

ŽILINSKÁ UNIVERZITA V ŽILINE
FAKULTA RIADENIA A INFORMATIKY

ROBUSTNÉ MODELY
V DISTRIBUČNÝCH ÚLOHÁCH

Dizertačná práca

(Evidenčné číslo 28360020193007)

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: doc. RNDr. Štefan Peško, CSc.

Prehlásenie

Čestne prehlasujem, že som túto dizertačnú prácu vypracovala samostatne a že som uviedla všetky použité pramene a literatúru, z ktorých som čerpala.

V Žiline, dňa 18.4.2019

Zuzana Borčinová

Pod'akovanie

Ďakujem môjmu školiteľovi doc. RNDr. Štefanovi Peškovi, CSc. za kvalitné odborné vedenie a inšpirujúce nápady počas celého trvania môjho doktorandského štúdia, ako aj ochotu a metodickú pomoc pri písaní dizertačnej práce.

Abstrakt

Distribučné úlohy súvisia s riadením prepravy tovaru a distribúcie služieb od dodávateľov k odberateľom. Zvyčajne sa predpokladá, že všetky vstupné údaje (prepravné náklady, čas obsluhy, požiadavky a pod.) sú v čase plánovania presne známe. V praxi však tieto dáta často bývajú neisté v dôsledku ich náhodnej povahy, výskytu chýb merania alebo z iných dôvodov. Neisté dáta môžu spôsobiť, že navrhnuté riešenie sa stane neprípustným. Robustná optimalizácia hľadá riešenie, ktoré je imúnne voči neistým vstupným údajom, za predpokladu, že rozdelenie pravdepodobnosti nie je známe. Namiesto toho predpokladáme, že neisté dáta patria ohraničenej množine neurčitostí. Cieľom robustnej optimalizácie je nájsť najlepšie riešenie, ktoré je prípustné pre ich všetky možné hodnoty z tejto množiny. V tejto práci sa zaoberáme na robustnou okružnou dopravnou úlohou, v ktorej majú byť zákazníci s neistými požiadavkami obslužení vozidlami s rovnakou kapacitou sústredenými v depe. Navrhli sme tri stratégie, ktoré pre danú množinu neurčitostí hľadali riešenie s minimálnymi neuspokojenými požiadavkami a minimálnymi prepravnými nákladmi. Dosiahnuté experimentálne výsledky ukázali, že nájdené robustné riešenia dokážu minimalizovať neuspokojené požiadavky zákazníkov, pričom ich cena je len o málo väčšia, než cena optimálneho deterministického riešenia.

Kľúčové slová: robustná optimalizácia, kapacitná okružná dopravná úloha, neisté požiadavky zákazníkov, celočíselné a zmiešané lineárne programovanie, paralelný mikrogenetický algoritmus

BORČINOVÁ ZUZANA: *Robustné modely v distribučných úlohách*. [Dizertačná práca]

Žilinská Univerzita v Žiline, Fakulta riadenia a informatiky, Katedra matematických metód a operačnej analýzy.

Vedúci: doc. RNDr. Štefan Peško, CSc.

FRI ŽU v Žiline, 2012 — 88 s.

Abstract

Distribution problems are concerned with the transfer of goods or services between manufacturing facilities, distribution centers and customers. Usually, we assume that input data (transportation costs, service times, demands etc.) is precisely known at the time of problem solving. However, in many real-life applications data are subject to uncertainty due to their random nature, measurement errors or other reasons. This random behavior of the problem data could cause the proposed feasible solution to become infeasible. Robust optimisation seeks the solution that is immune to this uncertainty under assumption that probability distribution is unknown. Instead we assume that the uncertain problem parameters belong to bounded uncertainty set. The objective of robust optimisation is to find the best solution which is feasible for any realization of the data from the given uncertainty set. In this work we consider the robust capacitated vehicle routing problem in which a set of uncertain customers' demands has to be served by a fleet of homogenous vehicles departing from a depot. We proposed three robust strategies which aim to minimise transportation costs and unsatisfied demands for the specific uncertainty set. Our experimental results show that obtained solutions can minimise unmet demands while incurring a small extra cost over deterministic optimal solution.

Keywords: robust optimisation, capacitated vehicle routing problem, uncertain customers' demands, integer and mixed linear programming, parallel micro genetic algorithm

BORČINOVÁ ZUZANA: *Robust models in distribution problems* [Dissertation thesis]

University of Žilina, Faculty of Management Science and Informatics, Department of mathematical methods.

Tutor: doc. RNDr. Štefan Peško, CSc.

FRI ŽU v Žiline, 2019 — 88 p.

Obsah

1	ÚVOD DO PROBLEMATIKY	6
1.1	Úvod	6
1.2	Definícia kapacitnej okružnej dopravnej úlohy	8
1.3	Matematický model kapacitnej okružnej dopravnej úlohy	9
1.4	Všeobecná definícia robustnej optimalizácie	10
1.5	Robustná lineárna optimalizácia	13
2	SÚČASNÝ STAV RIEŠENIA PROBLEMATIKY	15
2.1	Exaktné metódy	15
2.2	Heuristické metódy	16
2.2.1	Konštruktívne heuristiky	16
2.2.2	Zlepšujúce heuristiky	17
2.2.3	Metaheuristiky	18
2.3	CVRP s neistými parametrami	20
2.3.1	Stochastická CVRP	21
2.3.2	Robustná CVRP	21
3	ZVOLENÁ METODIKA PRÁCE A METÓDY SKÚMANIA	24
3.1	Spôsob modelovania neistých dát	24
3.2	Matematický model	25
3.3	Metódy generovania robustných riešení	25
3.3.1	Stratégie na nájdenie najhoršieho scenára	26

		2
	3.3.2	Metódy riešenia CVRP 26
3.4		Kritérium robustnej optimality 27
	3.4.1	Neuspokojené požiadavky 27
	3.4.2	Zvýšenie ceny riešenia 28
3.5		Testovacie inštancie 29
3.6		Softvérové nástroje 30
4	VÝSLEDKY PRÁCE	31
4.1		Nový model CVRP 31
	4.1.1	Model CVRP 2 34
	4.1.2	Výsledky experimentov 34
	4.1.3	Záver 35
4.2		Nová exaktná iteratívna metóda na riešenie CVRP 36
	4.2.1	Exaktná iteratívna metóda CVRP 3 36
	4.2.2	Výsledky experimentov 38
	4.2.3	Záver 39
4.3		Riešenie CVRP pomocou paralelného mikrogenetického algoritmu 40
	4.3.1	Mikrogenetický algoritmus μ GA 41
	4.3.2	Jedinec a jeho fitness 42
	4.3.3	Turnajový výber 44
	4.3.4	Kríženie 44
	4.3.5	Kontrola diverzity 45
	4.3.6	Paralelizácia algoritmu 46
	4.3.7	Paralelný mikrogenetický algoritmus $p\mu$ GA 47
	4.3.8	Výsledky experimentov 48
	4.3.9	Záver 50
4.4		Lexikografický model CVRP 52
	4.4.1	Model mBPP 53
	4.4.2	Definícia lexikografickej CVRP 53

	3
4.4.3 Model LCVRP	54
4.4.4 Výsledky experimentov	55
4.4.5 Záver	55
4.5 Riešenie robustnej okružnej dopravnej úlohy s neistými požiadavkami zákazníkov	57
4.5.1 Diskrétny scenár	57
4.5.2 Formálna definícia problému	58
4.5.3 Stratégie na riešenie RVRP	59
4.5.4 Výsledky experimentov	63
4.5.5 Záver	74
5 ZÁVER	75
5.0.1 Vedecký prínos práce	77
5.0.2 Praktický prínos práce	77
Literatúra	78

Zoznam obrázkov

1.1	Prípustné riešenie CVRP	9
2.1	Vľavo sú zákazníci i a j obslužení dvomi vozidlami, pokým vpravo len jedným	17
3.1	Optimálne deterministické riešenie CVRP	28
4.1	Prípustné riešenie CVRP, trasy vozidiel sú nahradené orientovanými cestami .	33
4.2	1-optimálna výmena $A = \{(7, 5)\}$, $B = \{(3, 5)\}$	37
4.3	Zlepšovanie riešenia v čase pre rôzne hodnoty δ	40
4.4	Procedúra <i>split</i>	43
4.5	Kríženie PMX	45
4.6	Vizualizácia riešení inštancie A-n36-k5	49
4.7	Vizualizácia riešení inštancie P-n40-k5	49
4.8	LCVRP $n = 3, p = 2, Q_{min} = 9$	52
4.9	Využitie kapacity vozidla	56
4.10	RVRP s neistými požiadavkami	58
4.11	Inštancie s neprípustným maximálnym scenárom	72
4.12	Inštancie s prípustným maximálnym scenárom	73

Zoznam tabuliek

4.1	Porovnanie časovej náročnosti modelov CVRP 1 a CVRP 2	35
4.2	Porovnanie doby výpočtu pre rôzne hodnoty δ	39
4.3	Porovnanie výsledkov podľa spôsobu generovania inicializačného riešenia . .	51
4.4	Porovnanie výsledkov získaných dvomi metódami	56
4.5	Vyhodnotenie deterministického optimálneho riešenia	66
4.6	Vyhodnotenie výsledkov stratégie <i>rvrp 1</i>	67
4.7	Vyhodnotenie výsledkov stratégie <i>rvrp 2</i>	68
4.8	Vyhodnotenie výsledkov stratégie <i>rvrp 3</i>	69

Kapitola 1

ÚVOD DO PROBLEMATIKY

1.1 Úvod

Názvom *distribučné úlohy* označujeme úlohy súvisiace s riadením prepravy tovaru, prípadne osôb a distribúcie služieb od dodávateľov k odberateľom s cieľom minimalizovať náklady. Patria k nim napríklad dopravné úlohy, ktoré riešia zostavenie optimálneho plánu rozvozu, resp. zvozu komodít, lokačné úlohy zaoberajúce sa optimálnym rozmiestnením ob-
služných stredísk, prirad'ovacie úlohy a podobne. Z dôvodu neustále rastúcej potreby šetriť palivo a čas, prípadne znižovať emisie sú tieto úlohy predmetom intenzívneho štúdia v oblasti operačného výskumu a matematického programovania.

Pri formulácii distribučných úloh sú dodávatelia definovaní svojimi kapacitami a odberatelia svojimi požiadavkami, pričom sa predpokladá, že všetky vstupné údaje sú pevne dané (*deterministické*). V praxi, však, nie sú tieto predpoklady zaručené, t. j. niektoré z údajov majú charakter náhodných veličín (*stochastické*). Na riešenie stochastických optimalizačných úloh sa používa stochastické modelovanie, ktoré súvisí s vytváraním a riešením stochastických modelov, zodpovedajúcich reálnej situácii s určitou pravdepodobnosťou, spolu s možnosťou robiť simulácie. Pomocou stochastického modelu sa dajú simulovať rozličné scenáre s rôznymi parametrami a pozorovať správanie modelu. Na základe analýzy chovania modelu je potom možné odvodiť informácie potrebné pre rozhodovanie. Pri tvorbe analytického stochastického modelu sa vždy vychádza z predpokladu, že pravdepodobnosť (rozdelenie, cha-

rakteristiky atď .), ktorou sa riadia náhodné parametre, je známa alebo sa dá presne odhadnúť.

V mnohých prípadoch je ťažké a niekedy aj nemožné presne odhadnúť rozdelenie pravdepodobnosti neistých vstupných údajov. Chybný odhad alebo nesprávna voľba pravdepodobnostného rozdelenia náhodných premenných môže viesť k chybám v stochastickom modeli. Táto skutočnosť je motiváciou k použitiu *robustných optimalizačných metód*, ktoré sú imúnne voči neurčitosti vstupných dát, čiže pri malých zmenách vstupných parametrov sa zachová výstup v blízkosti pôvodného výstupu a ani pri veľkých zmenách na vstupe nedôjde ku dramatickým zmenám, či dokonca katastrofálnym dôsledkom na výstupe. Pri robustnej optimalizácii je neistota modelovaná ako množina spojitých alebo diskretných hodnôt parametrov nazývaných scenáre. Hlavná myšlienka je vykonať optimalizáciu nad súborom scenárov a poskytnúť robustné riešenie, ktoré je kvalitné bez ohľadu na scenár (t. j. aj pri najhoršom možnom scenári).

Medzi najviac skúmané distribučné úlohy patrí *okružná dopravná úloha* – Vehicle Routing Problem (VRP). Úlohou VRP je naplánovať optimálne okružné trasy pre flotilu vozidiel, sústredených v centrálnom depe, aby obslúžili skupinu zákazníkov, pričom celkové prepravné náklady budú minimálne. K tejto úlohe sa v praktických aplikáciách často pridávajú ďalšie obmedzenia. Napríklad obsluha sa môže týkať súčasne rozvozu aj zberu komodít, pričom náklad vozidla po celej trase nesmie prekročiť jeho kapacitu, celková dĺžka trasy nesmie byť väčšia než predpísaný limit, zákazníci musia byť obslúžení v určených časových oknách, vozidlá flotily môžu byť heterogénne atď. Niektoré charakteristiky, ako sú požiadavky alebo časy jász, sa môžu dynamicky meniť. Nový trend riešenia VRP spočíva v zohľadnení neurčitosti vstupných parametrov, pričom sa hľadá robustné riešenie – Robust Vehicle Routing Problem (RVRP). Výskum sa zameriava na okružné dopravné úlohy s neistými požiadavkami, s neistými zákazníkmi, s časovými oknami a s neistými časmi jász a podobne. Táto práca je zameraná na *kapacitnú okružnú dopravnú úlohu* – Capacitated Vehicle Routing Problem (CVRP) a riešenie jej robustnej verzie *robustnej CVRP s neistými požiadavkami*.

1.2 Definícia kapacitnej okružnej dopravnej úlohy

CVRP môžeme definovať ako problém teórie grafov nasledovne [84]. Nech $\vec{G} = (V, \vec{H}, c)$ je úplný hranovo ohodnotený digraf, kde $V = \{0, 1, 2, \dots, n\}$ je množina vrcholov a $\vec{H} = \{(i, j), \forall i, j \in V, i \neq j\}$ je množina orientovaných hrán. Vrcholy $i = 1, 2, \dots, n$ reprezentujú zákazníkov s nezápornými požiadavkami d_i a vrchol 0 predstavuje depo. Každá hrana $(i, j) \in \vec{H}$ je ohodnotená nezápornou cenou c_{ij} , ktorá vyjadruje prepravné náklady medzi vrcholmi i a j . Budeme predpokladať, že matica ohodnotení hrán je symetrická, t. j. že platí $c_{ij} = c_{ji}$ pre všetky $i, j \in V$ a spĺňa trojuholníkovú nerovnosť $c_{ij} + c_{jk} \leq c_{ik}$ pre všetky $i, j, k \in V$.

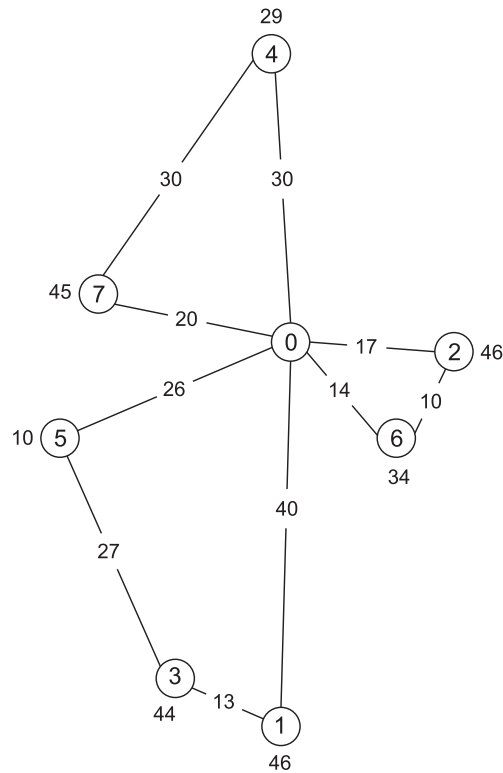
V depe je k dispozícii p homogénnych vozidiel, každé s kapacitou Q . Predpokladáme, že $p \geq p_{min}$, kde p_{min} je minimálny počet vozidiel potrebných na uspokojenie požiadaviek všetkých zákazníkov. Hodnota p_{min} sa dá určiť vyriešením úlohy o batohu – Bin Packing Problem (BPP) korešpondujúcej s CVRP. Vzhľadom na to, že BPP je NP-t'ážky problém, často sa namiesto p_{min} používa dolná hranica $\left\lceil \frac{\sum_{i=1}^n d_i}{Q} \right\rceil$. Navyše predpokladáme, že $d_i \leq Q$ pre každé $i = 1, 2, \dots, n$.

Cieľom CVRP je nájsť p cyklov, reprezentujúcich okružné trasy vozidiel, s minimálnou cenou, ktorá je definovaná ako súčet cien hrán patriacich cyklom, pričom:

- 1) každá trasa musí začínať a končiť v depe,
- 2) každý zákazník bude navštívený práve raz,
- 3) na žiadnej trase nebude prekročená kapacita vozidla.

Obrázok 1.1 znázorňuje príklad prípustného riešenia CVRP s $n = 7$ zákazníkmi a $p = 3$ vozidlami s rovnakou kapacitou $Q = 100$. Požiadavky zákazníkov sú zobrazené pri vrcholoch. Riešenie pozostáva z trás $r_1 = (0, 1, 3, 5, 0)$, $r_2 = (0, 7, 4, 0)$ a $r_3 = (0, 2, 6, 0)$ s cenami 106, 80 a 41. Cena tohto riešenia je 227.

CVRP je NP-t'ážky problém a je zovšeobecnením známeho problému obchodného cestujúceho – Traveling Salesman Problem (TSP), ktorý dostaneme, ak položíme $Q \geq \sum_{i=1}^n d_i$ a $p = p_{min} = 1$. Preto mnohé úspešné metódy riešenia CVRP vychádzajú z princípov riešenia TSP. Ako už bolo spomenuté, CVRP súvisí aj s BPP.



Obr. 1.1: Prípustné riešenie CVRP

1.3 Matematický model kapacitnej okružnej dopravnej úlohy

V literatúre možno nájsť niekoľko spôsobov modelovania CVRP. Ich podrobnú klasifikáciu publikovali napr. Laporte (2009) [53], Toth a Vigo (2014) [83].

Nasledujúci matematický model je založený na tzv. *tokovej formulácii* – vehicle flow formulation, ktorá pre každú hranu $(i, j) \in \vec{H}$ používa bivalentnú premennú x_{rij} , indikujúcu, či vozidlo $r, r \in \{1, 2, \dots, p\}$ v optimálnom riešení prechádza hranou (i, j) alebo nie, t. j.

$$x_{rij} = \begin{cases} 1, & \text{ak trasa } r \text{ v optimálnom riešení obsahuje hranu } (i, j), \\ 0, & \text{inak.} \end{cases} \quad (1.1)$$

CVRP formulujeme ako nasledujúcu úlohu bivalentného lineárneho programovania [45]:

Model CVRP 1

Minimalizujte

$$\sum_{r=1}^p \sum_{i=0}^n \sum_{\substack{j=0 \\ i \neq j}}^n c_{ij} x_{rij}, \quad (1.2)$$

za podmienok

$$\sum_{r=1}^p \sum_{\substack{i=0 \\ i \neq j}}^n x_{rij} = 1, \quad \forall j \in \{1, \dots, n\}, \quad (1.3)$$

$$\sum_{j=1}^n x_{r0j} = 1, \quad \forall r \in \{1, \dots, p\}, \quad (1.4)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^n x_{rij} = \sum_{i=0}^n x_{rji}, \quad \forall j \in \{0, \dots, n\}, r \in \{1, \dots, p\}, \quad (1.5)$$

$$\sum_{i=0}^n \sum_{\substack{j=1 \\ i \neq j}}^n d_j x_{rij} \leq Q, \quad \forall r \in \{1, \dots, p\}, \quad (1.6)$$

$$\sum_{r=1}^p \sum_{i \in S} \sum_{\substack{j \in S \\ i \neq j}} x_{rij} \leq |S| - 1, \quad \forall S \subseteq \{1, \dots, n\}, \quad (1.7)$$

$$x_{rij} \in \{0, 1\}, \quad \forall r \in \{1, \dots, p\}, i, j \in \{0, \dots, n\}, i \neq j. \quad (1.8)$$

Účelová funkcia (1.2) minimalizuje celkové prepravné náklady. Obmedzujúce podmienky (1.3) zaručujú, že každý zákazník bude obslužený práve jedným vozidlom. Podmienky toku (1.4) a (1.5) zaisťujú, že každé vozidlo opustí depo jediný raz a že počet vozidiel prichádzajúcich ku každému zákazníkovi a do depa je rovný počtu vozidiel odchádzajúcich. Kapacitné podmienky (1.6) garantujú, že na žiadnej trase súčet požiadaviek všetkých zákazníkov neprekročí kapacitu obslužného vozidla. Podmienky (1.7) sú anticyklické podmienky a zabezpečujú, že v riešení nebudú cykly, ktoré neprechádzajú depom. Posledné sú obligatórne podmienky (1.8) určujúce definičný obor premenných.

1.4 Všeobecná definícia robustnej optimalizácie

Pojem *robustný* má pri optimalizácii viacero významov. V praxi niekedy bývajú vstupné údaje neisté, nepresné alebo sa menia v čase, čo môže ovplyvniť optimálne riešenie nájdené pre aktuálne hodnoty parametrov tak, že sa stane suboptimálnym alebo dokonca neprípustným. *Robustné riešenie* je riešenie odolné voči odchýlkam vstupných dát.

Pri riešení optimalizačných problémov s neistými dátami sa používajú dva prístupy – stochastická a robustná optimalizácia. Dôležitým predpokladom pri *stochastickej optimalizácii*

(SO) je, že rozdelenie pravdepodobnosti neistých údajov musí byť známe. Stochastické programovanie zaviedol Dantzig (1955) [20], ktorý použil dve metódy na riešenie stochastických problémov: programovanie s pravdepodobnostnými obmedzeniami (chance-constrained programming) a stochastické programovanie s rekurziou (stochastic programming with recourse). Pri programovaní s pravdepodobnostnými obmedzeniami je prípustnosť riešenia v súlade s obmedzeniami zaručená s určitou pravdepodobnosťou. V stochastickom programovaní s rekurziou sú niektoré obmedzenia relaxované a zahrnuté do účelovej funkcie s predpokladom, že ak dôjde k porušeniu týchto podmienok v dôsledku náhodnej udalosti, budú prvotné rozhodnutia rekurzívne opravené. Stochastické modely sú silné, avšak majú dva hlavné nedostatky: rozdelenie pravdepodobnosti náhodných premenných musí byť známe a riešenie je prípustné len pre niektoré realizácie náhodných udalostí.

Robustná optimalizácia (RO) naopak nepredpokladá, že rozdelenie pravdepodobnosti neistých údajov je známe, ale namiesto toho vychádza z predpokladu, že neisté dáta patria do tzv. množiny neurčitostí [6]. Výsledkom RO je riešenie, ktoré je prípustné pre všetky realizácie náhodných udalostí v danej množine. Vo všeobecnosti, cieľom RO je optimalizovať najhorší prípad zo všetkých možných hodnôt náhodných premenných.

Všeobecná formulácia robustnej optimalizácie je:

Minimalizujte

$$f_0(\mathbf{x}),$$

za podmienok

$$f_i(\mathbf{x}, \mathbf{u}_i) \leq 0, \quad \forall \mathbf{u}_i \in \mathcal{U}_i, i = 1, \dots, m, \quad (1.9)$$

kde $\mathbf{x} \in R^n$ je vektor rozhodovacích premenných, $\mathbf{u}_i \in R^k$ sú *vektory scenárov* (scenario vectors) a $\mathcal{U}_i \subseteq R^k$ sú *množiny neurčitostí* (uncertainty sets). Cieľom (1.9) je nájsť riešenia \mathbf{x}^* s najmenšou cenou spomedzi všetkých tých riešení, ktoré sú prípustné pre všetky realizácie scenárov \mathbf{u}_i z množiny \mathcal{U}_i . Ak je \mathcal{U}_i jednoprvková, potom zodpovedajúca podmienka nemá žiadnu neurčitosť. Intuitívne, tento prístup ponúka určitú mieru ochrany prípustnosti riešenia pre problémy s parametrami, ktoré nie sú presne známe.

Robustný optimalizačný prístup má oproti stochastickému niekoľko výhod [79]. Po prvé, v mnohých prípadoch je ľahšie definovať množinu neurčitostí než odhadnúť rozdelenie prav-

depodobnosti. Po druhé, robustný prístup za určitých podmienok významne nezvyšuje zložitosť problému.

Podľa Kouvelis a Yu (1997) [50] sú pri RO dôležité nasledujúce rozhodnutia:

- 1) spôsob, akým sú modelované neisté dáta,
- 2) výber vhodného kritéria robustnej optimality – Robust Optimization Criteria (ROC),
- 3) voľba matematického modelu a metódy generovania robustných riešení.

Čo sa týka spôsobu modelovania neistých dát, najčastejšie sa reprezentujú pomocou konvexnej množiny (napr. mnohostena, kužeľa alebo elipsoidu) alebo určením hodnôt parametrov modelu vo forme intervalu, resp. diskrétneho scenára. Niektorí autori ako Kouvelis a Yu (1997) [50] pri výskume zistili, že robustné verzie niektorých polynomiálne riešiteľných kombinatorických optimalizačných problémov sú NP-t ťažké. Napríklad problém nájdenia najkratšej cesty v grafe je polynomiálny problém, kým jeho robustná verzia iba s dvomi scenármi pre ceny hrán je už NP-t ťažká. Bertsimas a Sim (2003) [10] navrhli obmedzenie počtu neistých parametrov, ktorým je povolené odchyliť sa od nominálnej hodnoty, na Γ – budget neurčitosti (budget of uncertainty). Tento prístup aplikovali na niekoľko problémov lineárneho alebo zmiešaného celočíselného programovania a ukázali, že pri riešení ich robustných verzií dôjde len k miernemu nárastu zložitosti. Konkrétne, úloha bivalentného lineárneho programovania s n neistými parametrami môže byť vyriešená riešením nanajviš $n + 1$ inštancií daného problému.

