

# **Modelování svozových úloh s dopravními omezeními**

Dominik Závada

---

Bakalářská práce  
2021



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav informatiky a umělé inteligence

Akademický rok: 2020/2021

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE (projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Dominik Závada  
Osobní číslo: A18087  
Studijní program: B3902 Inženýrská informatika  
Studijní obor: Softwarové inženýrství  
Forma studia: Prezenční  
Téma práce: Modelování svozových úloh s dopravními omezeními  
Téma práce anglicky: Modelling Waste Collection Problems with Traffic Constraints

### Zásady pro vypracování

1. Seznamte se s pojmy z oblasti matematické optimalizace, zejména se zaměřením na odpadové hospodářství a modely svozu odpadu.
2. Zpracujte rešerši existujících modelů svozu odpadu s ohledem na problematiku modelování (časových) zdržení v dopravních sítích.
3. Sestavte jednoduchý matematický model pro plánování svozu, naimplementujte jej ve zvoleném optimalizačním softwaru a výpočetně otestujte na zvolené úloze.
4. Proveďte důkladnou diskuzi dosažených výsledků. Popište omezení a limity zvoleného přístupu a vytyčte směry pro případný další výzkum.



Forma zpracování bakalářské práce: **Tištěná/elektronická**

**Seznam doporučené literatury:**

1. GHIANI, Gianpaolo, Gilbert LAPORTE a Roberto MUSMANNO. Introduction to logistics systems planning and control. Hoboken, NJ, USA: J. Wiley, c2004. ISBN 0-470-84917-7.
2. TOTH., Paolo a Daniele VIGO. Vehicle Routing, Problems, Methods, and Applications. Second edition. SIAM, 2014. ISBN: 978-1-611973-58-7.
3. BEN TICHA, Hamza, Nabil ABSI, Dominique FEILLET a Alain QUILLIOT. Vehicle routing problems with road-network information: State of the art. Networks. 2018, č. 72, s. 393-406. ISSN 0028-3045.
4. PIRES, Ana, Graça MARTINHO, Susana RODRIGUES a Maria Isabel GOMES. Sustainable Solid Waste Collection and Management. Springer, 2019. ISBN: 978-3-319-93199-9.s.
5. SCHRUIJVER, Alexander. Theory of linear and integer programming. Chichester: Wiley, c1986. ISBN 978-0-471-98232-6.

Vedoucí bakalářské práce: **Ing. Dušan Hrabec, PhD.**  
Ústav matematiky

Datum zadání bakalářské práce: **15. ledna 2021**  
Termín odevzdání bakalářské práce: **17. května 2021**

**doc. Mgr. Milan Adámek, Ph.D. v.r.**  
děkan



**prof. Mgr. Roman Jašek, Ph.D. v.r.**  
ředitel ústavu

Ve Zlíně dne 15. ledna 2021

**Prohlašuji, že**

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnaní případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Dominik Závada, v. r.  
podpis studenta

## **ABSTRAKT**

Tato bakalářská práce se zabývá modelováním svozových úloh s dopravními omezeními. Nejprve jsou popsány pojmy jako optimalizace v odpadovém hospodářství, matematická optimalizace, teorie výpočetní složitosti a vehicle routing problem. Sestrojený matematický model se řadí do kategorie úloh značených v anglické terminologii názvem time-dependent vehicle routing problem, patřící do třídy složitosti NP – Úplné. Výstupem práce je implementace matematického modelu v jazyce GAMS, který řeší optimalizaci pomocí smíšeného celočíselného programování. Závěrem jsou demonstrovány výsledky výpočtů v podobě přiložených grafů a map a jsou nastíněny možné cesty pro pozdější implementaci reálnými daty.

Klíčová slova: odpadové hospodářství, svoz odpadu, matematická optimalizace, smíšené celočíselné programování, MIP, teorie výpočetní složitosti, NP – Úplné, time-dependent vehicle routing problem, VRP, GAMS, dopravní omezení

## **ABSTRACT**

This thesis deals with modelling waste collection problems with traffic constraints. First, concepts of optimization in waste management, mathematical programming, computational complexity theory, and vehicle routing problem are described. The developed mathematical model belongs to the category of time-dependent vehicle routing problems, a family of NP – complete problems. Next, mixed integer programming implementation in GAMS language is done. In conclusion, the results are demonstrated in attached graphs and maps and possibilities of real data usage are outlined.

Keywords: waste management, waste collection, mathematical programming, mixed integer programming, MIP, computational complexity theory, NP – complete, time-dependent vehicle routing problem, VRP, GAMS, traffic delay

Tímto bych rád poděkoval svému vedoucímu práce Ing. Dušanovi Hrabcovi, Ph.D. za odborné vedení a konzultace, cenné rady, připomínky a čas, který mi po celou dobu věnoval. Také bych chtěl poděkovat Ing. Vlastimírovi Nevrlému, Ph.D. za konzultace, ochotu a předání zkušeností, které mi velmi pomohly při vypracování této práce. Nakonec bych chtěl poděkovat rodině, přátelům a všem, kteří mě po celou dobu studia podporovali.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 ODPADOVÉ HOSPODÁŘSTVÍ</b> .....	<b>11</b>
1.1 OPTIMALIZACE A LOGISTIKA SVOZU ODPADU .....	11
<b>2 MATEMATICKÁ OPTIMALIZACE</b> .....	<b>12</b>
2.1 ZÁKLADNÍ TŘÍDY MODELŮ .....	12
2.1.1 Lineární programování.....	12
2.1.2 Smíšené celočíselné programování .....	13
2.1.3 Nelineární programování .....	13
<b>3 TEORIE VÝPOČETNÍ SLOŽITOSTI</b> .....	<b>14</b>
3.1 TŘÍDY SLOŽITOSTI .....	14
3.1.3 Prostorová složitost .....	15
3.1.4 Třída P .....	15
3.1.5 Třída NP .....	15
3.1.6 Třída NP – Úplné .....	16
<b>4 VEHICLE ROUTING PROBLEM</b> .....	<b>18</b>
4.1 TYPY FORMULACÍ VRP .....	19
4.1.1 Vehicle flow formulations.....	19
4.1.2 Commodity flow formulations .....	20
4.1.3 Set partitioning problem.....	20
4.2 DRUHY VRP .....	20
4.2.1 Time-dependent vehicle routing problem .....	20
4.2.2 Dynamic vehicle routing problem.....	22
<b>5 REŠERŠE EXISTUJÍCÍCH MODELŮ SVOZU ODPADU</b> .....	<b>23</b>
<b>II PRAKTICKÁ ČÁST</b> .....	<b>24</b>
<b>6 MATEMATICKÝ MODEL SVOZU ODPADU</b> .....	<b>25</b>
6.2 POUŽITÁ DATA .....	25
6.3.1 Množiny .....	28
6.3.2 Proměnné.....	28
6.3.3 Binární proměnné.....	28
6.3.4 Parametry .....	29
6.3.6 Účelová funkce.....	30
6.3.7 Rovnice a omezení modelu .....	31
6.3.8 Omezení a podmínky pro přidání kapacity .....	33
6.3.9 Definice proměnných .....	34
6.4 ROZDÍL OPROTI OBECNÉMU MODELU.....	35
<b>7 IMPLEMENTACE MODELU DO SOFTWARE GAMS</b> .....	<b>36</b>
7.1 POUŽITÝ OPTIMALIZAČNÍ SOFTWARE .....	36

7.1.1	CPLEX.....	37
7.2	IMPLEMENTACE MODELU .....	37
7.2.1	Prvotní nastavení modelu.....	37
7.2.2	Zápis a vkládání dat do parametrů .....	38
7.2.3	Zápis množin.....	39
7.2.4	Zápis matic .....	41
7.2.5	Zápis proměnných.....	42
7.2.6	Zápis účelové funkce a omezení .....	43
7.2.7	Omezení pro výpočet .....	44
7.2.8	Nastavení modelu.....	45
7.3	PROBLÉMY PŘI IMPLEMENTACI A JEJICH ŘEŠENÍ .....	46
<b>8</b>	<b>VÝSLEDKY.....</b>	<b>47</b>
8.1	MĚŘENÍ PRO 1 VOZIDLO .....	47
8.2	MĚŘENÍ PRO 2 VOZIDLA .....	52
8.3	NEDOSTATKY ZVOLENÉHO PŘÍSTUPU .....	56
8.4	SMĚRY PRO BUDOUCÍ VÝZKUM.....	56
	<b>ZÁVĚR .....</b>	<b>57</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>58</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>61</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>62</b>
	<b>SEZNAM TABULEK.....</b>	<b>63</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>64</b>



## ÚVOD

Svoz odpadu každodenně slouží široké veřejnosti a je součástí života všech občanů. Při této činnosti může ovšem docházet k mnoha problémům a komplikacím, kterým se dá při použití správných metod částečně předejít. Jedním z takových problémů je zhoršující se dopravní situace ve městech. Ta vzniká v různých denních intervalech, které se neustále periodicky opakují a stále častěji tak dochází k dopravním omezením díky houstnoucí dopravě. Tyto dopravní špičky způsobují velká časová zdržení a dochází ke zvyšování celkových nákladů pro svozové firmy, jelikož vozidlo zde stojí a nevykonává žádnou činnost.

Příkladem řešení je optimalizace svozu odpadu pomocí vhodného modelu a jeho implementace do zvoleného optimalizačního softwaru. Zvolený přístup zohledňuje časová zdržení v dopravní infrastruktuře města a modeluje optimální trasu pro celý vozový park, aby docházelo k minimálnímu zdržení během vykonávání služeb.

Zvolený přístup je detailně rozebrán v první části bakalářské práce neboli části teoretické, která je rozdělena do pěti hlavních kapitol. První kapitola popisuje optimalizaci v odpadovém hospodářství za účelem dosažení minimalizaci nákladů. Druhá kapitola se zaměřuje na matematickou optimalizaci, kde jsou rozebrány možné druhy programování používané při řešení kombinatorických úloh. Další významnou kategorií je teorie výpočetní složitosti se zaměřením na třídy složitosti. Čtvrtá kapitola pojednává o VRP (vehicle routing problem), patřící do třídy složitosti NP – Úplných. Poslední částí teoretické práce je rozbor dostupných vědeckých publikací na danou problematiku, ze které byla následně vyhotovena rešerše.

Praktická část práce nejprve popisuje účelovou funkci a omezení sestaveného matematického modelu. V druhé kapitole je detailně prezentována implementace modelu do softwaru GAMS. Model byl následně otestován za použití dat získaných pomocí Google Maps. V poslední části je provedena důkladná diskuse dosažených výsledků.

