



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**SYSTÉM PRE PODPORU OPTIMALIZÁCIE SIETE
MESTSKEJ HROMADNEJ DOPRAVY**

SYSTEM FOR SUPPORTING THE OPTIMIZATION OF URBAN PUBLIC TRANSPORT NETWORK

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ KATONA

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. FRANTIŠEK ZBOŘIL, Ph.D.

BRNO 2025

Zadání bakalářské práce



163440

Ústav: Ústav inteligentních systémů (UITS)
Student: **Katona Lukáš**
Program: Informační technologie
Název: **Systém pro podporu optimalizace sítě městské hromadné dopravy**
Kategorie: Umělá inteligence
Akademický rok: 2024/25

Zadání:

1. Prostudujte problematiku vytváření sítě městské hromadné dopravy, její modelování a její optimalizace pro města do půl milionu obyvatel.
2. Pro město velikosti krajského města České republiky získejte nebo odhadněte data, ze kterých sestavte model zdejší městské hromadné dopravy.
3. Seznamte se s řešeními, které pro optimalizaci takové sítě používají metody umělé inteligence.
4. Vytvořte prostředí, které bude sloužit k modelování a optimalizaci takové sítě.
5. Na vhodně zvolených příkladech ověřte fungování vašeho systému a diskutujte dosažené výsledky.

Literatura:

- Kate Han, Lee A. Christie, Alexandru-Ciprian Zăvoianu, and John McCall. 2021. Optimising the introduction of connected and autonomous vehicles in a public transport system using macro-level mobility simulations and evolutionary algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '21). Association for Computing Machinery, New York, NY, USA, 315–316.
- Yentl Van Tendeloo and Hans Vangheluwe. 2018. Discrete event system specification modeling and simulation. In Proceedings of the 2018 Winter Simulation Conference (WSC '18). IEEE Press, 162–176.
-

Při obhajobě semestrální části projektu je požadováno:
První dva body zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Zbořil František, doc. Ing., Ph.D.**
Vedoucí ústavu: Kočí Radek, Ing., Ph.D.
Datum zadání: 1.11.2024
Termín pro odevzdání: 14.5.2025
Datum schválení: 31.10.2024

Abstrakt

Cielom tejto práce je optimalizácia mestskej hromadnej dopravy, konkrétne časového rozpisu jednej linky. Zmyslom tohto textu je popis celého riešenia daného problému, od analýzy až po výsledný systém. Informácie získané priamo z Dopravného podniku mesta Brno boli využité na zostrojenie simulačného modelu jednej linky mestskej hromadnej dopravy. Pomocou tohto modelu a genetického algoritmu sa systém pokúsi nájsť najoptimálnejší časový rozpis danej linky. Súčasťou práce je aj nástroj s používateľským rozhraním, ktorý je možné v praxi použiť na analýzu a optimalizáciu terajších časových rozpisov jednotlivých liniek. Nástroj poskytuje širokú škálu rôznych vstupov, ktorými môže analytik obmedziť isté vlastnosti výsledného rozpisu. Vďaka tomuto nástroju by sa mala práca analytikov, ktorí majú na starosť správu a vytváranie liniek, výrazne zjednodušiť a zvýšiť tak efektivitu mestskej hromadnej dopravy.

Abstract

The aim of this work is the optimization of urban public transport, specifically the timetable of one line. The purpose of this text is to describe the entire solution of the given problem, from analysis to the resulting system. Information obtained directly from the Brno Public Transport Company was used to construct a simulation model of one urban public transport line. Using this model and a genetic algorithm, the system will attempt to find the most optimal timetable for the given line. Part of this work is a tool with a user interface that can be used in practice to analyze and optimize the current timetables of individual lines. The tool provides a wide range of different inputs that an analyst can use to restrict certain properties of the resulting timetable. Thanks to this tool, the work of analysts who are responsible for managing and creating lines should be significantly simplified and thus increase the efficiency of urban public transport.

Klíčové slová

optimalizácia, mestská hromadná doprava, algoritmy inšpirované prírodou, genetický algoritmus, simulácia, DEVS

Keywords

optimization, urban public transport, nature-inspired algorithms, genetic algorithm, simulation, DEVS

Citácia

KATONA, Lukáš. *Systém pre podporu optimalizácie siete mestskej hromadnej dopravy*. Brno, 2025. Bakalárska práca. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce doc. Ing. František Zbořil, Ph.D.

Systém pre podporu optimalizácie siete mestskej hromadnej dopravy

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostante pod vedením pána doc. Ing. Františka Zbořila Ph.D. Ďalšie informácie mi poskytol pán Dr. Ing. Petr Peringer z jeho prezentácií predmetov IMS a SNT a pán Michael Kříž z Dopravného podniku mesta Brno. Uviedol som všetky literárne zdroje, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....

Lukáš Katona
23. apríla 2025

Podakovanie

Rád by som poďakoval pánovi Zbořilovi za jeho cenné rady a konzultácie, hlavne zo strany optimalizácie a algoritmov z odvetvia umelej inteligencie. Pánovi Křížovi by som rád poďakoval za jeho čas a ochotu poskytnúť informácie o fungovaní MHD v Brne a zároveň za jeho pripomienky z pohľadu možného používateľa systému. Nakoniec by som chcel poďakovať môjmu priateľovi Danovi Valníčkovi a jeho starému otcovi za to, že mi vybavili kontakt na pána Kříže a tým umožnili získať potrebné informácie o riešenom probléme priamo z prvej ruky.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 4 |
| 2 | Relevantné vlastnosti mestskej hromadnej dopravy | 6 |
| 2.1 | Konzultácia s analytikom Dopravného podniku | 6 |
| 2.2 | Získané informácie | 6 |
| 3 | Simulačný model linky mestskej hromadnej dopravy | 8 |
| 3.1 | Popis simulačného modelu | 8 |
| 3.2 | Implementácia simulačného modelu | 12 |
| 3.3 | Výstupy simulačného modelu | 13 |
| 4 | Optimalizácia časového rozpisu linky mestskej hromadnej dopravy | 18 |
| 4.1 | Vstupy potrebné od používateľa | 18 |
| 4.2 | Základná implementácia genetického algoritmu | 19 |
| 4.3 | Implementácia multi-objektívneho genetického algoritmu | 22 |
| 4.4 | Výstupy optimalizácie | 28 |
| 5 | Experimenty | 29 |
| 5.1 | Experiment s krátkou trasou a krátkym vozidlom | 30 |
| 5.2 | Experiment s krátkou trasou a dlhým vozidlom | 31 |
| 5.3 | Experiment s dlhou trasou a krátkym vozidlom | 32 |
| 5.4 | Experiment s dlhou trasou a dlhým vozidlom | 33 |
| 6 | Nástroj na analýzu a optimalizáciu časových rozpisov | 34 |
| 6.1 | Návrh užívateľského rozhrania | 34 |
| 6.2 | Implementácia nástroja | 38 |
| 7 | Záver | 39 |
| | Literatúra | 40 |
| A | Programová dokumentácia | 41 |
| B | Experiment s krátkou trasou a krátkym vozidlom | 42 |
| C | Experiment s krátkou trasou a dlhým vozidlom | 44 |
| D | Experiment s dlhou trasou a krátkym vozidlom | 46 |
| E | Experiment s dlhou trasou a dlhým vozidlom | 48 |

| | | |
|----------|--------------------------------------|-----------|
| F | Mail od pána Michala Hlaváčka | 50 |
| G | Finančný model DPMB | 54 |

Zoznam obrázkov

| | | |
|-----|---|----|
| 3.1 | Schéma simulačného modelu | 8 |
| 3.2 | Stavový automat modelu zastávky | 10 |
| 3.3 | Stavový automat modelu vozidla | 11 |
| 3.4 | Počet cestujúcich, ktorí prišli na zastávku v jednotlivých hodinách | 15 |
| 3.5 | Počet cestujúcich, ktorí vystúpili na zastávke v jednotlivých hodinách | 15 |
| 3.6 | Počet cestujúcich, ktorí museli čakať na ďalší spoj v jednotlivých hodinách | 16 |
| 3.7 | Čas, ktorý cestujúci strávili čakaním na zastávke v jednotlivých hodinách | 16 |
| 3.8 | Priemerná naplnenosť vozidla na jednotlivých zastávkach | 17 |
| 4.1 | Príklad Pareto frontu | 21 |
| 4.2 | Populácia na začiatku a na konci optimalizácie | 28 |
| 6.1 | Wireframe pre analýzu časového rozpisu | 35 |
| 6.2 | Wireframe pre optimalizáciu časového rozpisu | 37 |
| B.1 | Počet cestujúcich prichádzajúcich na zastávku za hodinu | 42 |
| B.2 | Priemerný čas strávený čakaním za hodinu | 42 |
| C.1 | Počet cestujúcich prichádzajúcich na zastávku za hodinu | 44 |
| C.2 | Priemerný čas strávený čakaním za hodinu | 44 |
| D.1 | Počet cestujúcich prichádzajúcich na zastávku za hodinu | 46 |
| D.2 | Priemerný čas strávený čakaním za hodinu | 46 |
| E.1 | Počet cestujúcich prichádzajúcich na zastávku za hodinu | 48 |
| E.2 | Priemerný čas strávený čakaním za hodinu | 48 |

Kapitola 1

Úvod

Mestskú hromadnú opravu využíva takmer každý človek a preto sa oplatí pracovať na jej efektívnosti. Zmyslom tejto práce je nazrieť do problematiky vytvárania a správy jednotlivých liniek mestskej hromadnej dopravy, ich analýze a optimalizácie. Ďalej vytvoriť nástroj, ktorý pomôže analytikom pri vytváraní nových liniek alebo optimalizácii existujúcich. Tento nástroj popísať a otestovať na reálnych dátach. Výsledkom celej práce by teda malo byť teoretické zefektívnenie mestskej hromadnej dopravy.

Ludia odnepamäti nachádzajú inspiráciu pre svoje vynálezy v prírode a tomuto trendu neunikla ani informatika, ktorá má od prírody zo všetkých oborov asi najďalej. Algoritmy inšpirované prírodou prinášajú riešenia na problémy, ktoré by sme inými spôsobmi vedeli vyriešiť len za nereálne dlhý čas. Táto časť informatiky ma veľmi zaujala a preto som sa rozhodol nájsť ďalšie využitie pre jeden z týchto algoritmov. Tieto algoritmy sa používajú pre komplexné problémy s veľkým množstvom navzájom sa ovplyvňujúcich premenných. Jedným z takýchto problémov môžu byť optimalizácie rôznych systémov.

Mestská hromadná doprava je systém, ktorý je pre mnoho ľudí veľmi dôležitý, preto sa oplatí investovať čas a námahu do jeho optimalizácie. Keďže aj mne je problém s dlhým čakaním na zastávke alebo preplneným autobusom blízko, rozhodol som sa venovať sa práve jemu.

Cieľom tejto práce je vytvoriť nástroj, ktorý za pomoci genetického algoritmu a používateľských obmedzení nájde čo najoptimálnejší časový rozpis jednej linky mestskej hromadnej dopravy. Zároveň sa tento nástroj bude dať použiť na analýzu už existujúcich rozpisov a ich porovnanie s novými riešeniami. Nástroj bude mať užívateľské rozhranie pre rýchlu manipuláciu so vstupnými dátami a prehľadné zobrazenie výsledkov.

Práca je rozdelená na 8 kapitol vrátane úvodu a záveru, z toho najdôležitejšie sú kapitoly 2 až ??.

Po tomto úvode nasleduje kapitola ??, v ktorej sú vysvetlené základné pojmy potrebné na porozumenie celej práce. Optimalizácia jednotlivých rozpisov pracuje so simulačným modelom, preto je potrebné poznať základy simulácií. V zhladom sa systém mestskej hromadnej dopravy, konkrétne príchody vozidiel na zastávky bola zvolená diskretná simulácia riadená udalosťami. Ďalej je v tejto kapitole vysvetlený genetický algoritmus, jeden z algoritmov inšpirovaných prírodou, ktorý bol zvolený na riešenie problému optimalizácie.

Na zostrojenie simulačného modelu treba najprv vyhodnotiť, ktoré vlastnosti reálneho systému sú pre nás relevantné. V kapitole 2 je popísaná konzultácia s analytikom Dopravného podniku mesta Brno, ktorý poskytol informácie z reálneho života. Popísal postupy a metódy, ktoré sa používajú pre optimalizáciu liniek teraz a vytýčil vlastnosti na ktoré je potrebné sa zamerať.

Zo získaných informácií bol vytvorený simulačný model, ktorého popis, stručné zhrnutie implementácie a jednotlivé experimenty s ním sú popísané v kapitole 3. Výstupy simulačného modelu sú základom pre optimalizáciu časového rozpisu linky. Ide o sériu grafov, na ktorých môže používateľ vidieť vyťaženosť linky v priebehu dňa a naprieč zastávkami. Tieto informácie poslúžili k validácii simulačného modelu a na ohodnotenie jednotlivých rozpisov.

V kapitole 4 je popísaný proces optimalizácie časového rozpisu linky mestskej hromadnej dopravy. Jednotlivé rozpisy ohodnotené na základe informácií získaných zo simulácie sa použijú ako jedinci v genetickom algoritme. Genetický algoritmus bude tieto rozpisy spájať a mutovať, aby našiel čo najlepšie riešenie. Výsledky optimalizácie budú porovnané s pôvodnými rozpismi a ohodnotené.

Samotný program by v praxi nebol moc použiteľný bez používateľského rozhrania. V kapitole ?? je popísaný návrh a implementácia užívateľského rozhrania analytického nástroja. Tento nástroj umožňuje používateľovi jednoducho zadať vstupné dáta, spustiť optimalizáciu a zobrazíť výsledky. Jednou z najdôležitejších častí tohto rozhrania je zadávanie používateľských obmedzení, ktoré musí výsledný časový rozpis splňovať.

Predposlednou kapitolou je kapitola 5, ktorá sa zaoberá testovaním optimalizácie a simulačného modelu. Na konci tejto kapitoly sa nachádza spätná väzba od analytika, ktorý poskytol informácie o reálnom systéme, a to z pohľadu možného budúceho používateľa tohto nástroja.

V závere práce 7 je uvedené zhrnutie výsledkov a možné budúce rozšírenia a vylepšenia nástroja.

Kapitola 2

Relevantné vlastnosti mestskej hromadnej dopravy

Optimalizácia mestskej hromadnej dopravy je veľmi široký pojem a tento úkon sa dá pre viesť v mnoho rôznych smeroch. Preto je dôležité vybrať si tú časť systému, na ktorej sa oplatí pracovať. Pre toto rozhodnutie sú potrebné informácie o reálnom systéme, doterajších postupoch, spôsoboch analýzy a optimalizácie.

2.1 Konzultácia s analytikom Dopravného podniku

Pre získanie potrebných informácií som sa obrátil na pána Michaela Kříže, analytika Dopravného podniku mesta Brno. Stretnutie s pánom Křížom prebehlo 7. 10. 2024 v priestoroch dopravného podniku. Vďaka získaným informáciám som sa rozhodol zamerať na analýzu a optimalizáciu časového rozpisu jednej linky mestskej hromadnej dopravy. Optimalizovať viac ako jednu linku naraz by pravdepodobne prinieslo lepšie výsledky, ale to len teoreticky. V praxi by sa to z finančných a časových dôvodov neoplatilo. Zhodli sme sa na tom, že najlepšie bude optimalizovať jednu linku a to iba jedným smerom, pretože druhý smer linky sa dá považovať za samostatnú linku, minimálne v rámci nárokov cestujúcich.

2.2 Získané informácie

Informácie, ktoré sú dôležité pre analýzu a optimalizáciu časového rozpisu linky mestskej hromadnej dopravy sú:

- **Počet zastávok** — potrebné pre zostavenie samotného rozpisu aj simulačného modelu, nie je dôležitý len počet, ale aj informácie o jednotlivých zastávkach.
- **Čas potrebný na prejdienie od jednej zastávky k druhej** — tiež je to súčasťou rozpisu, zároveň sa touto informáciou riadi plánovanie udalostí v simulačnom modeli, viac v kapitole 3.
- **Počet odchodov vozidla z jeho prvej zastávky za celý deň** — týmto sa myslí celkový počet vozidiel, respektíve ciest ktoré vozidlá danej linky za deň absolvujú, tento parameter je dôležitý pre optimalizáciu, pre dopravný podnik je výhodné mať čo najmenší počet vozidiel, ktoré sú v prevádzke, šetrí to palivo, údržbu aj ľudské zdroje.