Výber vhodného kritéria robustnej optimality ako sú napr. min-max, min-max regret, min-max relative regret, α -robustnosť, bw -robustnosť, pw -robustnosť, atď. závisí hlavne od účelu optimalizácie [18]. Cieľom kritérií typu min-max (Von Neumann, 1928 [65]) je minimalizovať možné straty pri najhoršom možnom scenári. Duálny problém max-min zas spočíva v maximalizácii minimálneho zisku. Obe kritériá min-max aj max-min sa používajú vtedy, keď z najhoršieho scenára môžu vyplývať veľké škody. Ďalšie ROC je lexikografické min-max kritérium (Dresher, 1961 [24]), ktoré minimalizuje nielen najhorší prípad, ale aj druhý najhorší, tretí najhorší atď. Rozšírením lexikografického min-max kritéria je α -robustnosť (Kalai, 2012 [47]), ktoré sa používa v prípadoch, keď sú neisté údaje modelované použitím

diskrétnych scenárov. Parameter α určuje tolerančný prah (threshold) obmedzujúci stupeň konzervativizmu riešenia. Rovnako ako lexikografické min-max kritérium aj α -robustnosť zoradí riešenia vzhľadom na najhoršie scenáre. Definuje sa tzv. vektor ideálnych riešení, pričom ideálne riešenie je najlepšie riešenie získané pre jeden scenár a používa sa na výpočet odchýliek. Pre dané riešenie ω , ideálne riešenie ω^* a množinu scenárov S , sa vypočíta rozdiel medzi cenou ω a ω^* v každom scenári $k \in S$ a dostaneme maximálnu odchýlku. Napríklad pre problém s dvoma scenármi, ak sú ideálne riešenia pre všetky scenáre dané vektorom $cena(\omega^*) = (10, 8)$ a cena riešenia ω je daná vektorom $cena(\omega) = (15, 10)$, tak maximálna odchýlka medzi $cena(\omega^*)$ a $cena(\omega)$ je 5. Pri kritériu bw -robustnosť (Roy, 2010 [74]) nezáporná hodnota b udáva cieľ, v zmysle ceny, ktorý má byť dosiahnutý alebo vylepšený v najväčšom počte scenárov, kým w je garantovaná hodnota najhoršej ceny bez ohľadu na scenár ($w \leq b$). Toto kritérium sa používa iba vtedy, keď scenáre zodpovedajú diskrétnym hodnotám a počet scenárov je veľký. Takisto ako pre kritériá typu min-max, môže byť bw -robustnosť určená ako bw -absolútna robustnosť, bw -absolútna odchýlka a bw -relatívna odchýlka. Kritérium pw -robustnosť (Gabrel, 2013 [28]) je rozšírením kritéria bw -robustnosť. Riešenie sa nazýva robustným vtedy, keď je pre všetky scenáre zaručená hodnota w a nedosiahne hodnotu b v p percentách scenárov.

1.5 Robustná lineárna optimalizácia

Ako sme už spomenuli, CVRP patrí medzi lineárne optimalizačné problémy. Pre robustnú verziu lineárneho optimalizačného problému je formulácia nasledovná [9]:

Minimalizujte

$$\mathbf{c}^\top \mathbf{x},$$

za podmienok

$$\mathbf{A} \mathbf{x} \leq \mathbf{b}, \quad \forall \mathbf{a}_1 \in \mathcal{U}_1, \dots, \mathbf{a}_m \in \mathcal{U}_m, \quad (1.10)$$

kde \mathbf{a}_i reprezentuje i -ty riadok v matici neurčitostí \mathbf{A} a obsahuje hodnoty z množiny $\mathcal{U}_i \subseteq \mathbb{R}^n$. Potom $\mathbf{a}_i^\top \mathbf{x} \leq b_i, \forall \mathbf{a}_i \in \mathcal{U}_i$ platí práve vtedy, keď

$$\max_{\mathbf{a}_i \in \mathcal{U}_i} \mathbf{a}_i^\top \mathbf{x} \leq b_i, \quad \forall i, \quad (1.11)$$

čo je podproblém, ktorý treba vyriešiť. Jeho štruktúra určuje zložitosť riešenia robustného optimalizačného problému.

Základná paradigma robustnej optimalizácie je založená na nasledujúcich troch predpokladoch [52]:

- Všetky rozhodovacie premenné $\mathbf{x} \in R^n$ reprezentujú rozhodnutia "tu a teraz", t. j. mali by nadobudnúť špecifickú numerickú hodnotu ako výsledok riešenia problému ešte predtým, než budú známe aktuálne dáta.
- Subjekt s rozhodovacou právomocou nesie plnú zodpovednosť za následky rozhodnutí jedine vtedy, ak aktuálne dáta sú z preddefinovanej množiny \mathcal{U} .
- Obmedzujúce podmienky problému sú "tvrdé", to znamená, že ten, kto rozhoduje, nesmie tolerovať porušenie podmienok, ak aktuálne dáta sú v preddefinovanej množine \mathcal{U} .

Kapitola 2

SÚČASNÝ STAV RIEŠENIA PROBLEMATIKY

V roku 1959 Dantzig a Ramser vo svojej práci "The truck dispatching problem-[21] prvýkrát uviedli matematickú formuláciu a algoritmické riešenie úlohy, ktorej cieľom bolo minimalizovať celkovú dĺžku trasy homogénnych vozidiel pri zásobovaní skupiny čerpacích staníc palivom z centrálného zdroja. O päť rokov neskôr, v roku 1964, Clarke a Wright [17] zovšeobecnil túto úlohu ako optimalizačný problém, ktorý sa stal známy ako *kapacitná okružná dopravná úloha* (CVRP) a navrhli prvú efektívnu heuristiku na jej riešenie. Pre širokú škálu praktických aplikácií sa CVRP stala jedným z najviac študovaných kombinatorických optimalizačných problémov, o čom svedčí aj veľké množstvo vedeckých publikácií. Autori predstavili niekoľko rôznych exaktných a heuristických metód riešenia VRP. Všeobecný prehľad výskumu môže čitateľ nájsť napr. v práci [53] alebo [83], z ktorej sme čerpali v tejto kapitole.

2.1 Exaktné metódy

Hoci CVRP patrí medzi NP-t'ážké problémy, existuje značné množstvo prác venovaných vývoju postupov na nájdenie exaktného riešenia. Exaktné algoritmy sú väčšinou založené na formulácii problému ako úlohy celočíselného lineárneho programovania (ILP). Úlohy ILP sa všeobecne riešia metódou *vetiev a hraníc* (Branch-and-Bound, BB) a metódou *vetiev a rezov* (Branch-and-Cut, BC), ktoré vychádzajú z tokovej formulácie CVRP. Začiatkom druhého

tisícročia bola najefektívnejšou exaktnou metódou metóda BC (Baldacci a kol., 2004 [4], Lysgaard a kol., 2004 [58]), ale i napriek jej sofistikovanosti, bolo možné nájsť optimálne riešenie inštancií s nanajvýš 50 zákazníkmi. V roku 2006 Fukasawa a kol. [27] predstavili exaktný algoritmus na riešenie CVRP *vetiev a rezov a oceňovania* (Branch-and-Cut-and-Price), ktorý ukázal, že metóda BC v kombinácii s metódou *generovania stĺpcov* (Column Generation) je omnoho efektívnejšia než každá z týchto techník samostatne (úspešne vyriešil inštancie až so 134 zákazníkmi). Odvtedy sa táto metóda stala základom v súčasnosti najúspešnejších exaktných algoritmov. Jeden z najnovších návrhol Pecin a kol.(2014) [70]. Jeho algoritmus dokázal vyriešiť všetky klasické inštancie z literatúry až s 200 zákazníkmi.

2.2 Heuristické metódy

Nakoľko sa v reálnej praxi vyskytujú inštancie s omnoho väčším počtom zákazníkov, na ich riešenie sú vhodnejšie heuristické metódy, ktoré sú schopné nájsť v rozumnom čase vysoko kvalitné približné riešenie. Heuristické metódy pre CVRP môžeme rozdeliť do dvoch hlavných kategórií: konštruktívne a zlepšujúce heuristiky a metaheuristiky.

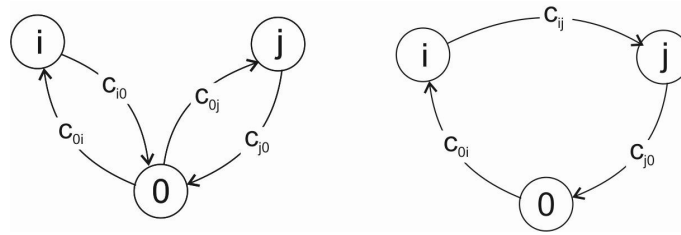
2.2.1 Konštruktívne heuristiky

Cieľom konštruktívnych heuristík je postupne vybudovať prípustné riešenie. Typické pre ne je, že pracujú pažravým a iteratívnym spôsobom.

Clarke-Wrightova metóda – Clarke-Wright's saving method (CWS) (Clarke a Wright, 1964 [17]) je asi najpopulárnejšia konštruktívna heuristika pre CVRP, a to pre svoju koncepcnú jednoduchosť a pomerne dobré výsledky. Princíp tejto metódy spočíva v tom, že sa snaží prepojiť vrcholy i, j tak, aby sa ich spojením čo najviac ušetrilo (obrázok 2.1). Spojením dvoch trás $(0, \dots, i, 0)$ a $(0, j, \dots, 0)$ do jednej $(0, \dots, i, j, \dots, 0)$ sa dosiahne úspora

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}.$$

Najprv sa vypočítajú úspory s_{ij} pre všetky $i, j = 1, \dots, n, i \neq j$ a zoradia sa zostupne. Algoritmus začína z počiatočného riešenia, v ktorom je každý zákazník obslužený iným vozidlom,



Obr. 2.1: Vľavo sú zákazníci i a j obslužení dvomi vozidlami, pokým vpravo len jedným

t. j. obsahuje spätočné trasy $(0, i, 0)$ pre $i = 1, \dots, n$. Potom postupuje od najväčších s_{ij} a prepája trasy hranou (i, j) , pod podmienkou, že nová trasa bude prípustná (požiadavky na trase neprekročia kapacitu vozidla). CWS má dve verzie – sekvenčnú a paralelnú. Prvá vytvára v danom čase jednu trasu (Mole a Jameson, 1976 [64]), kým pri druhej prebieha súčasne konštrukcia viacerých trás (Christofides a kol., 1979 [44]).

Dvojfázové metódy rozložia problém na dva podproblémy: rozdelenie vrcholov do skupín tak aby nebola prekročená kapacita vozidiel – *clustering* a vytvorenie okružných trás pre zákazníkov v jednotlivých skupinách – *routing*. Podľa poradia fáz potom rozlišujeme, tzv. *cluster-first route-second* (Gillett a Miller, 1974 [32]), resp. *route-first cluster-second* (Beasley, 1983 [5], Prins a kol., 2014 [72]).

2.2.2 Zlepšujúce heuristiky

Zlepšujúce heuristiky, väčšinou označované ako *local search*, prehľadáujú okolie východiskového riešenia x , t. j. aplikujú jednoduché modifikácie na x , pričom zachovávajú prípustnosť riešenia. Tieto modifikácie môžu byť vykonané buď v každej trase samostatne (*intra-route*) alebo pre každý pár rôznych trás (*inter-route*). Cieľom je nájsť lepšie riešenie x' , ktoré nahradí x . Používajú sa dve známe stratégie na výber riešenia x' z okolia:

- *prvé zlepšenie* (first improvement) – prehľadávanie okolia skončí akonáhle nájdeme prvé zlepšenie riešenia,
- *najlepšie zlepšenie* (best improvement) – prehľadá sa celé okolie riešenia a vyberie sa najlepšie riešenie.

Proces prehľadávanie okolia sa zastaví, keď sa riešenie už nedá týmto spôsobom vylepšiť.

V literatúre sa nachádza množstvo algoritmov prehl'adávaní okolia, najznámejšie sú:

2-opt a 2-opt* – odstránenie niektorých dvoch hrán z riešenia a ich následné nahradenie inými dvomi hranami v rámci jednej trasy (2-opt), resp. odstránenie niektorých dvoch hrán z rôznych trás a ich prepojenie iným spôsobom (2-opt*). Rovnako pracuje aj *k-opt*, ale nahradenie hrán sa s rastúcim *k* stáva čoraz náročnejšou na čas. Lin a Kernighan (1973) [56] implementovali *k-opt* algoritmus (Lin-Kernighanov algoritmus), v ktorom sa *k* dynamicky mení počas prehl'adávaní. Je síce pomalší než obyčajný 2-opt algoritmus, dosahuje však oveľa lepšie výsledky.

Relocate – presunutie zákazníka vnútri trasy, resp. z aktuálnej trasy do inej.

Swap – výmena dvoch zákazníkov v jednej trase alebo medzi dvomi trasami.

2.2.3 Metaheuristiky

V poslednom období sa čoraz viac stávajú populárnymi heuristickými metódami metaheuristiky, pretože vo všeobecnosti dokážu nájsť lepšie riešenie než klasické heuristiky, hoci s väčšou časovou a implementačnou náročnosťou. Metaheuristiky môžeme rozdeliť na *metaheuristiky lokálneho prehl'adávaní* (local search methods) a *metaheuristiky založené na populácii* (population-based heuristics). Množstvo metaheuristických metód prezentuje práca [31]. V tejto časti uvádzame stručný opis niekoľkých známych metaheuristik.

Metaheuristiky lokálneho prehl'adávaní

Metaheuristiky lokálneho prehl'adávaní štartujú z východiskového riešenia x_1 a v každom kroku (iterácii) sa posunú od riešenia x_k k riešeniu x_{k+1} v jeho okolí $N(x_k)$. Ak hodnota účelovej funkcie $f(x)$ znamená cenu riešenia x , potom $f(x_{k+1})$ nemusí byť nutne menšia než $f(x_k)$. Preto je nutné zabrániť zacykleniu iterovania, t. j. uviaznutiu v lokálnom minime.

Simulované žihanie – Simulated Annealing (SA). Metaheuristika SA bola inšpirovaná technikou z metalurgie, pri ktorej sa kov zohriaty na určitú teplotu pozvoľna ochladzuje, čím sa zvyšuje jeho pevnosť a odolnosť. Atómy sa pri vyššej teplote môžu v látke pohybovať a hľadať vhodné miesto s nižšou energiou. S klesajúcou teplotou sa energia atómov znižuje a atómy sa ukladajú na lepšie pozície. Na konci procesu je veľká šanca, že sa dosiahne stav

s menšou energiou. V algoritme SA je analógiou pre teplotu kovu parameter T a energiu atómov reprezentuje účelová funkcia $f(x)$. Uviaznutiu v lokálnom minime sa zabráni tým, že sa s určitou pravdepodobnosťou prijme aj horšie riešenie, t. j. ak pre nájdené riešenie x z $N(x_k)$ platí $f(x) \leq f(x_k)$, tak $x_{k+1} = x$. Inak

$$x_{k+1} = \begin{cases} x, & \text{s pravdepodobnosťou } p_k, \\ x_k, & \text{s pravdepodobnosťou } 1 - p_k, \end{cases} \quad (2.1)$$

kde p_k je zvyčajne klesajúca funkcia k a $f(x) - f(x_k)$ definovaná ako

$$p_k = \exp - \frac{f(x) - f(x_k)}{T_k},$$

pričom teplota T_k je klesajúca funkcia k . Túto heuristiku pri riešení VRP použil napríklad Osman (1993) [68].

Zakázané prehl'adávanie – Tabu Search (TS). Pri TS riešenie x_k prejde k najlepšiemu nezakázanému riešeniu x_{k+1} z $N(x_k)$. Aby nedošlo k uviaznutiu v lokálnom minime, po istú dobu sa pamätajú transformácie, ktoré vedú k nedávno preskúmaným riešeniam v *tabu zozname*. Ak tabu zoznam prekročí svoju maximálnu kapacitu, vylúči sa najstaršia transformácia. Tak sa tabu zoznam cyklicky obnovuje. Výpočet sa zastaví po uplynutí vopred stanoveného času alebo po vykonaní určitého počtu iterácií bez zlepšenia najlepšieho riešenia. Jeden z najnovších algoritmov pre VRP založených na TS, ktorý dáva dobré výsledky opísali Zachariadis a Kiranoudis (2010) [87].

Metaheuristiky založené na populácii

Tieto metódy sú inšpirované prírodou, napr. procesom evolúcie alebo sociálnym správaním spoločenských druhov akými sú mravce, včely a podobne. Pracuje sa pri nich s populáciou riešení, pričom sa riešenia môžu medzi sebou kombinovať, aby tak vznikli nové lepšie riešenia.

Genetický algoritmus – Genetic Algorithm (GA) simuluje proces evolúcie, ktorá sa riadi Darwinovým princípom prirodzeného výberu. GA pracuje s populáciou jedincov – *chromozómov*. Mierou kvality jedinca je hodnota *fitness*. Prvá populácia je vygenerovaná náhodne a následne sa vyvíja z generácie na generáciu aplikovaním operátorov výberu, genetickej mutácie

a kríženia jedincov, pričom kvalitnejší jedinec má väčšiu šancu prežiť a stať sa rodičom. Tieto kroky sa opakujú s cieľom zlepšovania jedincov v populácii. Cyklický proces obnovy populácie sa skončí po splnení vopred stanovených podmienok (napr. po dosiahnutí maximálneho počtu iterácií bez zlepšenia a pod.). V ideálnom prípade by výsledná populácia mala obsahovať jedinca, ktorého kvalita je globálnym optimom funkcie fitness. Úspešné implementácie GA na riešenie VRP predstavili napr. Prins (2004) [71], Mester a Bräysy (2007) [62].

Optimalizácia kolóniou mravcov – Ant Colony Optimisation (ACO) napodobňuje kooperatívne správanie mravcov pri hľadaní potravy. Mravce medzi sebou komunikujú zanechávaním pachovej stopy *feromónov* a tým označujú cestu k potrave. Samostatný mravec sa pohybuje náhodne a ak zacíti feromónovú stopu, nasleduje ju s určitou pravdepodobnosťou a posilní ju vlastným feromónom. Intenzita feromónu časom slabne. Algoritmus ACO (Dorigo, 2004 [22]) pracuje s populáciou mravcov, ktoré cyklicky generujú riešenia – *mravčie cesty*, pričom pravdepodobnosť použitia hrany (i, j) v ceste je

$$p_{ij} = \frac{\tau_{ij}^{\alpha} \left(\frac{1}{c_{ij}}\right)^{\beta}}{\sum_{k \in T} \tau_{ik}^{\alpha} \left(\frac{1}{c_{ik}}\right)^{\beta}},$$

kde τ_{ij} je intenzita feromónu na hrane (i, j) , c_{ij} je vzdialenosť vrcholov i, j , T množina nenavštievených vrcholov a α, β sú parametre, ktorými sa nastavuje vplyv feromónov a vzdialenosti. Po vytvorení mravčích ciest sa aktualizuje intenzita feromónu na hrane. Proces generovania mravčích ciest a aktualizácie intenzity feromónov sa opakuje, kým nie je splnená podmienka ukončenia. Jednu z najúspešnejších implementácií ACO algoritmu navrhli Riemann, Doerner a Hartl (2004) [73].

2.3 CVRP s neistými parametrami

Dvomi hlavnými typmi CVRP s neistými parametrami sú stochastická CVRP (Stochastic Vehicle Routing Problem, SVRP) a robustná CVRP (Robust Vehicle Routing Problem, RVRP). V tejto práci sa sústreďíme hlavne na RVRP, preto sa v časti SVRP obmedzíme len na vymenovanie niekoľkých najnovších prác venovaných tejto problematike.

2.3.1 Stochastická CVRP

Dobrym úvodom do problematiky SVRP sú práce [30] a [11], ktoré poskytujú prehľad formulácií SVRP. V probléme VRP sa najčastejšie objavujú tri stochastické parametre:

- *stochastické požiadavky* (stochastic demands) – množstvo komodít, ktoré majú byť odvezené resp. dovezené k zákazníkom je náhodné,
- *stochastickí zákazníci* (stochastic customers) – zákazníci buď majú alebo nemajú záujem o službu s určitou pravdepodobnosťou,
- *stochastické časy* (stochastic travel times)- doby obsluhy a časy jazd sa môžu náhodne meniť.

K výskumu VRP so stochastickými požiadavkami významne prispeli napr. Bertsimas (1992) [12], Laporte a kol. (2002) [54], Christiansen a Lysgaard (2007) [40], Secomandi a Margot (2009) [75], Sun (2014) [78]. Problému VRP so stochastickými zákazníkmi sa venovali Bertsimas (1988) [8], Waters (1989) [86]. VRP so stochastickými zákazníkmi a stochastickými požiadavkami formalizoval Bertsimas (1992) [12] a prvý exaktný algoritmus na jej riešenie navrhli Gendreau, Laporte a Séguin (1995) [29]. Sungur a kol. (2010) [81] prezentoval model a algoritmus na riešenie VRP so stochastickými zákazníkmi a stochastickými časmi obsluhy. Riešením VRP so stochastickými časmi sa zaoberali Kenyon a Morton (2003) [49], Li, Tian a Leung (2010) [57].

2.3.2 Robustná CVRP

V prípade RVRP je najviac študovaný prípad s neistými požiadavkami. Sungur a kol. (2008) [80] boli prví, ktorí skúmali tento problém. V ich práci sú vektory požiadaviek konštruované ako odchýlka od očakávanej hodnoty, ktorá prislúcha rôznym ohraničeným množinám. Robustná formulácia zohľadňuje neisté požiadavky v podmienkach, podobne ako zaviedli Ben-Tal a Nemirovski (1998) [6]. Autori pri riešení problému použili open source založený na metóde branch-and-cut a experimentovali s tromi rozdielnymi množinami inštancií z literatúry v rozsahu 15 až 100 zákazníkov. Ukázali, že robustné riešenia dokážu predísť ne-

uspokojeným požiadavkám, pričom vzniknú len veľmi malé dodatočné náklady v porovnaní s deterministickou verziou.

Moghaddam a kol. (2012) [63] sa s neistou distribúciou požiadaviek zákazníkov vysporiadali pomocou robustnej optimalizácie tak, že percento poruchy od nominálnej hodnoty požiadaviek definovali v podmienkach, pričom cieľom bolo minimalizovať cestovné náklady.

Inú štúdiu RVRP s neistými požiadavkami prezentovali Gounaris a kol. (2013) [33]. Autori vyvinuli robustné kapacitné nerovnosti (Robust Rounded Capacity Inequalities, RCI). V modeloch sú požiadavky zákazníkov ako náhodné premenné na pravých stranách podmienok a určujú minimálnu cenu dodacieho plánu, ktorý je prípustný pre všetky realizácie predpokladaných požiadaviek. Výsledky ich experimentov s 90 inštanciami s 15 až 135 zákazníkmi ukázali, že najlepšia robustná formulácia, toková formulácia s dvoma indexmi sa môže ešte viac zlepšiť pri použití RCI.

V novšej práci Gounaris a kol. (2014) [34] uviedli robustnú formuláciu VRP s neistými požiadavkami a implementovali metaheuristiku s použitím dvoch tried množín neurčitosti. Pri počítačových experimentoch s inštanciami do 483 zákazníkov a 38 vozidiel našli nové najlepšie riešenia pre všetkých 123 inštancií.

Výskumu VRP s neistými časmi jász modelovaných pomocou robustných scenárov sa venovali Han a kol. (2013) [41]. Obmedzili počet neistých parametrov, ktorým je dovolené odchyliť sa od nominálnej hodnoty (Bertsimas a Sim, 2003) [10]. Potom pre každú realizáciu (scenár) určili robustnú trasu a minimalizovali najhorší prípad zo všetkých scenárov. Využili dvojstupňové rekurzívne stochastické programovanie riešené metódou BB. Výsledky ukazujú, že takýto prístup získa dobré riešenia, keď penalizácia v účelovej funkcii je malá.

Práce autorov Toklu a kol. (2013) [82] sú zamerané na VRP s neistými prepravnými cenami modelovanými ako intervaly. Na minimalizáciu celkovej ceny prepravy použili algoritmus ACO, kde poruchy sú zohľadnené v koeficientoch účelovej funkcie smerom k horným hraniciam intervalov.

RVRP s neistými časmi jász skúmali Solano-Charis a kol. (2015) [77]. Množinu cien hrán nahradili množinou diskretných scenárov s cieľom nájsť množinu trás s použitím lexikografického min-max kritéria.

Ordóñez (2010) [67] sa vo svojej práci zamerail na VRP s viacerými neistými parametrami

– s neistými požiadavkami, časmi jazd, cenami a zákazníkmi. Autor popisuje rôzne robustné modely v závislosti od pôvodu neurčitosti, formulácie VRP a korelácie medzi neurčitými koeficientami. Neisté prepravné ceny sú zavedené v účelovej funkcii a neisté požiadavky a časy jazd v podmienkach. Neisté údaje sú modelované ako konvexné a ohraničené množiny neurčitostí a riešenie problému je založené na optimalizačnom prístupe podľa Ben-Tal a Nemirovského (1998) [6].

Riešenie VRP s neistými časmi jazd a požiadaviek je predmetom práce autorov Lee a kol. (2012) [55], pri ktorom na obmedzenie súčtu odchýliek časov jazd a odchýlky požiadaviek od nominálnych hodnôt definovali budžet neurčitosti podľa Bertsimas a Sim (2003) [10]. Robustnosť riešenia sa dosiahne hľadáním prípustného riešenia pre každý čas jazdy a požiadavku z množín neurčitosti s minimálnym časom prepravy. Riešenie vychádza z formulácie založenej na rozklade množiny (set partitioning formulation) a používa metódu generovania stĺpcov. Autorom sa podarilo optimálne vyriešiť inštancie s 20 až 40 zákazníkmi.