Možný přínos práce je minimalizovat čas svozu odpadu. Díky minimalizaci času trasy svozového vozidla se sníží jak náklady na cestu, tak i spotřeba paliva a s tím související snížení vyprodukovaných emisí za cestu. Pro možnou pozdější aplikaci do reálné dopravní sítě jsou vytyčeny směry sběru dat a případná vylepšení modelu na konci práce.

## **I. TEORETICKÁ ČÁST**

## 1 ODPADOVÉ HOSPODÁŘSTVÍ

Do odpadového hospodářství náleží mnoho odvětví, které mají různá zaměření. Je zde obsažen celý proces od sběru a přepravy, až po likvidaci nebo následné zpracování odpadu. Jeho prioritou je předcházení možného vzniku, avšak zabývá se i následnou recyklací. Patří mezi základní služby každé společnosti a důležité odvětví je sběr tuhého odpadu, u kterého je kladen velký důraz na logistiku a plánování svozových systému, aby byla zajištěna nejvyšší možná efektivita.

### 1.1 Optimalizace a logistika svozu odpadu

Sběr opadu je jedno z odvětví patřící do odpadového hospodářství. Jedná se o přesun určeného materiálu do sběrného místa pro pozdější využití nebo na odpadní skládky. Rychlé a neustálé centralizování obyvatelstva do měst vede k dramatickému zvyšování produkce odpadu. To má za následek zvyšování počtu umístěných sběrných nádob a zvyšování počtu tras určené k jejich vyvážení [1].

Důležitým úkolem se stává zajistit pravidelný a efektivní svoz. Při nízkém počtu svozů se odpadní nádoby nebo kontejnery přeplní a může nastat problém odhazování odpadků mimo nádoby nebo v nejzastříhanějších případech kontaminace půdy. Naopak při příliš častém svozu vzrůstají náklady a stává se příliš neefektivní a zbytečně nákladný. Kvůli velké spotřebě paliva sběr tuhého odpadu je nejvíce znečišťující a nákladná složka odpadového hospodářství. Představuje 50 % až 75 % celkových peněžních výdajů na sběr [2]. Vědní obory zabývající se touto problematikou jsou z hlavní části matematická optimalizace a logistika. Ta se zabývá plánováním, organizací a pohybem určeného materiálu při správné optimalizaci [3].

Jedním z řešení problému je optimalizace svozových tras, vzhledem k tomu, že stávající systémy vyžadují velké náklady na přepravu [1]. Největším omezením pro efektivní svoz ve městské infrastruktuře je samotná doprava. Ta způsobuje kolísání rychlosti a mnohdy zapříčiňuje stání v kolonách. Tato bakalářská práce navrhuje možný způsob, který zvolí optimální trasu podle aktuální dopravní situace. To vede ke snížení celkového času přepravy materiálu, což má za následek snížení nákladů a emisí vyprodukované vozidlem do ovzduší.

## 2 MATEMATICKÁ OPTIMALIZACE

Matematická optimalizace nebo také matematické programování je oblast matematiky, zaměřená na řešení složitých problémů. Využívá se v mnoha odvětvích, jako je logistika, nákladní doprava nebo odpadové hospodářství [1,2]. Metoda řešení pro optimalizační problémy je algoritmus, který do určité přesnosti počítá řešení problému. Účinnost těchto algoritmů se značně liší od úlohy a závisí na mnoha faktorech, jako jsou omezení funkcí nebo použité proměnné. Obecně se dá říct, že optimalizace je snaha najít extrémy (minima nebo maxima) účelové funkce na množině. Základní optimalizační úlohu můžeme formulovat následovně [4,5].

$$\begin{aligned} \min \quad & f_0(x) \\ \text{z. p.} \quad & f_i(x) \leq 0; \quad i = 1, \dots, m \\ & h_i(x) = 0; \quad i = 1, \dots, p \end{aligned}$$

Přičemž  $f_0(x)$  se nazývá účelová funkce. Ta minimalizuje hodnoty  $x$ , které splňují daná omezení  $f_i(x) \leq 0; \quad i = 1, \dots, m$  a  $h_i(x) = 0; \quad i = 1, \dots, p$ . Nalezený prvek  $x$  je poté označován jako optimální řešení a náleží do množiny  $x$ ,  $x \in \mathbb{R}^n$ . Pokud je množina proveditelných řešení prázdná, není známo řešení [5].  $f_i(x) \leq 0$  značí omezení nerovnosti a poslední  $h_i(x) = 0$  se označuje jako omezení rovnosti. Společně obě omezující funkce zobrazují  $\mathbb{R}^n \rightarrow \mathbb{R}$ . Za předpokladu, že nejsou definována žádná omezení  $m = p = 0$ . Pak říkáme, že problém je neomezený [4].

### 2.1 Základní třídy modelů

Obecně rozdělujeme optimalizační problémy na typy nebo třídy, které jsou charakterizované konkrétním cílem a omezeními. Ty poté definují vztahy mezi množinami a proměnnými [4]. Mezi základní druhy modelů patří lineární programování (LP), smíšené celočíselné programování (MIP, převzaté od slovního spojení z anglické terminologie mixed integer programming) a nelineární programování (NLP).

#### 2.1.1 Lineární programování

Řadí se mezi základní optimalizační problémy. Zabývá se problémem optimalizace lineární objektové funkce s lineární rovností či nerovností. Běžné využití má při hledání maximalizace zisku nebo minimalizace nákladů či hledání nejnižší ceny. Využívá se u formulování velké řady matematických problémů a dlouhodobě přispívá při řešení

alokačních a ekonomických problémů [6]. Zároveň se využívá při reálné aplikaci v dopravě, plánování výroby a je stavebním kamenem kombinatorické optimalizace. Lineární úlohu je možné formulovat následovně [4].

$$\begin{aligned} \min \quad & c^T x \\ \text{z. p.} \quad & a_i^T x \leq b_i; \quad i = 1, \dots, m \end{aligned}$$

Kde vektory  $c \in \mathbb{R}^n$  a  $a_1, a_2, \dots, a_m \in \mathbb{R}^n$  a skaláry  $b_1, b_2, \dots, b_m \in \mathbb{R}$ .

### 2.1.2 Smíšené celočíselné programování

U tohoto typu problému je na některé z rozhodovacích proměnných kladeny omezení celočíselnosti, jinak se rovnice s porovnáním s LP nemění. Pokud model používá pouze spojité proměnné, jedná se o LP. Při kombinaci spojitých a celých čísel problém náleží do třídy MIP. Použití celočíselných proměnných výrazně rozšiřuje rozsah použití při optimalizačních problémech, které lze definovat a vyřešit, například formulace kombinatorických úloh [2].

Nově přidaným faktorem oproti lineárnímu programování je přidání rozhodovací proměnné. Ta může dosahovat pouze hodnot 0 a 1 a označuje se jako binární proměnná. Používá se při modelování, kde je zapotřebí rozhodnutí ano/ne. Ovšem tyto proměnné způsobují, že daný problém není konvexní a narůstá výpočetní složitost. Pomocí MIP lze formulovat problémy jako je úloha obchodního cestujícího nebo vehicle routing problem (VRP).

### 2.1.3 Nelineární programování

Úloha do nelineárního programování patří v případě, když účelová funkce nebo některé definované omezení, při hledání maxima nebo minima, nejsou lineární. Neexistují žádné účinné metody pro řešení problémů s nelineárním programováním. U nelineárních problémů, které na první pohled vypadají jednoduše pouze s pár proměnnými mohou často být extrémně časově a výkonově náročné, zatímco problémy pro několik set proměnných až neřešitelné [4].

### 3 TEORIE VÝPOČETNÍ SLOŽITOSTI

Teorie složitosti má silný vliv v oblastech informatiky, matematiky a operačního výzkumu již od 60. let minulého století. Zabývá se výpočetními úlohami, kde zkoumá náročnost při výpočtu na použité zdroje (nejčastěji paměť a čas) a řešitelnost algoritmů. Tedy výpočetní složitost je obecné studium toho, čeho lze dosáhnout v omezeném čase. To lze také definovat jako otázku, zda kontrola správnosti řešení je těžší než hledání samotného řešení [7]. Nejslavnější otázkou je rovnost tříd P a NP, formulována od začátku 70. let. Úlohy se v oblasti teorie složitosti označují pod pojmem problémy. Nejznámější problémy jsou úloha obchodního cestujícího nebo vehicle routing problem. Výpočetní problém definujeme jako úkol řešený pomocí Turingova stroje použitím aplikací postupných kroků neboli algoritmu. Počítaný problém je považován za obtížný, pokud k dosažení jeho řešení jsou vyžadovány značné zdroje bez ohledu na použitý algoritmus [8].

Postupem času výzkum v oblasti výpočetní techniky expandoval exponenciální rychlostí a díky sofistikovanějším algoritmům a výpočetní technice se objevovali nové problémy. To vedlo k vytvoření hierarchie složitosti a zavedlo se rozdělení do jednotlivých klasifikačních tříd.

#### 3.1 Třídy složitosti

Jedná se o způsob organizace podobných typů problémů. Třída je sada problémů, jejichž řešení vyžaduje podobný rozsah času a prostoru [7]. Studium vztahů mezi třídami je hlavní oblastí výzkumu v oblasti teoretické informatiky. Existuje velké množství tříd, ale těmi nejdůležitějšími jsou třídy P a NP. Zde také platí otázka rovnosti zmiňovaných tříd, která se řadí mezi jednu ze sedmi problémů tisíciletí [9]. Na prokázání rovnosti vypsál dne 24. května 2000 Clayův matematický institut ve státě New Hampshire odměnu 1 000 000 amerických dolarů.

##### 3.1.1 Asymptotická složitost

Používá se ke klasifikaci složitosti algoritmů v závislosti na počtu vstupních dat. Jedním ze způsobů zápisu se používá Landauův symbol nebo také notace Omikronu, značící "velké O". Ten znázorňuje horní mez velikosti počtu kroků, které musíme vykonat na nalezení řešení. Takzvaně uvádí vždy nejhorší možný případ. Zápis je  $O(n)$ , kde číslo  $n$  znázorňuje velikost vstupních dat [10]. Ukázka vybraných tříd složitosti jsou znázorněny v Tabulce 1.

Tabulka 1 Příklad časové složitosti algoritmů

Notace	Název	$n = 10$	$n = 30$
$O(1)$	konstantní	1	1
$O(\log n)$	logaritmická	1	1,47
$O(n^2)$	kvadratická	100	900
$O(n^3)$	kubická	1000	27000
$O(n!)$	faktoriálová	3628800	2,65252859812E32

### 3.1.2 Časová složitost

Používá se u popisu počtu kroků, které jsou potřeba na vyřešení problému. Jednotky času jako minuty a sekundy se v teorii složitosti nepoužívají, protože jsou přímo úměrné na typ zvoleného výpočetního přístroje. Kroky jsou definovány jako počet základních operací, provedených během výpočtu. V informační technologii se používá časová složitost na zjištění, zda je algoritmus efektivní či nikoliv.

