

**M A S A R Y K O V A
U N I V E R Z I T A**

FAKULTA INFORMATIKY

**Generovanie vstupu pre flexibilné
smerovanie vozidiel**

Diplomová práca

PETER GÁBOR

Brno, jeseň 2024

**M A S A R Y K O V A
U N I V E R Z I T A**

FAKULTA INFORMATIKY

**Generovanie vstupu pre flexibilné
smerovanie vozidiel**

Diplomová práca

PETER GÁBOR

Vedúci práce: Mgr. Václav Sobotka

Department of Machine Learning and Data Processing

Brno, jeseň 2024



Vyhlásenie

Vyhlasujem, že táto diplomová práca je mojím pôvodným autorským dielom, ktoré som vypracoval samostatne. Všetky zdroje, pramene a literatúru, ktoré som pri vypracovaní používal alebo z nich čerpal, v práci riadne citujem s uvedením úplného odkazu na príslušný zdroj.

Prehlásenie o použití umelej inteligencie

Pri príprave tejto práce som za účelom rýchlejšieho vytvárania kódu použil ChatGPT Copilot.

Peter Gábor

Vedúci práce: Mgr. Václav Sobotka

Podakovanie

Na tomto mieste by som sa chcel poďakovať vedúcemu diplomovej práce Mgr. Václavovi Sobotkovi za cenné rady, trpezlivosť a odborné vedenie pri vytváraní tejto práce. Taktiež by som sa chcel poďakovať konzultantke doc. Mgr. Hanke Rudovej za doprevádzanie a firme Wereldo za poskytnuté dáta a námety. Vďaka patrí tiež MetaCentru za poskytnutie výpočtových zdrojov. Výpočtové prostriedky boli poskytnuté projektom e-INFRA CZ (ID:90254) podporovaným Ministerstvom školstva, mládeže a telovýchovy Českej republiky.

Zhrnutie

Efektívne plánovanie trás a celková logistika predstavuje pre prepravné spoločnosti poprednú prioritu, ktorá z veľkej miery určuje mieru optimálneho využitia zdrojov, s ktorými spoločnosť operuje, i spokojnosť zákazníkov. Táto práca, ktorá vznikla na podnet prepravnej spoločnosti Wereldo, skúma spôsoby, akými je možné obohatiť statické plánovanie tak, aby sa stalo viac flexibilným, schopným prijímať nové dynamické objednávky prichádzajúce počas dňa. Konkrétne do statického plánu integrujem dummy zákazníkov reprezentujúcich strategicky alokované zdroje, ktoré slúžia pre obsluhu možných objednávok prijatých počas dňa. Pre účely tejto práce som zhrnul aktuálne prístupy k riešeniu Vehicle Routing Problem (VRP). Na základe analýzy reálnych dát som naimplementoval generátor inštancií statického MAVRP. Navrhol som viacero stratégií pre generovanie dummy zákazníkov. Naimplementoval som event-driven systém, ktorý simuluje realizáciu plánu trás. Dynamické objednávky, ktoré počas dňa prichádzajú sú stotožňované s dummy zákazníkmi. Tento systém poslužil ako prostredie pre testovanie efektívnosti navrhnutých stratégií. Výsledky testov ukázali, že vloženie dummy zákazníkov do statického plánu trás výrazne zvyšuje jeho flexibilitu a schopnosť prijímať dynamické objednávky. Tento prístup tak zachováva výhody statického plánovania a súčasne otvára cestu pre dynamické vstupy.

Kľúčové slová

preprava, vehicle routing problem, dummy zákazník, dynamická objednávka, logistika, generátor inštancií, Wereldo.com, plán trás

Obsah

Úvod	1
1 Uvedenie do problematiky	2
2 State of the art	6
2.1 Static vehicle routing problem	6
2.1.1 Klasické heuristiky	6
2.1.2 Metaheuristiky	7
2.1.3 Univerzálne algoritmy	9
2.2 Dynamic vehicle routing problem	12
2.2.1 Metódy vkladania	12
2.2.2 Metódy reoptimalizácie	13
2.2.3 Predvídavé prístupy	14
3 Generátor inštancií	17
3.1 Statické inštancie	17
3.1.1 Geografická poloha	18
3.1.2 Hmotnosť	19
3.1.3 Objem	19
3.1.4 Výška	21
3.1.5 Čas obsluhy	21
3.1.6 Časové okná	22
3.2 Dynamické scenáre	22
4 Popis implementácie	25
4.1 Stratégie generovania dummy zákazníkov	26
4.1.1 Priestorová rovina	26
4.1.2 Časová rovina	27
4.1.3 Rovina alokácie zdrojov	28
4.2 Simulácia realizácie plánu trás počas dňa	30
4.2.1 Lokálne udalosti	31
4.2.2 Globálne udalosti	31
5 Experimenty	33
5.1 Datasetsy	33
5.2 Postup	35

5.3	Ladenie parametrov	36
5.4	Experiment	38
6	Záver	49
6.1	Perspektívy ďalšieho výskumu	50
	Bibliografia	51
A	Súbory	57
B	Podrobné výsledky z experimentu	58

Úvod

Kvalitná logistika a efektívne plánovanie trás predstavuje pre prepravné spoločnosti kritický bod, ktorý častokrát rozhoduje o ich úspechu a presadení sa na trhu. Obzvlášť v dnešnej dobe, kedy sa obchod široko otvára pre zákazníkov v blízkom okolí i zákazníkov v inom štáte či dokonca na inom kontinente, je dôležité premyslene plánovať. Obzvlášť keď má zákazník častokrát na výber mnohé možnosti prepravy. Navyše dynamika dnešnej doby si vyžaduje schopnosť prijímať nové nečakané objednávky a najrôznejšie požiadavky ako i rýchlo reagovať na zmeny. V tejto oblasti vyniká *vehicle routing problem*, ktorý sa prednostne zaoberá optimalizáciou trás pre množinu vozidiel obsluhujúcich plejádu zákazníkov. Do hry vstupujú viaceré obmedzenia ako sú časové okná, kapacita vozidiel či prioritné objednávky. Desaťročia výskumu a zdokonaľovania algoritmov priniesli veľmi spoľahlivé a efektívne riešenia pre statický deterministický VRP. Naproti tomu, dynamická povaha reálnych inštancií problémov, ktorá prináša nové objednávky a nečakaný vývoj situácie, predstavuje nemalú výzvu pre existujúce metódy.

V rámci tejto diplomovej práce spolupracujem s prepravnou firmou Wereldo, ktorá pre plánovanie trás používa statický deterministický algoritmus. Jeho nevýhodou je neschopnosť flexibilne reagovať na nové informácie, najmä v podobe dopredu neznámych objednávok od zákazníkov prichádzajúcich počas dňa. Vo svojej práci sa zameriavam na metódy, akými je možné upraviť statické plánovanie, aby bolo viac disponované prijímať dynamicky vstupujúcich zákazníkov. Konkrétne do statického plánu integrujem tzv. *dummy zákazníkov* reprezentujúcich potenciálnych zákazníkov, ktorí sa počas dňa môžu objaviť. Jedná sa o strategické body na trase, do ktorých je neskôr možné adaptívne vložiť nové objednávky.

V prvej kapitole uvádzam základné pojmy týkajúce sa VRP. Druhá kapitola predstavuje aktuálny *state of the art* v oblasti. Tretia kapitola je zameraná na popis generátora inštancií VRP, ktorý som naprogramoval. Následne vo štvrtej kapitole popisujem implementáciu stratégie integrácie dummy zákazníkov ako aj simulácie realizácie plánu trás počas dňa. Nakoniec piata kapitola sa venuje experimentálnej časti práce, kde testujem kvalitu a vhodnosť navrhnutých stratégií.

1 Uvedenie do problematiky

Vehicle Routing Problem (VRP) prvýkrát predstavili akademickému svetu Dantzig a Ramser v publikácii *The truck dispatching problem* v roku 1959 [1]. Pre danú množinu vozidiel problém spočíval v nájdení optimálnych trás, ktoré by spĺňali požiadavky prijaté od zákazníkov viazané na dané geografické lokality. VRP generalizuje *Travelling Salesman Problem* (TSP), ktorý pre danú množinu bodov a vzdialeností medzi nimi spočíva v nájdení trasy, ktorá začína a končí v tom istom bode a súčasne navštívi každý bod práve raz. Nájdenie optimálneho riešenia je ťažký optimalizačný problém. Použitie exaktných metód hľadajúcich preukázateľne optimálne riešenie je z tohto dôvodu obmedzené na relatívne malé vstupné inštancie. Pre praktické aplikácie sa preto často využívajú skôr heuristické a metaheuristické prístupy, nakoľko skutočné problémy sú často rozsiahle, pokrývajúce veľké množstvo zákazníkov k obsluženiu, zahŕňajúce celú radu často nehomogénnych vozidiel, dep i rôznorodých obmedzení.

Do VRP vstupujú viaceré entity, ktoré rôznym spôsobom ovplyvňujú celú problematiku: spôsob dopravy (cestná, letecká, ...), množina dopravných prostriedkov, s ktorými sa pracuje (ich množstvo, homogenita, resp. heterogenita, kapacitné obmedzenia a iné parametre), časové okná, možnosť odmietnuť zákazníka, účelová funkcia (priority pri optimalizácii) a iné entity. Z tohto dôvodu sa VRP neobjavuje v jednej jedinej podobe, ale môže sa objavovať v rôznych podobách. Podobách, ktoré odzrkadľujú akési podmnožiny entít, ktoré v danom prípade vstupujú do hry, resp. ktoré nie sú zanedbateľné. Medzi kanonické varianty VRP patria Capacitated Vehicle Routing Problem (CVRP), Vehicle Routing Problem with Time Windows (VRPTW) a Pickup and Delivery Problem with Time Windows (PDPTW).

Capacitated vehicle routing problem: Vozidlá majú obmedzenú maximálnu kapacitu, pričom požiadavky na trase nesmú dohromady túto kapacitu presiahnuť. Navyše zákazníci majú rôzne vysoké požiadavky.

Vehicle routing problem with time windows: Táto podoba sa obvykle viaže taktiež s obmedzenou kapacitou vozidla a predstavuje tak generalizáciu CVRP. Jednotlivé zastávky majú stanovené časové okná, t. j. časové intervaly, v ktorých je možné na danej zastávke naložiť či vyložiť náklad. Naloženie, resp. vyloženie nákladu trvá istý čas,

ktorý sa označuje ako *čas obsluhy* (service time). Do časového plánu vstupuje aj prípadné čakanie v prípade, že vozidlo dorazí na zastávku pred začiatkom časového okna. Ohraničenie časového okna môže byť *tvrdé* (hard time window), t. j. musí byť splnené, alebo *mäkké* (soft time window), kedy ohraničenie môže byť porušené, avšak s istou penalizáciou v účelovej funkcii [2]

Pickup and Delivery Problem with Time Windows : Táto podoba sa viaže s obmedzenou kapacitou vozidla rovnako, ako aj s časovými oknami, predstavujúc generalizáciu VRPTW. Plánovanie trás zahŕňa artikel, resp. položky, ktoré je potrebné vyzdvihnúť na jednom (pickup) mieste a doručiť na druhé (delivery). Tento princíp čiastočne určuje poradie, v ktorom musia byť zákazníci na daných miestach obslužení nakoľko prirodzene vyzdvihnutie musí predchádzať doručeniu. Navyše, miesto vyzdvihnutia a miesto doručenia sú spolu pevne zviazané a normálne musia byť obslužené jedným vozidlom, resp. v rámci jednej trasy.

Pre tieto kanonické varianty VRP sú k dispozícii benchmarkové sady, ktoré slúžia na evaluáciu prístupov či porovnávanie riešení problémov. Príkladmi sú CVRPLib [3, 4] (CVRP), Solomon benchmark [5, 6] (VRPTW) či dataset Li Lim [7, 8].

Podobne existujú mnohé iné podoby VRP, ktoré sa snažia zachytiť rôzne dodatočné atribúty a obmedzenia, a ktoré je možné najrôznejšie kombinovať s už spomenutými podobami alebo medzi sebou. Príkladom je *Heterogeneous Fleet Vehicle Routing Problem* (HVRP), kde vozidlá majú rôzne kapacity, prevádzkové náklady alebo iné parametre. Strategické plánovanie sa tak obohacuje o výber vhodného vozidla pre danú trasu. Ďalším príkladom je *Multi-Depot Vehicle Routing Problem* (MDVRP), kde existuje viacero dep, t. j. miest, z ktorých vozidlá môžu vychádzať a do ktorých sa môžu vracieť. V literatúre podoby VRP kombinujúce viacero takýchto vlastností sú známe tiež ako *Multi-Attribute Vehicle Routing Problems* (MAVRPs) [9].

Dosiaľ sa spomenuté podoby VRP hýbali v kategóriách *statického deterministického VRP*. Statický deterministický VRP stojí na dvoch nasledovných implicitných predpokladoch. Za prvé, všetky informácie vstupujúce do problému sú známe vopred, t. j. pred plánovaním trás, a následne sa nemenia (statický VRP). Za druhé, všetky informácie vstupujúce do problému sú isté, resp. nevstupujú do hry náhodné veličiny (deterministický VRP). V skutočnosti však tieto predpoklady

častokrát neplatia. VRP teda môže byť buď statický alebo dynamický, môže byť deterministický alebo stochastický.

Problém sa definuje ako dynamický, ak sa informácie vstupujúce do problému odкрývajú, resp. aktualizujú súbežne s určovaním trasy [10]. Jednoduchým príkladom dynamického prvku je dynamická objednávka, t. j. dopredu neznáma objednávka. Dynamický problém je navyše stochastický, ak zahŕňa neistotu týkajúcu sa času jazdy, presnej polohy zákazníka a podobne. Čas jazdy napríklad je približne známy, avšak odvíja sa tiež od dopravnej situácie. Lokality zákazníkov môžu mať známu priestorovú distribúciu, i keď presná poloha sa objaví až s prijatou objednávkou. Vo všeobecnosti, dynamické problémy sa charakterovo výrazne líšia od statických. Pre lepšie určenie miery dynamickosti daného problému sa používa *degree of dynamism* (DoD), miera zadefinovaná ako pomer dynamických objednávok k celkovému počtu objednávok [11, 12].

Výzkum práve v oblasti dynamického VRP zažíva prudký rozmach v akademickom svete za posledné desaťročia, čo je možné vďaka technologickému vývoju a naňho nadväzujúcim aplikáciám v logistike [13]. Za posledné desaťročia narástla výpočtová sila, čo umožňuje riešiť zložitejšie inštancie. Ďalším významným faktorom je pokročilá dátová analýza a nové prediktívne modely. Schopnosť zbierať a analyzovať obrovské množstvo dát, ako napr. profilovanie zákazníkov a následné predikcie dopytu či predikcie budúcich scenárov, prináša nové možnosti a umožňuje zainteresovaným spoločnostiam prijímať informované rozhodnutia. K rozmachu prispeli taktiež výrazné pokroky v geopriestorových technológiách, akými sú GPS a GIS. Ich dostupnosť, kvalita poskytnutých informácií, široké spektrum geografických údajov či informácie o aktuálnej dopravnej situácii, prispievajú k presnejšiemu plánovaniu trás, k flexibilnejším riešeniam a umožňujú mnohé iné priestorové predikcie a analýzy. V neposlednom rade kľúčovú úlohu zohráva i komunikácia v reálnom čase, ktorá je už dnes považovaná za úplnú samozrejmosť. Možnosť pohotovej komunikácie medzi dispečermi a vodičmi umožňuje zdieľať dôležité informácie a reagovať tak na nečakané a nové situácie.

Táto diplomová práca sa zaoberá VRP pre prepravu tovaru v spolupráci s firmou Wereldo. Wereldo ponúka zákazníkom komplexné logistické služby, od plánovania prepravy, cez naloženie, až po doručenie. Firma Wereldo používa pre plánovanie trás statický deterministický

riešič (algoritmus), konkrétne sa jedná o implementáciu algoritmu ALNS [14] pre MAVRP založený na PDPTW[15]. Vďaka desaťročiam výskumu a zdokonaľovania algoritmov sa riešenia pre statický VRP stali veľmi spoľahlivými a efektívnymi. Avšak problémom statického plánovania je práve predpoklad, že sú dopredu známe všetky informácie vstupujúce do problému a nič ďalšie sa už neobjaví, čo v praxi nie je až také časté. Ideálne by bolo ponechať si výhody a kvality statického plánovania a súčasne vedieť odpovedať aj na dynamické problémy. Problém, s ktorým firma Wereldo pracuje je zľahka dynamický. Výzvou pre firmu Wereldo je hľadať spôsob ako upraviť, resp. vylepšiť statické plánovanie, aby bolo flexibilnejšie, aby bolo viac disponované prijímať dynamicky vstupujúcich zákazníkov. Z tohto dôvodu sa vo svojej diplomovej práci snažím obohatiť statické plánovanie za účelom získania flexibilnejších plánov. Základnou myšlienkou je koncept *dummy zákazníkov*, resp. ich vloženie do vytváraných statických plánov. Toto riešenie priblížim v ďalších kapitolách.

2 State of the art

Táto kapitola poskytuje prehľad v oblasti aktuálneho výskumu a prístupov k riešeniu v oblasti VRP. Prístupy zahŕňajú statické modely, ktoré predpokladajú vopred známe všetky informácie i dynamické modely, ktoré sa adaptujú na základe príchodu nových informácií. V kapitole sa najprv zameriam na prístupy pre statické problémy, na klasické heuristiky, metaheuristiky a unified algorithms. Následne sa presuniem k prístupom pre dynamické problémy akými sú metódy vkladania, reoptimalizácie či predvídavé prístupy.

2.1 Static vehicle routing problem

U statického VRP sú dopredu známe všetky informácie vstupujúce do problému, čo umožňuje použitie štandardných modelov, pre ktoré sú známe efektívne algoritmy. Táto podkapitola člení prístupy k statickému VRP na *klasické heuristiky*, *metaheuristiky* a *unified algorithms*.

2.1.1 Klasické heuristiky

Na rozdiel od exaktných metód, heuristické metódy nezaručujú optimálne riešenie, avšak poskytujú častokrát kvalitné riešenie s veľmi rozumnou časovou zložitou. Nakoľko reálne inštancie VRP sú často rozsiahle a komplexné, a v praxi ťažko riešiteľné exaktnými metódami, heuristické metódy sa tešia veľkej popularite. Základné dve kategórie pre klasické heuristiky sú konštruktívne a vylepšujúce heuristiky [16].

Konštruktívne heuristiky budujú riešenie od nuly, riadiac sa stanoveným empirickým postupom. Nájdenie riešenia obvykle trvá krátko, avšak riešenie je častokrát značne vzdialené od optimálneho riešenia. Medzi často používané konštruktívne heuristiky patria insert method, saving method, a sweep method. U *insert method* každá trasa začína v depe a v každom kroku je do trasy pridaný najbližší z obslužitelných, neobslužených zákazníkov, na pozíciu, ktorá je najvýhodnejšia vzhľadom na účelovú funkciu, kým sa auto nenaplní [17]. U *saving method* iniciálne každý zákazník predstavuje jednu trasu a následne v každom kroku dochádza k takému zlúčeniu dvoch trás, ktoré povedie k najväčšej úspore v účelovej funkcii [18]. Táto metóda je forma hierarchického

aglomeratívneho clusteringu [19]. *Sweep metóda* zoradí zákazníkov na základe uhlu, ktorý zviaza ich spojnicu s vybraným depom, s iniciálne zvoleným smerom, napr. horizontálnou osou. V tomto poradí sú zákazníci pridávaný do trasy, kým sa kapacita vozidla nenaplní [20].

Vylepšujúce heuristiky prehľadávajú *okolie* (neighbourhood) aktuálneho riešenia a hľadajú v ňom lepšie. Okolie riešenia je množina riešení, ktoré je možné získať z daného riešenia pomocou dopredu určených povolených zmien, napr. tým, že prehodíme dvoch zákazníkov na jednej trase. Rýchlo konvergujú k lokálnemu optimu, čo ich robí účinnými pri riešení rozsiahlych inštancií VRP. Vylepšujúce heuristiky možno rozdeliť na vnútrotrasové a medzitrasové [21]. *Vnútrotrasové* pripúšťajú len zmeny v rámci trasy, zatiaľ čo *medzitrasové* pripúšťajú tiež zmeny, ktoré sa súčasne dotýkajú viacerých trás. Vnútrotrasové sa odvíjajú najmä od operátorov lokálneho vyhľadávania pre TSP akými sú napr. relocate, exchange alebo λ -opt [22, 23]. Viaceré medzitrasové vylepšujúce heuristiky vznikli ako rozšírenie vnútrotrasových. Medzi operátory s ktorými pracujú patria insert, swap, CROSS či λ -interchange [24, 25].