- **Počet odchodov vozidla z jeho prvej zastávky za hodinu** — myslí sa tým jeden riadok v časovom rozpise, mal by byť priamo úmerný priemernému počtu cestujúcich čakajúcich na danú linku v danej hodine.
- **Počet cestujúcich čakajúcich na jednotlivých zastávkach** — najdôležitejší parameter pre optimalizáciu, časový rozpis musí dostatočne uspokojiť potreby všetkých cestujúcich, zároveň je ťažké získať presné hodnoty. Pri reálnom použití by sa mala spraviť štúdia ktorá by tieto hodnoty získala. Na návrh pána Kříže som na testovacie účely použil hodnoty, ktoré sú úmerné počtu odchádzajúcich vozidiel v hodine a kapacite vozidiel, povedal, že linky v Brne sú už celkom dobre naplánované a preto by mali byť tieto hodnoty približne správne. Jedná sa o nemenný vstup optimalizácie, ktorému sa má časový rozpis prispôbiť, preto je teoreticky jedno aké hodnoty budú do systému zadané.
- **Doba čakania na zastávke** — druhý z parametrov podstatných pre optimalizáciu, systém by mal zostrojiť taký rozpis, kedy je celková doba čakania všetkých cestujúcich čo najmenšia.
- **Počet cestujúcich vo vozidle** — tretí dôležitý parameter pre optimalizáciu, vysoká preplnenosť vozidla má za dôsledok nekomfortné cestovanie, na druhej strane, príliš prázdne vozidlo je neefektívne využité zo strany dopravného podniku.
- **Kapacita vozidla** — každé vozidlo prenesie naraz len isté množstvo cestujúcich, zároveň sa od preplnenosti vozidla odvíja komfort cestujúcich.
- **Počet cestujúcich, ktorí nenásúpia kvôli preplnenosti vozidla** — v extrémnych prípadoch môže nastať, že bude vozidlo úplne plné a zvyšní cestujúci musia na zastávke čakať do príchodu ďalšieho vozidla, rozpisy pri ktorých sa toto bude diať budú veľmi negatívne ohodnocované aby sa táto zlá vlastnosť čo najviac eliminova, viac v kapitole o optimalizácii 4.
- **Počet vystupujúcich cestujúcich na jednotlivých zastávkach** — tento parameter je potrebný pre simuláciu, keďže do vozidla s istou kapacitou priebežne nastupuje veľa cestujúcich, musia z neho aj vystupovať, získanie presných hodnôt je ešte ťažšie ako pri predošlých parametroch, preto som sa rozhodol každej zastávke prideliť koeficient dôležitosti, ktorý bude určovať aký podiel cestujúcich vystúpi na danej zastávke. Na prvej zastávke je tento koeficient 0, keďže na nej cestujúci iba nastupujú, na poslednej zastávke je koeficient 1, keďže na nej cestujúci iba vystupujú a na zastávkach medzi nimi sa koeficient mení podľa toho aký podiel cestujúcich na danej zastávke priemerne vystupuje, prestupné zastávky budú mať vyšší koeficient ako tie, ktoré cestujúcich len zbierajú.

Po vytvorení simulačného modelu a implementácii optimalizácie som za pánom Křížom prišiel druhýkrát, ukázať mu výsledky, získať spätnú väzbu a zistiť jeho požiadavky na používateľské rozhranie tohto nástroja.

TODO navštíviť ho ešte raz a ukázať mu návrh užívateľského rozhrania.

Kapitola 3

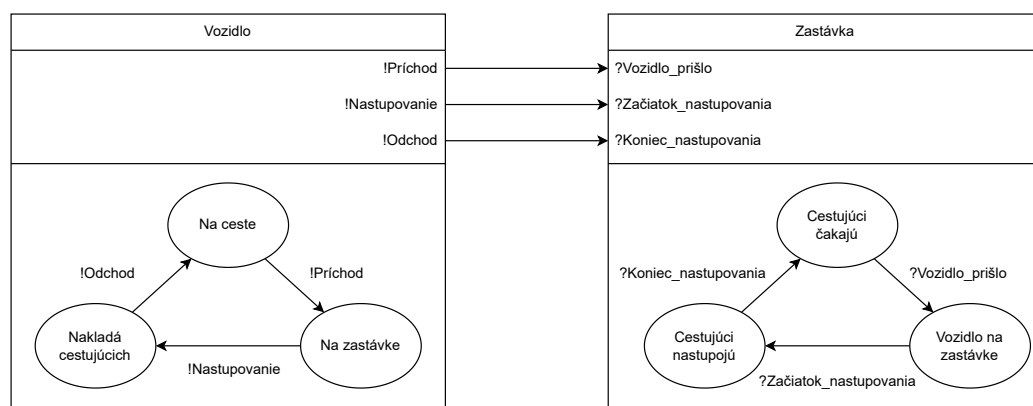
Simulačný model linky mestskej hromadnej dopravy

Druhým hlavným krokom po získaní potrebných informácií je zostrojenie simulačného modelu. Problém optimalizácie časového rozpisu linky mestskej hromadnej dopravy je veľmi komplexný a vytvoriť program ktorý by ho riešil analyticky by bolo veľmi náročné. Na modelovanie bol použitý formalizmus DEVS (Discrete Event Simulation System), po slovensky diskretná simulácia riadená udalosťami. Ide o typ simulácie, kde sa modelový čas posúva skokovo a to od jednej udalosti k druhej. Viac o tomto formalizme je napísané v kapitole ??.

3.1 Popis simulačného modelu

Na začiatok je potrebné definovať základné prvky modelu (viď obrázok 3.1), ich stavy a vzťahy medzi nimi. Simulačný model je zložený z troch hlavných častí, tými sú:

- **Kalendár udalostí** — dôležitý pre plánovanie udalostí v modeli
- **Model zastávky** — dôležitý pre príchod a odchod cestujúcich
- **Model vozidla** — dôležitý pre prepravu cestujúcich medzi zastávkami



Obr. 3.1: Schéma simulačného modelu

V simulácii sa budú nachádzať viaceré inštancie modelu zastávky, každá reprezentujúca jednu zastávku na linke. Vozidiel bude taktiež viacero, neberieme do úvahy fyzické vozidlá a fakt, že v reálnom živote sa tieto vozidlá na linke v priebehu dňa niekoľkokrát otočia. Dôležitá je trasa, ktorú prejde, preto jedna inštancia modelu vozidla reprezentuje jednu cestu, ktorú vozidlo prejde. Cestujúcich v modeli nie je potrebné reprezentovať ako samostatný model, sú reprezentovaní len jedným číslom, časom kedy prišli na zastávku. Sú súčasťou modelu zastávky, konkrétne tej na ktorej čakajú.

Kalendár udalostí

Kalendár udalostí je základným prvkom modelu, ktorý riadi priebeh simulácie. V kalendári sú uložené všetky udalosti, ktoré sa majú v modeli stať. Tieto udalosti sú v prípade tohto modelu príchody vozidiel na zastávky. V rámci jednej udalosti sa vykoná rutina obslúženia cestujúcich čakajúcich na zastávke. Všetky udalosti sú naplánované na začiatku simulácie, pretože vďaka časovému rozpisu presne vieme všetky príchody vozidiel na jednotlivé zastávky. Príchody na prvú zastávku sú dané hlavným časovým rozpisom, príchody na ostatné zastávky sú odvodené od času potrebného na prejde z jednej zastávky na druhú. Pseudokód plánovania udalostí je uvedený v algoritme 1. V priebehu simulácie už netreba žiadne ďalšie udalosti plánovať.

Algoritmus 1: Plánovanie udalostí

```
for čas v časovom rozpise počiatkovej zastávky do
    Vytvorí sa vozidlo;
    for zastávka na linke do
        Pridá sa udalosť príchodu vozidla na danú zastávku;
```

Po naplánovaní všetkých udalostí sa začne simulácia. Postupne sa vykonávajú jednotlivé udalosti a simulácia sa diskretné posúva vpred, až kým nie je kalendár udalostí prázdny alebo nie je dosiahnutý koncový čas simulácie. Pseudokód riadenia simulácie je popísaný v algoritme 2.

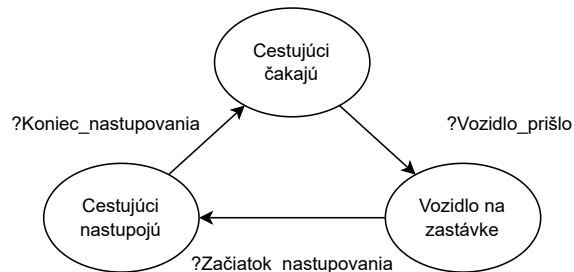
Algoritmus 2: Simulácia riadená udalosťami

```
while kalendár udalostí nie je prázdny do
    Vyberie sa udalosť s najmenším časom v kalendári;
    Udalosť sa odstráni z kalendára;
    if čas udalosti > koncový čas simulácie then
        Simulácia končí;
    Modelový čas sa nastaví na čas udalosti;
    Udalosť sa vykoná;
```

Pre potreby časového rozpisu stačí simulovať jeden deň, v prípade rôznych rozpisov v čase pracovných dní a v čase sviatkov a víkendov ide o kompletne iný rozpis, ktorý sa musí simulovať a optimalizovať samostatne. Všetky naplánované udalosti by mali byť v rámci jedného dňa, preto by simulácia mala končiť vždy vyprázdnením kalendára udalostí a nie prekročením koncového času.

Model zastávky

Model zastávky slúži na reprezentáciu jednotlivých zastávok na linke. Každá reálna zastávka je v systéme ako samostatná inštancia modelu zastávky. Je pasívna, to znamená, že jej vnútorný stav sa plne odvíja od prichádzajúcich vonkajších signálov.



Obr. 3.2: Stavový automat modelu zastávky

Ako je vidieť na obrázku 3.2, model zastávky má tri stavy:

- **Cestujúci čakajú** — na zastávke nie je žiadne vozidlo
- **Vozidlo na zastávke** — na zastávku práve prišlo vozidlo
- **Cestujúci nastupujú** — cestujúci nastupujú do vozidla

Dôležitejšie sú však vstupné signály, ktorými je celý stavový automat riadený. Zároveň sa pri prechodoch medzi stavmi vykonávajú akcie pre obsluhu cestujúcich.

Vstupný signál **?Vozidlo_prišlo** mení vnútorný stav zastávky z **Cestujúci čakajú** na **Vozidlo na zastávke**. Taktiež sa vypočíta dĺžka intervalu medzi posledným príchodom vozidla a terajším príchodom. Tento údaj slúži pre vygenerovanie prichádzajúcich cestujúcich.

Vstupný signál **?Začiatok_nastupovania** mení vnútorný stav zastávky z **Vozidlo na zastávke** na **Cestujúci nastupujú**. Pri tomto prechode sa vygenerujú časy všetkých cestujúcich, ktorí prišli od posledného príchodu vozidla.

Vstupný signál **?Koniec_nastupovania** mení vnútorný stav zastávky z **Cestujúci nastupujú** na **Cestujúci čakajú**. Prepíše sa čas posledného príchodu vozidla.

Generovanie cestujúcich

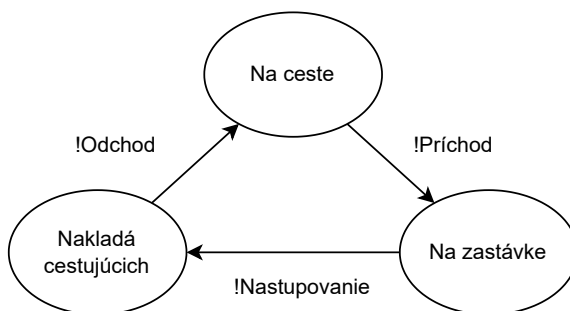
Cestujúci sa generujú na základe exponenciálneho rozdelenia s parametrom λ . Tento parameter je v rámci jednej zastávky každú hodinu iný, pretože počet prichádzajúcich cestujúcich sa v priebehu dňa mení. V algoritme 3 je popísaný proces generovania cestujúcich na zastávke.

Algoritmus 3: Generovanie cestujúcich

```
získanie  $\lambda$  pre danú zastáku a hodinu;  
while modelový čas < čas príchodu autobusu do  
    vygenerovanie času príchodu cestujúceho;  
    pridanie cestujúceho do fronty;  
    zmena modelového času na čas príchodu cestujúceho;
```

Model vozidla

Každé vozidlo, respektíve jedna trasa ktorú vozidlo podľa časového rozpisu prejde, je reprezentovaná jednou inštanciou modelu vozidla. Model vozidla je na rozdiel od modelu zastávky aktívnym prvkom, nemá žiadne vstupné signály, iba výstupné. Tieto výstupné signály sú prepojené so vstupnými signálmi modelu zastávky ako je vidieť na obrázku 3.1.



Obr. 3.3: Stavový automat modelu vozidla

Na obrázku 3.3 je vidieť stavový automat modelu vozidla. Model vozidla má tri stavy:

- **Na ceste** — vozidlo sa pohybuje medzi zastávkami
- **Na zastávke** — vozidlo práve prišlo na zastávku
- **Nakladá cestujúcich** — vozidlo nakladá cestujúcich

Výstupný signál **!Príchod** mení vnútorný stav vozidla z **Na ceste** na **Na zastávke**. Pri tomto prechode vystúpia cestujúci podľa koeficientu zastávky na ktorú vozidlo práve prišlo. Napríklad ak je vo vozidle 100 cestujúcich a koeficient zastávky je 0.7, pretože sa jedná o prestupnú zastávku, vystúpi 70 cestujúcich.

Výstupný signál **!Nastupovanie** mení vnútorný stav vozidla z **Na zastávke** na **Nakladá cestujúcich**. Zastávka, na ktorej sa vozidlo práve nachádza tento signál zachytí a vygeneruje nových cestujúcich. Pokiaľ to dovoľuje kapacita vozidla, všetci cestujúci nastúpia. V opačnom prípade nastúpia len tí, ktorí prišli skôr a tí čo prišli neskôr musia ostať čakať na ďalšie vozidlo.

Výstupný signál **!Odchod** mení vnútorný stav vozidla z **Nakladá cestujúcich** na **Na ceste**. Informuje sa príslušná zastávka a nič viac sa pri tomto prechode nevykonáva.

Obsluha zastávky

Každé vozidlo začína na prvej zastávke v rozpise, ale počas simulácie sa musí pohybovať medzi zastávkami. Proces ktorý vozidlo vykoná pri príchode na každú zastávku je popísaný v algoritme 4.

Algoritmus 4: Obsluha zastávky

prepojenie výstupných signálov vozidla so vstupnými signálmi zastávky;
aktivácia výstupného signálu **!Príchod**;
aktivácia výstupného signálu **!Nastupovanie**;
aktivácia výstupného signálu **!Odchod**;

Udalosť príchodu vozidla na zastávku

V rámci jednej udalosti sa vykoná rutina obsluženia cestujúcich, ktorí čakajú na zastávke. Táto rutina je popísaná v algoritme 4. Parametre tejto udalosti sú:

- **Čas príchodu** — čas, kedy vozidlo prichádza na zastávku
- **Zastávka** — zastávka, na ktorej sa udalosť vykonáva
- **Vozidlo** — vozidlo, ktoré prichádza na zastávku

Vozidlo začína udalosť v stave **Na ceste**, príde na zastávku uvedenú v parametri udalosti, vykoná rutinu nástupu a výstupu cestujúcich a odíde na ďalšiu zastávku. Za jednu udalosť teda vozidlo prejde všetkými svojimi vnútornými stavmi, udalosť končí v opačnom stave **Na ceste**. Zastávka ktorá je uvedená v parametri udalosti tiež prejde všetkými svojimi vnútornými stavmi na základe posielaných signálov od vozidla.

3.2 Implementácia simulačného modelu

Simulačný model je implementovaný v jazyku Python bez využitia externých knižníc. Trieda `EventCalendar` obsahuje kód základných operácií nad kalendárom udalostí:

- `isEmpty()` — vráti `True` ak je kalendár udalostí prázdny, inak `False`
- `addEvent(event)` — pridá udalosť do kalendára
- `getNextEvent()` — vráti udalosť s najmenším časom v kalendári a odstráni ju z kalendára

Trieda `Event` reprezentuje jednu udalosť. Obsahuje:

- metóda `__call__()` — vykoná akciu
- atribúty `action` a `actionArgument`

len metódu `__call__()`, ktorá vykoná udalosť uloženú v atribúte `action` s argumentami volania uloženými v `actionArgument`.

Modelový čas je implementovaný v triede `Simulation`. Obsahuje:

- metódy `forward(time)` a `getHour()`
- atribúty `startTime`, `currentTime` a `endTime`

Trieda `BusStop` reprezentuje jednu zastávku, rovnako tak trieda `Bus` reprezentuje jedno vozidlo. Tieto dve triedy sú písané podľa vzoru DEVS formalizmu. Obsahujú stavy, vstupné a výstupné signály a metódy, v ktorých sa nachádzajú akcie pre jednotlivé prechody medzi stavmi. Cestujúci sú uložení ako pole čísel, časov ich príchodu, v jednotlivých zastávkach v atribúte `waitingPassengersArrivalTimes`.

Na generovanie náhodného príchodu cestujúcich slúži trieda `RandomNumberGenerator`, ktorá obsahuje statické metódy na generovanie náhodných čísel.

Zber a agregácia štatistík (viď kapitola 3.3) sú implementované v triedach `Statistics`, `BusStatistics` a `BusStopStatistics`.

Podrobnejšie informácie o kóde sa nachádzajú v programovej dokumentácii, v prílohe A. Programová dokumentácia bola vygenerovaná pomocou nástroja Sphinx.