### 3.1.3 Prostorová složitost

Prostorová složitost udává velikost paměti potřebnou k vyřešení algoritmu. V případě Turingova stroje velikost paměti znázorňuje počet mezer na pásce, potřebných k vyřešení problému.

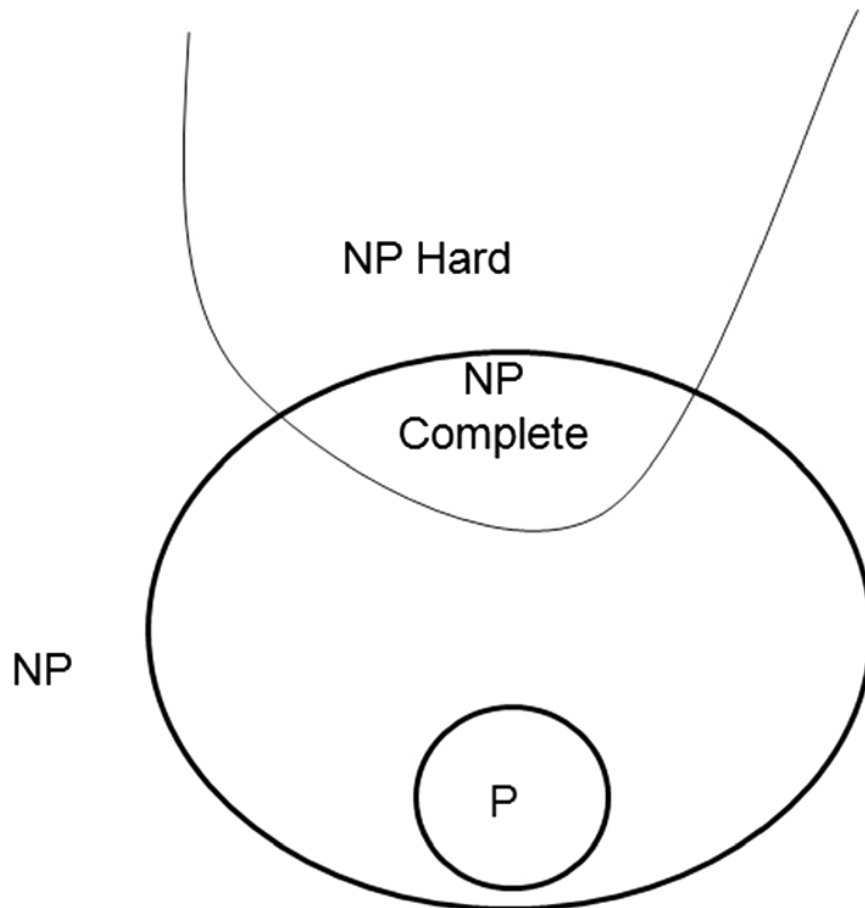
### 3.1.4 Třída P

Označovaná také jako PTIME nebo DTIME je základní třídou složitosti. Patří do nadmnožiny třídy NC a je podmnožinou třídy NP. Obsahuje sadu problémů řešitelné deterministickým Turingovým strojem v polynomiálním čase. Tyto problémy se označují za efektivní a jsou řešitelné v relativně krátkém čase [11]. Patří zde například hledání největšího společného dělitele.

### 3.1.5 Třída NP

Další třídou složitosti je NP (Nedeterministicky polynomiální), která se používá ke klasifikaci rozhodnutí problémů. Tyto úlohy se dají řešit pouze na nedeterministických strojích v polynomiálním čase. Nedeterministický stroj dokáže výpočet v každém kroku rozvést do  $n$  cest, ve kterých hledá řešení současně na rozdíl od deterministického stroje,

který v každé situaci provádí pouze jednu akci. Pokud alespoň jedna z cest uspěje, pak je známo řešení. Ekvivalentně se uvádí stroj, který dokáže uhodnout optimální cestu k hledanému řešení. Jednotlivý vztah mezi množinami je znázorněn v Obrázku 1.



Obrázek 1 Vztah množin P, NP a NP – Úplné [12]

### 3.1.6 Třída NP – Úplné

NP – úplné problémy jsou takové, pokud jsou na ně redukovatelné a převoditelné jakékoliv problémy z NP [7]. To znamená, že pokud dokážeme vyřešit problém v podmnožině NP – Úplných, řešení lze aplikovat na každý problém v množině NP.

Prokázání rovnosti  $P = NP$  by znamenalo prolomení bezpečnostních algoritmů veřejného klíče v oblasti kryptografie, jelikož v současné době bezpečnostní algoritmy závisí na neschopnosti zvládnout matematické problémy v reálném čase. Naopak díky optimalizaci by pro řadu odvětví, jako je logistika, znamenalo snížení nákladů na přepravu a efektivnější distribuci. Dokonalejší plánování výroby by mělo za následek minimalizaci výrobních nákladů a nižší tvorbu odpadu a přebytečného materiálu. Avšak tato otázka již od začátku 70. let stále zůstává nevyřešena [13].



Pravděpodobně nejznámějším problémem zde je úloha obchodního cestujícího. Úloha je konstruována na hledání nejkratší cesty mezi všemi městy v předem definované množině. Obchodní cestující musí navštívit a opustit každé město výhradně jednou a na konci trasy se musí vrátit zpět do startovního místa. Nejjednodušší algoritmus, využívá řešení "hrubou silou", kde způsob nalezení řešení je projít všechny kombinace cest a najít v tomto případě tu nejkratší cestu. Problém nastává při zahrnutí většího počtu měst do úlohy, jelikož počet cest a výpočetní složitost roste faktoriálně při přidání dalšího města k navštívení, ukázáno v Tabulce 1.

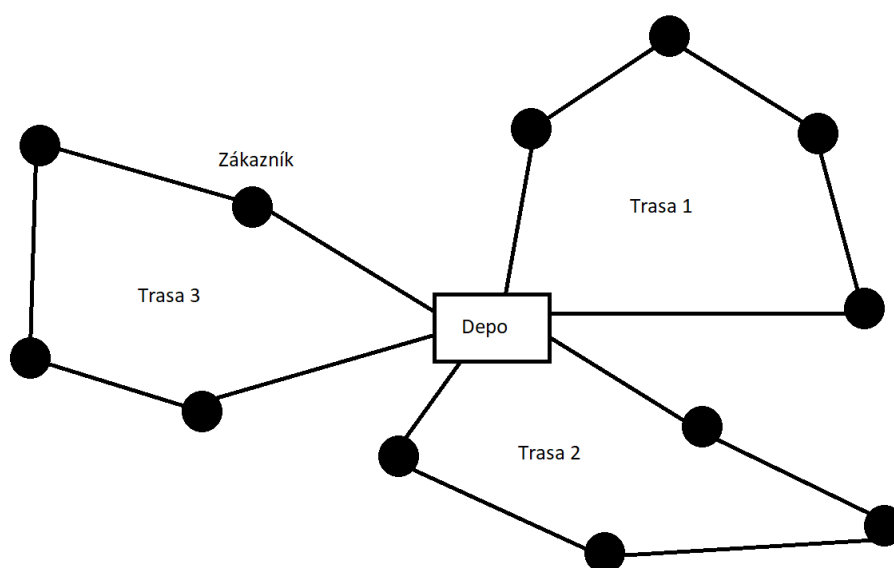
Mezi ostatní známé problémy patří zavazadlový problém, hledání hamiltonovské kružnice nebo problém dvou loupežníků.

## 4 VEHICLE ROUTING PROBLEM

V anglické terminologii běžně používaný termín, zkráceně označovaný jako VRP. Oficiální české pojmenování na rozdíl od již zmiňované úlohy obchodního cestujícího nemá. Společně s úlohou obchodního cestujícího náleží do třídy složitosti NP – Úplných, která byla popsána v podkapitole 3.1.6.

VRP je problém celočíselného programování a kombinatorické optimalizace. V současné době, díky neustálému zvyšování hustoty dopravy, je populárnější v oblastech městské logistiky, řízení dopravy zboží nebo lidí a svozu odpadu. Výzkum v oblasti VRP zefektivnilo návrhy tras a koordinaci vozového parku. Výsledky výzkumu přinášejí vyšší ekonomické a ekologické úspory. Optimalizace dopravních sítí může snížit potenciální náklady o 5 % až 30 % [14].

Obecná formulace spočívá v určení optimální trasy pro omezenou flotilu vozového parku. Ty musí plnit podmínku, že začátek a konec trasy se vykonává ze stejného počátečního místa neboli depa. Cílem je navštívit všechny uzly (body) v předem definované množině uzlů tak, že smíme přijet a opustit uzel výhradně jednou a zároveň minimalizovat celkové náklady společné s trasou. Silniční síť se v modelu sestaveného v praktické části, jakožto u VRP obecně, rozděluje do různého počtu uzlů, které jsou propojeny hranami značící hodnotu vzdálenosti nebo času, jak je znázorněno v Obrázku 2. VRP zohledňuje i jiné omezení než jen vzdálenost mezi jednotlivými body, takže se obecně nedá říct, optimální trasa je vždy ta nejkratší. Poprvé se tento problém objevil ve článku [15].



Obrázek 2 Příklad řešení trasy pomocí VRP

Pánové George Dantzig a John Ramser již v roce 1959 řešili problematiku optimální trasy pro přepravu benzínu flotilou nákladních aut. Nalezení řešení by přispělo ke snížení nákladů společné s přepravou. Při řešení uvažovali s kapacitou u jednotlivých aut. Jejich omezení bylo nastaveno, že kapacita nákladního auta je nižší než sumarizace potřeb všech zákazníků. V tomto případě nákladní auto nedokáže v jedné trase obsloužit všechny zákazníky a musí být naplánováno více tras nebo použito více aut.

Za předpokladu nezahrnutí omezení kapacity vozidla, nebo sumarizace potřeb zákazníků by byla menší nebo rovna kapacity nákladního vozidla, problém by byl formálně totožný s úlohou obchodního cestujícího (Travelling salesman problem), popsany v podkapitole 3.1.6, jelikož dopravce by byl schopen obsloužit každého zákazníka bez nutnosti znovu doplnění zásob.