Kapitola 3

ZVOLENÁ METODIKA PRÁCE A METÓDY SKÚMANIA

Vzhľadom na široký rozsah problematiky robustných distribučných úloh sme sa v tejto práci zamerali na robustnú kapacitnú dopravnú okružnú úlohu (RVRP) s neistými požiadavkami zákazníkov. Optimálnym riešením tohto problému je také riešenie, ktoré je prípustné pre všetky realizácie neistých požiadaviek.

Obtiažnosť riešenia RVRP, okrem veľkosti problému (počtu zákazníkov n), ovplyvňujú aj ďalšie faktory: neisté vstupné dáta, zvolená matematická formulácia a metóda riešenia. Preto je dôležité, v závislosti od cieľa a robustnej optimalizácie, zvoliť spôsob modelovania neistých dát, vybrať vhodný matematický model a metódu generovania robustných riešení a tiež určiť kritérium robustnej optimality.

3.1 Spôsob modelovania neistých dát

Neisté požiadavky môžu byť modelované rozličnými spôsobmi, napr. ako stochastické premenné so známym rozdelením pravdepodobnosti [12] alebo ako fuzzy premenné [25]. Pri robustnej optimalizácii sa však predpokladá, že rozdelenie pravdepodobnosti nie je známe a nedá sa ani odhadnúť. V našej práci sme použili metodológiu, ktorú zaviedli Ben-Tal a Nemirovski [6], [7] pre robustnú lineárnu, kvadratickú a konvexnú optimalizáciu a pre celočíselné programovanie ju rozšírili Bertsimas a Sim [10]. Táto metodológia predpokladá, že

neisté parametre patria danej ohraničenej množine neurčitostí. Napríklad, pre problém lineárneho programovania s neistým parametrom a je množina neurčitostí \mathcal{U}_a ohraničená mnohostenom, určeným sústavou podmienok vyjadrených lineárnymi nerovnicami:

$$\mathcal{U}_a = \{(1 + \xi_k) a_0 : |\xi_k| \leq \varepsilon\},$$

kde $0 \leq \varepsilon \leq 1$. To znamená, že realizácia $a_k = (1 + \xi_k) a_0$ neistého parametra v k -tom scenári je daná pomocou odchýlky od jeho nominálnej hodnoty a_0 , pričom ξ_k je náhodná premenná s rovnomerným rozdelením z intervalu $\langle -\varepsilon, \varepsilon \rangle$.

Neisté požiadavky zákazníkov d_i^k sme modelovali ako náhodné premenné,

$$d_i^k \in \langle d_i^0 - \varepsilon d_i^0, d_i^0 + \varepsilon d_i^0 \rangle \quad \forall i \in \{1, \dots, n\},$$

kde d_i^0 sú očakávané (nominálne) požiadavky a n je počet zákazníkov.

3.2 Matematický model

Existuje viacero rôznych matematických formulácií CVRP, z ktorých by sa dal odvodiť matematický model RVRP. Pri výbere je však potrebné zohľadniť obtiažnosť riešenia výsledného RVRP. V matematickej formulácii CVRP 1, ktorú sme uviedli v predchádzajúcej kapitole, počet nerovností v anticyklických podmienkach v (1.2) rastie exponenciálne s počtom zákazníkov. Preto sme navrhli nový, omnoho efektívnejší model CVRP 2, formulovaný ako úloha zmiešaného lineárneho programovania s polynomiálnym počtom anticyklických podmienok.

Okrem toho, v metodológii RO sa preferujú podmienky obsahujúce neisté parametre vo forme nerovností, nakoľko splniť rovnosť pre všetky hodnoty neistých parametrov je oveľa zložitejšie. Aj z tohto hľadiska je náš model CVRP 2 vhodnou formuláciou.

3.3 Metódy generovania robustných riešení

Podľa robustnej optimalizačnej metodológie, robustné riešenie RVRP je riešenie, ktoré je optimálne pre najhorší scenár z množiny \mathcal{U}_d . Takéto riešenie je *imúnne* voči neistým vstup-

ným dátam, pretože ak je prípustné pre najhorší prípad, tak potom je prípustné pre všetky možné prípady. Kľúčovým krokom pri riešení RVRP teda je identifikovať najhorší scenár.

3.3.1 Stratégie na nájdenie najhoršieho scenára

- *Stratégia maximálneho scenára* definuje maximálny scenár, v ktorom má každý zákazník najväčšiu požiadavku zo všetkých scenárov [80]. Následným vyriešením inštancie CVRP s maximálnym scenárom dostaneme robustné riešenie, ktoré síce môže mať väčšiu cenu, než optimálne riešenie CVRP s očakávaným (nominálnym) scenárom, ale na druhej strane predchádza neuspokojeným požiadavkám v prípade, že nastane iný scenár. Nevýhodou tejto stratégie je, že aj keď jednotlivé scenáre sú prípustné, maximálny scenár nemusí byť vždy prípustný. Z tohto dôvodu sme navrhli nasledujúcu stratégiu.
- *Stratégia najhoršieho prípustného scenára* vyberie u každého zákazníka do najhoršieho scenára čo najväčšiu požiadavku zo všetkých scenárov tak, aby výsledný scenár bol prípustný. Najhorší prípustný scenár získame riešením úlohy bivalentného lineárneho programovania. Riešenie inštancie CVRP s najhorším prípustným scenárom je vždy prípustné a v prípade, že sa najhorší prípustný scenár zhoduje s maximálnym scenárom, je aj robustné. Ak sa najhorší prípustný scenár nezhoduje s maximálnym, robustnosť riešenia nie je síce zaručená, ale neuspokojené požiadavky sú minimalizované.

Nájsť robustné riešenie znamená vyriešiť inštanciu CVRP s najhorším scenárom. Na riešenie CVRP sme modifikovali niektoré metódy známe z literatúry a okrem toho sme navrhli dve nové metódy - exaktnú iteratívnu metódu a lexikografickú metódu.

3.3.2 Metódy riešenia CVRP

- *Exaktná iteratívna metóda* je metódou lokálneho prehľadávania s najlepším zlepšením, ktorá iteratívne zlepšuje východiskové prípustné riešenie nahradením niektorých hrán inými, čo dosiahneme exaktným riešením jednoduchšieho problému.
- *Lexikografická metóda* je metóda, ktorá nájde optimálne riešenie CVRP, pričom bude využitá len najmenšia nutná kapacita vozidiel. Nevyužitá kapacita vozidiel predstavuje

rezervu pre prípad, že sa očakávané požiadavky náhodne zvýšia, výsledné riešenie je teda do istej miery robustné.

- *Paralelný mikrogenetický algoritmus* je metaheuristická metóda, ktorá je variantom genetického algoritmu s veľmi malou populáciou. Paralelný mikrogenetický algoritmus využíva benefity mikrogenetického algoritmu a paralelizácie výpočtov. Výsledkom je suboptimálne riešenie.

3.4 Kritérium robustnej optimality

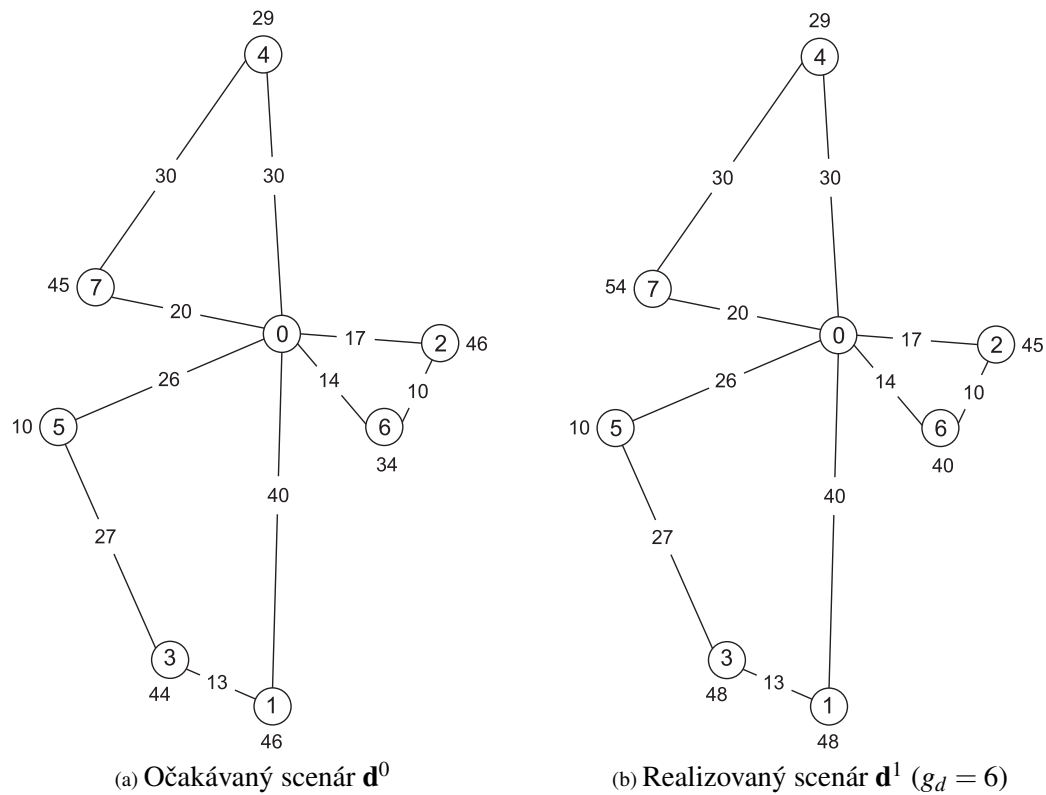
Prvoradým cieľom robustnej optimalizácie je nájsť riešenie, ktoré uspokojí všetky požiadavky zákazníkov bez ohľadu na to, ktorý scenár nastane. Takéto riešenie môže mať vyššiu cenu, než optimálne deterministické riešenie, čo je "daň" za robustnosť. Riešenia nájdené použitím jednotlivých stratégií sme posúdili z dvoch hľadísk: z hľadiska neuspokojených požiadaviek a z hľadiska navýšenia ceny riešenia [26].

3.4.1 Neuspokojené požiadavky

Ak by sme ignorovali premenlivosť požiadaviek a použili by sme deterministické optimálne riešenie, mohlo by dôjsť k tomu, že keď sa požiadavky zmenia, bude toto riešenie neprípustné. Napríklad, pre optimálne riešenie (obrázok 3.1) s trasami $r_1 = (0, 1, 3, 5, 0)$, $r_2 = (0, 7, 4, 0)$ a $r_3 = (0, 2, 6, 0)$, s vektorom očakávaných požiadaviek $\mathbf{d}^0 = (46, 46, 44, 29, 10, 34, 45)$ a kapacitou vozidiel $Q = 100$ by pri realizácii scenára $\mathbf{d}^1 = (48, 45, 48, 29, 10, 40, 54)$ bol súčet požiadaviek na trasách postupne 106, 83 a 85. Pretože na trase r_1 by bola prekročená kapacita vozidla, riešenie by bolo neprípustné.

Neuspokojené požiadavky sú požiadavky, ktoré na jednotlivých trasách prekročia kapacitu vozidla. Označme g_d neuspokojené požiadavky, ktoré vzniknú pri použití deterministického optimálneho riešenia a g_r neuspokojené požiadavky pri riešení získanom robustnou optimalizáciou. Ukazovateľ

$$\gamma = \frac{g_d - g_r}{\sum_{i=1}^n d_i^0} \quad (3.1)$$



Obr. 3.1: Optimálne deterministické riešenie CVRP

vyjadruje zníženie relatívneho množstva neuspokojených požiadaviek pri nájdenom riešení v porovnaní s optimálnym riešením, kde $\sum_{i=1}^n d_i^0$ je súčet očakávaných požiadaviek. Na základe tohto ukazovateľa môžeme porovnať riešenia získané rôznymi stratégiami, pričom väčšia hodnota γ znamená lepšie riešenie.

3.4.2 Zvýšenie ceny riešenia

Na porovnanie riešení nájdených rôznymi metódami z hľadiska ceny použijeme ukazovateľ ζ , ktorý predstavuje relatívne zvýšenie ceny z_r nájdeného riešenia voči cene z_d optimálneho deterministického riešenia:

$$\zeta = \frac{z_r - z_d}{z_d}. \quad (3.2)$$

Tento pomer nám dáva informáciu, o čo relatívne vyššie budú prepravné náklady, ak použijeme riešenie, ktoré predchádza neuspokojeným požiadavkám, namiesto deterministického optimálneho riešenia. Z toho vyplýva, že čím je hodnota ζ menšia, tým je riešenie lepšie.

3.5 Testovacie inštancie

Na testovanie a overovanie našich algoritmov a modelov sme využili klasické inštancie CVRP z repozitára dostupného na [90]. Databáza obsahuje pre každú inštanciu:

- počet zákazníkov n , kapacitu vozidiel Q , súradnice zákazníkov a ich požiadavky,
- najlepšie známe, resp. optimálne riešenie, ktoré je reprezentované ako množina okružných jász a tiež jeho cenu,
- ďalšie informácie, ako napr. zdroj, spôsob ich generovania a pod.

Inštancie sú zoskupené v sériách, z ktorých sme zvolili tieto:

- **Séria A** (Augerat [3]) - Vrcholy sú umiestnené na náhodných pozíciách v mriežke $(0,100) \times (0,100)$. Požiadavky zákazníkov sú náhodné hodnoty z rovnomerného rozdelenia na intervale $(1,30)$, avšak desatina z nich je vynásobená tromi.
- **Séria B** (Augerat [3]) - Podobne ako set A, ale zákazníci sú umiestnení v klastroch.
- **Séria E** (Christifides a Eilon [43]) - Informácie o ich generovaní nie sú známe.
- **Séria P** (Augerat [3]) - Dáta boli vytvorené modifikáciou inštancií zo sérií A, B a E zmenou ich kapacít a následne aj zmenou požadovaného počtu vozidiel (napr. inštancia P-n101-k4 pochádza z inštancie E-n101-k8)

Názvy inštancií obsahujú sériu, počet vrcholov (vrátane depa) a požadovaný počet vozidiel. Napr. názov E-n101-k8 označuje inštanciu zo série E so 100 zákazníkmi a 8 vozidlami. Testovacie inštancie sa tiež líšia polohou depa vzhľadom na umiestnenie zákazníkov.

Zvolené inštancie sme modifikovali na inštancie RVRP zohľadnením neistých požiadaviek zákazníkov. Pre každú inštanciu sme vygenerovali množinu neurčitostí \mathcal{U}_d s 5 scenármi s danou maximálnou odchýlkou ε tak, že v každom scenári požiadavky zákazníkov sú náhodné hodnoty z intervalu $\langle d_i^0 - \varepsilon d_i^0, d_i^0 + \varepsilon d_i^0 \rangle$, kde d_i^0 je očakávaná požiadavka zákazníka i z pôvodnej inštancie CVRP. Názvy nových inštancií, ktoré takto vznikli, sa skladajú z názvu originálnej inštancie a hodnoty ε použitej pri generovaní scenárov. Napr. názov A-n34-k5-e20 označuje inštanciu RVRP, ktorá vznikla modifikáciou inštancie A-n34-k5, pričom $\varepsilon = 0,2$ resp. 20%.

3.6 Softvérové nástroje

Všetky navrhnuté metódy sme implementovali v jazyku Python 3.4 [88] s využitím jeho modulov:

- NumPy - základný modul pre vedecké výpočty. Je určený pre prácu s rozsiahlymi viac-rozmernými poliami, lineárnu algebru, prácu s náhodnými číslami a pod.
- NetworkX - modul pre tvorbu, manipuláciu a skúmanie štruktúry a vlastností komplexných sietí.
- Matplotlib - knižnica na tvorbu 2D grafiky pre vedecké publikovanie, interaktívnu vizualizáciu, vývoj užívateľského rozhrania atď.
- mpi4py - modul pre distribuované paralelné programovanie na základe štandardu MPI (Message Passing Interface).

Úlohy lineárneho programovania sme riešili pomocou optimalizačného softvéru Gurobi 6.5 [89], ktorý dokáže využívať možnosti viacjadrových procesorov a používa implementácie najnovších algoritmov.

Kapitola 4

VÝSLEDKY PRÁCE

V tejto kapitole predstavíme postupne náš matematický model CVRP, vlastnú exaktnú a vylepšenú heuristickú metódu na riešenie na CVRP, nový lexikografický model CVRP a nakoniec tri robustné stratégie na riešenie RVRP s neistými požiadavkami zákazníkov. V každej časti uvedieme aj výsledky testovania navrhnutých modelov a metód a ich porovnanie s výsledkami z literatúry, ak sú známe.

4.1 Nový model CVRP

Počet nerovností v anticyklických podmienkach v (1.2) rastie exponenciálne s počtom zákazníkov. Nový model CVRP, ktorý predstavíme, je formulovaný ako úloha zmiešaného lineárneho programovania s polynomiálnym počtom anticyklických podmienok. Použijeme rozhodovacie premenné x_{ij} , kde

$$x_{ij} = \begin{cases} 1, & \text{ak optimálne riešenie obsahuje hranu } (i, j), \\ 0, & \text{inak.} \end{cases} \quad (4.1)$$

Aby sme zabezpečili spojitosť trasy a zabránili vzniku nežiadúcich cyklov, definujeme doplnkovú premennú y_i , $d_i \leq y_i \leq Q$ pre $i \in V - \{0\}$, ktorá vyjadruje využitú kapacitu (náklad) vozidla pri odchode od zákazníka i [48].

Pre každú dvojicu zákazníkov $i, j \in V - \{0\}, i \neq j$ vypočítame úsporu s_{ij} , ktorú získame ak

spojíme cykly $0 \rightarrow i \rightarrow 0$ a $0 \rightarrow j \rightarrow 0$ do jedného použitím hrany $i \rightarrow j$, t. j.

$$s_{ij} = c_{i0} + c_{0j} - c_{ij} \quad (4.2)$$

podobne, ako v Clarke-Wrightovej metóde (obrázok 2.1). Potom, namiesto minimalizácie celkovej ceny, budeme maximalizovať celkovú úsporu pri preprave.

Veta 4.1.1. *Maximalizovať celkovú úsporu pri preprave $\sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n s_{ij} x_{ij}$ je to isté ako minimalizovať celkové prepravné náklady $\sum_{i=0}^n \sum_{\substack{j=0 \\ i \neq j}}^n c_{ij} x_{ij}$.*

Dôkaz: Pre celkové prepravné náklady platí:

$$\sum_{i=0}^n \sum_{\substack{j=0 \\ i \neq j}}^n c_{ij} x_{ij} = \sum_{i=1}^n c_{i0} x_{i0} + \sum_{j=1}^n c_{0j} x_{0j} + \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n c_{ij} x_{ij},$$

čo môžeme podľa (4.2) zapísať ako

$$\sum_{i=0}^n \sum_{\substack{j=0 \\ i \neq j}}^n c_{ij} x_{ij} = \sum_{i=1}^n c_{i0} x_{i0} + \sum_{j=1}^n c_{0j} x_{0j} + \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n c_{i0} x_{ij} + \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n c_{0j} x_{ij} - \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n s_{ij} x_{ij},$$

Po jednoduchšej úprave dostaneme

$$\sum_{i=0}^n \sum_{\substack{j=0 \\ i \neq j}}^n c_{ij} x_{ij} = \sum_{i=1}^n \sum_{\substack{j=0 \\ i \neq j}}^n c_{i0} x_{ij} + \sum_{i=0}^n \sum_{\substack{j=1 \\ i \neq j}}^n c_{0j} x_{ij} - \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n s_{ij} x_{ij},$$

resp.

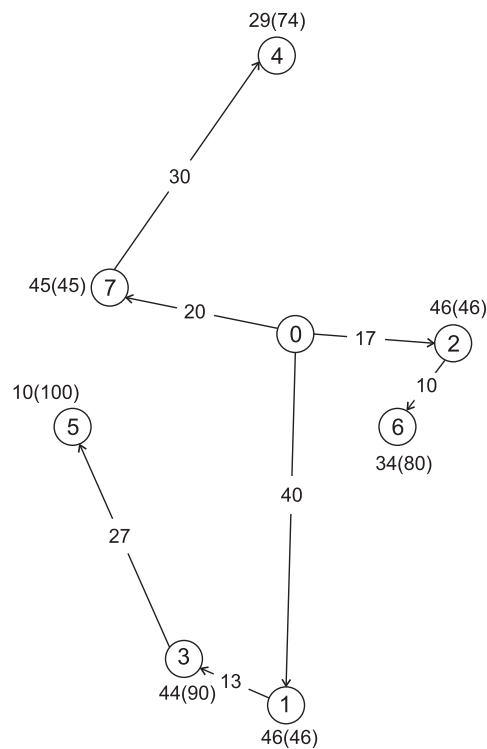
$$\sum_{i=0}^n \sum_{\substack{j=0 \\ i \neq j}}^n c_{ij} x_{ij} = \sum_{i=1}^n c_{i0} \sum_{\substack{j=0 \\ i \neq j}}^n x_{ij} + \sum_{j=1}^n c_{0j} \sum_{\substack{i=0 \\ i \neq j}}^n x_{ij} - \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n s_{ij} x_{ij}.$$

Pretože každý zákazník je obslužený práve jedným vozidlom, musí byť vstupný aj výstupný stupeň vrcholu z množiny $V - \{0\}$ rovný 1, t. j. $\sum_{\substack{i=0 \\ i \neq j}}^n x_{ij} = 1$ a $\sum_{\substack{j=0 \\ i \neq j}}^n x_{ij} = 1$. Z toho vyplýva, že

$$\sum_{i=0}^n \sum_{\substack{j=0 \\ i \neq j}}^n c_{ij} x_{ij} = \sum_{i=1}^n c_{i0} + \sum_{j=1}^n c_{0j} - \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n s_{ij} x_{ij},$$

alebo po úprave

$$\sum_{i=0}^n \sum_{\substack{j=0 \\ i \neq j}}^n c_{ij} x_{ij} + \sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n s_{ij} x_{ij} = \sum_{i=1}^n c_{i0} + \sum_{j=1}^n c_{0j}.$$



Obr. 4.1: Prípustné riešenie CVRP, trasy vozidiel sú nahradené orientovanými cestami

Z poslednej rovnosti je zrejmé, že súčet celkových prepravných nákladov a celkovej úspory pri preprave je konštantný. Preto platí, že maximalizáciou celkovej úspory pri preprave minimalizujeme celkové prepravné náklady. \square

Pre zjednodušenie modelovania každú prípustnú trasu $0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k \rightarrow 0$ nahradíme cestou z vrcholu 0 do vrcholu v_k , t. j. $0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k$. Obrázok 4.1 ilustruje príklad prípustného riešenia CVRP pre prípad zvozu tovaru. Pri každom vrchole i sú dve hodnoty – požiadavka d_i zákazníka i a náklad y_i vozidla pri odchode od zákazníka i (v zátvorkách). Napr. cesta $0 \rightarrow 2 \rightarrow 6$ reprezentuje trasu $0 \rightarrow 2 \rightarrow 6 \rightarrow 0$, zákazník 2 má požiadavku $d_2 = 46$ a požiadavka zákazníka 6 je $d_3 = 34$, vozidlo pri odchode od zákazníka 2 má využitú kapacitu $y_2 = 46$ a keď opúšťa zákazníka 6, je $y_6 = 80$.

4.1.1 Model CVRP 2

Maximalizujte

$$\sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n s_{ij} x_{ij}, \quad (4.3)$$

za podmienok

$$\sum_{j=1}^n x_{0j} = p, \quad (4.4)$$

$$\sum_{i=1}^n x_{i0} = 0, \quad (4.5)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^n x_{ij} = 1, \quad \forall j \in \{1, \dots, n\}, \quad (4.6)$$

$$\sum_{\substack{j=1 \\ i \neq j}}^n x_{ij} \leq 1, \quad \forall i \in \{1, \dots, n\}, \quad (4.7)$$

$$y_i + d_j x_{ij} - Q(1 - x_{ij}) \leq y_j, \quad \forall i, j \in \{1, \dots, n\}, i \neq j, \quad (4.8)$$

$$d_i \leq y_i \leq Q, \quad \forall i \in \{1, \dots, n\}, \quad (4.9)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in \{0, \dots, n\}, i \neq j. \quad (4.10)$$

V tejto formulácii účelová funkcia (4.3) maximalizuje celkovú úsporu pri preprave. Podmienky (4.4) a (4.5) stanovujú, že z depa vyjde presne p hrán, a do depa nevojde žiadna hrana, (4.6) a (4.7) určujú počet vozidiel prichádzajúcich a odchádzajúcich k/od zákazníkov. Skupina podmienok (4.8) predstavuje súčasne anticykliace podmienky a podmienky kontinuity toku, ktoré zaručujú, že riešenie neobsahuje cykly neincidentné s depom a tiež, že využitá kapacita vozidla je neklesajúca (v prípade zberu komodít) postupnosť v závislosti od požiadaviek zákazníkov prislúchajúcich tej istej trase. Podmienky (4.9) upresňujú dolné a horné ohraničenie y_i a (4.10) sú obligatórne podmienky.