2.1.2 Metaheuristiky

Konštruktíve heuristiky spolu s vylepšujúcimi tvoria tradičné jadro pre metaheuristiky. Metaheuristiky predstavujú istý rámec nad heuristikami, úroveň nad nimi, ktorá reguluje heuristické metódy za účelom účinnejšieho prehľadávania priestoru riešení [26]. Inak povedané, na metaheuristiky je možné sa dívať ako na šablóny (templates) pre algoritmy lokálneho vyhľadávania. Vo všeobecnosti sa považujú za veľmi efektívny prístup k mnohým ťažkým optimalizačným problémom [27]. Práve metaheuristiky napomáhajú, aby použité heuristické metódy neuviazli v lokálnych optimách, resp. aby nedošlo k prehľadávaniu len veľmi špecifického podpriestoru priestoru riešení. Navyše zabezpečujú vhodný kompromis medzi prehľadávaním stále nových oblastí prehľadávaného priestoru a zdokonaľovaním už objavených riešení (známy ako *diversification vs. intensification*). Na základe veľkosti množiny riešení udržiavanej počas výpočtu, možno metaheuristiky rozdeliť na *metódy založené na jednom riešení* a *metódy založené na populácii riešení* [16].

Metódy založené na jednom riešení

V každej iterácii prehľadávajú okolie aktuálne držaného riešenia v snahe nájsť lepšie riešenie. Niekedy sa nazývajú tiež neighbourhood-based alebo metódy lokálneho vyhľadávania. Patria sem metaheuristiky ako Tabu search, Variable neighbourhood search, Simulated Annealing (SA) alebo Large Neighborhood Search (LNS).

Tabu search sa zakladá na myšlienke uchovávať si v pamäti históriu prehľadávania a zamedziť tomu, aby sa prehľadávanie vrátilo späť k nedávno objaveným riešeniam [28, 29]. Týmto spôsobom bráni, aby prehľadávanie uviazlo v lokálnom minime. K ukladaniu používa dátovú štruktúru zvanú tabu zoznam.

Variable neighbourhood search prináša myšlienku systematickej zmeny okolia riešenia [30]. Zakladá sa najmä na troch pozorovaniach [31]. Po prvé, ak je riešenie lokálnym lokálnym minimom vzhľadom na dané okolie, nie je nutne lokálnym minimom vzhľadom na iné okolie. Po druhé, globálne minimum je globálnym minimom vzhľadom na všetky možné okolia. Po tretie, lokálne minimá vzhľadom na jedno alebo viaceré okolia sa často vyskytujú blízko seba. Striedaním viacerého prehľadávaných okolí sa dosahuje schopnosť prekonávať lokálne minimá.

Simulated annealing odzrkadľuje model energetických zmien počas procesu žihania v metalurgii [32]. V kontexte lokálneho vyhľadávania to znamená, že na začiatku sa nastaví nejaká iniciálna teplota, ktorá sa časom znižuje; a akceptačné kritérium pre prijatie nového najlepšieho riešenia je závislá od aktuálnej teploty. V praxi to často znamená, že v úvode môže dochádzať ku výrazným zmenám v riešení, a postupom času sa tieto zmeny stávajú viac konzervatívne, resp. nové riešenie je už len vyladením toho pôvodného bez podstatných zmien. Obvykle je prítomná silná náhodnosť v generovaní riešení v každej iterácii, napr. náhodný sampling z okolia riešenia.

Large neighborhood search sa odráža od pozorovania, že prehľadávanie širšieho okolia vedie k nájdeniu naozaj kvalitného lokálneho optima, a tak i k lepším finálnym riešeniam [33]. Pravdaže prehľadávanie širšieho okolia má vysokú časovú zložitosť, preto je často prehľadávanie regulované rôznymi heuristickými technikami. Okolie riešenia je definované pomocou metód *Ruin & Recreate*. Tieto stratégie pri hľadaní čo najlepšieho riešenia vezmú pôvodné riešenie, čas-

točne ho rozbijú (ruin) a chýbajúcu časť doplnia iným heuristickým spôsobom (recreate). Jedná sa o iteratívny proces. Vopred sú definované akceptačné kritéria, ktoré rozhodujú o tom, aké riešenie je možné prijať v danej iterácii. Termín Ruin & Recreate bol prvýkrát spomenutý v práci Schrimpf et al. (2000) [34], ktorí techniku aplikovali okrem iného práve i na VRPTW. Rozbitie riešenia spočívalo v odstránení vybraných zákazníkov, zákazníkov v určitom okruhu či zákazníkov z istého úseku trasy. Čiastočné riešenie bolo následne dotvorené použitím hladového algoritmu, ktorý zvyšných zákazníkov doplnil v náhodnom poradí s čo najnižšou cenou. Ruin metóda je zároveň nositeľom stochasticity, aby pri opakovanej aplikácii došlo vždy, resp. často k rozbitiu inej časti riešenia. Pri jej implementácii je tiež dôležité vhodne zvoliť aké percento riešenia sa má zachovať, resp. rozbiť. Zvoliť málo znamená prehľadávať len úzky podpriestor riešení, resp. úzke okolie, čo protirečí základnému princípu LNS. Zvoliť veľa zas znamená prakticky reinitializovať riešenie.

Metódy založené na populácii riešení

Do tejto kategórie patria evolučné algoritmy či algoritmy simulujúce kolektívnu inteligenciu (swarm intelligence alg.).

Evolučné algoritmy modelujú evolúciu v prírode, ktorá sa riadi zásadou, že silnejší jedinci, na rozdiel od slabších, splodia najskôr potomka s väčším potenciálom na prežitie [35]. V kontexte evolučných algoritmov jedinec predstavuje riešenie problému, a sila, resp. potenciál jedinca predstviuje účelovú funkciu riešenia. Táto biologická reprodukcia je v kontexte VRP typicky simulovaná operáciou zvanou *crossover* [36].

Algoritmy simulujúce kolektívnu inteligenciu modelujú zas systémy zložené z mnohých autonómnych spolupracujúcich jednotiek bez centralizovaného riadenia, akými sú kolónie mravcov či včiel [37]. Jeden zo súčasných trendov je tiež použitie viacúčelových evolučných algoritmov pre *multi-objective VRP*.

2.1.3 Univerzálne algoritmy

Pri pohľade na nezpočetné množstvo rôznych podôb, do ktorých je možné formulovať statický VRP, začína narastať potreba posunúť sa od

špecificky orientovaných metód k univerzálnejším prístupom, ktoré by bolo možné aplikovať na širokú škálu problémov s rôznymi podmienkami a nastaveniami. Existuje viacero takýchto algoritmov, ktoré ponúkajú veľmi kvalitné riešenia na mnohé MAVRPs. Označujú sa ako univerzálne algoritmy (unified algorithms).

Adaptive large neighborhood search

Základná myšlienka LNS sa ďalej rozvinula do viacerých podôb ako napr. *Adaptive Large Neighbourhood Search* (ALNS) [38, 14]. Veľkou výhodou tohto prístupu je univerzálnosť riešenia pre viaceré typy problémov, konkrétne pre VRPTW, CVRP, MDVRP, *split delivery VRP* a *open VRP*. Všetky typy problémov sa ešte pred samotným riešením transformujú do univerzálneho problému, ktorý v sebe zahŕňa všetkých päť typov, *rich pickup and delivery problem with time windows*. Hlavnou novinkou oproti LNS je adaptívna vrstva, ktorá vyberá spomedzi viacerých heuristik pre odoberanie i vkladanie zákazníkov, aby bolo prehľadávanie čo najúčinnnejšie. V každej iterácii je vybraný jeden Ruin algoritmus a jeden Recreate algoritmus. Výber je riadený stochasticky. Každému algoritmu (resp. okoliu, ktoré definuje) je priradené skóre na základe toho nakoľko prispel k zlepšeniu v minulosti. Čím vyššie skóre, tým väčšia šanca, že bude daný algoritmus zvolený. ALNS obvykle používa metaheuristiky ako SA, ako mechanizmus určujúci akceptačné kritéria.

Slack induction by string removals

Slack Induction by String Removals (SISR) predstavuje implementáciu LNS vyznačujúcu sa špecifickou ruin metódou [39]. Podobne ako ALNS, lokálne vyhľadávanie u SIRS preberá akceptačný mechanizmus SA. Lokálne vyhľadávanie je iteratívnym procesom. Pri každej iterácii sa aplikuje Ruin & Recreate na aktuálne riešenie. Následne sa rozhodne o prijatí nového aktuálneho riešenia. Ruin v SIRS sa riadi dvoma princípmi. Po prvé, odobrať dostatočné množstvo obslúžených zákazníkov z riešenia, aby tak vznikol dostatočne veľký priestor pre ich efektívnejšie preskladanie. Po druhé, metóda odstraňuje úseky trás (resp. zákazníkov na týchto úsekoch), ktoré sú blízko pri sebe. Recreate metóda má u SIRS tiež svoje špecifiká. Zvyšných zákazní-

kov zoradíme náhodne, podľa vzdialenosti od depa, podľa veľkosti objednávky alebo iným spôsobom, a následne ich hladovo pridávame jeden po druhom. Avšak nevyhodnocujú sa nutne všetky možné pozície, kam je zákazníka možné pridať. Zavedie sa koeficient β , ktorý predstavuje pravdepodobnosť, s ktorou sa vyhodnotenie danej pozície preskočí (tzv. blink rate).

Hybrid genetic search

Hybrid Genetic Search (HGS) je optimalizačný algoritmus, ktorý kombinuje výhody genetických algoritmov a lokálneho prehľadávania [9]. Princípom je udržiavať si rozmanitú množinu (tzv. populáciu) možných riešení a systematickým spôsobom sa dostávať k lepším. Algoritmus iteratívne vyberie dvojicu *rodičov* (riešení) z populácie, skombinuje ich, a vytvorí nové riešenie, *potomka*. Pre rekombináciu HGS aplikuje *ordered crossover* na permutačne ladenej reprezentácii rodičov [40]. Algoritmus následne spustí lokálne vyhľadávanie v okolí potomka, a vyberie z neho najlepšieho kandidáta, ktorého zaradí do populácie.

Jedným z kľúčových bodov algoritmu je udržiavanie si bohatej diverzity v populácii. Nízka diverzita by mohla viesť k predčasnej konvergencii algoritmu. Prehľadávanie priestoru riešení by sa tým obmedzilo len na veľmi špecifický podpriestor, ktorý len s malou pravdepodobnosťou bude obsahovať riešenie blízke optimu. Každému riešeniu je preto priradený rating, ktorý odzrkadľuje nie len jeho cenu ale aj jeho prínos k diverzite populácie. Zatiaľ, čo rekombinácia prispieva k diverzite v populácii, lokálne vyhľadávanie má viesť k výrazným zlepšeniam riešenia.

HGS tiež prichádza tiež so stratégiou riadeného prehľadávania neprípustných riešení. To znamená, že prekročenie stanovených limitov a podmienok je lineárne penalizované. Ako dôsledok, prehľadávanie sa nevzdiali prílišne z priestoru riešení vymedzeného danými podmienkami, avšak súčasne umožní sústrediť prehľadávanie do oblastí na hraniciach prípustnosti, kde sa s väčšou pravdepodobnosťou budú nachádzať riešenia blízke optimálnemu riešeniu [41].

Moderná implementácia HGS pre CVRP predstavuje viaceré zlepšenia pôvodného algoritmu. Jedným z nich je predstavenie dodatočného okolia riešenia nazývaného SWAP*, ktoré spočíva v zámene

dvoch zákazníkov na rôznych trasách voľnejším spôsobom, t. j. nie nutne vložením in place [36]. Prehľadávanie všetkých možností v prípade, že môžeme zákazníka vložiť na ľubovoľné miesto v danej trase by priamou implementáciou zabralo $\theta(n^3)$. Existujú avšak efektívnejšie stratégie prehľadávania, ktoré znižujú časovú zložitosť. Použitím takejto efektívnej implementácie operácie SWAP*, sa lokálne vyhľadávanie stáva omnoho účinnejším.

2.2 Dynamic vehicle routing problem

U dynamického VRP sa informácie vstupujúce do problému odкрývajú postupne. Hlavným zdrojom dynamickosti sú nové objednávky zákazníkov počas dňa, i keď dynamika môže byť daná i zmenou dopravnej situácie či zmenou v prioritách jednotlivých zákaziek. Prístupy k DVRP spadajú do troch hlavných prúdov: metódy vkladania, metódy reoptimalizácie a predvídavé prístupy.

2.2.1 Metódy vkladania

Metódy vkladania u dynamického VRP vstupujú do situácie, kedy je už pripravený statický plán odpovedajúci na vopred známe požiadavky zákazníkov, a úlohou je zintegrovat do statického plánu nové zákazky prichádzajúce počas dňa. Metódy vkladania predstavujú spôsoby, akými je možné postupne vkladať zákazníkov do statického plánu bez toho, aby plán iným spôsobom menili. Častokrát slúžia ako vstupný bod pre reoptimalizačné metódy.

Základným problémom je vloženie jedného zákazníka. Základnou heuristikou je *hladový prístup*, kde je zákazník vložený na pozíciu, ktorá predstavuje najlacnejšiu variantu.

V prípade, že je zákazníkov viac, otvára sa otázka, v akom poradí ich vkladať. Podobne ako pri jednom zákazníkovi, základnou heuristikou je hladový prístup. V každom kroku dôjde ku vloženiu, ktoré predstavuje najmenší nárast celkovej ceny riešenia. Inými slovami, najprv algoritmus spočíta minimálnu cenu vloženia pre každého zákazníka a následne vyberie, resp. pridá zákazníka s najlacnejšou minimálnou cenou vloženia na najlacnejšiu pozíciu [16]. Podobne sa scenár opakuje pre zvyšných zákazníkov.

Problémom tohto prístupu je, že najväčšie, resp. najťažšie obslužitelné zákazky sa dostanú na rad častokrát až ako posledné, resp. v neskorších fázach, kedy už aktuálny plán trás je menej flexibilný. Na tento problém sa snažia reagovať zrevidované obdoby hladového prístupu. *Farthest insert* v každom kroku vkladá do aktuálneho riešenia najvzdialenejšieho zákazníka na cenovo najvýhodnejšiu pozíciu [42]. S inou myšlienkou pracuje zasa *regret insert*, ktorý uprednostňuje zákazky, ktorých vloženie na druhú či i-tú najvýhodnejšiu pozíciu je podstatne drahšie než na tú najvýhodnejšiu [43, 44].

2.2.2 Metódy reoptimalizácie

Častokrát prosté vloženie nových zákaziek do existujúceho plánu bez ďalších zmien nemusí byť možné alebo môže byť veľmi drahé. Insertion methods samé osebe veľakrát nepostačujú. Z tohto dôvodu dochádza k prehodnoteniu existujúceho plánu. Prehodnoteniu, ktoré vedie k úprave naplánovaných trás, k ich reoptimalizácii. Tento proces nie je jednorazová záležitosť. Na reoptimalizáciu je možné pozeráť sa ako na istý rámec zahŕňajúci komponentu pre riešenie statického problému. Ako taká sa reoptimalizácia môže spájať s rôznymi ďalšími technikami, heuristikami i metaheuristikami akými sú metódy lokálneho vyhľadávania, simulated annealing, tabu-search a mnohé ďalšie. Na základe toho, kedy dochádza k reoptimalizácii trás, možno reoptimalizačné metódy rozdeliť na metódy založené na periodickej reoptimalizácii a metódy založené na kontinuálnej reoptimalizácii [11].

Metódy založené na periodickej reoptimalizácii začínajú na začiatku dňa iniciálnou optimalizáciou, t. j. vytvoria iniciálny plán trás. Následne dochádza k optimalizácii riešenia pre aktuálny stav vždy v istých, dopredu daných, časových intervaloch [45]. Prípadne ak dôjde k rozsiahlej zmene dostupných informácií. V oboch prípadoch sa v danej chvíli jedná o riešenie statického problému odzrkadľujúceho aktuálny stav. Vďaka tomu je možné tu využívať algoritmy pre statický VRP, ktorých evolúcia má za sebou rozsiahly výskum. Nevýhodou je, že aktualizáciu riešenia nie je možné urobiť dokým sa neskončia všetky optimalizácie. Dispečeri tak môžu reagovať až s istým oneskorením.

Metódy založené na kontinuálnej reoptimalizácii, na rozdiel od periodickej reoptimalizácie, sú založené na neustálej optimalizácii počas

celého dňa a ukladaní si vybraných údajov o dobrých riešeniach do adaptívnej pamäti [46]. Vždy keď dôjde k zmene dostupných informácií, systém zozbiera dáta z pamäte a odvíjajúc sa od nich upraví aktuálny plán trás. Výhodou je maximalizácia využitia výpočtovej sily. Nevýhodou je zložitejšia implementácia a veľká premenlivosť plánu trás. Vodiči tak častokrát, dokým neobslúžia danú zákazku, nevedia dopredu aké bude ich ďalšie smerovanie; čo v praxi pre vodiča častokrát predstavuje veľký diskomfort.

2.2.3 Predvídavé prístupy

Dosiaľ spomenuté metódy reoptimalizácie spadajú do kategórie *krátkozrakéj reoptimalizácie* [2]. Krátkozraká reoptimalizácia sa zameriava čisto na riešenie problému korešpondujúceho s aktuálnym stavom a neberie do úvahy budúci vývoj udalostí. Hľadá riešenie, ktoré najlepšie odpovedá na aktuálnu situáciu. Používa sa často v deterministickom prostredí, kde sa kladie dôraz na schopnosť rýchlo reagovať.

V skutočnosti však pri plánovaní je častokrát žiadané brať do úvahy dlhodobější horizont udalostí. Z tohto dôvodu existujú prístupy rozširujúce reoptimalizáciu s cieľom zabezpečiť dobré výsledky v dlhodobom horizonte [47]. Takáto reoptimalizácia sa zvyčajne označuje ako *predvídavá*. Medzi dôležité metódy predvídavej reoptimalizácie patria: Multiple Scenario Approach (MSA), heuristika založená na tzv. dvojitom horizonte, stratégia čakania a fruitful regions.

multiple scenario approach berie do úvahy budúce smerovanie udalostí skrze potenciálne scenáre v rámci istého časového horizontu [47]. Kľúčová myšlienka MSA spočíva vo vytváraní plánov, ktoré zahŕňajú existujúce ako aj potenciálne budúce objednávky zákazníkov. Budúce objednávky sú generované náhodne sledujúc určené pravdepodobnostné rozdelenie [48]. V priebehu dňa potom prichádzajú momenty, kedy je potrebné prijať rozhodnutie a vytýčiť tak isté smerovanie známe tiež ako *distinguished plan*. Ten vyberá *consensus function*. Táto funkcia sa vyznačuje tým, že nevyberá v prvom rade plán s najnižšími nákladmi, ale taký plán, ktorý sa čo najmenej odlišuje od ostatných. Inými slovami, vyberá čo najmenej záväznú stratégiu, čo je výhodné v prípade inštancií s vysokým percentom dynamických objednávok.

Heuristika založená na dvojitém horizonte sa z krátkodobého hľadiska snaží minimalizovať prejdenú vzdialenosť [49]. V dlhšom časovom horizonte sa zas snaží udržiavať časový buffer (time buffer alebo tiež slack), extra čas, ktorý plán trás robí značne flexibilným, schopným lepšie zvládať nečakané okolnosti ako zdržanie kvôli doprave alebo nové zákazky či požiadavky.

Stratégie čakania využívajú nevyhnutné čakacie časy na trasách k navýšeniu flexibility plánov pre DVRP. Vo VRP sa v dôsledku zavedených časových okien na niektorých miestach pozdĺž trasy stretávame s čakaním. U statického VRP, jedinou zmysluplnou stratégiou čakania je vyraziť čím skôr, drive-first (DF). U DVRP distribúcia času čakania môže mať vplyv na kvalitu riešenia. Protipríkladom k DF je voľba najneskoršieho odchodu, wait-first (WF). Výhodou WF je, že v čase odchodu sú často známe dodatočné informácie, resp. nové objednávky, a s nimi i potenciál plánu vozidiel. *Dynamic waiting strategy* zas vznikne ako špecifická kombinácia DF a WF [50]. Rozdelíme danú trasu na úseky (zóny) tak, aby každá zóna predstavovala množinu zákazníkov, ktorí sú blízko pri sebe. Inými slovami, medzi zákazníkmi v jednej zóne sú malé vzdialenosti a medzi zákazníkmi v odlišných zónach sú relatívne veľké vzdialenosti. *Dynamic waiting strategy* sa potom správa ako DF vo vnútri zóny, avšak v momente, kedy obslúži všetkých zákazníkov v danej zóne a má sa presunúť do inej, správa sa ako WF.