Na konci práce George Dantzig a Johna Ramser [15] formulovali, že matematický problém pro minimalizaci počtu najetých kilometrů je nezávislý na počtu vozového parku, pokud všechna vozidla mají stejnou kapacitu nákladního prostoru. Na začátku byl předpoklad, že nevyužitá kapacita nepřináší žádnou ztrátu, avšak později, pro zavedení do praxe, definovali náklady za nevyužitý objem. Problém se po přidání nákladů pak stává spíše minimalizací celkových nákladů než celkového počtu ujetých kilometrů. Náklady na jednotlivé trasy nezávisí pouze na jejich délce, ale také na náhodné podmínky, jako je závažnost provozu nebo dopravní omezení.

## 4.1 Typy formulací VRP

V modelování VRP lze použít tři hlavní různé modelovací přístupy. Prvním je vehicle flow formulations, ze kterého vychází sestrojený model v praktické části. Druhým je commodity flow formulations, který se týká kapacitního omezení a posledním je set partitioning problem [16].

### 4.1.1 Vehicle flow formulations

Tento přístup je obvykle základní verzí VRP. Při modelování používá celočíselné proměnné, které reprezentují každou hranu. Ty počítají, kolikrát vozidla prošla příslušnou hranou. Využívá se v případech, kdy celkové náklady se rovnají nákladům ze všech navštívených hran a nejdůležitějšími omezeními jsou hodnoty spojené s průchody mezi uzly [16].

#### 4.1.2 Commodity flow formulations

U tohoto přístupu jsou přidány další celočíselné proměnné přidružené k jednotlivým hranám, které znázorňují kapacitu u přidružených uzlů. Modely tohoto typu se dají použít u capacitated vehicle routing problem, zkráceně CVRP [16].

#### 4.1.3 Set partitioning problem

Model má exponenciální počet binárních proměnných, které jsou spojeny s jiným proveditelným obvodem. VRP poté zjišťuje, který oddíl má minimální náklady, přičemž musí splňovat všechna náležitá omezení [16].

### 4.2 Druhy VRP

Od prvního zavedení VRP v praxi [15] se neustále vyvíjejí aplikace problému s odlišnými požadavky a omezeními podle potřeb. To vedlo k zavedení široké škály druhů VRP, při nichž rychlost už není konstantní a je závislá na dopravě a doba jízdy mezi jednotlivými body není už jen funkcí pro ujetou vzdálenost [17].

#### 4.2.1 Time-dependent vehicle routing problem

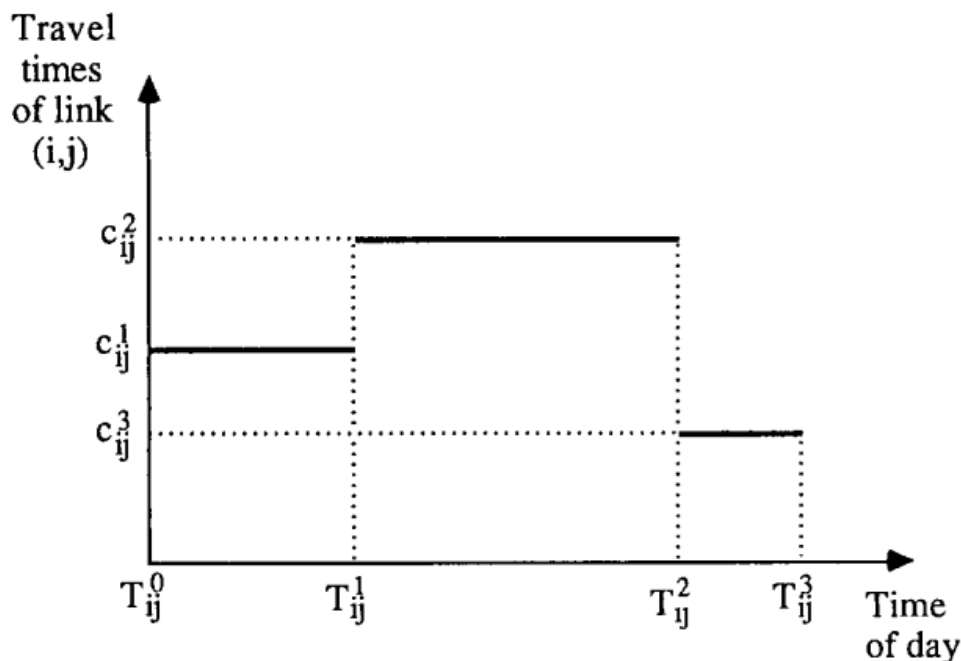
Matematickým modelem, kterým se v praktické části zabývám je v anglické terminologii označován jako time-dependent vehicle routing problem, zkráceně TDVRP, který získal v posledních letech mezi výzkumníky značnou pozornost [17,18].

Nejčastější uplatnění má v přetíženém městském prostředí, jelikož zohledňuje odlišné časové intervaly s různou délkou doby jízdy mezi uzly. Díky zmíněným intervalům dokáže lépe vyhodnotit optimální trasu skrz silniční infrastrukturu města. Čas mezi jednotlivými uzly nebo čas mezi uzlem a depem závisí na jejich vzdálenosti a na denní době, která rozděluje časové intervaly.

Uvažujme dobu jízdy mezi jednotlivými body (uzly) jako známou proměnnou. V městské dopravní síti skutečná doba jízdy není samo o sobě funkcí ujeté vzdálenosti. Je to dáno tím, že rychlost se mění v závislosti na okolnostech a tím pádem není konstantní. Změnu rychlosti mohou způsobit například rozdílná hustota dopravy, nehody na silnici, povětrnostní podmínky, změna ročního období anebo další náhodné události. Kolísání hustoty provozu má za následek kolísání rychlosti jízdy vozidel, což vede ke kolísání celkové doby jízdy. Další složka, která má za následek ovlivnění dopravní stálosti je samotná časová změna. Ta způsobuje dramatické zvýšení času stráveného na silnici během rušných hodinových,

denních, týdenních nebo sezónních cyklů, kvůli dramatickému zvýšení hustoty vozidel. Naopak snížení hustoty provozu v pozdních večerních či brzkých ranních hodinách [19].

Pokud neuvažujeme závislost měnící se dobu jízdy na příslušném denním čase, jak je znázorněno v Obrázku 3, může se celkově změnit struktura trasy. To může způsobit neoptimální cestu, kdy díky rozdílné trase bude vozidlo zdrženo vyšší dopravní hustotou provozu nebo stát v dopravní zácpě.



Obrázek 3 Přehled závislosti doby jízdy na příslušném čase [19]

Osa  $x$  představuje denní dobu rozdělený mezi 3 intervaly s odlišnou délkou. Osa  $y$  zobrazuje čas, potřebný na projetí hrany mezi uzly  $i$  a  $j$ . V Obrázku 3 trasa začíná v časovém intervalu  $T^0$ , kterému odpovídá hodnota proměnné  $C^1$ . S přibývajícím denním časem proběhne změna intervalu na  $T^1$ , ve kterém se celková doba jízdy zvyšuje na hodnotu  $C^2$ . Další změna nastává, kdy přibývajícím čas bude odpovídat intervalu  $T^2$ , ve kterém dosahuje hodnota  $C^3$  nejnižší hodnoty.

Čas potřebný na projetí dané hrany je nejvyšší v intervalu  $T^1$ . To může způsobit zhoršení dopravní situace nebo nastala jiná náhodná událost (zhoršení počasí, práce na vozovce nebo dopravní nehoda). Naopak nejnižší hodnota je v intervalu  $T^2$ , zde je dopravní situace nejlepší a odpovídá pozdním až nočním hodinám. Obecné VRP bez časové závislosti má konstantní dobu jízdy a změnu denního času vůbec nezohledňuje.

#### 4.2.2 Dynamic vehicle routing problem

U obecného VRP se předpokládá, že všechny vstupní hodnoty známe, ale to neplatí u aplikace na reálné prostředí. Zde jsou informace, například jestli daný uzel bude potřebovat obsloužit či nikoli neznámé během návrhu plánování trasy vozidla. Cílem je průběžně a dynamicky reagovat na nová vstupní data a události v reálném čase. Dále předvídat budoucí události, zejména jestli jsou k dispozici stochastické informace nebo je lze odvodit z předchozích událostí [20].

Dynamická optimalizace zkoumá a dokáže odvodit další rozhodnutí již během plánování trasy a cílem je vyřešit nejistotu v reálném čase. Takové řešení ale vyžaduje neustálou komunikaci mezi dispečerem a vozidlem. Tato metoda se také kombinuje s omezeními pro časová okna, kde zákazník má omezený čas na převzetí požadavku a flotila vozidel je směřována tak, aby vyhověli budoucím a okamžitým požadavkům, které se neustále vyvíjejí. Zde se požadavky dělí na statické (přijaly se před plánováním trasy) a dynamické (požadavky se přijaly až v průběhu vykonávání trasy).

Díky neustálému spojení s dispečerem vozidlo dokáže ihned reagovat na změnu trasy a obsloužit daný uzel v reálném čase. Ovšem tento přístup také klade větší důraz na použitou výpočetní a komunikační techniku.

## 5 REŠERŠE EXISTUJÍCÍCH MODELŮ SVOZU ODPADU

V Tabulce 2 je obsažen přehled vědeckých publikací zabývajících se VRP, TDVRP a svozem odpadu. Označené pole symbolem "✓" znamená zaměření článku na specifickou oblast problému. Ze seznamu dostupných prací je očividné, že problematika modelování s ohledem na časová omezení je nevyužívaná v oblasti odpadového hospodářství. Při hledání optimálního řešení je vhodné použití dostupných modelů a metod výpočtu řešení v oboru tranzitní dopravy a logistiky, kupříkladu ve článku Mugayskikh et al. (2018), kde je tato problematika využívána v reálných silničních sítích.

Tabulka 2 Rešerše dosavadní vědecké činnosti

Autor (rok vydání)	TDVRP	VRP	Svoz odpadu	Přeprava a logistika	Aplikace na dopravní síť
Allahyari et al. (2021)	✓				✓
Gmira et al. (2020)	✓			✓	✓
Derbel et al. (2020)		✓	✓		
Wu et al. (2020)		✓	✓	✓	✓
Pires et al. (2019)		✓	✓		
Ben Ticha et al. (2018)	✓	✓		✓	✓
Mugayskikh et al. (2018)	✓			✓	✓
Mat et al. (2018)	✓			✓	
Huang et al. (2017)	✓				✓
Han & Ponce-Cueto (2016)		✓	✓		
Mancini Simona (2014)	✓			✓	✓
Kok et al. (2012)	✓			✓	✓
Buhrkala et al. (2012)		✓	✓		✓
Donati et al. (2008)	✓			✓	✓
Malandraki & Daskin (1992)	✓				

Publikace Ben Ticha et al. (2018) a Mat et al. (2018) v Tabulce 2 zmiňují možné snížení nákladů použitím TDVRP v oboru odpadového hospodářství, avšak aplikují model pro optimalizaci v oboru logistiky a přepravní sítě. Naopak publikace Pires et al. (2019) optimalizuje svoz odpadu, avšak pomocí obecného VRP modelu. Tato bakalářská práce kombinuje oba přístupy a aplikuje vytvořený TDVRP model na svoz odpadu ve městě Zlín.