4.1.2 Výsledky experimentov

Obidva modely sme implementovali v jazyku Python 3.4 [88] a riešili pomocou solvera Gurobi 6.5 [89] na PC s parametrami Intel Xeon 32 cores, 2,4 GHz, 256 GB RAM. Experimenty sme vykonali na 8 klasických inštanciách z [90]. Nakoľko je CVRP NP-t'ážky prob-

Tabuľka 4.1: Porovnanie časovej náročnosti modelov CVRP 1 a CVRP 2

Instance	Q	n	p	Opt	Čas (sec)		Pomer
					CVRP 1	CVRP 2	CVRP 1/CVRP 2
P-n16-k8	35	15	8	450	233,72	0,43	543,5
P-n19-k2	160	18	2	212	9,09	4,72	1,9
P-n20-k2	160	19	2	216	8,80	2,52	3,5
P-n21-k2	160	20	2	211	3,22	0,73	4,4
P-n22-k2	160	21	2	216	4,96	3,35	1,5
E-n13-k4	6000	12	4	247	0,83	0,29	2,9
E-n22-k4	6000	21	4	375	3435,13	25,16	136,5
E-n23-k3	4500	22	3	569	25,34	1,94	13,0

Q–kapacita vozidla, n–počet zákazníkov, p–počet vozidiel, Opt – cena optimálneho riešenia

lém, obomi modelmi sa nám podarilo v prijateľnom čase exaktne vyriešiť len inšcie s počtom zákazníkov 12 až 22 a počtom vozidiel 2 až 8. V tabuľka 4.1 zobrazuje porovnanie časovej náročnosti týchto dvoch modelov na nájdenie optimálneho riešenia. Každá inšcia bola vyriešená päťkrát a v tabuľke je uvedená priemerná hodnota času výpočtu.

4.1.3 Záver

Z tabuľky 4.1 je evidentné, že model CVRP 1 je časovo náročnejší, než model CVRP 2. Dá sa to vysvetliť tým, že počet nerovností v anticyklických podmienkach v CVRP 1 rastie exponenciálne s počtom zákazníkov. Preto sa nám pri praktickom použití tejto formulácie osvedčilo ich postupné pridávanie počas optimalizačného procesu [69]. Porovnanie ukázalo, že formulácia CVRP 2 je efektívnejšia než CVRP 1. Tento model sme prezentovali na medzinárodnej konferencii KOI 2016 v Osijeku v septembri 2016 [13].

4.2 Nová exaktná iteratívna metóda na riešenie CVRP

Teraz opíšeme našu exaktnú metódu na riešenie CVRP [15], ktorej hlavná myšlienka spočíva v iteratívnom zlepšovaní východiskového prípustného riešenia S nahradením niektorých hrán inými, čo dosiahneme exaktným riešením jednoduchšieho problému pomocou modelu CVRP 2. Túto metódu môžeme teda zaradiť medzi metódy lokálneho prehľadávania s najlepším zlepšením.

4.2.1 Exaktná iteratívna metóda CVRP 3

Naša metóda je postavená na koncepte *k-optimálnosti* [56]:

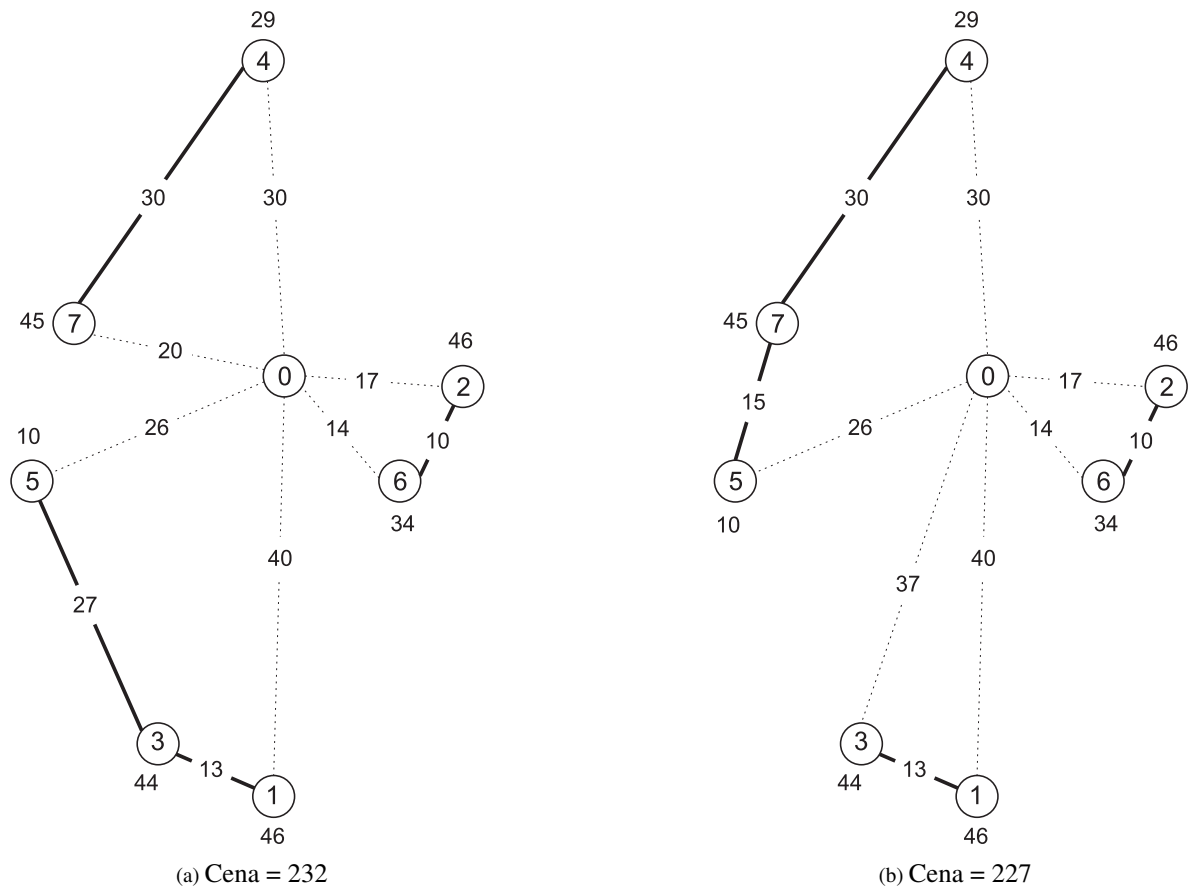
Definícia 4.2.1. Riešenie je *k-optimálne*, ak sa nedá zlepšiť výmenou žiadnych jeho k hrán inými k hranami.

Z tejto definície vyplýva, že každé *k-optimálne* riešenie je tiež *k'-optimálne* pre $1 \leq k' \leq k$. Takisto je zrejmé, že riešenie, ktoré má m hrán, je optimálne práve vtedy, keď je *m-optimálne*.

Nech S je prípustné riešenie CVRP s n zákazníkmi a p vozidlami, t.j. toto riešenie obsahuje $n + p$ hrán. Označme E množinu hrán riešenia S neincidentných s depom a počet jej prvkov m , $m = |E| = n + p$. Množinu ostatných hrán neincidentných s depom označme E' .

Metóda CVRP 3 v každej iterácii hľadá dve množiny hrán $A = \{a_1, a_2, \dots, a_k\}, A \subseteq E$ a $B = \{b_1, b_2, \dots, b_k\}, B \subseteq E'$ také, aby sme po odstránení hrán množiny A z množiny E a ich následnom nahradení hranami množiny B získali čo najlepšie riešenie (riešenie s čo najlepšou hodnotou účelovej funkcie). Takúto výmenu hrán voláme *k-optimálna výmena*. Obrázok 4.2 zobrazuje 1-optimálnu výmenu v riešení CVRP s počtom zákazníkov $n = 7$ a počtom vozidiel $p = 3$ s kapacitou $Q = 100$, pri ktorej bola hrana $(7, 5)$ nahradená hranou $(3, 5)$, pričom sa cena riešenia zlepšila z 232 na 227.

Ak je hľadanie úspešné, aktuálne riešenie sa nahradí novým nájdeným riešením. Tento proces sa opakuje dovtedy, kým nedosiahneme riešenie, ktoré sa už nedá zlepšiť výmenou žiadnych k hrán. V tom prípade platí, že aktuálne riešenie je podľa definície 4.2.1 *k-optimálne*. Počet vymieňaných hrán k , resp. počet hrán z , ktoré sa ponechajú, sa dynamicky mení podľa nasledovného algoritmu:



Obr. 4.2: 1-optimálna výmena

$$A = \{(7,5)\}, B = \{(3,5)\}$$

Algoritmus CVRP 3

Krok 1: Pomocou rýchlej heuristickej metódy nájdí počiatočné prípustné riešenie S_0 .

Polož $S = S_0$.

Krok 2: Vytvor zoznam E hrán prislúchajúcich riešeniu S , ktoré nie sú incidentné s depom.

Nech $m = |E|$.

Krok 3: Polož hodnoty $t = 1$, $m_1 = m - \delta$ a $m_2 = m - 1$, kde δ je vopred definovaná celočíselná hodnota $1 < \delta < m$.

Krok 4: Pomocou CVRP 2 nájdí optimálne riešenie S_t také, aby množina E_t hrán prislúchajúcich riešeniu S_t , ktoré nie sú incidentné s depom obsahovala minimálne m_1 a maximálne m_2

hrán z E , t.j. do modelu CVRP 2 pridáme podmienky:

$$\sum_{(i,j) \in E} x_{ij} = z, \quad (4.11)$$

$$m_1 \leq z \leq m_2, \quad (4.12)$$

kde celočíselná premenná z určuje, koľko hrán z E sa nachádza v množine E_t .

Krok 5: Ak je riešenie S_t lepšie než riešenie S , tak polož $S = S_t$ a pokračuj krokom 2.

Inak polož hodnoty $t = t + 1$, $m_1 = m_1 - \delta$ a $m_2 = m_2 - \delta$.

Krok 6: Ak $m_1 < 0$, polož $m_1 = 0$. Ak $m_2 \geq 0$, choď na krok 4, inak Stop.

Algoritmus CVRP 3 mení počet hrán vymieňaných hrán počnúc $k = 1$ a končiac $k = m$, čiže konečné riešenie je m -optimálne. Preto je táto metóda exaktná.

Ilustračný príklad: Majme riešenie, v ktorom $|E| = 11$. Voľbou parametra δ určíme, o koľko hrán množiny E aktuálneho riešenia menej sa má v nasledujúcej iterácii ponechať, resp. o koľko viac hrán sa má zameniť. Zvoľme $\delta = 3$, potom na začiatku $m_1 = 8$ a $m_2 = 10$, t.j. $8 \leq z \leq 10$ a $1 \leq k \leq 3$ alebo inými slovami ponechá sa 8 až 10 hrán a 1 až 3 hrany sa vymenia. Keď sa už riešenie nedá týmto spôsobom vylepšiť, dôjde k zmene počtu hrán, ktoré sa ponechajú a ktoré sa nahradia, na $5 \leq z \leq 7$ a $4 \leq k \leq 6$. V ďalšej iterácii to bude $2 \leq z \leq 4$ a $7 \leq k \leq 9$ a nakoniec $0 \leq z \leq 1$ a $10 \leq k \leq 11$. V poslednej fáze sa algoritmus CVRP 3 pokúsi zlepšiť riešenie výmenou všetkých 11 hrán a ak hľadanie bude neúspešné, tak skončí. Z toho vyplýva, že výsledné riešenie je 11-optimálne, a tým pádom je optimálne.

4.2.2 Výsledky experimentov

Algoritmus sme implementovali v jazyku Python 3.4 [88] a riešili pomocou solvera Gurobi 6.5 [89] na PC s parametrami Intel Xeon 32 cores, 2,4 GHz, 256 GB RAM. Experimenty sme vykonali na 7 klasických inštanciách z [90], pričom sme našli optimálne riešenie každej z nich. Na nájdenie počiatočného prípustného riešenia sme použili Clarke-Wrightovu metódu [17]. Každú inštanciu sme riešili pre rôzne hodnoty δ , výsledky sú v tabuľke 4.2.

Tabuľka 4.2: Porovnanie doby výpočtu pre rôzne hodnoty δ

Inštancia	n	Q	p	Štart	Opt	Čas (s)		
						$\delta = 2$	$\delta = 3$	$\delta = 4$
A-n32-k5	31	100	5	901	784	9	6	7
A-n34-k5	32	100	5	886	778	335	136	50
A-n36-k5	35	100	5	870	799	12 048	3 458	1847
A-n38-k5	37	100	5	828	730	81	76	134
A-n44-k6	43	100	6	1100	937	718	748	302
A-n53-k7	52	100	7	1167	1010	1165	5608	18 134
A-n55-k9	54	100	9	1291	1073	37 884	4 964	165 540

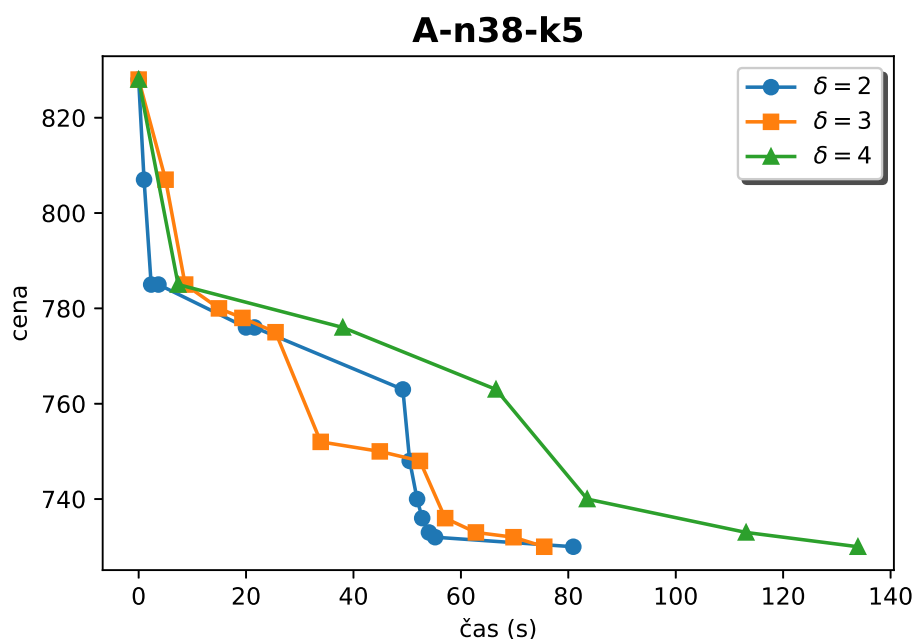
n – počet zákazníkov, Q – kapacita vozidla, p – počet vozidiel,

Štart – cena počiatočného riešenia, Opt – cena optimálneho riešenia

Obrázok 4.3 ilustruje vývoj riešenia inštancie A-n38-k5 s 37 zákazníkmi a 5 vozidlami, z počiatočnej ceny 828 až po optimálnu 730. Ilustrácia bola vytvorená použitím knižnice Matplotlib 3.0 [91].

4.2.3 Záver

Z našich experimentov vyplynulo, že nie je možné určiť, ktorá hodnota parametra δ je najvhodnejšia pre danú inštanciu. Tento fakt nás priviedol k myšlienke paralelizovať výpočet vzhľadom na parameter δ . Naša metóda sa javí ako slušná, nakoľko je schopná v prijateľnom čase exaktne vyriešiť aj inštancie s relatívne väčším počtom zákazníkov. Prípadne, ak by bol výpočet veľmi zdĺhavý, je možné ho prerušiť, pričom získame k -optimálne riešenie (posledné riešenie v CVRP 3).



Obr. 4.3: Zlepšovanie riešenia v čase pre rôzne hodnoty δ

4.3 Riešenie CVRP pomocou paralelného mikrogenetického algoritmu

Mikrogenetický algoritmus (μ GA) je verzia genetického algoritmu (GA), ktorý hľadá najlepšie prípustné (suboptimálne) riešenie problému na princípe darwinovskej evolúcie. Základný koncept GA po prvýkrát prezentoval v roku 1975 John Holland [42]. Ako sme už spomenuli, GA pracuje ako iteratívna procedúra s populáciou jedincov, z ktorých každý reprezentuje riešenie daného optimalizačného problému. Mierou kvality jedinca je hodnota fitness zodpovedajúca hodnote účelovej funkcie. Jedinci s lepšou fitness majú väčšiu šancu prežiť a reprodukovať sa. Pri reprodukcii (krížení) vzniknú noví jedinci – potomkovia, ktorí zdedia niektoré črty od oboch svojich rodičov a sformujú novú generáciu. Opakovaním tohto procesu sa priemerná fitness populácie, prípadne aj najlepšia fitness, z generácie na generáciu zlepšuje.

Na rozdiel od klasického GA, kde je populácia zložená z veľkého počtu jedincov (30 – 200), μ GA používa mikropopuláciu s 3 – 5 jedincami (Goldberg, 1989 [35]). Samozrejme, malá populácia vývojom rýchlo stratí svoju rôznorodosť (*diverzitu*), t.j. jedinci sa po krátkom čase začnú navzájom podobať a vývoj stagnuje. Môže to znamenať, že došlo k *predčasnej*

konvergencii, čiže riešenie uviazlo v lokálnom extrém. Aby evolučný proces mohol ďalej pokračovať, populácia sa reštartuje, pričom sa zachová len najlepší jedinec (*elitárstvo*) a zvyšní štyria sa náhodne vygenerujú. Elitárstvo zaručí, že najlepší jedinec v nasledujúcej generácii nebude horší od predchádzajúceho.

V nasledujúcich častiach najskôr schématicky zapíšeme štruktúru mikrogenetického algoritmu, ktorú navrhol a úspešne implementoval Krishnakumar v roku 1989 ([51]) a potom podrobnejšie objasníme jeho objekty a operácie adaptované na riešenie CVRP.

4.3.1 Mikrogenetický algoritmus μ GA

Krok 1: Vytvor náhodne populáciu s 5 jedincami (inicializácia) a choď na krok 3.

Krok 2: Vytvor populáciu so 4 jedincami náhodne a 1 najlepšieho jedinca z predchádzajúcej generácie (reštart).

Krok 3: Vypočítaj fitness jedincov.

Krok 4: Urči najlepšieho jedinca - toho uchovaj do nasledujúcej generácie (elitárstvo).

Krok 5: Pomocou turnajového výberu zvoľ 2 páry jedincov (rodičov) na reprodukciu.

Krok 6: Vykonaj kríženie a potomkov pridaj do novej generácie.

Krok 7: Vypočítaj fitness jedincov.

Krok 8: Skontroluj, či došlo k strate diverzity a ak nie, choď na krok 4.

Krok 9: Ak nebolo splnené ukončovacie kritérium, choď na krok 2., inak Stop.

Ukončovacím kritériom môže byť napríklad dosiahnutie predpísaného počtu generácií počítaných od začiatku vývoja alebo od posledného zlepšenia hodnoty fitness najlepšieho jedinca. Výsledným riešením je najlepší jedinec v aktuálnej generácii.

4.3.2 Jedinec a jeho fitness

Jedinec S v našom μ GA predstavuje riešenie CVRP a jeho fitness $f(S)$ je rovná cene tohto riešenia.

Jedinca budeme reprezentovať ako vektor (*chromozóm*), ktorého zložky (*gény*) sú zákazníci v takom poradí, v akom sú obslužení na jednotlivých trasách, bez oddeľovačov jednotlivých trás, t.j. $S = (S_1, S_2, \dots, S_n)$, kde $S_i \in \{1, \dots, n\}$ a $S_i \neq S_j$ ak $i \neq j$. Jedná sa o tzv. *permutačné kódovanie jedinca*. Obrázok 1.1 ilustruje riešenie CVRP (jedinca) s $n = 7$ zákazníkmi, kapacitou vozidiel $Q = 100$ a $p = 3$ trasami: $0 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 0$, $0 \rightarrow 7 \rightarrow 4 \rightarrow 0$ a $0 \rightarrow 2 \rightarrow 6 \rightarrow 0$. Jemu zodpovedajúci chromozóm teda je $S = (1, 3, 5, 7, 4, 2, 6)$ a jeho fitness $f(S) = 227$. V prípade potreby (napríklad pri výpočte hodnoty fitness $f(S)$) sa dá chromozóm S dekódovať pomocou procedúry *split* (Prins, 2004 [71]). Táto procedúra nájde optimálne miesta, na ktorých treba vektor S rozdeliť na jednotlivé trasy tak, že výsledné riešenie je prípustné a má minimálnu cenu (obr.4.4).

Procedúra *split*

Krok 1: Vytvor pomocný hranovo ohodnotený digraf $\vec{D} = (X, \vec{F}, w)$ s množinou vrcholov $X = \{0, 1, 2, \dots, n\}$. Do množiny hrán \vec{F} daj každú takú orientovanú hranu (i, j) , $i, j \in X$, $i < j$, pre ktorú platí, že trasa obsahujúca zákazníkov S_{i+1} až S_j je kapacitne prípustná, t.j. ak platí

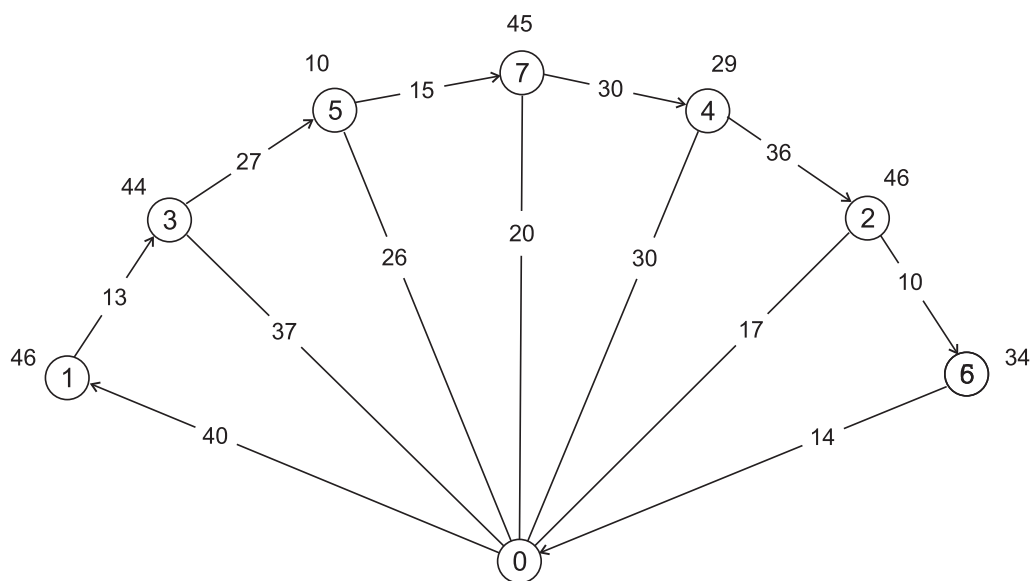
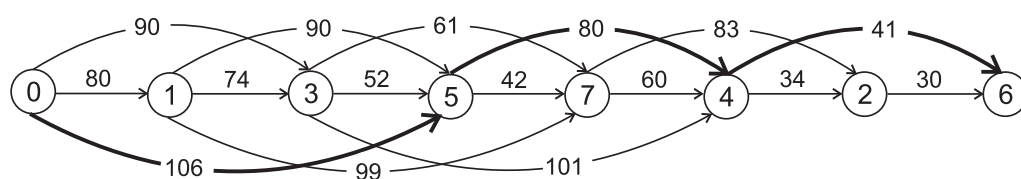
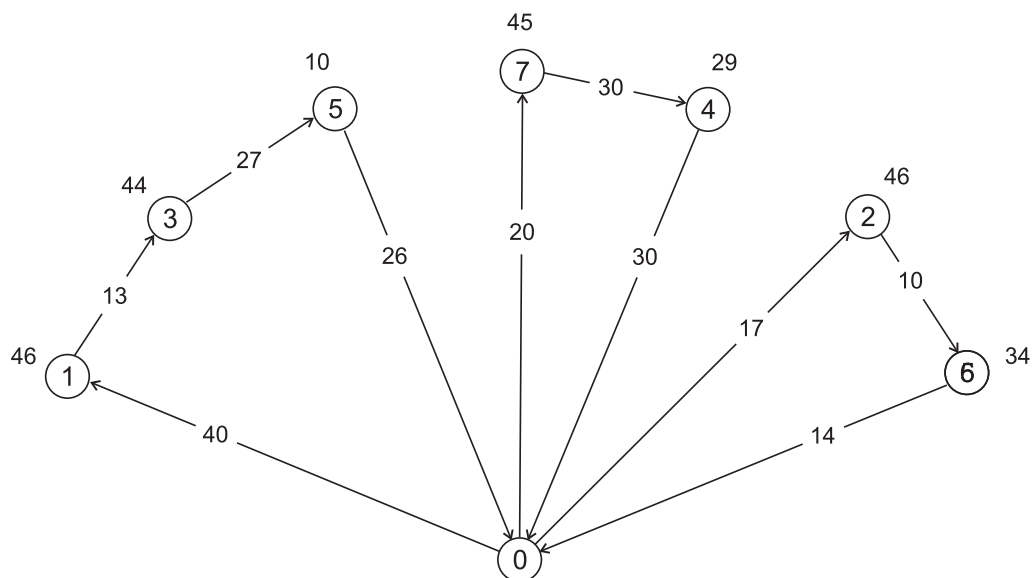
$$\sum_{k=i+1}^j d_{S_k} \leq Q.$$

Ohodnotenie w_{ij} hrany (i, j) sa rovná cene cyklu $(0, S_{i+1}, S_{i+2}, \dots, S_j, 0)$, t.j.

$$w_{ij} = c_{0, S_{i+1}} + \sum_{k=i+1}^{j-1} c_{S_k S_{k+1}} + c_{S_j 0}.$$

Krok 2: Nájdi v digrafe \vec{D} najkratšiu orientovanú cestu z vrcholu 0 do vrcholu n .

Najkratšia orientovaná cesta z vrcholu 0 do vrcholu n v digrafe \vec{D} udáva miesta, v ktorých bude vektor S optimálne rozdelený na prípustné trasy a jej dĺžka zodpovedá cene riešenia, čiže fitness $f(S)$.