3.3 Výstupy simulačného modelu

Hlavný zmysel simulácie je zber informácií, ktoré by sme inak museli získať z pozorovania reálnej prevádzky linky. Efektivitu každého jedného časového rozpisu, ktorý program vytvorí je potrebné otestovať a ohodnotiť aby sme našli ten najlepší z nich, skôr ako ho začneme používať v reálnej prevádzke. Na ohodnotenie jednotlivých rozpisov slúžia štatistiky získané počas simulácie. Štatistiky sa zbierajú pre každú zastávku aj pre každý autobus osobitne. Následne sa agregujú do jedného objektu **Statistics**, ktorý obsahuje:

- **totalNumberOfBuses** — počet spojov v priebehu dňa na linke
- **busStopStatistics** — agregované štatistiky zastávok
- **busStatistics** — agregované štatistiky vozidiel

Trieda **busStopStatistics** slúži na ukladanie informácií o jednotlivých zastávkach, ale ako bolo spomenuté vyššie tak aj na uloženie agregovaných dát. Táto trieda slúži na zber nasledujúcich informácií:

- **totalPassengersArrived** — počet cestujúcich, ktorí prišli na zastávku za celý deň
- **totalPassengersDeparted** — počet cestujúcich, ktorí vystúpili na zastávke za celý deň
- **totalPassengersLeftUnboarded** — počet cestujúcich, ktorí museli čakať na ďalší spoj
- **totalTimeSpentWaiting** — celkový čas, ktorý cestujúci strávili čakaním na zastávke
- **totalPassengersLeftUnboarded** — počet cestujúcich, ktorých po celom dni nezobral žiaden spoj
- **passengersArrivedPerHour** — počet cestujúcich, ktorí prišli na zastávku v jednotlivých hodinách
- **passengersDepartedPerHour** — počet cestujúcich, ktorí vystúpili na zastávke v jednotlivých hodinách
- **passengersLeftUnboardedPerHour** — počet cestujúcich, ktorí museli čakať na ďalší spoj v jednotlivých hodinách
- **timeSpentWaitingPerHour** — čas, ktorý cestujúci strávili čakaním na zastávke v jednotlivých hodinách

Trieda **busStatistics** má rovnaký účel ako trieda **busStopStatistics**, ale pre vozidlá. Zbiera informácie:

- **capacity** — kapacita vozidla
- **averageLoad** — priemerná naplnenosť vozidla
- **averageLoadInPercent** — priemerná naplnenosť vozidla v percentách
- **totalPassengersTransported** — celkový počet prevezených cestujúcich
- **loadPerBusStop** — naplnenosť vozidla na jednotlivých zastávkach
- **loadInPercentPerBusStop** — naplnenosť vozidla na jednotlivých zastávkach v percentách

Počet spojov v priebehu dňa na linke

Tento údaj slúži pre optimalizáciu, z pohľadu dopravného podniku chceme, aby bol počet spojov čo najmenší aby ušetrili na prevádzkových nákladoch. Z pohľadu cestujúcich chceme aby bol počet spojov čo najväčší, aby nemuseli čakať na spoj dlhšie ako je nutné. Je potrebné nájsť kompromis medzi týmito dvoma extrémami.

Počet cestujúcich, ktorí prišli na zastávku za celý deň

Celkový počet cestujúcich za deň je dobrým indikátorom toho, ako sú namáhané jednotlivé časti linky. V prípadoch, kedy na zastávky v strede trasy prichádza oveľa viac cestujúcich ako na koncové zastávky, posilňuje dopravný podnik práve túto strednú časť ďalšími spojmi. Tieto spoje však na koncové zastávky už nechodia.

Počet cestujúcich, ktorí vystúpili na zastávke za celý deň

Je potrebné aby sme v rámci simulácie ráтали aj s cestujúcimi, ktorí z vozidiel vystupujú, inak by sa kapacita vozidla hneď naplnila. Ide o dôležitú časť reálneho systému, ktorú nemôžeme zanedbať ani pri simulácii. Slúži na validáciu modelu.

Počet cestujúcich, ktorí museli čakať na ďalší spoj

Občas sa stane, že je vozidlo tak preplnené, že nie všetci cestujúci, ktorí čakali na zastávke do neho nastúpia. V reálnom živote ide asi o subjektívny pocit, či nastúpiť alebo nie. V simulačnom modeli cestujúci prestanú nastupovať v momente keď naplnenosť vozidla dosiahne jeho maximálnu kapacitu. Rozpisy, pri ktorých sa tento jav vyskytne budú mať horšie hodnotenie.

Celkový čas, ktorý cestujúci strávili čakaním na zastávke

Z pohľadu optimalizácie chceme, aby bol tento čas čo najnižší. Príliš dlhé čakanie na zastávke bude rozpisy ohodnocovať negatívne. V prípade, že cestujúci musel čakať na ďalší spoj, tento čas bude ešte dlhší. Treba rátať aj s cestujúcimi, ktorí na zastávku prídu tesne pred odchodom vozidla. Napríklad ak chodí spoj dvakrát za hodinu, je oveľa vyššia pravdepodobnosť, že cestujúci nepríde v náhodný čas, ale svoj príchod na zastávku si naplánuje.

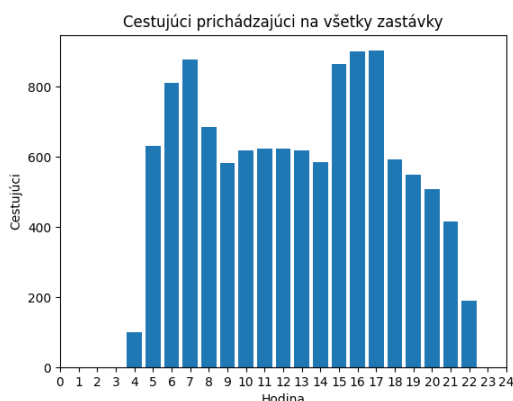
Počet cestujúcich, ktorých po celom dni nezobral žiaden spoj

Podobne ako v prípade cestujúcich, ktorí museli čakať na ďalší spoj, aj týchto cestujúcich by sme mali rátať a rozpis by mal byť ohodnotený veľmi negatívne. Tento jav by mohol v reálnom svete nastať asi iba ak by cestujúci išiel na posledný spoj v dni a toto vozidlo by bolo úplne preplnené. Ale keďže optimalizácia v tomto programe je riešená genetickým algoritmom, je veľká šanca že sa tento jav vyskytne častejšie. Táto vlastnosť je nežiaduca a je potrebné ju eliminovať.

Počet cestujúcich, ktorí prišli na zastávku v jednotlivých hodinách

Tento údaj má skôr informatívny charakter, pretože ide o vstup programu. Algoritmus sa snaží vytvoriť rozpis, ktorý by bol schopný prepraviť všetkých cestujúcich. Na grafe 3.4 je vidieť, že v priebehu dňa sa počet cestujúcich prichádzajúcich na zastávku mení. Obzvlášť

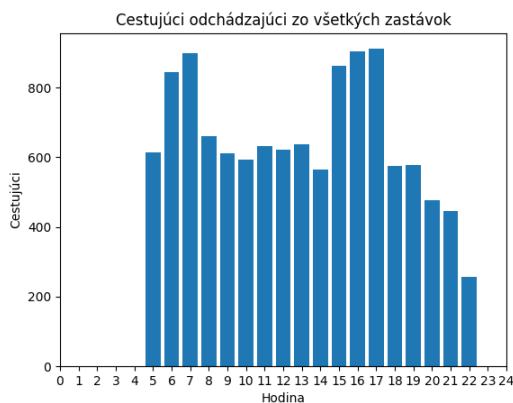
v ranných a poobedných hodinách je tento počet výrazne vyšší ako počas dňa. V týchto časoch by malo chodiť viac spojov.



Obr. 3.4: Počet cestujúcich, ktorí prišli na zastávku v jednotlivých hodinách

Počet cestujúcich, ktorí vystúpili na zastávke v jednotlivých hodinách

Ide o rovnaký údaj ako bol počet cestujúcich, ktorí vystúpili na zastávke za celý deň, ale rozdelený na jednotlivé hodiny dňa. Je vidieť, že graf 3.5 má podobný tvar ako graf 3.4. To je správne, pretože počet cestujúcich, ktorí vystúpili na zastávke by mal byť rovnaký ako počet cestujúcich, ktorí prišli na zastávku. Mohlo by sa stať, že tieto dve hodnoty by sa líšili, v prípade, že ostatnú cestujúcich, ktorí sa v priebehu dňa nezmestili do žiadneho vozidla.

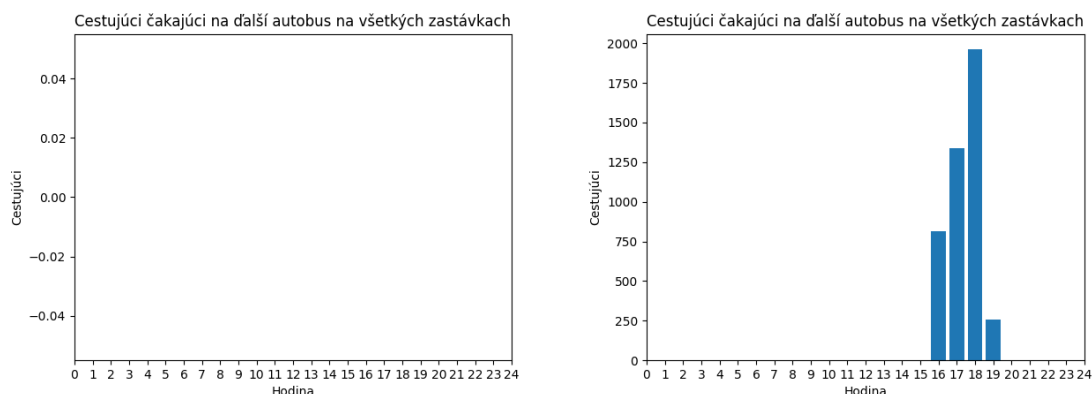


Obr. 3.5: Počet cestujúcich, ktorí vystúpili na zastávke v jednotlivých hodinách

Počet cestujúcich, ktorí museli čakať na ďalší spoj v jednotlivých hodinách

Na ľavej strane grafu 3.6 je vidieť, že v priebehu dňa žiaden z cestujúcich nemusel čakať na ďalší spoj. Toto je ideálny stav. Na pravej strane grafu 3.6 je vidieť, že v priebehu dňa sa počet cestujúcich, ktorí museli čakať na ďalší spoj mení. Ráno a okolo obeda nemusel na

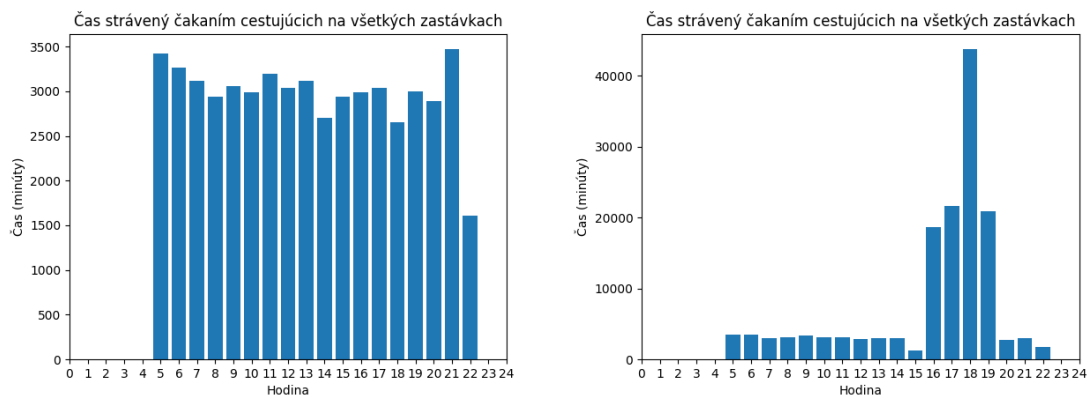
ďalší spoj čakať nikto, ale v poobednej špičke je vidieť zjavný nedostatok spojov, pretože cestujúci musia čakať na ďalší spoj.



Obr. 3.6: Počet cestujúcich, ktorí museli čakať na ďalší spoj v jednotlivých hodinách

Čas, ktorý cestujúci strávili čakaním na zastávke v jednotlivých hodinách

Čas strávený čakaním na zastávke by mal byť počas celého dňa približne rovnaký. Na ľavej strane grafu 3.7 je tento optimálny stav vidieť. Bez ohľadu na to, kedy cestujúci prídu na zastávku, čakajú na spoj rovnako dlho. Naopak na pravej strane grafu 3.7, ktorý je výstupom rovnakej simulácie ako pravá strana grafu 3.6, je zrejmé, že čas čakania na zastávke sa v poobedných hodinách výrazne zvýšil práve kvôli tomu, že cestujúci museli čakať na ďalší spoj.



Obr. 3.7: Čas, ktorý cestujúci strávili čakaním na zastávke v jednotlivých hodinách

Kapacita vozidla

Tento parameter je dôležitý, pretože na základe neho sa rozhoduje, koľko cestujúcich môže nastúpiť do vozidla. Zároveň slúži pre výpočet naplnenosti vozidla v percentách. Príliš malá alebo príliš veľká naplnenosť vozidla môže byť nežiaduca. Na ohodnotenie toho, ako moc je vozidlo naplnené teda treba poznať aj jeho kapacitu.

Priemerná naplnenosť vozidla

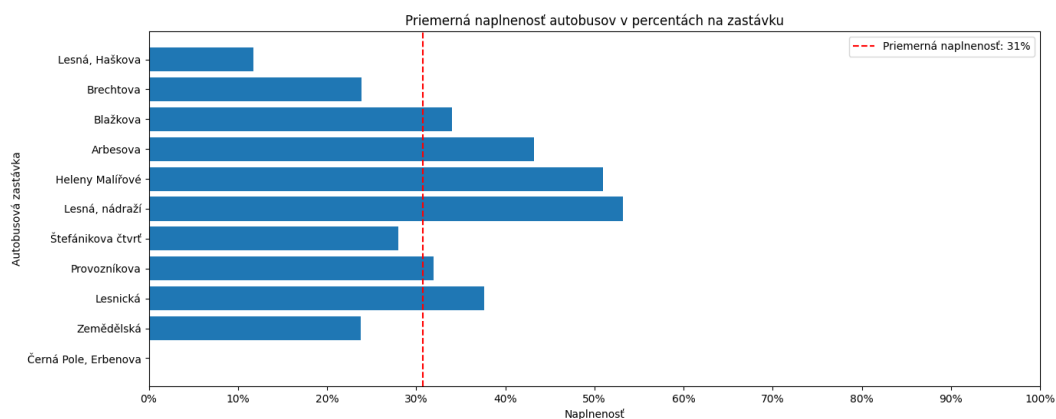
Priemerná naplnenosť vozidla je dôležitý parameter, ktorý nám hovorí, ako moc sú vozidlá využívané. Z pohľadu cestujúcich je lepšie, ak je vozidlo prázdnejšie, ale z pohľadu dopravného podniku je to nevyužitý potenciál. Preto je potrebné nájsť kompromis medzi týmito dvoma požiadavkami. To aká naplnenosť je optimálna je už úlohou analytika, ktorý bude tento program využívať. Povedzme, že sa rozhodne pre naplnenosť 70%. Potom čím ďalej bude priemerná naplnenosť vozidiel od tejto hodnoty, tým horšie bude rozpis ohodnotený.

Celkový počet prevezených cestujúcich

Celkový počet prevezených cestujúcich závisí od toho, koľko cestujúcich prišlo na jednotlivé zastávky a či sa im podarilo nastúpiť do vozidiel. V ideálnom prípade sa tento počet bude rovnať počtu cestujúcich, ktorí prišli na zastávky. V extrémnych prípadoch sa môže stať, že niektorí cestujúci zostanú na zastávkach a nezoberie ich žiadne vozidlo. V prípade štatistiky jednotlivých vozidiel, môže byť tento počet použitý na ich porovnanie. Počas dňa by mali všetky vozidlá prepraviť približne rovnaký počet cestujúcich. Ak tomu tak nie je a sú vyťažené nerovnomerne, rozpis by sa mal ohodnotiť negatívne.

Naplnenosť vozidla na jednotlivých zastávkach

Na grafe 3.8 je znázornené ako sa naplnenosť vozidiel mení počas ich jazdy. Konkrétne sa jedná o linku 46 v Brne. Naplnenosť sa postupne zvyšuje až kým nepríde na jednu z hlavných prestupných zastávok, Štefánikova čtvrt. Tu vystúpi veľké množstvo cestujúcich a naplnenosť vozidla sa zníži. Pokiaľ má linka príliš veľké výkivy v naplnenosti vozidiel naprieč svojou trasou, môže to byť znakom toho, že je potrebné trasu na určitých úsekoch posilniť inou linkou.



Obr. 3.8: Priemerná naplnenosť vozidla na jednotlivých zastávkach

Kapitola 4

Optimalizácia časového rozpisu linky mestskej hromadnej dopravy

Optimalizácia časového rozpisu je zložitý problém s veľkým množstvom premenných a obmedzení. Táto kapitola pojednáva o daných vstupoch, obmedzeniach, použitej optimalizačnej metóde a výstupoch optimalizácie.

Na optimalizáciu bol použitý genetický algoritmus. Genetický algoritmus je heuristická optimalizačná metóda, ktorá sa inšpiruje evolučnou biológiou. Je založená na princípe prírodného výberu, kde sa najlepšie jedince z populácie vyberajú na reprodukciu a vytvárajú novú generáciu. Tento proces sa opakuje, až kým sa nenájde optimálne riešenie alebo sa nedosiahne maximálny počet generácií. Na ohodnotenie jedinca sa používa fitness funkcia, ktorá hodnotí kvalitu riešenia na základe zadaných kritérií. V prípade časových rozpisov MHD sa na výpočet fitness funkcie používajú štatistiky zo simulácie, ktoré sú popísané v kapitole 3.3.