Koncept *fruitfull regions* sa zakladá na pozorovaní, že zákazníci v skutočnosti nie sú v priestore roztrúsení rovnomerne, ale sú často sústredení do istých zón, clusterov. Termínom *fruitful regions* sa označujú oblasti, z ktorých s veľkou pravdepodobnosťou vstúpi objednávky na transport [51]. Pre dané vozidlo je niekedy výhodné predvídavo navštíviť takúto oblasť v čase, ak sa v nej pravdepodobne v blízkom čase objaví nová objednávka. S konceptom *fruitful regions* úzko súvisí koncept *dummy* zákazníkov, ktorí sú častokrát umiestnení priamo do takejto oblasti alebo blízko nej, modelujúc tak potenciálneho zákazníka. Pre vyhradenie oblastí perspektívnych na nové objednávky sa častokrát používa clustering, s *dummy* zákazníkmi strategicky umiestnenými tak, aby sa ich poloha čo najmenej vzdialovala od novej polohy reálneho potenciálneho zákazníka. Na *dummy* zákazníka je tak možné sa pozerať ako na konkrétnu alokáciu prostriedkov (času

na trase a kapacity vozidla) práve v takejto perspektívnej oblasti. Integrácia dummy zákazníkov tu predstavuje proaktívnu metódu využívajúcu stochastickú informáciu. Metóda je zvlášť účinná, ak dostupné dáta z minulosti sú rôznorodé, obsahujúce vysokú štrukturálnu variabilitu [52]. Užitočné v tomto ohľade tiež môže byť prekonať predstavu, že objednávky prichádzajúce v rôzne dni sa riadi rovnakým vzorcom, rovnako ako že vzorec pre aktuálny deň sa nutne nachádza medzi vzorcami, ktoré sa už objavili v minulosti [53].

3 Generátor inšancií

Syntetické dáta v podobe inšancií VRP častokrát slúžia ako vstup pre rôzne metódy a algoritmy (riešiacie nejakú obdobu VRP) pre testovanie ich efektivity. Podobne generátor naimplementovaný v rámci tejto práce je súčasťou väčšieho systému slúžiaceho, okrem iného, pre testovanie navrhnutých stratégií pre generovanie dummy zákazníkov. Tento systém priblížim v štvrtej kapitole.

V podkapitole 3.1 predstavujem generátor inšancií statického VRP naimplementovaný na základe mapových podkladom z OpenStreet-Map (OSM) a analýzy reálnych dát poskytnutých firmou Wereldo. Podkapitola 3.2 sa týka rozšírenia funkcionality generátora o schopnosť vytvárať dynamické scenáre. Konkrétne sa jedná o postupnosť dynamických objednávok objavujúcich sa počas dňa.

3.1 Statické inšancie

Výstupom generátoru je inšancia štandardného statického rich VRP, resp. MAVRP. Inšancia zahŕňa množinu objednávok, množinu vozidiel, maticu vzájomných vzdialeností miest vstupujúcich do problému a maticu časov, ktoré korešpondujú s týmito vzdialenosťami.

Každá objednávka obsahuje polohu, časové okno a čas obsluhy pri vyzdvihnutí a doručení, ďalej hmotnosť, objem a výšku nákladu. Objednávky generujem na základe analýzy reálnych dát poskytnutých firmou Wereldo. Jedná sa o dáta za obdobie necelých troch mesiacov. Pri generovaní objednávky dochádza ku kontrole možnosti jej obsluhy. Prv než je daná objednávka zaradená do množiny objednávok, generátor vytvára kandidátov na objednávky. V prípade, že je kandidát neobslúžiteľný, t. j. napr. v prípade, že tento kandidát hmotnosťou prekračuje maximálnu nosnosť u všetkých vozidiel, sa kandidát zahodí, a proces generovania kandidátov pokračuje až dokým kandidát nesplní požadované vlastnosti. Až sa takýto kandidát nájde, generátor ho, ako validnú objednávku, vloží do množiny objednávok.

Ako predvolená množina vozidiel, ktorá vstupuje do inšancií je množina vozidiel, typovo podobných vozidlám, s ktorými pracovala firma Wereldo v období, z ktorého pochádzajú dáta. Táto množina pozostáva z dvanástich vozidiel a slúži na obsluhu približne sto

objednávok denne. Túto množinu si užívateľ môže navoliť, resp. ľubovoľne upraviť.

Pre generovanie matíc vzdialeností a časov využívam OpenRoute-Service (ORS) [54]. ORS je opensource geoinformačný systém voľne dostupný na GitHubu [55]. Systém poskytuje API služby pre navigáciu, výpočet optimálnych trás či analýzu oblasti dosiahnuteľnej v danom čase. Systém umožňuje voľbu módu pre cestovanie (auto, nákladné vozidlo a podobne), ako i zadanie špecifických podmienok pozdĺž trasy. ORS je postavený na dátach z OpenStreetMap. OpenStreetMap je projekt s cieľom vytvárať voľne dostupné geografické dáta spolu s ich následnou vizualizáciou do podoby máp či navigácie [56]. Dáta poskytované skrze OSM sú dostupné pod licenciou Open Database License (ODbL).

Pre testovanie riešení pre VRP sa bežne používajú benchmarkové inštanície ako napr. inštanície poskytované výskumnou organizáciou *Sintef* [57]. Napriek hodnote týchto inštancií, častokrát vychádzajú zo zjednodušených syntetických dát a neodrážajú komplexnosť skutočných problémov. Výhodou generátora naimplementovaného v rámci tejto práce je jeho schopnosť produkovať scenáre blízke realite. Použitie mapových podkladov OSM, matíc reálnych vzdialeností a časov pre danú trasu a analýza reálnych dát poskytnutých firmou Werelda umožňuje vytvárať realistickejšie inštanície viac použiteľné v praxi.

Následujúce sekcie podkapitoly 3.1 opisujú spôsob generovania jednotlivých atribútov u danej objednávky.

3.1.1 Geografická poloha

Poloha zákazníka sa odvíja od zamerania jeho činnosti. Vinárstvo budeme hľadať na inom mieste ako automobilovú firmu. Pre určenie charakteru zákazníkov Werelda som kategorizoval ich zákazníkov do sedemnástich kategórií. Analýza percentuálneho zastúpenia zákazníkov, zákaziek či hmotnosti prepraveného nákladu v daných kategóriách poskytla prehľad o typických zákazníkoch Werelda. Následne som tieto kategórie napamoval na kategórie prítomné v OSM. Hlavným problémom týchto kategórií bol fakt, že veľké množstvo firiem v OSM nie je zaradené do týchto kategórií, ale nesie iba všeobecné označenie ako priemyselná budova či priemyselná zóna. Pre geografickú polohu potenciálnych zákazníkov som z tohto dôvodu vybral všeobecnejšiu

kategóriu, ktorá sa v OSM označuje tagom *landuse=industrial*. Pod túto kategóriu spadá okolo 13 000 objektov v Českej republike, s 200 objektami v Karlovarskom kraji a 2000 v Středočeskom kraji. Táto kategória predstavuje pre generátor množinu potenciálnych pickup lokalít.

Čo sa týka delivery lokalít, nakoľko vyše 90 % všetkých objednávok v reálnych dátach smeruje do depa, vo svojej práci sa pre jednoduchosť obmedzím len na objednávky tohto typu.

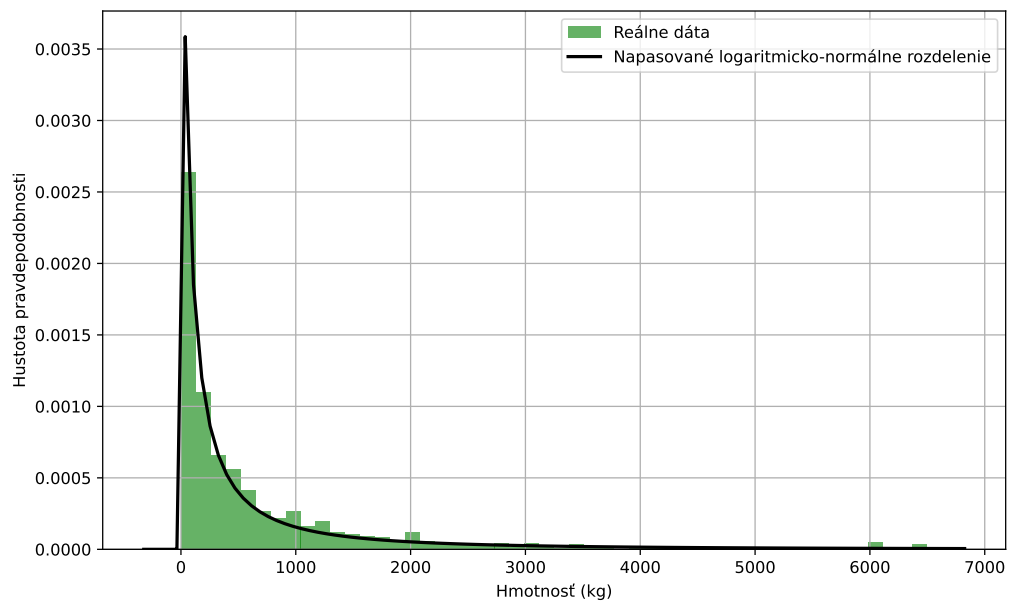
Pre polohu depa je možné vziať lokality reálne existujúcich dep prepravných spoločností, alebo vybrať lokality pre potenciálne depá. Miesta, kde by sa tieto depá mohli zriadiť. Na základe analýzy polohy reálnym dep som odvodil nasledovné priaznivé faktory pre polohu depa: lokalita v blízkosti väčšieho mesta (do 10 km), blízkosť diaľničného výjazdu, blízkosť železničnej trati, umiestnenie v priemyselnej oblasti, a označenie na mapách (OSM) ako uzol, priem. budova, priem. oblasť, sklad či komerčná budova. Rešpektujúc spomenuté podmienky som pre každý kraj v Českej republike vybral tri možné polohy pre depá.

3.1.2 Hmotnosť

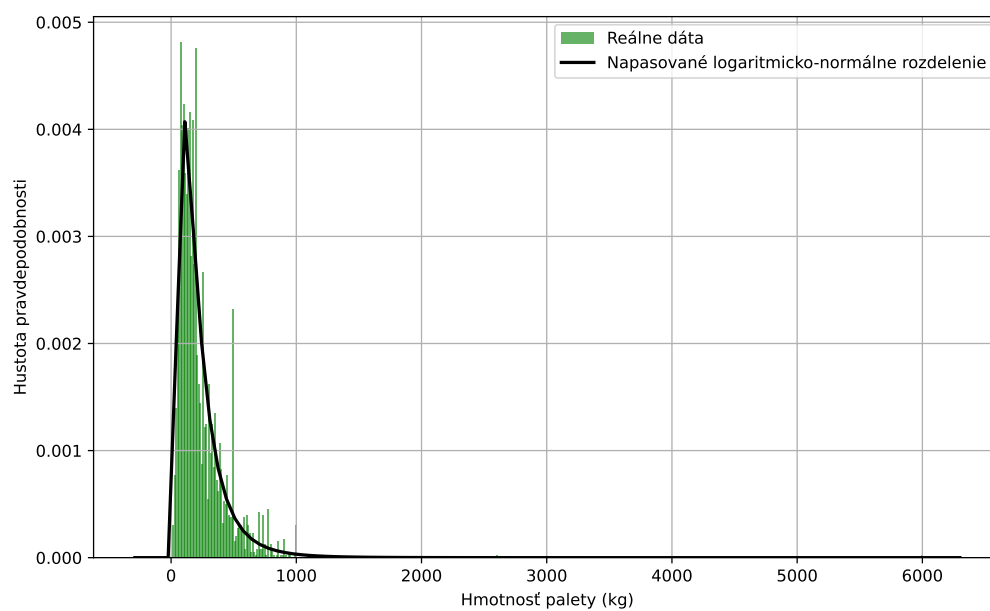
Hmotnosť nákladu k preprave generujem nezávisle, na základe analýzy pravdepodobnostného rozdelenia v poskytnutých dátach. Rozdelenie sa približuje logaritmicko-normálnemu rozdeleniu (viď obr. 3.1), ktoré som použil pre generovanie hmotnosti. Parametre rozdelenia som určil pomocou metódy *lognorm.fit* z knižnice *scipy* [58].

3.1.3 Objem

Objem nákladu k preprave generujem v závislosti od jeho hmotnosti. Konkrétne, generujem hmotnosť jednej palety, ktorou sa podelí hmotnosť celého nákladu. Podobne ako u hmotnosti, rozdelenie v poskytnutých dátach sa približuje logaritmicko-normálnemu rozdeleniu (viď obr. 3.2). Rovnako parametre rozdelenia sú určené pomocou metódy *lognorm.fit* z knižnice *scipy* [58]. Nakoľko zákazník zadáva počet paliet ako celé číslo, hodnota je následne zaokrúhlená.



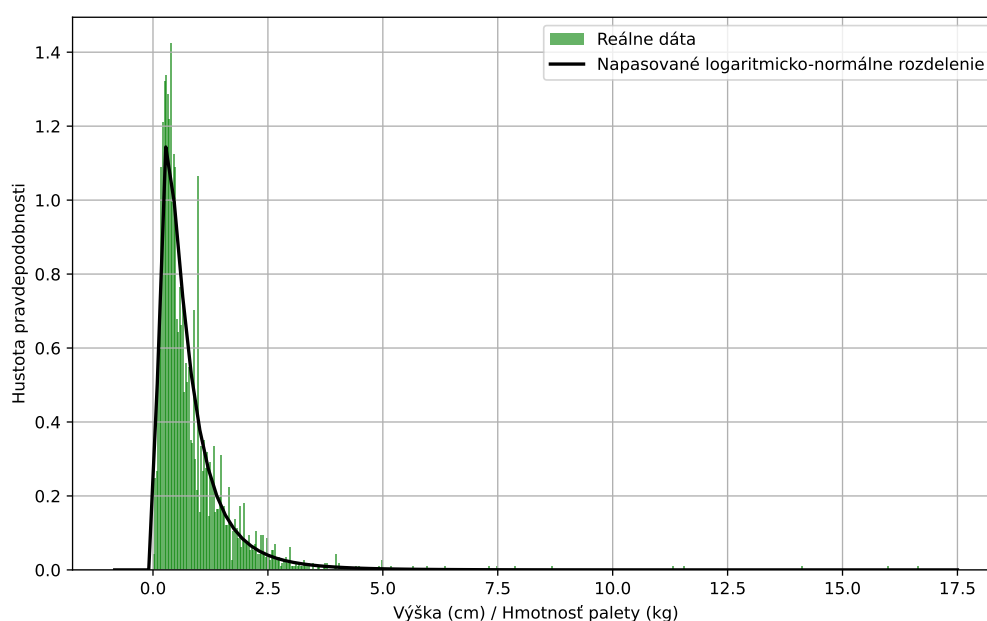
Obr. 3.1: Histogram hustoty pravdepodobnosti pre hmotnosti



Obr. 3.2: Histogram hustoty pravdepodobnosti hmotnosti pripadajúcich na jednu paletu

3.1.4 Výška

Výšku nákladu k preprave generujem v závislosti od jeho hmotnosti a počtu paliet. Konkrétne, generujem podiel výšky a hmotnosti, ktorá pripadá na jednu paletu. Rozdelenie v poskytnutých dátach sa približuje logaritmicke-normálnemu rozdeleniu (viď obr. 3.3). Predpokladám, že náklad je na paletách rozložený rovnomerne. Táto generovaná veličina, ako alternatíva k hustote materiálu nákladu, spolu s hmotnosťou nákladu a počtom paliet udáva výšku nákladu.



Obr. 3.3: Histogram hustoty pravdepodobnosti pre podiely výšky a hmotnosti pripadajúcej na jednu paletu

3.1.5 Čas obsluhy

Čas obsluhy pri vyzdvihnutí (pickup service time) aj čas obsluhy pri doručení (delivery service time) generujem nezávisle od iných veličín, na základe analýzy pravdepodobnostného rozdelenia v poskytnutých dátach. Analýza poskytnutých dát ukazuje len veľmi slabú koreláciu s inými veličinami, aj medzi sebou. Hodnoty pre čas obsluhy pri vyzdvihnutí sa pohybujú v rozmedzí od 4 minút po 53

minút, s priemernou hodnotou okolo 15 minút. Výraznou odchýlkou od logaritmicko-normálneho rozdelenia je hodnota 14 minút, ktorá sa vyskytuje najčastejšie, tvoriac asi dve percentá všetkých hodnôt. Hodnoty pre čas obsluhy pri doručení sa pohybujú v rozmedzí od 8 minút po 3 hodiny, s priemernou hodnotou okolo 17 minút. Výraznú odchýlku tu zas predstavuje najčastejšie vyskytujúca sa hodnota 20 minút, ktorá tvorí asi jedno a pol percenta všetkých hodnôt.

3.1.6 Časové okná

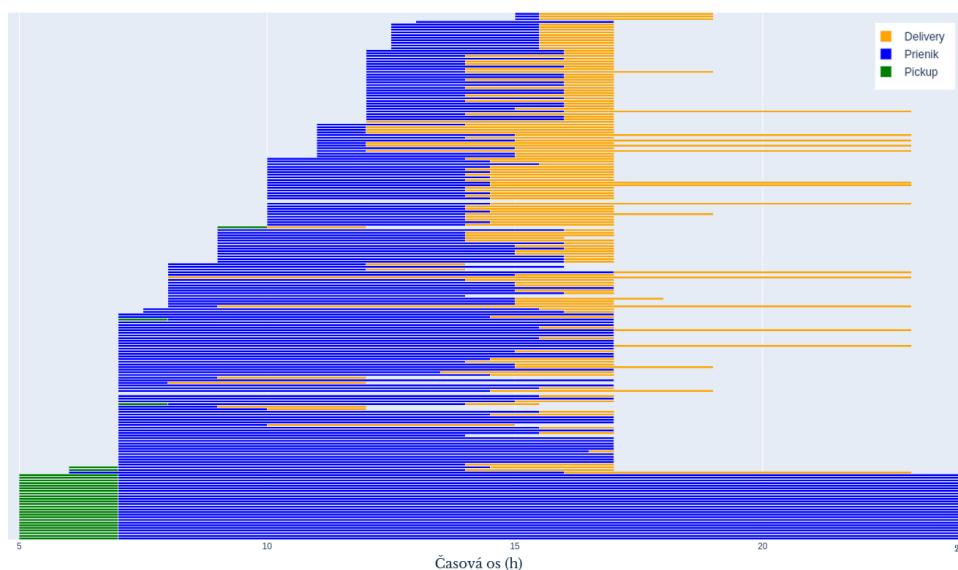
Jedná sa o dve časové okná (pickup a delivery), t. j. štyri hodnoty, ktoré spolu súvisia, a ktoré nemožno generovať osobitne bez výraznej straty kvality vygenerovaných dát. Pre generovanie som skúšal viaceré modely. Najlepšie výsledky ukazovali *Gaussian multivariate copula* [59] a generovanie založené na *Principal Component Analysis* [60]. Avšak ani s pomocou týchto modelov vygenerované dáta neboli charakterovo dostatočne podobné poskytnutým reálnym dátam. Z tohto dôvodu som pre generovanie časových okien zvolil použiť reálne dáta ako blackbox, z ktorej samplujem hodnoty s pravdepodobnosťou zodpovedajúcou pravdepodobnostnému rozdeleniu v týchto dátach. Časové okná v reálnych dátach sú pomerne široké, pričom časové okno pre doručenie je častokrát totožné s časovým oknom depa (viď obr. 3.4).

3.2 Dynamické scenáre

Dynamický scenár pozostáva z postupnosti prichádzajúcich nových objednávok spolu s časom ich príchodu. Hlavným parametrom dynamického scenára je DoD (podiel dyn. objednávok medzi všetkými objednávkami), od ktorého závisí počet dynamických objednávok v jednom scenári.

Väčšina atribútov dynamických objednávok sa hodnotami a ich pravdepodobnostným rozdelením neodlišuje od dopredu známych objednávok, navyše dynamické objednávky v reálnych dátach predstavujú len malé percento všetkých objednávok, preto atribúty dynamických objednávok generujem rovnako.

