzálnych IP solverov. Zatiaľ čo úlohy p-mediánu pre rozsiahle úlohy často časovo zlyhávajú, pokrývacie modely je možné vyriešiť v priebehu niekoľkých desiatok sekúnd. Podľa zdrojov [35] a [38] je to možné využiť pre riešenie úlohy návrhu verejného obslužného systému pomocou pokrývacieho modelu. Treba však podotknúť, že získanie optimálneho riešenia pokrývacej úlohy nezaručuje získanie optimálneho riešenia úlohy p-mediánu (2.1.07) – (2.1.12). Podstata aproximatívneho pokrývacieho prístupu spočíva v relaxácii priradenia zákazníka k najbližšiemu vybudovanému stredisku. Podľa zdroja [38], aby bolo možné ľubovoľnú celočíselnú vzdialenosť aproximovať, je potrebné rozdeliť postupnosť relevantných hodnôt z matice vzdialeností $\{d_{ij}\}$ do $r+1$ zón. Ide o náhradu oboru vzdialeností d_{ij} údajom, do ktorého z intervalov $(0, D_1)$, (D_1, D_2) , ..., (D_r, D_m) daná hodnota patrí. Na základe tejto informácie dokážeme ľubovoľnú vzdialenosť aproximovať zdola i zhora. Hodnota D_m predstavuje maximálnu relevantnú vzdialenosť, ktorá nemusí byť automaticky najväčšia vzdialenosť vyskytujúca sa v matici vzdialenosti. Pomocou deliacich bodov D_0, D_1, \dots, D_m definujeme konštanty e_k pre $k = 0, \dots, r$ podľa vzťahu (2.2.59):

$$e_k = D_{k+1} - D_k \quad \text{pre } k = 0, 1, \dots, r \quad (2.2.59)$$

Zavedme si umiestňovacie premenné y_i pre všetkých kandidátov na umiestnenie strediska z množiny I . Pre každého zákazníka j z množiny J a každú zónu k zavedme doplnkovú bivalentnú premennú x_{jk} , ktorá bude nadobúdať hodnotu jedna vtedy a len vtedy, keď vzdialenosť zákazníka j od strediska i je väčšia ako D_k . To je vyjadrené nasledovným predpisom (2.2.60).

$$x_{jk} = \begin{cases} 1 & \text{ak } d_{ij} > D_k \\ 0 & \text{ak } d_{ij} \leq D_k \end{cases} \quad (2.2.60)$$

Podľa zdrojov [35] a [38] platí tvrdenie, že ak hodnota d_{ij} padne do intervalu (D_k, D_{k+1}) , je odhadnutá zdola hodnotou D_k , zhora hodnotou D_{k+1} s maximálnou chybou e_k . Na základe uvedených definícií konštant e_k a premenných x_{jk} pre každého zákazníka j a každú zónu k môžeme aproximovať ľubovoľnú vzdialenosť zdola výrazom (2.2.61) a zhora výrazom (2.2.62).

$$e_0 x_{j1} + e_1 x_{j2} + e_2 x_{j3} + e_3 x_{j4} + \dots + e_{r-1} x_{jr} \quad (2.2.61)$$

$$e_0 x_{j0} + e_1 x_{j1} + e_2 x_{j2} + e_3 x_{j3} + \dots + e_r x_{jr} \quad (2.2.62)$$

Aby sme získali čo najlepší odhad hodnoty pôvodnej účelovej funkcie (2.1.07) je potrebné nastaviť hodnoty premenných x_{jk} pre každého zákazníka j a každú zónu k . Pre každú usporiadanú trojicu $[i, j, k] \in I \times J \times \{1, 2, \dots, r\}$ zavedieme takú 0-1 konštantu a_{ij}^k , ktorá bude nadobúdať svoju hodnotu podľa predpisu (2.2.63).

$$a_{ij}^k = \begin{cases} 1 & \text{ak } d_{ij} \leq D_k \\ 0 & \text{ak } d_{ij} > D_k \end{cases} \quad (2.2.63)$$

Potom matematický model pre horný odhad vzdialenosti vyzerať nasledovne:

$$\text{Minimalizujte} \quad \sum_{j \in J} \sum_{k=0}^r e_k x_{jk} \quad (2.2.64)$$

$$\text{Za podmienok:} \quad x_{jk} + \sum_{i \in I} a_{ij}^k y_i \geq 1 \quad \text{pre } j \in J, \quad k = \{0, 1, \dots, r\} \quad (2.2.65)$$

$$\sum_{i \in I} y_i = p \quad \text{pre } j \in J \quad (2.2.66)$$

$$x_{jk} \geq 0 \quad \text{pre } j \in J, \quad k = \{0, 1, \dots, r\} \quad (2.2.67)$$

$$y_i \in \{0, 1\} \quad \text{pre } i \in I \quad (2.2.68)$$

Účelová funkcia (2.2.64) poskytuje horný odhad pôvodnej účelovej funkcie (2.1.07). Sústava štruktúrnych podmienok (2.2.65) hovorí, že premenná x_{jk} môže nadobudnúť hodnotu nula, ak v dosahu D_k od zákazníka j existuje aspoň jedno vybudované stredisko. Inak bude $x_{jk} = 1$. Podmienka (2.2.66) limituje počet vybudovaných stredísk. Ostatné podmienky určujú definičný obor premenných. Keďže pre premenné x_{jk} platí čiastočná vlastnosť celočíselnosti, stačí zabezpečiť ich nezápornosť. Štruktúra modelu spôsobí, že budú nadobúdať práve jednu z hodnôt nula alebo jedna. Podobne je konštruovaný aj matematický model pre dolný odhad vzdialenosti, ktorý je podrobne popísaný v zdroji [38]. Otázkou však ostáva, ako pre daný počet $r+1$ zón určiť deliace body a ako zvýšiť presnosť navrhutej metódy vzhľadom k nárokom na výpočtový čas. Voľba hraníc jednotlivých zón je kľúčová vzhľadom na presnosť získaného riešenia. Na základe deliacich bodov D_1, D_2, \dots, D_r sa totiž počítajú konštanty pokrývacieho modelu e_k pre $k = 0, 1, \dots, r$, ktoré priamo ovplyvňujú presnosť riešenia úlohy. Zdroj [38] definuje úlohu optimálneho zónovania matice vzdialeností takto: Je daná matica vzdialeností $\{d_{ij}\}$ medzi kandidátmi na umiestnenie strediska i z množiny I a zákazníkmi j z množiny J , pričom pre každú vzdialenosť platí, že $d_{ij} \in \langle 0, d_m \rangle$. Hodnota d_m predstavuje najväčšiu vzdialenosť v matici. Úlohou je nájsť práve r celočíselných hodnôt D_1, D_2, \dots, D_r tak, aby platil vzťah (2.2.69) a celková chyba, ktorá vznikne náhradou vzdialenosti d_{ij} jej horným alebo dolným odhadom, bola minimálna.

$$0 = D_0 < D_1 < D_2 < \dots < D_r < D_{r+1} = d_m \quad (2.2.69)$$

Výpočet optimálnych deliacich bodov môže byť rozdielny pre horný a dolný odhad vzdialeností, hoci obidva modely spájajú rovnaké predpoklady. Pri tvorbe modelu na výpočet optimálnych deliacich bodov D_1, D_2, \dots, D_r vychádza z princípu, že jednotlivé vzdialenosti z matice $\{d_{ij}\}$ možno usporiadať do rastúcej postupnosti hodnôt $d_0 < d_1 < \dots < d_m$. S každou hodnotou d_h je spojená početnosť výskytov N_h tejto hodnoty v matici $\{d_{ij}\}$. Pre

každé celé číslo h z množiny $\{0, 1, \dots, m\}$ je zavedená hodnota N_h . Hodnota N_h daná vzťahom (2.2.70) udáva počet prvkov matice $\{d_{ij}\}$ rovných vzdialenosti d_h , pričom z každého stĺpca vylučuje $p - 1$ najväčších hodnôt.

$$N_h = \left| \left\{ d_{ij} : i \in I, j \in J, d_{ij} = d_h \right\} \right| \quad \text{pre } h = 0, 1, \dots, m \quad (2.2.70)$$

Ak existuje práve r rôznych hodnôt medzi d_0 a d_m , položia sa deliace body D_1, D_2, \dots, D_r rovné týmto hodnotám. Riešením pokrývacej úlohy (2.2.64) – (2.2.68) je získané exaktné riešenie pôvodnej úlohy (2.1.07) – (2.1.12). V opačnom prípade je získaný len odhad hodnoty pôvodnej účelovej funkcie, nakoľko o vzdialenosti medzi každým zákazníkom a jeho najbližším vybudovaným strediskom je informácia, že padne do niektorého z intervalov. Spomínané početnosti N_h nám požadovaný odhad neposkytnú, pretože odrážajú iba výskyty hodnôt d_h v matici $\{d_{ij}\}$. Ak pre každé $h = 0, 1, \dots, m$ zavedieme konštantu n_h vyjadrujúcu početnosť výskytov hodnoty d_h v zatiaľ neznámom optimálnom riešení, potom môžeme celkovú chybu odhadu vzdialeností minimalizovať pomocou deliacich bodov získaných riešením matematického modelu. Vzťah medzi hodnotou N_h a n_h je daná nasledovným vzťahom:

$$n_h = N_h e^{\frac{-d_h}{T}} \quad (2.2.71)$$

Symbol T reprezentuje parameter tzv. teplota, N_h je spomínaná frekvencia výskytu. Pre horný odhad vzdialeností optimálnych deliacich bodov matematický model vyzerá nasledovne (2.2.72) – (2.2.76):

$$\text{Minimalizujte} \quad \sum_{t=1}^m \sum_{h=1}^t (d_t - d_h) n_h z_{ht} \quad (2.2.72)$$

$$\text{Za podmienok:} \quad z_{(h-1)t} \leq z_{ht} \quad \text{pre } t = 2, 3, \dots, m; \quad h = 2, 3, \dots, t \quad (2.2.73)$$

$$\sum_{t=h}^m z_{ht} = 1 \quad \text{pre } h = 1, 2, \dots, m \quad (2.2.74)$$

$$\sum_{t=1}^{m-1} z_{tt} = r \quad (2.2.75)$$

$$z_{ht} \in \{0,1\} \quad \text{pre } t = 1,2, \dots, m; \quad h = 1,2, \dots, t \quad (2.2.76)$$

Účelová funkcia (2.2.72) minimalizuje celkovú chybu odhadu vzdialeností, pričom berie do úvahy očakávanú početnosť n_h každej hodnoty d_h v optimálnom riešení úlohy. Bivalentná premenná z_{ht} nadobúda hodnotu jedna práve vtedy, keď vzdialenosť d_h patrí do intervalu, ktorý končí hodnotou d_t . Väzobné podmienky (2.2.73) zabezpečujú, že vzdialenosť d_{h-1} môže patriť do intervalu, ktorý končí hodnotou d_t , len vtedy, ak všetky vzdialenosti medzi d_{h-1} a d_t patria do tohto intervalu. Podmienka (2.2.74) hovorí, že každá hodnota musí patriť práve do jedného intervalu. Posledná podmienka (2.2.75) zabezpečí práve r deliacich bodov. Nenulové hodnoty premenných z_{ht} indikujú vzdialenosti d_t , ktoré zodpovedajú deliacim bodom. Podrobnosti o radiálnom prístupe a deliacich bodoch pre dolný odhad vzdialenosti sú popísané v zdrojoch [35] a [38].

Výhodou tohto prístupu je možnosť riešiť úlohu návrhu verejného obslužného systému pomocou univerzálnych solverov, ktoré nevyžadujú zložité implementačné zručnosti. Všeobecne je známe, že riešenie rozsiahlych kapacitne obmedzených úloh je v rámci univerzálnych solverov pamäťovo aj časovo náročné. Výsledky numerických experimentov uvedené v zdrojoch [35] a [38] potvrdzujú, že radiálny prístup významným spôsobom rozširuje možnosti návrhu štruktúry verejného obslužného systému. Výhodou tohto prístupu je pomerne krátky čas výpočtu v porovnaní s exaktným riešením. Tento prístup však nezaručuje získanie optimálneho riešenia úlohy, čo je jeho nedostatkom. Dôvodom je, že formulácia úlohy radiálnym prístupom vedie k istej strate presnosti. K demonštrácií uvedených výhod a nevýhod boli v Tab. 3 porovnané prístupy z hľadiska presnosti riešenia. Presnosť riešenia je daná veľkosťou tzv. *gapu*, ktorý udáva percentuálny rozdiel hodnôt účelovej funkcie exaktného a približného riešenia. Tento *gap* je možné zapísať nasledovne:

$$gap = \frac{|ES - RÚF|}{ES} * 100 \quad (2.2.77)$$

Vo vzorci pre výpočet *gapu* (2.2.77) je hodnota účelovej funkcie exaktného riešenia označená symbolom *ES* (*exact solution*) a skutočnú hodnotu účelovej funkcie pokrývacieho riešenia *RÚF*. Exaktné riešenie s hodnotou účelovej funkcie *ES* bolo získané použitím univerzálného IP solveru na model lokačno-alokačnej úlohy. Výsledky experimentov v Tab. 3 sú prevzaté zo zdroja [38].

Tab. 3 Porovnanie presnosti pokrývacej metódy vzhľadom k optimálnemu riešeniu

I	p	Exaktné riešenie		Pokrývaci prístup r = 20, T = 1		
		ÚF	Čas [s]	RÚF	gap [%]	Čas [s]
500	250	953	26,5	953	0,00	0,90
500	167	1828	35,76	1828	0,00	0,99
500	125	2532	30,48	2534	0,08	2,17
500	100	2961	25,33	2967	0,20	1,13
500	50	4619	38,26	4890	5,87	7,80
500	25	5899	147,78	6075	2,98	16,13
600	300	939	49,97	939	0,00	1,29
600	200	1989	47,52	1989	0,00	1,30
600	150	2664	43,23	2664	0,00	1,57
600	120	3033	56,07	3035	0,07	1,83
600	60	4498	71,50	4532	0,76	10,04
600	30	6205	182,04	6462	4,14	104,7
700	350	963	80,03	963	0,00	2,29
700	233	1865	75,17	1865	0,00	2,60
700	175	2554	76,77	2554	0,00	3,09
700	140	3013	85,97	3013	0,00	2,83
700	70	4700	129,52	4716	0,34	2,76
700	35	6104	346,48	6356	4,13	58,66

¹ stĺpec |I| udáva veľkosť riešenej úlohy

² stĺpec p udáva hodnotu maximálneho počtu stredísk

⁴ stĺpec ÚF udáva hodnotu účelovej funkcie exaktného riešenia

⁴ stĺpec Čas [s] udáva čas trvania výpočtu jednotlivých metód

⁵ stĺpec RÚF udáva hodnota riešenia pokrývacej úlohy

⁶ stĺpec Gap [%] udáva percentuálny rozdiel hodnôt účelovej funkcie exaktného a približného riešenia

⁷ zelená farba – hodnota optimálneho riešenia

⁸ oranžová farba – najlepší výpočtový čas pri získaní optimálneho riešenia

2.2.6 Lagrangeová heuristika pre riešenie úlohy váženého p-mediánu

Lagrangeová heuristika popísaná v zdroji [39] rieši úlohu váženého p-mediánu, ktorú sme popísali v kapitole 2.1.2 matematickým modelom (2.2.78) – (2.2.83).

$$\text{Minimalizujte} \quad \sum_{i \in I} \sum_{j \in J} b_j d_{ij} z_{ij} \quad (2.2.78)$$

$$\text{Za podmienok:} \quad \sum_{i \in I} z_{ij} = 1 \quad \text{pre } j \in J \quad (2.2.79)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.2.80)$$

$$\sum_{i \in I} y_i \leq p \quad (2.2.81)$$

$$y_i \in \{0,1\} \quad \text{pre } i \in I \quad (2.2.82)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I, \quad j \in J \quad (2.2.83)$$

Vykonaním Lagrangeovej relaxácie podmienok (2.2.79) v matematickom modeli (2.2.75) – (2.2.80) získame relaxovanú úlohu (2.2.84) – (2.2.88), v ktorej parametre φ_j sú Lagrangeové multiplikátory.

$$\text{Minimalizujte} \quad \sum_{i \in I} \sum_{j \in J} b_j d_{ij} z_{ij} + \sum_{j \in J} \varphi_j \left(1 - \sum_{i \in I} z_{ij} \right) \quad (2.2.84)$$

$$\text{Za podmienok:} \quad z_{ij} \leq y_i \quad \text{pre } i \in I, \quad j \in J \quad (2.2.85)$$

$$\sum_{i \in I} y_i \leq p \quad (2.2.86)$$

$$y_i \in \{0,1\} \quad \text{pre } i \in I \quad (2.2.87)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I, \quad j \in J \quad (2.2.88)$$

Pre ľubovoľné nastavenie φ_j bude hodnota optimálneho riešenia relaxovanej úlohy (2.2.84) – (2.2.88) dolnou hranicou nerelaxovanej úlohy (2.2.78) – (2.2.83). Lagrangeová úloha predstavuje nastavenie takých φ_j , kde sa bude hodnota optimálneho riešenia relaxovanej úlohy (2.2.84) – (2.2.88) najviac približovať k hodnote pôvodnej úlohy (2.2.78) – (2.2.83). Pre zadané hodnoty Lagrangeových multiplikátorov φ_j vypočítame hodnoty V_i nasledovným vzťahom:

$$V_i = \sum_{j \in J} \min\{0, b_j d_{ij} - \varphi_j\} \quad \text{pre } i \in I \quad (2.2.89)$$

Následne vyberáme p umiestnení s najmenšou hodnotou V_i , kde hodnota V_i udáva mieru výhodnosti pridania strediska i do riešenia. Pre vybrané umiestnenia i nastavíme hodnoty premenných y_i na hodnotu jedna. Hornú hranicu HH nerelaxovanej úlohy získame priradením zákazníka k najbližšiemu umiestnenému stredisku, kde $y_i = 1$ a vyhodnotením vzťahu (2.2.90), ktorý odpovedá účelovej funkcii (2.2.78).

$$HH = \sum_{i \in I} \sum_{j \in J} b_j d_{ij} z_{ij} \quad (2.2.90)$$

Pri výpočte dolnej hranice DH nerelaxovanej úlohy získanej vzťahom (2.2.91), ktorý odpovedá účelovej funkcii (2.2.84), nastavujeme prirad'ovacie premenné z_{ij} pre všetky i na hodnotu jedna v prípade, ak platí, že $y_i = 1$ a $b_j d_{ij} - \varphi_j < 0$, inak sú prirad'ovacie premenné z_{ij} nastavené na hodnotu nula.

$$DH = \sum_{i \in I} \sum_{j \in J} (b_j d_{ij} - \varphi_j) z_{ij} + \sum_{j \in J} \varphi_j \quad (2.2.91)$$

Nastavenie prirad'ovacích premenných z_{ij} môže narušovať podmienky (2.2.79), v prípade kedy zákazníkovi nie je priradené žiadne stredisko a niektorí zákazníci naopak môžu byť priradení k viacerým strediskám. Čo nám znižuje hodnotu dolnej hranice, ktorá je zaťažená posledným členom vo vzťahu (2.2.84). Znížiť mieru zaťaženia posledného člena (2.2.84) a tým vyššiu hodnotu dolnej hranice získame vylepšením hodnôt Lagrangeových multiplikátorov φ_j pomocou subgradientovej metódy. Opakovaním postupu aktualizácie premenných z_{ij} , vyhodnotenia vzťahu (2.2.91) a zmenou hodnôt Lagrangeových multiplikátorov φ_j subgradientovou metódou získame najlepšiu dolnú hranicu DH . Podrobnejšie informácie o Lagrangeovom algoritme pre riešenie úlohy p -mediánu sú uvedené v zdroji [19].

Lagrangeové multiplikátory φ_j spolu s najlepšou dolnou hranicou DH nerelaxovanej úlohy a hornou hranicou riešenia HH vstupujú do metódy vetiev a hraníc. V každom vrchole stromu prehľadania sa vykonáva proces redukcie, kde sa identifikuje množinu prikázaných a zakázaných umiestnení stredísk. Množinu prikázaných umiestnení stredísk predstavujú strediská, kde premenná y_i je fixovaná na hodnotu jedna. Množinu zakázaných umiestnení stredísk predstavujú strediská, kde premenná y_i je fixovaná na hodnotu nula. Pri výbere prikázaných a zakázaných stredísk sú strediská usporiadané na základe hodnoty V_i tak, že $V_{[i]}$ predstavuje i -tu najmenšiu hodnotu V_i . $V_{[p]}$ predstavuje najväčšiu hodnotu V_i nachádzajúcu sa v najlepšom známom riešení. Zatiaľ čo $V_{[p+1]}$ je najmenšia hodnota V_i nezahrnutá v najlepšom známom riešení. Pridanie strediska i z množiny I v najlepšom známom riešení do množiny prikázaných umiestnení stredísk je na základe platnosti vzťahu. $UB < LB - V_i + V_{[p+1]}$. Pridané stredisko pri spracovaní daného vrchole patrí do množiny

prikázaných umiestnení stredísk všetkých jeho potomkoch. Pridanie strediska i z množiny I , ktoré sa nenachádza v najlepšom známom riešení do množiny zakázaných umiestnení stredísk je na základe platnosti vzťahu $UB < LB - V_i + V_{[p+1]}$. Pridané stredisko pri spracovaní daného vrchole patrí do množiny zakázaných umiestnení stredísk všetkých jeho potomkoch. Týmto postupom sa snažíme zredukovať množinu kandidátov na vetvenie, ktoré nám môžu priniesť kvalitnejšie riešenie. Zredukovaním množiny kandidátov sa urýchlí aj metóda vetiev a hraníc spracovaným menším počtu vrcholov. Bližší popis Lagrangeovej heuristiky nájdete v zdroji [39].

3 Ciele a metodika práce

3.1 Ciele práce

Cieľom dizertačnej práce je výskum metód navrhovania verejných obslužných systémov s exaktným optimalizačným jadrom pri využití prostriedkov aplikovanej informatiky. Tento hlavný cieľ je možné rozdeliť na jednotlivé čiastkové ciele.

Prvým čiastkovým cieľom je vylepšiť riešenie úlohy návrhu verejného obslužného systému získané Erlenkotterovým prístupom s Lagrangeovou relaxáciou. Tento čiastkový cieľ v sebe zahŕňa:

- návrh heuristických metód, ktoré vylepšia riešenie získané iteratívnym prístupom,
- obmedzenie vetviaceho algoritmu v metóde vetiev a hraníc, ktoré vylepší riešenie získané iteratívnym prístupom,
- porovnanie a výber najvhodnejšieho variantu, ktorý vylepší riešenie získané iteratívnym prístupom.

Druhým čiastkovým cieľom je vylepšenie Erlenkotterovho prístupu pre riešenie úlohy návrhu verejného obslužného systému s Lagrangeovou relaxáciou v najvhodnejšom nastavení Lagrangeovho multiplikátora. Tento čiastkový cieľ zahŕňa:

- analýza nákladov distribučného systému a vypracovanie hrubého odhadu nákladov distribučného systému,
- návrh iteratívneho prístupu s odhadom Lagrangeovho multiplikátora,
- porovnanie iteratívnych prístupov založených na metóde bisekcie alebo hrubom odhade najvhodnejšieho nastavenia Lagrangeovho multiplikátora.

Tretím čiastkovým cieľom je zovšeobecnenie Erlenkotterovho prístupu pre riešenie úlohy návrhu verejného obslužného systému. Tento čiastkový cieľ pre návrh exaktného algoritmu zahŕňa nasledovné podciele:

- riešenie umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk priamo v metóde vetiev a hraníc s využitím teórie duality,
- rozpracovanie teórie duality pre získanie dobrej dolnej hranice umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk,

- výskum prínosu resp. dopadu rôznych modifikácii duálneho prístupu k získaniu dolnej hranice riešenia pre prípady, kedy model kapacitne neobmedzenej umiestňovacej úlohy rozšírime o obmedzenie počtu vybudovaných stredísk,
- výskum prínosu alebo dopadu rôzneho výberu kandidáta na vetvenie v metóde vetiev a hraníc,
- výskum prínosu resp. dopadu rôznych modifikácii duálneho prístupu k získaniu hornej hranice riešenia umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk.

3.2 Metodika práce

Metodika práce sleduje cieľ, ktorým je výskum metód s exaktným optimalizačným jadrom pri využití prostriedkov aplikovanej informatiky za účelom získania efektívneho softvérového nástroja na podporu rozhodovania o štruktúre verejného obslužného systému.

Matematický model návrhu verejného obslužného systému, ktorý minimalizuje disjutilitu priemerného zákazníka odpovedá umiestňovacej úlohe o váženom p -mediáne. Ide o úlohu 0-1 matematického programovania, ktorá patrí k NP-tažkým úlohám, kde u všetkých doteraz známych exaktných algoritmov počet operácii resp. doba výpočtu rastie s veľkosťou riešenej úlohy rýchlejšie ako akákoľvek polynomiálna funkcia. Teda efektívnosť v prípade exaktnej metódy, resp. algoritmu, bude meraná dobou, za ktorú daný algoritmus nájde pre úlohu danej veľkosti optimálne riešenie.

V prípade aproximatívnej metódy, ktorá môže viesť k suboptimálnemu riešeniu, bude efektívnosť algoritmu pre daný rozmer úlohy meraná dvoma parametrami, jednak dobou potrebnou pre získanie výsledného riešenia a ďalej odchýlkou hodnoty výsledného riešenia od optimálneho riešenia (ak je známe) alebo od dolnej hranice optimálneho riešenia.

Skúmané metódy návrhu štruktúry verejného obslužného systému sú založené na princípe vetiev a hraníc. O algoritmoch založených na tomto princípe je známe, že ich účinnosť závisí na tom ako kvalitne pracujú na danom type úlohy procedúry vypočítavajúce horné a dolné hranice najlepšieho riešenia obsiahnutého v modelmi vymedzených podmnožinách riešení. Práve tu je kvalita určená dvoma parametrami a to dobou výpočtu a presnosťou získania hranice. Jednoduchou úvahou môžeme ukázať, že krajné prístupy k návrhom zmienených procedúr nevedú ku kvalitnej metóde návrhu verejného obslužného systému. Ak je určenie dolnej hranice navrhnutou procedúrou veľmi presné, je obvykle veľmi pomalé (jedná sa prevažne o heuristiku) a teda aj keď je v metóde vetiev a hraníc vďaka svojej presnosti realizovaná na menšom počte podmnožín riešení, trvá celkový výpočet veľmi dlho. Naopak, ak je procedúra rýchla je obvykle málo presná a je potom vykonaná na omnoho väčšom počte podmnožín množiny všetkých riešení a celkový výpočet trvá opäť veľmi dlho. Preto pri výskume metód riešenia vychádzam z hypotézy, že pre daný typ úlohy existuje nejaká špecifická „zlatá stredná cesta“, kedy existujú procedúry, v ktorých vhodný kompromis rýchlosti a presnosti dosiahne to, že celková doba metódy vetiev a hraníc bude malá. Overenie tejto hypotézy v skúmanom prípade určuje i metodiku výskumu. Tá bude spočívať v hľadaní procedúr pre určovanie horných a dolných hraníc

modelom definovaných podmnožín množiny prípustných riešení návrhu verejného obslužného systému s regulovanou presnosťou a dobou výpočtu a v štúdiu toho ako príslušný kompromis ovplyvní chod celej metódy.

Erlenkotterov prístup s Lagrangeovou relaxáciou umožňuje riešiť úlohu návrhu verejného obslužného systému. Použitím Lagrangeovej relaxácie dokážem transformovať umiestňovaciu úlohu s obmedzeným počtom vybudovaných stredísk na kapacitne neobmedzenú umiestňovaciu. Získanie optimálneho riešenia danej úlohy nie je zaručené, čo považujem za hlavnú nevýhodu tohto prístupu. Nezískanie optimálneho riešenia, ale nevylučuje možnosť vylepšiť riešenie získané algoritmom pMBBDual, ktorý realizuje Erlenkotterov prístup s Lagrangeovou relaxáciou. Pre naplnení tohto cieľa, rozanalyzujem a navrhmem možné spôsoby vylepšenia tohto iteratívneho prístupu.

Prvým spôsobom vylepšenia riešenia Erlenkotterovho prístupu s Lagrangeovou relaxáciou je doplnenie riešenia pomocou heuristickej metódy. V rámci naplnenia tohto podcieľa navrhmem tri heuristické metódy založené na rôznych prístupoch k vylepšeniu riešenia, porovnam jednotlivé metódy na základe výpočtového času a hodnoty získaného riešenia s univerzálnym IP solverom.

Druhým spôsobom vylepšenia riešenia Erlenkotterovho prístupu s Lagrangeovou relaxáciou je obmedzenie vetvenia v metóde vetiev a hraníc. V rámci naplnenia tohto podcieľa upravím metódu vytvárajúcu strom vetvenia dodaním obmedzujúcich podmienok do procesu vetvenia. Porovnam tento spôsob s univerzálnym IP-solverom a spomínanými heuristickými metódami na základe výpočtového času a hodnoty získaného riešenia, kde vyberiem najvhodnejšiu metódu vylepšenia.

V rámci naplnenia čiastkového cieľa, ktorým je vylepšenie Erlenkotterovho prístupu s Lagrangeovou relaxáciou na získanie najvhodnejšieho nastavenia Lagrangeovho multiplikátora analyzujem vzťah medzi hodnotou fixných nákladov a počtom umiestnených stredísk. Na základe analýzy vypracujem hrubý odhad nastavenia Lagrangeovho multiplikátora. Navrhmem iteratívny prístup založený na hrubom odhade Lagrangeovho multiplikátora, ktorý porovnam s iteratívnym prístupom k získaniu najvhodnejšieho nastavenia Lagrangeovho multiplikátora pomocou bisekčného algoritmu vo výpočtovom čase a počte vykonaných iterácii.

V rámci možnosti riešenia úlohy návrhu verejného obslužného systému aj priamo v metóde vetiev a hraníc zovšeobecnením Erlenkotterov prístup. Naplnenie tohto

čiasťkové cieľa znamená rozpracovať teóriu duality pre úlohu návrhu verejného obslužného systému, teda umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk. Na základe podrobného rozpracovania teórie duality pre riešený problém skonštruujem exaktný algoritmus založený na metóde vetiev a hraníc s Erlenkotterovským prístupom k dolnej a hornej hranici. V rámci návrhu a konštrukcie algoritmu navrhнем procedúru pre získanie dolnej hranice, hornej hranice, procedúru pre získanie kandidáta na vetvenie v metóde vetiev a hraníc a procedúru uskutočňujúcu metódu vetiev a hraníc.

V rámci naplnenia podcieľa, ktorým je výskum prínosu resp. dopadu rôznych modifikácií duálneho prístupu k získaniu dolnej hranice riešenia, navrhнем alternatívne metódy pre získanie dolnej hranice riešenia. Procedúry porovnam na základe nasledujúcich parametrov: výpočtového času algoritmu, počtu spracovaných vrcholov v metóde vetiev a hraníc a získanej dolnej hranice koreňa. Na testovanie kvality dolnej hranice, ktorú získam jednotlivými metódami, použijem hodnotu optimálneho riešenia úlohy získaného univerzálnym IP solverom ako hornú hranicu riešenia. Nastavená horná hranica riešenia umožňuje otestovať efektívnosť vetviaceho algoritmu pri znižovaní rozdielu medzi hornou hranicou riešenia a dolnými hranicami spracovaných vrcholov v metóde vetiev a hraníc pre navrhnuté metódy získania dolnej hranice. Kvalitná dolná hranica sa odzrkadlí menším počtom spracovaných vrcholov vo vetviacom algoritme, kratším výpočtovým časom ako aj hodnotou dolnej hranice koreňa bližšie k hodnote hornej hranice riešenia.

V rámci naplnenia podcieľa, ktorým je výskum prínosu resp. dopadu rôznych modifikácií duálneho prístupu k získaniu hornej hranice riešenia umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk navrhнем viaceré metódy pre získanie hornej hranice spracovaného vrcholu v metóde vetiev a hraníc, porovnam ich s už použitou procedúrou v navrhnutom algoritme a vyhodnotím ich použiteľnosť.

V rámci naplnenia podcieľa, ktorým je výskum prínosu alebo dopadu rôzneho výberu kandidáta na vetvenie v metóde vetiev a hraníc navrhнем viaceré metódy založené na vyhodnotení komplementárnych podmienok, porovnam ich s už použitou metódou pre získanie kandidáta na vetvenie v navrhnutom exaktnom algoritme a vyhodnotím ich použiteľnosť.

Navrhnuté algoritmy budú testované na benchmarkoch z cestnej siete Slovenskej republiky, prípadne na testovaných úlohách zo zdrojov uvedených v literatúre, pre porovnanie jednotlivých prístupov. Benchmarky z cestnej siete Slovenskej republiky

rozdelím do troch skupín na základe rozsahu riešenej úlohy, a to malé , stredné a veľké. Za malé benchmarky budem považovať testovacie úlohy o rozsahu do 100 zákazníkov a 100 kandidátov na stredisko a poslúžia na verifikáciu navrhnutých prístupov a metód. Za stredné benchmarky budem považovať úlohy o rozsahu od 100 do 1000 zákazníkov a 100 až 1000 kandidátov na stredisko. Stredné benchmarky rozsahom odpovedajú skoro všetkým Slovenským samosprávnym krajom. Za veľké benchmarky budem považovať úlohy rozmerovo väčšie ako 1000 zákazníkov a 1000 kandidátov na umiestnenie stredisko. Stredné a veľké benchmarky poslúžia na vyhodnotenie vhodnosti a použiteľnosti jednotlivých navrhnutých metód v rámci algoritmov pMBBDual a pMedBBDual ako aj celkovej použiteľnosti navrhnutých algoritmov pre riešenie úlohy návrhu verejného obslužného systému. Na verifikáciu navrhnutých algoritmov sú použité benchmarky troch typov v závislosti od riešenej úlohy, a to: benchmarky pozostávajúce len z matice vzdialeností pre riešenie úlohy p-mediánu, benchmarky pozostávajúce z matice vzdialeností a požiadaviek zákazníkov pre riešenie úlohy váženého p-mediánu a benchmarky pozostávajúce z matice vzdialeností a fixných nákladov, prípadne požiadaviek zákazníkov, pre riešenie kapacitne neobmedzenej umiestňovacej úlohy. Na základe tohto rozčlenenia benchmarkov ukážem možnosť riešenia rôznych úloh návrhu verejných obslužných systémov. Všetky experimenty budú realizované na PC s procesorom i7 Q720 a 8GB RAM.

4 Výsledky a ich zhodnotenie

V tejto kapitole ukážem ako som naplnil jednotlivé ciele zadania. Ukážem ako som vylepšil riešenie získané Erlenkotterovým prístupom s Lagrangeovou relaxáciou ako aj samotný prístup k získaniu najvhodnejšieho nastavenia Lagrangeovho multiplikátora. Ukážem zovšeobecnenie Erlenkotterovho prístupu pre riešenie úlohy návrhu verejného obslužného systému priamo v metóde vetiev a hraníc, na základe ktorého som skonštruoval exaktný algoritmus ako aj vylepšenie tohto exaktného algoritmu pre získanie dolnej hranice, hornej hranice a kandidáta na vetvenie.

4.1 Zlepšenie riešenia získaného iteratívnym prístupom

Rozšírenie Erlenkotterovho prístupu popísané v kapitole 2.2.3 umožnilo riešiť úlohu návrhu verejného obslužného systému transformáciou kapacitne neobmedzenej umiestňovacej úlohy pomocou Lagrangeovej relaxácie na umiestňovaciu úlohu s obmedzeným počtom vybudovaných stredísk. Na základe získaných poznatkov o Erlenkotterovom prístupe s Lagrangeovou relaxáciou pre riešenie umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk, ktoré sú popísané v kapitole 2.2.3, som sa zamerlal na podrobnejšie rozanalyzovanie nevýhod tohto prístupu a navrhol možné vylepšenie riešenia získaného týmto iteratívnym prístupom [14].

Vylepšenie riešenia získaného Erlenkotterovým prístupom s Lagrangeovou relaxáciou spočíva v zlepšení riešenia získaného algoritmom pMBBDual, popísaného v kapitole 2.2.3, za podmienky, že výsledkom nebolo optimálne riešenie. Zlepšenie získaného riešenia algoritmom pMBBDual realizujem pomocou návrhu vhodnej heuristickej metódy alebo formou obmedzenia vetvenia v metóde vetiev a hraníc. Tieto spôsoby vylepšenia však vychádzajú z predpokladu, že je k dispozícii nejaké už získané riešenie nazývané aj ako najlepšie získané prípustné riešenie, ktoré však nezodpovedá optimálnemu riešeniu, príp. neprípustného riešenia s najmenšou mierou neprípustnosti. Riešenie získané algoritmom pMBBDual je najlepším získaným prípustným riešením riešenej úlohy, kedy počet umiestnených stredísk je menší alebo rovný ako p a hodnota účelovej funkcie je najmenšia, aká mohla byť týmto algoritmom získaná. Na druhej strane, riešenie považujeme za neprípustné riešenie s najmenšou mierou neprípustnosti, ak počet umiestnených stredísk je

väčší ako p a je najbližšie k tejto hodnote p , pričom hodnota účelovej funkcie tohto riešenia je najmenšia, aká mohla byť algoritmom pMBBDual získaná.

4.1.1 Heuristické vylepšenia riešenia

Prvým spomínaným spôsobom vylepšenia je dodanie čo najlepšej kombinácie stredísk obsluhy do počtu p nejakou heuristickou metódou. Navrhol som tri vylepšovacie heuristiky s rôznymi stratégiami výberu stredísk obsluhy [10]. V nižšie popísaných vylepšovacích heuristikách, v ktorých sa mnohokrát prepočítava hodnota získaného riešenia, využijem mikrooptimalizačný výpočtový postup, ktorý je popísaný v zdroji [29].

Variant V1 – Vkladacia heuristika so stratégiou najlepší vhodný

Princíp vkladacej heuristiky so stratégiou najlepší vhodný je založený na predpoklade, že je k dispozícii najlepšie získané prípustné riešenie, ktoré pozostáva z menšieho počtu umiestnených stredísk ako p . Nech počet stredísk najlepšieho získaného prípustného riešenia algoritmom pMBBDual je r a p udáva maximálny možný počet možných umiestnení stredísk. Tu hľadám stredisko zo všetkých nezaradených stredísk s najväčším zlepšením hodnoty účelovej funkcie doterajšieho riešenia k novému riešeniu s $r+1$ počtom stredísk. Nájdene stredisko, ktoré najviac zlepší hodnotu riešenia, nazývané najlepším vhodným, zaradím do riešenia. Proces hľadania a vkladania stredísk opakujem pokiaľ počet umiestnených stredísk nie je rovný počtu p , kedy vkladacia heuristika skončí. Počet pridaných stredísk je teda rovný rozdielu $(p-r)$.

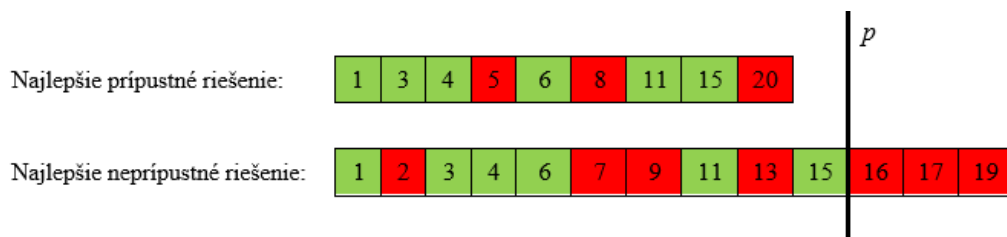
Variant V2 – Výmenná heuristika 1-1 so stratégiou prvý vhodný

Princíp výmennej heuristiky so stratégiou prvý vhodný je založený na predpoklade, že je k dispozícii neprípustné riešenie s najmenšou mierou neprípustnosti. Neprípustné riešenie s najmenšou mierou neprípustnosti predstavuje riešenie, kde počet umiestnených stredísk je väčší ako p , pričom hodnota účelovej funkcie tohto riešenia je najmenšia, aká mohla byť algoritmom pMBBDual získaná. Hľadám práve p stredísk zo stredísk umiestnených v získanom neprípustnom riešení, ktoré poskytujú najlepšiu hodnotu prípustného riešenia. Vložím p stredísk z tých, ktoré boli obsiahnuté v neprípustnom riešení s najmenšou mierou neprípustnosti získané algoritmom pMBBDual a vytvorím prípustné riešenie. Toto prípustné riešenie sa snažím vylepšiť 1-1 výmennou stredísk z prípustného riešenia so strediskami, ktoré neboli zaradené do tohto prípustného riešenia, ale boli v neprípustnom riešení

s najmenšou mierou neprípustnosti. Využívam stratégiu prvý vhodný, čo znamená, že výmena nastane vždy, ak nájdem prvé dve strediská, ktorých výmena zlepší riešenie. Výmenná 1-1 heuristika so stratégiou prvý vhodný končí, ak žiadna výmena prvkov neprinesie zlepšenie alebo hodnota riešenia odpovedá dolnej hranici riešenia.

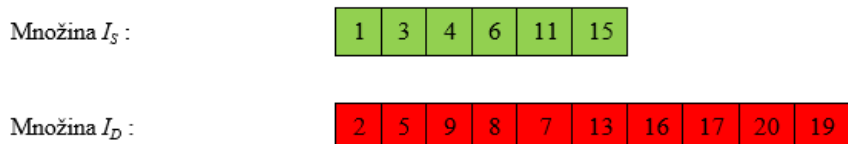
Variant V3 – Dvojfázová heuristika s prerozdelením stredísk

Princíp dvojfázovej heuristiky s prerozdelením stredísk demonštrujem na nasledujúcom príklade znázornenom na Obr. 2 až Obr. 5. Zoberme do úvahy, že algoritmom pMBBDual získam riešenie úlohy návrhu verejného obslužného systému, pričom toto riešenie neodpovedá optimálnemu riešeniu danej úlohy. Hodnota parametra p je rovná desať a počet umiestnených stredísk v najlepšom prípustnom riešení je deväť. Ako je znázornené na Obr. 2, potrebujem doplniť jedno stredisko do riešenia. Tento variant si vyžaduje, aby si algoritmus uchoval aj neprípustné riešenie s najmenšou mierou neprípustnosti získané algoritmom pMBBDual. Príklad neprípustného riešenia s najmenšou mierou neprípustnosti je znázornený na Obr. 2 s počtom vybudovaných stredísk trinásť. Čísla v Obr. 2 až Obr. 5 predstavujú identifikátory stredísk zo získaného riešenia. Tieto identifikátory označujú miesta, kde sa má umiestniť stredisko. Na Obr. 2 až Obr. 5 sú zelenou farbou označené strediská obsluhy, ktoré sa nachádzajú v oboch riešeniach, prípustnom aj neprípustnom riešení. Červenou farbou sú označené strediská obsluhy, ktoré sa nachádzajú iba v jednom z riešení



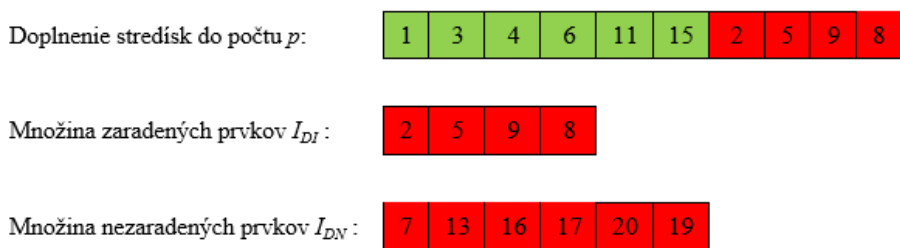
Obr. 2 Príklad najlepšieho prípustného a neprípustného riešenia

Ak mám k dispozícii prípustné aj neprípustné riešenie môžem realizovať prerozdelenie. Prerozdelenie stredísk spočíva vo vytvorení množín I_S a I_D z prvkov prípustného a neprípustného riešenia, ako je znázornené na Obr. 3. Množina I_S obsahuje strediská, ktoré sú spoločné v oboch riešeniach, znázornené v Obr. 2 až Obr. 5 zelenou farbou. Množina I_D obsahuje strediská, ktoré sa nachádzajú iba v jednom z riešení, znázornené v Obr. 2 až Obr. 5 červenou farbou.