(a) Chromozóm $S = (1, 3, 5, 7, 4, 2, 6)$ (b) Pomocný digraf \vec{D} a najkratšia orientovaná cesta(c) Optimálne rozdelený chromozóm S Obr. 4.4: Procedúra *split*

4.3.3 Turnajový výber

Výber jedincov pre reprodukciu je operácia, ktorá v GA simuluje prirodzený výber. Pri výbere jedinca medzi sebou rôznymi spôsobmi súťažia o možnosť zúčastniť sa reprodukcie (stať sa rodičmi). Hlavnou myšlienkou je, že kombináciou dobrých jedincov vzniknú lepší potomkovia. V μ GA použijeme metódu *binárneho turnajového výberu* (Goldberg, 1990 [36]).

Binárny turnajový výber

Krok 1: Premiešaj aktuálnu generáciu tak, aby poradie jedincov bolo náhodné.

Krok 2: Porovnaj fitness jedinca 1 s fitness jedinca 2. Jediniec s lepšou hodnotou fitness je vybraný ako rodič 1.

Krok 3: Porovnaj fitness jedinca 3 s fitness jedinca 4. Jediniec s lepšou hodnotou fitness je vybraný ako rodič 2.

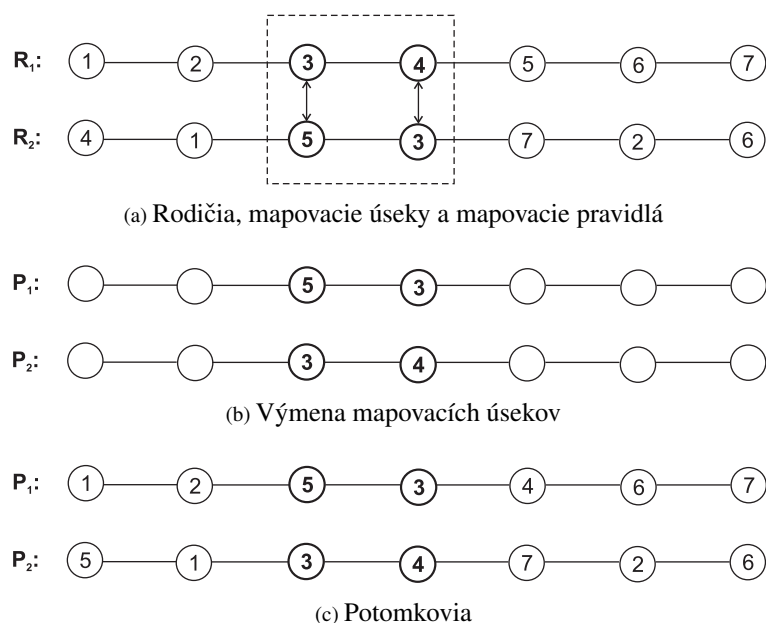
Tento proces sa opakuje, kým nie je vybraný požadovaný počet rodičovských párov, t.j. v našom prípade dva rodičovské páry. Po výbere nasleduje reprodukcia rodičov (kríženie) v jednotlivých pároch.

4.3.4 Kríženie

Pri permutačnom kódovaní je možné využiť niekoľko typov operátorov kríženia. Jedným z nich je *kríženie s čiastkovým mapovaním* (Partially Mapped Crossover, PMX) [37].

Kríženie PMX

Operátor PMX vytvorí z dvoch rodičovských chromozómov dva potomkovské chromozómy nasledovným spôsobom. Najprv sa náhodne vyberú dve miesta (*body kríženia*) vnútri rodičovských chromozómov. Podreťazce medzi bodmi kríženia (*mapovacie úseky*) určia *mapovacie pravidlá*. Mapovací úsek z druhého rodiča sa skopíruje na rovnaké miesto do prvého potomka. Zvyšné gény tohto potomka sa potom kopírujú z prvého rodiča, pričom gény, ktoré sú už v chromozóme prítomné, sa nahradia podľa mapovacích pravidiel. Analogicky sa postupuje aj pri formovaní druhého potomka, avšak sa vymenia úlohy rodičov.



Obr. 4.5: Kríženie PMX

Na obrázku 4.5 máme dvoch rodičov $R_1 = (1, 2, 3, 4, 5, 6, 7)$ a $R_2 = (4, 1, 5, 3, 2, 7, 2, 6)$. Tučným písmom sú zvýraznené mapovacie úseky, ktoré definujú mapovacie pravidlá $3 \leftrightarrow 5$ a $4 \leftrightarrow 3$. Výsledkom kríženia sú potomkovia $P_1 = (1, 2, 5, 3, 4, 6, 7)$ a $P_2 = (5, 1, 3, 4, 7, 2, 6)$.

4.3.5 Kontrola diverzity

Mikrogenetický algoritmus má tendenciu rýchlo skonvergovať ku stavu, kedy populácia obsahuje iba veľmi podobných alebo dokonca rovnakých jedincov a vtedy už nemožno očakávať progres. Na kontrolu diverzity použijeme parameter α , ktorý definujeme nasledovne [1]:

$$\alpha = \left(\frac{f_{max}(\cdot) - f_{min}(\cdot)}{f_{min}(\cdot)} \right),$$

kde $f_{max}(\cdot)$ je najväčšia a $f_{min}(\cdot)$ je najmenšia (najlepšia) fitness v aktuálnej generácii populácie. Ak $\alpha \leq 0,01$, tak usúdime, že došlo k strate diverzity.

Nevýhodou genetického algoritmu je dlhý čas výpočtu, čo je spôsobené veľkým počtom vyhodnotení účelovej funkcie (fitness jedincov). Avšak mikrogenetický algoritmus pracuje s malou populáciou, a preto je výpočtový čas v porovnaní s GA významne kratší a aj pamäťová náročnosť na uchovanie populácie je omnoho menšia. Pri riešení problému môžeme μ GA niekoľkokrát po sebe zopakovať a nakoniec spomedzi výsledných riešení jednotlivých

pokusov vyberieme najlepšie riešenie. Viacnásobným opakovaním výpočtu zvyšujeme pravdepodobnosť nájdenia kvalitného riešenia. Opakovanie výpočtu sa zastaví buď po vykonaní určitého počtu pokusov, alebo po uplynutí stanoveného času.

4.3.6 Paralelizácia algoritmu

Efektívnosť výpočtu môžeme zvýšiť jeho paralelizáciou, t.j. rozložením výpočtového výkonu na viac procesorov. V literatúre možno nájsť niekoľko stratégií paralelizácie GA, napr. [19], [76]. Najjednoduchším spôsobom je spustiť výpočet nezávisle na viacerých procesoroch, resp. jadrách, tzv. *Embarrassingly Parallel Computation* (EPC). Pri tomto spôsobe každý procesor (jadro) vykoná určenú časť zo stanoveného počtu pokusov a po skončení oznámi svoj najlepší výsledok. Ďalším spôsobom je paralelizácia, pri ktorej výpočet prebieha na viacerých nezávislých, ale navzájom komunikujúcich procesoroch. Napríklad, procesory môžu spolupracovať pri vyhodnotení fitness jedincov (*master-slave* model). Táto alternatíva má zmysel najmä pri úlohách, v ktorých vyhodnotenie účelovej funkcie je časovo náročné. Iná možnosť (*coarse-grained* - hrubozrnný model) je, že populácia je rozdelená na niekoľko podpopulácií medzi jednotlivé procesory. Každá podpopulácia sa potom vyvíja relatívne samostatne. Pri splnení vhodných podmienok (napr. po určitom počte generácií) si podpopulácie navzájom vymenia niektorých jedincov (*migrácia*) a následne pokračujú ďalej opäť izolovane. Pri ďalšej metóde (*fine-grained* - jemnozrnný model) sú jedinci umiestnení na veľkej dvojrozmernej mriežke a výber, kríženie a migrácia sa uskutočňuje medzi susediacimi jedincami. Využitie rôznych typov paralelných GA na riešenie VRP prezentovali autori Alba a Dorronsoro [2], Ochi a kol. [66] a Groër a kol. [38].

V nasledujúcej kapitole opíšeme paralelný mikrogenetický algoritmus, ktorý sme navrhli pre riešenie CVRP. Náš algoritmus pracuje na princípe hrubozrnného modelu, avšak namiesto podpopulácií veľkej populácie, sme použili mikropopulácie a proces migrácie sme nahradili dosadením *nasadeného jedinca*.

4.3.7 Paralelný mikrogenetický algoritmus $p\mu GA$

Najprv modifikujeme algoritmus μGA tak, že vstupom bude vektor reprezentujúci nasadeného jedinca a pri inicializácii (v kroku 1.) sa vytvorí populácia s nasadeným jedincom a 4 náhodne vygenerovanými jedincami. Každý procesor bude vykonávať μGA so svojou vlastnou mikropopuláciou, až kým nebude splnené ukončovacie kritérium. Túto periódu izolovaného vývoja mikropopulácie nazveme *epocha*.

Inicializácia prvého nasadeného jedinca

Prvý nasadený jedinec môže byť vytvorený buď náhodným vygenerovaním alebo použitím rýchlejšej konštruktívnej heuristiky, napr. Clarke-Wrightovej metódy.

Úlohy, ktoré vykonáva procesor počas jednej epochy

Krok 1: Prijmi nasadeného jedinca.

Krok 2: Inicializuj populáciu s 5 jedincami: 1 nasadený jedinec a 4 náhodne vygenerovaní jedinci.

Krok 3: Opakuj μGA kým nie je splnené ukončovacie kritérium.

Krok 4: Odovzdaj najlepšieho jedinca.

Jeden z procesorov je určený ako manažér úloh (*root*). Jeho úlohou je na začiatku každej novej epochy rozposlať (všetkým procesorom rovnakého) nasadeného jedinca a po jej skončení vyzbierať a vyhodnotiť výsledky od všetkých procesorov.

Operácie, ktoré vykonáva manažér úloh

Krok 1: Inicializuj prvého nasadeného jedinca.

Krok 2: Odošli nasadeného jedinca jednotlivým procesorom (*broadcast*).

Krok 3: Po skončení epochy prijmi výsledky od všetkých procesorov (*gather*).

Krok 4: Urči najlepší zo všetkých prijatých výsledkov a označ ho ako nového nasadeného jedinca.

Krok 5: Ak nebolo splnené ukončovacie kritérium, choď na krok 2., inak Stop.

Výpočet sa ukončí po stanovenom počte epoch a výsledkom bude posledný nasadený jedinec.

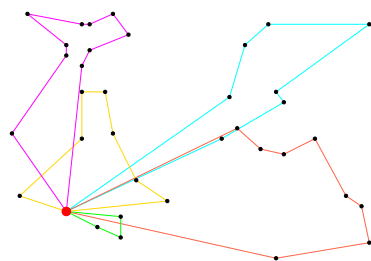
4.3.8 Výsledky experimentov

Naše experimenty boli zamerané na otestovanie navrhnutého mikrogenetického algoritmu pri riešení CVRP a porovnanie získaných výsledkov s optimálnymi výsledkami známymi z literatúry. Algoritmus $p\mu GA$ sme implementovali v jazyku Python 3.4 [88] s využitím modulu `mpi4py` (*Message Passing Interface for Python* [92]) pre paralelné výpočty. Testovanie sme realizovali na PC s procesorom Intel i7-5960X, 3.00GHz, 8 jadier a pamäťou 32GB. Experimenty boli vykonané na 16 klasických inštanciách pre CVRP od autorov Augerat a kol., 1995 [90]. Navrhnutý paralelný mikrogenetický algoritmus ($p\mu GA$) bol na každej inštancii ukončený po 2500 epochách, pričom ukončovacím kritériom modifikovaného algoritmu μGA bolo 50 generácií bez zlepšenia hodnoty účelovej funkcie. Prvý nasadený jedinec bol vytvorený dvomi spôsobmi - náhodným vygenerovaním a použitím Clarke-Wrightovej metódy.

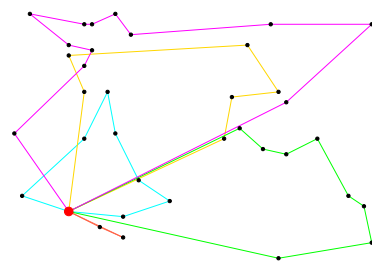
Výsledky testov sú zobrazené v tabuľke 4.3. V prvých troch stĺpcoch je názov inštancie, počet zákazníkov n a kapacita vozidiel Q . V ďalšom stĺpci je uvedené optimálne riešenie, pričom Opt predstavuje jeho cenu. Nasledujúce dva stĺpce obsahujú výsledky získané našim navrhnutým algoritmom s rozdielnymi spôsobmi inicializácie prvého nasadeného jedinca - náhodným vygenerovaním (stĺpec $p\mu GA/ rand$) a použitím Clarke-Wrightovej metódy (stĺpec $p\mu GA/ cws$). V jednotlivých výsledkoch je udaná cena prvého nasadeného jedinca (*Štart*), cena riešenia (*Rieš*) a relatívna odchýlka (*Odch*) nášho výsledku v porovnaní s optimálnym riešením, ktorú sme vypočítali pomocou vzorca:

$$Odch = \frac{Rieš - Opt}{Opt}.$$

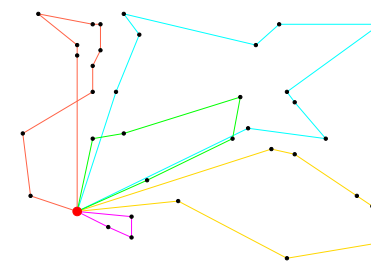
V poslednom riadku tabuľky sa nachádzajú priemerné odchýlky pre obe metódy inicializácie.



(a) Optimálne riešenie

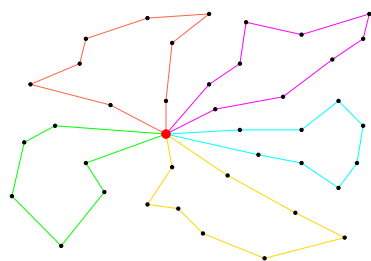


(b) Riešenie nájdené pμGA/rand

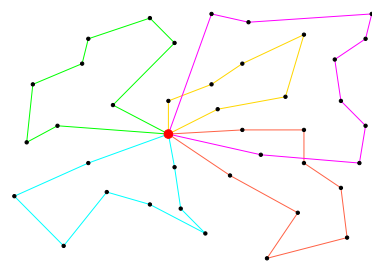


(c) Riešenie nájdené pμGA/cws

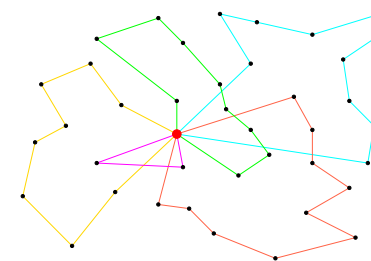
Obr. 4.6: Vizualizácia riešení inštancie A-n36-k5



(a) Optimálne riešenie



(b) Riešenie nájdené pμGA/rand



(c) Riešenie nájdené pμGA/cws

Obr. 4.7: Vizualizácia riešení inštancie P-n40-k5

Pre názornosť sú na obrázkoch 4.6 a 4.7 vizualizované riešenia dvoch vybraných inštancií, A-n36-k5 s depom v rohu a P-n40-k5 s depom v centre. Obrázky boli nakreslené pomocou NetworkX [93], modulu jazyka Python určeného pre prácu so sieťami.

4.3.9 Záver

Na základe našich výsledkov sa ukazuje, že algoritmus $p\mu$ GA je schopný nájsť optimálne alebo prijateľné suboptimálne riešenie. Pritom spôsob vytvorenia prvého nasadeného jedinca ovplyvňuje kvalitu výsledku. Pri náhodnom generovaní algoritmus štartoval s horším riešením než pri použití Clarke-Wrightovej metódy a dosiahol riešenia s priemernou relatívnou odchýlkou 14 %, kým pri druhom spôsobe len 7 %. V budúcom výskume plánujeme využiť $p\mu$ GA pri hľadaní robustného riešenia CVRP. Tento algoritmus sme prezentovali na medzinárodnej konferencii COMPENG 2018 vo Florencii v októbri 2018 [14].

Tabuľka 4.3: Porovnanie výsledkov podľa spôsobu generovania inicializačného riešenia

Inštancia	n	Q	Opt	p μ GA/rand			p μ GA/cws		
				Štart	Rieš	Odch	Štart	Rieš	Odch
P-n16-k8	15	35	450	630	450	0.00	482	450	0.00
P-n19-k2	18	160	212	471	212	0.00	236	212	0.00
P-n20-k2	19	160	216	464	218	0.01	237	234	0.08
P-n22-k2	21	160	216	491	235	0.09	246	240	0.11
P-n22-k8	21	3000	603	750	590	-0.02	656	590	-0.02
P-n40-k5	39	140	458	1041	504	0.10	509	494	0.08
P-n76-k4	75	350	593	1974	754	0.27	721	692	0.17
P-n101-k4	100	400	681	2075	971	0.43	816	804	0.18
A-n34-k5	33	100	778	1901	858	0.10	886	811	0.04
A-n36-k5	35	100	799	1880	870	0.09	870	828	0.04
A-n38-k5	37	100	730	2136	818	0.12	896	772	0.06
A-n44-k6	43	100	937	2455	1091	0.16	1100	1045	0.12
A-n53-k7	52	100	1010	2996	1209	0.20	1167	1092	0.08
A-n60-k9	59	100	1354	3474	1545	0.14	1603	1464	0.08
A-n69-k9	68	100	1159	3985	1381	0.19	1376	1256	0.08
A-n80-k10	79	100	1763	4745	2455	0.39	1961	1912	0.08
Priemerná relatívna odchýlka:						0.14			0.07

n – počet zákazníkov, Q – kapacita vozidiel, Opt – cena optimálneho riešenia, Štart – cena prvého nasadeného jedinca, Rieš – cena nájdeného riešenia, Odch – relatívna odchýlka

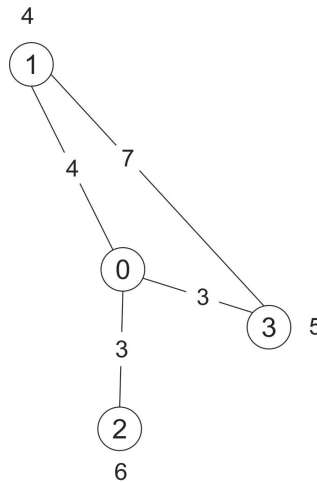
4.4 Lexikografický model CVRP

V našom modeli CVRP 2 sa využíva presne p vozidiel, pričom predpokladáme, že celkový dopyt nepresahuje kapacitu všetkých vozidiel, t.j. $\sum_{i=1}^n d_i \leq pQ$. Avšak v niektorých praktických situáciách celková kapacita všetkých vozidiel nepostačuje na to, aby boli uspokojené všetky požiadavky, hoci nie je menšia než celkový dopyt zákazníkov.

Napríklad majme CVRP s počtom zákazníkov $n = 3$ s požiadavkami $d_1 = 6$, $d_2 = 5$ a $d_3 = 4$, počtom vozidiel $p = 2$ s kapacitou $Q = 8$. Celková kapacita vozidiel $pQ = 16$ je väčšia než celkový dopyt všetkých zákazníkov $d_1 + d_2 + d_3 = 15$, a i napriek tomu nie je možné uspokojiť všetkých zákazníkov.

Ak by sme predsa len chceli využiť presne p vozidiel (napr. preto, že depo zamestnáva p vodičov), musíme určiť, najmenej akú kapacitu Q_{min} musia mať vozidlá, aby boli schopné obslúžiť všetkých zákazníkov. Je zrejmé, že ak chceme v našom príklade použiť 2 vozidlá s rovnakou kapacitou Q , tak táto kapacita musí byť aspoň $Q_{min} = 9$ (obr. 4.8).

Minimálnu potrebnú kapacitu Q_{min} určíme pomocou *modifikovaného problému kontajnerov* (mBPP), ktorý môžeme formulovať nasledovne: Treba rozmiestniť n objektov, každý s kladnou hmotnosťou d_i , $i = 1, \dots, n$, do p kontajnerov, $1 \leq p \leq n$, s čo najmenšou rovnakou kapacitou tak, aby celková hmotnosť objektov v žiadnom kontajneri nepresiahla jeho kapacitu.



Obr. 4.8: LCVRP $n = 3$, $p = 2$, $Q_{min} = 9$

Problém mBPP budeme modelovať pomocou bivalentných rozhodovacích premenných b_{ij} , kde

$$b_{ij} = \begin{cases} 1, & \text{ak objekt } j \text{ je umiestnený v kontajneri } i, \\ 0, & \text{inak} \end{cases} \quad (4.13)$$

a celočíselnej premennej q , $q > 0$, ktorá reprezentuje kapacitu kontajnera. Cieľom je minimalizovať q .

4.4.1 Model mBPP

Minimalizujte

$$q \quad (4.14)$$

za podmienok

$$\sum_{i=1}^p b_{ij} = 1, \quad \forall j \in \{1, \dots, n\}, \quad (4.15)$$

$$\sum_{j=1}^n b_{ij} \geq 1, \quad \forall i \in \{1, \dots, p\}, \quad (4.16)$$

$$\sum_{j=1}^n b_{ij} d_j \leq q, \quad \forall i \in \{1, \dots, p\}, \quad (4.17)$$

$$b_{ij} \in \{0, 1\}, \quad \forall i \in \{1, \dots, p\}, \forall j \in \{0, \dots, n\}. \quad (4.18)$$

Účelová funkcia (4.14) minimalizuje kapacitu kontajnera, resp. inými slovami hornú hranicu hmotnosti nákladu, ktorý do kontajnera smieme naložiť. Obmedzujúce podmienky (4.15) zaručujú, že každý objekt bude umiestnený do práve jedného kontajnera, nerovnice (4.16) zas zabezpečujú, že každý kontajner bude použitý. Ďalšie podmienky (4.17) sú kapacitné a (4.18) sú obligatórne podmienky.

4.4.2 Definícia lexikografickej CVRP

Podobne ako v CVRP, máme množinu n zákazníkov s istými požiadavkami, ktoré sú nedeliteľné. Označme D celkový dopyt všetkých zákazníkov, teda $D = \sum_{i=1}^n d_i$. Nech počet vozidiel s rovnakou kapacitou, ktorými disponuje depo je p .

Lexikografická kapacitná dopravná okružná úloha (LCVRP) má nájsť p prípustných trás tak, aby celkové prepravné náklady boli minimálne a pritom

- každá trasa začínala aj končila v depe,
- každý zákazník bol obslužený práve jedným vozidlom,
- na žiadnej trase nebola prekročená najmenšia nutná kapacita Q_{min} .

Daný problém môžeme riešiť dvojakým spôsobom. Prvá možnosť je vypočítať minimálnu kapacitu vozidla Q_{min} riešením korešpondujúceho mBPP a následne vyriešiť CVRP s $Q = Q_{min}$. Druhý spôsob, ako zobrať do úvahy požiadavku minimalizovať kapacitu vozidla, je použiť ďalšiu rozhodovaciu premennú q reprezentujúcu kapacitu vozidla (rovnako ako v mBPP) a upraviť účelovú funkciu pridaním člena Mq , kde M je veľké kladné číslo, tzv. penalizačná konštanta. Takto budeme mať možnosť v prvom rade minimalizovať kapacitu vozidla a popritom minimalizovať aj cenu riešenia.

Pretože v našom modeli CVRP 2 účelová funkcia namiesto minimalizácie prepravných nákladov maximalizuje celkovú úsporu, LCVRP sformulujeme matematicky takto:

4.4.3 Model LCVRP

Maximalizujte

$$\sum_{i=1}^n \sum_{j=1, i \neq j}^n s_{ij} x_{ij} - Mq, \quad (4.19)$$

za podmienok

$$\sum_{j=1}^n x_{0j} = p, \quad (4.20)$$

$$\sum_{i=0, i \neq j}^n x_{ij} = 1, \quad \forall j \in \{1, \dots, n\}, \quad (4.21)$$

$$\sum_{j=1, i \neq j}^n x_{ij} \leq 1, \quad \forall i \in \{1, \dots, n\}, \quad (4.22)$$

$$y_i + d_j x_{ij} - D(1 - x_{ij}) \leq y_j, \quad \forall i, j \in \{1, \dots, n\}, i \neq j, \quad (4.23)$$

$$d_i \leq y_i \leq q, \quad \forall i \in \{1, \dots, n\}, \quad (4.24)$$

$$q \geq \max\{[D/p], \max\{d_i\}\} \quad (4.25)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in \{0, \dots, n\}, i \neq j. \quad (4.26)$$

Podmienky (4.20), (4.21), (4.22), (4.24) a (4.26) sú rovnaké ako v modeli CVRP 2. Rovnice (4.23) vyjadrujú anticyklické podmienky a súčasne podmienky kontinuity toku a podmienka (4.25) vymedzuje dolnú hranicu premennej q .

4.4.4 Výsledky experimentov

Model LCVRP bol implementovaný v jazyku Python 3.4 [88] a riešený použitím optimalizačného softvéru Gurobi 6.5 [89]. Naše experimenty sme realizovali s 5 inštanciami z množiny klasických benchmarkov pre CVRP dostupných na [90] so 16 až 23 vrcholmi, pričom pri každej inštancii bolo nájdené optimálne riešenie. Aby sme overili správnosť výsledkov, všetky inštancie sme vyriešili aj pomocou druhej metódy (mBPP a CVRP 2). Z experimentov vyplynulo, že veľkosť penalizačnej konštanty M v účelovej funkcii (4.19) ovplyvňuje správnosť výsledkov a závisí od vstupných údajov. Napríklad, pre $M = 100$ sme pri inštancii P-n16-k8 dostali rovnaký výsledok ako pri druhej metóde, na rozdiel od E-n23-k3, kde výsledky získané obomi metódami boli odlišné. Z tohto dôvodu bolo potrebné nastaviť hodnotu M pre každú inštanciu tak, aby sme dosiahli rovnováhu medzi hodnotami členov $\sum_{i=1}^n \sum_{j=1, i \neq j}^n s_{ij} x_{ij}$ a Mq v účelovej funkcii (4.19). Toto sme docielili výpočtom podľa vzorca:

$$M = \left\lceil \frac{\max\{s_{ij}\}(n-p)}{\min\{d_i\}} \right\rceil, \quad (4.27)$$

kde $n - p$ je počet hrán, ktoré patria riešeniu a nie sú incidentné s depom. Experimentálne výsledky sú zobrazené v tabuľke 4.4.