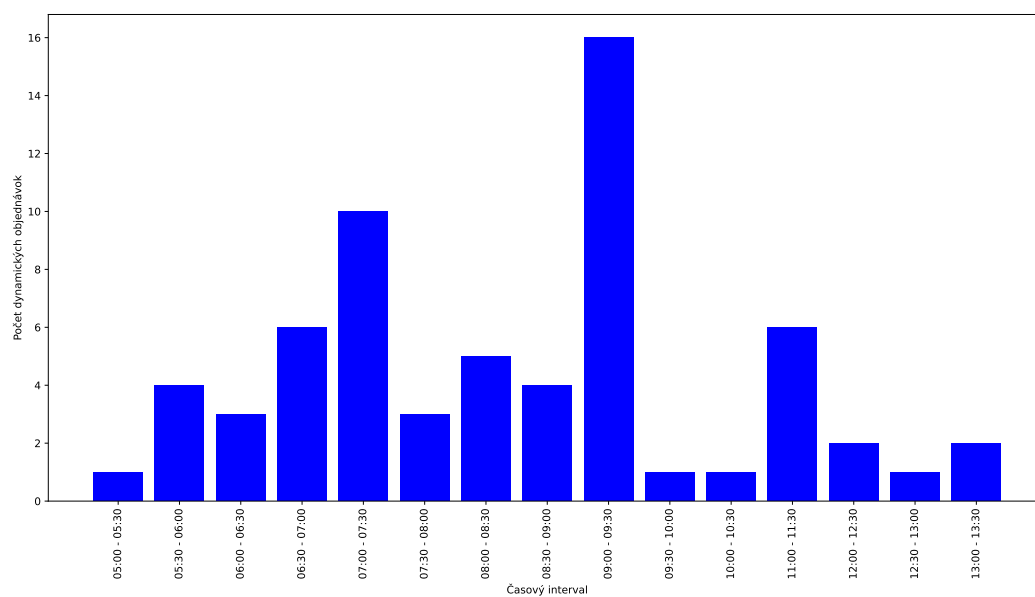
Odlišnosť v generovaní predstavujú časové okná, ktoré závisia od času vzniku dynamickej objednávky, t. j. času kedy zákazník objed-



Obr. 3.4: Ukážka reprezentatívnej vzorky dvesto časových okien zoradených podľa najskoršieho pickup času

návku zadal do systému. Pravdepodobnostné rozdelenie príchodu objednávok v čase v reálnych dátach (viď obr. 3.5) ukazuje tri výraznejšie vrcholy. Jeden okolo siedmej hodiny, druhý okolo deviatej a tretí okolo jedenástej. Pravdepodobnostné rozdelenie časov príchodu objednávky použité na generovanie zachytávam s pomocou *Gaussian KDE* [58].

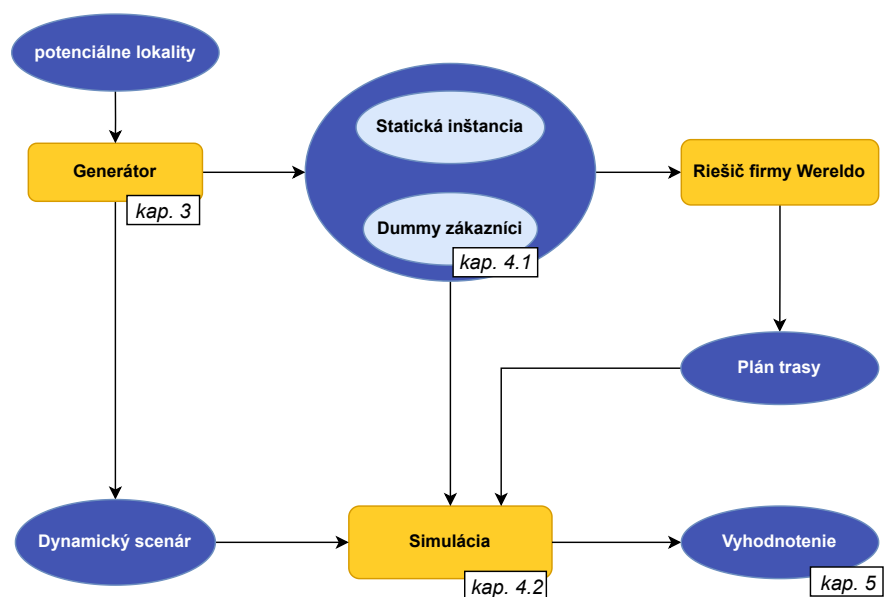
Samotné časové okná dynamických objednávok generujem zjednodušeným spôsobom. Predpokladám, že zákazník zadá objednávku v čase, kedy už je možné prepravovaný náklad vyzdvihnúť. Inak povedané, časové okno pre vyzdvihnutie začína v momente, kedy zákazník objednávku zadá do systému. Navyše neuvažujem urgentné objednávky, t. j. objednávky, ktoré je potrebné vyzdvihnúť v krátkom čase po zadaní objednávky. Jednak sa takéto objednávky vyskytujú v malom množstve medzi dynamickými objednávkami v reálnych dátach, jednak takéto objednávky výrazne menia charakter celého problému. Z tohto dôvodu pre koniec časového okna pre vyzdvihnutie, rovnako ako pre časové okno pre vyloženie, uvažujem iba obmedzenia zvyčajného časového okna pre depo, t. j. začínajúc o siedmej hodine a končiac o sedemnástej.



Obr. 3.5: Histogram počtov dynamických objednávok vytvorených vo vybraných intervaloch počas dňa

4 Popis implementácie

Generátor inštancií predstavený v kap. 3, ako aj generovanie dummy zákazníkov a simulácia realizácie plánu trás, ktoré predstavujem v tejto kapitole, sú súčasťou väčšieho systému (ilustrovanom na obr. 4.1), ktorý primárne slúži pre testovanie navrhnutých stratégií pre generovanie dummy zákazníkov.



Obr. 4.1: Diagram komplexného systému pre testovanie stratégií generovania dummy zákazníkov

Vstupom pre generátor je množina potenciálnych lokalít pre umiestnenie zákazníkov. Generátor produkuje inštanciu statického MAVRP, ktorú následne obohacuje o dummy zákazníkov. Táto obohatená inštancia slúži ako vstup pre algoritmickej riešič firmy Wereldo, ktorého výstupom je pripravený plán trás. Plán trás, spolu s obohatenou inštanciou a množinou dynamických scenárov (taktiež produkovaných generátorom) sú vstupom pre simuláciu. Simulácia krok po kroku realizuje pripravený plán trás, pričom sa snaží doň integrovať prichádzajúce dynamické objednávky do voľného podľa vybraného

dynamického scenára. Simulácia následne ako výstup vypíše koľko dynamických objednávok sa podarilo po plánu vložiť a aké množstvo zdrojov sa k tomu použilo.

V podkapitole 4.1 uvádzam jednu podstratégiu generovania dummy zákazníkov na priestorovej rovine, jednu na časovej rovine, a tri na rovine alokácie zdrojov. Kombináciou týchto podstratégií vzniknú dohromady tri stratégie ($1 \times 1 \times 3$) pre generovanie dummy zákazníkov. Množstvo alokovaných zdrojov pre vybranú stratégiu sa ešte následne odvíja od parametra štedrosti.

Podkapitola 4.2 predstavuje najprv statickú základňu systému, ktorý simuluje realizáciu plánu trás počas dňa, a následne jej rozšírenie o dynamické prvky.

4.1 Stratégie generovania dummy zákazníkov

Vygenerovať dummy zákazníkov znamená vygenerovať objednávky, ktoré sa pridajú do inštancie statického MAVRP. Plán trás pre takúto obohatenú inštanciu v sebe nesie zdroje navyše alokované pozdĺž trás. Plán sa tak stáva flexibilnejším, disponovaným pre integráciu potenciálnych dynamických objednávok. Vstupom pre generátor pri generovaní dummy zákazníkov je množina lokalít potenciálnych zákazníkov. Základné otázky vyvstávajúce pri generovaní dummy zákazníkov sú kde alokovať tieto extra zdroje, ktoré dummy zákazníci predstavujú (podsek. 4.1.1), v akom čase ich tam umiestniť (podsek. 4.1.2), a koľko zdrojov alokovať (podsek. 4.1.3).

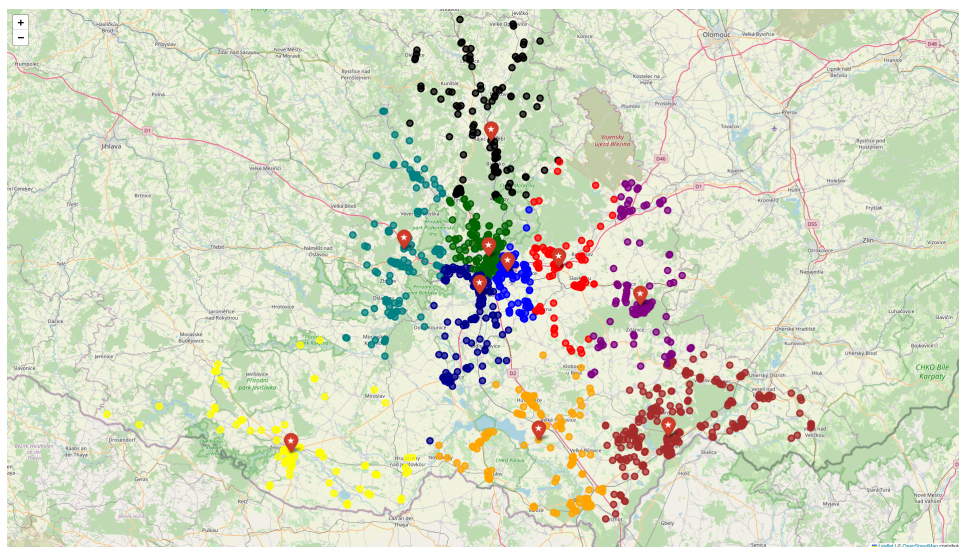
4.1.1 Priestorová rovina

Prvým predpokladom pre obsluhu dynamickej objednávky je schopnosť odkloniť vozidlo z dopredu naplánovanej trasy na miesto príšlej objednávky. Odklon na vzdialené miesto však častokrát nie je možný, a tak je dôležité dopredu predvídať, ktoré miesta, resp. oblasti má význam zaradiť do plánovanej trasy, i keď sa tam nenachádza žiaden dopredu známy zákazník k obsluhu.

Vybraný priestor, kraj v Českej republike, resp. geografické polohy potenciálnych zákazníkov rozdelím do klastrov pomocou algoritmu KMedoids [61]. Klastre v priestore predstavujú zhľuky potenciálnych

zákazníkov, resp. oblasti z ktorých s relatívne vysokou pravdepodobnosťou vstúdu nové objednávky počas dňa. Dummy zákazníkov umiestňujem práve do vnútra týchto klastrov, konkrétne ako ich *medoidy*. Medoid je konkrétny prvok v klastri, ktorý sa najmenej líši od ostatných prvkov. V tomto prípade to znamená prvok, ktorého súčet vzdialeností k ostatným prvkom je minimálny.

Počet klastrov k odvádzam od samotného geografického rozpočítania potenciálnych zákazníkov na vybranom území (o konkrétnej voľbe parametra k píšem v kapitole 5). Na obr. 4.2 je možné vidieť možné rozdelenie potenciálnych zákazníkov do klastrov.



Obr. 4.2: Potenciálni zákazníci v Juhomoravskom kraji rozdelení do desiatich klastrov. Špeciálne označené body predstavujú medoidy jednotlivých klastrov (vizualizácia je vytvorená na základe mapových podkladom OSM).

4.1.2 Časová rovina

Pre pokrytie dynamických objednávok prichádzajúcich počas dňa nestačí pokryť priestorovú rovinu. Vozidlo síce počas dňa môže prísť na miesto blízke lokalite vybranej dynamickej objednávky, avšak môže tam prísť príliš skoro, v čase, kedy o vybranej objednávke ešte systém

nemá informácie alebo príliš neskoro, v čase, kedy objednávku už nie je možné vyzdvihnúť. Inými slovami, pokryť dynamickú objednávku znamená byť schopný nachádzať sa na danej lokalite v čase, kedy je otvorené časové okno pre vyzdvihnutie daného nákladu.

Ako spomínam v kapitole 3, pre zjednodušenie predpokladám, že časové okno pre vyzdvihnutie začína v momente, kedy zákazník objednávku zadá do systému, a je zhora ohraničené iba koncom časového okna pre depo. Analyzované reálne dáta ukazujú, že nové dynamické objednávky prichádzajú takmer výlučne v doobedných hodinách. Za týchto predpokladov, ako jedinú stratégiu volím vytvorenie priestoru pre integrovanie dynamických objednávok v poobedných hodinách, kedy už sú známe skoro všetky dynamické objednávky. Z pravdepodobnostného rozdelenia časov príchodu nových dynamických objednávok (viď obr. 3.5), tretím a zároveň posledným vrcholným bodom je interval od 11:00 do 11:30. Po tomto intervale prišlo už len malé množstvo objednávok, s poslednou objednávkou o 13:22. Ako konkrétny čas pre začiatok časového okna pre vyzdvihnutie volím 12:30. Tento čas pokrýva ešte prudké klesanie z posledného vrcholného bodu, a s ním i väčšinu objednávok v jeho okolí, a zároveň pre pár okrajových hodnôt neukracuje zbytočne z poobedného intervalu pre možné obsluhovanie dynamických objednávok. V simulácii neuvažujem urgentné dynamické objednávky, ktoré už výrazne menia charakter problému, a s predpokladom, že zákazník bude spokojný s vybavením objednávky počas daného dňa, ohraničujem zhora časové okno pre vyzdvihnutie len koncom zvyčajného časového okna pre depo, o 17:00.

4.1.3 Rovina alokácie zdrojov

Ďalším predpokladom pre obsluhu dynamickej objednávky je dostatočná voľná kapacita vozidla a dostatočný čas pre naloženie, resp. vyloženie nákladu. S tým súvisí voľba pre množstvo zdrojov, ktoré chceme pre dynamické objednávky alokovať. Z analýzy reálnych dát, poznania DoD a počtu zákazníkov vo vybranom klastri odvádzam očakávaný počet dynamických objednávok vyšších z daného klastru a s ním i očakávané množstvo zdrojov potrebných pre ich obsluhu nasledovne:

$$\text{fraction}(k_i) = \frac{\text{locations}(k_i)}{\text{pickups}} \quad (4.1)$$

$$\text{requests}_{\text{dyn}}(k_i) = n \times \text{DoD} \times \text{fraction}(k_i) \quad (4.2)$$

$$\text{weight}_{\text{alloc}}(k_i) = \text{weight}_{\text{exp}} \times \text{requests}_{\text{dyn}}(k_i) \quad (4.3)$$

$$\text{time}_{\text{alloc}}(k_i) = (\text{time}_{\text{exp}} + \text{time}_{\text{travel}}(k_i)) \times \text{requests}_{\text{dyn}}(k_i) \quad (4.4)$$

- $\text{locations}(k_i)$ - počet lokalít v klastri k_i
- pickups - celkový počet pickup lokalít v kandidátnej množine
- $\text{requests}_{\text{dyn}}(k_i)$ - očakávaný počet dyn. objednávok z klastru k_i
- n - celkový počet objednávok
- cluster_i - klaster s indexom i
- $\text{time}_{\text{alloc}}(k_i)$ - čas alokovaný pre klaster k_i
- $\text{weight}_{\text{alloc}}(k_i)$ - hmotnosť alokovaná pre klaster k_i
- $\text{weight}_{\text{exp}}$ - priemerná hmotnosť u objednávky
- time_{exp} - priemerný pickup service time
- $\text{time}_{\text{travel}}(k_i)$ - očakávaný čas pre cestovanie medzi dvoma lokalitami v klastri k_i

Voľba potom spočíva v určení násobku, resp. zlomku tohto očakávaného množstva zdrojov potrebných pre obsluhu dynamických objednávok, ktoré alokujem. Tento násobok budem označovať ako *miera štedrosti* (o konkrétnom nastavení tohto parametra píšem v kapitole 5).

U času alokovaného pre service time je možné si všimnúť špecifický prírastok. I keď medoid slúži ako vhodný odrazový mostík, na pokrytie lokalít v klastri nestačí. Z tohto dôvodu k servis času dummy zákazníka pridávam očakávaný čas potrebný pre cestovanie medzi dvoma náhodnými lokalitami v rámci klastra (nakolko dummy zákazník môže predstavovať zdroje alokované pre viaceré dynamické objednávky, nevolím očakávaný čas potrebný pre cestovanie medzi

medoidom a náhodnou lokalitou v rámci klastra). Týmto spôsobom dummy zákazník priestorovo pokrýva značnú časť klastra.

Na kapacitnej rovine uvádzam tri stratégie. Prvou, priamočiarou stratégiou je umiestniť celkové množstvo alokovaných zdrojov pre daný klaster do jeho stredu, resp. medoidu ako jedného dummy zákazníka, resp. jednu dummy objednávku. Nevýhodou tejto stratégie je fakt, že ako jedna objednávka môže byť obslužená čisto jedným vozidlom, čo môže byť náročné pri vyššom množstve alokovaných zdrojov. V praxi tieto alokované zdroje môžu pokryť viaceré menšie objednávky, a môžu tak byť obslužené viacerými vozidlami.

Druhou stratégiou je celkové množstvo alokovaných zdrojov pre daný klaster rozdeliť na viacero menších dummy objednávok, konkrétne na objednávky o veľkosti jednej palety (v priemere okolo 228 kg). Pravdaže s vyšším počtom vozidiel, ktoré sa na objednávke podieľajú, sa zvyšuje aj celkový čas pre jej obsluženie. S touto stratégiou sa tak viaže alokácia väčšieho množstva času, ktorá túto účasť viacerých vozidiel pomáha umožniť. Avšak na obsluhu jednej objednávky sa zvyčajne nepodieľajú viac ako dve vozidlá, čo má za následok, že častokrát dôjde k prebytočnému alokovaniu času, najmä v prípade väčšieho množstva paletového miesta alokovaného pre klaster.

Ako stredná cesta má slúžiť tretia stratégia, pri ktorej je celkové množstvo alokovaných zdrojov pre daný klaster rozdelené na tzv. normo-dummy zákazníkov, kde jeden normo-dummy zákazník predstavuje objednávku s jednotlivými atribútmi rovnými priemerným hodnotám v analyzovaných dátach (približne 700 kg, 3 palety a 15 minútový pickup service čas).

4.2 Simulácia realizácie plánu trás počas dňa

Simulácia predstavuje *event-driven system*, ktorý simuluje realizáciu plánu trás počas dňa. Vstupmi pre simuláciu sú inštancia MAVRP obohatená o dummy zákazníkov, plán trasy pre túto inštanciu a scenár pre príchod dynamických objednávok počas dňa. Výstupom simulácie je súhrn údajov z vykonaného plánu trás. Konkrétne počet použitých vozidiel, celková najazdená vzdialenosť, celkový pracovný čas, počet obslužených dynamických objednávok a čas behu simulácie. Tieto metriky sa viažu na realizáciu plánu, ktorá v reakcii na dynamické

objednávky či nevyužitú alokované zdroje pôvodný plán upravuje. Zachytené údaje slúžia pre vyhodnocovanie stratégií pre generovanie dummy zákazníkov.

Simulácia je vnútorne vystavaná ako systém riadený udalosťami, ktoré určujú jeho stav, resp. prechody medzi stavmi, v ktorých sa nachádza. Stav systému je daný aktuálnym časom, polohou a stavom vozidiel, a plánom pre následné pokračovanie trás. Stav vozidla je daný aktuálnym zaťažením a naplnením jeho objemovej kapacity ako aj charakterom činnosti, ktorú vozidlo momentálne vykonáva – či cestuje, čaká alebo nakladá či vykladá tovar.

Simulácia rozlišuje dva typy udalostí, lokálne a globálne. Zatiaľ čo lokálne udalosti ovplyvňujú iba dané vozidlo, globálne udalosti majú vplyv na celý systém a vyžadujú si špecifické algoritmické kroky.

Až na príchod globálnych udalostí, simulácia jednoducho vykonáva dopredu daný statický plán. Plán trás sa vykonáva po krokoch, resp. sa posúva dopredu v čase o užívatelom stanovený časový interval, *tick*. S časom systém vnútorne pracuje tak, že každému vozidlu pridelí časový budget, s ktorým dané vozidlo pracuje, a spotrebúva na cestovanie, čakanie či nakladanie (resp. vykladanie). Systém vozidlám prideluje vždy len časový budget, ktorý nepresahuje čas najbližšej globálnej udalosti. Inými slovami, v čase príchodu globálnej udalosti končí časový budget každého vozidla, a nový budget je pridelený až po dokončení obslužných funkcií spojených s príslušnou globálnou udalosťou.