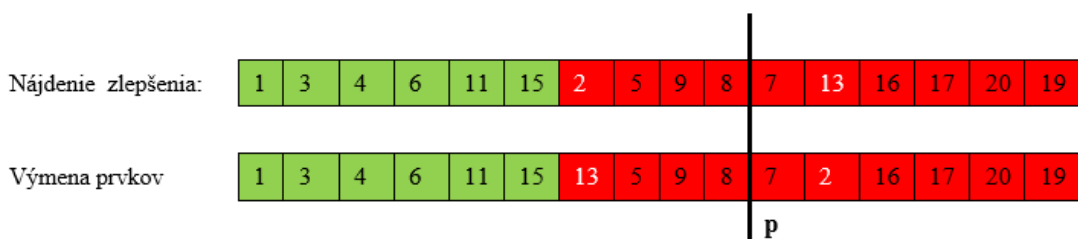
Obr. 3 Prerozdelenie stredísk na množiny I_S a I_D

Dvojfázová heuristika predstavuje kombináciu obmedzenej vsúvacej heuristiky s obmedzenou výmennou heuristikou. Prvou fázou je doplnenie stredísk z množiny I_S stredískami z množiny I_D do počtu p . Ako je znázornené na Obr. 4, toto doplnenie rozdeľuje množinu I_D na dve jej podmnožiny, množinu zaradených prvkov I_{DI} a množinu nezaradených prvkov I_{DN} . Táto prvá fáza odpovedá obmedzenej vsúvacej heuristike, keďže vsúvam prvky do riešenia a som obmedzený na výber prvkov iba s množiny I_D , nakoľko výber prvkov zo všetkých nezaradených stredísk je časovo náročný.



Obr. 4 Fáza doplnenia a rozdelenie množiny I_D na podmnožiny I_{DI} a I_{DN}

Po skončení prvej fázy nastáva druhá fáza - fáza výmeny, kedy pri zistení, že výmena je vhodná, sa dané prvky vymenia. Táto fáza odpovedá obmedzenej výmennej heuristike, obmedzenej keďže sa vymieňajú iba prvky z množiny I_D , pričom platí, že vymení sa iba prvok z množiny I_{DI} zaradený do riešenia za prvok z množiny I_{DN} nezaradený do riešenia (viď Obr. 5). Obmedzujem sa iba na výber nezaradeného prvku z množinu I_{DN} , nakoľko sa snažím zmenšiť dobu výpočtu a výber z celej množiny je časovo náročný.



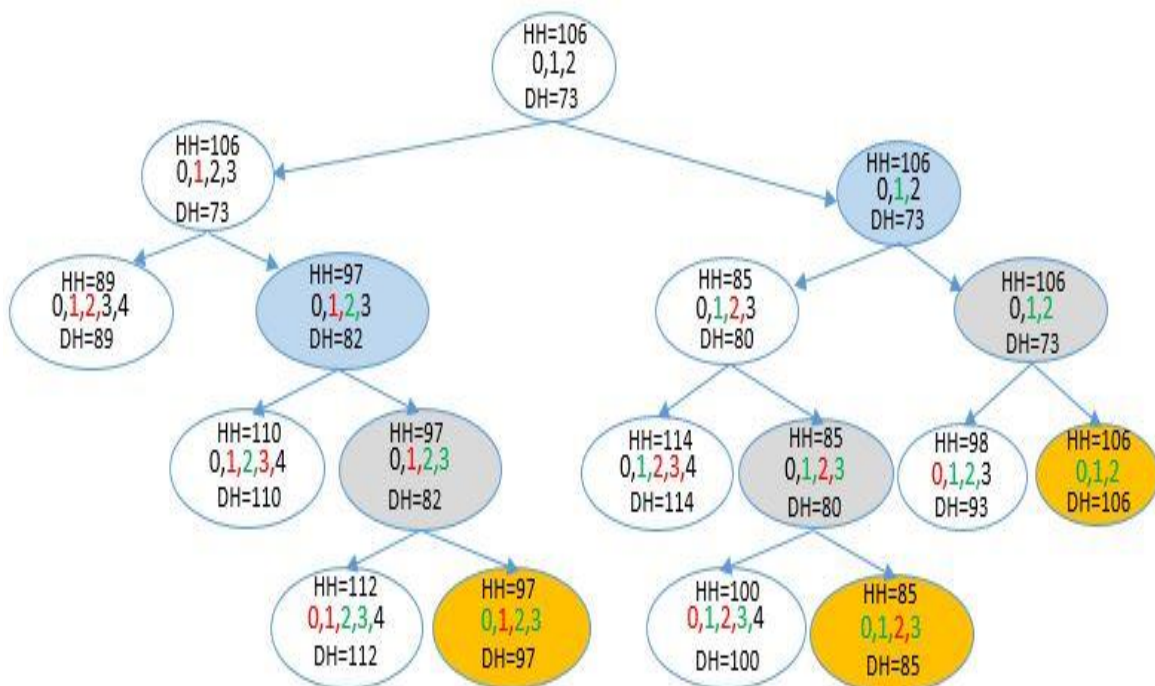
Obr. 5 Fáza výmeny prvkov pri zlepšení riešenia

Heuristika skončí, ak žiadna ďalšia výmena nezlepší riešenie. Pre vylepšenie riešenia je možné použiť aj iné typy heuristik.

4.1.2 Obmedzenie vetvenia v metóde vetiev a hraníc

Variant V4 – Erlenkotterov prístup s obmedzeným vetvením

Druhým spôsobom je možnosť obmedzenia vetvenia v metóde vetiev a hraníc pre zlepšenie riešenia získaného algoritmom pMBBDual. Vstupom do metódy vetiev a hraníc je najlepšie prípustné riešenie spolu so získanou hornou a dolnou hranicou riešenia ako aj hodnota Lagrangeovho multiplikátora pre neprípustne riešenie s najmenšou mierou neprípustnosti. Na základe Lagrangeovho multiplikátora dokážem získať neprípustné riešenie, ktoré sa vo vetvení snažím sprípustniť. Obmedzenie vetvenia spočíva vo fixácii najviac p stredísk na hodnotu jedna. Po fixácii p stredísk na hodnotu jedna, je vetvenie na hodnotu jedna v danej vetve ukončené, nakoľko už žiadne prípustné riešenie ďalej nezískam. Vetvím len fixáciou umiestňovacej premennej na hodnotu nula. Tento princíp demonštrujem na nasledovnom príklade znázornenom na Obr. 6.



Obr. 6 Demonštračný príklad s obmedzením vetvenia

Zoberme do úvahy úlohu s piatimi kandidátmi na stredisko a desiatimi zákazníkmi. Maximálny počet vybudovaných stredísk p je tri. Pre fixáciu na jedna je označenie

fixovaného prvku zelenou farbou, pre fixáciu na nula je označenie fixovaného prvku červenou farbou.

Postupujem ako v metóde vetiev a hraníc opísanej v kapitole 2.2.1. Na začiatku vyberiem z neprípustného riešenia prvých p stredísk, ktoré budú predstavovať riešenie pre koreň stromu riešenia a spracujem ho. Ak spracovaním vytvorím dva nové vrcholy, tie odpovedajú fixovaniu jedného vybraného strediska na hodnotu nula a na hodnotu jedna. Tento postup opakujem pre každý nespracovaný vrchol. Ak narazím na vrchol, kde počet fixovaných premenných na jedna je rovný p , vrchol vylúčim z ďalšieho prehl'adania a aktualizujem dolnú hranicu DH na hodnotu hornej hranice HH . Táto aktualizácia je znázornená zhodou DH a HH na Obr. 6 vo vrcholoch sfarbených oranžovou farbou. Na obrázku je farebne odlišnými vrcholmi demonštrované zvýšenie počtu fixovaných premenných na hodnotu jedna.

4.1.3 Predbežné experimenty

Na základe vylepšení algoritmu pMBBDual pre zlepšenie riešenia spomínanými spôsobmi som vytvoril štyri varianty, ktoré je potrebné otestovať a vyvodit' záver, ktorá z variant vylepšenia je najvhodnejšia. Kritériom pri testovaní je hodnota získaného riešenia aplikovaním jednotlivých variant a jej odchýlka od optimálneho riešenia. Dôležitým kritériom je aj čas trvania výpočtu pre jednotlivé varianty. Experimenty som realizoval na vybranej časti cestnej siete Slovenska, kde mohutnosť množiny I bola v rozmedzí 100 – 1000 kandidátov na stredisko a mohutnosť množiny J bola v rozmedzí 100 – 2916. Hodnota parametra p sa pohybovala od hodnoty 5 do 1000. Všetky experimenty boli realizované na PC s procesorom i7 Q720 a pamäťou 8GB RAM. Experimenty boli rozdelené podľa veľkosti na malé experimenty (100x100), stredné experimenty (315x315) a veľké experimenty (1000x2916). V predbežných experimentoch som porovnával výsledky získané univerzálnym IP solverom XPRESS-IVE [50], [51] s výsledkami algoritmu pMBBDual bez vylepšenia riešenia a s výsledkami algoritmu pMBBDual s vylepšením jednotlivými variantmi. Pri vylepšení bol vždy aplikovaný jeden variant. Čas výpočtu pre jednotlivé varianty (V1, V2, V3, V4) predstavuje súhrnný čas algoritmu pMBBDual a času vylepšenia práve realizovaným variantom.

Experimenty o rozsahu 100x100 sú realizované na úlohe p-mediánu definovanej na vybraných 100 obciach Banskobystrického kraja. Súbor, ktoré prislúchajú testovacej úlohe BB100x100 sú uvedené v prílohe 1.

Tab. 4 Vylepšenia algoritmu pMBBDual a ich porovnanie s XPRESS-IVE na úlohe BB100x100

p	XPRESS-IVE		pMBBDual				V1		V2		V3		V4		
	t[s]	OF	t[s]	OF _{UP}	OF _{RP}	NoF	t[s]	OF	t[s]	OF	t[s]	OF	t[s]	OF	
5	0,78	2391	0,18	2391	2391	2391	5	0,18	2391	0,18	2391	0,18	2391	0,18	2391
10	0,76	1275	0,05	1275	1275	1275	10	0,05	1275	0,05	1275	0,05	1275	0,05	1275
15	0,72	923	0,03	923	923	923	15	0,03	923	0,03	923	0,03	923	0,03	923
20	0,73	745	0,03	745	745	745	20	0,03	745	0,03	745	0,03	745	0,03	745
25	0,7	616	0,03	616	616	616	25	0,03	616	0,03	616	0,03	616	0,03	616
30	0,73	521	0,03	521	521	521	30	0,03	521	0,03	521	0,03	521	0,03	521
35	0,75	441	0,03	456	441	34	0,05	441	0,05	443	0,04	442	5,11	441	
40	0,75	373	0,03	385	373	39	0,04	373	0,06	373	0,04	373	263,91	373	
45	0,89	315	0,02	315	315	45	0,02	315	0,02	315	0,02	315	0,02	315	
50	0,86	263	0,02	315	260	45	0,04	263	0,04	263	0,04	263	0,04	263	
55	0,75	215	0,03	233	215	53	0,03	215	0,04	216	0,03	215	4,01	215	
60	0,75	176	0,02	198	174	57	0,03	176	0,03	176	0,03	176	0,04	176	
65	0,75	142	0,02	142	142	65	0,02	142	0,02	142	0,02	142	0,02	142	
70	0,73	112	0,03	142	112	65	0,03	112	0,04	115	0,04	114	6,29	112	
75	0,75	87	0,03	97	87	73	0,03	87	0,04	89	0,04	87	3600*	87	
80	0,75	64	0,02	97	62	73	0,04	64	0,03	64	0,03	64	0,2	64	
85	0,73	44	0,02	60	44	81	0,03	44	0,03	44	0,03	44	3600*	44	
90	0,73	25	0,02	28	25	89	0,03	25	0,03	25	0,03	25	489,17	25	
95	0,73	10	0,03	10	10	95	0,03	10	0,03	10	0,03	10	0,03	10	
100	0,39	0	0	0	0	100	0	0	0	0	0	0	0	0	

¹ stĺpec p udáva hodnotu maximálneho počtu stredísk

² stĺpec t[s] udáva čas trvania výpočtu pre jednotlivé metódy

³ stĺpec OF udáva hodnotu účelovej funkcie pre jednotlivé metódy

⁴ stĺpec OF_{UP} – hodnotu riešenia nerelaxovanej (pôvodnej) úlohy

⁵ stĺpec OF_{RP} – hodnotu riešenia LP-relaxácie – dolná hranica optimálneho riešenia pôvodnej úlohy

⁶ stĺpec NoF udáva počet vybudovaných stredísk

⁷ stĺpce XPRESS-IVE udávajú hodnoty parametrov pre optimálne riešenie získané pomocou univerzálneho solvera

⁸ stĺpce pMBBDual udávajú hodnoty výstupných parametrov riešenia získaného algoritmom pMBBDual bez vylepšenia

⁹ stĺpce V1, V2, V3, V4 udávajú hodnoty výstupných parametrov riešenia algoritmom pMBBDual s vylepšením pomocou variant V1, V2, V3 alebo V4

¹⁰ oranžová farba – hodnota optimálneho riešenia

¹¹ zelená farba – najlepší výpočtový čas pri získaní optimálneho riešenia

¹² sivá farba - algoritmus pMBBDual poskytol optimálne riešenie, tým pádom sme nemuseli nič vylepšovať

¹³ červená farba - algoritmus bol predčasne ukončený po 1 hodine výpočtu (3600*)

Na základe experimentov z Tab. 4 pri kritériu získania optimálneho riešenia sú najvhodnejšie varianty V1 a V4, kedy som vo všetkých prípadoch získal optimálne riešenie. Ak však zoberiem do úvahy čas trvania výpočtu sú vhodnejšie varianty V1, V2 a V3 oproti V4. Poskytnutie optimálneho riešenia variantom V4 stojí v mnohých prípadoch dlhý výpočtový čas vzhľadom na rozsahovo malý typ úlohy. Na základe týchto sledovaných parametrov je ako najvhodnejší variant V1 z hľadiska časového tak aj z hľadiska získania optimálneho riešenia. Varianty V2 a V3 napriek porovnateľnému výpočtovému času, niekedy aj lepšiemu, neposkytli získanie optimálneho riešenia v toľkých prípadoch ako

variant V1. Výber najvhodnejšieho variantu na základe vykonaných experimentov nie je jednoznačný. Preto otestujem varianty na rozsahovo väčších úlohách, kde sa ukáže ako obstoja jednotlivé prístupy v súťaži o najvhodnejší variant.

Experimenty o rozsahu 315x315 boli realizované na úlohe p-mediánu definovanej na množine všetkých obcí Žilinského kraja. Súborny, ktoré prislúchajú testovacej úlohe ZA315x315, sú uvedené v prílohe 1.

Tab. 5 Vylepšenia algoritmu pMBBDual a ich porovnanie s XPRESS-IVE na úlohe ZA315x315

p	XPRESS-IVE		pMBBDual				V1		V2		V3		V4	
	OF _{ES}	t[s]	OF _{UP}	OF _{RP}	t[s]	NoF	OF _{V1}	t[s]	OF _{V2}	t[s]	OF _{V3}	t[s]	OF _{V4}	t[s]
15	2803	14,13	2803	2803	2,52	15	2803	2,52	2803	2,52	2803	2,52	2803	2,52
30	1832	18,74	1832	1832	1,78	30	1832	1,78	1832	1,78	1832	1,78	1832	1,78
45	1400	13,65	1420	1400	2,15	44	1400	2,48	1403	2,35	1400	2,28	1400	3445,0
60	1138	16,92	1183	1138	2,07	57	1139	2,35	1139	2,48	1138	2,18	1138	17,5
75	944	13,6	977	944	1,69	72	945	2,01	944	2,08	945	1,81	944	2,0
90	801	13,53	828	801	1,51	87	801	1,68	801	1,68	801	1,61	801	3600*
105	686	13,69	707	686	1,44	102	687	1,74	695	1,74	688	1,61	690	3600*
120	595	13,69	637	595	1,41	113	595	2,08	604	1,88	597	1,61	648	3600*
135	516	13,5	637	505	1,48	113	519	1,74	516	1,74	517	1,68	516	1,7
150	441	13,59	506	441	1,65	137	441	1,74	450	2,08	444	1,74	503	3600*
165	372	13,72	506	366	1,68	137	376	1,74	375	1,81	374	1,81	379	3600*
180	312	14,12	340	312	1,55	173	312	1,68	329	2,28	315	1,68	427	3600*
195	257	14,45	340	252	1,48	173	257	1,68	269	1,95	259	1,81	283	3600*
210	212	13,96	340	192	1,65	173	212	1,74	212	1,68	212	1,81	212	1,7
225	167	14,4	206	167	1,48	212	167	1,61	173	2,01	167	1,61	193	3600*
240	123	14,2	206	122	1,55	212	123	1,61	128	1,68	125	1,61	127	3600*
255	93	14,29	107	93	1,28	248	93	1,61	95	2,82	93	1,41	183	3600*
270	63	13,77	107	63	1,34	248	63	1,61	65	2,01	63	1,54	107	3600*
285	33	13,71	107	33	1,58	248	33	1,68	35	1,68	33	1,74	37	3600*
300	14	13,74	23	14	1,58	291	14	1,61	14	1,81	14	1,74	42	3600*

¹ stĺpec p udáva hodnotu maximálneho počtu stredísk

² stĺpec t[s] udáva čas trvania výpočtu pre jednotlivé metódy

³ stĺpec OF udáva hodnotu účelovej funkcie pre jednotlivé metódy

⁴ stĺpec OF_{UP} – hodnotu riešenia nerelaxovanej (pôvodnej) úlohy

⁵ stĺpec OF_{RP} – hodnotu riešenia LP-relaxácie – dolná hranica optimálneho riešenia pôvodnej úlohy

⁶ stĺpec NoF udáva počet vybudovaných stredísk

⁷ stĺpce XPRESS-IVE udávajú hodnoty parametrov pre optimálne riešenie pomocou univerzálneho solvera

⁸ stĺpce pMBBDual udávajú hodnoty výstupných parametrov riešenia získaného algoritmom pMBBDual bez vylepšenia

⁹ stĺpce V1, V2, V3, V4 udávajú hodnoty výstupných parametrov riešenia algoritmom pMBBDual s vylepšením pomocou variant V1, V2, V3 alebo V4

¹⁰ oranžová farba – hodnota optimálneho riešenia

¹¹ zelená farba – najlepší výpočtový čas pri získaní optimálneho riešenia

¹² sivá farba - algoritmus pMBBDual poskytol optimálne riešenie, tým pádom sme nemuseli nič vylepšovať

¹³ červená farba - algoritmus bol predčasne ukončený po 1 hodine výpočtu (3600*)

Na základe vykonaných experimentov uvedených v Tab. 5 môžem potvrdiť predchádzajúci záver, že variant V4 je časovo náročný, čo spôsobuje takmer úplná enumerácia vetviaceho algoritmu. Preto je variant V4 nevhodný pre vylepšenie algoritmu

pMBBDual. Varianty V1, V2, V3 sú vo väčšine prípadov časovo menej náročnejšie ako V4, no rozdiely medzi týmito variantami V1, V2 a V3 nie sú až tak výrazne, aby som mohol na základe časového hľadiska nejaký ďalší vylúčiť. Ak zoberiem hľadisko získania optimálneho riešenia, tak variant V1 poskytuje v najväčšom počte prípadov optimálne riešenie, preto je považovaný za najvhodnejší variant vylepšenia riešenia algoritmu pMBBDual. Aby som našiel väčšie časové rozdiely medzi jednotlivými variantmi, tak som otestoval varianty V1, V2 a V3 na rozsahovo ešte väčších úlohách. Nakoľko variant V4 bol časovo náročný, tak som ho vylúčil z ďalších experimentov.

Experimenty zhrnuté v Tab. 6 o rozsahu 1000x2916 sú realizované na úlohe váženého p-mediánu definovanej na množine 1000 najväčších obcí Slovenska a množine všetkých obcí Slovenska. Súbory, ktoré prislúchajú testovacej úlohe SR1000x2916 sú uvedené v prílohe 1.

Tab. 6 Porovnanie zlepšenia riešenia algoritmu pMBBDual na úlohe SR1000x2916

P	pMBBDual				V1		V2		V3	
	t[s]	OF _{UP}	OF _{RP}	NoF	t[s]	OF	t[s]	OF	t[s]	OF
50	100,8	535448	535448	50	100,8	535448	100,8	535448	100,8	535448
100	35,4	321869	321869	100	35,4	321869	35,4	321869	35,4	321869
150	65,7	248373	248373	150	65,7	248373	65,7	248373	65,7	248373
200	113,0	207011	207011	200	113,0	207011	113,0	207011	113,0	207011
250	104,5	178645	178645	250	104,5	178645	104,5	178645	104,5	178645
300	132,4	160076	159044	297	143,0	159044	145,0	159044	142,4	159044
350	68,2	144076	144076	350	68,2	144076	68,2	144076	68,2	144076
400	68,4	132427	132427	400	68,4	132427	68,4	132427	68,4	132427
450	54,9	123775	123615	449	64,8	123615	64,5	123615	63,7	123615
500	68,2	116650	116388	498	72,6	116388	71,5	116388	70,7	116388
550	50,3	110558	110450	549	51,7	110450	56,0	110450	51,9	110450
600	46,0	105418	105418	600	46,0	105418	46,0	105418	46,0	105418
650	46,1	101209	101209	650	46,1	101209	46,1	101209	46,1	101209
700	40,0	98 229	97869	694	41,5	97869	42,1	97869	42,0	97869
750	38,6	95308	95158	747	40,0	95158	51,2	95158	38,8	95158
800	38,0	93000	92832	796	38,5	92832	57,0	92832	38,5	92832
850	38,4	90976	90906	848	39,1	90906	51,8	90907	38,6	90906
900	38,1	89648	89340	889	40,0	89340	45,6	89340	38,7	89340
950	39,4	88264	88164	945	39,5	88164	78,7	88164	38,5	88164
1000	0,3	87427	87427	1000	0,3	87427	0,3	87427	0,3	87427

¹ stĺpec *p* udáva hodnotu maximálneho počtu stredísk

² stĺpec *t[s]* udáva čas trvania výpočtu pre jednotlivé metódy

³ stĺpec *OF* udáva hodnotu účelovej funkcie pre jednotlivé metódy

⁴ stĺpec *OF_{UP}* – hodnotu riešenia nerelaxovanej (pôvodnej) úlohy

⁵ stĺpec *OF_{RP}* – hodnotu riešenia LP-relaxácie – dolná hranica optimálneho riešenia pôvodnej úlohy

⁶ stĺpce *pMBBDual* udávajú hodnoty výstupných parametrov riešenia získaného algoritmom pMBBDual bez vylepšenia

⁷ stĺpce *V1, V2, V3* udávajú hodnoty výstupných parametrov riešenia algoritmom pMBBDual s vylepšením pomocou variant V1, V2 alebo V3

⁸ oranžová farba – získaná hodnota optimálneho riešenia

⁹ zelená farba – najlepší výpočtový čas pri získaní optimálneho riešenia

¹⁰ sivá farba - algoritmus pMBBDual poskytol optimálne riešenie, tým pádom sme nemuseli nič vylepšovať

Na základe experimentov v Tab. 6 som pre zvolené hodnoty p vždy získal optimálne riešenie, nakoľko hodnota optimálneho riešenia odpovedala hodnote dolnej hranice riešenia. Vzhľadom na čas trvania výpočtu toto vylepšenie riešenia trvalo len zopár sekúnd, čím môžem varianty V1, V2 a V3 považovať za veľmi dobré vylepšenie riešenia získaného algoritmom pMBBDual. Ako časovo najlepší sa ukázal variant V3 vo väčšine prípadov. No na druhej strane časové rozdiely medzi variantmi V3 a V1 nie sú nejak veľké, na rozdiel od variantu V2.

4.1.4 Celkové zhodnotenie experimentov

Z realizovaných experimentov môžem vyvodit' záver, že variant V1 poskytuje získanie veľmi dobrého riešenia, hoci nezaručuje vo všetkých prípadoch získanie optimálneho riešenia. Treba však zdôrazniť, že variant V1 predstavuje heuristický prístup k riešeniu, čo znamená, že získané riešenie nemusí byť optimálne.

Na základe experimentov som zistil, že variant V2 dokáže vylepšiť algoritmus a je výsledkovo porovnateľný s variantom V1. No ukázalo sa aj, že variant V2 môže byť časovo náročnejší a optimálne riešenie nebolo dosiahnuté v takom počte ako s variantom V1, kde som väčšinou optimálne riešenie dosiahol.

Na základe experimentov som zistil, že variant V3 dokáže vylepšiť algoritmus a je výsledkovo porovnateľný s variantom V1 z časového hľadiska ako aj pri získaní optimálneho riešenia. Variant V3 dosahoval vo väčšine lepšie výpočtové časy ako variant V1. Z hľadiska početnosti získania optimálneho riešenia bol však horší ako V1, avšak lepší v porovnaní s variantom V2.

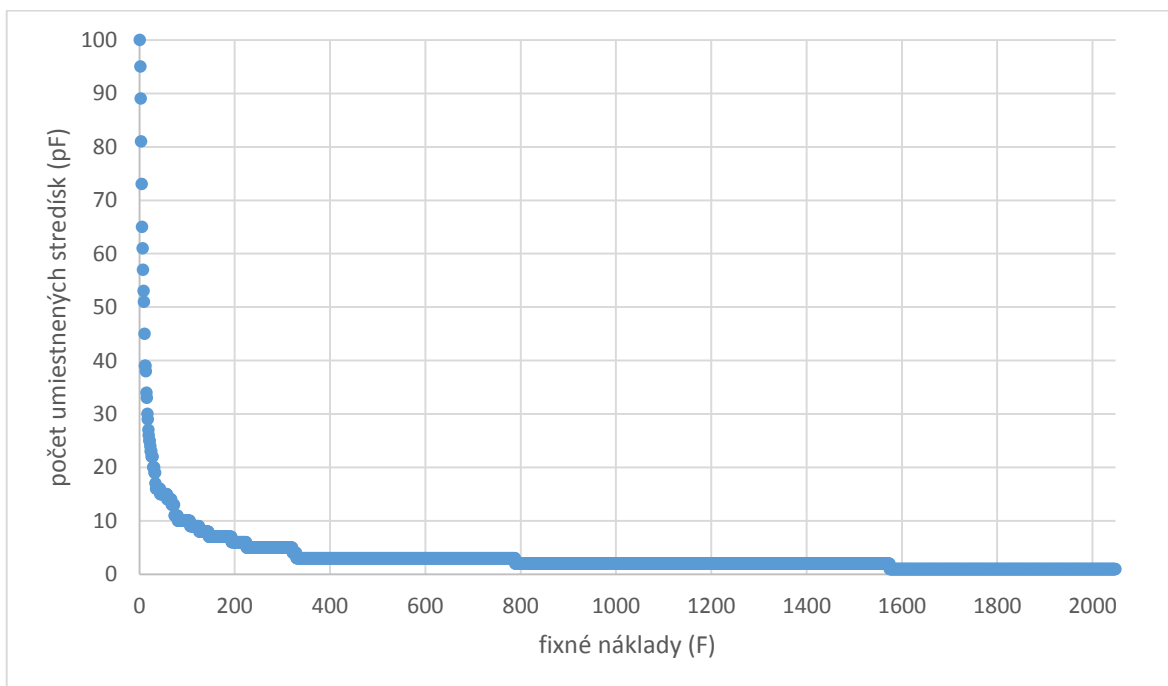
Variant V4, ktorý obmedzuje vetvenie, nie je vhodný pre vylepšenie algoritmu pMBBDual, nakoľko má tendenciu exponenciálne rásť čas výpočtu spôsobený skoro úplnou enumeráciou vetvenia v metóde vetiev a hraníc.

Najvhodnejším variantom pre vylepšenie riešenia získaného algoritmom pMBBDual je variant V1, teda vkladacia heuristika so stratégiou najlepší vhodný.

Keďže algoritmus pMBBDual neposkytol vždy optimálne riešenie, vedie to k hypotéze či riešenie úlohy návrhu verejného obslužného systému priamou metódou založenou na zovšeobecnení Erlenkotterovho prístupu bez využitia Lagrangeovej relaxácie neprinesie lepšie výsledky.

4.2 Vylepšenie Erlenkotterovho prístupu s Lagrangeovou relaxáciou

Rozšírenie Erlenkotterovho prístupu popísané v kapitole 2.2.3 umožnilo riešiť úlohu návrhu verejného obslužného systému transformáciou kapacitne neobmedzenej umiestňovacej úlohy pomocou Lagrangeovej relaxácie, ktorá viedla na umiestňovaciú úlohu s obmedzeným počtom vybudovaných stredísk. Týmto prístupom je realizované iteratívne vyhľadávanie Lagrangeovho multiplikátora Lg pomocou metódy bisekcie. Pre každú hodnotu Lagrangeovho multiplikátora získanú algoritmom bisekcie sa vykonáva algoritmus BBDual popísaný v kapitole 2.2.1, ktorým získam počet umiestnených stredísk pri navýšených fixných nákladoch všetkých kandidátov na stredisko o hodnotu Lagrangeového multiplikátora. Bisekciou sa pokúšam nájsť najlepšie nastavenie Lagrangeovho multiplikátora, ktorému odpovedá najlepšie získané prípustné riešenie. Získanie najlepšieho nastavenia Lagrangeovho multiplikátora pomocou algoritmu bisekcie môže viesť k vykonaniu veľkého počtu iterácií, čo negatívne vplyva na výpočtový čas. Získanie najlepšieho nastavenia Lagrangeovho multiplikátora je možné získať aj odhadom nákladov distribučného systému. Pri zadaných nulových fixných nákladoch je hodnota odhadu požadovaných nákladov pre získanie najlepšieho prípustného riešenia rovná hodnote Lagrangeovho multiplikátora, ktorý určuje navýšenie pôvodne nulových fixných nákladov.



Obr. 7 Demonštračný príklad závislosti počtu vybudovaných stredísk na fixných nákladoch

Na základe analýzy závislosti medzi počtom vybudovaných stredísk p_F a požadovanom nastavení fixných nákladov, čo je demonštrované na Obr. 7, sa pokúsim odhadnúť náklady distribučného systému f pre získanie požadovaného počtu umiestnených stredísk p . Návrhom iteratívneho prístupu s odhadom nákladov na získanie požadovaného počtu umiestnených stredísk p sa snažím dosiahnuť najlepšie nastavenie Lagrangeovho multiplikátora s menším počtom iterácií algoritmu pMBBDual ako som získal pôvodne bisekčným algoritmom.

4.2.1 Odhad nákladov distribučného systému

Všeobecne môžem náklady distribučného systému $N(p)$ odhadnúť ako súčet súčinov fixných nákladov f s počtom stredísk p a konštanty c s polomerom r obsluhovaných oblastí nasledovne:

$$N(p) = fp + cr \quad (4.2.01)$$

Pri rovnomernom rozložení požiadaviek budú náklady úmerné strednej vzdialenosti, ktorá je za predpokladu kruhových obsluhovaných oblastí priamo úmerná polomeru r . Vzťah medzi polomerom r a počtom stredísk p je možné vyjadriť cez obsluhovanú plochu S .

$$S = p \pi r^2 \rightarrow r = \sqrt{\frac{S}{\pi p}} \quad (4.2.02)$$

Upravím vzťah (4.2.01) nahradením polomeru r vzťahom (4.2.02):

$$N(p) = fp + c \sqrt{\frac{S}{\pi p}} = fp + c \sqrt{\frac{S}{\pi}} p^{-\frac{1}{2}} \quad (4.2.03)$$

Vzťah medzi fixnými nákladmi f a počtom p získam na základe prvej derivácie (4.2.04), druhej derivácie (4.2.05) a určenia stacionárnych bodov (4.2.06).

$$\frac{d N(p)}{d p} = f - \frac{c}{2} \sqrt{\frac{S}{\pi}} p^{-\frac{3}{2}} \quad (4.2.04)$$

$$\frac{d^2 N(p)}{d p^2} = \frac{3 c}{4} \sqrt{\frac{S}{\pi}} p^{-\frac{5}{2}} > 0 \quad (4.2.05)$$

$$\frac{d N(p)}{d p} = 0 \rightarrow f = \frac{c}{2} \sqrt{\frac{S}{\pi}} p^{-\frac{3}{2}} \quad (4.2.06)$$

Nahradením konštant vo vzťahu (4.2.06) konštantou K zjednoduším vzťah nasledovne:

$$K = \frac{c}{2} \sqrt{\frac{S}{\pi}} \rightarrow p = \frac{K}{f^{\frac{2}{3}}} \quad (4.2.07)$$

Vychádzajme z hypotézy, že ak fixné náklady f sú rovné nule dostaneme počet stredísk p rovný m , kde m je počet všetkých kandidátov na umiestnenie stredísk. Dosadenie nulových nákladov f do vzťahu (4.2.07) predstavuje delenie nulou, čo požadovaný počet m neprinesie. Preto upravím vzťah (4.2.07), kde δ bude predstavovať konštantu, ktorá zabezpečí získanie počtu m stredísk pre hodnotu f rovnú nule. Hodnotu δ môžem vyjadriť nasledovne:

$$p = \frac{K}{\delta + f^{\frac{2}{3}}} \rightarrow m = \frac{K}{\delta} \rightarrow \delta = \frac{K}{m} \quad (4.2.08)$$

Nahradením δ vo vzťahu (4.2.08) vyjadrim závislosť počtu p na nákladoch f nasledovne:

$$p = \frac{K}{\frac{K}{m} + f^{\frac{2}{3}}} \quad (4.2.09)$$

Vo všeobecnosti zo vzťahu (4.2.09) vyplýva, že pre ľubovoľné náklady F môžem získať počet stredísk p_F vzťahom (4.2.10). Konštantu K je potom zo vzťahu možné vyjadriť nasledovne:

$$p_F = \frac{K}{\frac{K}{m} + F^{\frac{2}{3}}} \rightarrow K = \frac{p_F m}{m - p_F} F^{\frac{2}{3}} \quad (4.2.10)$$

Zavedením substitúcie konštanty K do vzťahu (4.2.09) dostanem vzťah medzi požadovaným počtom stredísk p s hľadanou hodnotou fixných nákladov f a počtom stredísk p_F s fixnými nákladmi F :

$$p = \frac{m}{1 + \frac{m - p_F}{p_F} f^{\frac{2}{3}} F^{\frac{2}{3}}} \quad (4.2.11)$$

Hrubý odhad neznámych nákladov pre požadovaný počet stredísk p získam vyjadrením f zo vzťahu (4.2.11) nasledovne:

$$f = \left(\frac{m - p}{m - p_F} \frac{p_F}{p} \right)^{\frac{3}{2}} F \quad (4.2.12)$$

Na základe uvedených hypotéz som získal hrubý odhad nákladov f , čo znamená, že ani odhadnutý exponent $3/2$ nemusí byť najvhodnejší. Vo všeobecnosti môžem vzťah (4.2.12) zapísať pomocou exponentu α nasledovne:

$$f = \left(\frac{m - p}{m - p_F} \frac{p_F}{p} \right)^{\alpha} F \quad (4.2.13)$$

Experimentálne sa pokúsim získať hodnotu exponenta α , ktorý by zabezpečil najvhodnejší odhad hodnoty nákladov f . Pre získanie odhadu f je potrebné okrem hodnoty exponenta vedieť resp. spočítať pre nejaké zvolené náklady F počet stredísk p_F .

4.2.2 Iteratívny prístup s odhadom Lagrangeovho multiplikátora

Získaný hrubý odhad predstavuje nastavenie Lagrangeovho multiplikátora v jednotlivých iteráciách za podmienky nulových počiatkových fixných nákladov F_0 . Potom

najlepšie nastavenie fixných nákladov F (Lagrangeovho multiplikátora), ktorému prislúcha najlepšie získané prípustné riešenie, je získané nasledovným algoritmom:

0. Zvoľte „ad hoc“ nejaké fixné náklady F , ktorých hodnota bude dostatočne veľká alebo rovná hornému odhadu F , pre ktoré je hodnota p_F rovná jednej, hodnotu exponenta α a r predstavujúce koeficient delenia pre intervalové vyhľadávanie.
1. Nastavte náklady $f_{mn} = 0$ a vypočítajte počet stredísk p_{mn} pomocou algoritmu BBDual. Nastavte $f_{mx} = F$, $p_{mx} = p_F$
2. Pre požadované p vypočítajte f podľa vzťahu (4.2.13).
3. Ak platí, že $f_{mn} \leq f \leq f_{mx}$ nastavte hodnotu nového $F = f$, inak nastavte hodnotu F podľa vzťahu:

$$F = \max\{1, (f_{mx} - f_{mn})/r\} \quad (4.2.14)$$

4. Pre F spočítajte algoritmom BBDual novú hodnotu p_F .
5. Ak $p_F > p$, položte $f_{mn} = F$, $p_{mn} = p_F$.
6. Ak $p_F < p$, položte $f_{mx} = F$, $p_{mx} = p_F$.
7. Ak $p_F = p$ alebo platí, že $f_{mx} - f_{mn} = 1$ skončte, inak späť na krok 2.

Takto skonštruovaný algoritmus pre získanie najlepšieho nastavenia fixných nákladov F zaručí získanie najlepšieho prípustného riešenia, ktoré sa rovná získanému riešeniu algoritmom pMBBDual (viď kapitola 2.2.3).

Musíme si uvedomiť, že v kroku 0 navrhnutého algoritmu je potrebné určiť hodnotu fixných nákladov F , ktorých hodnota bude dostatočne veľká alebo rovná hornému odhadu F , pre ktoré je hodnota p_F rovná jednej, hodnotu exponentu α a koeficient delenia pre intervalové vyhľadávanie r . Počiatočné fixné náklady F som nastavil na rovnakú hodnotu ako bola nastavená v bisekčnom algoritme pMBBDual. Koeficient delenia r som nastavil na hodnotu 8 a exponent α na fixnú hodnotu 1,5. Treba si však uvedomiť, že exponent α nemusí byť len fixný, ale môže sa aj počas výpočtu meniť na základe vyhodnotenia predošlého kroku. Vhodnosť zvolených nastavení parametrov α a r ako aj hypotézu o aktualizácii parametra α počas výpočtu sa budem snažiť experimentálne potvrdiť alebo vyvrátiť na benchmarkoch zo slovenskej cestnej siete a nájsť ich najlepšie nastavenie.

Experimenty s fixným nastavením parametra α

Experimenty pre získanie vhodnej hodnoty parametra α boli realizované na benchmarkoch z Trenčianskeho a Žilinského kraja na úlohách p -mediánu a váženého p -mediánu s hodnotou parametra $r = 8$, počiatočným nastavením fixných nákladov $F = 2048$ a fixným nastavením sledovaného parametra α počas celej doby výpočtu. Súborny, ktoré prislúchajú testovaním úlohám ZA315x315 a TN276x276 sú uvedené v prílohe 1.

Tab. 7 Výber vhodného nastavenia fixného parametra α pre rôzne typy úloh

α	Úlohy p -mediánu				Úlohy váženého p -mediánu			
	TN276x276		ZA315x315		TN276x276		ZA315x315	
	priemer	maximum	priemer	maximum	priemer	maximum	priemer	maximum
2	8,05	20	8,49	21	11,56	33	11,88	28
1,9	7,91	19	8,37	20	11,35	32	11,11	26
1,8	7,88	18	8,19	17	10,84	22	10,25	27
1,7	7,83	17	8,04	18	10,33	22	9,43	27
1,6	7,63	15	7,99	16	9,65	19	8,87	24
1,5	7,57	22	7,87	18	9,29	22	8,42	21
1,4	7,43	12	7,75	26	8,77	17	7,91	18
1,3	7,37	15	7,54	23	8,30	20	7,63	22
1,2	7,24	14	7,42	17	7,97	25	7,42	20
1,1	7,23	11	7,33	15	7,54	16	7,56	17
1	7,26	19	7,23	11	7,47	20	7,93	23
0,9	7,21	14	7,25	15	7,67	17	8,38	19
0,8	7,39	13	7,15	15	8,20	20	8,96	21
0,7	7,70	16	7,43	15	9,01	19	9,68	22
0,6	7,95	15	7,85	16	9,97	24	10,61	23

¹ stĺpec α udáva hodnotu exponenta vo vzťahu (4.2.13)

² stĺpec *priemer* udáva priemerný počet iterácií v testovanom benchmarku pre riešenú úlohu

³ stĺpec *maximum* udáva najväčší počet iterácií v testovanom benchmarku pre riešenú úlohu

⁴ stĺpec TN276x276 – testovací benchmark pre Trenčiansky kraj

⁵ stĺpec ZA315x315 – testovací benchmark pre Žilinský kraj

⁶ zelená farba – najlepšia získaná hodnota pre meraný parameter v danom stĺpci

⁷ oranžová farba – dobrá získaná hodnota pre meraný parameter v danom stĺpci

Výsledky vyhodnotenia realizovaných experimentov v Tab. 7 ukázali, že ako vhodné fixné nastavenie parametra α môžeme považovať hodnoty od 0,8 po 1,2. Ako najvhodnejšie nastavenie parametra α som po zvážení maximálneho počtu iterácií a priemerného počtu iterácií vybral hodnotu parametra α rovnú 1,1. Pre vybrané α rovné 1,1 som experimentálne otestoval aj vhodnosť nastavenia parametra r na rovnakých benchmarkoch a úlohách ako som testoval parameter α .

Výsledky vyhodnotenia realizovaných experimentov v Tab. 8 ukázali, že nastavenie parametra r na hodnotu 8 je najvhodnejšie, nakoľko som získal najmenšie priemerné počty iterácií vo väčšine testovaných prípadov. Aj keď pre hodnotu parametra $r = 7$ som získal

kvalitné štatistiky, hoci o malý rozdiel bol lepší výsledok pre parameter r nastavený na hodnotu 8.

Tab. 8 Výber vhodného nastavenia parametra r s fixným α pre rôzne typy úloh

r	Úlohy p -mediánu				Úlohy váženého p -mediánu			
	TN276x276		ZA315x315		TN276x276		ZA315x315	
	priemer	maximum	priemer	maximum	priemer	maximum	priemer	maximum
8	7,23	11	7,33	15	7,54	16	7,56	17
7	7,82	12	7,71	15	7,80	17	7,36	18
6	8,21	13	8,15	17	8,03	15	7,38	18
5	8,18	12	8,17	11	8,01	15	7,78	16
4	8,79	13	8,75	13	8,41	12	7,96	15
3	9,59	13	9,60	12	8,84	13	8,58	14
2	12,21	15	12,32	14	11,00	15	10,78	17

¹ stĺpec r udáva hodnotu r -sekcie vo vzťahu (4.2.14)

² stĺpec *priemer* udáva priemerný počet iterácií v testovanom benchmarku pre riešenie úlohu

³ stĺpec *maximum* udáva najväčší počet iterácií v testovanom benchmarku pre riešenie úlohu

⁴ stĺpec TN276x276 – testovací benchmark pre Trenčiansky kraj

⁵ stĺpec ZA315x315 – testovací benchmark pre Žilinský kraj

⁶ zelená farba – najlepšia získaná hodnota pre meraný parameter v danom stĺpci

⁷ oranžová farba – dobrá získaná hodnota pre meraný parameter v danom stĺpci

Experimenty s adaptívnym nastavením parametra α

Pod pojmom adaptívne nastavenie parametra α rozumieme meniacu sa hodnotu parametra α v závislosti od podielu hľadanej hodnoty p a hodnoty počtu umiestnených stredísk p_F v predchádzajúcom kroku. To sa prejavuje v navrhnutom algoritme doplnením kroku 5 a 6 o nasledovné nastavenie parametra α :

$$\alpha = \alpha \pm \frac{p}{p_F} \quad (4.2.15)$$

Pri testovaní vhodného nastavenia parametra α v prípade adaptívneho prístupu sa jedná o určenie počiatkovej hodnoty α , ktorá sa bude počas výpočtu aktualizovať.

Experimenty pre získanie vhodnej počiatkovej hodnoty parametra α boli realizované na benchmarkoch z Trenčianskeho a Žilinského kraja na úlohách p -mediánu a váženého p -mediánu s hodnotou parametra $r = 8$, počiatkovým nastavením fixných nákladov $F = 2048$ a adaptívnym nastavením sledovaného parametra α .

Výsledky vyhodnotenia realizovaných experimentov v Tab. 9 ukázali, že vzhľadom k aktualizácii hodnoty parametra α na základe predošlého kroku je jeho počiatkové nastavenie zanedbateľné, nakoľko rozdiely sú minimálne. Napriek tomu som sa rozhodol považovať nastavenie parametra $\alpha = 0,6$ za najvhodnejšie na základe vyhodnotenia priemerného počtu iterácií, kedy bol v priemere vykonaný najmenší počet iterácií vo väčšine prípadov.