4.1 Vstupy potrebné od používateľa

Na to aby mohol algoritmus fungovať, je potrebné zadať niekoľko vstupných parametrov. Najdôležitejším z nich sú informácie o zastávkach. O každej zastávke je potrebné zadať:

- **Názov zastávky** — názov zastávky, ktorý sa zobrazuje na výstupoch
- **Čas príchodu** — čas potrebný na prejdienie vzdialenosti od prvej zastávky po túto zastávku
- **Štatistika o príchodoch cestujúcich** — priemerný počet prichádzajúcich cestujúcich v každej hodine dňa
- **Pravdepodobnosť výstupu cestujúceho** — údaj potrebný pre priebežné vyprázdenie vozidla

Tieto informácie sa zadávajú vo forme textového súboru, každá zastávka je na samostatnom riadoku a jednotlivé údaje sú oddelené dvojbodkou. Príklad toho ako by mali vyzerat záznamy o zastávkach:

```
Lesná, Haškova:0:[(5, 60), (6, 90), ..., (21, 40), (22, 30)]:0  
Brechtova:1:[(5, 60), (6, 90), ..., (21, 40), (22, 30)]:0.1  
Blažkova:2:[(5, 60), (6, 90), ..., (21, 40), (22, 30)]:0.1  
...
```

Ďalšie dôležité informácie o linke sú:

- **Kapacita vozidla** — maximálny počet cestujúcich, ktorý sa zmestí do vozidla
- **Miest na sedenie** — počet miest na sedenie v vozidle
- **Celkové náklady** — udávané v Kč/100 miesto-km (tabuľková hodnota)
- **Dĺžka trasy** — dĺžka trasy v km

Kapacita vozidla a počet miest na sedenie sú potrebné pre výpočet naplnenosti vozidla, od ktorého sa potom odvíja spokojnosť cestujúcich. Celkové náklady na 100 miestokilometrov a dĺžka trasy sú potrebné pre výpočet nákladov na prevádzku linky. Podrobnejší popis výpočtu celkovej spokojnosti a nákladov na prevádzku je v kapitole 4.2.

Potom je potrebné zadať parametre genetického algoritmu, ktoré sú:

- **Veľkosť populácie** — počet jedincov v jednej generácii
- **Počet generácií** — počet generácií, ktoré sa majú vykonať
- **Pravdepodobnosť mutácie** — pravdepodobnosť, že sa v jedincovi vyskytne mutácia
- **Maximálny počet spojov za hodinu** — maximálny počet spojov, ktorý môže byť vygenerovaný za hodinu

Poslednou informáciou, ktorá je od používateľa potrebná sú obmedzenia na počet spojov za hodinu. Používateľ takto môže nastaviť fixný počet spojov, ktorý sa má vygenerovať za hodinu. Toto sa využíva hlavne pri plánovaní návaznosti liniek, kedy je potrebné aby sa niektoré spoje stretli na prestupnej zastávke. Obvykle sa to robí tak, že sa nastaví rovnaký počet spojov pre všetky prestupné linky, ktoré sa na tejto zastávke stretávajú. Toto obmedzenie využíva aj predbežná analýza, ktorá sa vykonáva pred optimalizáciou. V prípade ak pre niektorú hodinu neexistuje údaj o priemernom počte prichádzajúcich cestujúcich naprieč všetkými zastávkami, nastaví sa fixný počet spojov v danej hodine na 0. Tento krok výrazne prispieva k lepším výsledkom optimalizácie, pretože dopredu vieme, že v tejto hodine nebude žiaden spoj potrebný. Zároveň s nižším počtom spojov vo všetkých rozpisoch počas optimalizácie sa výpočet výrazne urýchli.

4.2 Základná implementácia genetického algoritmu

Na optimalizáciu časového rozpisu bol použitý genetický algoritmus. Genetický algoritmus je heuristická optimalizačná metóda, ktorá sa inšpiruje Darwinovou teóriou evolúcie. Podľa citátu od Charlesa Darwina, "V prírode je prežitie a úspech otázkou tých najschopnejších a najpríťažlivejších", aj genetický algoritmus sa snaží nájsť najlepšie riešenie problému pomocou evolučného procesu. Používajú sa postupy ako je náhodná mutácia jedince a rozne typy výberu a kríženia jedincov. Myšlienka je, že najlepší jedinci v generácii buď splňujú požiadavky alebo sú veľmi blízko k optimálnemu riešeniu. Predpokladá sa, že tieto jedinci majú najlepšie predispozície na to, aby ich potomkovia boli o krok bližšie k optimálnemu riešeniu. Využitie genetického algoritmu je efektívne v prípadoch, kedy by deterministické metódy trvali príliš dlho alebo by vôbec neboli schopné nájsť optimálne riešenie. Genetický

algoritmus bol prvýkrát predstavený okolo roku 1960 Johnom Hollandom na Univerzite v Michigane (viď [4]).

Algoritmus 5: Genetický algoritmus

```
Inicializácia prvej populácie;  
Definovanie fitness funkcie;  
Výpočet fitness funkcie pre každého jedinca v populácii;  
while nie je nájdené optimálne riešenie alebo nebol dosiahnutý maximálny počet  
  generácií do  
    Výber rodičov z populácie;  
    Kríženie rodičov na vytvorenie potomkov;  
    Mutácia potomkov;  
    Výpočet fitness funkcie pre každého potomka;  
Výber najlepšieho jedinca z populácie;
```

V prípade optimalizácie časového rozpisu MHD sa jedincami stávajú samotné časové rozpisy. Na základe simulácie jedného dňa na linke s použitým časovým rozpisom sa získajú štatistiky, ktoré sa následne použijú na ohodnotenie kvality jedinca.

Výpočet cieľov, spokojnosť cestujúcich a náklady na prevádzku

Pre ohodnotenie kvality časového rozpisu sa zvolili dva hlavné ciele:

- **Spokojnosť cestujúcich** — priemerné percentuálne zhodnotenie spokojnosti všetkých cestujúcich, ktorí prišli za deň na zastávku
- **Náklady na prevádzku** — celkové náklady na prevádzku linky za deň, udávané v Kč/100 miesto-km

Výpočet spokojnosti cestujúceho sa vykonáva v čase nástupu do vozidla. Zhodnocuje sa aktuálna naplnenosť vozidla a na základe toho sa vypočíta percentuálne zhodnotenie spokojnosti. Potrebne parametre pre výpočet sú kapacita vozidla, počet miest na sedenie a aktuálna naplnenosť vozidla. Výpočet sa riadi nasledujúcimi troma podmienkami:

1. Ak sú vo vozidle voľné miesta na sedenie, cestujúci si sadne a jeho spokojnosť je 1
2. Ak je vozidlo úplne plné a cestujúci sa do neho nezmestí, jeho spokojnosť je 0
3. Ak je vo vozidle miesto, ale už len na státie, spokojnosť vozidla závisí od aktuálnej naplnenosti vozidla a je vypočítaná ako:

$$\text{spokojnosť} = 1 - \frac{\text{aktualna naplnenosť vozidla} - \text{počet miest na sedenie}}{\text{kapacita vozidla} - \text{počet miest na sedenie}} \quad (4.1)$$

Nakonci sa vypočíta priemer spokojnosti všetkých cestujúcich. Hodnota sa pohybuje od 0 do 1 a genetický algoritmus uprednostňuje vyššie hodnoty, teda spokojnejších cestujúcich.

Celkové náklady na prevádzku linky sú vypočítané z počtu spojov v časovom rozpise, kapacity použitých vozidiel, dĺžky trasy a nákladov na jeden miestokilometer. Jeden miestokilometer je definovaný ako vzdialenosť jedného kilometra, ktorú prejde jedno miesto vo vozidle. Potom vozidlo s kapacitou 80 miest, ktoré prejde 10 km, prejde 800 miestokilometrov. Miestokilometre sa môžu násobiť aj počtom vozidiel, potom dve takéto vozidlá

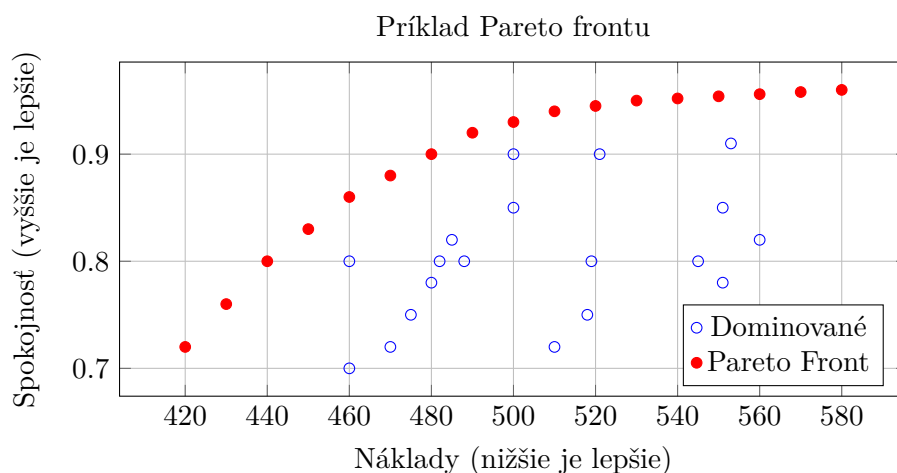
by prešli 1600 miestokilometrov. Informácie o tejto jednotke sú prevzaté od pána Michala Hlaváčka, zamestnanca Dopravného podniku mesta Brno, pomocou mailovej komunikácie, ktorá je dostupná v prílohe F. Tento údaj je potrebný pretože náklady vo finančnom pláne Dopravného podniku mesta Brno sú udávané v českých korunách na 100 miestokilometrov. Finančný plán na rok 2025 je v prílohe G. Celý výpočet nákladov na prevádzku linky teda závisí od štyroch premenných a je daný vzorcom:

$$\text{celkové náklady} = \frac{\text{dĺžka trasy} \cdot \text{počet spojov} \cdot \text{kapacita vozidla}}{100} \cdot \text{náklady} \quad (4.2)$$

Tieto dva ciele sú v konflikte, pretože čím viac spojov je na linke, tým je spokojnosť cestujúcich vyššia, ale náklady na prevádzku sú tiež vyššie. Tento problém rieši multi-objektívna optimalizácia, ktorá sa snaží nájsť optimálne riešenie pre oba ciele naraz.

Genetický algoritmus s viacero cieľmi

Genetický algoritmus, v ktorom sa snažíme dosiahnuť optimálneho kompromisu medzi viacero cieľmi sa nazýva multi-objektívny genetický algoritmus. Jeho myšlienka spočíva v nájdení Pareto frontu, čo je množina všetkých možných riešení, ktoré sú optimálne z pohľadu všetkých cieľov. To znamená, že neexistuje žiaden iný jedinec, ktorý by bol lepší v oboch cieľoch naraz. Ak je jeden jedinec lepší v oboch cieľoch, hovoríme, že dominuje iného jedinca. V prípade, že dominuje iba v jednom, ale v druhom nie, patria do rovnakej skupiny, sú si rovnocenní. Ak je jedinec horší v oboch cieľoch, hovoríme, že je dominovaný. Každá generácia sa rozdelí do rovnocenných skupín, pri výbere rodičov sa uprednostňujú jedinci z lepšej skupiny. Najlepšia skupina sa nazýva Pareto front, ide o množinu takých jedincov, nad ktorými nedominuje žiaden iný jedinec. Na obrázku 4.1 je znázornený Pareto front pre dva ciele, spokojnosť a náklady. Multi-objektívne genetické algoritmy dokážu pracovať aj s viac ako dvoma cieľmi. Pre optimalizáciu časového rozpisu však stačia dva, jeden zo strany zákazníka (spokojnosť) a druhý zo strany dopravného podniku (náklady). Tieto základné informácie o fungovaní multi-objektívnych genetických algoritmov s použitím Pareto frontu sú prevzaté z článku [5].



Obr. 4.1: Príklad Pareto frontu

4.3 Implementácia multi-objektívneho genetického algoritmu

Pre účely multi-objektívnej optimalizácie bol použitý algoritmus NSGA-II (Non-dominated Sorting Genetic Algorithm II). V článku [2] je porovnanie tohto algoritmu s inými multi-objektívnymi genetickými algoritmami. NSGA-II vo viacerých testoch preukázal, že má lepšie rozloženie a diverzitu jedincov v Pareto fronte ako iné algoritmy. Práve pre túto vlastnosť bol zvolený aj pre túto prácu. Okrem testov je v článku [2] aj podrobný popis fungovania algoritmu. Algoritmy 7, 8, 9 a 10 boli z tohto článku prevzaté a mierne upravené pre účely tejto práce.

NSGA-II je založený na triedení jedincov do frontov na základe ich dominancie nad inými jedincami. Pri výbere jedincov pri krížení sa uprednostňujú jedinci z lepších frontov. V prípade, že treba porovnať dvoch jedincov z rovnakého frontu, použije sa zhluková vzdialenosť. Zhluková vzdialenosť zaisťuje vysokú diverzitu jedincov, pretože sa uprednostňujú jedinci, ktorí sú od seba najďalej, teda majú veľké rozdiely v hodnotách cieľov.

Oproti klasickému genetickému algoritmu NSGA-II spracováva elitizmus inak. V NSGA-II neexistuje konštanta, ktorá by udávala počet elitných jedincov, ktorí sa prenású do ďalšej generácie. Pri vytváraní novej generácie sa najprv vytvorí nová populácia z rodičov a ich potomkov. Táto, dvojnásobne veľká populácia sa následne rozdelí do frontov a pre každého jedinca sa vypočíta zhluková vzdialenosť na základe jeho vzdialenosti od svojich susedov vo fronte. Potom sa najlepšie fronty prenású do novej generácie. Posledný prenášaný front sa pravdepodobne nezmesť celý, vtedy sa vyberie toľko jedincov z daného frontu aby bola populácia rovnako veľká naprieč všetkými generáciami. Jedinci z posledného frontu sa vyberajú na základe zhlukovej vzdialenosti, aby sa zachovala vyššia diverzita.

Inicializácia prvej generácie

Populácia prvej generácie sa musí spracovať trochu inak ako ostatné generácie. Hlavný cyklus algoritmu NSGA-II sa začína spojením rodičov a potomkov do jedného celku. Na začiatku však žiadnych potomkov nemáme. Algoritmus 6 popisuje ako sa vytvorí populácia prvej generácie a jej potomkovia. Najprv sa vytvorí prvá generácia jedincov, tým sa náhodne vygenerujú chromozómy. Následne sa tieto jedinci roztriedia na fronty pomocou rýchleho nedominantného triedenia. V rámci každého frontu sa vypočíta zhluková vzdialenosť každého jedinca. Na základe získaných ohodnotení sa vyberú tí lepší rodičia a za pomoci kríženia a mutácie sa vytvorí prvá generácia potomkov.

Algoritmus 6: Inicializácia prvej generácie

Input: Veľkosť populácie N
 $t \leftarrow 0$ (počiatočná generácia);
 $P_t \leftarrow \emptyset$ (inicializácia populácie);
for $i = 0$ **do** $N - 1$ **do**
 $p_i \leftarrow \text{vytvorenie_jedinca}()$;
 $P_t \leftarrow P_t \cup \{p_i\}$;
 $F \leftarrow \text{rýchle_nedominantné_triedenie}(P_t)$;
for *každý front* $f \in F$ **do**
 $\text{priradenie_zhlukovej_vzdialenosti}(f)$;
 $Q_t \leftarrow \text{vytvorenie_nových_potomkov}(P_{t+1})$ (kríženie a mutácia);

Hlavná slučka algoritmu NSGA-II

Po inicializácii prvej generácie sa ostatné môžu spracovávať rovnako. Ako je vidieť v algoritme 7, hlavná slučka začína spojením rodičov a potomkov do jednej populácie. Novo vzniknutá populácia sa následne rozdelí do frontov na základe ich dominancie. Následuje kombinovaný cyklus na spracovanie jednotlivých frontov, ktorý ma dve úlohy:

1. Priradenie zhlukovej vzdialenosti jedincov v aktuálnom fronte
2. Priradenie jedincov aktuálneho frontu do novej generácie

Tento cyklus sa vykonáva pokým sa celý nasledujúci front zmestí do novej generácie. Posledný front, ten ktorý sa už nezместí celý, sa musí spracovať osobitne. Prvky z tohto frontu sa usporiadajú na základe zhlukovej vzdialenosti a do novej generácie sa prenesú len tie jedince, ktoré sa tam zmestia. Z novovzniknutej populácie sa následne vytvorí nová generácia potomkov. Vo všeobecnosti sa táto slučka vykonáva pokiaľ nie je dosiahnutý maximálny počet generácií alebo pokiaľ nebol nájdený optimálny jedinec. V prípade tejto práce sa však bere do úvahy iba maximálny počet generácií.