4.2.1 Lokálne udalosti

Lokálnymi udalosťami sú príchod vozidla na zastávku, začiatok nakladania tovaru a odchod zo zastávky. Tieto udalosti nastanú, keď sa aktuálny čas simulácie rovná času, kedy táto udalosť nastane v pláne. Lokálne udalosti si vyžadujú iba aktualizáciu stavu, napr. zmenu momentálne naplnenej kapacity vozidla alebo odstránenie zastávky, ktorú už vozidlo navštívilo z plánu pre pokračovanie jeho trasy.

4.2.2 Globálne udalosti

Globálnou udalosťou v implementovanom systéme je príchod novej dynamickej objednávky. V momente, kedy príde nová dynamická

objednávka, sa ju systém snaží integrovať do aktuálneho plánu trás. Časy príchodov dynamických objednávok sú súčasťou generátorom vygenerovaných dynamických scenárov. Plány, s ktorými pracujem, už v sebe nesú proaktívne opatrenia pre integráciu potenciálnych dynamických objednávok v podobe dummy zákazníkov umiestnených do trás. Títo dummy zákazníci, predstavujúci alokované zdroje pre možné dynamické objednávky, sú práve priestorom, do ktorého sa systém snaží vložiť dynamickú objednávku.

Vkladanie dynamických zákazníkov do plánu preto prebieha stotožňovaním dynamických zákazníkov s dummy zákazníkmi v pláne. Systém vyberie jedného dummy zákazníka, s ktorým chce dynamickú objednávku stotožniť, resp. ktorou ho chce nahradiť, spočíta, či je táto operácia možná, a v prípade pozitívnej odpovede, dummy zákazníka nahradí dynamickou objednávkou. Toto nahradenie môže byť tak tiež čiastočné. Množstvo alokovaných zdrojov, ktoré vybraný dummy zákazník reprezentuje môže značne presahovať zdroje potrebné pre obsluhu prišlej dynamickej objednávky. V takomto prípade systém dynamickú objednávku vloží do trasy pred dummy zákazníka, a dummy zákazníkovi odoberie len také množstvo alokovaných zdrojov, ktoré postačia pre obsluhu danej dynamickej objednávky.

Dummy zákazníkov, s ktorými je prišlú dynamickú objednávku možné stotožniť, môže byť viacero. Systém, ktorý som naimplementoval radí dummy zákazníkov ako kandidátov na stotožnenie v poradí danom vzdialenosťou od miesta vyzdvihnutia dynamickej objednávky, od najbližšieho po najvzdialenejšieho. Prvým, najbližším kandidátom je tak priamo medoid klastru, z ktorého vzíde dynamická objednávka.

Vzhľadom na jednu z použitých stratégií pre generovanie dummy zákazníkov, ktorá nahrádza jedného dummy zákazníka, väčším počtom menších dummy zákazníkov (vzhľadom na množstvo alokovaných zdrojov), predstavujem v simulácii koncept tzv. *super-dummy zákazníka*. Jedná sa o objekt reprezentujúci všetkých dummy zákazníkov na jednej fyzickej lokalite v rámci jednej trasy. Týmto spôsobom systém môže jednoduchšie operovať s alokovanými zdrojmi na vybranej lokalite.

V prípade, že vozidlo vyráža zo zastávky, a nasledujúcou zastávkou by bol dummy zákazník, simulácia dummy zákazníka na nasledujúcej zastávke odstráni z plánu a vozidlo vyrazí na ďalšiu zastávku.

5 Experimenty

Cieľom experimentu je otestovať a následne vyhodnotiť navrhnuté stratégie pre generovanie dummy zákazníkov.

Prvá podkapitola opisuje prípravu dátových podkladov problémov, na ktorých je experiment vyhodnocovaný. Druhá podkapitola vykresľuje pracovný postup experimentu. V tretej kapitole je obsiahnuté ladenie vstupných parametrov. Štvrtá podkapitola prezentuje výsledky experimentu a následné súhrnné zhodnotenie pre každú z navrhnutých stratégií.

Výpočty v rámci experimentu prebiehali na výpočtovom klastri Kirke, v rámci zdrojov poskytnutých MetaCentrom [62]. Stroje v rámci klastra Kirke používajú OS Debian 12 a disponujú 64 procesormi AMD EPYC 7532. Výpočty boli paralelizované na 16 jadrách, s celkovým použitím RAM 12 GB.

5.1 Datasetsy

Pre dátové podklady problémov, na ktorých je experiment vyhodnocovaný sú použité syntetické dáta vytvorené naimplementovaným generátorom. Dátaset sa skladá z viacerých inštancií statického MAVRP, množín lokalít potenciálnych zákazníkov, ktoré s nimi korešpondujú, a scenárov počas dňa prichádzajúcich dynamických objednávok.

Pre geografickú oblasť, v ktorej sa experiment odohráva bol vybraný Juhomoravský kraj (JHM), primárne v ktorom firma Wereldo pôsobí. Tento kraj obsahuje približne jeden a pol tisíc lokalít v OSM označených tagom *landuse=industrial*, ktoré predstavujú kandidátne lokality pre potenciálnych zákazníkov (viď podsek. 3.1.1). Vzhľadom na to, že s tak veľkým počtom potenciálnych zákazníkov pracujú značne väčšie firmy, množinu lokalít potenciálnych zákazníkov vyberám spo-medzi podmnožín kandidátnych lokalít v JHM s kardinalitou 428 (30 % kandidátnych lokalít).

Každý z problémov, s ktorými pracujem obsahuje 150 objednávok, z toho časť je dopredu známa, a zvyšné, dynamické objednávky sa objavujú postupne počas dňa. Problémy objavujúce sa v dataseete majú rôzne stupne dynamizmu (DoD), konkrétne 5 %, 10 %, 15 % a 20 %. Dopredu známe inštancie sú súčasťou inštancie statického MAVRP,

zatiaľ čo dynamické sa objavujú v scenároch (počas dňa prichádzajúcich dynamických objednávok). S každou inštanciou sa viaže viacero možných scenárov, pričom každá inštancia má vlastný súbor scenárov. Konkrétne sa jedná o súbor 10 000 scenárov pre inštanciu.

V inštancii pracujem s množinou vozidiel, ktoré sú typovo podobné vozidlám, ktoré používa firma Wereldo. Počet vozidiel som stanovil na 32. Vysoký počet vozidiel sa odvíja od dummy zákazníkov, ktorými bude inštancia doplnená. Vysoký počet vozidiel slúži tiež ako preventívne opatrenie, aby sa zamedzilo výskytu neobslúžených objednávok v pláne trás pre danú inštanciu. Plánu trás, s neobslúženými objednávkami totiž riešič firmy Wereldo udeľuje penalizáciu vo forme pridanej prejdenej vzdialenosti, ktorá by výsledky dát mohla značne skresliť.

Vytvorenie datasetu prebieha v troch nasledovných krokoch. Prvý krok spočíva vo vygenerovaní n podmnožín z množiny kandidátnych lokalít pre potenciálnych zákazníkov v JHM (kde n je parameter experimentu). Voľbe n predchádza ladenie parametrov vo forme nultého experimentu. Druhým krokom je vygenerovať n inštancií statického MAVRP korešpondujúcich s vygenerovanými podmnožinami ($\frac{n}{4}$ pre každú hodnotu DoD). Tretí krok spočíva vo vygenerovaní súboru 10 000 scenárov (príchodu dyn. objednávok) pre každú inštanciu.

Tabuľka 5.1: Parametre pre dataset a ich zvolená hodnota

Parameter	Hodnota
Geograf. oblasť	JHM
Počet potenciálnych zákazníkov	428
Počet objednávok v probléme	150
Počet inštancií	n
DoD	5 %, 10 %, 15 %, 20 %
Počet scenárov pre inštanciu	10 000
Počet vozidiel	32
Typ flotily	Heterogénna

5.2 Postup

Experiment slúži pre vyhodnotenie navrhnutých stratégií pre generovanie dummy zákazníkov. Navrhnuté stratégie sú tri. Realizácia týchto stratégií závisí od dvoch parametrov, s ktorými tieto stratégie operujú. Prvým parametrom je miera štedrosti, ktorá určuje násobok, resp. zlomok očakávaného množstva zdrojov potrebných pre obsluhu dynamických objednávok, ktoré alokujem. Parameter označujem veľkým písmenok G , nakoľko v experimente sa nejedná len o jednu konkrétnu hodnotu, ale o množinu hodnôt, resp. škálu štedrosti. Druhým parametrom je počet klastrov k , do ktorých rozdeľujem lokality potenciálnych zákazníkov, a do ktorých medoidov umiestňujem dummy zákazníkov. Tieto parametre stratégií generovania dummy zákazníkov nie sú implicitne určené, ale sú predmetom skúmania a ich voľba je výsledkom ladenia parametrov v rámci nultého experimentu. K trom navrhnutým stratégiám sa pridá ešte stratégia, ktorá nevkladá do inštancie žiadnych dummy zákazníkov. Táto identita, ktorá ponecháva inštanciu nezmenenú slúži ako referenčný bod pre navrhnuté stratégie.

Navrhnuté stratégie generovania dummy zákazníkov obohatia inštanciu statického MAVRP, aby vzniklý plán trás pre túto inštanciu bol flexibilnejší a schopný integrovať dynamické objednávky. Plán trás pre obohatenú inštanciu získavam ako riešenie, výstup z algoritmickeho riešiča firmy Wereldo. Nakoľko riešič nevráti pri opakovanom behu na danom vstupe vždy rovnaké riešenie, t. j. vnáša do vyhodnocovania stratégií istú stochasticitu, opakujem beh pre každú obohatenú inštanciu l -krát. Voľbe l predchádza ladenie parametrov vo forme nultého experimentu. Časový limit pre hľadanie riešenia bol zvolený na 5 minút. Jedná sa o relatívne štedrý čas, ktorý má zabrániť prílišnej variabilite v riešeníach pre danú inštanciu.

Riešenia obohatených inštancií, spolu so samotnými obohatenými inštanciami a scenármi príchodu dynamických objednávok sú vstupom pre simuláciu realizácie plánu trás. Pre každý scenár, s ktorým simulácia spustí danú obohatenú inštanciu sa do CSV súboru zapíše riadok s vybranými údajmi, resp. metrikami, ktoré zaznamenávame. Konkrétne sa jedná o tieto údaje: počet použitých vozidiel, celková najazdená vzdialenosť, celkový pracovný čas, počet obslužených dynamických objednávok a čas behu simulácie. Jedná sa o metriky viazané

na realizáciu plánu trás simuláciou, ktorá v reakcii na dynamické objednávky či nevyužité alokované zdroje pôvodný plán upravuje. Čas behu simulácie sa viaže na jeden scenár príchodu dynamických objednávok. Na základe týchto metrík sú evaluované navrhnuté stratégie generovania dummy zákazníkov.

Priebeh experimentu na už vygenerovanom datasete prebieha teda v štyroch krokoch. Prvý krok spočíva v obohatení každej z n inštancií statického MAVRP s pomocou troch stratégií generovania dummy zákazníkov, pri zvolenom počte klastrov k a s mierou štedrosti rovnou postupne každej hodnote zo škály G . Druhým krokom je nájdenie riešenia, t. j. plánu trás pre $n \times 3 \times |G|$ obohatených inštancií pomocou algoritmickeho riešiča firmy Wereldo. Každú inštanciu vložíme do riešiča l -krát. Tretím krokom je spustenie simulácie s príslušnými vstupmi. Štvrtým krokom je analýza a vyhodnotenie dát zozbieraných z výstupov simulácie.

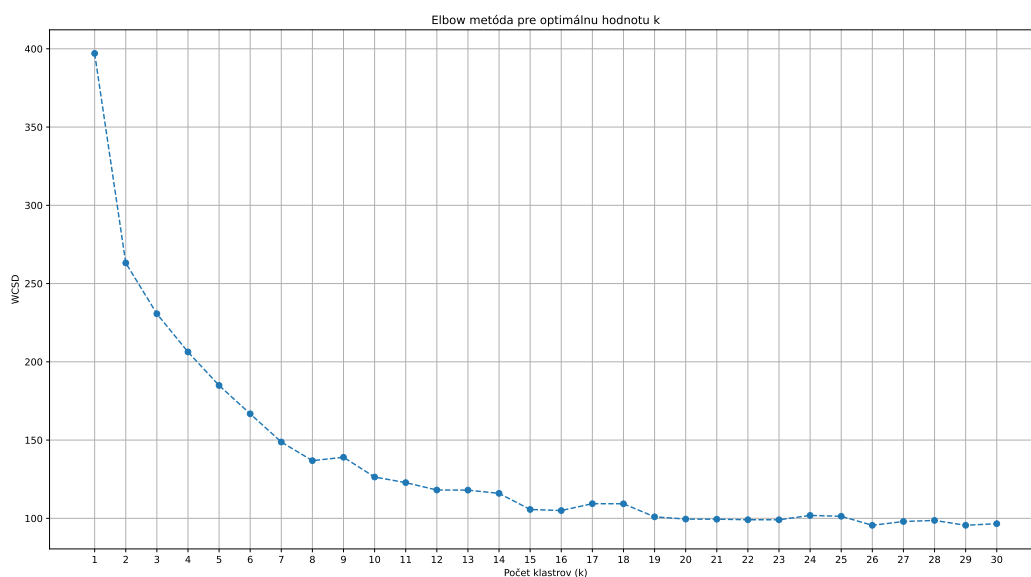
Tabuľka 5.2: Parametre experimentu

Parameter	Hodnota
Počet stratégií pre generovanie dummy zákazníkov	$3 + 1$
Škála štedrosti	G
Počet klastrov	k
Počet riešení pre inštanciu	l
Časový limit pre výpočet riešenia inštancie	5 minút
Simuláciou merané metriky	Počet použitých vozidiel, celková najazdená vzdialenosť, celkový pracovný čas, počet obslužených dynamických objednávok, čas behu simulácie

5.3 Ladenie parametrov

Prvým parametrom, ktorý nie je implicitne zvolený, je počet klastrov k . Voľbu k odvádzam od samotného geografického rozloženia lokalít

potenciálnych zákazníkov v rámci danej oblasti. Kvalita klastrov sa štandardne odvíja od dvoch kvalít. Prvou je podobnosť prvkov v rámci jedného klastra. Druhou je odlišnosť prvkov v rámci rôznych klastrov. Podobnosť, resp. odlišnosť prvkov v kontexte dizajnu experimentu sa viaže na vzdialenosť prvkov. Vzhľadom na to, že sa snažíme minimalizovať vzdialenosť potenciálnej dynamickej objednávky od dummy zákazníka, t. j. od medoidu daného klastra, dôležitejšou kvalitou pre nás je kompaktnosť klastrov, čiže malá vzdialenosť lokalít v rámci jedného klastra. Pre výber vhodnej hodnoty k som použil Elbow metódu, ktorá meria sumu štvorcov vzdialeností bodov od centra klastra [61]. S poklesom na y-osi klesá i nekompaktnosť klastrov. Avšak s veľkým nárastom na x-osi, sa klastre môžu stať príliš malé. Navyše počet klastrov by mal byť vhodný pre ľubovoľnú podmnožinu z kandidátnej množiny lokalít, čo môže byť problém pri vysokom počte klastrov. Vhodné voľby pre k sa tak pohybujú v ľavej polovici vizualizácie Elbow metódy na obr. 5.1, kde nekompaktnosť klesá za cenu malého nárastu počtu klastrov. Ako konkrétnu voľbu k som vybral hodnotu 10. Ako vhodné alternatívy na základe Elbow metódy by mohli poslúžiť i hodnoty ako 8, 12 či 15.



Obr. 5.1: Vizualizácia Elbow metódy pre interval od jedna po tridsať klastrov

Voľba zvyšných parametrov, ktoré neboli implicitne určené (n , G a l) bola vykonaná na základe nultého experimentu. Priebeh nultého experimentu bol rovnaký ako u samotného experimentu až na vybrané parametre. Nultý experiment používal dataset s 12 inštanciami statického MAVRP (n). Stratégie generovania dummy zákazníkov boli realizované s parametrami $k = 10$ (rovnako ako v ostrom experimente) a $G = \{0.2, 0.4, \dots, 2\}$. Nakoniec, pre každú obohatenú inštanciu boli spočítané 3 riešenia (l). Medzi plánmi trás pre jednu inštanciu bola pomerne vysoká variabilita v riešeníach, z čoho som usúdil, že tri riešenia pre inštanciu sú málo. Alokácia dvojnásobku očakávaných zdrojov potrebných pre obsluhu dynamických objednávok bola zbytočne vysoká, a tak som znížim horné ohraničenie škály štedrosti na 1.7, pričom som dvojnásobne zvýšil granularitu, aby bolo možné presnejšie pozorovať zmeny v sledovaných metrikách. Nakoniec som rozšíril dataset na 20 inštancií statického MAVRP.

5.4 Experiment

Na základe Elbow metódy pre výber počtu klastrov k a výsledkov z nultého experimentu som zvolil hodnoty pre zvyšné parametre n , G a l . Dataset obsahuje 20 inštancií statického MAVRP, používajúc vybraných 20 korešpondujúcich podmnožín z kandidátnej množiny potenciálnych zákazníkov ($n = 20$). Počet klastrov, do ktorých rozdeľujem lokality potenciálnych zákazníkov je 10 ($k = 10$). Každá stratégia generovania dummy zákazníkov sa vyhodnocuje pre miery štedrosti zo škály $G = \{0.1, 0.2, \dots, 1.7\}$. Vzniklo tak 1020 inštancií obohatených o dummy zákazníkov (20 zákl. inštancií $\times 3$ stratégie $\times 17$ hodnôt na škále štedrosti). Pre každú inštanciu bolo spočítaných 10 riešení. Spolu 10 200 riešení obohatených inštancií.

Ako referenčný bod pre navrhnuté stratégie generovania dummy zákazníkov slúžia namerané hodnoty metrík pre nultú stratégiu, ktorá ponecháva zákl. inštanciu nezmenenú (viď obr. 5.4).

Algoritmický riešič firmy Wereldo je naimplementovaný tak, aby primárne minimalizoval u plánu trás najazdenú vzdialenosť. Z tohto dôvodu podobne analýza výsledkov z experimentu sa primárne zameriava práve na najazdenú vzdialenosť vo vzťahu k počtu obslužených

Tabuľka 5.3: Ladené parametre pre ostrý experiment

Parameter	Hodnota
Počet zákl. inštancií	$n = 20$
Počet klastrov	$k = 10$
Škála štedrosti	$G = \{0.1, 0.2, \dots, 1.7\}$
Počet riešení pre inštanciu	$l = 10$

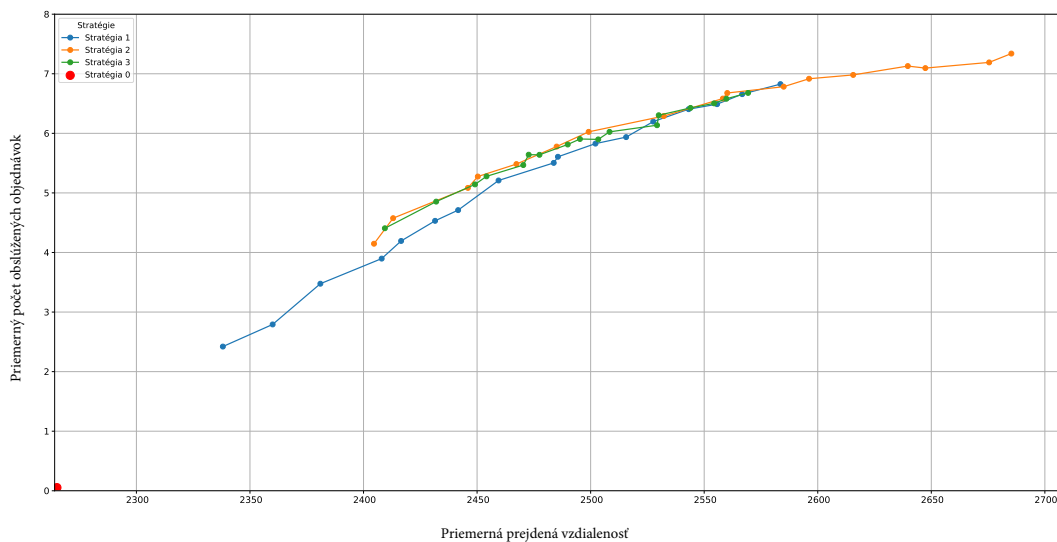
Tabuľka 5.4: Priemerné hodnoty metrík u plánov trás bez dummy zákazníkov

DoD	Počet vozidiel	Najazdená vzdialenosť (km)	Pracovný čas (h)
5	14.08	2264.27	94.93
10	12.8	2052.62	86.39
15	12.46	1829.69	81.80
20	12.6	2045.06	82.80

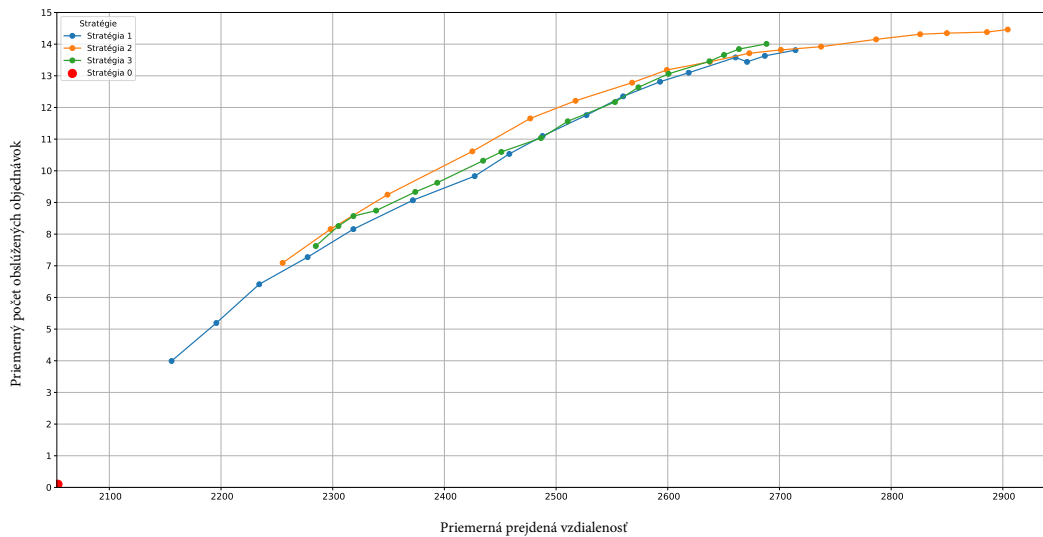
dynamických objednávok. Do úvahy však budeme brať i počet použitých vozidiel i ďalšie metriky.