Tab. 9 Vyber vhodného nastavenia adaptívneho parametra α pre rôzne typy úloh

α	Úlohy p -mediánu				Úlohy váženého p -mediánu			
	TN276x276		ZA315x315		TN276x276		ZA315x315	
	priemer	maximum	priemer	maximum	priemer	maximum	priemer	maximum
2	7,68	13	7,78	16	9,11	17	9,42	28
1,9	7,72	14	7,81	20	9,07	19	9,53	26
1,8	7,72	19	7,83	23	9,17	21	9,49	27
1,7	7,69	13	7,81	17	9,10	21	9,44	27
1,6	7,71	14	7,77	15	9,18	24	9,41	24
1,5	7,70	13	7,75	13	9,06	20	9,43	21
1,4	7,69	13	7,76	15	9,01	20	9,61	18
1,3	7,68	13	7,82	15	9,18	25	9,51	22
1,2	7,71	13	7,79	13	9,20	20	9,43	20
1,1	7,69	14	7,80	19	9,15	19	9,33	17
1	7,71	20	7,73	13	9,33	20	9,39	23
0,9	7,67	14	7,76	15	9,26	17	9,29	19
0,8	7,68	14	7,76	16	9,61	20	9,22	21
0,7	7,76	20	7,73	14	9,39	20	9,06	22
0,6	7,61	14	7,73	16	9,40	21	9,00	23

¹ stĺpec α udáva počiatočnú hodnotu exponenta vo vzťahu (4.2.13)

² stĺpec *priemer* udáva priemerný počet iterácií v testovanom benchmarku pre riešenú úlohu

³ stĺpec *maximum* udáva najväčší počet iterácií v testovanom benchmarku pre riešenú úlohu

⁴ stĺpec TN276x276 – testovací benchmark pre Trenčiansky kraj

⁵ stĺpec ZA315x315 – testovací benchmark pre Žilinský kraj

⁶ zelená farba – najlepšia získaná hodnota pre meraný parameter v danom stĺpci

⁷ oranžová farba – dobrá získaná hodnota pre meraný parameter v danom stĺpci

Nastavenie parametra $\alpha = 0,6$ v adaptívnom prístupe som využil pri zisťovaní vhodného nastavenia parametra r , ktorý bol testovaný na rovnakých benchmarkoch ako parameter α . Výsledky vyhodnotenia realizovaných experimentov v Tab. 10 ukázali, že hodnota parametra $r = 8$ je najvhodnejšia, keďže priemerný počet iterácií bol vo väčšine prípadov najnižší. Aj keď môžem povedať, že hodnoty r rovné 5 alebo 6 sú takisto vhodné pre navrhnutý iteratívny algoritmus, nakoľko rozdiely medzi r sú minimálne.

Tab. 10 Vyber vhodného nastavenia parametra r s adaptívnym α pre rôzne typy úloh

r	Úlohy p -mediánu				Úlohy váženého p -mediánu			
	TN276x276		ZA315x315		TN276x276		ZA315x315	
	priemer	maximum	priemer	maximum	priemer	maximum	priemer	maximum
8	7,61	14	7,73	16	9,40	21	9,00	23
7	8,34	17	8,24	13	9,34	22	9,63	19
6	8,32	14	8,51	13	9,74	18	9,23	15
5	8,30	13	8,43	17	9,33	15	9,34	17
4	9,12	14	9,16	14	9,78	15	9,32	17
3	9,87	16	9,90	15	10,03	17	9,86	16
2	12,43	16	12,42	14	11,46	15	11,37	15

¹ stĺpec r udáva hodnotu r -sekcíe vo vzťahu (4.2.14)

² stĺpec *priemer* udáva priemerný počet iterácií v testovanom benchmarku pre riešenú úlohu

³ stĺpec *maximum* udáva najväčší počet iterácií v testovanom benchmarku pre riešenú úlohu

⁴ stĺpec TN276x276 – testovací benchmark pre Trenčiansky kraj

⁵ stĺpec ZA315x315 – testovací benchmark pre Žilinský kraj

⁶ zelená farba – najlepšia získaná hodnota pre meraný parameter v danom stĺpci

⁷ oranžová farba – dobrá hodnota pre meraný parameter v danom stĺpci

4.2.3 Porovnanie navrhnutých iteratívnych algoritmov

Navrhnuté iteratívne algoritmy v kapitole 4.2.2 porovnam s algoritmom pMBBDual, ktorý je popísaný v kapitole 2.2.3, vo výpočtovom čase a počte vykonaných iterácií (viď príloha 2), priemerného počtu vykonaných iterácií a maximálneho počtu iterácií (viď Tab. 11) vyskytujúceho sa v testovaných benchmarkoch.

Tab. 11 Porovnanie iteratívnych algoritmov na benchmarkoch ZA315x315 a TN276x276

Získanie Lagrangeovho multiplikátora	Úlohy p -mediánu				Úlohy váženého p -mediánu			
	TN276x276		ZA315x315		TN276x276		ZA315x315	
	<i>ppI</i>	<i>maxI</i>	<i>ppI</i>	<i>maxI</i>	<i>ppI</i>	<i>maxI</i>	<i>ppI</i>	<i>maxI</i>
Bisekcia	13,56	14	13,10	14	12,61	14	11,95	14
Hrubý odhad s fixným α	7,23	11	7,33	15	7,54	16	7,56	17
Hrubý odhad s adaptívnym α	7,61	14	7,73	16	9,40	21	9,00	23

¹ stĺpec *Získanie Lagrangeovho multiplikátora* udáva realizovaný iteratívny prístup pre získanie Lagrangeovho multiplikátora

² stĺpec *ppI* udáva priemerný počet iterácií v testovanom benchmarku pre riešenie úlohu

³ stĺpec *maxI* udáva najväčší počet iterácií v testovanom benchmarku pre riešenie úlohu

⁴ stĺpec TN276x276 – testovací benchmark pre Trenčiansky kraj

⁵ stĺpec ZA315x315 – testovací benchmark pre Žilinský kraj

⁶ riadok Bisekcia udáva získanie Lagrangeovho multiplikátora bisekčným algoritmom

⁷ riadok Hrubý odhad s fixným α udáva získanie Lagrangeovho multiplikátora hrubým odhadom s nemeniacim sa parametrom α

⁸ riadok Hrubý odhad s adaptívnym α udáva získanie Lagrangeovho multiplikátora hrubým odhadom s meniacim sa parametrom α

⁹ zelená farba – najlepší maximálny počet vykonaných iterácií v testovanom benchmarku

¹⁰ oranžová farba – najlepší priemerný počet vykonaných iterácií v testovanom benchmarku

Vyhodnotením realizovaných experimentov sa ukázalo, že hrubým odhadom dokážem získať najlepšie prípustné riešenie za kratší výpočtový čas a s menším počtom iterácií vo väčšine prípadov (viď príloha 2). Metóda bisekcie zaručuje vykonanie maximálne 14 iterácií vo väčšine testovaných prípadov, no pri metóde využívajúcej hrubý odhad je občas počet iterácií *maxI* väčší. Avšak v priemernom počte vykonaných iterácií *ppI* je metóda využívajúca hrubý odhad oveľa lepšia či už s fixným alebo adaptívnym parametrom α . (viď Tab. 11) Pri porovnaní metód využívajúcich hrubý odhad Lagrangeovho multiplikátora s fixným a adaptívnym parametrom α vo väčšine prípadov je lepšie fixné nastavenie parametra α . To sa prejavilo kratším výpočtovým časom a menším počtom vykonaných iterácií (viď príloha 2). Vhodnosť fixného nastavenia parametra α potvrdzuje najmenší priemerný počet iterácií (viď Tab. 11). Na základe vykonaných experimentov som dospel k záveru, že vylepšenie algoritmu pMBBDual je možné nahradením algoritmu bisekcie na získanie Lagrangeovho multiplikátora algoritmom s hrubým odhadom Lagrangeovho multiplikátora popísaný v kapitole 4.2.2. Z algoritmov s hrubým odhadom Lagrangeovho

multiplikátora, ktoré boli porovnávané v Tab. 11, za vhodnejší považujem odhad s nastavenými parametrami $r = 8$ a fixným $\alpha = 1,1$.

Podrobné porovnanie základného a vylepšeného algoritmu pMBBDual s vylepšovacou heuristikou som realizoval na slovenských krajoch a na celej cestnej sieti Slovenska s vybranými hodnotami maximálneho počtu umiestnených stredísk p na základe predpisu (4.2.16), kde hodnota p_1 predstavuje prvú hodnotu p , q udáva rozmedzie medzi jednotlivými hodnotami p , k udáva krok, v ktorom počítam dané p a *počet* udáva počet vykonaných experimentov na testovanom benchmarku.

$$p = p_1 + q(k - 1) \quad k = 1, \dots, \text{počet} \quad (4.2.16)$$

Hodnoty parametrov p_1 , q a *počet*, ktoré sú uvedené v Tab. 12, boli zvolené tak, aby som obsiahol rozsah celého testovaného benchmarku a počet experimentov na danom benchmarku bol prispôsobený veľkosti riešenej úlohy pre možné vyvodenie záveru. V Tab. 12 sú podrobne popísané testované kraje spolu s ich označením, veľkosťou a nastavenými parametrami.

Tab. 12 Popis vykonaných experimentov na slovenskej cestnej sieti

Testovaný kraj	Benchmark	I	J	Testované p		
				p_1	q	počet
Bratislavský	BA87x87	87	87	5	5	15
Banskobystrický	BB515x515	515	515	15	20	25
Košický	KE460x460	460	460	15	20	23
Prešovský	PO664x664	664	664	15	20	33
Nitriansky	NR350x350	350	350	15	15	23
Trenčiansky	TN276x276	276	276	15	15	18
Tmavský	TT249x249	246	246	15	15	16
Žilinský	ZA315x315	315	315	15	15	20
Slovensko	CSR2916x2916	2916	2916	50	50	58

¹ stĺpec *Testovaný kraj* udáva kraj Slovenska, ktorého cestná sieť bola použitá

² stĺpec *Benchmark* udáva označenie dát pre testovanú úlohu

³ stĺpec *|I|* udáva mohutnosť množiny kandidátov na umiestnenie strediska

⁴ stĺpec *|J|* udáva mohutnosť množiny zákazníkov

⁵ stĺpec p_1 udáva prvé testované p

⁶ stĺpec q udáva rozmedzie medzi jednotlivými testovanými hodnotami p

⁷ stĺpec *počet* udáva počet testovaných hodnôt p pre najlepší výpočtový čas v testovanom benchmarku pre danú úlohu

Základný algoritmus pMBBDual s vylepšovacou heuristikou V1 je reprezentovaný algoritmom, ktorý je popísaný v kapitole 2.2.3, kde najlepšie nastavenia Lagrangeovho multiplikátora je získané bisekčným algoritmom. Vylepšenie získaného riešenia bolo dosiahnuté vkladacou heuristikou so stratégiou najlepši vhodný (viď kapitola 4.1.1).

Vylepšený algoritmus pMBBDual s vylepšovacou heuristikou V1 je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.2.2, kde metódou hrubého odhadu Lagrangeovho multiplikátora s fixnou hodnotou exponenta $\alpha = 1,1$ a parametrom delenia $r = 8$ hľadám najlepšie nastavenie Lagrangeovho multiplikátora. Vylepšenie získaného riešenia bolo dosiahnuté vkladacou heuristikou so stratégiou najlepší vhodný (viď kapitola 4.1.1).

Experimenty realizované na benchmarkoch slovenskej cestnej siete pre jednotlivé kraje Slovenska a celého Slovenska sú uvedené v prílohe 3. Súbory, ktoré prislúchajú testovacím úlohám sú uvedené v prílohe 1.

Vyhodnotenie experimentov je uvedené v Tab. 13 v podobe *PoR* počtu získaných optimálnych riešení k celkovému počtu vykonaných experimentov pre testovaný benchmark v percentách (4.2.17), percentuálnom vyjadrení relatívnej odchýlky *Gap* medzi získanou hodnotou riešenia *HH* a hodnotou optimálneho riešenia *ES* v priemere (4.2.18), priemerného výpočtového času *pt* z časov t jednotlivých experimentov (4.2.19), priemerný počet vykonaných iterácií *ppI* (4.2.20) a počtu predčasných ukončení algoritmu po uplynutí jednej hodiny výpočtového času *ppU*. Parameter *pocet* udáva počet vykonaných experimentov a p_{pocet} udáva posledné testované p v benchmarku získané vzťahom (4.2.16). Parameter *NoR* udáva počet optimálnych riešení pre testovaný benchmark a parameter *pI* udáva počet vykonaných iterácií v jednotlivých experimentoch.

$$PoR = \frac{NoR}{pocet} * 100 \quad (4.2.17)$$

$$Gap = \left(\sum_{p=p_1}^{p_{pocet}} \frac{HH_p - ES_p}{ES_p} * 100 \right) / pocet \quad (4.2.18)$$

$$pt = \left(\sum_{p=p_1}^{p_{pocet}} t_p \right) / pocet \quad (4.2.19)$$

$$ppI = \left(\sum_{p=p_1}^{p_{pocet}} pI_p \right) / pocet \quad (4.2.20)$$

Tab. 13 Porovnanie iteratívnych prístupov na slovenskej cestnej sieti

úloha	pMBBDual + V1			Základný			Vylepšený		
	Testovaný kraj	PoR [%]	Gap [%]	pt[s]	ppI	ppU	pt[s]	ppI	ppU
p-medián	Bratislavský	93	0,03	0,07	13,1	0	0,05	7,4	0
	Banskobystrický	72	0,04	38,99	13,7	0	11,49	5,9	0
	Košický	78	0,09	188,63	13,6	0	42,97	7,5	0
	Prešovský	76	0,07	696,23	13,6	1	383,15	7,5	0
	Nitriansky	83	0,04	42,12	13,5	0	6,60	7,4	0
	Trenčiansky	83	0,04	0,58	13,6	0	0,22	7,2	0
	Trnavský	81	0,07	5,63	13,6	0	2,03	7,4	0
	Žilinský	75	0,10	1,83	13,7	0	0,49	6,2	0
	Slovensko	-	-	3600,00	2,0	58	3600,00	2,0	58
vážený p-medián	Bratislavský	100	0,00	0,02	10,0	0	0,02	8,3	0
	Banskobystrický	100	0,00	0,96	12,6	0	0,23	6,0	0
	Košický	100	0,00	0,48	12,2	0	0,21	8,1	0
	Prešovský	97	0,00	4,43	13,4	0	1,17	7,8	0
	Nitriansky	100	0,00	0,31	12,8	0	0,16	8,4	0
	Trenčiansky	100	0,00	0,13	12,6	0	0,08	7,8	0
	Trnavský	100	0,00	0,11	12,3	0	0,09	8,9	0
	Žilinský	95	0,00	0,13	12,1	0	0,07	6,8	0
	Slovensko	93	0,00	635,47	13,5	0	189,98	6,3	0

¹ stĺpec *úloha* udáva typ riešenej úlohy

² stĺpec *Testovaný kraj* udáva kraj Slovenska, ktorého cestná sieť bola použitá

³ stĺpec *PoR[%]* udáva počet získaných optimálnych riešení k celkovému počtu vykonaných experimentov v percentách

⁴ stĺpec *Gap[%]* udáva percentuálne vyjadrenie relatívnej odchýlky medzi hodnotou *HH* a hodnotou *ES* v priemere

⁵ stĺpec *pt[s]* udáva priemerný čas výpočtu v sekundách

⁶ stĺpec *ppI* udáva priemerný počet iterácií

⁷ stĺpec *ppU* udáva počet predčasných ukončení algoritmu

⁸ stĺpce *Základný* udávajú získanie riešenia základným algoritmom pMBBDual s vylepšovacou heuristikou V1

⁹ stĺpce *Vylepšený* udávajú získanie riešenia vylepšeným algoritmom pMBBDual s vylepšovacou heuristikou V1

¹⁰ oranžová farba – získanie najlepšieho výpočtového času

Štatistiky v Tab. 13 potvrdili predpoklady, že využitie hrubého odhadu Lagrangeovho multiplikátora je vhodnejšie ako získavanie Lagrangeovho multiplikátora bisekčným algoritmom. Získanie Lagrangeovho multiplikátora hrubým odhadom umožňuje dosiahnuť rovnaké riešenie za menší výpočtový čas *pt* v priemere s menším počtom vykonaných iterácií *ppI* v porovnaní s bisekčným algoritmom. Veľký počet iterácií ovplyvňuje dĺžku výpočtového času, čo pri časovom obmedzení algoritmu môže spôsobiť jeho predčasné ukončenie. To je vidieť v počte predčasných ukončení algoritmu *ppU*, kde základný algoritmus pMBBDual bol v Prešovskom kraji raz predčasne ukončený. Avšak ani jedným prístupom sa nepodarilo spočítať celé Slovensko, kde sa negatívne prejavila iteratívnosť, kedy už pri hodnote Lagrangeovho multiplikátora $F = 2048$ algoritmus nedokázal poskytnúť dobré riešenie za požadovaný čas.

4.3 Zovšeobecnený Erlenkotterov prístup

Erlenkotterov prístup k riešeniu kapacitne neobmedzenej umiestňovacej úlohy (UFLP), ktorý som popísal v kapitole 2.2.1, je založený na poznatkoch z teórie duality. Využíva vzťah medzi primárnym a duálnym riešením, kde na základe získania dobrého prípustného duálneho riešenia pre špecifikovanú podúlohu sa snaží skonštruovať dobré primárne riešenie. Erlenkotter navrhol algoritmus, kde tento prístup použil pre spracovanie každého vrcholu stromu riešenia v metóde vetiev a hraníc. Dolnú hranicu vrcholu v metóde vetiev a hraníc získava riešením duálneho modelu kapacitne neobmedzenej umiestňovacej úlohy, ktorú sa snaží heuristickou metódou rýchlo odhadnúť. Horná hranica vrcholu ako aj najlepšie nájdené riešenie je získané konštrukciou primárneho riešenia v snahe dodržať komplementárne podmienky v čo najväčšej možnej miere, teda s čo najmenším gapom medzi hornou a dolnou hranicou. Hornú hranicu udáva hodnota účelovej funkcie primárneho riešenia.

Tento prístup chcem zovšeobecniť v snahe riešiť úlohu návrhu verejného obslužného systému ako umiestňovaciú úlohu s obmedzeným počtom vybudovaných obslužných stredísk a získať tak exaktný algoritmus na riešenie tejto úlohy. Návrh exaktného algoritmu založeného na metóde vetiev a hraníc vyžaduje definovanie primárneho matematického modelu a k nemu prislúchajúci duálny model spolu s podmienkami komplementárnosti na výpočet dolnej a hornej hranice riešenia, určenie spôsobu vetvenia a schému prehľadania stromu riešení v metóde vetiev a hraníc.

4.3.1 Matematický model úlohy návrhu verejného obslužného systému

Základný model úlohy návrhu verejného obslužného systému ako úlohy umiestňovacej s obmedzeným počtom vybudovaných stredísk som definoval v kapitole 2.1.1. Nadviažem na tento model a skonštruujem podrobnejší model úlohy návrhu verejného obslužného systému. V základnom modeli som pracoval s dvoma množinami: množinou kandidátov na stredisko I a množinou zákazníkov J . V podrobnejšom modeli úlohy budem brať do úvahy, to že niektoré umiestnenia stredísk môžu byť zakázané a pri niektorých zase môže byť vyžadované ich obsadenie strediskom. Tento zákaz resp. príkaz umiestnenia strediska v mieste i je vykonaný formou fixácie stredísk v metóde vetiev a hraníc v procese vetvenia. V podrobnejšom modeli budem pracovať s množinou zákazníkov J a množinou

I rozčlenenou vzhľadom k základnému modelu (2.1.1) – (2.1.6). Z množiny I vytvoríme tri podmnožiny, s ktorými budem pracovať pri skonštruovaní modelu riešenej úlohy:

1. Množina prikázaných stredísk I_1 je podmnožina množiny všetkých možných umiestnení stredísk, kde je prikázané umiestniť stredisko.
2. Množina zakázaných stredísk I_0 je podmnožina množiny všetkých možných umiestnených stredísk, kde je zakázané umiestniť stredisko.
3. Množina voľných stredísk I_V je podmnožina množiny všetkých možných umiestnení stredísk, kde je povolené umiestniť stredisko.

Množiny stredísk I_1 , I_0 a I_V sú disjunktné. Pracuje sa tu s dvomi druhmi rozhodnutí. Rozhodnutie o umiestnení resp. neumiestnení strediska v mieste i z množiny I je modelované sadou bivalentných premenných y_i pre i z množiny I . Premenná y_i nadobúda hodnotu rovnú jednej, ak umiestnim stredisko v mieste i . Premenná y_i je rovná nule, ak neumiestnim stredisko v mieste i . Rozhodnutie o priradení resp. nepriradení zákazníka j z množiny J k stredisku i z množiny I je modelované sadou premenných z_{ij} , kde z_{ij} je rovné jednej, ak priradím zákazníka j k stredisku i . Premenná z_{ij} je rovná nule, ak nepriradím zákazníka j k stredisku i . Náklady spojené s uspokojením požiadavky zákazníka j z miesta i za určitú dobu sú modelované konštantami c_{ij} . Konštanty f_i predstavujú fixné náklady na udržanie obslužného strediska v mieste i na určitú dobu. Konštantu p udáva horné ohraničenie počtu vybudovaných stredísk.

Na základe takéhoto rozčlenenia množiny I a definovaní rozhodovacích premenných je možné sformulovať model úlohy návrhu verejného obslužného systému nasledovne:

$$\text{Minimalizujte} \quad \sum_{i \in I_1} f_i + \sum_{i \in I_V} f_i y_i + \sum_{i \in I_1 \cup I_V} \sum_{j \in J} c_{ij} z_{ij} \quad (4.3.01)$$

$$\text{Za podmienok:} \quad \sum_{i \in I_1 \cup I_V} z_{ij} = 1 \quad \text{pre } j \in J \quad (4.3.02)$$

$$z_{ij} \leq y_i \quad \text{pre } i \in I_V, \quad j \in J \quad (4.3.03)$$

$$\sum_{i \in I_V} y_i \leq p - |I_1| \quad (4.3.04)$$

$$y_i \in \{0,1\} \quad \text{pre } i \in I_V \quad (4.3.05)$$

$$z_{ij} \in \{0,1\} \quad \text{pre } i \in I_V \cup I_1, \quad j \in J \quad (4.3.06)$$

Účelová funkcia (4.3.01) predstavuje celkové náklady uspokojenia požiadaviek zákazníkov a vybudovania stredísk, ktoré minimalizujeme. Podmienky (4.3.02) zabezpečujú, aby každý zákazník j z množiny J bol priradený k práve jednému vybudovanému stredisku v mieste i z množín I_1 alebo I_V . Podmienky (4.3.03) zabezpečujú, aby nebolo možné priradiť zákazníka j miestu i (kandidátovi na umiestnenie strediska), kde stredisko nie je vybudované. Vybudovanie maximálne p stredísk zabezpečuje podmienka (4.3.04). Podmienky (4.3.05) a (4.3.06) určujú obor hodnôt rozhodovacích premenných.

V modeli nevystupuje množina I_0 , keďže miesta, kde je zakázané umiestniť strediská obsluhy nepredstavujú žiadne fixné náklady na ich udržanie. Naopak v modeli vystupuje množina voľných stredísk I_V , ktorá pozostáva z kandidátov, kde sa rozhoduje či v daných miestach bude umiestnené stredisko. Pri prioritne umiestnených strediskách, ktoré obsahuje množina I_1 rozhodujem len o priradení zákazníka k stredisku z množiny I_1 . Avšak zákazník nemusí byť priradený stredisku z množiny I_1 , nakoľko môže byť priradený bližšiemu vybudovanému stredisku z množiny I_V , o ktorého vybudovaní sa rozhodlo. V podmienke (4.3.04) je označená mohutnosť množiny $|I_1|$, ktorá určuje počet prvkov množiny I_1 . Nakoľko sú už nejaké strediská prioritne určené, potrebujem vybrať najviac $(p - |I_1|)$. Keďže rozhodovacie premenné majú celočíselný obor hodnôt, tento model sa nazýva diskretným modelom úlohy návrhu verejného obslužného systému. Tento model je možné upraviť na spojitý model úlohy návrhu verejného obslužného systému s využitím poznatkov a metód, ktoré Erlenkotter [22] aplikoval pri tvorbe spojitého modelu kapacitne neobmedzenej umiestňovacej úlohy popísanej v kapitole 2.2.1. Na základe podmienky (4.3.03), ktorá hodnotou premennej y_i zhora obmedzuje hodnotu premennej z_{ij} , je možné obor hodnôt premennej z_{ij} (4.3.06) zapísať v tvare (4.3.07) :

$$z_{ij} \geq 0 \quad \text{pre } i \in I_V \cup I_1, \quad j \in J \quad (4.3.07)$$

Uskutočnenie LP-relaxácie rozhodovacej premennej y_i v modeli (4.3.01 - 4.3.06) a úprava diskretného matematického modelu úlohy návrhu verejného obslužného systému zmenia model na nasledovný tvar:

$$\text{Minimalizujte} \quad \sum_{i \in I_1} f_i + \sum_{i \in I_V} f_i y_i + \sum_{i \in I_1 \cup I_V} \sum_{j \in J} c_{ij} z_{ij} \quad (4.3.08)$$

$$\text{Za podmienok:} \quad \sum_{i \in I_1 \cup I_V} z_{ij} = 1 \quad \text{pre } j \in J \quad (4.3.09)$$

$$y_i - z_{ij} \geq 0 \quad \text{pre } i \in I_V, \quad j \in J \quad (4.3.10)$$

$$\sum_{i \in I_V} y_i \leq p - |I_1| \quad (4.3.11)$$

$$y_i \geq 0 \quad \text{pre } i \in I_V \quad (4.3.12)$$

$$z_{ij} \geq 0 \quad \text{pre } i \in I_V \cup I_1, \quad j \in J \quad (4.3.13)$$

4.3.2 Duálny model LP-relaxácie úlohy návrhu verejného obslužného systému

Na základe teórie duality [31] je možné skonštruovať k modelu neceločíselného lineárneho programovania (4.3.08) – (4.3.13) úlohy návrhu verejného obslužného systému aj príslušný duálny model, ktorý vyzerá nasledovne:

$$\text{Maximalizujte} \quad \sum_{j \in J} v_j + (p - |I_1|) x \quad (4.3.14)$$

$$\text{Za podmienok:} \quad \sum_{j \in J} w_{ij} + x \leq f_i \quad \text{pre } i \in I_V \quad (4.3.15)$$

$$v_j - w_{ij} \leq c_{ij} \quad \text{pre } i \in I_V, \quad j \in J \quad (4.3.16)$$

$$v_j \leq c_{ij} \quad \text{pre } i \in I_1, \quad j \in J \quad (4.3.17)$$

$$v_j \in R \quad \text{pre } j \in J \quad (4.3.18)$$

$$w_{ij} \geq 0 \quad \text{pre } i \in I_V, \quad j \in J \quad (4.3.19)$$

$$x \leq 0 \quad (4.3.20)$$

V duálnom modeli premenné v_j korešpondujú s podmienkami (4.3.09) z primárneho modelu, ktoré zabezpečujú, aby zákazníkovi j z množiny J bolo priradené práve jedno stredisko i z množiny I . Premenné w_{ij} korešpondujú s podmienkami (4.3.10) z primárneho modelu, ktorá zabezpečí priradenie zákazníka len vybudovaným strediskám, teda kde $y_i=1$. Premenná x odpovedá štrukturálnej podmienke (4.3.11) a nadobúda nekladnú hodnotu.

Premenné y_i v spojitom modeli odpovedajú podmienkam (4.3.15). Premenné z_{ij} odpovedajú podmienkam (4.3.16) – (4.3.17). Premenné v_j sú neobmedzené duálne premenné s oborom hodnôt \mathbb{R} . Ak vyjadríme premennú v_j z podmienky (4.3.16), potom podmienka bude mať tvar (4.3.21):

$$v_j \leq c_{ij} + w_{ij} \quad \text{pre } i \in I_V, \quad j \in J \quad (4.3.21)$$

Premenná w_{ij} podľa vzťahu (4.3.19) a konštanta c_{ij} nadobúdajú nezáporné hodnoty, teda ich súčet nemôže byť záporný. Z čoho vyplýva, že postačujúcou podmienkou platnosti vzťahu (4.3.21) je nulová hodnota premennej v_j . Vzhľadom na účelovú funkciu (4.3.14), kde maximalizujem premenné v_j a súčin $(p - |I_1|)x$ nárast účelovej funkcie zabezpečia iba nezáporné premenné v_j . Súčin $(p - |I_1|)x$ bude nekladný, nakoľko premenná x dáva nekladné hodnoty a pri dodržaní prípustného riešenia mohutnosť množiny $|I_1|$ nemôže byť väčšia ako konštanta p . Na základe dodržania podmienok (4.3.21) a snahe maximalizovať účelovú funkciu (4.3.14) postačia nezáporné hodnoty premenných v_j , teda obmedziť premenné v_j na obor \mathbb{R}^+ . Premenné w_{ij} podľa (4.3.19) nadobúdajú nezáporné hodnoty. Ak vyjadrím premennú w_{ij} z podmienky (4.3.21), potom podmienka bude vyzeráť nasledovne (4.3.22):

$$w_{ij} \geq v_j - c_{ij} \quad \text{pre } i \in I_V, \quad j \in J \quad (4.3.22)$$

Ak podmienky (4.3.19) a (4.3.22) zlúčim, dokážem vyjadriť hodnotu premenných w_{ij} nasledovným vzťahom:

$$w_{ij} \geq \max\{0, v_j - c_{ij}\} \quad \text{pre } i \in I_V, \quad j \in J \quad (4.3.23)$$

Postačujúcou požiadavkou dodržania podmienky (4.3.19) je minimálna hodnota premennej w_{ij} daná nasledovným vzťahom:

$$w_{ij} = \max\{0, v_j - c_{ij}\} \quad \text{pre } i \in I_V, \quad j \in J \quad (4.3.24)$$

Zavedením tejto substitúcie do duálneho modelu dostanem redukovaný duálny model, kde sa vyskytujú iba premenné v_j . Aby som mal podmienky v tvare rovnosti zavediem doplnkové premenné u_i pre všetky možné umiestnenia i z množiny I_V . Redukovaný duálny model potom vyzerá nasledovne:

$$\text{Maximalizujte } \sum_{j \in J} v_j + (p - |I_1|)x \quad (4.3.25)$$

$$\text{Za podmienok: } \sum_{j \in J} \max\{0, v_j - c_{ij}\} + x + u_i = f_i \quad \text{pre } i \in I_V \quad (4.3.26)$$

$$v_j \leq c_{ij} \quad \text{pre } i \in I_1, \quad j \in J \quad (4.3.27)$$

$$v_j \geq 0 \quad \text{pre } j \in J \quad (4.3.28)$$

$$x \leq 0 \quad (4.3.29)$$

$$u_i \geq 0 \quad \text{pre } i \in I_V \quad (4.3.30)$$

Redukovaný duálny model pozostáva z účelovej funkcie (4.3.25). V podmienke (4.3.26) je zahrnutá spomínaná substitúcia (4.3.24), kde som nahradil premennú w_{ij} v modeli (4.3.14) – (4.3.20) a upravil podmienku (4.3.15) na tvar rovnosti vložení doplnkovej premennej u_i . Podmienky (4.3.28) – (4.3.30) sú obligatórne, teda zabezpečujú obor hodnôt rozhodovacích duálnych premenných v_j , x a doplnkových premenných u_i .

4.3.3 Výpočet dolnej hranice účelovej funkcie

Erlenkotterov prístup k dolnej hranici pri riešení kapacitne neobmedzenej umiestňovacej úlohy (viď kapitola 2.2.1) vychádza z lineárnej relaxácie pôvodnej úlohy a riešenia príslušnej duálnej úlohy. Získané poznatky z Erlenkotterovho princípu a teórie duality som aplikoval na získanie dolnej hranice úlohy návrhu verejného obslužného systému. Tieto poznatky sú:

1. Množina prípustných riešení relaxovanej úlohy obsahuje všetky riešenia pôvodnej úlohy.
2. Hodnota účelovej funkcie optimálneho riešenia minimalizačnej relaxovanej úlohy je dolnou hranicou hodnôt účelovej funkcie riešení v spracovávanej vetve.
3. Získanie optimálneho riešenia minimalizačnej relaxovanej úlohy vyžaduje riešenie rozsiahlej úlohy lineárneho programovania, čo kladie veľké nároky na čas výpočtu, a preto je táto dolná hranica iba odhadovaná zdola hodnotou riešenia duálnej úlohy.
4. Hodnota účelovej funkcie každého prípustného riešenia duálneho modelu poskytuje dolnú hranicu (DH) hodnoty účelovej funkcie ľubovoľného prípustného riešenia primárnej úlohy.

Dolná hranica úlohy návrhu verejného obslužného systému je tak získaná nasledovne:

$$DH = \sum_{j \in J} v_j + (p - |I_1|) x \quad (4.3.31)$$

Tento vzťah (4.3.31) odpovedá účelovej funkcii duálneho modelu (4.3.25). Na získanie dolnej hranice ako hodnoty duálneho riešenia je potrebné definovať východiskové riešenie, ktoré je neskôr vylepšované. Východiskové riešenie získam nasledovným nastavením duálnych premenných:

$$v_j = \min \{c_{ij} : i \in I_V \cup I_1\} \quad \text{pre } j \in J \quad (4.3.32)$$

$$u_i = f_i \quad \text{pre } i \in I_V \quad (4.3.33)$$

$$x = 0 \quad (4.3.34)$$

Na zlepšenie dolnej hranice som navrhol heuristický postup nazývaný ako korekčný vzostupný duálny algoritmus CDA (**Correction Dual Ascent Algorithm**). Algoritmus vychádza zo vzostupného duálneho algoritmu (DA), ktorý navrhol Erlenkotter, vylepšili Janáček a Buzna a je popísaný v kapitole 2.2.1. Korekčný vzostupný duálny algoritmus CDA má nasledovný predpis:

0. {Inicializácia} Určte počiatočné prípustné duálne riešenie $\mathbf{u}, \mathbf{v}, \mathbf{x}$: $u_i = f_i$ pre $i \in I_V$, $v_j = \min\{c_{ij} : i \in I_V \cup I_1\}$ pre $j \in J$ a $x = 0$. Podľa vzťahu (4.3.31) určte hodnotu DH aktuálnej dolnej hranice.
1. Usporiadajte množinu zákazníkov J do sekvencie $j_1, j_2, \dots, j_k, \dots, j_n$ vzostupne na základe kardinality množiny voľných stredísk pre jednotlivých zákazníkov $K_j = \{i \in I_V, v_j \geq c_{ij}\}$.
2. Pre každé $j \in J$ vykonajte nasledujúci postup: Určte $\delta = \min\{\min\{u_i : i \in I_V, v_j \geq c_{ij}\}, \min\{c_{ij} - v_j : i \in I_V, v_j < c_{ij}\}, \min\{c_{ij} - v_j : i \in I_1, v_j < c_{ij}\}\}$. Ak je $\delta > 0$, tak pre každé $i \in I_V$, pre ktoré platí $v_j \geq c_{ij}$, položte $u_i = u_i - \delta$, upravte v_j podľa vzťahu $v_j = v_j + \delta$, zvýšte $DH = DH + \delta$ a aktualizujte pozíciu zákazníka j v sekvencii všetkých zákazníkov $j_1, j_2, \dots, j_k, \dots, j_n$.
3. Vyberte v poradí doposiaľ nespracovaného zákazníka j^* z usporiadanej množiny J pokiaľ platí, že $|\{u_i = 0, i \in I_V\}| > (p - |I_1|)$. Inak skonči.
4. Zálohujte si hodnoty duálnych premenných x, v_j, u_i , dolnú hranicu DH a aktuálnu

sekvenciu zákazníkov $j_1, j_2, \dots, j_k, \dots, j_n$.

5. Určte $\Delta x = \min((c_{ij^*} - v_{j^*}) \geq 0, i \in I_V \cup I_1)$ a nastavte duálne premenné nasledovne: $x = x - \Delta x, v_{j^*} = v_{j^*} + \Delta x$ a pre každé $i \in I_V$, kde $v_j < c_{ij}$ hodnotu $u_i = u_i + \Delta x$. Upravte pozíciu zákazníka j^* v usporiadanej množine J .
6. Pre každé $j \in J$ vykonajte nasledujúci postup: Urči $\delta = \min\{\min\{u_i: i \in I_V, v_j \geq c_{ij}\}, \min\{c_{ij} - v_j: i \in I_V, v_j < c_{ij}\}, \min\{c_{ij} - v_j: i \in I_1, v_j < c_{ij}\}\}$. Ak je $\delta > 0$, tak pre každé $i \in I_V$, pre ktoré platí $v_j \geq c_{ij}$, položte $u_i = u_i - \delta$, upravte v_j podľa vzťahu $v_j = v_j + \delta$, zvýšte $DH = DH + \delta$ a aktualizujte pozíciu zákazníka j v usporiadanej množine J .
7. Ak sa hodnota dolnej hranice DH zvýšila, choďte na krok 3, inak obnovte hodnoty duálnych premenných, dolnej hranice a sekvencie zo zálohy vykonanej v kroku 4 a choďte na krok 3.

CDA procedúrou získam riešenie, v ktorom pre každého zákazníka $j \in J$ existuje aspoň jedno umiestnenie $i \in I_V \cup I_1$, pre ktoré platí $v_j \geq c_{ij}$ a $u_i = 0$, teda zákazník j sa „opiera“ o umiestnenie i alebo naopak, umiestnenie i sa „opiera“ o zákazníka j . Konkrétny zákazník j sa musí opierať aspoň o jedno umiestnenie i , ale môže sa opierať aj o viacero umiestnení. Taktiež každé umiestnenie i sa môže opierať o jedného alebo viacerých zákazníkov j alebo sa nemusí opierať o žiadneho zákazníka. V CDA procedúre je zmena duálnych premenných viazaná na určenie Δx , ktorej hodnota je viazaná na aktuálneho spracovávaného zákazníka.

4.3.4 Výpočet hornej hranice riešenia

Erlenkotterov prístup k získaniu hornej hranice a prípustného primárneho riešenia je založený na poznatkoch z teórie duality. Jeho snahou je využiť duálne riešenie získané pri výpočte dolnej hranice daného vrcholu. Na základe vety o komplementárnosti. Erlenkotter [22] sformuloval komplementárne podmienky v snahe získať primárne riešenie k duálnemu riešeniu na základe dodržania komplementárnych podmienok čo v najväčšej miere. Erlenkotter sa pri konštrukcii primárneho riešenia snaží získať čo najmenší možný rozdiel medzi hornou a dolnou hranicou.

Môj prístup k získaniu hornej hranice ako aj primárneho riešenia vychádza z týchto Erlenkotterových poznatkov. Získané poznatky z Erlenkotterovho prístupu, ktoré som využil pri konštrukcii prípustného primárneho riešenia sú:

1. Hodnota účelovej funkcie prípustného primárneho riešenia odpovedá hornej hranici prislúchajúcej najlepšiemu celočíselnému riešeniu.
2. Hodnota hornej hranice prislúchajúcej najlepšiemu celočíselnému riešeniu podľa hodnôt účelových funkcií nájdených prípustných riešení primárnej úlohy sa aktualizuje počas výpočtu.
3. Hodnota účelovej funkcie každého prípustného riešenia z dvojice primárno-duálnych úloh, ktorú maximalizujeme je menšia alebo rovná hodnote účelovej funkcie ľubovoľného prípustného riešenia druhej úlohy, v ktorej je účelová funkcia minimalizovaná (Slabá veta o dualite).
4. Ak jedna z dvojice primárnej a duálnej úlohy má optimálne riešenie, potom ho má i tá druhá úloha a hodnoty účelových funkcií optimálnych riešení sú rovné (Silná veta o dualite).
5. Pre optimálne riešenie dvojice primárnej a duálnej úlohy platí, že buď je nerovnosť jednej zo sústav splnená ako rovnosť alebo je premenná druhej sústavy rovná nule (Veta o komplementárnosti).

Prípustné primárne riešenie konštruujem tak, aby podmienky pre primárnu a duálnu úlohu boli splnené čo najtesnejšie. Pre všetky dvojice prípustných riešení oboch sústav platia teda nasledujúce podmienky (4.3.35) – (4.3.39):

$$(y_i - z_{ij}) \max\{0, v_j - c_{ij}\} \geq 0 \quad \text{pre } i \in I_V, \quad j \in J \quad (4.3.35)$$

$$\left(\sum_{i \in I_V} y_i - p + |I_1| \right) x \geq 0 \quad (4.3.36)$$

$$u_i y_i \geq 0 \quad \text{pre } i \in I_V \quad (4.3.37)$$

$$\max\{0, c_{ij} - v_j\} z_{ij} \geq 0 \quad \text{pre } i \in I_V, \quad j \in J \quad (4.3.38)$$

$$(c_{ij} - v_j) z_{ij} \geq 0 \quad \text{pre } i \in I_1, \quad j \in J \quad (4.3.39)$$

Ak sú podmienky (4.3.35) – (4.3.39) splnené na rovnosť, tak dané riešenia sú optimálne pre obe úlohy a hovoríme o tzv. komplementárnych podmienkach. Zovšeobecnením Erlenkotterovho prístupu som získal novú podmienku (4.3.36). V prípade, že niektoré podmienky nie sú splnené na rovnosť, bude hodnota účelovej funkcie primárneho riešenia

$F_P(y, z)$ vyššia ako hodnota duálneho riešenia $F_D(u, v)$, a to práve o súčet ľavých strán podmienok (4.3.35) – (4.3.39):

$$\begin{aligned}
 gap = F_P(y, z) - F_D(u, v, x) &= \sum_{i \in I_V} \sum_{j \in J} (y_i - z_{ij}) \max\{0, v_j - c_{ij}\} + \sum_{i \in I_V} u_i y_i + \\
 &+ \left(\sum_{i \in I_V} y_i - p + |I_1| \right) x + \sum_{i \in I_V} \sum_{j \in J} \max\{0, c_{ij} - v_j\} z_{ij} + \sum_{i \in I_1} \sum_{j \in J} (c_{ij} - v_j) z_{ij} \quad (4.3.40)
 \end{aligned}$$

Kvalita hornej hranice pre daný vrchol je závislá na dodržiavaní komplementárnych podmienok. Preto pri konštrukcii prípustného primárneho riešenia sa snažím získať čo najmenší *gap* medzi hornou a dolnou hranicou vrcholu a budem postupovať podľa nasledovných zásad:

1. Dodržať podmienky (4.3.38) – (4.3.39) priradovaním zákazníka j len k tým umiestneniam i , o ktoré sa zákazník opiera ($v_j \geq c_{ij}$).
2. Dodržať podmienky (4.3.37) umiestňovaním stredísk obsluhy z množiny možných umiestnení I_V , o ktoré sa opiera aspoň jeden zákazník ($u_i = 0$).
3. Dodržať podmienku (4.3.36) umiestnením presne p -stredísk resp. udržiavaním nulovej hodnoty premennej x a potom môžem umiestniť nanajvýš p stredísk.

Rozdiel medzi riešeniami pri dodržaní spomenutých zásad udávajú podmienky (4.3.35). Pomocou heuristického postupu sa budeme snažiť nájsť najmenšiu množinu umiestnení, o ktorú sa opierajú zákazníci tak, aby každý sa opieral aspoň o jedno umiestnenie z hľadanej podmnožiny. Algoritmus pre získanie minimálnej množiny umiestnení I^* má nasledovné kroky:

0. (Inicializácia) Položte $I^* = I_1$.
1. Pokiaľ $|I^*| < p$ postupne vložte do I^* všetky $i \in I_V$, pre ktoré platí: $u_i = 0$, existuje aspoň jeden zákazník j , ktorý sa opiera iba o i (t.j. $v_j \geq c_{ij}$ a pre všetky $k \neq i$, kde $u_k = 0$ je $v_j < c_{ik}$).
2. Preberte všetky $j \in J$, pokiaľ pre všetky $i \in I^*$ platí, že $v_j < c_{ij}$ a súčasne $|I^*| < p$, kde nájdite $i^* = \text{solmin}\{c_{ij} : i \in I_V - I^*, u_i = 0\}$ a položte $I^* = I^* \cup \{i^*\}$.

3. Pokiaľ $|I^*| = p$ alebo $x = 0$ skončí, inak postupne vkladajte také i^* do množiny I^* , pre ktoré platí:

$$i^* = \operatorname{solmin} \left\{ u_i + \sum_{j \in J} (v_j - c_{ij}) : i \in I_V - I^* \right\} \quad (4.3.41)$$

Pri navrhnutom algoritme pre získanie minimálnej množiny umiestnení sa snažím, aby podmienky (4.3.35) – (4.3.39) platili na rovnosť v čo najväčšej miere. To znamená, že v prípade nedodržania niektorých komplementárnych podmienok, aby bol vzniknutý *gap* (4.3.40) minimálny. Avšak splnenie podmienky (4.3.36) na rovnosť považujem za prvoradé oproti rovnosti ostatných podmienok. Dodržaním tejto podmienky na rovnosť sa otvára možnosť získania lepšieho riešenia aj menšieho rozdielu medzi dolnou a hornou hranicou *gap* za cenu porušenia niektorej z podmienok (4.3.35), (4.3.37) – (4.3.39).

Tento *gap* sa potom snažím znížiť algoritmom duálnych úprav DAD (**Dual Adjustment Algorithm**), ktorý navrhol Erlenkotter. Algoritmus duálnych úprav podrobne popísaný v kapitole 2.2.1 je založený na zmenšení jednej hodnoty duálnej premennej v_j , ktoré vytvorí voľnú „kapacitu“ aspoň u dvoch umiestnení i_1, i_2 a následné zvýšenie aspoň dvoch hodnôt iných dvoch duálnych premenných v_j .