Algoritmus 7: Hlavná slučka algoritmu NSGA-II

```
 $R_t \leftarrow P_t \cup Q_t$  (zlučovanie rodičov a potomkov);  
 $F \leftarrow \text{rýchle\_nedominantné\_triedenie}(R_t)$ ;  
 $P_{t+1} \leftarrow \emptyset$  (inicializácia novej populácie);  
 $i \leftarrow 0$ ;  
while  $|P_{t+1}| + |F_i| \leq N$  do  
    priradenie_zhlukovej_vzdialenosti( $F_i$ );  
     $P_{t+1} \leftarrow P_{t+1} \cup F_i$ ;  
     $i \leftarrow i + 1$ ;  
 $\text{sort}(F_i, \prec_n)$ ;  
 $P_{t+1} \leftarrow P_{t+1} \cup F_i[0 : N - |P_{t+1}|]$ ;  
 $Q_{t+1} \leftarrow \text{vytvorenie\_nových\_potomkov}(P_{t+1})$  (kríženie a mutácia);  
 $t \leftarrow t + 1$ ;
```

Rýchle nedominantné triedenie

Rýchle nedominantné triedenie je základom NSGA-II. Triedi jedincov do frontov na základe ich dominancie nad inými jedincami. Jedince dominuje iného jedinca iba v prípade, ak je vo všetkých cieľoch aspoň rovnako dobrý a v aspoň jednom je lepší. Jedince, ktoré sa navzájom nedominujú, patria do rovnakej skupiny, frontu, preto sa toto triedenie nazýva nedominantné. Triedenie sa vykonáva v dvoch krokoch:

- Výpočet S_p a n_p pre každého jedinca a pridanie jedincov bez dominancie do prvého frontu
- Vytvorenie zvyšných frontov za základe získaných hodnôt S_p a n_p z predchádzajúceho kroku

V prvom kroku sa musia porovnať všetci jedinci navzájom. Pre každého jedinca p v prvom cykle algoritmu 8 sa vypočíta množina S_p , ktorá obsahuje všetkých jedincov, ktorých

dominuje p a počet n_p , ktorý udáva počet jedincov, ktorí dominujú p . Pokiaľ je $n_p = 0$, jedinec p nie je nikdy dominovaný a pripojí sa do prvého frontu. Taktiež sa mu priradí hodnota $p_{rank} = 1$. Rank je hodnota, ktorá udáva v akom fronte sa jedinec nachádza, čím nižšia hodnota, tým je jedinec lepší.

Následuje druhý hlavný cyklus algoritmu 8, ktorý vytvorí zvyšné fronty. Princíp je taký, že pre každého jedinca p v posledne vytvorenom fronte sa upraví hodnoty n_q všetkých jedincov q z množiny S_p . To v podstate znamená odstránenie jedinca p z riešeného problému. Na konci tohto cyklu by sa mal z problému odstrániť celý front F_i . TO znamená, že niektorí jedinci by sa mali stať tými najlepšími v tomto obmedzenom probléme. Práve títo jedinci sa pridávajú do ďalšieho frontu F_{i+1} . Tento cyklus sa opakuje až pokiaľ sa do najnovšieho frontu nemá kto pridať.

Toto triedenie teda iteratívne nachádza najlepší front a odstraňuje ho z problému až pokiaľ sa nenájdu všetky fronty.

Algoritmus 8: Rýchle nedominantné triedenie

Input: Populácia P

Output: Fronty nedominovaných jedincov F

$F \leftarrow \emptyset$ (množina frontov);

for každý jedinec $p \in P$ **do**

$S_p \leftarrow \emptyset$ (množina jedincov, ktorých dominuje p);

$n_p \leftarrow 0$ (počet jedincov, ktorí dominujú p);

for každý jedinec $q \in P : q \neq p$ **do**

if p dominuje q **then**

$S_p \leftarrow S_p \cup \{q\}$;

else if q dominuje p **then**

$n_p \leftarrow n_p + 1$;

if $n_p = 0$ **then**

$p_{rank} \leftarrow 1$;

$F_1 \leftarrow F_1 \cup \{p\}$ (pridanie jedinca p do prvého frontu);

$i \leftarrow 1$;

while $F_i \neq \emptyset$ **do**

$Q \leftarrow \emptyset$ (množina jedincov ďalšieho frontu);

for každý jedinec $p \in F_i$ **do**

for každý jedinec $q \in S_p$ **do**

$n_q \leftarrow n_q - 1$;

if $n_q = 0$ **then**

$q_{rank} \leftarrow i + 1$;

$Q \leftarrow Q \cup \{q\}$;

$i \leftarrow i + 1$;

$F_i \leftarrow Q$;

return F ;

Priradenie zhlukovej vzdialenosti

V rámci jedného frontu sa jedinci porovnávajú na základe zhlukovej vzdialenosti. Zhluková vzdialenosť je definovaná ako vzdialenosť jedinca od jeho susedov vo fronte. Vypočítava sa pre každý cieľ zvlášť a následne sa spočíta. Najdôležitejšou časťou algoritmu 9 je cyklus, ktorý sa vykonáva pre každý cieľ c . Najprv sa jedinci zoradia podľa dosiahnutých hodnôt cieľa c . Následne sa vytýčia krajné body, jedincom s najmenšou a najväčšou hodnotou cieľa c sa priradí nekonečná zhluková vzdialenosť, pretože sú najviac diverzní. Pre všetkých ostatných jedincov vo fronte a pre cieľ c sa vypočíta zhluková vzdialenosť ako podiel rozdielu hodnôt susedov a rozdielu krajných jedincov. Tieto dielčie vzdialenosti sa postupne sčítavajú do celkovej zhlukovej vzdialenosti jedinca.

Algoritmus 9: Priradenie zhlukovej vzdialenosti

Input: Front L

$l \leftarrow |L|$ (počet jedincov frontu);

for každý jedinec $p \in L$ **do**

$p_{vzdialenosť} \leftarrow 0$ (inicializácia zhlukovej vzdialenosti);

for každý cieľ c **do**

$L \leftarrow \text{sort}(L, c)$ (zoradenie jedincov podľa hodnôt cieľa c);

$L[0] \leftarrow \infty$ (priradenie nekonečnej vzdialenosti najmenšiemu jedincovi);

$L[l-1] \leftarrow \infty$ (priradenie nekonečnej vzdialenosti najväčšiemu jedincovi);

for $i = 1$ **do** $l - 2$ **do**

$L[i]_{vzdialenosť} \leftarrow L[i]_{vzdialenosť} + \frac{L[i+1].c - L[i-1].c}{L[l-1].c - L[0].c};$

Porovnávací operátor \prec_n

Informácie získané z rýchleho nedominantného triedenia 8 a priradenia zhlukovej vzdialenosti 9 sa použijú na porovnanie dvoch jedincov. Pokiaľ je jedinec p z lepšieho frontu ako jedinec q , teda má lepší, nižší rank, p dominuje q . Ak sú jedinci z rovnakého frontu, porovnávajú sa na základe zhlukovej vzdialenosti. Dominuje ten, ktorý má väčšiu zhlukovú vzdialenosť, to znamená, že sa výraznejšie odlišuje od svojich susedov vo fronte. Tento porovnávací operátor je znázornený v algoritme 10.

Algoritmus 10: Porovnávací operátor \prec_n

Input: Jedinci p a q

Output: True ak p dominuje q , inak False

return $p_{rank} < q_{rank} \vee (p_{rank} = q_{rank} \wedge p_{vzdialenosť} > q_{vzdialenosť})$;

(dá sa rozšíriť aj o obmedzenia);

Toto porovnávanie sa dá rozšíriť aj o rôzne obmedzenia. Napríklad, v rámci optimalizácie časového rozpisu máme dva ciele, spokojnosť zákazníkov a celkové náklady, ale porovnávanie sme rozšírili o to, v ktorom rozpise dôjde menej často k úplnému preplneniu vozidla. Sledujeme počet prípadov, kedy sa nejaký cestujúci nezmestil do vozidla. V prípade, že aspoň jeden z porovnávaných jedincov má tento počet iný ako 0, vyhráva ten, ktorý má tento počet menší. Všeobecne môže byť obmedzení aj viac, v takom prípade sa porovnáva počet

porušenia daných obmedzení. Napríklad ak by sme mali tri rôzne obmedzenia, časový rozpis môže porušovať 0, 1, 2 alebo 3 z nich. Vyhráva ten rozpis ktorý ich porušuje menej. V prípade ak oba rozpisy porušujú rovnaký počet obmedzení, porovnáme ich podľa operátora \prec_n . Pre potreby tejto práce máme len jedno spomínané obmedzenie, bližšie informácie k implementácii viacerých obmedzení sú v článku [2].

Vytvorenie nových potomkov

Genetický algoritmus by sa nezaobišiel bez vytvárania nových potomkov. Tento proces sa dá rozdeliť na tri hlavné časti:

1. Výber rodičov
2. Kríženie rodičov
3. Mutácia potomkov

Z každého kríženia vzniknú dvaja noví potomkovia, to znamená, že tento proces asa musí vykonať $N/2$ krát, pokiaľ je veľkosť populácie N . Druhá možnosť ako ukončiť tento proces je znázornená v algoritme 11, kde sa noví potomkovia pridávajú do množiny Q pokiaľ je ich počet menší ako N .

Algoritmus 11: Vytvorenie nových potomkov

Input: Populácia P

Output: Noví potomkovia Q

$Q \leftarrow \emptyset$ (inicializácia nových potomkov);

while $|Q| < N$ **do**

```

     $p_1 \leftarrow \text{výber\_rodiča}(P);$ 
     $p_2 \leftarrow \text{výber\_rodiča}(P);$ 
     $q_1, q_2 \leftarrow \text{kríženie\_rodičov}(p_1, p_2);$ 
     $\text{mutácia\_jedinca}(q_1, );$ 
     $\text{mutácia\_jedinca}(q_2);$ 
     $Q \leftarrow Q \cup \{q_1, q_2\};$ 

```

return Q ;

Výber rodiča

Pred tým ako sa budú rodičia medzi sebou krížiť, je potrebné ich vybrať. Pre túto prácu bol zvolený turnajový výber rodiča. Turnajový výber funguje tak, že sa náhodne vyberie k jedincov z populácie, z ktorých sa turnajovým spôsobom vyberie ten najlepší. V algoritme 12 je zvolený turnajový výber s $k = 2$, *binary tournament*, teda sa vyberú dvaja náhodní jedinci a porovnajú sa pomocou porovnávacieho operátora \prec_n .

Turnajový výber bol pre túto prácu zvolený, pretože je jednoduchý na implementáciu a má dobré výsledky v porovnaní s inými typmi výberu rodičov. V článku [3] je porovnanie roznych typov výberu rodičov. Turnajový výber má veľmi podobné výsledky ako náhodný výber rodiča, pravdepodobne z dôvodu malého počtu jedincov v turnaji. Vyšší počet by však mohol celú optimalizáciu spomaliť. Pokusmi sme odhadli počet jedincov v turnaji na 2.

Algoritmus 12: Výber rodiča

Input: Populácia P **Output:** Rodič p $p_1 \leftarrow$ náhodný jedinec z P ; $p_2 \leftarrow$ náhodný jedinec z P ;**if** $p_1 \prec_n p_2$ **then** **return** p_1 ;**else** **return** p_2 ;

Kríženie rodičov

Kríženie znamená výmenu génov v chromozónoch rodičov. Tým vznikajú noví jedinci, potomkovia. V algoritme 13 je znázornené kríženie dvoch rodičov. Pre každý gén v chromozónoch rodičov sa náhodne vyberie jeden z rodičov a jeho gén sa priradí prvému potomkovi. Gén z druhého rodiča sa priradí druhému potomkovi. Tento proces sa opakuje pre všetky gény v chromozónoch rodičov. Je vidieť, že takýmto krížením vzniknú dvaja noví jedinci, preto stačí toto kríženie vykonať len $N/2$ krát, pokiaľ je veľkosť populácie N .

Tento typ kríženia je známy ako *uniform crossover* a je veľmi populárny v genetických algoritmoch. Porovnanie rôznych typov kríženia je v článku [1]. Na základe výsledkov v tomto článku bol zvolený typ kríženia s najväčšou diverzitou výsledkov aby sa zamedzilo predčasnej konvergencii, teda aby sa algoritmus nezastavil na lokálnom minime.

Algoritmus 13: Kríženie rodičov

Input: Rodičia p_1 a p_2 **Output:** Potomkovia q_1 a q_2 **for** každý gén g v p_1 **do** **if** $\text{random}(0, 1) < 0,5$ **then** $q_1[g] \leftarrow p_1[g]$; $q_2[g] \leftarrow p_2[g]$; **else** $q_1[g] \leftarrow p_2[g]$; $q_2[g] \leftarrow p_1[g]$;**return** q_1, q_2 ;

Mutácia jedinca

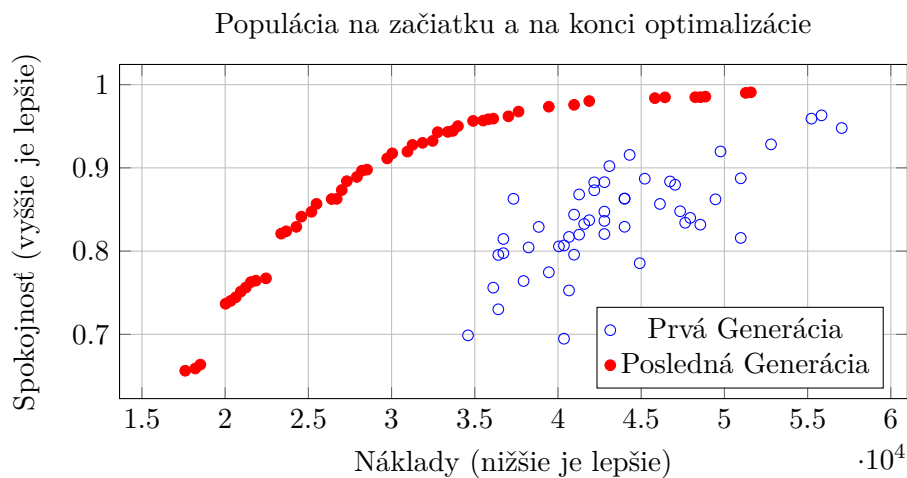
Posledným krokom v procese vytvorenia nových potomkov je mutácia. Mutácia zvyšuje diverzitu jedincov a zabraňuje predčasnému uviaznutiu v lokálnych extrémoch. V algoritme 14 je znázornená mutácia jedinca. Pre každý gén v chromozóme sa náhodne vyberie číslo z intervalu $[0, 1]$. Ak je toto číslo menšie ako pravdepodobnosť mutácie, gén sa nahradí náhodným génom.

Algoritmus 14: Mutácia jedinca

Input: Jedinca p
for každý gén g v p **do**
 if $\text{random}(0, 1) < \text{pravdepodobnosť mutácie}$ **then**
 $p[g] \leftarrow$ náhodný gén;
end for

4.4 Výstupy optimalizácie

Výstupom klasického genetického algoritmu by mal byť jedinec s najlepším ohodnotením. V prípade NSGA-II je to front s najlepšími jedincami. Keďže všetky jedince v najlepšom fronte sú rovnako dobré, nedá sa jednoznačne určiť, ktorý z nich je lepší. Ich ohodnotenie je závislé od viacerých cieľov, v prípade tejto práce dvoch. Po skončení optimalizácie sa celá posledná generácia približuje optimálnemu frontu riešení. To znamená, že všetky časové rozpisy v poslednej generácii sú rovnako dobrým výsledkom. Pre výber toho najlepšieho jedinca je potrebné rozhodnúť sa, ktorý z cieľov je dôležitejší, prípadne ich váhy. Táto práca sa už musí nechať na používateľa programu, ktorý sa sám rozhodne, ktorý z rozpisov je pre neho lepší. Z výsledkov priebežných experimentov počas vývoja programu sa ukázalo, že rozpisy s vyššou spokojnosťou cestujúcich majú z pravidla aj vyššie celkové náklady. Ako postupne klesajú náklady, klesá aj spokojnosť cestujúcich, pretože sa znižuje počet spojov, to má za následok preplnenejšie vozidlá a dlhšie čakacie doby na zastávkach. Tieto experimenty sú bližšie popísané v kapitole 5.



Obr. 4.2: Populácia na začiatku a na konci optimalizácie

Na obrázku 4.2 je vidieť priaznivé výsledky optimalizácie. Algoritmus NSGA-II sa po 100 generáciách priblížil k optimálnemu frontu riešení. Oproti prvej generácii sa značne znížili náklady a zvýšila spokojnosť cestujúcich. Tento výsledok ukazuje, že genetický algoritmus dokáže efektívne nájsť kompromis medzi protichodnými cieľmi, ako sú náklady a spokojnosť. Použitie NSGA-II umožnilo zachovať diverzitu riešení, čo poskytuje používateľovi možnosť výberu najvhodnejšieho časového rozpisu podľa jeho preferencií.

Kapitola 5

Experimenty

V tejto kapitole sú podrobne popísané experimentálne scenáre, ktoré boli navrhnuté na posúdenie výkonnosti optimalizačného algoritmu aplikovaného na plánovanie liniek MHD. Cieľom bolo preskúmať, ako algoritmus reaguje na zmeny v parametroch ako sú dĺžka trasy, dĺžka vozidla a počet zastávok.