Prvým skúmaným vzťahom je závislosť počtu obslužených objednávok od najazdenej vzdialenosti. Táto závislosť pre danú hodnotu DoD je zachytená na obrázkoch 5.2, 5.3, 5.4 a 5.5.

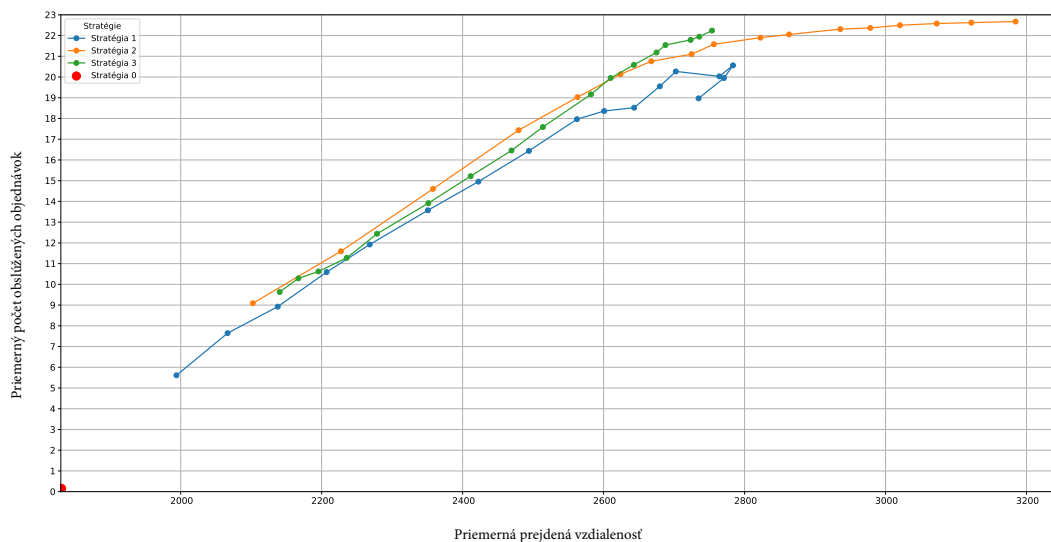
Prvým pozorovaním je, že prvá stratégia pri DoD 15% a 20% vytvára pri vysokej miere štedrosti vo veľkých klastroch tak veľkých dummy zákazníkov, že sa stávajú neobslúžiteľnými, resp. veľmi ťažko obslužiteľnými. To vysvetľuje nečakaný pokles najazdenej vzdialenosti spôsobený tým, že niektoré zákazky, ktoré riešiť nevedel zakomponovať do plánu trás, nám z neho vypadli. Napríklad pre DoD 20% v 11. inštancii vznikne pri miere štedrosti 1.4 dummy zákazník presahujúci maximálnu kapacitu u všetkých vozidiel. Táto degradácia prvej stratégie pri vyšších hodnotách DoD je postupná. Počnúc prvým neobslúženým dummy zákazníkom v najväčšom klastri. Následne ako sa zvyšuje miera štedrosti, vznikajú prívětý dummy zákazníci v ďalších



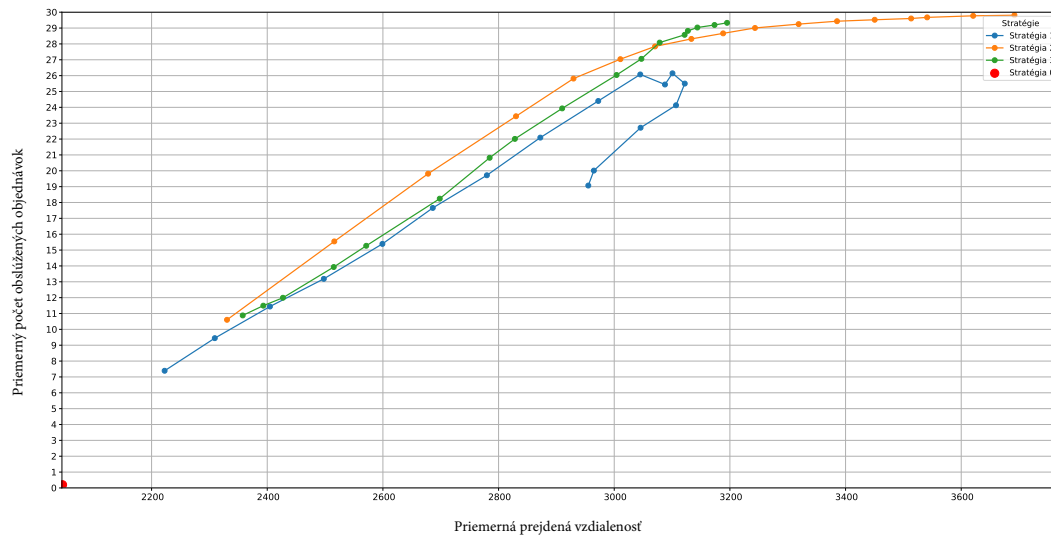
Obr. 5.2: Závislosť počtu obslužených dyn. objednávok od prejdenej vzdialenosti (DoD: 5)



Obr. 5.3: Závislosť počtu obslužených dyn. objednávok od prejdenej vzdialenosti (DoD: 10)



Obr. 5.4: Závislosť počtu obslužených dyn. objednávok od prejdenej vzdialenosti (DoD: 15)



Obr. 5.5: Závislosť počtu obslužených dyn. objednávok od prejdenej vzdialenosti (DoD: 20)

klastroch (alebo v ďalších inštanciách s o niečo menším najväčším klastrom), a vypadávajú z plánu trás.

Druhým pozorovaním je fakt, že druhá stratégia má pri danej miere štedrosti výrazne vyššiu prejdenú vzdialenosť ako zvyšné stratégie. To je dané tým, že v prípade druhej stratégie alokujeme pre daný klaster nadmerný čas, ktorý slúži na to, aby mohla byť potenciálne každá paleta obslužená iným vozidlom. Naopak prvá stratégia je schopná obslúžiť niekoľko dynamických objednávok už pri veľmi malej vzdialenosti prejdenej navyše.

Tretím pozorovaním je, že druhá stratégia v značnom miere dominuje. Keď sa však druhá stratégia začne blížiť k maximálnemu počtu obslužených objednávok, jej rast sa spomalí výraznejšie než u tretej stratégie, ktorá ku maximálnemu počtu obslužených zákazníkov (v poslednom úseku) konverguje rýchlejšie, resp. s nižšou najazdenou vzdialenosťou. Dominancia stratégie vzhľadom na prejdenú vzdialenosť však ešte neznamena dominanciu vo všeobecnosti.

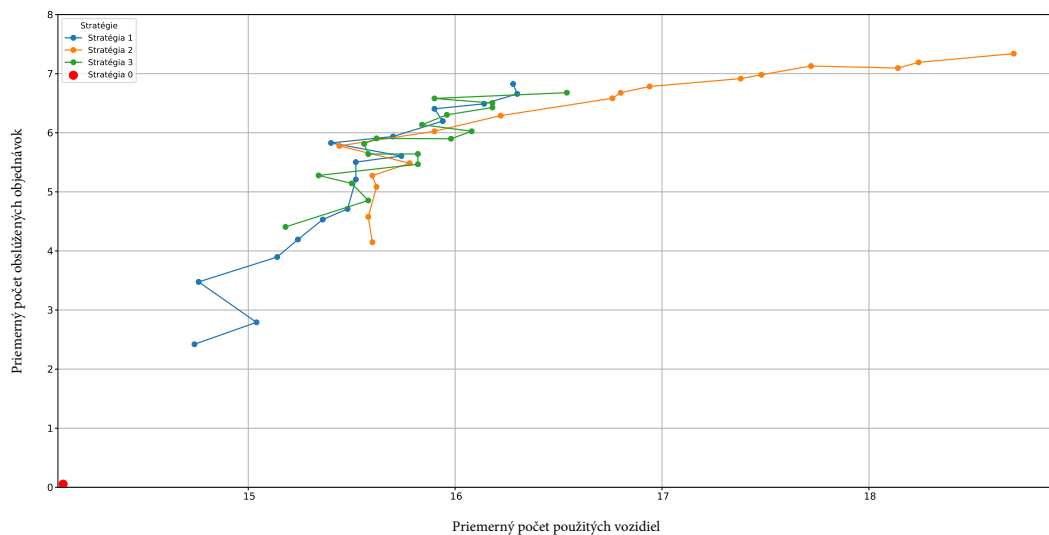
Je taktiež možné si všimnúť, že s rastúcim DoD sa trendy udávané stratégiami stávajú stabilnejšími. Variabilita, resp. rozptyly v hodnotách najazdenej vzdialenosti, sú pomerne stabilné (viď B.1 a B.2), majú väčší relatívny dopad u nižších DoD.

Druhým skúmaným vzťahom je závislosť počtu obslužených objednávok od počtu použitých vozidiel. Táto závislosť pre danú hodnotu DoD je zachytená na obrázkoch 5.6, 5.7, 5.8 a 5.9.

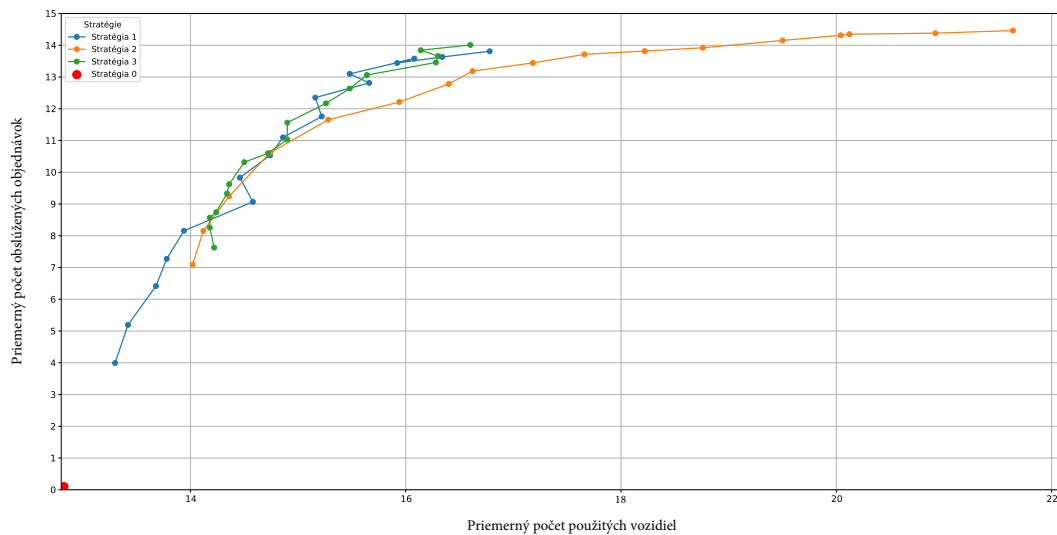
Rovnako ako u závislosti počtu obslužených objednávok od najazdenej vzdialenosti, sledujeme tú istú degradáciu prvej stratégie pri DoD 15 a 20% pri vyšších hodnotách miery štedrosti. Podobne sledujeme, že druhá stratégia pri danej miere štedrosti využíva značne viac vozidiel. Naproti tomu, prvá stratégia už pri veľmi nízkom počte vozidiel navyše dokáže obslúžiť niekoľko dynamických objednávok.

S rastúcim DoD sa trendy udávané stratégiami stávajú stabilnejšími, čo je dané poklesom rozptylu u počtu použitých vozidiel (viď B.1 a B.2).

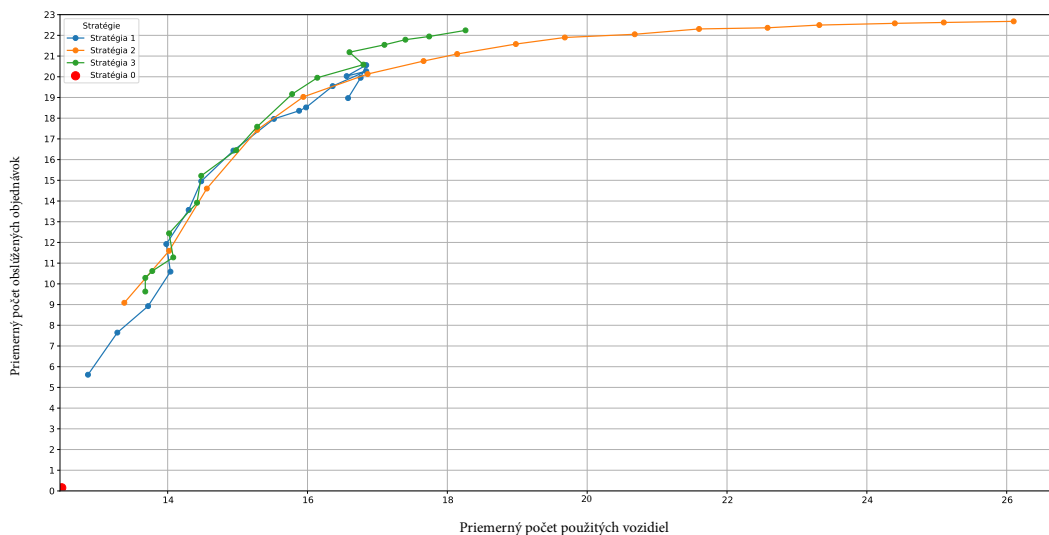
Pri porovnaní stratégií je možné si všimnúť dominanciu prvej a tretej stratégie nad druhou stratégiou. Pričom prvá stratégia pri DoD 15 a 20 % dominanciu stráca kvôli spomenutej degradácii. Naopak dominancia tretej stratégie pri DoD 20 % je už veľmi výrazná. Opäť musíme brať do úvahy, že sa jedná len o dominanciu vzhľadom na jednu z metrík ceny, ktorú zaplatíme za obsluženie dynamických



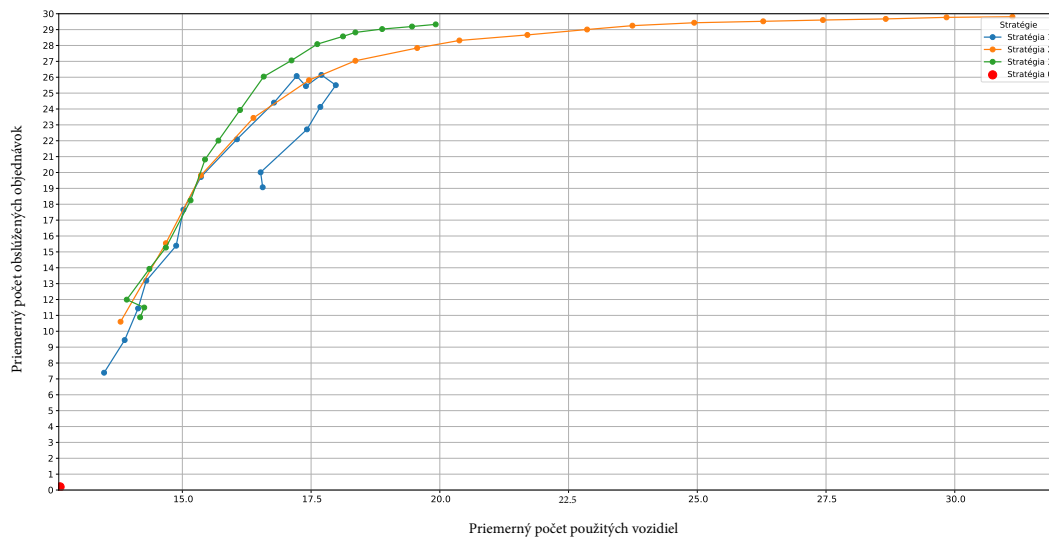
Obr. 5.6: Závislosť počtu obslužených dyn. objednávok od počtu použitých vozidiel (DoD: 5)



Obr. 5.7: Závislosť počtu obslužených dyn. objednávok od počtu použitých vozidiel (DoD: 10)



Obr. 5.8: Závislosť počtu obslužených dyn. objednávok od počtu použitých vozidiel (DoD: 15)



Obr. 5.9: Závislosť počtu obslužených dyn. objednávok od počtu použitých vozidiel (DoD: 20)

objednávok. Rovnako je potrebné brať do úvahy, že algoritmicke riešič firmy Wereldo nemá explicitne danú preferenciu pre nízky počet vozidiel.

Tretou skúmanou veličinou, ktorá úzko súvisí s počtom použitých vozidiel je priemerný pracovný čas pripadajúci na jedno vozidlo. Vo väčšine prípadov je žiadúce, aby vozidlá i šoféri boli dostatočne vyťažení počas celej pracovnej doby. Pri pohľade do tabuľky 5.5 je možné si všimnúť nižší priemerný pracovný čas u druhej stratégie, rozdiel sa stáva výrazným pri vyšších DoD.