4.3.5 Schéma prehľadovania stromu riešení v metóde vetiev a hraníc

Schéma prehľadania je určená elementárnym pravidlom, ktoré hovorí o tom, ktorý z dvoch novovytvorených vrcholov so spoločným predchodcom bude vybraný a spracovaný ako prvý. Dvoma novovytvorenými vrcholmi chápeme vrchol získaný príkazom a vrchol získaný zákazom. Vrchol, ktorý sme vytvorili fixovaním získaného kandidáta na vetvenie na hodnotu jedna je vrchol získaný príkazom. Vrchol, ktorý sme vytvorili fixáciou získaného kandidáta na vetvenie na hodnotu nula je vrchol získaný zákazom. V algoritme pMedBBDual, ktorý realizuje zovšeobecnený prístup, sa strom riešení prehľadáva do hĺbky rovnako ako v algoritme BBDual [33]. Pri prehľadaní do hĺbky v algoritme pMedBBDual je elementárnym pravidlom výber vrcholu s menšou hodnotou dolnej hranice, pričom výnimkou je prednosť v spracovaní vrcholu, ktorý bude vylúčený z ďalšieho prehľadania. To znamená, že v prípade vytvorenia dvoch nových vrcholov so spoločným predchodcom, ak dolná hranica vrcholu získaného zákazom je väčšia ako dolná hranica vrcholu získaného

príkazom, potom ako prvý spracujeme vrchol získaný príkazom v prípade, že hodnota dolnej hranice vrcholu získaného zákazom je menšia ako hodnota hornej hranice vrcholu získaného zákazom, teda vrchol nie je možné vylúčiť z prehľadania. Potom vrchol získaný zákazom si odložíme do zásobníka nespracovaných vrcholov. V opačnom prípade ako prvý spracujeme vrchol získaný zákazom, ak vrchol získaný príkazom nie je možné vylúčiť z prehľadania. Potom vrchol získaný príkazom odložíme do zásobníka pre neskoršie spracovanie pri návrate (backtracking).

4.3.6 Spôsob vetvenia exaktného algoritmu pMedBBDual

Spôsob vetvenia predstavuje vytváranie stromu prehľadania, ktoré je v algoritme pMedBBDual zabezpečené fixovaním jednotlivých premenných y_i , ktoré rozhodujú o umiestnení strediska. Tento spôsob vetvenia použil aj Janáček a Buzna [33] v algoritme BBDual pri riešení kapacitne neobmedzenej umiestňovacej úlohy popísaný v kapitole 2.2.1. Fixáciou sa zredukuje množina riešení vypustením jednoznačne určených premenných y_i , ktoré vhodným použitím prohibítivnej konštanty v účelovej funkcii vynúti minimalizáciu. Poradie fixácii jednotlivých premenných y_i závisí od vyhodnotenia narušenia komplementárnych podmienok (4.3.35) – (4.3.39) pre jednotlivé strediská i . Procedúra pre získanie hornej hranice nám poskytuje dodržanie podmienky (4.3.36), preto túto podmienku nie je nutné vyhodnocovať. Podmienka (4.3.39) je viazaná len k už fixovaným premenným y_i na hodnotu jedna, teda nemôžem vybrať dvakrát toho istého kandidáta na fixáciu, aj keď je porušená táto podmienka. Množina kandidátov na fixáciu je tvorená iba doposiaľ nefixovanými premennými y_i , čo nám podmienku (4.3.39) vylučuje z vyhodnotenia. Vyhodnotenie narušenia podmienok (4.3.35), (4.3.37), (4.3.38) realizujem pomocou procedúry na získanie vhodného kandidáta na fixáciu. Hľadanie najväčšieho porušenia podmienok je časovo náročné, preto som sa zameril na nájdenie prvého porušenia komplementárnych podmienok (4.3.35), (4.3.37), (4.3.38). Výber vhodného kandidáta zníži *gap* medzi hornou a dolnou hranicou riešenia (4.3.40). Procedúra pre získanie kandidáta na fixáciu má nasledovný predpis:

0. Inicializujte hľadaného kandidáta na fixáciu $i^* = -1$ a množinu potencionálnych kandidátov na fixáciu $I_K = I^* - I_L$.

1. Preberte postupne umiestnenia $i \in I_K$, kým nenájdete prvé porušenie podmienky (4.3.35) a aktualizujte i^* rovné umiestneniu i , ktoré porušilo túto podmienku (4.3.35).
2. Ak $i^* = -1$, preberte postupne umiestnenia $i \in I_K$, kým nenájdete prvé porušenie podmienky (4.3.37), teda hodnotu premennej $u_i > 0$, a aktualizujte i^* rovné umiestneniu i , ktoré porušilo túto podmienku (4.3.37).
3. Ak $i^* = -1$, preberte postupne umiestnenia $i \in I_K$, kým nenájdete prvé porušenie podmienky (4.3.38), aktualizujte i^* rovné umiestneniu i , ktoré porušilo túto podmienku (4.3.38).
4. Ak $i^* = -1$, nastavte i^* podľa nasledovného vzťahu (4.3.42):

$$i^* = \operatorname{solmin} \left\{ f_i - \sum_{j \in J} \max\{0, v_j - c_{ij}\} : i \in I^* - I_1 \right\} \quad (4.3.42)$$

Aj splnenie podmienok (4.3.35) – (4.3.38) nemusí znamenať získanie nulového rozdielu *gap* (4.3.40) a tým vylúčenie vrcholu z prehľadania. Rozdiel medzi získanou dolnou a hornou hranicou udávajú podmienky (4.3.39), preto je ďalší kandidát vybraný cez minimálnu hodnotu duálnej premennej x vzťahom (4.3.42). Týmto navrhnutým algoritmom pre získanie vhodného kandidáta na fixáciu sa snažím znížiť *gap*, a tým aj počet spracovaných vrcholov v metóde vetiev a hraníc.

4.3.7 Úplná fixácia v metóde vetiev a hraníc

Úplná fixácia v metóde vetiev a hraníc predstavuje zafixovanie všetkých stredísk na hodnotu nula alebo jedna. Pri postupnej fixácii kandidátov získaných na vetvenie pomocou navrhnutého algoritmu musím rozlíšiť prípady, kedy sa nepodarilo vylúčiť vrchol z prehľadania a súčasne nie je vhodné ďalej vetviť v metóde vetiev a hraníc. Využitím úplnej fixácie sa snažím obmedziť ďalšiemu zbytočnému vetveniu v algoritme. Ďalšie vetvenie v algoritme by viedlo k získaniu neprípustného riešenia, nenájdeniu kandidáta na vetvenie alebo porušeniu podmienky komplementárnosti(4.3.38). Tieto prípady sa viažu na zákaz vetvenia na hodnotu nula alebo jedna a sú to:

Prípád 0: Spracovanie vrcholu, kde nie je už možné vetviť na hodnotu 0

a) $|I_0| + |I_1| = |I|$

V tomto prípade boli fixované všetky možné umiestnenia, z čoho vyplýva, že nemám ďalšieho kandidáta na vetvenie, teda množina možných kandidátov na fixáciu je prázdna.

b) $|I_0| = |I| - p$

Dodržanie komplementárnej podmienky (4.3.36) pri hodnote x menšej ako nula zabezpečí umiestnenie práve p stredísk. V tomto prípade to vedie k zákazu fixácie ďalších kandidátov na hodnotu nula a prioritne fixovanie ostatných nefixovaných umiestnení na hodnotu jedna.

c) $|I_0| = |I| - 1$

Za prípustné riešenie považujeme riešenie, ktoré obsahuje aspoň jedno umiestnené stredisko. Dodržanie komplementárnej podmienky (4.3.36) zabezpečuje hodnota duálnej premennej x rovná nule, čím sa otvára priestor umiestniť menej ako p stredísk. V tomto prípade to znamená, že pri fixácii len na hodnotu nula je nulová fixácia posledného nefixovaného strediska pre získanie prípustného riešenia zakázaná, pričom je prioritne stredisko fixované na hodnotu jedna.

Prípád 1: Spracovanie vrcholu, kde nie je už možné vetviť na hodnotu 1

a) $|I_1| = p$

Dodržanie komplementárnej podmienky (4.2.36) pri hodnote x menšej ako nula zabezpečí umiestnenie p stredísk. V tomto prípade to vedie k zákazu fixácie ďalších kandidátov na jedna, nakoľko už počet fixovaných premenných na hodnotu jedna je presne p a fixácia ďalšieho strediska na hodnotu jedna by znamenalo získanie neprípustného riešenia. Preto sú prioritne fixované ostatné nefixované umiestnenia na hodnotu nula.

b) $|I_0| + |I_1| = |I|$

V tomto prípade boli fixované všetky možné umiestnenia, z čoho vyplýva, že nemáme ďalšieho kandidáta na vetvenie, t.j. množina možných kandidátov na fixáciu je prázdna.

Úplnou fixáciou v spomenutých prípadoch získam prípustné riešenie pomerne rýchlo a jednoducho, pričom ako hodnotu dolnej hranice stanovím hodnotu hornej hranice a tým daný vrchol stromu riešeni môžem vylúčiť. Z daných prípadov je možné vidieť spôsoby dodržania podmienky (4.3.36). Ak dodržíme podmienku nulovou hodnotou duálnej premennej x , ktorá je získaná pri výpočte dolnej hranice, riešim tak kapacitne neobmedzenú umiestňovaciu úlohu a tomu aj prispôbujem hranice medzi vhodným a nevhodným vetvením algoritmu. Naopak ak dodržaním podmienky (4.3.36) zabezpečujem umiestnením presne p stredísk riešime úlohu p -mediánu, prípadne váženého p -mediánu (2.1.2) a zase tejto úlohe prispôbujem hranice medzi vhodným a nevhodným vetvením algoritmu. Ošetrovanie týchto prípadov vedie k efektívnemu hľadaniu optimálneho riešenia v metóde vetiev a hraníc.

4.3.8 Predbežné experimenty

Takto skonštruovaný algoritmus [9] som testoval na benchmarkoch zo slovenskej cestnej siete s ohľadom na výpočtový čas, počet spracovaných vrcholov, hodnotu dolnej hranice koreňa, hodnotu dolnej hranice riešenia a hodnotu hornej hranice riešenia odpovedajúcu najlepšiemu nájdenému riešeniu. Všetky experimenty boli realizované na PC s procesorom i7 Q720 a 8GB RAM. Hodnotu riešenia získaného algoritmom pMedBBDual ako aj hodnotu výpočtového času sme porovnávali s úlohou riešenou univerzálnym IP solverom. Súbory, ktoré prislúchajú testovacím úlohám sú uvedené v prílohe 1. Experimenty o rozsahu 100×100 sú realizované na úlohe p -mediánu definovanej na vybraných 100 obciach Banskobystrického kraja.

Experimenty v Tab. 14 ukázali, že algoritmus pMedBBDual získal optimálne riešenie vo väčšine prípadov. Prípady, kedy som nezískal optimálne riešenie vyplynuli z toho, že výpočtový čas bol vzhľadom na rozsah riešenej úlohy veľký. Pre riešene úlohy som stanovil maximálny výpočtový čas riešenej úlohy jednu hodinu, ktorá bola v niektorých prípadoch aj dosiahnutá a tým bol predčasne ukončený algoritmus. Toto predčasne skončenie algoritmu spôsobilo nezískanie vždy optimálneho riešenia. Ak porovnam výpočtové časy algoritmu pMedBBDual a univerzálného IP solvera XPRESS-IVE, tak výpočtový čas získaný IP solverom je kratší vo všetkých prípadoch, niekedy až niekoľko násobne. Tento pomerne veľký výpočtový čas bol podmienený spracovaním veľkého počtu vrcholov vo vetvení. Ak si zoberiem kvalitu získanej dolnej hranice, tak Tab. 14 ukazuje v niektorých prípadoch získanie nekvalitnej dolnej hranice. Kvalitu dolnej hranice posudzujem na základe čo

najväčšieho možného priblíženia k hodnote hľadaného optimálneho riešenia. V prípadoch, kde hodnota p bola 15, 25, 60 som získal veľmi kvalitnú dolnú hranicu, keďže dolná hranica sa rovnala hodnote optimálneho riešenia. Avšak v prípadoch, kde hodnota p bola 55 a 85, hodnota dolnej hranice bola nekvalitná. Experimenty tiež ukázali, že pokiaľ dolná hranica koreňa nie je kvalitná, tak nájdenie optimálneho riešenia vetviacim algoritmom spôsobí spracovanie veľkého množstva spracovaných vrcholov. Tiež sa ukázalo, že vetvením nedokážem výrazne vylepšiť získanú dolnú hranicu riešenia oproti dolnej hranice koreňa stromu vetvenia.

Tab. 14 Porovnanie algoritmu pMedBBDual s XPRESS-IVE na benchmarku BB100x100

p	XPRESS-IVE		pMedBBDual					
	OF	$t[s]$	DH_K	DH_R	HH	$t[s]$	noN	
5	2069	1,04	2032	2069	2069	2,74	555	
10	1275	1,01	1265	1275	1275	25,07	12719	
15	923	0,95	923	923	923	1,32	369	
20	745	0,93	738	745	745	9,14	3531	
25	616	0,89	616	616	616	1,71	521	
30	521	0,97	520	521	521	1,58	345	
35	441	0,95	430	337	444	3 600*	3677965	
40	373	0,95	340	341	382	3 600*	4322839	
45	315	1,23	250	251	324	3 600*	5084461	
50	263	1,12	160	161	274	3 600*	5994615	
55	215	0,98	70	72	242	3 600*	6189543	
60	176	1,00	176	176	176	2,09	291	
65	142	1,04	141	142	142	3 600*	2753881	
70	112	1,00	106	86	112	3 600*	5692943	
75	87	0,96	71	71	88	3 600*	8806833	
80	64	0,96	36	36	65	3 600*	10740197	
85	44	0,95	1	4	44	3 600*	12314193	
90	25	1,00	12	12	25	3 600*	15186543	
95	10	0,98	4	10	10	86,44	479555	

¹ stĺpec p udáva hodnotu maximálneho počtu stredísk

² stĺpec OF udáva hodnotu optimálneho riešenia získanú pomocou univerzálneho IP solvera XPRESS-IVE

³ stĺpec $t[s]$ udáva čas trvania výpočtu

⁴ stĺpec DH_K udáva hodnotu dolnej hranice koreňa v algoritme pMedBBDual

⁵ stĺpec DH_R udáva hodnotu dolnej hranice riešenia v algoritme pMedBBDual

⁶ stĺpec HH udáva hodnotu hornej hranice riešenia v algoritme pMedBBDual

⁷ stĺpec noN udáva počet spracovaných vrcholov v metóde vetiev a hraníc

⁸ červená farba – predčasné zastavenie algoritmu po 1 hodine výpočtu (3600*)

⁹ zelená farba – získanie hodnoty optimálneho riešenia algoritmom pMedBBDual

4.4 Vylepšenia exaktného optimalizačného jadra

Na základe vyhodnotenia experimentov z Tab. 14 sa pokúsím bližšie analyzovať, prečo sú niektoré prípady tak časovo náročné. Otestujem kvalitu získania dolnej hranice pre jednotlivé vrcholy stromu prehľadania dosadením hodnoty optimálneho riešenia získaného univerzálnym IP solverom riešením lokačno-alokačného modelu do hornej hranice riešenia.

Horná hranica riešenia predstavuje doposiaľ najlepšie nájdené riešenie. Keďže hodnota optimálneho riešenia predstavuje hornú hranicu riešenie počas behu algoritmu v metóde vetiev a hraníc táto možnosť môže vylúčiť veľké množstvo spracovaných vrcholov, ktoré by sme pôvodne museli spracovať. V prípade, že počet spracovaných vrcholov by nám výrazne poklesol, môžeme zhodnotiť, že mám kvalitnú dolnú hranicu, no menej kvalitnú hornú hranicu riešenia. Naopak, ak počet spracovaných vrcholov výrazne nepoklesol, tak nemôžem získať dolnú hranicu považovať za kvalitnú. Otestujem to na rovnakej úlohe BB100x100, ale s najlepšou možnou hornou hranicou.

Tab. 15 Porovnanie kvality dolnej hranice na benchmarku BB100x100

p	$pMedBBDual$					$pMedBBDual(OPT)$			
	DH_K	DH_R	HH	$t[s]$	nN	DH_R	HH_N	$t_o[s]$	noN
5	2032	2069	2069	2,74	555	2069	2069	1,58	67
10	1265	1275	1275	25,07	12719	1275	1275	5,78	1227
15	923	923	923	1,32	369	923	923	0,02	1
20	738	745	745	9,14	3531	745	745	9,56	3163
25	616	616	616	1,71	521	616	616	0,01	1
30	520	521	521	1,58	345	521	521	0,74	87
35	430	337	444	3 600*	3677965	337	441	3 600*	3674234
40	340	341	382	3 600*	4322839	341	373	3 600*	4527215
45	250	251	324	3 600*	5084461	251	315	3 600*	5103254
50	160	161	274	3 600*	5994615	161	263	3 600*	5884235
55	70	72	242	3 600*	6189543	72	215	3 600*	6189543
60	176	176	176	2,09	291	176	176	0,12	1
65	141	142	142	3 600*	2753881	142	142	3 600*	2186423
70	106	86	112	3 600*	5692943	86	112	3 600*	5664432
75	71	71	88	3 600*	8806833	71	87	3 600*	8795683
80	36	36	65	3 600*	10740197	36	64	3 600*	11045197
85	1	4	44	3 600*	12314193	4	44	3 600*	12635231
90	12	12	25	3 600*	15186543	12	25	3 600*	15223646
95	4	10	10	86,44	479555	10	10	86,06	479555

¹ stĺpec p udáva hodnotu maximálneho počtu stredísk

² stĺpec DH_K udáva hodnotu dolnej hranice koreňa

³ stĺpec DH_R udáva hodnotu dolnej hranice riešenia

⁴ stĺpec HH udáva hodnotu hornej hranice riešenia

⁵ stĺpec $t[s]$ udáva čas trvania výpočtu pre algoritmus $pMedBBDual$

⁶ stĺpec nN udáva počet spracovaných vrcholov v metóde vetiev a hraníc algoritmu $pMedBBDual$

⁷ stĺpec HH_N udáva hodnotu hornej hranice riešenia prednastavenú na hodnotu optimálneho riešenia

⁸ stĺpec $t_o[s]$ udáva čas trvania výpočtu pre algoritmus $pMedBBDual(OPT)$

⁹ stĺpec noN udáva počet spracovaných vrcholov v metóde vetiev a hraníc algoritmu $pMedBBDual(OPT)$

¹⁰ červená farba – predčasné zastavenie algoritmu po 1 hodine výpočtu (3600*)

¹¹ zelená farba – získanie optimálneho riešenia

¹² $pMedBBDual(OPT)$ – algoritmus $pMedBBDual$ s prednastavenou hornou hranicou, ktorá je rovná hodnote optimálneho riešenia

Experimenty v Tab. 15 ukázali, že vo väčšine prípadov nám procedúra neposkytuje kvalitnú dolnú hranicu DH_R , čo sa odzrkadľuje na výpočtovom čase t a vysokom počte spracovávaných vrcholov nN . Ani nastavením najlepšej hornej hranici som nedokázal získať

riešenie za prijateľný čas t_0 a musíme spracovať veľké množstvo vrcholov $n \in N$. Touto analýzou som dospel k záveru, že je potrebné vylepšiť procedúru na získanie dolnej hranice.

4.4.1 Vylepšenie výpočtu dolnej hranice

Na základe získaných poznatkov o CDA procedúre, ktorá poskytla nepostačujúcu dolnú hranicu pre exaktný algoritmus, som navrhol novú procedúru založenú na inkrementačnom princípe. Tento inkrementačný korekčný duálny algoritmus [12] pre získanie dobrej dolnej hranice má nasledovný predpis:

0. {Inicializácia} Určte počiatočné prípustné duálne riešenie $\mathbf{u}, \mathbf{v}, \mathbf{x}$: $u_i = f_i$ pre $i \in I_V$, $v_j = \min\{c_{ij} : i \in I_V \cup I_1\}$ pre $j \in J$ a $x = 0$. Podľa vzťahu (4.3.31) určte hodnotu DH aktuálnej dolnej hranice.
1. Usporiadajte množinu zákazníkov J do sekvencie $j_1, j_2, \dots, j_k, \dots, j_n$ vzostupne na základe kardinality jednotlivých zákazníkov $K_j = |\{i \in I_V, v_j \geq c_{ij}\}|$.
2. Pre každé $j \in J$ vykonajte nasledujúci postup: Urč $\delta = \min\{\min\{u_i : i \in I_V, v_j \geq c_{ij}\}, \min\{c_{ij} - v_j : i \in I_V, v_j < c_{ij}\}, \min\{c_{ij} - v_j : i \in I_1, v_j < c_{ij}\}\}$. Ak je $\delta > 0$, tak pre každé $i \in I_V$, pre ktoré platí $v_j \geq c_{ij}$, položte $u_i = u_i - \delta$, upravte v_j podľa vzťahu $v_j = v_j + \delta$, zvýš $DH = DH + \delta$ a aktualizujte pozíciu zákazníka j v sekvencii.
3. Opakujte pokiaľ platí, že $|\{u_i = 0, i \in I_V\}| > (p - |I_V|)$. Inak skonči.
4. Zálohujte si hodnoty duálnych premenných x, v_j, u_i , dolnú hranicu DH a aktuálnu sekvenciu zákazníkov $j_1, j_2, \dots, j_k, \dots, j_n$.
5. Nastavte duálne premenné nasledovne: $x = x - 1$ a pre každé $i \in I_V$ hodnotu $u_i = u_i + 1$.
6. Pre každé $j \in J$ vykonajte nasledujúci postup: Určte $\delta = \min\{\min\{u_i : i \in I_V, v_j \geq c_{ij}\}, \min\{c_{ij} - v_j : i \in I_V, v_j < c_{ij}\}, \min\{c_{ij} - v_j : i \in I_1, v_j < c_{ij}\}\}$. Ak je $\delta > 0$, tak pre každé $i \in I_V$, pre ktoré platí $v_j \geq c_{ij}$, polož $u_i = u_i - \delta$, upravte v_j podľa vzťahu $v_j = v_j + \delta$, zvýšte $DH = DH + \delta$ a aktualizujte pozíciu zákazníka j v sekvencii.
7. Ak hodnota dolnej hranice DH sa zvýšila, návrat na krok 3, inak obnovte hodnoty duálnych premenných, dolnej hranice a sekvencie zo zálohy vykonanej v kroku 4 a skončte.

Zmena duálnych premenných v ICDA procedúre nie je viazaná na žiadneho zákazníka

oproti CDA procedúre. Zmena premennej x v CDA procedúre bola určená vzťahom:

$$\Delta x = \min((c_{ij^*} - v_{j^*}) \geq 0, i \in I_v \cup I_1) \quad (4.4.01)$$

Ak porovnáam CDA a ICDA procedúru a zapíšem zmenu duálnych premenných zápisom ako v CDA procedúre, tak tá istá zmena v ICDA je o hodnotu jedna (4.4.02):

$$\Delta x = 1 \quad (4.4.02)$$

ICDA procedúru som verifikoval na tom istom benchmarku BB100x100 a porovnal s CDA procedúrou, kde horná hranica riešenia bola určená hodnotou optimálneho riešenia získaného univerzálnym IP solverom riešením lokačno-alokačného modelu (viď Tab. 16).

Tab. 16 Porovnanie procedúr pre získanie dolnej hranice na benchmarku BB100x100

p	CDA				ICDA			
	DH	HH	$t[s]$	noN	DH	HH	$t[s]$	noN
5	2032	2069	1,58	67	2034	2069	0,321	23
10	1265	1275	5,78	1227	1272	1275	0,121	27
15	923	923	0,02	1	923	923	0,01	1
20	738	745	9,56	3163	741	745	1,307	359
25	616	616	0,01	1	616	616	0,054	1
30	520	521	0,74	87	520	521	0,152	89
35	430	441	3 600*	3674234	440	441	28,419	17073
40	340	373	3 600*	4527215	373	373	0,003	1
45	250	315	3 600*	5103254	314	315	0,109	27
50	160	263	3 600*	5884235	262	263	1098,18	868311
55	70	215	3 600*	6189543	215	215	0,002	1
60	176	176	0,12	1	176	176	0,003	1
65	141	142	3 600*	2186423	142	142	0,003	1
70	106	112	3 600*	5664432	112	112	0,004	1
75	71	87	3 600*	8795683	87	87	0,001	1
80	36	64	3 600*	11045197	64	64	0,002	1
85	1	44	3 600*	12635231	44	44	0,002	1
90	12	25	3 600*	15223646	25	25	0,001	1
95	4	10	86,06	479555	10	10	0,001	1

¹ stĺpec p udáva hodnotu maximálneho počtu stredísk

² stĺpec DH – hodnota dolnej hranice koreňa pre jednotlivé procedúry

³ stĺpec HH udáva hornú hranicu riešenia prednostne nastavenú na hodnotu optimálneho riešenia

⁴ stĺpec $t[s]$ udáva čas trvania výpočtu algoritmu pre jednotlivé procedúry

⁵ stĺpec noN udáva počet spracovaných vrcholov v metóde vetiev a hraníc pre jednotlivé procedúry

⁶ stĺpce CDA udávajú výstupné parametre výpočtu pre CDA procedúru

⁷ stĺpce ICDA udávajú výstupné parametre výpočtu pre ICDA procedúru

⁸ červená farba – predčasné zastavenie algoritmu po 1 hodine výpočtu

⁹ zelená farba – získanie lepšej dolnej hranice koreňa

¹⁰ oranžová farba – lepší výpočtový čas

Na základe vykonaných experimentov (viď Tab. 16) môžem skonštatovať, že ICDA procedúra nám poskytuje lepšiu dolnú hranicu v porovnaní s dolnou hranicou získanou

pomocou CDA procedúry. To možno demonštrovať lepším výpočtovým časom ako aj menším počtom spracovaných vrcholov. Na základe tejto skutočnosti je ICDA procedúra na získanie dolnej hranice riešenia vhodnejšia pre exaktný algoritmus ako CDA procedúra. Experimenty však ukázali, že v niektorých prípadoch som musel spracovať veľké množstvo vrcholov, (viď p rovné 50), aj keď oveľa menej ako predtým. Otázkou je, čo viedlo k tejto anomálii.

4.4.2 Vplyv fixácie na získanie dobrej dolnej hranice

Algoritmus pMedBBDual využíva pri vytváraní stromu riešenia odloženie nespracovávaného vrcholu do zásobníka doposiaľ nespracovaných vrcholov spolu s hodnotami duálnych premenných v_j a x . Premenné u_i si dokážem dodatočne prepočítať. Tieto vrcholy pri backtrackingu vyťahujem a spracovávam ich. Treba si však uvedomiť, že pri spracovávaní vrcholov sa snažím nájsť kandidáta na umiestnenie strediska, na základe fixácie ktorého budú vytvorené ďalšie dva nové vrcholy. V duálnom modeli sa fixácia na hodnotu jedna prejavuje zmenou podmienky (4.3.26) na (4.3.27). Pokiaľ kandidát na umiestnenie strediska patrí do množiny nefixovaných stredísk musím spĺňať podmienky (4.2.26), fixáciou na jedna sa to mení na spĺňanie podmienky (4.3.27). Z čoho vyplýva aktualizácia hodnôt duálnych premenných v_j podľa vzťahu (4.4.03).

$$v_j = c_{ij} \quad \text{pre } i \in I_1, \quad j \in J \quad (4.4.03)$$

Pokúsil som sa neukladať tieto hodnoty do zásobníka a každý vrchol spracovať ako koreň stromu riešenia, teda z východiskovej pozície (4.4.04 – 4.4.06).

$$v_j = \min \{c_{ij} : i \in I_V \cup I_1\} \quad \text{pre } j \in J \quad (4.4.04)$$

$$u_i = f_i \quad \text{pre } i \in I_V \quad (4.4.05)$$

$$x = 0 \quad (4.4.06)$$

Týmto som chcel overiť potrebnosť ukladania duálnych premenných v_j a x do zásobníka. To som experimentálne realizoval na benchmarku z Banskobystrického kraja vo výpočtovom čase a počte spracovaných vrcholov. V Tab. 17 sú uvedené výsledky spomínaného porovnania, ktoré ukázali, že ukladanie hodnôt duálnych premenných do zásobníka nie je potrebné. Je výhodnejšie vychádzať z východiskovej pozície (4.4.04) –

(4.4.06) pri získavaní dolnej hranice vrcholu v metóde vetiev a hraníc, čo je demonštrované menším počtom spracovaných vrcholov a kratším výpočtovým časom.

Tab. 17 Porovnanie efektívnosti získania dolnej hranice na benchmarku BB100x100

p	$pMedBBDual$ (Zasobník)				$pMedBBDual(Start)$			
	DH	HH	$t[s]$	nN	DH	HH	$t[s]$	nN
5	2034	2069	0,321	23	2034	2069	1,482	51
10	1272	1275	0,121	27	1272	1275	0,133	17
15	923	923	0,01	1	923	923	0,005	1
20	741	745	1,307	359	741	745	0,602	161
25	616	616	0,054	1	616	616	0,004	1
30	520	521	0,152	89	520	521	0,220	85
35	440	441	28,419	17073	440	441	0,149	87
40	373	373	0,003	1	373	373	0,002	1
45	314	315	0,109	27	314	315	0,034	19
50	262	263	1098,18	868311	262	263	0,050	33
55	215	215	0,002	1	215	215	0,003	1
60	176	176	0,003	1	176	176	0,002	1
65	142	142	0,003	1	142	142	0,003	1
70	112	112	0,004	1	112	112	0,002	1
75	87	87	0,001	1	87	87	0,001	1
80	64	64	0,002	1	64	64	0,002	1
85	44	44	0,002	1	44	44	0,001	1
90	25	25	0,001	1	25	25	0,003	1
95	10	10	0,001	1	10	10	0,001	1

¹ stĺpec p udáva hodnotu maximálneho počtu stredísk

² stĺpec DH – hodnota dolnej hranice koreňa

³ stĺpec HH – hodnota hornej hranice riešenia

⁴ stĺpec $t(s)$ udáva čas trvania výpočtu algoritmu

⁵ stĺpec nN udáva počet spracovaných vrcholov v metóde vetiev a hraníc

⁶ stĺpce $pMedBBDual(Zasobník)$ udávajú využívanie hodnôt zo zásobníka

⁷ stĺpce $pMedBBDual(Start)$ udávajú využívanie východiskového nastavenia

⁸ červená farba udáva nežiadúci počet spracovaných vrcholov

⁹ zelená farba udáva menší počet spracovaných vrcholov

¹⁰ oranžová farba – dolná hranica je rovná hodnote optimálneho riešenia

4.4.3 Výber kandidáta na vetvenie

V kapitole 4.3.6 som popisoval výber kandidáta na vetvenie, ktorý je založený na postupnom vyhodnocovaní dodržania komplementárnych podmienok (4.4.07) – (4.4.09).

$$(y_i - z_{ij}) \max\{0, v_j - c_{ij}\} = 0 \quad \text{pre } i \in I_V, \quad j \in J \quad (4.4.07)$$

$$u_i y_i = 0 \quad \text{pre } i \in I_V \quad (4.4.08)$$

$$\max\{0, c_{ij} - v_j\} z_{ij} = 0 \quad \text{pre } i \in I_V, \quad j \in J \quad (4.4.09)$$

Na základe toho bol skonštruovaný algoritmus pre výber kandidáta na vetvenie, kde vybraný kandidát je fixovaný na hodnotu nula a jedna, čím boli vytvorené nové vrcholy

v strome riešení. Tento navrhnutý algoritmus popísaný v kapitole 4.3.6 vyhodnocuje podmienky v nasledovnom poradí:

1. Dodržanie podmienok (4.4.07)
2. Dodržanie podmienok (4.4.08)
3. Dodržanie podmienok (4.4.09)

Naskytuje sa tu otázka, či dané poradie vyhodnotenia podmienok je najvhodnejšie pre výber kandidáta na vetvenie [11]. Preto som pristúpil k vytvoreniu všetkých možných kombinácií vyhodnotenia komplementárnych podmienok (4.4.07) – (4.4.09), ktoré som označil ako varianty funkcie pre získanie kandidáta na vetvenie a zhrnul v Tab. 18.

Tab. 18 Poradie vyhodnotenia dodržania komplementárnych podmienok pre jednotlivé varianty

Variant	Poradie vyhodnotenia dodržania komplementárnych podmienok		
	1.	2.	3.
V1	(4.3.07)	(4.3.08)	(4.3.09)
V2	(4.3.07)	(4.3.09)	(4.3.08)
V3	(4.3.09)	(4.3.08)	(4.3.07)
V4	(4.3.08)	(4.3.09)	(4.3.07)
V5	(4.3.08)	(4.3.07)	(4.3.09)
V6	(4.3.09)	(4.3.07)	(4.3.08)

¹ stĺpec *Variant* udáva jednotlivé varianty

² stĺpec 1. udáva, ktorá podmienka sa vyhodnotí ako prvá

³ stĺpec 2. udáva, ktorá podmienka sa vyhodnotí ako druhá

⁴ stĺpec 3. udáva, ktorá podmienka sa vyhodnotí ako tretia

Variant V1 predstavuje funkciu s pôvodným usporiadaním vyhodnotenia podmienok popísaných v kapitole 4.3.6 v krokoch 2, 3 a 4 navrhnutého postupu pre získanie kandidáta na vetvenie. Tento variant V1 porovnam s ostatnými variantami na základe výpočtového času, kde funkcia pre získanie kandidáta sa bude s pôvodnou líšiť iba v spomínaných krokoch 2, 3 a 4. Experimenty na overenie vhodnosti použitej funkcie pre získanie kandidáta na vetvenia oproti ostatným funkciám realizujem na benchmarkoch BB100x100, ZA315x315, NR350x350 zo slovenskej cestnej siete.

Experimenty na BB100x100 ukázali malé časové rozdiely, na základe ktorých nie je možné vyvodiť adekvátny záver (viď príloha 4). Do Tab. 19 som vybral výsledky z benchmarku ZA315x315 odpovedajúce Žilinskému kraju, kde najmarkantnejšie vidieť rozdiely vo výpočtových časoch jednotlivých variantov funkcie.

Tab. 19 Porovnanie variantov funkcie vo výpočtovom čase na benchmarku ZA315x315

<i>p</i>	Výpočtový čas v sekundách [s]					
	V1	V2	V3	V4	V5	V6
24	511	644	928	105	81	914
27	151	94	501	451	403	490
30	78	125	206	546	379	178
51	843	1 222	918	273	424	916
54	194	160	2 035	743	861	2 027
55	317	312	3 600*	1 046	1 059	3 600*
58	3 600*	2 118	3 600*	1 468	1 729	3 600*
59	3 600*	3 058	2 721	1 102	899	2 676
60	3 313	3 600*	1 954	614	495	1 962
62	710	1 205	3 600*	1 040	1 075	3 600*
63	591	641	2 242	484	462	2 217
65	524	891	3 600*	713	597	3 600*

¹ stĺpec *p* udáva hodnotu maximálneho počtu stredísk

² stĺpec V1 udáva výpočtový čas algoritmu získaný variantom V1

³ stĺpec V2 udáva výpočtový čas algoritmu získaný variantom V2

⁴ stĺpec V3 udáva výpočtový čas algoritmu získaný variantom V3

⁵ stĺpec V4 udáva výpočtový čas algoritmu získaný variantom V4

⁶ stĺpec V5 udáva výpočtový čas algoritmu získaný variantom V5

⁷ stĺpec V6 udáva výpočtový čas algoritmu získaný variantom V6

⁸ červená farba – predčasné zastavenie algoritmu po 1 hodine výpočtu (3600*)

⁹ zelená farba – najkratší výpočtový čas zo všetkých variantov

Experimenty ukázali, že najkratší výpočtový čas algoritmu sme získali s rôznymi variantmi funkcie pre získanie kandidátov na vetvenie. Na druhej strane výpočtový čas v niektorých prípadoch prevýšil jednu hodinu, čo vzhľadom na veľkosť riešenej úlohy nepovažujem za priaznivé. Variantmi V4 a V5 som získal vo všetkých prípadoch výpočtový čas menší ako hodina, čo predurčuje tieto varianty, aby som z nich vybral jeden ako najvhodnejší variant. Na základe vykonaných experimentov, ale nebolo možné jasne vybrať jeden variant, preto som pristúpil k štatistickému porovnaniu jednotlivých variantov.

Na určenie najvhodnejšieho variantu som zaviedol tieto štatistiky:

1. Priemerný čas výpočtu – aritmetický priemer časov výpočtu algoritmu pre všetky hodnoty *p*.

$$t_A = \left(\sum_{p=1}^m t_p \right) / m \quad (4.4.10)$$

2. Relatívny čas výpočtu – suma pomeru času výpočtu variantu k variantu s najlepším časom výpočtu.

$$t_R = \sum_{p=1}^m (t_p / \min_{F1..F7} t_p) \quad (4.4.11)$$

3. Priemerný počet získaných kandidátov - aritmetický priemer počtu získaných kandidátov v algoritme pre všetky hodnoty p .

$$NoC_A = \left(\sum_{p=1}^m NoC_p \right) / m \quad (4.4.12)$$

4. Relatívny počet získaných kandidátov - suma pomeru počtu získaných kandidátov variantu k variantu s najnižším počtom získaných kandidátov.

$$NoC_R = \sum_{p=1}^m (NoC_p / \min_{F1..F6} NoC_p) \quad (4.4.13)$$

Na základe týchto štatistík chcem porovnať v Tab. 20 jednotlivé varianty funkcie pre získanie kandidáta na vetvenie a vybrať najvhodnejší variant pre navrhnutý exaktný algoritmus.

Tab. 20 Vyhodnotenie štatistík na úlohe BB100x100, ZA315x315 a NR350x350

Variant	BB100x100				ZA315x315				NR350x350			
	NoC _A	NoC _R	t _A	t _R	NoC _A	NoC _R	t _A	t _R	NoC _A	NoC _R	t _A	t _R
V1	162,04	191,71	0,39	123,70	3699	992	142	1244	16870	516	552	535
V2	430,26	462,51	0,58	161,02	4560	1025	178	1304	34846	3441	1143	4740
V3	171,00	211,54	0,50	152,58	10909	1553	312	2054	36035	4098	1183	4408
V4	165,57	194,96	0,46	132,91	1854	480	75	510	12013	443	357	426
V5	166,33	195,67	0,51	146,21	1923	490	72	469	12352	427	355	403
V6	168,98	207,06	0,54	163,89	10792	1431	288	1996	36543	4145	1175	4395

¹ stĺpec Variant udáva jednotlivé varianty

² stĺpec NoC_A udáva, priemerný počet získaných kandidátov počas vetvenia

³ stĺpec NoC_R udáva relatívny počet získaných kandidátov počas vetvenia

⁴ stĺpce t_A udáva priemerný čas výpočtu

⁵ stĺpce t_R udáva relatívny čas výpočtu

⁶ zelená farba – najvhodnejší variant na základe danej štatistiky

Na základe vykonaných štatistík som zistil, že variant V1 bol najvhodnejší pri rozsahovo malej úlohe, no keď sme začali testovať rozsahovo väčšie úlohy, vhodnosť tohto variantu sa nepotvrdila. Je vidieť, že variant V1 sa oproti ostatným variantom v štatistikách úlohy BB100x100 veľmi nelíši. Čo však nemožno povedať o variantoch V4 a V5 voči ostatným variantom pre rozsahovo väčšiu úlohu. Pre úlohách ZA315x315, NR350x350 je vidieť niekoľko násobne menšie hodnoty štatistík pri variantoch V4 a V5. Spoločným znakom variantov V4 a V5 je vyhodnotenia komplementárnych podmienok (4.4.14) ako prvých v poradí, teda či hodnota doplnkovej premennej u_i je nulová.

$$u_i y_i = 0 \quad \text{pre } i \in I_V \quad (4.4.14)$$

Pri štatistikách, kde som bral do úvahy počet získaných kandidátov, ako najvhodnejší variant je variant V4. Avšak pri štatistikách, kde sme brali do úvahy výpočtový čas, ako najvhodnejší variant je variant V5. Rozdiely v štatistikách medzi variantmi V4 a V5 nie sú veľké, teda výber najvhodnejšieho kandidáta je založený na dôležitosti jednotlivých štatistických kritérií.

Na potvrdenie vhodnosti vybraných variantov som využil štatistický test, konkrétne dvojvýberový t-test s nerovnosťou rozptylov (t-Test: Two-Sample Assuming Unequal Variances) podrobne popísaný v zdroji [40]. Testy som realizoval pomocou nástroja *Data Analysis* v prostredí *Microsoft Excel* [3] na hladine významnosti $\alpha = 0,05$, kde boli porovnávané dvojice variantov navzájom. Podrobné výsledky štatistických testov sú uvedené v prílohe 5. V Tab. 21 vyhodnocujem výhodnosť variantov funkcie medzi sebou na základe vyhodnotenia hypotézy a jej alternatív v dvojvýberovom t-teste. To znamená, že je uvedený ten variant z testovanej dvojice variantov (variant z riadka alebo variant zo stĺpca), ktorý je na základe vyhodnotenia štatistického testu vhodnejší pre algoritmus pMedBBDual. V prípade ak sa varianty rovnajú, je uvedený vzťah rovnosti.

Tab. 21 Porovnanie vhodnosti jednotlivých variantov funkcie na základe dvojvýberového t-testu

<i>Variant</i>	V1	V2	V3	V4	V5	V6
V1	-	V1	V1	V4	V5	V1
V2	V1	-	V2	V4	V5	V2
V3	V1	V2	-	V4	V5	V3=V6
V4	V4	V4	V4	-	V4=V5	V4
V5	V5	V5	V5	V5=V4	-	V5
V6	V1	V2	V6=V3	V4	V5	-

¹ stĺpec *Variant* udáva jednotlivé varianty

² Symbol - udáva, že nie je možné testovať rovnaké varianty medzi sebou

Štatistický test potvrdil predošlé závery, že najvhodnejšie varianty funkcie pre získanie kandidáta na vetvenie sú V4 a V5. Keďže je prvoradé získanie optimálneho riešenia za čo najkratší čas, najvhodnejší variant získania kandidáta na vetvenie predstavuje variant V5.

4.4.4 Vylepšenie hornej hranice

V kapitole 4.3.4 som popísal procedúru pre získanie minimálnej množiny umiestnení založenú na konštrukcii primárneho riešenia s minimálnym rozdielom *gap* medzi hornou a dolnou hranicou:

$$\begin{aligned}
gap = F_p(y, z) - F_D(u, v, x) &= \sum_{i \in I_V} \sum_{j \in J} (y_i - z_{ij}) \max\{0, v_j - c_{ij}\} + \sum_{i \in I_V} u_i y_i + \\
&+ \left(\sum_{i \in I_V} y_i - p + |I_1| \right) x + \sum_{i \in I_V} \sum_{j \in J} \max\{0, c_{ij} - v_j\} z_{ij} + \sum_{i \in I_1} \sum_{j \in J} (c_{ij} - v_j) z_{ij} \quad (4.4.15)
\end{aligned}$$

Tento *gap* udávajú nespĺnené komplementárne podmienky. Vychádzam z duálneho riešenia, kde cez dodržanie komplementárnych podmienok (4.4.16 – 4.4.20) konštruujem primárne riešenie.

$$(y_i - z_{ij}) \max\{0, v_j - c_{ij}\} = 0 \quad \text{pre } i \in I_V, \quad j \in J \quad (4.4.16)$$

$$\left(\sum_{i \in I_V} y_i - p + |I_1| \right) x = 0 \quad (4.4.17)$$

$$u_i y_i = 0 \quad \text{pre } i \in I_V \quad (4.4.18)$$

$$\max\{0, c_{ij} - v_j\} z_{ij} = 0 \quad \text{pre } i \in I_V, \quad j \in J \quad (4.4.19)$$

$$(c_{ij} - v_j) z_{ij} = 0 \quad \text{pre } i \in I_1, \quad j \in J \quad (4.4.20)$$

Získanie primárneho riešenia závisí aj od poradia spracovaných zákazníkov. Na základe rozličného poradia zákazníkov som navrhol varianty funkcie pre získanie minimálnej množiny umiestnení [13]:

1. Variant F1 predstavuje Erlenkotterovo usporiadanie zákazníkov. Tento variant odpovedá algoritmu, ktoré som navrhol a podrobne popísal v kapitole 4.3.4.
2. Variant F2 predstavuje preberanie zákazníkov po poradí ako boli zákazníci usporiadaný na vstupe.
3. Variant F3 predstavuje preberanie zákazníkov v poradí, ktoré som získal ICDA procedúrou.
4. Variant F4 predstavuje preberanie zákazníkov v náhodnom poradí.
5. Variant F5 predstavuje vykonanie variantu F4 dvakrát po sebe a výber riešenia s lepšou (menšou) hornou hranicou.