## **II. PRAKTICKÁ ČÁST**



## 6 MATEMATICKÝ MODEL SVOZU ODPADU

### 6.1 Definice úlohy

Úloha řešená níže se zabývá optimální trasou svozového vozidla pro rozdílné časové zdržení v dopravních sítích. Jelikož dopravní situace každým rokem houstne a mnohdy se liší každou hodinou, zvolený přístup je znázorněn v Tabulce 3 a rozděluje den na 7 časových intervalů, jako při řešení ve článku [17]. Ty jsou různě dlouhé a dělíme je na 2, 4 nebo 6 hodin podle toho, ve které denní době se právě nacházejí. Každé svozové vozidlo začíná a končí ve stanoveném depu. Informace potřebné pro tvorbu trasy jsou: doby jednotlivých průjezdů mezi jednotlivými kontejnery v závislosti na denní době, časy zahájení trasy, kapacita svozového vozidla, kapacity dostupných kontejnerů, doby potřebné pro obsluhu každého kontejneru a počet svozových aut.

Tabulka 3 Rozdělení časových intervalů

0:00 – 6:00 interval $m_1$
6:00 – 8:00 interval $m_2$
8:00 – 10:00 interval $m_3$
10:00 – 14:00 interval $m_4$
14:00 – 16:00 interval $m_5$
16:00 – 20:00 interval $m_6$
20:00 – 24:00 interval $m_7$

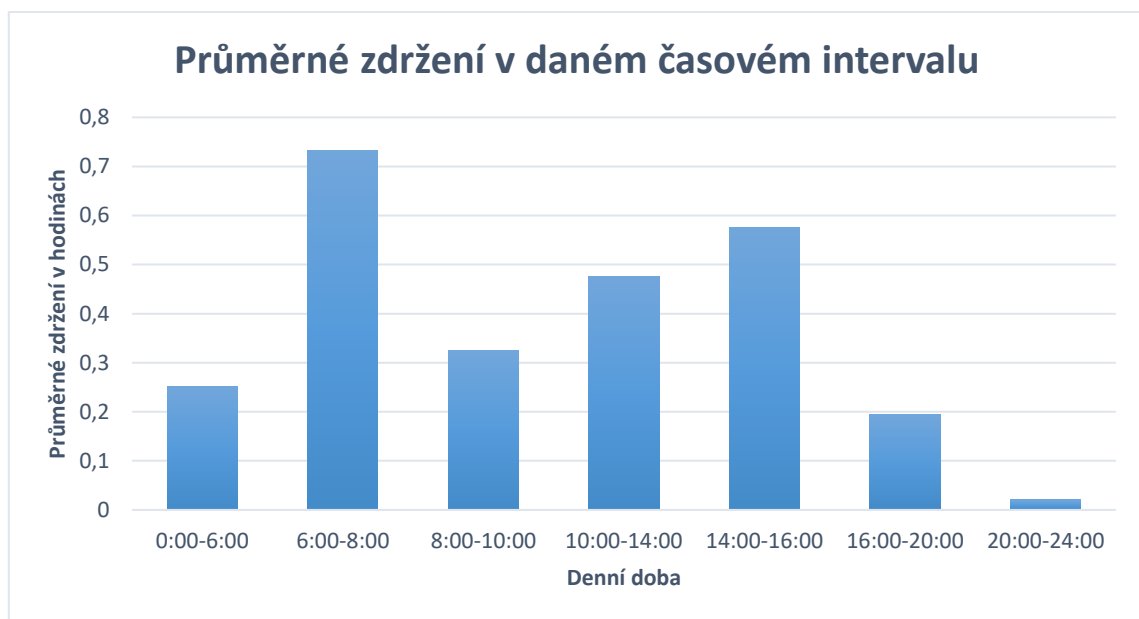
Řešení bude dosaženo za použití celočíselného lineárního programování v reálném čase. Model bude otestován za pomoci optimalizačního softwaru GAMS. Data použitá k ověření modelu jsou pouze teoretická a nejsou použita na reálnou aplikaci svozu odpadu. Výsledky budou obsaženy v grafech na konci práce.

### 6.2 Použitá data

Data použitá k otestování funkčnosti modelu pochází z města Zlín. V městské zástavbě tohoto typu se vozidlo nepohybuje konstantní (povolenou) rychlostí, proto je optimální znázornit vzdálenost mezi dvěma body na mapě pomocí časové informace / vzdálenosti. Data byla vygenerována zvlášť pro každý interval. Časy průjezdu ulicemi použité v modelu

jsou získány pomocí Google Maps, kde jsou znázorněny vzdálenosti v minutách pro různou denní dobu.

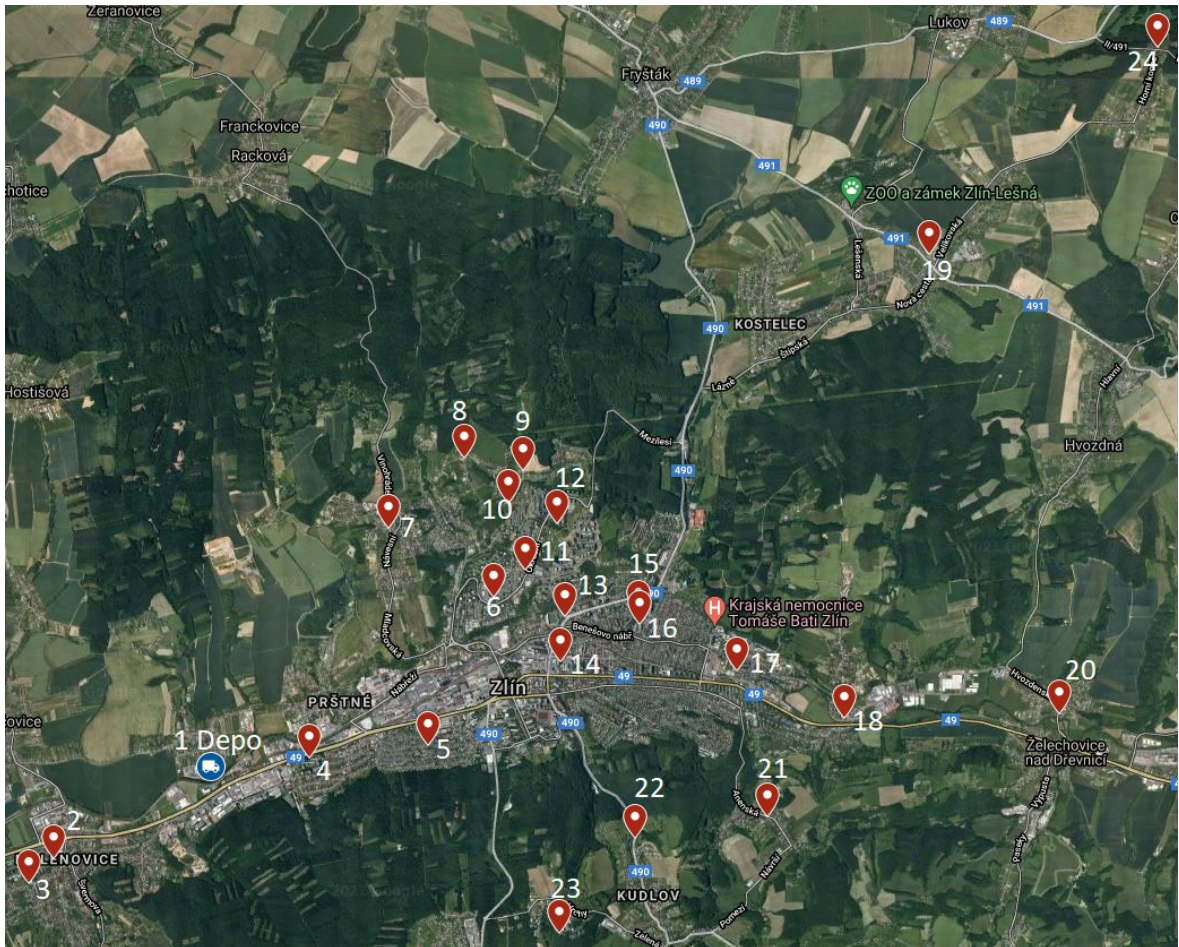
V Obrázku 4 je graficky znázorněn přehled časových zdržení způsobené dopravními omezeními. Nejnižší zdržení představuje interval  $m_7$ . Zde je zdržení zanedbatelné díky nízké hustotě dopravy. Naopak interval  $m_2$  představuje dopravní špičku. Hustota dopravy je zde vysoká, vyskytují se časté dopravní zácpy a zdržení díky dopravní situaci narůstá. Intervaly  $m_3$ ,  $m_4$  a  $m_5$  zobrazují mírnou hustotu dopravy, zácpy jen místy a zdržení není značné. Poslední intervaly  $m_1$  a  $m_6$  představují klidný městský provoz.



Obrázek 4 Graf přehledu zdržení v časových intervalech

Pro přehled a zjištění ujeté vzdálenosti vozidly na svozové cestě jsou v modelu použity také vzdálenosti mezi body v kilometrech. Tyto vzdálenosti jsou rovněž převzaty z Google Maps a vloženy do modelu. V celkovém přehledu ujetých kilometrů na konci práce jde vidět, že sestavený model využívá vždy odlišnou kombinaci trasy, tudíž ujeté kilometry se často liší v závislosti na denní době.

Poslední použité data jsou kapacity kontejnerů. Pro případovou studii jsou použity nádoby na elektro odpad firmy Asekol a.s. Každý kontejner reprezentuje jeden uzel v modelu. Přehled umístění všech kontejnerů a depa je zobrazeno v Obrázku 5. Objem každé nádoby činí  $2,15m^3$  a jejich rozmístění je dáno podle počtu obyvatel, kde na jeden kontejner připadá alespoň 1000 obyvatel [21].



Obrázek 5 Mapa města Zlína s vyznačenými kontejnery a depem

### 6.3 Sestavení teoretického modelu

Před implementací modelu do optimalizačního softwaru GAMS byl první sestaven teoretický matematický model. V této podkapitole jsou první podrobně vysvětleny všechny parametry, množiny, proměnné a jejich omezení použité v modelu. Následně v druhé sekci jsou popsány omezení a účelová funkce modelu, která minimalizuje celkový čas cesty svozu. Na závěr této podkapitoly jsou popsány omezení matematického modelu. Omezení jsou zcela zásadní při tvorbě a následně dosažení správného řešení modelu.