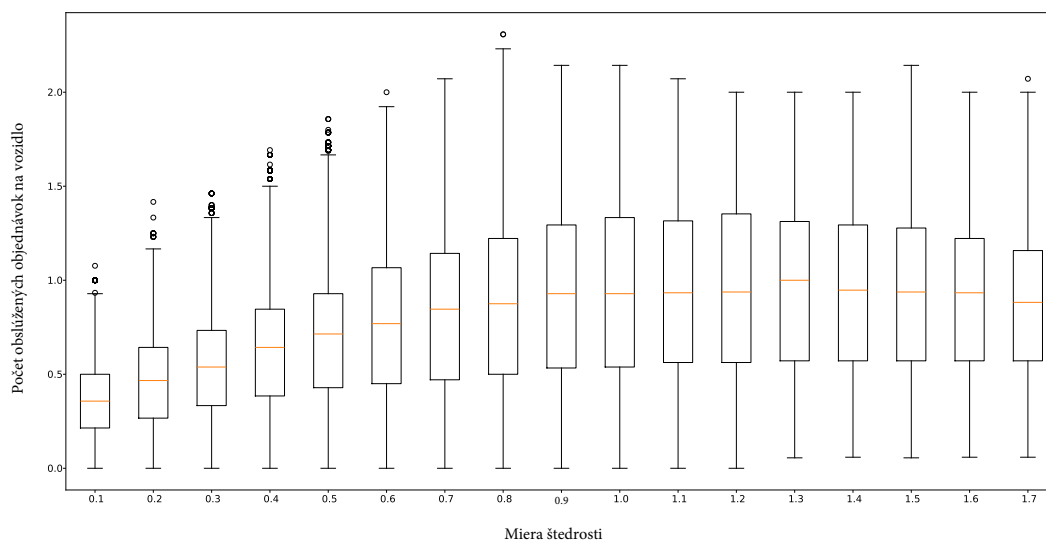
Tabuľka 5.5: Priemerný pracovný čas jedného vozidla

DoD	Priemerný pracovný čas pri str. I. (h)	Priemerný pracovný čas pri str. II. (h)	Priemerný pracovný čas pri str. III. (h)
5	6.59	6.33	6.53
10	6.65	6.11	6.62
15	6.54	5.85	6.52
20	6.54	5.65	6.54

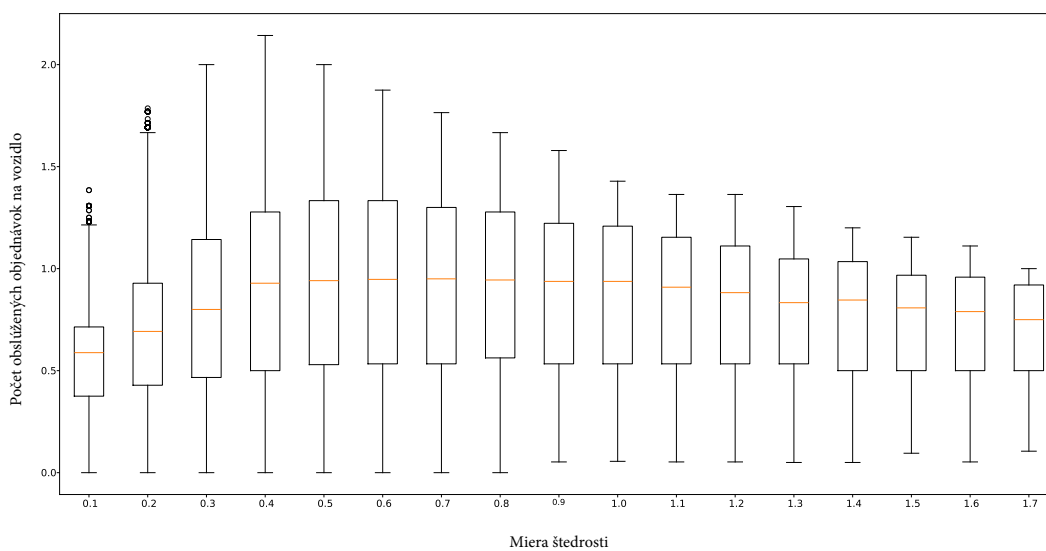
Štvrtou a poslednou skúmanou veličinou je priemerný počet obslužených dynamických objednávok pripadajúcich na jedno vozidlo (viď obr. 5.10, 5.11 a 5.12). Túto informáciu síce ide priamo vyčítať už z načrtnutých grafov zobrazujúcich závislosť počtu obslužených dyn. objednávok od počtu použitých vozidiel, avšak týmto spôsobom lepšie vyniknú miery štedrosti, pri ktorých dochádza k efektívnemu využitiu vozidiel pre obsluhu dynamických objednávok.

U prvej stratégie vrchol nastáva pri miere štedrosti 1 – 1.2. Vrchol u druhej stratégie nastáva pri miere štedrosti 0.5 – 0.6. Tretia stratégia vrcholí efektívnym využitím vozidiel pre obsluhu dyn. objednávok pri miere štedrosti 1.3. Pri pohľade na túto veličinu si však je vhodné uvedomiť, že nám nepodáva informáciu o tom, koľko vozidiel bolo do obsluhy dyn. objednávok zapojených. Neumožniac detekovať či dyn. objednávky neobsluhuje len veľmi malé množstvo vybraných vozidiel.

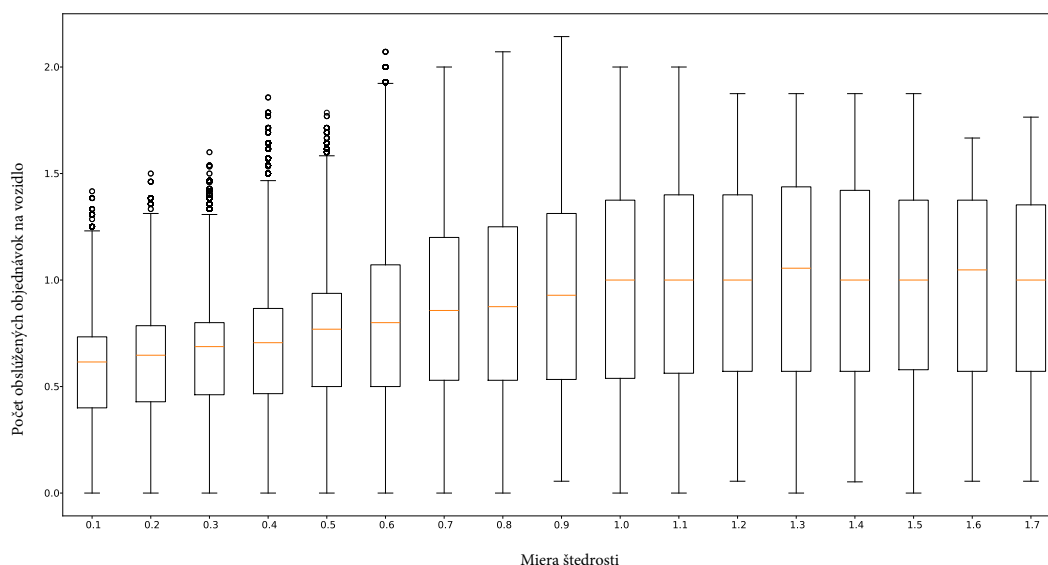
Meraný čas behu simulácie pre jeden scenár príchodu dynamických objednávok je udaný v milisekundách, s priemernou hodnotou asi 9.22 ms. Jedná sa tak o relatívne výpočtovo lacnú operáciu



Obr. 5.10: Priemerný počet obslužených dyn. objednávok pripadajúcich na jedno vozidlo (stratégia I.)



Obr. 5.11: Priemerný počet obslužených dyn. objednávok pripadajúcich na jedno vozidlo (stratégia II.)



Obr. 5.12: Priemerný počet obslužených dyn. objednávok pripadajúcich na jedno vozidlo (stratégia III.)

z hľadiska času. Pre hrubý prehľad, vygenerovanie jednej základnej inštancie, bez dummy zákazníkov, trvá asi 14 sekúnd, kde najdrahšou operáciou je generovanie matíc časov a vzdialeností medzi všetkými potenciálnymi zákazníkmi. Naopak obohatenie základnej inštancie o dummy zákazníkov, kde už ku generovaniu matíc nedochádza je časovo výrazne lacnejšia operácia.

Pri súhrnnom pohľade na prvú stratégiu je možné povedať, že jej potenciál spočíva v schopnosti obslúžiť malé množstvo dyn. objednávok s vynaložením relatívne malého množstva zdrojov. Jej hlavnou nevýhodou je degradácia pri veľkom množstve alokovaných zdrojov pre klaster, čo môže byť problémom hlavne u nerovnomerného rozdelenia lokalít do klastrov, pri ktorom vznikne niekoľko veľkých klastrov.

Druhá stratégia vyniká vysokým počtom obslužených dyn. objednávok za cenu relatívne nízkej najazdenej vzdialenosti. Navrch má len tretia stratégia pri tesnom približovaní sa k obslúženiu všetkých objednávok. Nevýhodou druhej stratégie je tendencia vtiahnuť do plánu trás veľké množstvo vozidiel aj za cenu toho, že vozidlá budú menej vyťažené, čo je typicky nežiadúce.

Tretia stratégia, rozdelením zdrojov na normo-dummy zákazníkov, prekonáva hlavné obmedzenie prvej stratégie, pričom jej použitím nedochádza k nadmernému alokovaniu času v rámci klastra ako pri druhej stratégii. Oproti druhej stratégii mierne zaostáva v počte obslužených dyn. objednávok vzhľadom na prejdenú vzdialenosť. Na druhej strane však pri obsluhu používa menší počet vozidiel. Na rozdiel od druhej stratégie sa tesne približuje k obslúženiu všetkých dynamic-kých objednávok za nižšiu cenu prejdenej vzdialenosti a použitých vozidiel, čo môže byť výhoda v prípade skutočne vysokej priority pre obsluhu dynamických objednávok.

Podrobnejšie výsledky z experimentu je možné nájsť v príl. B, ktorá obsahuje rozptyly a smerodajné odchylky vybraných veličín, ako i priemerné hodnoty metrík z dát zoskupených podľa DoD, stratégie a miery štedrosti.

6 Záver

Táto diplomová práca sa chopila výzvy prepravnej spoločnosti Wereldo, vniesť väčšiu flexibilitu do plánov trás, a to prechodom od rýdzo statického plánovania k plánom disponovaným k integrácii dynamických objednávok prichádzajúcich počas dňa. Umiestnením dummy zákazníkov, ktorí predstavujú strategicky alokované zdroje pre obsluhu dynamických objednávok, do inštancii statického MAVRP, práca doplnila chýbajúcu funkcionality statického plánovania oproti dynamickému.

Za účelom testovania a vyhodnotenia navrhnutých stratégií bol vytvorený komplexný rámec. Na základe reálnych dát poskytnutých firmou Wereldo a s použitím dátových podkladov z OSM som naimplementoval generátor inštancií statického MAVRP. Funkcionalita generátora je rozšírená o schopnosť generovať scenáre príchodu dynamických objednávok počas dňa. Navrhol som tri stratégie pre generovanie dummy zákazníkov, ktorými boli inštancie obohatené. Pre obohatené inštancie boli spočítané plány trás s použitím algoritmickeho riešiča firmy Wereldo. Ďalej som naimplementoval event-driven systém, ktorý simuluje realizáciu plánu trás, a ktorý slúžil ako prostredie pre testovanie efektívnosti navrhnutých stratégií. Realizácia plánu trás spočíva jednak v aktualizácii statického plánu v čase, jednak v stotožňovaní prišlých dynamických objednávok s dummy zákazníkmi umiestnenými pozdĺž trás. Meranými metrikami plánu trás boli počet použitých vozidiel, najazdená vzdialenosť, pracovný čas a počet obslužených dynamických objednávok. Na základe údajov zaznamenávaných metrik som vyhodnotil stratégie a podal súhrnný pohľad na potenciál i nevýhody použitia každej stratégie. V rámci príloh diplomovej práce som zverejnil generátor inštancií statického MAVRP, ako i reprezentatívne sady ním vygenerovaných inštancií.

Výsledky testov ukázali, že pridanie dummy zákazníkov do statického plánu trás výrazne zvyšuje jeho flexibilitu a disponibilitu prijímať dynamické objednávky. Testy rovnako ukázali náklady spojené s prijímaním dynamických objednávok. Výsledky zo simulácie spolu so štruktúrovaným dizajnom experimentu prinášajú hodnotné poznatky o kvalitách jednotlivých stratégií odskúšaných pre najrôznejšie dy-

namické scenáre, a ponúkajú usmernenia pre implementáciu týchto stratégií v skutočnom prostredí.

6.1 Perspektívy ďalšieho výskumu

Práca ponúka široký priestor pre návaznú akademickú či výskumnú činnosť. Bolo by možné bližšie skúmať vhodnú voľbu pre počet klastrov, do ktorých sú rozdelení potenciálni zákazníci vzhľadom na DoD. Počet klastrov sa síce primárne odvíja od geografického rozpoloženia potenciálnych zákazníkov v oblasti, avšak súvisí aj s množstvom zdrojov, ktoré v nich alokujeme (čo sa prejavilo napr. v degradácií prvej stratégie pri vysokom množstve alokovaných zdrojov v klastri). Vhodným postupom by mohlo byť odraziť sa od Elbow metódy, ako metódy, ktorá nám ponúka množinu vhodných kandidátov. Táto množina by v experimentoch slúžila ako škála pre počet klastrov k .

Pre lepšie porovnanie nákladov spojených s obsluhou objednávok známych dopredu a dynamických objednávok by bolo možné postaviť vedľa seba plán trás s dummy zákazníkmi do ktorého sú postupne integrované dyn. objednávky a plán trás, kde sú všetky objednávky dopredu známe. Takéto porovnanie by si však vyžadovalo výpočet plánu trás pre mnohé scenáre, čo je výpočtovo drahá operácia na čas. Možnou cestou by bola efektívnejšia implementácia algoritmu pre výpočet plánu trás, založená napr. na algoritme SISR.

Bibliografia

1. DANTZIG, George B.; RAMSER, John H. The Truck Dispatching Problem. *Management Science*. 1959, roč. 6, č. 1, s. 80–91.
2. OJEDA RIOS, Brenner Humberto; XAVIER, Eduardo C.; MIY-AZAWA, Flávio K.; AMORIM, Pedro; CURCIO, Eduardo; SANTOS, Maria João. Recent dynamic vehicle routing problems: A survey. *Computers Industrial Engineering*. 2021, roč. 160, s. 107604.
3. UCHOA, E.; PECIN, D.; PESSOA, A.; POGGI, M.; VIDAL, T.; SUBRAMANIAN, A. New Benchmark Instances for the Capacitated Vehicle Routing Problem. *European Journal of Operational Research*. 2017, roč. 257, č. 3, s. 845–858.
4. AL., E. Uchoa et. *CVRPLib: A library for CVRP instances*. 2017. Dostupné tiež z: <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>. [cit. 2024-12-16].
5. SOLOMON, M. M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*. 1987, roč. 35, č. 2, s. 254–265.
6. SOLOMON, M. M. *Solomon Benchmark Instances for VRPTW*. 1987. Dostupné tiež z: <https://www.sintef.no/projectweb/top/vrptw/>. [cit. 2024-12-16].
7. LI, H.; LIM, A. A Metaheuristic for the Pickup and Delivery Problem with Time Windows. *International Journal on Artificial Intelligence Tools*. 2001, roč. 10, č. 4, s. 431–451.
8. LI, H.; LIM, A. *Li & Lim Benchmark Instances for PDPTW*. 2001. Dostupné tiež z: <https://www.sintef.no/projectweb/top/pdptw/>. [cit. 2024-12-16].
9. VIDAL, Thibaut; CRAINIC, Teodor Gabriel; GENDREAU, Michel; LAHRICHI, Nicolas; REI, Walter. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Operations Research*. 2012, roč. 60, č. 3, s. 611–624.
10. PSARAFTIS, Harilaos N. Dynamic vehicle routing problems. In: GOLDEN, Bruce; ASSAD, Arjang (ed.). *Vehicle routing: methods and studies*. Elsevier Science Publishers BV, 1988, s. 223–248.

11. PILLAC, Vincent; GENDREAU, Michel; GUÉRET, Christophe; MEDAGLIA, Andrés. A review of dynamic vehicle routing problems. *European Journal of Operational Research*. 2008, roč. 225, č. 1, s. 11–17.
12. LUND, K.; MADSEN, O. B. G.; RYGAARD, J. M. Vehicle routing problems with varying degrees of dynamism. *IMM Institute of Mathematical Modelling*. 1996. Technical report.
13. PSARAFTIS, Harilaos; WEN, Min; KONTOVAS, Christos. Dynamic Vehicle Routing Problems: Three Decades and Counting. *Networks*. 2015, roč. 67.
14. PISINGER, David; RØPKE, Søren. Large Neighborhood Search. In: GENDREAU, Michel (ed.). *Handbook of Metaheuristics*. 2. vyd. Springer, 2010, s. 399–420.
15. SASSMANN, Vojtěch; RUDOVÁ, Hana; GABONNAY, Michal; SOBOTKA, Václav. Real-World Vehicle Routing Using Adaptive Large Neighborhood Search. In: PÉREZ CÁCERES, Leslie; STÜTZLE, Thomas (ed.). *Evolutionary Computation in Combinatorial Optimization*. Cham: Springer Nature Switzerland, 2023, s. 34–49.
16. LIU, Fei; LU, Chengyu; GUI, Lin; ZHANG, Qingfu; TONG, Xiaoliang; YUAN, Mingxuan. Heuristics for Vehicle Routing Problem: A Survey and Recent Advances. 2023.
17. SOLOMON, Marius M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*. 1987, roč. 35, č. 2, s. 254–265.
18. CLARKE, G.; WRIGHT, J.W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*. 1964, roč. 12, s. 568–581.
19. NIELSEN, Frank. Hierarchical Clustering. In: 2016, s. 195–211.
20. GILLET, B.E.; MILLER, L.R. A heuristic algorithm for the vehicle dispatch problem. *Operations Research*. 1974, roč. 22, s. 340–349.
21. VAN BREEDAM, Alex. Improvement heuristics for the Vehicle Routing Problem based on simulated annealing. *European Journal of Operational Research*. 1995, roč. 86, č. 3, s. 480–490.

22. APPLGATE, D.L.; BIXBY, R.E.; CHVÁTAL, V.; COOK, W.J. *The Traveling Salesman Problem*. The Traveling Salesman Problem. Princeton: Princeton University Press, 2011.
23. LIN, S. Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal*. 1965, roč. 44, s. 2245–2269.
24. OSMAN, I.H. Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem. *Annals of Operations Research*. 1993, roč. 41, s. 421–451.
25. TAILLARD, É.; BADEAU, P.; GENDREAU, M.; GUERTIN, F.; POTVIN, J.Y. A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. *Transportation Science*. 1997, roč. 31, s. 170–186.
26. SØRENSEN, K.; GLOVER, F. Metaheuristics. *Encyclopedia of Operations Research and Management Science*. 2013, roč. 62, s. 960–970.
27. GENDREAU, Michel; POTVIN, Jean-Yves et al. *Handbook of Metaheuristics*. Zv. 2. Springer, 2010.
28. GLOVER, F. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers & Operations Research*. 1986, roč. 13, s. 533–549.
29. ZACHARIADIS, Emmanouil E.; KIRANOUDIS, Chris T. A local search metaheuristic algorithm for the vehicle routing problem with simultaneous pick-ups and deliveries. *Expert Systems with Applications*. 2011, roč. 38, č. 3, s. 2717–2726.
30. HANSEN, Pierre; MLADENović, Nenad. An Introduction to Variable Neighborhood Search. In: *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Ed. VOSS, Stefan; MARTELLO, Silvano; OSMAN, Ibrahim H.; ROUCAIROL, Catherine. Boston, MA: Springer US, 1999, s. 433–458.
31. HANSEN, Pierre; MLADENović, Nenad. Variable Neighborhood Search. In: *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Ed. BURKE, Edmund K.; KENDALL, Graham. Boston, MA: Springer US, 2005, s. 211–238.
32. KIRKPATRICK, S.; GELATT, C.D.; VECCHI, M.P. Optimization by simulated annealing. *Science*. 1983, roč. 220, s. 671–680.

33. SHAW, Paul. Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. In: *CP-98: Fourth International Conference on Principles and Practice of Constraint Programming*. Springer, 1998, zv. 1520, s. 417–431. Lecture Notes in Computer Science.
34. SCHRIMPF, Günther; SCHNEIDER, Jörg; STAMM-WILBRANDT, Hermann; DUECK, Gerhard. Record Breaking Optimization Results Using the Ruin and Recreate Principle. *Journal of Computational Physics*. 2000, roč. 159, č. 2, s. 139–171.
35. HOLLAND, J.H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
36. VIDAL, Thibaut. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Computers Operations Research*. 2022, roč. 140, s. 105643.
37. DORIGO, M.; BIRATTARI, M.; STÜTZLE, T. Ant colony optimization. *IEEE Computational Intelligence Magazine*. 2006, roč. 1, s. 28–39.
38. PISINGER, David; RØPKE, Stefan. A General Heuristic for Vehicle Routing Problems. *Computers and Operations Research*. 2007, roč. 34, č. 8, s. 2403–2435.
39. CHRISTIAENS, Jan; BERGHE, Greet Vanden. Slack Induction by String Removals for Vehicle Routing Problems. *Transportation Science*. 2020, roč. 54, č. 2, s. 417–433.
40. OLIVER, I.; SMITH, D.; HOLLAND, J.R. A Study of Permutation Crossover Operators on the Traveling Salesman Problem. In: GREFFENSTETTE, John (ed.). *Genetic Algorithms and their Applications: Proceedings of the Second International Conference*. 1987, s. 224–230.
41. GLOVER, Fred; HAO, Jin-Kao. The case for strategic oscillation. *Annals of Operations Research*. 2011, roč. 183, č. 1, s. 163–173.
42. RENAUD, J.; BOCTOR, F. F.; OUENNICHE, J. A heuristic for the pickup and delivery traveling salesman problem. *Computers & Operations Research*. 2000, roč. 27, s. 905–916.