4.4.5 Záver

Lexikografický model CVRP, ktorý sme predstavili v tejto kapitole, je vhodný pre menšie regionálne prepravné spoločnosti, ktoré disponujú vozidlami rovnakého typu. Benefitom homogénneho vozidlového parku je výhodnejšia nákupná cena alebo cena leasingu vozidiel, ako aj menej komplikovaná a lacnejšia údržba. Využívanie minimálnej nožnej kapacity vozidiel síce vedie k malému nárastu ceny prepravy v porovnaní s využitím plnej kapacity, ale na druhej strane sa znižujú rozdiely medzi nákladmi vozidiel a predchádza sa ich preťažovaniu (obr. 4.9), čo prispieva k znižovaniu výdavkov na údržbu vozidlového parku. Navyiac, nevy-

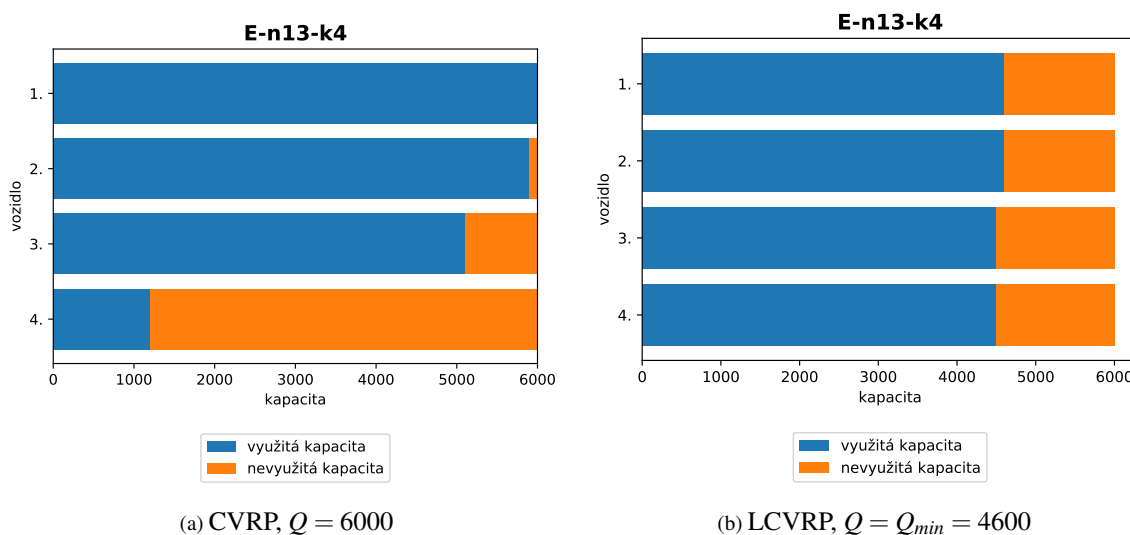
užitá kapacita vozidiel predstavuje rezervu pre prípad, že sa požiadavky neočakávane zvýšia, čiže výsledné riešenie do istej miery predchádza neuspokojeným požiadavkám.

Tento model sme prezentovali na medzinárodnej konferencii SOLI 2017 v Bari v septembri 2017 [16].

Tabuľka 4.4: Porovnanie výsledkov získaných dvomi metódami

Inštancia	n	p	Q	Opt	mBPP& CVRP 2		LCVRP		
					Q_{min}	$Cena$	M	Q_{min}	$Cena$
P-n16-k8	15	8	35	450	33	468	67	33	468
P-n19-k2	18	2	160	212	154	223	411	154	223
P-n21-k2	20	2	160	211	149	211	231	149	211
E-n13-k4	12	4	6000	247	4600	315	1	4600	315
E-n23-k3	22	3	4500	569	4100	570	39	4100	570

n – počet zákazníkov, p – počet vozidiel, Q – kapacita vozidla, Opt – cena optimálneho riešenia, Q_{min} – minimálna kapacita vozidla, $Cena$ – cena riešenia, M – penalizačná konštanta



Obr. 4.9: Využitie kapacity vozidla

4.5 Riešenie robustnej okružnej dopravnej úlohy s neistými požiadavkami zákazníkov

Problém robustnej okružnej dopravnej úlohy (RVRP) s neistými požiadavkami zákazníkov spočíva v tom, že kým v CVRP sú požiadavky zákazníkov vopred exaktne dané (deterministické), v RVRP majú povahu náhodných premenných, pričom však rozdelenie pravdepodobnosti týchto náhodných premenných nie je známe.

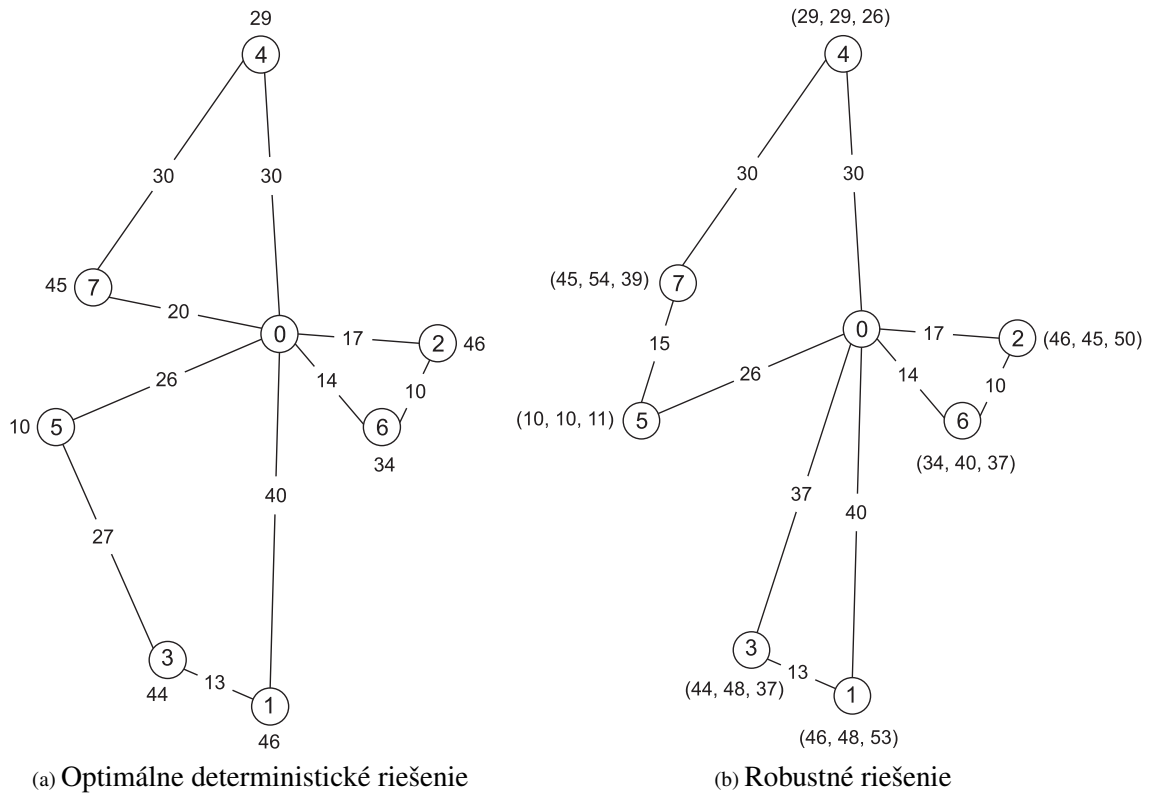
Za predpokladu, že trasy vozidiel musia byť určené ešte predtým, než budú známe aktuálne hodnoty požiadaviek zákazníkov a nebudú sa môcť dodatočne meniť, môže náhodné správanie požiadaviek spôsobiť, že navrhnuté prípustné riešenie sa stane neprípustným (lebo ak na niektorej trase bude prekročená kapacita vozidla, tak niektorí zákazníci nebudú obslužení). Napríklad, obrázok 4.10a znázorňuje optimálne riešenie CVRP pre $n = 7$ zákazníkov s požiadavkami $\mathbf{d} = (46, 46, 44, 29, 10, 34, 45)$, $p = 3$ vozidiel s kapacitou $Q = 100$. Jeho cena je 227. Je zrejmé, že ak na trase $0 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 0$ dôjde k zvýšeniu požiadavky niektorého zákazníka, nebude možné obslužiť ich všetkých.

S touto úlohou sa môžeme stretnúť v mnohých praktických situáciách [46], ako napríklad dodávka ropných produktov, priemyselných plynov a vykurovacieho oleja pre domácnosť [39], distribúcia zdravotníckeho materiálu do miest v stave núdze [23], doručovanie zásielok do nemocníc a reštaurácií [85], zvoz a rozvoz pošty, balíkov a recyklovaného materiálu z kancelárií a priemyselných závodov [59] a podobne.

4.5.1 Diskrétny scenár

Neistotu budeme modelovať pomocou *diskrétnych scenárov* tak, že v každom scenári budú definované aktuálne požiadavky zákazníkov. Cieľom je nájsť *robustné riešenie*, čiže riešenie (s minimálnymi prepravnými nákladmi), ktoré bude "dobré" pre každý scenár, t. j. v každom scenári budú uspokojené všetky požiadavky.

Napríklad, na obrázku 4.10b je zobrazené robustné riešenie problému s tromi scenármi $\mathbf{d}^0 = (46, 46, 44, 29, 10, 34, 45)$, $\mathbf{d}^1 = (48, 45, 48, 29, 10, 40, 54)$ a $\mathbf{d}^2 = (53, 50, 37, 26, 11, 37, 39)$. Jeho cena je 232, čo síce je o 2,2% viac oproti zodpovedajúcemu optimálnemu determinis-



Obr. 4.10: RVRP s neistými požiadavkami

tickému riešeniu CVRP na obrázku 4.10a, ale bez ohľadu na to, ktorý scenár nastane, všetci zákazníci budú obslužení.

4.5.2 Formálna definícia problému

Pri formulácii RVRP budeme predpokladať, že neisté požiadavky zákazníkov patria danej ohraničenej množine diskrétnych scenárov

$$\mathcal{U}_d = \{\mathbf{d}^k \in R^n | k \in \{1, \dots, s\}\},$$

kde \mathbf{d}^k je vektor požiadaviek v scenári k a s je počet scenárov. Budeme uvažovať takú množinu \mathcal{U}_d , v ktorej sú vektory \mathbf{d}^k konštruované pomocou odchýlok od vektora \mathbf{d}^0 očakávaných hodnôt, t. j. pre požiadavku d_j^k zákazníka j v scenári k platí

$$d_j^k \in \langle d_j^0 - \varepsilon d_j^0, d_j^0 + \varepsilon d_j^0 \rangle \quad \forall j \in \{1, \dots, n\},$$

kde ε je konštanta, ktorá určuje maximálnu možnú relatívnu odchýlku od nominálnej hodnoty, pričom

$$(1 + \varepsilon) \sum_{i=1}^n d_i^k \leq pQ \text{ pre } \forall k \in \{1, \dots, s\}.$$

RVRP hľadá riešenie, ktoré je prípustné pre všetky scenáre z množiny \mathcal{U}_d . To znamená, že formuláciu RVRP s neistými požiadavkami môžeme odvodiť z formulácie CVRP (Model CVRP 2) ako nasledujúcu úlohu MILP:

Model RVRP 1

Maximalizujte

$$\sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n s_{ij} x_{ij}, \quad (4.28)$$

za podmienok

$$\sum_{j=1}^n x_{0j} = p, \quad (4.29)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^n x_{ij} = 1, \quad \forall j \in \{1, \dots, n\}, \quad (4.30)$$

$$\sum_{\substack{j=1 \\ i \neq j}}^n x_{ij} \leq 1, \quad \forall i \in \{1, \dots, n\}, \quad (4.31)$$

$$y_i + d_j x_{ij} - Q(1 - x_{ij}) \leq y_j, \quad \forall d \in \mathcal{U}_d, \forall i, j \in \{1, \dots, n\}, i \neq j, \quad (4.32)$$

$$d_i \leq y_i \leq Q, \quad \forall d \in \mathcal{U}_d, \forall i \in \{1, \dots, n\}, \quad (4.33)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in \{0, \dots, n\}, i \neq j. \quad (4.34)$$

4.5.3 Stratégie na riešenie RVRP

Pretože RVRP je NP-t'azký kombinatorický problém, riešiť exaktne úlohu formulovanú v modeli RVRP 1 je možné len pre inštanície s malým počtom zákazníkov n a scenárov s . Riešenie sa však dá zjednodušiť na základe toho, že množina scenárov \mathcal{U}_d je ohraničená. Pre takúto množinu sa dá dokázať, že výsledný RVRP problém je inštanciou CVRP [80].

Veta 4.5.1. *Nech $d_j^m = \max\{\max_k d_j^k, d_j^0\}$ pre $j \in \{1, \dots, n\}, k \in \{1, \dots, s\}$. Potom pre definovanú množinu \mathcal{U}_d môžeme podmienky (4.32) nahradiť podmienkami*

$$y_i + d_j^m x_{ij} - Q(1 - x_{ij}) \leq y_j, \quad \forall i, j \in \{1, \dots, n\}, i \neq j. \quad (4.35)$$

Dôkaz: Upravme podmienky (4.32) do tvaru

$$y_j - y_i + Q(1 - x_{ij}) \geq d_j x_{ij}, \quad \forall d \in \mathcal{U}_d, \forall i, j \in \{1, \dots, n\}, i \neq j.$$

Potom, ak nahradíme \mathcal{U}_d podľa definície, dostaneme

$$y_j - y_i + Q(1 - x_{ij}) \geq d_j^k x_{ij}, \quad \forall k \in \{1, \dots, s\}, \forall i, j \in \{1, \dots, n\}, i \neq j.$$

Pretože je množina \mathcal{U}_d ohraničená, musí byť

$$y_j - y_i + Q(1 - x_{ij}) \geq \max_k d_j^k x_{ij}, \quad \forall i, j \in \{1, \dots, n\}, i \neq j.$$

Ak k týmto nerovnostiam pridáme podmienku

$$y_j - y_i + Q(1 - x_{ij}) \geq d_j^0 x_{ij}, \quad \forall i, j \in \{1, \dots, n\}, i \neq j,$$

dostaneme

$$y_j - y_i + Q(1 - x_{ij}) \geq \max\{\max_k d_j^k, d_j^0\} x_{ij}, \quad \forall i, j \in \{1, \dots, n\}, i \neq j,$$

t.j.

$$y_j - y_i + Q(1 - x_{ij}) \geq d_j^m x_{ij}, \quad \forall i, j \in \{1, \dots, n\}, i \neq j.$$

□

Stratégia maximálneho scenára

Na základe dokázaného tvrdenia bude optimálne riešenie úlohy so scenárom \mathbf{d}^m prípustné pre všetky scenáre z množiny \mathcal{U}_d , a tak robustné riešenie môžeme hľadať optimalizáciou inštancie CVRP s vektorom požiadaviek \mathbf{d}^m . Stačí teda riešiť nasledujúcu úlohu MILP odvodenú z modelu CVRP 2:

Model RVRP 2

Maximalizujte

$$\sum_{i=1}^n \sum_{\substack{j=1 \\ i \neq j}}^n s_{ij} x_{ij}, \quad (4.36)$$

za podmienok

$$\sum_{j=1}^n x_{0j} = p, \quad (4.37)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^n x_{ij} = 1, \quad \forall j \in \{1, \dots, n\}, \quad (4.38)$$

$$\sum_{\substack{j=1 \\ i \neq j}}^n x_{ij} \leq 1, \quad \forall i \in \{1, \dots, n\}, \quad (4.39)$$

$$y_i + d_j^m x_{ij} - Q(1 - x_{ij}) \leq y_j, \quad \forall i, j \in \{1, \dots, n\}, i \neq j, \quad (4.40)$$

$$d_i^m \leq y_i \leq Q, \quad \forall i \in \{1, \dots, n\}, \quad (4.41)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in \{0, \dots, n\}, i \neq j. \quad (4.42)$$

Je potrebné si však uvedomiť, že táto inštancia je zvyčajne kapacitne tesnejšie obmedzená, než príslušná inštancia s vektorom očakávaných požiadaviek \mathbf{d}^0 , a preto výpočet bude zdĺhavejší.

V závislosti od povahy scenárov, môže nastať situácia, keď scenár \mathbf{d}^m nie je prípustný, hoci jednotlivé scenáre z množiny \mathcal{U}_d sú prípustné alebo dokonca, keď robustné riešenie neexistuje. V takýchto prípadoch sa pokúsime nájsť také riešenie, pri ktorom bude čo najmenej neuspokojených požiadaviek.

Stratégia najhoršieho prípustného scenára

Hlavným cieľom robustnej optimalizácie vo všeobecnosti je optimalizovať najhorší prípad zo všetkých možných scenárov. Ako sme ukázali v predchádzajúcej časti, môžeme za najhorší scenár považovať vektor požiadaviek \mathbf{d}^m so zložkami $d_j^m = \max_k \{\max_k d_j^k, d_j^0\}$, t. j. s najväčšími požiadavkami zákazníkov zo všetkých scenárov. Avšak tento scenár nemusí byť prípustný. Preto budeme hľadať *najhorší prípustný scenár*, čiže taký vektor požiadaviek \mathbf{d}^h , ktorého zložky $d_j^h \in \{d_j^k, k = 0, 1, \dots, s\}$ budú čo najväčšie, resp. ich súčet bude čo najväčší.

Použijeme rozhodovacie premenné t_{ijk} , kde

$$t_{ijk} = \begin{cases} 1, & \text{ak vozidlo } i \text{ obslúži požiadavku } d_j^k, \\ 0, & \text{inak.} \end{cases} \quad (4.43)$$

Našu úlohu nájsť najhorší prípustný scenár sformulujeme ako nasledujúcu úlohu bivalentného lineárneho programovania:

Model RVRP 3

Maximalizujte

$$\sum_{i=1}^p \sum_{j=1}^n \sum_{k=0}^s t_{ijk} d_j^k, \quad (4.44)$$

za podmienok

$$\sum_{j=1}^n \sum_{k=1}^s t_{ijk} \geq 1, \quad \forall i \in \{1, \dots, p\}, \quad (4.45)$$

$$\sum_{i=1}^p \sum_{k=0}^s t_{ijk} = 1, \quad \forall j \in \{1, \dots, n\}, \quad (4.46)$$

$$\sum_{j=1}^n \sum_{k=0}^s t_{ijk} d_j^k \leq Q, \quad \forall i \in \{1, \dots, p\}, \quad (4.47)$$

$$t_{ijk} \in \{0, 1\}, \quad \forall i \in \{1, \dots, p\}, \forall j \in \{0, \dots, n\}, \forall k \in \{0, \dots, s\}. \quad (4.48)$$

Účelová funkcia (4.44) maximalizuje celkovú sumu uspokojených požiadaviek. Podmienky (4.45) zabezpečujú, že každé vozidlo bude využité, t. j. obslúži aspoň jedného zákazníka. Skupina podmienok (4.46) zas stanovuje, že u každého zákazníka bude obslúžená požiadavka z práve jedného scenára a práve jedným vozidlom. Podmienky (4.47) sú kapacitné podmienky zaručujúce, že nebude prekročená kapacita žiadneho vozidla a (4.48) sú obligatórne podmienky.

Model RVRP 3 využijeme nielen na výpočet hľadaného najhoršieho scenára \mathbf{d}^h , ale súčasne aj na nájdenie prípustného riešenia, ktoré bude inicializačným riešením pre optimalizačné metódy, ktorými robustnú optimalizáciu dokončíme. V prípade inštancií s malým počtom zákazníkov to bude algoritmus CVRP 3 a pre väčšie úlohy zas algoritmus μ GA.

Stratégia najmenšej nutnej kapacity

Príčinou toho, že scenár \mathbf{d}^m nie je prípustný, je nepostačujúca celková kapacita vozidiel pQ . Jednou z možností, ako sa vysporiadať s týmto problémom, je použitie lexikografického modelu LCVRP, pomocou ktorého nájdeme optimálne riešenie inštancie so scenárom \mathbf{d}^m s využitím najmenšej nutnej kapacity vozidiel Q_{min} .

Minimálnu potrebnú kapacitu získame riešením nasledujúcej úlohy odvodenej z modelu mBPP:

Model RVRP 4

Minimalizujte

$$q \tag{4.49}$$

za podmienok

$$\sum_{i=1}^p b_{ij} = 1, \quad \forall j \in \{1, \dots, n\}, \tag{4.50}$$

$$\sum_{j=1}^n b_{ij} \geq 1, \quad \forall i \in \{1, \dots, p\}, \tag{4.51}$$

$$\sum_{j=1}^n b_{ij} d_j^m \leq q, \quad \forall i \in \{1, \dots, p\}, \tag{4.52}$$

$$b_{ij} \in \{0, 1\}, \quad \forall i \in \{1, \dots, p\}, \forall j \in \{0, \dots, n\}, \tag{4.53}$$

kde celočíselná premenná q , $q > 0$ v účelovej funkcii (4.49) modeluje minimálnu nutnú kapacitu Q_{min} a bivalentné premenné b_{ij} modelujú rozhodnutie, či vozidlo i obsluží zákazníka j .

4.5.4 Výsledky experimentov

Experimenty sme vykonali so 6 inštanciami z [90] s malým počtom zákazníkov (15 až 22) a so 6 inštanciami s väčším počtom zákazníkov (33 až 100). Najprv sme tieto inštancie modifikovali tak, aby v nich boli zahrnuté neisté požiadavky. Z deterministických (očakávaných) požiadaviek sme pre každú inštanciu vygenerovali 4 množiny \mathcal{U}_d obsahujúce po 5 scenárov požiadaviek so zadanou maximálnou odchýlkou ε , konkrétne $\varepsilon \in \{0,05; 0,1; 0,15; 0,2\}$.

Tabuľka 4.5 obsahuje vyhodnotenie deterministického optimálneho riešenia pri realizáciách vygenerovaných scenárov. Pri každej inštancii sme uviedli charakteristiku D , ktorá označuje súčet očakávaných požiadaviek všetkých zákazníkov

$$D = \sum_{i=1}^n d_i^0$$

a charakteristiku τ vyjadrujúcu pomer medzi súčtom očakávaných požiadaviek a celkovou kapacitou vozidiel [90]

$$\tau = \frac{\sum_{i=1}^n d_i^0}{pQ}.$$

Na základe tejto charakteristiky si môžeme utvoriť predstavu o tom, ako tesne je daná inštancia kapacitne obmedzená.

V stĺpci z_d je uvedená cena deterministického optimálneho riešenia s očakávanými požiadavkami a stĺpec g_d obsahuje maximálnu hodnotu neuspokojených požiadaviek, ktoré by vznikli, ak by sme pri realizácii jednotlivých scenárov použili optimálne riešenie. Napríklad, pre optimálne riešenie s trasami $(0, 1, 3, 5, 0)$, $(0, 7, 4, 0)$ a $(0, 2, 6, 0)$, s kapacitou vozidiel $Q = 100$ a vektorom očakávaných požiadaviek $\mathbf{d}^0 = (46, 46, 44, 29, 10, 34, 45)$ by pri realizácii scenára $\mathbf{d}^1 = (48, 45, 48, 29, 10, 40, 54)$ bol súčet požiadaviek na trasách 106, 83 a 85 a pri realizácii scenára $\mathbf{d}^2 = (53, 50, 37, 26, 11, 37, 39)$ by bol súčet požiadaviek na trasách 101, 65 a 87. Maximálna hodnota neuspokojených požiadaviek teda je $g_d = 6$.

Pri niektorých inštanciách je $g_d = 0$, čo znamená, že deterministické optimálne riešenie je prípustné pre všetky scenáre, t. j. je robustné.

Postup pri jednotlivých stratégiách

Stratégia maximálneho scenára (rvrp 1):

1. výpočet maximálneho scenára \mathbf{d}^m so zložkami $d_j^m = \max\{\max_k d_j^k, d_j^0\}$ pre $j \in \{1, \dots, n\}$,
 $k \in \{1, \dots, s\}$,
2. vyriešenie inštancie CVRP so scenárom \mathbf{d}^m a kapacitou Q .

Stratégia najhoršieho prípustného scenára (rvrp 2):

1. exaktný výpočet najhoršieho prípustného scenára \mathbf{d}^h použitím modelu RVRP 3,
2. vyriešenie inštancie CVRP so scenárom \mathbf{d}^h a kapacitou Q .

Stratégia najmenšej nutnej kapacity (rvrp 3):

1. exaktný výpočet najmenšej nutnej kapacity Q_{min} pri scenári \mathbf{d}^m použitím modelu RVRP 4,
2. vyriešenie inštancie CVRP so scenárom \mathbf{d}^m a kapacitou Q_{min} .

Na vyriešenie inštancií CVRP sme pri malom počte zákazníkov použili exaktnú metódu (algoritmus CVRP 3 s parametrom $\delta = 2$) a pri väčšom počte zákazníkov heuristickú metódu (algoritmus μ GA s parametrami: 50 generácií bez zlepšenia, 5000 epoch). Inicializačné riešenie pre uvedené metódy sme v stratégiách *rvrp 1* a *rvrp 3* získali pomocou Clarke-Wrightovej metódy [17] a v stratégii *rvrp 2* riešením modelu RVRP 3.

Všetky stratégie robustnej optimalizácie sme implementovali v jazyku Python 3.4 [88]. Exaktné metódy sme riešili pomocou solvera Gurobi 6.5 [89] na PC s parametrami Intel Xeon 32 jadier, 2,4 GHz, 256 GB RAM. Paralelné výpočty sme realizovali na univerzitnom klastri Cluster UNIZA s nasledovnou špecifikáciou ¹: 46 výpočtových uzlov (2 x 6 jadrový Intel Xeon, 2.27 GHz, 96 GB RAM), 2 výpočtové uzly (2 x 10 jadrový Intel Xeon, 2.27 GHz, 256 GB RAM).