Každý experiment bol vykonaný s rovnakými parametrami pre genetický algoritmus, aby sa zabezpečila konzistentnosť výsledkov:

- **Počet jedincov v populácii** — 50 pre krátku trasu, 10 pre dlhú trasu ¹
- **Počet generácií** — 100
- **Pravdepodobnosť mutácie** — 5%
- **Maximálny počet spojov za hodinu** — 15

Ostatné parametre sú špecifické pre jednotlivé experimenty a sú uvedené v tabuľkách v príslušných sekciách.

Parametre zariadenia na ktorom boli experimenty vykonané sú:

- **Operačný systém** — Windows 11 Pro 64-bit
- **Procesor** — Intel Core i7-9750H CPU @ 2.60GHz
- **RAM** — 16 GB DDR4

Všetky experimenty boli vykonané na rovnakom zariadení, aby sa zabezpečila konzistentnosť výsledkov, a v rovnakom čase, aby sa minimalizoval vplyv vonkajších faktorov. Každý experiment trval približne od 45 minút do 1.5 hodiny v závislosti od dĺžky trasy a počtu zastávok.

¹Tento počet bol znížený z dôvodu dlhšej doby výpočtu.

5.1 Experiment s krátkou trasou a krátkým vozidlem

Cílem tohoto experimentu je zjistit, ako sa algoritmus vyrovnáva s krátkou trasou a krátkým vozidlom. Ako referencia bola zvolená linka 46 v Brne, ktorej trasa pozostáva z 11 zastávok. Parametry experimentu sú uvedené v tabuľke 5.1.

| Vstup | Hodnota |
|------------------|------------------------|
| Kapacita vozidla | 80 |
| Miest na sedenie | 30 |
| Celkové náklady | 92,82 Kč/100 miesto-km |
| Dĺžka trasy | 3,8 km |

Tabuľka 5.1: Parametre vozidla a trasy

Ako je vidieť v tabuľke 5.2, optimalizácia prebehla úspešne. Algoritmus bol schopný vygenerovať rozpis, pri ktorom je spokojnosť cestujúcich 94% a počet cestujúcich. Vozidlá sú v priemere zaplnené do jednej tretiny, pretože ich je až 115, to je o 3 spoje menej ako je aktuálny rozpis linky 46.

| Výstup | Hodnota |
|----------------------------------|--------------|
| Počet príchodov cestujúcich | 11603 |
| Počet prevezených cestujúcich | 11603 |
| Počet neobslužených cestujúcich | 0 (0,0 %) |
| Celkový čas strávený čakaním | 54625 min |
| Priemerný čas strávený čakaním | 5 min |
| Celkové náklady | 32450 Kč |
| Priemerná spokojnosť cestujúcich | 94,0 % |
| Celkový počet vozidiel | 115 |
| Priemerná naplnenosť vozidiel | 25 (31,02 %) |

Tabuľka 5.2: Výstupy simulácie

Celý rozpis zastávok, časový rozpis a grafy podrobnejšie popisujúce tento experiment sú uvedené v prílohe B.

5.2 Experiment s krátkou trasou a dlhým vozidlom

Cieľom tohto experimentu je zistiť, ako sa algoritmus vyrovnáva s krátkou trasou a dlhým vozidlom. Ako referencia bola zvolená linka 46 v Brne, ktorej trasa pozostáva z 11 zastávok. Parametry experimentu sú uvedené v tabuľke 5.3.

| Vstup | Hodnota |
|------------------------|------------------------|
| Kapacita vozidla | 160 |
| Počet miest na sedenie | 60 |
| Celkové náklady | 92,82 Kč/100 miesto-km |
| Dĺžka trasy | 3,8 km |

Tabuľka 5.3: Parametre vozidla a trasy

V tabuľke 5.4 sú vidieť zmeny oproti predošlému experimentu. Navýšenie kapacity jedného vozidla má vplyv na:

- Priemerná čakacia doba sa zvýšila o 2 minúty
- Spokojnosť cestujúcich sa zvýšila o 4,54%
- Počet vozidiel sa znížil o 43 kusov
- Priemerná naplnenosť vozidiel sa znížila o 6,42%

| Výstup | Hodnota |
|----------------------------------|-------------|
| Počet príchodov cestujúcich | 11528 |
| Počet prevezených cestujúcich | 11528 |
| Počet neobslužených cestujúcich | 0 (0,0 %) |
| Celkový čas strávený čakaním | 86013 min |
| Priemerný čas strávený čakaním | 7 min |
| Celkové náklady | 40633 Kč |
| Priemerná spokojnosť cestujúcich | 98,54 % |
| Celkový počet vozidiel | 72 |
| Priemerná naplnenosť vozidiel | 39 (24,6 %) |

Tabuľka 5.4: Výstupy simulácie

Celý rozpis zastávok, časový rozpis a grafy podrobnejšie popisujúce tento experiment sú uvedené v prílohe C.

5.3 Experiment s dlhou trasou a krátkym vozidlom

Cieľom tohto experimentu je zistiť, ako sa algoritmus vyrovnáva s dlhou trasou a krátkym vozidlom. Ako referencia bola zvolená okružná linka 84 v Brne, ktorej trasa pozostáva zo 44 zastávok. Parametry experimentu sú uvedené v tabuľke 5.5.

| Vstup | Hodnota |
|------------------------|------------------------|
| Kapacita vozidla | 80 |
| Počet miest na sedenie | 30 |
| Celkové náklady | 92,82 Kč/100 miesto-km |
| Dĺžka trasy | 23,0 km |

Tabuľka 5.5: Parametre vozidla a trasy

Z tabuľky 5.6 je vidieť, že algoritmus bol schopný vygenerovať rozpis, pri ktorom je spokojnosť cestujúcich viac ako 98%. Priemerná čakacia doba, počet vozidiel a priemerná naplnenosť vozidiel sú veľmi podobné ako v prípade krátkej trasy. Oproti nej však vzrástol počet cestujúcich a teda aj celkový čas strávený čakaním. Taktiež sa zvýšili celkové náklady na prevádzku.

| Výstup | Hodnota |
|----------------------------------|--------------|
| Počet príchodov cestujúcich | 49041 |
| Počet prevezených cestujúcich | 49041 |
| Počet neobslužených cestujúcich | 0 (0,0 %) |
| Celkový čas strávený čakaním | 253052 min |
| Priemerný čas strávený čakaním | 5 min |
| Celkové náklady | 189576 Kč |
| Priemerná spokojnosť cestujúcich | 98,1 % |
| Celkový počet vozidiel | 111 |
| Priemerná naplnenosť vozidiel | 22 (27,67 %) |

Tabuľka 5.6: Výstupy simulácie

Celý rozpis zastávok, časový rozpis a grafy podrobnejšie popisujúce tento experiment sú uvedené v prílohe D.

5.4 Experiment s dlhou trasou a dlhým vozidlom

Cieľom tohto experimentu je zistiť, ako sa algoritmus vyrovnáva s dlhou trasou a dlhým vozidlom. Ako referencia bola zvolená okružná linka 84 v Brne, ktorej trasa pozostáva zo 44 zastávok. Parametry experimentu sú uvedené v tabuľke 5.7.

| Vstup | Hodnota |
|------------------------|------------------------|
| Kapacita vozidla | 160 |
| Počet miest na sedenie | 60 |
| Celkové náklady | 92,82 Kč/100 miesto-km |
| Dĺžka trasy | 23,0 km |

Tabuľka 5.7: Parametre vozidla a trasy

Výstupy experimentu je možné vidieť v tabuľke 5.8. Oproti experimentu s dlhou cestou ale krátkym vozidlom sa zmenili nasledovné parametre:

- Priemerná čakacia doba sa zvýšila o 2 minúty
- Spokojnosť cestujúcich sa zvýšila o 1,88%
- Počet vozidiel sa znížil o 29 kusov
- Priemerná naplnenosť vozidiel sa znížila o 9,04%

| Výstup | Hodnota |
|----------------------------------|--------------|
| Počet príchodov cestujúcich | 48358 |
| Počet prevezených cestujúcich | 48358 |
| Počet neobslužených cestujúcich | 0 (0,0 %) |
| Celkový čas strávený čakaním | 314379 min |
| Priemerný čas strávený čakaním | 7 min |
| Celkové náklady | 280094 Kč |
| Priemerná spokojnosť cestujúcich | 99,98 % |
| Celkový počet vozidiel | 82 |
| Priemerná naplnenosť vozidiel | 30 (18,63 %) |

Tabuľka 5.8: Výstupy simulácie

Celý rozpis zastávok, časový rozpis a grafy podrobnejšie popisujúce tento experiment sú uvedené v prílohe E.

Kapitola 6

Nástroj na analýzu a optimalizáciu časových rozpisov

Poslednou časťou tejto práce je implementácia nástroja na analýzu a optimalizáciu časových rozpisov. Tento nástroj bude slúžiť ako rozhranie medzi používateľom a optimalizačným algoritmom. Nástroj bude mať dve hlavné funkcie:

- **Analýza časového rozpisu** — nástroj prevezme existujúci časový rozpis a vykoná analýzu jeho kvality. Na to poslúži spustenie simulácie, zber štatistík a ich vizualizácia.
- **Optimalizácia časového rozpisu** — nástroj vygeneruje nové časové rozpisy na základe vstupných parametrov a požiadaviek používateľa. Na to poslúži optimalizačný algoritmus NSGA-II.

Užívateľské rozhranie by malo byť intuitívne a jednoduché. Hlavným cieľom tejto práce je optimalizácia, preto je kladený väčší dôraz na ňu ako na samotné užívateľské rozhranie. Toto rozhranie, bez ohľadu na to ako moc jednoduché bude, je dôležité, preto tento krok nesmie byť zanedbaný.

6.1 Návrh užívateľského rozhrania

Prvým krokom pri tvorbe užívateľských rozhraní je návrh. Najprv si treba premyslieť, čo všetko bude nástroj robiť a aké funkcie bude mať. Tento nástroj by mal vedieť analyzovať nahrané časové rozpisy, vizualizovať zistenia zo simulácie, prezentovať ich formou grafov a štatistík. Ďalej by mal vedieť optimalizovať časové rozpisy. Výpočet optimalizácie pomocou genetického algoritmu je časovo náročný úkon a preto by bolo vhodné používateľa informovať o priebehu optimalizácie. Po dokončení optimalizácie by mal mať používateľ možnosť vybrať si rozpis, ktorý sa mu páči najviac a uložiť si ho. Malo by byť možné analyzovať aj uložené rozpisy. Prípadne by nástroj mohol ponúkať exportovanie analýzy rozpisu.

Všetky tieto funkcie sa dajú rozdeliť medzi dve stránky, stránku pre analýzu a stránku pre optimalizáciu. Na prepínanie medzi stránkami poslúži jednoduché menu v hornej časti okna. Na ich návrh boli použité wireframy.

Stránka pre analýzu časového rozpisu

Na stránke analýzy časového rozpisu bude možné:

- nahrať informácie o zastávkach
- nahrať informácie o časovom rozpise
- nahrať uložený časový rozpis
- zadať parametre pre simuláciu
- Resetovať stránku
- spustiť simuláciu
- zobrazíť výsledky simulácie, vrátane grafov a štatistík
- uložiť výsledky analýzy

The wireframe shows a web interface for simulation analysis. At the top, there's a header with 'Názov aplikácie' on the left and two buttons, 'Analýza' and 'Optimizácia', on the right. Below the header, the main area is divided into several sections. On the left, there's a large dashed box labeled 'Nahratie súboru s informáciami o zastávkach'. To its right, another dashed box is labeled 'Nahratie súboru s informáciami o časovom rozpise'. Below these, there's a button 'Vybrať uložený rozpis'. Further down, there are four input fields: 'Kapacita vozidla', 'Miest na sedenie', 'Náklady na 100 miesto-km', and 'Dĺžka trasy'. Below these fields are two buttons: 'Resetovať' and 'Analyzovať'. At the bottom, there's a large dashed box labeled 'Výsledky analýzy' with the subtitle 'Jedno číselné výstupy' and 'grafy'. Below this box is a button 'Uložiť analýzu'.

Obr. 6.1: Wireframe pre analýzu časového rozpisu

Na obrázku 6.1 je znázornený wireframe pre túto stránku. Funkcie spomenuté vyššie sú usporiadané rovnako ako jednotlivé elementy stránky. Pre nahranie informácií o zastávke a novom časovom rozpise bude slúžiť vyberač súborov. Informácie by mali byť uložené v textovom súbore. Užívateľské rozhranie by malo obsahovať nápovedu, ktorá používateľovi ukáže aký formát majú mať vstupné súbory.

Formát potrebný na spracovanie informácií o zastávkach:

Lesná, Haškova:0:[(5, 60), (6, 90), ..., (21, 40), (22, 30)]:0

Brechtova:1:[(5, 60), (6, 90), ..., (21, 40), (22, 30)]:0.1

Blažkova:2:[(5, 60), (6, 90), ..., (21, 40), (22, 30)]:0.1

...

Formát potrebný na spracovanie informácií o časovom rozpise:

04:

05:01,11,21,31,41,51

06:01,08,14,21,28,34,41,48,54

07:01,08,14,21,28,34,41,48,54

08:01,08,18,28,38,48,58

09:08,18,28,38,48,58

...

Výber uloženého rozpisu môže byť realizovaný formou rozbaľovacej ponuky (dropdown). Rozpisy v tejto ponuke by mali byť pomenované, aby bol výber prehľadný. Vstupné parametre **Kapacita vozidla**, **Miest na sedenie**, **Náklady na 100 miesto-km** a **Dĺžka trasy** budú realizované jednoduchými textovými vstupmi, s obmedzením len na numerické znaky, prípadne číselnými obmedzeniami potrebnými pre dané parametre. Tlačítko **Resetovať** vymaže všetky vstupy aj výstupy analýzy. Tlačítko **Analyzovať** spustí simuláciu časového rozpisu a zobrazí výsledky analýzy. Medzi tieto výsledky budú patriť agregované hodnoty:

- Počet príchodov cestujúcich
- Počet prevezených cestujúcich
- Počet neobslužených cestujúcich
- Celkový čas strávený čakaním
- Priemerný čas strávený čakaním
- Celkové náklady
- Priemerná spokojnosť cestujúcich
- Celkový počet vozidiel
- Priemerná naplnenosť vozidiel

Taktiež budú na stránke zobrazené grafy s hodnotami rôznych štatistík za hodinu aby boli vidieť zmeny v priebehu dňa, konkrétne:

- Cestujúci prichádzajúci za hodinu
- Priemerný čas strávený čakaním za hodinu
- Počet prípadov kedy sa cestujúci nezmestili do vozidla za hodinu

Posledným grafom bude:

- Priemerná naplnenosť vozidiel naprieč zastávkami

Po stlačení tlačítka **Uložiť analýzu** sa otvorí vyberač súborov, aby si používateľ mohol vybrať kam sa analýza uloží. Analýza sa bude ukladať vo forme textového súboru.

Stránka pre optimalizáciu časového rozpisu

Na stránke optimalizácie časového rozpisu bude možné:

- nahrať informácie o zastávkach
- zadať parametre pre optimalizáciu
- Resetovať stránku
- spustiť optimalizáciu
- zobrazíť priebežne dosiahnuté ciele aktuálne spracovanej generácie
- vybrať rozpis z generácie
- uložiť vybraný rozpis

The wireframe shows a web interface for time schedule optimization. At the top, there is a header with the text 'Názov aplikácie' on the left and two buttons, 'Analýza' and 'Optimalizácia', on the right. Below the header is a large dashed box containing the text 'Nahratie súboru s informáciami o zastávkach'. Underneath this box are eight input fields arranged in two rows of four. The first row contains 'Veľkosť populácie', 'Počet generácií', 'Pravdepodobnosť mutácie', and 'Maximálny počet spojov za hodinu'. The second row contains 'Kapacita vozidla', 'Miest na sedenie', 'Náklady na 100 miesto-km', and 'Dĺžka trasy'. Below these input fields are two buttons: 'Resetovať' and 'Optimalizovať'. At the bottom of the main content area is another large dashed box containing the text 'Priebežné výsledky optimalizácie' and 'Graf s dosiahnutými cieľmi všetkých jednotníc v aktuálne dosiahnutej generácii'. Below this box is a button labeled 'Uložiť rozpis'.

Obr. 6.2: Wireframe pre optimalizáciu časového rozpisu

Na obrázku 6.2 je znázornený wireframe pre túto stránku. Pre nahranie informácií o zastávke bude slúžiť rovnaký vyberač súborov ako na stránke analýzy. Vyberač pre časový rozpis tu nie je potrebný, pretože z hľadiska optimalizácie je časový rozpis výstupom a nie vstupom. Číselné vstupné parametre pre simuláciu sú rovnaké ako na stránke analýzy, ale pre genetický algoritmus tu sú štyri nové a to **Veľkosť populácie**, **Počet generácií**, **Pravdepodobnosť mutácie** a **Maximálny počet spojov za hodinu**. Tlačítko **Resetovať** vymaže všetky vstupy aj výstupy optimalizácie. Tlačítko **Optimalizovať** spustí optimalizáciu časového rozpisu. Používateľ by mal počas optimalizácie byť schopný vidieť:

- V koľkej generácii sa algoritmus nachádza
- Koľko generácií bude algoritmus spracovávať

- Kedy optimalizácia začala
- Kedy optimalizácia skončila (ak už skončila)
- Ako dlho optimalizácia trvala (ak už skončila)
- Graf s priebežnými dosiahnutými cieľmi jednotlivých jedincov v aktuálne dosiahnutej generácii

Po dokončení optimalizácie by mal byť používateľ schopný prezrieť si všetky rozpisy v poslednej generácii. To bude zabezpečené posuvným tiahlom, a zobrazením aktuálne vybraného rozpisu. Tiahlo bude mať toľko krokov, koľko je rozpisov v generácii. Vyplnením textového vstupu pre názov rozpisu a stlačením tlačítka **Uložiť rozpis** sa rozpis uloží. Nepomenovaný rozpis sa označí časom a dátumom vytvorenia.