43. DIANA, Marco; DESSOUKY, Maged M. A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows. *Transportation Research Part B: Methodological*. 2004, roč. 38, č. 6, s. 539–557.
44. POTVIN, Jean-Yves; ROUSSEAU, Jean-Marc. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research*. 1993, roč. 66, č. 3, s. 331–340.
45. CHEN, Zhi-Long; XU, Hang. Dynamic Column Generation for Dynamic Vehicle Routing with Time Windows. *Transportation Science* [online]. 2006, roč. 40, č. 1, s. 74–88 [cit. 2024-12-10].
46. TAILLARD, É. D.; GAMBARDILLA, L. M.; GENDREAU, M.; POTVIN, J.-Y. Adaptive memory programming: A unified view of metaheuristics. *European Journal of Operational Research*. 2001, roč. 135, č. 1, s. 1–16.
47. POWELL, W. B.; SIMAO, H. P.; BOUZAIENE-AYARI, B. Approximate dynamic programming in transportation and logistics: a unified framework. *EURO Journal on Transportation and Logistics*. 2012, roč. 1, s. 237–284.
48. BENT, Russell W.; HENTENRYCK, Pascal Van. Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers. *Operations Research*. 2004, roč. 52, č. 6, s. 977–987.
49. MITROVIĆ-MINIĆ, Snežana; KRISHNAMURTI, Ramesh; LAPORTE, Gilbert. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*. 2004, roč. 38, č. 8, s. 669–685.
50. MINIC, Snezana; LAPORTE, Gilbert. Waiting strategies for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological*. 2004, roč. 38, s. 635–655.
51. HEMERT, Jano I. van; LA POUTRÉ, J. A. Dynamic Routing Problems with Fruitful Regions: Models and Evolutionary Computation. In: *Parallel Problem Solving from Nature - PPSN VIII*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, s. 692–701.

52. BOCK, Stefan; FERRUCCI, Francesco; GENDREAU, Michel. A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods. *European Journal of Operational Research*. 2013, roč. 225, s. 130–141.
53. FERRUCCI, Francesco; BOCK, Stefan. Pro-active real-time routing in applications with multiple request patterns. *European Journal of Operational Research*. 2016, roč. 253, č. 2, s. 356–371.
54. OPENROUTESERVICE. *OpenRouteService API*. 2024. Dostupné tiež z: <https://www.openrouteservice.org/>. [cit. 2024-12-16].
55. OPENROUTESERVICE. *OpenRouteService API*. 2024. Dostupné tiež z: <https://github.com/GIScience/openrouteservice>. [cit. 2024-12-16].
56. FOUNDATION, OpenStreetMap. *OpenStreetMap*. 2024. Dostupné tiež z: <https://www.openstreetmap.org/>. [cit. 2024-12-16].
57. SINTEF. *Benchmark Instances for the Vehicle Routing Problem (VRP)*. 2024. Dostupné tiež z: <https://www.sintef.no/en/projects/vrp-benchmark-instances/>. [cit. 2024-12-16].
58. VIRTANEN, Pauli; GOMMERS, Ralf; OLIPHANT, Travis E. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*. 2020, roč. 17, s. 261–272.
59. LAB, DAI. *Copulas: Multivariate Modeling with Python*. 2018. Dostupné tiež z: <https://github.com/sdv-dev/Copulas>. [cit. 2024-12-16].
60. DEVELOPERS, Scikit-learn. *Scikit-learn: Machine Learning in Python*. 2020. Dostupné tiež z: <https://scikit-learn.org/stable/>. [cit. 2024-12-16].
61. LEMAIRE, Vincent; COELHO, Arturo; PETIT, Magali et al. *scikit-learn-extra: A set of complementary tools for scikit-learn*. 2020. Dostupné tiež z: <https://github.com/scikit-learn-contrib/scikit-learn-extra>. [cit. 2024-12-16].
62. METACENTRUM. *Metacentrum: National e-Infrastructure for High Performance Computing*. [B.r.]. Dostupné tiež z: <https://metacentrum.cz>. [cit. 2024-12-16].

A Súbory

Súčasťou diplomovej práce sú nasledujúce priložené súbory:

1. **dataset_tuning.zip, dataset_experiment.zip**

Dataset použitý pri nultom experimente (dataset_tuning.zip) a ostrom experimente (dataset_experiment.zip). Každý dataset obsahuje podmnožiny kandidátnych lokalít pre potenciálnych zákazníkov, základné inštancie (bez dummy zákazníkov) a scenáre príchodu dynamických objednávok počas dňa. Množina vozidiel použitá pri experimentoch (ako citlivá informácia) bola nahradená množinou prototypných vozidiel.

2. **generator.zip**

Adresár projektu, ktorý obsahuje zdrojové kódy a iné súbory potrebné pre spustenie naimplementovaného generátora. Adresár tiež obsahuje predgenerované súbory s potenciálnymi zákazníkmi v jednotlivých krajoch ČR.

B Podrobné výsledky z experimentu

Tabuľka B.1: Rozptyly v hodnotách pre daný DoD a stratégiu

DoD	Strat.	Rozptyl v najaz- denej vzdiale- nosti	Rozptyl v počte obslúžených ob- jednávok	Rozptyl v po- čte vozidiel
5	1	48150.99	1.40	1.91
5	2	44318.89	1.22	1.71
5	3	47920.82	1.39	1.86
10	1	125643.90	1.96	1.6
10	2	99059.6	1.57	1.34
10	3	121055.22	1.89	1.55
15	1	26622.29	4.65	1.13
15	2	31251.94	1.70	1.2
15	3	27036.66	2.46	1.23
20	1	41357.90	9.85	1.54
20	2	59386.84	1.97	1.18
20	3	43006.53	3.17	1.08

Tabuľka B.2: Smerodatné odchyľky v hodnotách pre daný DoD a stratégiu

DoD	Strat.	Smerodatná odchyľka v najzdenej vzdialenosti	Smerodatná odchyľka v počte obslužených objednávok	Smerodatná odchyľka v počte vozidiel
5	1	219.40	1.18	1.38
5	2	210.48	1.10	1.31
5	3	218.92	1.18	1.36
10	1	354.46	1.40	1.26
10	2	314.87	1.25	1.16
10	3	347.93	1.38	1.24
15	1	163.17	2.16	1.06
15	2	176.75	1.30	1.10
15	3	164.45	1.57	1.11
20	1	203.36	3.14	1.24
20	2	243.74	1.40	1.09
20	3	207.38	1.78	1.04

Tabuľka B.3: Priemerné hodnoty jednotlivých veličín z dát zoskúpe-
ných podľa stratégie a miery štedrosti (DoD: 5%)

Stratégia	Miera šted- rosti	Počet vozi- diel	Najazdená vzdiale- nosť (km)	Pracovný čas (h)	Obslúžených objednávok	Čas behu (ms)
1	0.1	14.74	2338.06	97.78	2.42	7.88
1	0.2	15.04	2359.98	98.62	2.79	7.86
1	0.3	14.76	2381	98.66	3.48	7.70
1	0.4	15.14	2407.99	99.62	3.90	7.82
1	0.5	15.24	2416.56	100.03	4.19	7.73
1	0.6	15.36	2431.51	100.60	4.53	7.65
1	0.7	15.48	2441.66	101.04	4.71	7.67
1	0.8	15.52	2459.45	101.44	5.21	7.62
1	0.9	15.52	2483.75	102.33	5.50	7.51
1	1.0	15.74	2485.60	102.42	5.61	7.50
1	1.1	15.40	2502.12	102.68	5.83	7.67
1	1.2	15.70	2515.63	103.14	5.94	7.63
1	1.3	15.94	2527.58	104.10	6.20	8.13
1	1.4	15.90	2543.16	104.05	6.41	8.08
1	1.5	16.14	2555.67	104.26	6.49	8.17
1	1.6	16.30	2566.72	105.49	6.66	8.32
1	1.7	16.28	2583.57	106.18	6.83	8.25
2	0.1	15.60	2404.62	99.92	4.15	7.64
2	0.2	15.58	2413.03	100.72	4.58	7.70
2	0.3	15.62	2445.96	100.98	5.08	7.58
2	0.4	15.60	2450.32	101.16	5.27	7.69
2	0.5	15.78	2467.33	102.03	5.48	7.76
2	0.6	15.44	2485.03	101.64	5.78	7.73
2	0.7	15.90	2499.12	102.95	6.03	8.01
2	0.8	16.22	2532.20	104.52	6.29	8.31
2	0.9	16.76	2558.22	105.53	6.58	8.50
2	1.0	16.80	2560.16	106.06	6.68	8.44
2	1.1	16.94	2585	107.29	6.78	8.64
2	1.2	17.38	2596.17	107.71	6.92	8.62
2	1.3	17.48	2615.60	108.30	6.98	9.48
2	1.4	17.72	2639.66	108.76	7.13	9.59
2	1.5	18.14	2647.37	109.42	7.10	9.92
2	1.6	18.24	2675.47	110.56	7.19	10.24
2	1.7	18.70	2685.25	111.17	7.34	10.23
3	0.1	15.18	2409.40	99.38	4.41	7.70
3	0.2	15.58	2432	101.34	4.85	7.55
3	0.3	15.50	2449.08	100.82	5.14	7.59
3	0.4	15.34	2454.16	101.39	5.28	7.63
3	0.5	15.82	2470.34	102.07	5.47	7.48
3	0.6	15.82	2472.73	102.48	5.64	7.64
3	0.7	15.58	2477.46	101.72	5.64	7.66
3	0.8	15.56	2489.90	101.58	5.81	7.67
3	0.9	15.62	2495.29	102.20	5.90	7.55
3	1.0	15.98	2503.36	103.12	5.90	7.53
3	1.1	16.08	2508.31	104	6.02	7.55
3	1.2	15.84	2529.24	103.43	6.14	7.53
3	1.3	15.96	2529.98	103.78	6.30	8.08
3	1.4	16.18	2543.97	104.43	6.43	8.17
3	1.5	16.18	2554.38	105.04	6.50	8.45
3	1.6	15.90	2559.77	104.32	6.58	8.49
3	1.7	16.54	2569.33	105.99	6.68	8.45

Tabuľka B.4: Priemerné hodnoty jednotlivých veličín z dát zoskúpe-
ných podľa stratégie a miery štedrosti (DoD: 10%)

Stratégia	Miera šted- rosti	Počet vozi- diel	Najazdená vzdiale- nosť (km)	Pracovný čas (h)	Obslúžených objednávok	Čas behu (ms)
1	0.1	13.30	2155.73	89.96	3.99	8.18
1	0.2	13.42	2195.80	91.19	5.19	7.92
1	0.3	13.68	2234.22	92.16	6.42	7.78
1	0.4	13.78	2277.54	93.33	7.27	7.78
1	0.5	13.94	2318.49	94.42	8.16	7.79
1	0.6	14.58	2371.80	96.24	9.07	7.65
1	0.7	14.46	2427.12	97.17	9.83	7.42
1	0.8	14.74	2458.14	98.11	10.53	7.68
1	0.9	14.86	2487.86	99.22	11.10	7.73
1	1.0	15.22	2527.24	100.44	11.76	7.71
1	1.1	15.16	2560.04	101.23	12.35	7.67
1	1.2	15.66	2593.11	102.61	12.81	7.71
1	1.3	15.48	2618.77	102.91	13.10	8.46
1	1.4	16.08	2660.63	104.45	13.58	8.42
1	1.5	15.92	2670.99	104.38	13.44	8.45
1	1.6	16.34	2686.93	105.34	13.63	8.44
1	1.7	16.78	2714.38	106.05	13.81	8.48
2	0.1	14.02	2255.17	93.00	7.09	7.73
2	0.2	14.12	2298.06	93.82	8.16	7.69
2	0.3	14.36	2349.05	95.47	9.25	7.99
2	0.4	14.74	2425.01	97.83	10.61	8.29
2	0.5	15.28	2476.88	99.76	11.65	8.55
2	0.6	15.94	2517.43	101.27	12.21	9.05
2	0.7	16.40	2568.10	103.07	12.78	9.30
2	0.8	16.62	2599.18	104.10	13.18	9.52
2	0.9	17.18	2637.78	105.53	13.44	10.00
2	1.0	17.66	2673.00	107.11	13.71	10.14
2	1.1	18.22	2701.29	108.48	13.82	10.48
2	1.2	18.76	2737.29	109.25	13.92	10.76
2	1.3	19.50	2786.56	111.23	14.15	11.88
2	1.4	20.04	2825.97	112.32	14.31	12.26
2	1.5	20.12	2849.92	113.03	14.35	12.50
2	1.6	20.92	2885.79	114.41	14.38	12.97
2	1.7	21.64	2904.39	115.93	14.46	13.22
3	0.1	14.22	2284.86	93.96	7.62	7.85
3	0.2	14.18	2305.10	93.54	8.25	7.63
3	0.3	14.18	2318.51	94.60	8.57	7.78
3	0.4	14.24	2338.86	94.91	8.74	7.73
3	0.5	14.34	2373.91	96.13	9.33	7.72
3	0.6	14.36	2393.63	96.24	9.62	7.71
3	0.7	14.50	2434.66	97.57	10.32	7.84
3	0.8	14.72	2451.02	98.21	10.60	7.88
3	0.9	14.90	2486.46	98.85	11.03	8.00
3	1.0	14.90	2510.39	99.84	11.56	8.03
3	1.1	15.26	2552.72	100.88	12.17	8.06
3	1.2	15.48	2573.70	101.94	12.64	8.18
3	1.3	15.64	2600.65	103.48	13.06	9.18
3	1.4	16.28	2637.33	105.01	13.46	9.27
3	1.5	16.30	2650.38	105.28	13.66	9.35
3	1.6	16.14	2663.76	105.32	13.84	9.41
3	1.7	16.60	2688.40	106.48	14.01	9.61

Tabuľka B.5: Priemerné hodnoty jednotlivých veličín z dát zoskúpe-
ných podľa stratégie a miery štedrosti (DoD: 15%)

Stratégia	Miera šted- rosti	Počet vozi- diel	Najazdená vzdiale- nosť (km)	Pracovný čas (h)	Obslúžených objednávok	Čas behu (ms)
1	0.1	12.86	1994.02	85.54	5.61	8.24
1	0.2	13.28	2066.54	88.15	7.65	8.04
1	0.3	13.72	2137.76	89.87	8.92	7.99
1	0.4	14.04	2207.1	91.58	10.59	7.74
1	0.5	13.98	2268.19	93.31	11.92	7.90
1	0.6	14.30	2350.74	95.03	13.57	7.74
1	0.7	14.48	2422.37	96.87	14.95	7.76
1	0.8	14.94	2494.03	98.76	16.44	7.77
1	0.9	15.52	2562.2	101.21	17.97	7.84
1	1.0	15.88	2600.74	102.96	18.36	7.61
1	1.1	15.98	2643.31	104.03	18.52	7.63
1	1.2	16.36	2679.74	105.11	19.55	7.73
1	1.3	16.84	2702.52	106.3	20.27	8.68
1	1.4	16.56	2764.17	106.74	20.03	8.59
1	1.5	16.84	2783.63	107.64	20.56	8.61
1	1.6	16.76	2770.8	106.87	19.95	8.52
1	1.7	16.58	2734.77	106.3	18.97	8.41
2	0.1	13.38	2102.37	88.52	9.09	7.82
2	0.2	14.02	2227.34	92.17	11.59	8.24
2	0.3	14.56	2357.93	95.67	14.6	8.68
2	0.4	15.28	2479.41	99.25	17.43	9.14
2	0.5	15.94	2563.01	102.32	19.03	9.56
2	0.6	16.86	2624.01	105.34	20.13	10.16
2	0.7	17.66	2667.69	107.5	20.76	10.51
2	0.8	18.14	2724.8	109.76	21.1	11.19
2	0.9	18.98	2756.49	110.48	21.58	11.51
2	1.0	19.68	2822.55	112.53	21.9	11.82
2	1.1	20.68	2863.12	115.25	22.05	12.29
2	1.2	21.60	2935.75	117.47	22.31	12.82
2	1.3	22.58	2978.45	120.24	22.37	14.55
2	1.4	23.32	3020.46	121.81	22.5	15.08
2	1.5	24.40	3072.61	124.11	22.58	15.60
2	1.6	25.10	3121.67	125.9	22.62	16.20
2	1.7	26.10	3184.5	128.66	22.68	16.75
3	0.1	13.68	2140.48	90.23	9.63	7.83
3	0.2	13.68	2167.06	90.42	10.29	7.86
3	0.3	13.78	2195.49	91.45	10.62	7.75
3	0.4	14.08	2235.61	92.53	11.28	7.81
3	0.5	14.02	2278.69	93.24	12.44	8.07
3	0.6	14.42	2351.42	95.61	13.92	8.11
3	0.7	14.48	2411.41	97	15.22	8.23
3	0.8	14.98	2469.34	99.56	16.46	8.15
3	0.9	15.28	2513.89	101.38	17.59	8.61
3	1.0	15.78	2582.08	103.28	19.16	8.71
3	1.1	16.14	2610.04	104.44	19.95	8.86
3	1.2	16.80	2643.02	106.91	20.59	9.00
3	1.3	16.60	2675.13	107.03	21.19	10.18
3	1.4	17.10	2687.76	108.3	21.54	10.25
3	1.5	17.40	2723.45	109.35	21.79	10.39
3	1.6	17.74	2735.82	110.03	21.95	10.62
3	1.7	18.26	2753.84	111.48	22.24	10.78

Tabuľka B.6: Priemerné hodnoty jednotlivých veličín z dát zoskúpe-
ných podľa stratégie a miery štedrosti (DoD: 20%)

Stratégia	Miera šted- rosti	Počet vozi- diel	Najazdená vzdiale- nosť (km)	Pracovný čas (h)	Obslúžených objednávok	Čas behu (ms)
1	0.1	13.48	2222.55	88.98	7.39	8.29
1	0.2	13.88	2309.3	91.03	9.45	7.92
1	0.3	14.14	2404.63	93.9	11.44	7.57
1	0.4	14.30	2497.83	95.9	13.19	7.63
1	0.5	14.88	2599.12	98.98	15.39	7.68
1	0.6	15.02	2686.24	100.82	17.66	7.76
1	0.7	15.36	2779.77	104.03	19.72	7.74
1	0.8	16.06	2872.19	106.62	22.09	7.97
1	0.9	16.78	2972.35	110.12	24.4	7.89
1	1.0	17.22	3045	111.11	26.07	7.96
1	1.1	17.40	3087.7	111.49	25.44	7.94
1	1.2	17.70	3100.71	111.87	26.15	8.00
1	1.3	17.98	3121.93	112.8	25.50	8.67
1	1.4	17.68	3106.84	111.24	24.13	8.54
1	1.5	17.42	3045.55	109.92	22.72	8.43
1	1.6	16.52	2964.78	106.78	20.01	8.20
1	1.7	16.56	2955.16	106.62	19.07	8.12
2	0.1	13.80	2330.43	91.64	10.6	7.78
2	0.2	14.68	2515.98	96.75	15.55	8.52
2	0.3	15.36	2678.16	101.32	19.82	9.16
2	0.4	16.38	2830.17	106.36	23.44	9.86
2	0.5	17.46	2929.58	109.98	25.81	10.48
2	0.6	18.36	3010.62	113.22	27.04	11.36
2	0.7	19.56	3070.57	115.81	27.84	11.85
2	0.8	20.38	3133.35	117.91	28.32	12.38
2	0.9	21.70	3188.31	120.31	28.67	13.04
2	1.0	22.86	3243.39	124.14	29.01	13.62
2	1.1	23.74	3318.89	126.08	29.25	14.42
2	1.2	24.94	3385.25	128.49	29.43	15.14
2	1.3	26.28	3450.36	131.73	29.52	17.41
2	1.4	27.44	3513.4	134.32	29.6	18.09
2	1.5	28.66	3541.1	136.99	29.67	19.01
2	1.6	29.84	3620.63	139.58	29.77	19.90
2	1.7	31.12	3692.03	141.82	29.81	21.08
3	0.1	14.18	2357.55	92.49	10.88	7.73
3	0.2	14.26	2393.23	93.63	11.49	7.69
3	0.3	13.92	2427.16	93.72	11.99	7.58
3	0.4	14.36	2515.02	96.76	13.93	7.87
3	0.5	14.68	2571.17	98.25	15.27	8.09
3	0.6	15.16	2698.28	101.44	18.24	8.21
3	0.7	15.44	2784.56	104.8	20.82	8.52
3	0.8	15.70	2828.24	105.77	22.01	8.94
3	0.9	16.12	2910	107.93	23.93	9.07
3	1.0	16.58	3004.07	111.21	26.04	9.29
3	1.1	17.12	3046.97	112.61	27.06	9.55
3	1.2	17.62	3078.55	114.3	28.08	9.79
3	1.3	18.12	3121.73	115.23	28.57	11.03
3	1.4	18.36	3127.33	116.39	28.82	11.24
3	1.5	18.88	3143.79	116.97	29.03	11.46
3	1.6	19.46	3173.38	119.32	29.19	11.71
3	1.7	19.92	3195.08	119.69	29.33	11.91