6. Variant F6 predstavuje preberanie zákazníkov po poradí, avšak vyberá sa z maximálne 3 najlepších stredísk pre daného zákazníka.

Experimenty na overenie vhodnosti použitej funkcie pre získanie minimálnej množiny umiestnení, ktorej hodnota udáva hornú hranicu vrcholu realizujem na benchmarkoch BB100x100, ZA315x315, TT276x276 zo slovenskej cestnej siete. Experimenty na benchmarku BB100x100 ukázali malé časové rozdiely, na základe ktorých nie je možné vyvodiť adekvátny záver (viď príloha 6). Do Tab. 22 som vybral výsledky z benchmarku ZA315x315 odpovedajúce Žilinskému kraju, kde najmarkantnejšie vidieť rozdiely vo výpočtových časoch jednotlivých variantov funkcie.

Tab. 22 Porovnanie variantov funkcie vo výpočtovom čase na benchmarku ZA315x315

<i>p</i>	Výpočtový čas (s)					
	<i>F1</i>	<i>F2</i>	<i>F3</i>	<i>F4</i>	<i>F5</i>	<i>F6</i>
1	4,27	264,71	383,06	7,65	8,24	4,16
9	42,96	266,27	365,07	197,13	113,25	68,72
15	195,21	422,47	613,78	1 193,09	105,03	84,42
27	475,84	729,19	653,10	504,02	416,62	223,07
44	42,53	29,65	42,14	87,48	15,44	10,07
50	326,47	522,17	3 600*	1 188,26	760,31	207,77
68	98,71	3 600*	3 600*	3 600*	3 600*	3 600*
101	23,92	8,30	3 600*	3 600*	206,26	5,10
106	24,08	12,99	9,40	19,39	4,33	724,10
120	36,25	3 600*	3 600*	10,40	3 600*	3 600*
150	20,66	3 600*	3 600*	3 600*	3 600*	3 600*
180	16,67	3 600*	3 600*	3 600*	3 600*	3 600*
200	14,12	8,86	3 600*	8,19	3 600*	10,87
220	13,16	7,58	6,71	5,23	19,33	5,50
250	9,34	4,23	4,83	0,13	4,56	5,50
280	7,52	7,72	7,38	7,45	4,63	5,91
300	5,95	0,07	0,01	0,01	4,16	3,36

¹ stĺpec *p* udáva hodnotu maximálneho počtu stredísk

² stĺpec *F1* udáva výpočtový čas algoritmu získaný variantom F1

³ stĺpec *F2* udáva výpočtový čas algoritmu získaný variantom F2

⁴ stĺpec *F3* udáva výpočtový čas algoritmu získaný variantom F3

⁵ stĺpec *F4* udáva výpočtový čas algoritmu získaný variantom F4

⁶ stĺpec *F5* udáva výpočtový čas algoritmu získaný variantom F5

⁷ stĺpec *F6* udáva výpočtový čas algoritmu získaný variantom F6

⁸ červená farba – predčasné zastavenie algoritmu po 1 hodine výpočtu (3600*)

⁹ zelená farba – najkratší výpočtový čas zo všetkých variantov

Experimenty ukázali, že najkratší výpočtový čas algoritmu som získal s rôznymi variantami funkcie pre získanie kandidátov na vetvenie. Variant V2 neposkytol ani v jednom prípade lepší výpočtový čas, z čoho je možné usúdiť, že tento variant nebude najvhodnejší. Na druhej strane výpočtový čas v niektorých prípadoch prevýšil jednu hodinu, čo vzhľadom na veľkosť riešenej úlohy nepovažujem za priaznivé. Variant V1 ako jediný poskytol

získanie optimálneho riešenia vo všetkých prípadoch, keďže variantom V1 som získal vo všetkých prípadoch výpočtový čas menší ako jedna hodina. Čo variant V1 predurčuje, aby bol najvhodnejší. Na základe vykonaných experimentov, ale nebolo možné jasne vybrať jeden variant, preto som pristúpil k štatistickému porovnaniu jednotlivých variantov. Na určenie najvhodnejšieho variantu som zaviedol tieto štatistiky:

1. Priemerný čas výpočtu – aritmetický priemer časov výpočtu algoritmu pre všetky hodnoty p .

$$t_A = \left(\sum_{p=1}^m t_p \right) / m \quad (4.4.21)$$

2. Relatívny čas výpočtu – suma pomeru času výpočtu variantu k variantu s najlepším časom výpočtu.

$$t_R = \sum_{p=1}^m (t_p / \min_{F1..F7} t_p) \quad (4.4.22)$$

3. Priemerný počet spracovaných vrcholov - aritmetický priemer počtu spracovaných vrcholov v algoritme pre všetky hodnoty p .

$$NoN_A = \left(\sum_{p=1}^m NoN_p \right) / m \quad (4.4.23)$$

4. Relatívny počet spracovaných vrcholov - suma pomeru počtu spracovaných vrcholov variantu k variantu s najnižším počtom spracovaných vrcholov.

$$NoN_R = \sum_{p=1}^m (NoN_p / \min_{F1..F6} NoN_p) \quad (4.4.24)$$

Pomocou štatistík chcem porovnať v Tab. 23 jednotlivé varianty funkcie pre získanie minimálnej množiny umiestnení a vybrať najvhodnejší variant pre navrhnutý exaktný algoritmus.

Na základe vykonaných štatistík (viď Tab. 23) som zistil, že už pri malých úlohách BB100x100 variant F3 nie je vhodný nakoľko skúmané hodnoty boli oproti ostatným variantom niekoľkonásobne vyššie. No čo sa týka výberu najvhodnejšieho variantu z ostatných variantov nie je možné určiť práve jeden variant. Preto som sa zameral na

rozsahovo väčšie úlohy, ktoré odpovedajú Žilinskému a Trenčianskemu kraju (ZA315x315, TN276x276). Štatistiky pri rozsahovo väčších úlohách ukázali, že variant V1 je najvhodnejší pre exaktný algoritmus pMedBBDual. Vhodnosť variantu V1 je vidieť, v mnohonásobne lepších štatistikách oproti ostatným variantom.

Tab. 23 Vyhodnotenie štatistík pre úlohy BB100x100, ZA315x315 a TN276x276

Variant	BB100x100				ZA315x315				TN276x276			
	NoN _A	NoN _R	t _A	t _R	NoN _A	NoN _R	t _A	t _R	NoN _A	NoN _R	t _A	t _R
F1	333,66	4588	0,70	693	3847	29162	76	1307	1095	26129	15	833
F2	565,76	1376	0,93	540	121402	28512	1614	21134	100571	223602	780	15756
F3	9744,74	6002	9,67	5047	137740	39605	1822	28873	154894	363417	1374	40095
F4	2213,82	1438	2,83	1394	121319	28852	1678	22786	124853	289265	1113	26286
F5	1027,78	700	1,73	849	174087	81448	1794	30917	122992	293208	828	17365
F6	505,20	3108	0,67	579	70853	21087	1526	20104	132301	311182	947	20166

¹ stĺpec Variant udáva jednotlivé varianty

² stĺpec NoN_A udáva, priemerný počet spracovaných vrcholov

³ stĺpec NoN_R udáva relatívny počet spracovaných vrcholov

⁴ stĺpce t_A udáva priemerný čas výpočtu

⁵ stĺpce t_R udáva relatívny čas výpočtu

⁶ zelená farba – najvhodnejší variant na základe danej štatistiky

Na potvrdenie vhodnosti vybraného variantu som využil štatistický test, konkrétne dvojvýberový t-test s nerovnosťou rozptylov (t-Test: Two-Sample Assuming Unequal Variances) podrobne popísaný v zdroji [40]. Testy som realizoval pomocou nástroja *Data Analysis* v prostredí *Microsoft Excel* [3] na hladine významnosti $\alpha = 0,05$, kde boli porovnávané dvojice variantov navzájom. Podrobné výsledky štatistických testov sú uvedené v prílohe 7. V Tab. 24 vyhodnocujem vhodnosť variantov funkcie medzi sebou na základe vyhodnotenia hypotézy a jej alternatív v dvojvýberovom t-teste (viď príloha 7). To znamená, že je uvedený ten variant z testovanej dvojice variantov (variant z riadka alebo variant zo stĺpca), ktorý je na základe vyhodnotenia štatistického testu vhodnejší pre algoritmus pMedBBDual. V prípade ak sa varianty rovnajú je uvedený vzťah rovnosti.

Tab. 24 Porovnanie vhodnosti jednotlivých variantov funkcie na základe dvojvýberového t-testu

Variant	F1	F2	F3	F4	F5	F6
F1	-	F1	F1	F1	F1	F1
F2	F1	-	F2=F3	F2=F4	F2=F5	F2=F6
F3	F1	F3=F2	-	F3=F4	F3=F5	F6
F4	F1	F4=F2	F4=F3	-	F4=F5	F4=F6
F5	F1	F5=F2	F5=F3	F5=F4	-	F6
F6	F1	F6=F2	F6	F6=F4	F6	-

¹ stĺpec Variant udáva jednotlivé varianty

² Symbol – udáva, že nie je možné testovať rovnaké varianty medzi sebou

Štatistický test potvrdil predošlé závery, že funkcia FI pre získanie minimálnej množiny umiestnení založená na Erlenkotterovom usporiadaní zákazníkov, ktorú som popísal v kapitole 4.3.4, je najvhodnejšia. Nepodarilo sa nájsť vhodnejší variant funkcie, ktorý by poskytol lepšie výsledky, no na druhej strane som potvrdil vhodnosť navrhnutého algoritmu z kapitoly 4.3.4.

4.4.5 Porovnanie algoritmov so zovšeobecným Erlenkotterovým prístupom

Porovnanie algoritmov založených na zovšeobecnení Erlenkotterovho prístupu som realizoval na Slovenských krajoch a na celej cestnej sieti Slovenska s vybranými hodnotami maximálneho počtu umiestnených stredísk p na základe predpisu (4.4.25), kde hodnota p_1 predstavuje prvú hodnotu p , q udáva rozmedzie medzi jednotlivými hodnotami p , k udáva krok, v ktorom počítam dané p a *počet* udáva počet vykonaných experimentov na testovanom benchmarku.

$$p = p_1 + q(k - 1) \quad k = 1, \dots, \text{počet} \quad (4.4.25)$$

Hodnoty parametrov p_1 , q a *počet*, ktoré sú uvedené v Tab. 25, boli zvolené tak, aby som obsiahol rozsah celého testovaného benchmarku a počet experimentov na danom benchmarku bol prispôsobený veľkosti riešenej úlohy pre možné vyvodenie záveru. V Tab. 25 sú podrobne popísané testované kraje spolu s ich označením, veľkosťou a nastavenými parametrami.

Tab. 25 Popis vykonaných experimentov na slovenskej cestnej sieti

Testovaný kraj	Benchmark	I	J	Testované p		
				p_1	q	počet
Bratislavský	BA87x87	87	87	5	5	15
Banskobystrický	BB515x515	515	515	15	20	25
Košický	KE460x460	460	460	15	20	23
Prešovský	PO664x664	664	664	15	20	33
Nitriansky	NR350x350	350	350	15	15	23
Trenčiansky	TN276x276	276	276	15	15	18
Trnavský	TT249x249	246	246	15	15	16
Žilinský	ZA315x315	315	315	15	15	20
Slovensko	CSR2916x2916	2916	2916	50	50	58

¹ stĺpec *Testovaný kraj* udáva kraj Slovenska, ktorého cestná sieť bola použitá

² stĺpec *Benchmark* udáva označenie dát pre testovanú úlohu

³ stĺpec *|I|* udáva mohutnosť množiny kandidátov na umiestnenie strediska

⁴ stĺpec *|J|* udáva mohutnosť množiny zákazníkov

⁵ stĺpec p_1 udáva prvé testované p

⁶ stĺpec q udáva rozmedzie medzi jednotlivými testovanými hodnotami p

⁷ stĺpec *počet* udáva počet testovaných hodnôt p pre najlepší výpočtový čas v testovanom benchmarku pre danú úlohu

Základný algoritmus pMedBBDual je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.3, pozostávajúci hlavne z CDA procedúry na získanie dolnej hranice (viď kapitola 4.3.3), spôsobom spracovania vrcholu z uložených hodnôt duálnych premenných v zásobníku, algoritmu pre získanie minimálnej množiny umiestnených stredísk (viď 4.3.4) a algoritmu pre získanie kandidáta na vetvenie (viď kapitola 4.3.6).

Vylepšený algoritmus pMedBBDual je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.4, pozostávajúci hlavne z ICDA procedúry na získanie dolnej hranice (viď kapitola 4.4.1), spôsobu spracovania vrcholu stromu vetvenia s počiatočným nastavením duálnych premenných (viď kapitola 4.4.2), algoritmu pre získanie minimálnej množiny umiestnených stredísk FI (viď 4.4.4) a algoritmu pre získanie kandidáta na vetvenie $V5$ (viď kapitola 4.4.3).

Experimenty realizované na benchmarkoch slovenskej cestnej siete pre jednotlivé kraje a celé Slovensko sú uvedené v prílohe 8. Vyhodnotenie experimentov je uvedené v Tab. 26 v podobe PoR počtu získaných optimálnych riešení k celkovému počtu vykonaných experimentov pre testovaný benchmark v percentách (4.4.26), percentuálnom vyjadrení relatívnej odchýlky Gap medzi získanou hodnotou riešenia HH a hodnotou optimálneho riešenia ES v priemere (4.4.27), priemerného výpočtového času pt z časov t jednotlivých experimentov (4.4.28) a počtu predčasných ukončení algoritmu po uplynutí jednej hodiny výpočtového času ppU . Parameter $pocet$ udáva počet vykonaných experimentov a p_{pocet} udáva posledné testované p v benchmarku získané vzťahom (4.4.25). Parameter NoR udáva počet optimálnych riešení pre testovaný benchmark.

$$PoR = \frac{NoR}{pocet} * 100 \quad (4.4.26)$$

$$Gap = \left(\sum_{p=p_1}^{p_{pocet}} \frac{HH_p - ES_p}{ES_p} * 100 \right) / pocet \quad (4.4.27)$$

$$pt = \left(\sum_{p=p_1}^{p_{pocet}} t_p \right) / pocet \quad (4.4.28)$$

Tab. 26 Porovnanie algoritmov založených na zovšeobecnenom Erlenkotterovom prístupe

<i>pMedBBDual</i>		ZÁKLADNÝ				VYLEPŠENÝ			
úloha	Testovaný kraj	PoR [%]	Gap [%]	pt[s]	ppU	PoR [%]	Gap [%]	pt[s]	ppU
p-medián	Bratislavský	80	0,69	1153,33	5	100	0,00	0,15	0
	Banskobystrický	80	0,34	2097,54	6	84	0,01	898,30	5
	Košický	87	0,06	1565,26	5	87	0,05	1804,65	11
	Prešovský	73	2,64	3301,86	28	85	0,03	1241,65	10
	Nitriansky	74	1,15	1270,80	6	96	0,00	577,17	3
	Trenčiansky	78	0,51	1040,56	5	100	0,00	4,84	0
	Trnavský	94	0,45	597,15	2	100	0,00	361,59	1
	Žilinský	50	4,68	2083,02	11	100	0,00	125,43	0
	Slovensko	-	-	3600,00	58	-	-	3487,59	56
vážený p-medián	Bratislavský	94	0,49	868,05	4	100	0,00	0,39	0
	Banskobystrický	60	3,90	2505,25	12	100	0,00	207,92	0
	Košický	39	7,48	2989,27	17	100	0,00	147,70	0
	Prešovský	27	6,14	3380,25	29	97	0,00	762,86	3
	Nitriansky	43	8,83	2279,40	14	100	0,00	134,37	0
	Trenčiansky	56	4,87	2226,30	11	100	0,00	9,15	0
	Trnavský	75	0,83	939,89	4	100	0,00	6,98	0
	Žilinský	50	9,14	2221,46	12	100	0,00	13,43	0
	Slovensko	2	20,23	3600,00	58	0	17,90	3600,00	58

¹ stĺpec *úloha* udáva typ riešenej úlohy

² stĺpec *Testovaný kraj* udáva kraj Slovenska, ktorého cestná sieť bola použitá

³ stĺpec *PoR [%]* udáva počet získaných optimálnych riešení k celkovému počtu vykonaných experimentov v percentách

⁴ stĺpec *Gap [%]* udáva percentuálne vyjadrenie relatívnej odchýlky medzi hodnotou *HH* a hodnotou *ES* v priemere

⁵ stĺpec *pt[s]* udáva priemerný čas výpočtu v sekundách

⁶ stĺpec *ppU* udáva počet predčasných ukončení algoritmu

⁷ stĺpec *Základný pMedBBDual* udávajú získanie riešenia základným algoritmom pMedBBDual

⁸ stĺpce *Vylepšený pMedBBDual* udávajú získanie riešenia vylepšeným algoritmom pMedBBDual

⁹ zelená farba – najlepšia hodnota parametra *PoR*

¹⁰ oranžová farba – získanie najlepšieho výpočtového času

¹¹ Symbol – nepoznáme hodnotu štatistiky nakoľko nepoznáme optimálne riešenia pre danú úlohu

Štatistiky uvedené v Tab. 26 ukazujú, že vylepšený algoritmus je v porovnaní so základným algoritmom lepší a to vo všetkých meraných štatistikách. Výnimkou je Košický kraj pri úlohe p-mediánu, kde základný algoritmus bol v priemernom čase *pt* lepší, čo sa odzrkadlilo menším počtom predčasných zastavení algoritmu *ppU*, avšak v získaní optimálnych riešení *PoR* sa to neprejavilo, dokonca *Gap* bol lepší v prípade vylepšeného algoritmu, čo znamená získanie kvalitnejších riešení vylepšeným algoritmom. V ostatných prípadoch pri úlohe p-mediánu vylepšený algoritmus je lepší vo všetkých meraných štatistikách, aj keď sa stalo, že algoritmus musel byť predčasne ukončený. Pri úlohe váženého p-mediánu je vylepšený algoritmus jednoznačne lepší, či už z časového hľadiska alebo kvality získaného riešenia.

4.5 Porovnanie iteratívneho a zovšeobecneného prístupu

Porovnanie iteratívneho a zovšeobecneného algoritmu som realizoval na Beasleyho úlohách [6], slovenských krajoch ako aj celej cestnej sieti Slovenska s vybranými hodnotami maximálneho počtu umiestnených stredísk p na základe predpisu (4.5.01), kde hodnota p_1 predstavuje prvú hodnotu p , q udáva rozmedzie medzi jednotlivými hodnotami p , k udáva krok, v ktorom počítam dané p a *počet* udáva počet vykonaných experimentov na testovanom benchmarku.

$$p = p_1 + q(k - 1) \quad k = 1, \dots, \text{počet} \quad (4.5.01)$$

Hodnoty parametrov p_1 , q a *počet*, ktoré sú uvedené v Tab. 27, boli zvolené tak, aby som obsiahol rozsah celého testovaného benchmarku a počet experimentov na danom benchmarku bol prispôsobený veľkosti riešenej úlohy pre možné vyvodenie záveru. V Tab. 27 sú podrobne popísané testované kraje spolu s ich označením, veľkosťou a nastavenými parametrami.

Tab. 27 Popis vykonaných experimentov

Testovaný kraj	Benchmarky	I	J	Testované p		
				p_1	q	počet
Bratislavský	BA87x87	87	87	5	5	15
Banskobystrický	BB515x515	515	515	15	20	25
Košický	KE460x460	460	460	15	20	23
Prešovský	PO664x664	664	664	15	20	33
Nitriansky	NR350x350	350	350	15	15	23
Trenčiansky	TN276x276	276	276	15	15	18
Trnavský	TT249x249	246	246	15	15	16
Žilinský	ZA315x315	315	315	15	15	20
Slovensko	CSR2916x2916	2916	2916	50	50	58
Beasley	A,B,C	100	1000	100	0	3
Slovensko (1000)	1000_101 – 1000_107	1000	2916	1000	0	7

¹ stĺpec *Testovaný kraj* udáva kraj Slovenska, ktorého cestná sieť bola použitá

² stĺpec *Benchmarky* udáva označenie dát pre testovanú úlohu

³ stĺpec *|I|* udáva mohutnosť množiny kandidátov na umiestnenie strediska

⁴ stĺpec *|J|* udáva mohutnosť množiny zákazníkov

⁵ stĺpec p_1 udáva prvé testované p

⁶ stĺpec q udáva rozmedzie medzi jednotlivými testovanými hodnotami p

⁷ stĺpec *počet* udáva počet testovaných hodnôt p pre najlepší výpočtový čas v testovanom benchmarku pre danú úlohu

Vylepšený algoritmus pMBBDual s vylepšovacou heuristikou V1 je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.2.2, kde metódou hrubého odhadu Lagrangeovho multiplikátora s fixnou hodnotou exponenta $\alpha = 1,1$ a parametrom delenia $r = 8$ hľadám

najlepšie nastavenie Lagrangeovho multiplikátora. Vylepšenie získaného riešenia bolo dosiahnuté vkladacou heuristikou so stratégiou najlepší vhodný (viď kapitola 4.1.1).

Vylepšený algoritmus pMedBBDual je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.4, pozostávajúci hlavne z ICDA procedúry na získanie dolnej hranice (viď kapitola 4.4.1), spôsobu spracovania vrcholu stromu vetvenia s počiatočným nastavením duálnych premenných (viď kapitola 4.4.2), algoritmu pre získanie minimálnej množiny umiestnených stredísk *FI* (viď 4.4.4) a algoritmu pre získanie kandidáta na vetvenie *V5* (viď kapitola 4.4.3).

Experimenty realizované na benchmarkoch slovenskej cestnej siete pre jednotlivé kraje a celé Slovensko [48] a Beasleyho benchmarkoch [6], [43] sú uvedené v prílohe 9. Súbor, ktoré prislúchajú testovaným úlohám sú uvedené v prílohe 1.

Vyhodnotenie experimentov je uvedené v Tab. 28 v podobe *PoR* počtu získaných optimálnych riešení k celkovému počtu vykonaných experimentov pre testovaný benchmark v percentách (4.5.02), percentuálnom vyjadrení relatívnej odchýlky *Gap* medzi získanou hodnotou riešenia *HH* a hodnotou optimálneho riešenia *ES* v priemere (4.5.03), priemerného výpočtového času *pt* z časov *t* jednotlivých experimentov (4.5.04) a počtu predčasných ukončení algoritmu po uplynutí jednej hodiny výpočtového času *ppU*. Parameter *pocet* udáva počet vykonaných experimentov a *p_{pocet}* udáva posledné testované *p* v benchmarku získané vzťahom (4.5.01). Parameter *NoR* udáva počet optimálnych riešení pre testovaný benchmark.

$$PoR = \frac{NoR}{pocet} * 100 \quad (4.5.02)$$

$$Gap = \left(\sum_{p=p_1}^{p_{pocet}} \frac{HH_p - ES_p}{ES_p} * 100 \right) / pocet \quad (4.5.03)$$

$$pt = \left(\sum_{p=p_1}^{p_{pocet}} t_p \right) / pocet \quad (4.5.04)$$

Tab. 28 Porovnanie vylepšených algoritmov založených na Erlenkotterovom prístupe

úloha	Testovaný kraj	Vylepšený pMBBDual+VI				Vylepšený pMedBBDual			
		PoR [%]	Gap[%]	pt[s]	ppU	PoR [%]	Gap [%]	pt[s]	ppU
p-medián	Bratislavský	93	0,03	0,05	0	100	0,00	0,15	0
	Banskobystrický	72	0,04	11,49	0	84	0,01	898,30	5
	Košický	78	0,09	42,97	0	87	0,05	1804,65	11
	Prešovský	76	0,07	383,15	0	85	0,03	1241,65	10
	Nitriansky	83	0,04	6,60	0	96	0,00	577,17	3
	Trenčiansky	83	0,04	0,22	0	100	0,00	4,84	0
	Trnavský	81	0,07	2,03	0	100	0,00	361,59	1
	Žilinský	75	0,10	0,49	0	100	0,00	125,43	0
	Slovensko	-	-	3600,00	58	-	-	3487,59	56
vážený p-medián	Bratislavský	100	0,00	0,02	0	100	0,00	0,39	0
	Banskobystrický	100	0,00	0,23	0	100	0,00	207,92	0
	Košický	100	0,00	0,21	0	100	0,00	147,70	0
	Prešovský	97	0,00	1,17	0	97	0,00	762,86	3
	Nitriansky	100	0,00	0,16	0	100	0,00	134,37	0
	Trenčiansky	100	0,00	0,08	0	100	0,00	9,15	0
	Trnavský	100	0,00	0,09	0	100	0,00	6,98	0
	Žilinský	95	0,00	0,07	0	100	0,00	13,43	0
	Slovensko	93	0,00	189,98	0	0	17,90	3600,00	58
UFLP	Beasley	100	0,00	0,74	0	100	0,00	4,51	0
	Slovensko1000	100	0,00	10,45	0	86	0,00	1450,46	2

¹ stĺpec *úloha* udáva typ riešenej úlohy

² stĺpec *Testovaný kraj* udáva kraj Slovenska, ktorého cestná sieť bola použitá

³ stĺpec *PoR[%]* udáva počet získaných optimálnych riešení k celkovému počtu vykonaných experimentov v percentách

⁴ stĺpec *Gap[%]* udáva percentuálne vyjadrenie relatívnej odchýlky medzi hodnotou *HH* a hodnotou *ES* v priemere

⁵ stĺpec *pt[s]* udáva priemerný čas výpočtu v sekundách

⁶ stĺpec *ppU* udáva počet predčasných ukončení algoritmu

⁷ stĺpce *Vylepšený pMBBDual+VI* udávajú získanie riešenia vylepšeným algoritmom pMBBDual s vylepšovacou heuristikou

⁸ stĺpce *Vylepšený pMedBBDual* udávajú získanie riešenia vylepšeným algoritmom pMedBBDual

⁹ zelená farba – najlepšia hodnota parametra *PoR*

¹⁰ oranžová farba – získanie najlepšieho výpočtového času

¹¹ Symbol – nepoznáme hodnotu štatistiky nakoľko nepoznáme optimálne riešenia pre danú úlohu

Štatistiky z vykonaných experimentov uvedené v Tab. 28 ukazujú, že algoritmus pMBBDual s vylepšovacou heuristikou je v porovnaní s algoritmom pMedBBDual mnohonásobne lepší v priemernom výpočtovom čase *pt*. Avšak algoritmus pMedBBDual poskytuje získanie optimálneho riešenia percentuálne vo väčšom počte testovaných prípadov *PoR*. Čo sa odzrkadľuje aj v získanej lepšej hodnote parametra *Gap*. Výnimkou sú riešené kapacitne neobmedzené umiestňovacie úlohy (UFLP), kedy optimálne riešenie bolo percentuálne získané vo väčšom počte testovaných prípadov *PoR* iteratívnym algoritmom pMBBDual. Napriek tomu, že semi-exaktný algoritmus pMBBDual je iteratívny algoritmus dokázal som ním spočítať úlohu váženého p-mediánu na celej Slovenskej sieti do jednej hodiny s 93% získaných optimálnych riešení, čo sa v prípade algoritmu pMedBBDual nepodarilo. Nevýhoda iteratívnosti algoritmu pMBBDual sa prejavila pri riešení úlohy p-mediánu na celej Slovenskej cestnej sieti, kde získanie optimálneho riešenia v 2. iterácii pri nastavenom Lagrangeovom multiplikátore $Lg = 2048$ bolo veľmi časovo náročné.

Algoritmus pMedBBDual poskytol riešenie na úlohe p -mediánu na celom Slovensku za jednu hodinu výpočtového času, avšak či získané riešenie je optimálne neviem posúdiť vo väčšine prípadov, nakoľko bol algoritmus predčasne ukončený. Treba však zdôrazniť, že algoritmus pMedBBDual bez časového obmedzenia výpočtu vždy poskytne optimálne riešenie.

4.6 Kompozičný prístup k riešeniu úlohy návrhu VOS

Iteratívny semi-exaktný algoritmus pMBBDual s vylepšovacou heuristikou dokáže získať dobré prípustné riešenie za krátky výpočtový čas. Pod dobrým prípustným riešením rozumiem riešenie blízko optimálneho riešenia. Zovšeobecnený exaktný algoritmus pMedBBDual dokáže získať optimálne riešenie, ale za mnohonásobne vyšší výpočtový čas. Získanie optimálneho riešenia riešenej úlohy za adekvátny výpočtový čas (jedna hodina výpočtu) vidím v kombinácii vyššie uvedených prístupov tzv. kompozičný prístup. Získanie optimálneho riešenia úlohy predstavuje rovnosť dolnej hranice riešenia DH a hornej hranice riešenia HH .

Kompozičný prístup predstavuje vylepšenie riešenia získaného iteratívnym algoritmom pMBBDual s vylepšovacou heuristikou aplikovaním zovšeobecneného algoritmu pMedBBDual vychádzajúceho z hodnoty neprípustného riešenia s najmenšou mierou neprípustnosti získaného algoritmom pMBBDual v jednej z jeho iterácií. Riešenie získané algoritmom pMBBDual považujem za neprípustné riešenie s najmenšou mierou neprípustnosti, ak počet umiestnených stredísk je väčší ako p a je najbližšie k tejto hodnote p , pričom hodnota účelovej funkcie tohto riešenia je najmenšia, aká mohla byť algoritmom pMBBDual získaná. V Tab. 29 som porovnal navrhnuté algoritmy s kompozičným prístupom na základe GoR počtu získaných optimálnych riešení k celkovému počtu vykonaných experimentov pre testovaný benchmark v percentách (4.6.01), percentuálnom vyjadrení relatívnej odchýlky Gap medzi získanou hodnotou riešenia HH a dolnou hranicou riešenia DH v priemere (4.6.02) a priemerného výpočtového času pt z časov t jednotlivých experimentov (4.6.03). Parameter $pocet$ udáva počet vykonaných experimentov a p_{pocet} udáva posledné testované p v benchmarku získané vzťahom (4.5.01). Parameter NoR udáva počet optimálnych riešení (dolná hranica riešenia je rovná hornej hranice riešenia) pre testovaný benchmark.

$$GoR = \frac{NoR}{pocet} * 100 \quad (4.6.01)$$

$$Gap = \left(\sum_{p=p_1}^{p_{pocet}} \frac{HH_p - DH_p}{DH_p} * 100 \right) / pocet \quad (4.6.02)$$

$$pt = \left(\sum_{p=p_1}^{p_{pocet}} t_p \right) / pocet \quad (4.6.03)$$

Podrobné experimenty s kompozičným prístupom, na základe ktorých som jednotlivé kraje Slovenska vyhodnocoval sú uvedené v prílohe 10.

Tab. 29 Porovnanie vylepšených algoritmov s kompozičným prístupom v kvalite riešenia a čase

úloha	Testovaný kraj	Vylepšený pMBBDual+V1			Vylepšený pMedBBDual			Kompozičný algoritmus		
		GoR[%]	Gap[%]	pt [s]	GoR[%]	Gap[%]	pt[s]	GoR[%]	Gap[%]	pt[s]
p-medián	Bratislavský	73	0,26	0,05	100	0,00	0,15	100	0,00	0,16
	Banskobystrický	68	0,49	11,49	80	0,12	898,30	84	0,01	567,18
	Košický	70	0,37	42,97	52	0,18	1804,65	83	0,07	803,13
	Prešovský	70	0,58	383,15	70	0,15	1241,65	94	0,00	659,77
	Nitriansky	74	0,66	6,60	87	0,07	577,17	100	0,00	10,65
	Trenčiansky	72	0,26	0,22	100	0,00	4,84	100	0,00	1,10
	Trnavský	69	0,47	2,03	94	0,06	361,59	100	0,00	23,54
	Žilinský	60	0,95	0,49	100	0,00	125,43	100	0,00	4,10
	Slovensko	-	-	3600,00	3	10,46	3487,59	-	-	-
vážený p-medián	Bratislavský	100	0,00	0,02	100	0,00	0,39	100	0,00	0,02
	Banskobystrický	100	0,00	0,23	100	0,00	207,92	100	0,00	0,23
	Košický	100	0,00	0,21	100	0,00	147,70	100	0,00	0,21
	Prešovský	97	0,00	1,17	97	0,00	762,86	100	0,00	3,73
	Nitriansky	100	0,00	0,16	100	0,00	134,37	100	0,00	0,16
	Trenčiansky	100	0,00	0,08	100	0,00	9,15	100	0,00	0,08
	Trnavský	100	0,00	0,09	100	0,00	6,98	100	0,00	0,09
	Žilinský	95	0,00	0,07	100	0,00	13,43	100	0,00	0,23
	Slovensko	93	0,00	189,98	0	17,95	3600,00	93	0,00	423,61
UFLP	Beasley	100	0,00	0,74	100	0,00	4,51	100	0,00	0,74
	Slovensko1000	100	0,00	10,45	71	0,00	1450,46	100	0,00	10,45

¹ stĺpec *úloha* udáva typ riešenej úlohy

² stĺpec *Testovaný kraj* udáva kraj Slovenska, ktorého cestná sieť bola použitá

³ stĺpec *GoR[%]* udáva počet získaných optimálnych riešení k celkovému počtu vykonaných experimentov v percentách ($DH=HH$)

⁴ stĺpec *Gap[%]* udáva percentuálne vyjadrenie relatívnej odchýlky medzi hodnotou HH a hodnotou DH v priemere

⁵ stĺpec *pt[s]* udáva priemerný čas výpočtu v sekundách

⁶ stĺpec *Vylepšený pMBBDual+V1* udávajú získanie riešenia vylepšeným algoritmom pMBBDual s vylepšovacou heuristikou

⁷ stĺpec *Vylepšený pMedBBDual* udávajú získanie riešenia vylepšeným algoritmom pMedBBDual

⁸ stĺpec *Kompozičný algoritmus* udávajú získanie riešenia kompozičným prístupom popísaný vyššie.

⁹ zelená farba – najlepšie hodnoty parametra *GoR*

¹⁰ oranžová farba – získanie najlepšieho výpočtového času medzi kompozičným prístupom a algoritmom pMedBBDual

¹¹ Symbol – nepoznáme hodnotu štatistiky, nakoľko sme nedokázali dospieť k adekvátnemu riešeniu

Na základe vykonaných štatistík (vid' Tab. 29) môžem vyvodit' záver, že využitím kompozičného prístupu pre riešenie úlohy návrhu verejného obslužného systému dokážem získať optimálne riešenie úlohy v priemere za kratší výpočtový čas ako by sme to riešili zovšeobecneným algoritmom pMedBBDual. Kompozičným prístupom sa podarilo získať optimálne riešenie vo väčšom počte riešených prípadov ako s navrhnutými algoritmi pMBBDual s vylepšovacou heuristikou alebo algoritmom pMedBBDual pri obmedzení výpočtového času na jednu hodinu. Riešenie testovaných úloh váženého p-mediánu na celom Slovensku sa nám nepodarilo vylepšiť algoritmom pMedBBDual v kompozičnom prístupe ani po jednej hodine výpočtu. Kompozičný prístup nie je možné využiť pri úlohe p-mediánu na celom Slovensku, keďže tento prístup sa zakladá na získaní dobrého prípustného riešenia algoritmom pMBBDual s vylepšovacou heuristikou za krátky výpočtový čas, čo som v tomto prípade nezískal.

4.7 Zhrnutie

V predchádzajúcich podkapitolách som prezentoval výsledky svojho výskumu v oblasti navrhovania verejných obslužných systémov s exaktným optimalizačným jadrom založených na Erlenkotterovom princípe. Navrhol som semi-exaktný iteratívny algoritmus na riešenie úlohy návrhu verejného obslužného systému založený na Lagrangeovej relaxácii s najvhodnejším nastavením Lagrangeovho multiplikátora metódou bisekcie alebo hrubým odhadom, ktorého riešenie som vylepšil vkladacou heuristikou so stratégiou najlepšie vhodný. Algoritmus s vylepšovacou heuristikou poskytuje dobré prípustné riešenie, no získanie optimálneho riešenia nie je zaručené. Rozpracoval som teóriu duality úlohy návrhu verejného obslužného systému v snahe riešiť úlohu návrhu VOS priamo v metóde vetiev a hraníc. Navrhol som exaktný algoritmus založený na zovšeobecnenom Erlenkotterovom prístupe k získaniu dolnej a hornej hranici riešenia, výberu kandidáta na vetvenie. Exaktný algoritmus som vylepšil výberom vhodnej procedúry pre získanie dolnej hranice, vhodnej funkcie pre získanie kandidáta na vetvenie a vhodnej funkcie pre získanie hornej hranice a prípustného riešenia. Navrhnutý exaktný algoritmus poskytuje získanie optimálneho riešenia, no za cenu dlhšieho výpočtového času, niekedy však až extrémne dlhého vzhľadom k veľkosti riešenej úlohy. V snahe nájsť kompromis medzi výpočtovým časom a získaním optimálneho riešenia som navrhol kompozičný prístup k riešeniu úlohy návrhu verejného systému pozostávajúci z riešenia úlohy návrhu VOS iteratívnym algoritmom s hrubým odhadom Lagrangeovho multiplikátora s vylepšovacou heuristikou, ktorý pri získaní prípustného riešenia, ale nie optimálneho, využije

získanie optimálne riešenie exaktným algoritmom so zovšeobecnením Erlenkotterovým prístupom. Výsledky numerických experimentov potvrdzujú, že kompozičný algoritmus dokáže získať optimálne riešenie úlohy návrhu VOS vo väčšine prípadov za pomerne krátky výpočtový čas vzhľadom k rozsahu testovanej úlohy, čo považujem za prínos tejto práce. Treba však zdôrazniť, že tento kompozitný algoritmus má svoje limity, keď sa nám nepodarilo spočítať úlohu p-mediánu na celom Slovensku. Optimálne riešenie som získal semi-exaktným iteratívnym algoritmom v 86% testovaných úloh, zovšeobecneným algoritmom v 77% testovaných úloh a kompozičným algoritmom v 97% testovaných úloh pri obmedzení výpočtového času na jednu hodinu a získaniu aspoň nejakého adekvátneho riešenia (tzn. bez úloh p-mediánu na Slovensku). Treba však zdôrazniť, že som bral do úvahy optimálne riešenia len s nulovým rozdielom (*gap*) medzi dolnou a hornou hranicou riešenia.

5 Záver

V práci som sa zaoberal aplikovaným výskumom v oblasti navrhovania verejných obslužných systémov s exaktným optimalizačným jadrom s využitím prostriedkov aplikovanej informatiky za účelom získania efektívneho softvérového nástroja na podporu rozhodovania o štruktúre verejného obslužného systému.

V úvodnej časti práce som sa zaoberal súčasným stavom riešenej problematiky v oblasti súvisiacej s cieľom práce. V tejto časti som prezentoval úlohu návrhu verejného obslužného systému a jeho špecifiká, kde som popísal niekoľko typových úloh spadajúcich do uvedenej oblasti. Analyzoval som najpoužívanejšie exaktné a semi-exaktné algoritmy z domácich aj zahraničných literárnych prameňov, ktoré boli zamerané na riešenie danej problematiky. Špeciálne som sa venoval Erlenkotterovmu prístupu k riešeniu kapacitne neobmedzenej umiestňovacej úlohy, ktorý je založený na metóde vetiev a hraníc a na vzťahoch z teórie duality k získaniu rýchlej a dobrej dolnej a hornej hranice riešenia.

Vo vlastnej výskumnej práci som sa zaoberal otázkou do akej miery je možné a výhodné využitie Erlenkotterovho prístupu k návrhu verejného obslužného systému spočívajúceho v riešení umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk v rámci výskumu metód navrhovania verejných obslužných systémov s exaktným optimalizačným jadrom pri využití prostriedkov aplikovanej informatiky.

Navrhol som iteratívny prístup s využitím Lagrangeovej relaxácie, ktorý umožnil transformovať úlohu návrhu verejného obslužného systému na kapacitne neobmedzenú umiestňovaciu úlohu. Pre implementáciu iteratívneho prístupu som zostrojil semi-exaktný iteratívny algoritmus založený na Erlenkotterovom prístupe a Lagrangeovej relaxácii p-mediánovej podmienky. Erlenkotterovým prístupom som opakovane riešil kapacitne neobmedzenú umiestňovaciu úlohu s meniacou sa hodnotou Lagrangeovho multiplikátora. Nastavenie Lagrangeovho multiplikátora, ktoré bolo pôvodne získavané metódou bisekcie som urýchlil metódou odhadu Lagrangeovho multiplikátora. Semi-exaktný iteratívny algoritmus, ktorý poskytuje dolnú a hornú hranicu riešenia, som rozšíril o vylepšovaciu heuristiku pre získanie lepšieho riešenia, pokiaľ algoritmus predtým nedosiahol optimálne riešenie. Dosiachnutie optimálneho riešenia bolo rozpoznateľné na základe rovnosti dolnej a hornej hranice riešenia.

Navrhol som zovšeobecnený Erlenkotterov prístup na riešenie úlohy návrhu verejného obslužného systému založený na metóde vetiev a hraníc a poznatkoch z Erlenkotterovho prístupu. Na základe podrobného rozpracovania teórie duality pre riešený problém som skonštruoval exaktný algoritmus pre riešenie umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk. V rámci konštrukcie exaktného algoritmu som navrhol procedúru pre získanie dolnej hranice, hornej hranice, procedúru pre získanie kandidáta na vetvenie v metóde vetiev a hraníc a procedúru uskutočňujúcu samotnú metódu vetiev a hraníc.

Modifikoval som duálny prístup na získanie dolnej hranice riešenia realizovaný duálnym vzostupným algoritmom (DA) pre kapacitne neobmedzenú umiestňovaciu úlohu. Navrhol som alternatívne metódy pre získanie dolnej hranice riešenia úlohy návrhu verejného obslužného systému, ktoré som porovnal a vybral najvhodnejšiu metódu.

Modifikoval som duálny prístup k získaniu hornej hranice riešenia umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk. Navrhol som viaceré metódy pre získanie hornej hranice spracovaného vrcholu v metóde vetiev a hraníc založené na dodržiavaní komplementárnych podmienok pri konštrukcii primárneho riešenia s rôznym poradím spracovania zákazníkov, porovnal ich a vybral najvhodnejšiu metódu.

Modifikoval som výber kandidáta na vetvenie v metóde vetiev a hraníc a navrhol som viaceré metódy založené na rôznom poradí vyhodnotenia komplementárnych podmienok, porovnal som ich a vybral som najvhodnejšiu metódu.

Výsledkom výskumnej práce bol návrh základných algoritmov s Erlenkotterovým prístupom k riešeniu danej problematiky a ich vylepšených verzií získaných výskumom jednotlivých procedúr algoritmu pre riešenie úlohy p-mediánu, váženého p-mediánu alebo kapacitne neobmedzenej umiestňovacej úlohy ako úlohy návrhu verejného obslužného systému. Urýchlil som postup najvhodnejšieho nastavenia Lagrangeovho multiplikátora metódou hrubého odhadu, ktorá priniesla v porovnaní s pôvodnou metódou bisekcie v semi-exaktnom iteratívnom algoritme menší počet iterácií a tým aj celkovo menší výpočtový čas algoritmu. Semi-exaktný iteratívny algoritmus poskytuje optimálne riešenie, prípadne suboptimálne riešenie, ktoré zlepší pomocou vylepšovacej heuristiky.

V rámci vylepšenia zovšeobecneného exaktného prístupu som navrhol a experimentálne verifikoval takú kombináciu procedúr, ktorá urýchlila získanie optimálneho riešenia oproti pôvodnej verzii zovšeobecneného exaktného algoritmu.

Porovnal som vylepšené verzie iteratívneho a zovšeobecneného algoritmu s ohľadom na výpočtový čas a kvalitu získaného riešenia. Iteratívny algoritmus poskytuje kvalitné prípustné riešenie za pomerne krátku dobu výpočtu v závislosti od veľkosti riešenej úlohy. Naopak exaktný algoritmus poskytuje optimálne riešenie úlohy, ale niekedy za cenu extrémne dlhého výpočtového času. Preto som skúmal, aké dobré riešenie je schopný algoritmus získať v obmedzenej dobe výpočtu, čo v prípade, že vetviaci proces bol predčasne ukončený po jednej hodine, viedlo k zníženiu kvality získaného riešenia.

Ďalej som navrhol kompozičný prístup, ktorý kombinuje výhody navrhnutých algoritmov. V prvej fáze kompozičného algoritmu je získané riešenie iteratívnym semi-exaktným algoritmom s vylepšovacou heuristikou. V druhej fáze je toto riešenie spracované exaktným algoritmom spolu so získanou dolnou a hornou hranicou riešenia.