Vyhodnotenie výsledkov robustnej optimalizácie použitím jednotlivých stratégií je zobrazené v tabuľkách:

- stratégia *rvrp 1* 4.6,
- stratégia *rvrp 2* 4.7,
- stratégia *rvrp 3* 4.8,

pričom každá tabuľka je rozdelená na dve časti – pre inštancie s malým počtom zákazníkov a pre inštancie s väčším počtom zákazníkov. Výsledky v druhej časti sú najlepšie riešenia získané z 5 pokusov.

¹<https://nic.uniza.sk/hpc/doku.php?id=hpc:klastery>

Tabuľka 4.5: Vyhodnotenie deterministického optimálneho riešenia

Inštancia	D	Q	τ	z_d	g_d	g_d/D	Inštancia	D	Q	τ	z_d	g_d	g_d/D
P-n16-k8-e5	246	35	0,88	450	5	0,020	P-n40-k5-e5	618	140	0,88	458	0	0,000
P-n16-k8-e10	246	35	0,88	450	1	0,004	P-n40-k5-e10	618	140	0,88	458	0	0,000
P-n16-k8-e15	246	35	0,88	450	4	0,016	P-n40-k5-e15	618	140	0,88	458	0	0,000
P-n16-k8-e20	246	35	0,88	450	5	0,020	P-n40-k5-e20	618	140	0,88	458	0	0,000
P-n20-k2-e5	310	160	0,97	216	0	0,000	P-n76-k4-e5	1364	350	0,97	593	0	0,000
P-n20-k2-e10	310	160	0,97	216	0	0,000	P-n76-k4-e10	1364	350	0,97	593	0	0,000
P-n20-k2-e15	310	160	0,97	216	2	0,006	P-n76-k4-e15	1364	350	0,97	593	0	0,000
P-n20-k2-e20	310	160	0,97	216	2	0,006	P-n76-k4-e20	1364	350	0,97	593	2	0,001
P-n23-k8-e5	313	40	0,98	529	1	0,003	P-n101-k4-e5	1458	400	0,91	681	0	0,000
P-n23-k8-e10	313	40	0,98	529	1	0,003	P-n101-k4-e10	1458	400	0,91	681	0	0,000
P-n23-k8-e15	313	40	0,98	529	5	0,016	P-n101-k4-e15	1458	400	0,91	681	0	0,000
P-n23-k8-e20	313	40	0,98	529	10	0,032	P-n101-k4-e20	1458	400	0,91	681	0	0,000
E-n13-k4-e5	18200	6000	0,76	247	69	0,004	A-n34-k5-e5	460	100	0,92	778	0	0,000
E-n13-k4-e10	18200	6000	0,76	247	143	0,008	A-n34-k5-e10	460	100	0,92	778	0	0,000
E-n13-k4-e15	18200	6000	0,76	247	438	0,024	A-n34-k5-e15	460	100	0,92	778	0	0,000
E-n13-k4-e20	18200	6000	0,76	247	390	0,021	A-n34-k5-e20	460	100	0,92	778	0	0,000
E-n22-k4-e5	22500	6000	0,94	375	28	0,001	A-n60-k9-e5	829	100	0,92	1354	0	0,000
E-n22-k4-e10	22500	6000	0,94	375	246	0,011	A-n60-k9-e10	829	100	0,92	1354	0	0,000
E-n22-k4-e15	22500	6000	0,94	375	360	0,016	A-n60-k9-e15	829	100	0,92	1354	0	0,000
E-n22-k4-e20	22500	6000	0,94	375	443	0,020	A-n60-k9-e20	829	100	0,92	1354	0	0,000
E-n23-k3-e5	10189	4500	0,75	569	28	0,003	A-n80-k10-e5	942	100	0,94	1763	0	0,000
E-n23-k3-e10	10189	4500	0,75	569	202	0,020	A-n80-k10-e10	942	100	0,94	1763	0	0,000
E-n23-k3-e15	10189	4500	0,75	569	75	0,007	A-n80-k10-e15	942	100	0,94	1763	0	0,000
E-n23-k3-e20	10189	4500	0,75	569	203	0,020	A-n80-k10-e20	942	100	0,94	1763	0	0,000

Tabuľka 4.6: Vyhodnotenie výsledkov stratégie *rvrp 1*

Inštancia	z_r	g_r	ζ	γ	Inštancia	z_r	g_r	ζ	γ
P-n16-k8-e5	450	0	0,000	0,020	P-n40-k5-e5	458	0	0,000	0,000
P-n16-k8-e10	462	0	0,027	0,004	P-n40-k5-e10	458	0	0,000	0,000
P-n16-k8-e15	479	0	0,064	0,016	P-n40-k5-e15	461	0	0,007	0,000
P-n16-k8-e20	467	0	0,038	0,020	P-n40-k5-e20	468	0	0,022	0,000
P-n20-k2-e5	217	0	0,005	0,000	P-n76-k4-e5	605	0	0,020	0,000
P-n20-k2-e10	n	n	n	n	P-n76-k4-e10	627	0	0,053	0,000
P-n20-k2-e15	n	n	n	n	P-n76-k4-e15	n	n	n	n
P-n20-k2-e20	n	n	n	n	P-n76-k4-e20	n	n	n	n
P-n23-k8-e5	577	0	0,091	0,003	P-n101-k4-e5	700	0	0,028	0
P-n23-k8-e10	n	n	n	n	P-n101-k4-e10	690	0	0,013	0
P-n23-k8-e15	n	n	n	n	P-n101-k4-e15	685	0	0,006	0
P-n23-k8-e20	n	n	n	n	P-n101-k4-e20	707	0	0,038	0
E-n13-k4-e5	277	0	0,121	0,004	A-n34-k5-e5	778	0	0,000	0,000
E-n13-k4-e10	277	0	0,121	0,008	A-n34-k5-e10	785	0	0,009	0,000
E-n13-k4-e15	284	0	0,150	0,024	A-n34-k5-e15	791	0	0,017	0,000
E-n13-k4-e20	285	0	0,154	0,021	A-n34-k5-e20	795	0	0,022	0,000
E-n22-k4-e5	382	0	0,019	0,001	A-n60-k9-e5	1360	0	0,004	0,000
E-n22-k4-e10	n	n	n	n	A-n60-k9-e10	1359	0	0,004	0,000
E-n22-k4-e15	n	n	n	n	A-n60-k9-e15	1373	0	0,016	0,000
E-n22-k4-e20	n	n	n	n	A-n60-k9-e20	1430	0	0,056	0,000
E-n23-k3-e5	572	0	0,005	0,003	A-n80-k10-e5	1805	0	0,024	0,000
E-n23-k3-e10	572	0	0,005	0,020	A-n80-k10-e10	1840	0	0,044	0,000
E-n23-k3-e15	572	0	0,005	0,007	A-n80-k10-e15	1836	0	0,041	0,000
E-n23-k3-e20	572	0	0,005	0,020	A-n80-k10-e20	1866	0	0,058	0,000

Tabuľka 4.7: Vyhodnotenie výsledkov stratégie *rvrp 2*

Inštančia	z_r	g_r	ζ	γ	Inštančia	z_r	g_r	ζ	γ
P-n16-k8-e5	450	0	0,000	0,020	P-n40-k5-e5	458	0	0,000	0,000
P-n16-k8-e10	462	0	0,027	0,004	P-n40-k5-e10	458	0	0,000	0,000
P-n16-k8-e15	479	0	0,064	0,016	P-n40-k5-e15	461	0	0,007	0,000
P-n16-k8-e20	467	0	0,038	0,020	P-n40-k5-e20	468	0	0,022	0,000
P-n20-k2-e5	217	0	0,005	0,000	P-n76-k4-e5	597	0	0,007	0,000
P-n20-k2-e10	217	0	0,005	0,000	P-n76-k4-e10	622	0	0,049	0,000
P-n20-k2-e15	219	0	0,014	0,006	P-n76-k4-e15	625	0	0,054	0,000
P-n20-k2-e20	217	2	0,009	0,000	P-n76-k4-e20	625	0	0,054	0,000
P-n23-k8-e5	577	0	0,091	0,003	P-n101-k4-e5	685	0	0,006	0,000
P-n23-k8-e10	605	1	0,144	0,000	P-n101-k4-e10	704	0	0,034	0,000
P-n23-k8-e15	529	5	0,000	0,000	P-n101-k4-e15	696	0	0,022	0,000
P-n23-k8-e20	618	8	0,168	0,006	P-n101-k4-e20	711	0	0,044	0,000
E-n13-k4-e5	277	0	0,121	0,004	A-n34-k5-e5	778	0	0,000	0,000
E-n13-k4-e10	277	0	0,121	0,008	A-n34-k5-e10	778	0	0,000	0,000
E-n13-k4-e15	284	0	0,150	0,024	A-n34-k5-e15	786	0	0,010	0,000
E-n13-k4-e20	285	0	0,154	0,021	A-n34-k5-e20	805	0	0,035	0,000
E-n22-k4-e5	382	0	0,019	0,001	A-n60-k9-e5	1358	0	0,003	0,000
E-n22-k4-e10	440	3	0,173	0,011	A-n60-k9-e10	1365	0	0,008	0,000
E-n22-k4-e15	619	13	0,651	0,015	A-n60-k9-e15	1379	0	0,018	0,000
E-n22-k4-e20	471	355	0,256	0,004	A-n60-k9-e20	1429	0	0,055	0,000
E-n23-k3-e5	572	0	0,005	0,003	A-n80-k10-e5	1809	0	0,026	0,000
E-n23-k3-e10	572	0	0,005	0,020	A-n80-k10-e10	1819	0	0,032	0,000
E-n23-k3-e15	572	0	0,005	0,007	A-n80-k10-e15	1824	0	0,035	0,000
E-n23-k3-e20	572	0	0,005	0,020	A-n80-k10-e20	1917	0	0,087	0,000

Tabuľka 4.8: Vyhodnotenie výsledkov stratégie *rvrp 3*

Inštancia	Q_{min}	z_r	g_r	ζ	γ	Inštancia	Q_{min}	z_r	g_r	ζ	γ
P-n16-k8-e5	33	468	0	0,040	0,020	P-n40-k5-e5	124	497	0	0,085	0,000
P-n16-k8-e10	34	473	0	0,051	0,004	P-n40-k5-e10	125	496	0	0,083	0,000
P-n16-k8-e15	35	479	0	0,064	0,016	P-n40-k5-e15	128	485	0	0,059	0,000
P-n16-k8-e20	35	467	0	0,038	0,020	P-n40-k5-e20	132	496	0	0,083	0,000
P-n20-k2-e5	157	217	0	0,005	0,000	P-n76-k4-e5	341	629	0	0,061	0,000
P-n20-k2-e10	163	217	0	0,005	0,000	P-n76-k4-e10	346	636	0	0,073	0,000
P-n20-k2-e15	171	217	2	0,005	0,000	P-n76-k4-e15	352	613	0	0,034	0,000
P-n20-k2-e20	178	217	2	0,005	0,000	P-n76-k4-e20	368	648	0	0,093	0,000
P-n23-k8-e5	40	572	0	0,081	0,003	P-n101-k4-e5	365	721	0	0,059	0,000
P-n23-k8-e10	41	618	0	0,168	0,003	P-n101-k4-e10	370	793	0	0,164	0,000
P-n23-k8-e15	43	534	5	0,009	0,000	P-n101-k4-e15	375	761	0	0,117	0,000
P-n23-k8-e20	45	562	10	0,062	0,000	P-n101-k4-e20	391	752	0	0,104	0,000
E-n13-k4-e5	4757	379	0	0,534	0,004	A-n34-k5-e5	92	813	0	0,045	0,000
E-n13-k4-e10	4804	343	0	0,389	0,008	A-n34-k5-e10	93	807	0	0,037	0,000
E-n13-k4-e15	5101	360	0	0,457	0,024	A-n34-k5-e15	95	853	0	0,096	0,000
E-n13-k4-e20	5242	379	0	0,534	0,021	A-n34-k5-e20	99	801	0	0,030	0,000
E-n22-k4-e5	5830	639	0	0,704	0,001	A-n60-k9-e5	93	1446	0	0,068	0,000
E-n22-k4-e10	6057	573	0	0,528	0,011	A-n60-k9-e10	93	1463	0	0,081	0,000
E-n22-k4-e15	6141	669	0	0,784	0,016	A-n60-k9-e15	96	1501	0	0,109	0,000
E-n22-k4-e20	6400	679	104	0,811	0,015	A-n60-k9-e20	99	1515	0	0,119	0,000
E-n23-k3-e5	4274	570	0	0,002	0,003	A-n80-k10-e5	95	1907	0	0,082	0,000
E-n23-k3-e10	4454	570	0	0,002	0,020	A-n80-k10-e10	95	1900	0	0,078	0,000
E-n23-k3-e15	4356	570	0	0,002	0,007	A-n80-k10-e15	97	1888	0	0,071	0,000
E-n23-k3-e20	4478	570	0	0,002	0,020	A-n80-k10-e20	99	1974	0	0,120	0,000

Kvalitu nájdeného riešenia z hľadiska neuspokojených požiadaviek sme ohodnotili pomocou ukazovateľa γ , ktorý predstavuje zníženie relatívneho množstva neuspokojených požiadaviek pri použití nájdeného riešenia namiesto optimálneho deterministického riešenia (3.1).

Ďalším ukazovateľom, ktorý sme použili na ohodnotenie nájdeného riešenia z hľadiska ceny, je relatívne zvýšenie ceny nájdeného riešenia voči cene optimálneho riešenia (3.2).

Analýza výsledkov

Neuspokojené požiadavky

Vo všetkých prípadoch, keď bol maximálny scenár \mathbf{d}^m prípustný, stratégia *rvrp 2* našla pomocou modelu RVRP 3 najhorší prípustný scenár \mathbf{d}^h zhodný s maximálnym scenárom ($\mathbf{d}^h = \mathbf{d}^m$). Preto sú výsledky získané stratégiami *rvrp 1* a *rvrp 2* pri inštanciách riešených exaktne rovnaké. Pri heuristickom riešení sa výsledky líšia, čo však vyplýva z povahy heuristických metód. Obom stratégiam sa podarilo nájsť robustné riešenia, ktoré sú schopné uspokojiť všetky požiadavky zákazníkov vo všetkých scenároch. Rovnako aj riešenia nájdené stratégiou *rvrp 3* sú všetky robustné.

Inštancie, pri ktorých bol maximálny scenár neprípustný, sú v tabuľkách zvýraznené. Tieto inštancie sa vyznačujú pomerne tesným kapacitným obmedzením ($\tau \geq 0,94$). Najmenšia kapacita nutná na to, aby bol scenár \mathbf{d}^m prípustný, je v týchto prípadoch $Q_{min} \geq Q$ (tabuľka 4.8). Z tohto dôvodu stratégia *rvrp 1* zlyhala (v tabuľke 4.6 je uvedený údaj "n"). Na rozdiel od nej, stratégie *rvrp 2* a *rvrp 3* našli riešenia, ktoré síce neboli všetky robustné ($g_r \geq 0$), ale množstvo neuspokojených požiadaviek bolo prijateľné. Heuristickou metódou obe stratégie vygenerovali len robustné riešenia ($g_r = 0$).

Zaujímavé je teda porovnanie výsledkov stratégií *rvrp 2* a *rvrp 3* z hľadiska neuspokojených požiadaviek v tých prípadoch, keď stratégia *rvrp 1* nebola úspešná. Graf 4.11a zobrazuje zníženie relatívneho množstva neuspokojených požiadaviek γ pri použití riešení získaných stratégiami *rvrp 2* a *rvrp 3*. Z predchádzajúcej kapitoly vieme, že čím väčšia je hodnota γ , tým je riešenie lepšie. Stratégia *rvrp 2* bola lepšia v dvoch prípadoch a *rvrp 3* v štyroch prípadoch z jedenástich. V piatich prípadoch boli rovnaké.

Cena riešenia

Aj pri tomto hľadisku porovnáme osobitne výsledky pre inštalácie, pri ktorých bol maximálny scenár neprípustný. Graf 4.11b vizualizuje relatívne zvýšenie ceny ζ pri stratégiách *rvrp 2* a *rvrp 3*, pričom menšia hodnota ζ znamená lepšie riešenie. Stratégia *rvrp 2* bola lepšia v šiestich prípadoch, *rvrp 3* v troch a v dvoch prípadoch z jedenástich boli rovnaké.

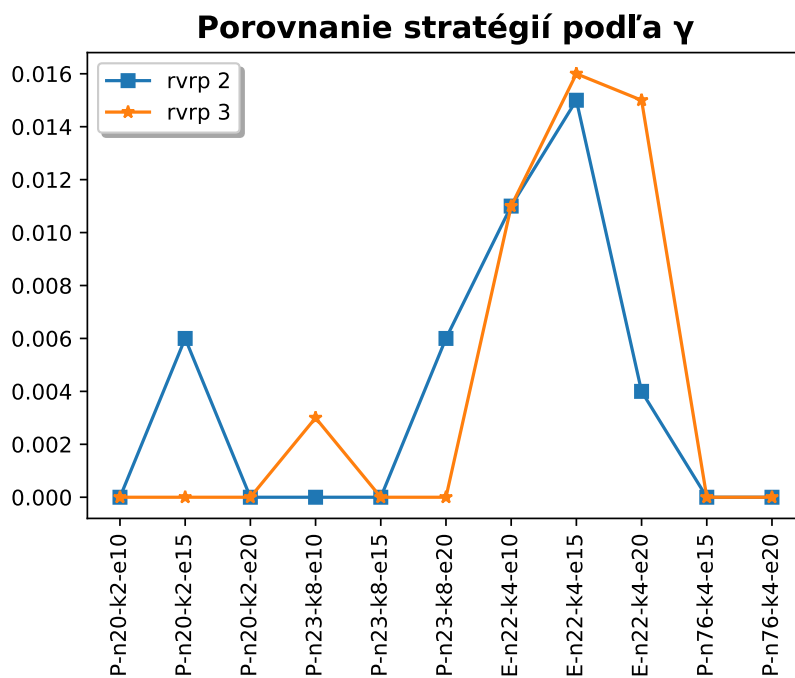
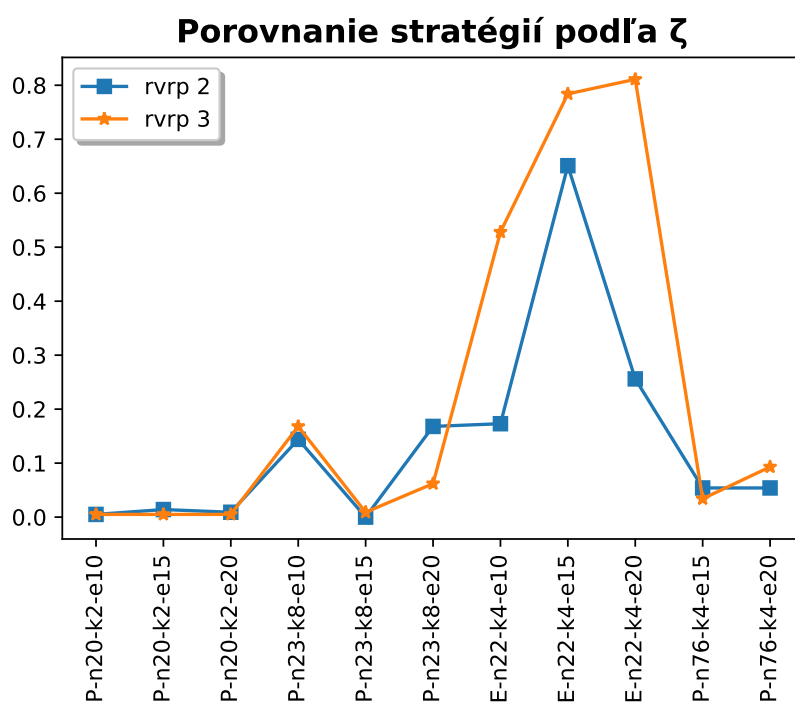
K zvýšeniu ceny o viac ako 50% došlo v riešeniach získaných stratégiou *rvrp 3* v troch prípadoch, kým u stratégie *rvrp 2* len v jednom.

Pri ostatných inštaláciách môžeme porovnať všetky tri stratégie. Najprv porovnáme výsledky pre inštalácie s malým počtom zákazníkov (riešené exaktne), a potom pre inštalácie s väčším počtom zákazníkov (riešené heuristicky). Opäť platí, že exaktné riešenia získané stratégiami *rvrp 1* a *rvrp 2* sú rovnaké, preto sú v grafe 4.12a zobrazené len výsledky stratégií *rvrp 2* a *rvrp 3*. Heuristické výsledky všetkých troch stratégií sú rozdielne, ako môžeme vidieť v grafe 4.12b.

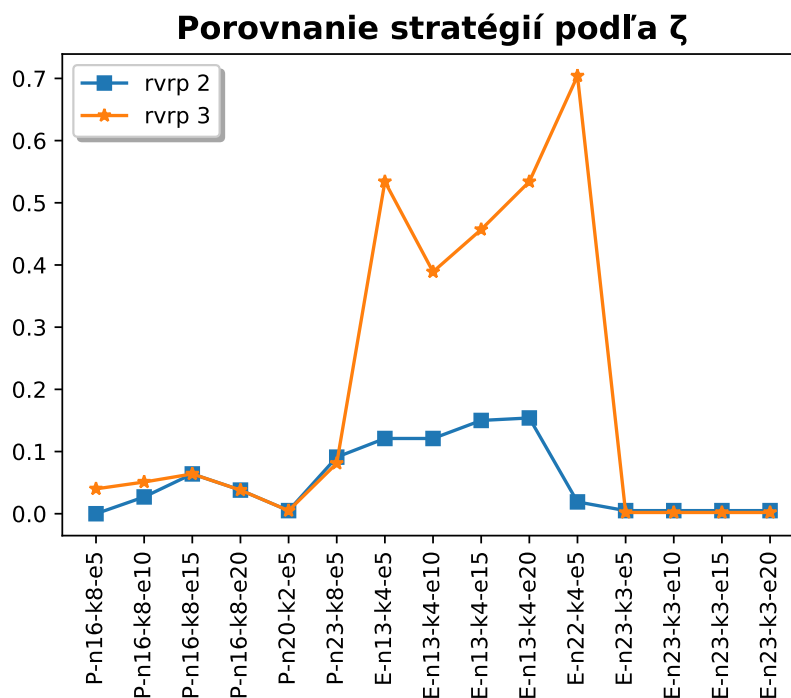
Najhoršie výsledky z hľadiska ceny dosiahla stratégia *rvrp 3*, ktorá bola najhoršia vo všetkých prípadoch okrem dvoch. Stratégia *rvrp 2* bola v porovnaní so stratégiou *rvrp 1* lepšia v deviatich prípadoch, kým *rvrp 1* v ôsmich z dvadsiatich dvoch prípadov riešených heuristicky. Vo zvyšných prípadoch boli rovnaké. Môžeme teda hovoriť o približne rovnakej úspešnosti. Riešenia s relatívnym zvýšením ceny väčším ako 50% sme dostali len stratégiou *rvrp 3*, a to v dvoch prípadoch. V riešeniach stratégií *rvrp 1* a *rvrp 2* sa cena zvýšila nanajvýš o 15,4%.

Cena versus neuspokojené požiadavky

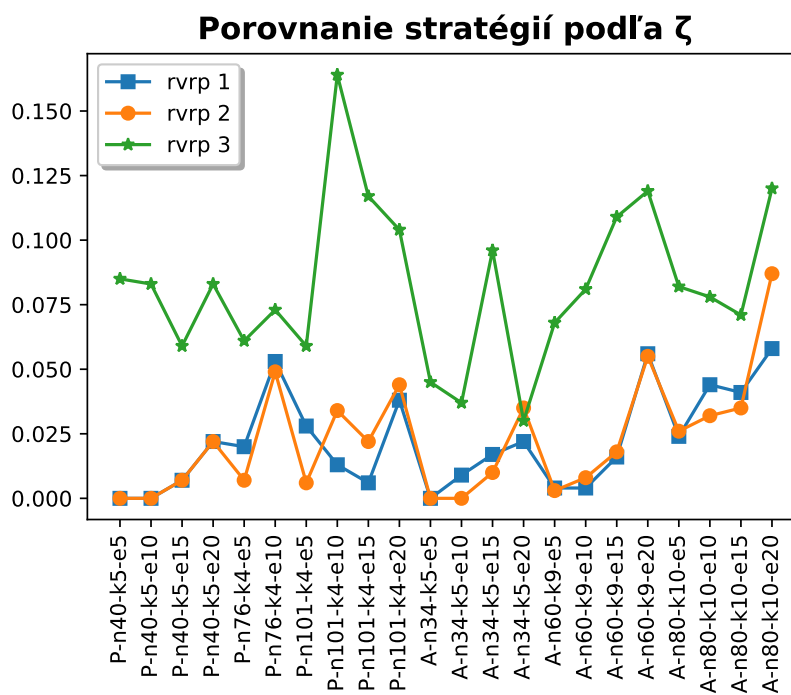
Pri rozhodovaní, ktoré riešenie je lepšie, treba nájsť vyvážený kompromis medzi týmito dvomi hľadiskami. Napríklad pri inštalácii E-n22-k4-e15 stratégia *rvrp 2* našla riešenie s cenou 619 a množstvom neuspokojených požiadaviek 13, naopak stratégia *rvrp 3* našla riešenie s väčšou cenou 669, ale so žiadnymi neuspokojenými požiadavkami. Na jednej strane, nižšia cena znamená väčší aktuálny profit, na druhej strane menej neuspokojených požiadaviek vedie k väčšej spokojnosti zákazníkov a tým aj k vyššiemu profitu v budúcnosti.