### 6.3.1 Množiny

Tabulka 4 Popis množin

$I$	Množina uzlů (kontejnery a depo), $i \in I$ . Množina $I$ představuje kontejnery určené ke svozu a depo, které se nachází na první pozici v množině.
$M$	Množina časových intervalů; $m \in M$
$I_1$	Množina dep; $i_1 \in I_1$
$I_2$	Množina kontejnerů bez depa; $I_2 = I \setminus I_1$
$I_v$	Množina virtuálních dep; $i_v \in I_v$
$I_a$	Množina uzlů a virtuálních dep; $i_a \in I_a$ ; $I_a = I \cup I_v$

### 6.3.2 Proměnné

Tabulka 5 Popis proměnných

$t^{overall}$	Hodnota účelové funkce – Sumarizace časů odjezdu z každého uzlu [h]
$t^{final}$	Konečný celkový čas svozu [h]
$u_i$	Pomocná proměnná; $i \in I_2$
$t_i$	Čas odjezdu z uzlu $i$ [h]; $i \in I$
$w_i$	Kapacita vozidla při příjezdu do uzlu $i$ [m <sup>3</sup> ]; $i \in I$

### 6.3.3 Binární proměnné

Tabulka 6 Popis binární proměnné

$x_{i,j,m}$	Binární proměnná, $x_{i,j,m} \in \{0, 1\}$ . Pokud vozidlo přímo směřuje z uzlu $i$ do uzlu $j$ v časovém intervalu $m$ , pak hodnota binární proměnné $x_{i,j,m}$ je 1, v opačném případě je hodnota vždy 0; $i, j \in I$ ; $m \in M$
-------------	--

## 6.3.4 Parametry

Tabulka 7 Popis parametrů

$V$	Počet svozových vozidel
$n$	Počet uzlů k navštívení, zahrnující depa
$M^1$	Velké celé číslo používané v omezeních
$M^2$	Celé číslo, znázorňuje maximální čas, kdy vozidlo může dojet do depa
$St$	Čas odjezdu svozového vozidla z depa [h]
$MaxW$	Maximální povolená kapacita svozového vozu [m <sup>3</sup> ]
$c_i^{servis}$	Čas potřebný pro obsluhu kontejneru (uzlu) $i$ [h]; $i \in I$
$d_i$	Zaplňená kapacita kontejneru (uzlu) $i$ [m <sup>3</sup> ]; $i \in I$
$l_{i,j}$	Vzdálenosti mezi kontejnery (uzly) $i, j$ [km]; $i, j \in I$
$l^{final}$	Celkový počet kilometrů ujetých vozidly [km]
$c_{i,j,m}$	Doba jízdy mezi uzly $i, j$ v závislosti na denním intervalu $m$ ; $i, j \in I$ ; $m \in M$
$t_{i,j,m}^1$	Hodnota horní hranice časového intervalu $m$ . Uvádí maximální hodnotu času pro hranu mezi uzly $i, j$ ; $i, j \in I$ ; $m \in M$
$\dot{l}_{i,j}$	Vzdálenost mezi kontejnery (uzly) $i, j$ ; $i, j \in I$

Optimální parametr pro elektro kontejnery je kapacita. Jelikož se zde mohou vyskytovat baterie, či další nebezpečná zařízení nebo nářadí, nespadá v úvahu odpad stlačit a ušetřit více místa. Při zaplnění maximální kapacity je potřeba nádobu vyvézt. Pro čas potřebný na vyprázdnění a přípravu kontejneru na následné znovupoužití slouží parametr  $c_i^{servis}$ , kde je pro každý kontejner uvedená hodnota a posléze započítána do omezení.

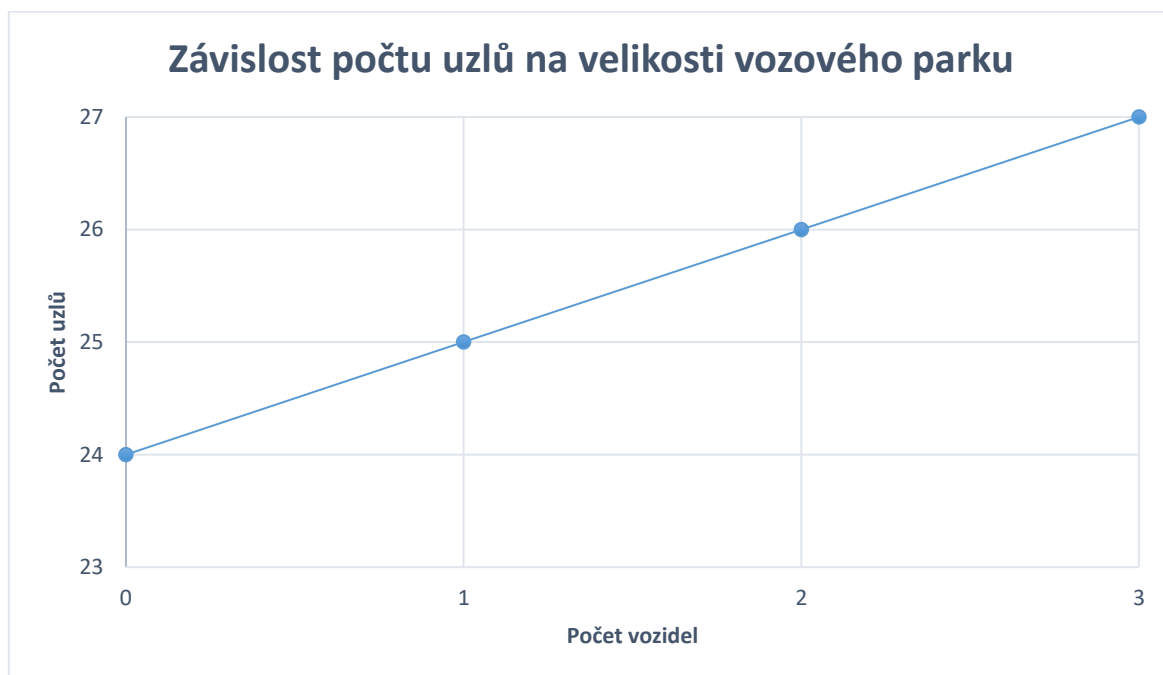
Matrice  $C$  představuje pro model hlavní zdroj dat. Jsou zde uvedeny všechny naměřené hodnoty doby jízdy mezi jednotlivými uzly v každém časovém intervalu. Pomocí omezení v modelu jsou posléze vybrána konkrétní data pro výpočet.

Matrice  $T^1$  musí začínat v čase 0:00 a končit do 24:00 aby byla dodržena správnost výsledků. V opačném případě by výsledný čas byl nesprávný a byl v rozporu s denním časem.

### 6.3.5 Předpoklady nutné k sestavení modelu

Za předpokladu, že uvažujeme uzel  $i_1$  jako depo, musíme množinu  $I$  rozšířit o počet použitých vozidel  $V$  ( $i = 1, 2, \dots, I + V$ ), jak je znázorněno v Obrázku 6. Tyto přidání uzly, nebo také "virtuální depo", odpovídají vracejícím se vozidlům zpět do depa, jako ve článku [19].

V případě nezohlednění této skutečnosti model není schopný započítat vracející se vozidla zpět do depa a konečný výsledek bude v rozporu se skutečností.



Obrázek 6 Graf závislosti počtu uzlů na vozidlech

### 6.3.6 Účelová funkce

Účelové funkce mají za úkol minimalizovat náklady, které je nutné optimalizovat. V případě modelu řešeného níže hrají zásadní roli při minimalizaci celkového času vozidla.

Funkce (1) minimalizuje čas potřebný na projetí všech kontejnerů. Zaručuje výběr optimální trasy a eliminuje nežádoucí zpoždění způsobené zhoršenou dopravní situací. Přínosem je minimalizace zdržení mezi projetí každou hranou, šetrnější dopad na dopravní situaci, zmenšení emisí vyprodukované vozidlem a nižší celkové náklady.

$$\text{Min } t^{\text{overall}} = \sum_{i_a=1}^{n+V} t_{i_a} \quad (1)$$

### 6.3.7 Rovnice a omezení modelu

Rovnice jsou nezbytně nutné pro správnou funkci zadaného problému. Přiřazují hodnoty proměnným a zároveň zahrnují veškeré podmínky a důležitá omezení. Provádí výpočet a obsahují hlavní logiku modelu.

Rovnice (2) počítá čas příjezdu zpět do depa. Pokud je v modelu zahrnuto více vozidel, započítá se jen výsledný čas každého svozového vozidla. Do proměnné  $t^{final}$  se započítává doba projetí každou hranou v modelu a zároveň čas potřebný k obsluze určených nádob.

$$t^{final} = \sum_{i_v=1}^V t_{i_v} - St \quad (2)$$

Model pracuje se začátečním časem, konkrétně v rovnici (2). Je důležité výsledný čas  $t_i$  odečíst od času  $St$ , jelikož mimo půlnoc nikdy nezačíná nulou. Nebýt rozdílu hodnot, výsledek by byl podstatně větší a nepřesný od skutečného času.