Prínosy mojej dizertačnej práce môžem rozdeliť do dvoch oblastí. V praktickej oblasti som navrhol kolekciu nástrojov pre úlohy návrhu verejného obslužného systému. Jednotlivé nástroje sa líšia na základe preferovaného kritéria, či je to časové kritérium alebo kritérium exaktnosti. Vedie to k výberu nástroja na základe užívateľom preferovaného kritéria. Prínosom práce v teoretickej oblasti sú poznatky o exaktnosti a semi-exaktnosti metód riešenia. Exaktnosť predstavuje poskytnutie optimálneho riešenia, zatiaľ čo semi-exaktnosť predstavuje poskytnutie prípustného riešenia aj s dolnou hranicou riešenia. Študoval som počet prípadov, kedy iteratívny algoritmus získa exaktné riešenie, pričom beriem do úvahy len optimálne riešenia s nulovým rozdielom medzi dolnou a hornou hranicou riešenia. Keď nebolo získané exaktné riešenie, vyhodnocoval som maximálnu možnú odchýlku získaného riešenia od exaktného riešenia, čo je rozdiel medzi získanou hornou a dolnou hranicou riešenia. V zovšeobecnenom exaktnom algoritme som študoval počet prípadov, kedy algoritmus pri časovo obmedzenom behu získa exaktné riešenie. V prípade prekročenia určeného času algoritmu (jedna hodina) som vyhodnocoval odchýlku získaného riešenia od dolnej hranice riešenia.

Z tohto pohľadu možno uvedené softvérové nástroje na podporu rozhodovania o štruktúre verejného obslužného systému považovať za prínos v oblasti aplikovanej informatiky. Výsledky môjho výskumu poukázali na prínosy a limity využitia Erlenkotterovho prístupu k návrhu verejného obslužného systému spočívajúceho v riešení umiestňovacej úlohy s obmedzeným počtom vybudovaných stredísk. Týmto som splnil zadané ciele práce

Zoznam použitej literatúry

- [1] AVELLA, P., SASSANO, A., VASSIL'EV, I. (2007). *Computational study of large scale p-median problems*. In *Mathematical Programming* 109, 2007, pp. 89-114
- [2] BALINSKI, M. (1965). *Integer programming: methods, uses computation*. In *Management Science*, 12, 1965, pp. 254–313
- [3] BARILLA, J., SIMR, P., SÝKOROVÁ, K. (2013). *Microsoft Excel 2013: Podrobná uživatelská príručka*, Brno: Computer Press, 2013, ISBN 978-80-251-4114-4, 496 s.
- [4] BEASLEY, J. E. (1993). *Lagrangean heuristics for location problems*. In *European Journal of Operational Research* 65(3), 1993, pp. 383-399
- [5] BEASLEY, J. E. (1993). *Lagrangean relaxation*. In *Modern Heuristic Techniques for Combinatorial Problems*, Oxford Blackwell Scientific Publications, London, 1993, ISBN 0-632-03238-3, pp. 243-303
- [6] BEASLEY J. E. (1990), *OR Library: Distributing Test Problems by Electronic Mail*. *Journal of the Operational Research Society*, 41(11), pp. 1069-1072
- [7] BENDÍK J. (2013). *Exaktný algoritmus na riešenie umiestňovacích úloh v návrhu verejného obslužného systému*: diplomová práca, Žilina: Žilinská univerzita v Žiline, 88 s.
- [8] BENDÍK, J. (2013). *Exact algorithm for Location Problem Solving in Public Service System Design*. In: *Úlohy diskretní optimalizace v dopravní praxi 2013 : SW nástroje na řešení úloh racionalizace. Objektivizace a optimalizace pokrytí území veřejnými obslužnými systémy*: Pardubice, Česká republika 28.-29.10.2013, ISBN 978-80-7395-744-5, pp. 7-15.
- [9] BENDÍK, J. (2014). *Generalization of the Erlenkotter approach for solving of the public service system design*. In. *Proceedings of ICTTE 2014: International Conference on Traffic and Transport Engineering: Belgrade, Serbia, November 27-28, 2014*, ISBN 978-86-916153-1-4, pp. 12-17.
- [10] BENDÍK, J. (2014) *Heuristics for improving the solution of p-median location problem with Erlenkotter approach*. In *Proceedings of the 10th international conference Digital technologies: Žilina, Slovakia, 9-11 July, 2014*, ISBN 978-1-4799-3301-3, pp. 7-11.

- [11] BENDÍK, J. (2015). *Improving the exact algorithm for solving the public service system design in the branching*. In SOR '15 : Proceedings of the 13th International Symposium on Operational Research: Bled, Slovenia, September 23-25, 2015, ISBN 978-961-6165-45-7, pp. 215-220.
- [12] BENDÍK, J. (2015). *Increment approach for obtaining the lower bound in public service system design*. In Proceedings of the 4th international symposium and 26th national conference on operational research: Chania-Greece, June 4-6, 2015, ISBN 978-618-80361-4-7, pp. 230-234.
- [13] BENDÍK, J. (2015). *Selection of minimal set of locations in the public service system design*. In Proceedings of Informatics 2015 : IEEE 13th international scientific conference on informatics: Poprad, Slovakia, November 18-20, 2015, ISBN 978-1-4673-9867-1, pp. 47-51.
- [14] BENDÍK, J. (2014). *Solving the p -median location problem with the Erlenkotter approach in public service system design*. In 4th Student Conference on Operational Research: Nottingham, UK, May 2-4, 2014, Saarbrücken/Wadern: Dagstuhl Publishing, ISBN 978-3-939897-67-5, pp. 25-33.
- [15] BUZNA, Ľ. (2010). *Informatické nástroje pre návrh obslužných systémov na priestorovo rozľahlých sieťach*: habilitačná práca, Žilina: Žilinská univerzita v Žiline, Fakulta riadenia a informatiky, 2010, 91 s.
- [16] CORNUÉJOLS, G., NEMHAUSER, G.L., WOLSEY, L.A. (1980). *A canonical representation of simple plant location problems and its applicaitons*. In SIAM Journal on Algebraic and Discrete Methods 1 (3), 1980, pp. 261-272
- [17] CORNUÉJOLS, G., NEMHAUSER, G.L., WOLSEY, L.A. (1990). *The uncapacitated facility location problem*. In P. B. Mirchandani, P., B., Francis, R., L., (Eds.), Discrete location theory, New York: Wiley, 1990, pp. 119–171
- [18] CUDRÁK, P. (2011). *Informatický nástroj na hľadanie stredísk verejného obslužného systému v rozľahlej sieti*: diplomová práca, Žilina: Žilinská univerzita v Žiline, 83 s.
- [19] DASKIN, M. S. (2013). *Network and Discrete Location. Models, Algorithms, and Applications*. John Wiley & Sons, New York, 2013
- [20] DREZNER, Z., HAMACHER, H.W. (2002) *Facility location: Applications and theory*, Berlin: Springer, 2002.

- [21] DREZNER, Z. (1984) *The p-Centre Problem - Heuristic and Optimal Algorithms*. The Journal of the Operational Research Society, Vol. 35, No. 8, 1984, pp. 741 -748.
- [22] ERLINKOTTER, D. (1978). *A Dual-Based Procedure for Uncapacitated Facility Location*. Operations Research, Vol. 26, No 6, 1978, pp. 992-1009
- [23] GALVAO, R. D. (1980). *A dual-bounded algorithm for the p-median problem*. In Operations Research 28, 1980, pp. 1112-1121
- [24] GARCIA, S., LABBÉ, M., MARÍN, A. (2011). *Solving large p-median problems with a radius formulation*. In INFORMS Journal on Computing 23 (4), 2011, pp. 546-556
- [25] GAREY, M. R., JOHNSON, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979
- [26] HAKIMI, S. L., KARIV, O. (1979). *An algorithmic approach to network location problems II: The p-medians*. In SIAM J. Appl. Math. Oper. Res. 37(3), pp. 539–560
- [27] JANÁČEK, J. a kol. (2010). *Navrhovanie územne rozľahlých obslužných systémov*, Žilina: EDIS – vydavateľstvo ŽU, 2010, ISBN 978-80-554-0219-2, 404 s.
- [28] JANÁČEK, J. (2000). *Dopravno-optimálny rozklad regiónu*, In Communications - scientific letters of the University of Žilina 2(4), pp. 35-42.
- [29] JANÁČEK, J. (2014). *Mikrooptimalizace výpočetních postupů v metaheuristikách pro umísťovací úlohy*. In Úlohy diskretní optimalizace v dopravní praxi 2014 : SW podpora rozhodování v inteligentních dopravních systémech, Pardubice, 2014, ISBN 978-80-7395-867-1. pp. 4-9.
- [30] JANÁČEK, J., KOVAČIKOVÁ, J. (1997). *Exact Solution Techniques for Large Location Problems*. In Proceedings of the Mathematical Methods in Economics 1997, Ostrava, 9.-11.9.1997, pp. 80-84
- [31] JANÁČEK, J. (2003). *Matematické programování*. Druhé opravené vydanie, Žilina: EDIS – vydavateľstvo ŽU, 2003, ISBN 80-8070-054-0, 225 s.
- [32] JANÁČEK, J. (2006). *Optimalizace na dopravních sítích*. Druhé prepracované vydanie. Žilina: EDIS – vydavateľstvo ŽU, 2006, ISBN 80-8070-586-0, 248 s.
- [33] JANÁČEK, J., BUZNA, E. (2008). *An acceleration of Erlenkotter-Körkel's algorithms for uncapacitated facility location problem*. In Annals of Operations Research, 2008, Vol. 164, No. 1, ISSN 0254-5330, pp. 97-109.

- [34] JANÁČEK, J.; KOHÁNI, M.; BUZNA, Ľ. (2014) *An Approximation Algorithm for the Facility Location Problem with Lexicographic Minimax Objective* In Journal of Applied Mathematics, 2014, ISSN 1110-757X, pp. 1-12.
- [35] JANÁČEK, J., KVET, M. (2012). *Relevant network distances for approximate approach to the p-median problem*. In Operations Research Proceedings 2012: Selected Papers of the International Conference on Operations Research (OR 2012), September 4-7 2012, Leibniz Univesität Hannover, Germany, Berlin Heidelberg: Springer-Verlag, 2012, s. 99
- [36] JANÁČEK, J. (2008). *Approximate Covering Models of Location Problems*. In Lecture Notes in Management Science: Proceedings of the 1st International Conference on Applied Operational Research - ICAOR '08, Vol. 1, September 2008, Yerevan, Armenia, ISSN 2008-0050, pp. 53-61.
- [37] KÖRKEL, M. (1989). *On the exact solution of large – scale simple plant location problem*. In European Journal of Operational Research 39, North Holland, pp.157-173.
- [38] KVET, M. (2013) *Navrhovanie verejných obslužných systémov metódami pokrývania: dizertačná práca*, Žilina: Žilinská Univerzita, 2013, 143 s.
- [39] LAPORTE, G.; NICKEL, S. (2015) *Location Science*, New York: Springer-Verlag, 2015, pp. 642
- [40] MARKECHOVÁ, D.; TIRPÁKOVÁ, A.; STEHLÍKOVÁ, B. (2011). *Základy štatistiky pre pedagógov*, Nitra: FPV UKF, 2011, p. 405.
- [41] MLADENOVÍČ, N. et al. (2007). *The p-median problem: a survey of metaheuristic approaches*. In European Journal of Operational Research, 179(3), 2007, pp. 927–939.
- [42] OGRYCZAK, W.; ŚLIWIŃSKI, T. (2006). *On Direct Methods for Lexicographic Min-Max Optimization*, ICCSA, vol. 3982, 2006, pp. 802- 811.
- [43] OR-Lib benchmarks, [posledný prístup 2016-04-13], dostupné na internete: <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/pmedinfo.html>
- [44] PLESNÍK, J. (1983) *Grafové algoritmy*, Bratislava: VEDA, 1983.
- [45] REESE, J. (2006). *Solution methods for the p-median problem: An annotated bibliography*. In *Networks* 48(3), pp. 125–142, DOI 10.1002/net.20128, Published online in Wiley InterScience (www.interscience.wiley.com)

- [46] RESENDE, M.G.C. and WERNECK, R. F.. (2003). *On the Implementation of a Swap-Based Local Search Procedure for the p-Median Problem*. In R.E. Ladner (ed.), Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments (ALENEX'03), SIAM, pp. 119–127.
- [47] RESENDE, M. G. C., WERNECK, R. F. (2004). *A Hybrid Heuristic for the p-Median Problem*. In Journal of Heuristics, volume 10, number 1, 2004, pp. 59-88
- [48] SZENDREYOVÁ A. *Benchmarks - vzorové riešené úlohy*, [posledný prístup 2016-04-13], dostupné na internete: <http://frdsa.fri.uniza.sk/~betka/indexPovodny.html>.
- [49] TOMAN, Ľ. (2013). *Sofistikované nástroje na podporu rozhodovania v podmienkach neistoty pri návrhu verejných obslužných systémov evakuačného typu*: dizertačná práca, Žilina: Žilinská univerzita v Žiline, Fakulta riadenia a informatiky, 2013
- [50] XPRESS Optimization Suite - Student Edition, [posledný prístup 2016-04-13], dostupné na internete: <https://community.fico.com/download.jspa>
- [51] XPRESS-Mosel “User guide”, Dash Associates, Blisworth, 2005, UK, 99 s.

Zoznam obrázkov a grafov

Obr. 1 Distribučný systém s relevantnými nákladmi	19
Obr. 2 Príklad najlepšieho prípustného a neprípustného riešenia	60
Obr. 3 Prerozdelenie stredísk na množiny I_S a I_D	61
Obr. 4 Fáza doplnenia a rozdelenie množiny I_D na podmnožiny I_{DI} a I_{DN}	61
Obr. 5 Fáza výmeny prvkov pri zlepšení riešenia.....	61
Obr. 6 Demonštračný príklad s obmedzením vetvenia	62
Obr. 7 Demonštračný príklad závislosti počtu vybudovaných stredísk na fixných nákladoch	68

Zoznam tabuliek

Tab. 1 Porovnanie získaných riešení nástrojmi XPRESS-IVE a algoritmom pMBBDual	38
Tab. 2 Výsledky experimentov algoritmu ZEBRA.....	42
Tab. 3 Porovnanie presnosti pokrývacej metódy vzhľadom k optimálnemu riešeniu.....	48
Tab. 4 Vylepšenia algoritmu pMBBDual a ich porovnanie s XPRESS-IVE na úlohe BB100x100	64
Tab. 5 Vylepšenia algoritmu pMBBDual a ich porovnanie s XPRESS-IVE na úlohe ZA315x315	65
Tab. 6 Porovnanie zlepšenia riešenia algoritmu pMBBDual na úlohe SR1000x2916.....	66
Tab. 7 Výber vhodného nastavenia fixného parametra α pre rôzne typy úloh.....	73
Tab. 8 Výber vhodného nastavenia parametra r s fixným α pre rôzne typy úloh	74
Tab. 9 Vyber vhodného nastavenia adaptívneho parametra α pre rôzne typy úloh	75
Tab. 10 Vyber vhodného nastavenia parametra r s adaptívnym α pre rôzne typy úloh	75
Tab. 11 Porovnanie iteratívnych algoritmov na benchmarkoch ZA315x315 a TN276x276	76
Tab. 12 Popis vykonaných experimentov na slovenskej cestnej sieti	77
Tab. 13 Porovnanie iteratívnych prístupov na slovenskej cestnej sieti	79
Tab. 14 Porovnanie algoritmu pMedBBDual s XPRESS-IVE na benchmarku BB100x100.....	95
Tab. 15 Porovnanie kvality dolnej hranice na benchmarku BB100x100	96
Tab. 16 Porovnanie procedúr pre získanie dolnej hranice na benchmarku BB100x100.....	98
Tab. 17 Porovnanie efektívnosti získania dolnej hranice na benchmarku BB100x100	100
Tab. 18 Poradie vyhodnotenia dodržania komplementárnych podmienok pre jednotlivé varianty	101
Tab. 19 Porovnanie variantov funkcie vo výpočtovom čase na benchmarku ZA315x315	102
Tab. 20 Vyhodnotenie štatistík na úlohe BB100x100, ZA315x315 a NR350x350	103
Tab. 21 Porovnanie vhodnosti jednotlivých variantov funkcie na základe dvojvýberového t-testu	104
Tab. 22 Porovnanie variantov funkcie vo výpočtovom čase na benchmarku ZA315x315	106
Tab. 23 Vyhodnotenie štatistík pre úlohy BB100x100, ZA315x315 a TN276x276.....	108
Tab. 24 Porovnanie vhodnosti jednotlivých variantov funkcie na základe dvojvýberového t-testu	108
Tab. 25 Popis vykonaných experimentov na slovenskej cestnej sieti	109
Tab. 26 Porovnanie algoritmov založených na zovšeobecnenom Erlenkotterovom prístupe	111
Tab. 27 Popis vykonaných experimentov	112
Tab. 28 Porovnanie vylepšených algoritmov založených na Erlenkotterovom prístupe	114
Tab. 29 Porovnanie vylepšených algoritmov s kompozičným prístupom v kvalite riešenia a čase.....	116
Tab. 30 Súbory použité pre úlohu p-mediánu	131
Tab. 31 Súbory použité pre úlohu váženého p-mediánu.....	132
Tab. 32 Súbory použité pre kapacitne neobmedzenú umiestňovaciu úlohu	132
Tab. 33 Porovnanie metód nastavenia Lagrangeovho multiplikátora na benchmarku ZA315x315	133
Tab. 34 Porovnanie iteratívnych algoritmov na Bratislavskom kraji.....	134
Tab. 35 Porovnanie iteratívnych algoritmov na Banskobystrickom kraji.....	135
Tab. 36 Porovnanie iteratívnych algoritmov na Trenčianskom kraji	135

Tab. 37 Porovnanie iteratívnych algoritmov na Košickom kraji	136
Tab. 38 Porovnanie iteratívnych algoritmov na Nitrianskom kraji.....	136
Tab. 39 Porovnanie iteratívnych algoritmov na Prešovskom kraji	137
Tab. 40 Porovnanie iteratívnych algoritmov na Trnavskom kraji.....	138
Tab. 41 Porovnanie iteratívnych algoritmov na Žilinskom kraji	138
Tab. 42 Porovnanie iteratívnych algoritmov na slovenskej cestnej sieti.....	139
Tab. 43 Porovnanie variantov funkcie vo výpočtovom čase na benchmarku BB100x100	140
Tab. 44 Dvojvýberový t – test pre výber vhodného variantu funkcie pre získanie kandidáta na vetvenie	141
Tab. 45 Porovnanie variantov funkcie vo výpočtovom čase na benchmarku BB100x100	142
Tab. 46 Dvojvýberový t – test pre získanie minimálnej množiny umiestnení	143
Tab. 47 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Bratislavskom kraji	145
Tab. 48 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Banskobystrickom kraji ...	145
Tab. 49 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Košickom kraji	146
Tab. 50 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Nitrianskom kraji	146
Tab. 51 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Prešovskom kraji	147
Tab. 52 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Trenčianskom kraji	148
Tab. 53 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Trnavskom kraji.....	148
Tab. 54 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Žilinskom kraji	149
Tab. 55 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Slovensku.....	150
Tab. 56 Porovnanie algoritmov na Bratislavskom kraji.....	152
Tab. 57 Porovnanie algoritmov na Banskobystrickom kraji.....	152
Tab. 58 Porovnanie algoritmov na Košickom kraji	153
Tab. 59 Porovnanie algoritmov na Nitrianskom kraji.....	153
Tab. 60 Porovnanie algoritmov na Prešovskom kraji	154
Tab. 61 Porovnanie algoritmov na Trenčianskom kraji.....	155
Tab. 62 Porovnanie algoritmov na Trnavskom kraji	155
Tab. 63 Porovnanie algoritmov na Žilinskom kraji	156
Tab. 64 Porovnanie algoritmov pre kapacitne neobmedzenú umiestňovaciu úlohu	156
Tab. 65 Porovnanie algoritmov na Slovenskej cestnej sieti pre úlohu váženého p-mediánu.....	157
Tab. 66 Porovnanie algoritmov s kompozičným algoritmom na úlohe p-mediánu	159
Tab. 67 Porovnanie algoritmov s kompozičným algoritmom na úlohe váženého p-mediánu	160

Zoznam príloh

Príloha 1: Popis úloh a vstupných súborov

Príloha 2: Porovnanie metód nastavenia Lagrangeovho multiplikátora

Príloha 3: Porovnanie iteratívnych algoritmov

Príloha 4: Experimenty pre výber kandidáta na vetvenie

Príloha 5: Štatistiky pre výber kandidáta na vetvenie

Príloha 6: Experimenty pre výber varianty minimálnej množiny umiestnení

Príloha 7: Štatistiky pre získanie hornej hranice

Príloha 8: Porovnanie algoritmov zovšeobecneného Erlenkotterovho prístupu

Príloha 9: Porovnanie iteratívneho a zovšeobecneného algoritmu

Príloha 10: Porovnanie navrhnutých algoritmov s kompozičným prístupom

Príloha 11: CD príloha

Príloha 1: Popis úloh a vstupných súborov

Popis úloh

Príklady, na ktorých som testoval navrhnuté algoritmy boli z reálnej slovenskej cestnej siete [48] alebo zo zdroja [6]. Rozsahy testovacích príkladov boli rôzne v snahe čo najlepšie otestovať navrhnuté algoritmy a ich jednotlivé časti.

Popis súborov a ich štruktúra

- Benchmark - označenie testovacieho súboru
- Rozsah - rozsah riešenej úlohy (uvádzané ako počet kandidátov na stredisko x počet zákazníkov)
- Požiadavky zákazníkov - súbor s požiadavkami jednotlivých odberateľov (prvý údaj v súbore je počet odberateľov, ostatné údaje sú konkrétne hodnoty požiadaviek)
- Fixné náklady - súbor s fixnými nákladmi jednotlivých kandidátov na strediská (prvý údaj v súbore je počet kandidátov, ostatné údaje sú konkrétne hodnoty fixných nákladov)
- Matica vzdialenosti - súbor s maticou vzdialenosti jednotlivých odberateľov od kandidátov na stredisko (prvý údaj v súbore je počet kandidátov, druhý údaj počet odberateľov, ostatné údaje sú konkrétne hodnoty vzdialenosti medzi kandidátmi a odberateľmi)

Súbory použité pre úlohu p-mediánu

Tab. 30 Súbory použité pre úlohu p-mediánu

Benchmark	Súbory resp. prednastavené hodnoty		
	Matica vzdialenosti	Fixné náklady	Požiadavky zákazníkov
BB100x100	S_CBB_0100_S001_D.txt	nulové	jednotkové
BA87x87	S_CBA_0087_0001_D.txt	nulové	jednotkové
BB515x515	S_CBB_0515_0001_D.txt	nulové	jednotkové
KE460x460	S_CKE_0460_0001_D.txt	nulové	jednotkové
PO664x664	S_CPO_0664_0001_D.txt	nulové	jednotkové
NR350x350	S_CNR_0350_0001_D.txt	nulové	jednotkové
TN276x276	S_CTN_0276_0001_D.txt	nulové	jednotkové
TT249x249	S_CTT_0249_0001_D.txt	nulové	jednotkové
ZA315x315	S_CZA_0315_0001_D.txt	nulové	jednotkové
SR2916x2916	V_CSR_2916_0001_D.txt	nulové	jednotkové

Súbory použité pre úlohu váženého p-mediánu

Tab. 31 Súbory použité pre úlohu váženého p-mediánu

<i>Benchmark</i>	<i>Súbory resp. prednastavené hodnoty</i>		
	<i>Matica vzdialenosti</i>	<i>Fixné náklady</i>	<i>Požiadavky zákazníka</i>
SR1000x2916	V_CSR_1000_0100_D.txt	nulové	V_CSR_1000_0000_B.txt
BA87x87	S_CBA_0087_0001_D.txt	nulové	S_CBA_0087_0001_B.txt
BB515x515	S_CBB_0515_0001_D.txt	nulové	S_CBB_0515_0001_B.txt
KE460x460	S_CKE_0460_0001_D.txt	nulové	S_CKE_0460_0001_B.txt
PO664x664	S_CPO_0664_0001_D.txt	nulové	S_CPO_0664_0001_B.txt
NR350x350	S_CNR_0350_0001_D.txt	nulové	S_CNR_0350_0001_B.txt
TN276x276	S_CTN_0276_0001_D.txt	nulové	S_CTN_0276_0001_B.txt
TT249x249	S_CTT_0249_0001_D.txt	nulové	S_CTT_0249_0001_B.txt
ZA315x315	S_CZA_0315_0001_D.txt	nulové	S_CZA_0315_0001_B.txt
SR2916x2916	V_CSR_2916_0001_D.txt	nulové	V_CSR_2916_0000_B.txt

Súbory použité pre kapacitne neobmedzenú umiestňovaciu úlohu

Tab. 32 Súbory použité pre kapacitne neobmedzenú umiestňovaciu úlohu

<i>Benchmark</i>	<i>Súbory resp. prednastavené hodnoty</i>		
	<i>Matica vzdialenosti</i>	<i>Fixné náklady</i>	<i>Požiadavky zákazníka</i>
A	Lokacna_C_UncapIntA.txt	Lokacna_F_UncapIntA.txt	jednotkové
B	Lokacna_C_UncapIntB.txt	Lokacna_F_UncapIntB.txt	jednotkové
C	Lokacna_C_UncapIntC.txt	Lokacna_F_UncapIntC.txt	jednotkové
1000_0101	V_CSR_1000_0100_D.txt	V_CSR_1000_0101_F.txt	V_CSR_1000_0000_B.txt
1000_0102	V_CSR_1000_0100_D.txt	V_CSR_1000_0102_F.txt	V_CSR_1000_0000_B.txt
1000_0103	V_CSR_1000_0100_D.txt	V_CSR_1000_0103_F.txt	V_CSR_1000_0000_B.txt
1000_0104	V_CSR_1000_0100_D.txt	V_CSR_1000_0104_F.txt	V_CSR_1000_0000_B.txt
1000_0105	V_CSR_1000_0100_D.txt	V_CSR_1000_0105_F.txt	V_CSR_1000_0000_B.txt
1000_0106	V_CSR_1000_0100_D.txt	V_CSR_1000_0106_F.txt	V_CSR_1000_0000_B.txt
1000_0107	V_CSR_1000_0100_D.txt	V_CSR_1000_0107_F.txt	V_CSR_1000_0000_B.txt

Príloha 2: Porovnanie metód nastavenia Lagrangeovho multiplikátora

Tab. 33 Porovnanie metód nastavenia Lagrangeovho multiplikátora na benchmarku ZA315x315

p	Úlohy p -mediánu						Úlohy váženého p -mediánu					
	Bisekcia		Hrubý odhad				Bisekcia		Hrubý odhad			
	pI	$t[s]$	fixné α		adaptívne α		pI	$t[s]$	fixné α		adaptívne α	
			pI	$t[s]$	pI	$t[s]$			pI	$t[s]$	pI	$t[s]$
15	10	4,21	5	1,49	7	2,09	6	0,29	12	0,65	5	0,33
30	12	2,01	8	1,66	8	0,70	10	0,19	14	0,20	18	0,38
45	14	3,06	7	0,39	8	0,58	7	0,09	6	0,18	7	0,10
60	14	1,87	6	1,49	9	0,90	8	0,09	5	0,03	7	0,08
75	14	1,56	6	0,43	9	0,79	14	0,17	9	0,10	8	0,13
90	14	1,40	7	0,27	8	0,44	14	0,23	8	0,08	10	0,15
105	14	1,38	5	0,89	8	1,95	14	0,11	7	0,07	8	0,08
120	14	1,47	5	0,27	8	1,85	11	0,11	6	0,04	8	0,05
135	14	2,36	8	0,34	9	0,61	10	0,07	5	0,04	8	0,05
150	14	1,45	7	0,31	9	0,76	12	0,08	7	0,04	8	0,04
165	14	1,45	8	0,34	9	1,00	14	0,12	6	0,05	7	0,04
180	14	1,43	6	0,25	9	0,36	14	0,14	6	0,06	8	0,09
195	14	1,40	5	0,22	9	0,32	14	0,14	6	0,07	8	0,07
210	14	1,45	7	0,47	9	0,33	11	0,09	5	0,03	8	0,05
225	14	1,36	4	0,14	8	0,28	14	0,11	6	0,05	9	0,05
240	14	1,59	7	0,35	8	0,27	13	0,09	6	0,02	10	0,05
255	14	1,33	5	0,14	7	0,47	14	0,11	6	0,04	12	0,05
270	14	1,43	4	0,15	7	0,28	14	0,15	5	0,03	9	0,04
285	14	1,48	7	0,34	6	0,19	14	0,10	5	0,02	10	0,05
300	14	1,36	6	0,30	5	0,89	14	0,11	5	0,02	6	0,03

¹ stĺpec p udáva maximálny počet vybudovaných stredísk

² stĺpec pI udáva počet iterácií v testovanom benchmarku pre riešenú úlohu

³ stĺpec $t[s]$ udáva výpočtový čas v testovanom benchmarku pre riešenú úlohu

⁴ stĺpec Bisekcia udáva získanie Lagrangeovho multiplikátora bisekčným algoritmom

⁵ stĺpec Hrubý odhad s fixným α udáva získanie Lagrangeovho multiplikátora hrubým odhadom s nemeniacim sa parametrom α

⁶ stĺpec Hrubý odhad s adaptívnym α udáva získanie Lagrangeovho multiplikátora hrubým odhadom s meniacim sa parametrom α

⁷ zelená farba – najlepší výpočtový čas v testovanom benchmarku pre danú úlohu

⁸ oranžová farba – najmenší počet vykonaných iterácií v testovanom benchmarku pre danú úlohu

Príloha 3: Porovnanie iteratívnych algoritmov

Experimenty na porovnanie iteratívnych algoritmov som vykonával pre úlohu p -mediánu a úlohu váženého p -mediánu na jednotlivých slovenských krajoch a celom Slovensku. V Tab. 34 až Tab. 42 je porovnávaný základný a vylepšený algoritmus pMBBDual s vylepšovacou heuristikou V1 vo výpočtom čase, kvalite získaného riešenia a počte vykonaných iterácií.

Základný algoritmus pMBBDual s vylepšovacou heuristikou V1 je reprezentovaný algoritmom, ktorý je popísaný v kapitole 2.2.3, kde najlepšie nastavenia Lagrangeovho multiplikátora je získané bisekčným algoritmom. Vylepšenie získaného riešenia bolo dosiahnuté vkladacou heuristikou so stratégiou najlepší vhodný (viď kapitola 4.1.1).

Vylepšený algoritmus pMBBDual s vylepšovacou heuristikou V1 je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.2.2, kde metódou hrubého odhadu Lagrangeovho multiplikátora s fixnou hodnotou exponenta $\alpha = 1,1$ a parametrom delenia $r = 8$ hľadám najlepšie nastavenie Lagrangeovho multiplikátora. Vylepšenie získaného riešenia bolo dosiahnuté vkladacou heuristikou so stratégiou najlepší vhodný (viď kapitola 4.1.1).

V Tab. 34 až Tab. 42 hodnota p udáva testovaný maximálny počet vybudovaných stredísk, DH udáva dolnú hranicu riešenia, HH udáva hodnotu získaného riešenia, t udáva získaný výpočtový čas, pI udáva počet vykonaných iterácií, farebne je vyznačené získanie optimálneho riešenia (v stĺpcoch HH) a symbolom 3600* je vyznačené predčasné ukončenie algoritmu po jednej hodine výpočtu (v stĺpcoch t).

Tab. 34 Porovnanie iteratívnych algoritmov na Bratislavskom kraji

BA	Úloha p -mediánu								Úloha váženého p -mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	$t[s]$	pI	DH	HH	$t[s]$	pI	DH	HH	$t[s]$	pI	DH	HH	$t[s]$
5	840	840	0,11	8	840	840	0,12	7	32891	32891	0,08	6	32891	32891	0,05	4
10	541	541	0,09	8	541	541	0,04	4	18509	18509	0,03	4	18509	18509	0,03	5
15	402	405	0,18	14	402	405	0,18	10	11763	11763	0,01	3	11763	11763	0,02	6
20	317	317	0,11	14	317	317	0,10	9	7696	7696	0,02	7	7696	7696	0,02	6
25	260	261	0,13	14	260	261	0,06	8	5604	5604	0,03	9	5604	5604	0,03	10
30	219	219	0,05	14	219	219	0,06	8	4274	4274	0,02	8	4274	4274	0,01	5
35	184	187	0,05	14	184	187	0,06	9	3248	3248	0,03	7	3248	3248	0,02	7
40	157	159	0,07	14	157	159	0,04	7	2444	2444	0,02	12	2444	2444	0,02	13
45	134	134	0,03	14	134	134	0,04	6	1812	1812	0,02	9	1812	1812	0,01	6
50	111	111	0,03	11	111	111	0,03	5	1326	1326	0,02	13	1326	1326	0,02	10
55	91	91	0,09	14	91	91	0,03	7	948	948	0,02	14	948	948	0,02	11
60	71	71	0,04	14	71	71	0,03	7	685	685	0,02	14	685	685	0,02	10
65	55	55	0,03	14	55	55	0,02	8	485	485	0,02	14	485	485	0,04	12
70	40	40	0,05	14	40	40	0,01	8	311	311	0,02	14	311	311	0,02	9
75	25	25	0,03	14	25	25	0,01	8	165	165	0,02	12	165	165	0,01	11
80	12	12	0,03	14	12	12	0,01	8	66	66	0,03	14	66	66	0,02	10
85	2	2	0,02	13	2	2	0,01	6	14	14	0,01	10	14	14	0,01	6

Tab. 35 Porovnanie iteratívnych algoritmov na Banskobystrickom kraji

BB	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	t[s]	pI	DH	HH	t[s]	pI	DH	HH	t[s]	pI	DH	HH	t[s]
15	6399	6399	53,50	9	6399	6399	3,41	6	46716	46716	0,96	5	46716	46716	0,63	5
35	3919	3919	149,05	12	3919	3919	75,96	5	24172	24172	0,71	9	24172	24172	0,24	6
55	2927	2930	25,55	14	2927	2930	3,29	7	16461	16461	0,33	5	16461	16461	0,32	7
75	2388	2389	148,93	14	2388	2389	78,97	7	12457	12457	0,52	8	12457	12457	0,19	6
95	2000	2000	68,39	14	2000	2000	1,63	5	9842	9842	1,10	13	9842	9842	0,39	8
115	1708	1708	47,65	14	1708	1708	34,61	6	7980	7980	1,05	14	7980	7980	0,31	7
135	1479	1479	49,11	14	1479	1479	34,81	5	6596	6596	0,67	11	6596	6596	0,17	5
155	1296	1301	48,78	14	1296	1301	2,06	5	5517	5517	0,85	13	5517	5517	0,25	8
175	1148	1148	22,46	14	1148	1148	1,93	5	4581	4581	0,75	14	4581	4581	0,22	7
195	1023	1023	22,45	14	1023	1023	30,16	5	3826	3826	0,68	13	3826	3826	0,15	5
215	903	915	23,27	14	903	915	4,07	6	3218	3218	0,68	14	3218	3218	0,19	6
235	814	814	23,46	14	814	814	1,24	7	2703	2703	1,00	14	2703	2703	0,16	5
255	714	716	22,65	14	714	716	0,91	7	2254	2254	1,30	14	2254	2254	0,30	6
275	614	636	22,86	14	614	636	0,77	7	1868	1868	1,27	14	1868	1868	0,13	5
295	555	555	22,24	14	555	555	0,63	6	1532	1532	1,11	14	1532	1532	0,11	6
315	475	475	22,37	14	475	475	0,56	5	1242	1242	1,09	14	1242	1242	0,25	7
335	395	409	22,25	14	395	409	1,46	7	999	999	1,14	14	999	999	0,25	6
355	348	348	22,53	14	348	348	0,51	5	789	789	1,05	14	789	789	0,19	5
375	288	288	22,20	14	288	288	0,65	4	619	619	1,08	14	619	619	0,19	6
395	228	228	22,42	14	228	228	0,81	6	473	473	1,06	14	473	473	0,16	6
415	168	173	22,49	14	168	173	1,04	6	353	353	1,15	14	353	353	0,23	6
435	133	133	22,56	14	133	133	0,68	5	240	240	1,10	14	240	240	0,20	7
455	93	93	22,49	14	93	93	2,39	7	158	158	1,15	14	158	158	0,20	5
475	53	53	22,77	14	53	53	2,49	7	82	82	1,07	14	82	82	0,13	4
495	19	19	22,39	14	19	19	2,19	6	32	32	1,12	14	32	32	0,23	5

Tab. 36 Porovnanie iteratívnych algoritmov na Trenčianskom kraji

TN	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	t[s]	pI	DH	HH	t[s]	pI	DH	HH	t[s]	pI	DH	HH	t[s]
15	2178	2178	0,82	9	2178	2178	0,56	5	24567	24567	0,23	7	24567	24567	0,24	5
30	1382	1382	0,81	14	1382	1382	0,32	6	14769	14769	0,14	7	14769	14769	0,13	5
45	1022	1022	0,58	13	1022	1022	0,32	9	9920	9920	0,20	8	9920	9920	0,14	6
60	818	820	0,62	14	818	820	0,29	9	7280	7280	0,25	11	7280	7280	0,15	9
75	680	682	0,70	14	680	682	0,32	9	5512	5512	0,18	14	5512	5512	0,10	9
90	578	578	0,54	12	578	578	0,19	7	4249	4249	0,09	12	4249	4249	0,06	8
105	488	495	0,59	14	488	495	0,21	7	3348	3348	0,11	14	3348	3348	0,07	8
120	418	419	0,58	14	418	419	0,21	7	2658	2658	0,12	14	2658	2658	0,10	9
135	358	358	0,52	14	358	358	0,15	7	2072	2072	0,13	14	2072	2072	0,07	8
150	298	298	0,53	14	298	298	0,16	7	1569	1569	0,12	14	1569	1569	0,05	7
165	238	244	0,64	14	238	244	0,18	7	1187	1187	0,10	14	1187	1187	0,05	9
180	199	199	0,50	14	199	199	0,15	8	858	858	0,10	14	858	858	0,04	10
195	154	154	0,52	14	154	154	0,16	8	606	606	0,10	14	606	606	0,05	9
210	109	109	0,54	14	109	109	0,18	8	401	401	0,10	14	401	401	0,04	8
225	78	78	0,49	14	78	78	0,15	7	242	242	0,09	14	242	242	0,03	8
240	48	48	0,50	14	48	48	0,15	7	136	136	0,10	14	136	136	0,04	8
255	21	21	0,49	14	21	21	0,15	6	62	62	0,10	14	62	62	0,03	7
270	6	6	0,51	14	6	6	0,15	6	11	11	0,10	14	11	11	0,04	8

Tab. 37 Porovnanie iteratívnych algoritmov na Košickom kraji

KE	Úloha p -mediánu								Úloha váženého p -mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	$t[s]$	pI	DH	HH	$t[s]$	pI	DH	HH	$t[s]$	pI	DH	HH	$t[s]$
15	4756	4756	4,83	7	4756	4756	6,04	8	46969	46969	0,27	2	46969	46969	0,19	2
35	2933	2933	12,62	12	2933	2933	15,47	8	25000	25000	0,80	8	25000	25000	0,29	6
55	2222	2222	120,26	13	2222	2222	25,83	7	17410	17410	0,33	9	17410	17410	0,14	5
75	1806	1806	272,19	14	1806	1806	151,47	8	13192	13192	0,45	10	13192	13192	0,20	6
95	1506	1508	272,19	14	1506	1508	216,44	9	10342	10342	0,45	12	10342	10342	0,46	12
115	1269	1273	148,71	14	1269	1273	79,92	8	8228	8228	0,34	10	8228	8228	0,19	7
135	1094	1098	402,12	14	1094	1098	28,22	8	6695	6695	0,26	7	6695	6695	0,19	6
155	957	963	428,42	14	957	963	154,16	8	5575	5575	0,52	14	5575	5575	0,27	8
175	841	845	720,75	14	841	845	175,70	7	4641	4641	0,63	14	4641	4641	0,23	7
195	744	744	503,58	14	744	744	42,74	6	3867	3867	0,58	14	3867	3867	0,16	6
215	644	657	502,51	14	644	657	45,83	7	3227	3227	0,57	14	3227	3227	0,20	8
235	574	574	79,99	14	574	574	3,77	7	2680	2680	0,56	14	2680	2680	0,22	10
255	494	494	79,46	14	494	494	3,93	7	2206	2206	0,51	12	2206	2206	0,20	10
275	414	431	79,99	14	414	431	3,95	7	1789	1789	0,47	14	1789	1789	0,23	12
295	371	371	78,92	14	371	371	3,81	8	1410	1410	0,47	14	1410	1410	0,20	10
315	311	311	78,92	14	311	311	3,82	8	1083	1083	0,45	14	1083	1083	0,21	9
335	251	251	78,92	14	251	251	3,87	8	837	837	0,67	14	837	837	0,20	9
355	191	192	78,92	14	191	192	3,99	8	630	630	0,48	14	630	630	0,19	9
375	152	152	78,92	14	152	152	3,76	8	451	451	0,42	14	451	451	0,20	11
395	112	112	78,92	14	112	112	3,90	8	298	298	0,43	14	298	298	0,15	9
415	72	72	78,92	14	72	72	3,96	7	178	178	0,48	14	178	178	0,20	8
435	32	32	79,46	14	32	32	4,00	7	83	83	0,44	14	83	83	0,17	8
455	5	5	78,92	14	5	5	3,74	6	11	11	0,44	14	11	11	0,16	8

Tab. 38 Porovnanie iteratívnych algoritmov na Nitrianskom kraji

NR	Úloha p -mediánu								Úloha váženého p -mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	$t[s]$	pI	DH	HH	$t[s]$	pI	DH	HH	$t[s]$	pI	DH	HH	$t[s]$
15	3378	3378	10,25	9	3378	3378	8,21	7	40952	40952	0,41	5	40952	40952	0,60	8
30	2316	2316	12,54	12	2316	2316	5,77	7	25445	25445	0,39	8	25445	25445	0,32	8
45	1812	1812	130,24	13	1812	1812	58,16	8	18525	18525	0,41	10	18525	18525	0,51	12
60	1472	1472	66,99	12	1472	1472	12,98	7	14496	14496	0,61	12	14496	14496	0,14	5
75	1236	1236	39,92	12	1236	1236	17,47	7	11512	11512	0,34	9	11512	11512	0,16	6
90	1056	1056	42,05	14	1056	1056	20,38	8	9278	9278	0,56	14	9278	9278	0,34	9
105	916	923	42,67	14	916	923	3,41	8	7633	7633	0,39	14	7633	7633	0,15	8
120	805	805	39,07	14	805	805	1,98	8	6371	6371	0,39	14	6371	6371	0,19	10
135	707	707	39,04	14	707	707	3,20	8	5304	5304	0,37	14	5304	5304	0,14	8
150	617	626	41,13	14	617	626	1,99	8	4386	4386	0,27	14	4386	4386	0,08	6
165	549	549	39,07	14	549	549	1,67	7	3621	3621	0,23	13	3621	3621	0,07	6
180	474	481	39,84	14	474	481	1,41	6	2965	2965	0,25	14	2965	2965	0,08	7
195	399	421	38,94	14	399	421	1,63	7	2414	2414	0,24	14	2414	2414	0,08	7
210	360	360	38,21	14	360	360	1,47	7	1954	1954	0,24	14	1954	1954	0,09	8
225	300	300	39,53	14	300	300	1,38	7	1556	1556	0,25	14	1556	1556	0,10	11
240	240	252	39,22	14	240	252	1,37	7	1194	1194	0,24	14	1194	1194	0,09	9
255	207	207	38,75	14	207	207	1,52	8	893	893	0,23	14	893	893	0,09	9
270	162	162	38,60	14	162	162	1,46	8	655	655	0,24	14	655	655	0,09	10
285	117	118	38,44	14	117	118	1,53	8	457	457	0,23	14	457	457	0,09	10
300	88	88	38,52	14	88	88	1,19	8	301	301	0,23	14	301	301	0,08	10
315	58	58	38,62	14	58	58	1,24	8	174	174	0,24	14	174	174	0,09	10
330	28	28	38,65	14	28	28	1,26	7	78	78	0,24	14	78	78	0,10	8
345	5	5	38,51	14	5	5	1,21	6	14	14	0,22	14	14	14	0,09	8