(a) Zníženie relatívneho množstva neuspokojených požiadaviek γ (b) Relatívne zvýšenie ceny ζ

Obr. 4.11: Inštancie s neprípustným maximálnym scenárom



(a) Inštancie riešené exaktne



(b) Inštancie riešené heuristicky

Obr. 4.12: Inštancie s prípustným maximálnym scenárom

4.5.5 Záver

Cieľom nami navrhnutej robustnej optimalizácie bolo nájsť riešenie s minimálnym množstvom neuspokojených požiadaviek a s minimálnou cenou.

Výsledky experimentov ukázali, že stratégia maximálneho scenára *rvrp 1* nájde robustné riešenie len v prípade, ak maximálny scenár je prípustný a v opačnom prípade zlyhá. Stratégia najhoršieho prípustného scenára *rvrp 2* generuje rovnaké výsledky ako *rvrp 1* a navyše sa dokáže vysporiadať aj s prípadmi, keď je *rvrp 1* neúspešná.

Aj stratégia najmenej nutnej kapacity *rvrp 3* nájde prijateľné riešenie bez ohľadu na to, či je maximálny scenár prípustný alebo nie. Avšak riešenia tejto stratégie *rvrp 3* vo väčšine prípadov vedú k najväčšiemu zvýšeniu ceny.

Na základe analýzy našich experimentálnych výsledkov môžeme konštatovať, že z troch stratégií na riešenie RVRP s neistými požiadavkami je stratégia najhoršieho prípustného scenára *rvrp 2* najschopnejšia generovať riešenia, ktoré minimalizujú množstvo neuspokojených požiadaviek pričom ich cena je len o málo vyššia.

Kapitola 5

ZÁVER

V našej práci sme sa zaoberali riešením distribučných úloh s neistými parametrami pomocou robustnej optimalizácie. Vzhľadom na široký rozsah tejto problematiky sme sa zamerali na kapacitnú okružnú dopravnú úlohu s neistými požiadavkami zákazníkov. Cieľom bolo navrhnúť robustné modely na riešenie tejto úlohy.

V úvode sme opísali teoretický základ s účelom definovať a vysvetliť základné pojmy skúmanej problematiky. Druhá kapitola obsahuje prehľad exaktných a heuristických metód používaných pri riešení CVRP. Osobitná časť tejto kapitoly je venovaná CVRP s neistými parametrami a zahŕňa prehľad metód so stochastickým i robustným prístupom. Zdrojom bola prevažne zahraničná literatúra, nakoľko literatúra venovaná problematike robustnej optimalizácie v slovenskom, resp. českom jazyku prakticky neexistuje.

Ďalej nasleduje opis zvolenej metodiky práce a metód skúmania. Opísali sme spôsob modelovania neistých požiadaviek zákazníkov pomocou diskretných scenárov a dve stratégie na nájdenie najhoršieho scenára. Navrhli sme vhodnú deterministickú formuláciu CVRP, ktorú sme potom použili pri odvodení robustnej formulácie. Taktiež sme predstavili dve exaktné a jednu heuristickú metódu na riešenie CVRP, aby sme ich neskôr využili pri riešení robustnej CVRP. Okrem toho sme objasnili kritériá, podľa ktorých sme posudzovali úspešnosť navrhnutých robustných stratégií. Táto kapitola obsahuje aj špecifikáciu testovacích inštancií a softvérových nástrojov použitých pri našich experimentoch.

V kapitole venovanej výskumu sme prezentovali náš prístup k riešeniu robustnej CVRP s neistými požiadavkami zákazníkov. Najskôr sme navrhli a implementovali nový model

CVRP 2 na riešenie CVRP založený na tokovej formulácii, ktorý sa vyznačuje polynomiálnym počtom anticyklických podmienok. Tento model je formuláciou vhodnou pre odvodenie takej formulácie RVRP, ktorú nebude príliš zložité riešiť.

Aby sme boli schopní v prijateľnom čase exaktne riešiť aj úlohy s väčším počtom zákazníkov, vyvinuli sme efektívnu exaktnú metódu CVRP 3 postavenú na koncepte k -optimálnosti. Naša metóda iteratívne zlepšuje počiatočné prípustné riešenie nahradením niektorých hrán inými. Zlepšenie riešenia pritom dosiahne exaktným riešením jednoduchšieho problému, vďaka čomu nájde optimálne riešenie omnoho rýchlejšie než CVRP 2. Navyše, ak by bol výpočet príliš zdĺhavý, môžeme ho prerušiť a získame tak k -optimálne riešenie.

Vzhľadom na to, že CVRP patrí do skupiny NP-t ťažkých problémov, nie je možné pomocou tohto modelu exaktne riešiť úlohy s veľkým rozsahom, a preto sa na ich riešenie využívajú heuristické a metaheuristické metódy. My sme na tento účel navrhli a experimentálne overili paralelný mikrogenetický algoritmus $p\mu$ GA, ktorý profituje z výhod mikrogenetického algoritmu a paralelného výpočtu. Pri paralelizácii mikrogenetického algoritmu sme vychádzali z princípu hrubozrného modelu, ale namiesto podpopulácií veľkej populácie sme použili mikropopulácie a proces migrácie sme nahradili dosadením nasadeného jedinca.

Nový prístup k riešeniu CVRP predstavuje návrh lexikografického modelu LCVRP, ktorého cieľom je nájsť riešenie s minimálnou cenou, pričom bude využitá len najmenšia nutná kapacita vozidiel. Takéto riešenie má síce väčšiu cenu v porovnaní s cenou riešenia pri využití plnej kapacity, ale na druhej strane nevyužitá kapacita vozidiel môže slúžiť ako rezerva pre prípad, že dôjde k neočakávanému nárastu požiadaviek zákazníkov. Vďaka tomu je riešenie LCVRP do určitej miery robustné.

V poslednej fáze výskumu sme sa venovali robustnej CVRP s neistými požiadavkami zákazníkov, ktorej cieľom je nájsť také riešenie, ktoré bez ohľadu na to, aký scenár nastane, uspokojí všetkých zákazníkov. Niekedy však také riešenie neexistuje, a preto sme v našom prístupe k riešeniu tejto úlohy hľadali riešenie s minimálnymi neuspokojenými požiadavkami a minimálnou cenou. Navrhli sme tri robustné stratégie na riešenie robustnej CVRP s neistými požiadavkami zákazníkov – stratégiu maximálneho scenára *rvrp 1*, stratégiu najhoršieho prípustného scenára *rvrp 2* a stratégiu najmenšej nutnej kapacity *rvrp 3*. Experimentálne výsledky dosiahnuté aplikáciou týchto stratégií sme porovnali z hľadiska ceny i z hľadiska neuspoko-

jených požiadaviek. Na základe analýzy výsledkov bola z hľadiska vyváženého kompromisu medzi cenou a neuspokojenými požiadavkami najúspešnejšou stratégia *rvrp 2*.

Na záver zhrnieme vedecký a praktický prínos práce.

5.0.1 Vedecký prínos práce

Za hlavný vedecký prínos tejto práce možno považovať:

- prehľad súčasného stavu riešenia distribučných úloh s využitím robustnej optimalizácie,
- teoretický návrh prístupu k riešeniu problematiky robustných modelov,
- softvérové riešenia a verifikáciu navrhnutých prístupov,
- inovatívne metódy na riešenie deterministickej i robustnej CVRP, využiteľné aj v iných oblastiach kombinatorickej optimalizácie:
 - novú iteratívnu exaktnú metódu na riešenie CVRP,
 - efektívny paralelný mikrogenetický algoritmus na riešenie CVRP,
 - lexikografický model CVRP,
 - nový prístup k riešeniu robustnej CVRP s neistými požiadavkami zákazníkov.

5.0.2 Praktický prínos práce

Navrhnuté modely nájdu praktické využitie v oblastiach ako sú napríklad:

- miestna preprava tovaru (zásobovanie),
- kuriérske služby,
- mestský zber komunálneho a separovaného odpadu,
- mobilné gastronomické služby (cateringový servis),
- školská autobusová doprava,
- zvoz recyklovateľných obalov.

Literatúra

- [1] ALAJMI, A., WRIGHT, J.: Selecting the most efficient genetic algorithm sets in solving unconstrained building optimization problem. *International Journal of Sustainable Built Environment*, vol. 3, no. 1, 18–26, 2014.
- [2] ALBA, E., DORRONSORO, B.: Solving the vehicle routing problem by using cellular genetic algorithms, in: *Evolutionary Computation in Combinatorial Optimization (EvoCOP)*, Lecture Notes in Computer Science, vol. 3004, Springer-Verlag, Berlin, 11–20, 2004.
- [3] AUGERAT, P., BELENGUER, J., BENAVENT, E., CORBERÁN, A., NADDEF, D., RINALDI, G.: Computational results with a branch and cut code for the capacitated vehicle routing problem. Tech. Rep. 949-M, Universite Joseph Fourier, Grenoble, France, 1995.
- [4] BALDACCIO R., HADJICONSTANTINO E., MIGNOZZI A.: An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation. *Oper. Res.* 52:723–738, 2004.
- [5] BEASLEY, J. E.: Route first-cluster second methods for vehicle routing. *Omega*, 11(4), 403–408, 1983.
- [6] BEN-TAL, A., NEMIROVSKI, A.: Robust convex optimization. *Mathematics of Operations Research*, 23(4), 769–805, 1998.
- [7] BEN-TAL, A., NEMIROVSKI, A.: Robust Solutions of Uncertain Linear Programs, *OR Letters* v. 25, 1-13, 1999.

- [8] BERTSIMAS, D.: Probabilistic combinatorial optimization problems. Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Mathematics, 1988.
- [9] BERTSIMAS, D., BROWN, D. B., CARAMANIS, C.: Theory and applications of robust optimization. *SIAM Rev.* 53(3), 464–501, 2011.
- [10] BERTSIMAS, D., SIM, M.: Robust discrete optimization and network flows. *Mathematical Programming Series B*, 98(1-3), 49–71, 2003.
- [11] BERTSIMAS, D., SIMCHI-LEVI, D.: A new generation of vehicle routing research: robust algorithms, addressing uncertainty. *Operations Research*, 44(2), 286–304, 1996.
- [12] BERTSIMAS, D.: A vehicle routing problem with stochastic demand. *Operations Research*, 40(3), 574–585, 1992.
- [13] BORČINOVÁ, Z.: Two models of the capacitated vehicle routing problem. In: *Croatian operational research review*. - ISSN 1848-0225. - Vol. 8, no. 2, s. 463-469, 2017.
- [14] BORČINOVÁ, Z.: Solving the capacitated vehicle routing problem using a parallel micro genetic algorithm. In: *2018 IEEE Workshop on Complexity in Engineering (COMPENG)* - ISBN 978-1-5386-5338-8. - s. [1-6], 2018.
- [15] BORČINOVÁ, Z., PEŠKO, Š.: New exact iterative method for the capacitated vehicle routing problem. In: *Communications : scientific letters of the University of Žilina*. - ISSN 1335-4205. - Vol. 18, no. 3, s. 19-21, 2016.
- [16] BORČINOVÁ, Z., PEŠKO, Š.: The lexicographical capacitated vehicle routing problem. In: *SOLI 2017 : IEEE international conference on Service operations and logistics, and informatics : September 18-20, 2017 Bari, Italy* - ISBN 978-1-5090-5847-1. - [4] s., 2017.
- [17] CLARKE, G., WRIGHT, J. W.: Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4), 568–581, 1964.

- [18] COCO, A. A., SOLANO-CHARRIS, E. L., SANTOS, A. C., PRINS, C., NORONHA, T.: Robust optimization criteria: state-of-the-art and new issues. Technical Report UTT-LOSI-14001. ISSN:2266-5064. Université de Technologie de Troyes, 2014a.
- [19] CRAINIC, T.: Parallel solution methods for vehicle routing problems. In: Golden, B., Raghavan, S., Wasil, E. eds., *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York, 171–198, 2008.
- [20] DANTZIG, G. B.: Linear Programming under uncertainty. *Management Science*, 1(3-4), 197–206, 1955.
- [21] DANTZIG, G. B., RAMSER, J. H.: The truck dispatching problem. *Management Science*, 6(1), 80–91, 1959.
- [22] DORIGO, M., BLUM, C.: Ant colony optimization theory: A survey. *Theoret Comput Sci*, 344 (2–3), pp. 243–278, 2005.
- [23] DESSOUKY, M., ORDÓÑEZ, F., JIA, H., SHEN, Z. Rapid distribution of medical supplies. *Patient flow: Reducing delay management in health care systems*. Springer, USA, 309–339, 2005.
- [24] DRESHER, M.: *Games of Strategy*. Prentice-Hall, 1961.
- [25] ERBAO, C., MINGYONG, L.: A hybrid differential evolution algorithm to vehicle routing problem with fuzzy demands. *Journal of Computational and Applied Mathematics*. 231. 302-310, 2009.
- [26] ERBAO, C., MINGYONG, L., HONGMING, Y.: Open vehicle routing problem with demand uncertainty and its robust strategies. *Expert Systems with Applications: An International Journal*. 41. 3569-3575, 2014.
- [27] FUKASAWA, R., LONGO, H., LYSGAARD, J., ARAGÃO, M. P., REIS, M., UCHOA, E., WERNECK, R. F.: Robust Branch-and-Cut-and-Price for the Capacitated Vehicle Routing Problem. *Mathematical Programming Series A*, 106(3), 491–511, 2006.

- [28] GABREL, V., MURAT, C., THIELE, A.: La pw-robustesse: pourquoi un nouveau critère de robustesse et comment l'appliquer? In: 14ème congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF), 2013.
- [29] GENDREAU, M., LAPORTE, G., SEGUIN, R.: An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science*, 29(2), 143–155, 1995.
- [30] GENDREAU, M., LAPORTE, G., SEGUIN, R.: Stochastic vehicle routing. *European Journal of Operational Research*, 88(1), 3–12, 1996.
- [31] GENDREAU, M., POTVIN, J. Y.: *Handbook of Metaheuristics*, Second Edition, Springer, New York, 2010
- [32] GILLETT, B. E., MILLER, L. R.: A heuristic algorithm for the Vehicle-Dispatch problem. *Operations Research*, 22(2), 340–349, 1974.
- [33] GOUNARIS, C. E., WIESEMANN, W., FLOUDAS, C. A.: The Robust Capacitated Vehicle Routing Problem Under Demand Uncertainty. *Operations Research*, 61(3), 677–693, 2013.
- [34] GOUNARIS, E. C., REPOUSSIS, P. P., TARANTILIS, C. D., WIESEMANN, W., FLOUDAS, C. A.: An Adaptive Memory Programming framework for the Robust Capacitated Vehicle Routing Problem. *Transportation Science*. doi:10.1287/trsc.2014.055, 2014.
- [35] GOLDBERG, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, MA, 1989.
- [36] GOLDBERG, D. E., KORB, B., DEB, K.: Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3, 493–530, 1990.
- [37] GOLDBERG, D. E., LINGLE, R.: Alleles Loci and the TSP. *Proceedings of the First International Conference on Genetic Algorithms and Their Application*, Lawrence Erlbaum, New Jersey, 54–159, 1985.

- [38] GROËR, Ch., GOLDEN, B., WASIL, E. A.: A Parallel Algorithm for the Vehicle Routing Problem. *INFORMS Journal on Computing*, 23, 315–330, 2011.
- [39] CHEPURI, K., HOMEM de MELLO, H.. Solving the vehicle routing with stochastic demands using the cross entropy method. *Annals of Operations Research*, 135:153–181, 2005.
- [40] CHRISTIANSEN, C., LYSGAARD, J.: A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands, *Operation Research Letters*, 35(6), 773–781, 2007.
- [41] HAN, J., LEE, C., PARK, S.: A Robust Scenario Approach for the Vehicle Routing Problem with Uncertain Travel Times. *Transportation Science*, 48(3), 373–390, 2013 .
- [42] HOLLAND, H.: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975.
- [43] CHRISTOFIDES, N., EILON, S.: An algorithm for the vehicle dispatching problem. *Journal of the Operational Research Society*, Vol. 20, No. 3, pp. 309–318, 1969.
- [44] CHRISTOFIDES, N., MIGNOZZI, A., TOTH, P.: The vehicle routing problem. Pages 315–338 of: *Combinatorial Optimization*. Wiley, Chichester, 1979.
- [45] JANÁČEK, J., JÁNOŠÍKOVÁ, L., KOHÁNI, M.: *Modelovanie a optimalizácia*, EDIS vydavateľstvo ŽU, 2013.
- [46] JUAN, A., GRASMAN, S., FAULIN, J., RIERA, D., MÉNDEZ, C., RUIZ, B.: Applying Simulation and Reliability to Vehicle Routing Problems with Stochastic Demands. *CEUR Workshop Proceedings*, 589, 2009.
- [47] KALAŘ, R., LAMBORAY, C., VANDERPOOTEN, D.: Lexicographic α -robustness: An alternative to min-max criteria. *European Journal of Operational Research*, 220(3), 722–728, 2012.

- [48] KARA, I.: Tightening Bounding Constraints of the Miller-Tucker-Zemlin Based Formulation of the Capacitated Vehicle Routing Problems and Some Extensions, Proc. of the 2nd intern. conference on Manufacturing Engineering, Quality and Production Systems, edited by C. Panait et al., WSEAS Press, Constantza, pp.137-142, 2010.
- [49] KENYON, A. S., MORTON, D.: Stochastic vehicle routing with random travel times. *Transportation Science*, 37(1), 69–82, 2003.
- [50] KOUVELIS, P., YU, G.: *Robust discrete optimization and its applications*. Kluwer Academic Publishers, Norwell, MA, 1997.
- [51] KRISHNAKUMAR, K.: Micro-Genetic Algorithms for Stationary and Non-Stationary Function Optimization. *SPIE Proceedings: Intelligent Control and Adaptive Systems*, Vol. 1196, 289–296, 1989.
- [52] L. GORISSEN, B., YANIKOĞLU, İ, den HERTOĞ, D.: A Practical Guide to Robust Optimization. *Omega*. 53, 124-137, 2015.
- [53] LAPORTE, G.: Fifty Years of Vehicle Routing. *Transportation Science*, 43(4), 408–416, 2009.
- [54] LAPORTE, G., LOUVEAUX, F., HAMME, L. V.: An Integer L-Shaped Algorithm for the Capacitated Vehicle Routing Problem with Stochastic Demands. *Operations Research*, 50(3), 415–423, 2002.
- [55] LEE, C., LEE, K., PARK, S.: Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the OR Society*, 63, 1294–1306, 2012 .
- [56] LIN, S., KERNIGHAN, B. W.: An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Operations Research*. 21 (2): 498–516, 1973.
- [57] LI, X., TIAN, P., LEUNG, S. C.H.: Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm, *International Journal of Production Economics*, Elsevier, vol. 125(1), 137-145, 2010.

- [58] LYSGAARD, J., LETCHFORD, A. N., EGGLESE, R. W.: A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2), 423–445, 2004.
- [59] MARKOVIC, H., CAVAR, I., CARIC, T.: Using data mining to forecast uncertain demands in stochastic vehicle routing problem. In *Proceedings of the 13th International Symposium on Electronics in Transport (ISEP)*, Slovenia, 2005.
- [60] MARTELLO, S., TOTH, P.: *Knapsack Problems: Algorithms and Computer Implementations*, Wiley, Chichester, 1990.
- [61] MARTIN, W. N., LIENING, J., COHOON, J. P.: Island (migration) models: Evolutionary algorithms based on punctuated equilibria. *Evolutionary Computation 2: Advanced Algorithms and Operators*, T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds. London, U.K.: IOP Publishing, vol. 2, 101–124, 2000.
- [62] MESTER, D., BRÄYSY, O.: Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers & Operations Research*, 34(10), 2964–2975, 2007.
- [63] MOGHADDAM, B. F., RUIZ, R., SADJADI, S. J.: Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Computers & Industrial Engineering*, 62(1), 306–317, 2012 .
- [64] MOLE, R. H., JAMESON, S. R.: A Sequential Route-Building Algorithm Employing a Generalised Savings Criterion. *Operational Research Quarterly*, 27(2), 503–511, 1976.
- [65] NEUMANN, J. V.: Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1), 295–320, 1928.
- [66] OCHI, L. S., VIANNA, D. S., DRUMMOND, L. M. A., VICTOR, A. O.: A Parallel Evolutionary Algorithm for the Vehicle Routing Problem with Heterogeneous Fleet. *Future Generation Computer Systems*, 14(3): 285–292, 1998.
- [67] ORDÓÑEZ, F.: *Robust Vehicle Routing*. Chap. 8, pages 153–178, 2010.

- [68] OSMAN, I.: Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*, 41(4), 421–451, 1993.
- [69] PALÚCH, S., PEŠKO, Š.: *Kvantitatívne metódy v logistike*, EDIS vydavateľstvo ŽU, 2006.
- [70] PECIN, D., PESSOA, A., POGGI, M., UCHOA, E.: Improved Branch-Cut-and-Price for Capacitated Vehicle Routing. Pages 393–403 of: *Integer Programming and Combinatorial Optimization*. Lecture Notes in Computer Science, vol. 8494. Springer International Publishing, 2014.
- [71] PRINS, C.: A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12), 1985–2002, 2004.
- [72] PRINS, C., LACOMME, P., PRODHON, C.: Order-first split-second methods for vehicle routing problems: A review. *Transportation Research Part C*, 40, 179–200, 2014.
- [73] RIEMANN, M., DOERNER, K., HARTL, R. F.: Analyzing a Unified Ant System for the VRP and Some of its Variants. *Applications of Evolutionary Computing*, 2611:300–310, 2003.
- [74] ROY, B.: Robustness in operational research and decision aiding: A multi-faceted issue. *European Journal of Operational Research*, 200(3), 629–638, 2010.
- [75] SECOMANDI, N., MARGOT, F.: Reoptimization Approaches for the Vehicle-Routing Problem with Stochastic Demands. *Operations Research*, 57(1), 214–230, 2009.
- [76] SEKAJ, I.: *Evolučné výpočty a ich využitie v praxi*, IRIS, Bratislava, 2005.
- [77] SOLANO-CHARRIS, E. L., PRINS, C., SANTOS, A. C.: Local Search Based Metaheuristics for the Robust Vehicle Routing Problem with Discrete Scenarios. *Applied Soft Computing*, 32, 518–531, 2015.
- [78] SUN, L.: A New Robust Optimization Model for the Vehicle Routing Problem with Stochastic Demands. *Journal of Interdisciplinary Mathematics*, 17(3), 287–309, 2014.

- [79] SUN, L., WANG, B.: Robust Optimisation Approach for Vehicle Routing Problems with Uncertainty, *Mathematical Problems in Engineering*, vol. 2015, Article ID 901583, 8 pages, 2015.
- [80] SUNGUR, I., ORDÓÑEZ, F., DESSOUKY, M.: A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 40(5), 509–523, 2008 .
- [81] SUNGUR, I., REN, Y., ORDÓÑEZ, F., DESSOUKY, M., ZHONG, H.: A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science*, 44(2), 193–205, 2010.
- [82] TOKLU, N. E., MONTEMANNI, R., GAMBARDELLA, L. M.: An ant colony system for the capacitated vehicle routing problem with uncertain travel costs. Pages 32–39 of: *IEEE Symposium on Swarm Intelligence (SIS)*, 2013.
- [83] TOTH, P., VIGO, D. : *Vehicle routing: Problems, Methods and Applications*, Second Edition. Philadelphia: SIAM, 2014.
- [84] TOTH, P., VIGO, D.: Models, Relaxations and Exact Approaches for Capacitated Vehicle Routing Problem, *Discrete Applied Mathematics*, 123: 487-512, 2002.
- [85] YANG, W., MATHUR, K., BALLOU, R.: Stochastic vehicle routing problem with restocking. *Transportation Science*, 34(1):99–112, 2000.
- [86] WATERS, C.: Vehicle-scheduling problems with uncertainty and omitted customer. *Journal of the Operational Research Society*, 4(12), 1099–1108, 1989 .
- [87] ZACHARIADIS, E. E., KIRANOUDIS, C. T.: A strategy for reducing the computational complexity of local search-based methods for the vehicle routing problem. *Computers & operations research*, 37 (12), 2089-2105, 2010.
- [88] *The Python Language Reference, 1990 - 2016*, Python Software Foundation, <http://docs.python.org/3/>.

- [89] Gurobi Optimizer Reference Manual, version 6.5, 2016, Gurobi Optimization, Inc., <http://www.gurobi.com>.
- [90] Branch Cut and Price Resource Web, Vehicle Routing Data Sets USA, <http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/>.
- [91] Matplotlib 3.0, 2002 - 2018, The Matplotlib development team, <https://matplotlib.org/>.
- [92] Message Passing Interface (MPI) standard for the Python programming language, <https://bitbucket.org/mpi4py/mpi4py>.
- [93] NetworkX 2.2, 2004 - 2018, NetworkX Developers, <https://networkx.github.io/documentation/stable/>

Vlastné publikácie

BORČINOVÁ, Z.: Two models of the capacitated vehicle routing problem. In: Croatian operational research review. - ISSN 1848-0225. - Vol. 8, no. 2, s. 463-469, 2017.

BORČINOVÁ, Z.: Solving the capacitated vehicle routing problem using a parallel micro genetic algorithm. In: 2018 IEEE Workshop on Complexity in Engineering (COMPENG) - ISBN 978-1-5386-5338-8. - s. [1-6], 2018.

BORČINOVÁ, Z., PEŠKO, Š.: New exact iterative method for the capacitated vehicle routing problem. In: Communications : scientific letters of the University of Žilina. - ISSN 1335-4205. - Vol. 18, no. 3, s. 19-21, 2016.

BORČINOVÁ, Z., PEŠKO, Š.: The lexicographical capacitated vehicle routing problem. In: SOLI 2017 : IEEE international conference on Service operations and logistics, and informatics : September 18-20, 2017 Bari, Italy - ISBN 978-1-5090-5847-1. - [4] s., 2017.