Pro počet vozidel, které mohou vyjet z depa slouží rovnice (3). Bez tohoto omezení by množství aut nebylo regulováno.

$$\sum_{j=2}^n \sum_{m=1}^M x_{1,j,m} = V \quad (3)$$

Podobně jak v rovnici (3) slouží omezení (4) na regulaci vozidel. Jediný rozdíl je, že zde se redukuje počet vozidel směřující do depa. Je nutné dodržet stejnou hodnotu  $V$  pro oba typy rovnic.

$$\sum_{i=2}^n \sum_{m=1}^M x_{i,1,m} = V \quad (4)$$

Návštěvu každého uzlu (kontejneru) právě jednou určuje omezení (5). Výjimku tvoří depo neboli uzel 1, pro které omezení neplatí. Omezení (6) zastává opačnou roli a zaručuje jej opustit právě jednou.

$$\sum_{j=2}^{n+V} \sum_{m=1}^M x_{i,j,m} = 1; \quad i = 2, 3, \dots, n + V \quad (5)$$

Dohromady rovnice (5) a (6) zajistí sběr každého kontejneru pouze jednou a nevznikne možnost navštívit vícekrát na stejné místo. Tyto podmínky tvoří celek a nelze je použít každou zvlášť.

$$\sum_{i=2}^{n+V} \sum_{m=1}^M x_{i,j,m} = 1; \quad j = 2, 3, \dots, n + V \quad (6)$$

Podmínka (7) zamezuje tvorbu dílčích okruhů trasy. Proměnná  $u_i$  obsahuje reálná čísla a nabývá libovolných hodnot. Binární proměnná  $x_{i,j,m}$  dosahuje hodnoty 1 jestli existuje cesta mezi místy  $i$  a  $j$  v rámci okruhu a v opačném případě nabude hodnoty 0, což značí že mezi místy  $i$  a  $j$  se cesta nenachází. Bez této podmínky by trasa byla nekonzistentní a existovaly by nežádoucí dílčí okruhy.

$$u_i - u_j + n \cdot x_{i,j,m} \leq n - 1; \quad i, j = 2, 3, \dots, n; m = 1, 2, \dots, M; i \neq j \quad (7)$$

K nastavení začátečního času slouží rovnice (8). Proměnná  $St$  nastavuje stejný čas všem vozidlům vyjíždějící z depa a měla by být nižší než maximální povolená doba, kdy vozidlo může přijet do uzlu 1. Pokud vynecháme podmínku, začáteční čas  $St$  by byl vždy nulový. Start vozidel by byl vždy o půlnoci nehledě na denní čas.

$$t_1 = St \quad (8)$$

Časové omezení (9), počítá čas odjezdu z uzlu  $j$ . Zajišťuje splnění podmínky pouze v případě, kdy proměnná  $x_{i,j,m}$  je rovno 1, podobně jak ve článku [19]. Při hodnotě 0 se podmínka nesplní díky násobení nulou s velkým celým číslem  $M^1$ . V souladu s účelovou funkcí (1) zapříčiní, že vozidlo nebude v žádném okamžiku na svozové trati stát a čekat na vhodný časový interval  $m$ , kde hodnota odjezdu z uzlu  $j$  by byla nižší než ve stávajícím časovém intervalu.

$$t_j - t_i - x_{i,j,m} \cdot M^1 \geq c_j^{servis} + c_{i,j,m} - M^1$$

$$i = 1, 2, \dots, n; j = 2, 3, \dots, n + V; m = 1, 2, \dots, M; i \neq j \quad (9)$$



Pro správný výběr časového intervalu  $m$  v závislosti na denní době slouží omezení (10) a (11). Výběr se porovnává s časem odjezdu z uzlu  $i$  a v případě že stanovený čas bude vyšší, než dovoluje daný časový interval  $m$ , zvolí se interval  $m + 1$ .

$$T_{i,j,m}^1 + M^2 \geq t_i + x_{i,j,m} \cdot M^2$$

$$i = 1, 2, \dots, n; j = 2, 3, \dots, n + V; m = 1, 2, \dots, M; i \neq j \quad (10)$$

Platnost podmínky probíhá pouze v případě dosažení  $x_{i,j,m}$  hodnoty 1, pouze tehdy se vzájemně vynulují velká čísla  $M^2$ .

V omezení (11) provádí kontrola správnosti zvoleného časového intervalu. Rozdíl hodnot  $t_i$  je s intervalem  $m$  o jedna nižší je po celou dobu trasy větší nebo rovno 0. V opačném případě by došlo ke zvolení nesprávného paralelního spojení  $m$  vůči denní době.

$$t_i - x_{i,j,m} \cdot T_{i,j,m-1}^1 \geq 0$$

$$i = 1, 2, \dots, n; j = 2, 3, \dots, n + V; m = 1, 2, \dots, M; i \neq j \quad (11)$$

Časové omezení návštěvy všech uzlů je obsaženo v rovnici (12).

$$t_i \leq M^2; i = 1, 2, \dots, n \quad (12)$$

Rovnice (13) slouží na určení celkové vzdálenosti ujeté jedním nebo více vozidly. Vzdálenost se počítá od vyjetí z depa do přijetí zpět.

$$l^{final} = \sum_{i=1}^{n+V} x_{i,j,m} \cdot l_{i,j} \quad (13)$$

### 6.3.8 Omezení a podmínky pro přidání kapacity

Rovnice (14) značí, že nákladový prostor vozidel vyjíždějících z depa je prázdný. Podmínka slouží i pro opakované přijetí do depa, kde se nákladový prostor vždy vyprázdní.

$$w_1 = 0 \quad (14)$$

K určení maximální kapacity nákladového prostoru slouží omezení (15), jenž nastavuje hodnota  $MaxW$ . Po dosažení maxima povolené kapacity, nebo kdyby hrozilo překročení stanovené hodnoty při obsluze dalšího uzlu, vozidlo nemůže pokračovat ve stávajícím svozu a směřuje zpět do depa. Po vyprázdnění nákladu může pokračovat v přerušené trase.

$$w_i \leq MaxW; i = 1, 2, \dots, n \quad (15)$$

Poslední důležité omezení (16) souvisí s kapacitou nákladu při odjezdu z uzlu  $j$ . Ta nesmí být nižší jak sumarizace všech předchozích uzlech. Společně podmínky (14), (15) a (16) přidávají do modelu kapacitu vozu podobně, jak řešili problém ve člancích [19,22].

$$d_j - MaxW \leq w_j - w_i - MaxW \sum_{m=1}^M x_{i,j,m}$$

$$i = 1, 2, \dots, n; j = 2, 3, \dots, n + V; i \neq j \quad (16)$$

### 6.3.9 Definice proměnných

Rovnice (17) a (18) určují podmínky nezápornosti pro kapacitu vozidla a denní čas. Omezení (19) definuje nezápornost pomocné proměnné  $u_i$  rovnice pro zamezení tvorbu dílčích okruhů cesty. Rovnice (20) a (21) definují nezápornost celkového času svozu a konečného času svozu a poslední (22) značí definici binární proměnné.

$$w_i \geq 0; i = 1, 2, \dots, n \quad (17)$$

$$t_i \geq 0; i = 1, 2, \dots, n \quad (18)$$

$$u_i \geq 0; i = 2, 3, \dots, n \quad (19)$$

$$t^{overall} \geq 0 \quad (20)$$

$$t^{final} \geq 0 \quad (21)$$

$$x_{i,j,m} \in \{0, 1\}; i, j = 1, 2, \dots, n; m = 1, 2, \dots, M \quad (22)$$

## 6.4 Rozdíl oproti obecnému modelu

Sestavený model v rámci této bakalářské práce se v pár bodech liší od obecného. První znatelnou úpravou je vložení časových interval. Díky tomu umožňuje zvolený přístup vždy výběr optimální cesty pro jakoukoliv denní dobu.

Druhým bodem je přidání kapacity pro vozový park. Kapacita je důležitým prvkem pro pozdější aplikaci do reálné situace. Se započítáním kapacity souvisí zvyšování celkových výdajů. Také zvyšuje čas, jelikož se v průběhu trasy musí vyprázdnit. Tomuto lze předejít použitím více svozových vozů, kde denní doba příjezdu do depa je nižší, avšak samotná cesta je znatelně delší. Použití zmíněných přístupů je znázorněno a porovnáno ve grafech na konci práce.

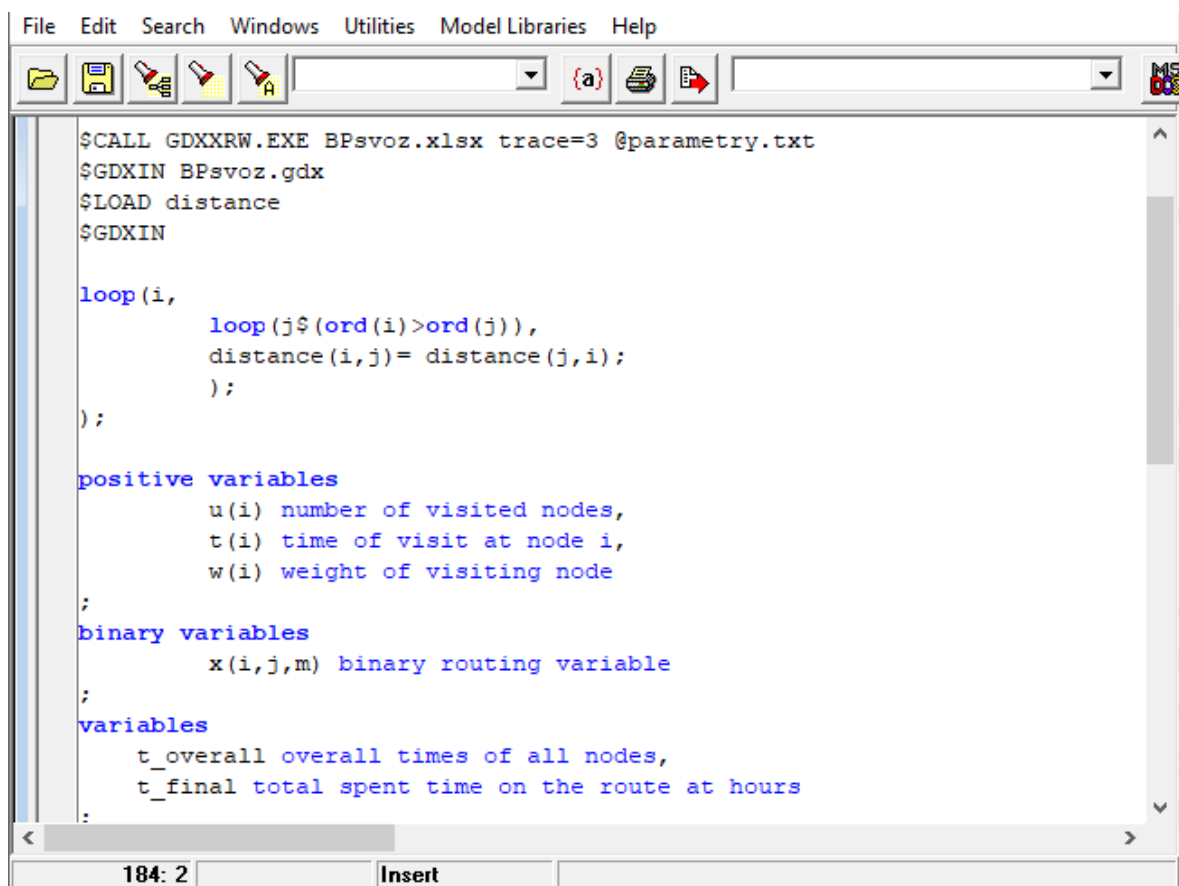
Dalším přidaným prvkem je čas strávený při obsluze. Tato hodnota byla nastavena na 6 minut u všech bodů v modelu. Avšak model umožňuje změnit dobu obsluhy na různé hodnoty.

Posledním rozdílem je nastavení maximální kapacity svozových vozidel, stanovená na  $20m^3$ . Tato hodnota je pro celý vozový park stejná.