Tab. 39 Porovnanie iteratívnych algoritmov na Prešovskom kraji

PO	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	t[s]	pI	DH	HH	t[s]	pI	DH	HH	t[s]	pI	DH	HH	t[s]
15	8546	8546	22,92	7	8546	8546	23,53	5	55422	55422	6,84	8	55422	55422	1,25	4
35	5271	5271	175,82	8	5271	5271	330,12	7	32281	32281	5,38	12	32281	32281	1,91	11
55	4031	4038	3161,19	14	4031	4038	2111,22	10	23268	23268	3,47	13	23268	23268	0,81	6
75	3308	3308	3600*	14	3308	3308	2562,95	7	18273	18273	6,15	13	18273	18273	1,75	9
95	2814	2815	1704,05	14	2814	2815	2174,71	8	14788	14788	4,63	12	14788	14788	2,40	11
115	2425	2426	425,56	14	2425	2426	657,23	8	12341	12341	6,39	14	12341	12341	1,87	9
135	2147	2147	993,83	14	2147	2147	428,90	9	10429	10429	4,11	12	10429	10429	0,91	7
155	1931	1931	784,18	14	1931	1931	206,45	8	8904	8904	6,26	14	8904	8904	1,60	7
175	1743	1743	732,65	14	1743	1743	519,07	8	7689	7689	4,32	13	7689	7689	1,15	8
195	1581	1587	670,95	14	1581	1587	419,10	9	6656	6656	5,56	14	6656	6656	1,43	9
215	1444	1447	486,84	14	1444	1447	170,71	8	5791	5791	8,27	14	5791	5791	2,40	8
235	1323	1323	511,01	14	1323	1323	167,82	8	5055	5055	6,22	12	5055	5055	1,85	8
255	1203	1214	509,92	14	1203	1214	161,75	7	4414	4414	4,52	14	4414	4414	0,97	8
275	1112	1112	494,84	14	1112	1112	129,11	6	3843	3844	4,35	14	3843	3844	1,06	8
295	1012	1012	495,93	14	1012	1012	129,33	6	3347	3347	3,85	14	3347	3347	1,12	9
315	912	929	495,37	14	912	929	320,76	7	2907	2907	3,83	14	2907	2907	0,86	8
335	845	845	439,58	14	845	845	114,40	7	2502	2502	3,72	14	2502	2502	0,87	8
355	765	765	442,62	14	765	765	115,69	7	2154	2154	3,48	14	2154	2154	0,75	8
375	685	685	443,38	14	685	685	115,52	7	1850	1850	3,57	14	1850	1850	0,83	8
395	605	624	442,40	14	605	624	117,93	7	1588	1588	3,62	14	1588	1588	0,86	8
415	525	564	443,87	14	525	564	135,87	7	1348	1348	3,79	14	1348	1348	1,01	8
435	501	501	439,27	14	501	501	146,17	8	1139	1139	3,64	14	1139	1139	0,86	7
455	441	441	438,94	14	441	441	116,57	8	946	946	3,53	14	946	946	0,79	8
475	381	381	438,85	14	381	381	116,66	8	774	774	3,52	14	774	774	0,76	7
495	321	321	440,30	14	321	321	116,99	8	618	618	3,28	12	618	618	0,73	6
515	261	274	439,03	14	261	274	119,27	8	498	498	3,74	14	498	498	1,02	8
535	234	234	471,86	14	234	234	133,74	8	391	391	3,80	14	391	391	0,91	7
555	194	194	471,33	14	194	194	135,01	8	308	308	3,54	14	308	308	0,87	7
575	154	154	471,48	14	154	154	134,16	8	228	228	3,79	14	228	228	1,02	7
595	114	114	471,95	14	114	114	133,29	8	162	162	3,56	14	162	162	0,86	8
615	74	74	472,02	14	74	74	126,96	7	102	102	3,87	14	102	102	1,10	8
635	34	34	472,15	14	34	34	126,80	7	56	56	3,60	14	56	56	0,85	8
655	9	9	471,35	14	9	9	126,25	6	16	16	3,85	14	16	16	1,09	8

Tab. 40 Porovnanie iteratívnych algoritmov na Trnavskom kraji

TT	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
p	DH	HH	t[s]	pI	DH	HH	t[s]	pI	DH	HH	t[s]	pI	DH	HH	t[s]	pI
15	2064	2079	2,07	14	2064	2079	1,68	9	27831	27831	0,34	9	27831	27831	0,63	17
30	1336	1336	0,59	8	1336	1336	0,32	4	16306	16306	0,23	9	16306	16306	0,30	14
45	1004	1006	7,16	14	1004	1006	4,81	8	11631	11631	0,11	11	11631	11631	0,04	6
60	792	792	35,61	14	792	792	13,53	8	8606	8606	0,12	10	8606	8606	0,07	6
75	635	635	17,92	14	635	635	8,02	8	6445	6445	0,08	12	6445	6445	0,02	6
90	522	523	2,42	14	522	523	0,53	8	4950	4950	0,10	13	4950	4950	0,03	6
105	433	438	2,44	14	433	438	0,38	8	3799	3799	0,08	13	3799	3799	0,02	5
120	363	363	2,42	14	363	363	0,32	6	2890	2890	0,07	13	2890	2890	0,03	6
135	288	303	2,49	14	288	303	0,40	7	2212	2212	0,09	14	2212	2212	0,05	7
150	242	242	2,63	14	242	242	0,29	7	1639	1639	0,07	8	1639	1639	0,03	4
165	194	194	2,41	13	194	194	0,63	7	1196	1196	0,07	14	1196	1196	0,05	9
180	149	149	2,38	14	149	149	0,30	8	841	841	0,09	14	841	841	0,05	10
195	104	104	2,41	14	104	104	0,31	8	563	563	0,07	14	563	563	0,04	10
210	71	71	2,37	14	71	71	0,30	8	338	338	0,06	14	338	338	0,04	13
225	41	41	2,38	14	41	41	0,31	8	160	160	0,07	14	160	160	0,05	16
240	11	11	2,40	14	11	11	0,32	7	33	33	0,07	14	33	33	0,03	7

Tab. 41 Porovnanie iteratívnych algoritmov na Žilinskom kraji

ZA	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
p	DH	HH	t[s]	pI	DH	HH	t[s]	pI	DH	HH	t[s]	pI	DH	HH	t[s]	pI
15	2803	2803	3,26	10	2803	2803	1,53	5	36696	36696	0,40	6	36696	36696	0,47	12
30	1832	1832	1,51	12	1832	1832	1,50	8	22209	22209	0,17	10	22209	22209	0,18	14
45	1400	1400	1,93	14	1400	1400	0,35	7	15756	15756	0,10	7	15756	15756	0,10	6
60	1138	1139	2,23	14	1138	1139	1,55	6	11771	11771	0,09	8	11771	11771	0,04	5
75	944	945	1,97	14	944	945	0,39	6	9023	9023	0,23	14	9023	9023	0,10	9
90	801	801	1,71	14	801	801	0,26	7	7013	7013	0,11	14	7013	7013	0,08	8
105	686	687	1,73	14	686	687	0,82	5	5594	5594	0,13	14	5594	5594	0,05	7
120	595	595	1,74	14	595	595	0,27	5	4476	4476	0,10	11	4476	4476	0,04	6
135	505	519	1,76	14	505	519	0,32	8	3553	3553	0,07	10	3553	3553	0,03	5
150	441	441	1,73	14	441	441	0,28	7	2788	2788	0,09	12	2788	2788	0,04	7
165	366	376	1,80	14	366	376	0,31	8	2172	2172	0,10	14	2172	2172	0,04	6
180	312	312	1,69	14	312	312	0,22	6	1684	1684	0,21	14	1684	1684	0,06	6
195	252	257	1,73	14	252	257	0,20	5	1265	1266	0,13	14	1265	1266	0,05	6
210	192	212	1,71	14	192	212	0,43	7	923	923	0,08	11	923	923	0,03	5
225	167	167	1,68	14	167	167	0,14	4	654	654	0,09	14	654	654	0,04	6
240	122	123	1,69	14	122	123	0,32	7	434	434	0,08	13	434	434	0,02	6
255	93	93	1,69	14	93	93	0,12	5	265	265	0,10	14	265	265	0,03	6
270	63	63	1,65	14	63	63	0,13	4	146	146	0,08	14	146	146	0,02	5
285	33	33	1,67	14	33	33	0,31	7	69	69	0,08	14	69	69	0,03	5
300	14	14	1,62	14	14	14	0,28	6	20	20	0,09	14	20	20	0,02	5

Tab. 42 Porovnanie iteratívnych algoritmov na slovenskej cestnej sieti

SR	Úloha váženého p-mediánu							
	Základný				Vylepšený			
p	DH	HH	t[s]	pI	DH	HH	t[s]	pI
50	533232	533232	578,75	10	533232	533232	192,66	6
100	315963	315963	290,98	10	315963	315963	151,98	8
150	237216	237216	1250,37	9	237216	237216	267,21	5
200	191975	191975	253,40	11	191975	191975	98,04	7
250	161396	161396	380,64	11	161396	161396	136,32	7
300	138936	138936	1232,92	12	138936	138936	288,69	8
350	122124	122124	679,41	13	122124	122124	1101,08	9
400	108371	108371	1443,38	12	108371	108371	306,32	5
450	96887	96887	1643,36	14	96887	96887	518,04	7
500	87070	87070	1025,69	14	87070	87070	591,82	8
550	78632	78633	794,30	14	78632	78633	177,41	7
600	71379	71379	1122,60	13	71379	71379	724,68	6
650	65081	65081	1071,86	14	65081	65081	498,82	7
700	59530	59530	1202,32	14	59530	59530	359,89	6
750	54615	54615	1127,70	14	54615	54615	590,28	6
800	50213	50213	823,29	14	50213	50213	176,94	7
850	46247	46248	896,57	14	46247	46248	197,68	6
900	42642	42642	776,32	11	42642	42642	299,15	6
950	39361	39361	631,90	14	39361	39361	334,84	6
1000	36352	36352	547,88	14	36352	36352	134,40	6
1050	33628	33628	552,71	14	33628	33628	118,88	6
1100	31104	31105	604,25	14	31104	31105	318,82	7
1150	28735	28735	637,27	14	28735	28735	118,07	6
1200	26550	26550	826,51	14	26550	26550	202,83	5
1250	24506	24506	820,07	14	24506	24506	230,39	7
1300	22627	22627	791,62	14	22627	22627	191,44	5
1350	20860	20860	882,62	14	20860	20860	396,96	6
1400	19240	19240	481,30	14	19240	19240	154,60	7
1450	17726	17726	493,38	14	17726	17726	87,00	7
1500	16317	16317	521,03	14	16317	16317	114,70	7
1550	14974	14974	562,64	14	14974	14974	133,27	7
1600	13742	13742	454,46	14	13742	13742	59,21	6
1650	12542	12544	470,03	14	12542	12544	155,71	8
1700	11456	11456	456,34	14	11456	11456	80,07	6
1750	10429	10429	456,07	14	10429	10429	109,66	6
1800	9446	9446	429,23	12	9446	9446	163,08	6
1850	8546	8546	461,44	14	8546	8546	62,18	7
1900	7731	7731	438,89	14	7731	7731	45,80	6
1950	6970	6970	439,97	14	6970	6970	89,73	7
2000	6238	6238	429,50	14	6238	6238	36,63	6
2050	5583	5583	428,96	14	5583	5583	36,95	6
2100	4983	4983	446,95	14	4983	4983	55,19	6
2150	4383	4383	464,93	14	4383	4383	65,38	7
2200	3864	3864	432,72	14	3864	3864	36,38	6
2250	3373	3373	423,86	14	3373	3373	25,19	5
2300	2923	2923	440,77	14	2923	2923	41,57	6
2350	2514	2514	431,64	14	2514	2514	34,63	5
2400	2114	2114	450,43	14	2114	2114	52,86	6
2450	1790	1790	427,08	14	1790	1790	30,86	5
2500	1490	1490	445,60	14	1490	1490	46,16	5
2550	1190	1190	466,00	14	1190	1190	60,92	5
2600	947	947	414,46	14	947	947	22,95	5
2650	747	747	435,40	14	747	747	35,60	4
2700	547	547	457,15	14	547	547	130,73	7
2750	370	370	422,25	14	370	370	25,03	4
2800	220	220	443,19	14	220	220	44,78	6
2850	110	110	427,89	14	110	110	135,86	8
2900	14	14	414,73	14	14	14	122,64	6

Príloha 4: Experimenty pre výber kandidáta na vetvenie

Tab. 43 Porovnanie variantov funkcie vo výpočtovom čase na benchmarku BB100x100

<i>p</i>	<i>Výpočtový čas v sekundách s </i>					
	<i>V1</i>	<i>V2</i>	<i>V3</i>	<i>V4</i>	<i>V5</i>	<i>V6</i>
5	0,27	0,20	0,07	0,27	0,20	0,07
10	0,07	0,01	0,07	0,01	0,01	0,07
15	0,13	0,13	0,27	0,40	0,47	0,20
20	0,07	0,67	1,28	0,40	0,40	1,48
25	0,34	0,54	0,54	0,54	0,47	0,54
30	0,34	0,34	0,40	0,40	0,34	0,40
35	9,73	13,15	7,72	3,83	3,69	9,19
40	0,34	0,40	0,34	0,34	0,27	0,40
45	0,40	0,34	0,40	0,40	0,34	0,47
50	0,40	0,34	0,94	0,34	0,27	1,28
55	0,34	0,40	0,28	0,28	0,20	0,34
60	0,26	0,34	0,26	0,26	0,27	0,34
65	0,27	0,40	0,27	0,27	0,27	0,34
70	0,20	0,34	0,20	0,27	0,27	0,27
75	0,18	0,27	0,18	0,19	0,18	0,27
80	0,20	0,21	0,21	0,20	0,20	0,22
85	0,20	0,27	0,13	0,18	0,20	0,21
90	0,20	0,20	0,23	0,20	0,13	0,19
95	0,12	0,13	0,13	0,12	0,20	0,12

¹ stĺpec *p* udáva hodnotu maximálneho počtu stredísk

² stĺpec *V1* udáva výpočtový čas algoritmu získaný variantom *V1*

³ stĺpec *V2* udáva výpočtový čas algoritmu získaný variantom *V2*

⁴ stĺpec *V3* udáva výpočtový čas algoritmu získaný variantom *V3*

⁵ stĺpec *V4* udáva výpočtový čas algoritmu získaný variantom *V4*

⁶ stĺpec *V5* udáva výpočtový čas algoritmu získaný variantom *V5*

⁷ stĺpec *V6* udáva výpočtový čas algoritmu získaný variantom *V6*

⁸ zelená farba – najkratší výpočtový čas zo všetkých variantov

Príloha 5: Štatistiky pre výber kandidáta na vetvenie

Dvojvýberovým t-testom s nerovnosťou rozptylov (t-Test: Two-Sample Assuming Unequal Variances) testujem hypotézu $H_0: \mu_x = \mu_y$ proti dvojstrannej alternatíve $H_1: \mu_x \neq \mu_y$ alebo jednostranným alternatívam $H_{1+}: \mu_x > \mu_y$ resp. $H_{1-}: \mu_x < \mu_y$ na hladine významnosti $\alpha=0,05$. Inak povedané, zaujíma nás, aký veľký musí byť rozdiel medzi výberovými strednými hodnotami (aritmetickými priermi X a Y , aby som mohol spochybníť predpoklad.

Tab. 44 Dvojvýberový t – test pre výber vhodného variantu funkcie pre získanie kandidáta na vetvenie

Variant		V1	V2	V3	V4	V5	V6
Stredná hodnota		141,7397	560,1412	1160,433	75,34807	72,11531	1158,928
Rozptyl		308100,3	1341439	2477856	36003,63	35661,58	2469598
Porovnanie		315	315	315	315	315	315
Hypotéza rozdielu stredných hodnôt		0	0	0	0	0	0
V1	Stupeň voľnosti df	-	451	391	386	386	391
	Testovacie kritérium T	-	-5,78185	-10,8321	2,008742	2,1076	-10,8321
	$P(T \leq t)$ jednostranne	-	6,91E-09	2,18E-24	0,02263	0,017855	2,18E-24
	t kritické jednostranne	-	1,648239	1,64876	1,648811	1,648811	1,64876
	$P(T \leq t)$ dvojstranne	-	1,38E-08	4,35E-24	0,04526	0,03571	4,35E-24
	t kritické dvojstranne	-	1,965238	1,96605	1,966129	1,966129	1,96605
V2	Stupeň voľnosti df	451	-	577	331	331	577
	Testovacie kritérium T	5,781854	-	-5,45162	7,331198	7,381001	-5,44385
	$P(T \leq t)$ jednostranne	6,91E-09	-	3,7E-08	8,82E-13	6,4E-13	3,86E-08
	t kritické jednostranne	1,648239	-	1,647499	1,64947	1,64947	1,647499
	$P(T \leq t)$ dvojstranne	1,38E-08	-	7,41E-08	1,76E-12	1,28E-12	7,72E-08
	t kritické dvojstranne	1,965238	-	1,964084	1,967157	1,967157	1,964084
V3	Stupeň voľnosti df	391	577	-	323	323	628
	Testovacie kritérium T	10,83207	5,451622	-	12,14642	12,18344	0,012004
	$P(T \leq t)$ jednostranne	2,18E-24	3,7E-08	-	1,61E-28	1,18E-28	0,495213
	t kritické jednostranne	1,64876	1,647499	-	1,649585	1,649585	1,647284
	$P(T \leq t)$ dvojstranne	4,35E-24	7,41E-08	-	3,22E-28	2,36E-28	0,990426
	t kritické dvojstranne	1,96605	1,964084	-	1,967336	1,967336	1,963749
V4	Stupeň voľnosti df	386	331	323	-	628	323
	Testovacie kritérium T	-2,00874	-7,3312	-12,1464	-	0,214325	-12,1496
	$P(T \leq t)$ jednostranne	0,02263	8,82E-13	1,61E-28	-	0,415181	1,57E-28
	t kritické jednostranne	1,648811	1,64947	1,649585	-	1,647284	1,649585
	$P(T \leq t)$ dvojstranne	0,04526	1,76E-12	3,22E-28	-	0,830363	3,14E-28
	t kritické dvojstranne	1,966129	1,967157	1,967336	-	1,963749	1,967336
V5	Stupeň voľnosti df	386	331	323	628	-	323
	Testovacie kritérium T	-2,1076	-7,381	-12,1834	-0,21433	-	-12,1866
	$P(T \leq t)$ jednostranne	0,017855	6,4E-13	1,18E-28	0,415181	-	1,15E-28
	t kritické jednostranne	1,648811	1,64947	1,649585	1,647284	-	1,649585
	$P(T \leq t)$ dvojstranne	0,03571	1,28E-12	2,36E-28	0,830363	-	2,3E-28
	t kritické dvojstranne	1,966129	1,967157	1,967336	1,963749	-	1,967336
V6	Stupeň voľnosti df	391	577	628	323	323	-
	Testovacie kritérium T	10,83214	5,443848	-0,012	12,14955	12,18663	-
	$P(T \leq t)$ jednostranne	3,86E-08	3,86E-08	0,495213	1,57E-28	1,15E-28	-
	t kritické jednostranne	1,647499	2,332828	1,647284	1,649585	1,649585	-
	$P(T \leq t)$ dvojstranne	7,72E-08	7,72E-08	0,990426	3,14E-28	2,3E-28	-
	t kritické dvojstranne	1,964084	2,584377	1,963749	1,967336	1,967336	-

¹ symbol – udáva, že štatistiky medzi rovnakými variantmi nie je možné vykonať

² štatistický test bol vykonaný medzi každými dvoma variantmi, ktoré v tabuľke odpovedajú označením riadkom a označením stĺpcom

³ popis jednotlivých štatistických pojmov je uvedený v zdroj [40]

Príloha 6: Experimenty pre výber varianty minimálnej množiny umiestnení

Tab. 45 Porovnanie variantov funkcie vo výpočtovom čase na benchmarku BB100x100

p	Výpočtový čas [s]					
	$F1$	$F2$	$F3$	$F4$	$F5$	$F6$
5	7,61	2,56	3,44	2,61	2,65	1,70
10	0,42	0,96	1,33	0,39	0,70	0,21
15	0,08	0,49	0,41	0,30	0,33	0,16
20	0,58	1,02	0,60	0,42	0,57	0,66
25	0,45	0,34	0,22	0,38	0,83	0,17
30	1,42	0,39	0,44	0,27	0,54	0,22
35	0,95	0,48	1,31	0,82	0,73	0,25
40	0,60	0,22	0,35	0,37	0,15	0,18
45	0,68	1,04	1,33	2,83	4,27	0,96
50	0,48	8,88	263,35	35,08	23,79	12,08
55	0,45	0,28	0,43	0,29	0,27	0,22
60	0,38	0,19	0,21	0,17	0,18	0,18
65	0,35	0,17	0,19	0,16	0,16	0,22
70	0,33	0,20	0,23	0,68	0,21	0,21
75	0,32	0,15	0,19	0,34	0,08	0,18
80	0,25	0,16	0,09	0,01	0,01	0,16
85	0,24	0,03	0,12	0,01	0,07	0,17
90	0,21	0,00	0,01	0,02	0,01	0,18
95	0,19	0,07	0,08	0,06	0,14	0,10

¹ stĺpec p udáva hodnotu maximálneho počtu stredísk

² stĺpec $F1$ udáva výpočtový čas algoritmu získaný variantom $F1$

³ stĺpec $F2$ udáva výpočtový čas algoritmu získaný variantom $F2$

⁴ stĺpec $F3$ udáva výpočtový čas algoritmu získaný variantom $F3$

⁵ stĺpec $F4$ udáva výpočtový čas algoritmu získaný variantom $F4$

⁶ stĺpec $F5$ udáva výpočtový čas algoritmu získaný variantom $F5$

⁷ stĺpec $F6$ udáva výpočtový čas algoritmu získaný variantom $F6$

⁸ zelená farba – najkratší výpočtový čas zo všetkých variantov

Príloha 7: Štatistiky pre získanie hornej hranice

Dvojvýberovým t-testom s nerovnosťou rozptylov (t-Test: Two-Sample Assuming Unequal Variances) testujem hypotézu $H_0: \mu_x = \mu_y$ proti dvojstrannej alternatíve $H_1: \mu_x \neq \mu_y$ alebo jednostranným alternatívam $H_{1+}: \mu_x > \mu_y$ resp. $H_{1-}: \mu_x < \mu_y$ na hladine významnosti $\alpha=0,05$. Inak povedané, zaujíma nás, aký veľký musí byť rozdiel medzi výberovými strednými hodnotami (aritmetickými priermi X a Y , aby som mohol spochybníť predpoklad.

Tab. 46 Dvojvýberový t – test pre získanie minimálnej množiny umiestnení

Variant		F1	F2	F3	F4	F5	F6
Stredná hodnota		62,01514	1614,398	1821,518	1678,244	1793,268	1526,501
Rozptyl		36263,27	2969124	3101277	3072715	3092671	2959596
Porovnania		315	315	315	315	315	315
Hypotéza rozdielu stredných hodnôt		0	0	0	0	0	0
F1	Stupeň voľnosti df	-	322	321	321	321	322
	Testovacie kritérium T	-	-15,8929	-17,6299	-16,2686	-17,3707	-15,0169
	$P(T \leq t)$ jednostranne	-	1,08E-42	1,99E-49	4,02E-44	2,05E-48	2,64E-39
	t kritické jednostranne	-	1,6496	1,649614	1,649614	1,649614	1,6496
	$P(T \leq t)$ dvojstranne	-	2,15E-42	3,99E-49	8,04E-44	4,09E-48	5,28E-39
	t kritické dvojstranne	-	1,967359	1,967382	1,967382	1,967382	1,967359
F2	Stupeň voľnosti df	322	-	628	628	628	628
	Testovacie kritérium T	15,89293	-	-1,492	-0,461	-1,28942	0,640691
	$P(T \leq t)$ jednostranne	1,08E-42	-	0,068101	0,322478	0,098864	0,260979
	t kritické jednostranne	1,6496	-	1,647284	1,647284	1,647284	1,647284
	$P(T \leq t)$ dvojstranne	2,15E-42	-	0,136201	0,644956	0,197728	0,521957
	t kritické dvojstranne	1,967359	-	1,963749	1,963749	1,963749	1,963749
F3	Stupeň voľnosti df	321	628	-	628	628	628
	Testovacie kritérium T	17,62993	1,492	-	1,023387	0,201459	2,12684
	$P(T \leq t)$ jednostranne	1,99E-49	0,068101	-	0,153259	0,420202	0,016911
	t kritické jednostranne	1,649614	1,647284	-	1,647284	1,647284	1,647284
	$P(T \leq t)$ dvojstranne	3,99E-49	0,136201	-	0,306519	0,840405	0,033823
	t kritické dvojstranne	1,967382	1,963749	-	1,963749	1,963749	1,963749
F4	Stupeň voľnosti df	321	628	628	-	628	628
	Testovacie kritérium T	16,26856	0,461003	-1,02339	-	-0,82218	1,096533
	$P(T \leq t)$ jednostranne	4,02E-44	0,322478	0,153259	-	0,205644	0,136633
	t kritické jednostranne	1,649614	1,647284	1,647284	-	1,647284	1,647284
	$P(T \leq t)$ dvojstranne	8,04E-44	0,644956	0,306519	-	0,411289	0,273266
	t kritické dvojstranne	1,967382	1,963749	1,963749	-	1,963749	1,963749
F5	Stupeň voľnosti df	321	628	628	628	-	628
	Testovacie kritérium T	17,37071	1,289415	-0,20146	0,822176	-	1,924548
	$P(T \leq t)$ jednostranne	2,05E-48	0,098864	0,420202	0,205644	-	0,027369
	t kritické jednostranne	1,649614	1,647284	1,647284	1,647284	-	1,647284
	$P(T \leq t)$ dvojstranne	4,09E-48	0,197728	0,840405	0,411289	-	0,054737
	t kritické dvojstranne	1,967382	1,963749	1,963749	1,963749	-	1,963749
F6	Stupeň voľnosti df	322	628	628	628	628	-
	Testovacie kritérium T	15,01688	-0,64069	-2,12684	-1,09653	-1,92455	-
	$P(T \leq t)$ jednostranne	2,64E-39	0,260979	0,016911	0,136633	0,027369	-
	t kritické jednostranne	1,6496	1,647284	1,647284	1,647284	1,647284	-
	$P(T \leq t)$ dvojstranne	5,28E-39	0,521957	0,033823	0,273266	0,054737	-
	t kritické dvojstranne	1,967359	1,963749	1,963749	1,963749	1,963749	-

¹ symbol – udáva, že štatistiky medzi rovnakými variantmi nie je možné vykonať

² štatistický test bol vykonaný medzi každými dvoma variantmi, ktoré v tabuľke odpovedajú označením riadkom a označením stĺpcom

³ popis jednotlivých štatistických pojmov je uvedený v zdroji [40]

Príloha 8: Porovnanie algoritmov zovšeobecneného Erlenkotterovho prístupu

Experimenty na porovnanie zovšeobecnených algoritmov som vykonával pre úlohu p -mediánu a úlohu váženého p -mediánu na jednotlivých Slovenských krajoch a celom Slovensku. V Tab. 47 až Tab. 55 je porovnávaný základný a vylepšený algoritmus p MedBBDual vo výpočtom čase, kvalite získaného riešenia a počte spracovaných vrcholov v metóde vetiev a hraníc.

Základný algoritmus p MedBBDual je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.3, pozostávajúci hlavne z CDA procedúry na získanie dolnej hranice (viď kapitola 4.3.3), spôsobom spracovania vrcholu z uložených hodnôt duálnych premenných v zásobníku, algoritmu pre získanie minimálnej množiny umiestnených stredísk (viď 4.3.4) a algoritmu pre získanie kandidáta na vetvenie (viď kapitola 4.3.6).

Vylepšený algoritmus p MedBBDual je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.4, pozostávajúci hlavne z ICDA procedúry na získanie dolnej hranice (viď kapitola 4.4.1), spôsobu spracovania vrcholu stromu vetvenia s počiatočným nastavením duálnych premenných (viď kapitola 4.4.2), algoritmu pre získanie minimálnej množiny umiestnených stredísk FI (viď 4.4.4) a algoritmu pre získanie kandidáta na vetvenie $V5$ (viď kapitola 4.4.3).

V Tab. 47 až Tab. 55 hodnota p udáva testovaný maximálny počet vybudovaných stredísk, DH udáva dolnú hranicu riešenia, HH udáva hodnotu získaného riešenia, $t[s]$ udáva získaný výpočtový čas, nN udáva počet spracovaných vrcholov v metóde vetiev a hraníc, farebne je vyznačené získanie optimálneho riešenia (v stĺpcoch HH) a symbolom 3600* je vyznačené predčasné ukončenie algoritmu po jednej hodine výpočtu (v stĺpcoch $t[s]$).

Tab. 47 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Bratislavskom kraji

BA	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]
5	840	840	0,26	9	840	840	0,19	7	32891	32891	1,27	977	32891	32891	0,49	1
10	541	541	0,97	489	541	541	0,43	193	18509	18509	0,17	189	18509	18509	1,90	19
15	403	403	0,38	75	403	403	0,14	73	11763	11763	0,18	191	11763	11763	1,09	23
20	317	317	6,47	9107	317	317	0,10	85	7696	7696	152,46	216991	7696	7696	0,48	19
25	261	261	76,67	117373	261	261	0,76	1099	5604	5604	0,06	65	5604	5604	0,23	15
30	219	219	0,39	43	219	219	0,07	107	3844	4274	3600*	5388831	4274	4274	0,29	29
35	187	187	0,34	41	187	187	0,03	41	3248	3248	196,37	365825	3248	3248	0,14	17
40	159	159	0,87	233	159	159	0,10	135	2444	2444	3,06	6145	2444	2444	0,60	83
45	134	134	0,83	187	134	134	0,12	177	1812	1812	0,04	133	1812	1812	0,01	1
50	109	111	3600*	6978815	111	111	0,00	1	1326	1326	0,41	135	1326	1326	0,34	105
55	84	91	3600*	7223929	91	91	0,13	217	948	948	0,03	139	948	948	0,29	117
60	59	75	3600*	11187291	71	71	0,14	325	685	685	1,88	6099	685	685	0,21	121
65	34	57	3600*	13789177	55	55	0,00	5	398	485	3600*	9870037	485	485	0,21	135
70	9	41	3600*	16908857	40	40	0,11	289	114	337	3600*	11237263	311	311	0,23	175
75	25	25	1,02	297	25	25	0,10	287	23	165	3600*	16141853	165	165	0,01	1
80	12	12	1518,41	9322847	12	12	0,00	1	66	66	0,87	167	66	66	0,10	153
85	2	2	0,07	197	2	2	0,05	175	14	14	0,07	461	14	14	0,07	173

Tab. 48 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Banskobystrickom kraji

BB	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]
15	6399	6399	293,42	503	6399	6399	538,48	1893	46716	46716	136,04	3261	46716	46716	103,97	29
35	3896	3942	3600*	42745	3896	3922	3600*	26167	24113	24172	3600*	26755	24172	24172	149,45	143
55	2929	2929	645,61	3811	2917	2929	3600*	46715	15701	18692	3600*	27929	16461	16461	61,42	115
75	2374	2394	3600*	4455	2375	2390	3600*	60395	12457	12457	681,73	4265	12457	12457	1237,02	2401
95	2000	2000	354,84	1145	2000	2000	2933,98	39671	9257	11185	3600*	32413	9842	9842	2119,03	12025
115	1698	1738	3600*	26535	1701	1710	3600*	70539	7903	8307	3600*	24837	7980	7980	242,48	1681
135	1479	1479	1884,17	1871	1470	1480	3600*	63609	6596	6596	10,27	3	6596	6596	118,53	1301
155	1298	1298	600,73	531	1298	1298	22,37	547	5517	5517	810,81	1571	5517	5517	126,01	1135
175	1136	1184	3600*	34875	1148	1148	93,25	1729	4579	4581	3600*	30661	4581	4581	45,48	467
195	1023	1023	875,87	1995	1023	1023	119,93	2001	3679	4221	3600*	46741	3826	3826	0,10	1
215	914	914	974,52	1545	914	914	24,75	859	2779	3927	3600*	53681	3218	3218	146,78	1915
235	814	814	1063,72	1917	814	814	100,66	1921	2689	2816	3600*	37405	2703	2703	84,59	1309
255	714	716	3600*	35807	715	715	91,82	2003	2254	2254	1325,46	1939	2254	2254	123,26	1931
275	614	652	3600*	49297	635	635	73,15	1851	1850	1938	3600*	37767	1868	1868	107,06	1907
295	555	555	1359,40	1881	555	555	68,50	1879	1532	1532	1221,39	1643	1532	1532	58,08	1269
315	475	475	1414,97	1913	475	475	82,08	2019	1232	1310	3600*	45359	1242	1242	79,62	1905
335	408	408	1961,44	1315	408	408	30,61	1313	932	1172	3600*	81731	999	999	75,75	1915
355	348	348	1947,49	1797	348	348	47,30	1797	789	789	1586,33	1071	789	789	29,13	1053
375	288	288	2022,48	1819	288	288	51,96	1885	609	638	3600*	61911	619	619	66,87	1755
395	228	228	1920,93	1821	228	228	51,42	1885	473	473	1506,33	1323	473	473	36,24	1323
415	173	173	2512,71	1971	173	173	38,63	1697	353	353	2234,39	1821	353	353	49,74	1821
435	133	133	2307,23	1451	133	133	10,08	383	240	240	1995,48	1557	240	240	41,22	1581
455	93	93	2669,04	1821	93	93	37,70	1811	158	158	2392,34	1751	158	158	40,38	1751
475	53	53	2430,00	1825	53	53	40,88	1843	78	82	3600*	204953	82	82	30,80	1215
495	13	19	3600*	290109	19	19	0,07	3	32	32	1929,60	1277	32	32	25,05	1277

Tab. 49 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Košickom kraji

KE	Úloha p -mediánu								Úloha váženého p -mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	$t[s]$	nN	DH	HH	$t[s]$	nN	DH	HH	$t[s]$	nN	DH	HH	$t[s]$
15	4756	4756	45,65	41	4756	4756	734,80	3821	46969	46969	1,82	1	46969	46969	3,51	1
35	2920	2945	3600*	35255	2933	2933	947,08	11519	24597	26278	3600*	28409	25000	25000	1,43	1
55	2222	2222	984,38	1033	2209	2222	3600*	69599	17410	17410	1562,94	18851	17410	17410	127,42	341
75	1795	1806	3600*	5461	1795	1806	3600*	68373	12346	14903	3600*	42843	13192	13192	70,03	263
95	1508	1508	2396,00	2263	1503	1508	3600*	81929	7306	13488	3600*	49403	10342	10342	1241,98	7115
115	1270	1270	1580,09	66211	1266	1271	3600*	105251	7938	9440	3600*	52389	8228	8228	42,65	171
135	1097	1097	830,82	1883	1093	1097	3600*	121023	6690	6695	3600*	44331	6695	6695	849,41	4671
155	956	962	3600*	55989	957	962	3600*	159477	5568	5575	3600*	41649	5575	5575	93,62	509
175	841	844	3600*	37033	842	846	3600*	193125	4606	4816	3600*	48777	4641	4641	216,16	1729
195	741	744	3600*	4123	742	744	3600*	145193	3864	3868	3600*	62761	3867	3867	64,98	469
215	654	654	647,88	1831	653	654	3600*	176439	3201	3355	3600*	61409	3227	3227	208,47	2067
235	574	574	806,74	1837	573	574	3600*	190625	2541	3068	3600*	82917	2680	2680	43,92	991
255	494	494	1442,33	2277	493	494	3600*	194221	1881	2729	3600*	94601	2206	2206	2,97	45
275	431	431	835,53	951	431	431	31,58	1457	1783	1819	3600*	62675	1789	1789	86,59	1663
295	371	371	862,87	1563	371	371	30,79	1565	1363	1613	3600*	108247	1410	1410	82,90	1683
315	311	311	870,05	1569	311	311	31,21	1577	1083	1083	3109,31	44329	1083	1083	32,77	755
335	251	251	829,34	1569	251	251	31,14	1577	837	837	1019,83	1615	837	837	60,72	1591
355	192	192	1209,59	1743	192	192	29,00	1723	630	630	1039,69	1659	630	630	49,69	1665
375	152	152	1027,28	869	152	152	8,92	585	451	451	818,61	983	451	451	23,71	981
395	112	112	1192,37	1337	112	112	18,24	1249	291	299	3600*	152323	298	298	21,06	1007
415	72	72	1220,84	1361	72	72	21,15	1469	131	207	3600*	304865	178	178	33,89	1593
435	32	32	1216,87	1361	32	32	22,97	1713	16	108	3600*	446313	83	83	23,12	1271
455	5	5	2,28	1	5	5	0,01	1	3	11	3600*	687719	11	11	16,07	1057

Tab. 50 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Nitrianskom kraji

NR	Úloha p -mediánu								Úloha váženého p -mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	$t[s]$	nN	DH	HH	$t[s]$	nN	DH	HH	$t[s]$	nN	DH	HH	$t[s]$
15	3378	3378	94,75	929	3378	3378	65,18	352	40952	40952	18,50	699	40952	40952	12,13	9
30	2316	2316	518,16	1855	2316	2316	653,34	6810	24974	27010	3600*	38931	25445	25445	34,07	69
45	1812	1812	3402,94	9531	1799	1812	3600*	57427	18209	18955	3600*	51475	18525	18525	490,28	2109
60	1462	1474	3600*	59713	1466	1472	3600*	61936	11819	17402	3600*	68099	14496	14496	770,12	4279
75	1228	1262	3600*	71951	1231	1237	3600*	67045	5429	15394	3600*	84825	11512	11512	364,52	2021
90	1056	1056	100,63	181	1056	1056	1406,63	34660	9278	9278	80,23	781	9278	9278	637,99	3979
105	911	937	3600*	85211	921	921	30,83	498	7604	7657	3600*	73459	7633	7633	417,80	3243
120	805	805	156,49	941	805	805	98,11	2610	6344	6550	3600*	78997	6371	6371	64,03	933
135	707	707	152,39	989	707	707	23,79	578	5084	6093	3600*	117007	5304	5304	33,73	457
150	617	638	3600*	115267	624	624	20,89	495	3824	5453	3600*	135945	4386	4386	35,95	755
165	527	594	3600*	153711	549	549	29,61	664	2564	4874	3600*	155781	3621	3621	21,33	445
180	437	538	3600*	181597	480	480	25,94	685	2965	2965	219,11	3183	2965	2965	22,97	749
195	420	420	238,52	605	420	420	6,98	274	2333	2637	3600*	152679	2414	2414	29,86	911
210	360	360	255,83	1249	360	360	21,71	633	1703	2303	3600*	191633	1954	1954	33,55	1193
225	300	300	252,72	1283	300	300	21,51	633	1556	1556	162,97	615	1556	1556	17,15	623
240	252	252	353,96	935	252	252	9,54	468	1194	1194	785,93	18455	1194	1194	17,50	695
255	207	207	361,61	1129	207	207	13,64	594	893	893	185,31	685	893	893	13,32	657
270	162	162	343,72	1189	162	162	13,75	596	655	655	256,07	1137	655	655	20,63	1195
285	118	118	396,86	1143	118	118	9,23	524	443	457	3600*	238311	457	457	16,74	1105
300	88	88	219,13	353	88	88	6,21	323	233	346	3600*	374047	301	301	10,28	837
315	58	58	378,76	901	58	58	7,41	406	23	212	3600*	621719	174	174	10,97	945
330	28	28	401,08	1299	28	28	10,54	650	78	78	304,93	989	78	78	10,06	997
345	5	5	0,96	1	5	5	0,01	1	14	14	12,53	4349	14	14	5,43	693

Tab. 51 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Prešovskom kraji

PO	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]
15	8546	8546	1065,60	321	8480	8611	3600*	2865	55422	55422	193,66	1353	55422	55422	505,73	77
35	5244	5271	3600*	939	5232	5271	3600*	7829	29533	35357	3600*	10399	32281	32281	489,09	219
55	4007	4033	3600*	1195	4006	4032	3600*	12237	23224	23321	3600*	10367	23268	23268	286,69	213
75	3292	3334	3600*	11587	3288	3309	3600*	18237	17668	19824	3600*	12325	18273	18273	238,37	233
95	2792	2854	3600*	10545	2806	2814	3600*	20251	14766	15000	3600*	9543	14788	14788	1108,10	1647
115	2426	2426	980,66	341	2426	2426	588,29	3933	12234	12973	3600*	11163	12341	12341	1926,29	4041
135	2141	2147	3600*	1913	2143	2147	3600*	19321	9749	12215	3600*	18259	10429	10429	173,95	389
155	1926	1931	3600*	1987	1927	1931	3600*	21565	8904	8904	1814,37	1395	8890	8904	3600*	10633
175	1743	1743	2504,10	2713	1740	1743	3600*	27105	7663	7807	3600*	8927	7689	7689	936,30	2751
195	1581	1584	3600*	2911	1581	1584	3600*	30989	6656	6656	2379,87	2181	6656	6656	1573,57	5813
215	1441	1447	3600*	15903	1441	1445	3600*	36505	5791	5791	2744,65	2041	5770	5792	3600*	17717
235	1322	1323	3600*	905	1323	1323	943,91	9501	5042	5085	3600*	9427	5055	5055	534,72	2937
255	1211	1212	3600*	1689	1212	1212	73,69	1075	4406	4433	3600*	8787	4414	4414	367,22	1701
275	1111	1113	3600*	2971	1112	1112	314,43	2465	3766	4236	3600*	15261	3843	3843	240,52	1299
295	1011	1012	3600*	3919	1012	1012	363,68	2609	3214	3799	3600*	26261	3344	3347	3600*	22401
315	923	925	3600*	2351	925	925	299,11	2645	2904	2907	3600*	2499	2907	2907	2660,20	18331
335	844	845	3600*	2399	845	845	191,93	2153	2502	2502	3553,32	529	2502	2502	323,20	2625
355	763	765	3600*	2493	765	765	250,67	2537	2154	2154	3600*	2425	2154	2154	335,54	2439
375	683	685	3600*	2483	685	685	259,61	2573	1834	1938	3600*	29705	1850	1850	331,25	2553
395	621	621	3600*	2629	621	621	230,20	2641	1569	1605	3600*	8709	1588	1588	265,75	2343
415	561	561	3600*	1071	561	561	94,56	1639	1348	1348	3600*	2517	1348	1348	586,26	5877
435	501	501	3600*	2339	501	501	167,88	2357	1139	1199	3600*	1379	1139	1139	217,43	2383
455	441	441	3600*	2339	441	441	168,97	2357	939	964	3600*	3317	946	946	188,98	2281
475	381	381	3600*	2341	381	381	170,32	2357	739	854	3600*	45521	774	774	194,88	2293
495	321	321	3600*	2321	321	321	171,24	2357	539	748	3600*	50751	618	618	0,21	1
515	261	274	3600*	18893	274	274	118,64	2083	454	540	3600*	52015	498	498	163,21	2451
535	201	234	3600*	40453	234	234	23,88	381	391	413	3600*	739	391	391	138,51	2293
555	141	194	3600*	51479	194	194	90,58	1837	308	329	3600*	681	308	308	115,43	2151
575	81	154	3600*	63751	154	154	106,82	2147	228	279	3600*	769	228	228	135,29	2431
595	21	117	3600*	81675	114	114	112,47	2205	162	175	3600*	521	162	162	94,49	1975
615	74	91	3600*	1775	74	74	116,38	2277	102	135	3600*	675	102	102	119,19	2377
635	34	54	3600*	1839	34	34	117,31	2277	42	56	3600*	127047	56	56	55,83	1271
655	9	9	11,16	1	9	9	0,04	1	16	17	3600*	261	16	16	67,11	1455

Tab. 52 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Trenčianskom kraji

TN	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]
15	2178	2178	52,59	2851	2178	2178	1,65	21	21104	27654	3600*	111817	24567	24567	0,74	1
30	1382	1382	50,57	3983	1382	1382	4,90	229	14493	14794	3600*	139943	14769	14769	9,94	45
45	1022	1022	6,89	31	1022	1022	4,92	291	9920	9920	0,70	3	9920	9920	10,03	93
60	818	820	3600*	183883	819	819	2,87	259	7218	7349	3600*	117071	7280	7280	13,96	181
75	682	682	30,59	177	682	682	2,40	255	5452	5608	3600*	120809	5512	5512	17,93	627
90	574	581	3600*	168019	578	578	2,31	343	3727	5151	3600*	215817	4249	4249	15,17	387
105	493	493	50,10	779	493	493	5,45	815	3342	3348	3600*	249305	3348	3348	10,69	603
120	417	419	3600*	300327	418	418	13,32	1363	2658	2658	36,64	329	2658	2658	7,81	327
135	358	358	51,40	405	358	358	5,22	643	2072	2072	59,50	1001	2072	2072	18,58	999
150	298	298	97,02	1033	298	298	9,67	1033	1532	1666	3600*	255023	1569	1569	9,92	625
165	244	244	102,81	1083	244	244	9,07	1203	992	1437	3600*	366213	1187	1187	15,38	985
180	199	199	88,50	765	199	199	5,21	865	452	1195	3600*	461673	858	858	9,19	785
195	154	154	100,74	979	154	154	6,78	1015	606	606	82,79	907	606	606	8,77	867
210	109	109	97,92	923	109	109	6,89	1043	401	401	82,55	641	401	401	5,04	641
225	64	78	3600*	748963	78	78	1,61	307	221	254	3600*	541599	242	242	2,94	467
240	19	52	3600*	1008591	48	48	4,86	993	136	136	183,08	703	136	136	3,86	715
255	21	21	0,50	1	21	21	0,02	1	57	62	3600*	1006775	62	62	2,68	595
270	6	6	0,47	1	6	6	0,01	1	11	11	27,88	9909	11	11	2,04	547