6.2 Implementácia nástroja

Po návrhu následuje implementácia. Predtým však treba vybrať technológie, ktoré sú na to vhodné a navrhnuť architektúru aplikácie.

Použité technológie

Ako programovací jazyk bol zvolený **Python** a to preto, lebo je veľmi verzatilný. Výhodnejšie by možno bolo naprogramovať optimalizačný algoritmus v jazyku **C** alebo **C++**, ktoré sú rýchlejšie.

Python bol však vybraný vďaka jednoduchému použitiu a tomu, že sa v ňom dá pohodlne napísať ako backend, tak aj frontend. Na to aby sa mohol v tomto jazyku písať aj frontend bol využitý framework **Reflex** ¹.

Vďaka tomtu frameworku je možné písať webové komponenty a celé stránky priamo v jazyku **Python**. Každý komponent je písaný formou funkcie, ktorá vracia jeho obsah. Tým sa dajú komponenty zhlukovať do väčších komponentov alebo stránok. Vstupy komponentov sa dajú predávať ako parametre jeho funkcie, alebo môže komponent pracovať s premennými, ktoré sú súčasťou triedy **State**. **State** je trieda, ktorá počas behu aplikácie uchováva jej stav. K premenným tohto stavu sa dá jednoducho pristupovať, čítať ich hodnoty aj prepisovať ich. Takýchto tried môže byť v aplikácii viacej a jeden komponent môže pracovať s premennými aj viacerých stavov. Stav okrem premenných obsahuje aj metódy, ktoré sa volajú pri danej udalosti. Medzi udalosti patrí napríklad stlačenie konkrétneho tlačítka, alebo zmena hodnoty v jednom z textových vstupov. V týchto metódach sa môže pristupovať aj k premenným iného stavu.

Takáto verzatilita jazyka a frameworku bola pri tejto práci veľmi výtaná. Treba si však dávať pozor, pretože priveľká sloboda v písaní kódu môže mať za následok neprehľadný, príliš previazaný kód, ktorý sa bude v budúcnosti ťažko rozširovať. Pretože sa s týmito technológiami dá aplikácia napísať naozaj veľmi veľa spôsobmi, je potrebné zvoliť si dobrú architektúru, rozčlenenie aplikácie na moduly, a tie na jednotlivé webové komponenty alebo backendové skripty.

Architektúra aplikácie

¹<https://reflex.dev/>

Kapitola 7

Záver

Literatúra

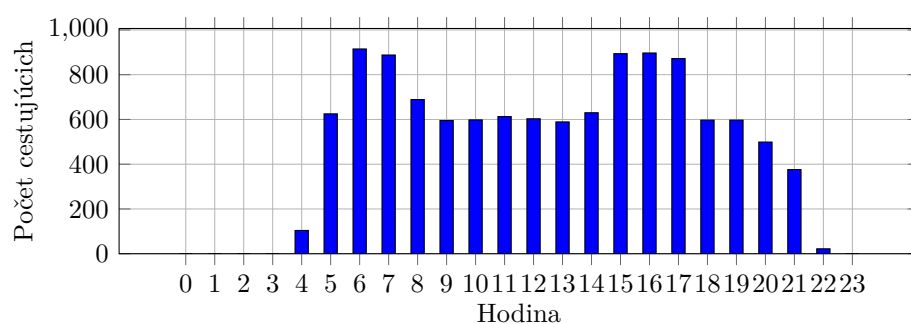
- [1] BALA, A. a SHARMA, A. K. A comparative study of modified crossover operators. In: *2015 Third International Conference on Image Information Processing (ICIIP)*. 2015, s. 281–284.
- [2] DEB, K.; PRATAP, A.; AGARWAL, S. a MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, zv. 6, č. 2, s. 182–197.
- [3] FARIAS, R. G. G. a DE MAGALHÃES, C. S. Parent Selection Strategies in Niching Genetic Algorithms. In: *2018 IEEE Congress on Evolutionary Computation (CEC)*. 2018, s. 1–8.
- [4] IMMANUEL, S. D. a CHAKRABORTY, U. K. Genetic Algorithm: An Approach on Optimization. In: *2019 International Conference on Communication and Electronics Systems (ICCES)*. 2019, s. 701–708.
- [5] NGATCHOU, P.; ZAREI, A. a EL SHARKAWI, A. Pareto Multi Objective Optimization. In: *Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems*. 2005, s. 84–91.

Príloha A

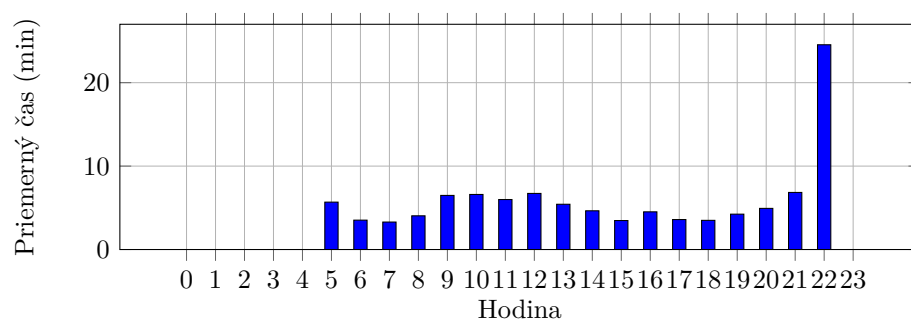
Programová dokumentácia

Príloha B

Experiment s krátkou trasou a krátkym vozidlom



Obr. B.1: Počet cestujúcich prichádzajúcich na zastávku za hodinu



Obr. B.2: Priemerný čas strávený čakaním za hodinu

| Zastávka | # |
|----------------------|----------|
| Lesná, Haškova | 0 |
| Brechtova | 1 |
| Blažkova | 2 |
| Arbesova | 3 |
| Heleny Malířové | 4 |
| Lesná, nádraží | 5 |
| Štefánikova čtvrť | 6 |
| Provozníková | 7 |
| Lesnická | 9 |
| Zemědělská | 10 |
| Černá Pole, Erbenova | 11 |

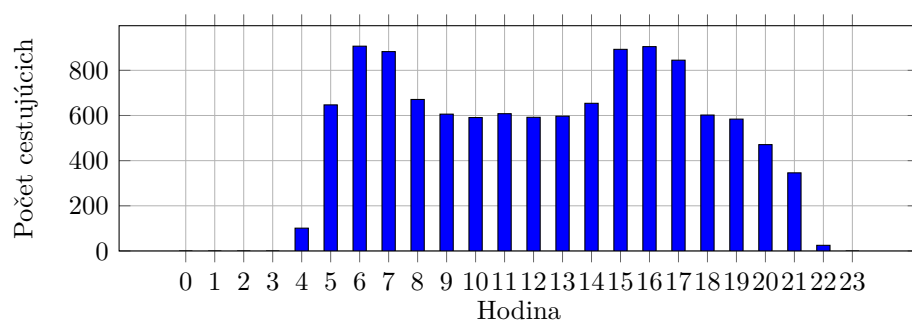
Tabuľka B.1: Rozpis zastávok

| h | Odchody |
|----------|--|
| 05 | 00, 10, 20, 30, 40, 50 |
| 06 | 00, 06, 12, 18, 24, 30, 36, 42, 48, 54 |
| 07 | 00, 06, 13, 20, 26, 33, 40, 46, 53 |
| 08 | 00, 08, 17, 25, 34, 42, 51 |
| 09 | 00, 15, 30, 45 |
| 10 | 00, 12, 24, 36, 48 |
| 11 | 00, 12, 24, 36, 48 |
| 12 | 00, 15, 30, 45 |
| 13 | 00, 08, 17, 25, 34, 42, 51 |
| 14 | 00, 10, 20, 30, 40, 50 |
| 15 | 00, 06, 12, 18, 24, 30, 36, 42, 48, 54 |
| 16 | 00, 10, 20, 30, 40, 50 |
| 17 | 00, 06, 12, 18, 24, 30, 36, 42, 48, 54 |
| 18 | 00, 07, 15, 22, 30, 37, 45, 52 |
| 19 | 00, 08, 17, 25, 34, 42, 51 |
| 20 | 00, 10, 20, 30, 40, 50 |
| 21 | 00, 15, 30, 45 |
| 22 | 00 |

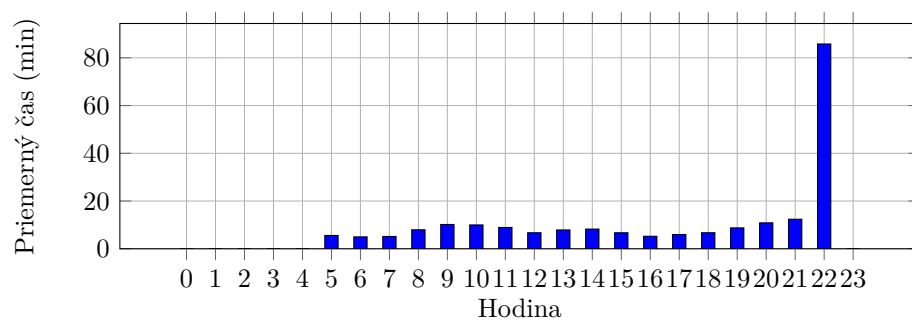
Tabuľka B.2: Časový rozpis

Príloha C

Experiment s krátkou trasou a dlhým vozidlom



Obr. C.1: Počet cestujúcich prichádzajúcich na zastávku za hodinu



Obr. C.2: Priemerný čas strávený čakaním za hodinu

| Zastávka | # |
|----------------------|----------|
| Lesná, Haškova | 0 |
| Brechtova | 1 |
| Blažkova | 2 |
| Arbesova | 3 |
| Heleny Malířové | 4 |
| Lesná, nádraží | 5 |
| Štefánikova čtvrť | 6 |
| Provozníková | 7 |
| Lesnická | 9 |
| Zemědělská | 10 |
| Černá Pole, Erbenova | 11 |

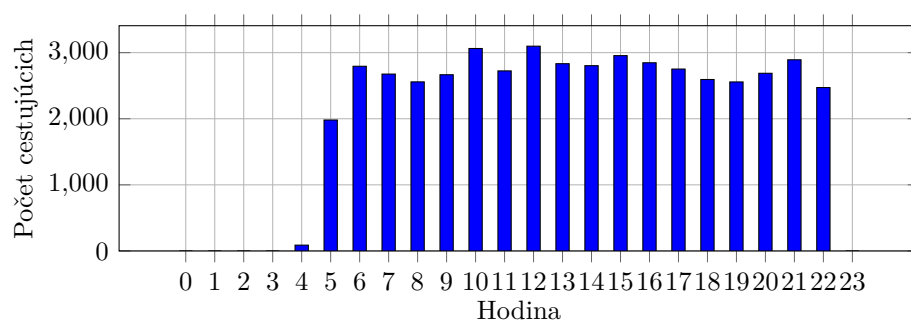
Tabuľka C.1: Rozpis zastávok

| h | Odchody |
|----------|------------------------|
| 05 | 00, 10, 20, 30, 40, 50 |
| 06 | 00, 10, 20, 30, 40, 50 |
| 07 | 00, 10, 20, 30, 40, 50 |
| 08 | 00, 20, 40 |
| 09 | 00, 20, 40 |
| 10 | 00, 20, 40 |
| 11 | 00, 15, 30, 45 |
| 12 | 00, 12, 24, 36, 48 |
| 13 | 00, 20, 40 |
| 14 | 00, 15, 30, 45 |
| 15 | 00, 12, 24, 36, 48 |
| 16 | 00, 10, 20, 30, 40, 50 |
| 17 | 00, 12, 24, 36, 48 |
| 18 | 00, 15, 30, 45 |
| 19 | 00, 20, 40 |
| 20 | 00, 20, 40 |
| 21 | 00, 30 |
| 22 | 00 |

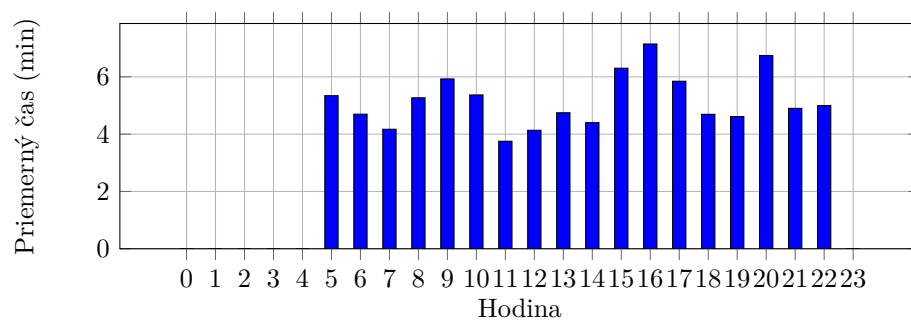
Tabuľka C.2: Časový rozpis

Príloha D

Experiment s dlhou trasou a krátkym vozidlom



Obr. D.1: Počet cestujúcich prichádzajúcich na zastávku za hodinu



Obr. D.2: Priemerný čas strávený čakaním za hodinu

| Zastávka | # |
|-------------------------|----------|
| Židenice, Stará osada | 0 |
| Gajdošova | 2 |
| Otakara Ševčíka | 3 |
| Škroupova | 4 |
| Tržní | 6 |
| Hladíkova | 8 |
| Autobusové nádraží | 10 |
| Opuštěná | 12 |
| Křídlovická | 15 |
| Poříčí | 16 |
| Mendlovo náměstí | 20 |
| Křížkovského | 21 |
| Výstaviště | 22 |
| Riviéra | 24 |
| Pisárky | 26 |
| Anthropos | 28 |
| Pod Jurankou | 29 |
| Veslařská | 30 |
| Jundrov, hřiště | 31 |
| Jundrovský most | 32 |
| Vozovna Komín | 34 |
| Hlavní | 35 |
| Štursova | 36 |
| Rosického náměstí | 37 |
| Přívrat | 39 |
| Záhřebská | 40 |
| Skácelova | 42 |
| Slovanské náměstí | 43 |
| Husitská | 44 |
| Semilasso | 46 |
| Královo Pole, nádraží | 47 |
| Mojmírovo náměstí | 48 |
| Kociánka | 49 |
| Královopolská strojárna | 50 |
| Divišova čtvrť | 51 |
| U Tunýlku | 52 |
| Halasovo náměstí | 54 |
| Poliklinika Lesná | 55 |
| Lesná, nádraží | 56 |
| Štefánikova čtvrť | 57 |
| Merhautova | 59 |
| Tomkovo náměstí | 61 |
| Židenice, kasárna | 63 |
| Židenice, Stará osada | 65 |

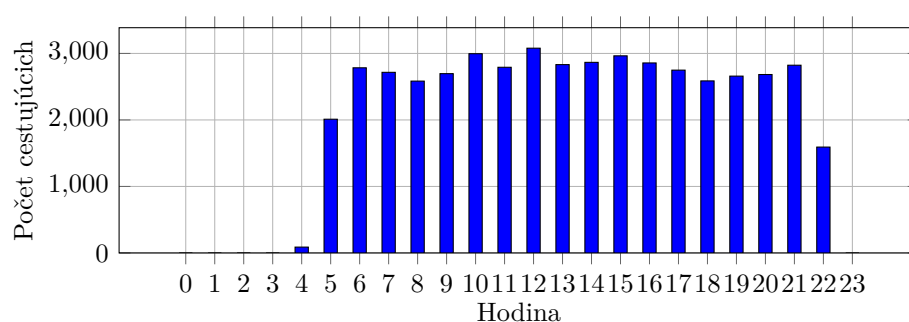
Tabulka D.1: Rozpis zastávek

| h | Odchody |
|----------|--------------------------------|
| 05 | 00, 10, 20, 30, 40, 50 |
| 06 | 00, 07, 15, 22, 30, 37, 45, 52 |
| 07 | 00, 10, 20, 30, 40, 50 |
| 08 | 00, 12, 24, 36, 48 |
| 09 | 00, 12, 24, 36, 48 |
| 10 | 00, 07, 15, 22, 30, 37, 45, 52 |
| 11 | 00, 07, 15, 22, 30, 37, 45, 52 |
| 12 | 00, 10, 20, 30, 40, 50 |
| 13 | 00, 07, 15, 22, 30, 37, 45, 52 |
| 14 | 00, 12, 24, 36, 48 |
| 15 | 00, 15, 30, 45 |
| 16 | 00, 12, 24, 36, 48 |
| 17 | 00, 10, 20, 30, 40, 50 |
| 18 | 00, 07, 15, 22, 30, 37, 45, 52 |
| 19 | 00, 15, 30, 45 |
| 20 | 00, 10, 20, 30, 40, 50 |
| 21 | 00, 08, 17, 25, 34, 42, 51 |
| 22 | 00, 10, 20, 30, 40, 50 |