Tab. 53 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Trnavskom kraji

TT	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]
15	2064	2064	11,61	143	2064	2064	46,86	1153	27831	27831	9,59	1107	27831	27831	18,06	45
30	1336	1336	187,35	1443	1336	1336	480,86	23583	16043	16906	3600*	131763	16306	16306	9,19	57
45	1005	1005	280,50	2347	1005	1005	1616,57	112977	11631	11631	270,79	18235	11631	11631	11,33	121
60	792	792	1426,27	10003	784	792	3600*	331677	8541	8779	3600*	153901	8606	8606	8,06	109
75	635	635	38,30	539	635	635	8,01	1117	6445	6445	49,80	2527	6445	6445	0,06	1
90	498	561	3600*	332175	523	523	1,41	235	4950	4950	0,66	3	4950	4950	0,04	1
105	433	438	3600*	265059	438	438	4,99	999	3799	3799	19,17	313	3799	3799	9,12	297
120	363	363	38,64	941	363	363	4,53	661	2761	3001	3600*	267365	2890	2890	7,63	335
135	302	302	49,79	395	302	302	2,04	387	2212	2212	40,56	891	2212	2212	16,28	897
150	242	242	54,57	937	242	242	6,03	961	1639	1639	35,38	399	1639	1639	6,12	395
165	194	194	0,26	1	194	194	0,00	1	1196	1196	36,64	559	1196	1196	6,20	559
180	149	149	69,37	833	149	149	4,27	857	791	872	3600*	495857	841	841	7,25	753
195	104	104	67,11	833	104	104	4,49	887	563	563	78,04	495	563	563	4,02	495
210	71	71	0,29	1	71	71	0,00	1	338	338	74,69	467	338	338	3,20	467
225	41	41	53,98	551	41	41	1,96	513	160	160	3,18	903	160	160	2,68	453
240	11	11	76,33	929	11	11	3,36	881	33	33	19,76	7405	33	33	2,37	525

Tab. 54 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Žilinskom kraji

ZA	Úloha p -mediánu								Úloha váženého p -mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
	p	DH	HH	$t[s]$	nN	DH	HH	$t[s]$	nN	DH	HH	$t[s]$	nN	DH	HH	$t[s]$
15	2803	2803	129,14	2689	2803	2803	60,38	887	36696	36696	24,17	1283	36696	36696	17,09	13
30	1832	1832	37,49	123	1832	1832	291,23	7357	22209	22209	52,89	2161	22209	22209	4,14	9
45	1400	1400	980,18	40879	1400	1400	21,85	703	15756	15756	206,46	3413	15756	15756	28,14	141
60	1138	1138	134,92	1025	1138	1138	1761,89	85877	11771	11771	33,64	643	11771	11771	0,16	1
75	942	945	3600*	173089	944	944	21,39	973	9023	9023	51,64	937	9023	9023	48,96	675
90	801	801	94,19	1207	801	801	16,94	983	7013	7013	101,84	1681	7013	7013	27,67	669
105	679	697	3600*	110917	686	686	13,38	909	5594	5594	145,09	3025	5594	5594	17,28	287
120	559	651	3600*	178885	595	595	29,94	1793	4459	4480	3600*	148687	4476	4476	26,50	655
135	439	590	3600*	220747	516	516	4,62	503	3349	3930	3600*	177343	3553	3553	0,06	1
150	319	538	3600*	248933	441	441	203,05	21119	2738	2907	3600*	173363	2788	2788	0,05	1
165	199	479	3600*	272521	372	372	13,36	1117	2167	2173	3600*	230599	2172	2172	15,28	689
180	312	312	173,63	1181	312	312	14,18	1175	1676	1684	3600*	192479	1684	1684	22,94	993
195	251	258	3600*	202915	257	257	11,56	1101	1196	1479	3600*	273519	1265	1265	7,42	405
210	191	221	3600*	312711	212	212	6,71	813	716	1223	3600*	338199	923	923	12,57	787
225	131	183	3600*	391103	167	167	10,40	1173	236	979	3600*	403707	654	654	12,70	953
240	122	123	3600*	322809	123	123	10,05	1247	414	502	3600*	374355	434	434	5,99	509
255	77	96	3600*	409865	93	93	3,90	557	174	380	3600*	518093	265	265	8,92	907
270	63	63	260,65	1143	63	63	6,92	1107	128	160	3600*	490333	146	146	5,25	721
285	33	33	249,52	1145	33	33	6,78	1083	56	69	3600*	614543	69	69	4,56	727
300	14	14	0,71	1	14	14	0,01	1	20	20	613,32	13709	20	20	2,89	593

Tab. 55 Porovnanie algoritmov zovšeobecného Erlenkotterovho prístupu na Slovensku

SR	Úloha p-mediánu								Úloha váženého p-mediánu							
	Základný				Vylepšený				Základný				Vylepšený			
p	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]	nN	DH	HH	t[s]	nN
50	43824	45995	3600*	5	43825	45707	3600*	31	532434	538352	3600*	3	530812	542459	3600*	11
100	30053	31079	3600*	3	30026	31202	3600*	61	315822	315963	3600*	5	315439	317897	3600*	21
150	24012	25367	3600*	5	24015	25453	3600*	79	236928	238334	3600*	5	236954	241067	3600*	39
200	20320	21367	3600*	5	20314	21349	3600*	87	186700	208082	3600*	1079	191893	195416	3600*	55
250	17742	18806	3600*	5	17812	19079	3600*	89	137500	195447	3600*	919	161240	162327	3600*	61
300	15900	16606	3600*	3	15897	16565	3600*	103	88300	190639	3600*	911	138795	141645	3600*	75
350	14401	15295	3600*	3	14396	15230	3600*	107	117070	135007	3600*	1077	121899	124377	3600*	91
400	13151	14064	3600*	3	13170	13889	3600*	111	95770	129188	3600*	825	108274	111130	3600*	93
450	12120	12898	3600*	3	12125	12788	3600*	101	95621	103367	3600*	3	96774	98500	3600*	103
500	11219	12059	3600*	3	11214	11955	3600*	121	83022	99604	3600*	1073	86976	89340	3600*	105
550	10434	11116	3600*	3	10440	11078	3600*	111	78568	78917	3600*	3	78554	81127	3600*	129
600	9735	10559	3600*	3	9736	10378	3600*	123	71191	72811	3600*	3	71288	73861	3600*	139
650	9106	10056	3600*	3	9100	9520	3600*	175	65012	65447	3600*	3	64996	66870	3600*	137
700	8562	9466	3600*	3	8559	9345	3600*	193	59269	61646	3600*	3	59468	60970	3600*	145
750	8067	8885	3600*	3	8069	8808	3600*	199	54555	55513	3600*	3	54576	56294	3600*	173
800	7617	7937	3600*	3	7611	7880	3600*	183	50161	50912	3600*	3	50200	51877	3600*	163
850	7166	7478	3600*	3	7211	7628	3600*	189	45811	48765	3600*	3	46230	47937	3600*	149
900	6812	7031	3600*	3	6842	7554	3600*	275	42524	43642	3600*	3	42617	44127	3600*	161
950	6494	6844	3600*	3	6492	6690	3600*	311	39331	39738	3600*	3	39344	40583	3600*	193
1000	6166	6931	3600*	3	6165	6898	3600*	415	36332	36527	3600*	3	36336	37993	3600*	229
1050	5866	6218	3600*	3	5865	6162	3600*	415	33566	34396	3600*	3	33619	35128	3600*	237
1100	5566	5709	3600*	3	5566	5697	3600*	361	30816	33229	3600*	3	31095	32912	3600*	263
1150	5305	6271	3600*	3	5305	6188	3600*	377	28067	31554	3600*	1411	28723	30125	3600*	257
1200	5055	5573	3600*	3	5055	5493	3600*	441	25317	30210	3600*	1065	26530	28260	3600*	281
1250	4805	4920	3600*	3	4805	4866	3600*	439	22567	28998	3600*	843	24472	25735	3600*	307
1300	4555	4719	3600*	3	4556	4712	3600*	415	19817	27784	3600*	821	22616	24593	3600*	349
1350	4305	4582	3600*	3	4306	4563	3600*	415	17067	26636	3600*	801	20854	22563	3600*	353
1400	4055	4454	3600*	3	4092	5218	3600*	1083	17471	22833	3600*	1109	19236	20979	3600*	365
1450	3805	4304	3600*	3	3892	4548	3600*	1075	15221	21794	3600*	849	17721	19254	3600*	363
1500	3555	4170	3600*	3	3692	3943	3600*	1073	16313	16801	3600*	1	16310	18655	3600*	401
1550	3305	4033	3600*	3	3493	3536	3600*	879	14961	15521	3600*	1	14968	16518	3600*	399
1600	3055	3900	3600*	3	3293	3417	3600*	875	13611	14873	3600*	1	13738	15391	3600*	373
1650	2805	3768	3600*	3	3093	3305	3600*	873	12261	13971	3600*	1047	12540	13951	3600*	449
1700	2555	3639	3600*	3	2893	3184	3600*	869	10911	13227	3600*	809	11452	12958	3600*	567
1750	2305	3516	3600*	3	2711	3849	3600*	1187	10402	11065	3600*	1	10426	11721	3600*	627
1800	2561	3396	3600*	3	2561	3383	3600*	1177	9446	9471	3600*	1	9446	10538	3600*	689
1850	2411	2848	3600*	3	2411	2835	3600*	1169	8546	8935	3600*	1	8546	10067	3600*	695
1900	2261	2378	3600*	3	2261	2363	3600*	1165	7731	7920	3600*	1	7731	8927	3600*	675
1950	2111	2162	3600*	3	2111	2164	3600*	997	6970	7156	3600*	1	6970	8064	3600*	719
2000	1961	2073	3600*	3	1961	2075	3600*	1073	6238	6312	3600*	1	6238	7109	3600*	773
2050	1811	1982	3600*	3	1811	1991	3600*	1071	5538	5908	3600*	1	5583	6467	3600*	867
2100	1661	1862	3600*	3	1661	1879	3600*	1035	4838	5495	3600*	1527	4983	6153	3600*	819
2150	1511	1743	3600*	3	1511	1758	3600*	1063	4138	5156	3600*	1257	4382	4882	3600*	999
2200	1361	1635	3600*	3	1361	1639	3600*	1103	3438	4785	3600*	1143	3864	4582	3600*	1061
2250	1211	1524	3600*	3	1211	1521	3600*	1125	2738	4456	3600*	1043	3373	3894	3600*	1107
2300	1090	1460	3600*	3	1090	1452	3600*	1075	2038	4095	3600*	987	2923	3644	3600*	1107
2350	990	1073	3600*	3	990	1067	3600*	1065	1338	3714	3600*	939	2514	3062	3600*	1081
2400	890	904	3600*	3	890	902	3600*	1271	638	3355	3600*	873	2114	2826	3600*	1081
2450	790	816	3600*	3	790	810	3600*	1269	783	3131	3600*	1105	1790	2347	3600*	1719
2500	690	736	3600*	3	690	721	3600*	1263	183	2830	3600*	1063	1490	2154	3600*	1717
2550	590	648	3600*	3	590	634	3600*	1257	914	1733	3600*	1701	1190	1978	3600*	1721
2600	490	560	3600*	3	490	547	3600*	1261	514	1541	3600*	1491	947	1234	3600*	2047
2650	390	472	3600*	3	390	463	3600*	1255	114	1342	3600*	1409	747	1109	3600*	2061
2700	290	385	3600*	3	290	377	3600*	1245	547	714	3600*	3	547	973	3600*	2051
2750	190	295	3600*	3	190	288	3600*	1241	370	392	3600*	3	370	543	3600*	2277
2800	90	206	3600*	3	114	114	1,82	1	220	298	3600*	3	220	431	3600*	2285
2850	64	64	3600*	3	64	64	7,13	3	70	174	3600*	2113	110	167	3600*	2411
2900	14	15	3600*	3	14	15	3600*	1569	10	30	3600*	1	14	31	3600*	2185

Príloha 9: Porovnanie iteratívneho a zovšeobecneného algoritmu

Experimenty na porovnávanie iteratívneho a zovšeobecneného algoritmu som realizoval pre úlohu p -mediánu, úlohu váženého p -mediánu a kapacitne neobmedzenú umiestňovaciu úlohu na jednotlivých Slovenských krajoch, celom Slovensku [48] alebo Beasleyho úlohách [43]. V Tab. 56 až Tab. 65 je porovnávaný vylepšený algoritmus pMBBDual s vylepšovacou heuristikou V1 s vylepšeným algoritmom pMedBBDual vo výpočtom čase, kvalite získaného riešenia.

Vylepšený algoritmus pMBBDual s vylepšovacou heuristikou V1 je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.2.2, kde metódou hrubého odhadu Lagrangeovho multiplikátora s fixnou hodnotou exponenta $\alpha = 1,1$ a parametrom delenia $r = 8$ hľadám najlepšie nastavenie Lagrangeovho multiplikátora. Vylepšenie získaného riešenia bolo dosiahnuté vkladacou heuristikou so stratégiou najlepší vhodný (viď kapitola 4.1.1).

Vylepšený algoritmus pMedBBDual je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.4, pozostávajúci hlavne z ICDA procedúry na získanie dolnej hranice (viď kapitola 4.4.1), spôsobu spracovania vrcholu stromu vetvenia s počiatočným nastavením duálnych premenných (viď kapitola 4.4.2), algoritmu pre získanie minimálnej množiny umiestnených stredísk F1 (viď 4.4.4) a algoritmu pre získanie kandidáta na vetvenie V5 (viď kapitola 4.4.3).

V Tab. 56 až Tab. 65 hodnota p udáva testovaný maximálny počet vybudovaných stredísk, DH udáva dolnú hranicu riešenia, HH udáva hodnotu získaného riešenia, $t[s]$ udáva získaný výpočtový čas, farebne je vyznačené získanie optimálneho riešenia (v stĺpcoch HH) a symbolom 3600* je vyznačené predčasné ukončenie algoritmu po jednej hodine výpočtu (v stĺpcoch $t[s]$). V Tab. 64 sú výsledky realizovaných experimentov kapacitne neobmedzenej umiestňovacej úlohy.

Tab. 56 Porovnanie algoritmov na Bratislavskom kraji

BA	Úloha p -mediánu						Úloha váženého p -mediánu					
	p MBBDual			p MedBBDual			p MBBDual			p MedBBDual		
p	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$
5	840	840	0,12	840	840	0,19	32891	32891	0,05	32891	32891	0,49
10	541	541	0,04	541	541	0,43	18509	18509	0,03	18509	18509	1,90
15	402	405	0,18	403	403	0,14	11763	11763	0,02	11763	11763	1,09
20	317	317	0,10	317	317	0,10	7696	7696	0,02	7696	7696	0,48
25	260	261	0,06	261	261	0,76	5604	5604	0,03	5604	5604	0,23
30	219	219	0,06	219	219	0,07	4274	4274	0,01	4274	4274	0,29
35	184	187	0,06	187	187	0,03	3248	3248	0,02	3248	3248	0,14
40	157	159	0,04	159	159	0,10	2444	2444	0,02	2444	2444	0,60
45	134	134	0,04	134	134	0,12	1812	1812	0,01	1812	1812	0,01
50	111	111	0,03	111	111	0,00	1326	1326	0,02	1326	1326	0,34
55	91	91	0,03	91	91	0,13	948	948	0,02	948	948	0,29
60	71	71	0,03	71	71	0,14	685	685	0,02	685	685	0,21
65	55	55	0,02	55	55	0,00	485	485	0,04	485	485	0,21
70	40	40	0,01	40	40	0,11	311	311	0,02	311	311	0,23
75	25	25	0,01	25	25	0,10	165	165	0,01	165	165	0,01
80	12	12	0,01	12	12	0,00	66	66	0,02	66	66	0,10
85	2	2	0,01	2	2	0,05	14	14	0,01	14	14	0,07

Tab. 57 Porovnanie algoritmov na Banskobystrickom kraji

BB	Úloha p -mediánu						Úloha váženého p -mediánu					
	p MBBDual			p MedBBDual			p MBBDual			p MedBBDual		
p	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$
15	6399	6399	3,41	6399	6399	538,48	46716	46716	0,63	46716	46716	103,97
35	3919	3919	75,96	3896	3922	3600*	24172	24172	0,24	24172	24172	149,45
55	2927	2930	3,29	2917	2929	3600*	16461	16461	0,32	16461	16461	61,42
75	2388	2389	78,97	2375	2390	3600*	12457	12457	0,19	12457	12457	1237,02
95	2000	2000	1,63	2000	2000	2933,98	9842	9842	0,39	9842	9842	2119,03
115	1708	1708	34,61	1701	1710	3600*	7980	7980	0,31	7980	7980	242,48
135	1479	1479	34,81	1470	1480	3600*	6596	6596	0,17	6596	6596	118,53
155	1296	1301	2,06	1298	1298	22,37	5517	5517	0,25	5517	5517	126,01
175	1148	1148	1,93	1148	1148	93,25	4581	4581	0,22	4581	4581	45,48
195	1023	1023	30,16	1023	1023	119,93	3826	3826	0,15	3826	3826	0,10
215	903	915	4,07	914	914	24,75	3218	3218	0,19	3218	3218	146,78
235	814	814	1,24	814	814	100,66	2703	2703	0,16	2703	2703	84,59
255	714	716	0,91	715	715	91,82	2254	2254	0,30	2254	2254	123,26
275	614	636	0,77	635	635	73,15	1868	1868	0,13	1868	1868	107,06
295	555	555	0,63	555	555	68,50	1532	1532	0,11	1532	1532	58,08
315	475	475	0,56	475	475	82,08	1242	1242	0,25	1242	1242	79,62
335	395	409	1,46	408	408	30,61	999	999	0,25	999	999	75,75
355	348	348	0,51	348	348	47,30	789	789	0,19	789	789	29,13
375	288	288	0,65	288	288	51,96	619	619	0,19	619	619	66,87
395	228	228	0,81	228	228	51,42	473	473	0,16	473	473	36,24
415	168	173	1,04	173	173	38,63	353	353	0,23	353	353	49,74
435	133	133	0,68	133	133	10,08	240	240	0,20	240	240	41,22
455	93	93	2,39	93	93	37,70	158	158	0,20	158	158	40,38
475	53	53	2,49	53	53	40,88	82	82	0,13	82	82	30,80
495	19	19	2,19	19	19	0,07	32	32	0,23	32	32	25,05

Tab. 58 Porovnanie algoritmov na Košickom kraji

KE	Úloha p -mediánu						Úloha váženého p -mediánu					
	$pMBBDual$			$pMedBBDual$			$pMBBDual$			$pMedBBDual$		
p	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$
15	4756	4756	6,04	4756	4756	734,80	46969	46969	0,19	46969	46969	3,51
35	2933	2933	15,47	2933	2933	947,08	25000	25000	0,29	25000	25000	1,43
55	2222	2222	25,83	2209	2222	3600*	17410	17410	0,14	17410	17410	127,42
75	1806	1806	151,47	1795	1806	3600*	13192	13192	0,20	13192	13192	70,03
95	1506	1508	216,44	1503	1508	3600*	10342	10342	0,46	10342	10342	1241,98
115	1269	1273	79,92	1266	1271	3600*	8228	8228	0,19	8228	8228	42,65
135	1094	1098	28,22	1093	1097	3600*	6695	6695	0,19	6695	6695	849,41
155	957	963	154,16	957	962	3600*	5575	5575	0,27	5575	5575	93,62
175	841	845	175,70	842	846	3600*	4641	4641	0,23	4641	4641	216,16
195	744	744	42,74	742	744	3600*	3867	3867	0,16	3867	3867	64,98
215	644	657	45,83	653	654	3600*	3227	3227	0,20	3227	3227	208,47
235	574	574	3,77	573	574	3600*	2680	2680	0,22	2680	2680	43,92
255	494	494	3,93	493	494	3600*	2206	2206	0,20	2206	2206	2,97
275	414	431	3,95	431	431	31,58	1789	1789	0,23	1789	1789	86,59
295	371	371	3,81	371	371	30,79	1410	1410	0,20	1410	1410	82,90
315	311	311	3,82	311	311	31,21	1083	1083	0,21	1083	1083	32,77
335	251	251	3,87	251	251	31,14	837	837	0,20	837	837	60,72
355	191	192	3,99	192	192	29,00	630	630	0,19	630	630	49,69
375	152	152	3,76	152	152	8,92	451	451	0,20	451	451	23,71
395	112	112	3,90	112	112	18,24	298	298	0,15	298	298	21,06
415	72	72	3,96	72	72	21,15	178	178	0,20	178	178	33,89
435	32	32	4,00	32	32	22,97	83	83	0,17	83	83	23,12
455	5	5	3,74	5	5	0,01	11	11	0,16	11	11	16,07

Tab. 59 Porovnanie algoritmov na Nitrianskom kraji

NR	Úloha p -mediánu						Úloha váženého p -mediánu					
	$pMBBDual$			$pMedBBDual$			$pMBBDual$			$pMBBDual$		
p	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$
15	3378	3378	8,21	3378	3378	65,18	40952	40952	0,60	40952	40952	12,13
30	2316	2316	5,77	2316	2316	653,34	25445	25445	0,32	25445	25445	34,07
45	1812	1812	58,16	1799	1812	3600*	18525	18525	0,51	18525	18525	490,28
60	1472	1472	12,98	1466	1472	3600*	14496	14496	0,14	14496	14496	770,12
75	1236	1236	17,47	1231	1237	3600*	11512	11512	0,16	11512	11512	364,52
90	1056	1056	20,38	1056	1056	1406,63	9278	9278	0,34	9278	9278	637,99
105	916	923	3,41	921	921	30,83	7633	7633	0,15	7633	7633	417,80
120	805	805	1,98	805	805	98,11	6371	6371	0,19	6371	6371	64,03
135	707	707	3,20	707	707	23,79	5304	5304	0,14	5304	5304	33,73
150	617	626	1,99	624	624	20,89	4386	4386	0,08	4386	4386	35,95
165	549	549	1,67	549	549	29,61	3621	3621	0,07	3621	3621	21,33
180	474	481	1,41	480	480	25,94	2965	2965	0,08	2965	2965	22,97
195	399	421	1,63	420	420	6,98	2414	2414	0,08	2414	2414	29,86
210	360	360	1,47	360	360	21,71	1954	1954	0,09	1954	1954	33,55
225	300	300	1,38	300	300	21,51	1556	1556	0,10	1556	1556	17,15
240	240	252	1,37	252	252	9,54	1194	1194	0,09	1194	1194	17,50
255	207	207	1,52	207	207	13,64	893	893	0,09	893	893	13,32
270	162	162	1,46	162	162	13,75	655	655	0,09	655	655	20,63
285	117	118	1,53	118	118	9,23	457	457	0,09	457	457	16,74
300	88	88	1,19	88	88	6,21	301	301	0,08	301	301	10,28
315	58	58	1,24	58	58	7,41	174	174	0,09	174	174	10,97
330	28	28	1,26	28	28	10,54	78	78	0,10	78	78	10,06
345	5	5	1,21	5	5	0,01	14	14	0,09	14	14	5,43

Tab. 60 Porovnanie algoritmov na Prešovskom kraji

PO	Úloha p -mediánu						Úloha váženého p -mediánu					
	p MBBDual			p MedBBDual			p MBBDual			p MedBBDual		
p	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$
15	8546	8546	23,53	8480	8611	3600*	55422	55422	1,25	55422	55422	505,73
35	5271	5271	330,12	5232	5271	3600*	32281	32281	1,91	32281	32281	489,09
55	4031	4038	2111,22	4006	4032	3600*	23268	23268	0,81	23268	23268	286,69
75	3308	3308	2562,95	3288	3309	3600*	18273	18273	1,75	18273	18273	238,37
95	2814	2815	2174,71	2806	2814	3600*	14788	14788	2,40	14788	14788	1108,10
115	2425	2426	657,23	2426	2426	588,29	12341	12341	1,87	12341	12341	1926,29
135	2147	2147	428,90	2143	2147	3600*	10429	10429	0,91	10429	10429	173,95
155	1931	1931	206,45	1927	1931	3600*	8904	8904	1,60	8890	8904	3600*
175	1743	1743	519,07	1740	1743	3600*	7689	7689	1,15	7689	7689	936,30
195	1581	1587	419,10	1581	1584	3600*	6656	6656	1,43	6656	6656	1573,57
215	1444	1447	170,71	1441	1445	3600*	5791	5791	2,40	5770	5792	3600*
235	1323	1323	167,82	1323	1323	943,91	5055	5055	1,85	5055	5055	534,72
255	1203	1214	161,75	1212	1212	73,69	4414	4414	0,97	4414	4414	367,22
275	1112	1112	129,11	1112	1112	314,43	3843	3844	1,06	3843	3843	240,52
295	1012	1012	129,33	1012	1012	363,68	3347	3347	1,12	3344	3347	3600*
315	912	929	320,76	925	925	299,11	2907	2907	0,86	2907	2907	2660,20
335	845	845	114,40	845	845	191,93	2502	2502	0,87	2502	2502	323,20
355	765	765	115,69	765	765	250,67	2154	2154	0,75	2154	2154	335,54
375	685	685	115,52	685	685	259,61	1850	1850	0,83	1850	1850	331,25
395	605	624	117,93	621	621	230,20	1588	1588	0,86	1588	1588	265,75
415	525	564	135,87	561	561	94,56	1348	1348	1,01	1348	1348	586,26
435	501	501	146,17	501	501	167,88	1139	1139	0,86	1139	1139	217,43
455	441	441	116,57	441	441	168,97	946	946	0,79	946	946	188,98
475	381	381	116,66	381	381	170,32	774	774	0,76	774	774	194,88
495	321	321	116,99	321	321	171,24	618	618	0,73	618	618	0,21
515	261	274	119,27	274	274	118,64	498	498	1,02	498	498	163,21
535	234	234	133,74	234	234	23,88	391	391	0,91	391	391	138,51
555	194	194	135,01	194	194	90,58	308	308	0,87	308	308	115,43
575	154	154	134,16	154	154	106,82	228	228	1,02	228	228	135,29
595	114	114	133,29	114	114	112,47	162	162	0,86	162	162	94,49
615	74	74	126,96	74	74	116,38	102	102	1,10	102	102	119,19
635	34	34	126,80	34	34	117,31	56	56	0,85	56	56	55,83
655	9	9	126,25	9	9	0,04	16	16	1,09	16	16	67,11

Tab. 61 Porovnanie algoritmov na Trenčianskom kraji

TN	Úloha p-mediánu						Úloha váženého p-mediánu					
	pMBBDual			pMedBBDual			pMBBDual			pMBBDual		
p	DH	HH	t[s]	DH	HH	t[s]	DH	HH	t[s]	DH	HH	t[s]
15	2178	2178	0,56	2178	2178	1,65	24567	24567	0,24	24567	24567	0,74
30	1382	1382	0,32	1382	1382	4,90	14769	14769	0,13	14769	14769	9,94
45	1022	1022	0,32	1022	1022	4,92	9920	9920	0,14	9920	9920	10,03
60	818	820	0,29	819	819	2,87	7280	7280	0,15	7280	7280	13,96
75	680	682	0,32	682	682	2,40	5512	5512	0,10	5512	5512	17,93
90	578	578	0,19	578	578	2,31	4249	4249	0,06	4249	4249	15,17
105	488	495	0,21	493	493	5,45	3348	3348	0,07	3348	3348	10,69
120	418	419	0,21	418	418	13,32	2658	2658	0,10	2658	2658	7,81
135	358	358	0,15	358	358	5,22	2072	2072	0,07	2072	2072	18,58
150	298	298	0,16	298	298	9,67	1569	1569	0,05	1569	1569	9,92
165	238	244	0,18	244	244	9,07	1187	1187	0,05	1187	1187	15,38
180	199	199	0,15	199	199	5,21	858	858	0,04	858	858	9,19
195	154	154	0,16	154	154	6,78	606	606	0,05	606	606	8,77
210	109	109	0,18	109	109	6,89	401	401	0,04	401	401	5,04
225	78	78	0,15	78	78	1,61	242	242	0,03	242	242	2,94
240	48	48	0,15	48	48	4,86	136	136	0,04	136	136	3,86
255	21	21	0,15	21	21	0,02	62	62	0,03	62	62	2,68
270	6	6	0,15	6	6	0,01	11	11	0,04	11	11	2,04

Tab. 62 Porovnanie algoritmov na Trnavskom kraji

TT	Úloha p-mediánu						Úloha váženého p-mediánu					
	pMBBDual			pMedBBDual			pMBBDual			pMBBDual		
p	DH	HH	t[s]	DH	HH	t[s]	DH	HH	t[s]	DH	HH	t[s]
15	2064	2079	1,68	2064	2064	46,86	27831	27831	0,63	27831	27831	18,06
30	1336	1336	0,32	1336	1336	480,86	16306	16306	0,30	16306	16306	9,19
45	1004	1006	4,81	1005	1005	1616,57	11631	11631	0,04	11631	11631	11,33
60	792	792	13,53	784	792	3600*	8606	8606	0,07	8606	8606	8,06
75	635	635	8,02	635	635	8,01	6445	6445	0,02	6445	6445	0,06
90	522	523	0,53	523	523	1,41	4950	4950	0,03	4950	4950	0,04
105	433	438	0,38	438	438	4,99	3799	3799	0,02	3799	3799	9,12
120	363	363	0,32	363	363	4,53	2890	2890	0,03	2890	2890	7,63
135	288	303	0,40	302	302	2,04	2212	2212	0,05	2212	2212	16,28
150	242	242	0,29	242	242	6,03	1639	1639	0,03	1639	1639	6,12
165	194	194	0,63	194	194	0,02	1196	1196	0,05	1196	1196	6,20
180	149	149	0,30	149	149	4,27	841	841	0,05	841	841	7,25
195	104	104	0,31	104	104	4,49	563	563	0,04	563	563	4,02
210	71	71	0,30	71	71	0,02	338	338	0,04	338	338	3,20
225	41	41	0,31	41	41	1,96	160	160	0,05	160	160	2,68
240	11	11	0,32	11	11	3,36	33	33	0,03	33	33	2,37

Tab. 63 Porovnanie algoritmov na Žilinskom kraji

ZA	Úloha p -mediánu						Úloha váženého p -mediánu					
	p MBBDual			p MedBBDual			p MBBDual			p MBBDual		
p	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$	DH	HH	$t[s]$
15	2803	2803	1,53	2803	2803	60,38	36696	36696	0,47	36696	36696	17,09
30	1832	1832	1,50	1832	1832	291,23	22209	22209	0,18	22209	22209	4,14
45	1400	1400	0,35	1400	1400	21,85	15756	15756	0,10	15756	15756	28,14
60	1138	1139	1,55	1138	1138	1761,89	11771	11771	0,04	11771	11771	0,16
75	944	945	0,39	944	944	21,39	9023	9023	0,10	9023	9023	48,96
90	801	801	0,26	801	801	16,94	7013	7013	0,08	7013	7013	27,67
105	686	687	0,82	686	686	13,38	5594	5594	0,05	5594	5594	17,28
120	595	595	0,27	595	595	29,94	4476	4476	0,04	4476	4476	26,50
135	505	519	0,32	516	516	4,62	3553	3553	0,03	3553	3553	0,06
150	441	441	0,28	441	441	203,05	2788	2788	0,04	2788	2788	0,05
165	366	376	0,31	372	372	13,36	2172	2172	0,04	2172	2172	15,28
180	312	312	0,22	312	312	14,18	1684	1684	0,06	1684	1684	22,94
195	252	257	0,20	257	257	11,56	1265	1266	0,05	1265	1265	7,42
210	192	212	0,43	212	212	6,71	923	923	0,03	923	923	12,57
225	167	167	0,14	167	167	10,40	654	654	0,04	654	654	12,70
240	122	123	0,32	123	123	10,05	434	434	0,02	434	434	5,99
255	93	93	0,12	93	93	3,90	265	265	0,03	265	265	8,92
270	63	63	0,13	63	63	6,92	146	146	0,02	146	146	5,25
285	33	33	0,31	33	33	6,78	69	69	0,03	69	69	4,56
300	14	14	0,28	14	14	0,01	20	20	0,02	20	20	2,89

Tab. 64 Porovnanie algoritmov pre kapacitne neobmedzenú umiestňovaciu úlohu

Úlohy	Benchmark	Vylepšený p MBBDual+VI				Vylepšený p MedBBDual			
		DH	HH	NoF	$t[s]$	DH	HH	NoF	$t[s]$
Beasley	A	17 156 455	17 156 455	4	0,30	17156455	17156455	4	1,80
	B	12 979 087	12 979 087	7	0,76	12 979 087	12 979 087	7	3,47
	C	11 505 581	11 505 581	9	1,15	11 505 581	11 505 581	9	8,27
Slovensko (1000)	1000_0101	18 718 735	18 718 735	1	44,13	18 718 735	18 718 735	1	49,58
	1000_0102	356 725	356 725	181	4,96	356 725	356 725	181	1500,78
	1000_0103	270 383	270 383	284	14,38	270 342	270 386	285	3600*
	1000_0104	229 674	229 674	366	4,54	229 674	229 674	366	3440,02
	1000_0105	186 790	186 790	479	3,25	186 787	186 790	479	3600*
	1000_0106	141 950	141 950	686	1,42	141 950	141 950	686	1,84
	1000_0107	87 427	87 427	1 000	0,48	87 427	87 427	1 000	0,90

¹ stĺpec *Úlohy* udáva označenie skupiny úloh

² stĺpec *Benchmark* udáva testovanú úlohu

³ stĺpec *DH* – hodnota dolnej hranice riešenia

⁴ stĺpec *HH* udáva hornú hranicu riešenia

⁵ stĺpec *NoF* udáva počet umiestnených stredísk

⁶ stĺpec *t[s]* udáva čas trvania výpočtu algoritmu

⁷ červená farba – predčasné zastavenie algoritmu po 1 hodine výpočtu

⁸ zelená farba – získanie optimálneho riešenia

Tab. 65 Porovnanie algoritmov na Slovenskej cestnej sieti pre úlohu váženého p-mediánu

SR	Úloha váženého p-mediánu					
	p	pMBBDual			pMedBBDual	
DH		HH	t[s]	DH	HH	t[s]
50	533232	533232	192,66	530812	542459	3600*
100	315963	315963	151,98	315439	317897	3600*
150	237216	237216	267,21	236954	241067	3600*
200	191975	191975	98,04	191893	195416	3600*
250	161396	161396	136,32	161240	162327	3600*
300	138936	138936	288,69	138795	141645	3600*
350	122124	122124	1101,08	121899	124377	3600*
400	108371	108371	306,32	108274	111130	3600*
450	96887	96887	518,04	96774	98500	3600*
500	87070	87070	591,82	86976	89340	3600*
550	78632	78633	177,41	78554	81127	3600*
600	71379	71379	724,68	71288	73861	3600*
650	65081	65081	498,82	64996	66870	3600*
700	59530	59530	359,89	59468	60970	3600*
750	54615	54615	590,28	54576	56294	3600*
800	50213	50213	176,94	50200	51877	3600*
850	46247	46248	197,68	46230	47937	3600*
900	42642	42642	299,15	42617	44127	3600*
950	39361	39361	334,84	39344	40583	3600*
1000	36352	36352	134,40	36336	37993	3600*
1050	33628	33628	118,88	33619	35128	3600*
1100	31104	31105	318,82	31095	32912	3600*
1150	28735	28735	118,07	28723	30125	3600*
1200	26550	26550	202,83	26530	28260	3600*
1250	24506	24506	230,39	24472	25735	3600*
1300	22627	22627	191,44	22616	24593	3600*
1350	20860	20860	396,96	20854	22563	3600*
1400	19240	19240	154,60	19236	20979	3600*
1450	17726	17726	87,00	17721	19254	3600*
1500	16317	16317	114,70	16310	18655	3600*
1550	14974	14974	133,27	14968	16518	3600*
1600	13742	13742	59,21	13738	15391	3600*
1650	12542	12544	155,71	12540	13951	3600*
1700	11456	11456	80,07	11452	12958	3600*
1750	10429	10429	109,66	10426	11721	3600*
1800	9446	9446	163,08	9446	10538	3600*
1850	8546	8546	62,18	8546	10067	3600*
1900	7731	7731	45,80	7731	8927	3600*
1950	6970	6970	89,73	6970	8064	3600*
2000	6238	6238	36,63	6238	7109	3600*
2050	5583	5583	36,95	5583	6467	3600*
2100	4983	4983	55,19	4983	6153	3600*
2150	4383	4383	65,38	4382	4882	3600*
2200	3864	3864	36,38	3864	4582	3600*
2250	3373	3373	25,19	3373	3894	3600*
2300	2923	2923	41,57	2923	3644	3600*
2350	2514	2514	34,63	2514	3062	3600*
2400	2114	2114	52,86	2114	2826	3600*
2450	1790	1790	30,86	1790	2347	3600*
2500	1490	1490	46,16	1490	2154	3600*
2550	1190	1190	60,92	1190	1978	3600*
2600	947	947	22,95	947	1234	3600*
2650	747	747	35,60	747	1109	3600*
2700	547	547	130,73	547	973	3600*
2750	370	370	25,03	370	543	3600*
2800	220	220	44,78	220	431	3600*
2850	110	110	135,86	110	167	3600*
2900	14	14	122,64	14	31	3600*

Príloha 10: Porovnanie navrhnutých algoritmov s kompozičným prístupom

Experimenty na porovnanie iteratívneho, zovšeobecneného a kompozičného algoritmu som realizoval pre úlohu p-mediánu a úlohu váženého p-mediánu na jednotlivých slovenských krajoch a celom Slovensku, kde som nezískal optimálne riešenie iteratívnym algoritmom. V Tab. 66 až Tab. 67 je porovnávaný vylepšený algoritmus pMBBDual s vylepšovacou heuristikou V1 a vylepšený algoritmus pMedBBDual s kompozičným algoritmom vo výpočtom čase a kvalite získaného riešenia ($DH=HH$).

Vylepšený algoritmus pMBBDual s vylepšovacou heuristikou V1 je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.2.2, kde metódou hrubého odhadu Lagrangeovho multiplikátora s fixnou hodnotou exponenta $\alpha = 1,1$ a parametrom delenia $r = 8$ hľadám najlepšie nastavenie Lagrangeovho multiplikátora. Vylepšenie získaného riešenia bolo dosiahnuté vkladacou heuristikou so stratégiou najlepší vhodný (viď kapitola 4.1.1).

Vylepšený algoritmus pMedBBDual je reprezentovaný algoritmom, ktorý je popísaný v kapitole 4.4, pozostávajúci hlavne z ICDA procedúry na získanie dolnej hranice (viď kapitola 4.4.1), spôsobu spracovania vrcholu stromu vetvenia s počiatočným nastavením duálnych premenných (viď kapitola 4.4.2), algoritmu pre získanie minimálnej množiny umiestnených stredísk F1 (viď 4.4.4) a algoritmu pre získanie kandidáta na vetvenie V5 (viď kapitola 4.4.3).

Kompozičný algoritmus predstavuje vylepšenie riešenia získaného iteratívnym algoritmom pMBBDual s vylepšovacou heuristikou aplikovaním zovšeobecneného algoritmu pMedBBDual vychádzajúceho z hodnoty neprípustného riešenia s najmenšou mierou neprípustnosti získaného algoritmom pMBBDual.

V Tab. 66 až Tab. 67 predstavuje *Kraj* testovanú oblasť, hodnota p udáva maximálny počet vybudovaných stredísk, DH udáva dolnú hranicu riešenia, HH udáva hodnotu získaného riešenia, $t[s]$ udáva získaný výpočtový čas, farebne je vyznačené získanie optimálneho riešenia (v stĺpcoch HH) pri rovnosti hraníc a symbolom 3600* je vyznačené predčasné ukončenie algoritmu po jednej hodine výpočtu (v stĺpcoch $t[s]$).

Tab. 66 Porovnanie algoritmov s kompozičným algoritmom na úlohe p-mediánu

		Úloha p-mediánu								
		pMBBDual+VI			pMedBBDual			Kompozičný algoritmus		
Kraj	p	DH	HH	t[s]	DH	HH	t[s]	DH	HH	t[s]
Bratislavský	15	402	405	0,18	403	403	0,14	403	403	0,55
	25	260	261	0,06	261	261	0,76	261	261	1,57
	35	219	219	0,06	219	219	0,07	187	187	0,04
	40	184	187	0,06	187	187	0,03	159	159	0,09
Banskobystrický	55	2927	2930	3,29	2917	2929	3600*	2927	2929	3600*
	75	2388	2389	78,97	2375	2390	3600*	2388	2389	3600*
	155	1296	1301	2,06	1298	1298	22,37	1298	1298	18,15
	215	903	915	4,07	914	914	24,75	913	914	3600*
	255	714	716	0,91	715	715	91,82	714	715	3600*
	275	614	636	0,77	635	635	73,15	635	635	93,16
	335	395	409	1,46	408	408	30,61	408	408	38,69
	415	168	173	1,04	173	173	38,63	173	173	1,81
Košický	115	1269	1273	79,92	1266	1271	3600*	1270	1270	3566,91
	135	1094	1098	28,22	1093	1097	3600*	1094	1097	3600*
	155	957	963	154,16	957	962	3600*	957	961	3600*
	175	841	845	175,70	842	846	3600*	841	845	3600*
	215	644	657	45,83	653	654	3600*	653	654	3600*
	275	414	431	3,95	431	431	31,58	431	431	3,94
	355	191	192	3,99	192	192	29,00	191	192	4,49
Prešovský	55	4031	4038	2111,22	4006	4032	3600*	4031	4032	3600*
	95	2814	2815	2174,71	2806	2814	3600*	2814	2814	3336,73
	115	2425	2426	657,23	2426	2426	588,29	2426	2426	893,06
	195	1581	1587	419,10	1581	1584	3600*	1584	1584	1739,36
	215	1444	1447	170,71	1441	1445	3600*	1444	1445	3600*
	255	1203	1214	161,75	1212	1212	73,69	1212	1212	378,05
	315	912	929	320,76	925	925	299,11	925	925	988,95
	395	605	624	117,93	621	621	230,20	621	621	507,44
	415	525	564	135,87	561	561	94,56	561	561	292,94
	515	261	274	119,27	274	274	118,64	274	274	180,26
Nitriansky	105	916	923	3,41	921	921	30,83	921	921	71,77
	150	617	626	1,99	624	624	20,89	624	624	6,99
	180	474	481	1,41	480	480	25,94	480	480	16,79
	195	399	421	1,63	420	420	6,98	420	420	5,37
	240	240	252	1,37	252	252	9,54	252	252	1,68
	285	117	118	1,53	118	118	9,23	118	118	1,69
Trenčiansky	60	818	820	0,29	819	819	2,87	819	819	2,98
	75	680	682	0,32	682	682	2,40	682	682	0,5
	105	488	495	0,21	493	493	5,45	493	493	11,68
	120	418	419	0,21	418	418	13,32	418	418	1,53
	165	238	244	0,18	244	244	9,07	244	244	0,26
Trnavský	15	2064	2079	1,68	2064	2064	46,86	2064	2064	50,45
	45	1004	1006	4,81	1005	1005	1616,57	1005	1005	290,29
	90	522	523	0,53	523	523	1,41	523	523	2,16
	105	433	438	0,38	438	438	4,99	438	438	7,18
	135	288	303	0,40	302	302	2,04	302	302	1,96
Žilinský	60	1138	1139	1,55	1138	1138	1761,89	1138	1138	16,26
	75	944	945	0,39	944	944	21,39	944	944	19,32
	105	686	687	0,82	686	686	13,38	686	686	16,03
	135	505	519	0,32	516	516	4,62	516	516	6,81
	165	366	376	0,31	372	372	13,36	372	372	16,61
	195	252	257	0,20	257	257	11,56	257	257	0,33
	210	192	212	0,43	212	212	6,71	212	212	0,69
240	122	123	0,32	123	123	10,05	123	123	0,5	

Tab. 67 Porovnanie algoritmov s kompozičným algoritmom na úlohe váženého p -mediánu

		<i>Úloha váženého p-mediánu</i>								
		<i>$pMBBDual+V1$</i>			<i>$pMedBBDual$</i>			<i>Kompozičný algoritmus</i>		
<i>Kraj</i>	<i>p</i>	<i>DH</i>	<i>HH</i>	<i>$t[s]$</i>	<i>DH</i>	<i>HH</i>	<i>$t[s]$</i>	<i>DH</i>	<i>HH</i>	<i>$t(s)$</i>
Žilinský	195	1265	1266	0,05	1265	1265	7,42	1265	1265	3,31
Prešovský	275	3843	3844	1,06	3843	3843	240,52	3843	3843	85,52
Slovensko	550	78632	78633	177,41	78554	81127	3600*	78632	78633	3600*
	850	46247	46248	197,68	46230	47937	3600*	46247	46248	3600*
	1100	31104	31105	318,82	31095	32912	3600*	31104	31105	3600*
	1650	12542	12544	155,71	12540	13951	3600*	12542	12544	3600*

Príloha 11: CD príloha

CD príloha obsahuje:

1. Dizertačnú prácu (formát *pdf*)
2. Publikácie autora
3. Softvérový nástroj na riešenie úlohy návrhu VOS