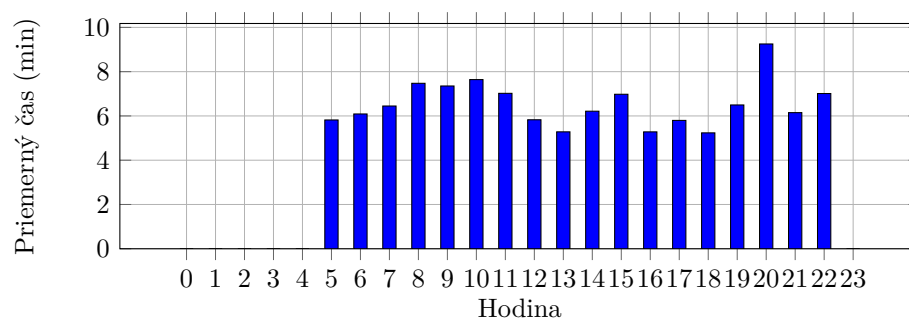
Tabulka D.2: Časový rozpis

Príloha E

Experiment s dlhou trasou a dlhým vozidlom



Obr. E.1: Počet cestujúcich prichádzajúcich na zastávku za hodinu



Obr. E.2: Priemerný čas strávený čakaním za hodinu

| Zastávka | # |
|-------------------------|----------|
| Židenice, Stará osada | 0 |
| Gajdošova | 2 |
| Otakara Ševčíka | 3 |
| Škroupova | 4 |
| Tržní | 6 |
| Hladíkova | 8 |
| Autobusové nádraží | 10 |
| Opuštěná | 12 |
| Křídlovická | 15 |
| Poříčí | 16 |
| Mendlovo náměstí | 20 |
| Křížkovského | 21 |
| Výstaviště | 22 |
| Riviéra | 24 |
| Pisárky | 26 |
| Anthropos | 28 |
| Pod Jurankou | 29 |
| Veslařská | 30 |
| Jundrov, hřiště | 31 |
| Jundrovský most | 32 |
| Vozovna Komín | 34 |
| Hlavní | 35 |
| Štursova | 36 |
| Rosického náměstí | 37 |
| Přívrat | 39 |
| Záhřebská | 40 |
| Skácelova | 42 |
| Slovanské náměstí | 43 |
| Husitská | 44 |
| Semilasso | 46 |
| Královo Pole, nádraží | 47 |
| Mojmírovo náměstí | 48 |
| Kociánka | 49 |
| Královopolská strojárna | 50 |
| Divišova čtvrť | 51 |
| U Tunýlku | 52 |
| Halasovo náměstí | 54 |
| Poliklinika Lesná | 55 |
| Lesná, nádraží | 56 |
| Štefánikova čtvrť | 57 |
| Merhautova | 59 |
| Tomkovo náměstí | 61 |
| Židenice, kasárna | 63 |
| Židenice, Stará osada | 65 |

Tabulka E.1: Rozpis zastávek

| h | Odchody |
|----------|------------------------|
| 05 | 00, 12, 24, 36, 48 |
| 06 | 00, 12, 24, 36, 48 |
| 07 | 00, 15, 30, 45 |
| 08 | 00, 15, 30, 45 |
| 09 | 00, 15, 30, 45 |
| 10 | 00, 15, 30, 45 |
| 11 | 00, 12, 24, 36, 48 |
| 12 | 00, 10, 20, 30, 40, 50 |
| 13 | 00, 12, 24, 36, 48 |
| 14 | 00, 15, 30, 45 |
| 15 | 00, 10, 20, 30, 40, 50 |
| 16 | 00, 12, 24, 36, 48 |
| 17 | 00, 10, 20, 30, 40, 50 |
| 18 | 00, 12, 24, 36, 48 |
| 19 | 00, 20, 40 |
| 20 | 00, 12, 24, 36, 48 |
| 21 | 00, 12, 24, 36, 48 |
| 22 | 00 |

Tabulka E.2: Časový rozpis

Príloha F

Mail od pána Michala Hlaváčka

From: Hlaváček Michal <mhlavacek@dpmb.cz>
Date: 3/25/25 22:10 (GMT+01:00)
To: 'xkaton00' <xkaton00@stud.fit.vutbr.cz>
Cc: Kříž Michael <mkriz@dpmb.cz>
Subject: RE: Bakalárska práca - optimalizácia MHD

Dobrý den,

jednoduchá otázka, složitá odpověď, přesto se pokusím to popsat srozumitelně a jednoduše, sám nejsem ekonom.

Jako nejvhodnější postup pro vyčíslení vícenákladů nebo naopak úspor u změny dopravního řešení či parametrů jednotlivých linek bych doporučil **výpočet přes přepravní výkony**, které jsou základní jednotkou, ve které DPMB "obchoduje" s městem Brnem.

V příloze zasílám platný Finanční model, který je součástí smluvního vztahu mezi Brnem a DPMB, tudíž se jedná o veřejně přístupný dokument.

Je vidět, že část nákladů je variabilních, ale většina je fixních – resp. část variabilních nákladů je zde považována za fixní, což má své důvody. Např. pokud nějakými úsporami ušetřím jeden autobus do výpravy, ušetříme naftu a hodiny řidiče, ale nebude propuštěn mechanik nebo paní na předprodeji jízdních dokladů, ani se tím nezlevní dovoz mýdla na WC na konečné. V případě nějakých opravdu zásadních změn by zřejmě došlo ke změně výše "fixních" nákladů, ale ne tak výrazně, aby bylo nutné se tím zabývat.

Dále je z tabulky možné zjistit celkovou výši kompenzace a ztráty.

Doporučený zjednodušený postup pro ekonomickou kalkulaci navržených změn:

- pro každou kategorii vozu spočítat roční změnu výkonů – vozokilometrů – tj. denní vozové kilometry vynásobit počty provozních dnů
- přepočíst vozokilometry na místokilometry – tj. roční výkony vynásobit obsaditelností
- takto získané přepravní výkony vydělit 100 (údaje ve finančním modelu se vztahují ke 100 mkm)
- celkový objem přepravních výkonů (mkm/100) v jednotlivých typech dopravy vynásobit finančním oceněním jednotky 100 mkm opět pro jednotlivé druhy dopravy – viz příloha, řádek kompenzace (tramvaj 55,59 – trolejbus 72,97 – autobus 66,35 – lodě nás snad nezajímají)
- výsledkem je finanční hodnota výkonů (náklad) za celý kalendářní rok

Nápověda 1:

Počty provozních dnů pro rok 2025: pracovní dny – 188, soboty – 82, neděle – 33, pracovní dny prázdniny – 62

Nápověda 2:

Statistická obsaditelnost pro výpočet mkm

| Kategorie vozidla | Vozidla ve flotile DPMB | Obsaditelnost |
|--|---|---------------|
| Tramvaj – sólo (cca 15 m) | T3, T6, VarioLF | 110 |
| Tramvaj – kloubová (cca 20 m) | K2, Anitra, VarioLF2, EVO 2 | 157 |
| Tramvaj – dlouhá (cca 30 m) | K3, KT8, 13T, 45T | 220 |
| Trolejbus – klasický a velký trolejbus (cca 12/15 m) | 21 Tr, 26 Tr, 32 Tr, SOR TNS 12 | 70 |
| Trolejbus – kloubový trolejbus (cca 18 m) | 27 Tr, 31 Tr | 120 |
| Autobus – minibus (cca 8 m) | Dekstra LF 38, Isuzu NOVO | 35 |
| Autobus – malý autobus (cca 10 m) | SOR ICN 9,5 | 50 |
| Autobus – klasický a velký autobus (cca 12/15 m) | Crossway LE 12M, Crossway 14,5M, Citelis 12M, Urbanway 12M, SOR NBG 12, SOR NS 12 | 70 |
| Autobus – kloubový autobus (cca 18 m) | Urbanway 18M, Solaris Urbino 18 | 110 |

Příklad:

Rozhodnul jsem se nahradit o víkendu 10 km jízdy soupravy 2 x T3, místo které pojede 10 km kloubový autobus.

Výpočet tramvaj: 10 km x 115 dní = 1 150 vozkm = 253 000 mkm (každý vůz soupravy, tj. 1 150 x 220) = při vydělení 100 = 2 530 x 55,59 = roční úspora 140 642,70 Kč

Výpočet autobus: 10 km x 115 dní = 1 150 vozkm = 126 500 mkm (1 150 x 110) = při vydělení 100 = 1 256 x 66,35 = roční navýšení 83 932,75 Kč.

Dosáhli jsme tak roční úspory 56 709,95 Kč.

Jak je patrné, jedná se o značně zjednodušený výpočet, ale domnívám se, že pro potřeby teoretických kalkulací by měl být dostačující.

Samozřejmě, i když jsou peníze vždy až na prvním místě, je třeba při návrzích změn vzít v potaz celou řadu dalších aspektů.

S pozdravem

Michal Hlaváček

Odbor přípravy provozu
Dopravní podnik města Brna, a.s.
p.p. 46, Hlinky 151, 656 46 Brno
pracoviště: Hlinky 151, Brno - Pisárky
Tel.: 543 171 409
E-mail: mhlavacek@dpmb.cz



125 let elektrické tramvaje v Brně
1900–2025



95 let autobusové dopravy v Brně
1930–2025



25 let nočních rozjezdů v Brně
2000–2025

Jezdíme pro vás



From: Kříž Michael <mkriz@dpmb.cz>
Sent: Thursday, March 13, 2025 2:10 PM
To: 'xkaton00' <xkaton00@stud.fit.vutbr.cz>
Cc: Hlaváček Michal <mhlavacek@dpmb.cz>
Subject: RE: Bakalárska práca - optimalizácia MHD

Dobré odpoledne,
veřejná hromadná doprava je už ze svého principu vždy ztrátová a je nutné její provoz dotovat z veřejných zdrojů.

Co se ale týče podrobnějšího ekonomického vyhodnocení, nemám sám takový přehled, abych si troufl prezentovat nějaká konkrétní čísla.

Více k tomu ví kolega Michal Hlaváček, kterého přidávám do kopie a který se Vám ozve.

Děkuji za spolupráci a přeji pěkný den.

Ing. Michael Kříž

Analytik – administrátor dopravních aplikací
Oddělení dopravních informací
Odbor přípravy provozu

Dopravní podnik města Brna, a.s.
Hlinky 151, 603 00 Brno
pracoviště: Novobranská 18, 602 00 Brno

e-mail: mkriz@dpmb.cz
telefon: 5 4317 4261





From: xkaton00 <xkaton00@stud.fit.vutbr.cz>**Sent:** Friday, March 7, 2025 1:23 PM**To:** Kříž Michael <mkriz@dpmb.cz>**Subject:** Bakalárska práca - optimalizácia MHD

Dobrý deň,

Píšem ohľadom mojej bakalár práce na tému optimalizácia MHD. Pred nedávnom sme sa osobne stretli a veľmi mi to pomohlo. Chcel by som Vás požiadať o ďalšie stretnutie. Ak by ste mali čas niekedy v blízkej budúcnosti, rád by som Vám ukázal doterajšie výsledky a opýtal sa na pár nových informácií. V rámci optimalizácie mi vedúci práce odporučil ohodnocovať jednotlivé rozpisy liniek z ekonomického pohľadu. Momentálne to mám spravené len ako taký abstraktný bodový systém pretože si vôbec neviem predstaviť aké všetky náklady sú spojené s prevádzkou jednej linky a aký je priemerný zárobok za jedného cestujúceho. Tieto informácie potrebujem aby som vedel ohodnotiť či je daný vygenerovaný rozpis stratový alebo ziskový. Chápem že tieto presné informácie môžu byť dôverné, ale minimálne by som potreboval sa o tejto téme porozprávať aby som mal aspoň v akých jednotkách sa tieto čísla približne pohybujú. Samozrejme, čím bližšie budú získané informácie k realite, tým kvalitnejší by mal byť výsledok mojej práce.

Za skorú odpoveď vopred ďakujem.
S pozdravom, Lukáš Katona.

Sent from my Galaxy

Príloha G

Finančný model DPMB

Príloha č. 6 Smlouvy

Finanční model nákladů a výnosů pro rok 2025

| Typ | Položka | číslo řádku | ED | | TD | | AD | | Lodě | |
|---------------|---------------------------|-------------|-----------|----------------|-----------|----------------|-----------|----------------|-----------|----------------|
| | | | v tis. Kč | Kč/100 míst.km | v tis. Kč | Kč/100 míst.km | v tis. Kč | Kč/100 míst.km | v tis. Kč | Kč/100 míst.km |
| výnosy | Tržební energie a palivo | 1 | 204 508 | 8,50 | 43 627 | 7,73 | 269 594 | 19,15 | 1 368 | 21,39 |
| | | 2 | 0 | 0,00 | 3 697 | 0,63 | 16 795 | 1,19 | 0 | 0,00 |
| fixní | Kola, pneu | 3 | 96 423 | 4,01 | 16 723 | 2,85 | 59 161 | 4,20 | 915 | 14,31 |
| | Matériál | 4 | 9 260 | 0,38 | 3 856 | 0,66 | 5 089 | 0,36 | 2,72 | 12,89 |
| | Vetrační energie a palivo | 5 | 11 520 | 0,48 | 1 200 | 0,20 | 4 282 | 0,30 | 831 | 12,89 |
| | Externí opravy | 6 | 16 890 | 0,70 | 4 869 | 0,83 | 12 285 | 0,87 | 962 | 15,05 |
| | Externí služby | 7 | 339 624 | 14,07 | 76 072 | 12,98 | 367 198 | 26,08 | 4 348 | 69,00 |
| x | Celkem (1+2+3+4+5+6+7) | 8 | 434 486 | 18,06 | 195 094 | 33,30 | 481 804 | 34,21 | 11 342 | 177,36 |
| | | 9 | 146 631 | 6,18 | 66 742 | 11,38 | 164 825 | 11,70 | 3 880 | 60,67 |
| fixní | Osobní náklady | 10 | 585 096 | 24,24 | 261 836 | 44,69 | 646 629 | 45,92 | 15 221 | 238,04 |
| | | 11 | 0 | 0,00 | 9 | 0,02 | 11 | 0,00 | 0 | 0,02 |
| fixní | Celkové náklady | 12 | 319 471 | 13,28 | 80 176 | 13,68 | 91 543 | 6,50 | 9 825 | 163,64 |
| | | 13 | 0 | 0,00 | 0 | 0,00 | 0 | 0,00 | 0 | 0,00 |
| | | 14 | 0 | 0,00 | 0 | 0,00 | 16 | 0,00 | 2 | 0,03 |
| | | 15 | 9 893 | 0,42 | 3 694 | 0,61 | 12 734 | 0,90 | 247 | 3,93 |
| | | 16 | 28 568 | 1,06 | 9 543 | 1,58 | 21 593 | 1,53 | 469 | 7,36 |
| | | 17 | 465 668 | 20,23 | 78 037 | 13,32 | 29 144 | 17,9 | 2 059 | 32,76 |
| | | 18 | 104 656 | 4,34 | 33 462 | 6,05 | 72 023 | 5,11 | 3 971 | 62,20 |
| | | 19 | 112 091 | 4,66 | 24 413 | 4,17 | 70 111 | 4,96 | 344 | 5,38 |
| | | 20 | 1 980 007 | 82,30 | 568 945 | 97,11 | 1 307 004 | 92,82 | 32 358 | 505,98 |
| | | 21 | 546 810 | 22,73 | 121 996 | 20,82 | 331 194 | 23,52 | 12 128 | 189,64 |
| | | 22 | 38 090 | 1,50 | 8 052 | 1,37 | 21 859 | 1,55 | 1 159 | 18,12 |
| | | 23 | 69 407 | 2,88 | 14 208 | 2,43 | 26 147 | 1,86 | 0 | 0,00 |
| Výnosy | Tržby z jízdního | 24 | 652 307 | 27,11 | 144 256 | 24,62 | 379 200 | 26,53 | 13 286 | 207,76 |
| | | 25 | 1 337 600 | 55,59 | 427 534 | 72,97 | 934 336 | 66,35 | 19 233 | 300,75 |
| | | 26 | x | x | x | x | x | x | x | x |
| | | 27 | x | x | x | x | x | x | x | x |
| | | 28 | x | x | x | x | x | x | x | x |
| Ztrata ze ZVS | Ztrata ze ZVS | 29 | 1 327 700 | 55,18 | 424 690 | 72,49 | 927 803 | 65,59 | 19 071 | 298,22 |
| | | 30 | 2 405 981 | 99,58 | 593 822 | 1 408 165 | 6 356 | 0,10 | 0 | 0,00 |

* Procentní výše výnosů z výnosů v Kč a náklady na jednotku výkonu jsou zobrazeny pro nákladovou výšku, která umožňuje se zúčtovat náklady na 2 desetinná místa, přičemž výše výnosů v tis. Kč

Celkem ztráta
Celkem kompenzace
Čistý příjem
Místové km (v tis. km)

2 699 264
2 718 705
19 442
4 406 423