## 7 IMPLEMENTACE MODELU DO SOFTWARE GAMS

V této kapitole je podrobně popsána implementace sestrojeného modelu do optimalizačního softwaru GAMS. Ukázka zmiňovaného prostředí znázorňuje Obrázek 7. Podkapitoly jsou dále rozděleny na Použitý optimalizační software, Implementace do modelu a poslední jsou Problémy při implementaci a jejich následné řešení.

Zvolení vhodného optimalizačního nástroje výrazně urychlí práci nad modelem a následným výpočtem. Zápis modelu byl proveden v anglickém jazyce a za využití terminologie, která je zcela výhradně používána v této problematice.

The image shows a screenshot of the GAMS (General Algebraic Modeling System) software interface. The window title is "GAMS" and the menu bar includes "File", "Edit", "Search", "Windows", "Utilities", "Model Libraries", and "Help". The toolbar contains icons for file operations and execution. The main text area displays the following GAMS script:

```
$CALL GDXXRW.EXE BPsvoz.xlsx trace=3 @parametry.txt
$GDXIN BPsvoz.gdx
$LOAD distance
$GDXIN

loop(i,
    loop(j$(ord(i)>ord(j)),
        distance(i,j)= distance(j,i);
    );
);

positive variables
    u(i) number of visited nodes,
    t(i) time of visit at node i,
    w(i) weight of visiting node
;

binary variables
    x(i,j,m) binary routing variable
;

variables
    t_overall overall times of all nodes,
    t_final total spent time on the route at hours
;
```

The status bar at the bottom shows the cursor position "184: 2" and the word "Insert".

Obrázek 7 Ukázka prostředí GAMSu

### 7.1 Použitý optimalizační software

Programovací jazyk GAMS (General Algebraic Modeling System) byl vyvinut v 70. letech minulého století jako projekt pro Světovou banku. Jako první svého druhu kombinoval jazyk matematické algebry s tradičními koncepty počítačového programování. Díky tomu efektivně popisoval a řešil optimalizační úlohy. Dnes je algebraické modelování považováno za nejproduktivnější způsob implementace optimalizačních problémů [23].

GAMS jako takový se skládá z překladače kódu a řady použitých solverů (řešitelů). Umožňuje řešit jak lineární, tak nelineární a smíšené celočíselné optimalizační problémy. Použitý solver (řešitel) při výpočtu implementovaného modelu je IBM ILOG CPLEX Optimization studio, zkráceně CPLEX. Ten je běžně používaný na úlohy typu VRP, jelikož model používá kombinaci celých čísel a binární proměnné (MIP).

### 7.1.1 CPLEX

Nástroj vyvinutý k řešení rozsáhlých problémů lineární optimalizace a úloh s celočíselným programováním. Vytváří přesná a logická rozhodnutí za účelem zvýšení efektivity, snížení nákladů a také zvýšení ziskovosti. Původně byl navržen pro implementaci v programovacím jazyce C. Dnes již poskytuje rozhraní i pro jazyky C++, C#, Python a je přístupný i pro optimalizační softwary [24].

## 7.2 Implementace modelu

V této sekci je detailně popsána implementace matematického modelu. Jakožto každý programovací jazyk i GAMS má svou syntaxi zápisu parametrů, množin, proměnných a rovnic, která se musí dodržet pro správný chod modelu. K tomuto účelu slouží dokumentace jazyka, která poskytuje uživatelům souhrnný přehled používané terminologie. Obsahuje přesné definice, správné oddělování mezer, psaní speciálních znaků a tvorbu funkcí či dalších nástrojů při psaní modelu. Kód modelu je ze softwaru GAMS přepsán do tabulek kvůli lepší ucelenosti a čitelnosti.

### 7.2.1 Prvotní nastavení modelu

Kód 1 Počáteční nastavení modelu

```
1   $eolcom //  
2   $set depot i1
```

Na začátku modelu je přehlednější definovat počátečnímu místu slovní označení. To probíhá na řádce 2, který přiřazuje označení "depot" uzlu i1. Řádek 1 předefinuje symbol poznámky na konci řádku. Ve výchozím nastavení jsou symboly inicializovány na !!. Pro vytvoření poznámky stačí napsat // a automaticky vzniklý text za těmito symboly značí poznámky či komentáře.

### 7.2.2 Zápis a vkládání dat do parametrů

Parametr lze deklarovat v jedné nebo více sadách. Může být zadán jako číslo, které se v průběhu času nemění, ale i naopak. Skládá se ze 2 částí a hodnoty. První částí je jméno, které musí obsahovat každý parametr a druhou volitelnou částí je indexace vektoru dat. Každá uvedená hodnota poté odpovídá na jednu nebo více rozměrnou množinu prvků, jak je uvedeno v Kódu 2 na řádce 10. Při neinicializování je výchozí hodnota 0. Deklarovat prázdný parametr je možné pouze v případě, kdy se použije s příkazem `$onEmpty` [25].

Kód 2 Zápis sady parametrů v GAMSu

```
3   Parameter
4   v 'number of vehicles' /1/,
5   n 'number of nodes to visit' /24/,
6   M1 'big number 1' /1000/,
7   M2 'big number 2' /24/,
8   MaxW 'maximum weight capacity of vehicle' /30/,
9   St 'starting time at node 1' /0/,
10  distance(i,j) 'distance between nodes',
11  distance_final 'distance of all nodes';
```

Sada parametrů se vkládá pomocí klíčového slova `Parameter`. Parametry na řádce 4 až 10 jsou vstupní parametry. Před začátkem výpočtu je nutné nastavit hodnoty k počtu svozových vozů, počtům uzlů k navštívení, čísel M1 a M2, maximální kapacitě svozových vozidel a denní době začátku svozu v hodinách. Parametr `distance` na řádce 10 představuje hrany mezi uzly `i` a `j`. Parametr `distance_final` je výstupní parametr, tudíž před začátkem výpočtu nabývá hodnoty 0.

Kód 3 Vkládání dat do parametrů c\_servis, d

```
12 Parameter c_servis(i) 'service time at node i'
13 /i2 0.1
14 i3 0.1
15 i4 0.1/;
16 Parameter d(i) 'weight of each node'
17 /i2 1.04
20 i3 1.8
21 i4 0.7/;
```

První polovina obsahuje inicializaci parametru `c_servis(i)`, udávající hodnotu časového zdržení na obsluhu každého uzlu. Ve druhé polovině inicializujeme kapacitu každého uzlu neboli parametr `d(i)`. Pro oba parametry je znázorněno vložení dat přímo do modelu. Pro docílení správného výsledku, musí být vložena data ještě před zahájením výpočtu. Ty se do modelu mohou vkládat i externě, pro takové vložení se využívají speciální příkazy v ukázce kódu 4.

Kód 4 Vkládání externích dat z Microsoft Excel

```
22 $CALL GDXXRW.EXE BPsvoz.xlsx trace=3 @parametry.txt
23 $GDXIN BPsvoz.gdx
24 $LOAD distance
25 $GDXIN
```

Data z Microsoft Excel lze exportovat voláním příkazu `$CALL` nástroje `GDXXRW`, který postupně čte z `.xlsx` souboru a následně zapisuje do souboru s příponou `GDX`. Pro import dat ze souboru `GDX` slouží příkazy `$GDXIN` na řádcích 23 a 25. První volání slouží ke čtení souboru a druhé volání k následnému zavření. Příkaz `$LOAD` importuje příslušný parametr do modelu, který musí být již předem definovaný.

### 7.2.3 Zápis množin

Jak již bylo zmíněno v podkapitole zápisu parametrů, množiny mohou být také zapsány a deklarovány v jedné nebo více sadách. Seznam prvků v množině je oddělen lomítkem z obou stran. Syntaxe pro zápis prvku je umožněna pomocí znaku hvězdičky. Postupný zápis prvků musí být oddělen od ostatních buď koncem řádku nebo čárkou mezi nimi. Tyto prvky představují jednotlivé uzly a jejich hrany. Uzel je reprezentován jako index `i`, kdežto hrana

je vždy znázorněna pomocí dvou indexů  $i$  a  $j$ . První index vždy značí bod odkud vyjíždíme a druhý index značí uzel kam směřujeme.

Kód 5 Zápis množin v GAMSu

```
26 sets
27 i /i1*i26/
28 ;
29 alias (i,j);
30 sets
31 m /m1*m7/;
32 set e(i,j,m) 'edges between nodes';
33 e(i,j,m)$c(i,j,m) = yes;
34 set i1(i) /%depot%/;
35 set i2(i);
36 i2(i)$ (not i1(i)) = yes;
37 set i_v(i) 'all virtual depots'
38 /i25
39 i26/;
```

Zde je znázorněno vkládání pomocí klíčových slov `sets` pro sadu množin a `set` pro jednotlivé množiny zvlášť. Dále se zde objevuje jak automatické vkládání prvků množiny pomocí speciálního znaku hvězdičky, tak i výpis jednotlivých prvků pomocí oddělení řádku.

Před provedení výpočtu je nutné definovat depo ve množině `i1` a virtuální depa v množině `i_v`. Díky již zmiňovaného příkazu `$set depot i1` je jeho slovní označení přehledné a nezaměnitelné od ostatních uzlů. Virtuální depa jsou nutné definovat kvůli vzniklému problému, který je podrobně popsán na konci kapitoly.

Jelikož model počítá s hranami definované v množině  $i$ , je nutné jí přiřadit další název. K tomu slouží příkaz `alias`, kdy určené množině přiřadíme alternativní název. Původní a alternativní název vždy budou obsahovat stejné prvky. Na pořadí jmen nezáleží a jediným omezením je nutnost definovat alespoň jednu z uvedených množin před použitím příkazu.



### 7.2.4 Zápis matic

Maticе představují přehlednější a jednodušší metodu pro zadávání dat než přístup založený na seznamu. Na rozdíl od parametrů musí být matice minimálně dvourozměrná. Při použití více než dvou rozměrů se jednotlivě oddělují znakem tečky. Data se poté uspořádávají do sloupců oddělené od sebe mezerami. Při nedodržení správné syntaxe zápisu dat nedojde ke správnému přečtení a následnému zpracování.

Kód 6 Ukázka zápisu dat do matice C

```

40  table c(i,j,m) 'travel time from node i to j during m'
41      i1.m1  i2.m1  i3.m1  i1.m2  i2.m2  i3.m2
42  i1          6      7          8      9
43  i2      6          1      9          2
44  i3      6      1          10      3
45  ;
46  c(i,j,m) = c(i,j,m) / 60; // conversion from min to hours

```

Kód 6 představuje deklaraci matice a poté následné naplnění vstupními daty. Prázdné místa nebo také 0 značí, že neexistuje hrana mezi stejnými uzly. Absence hrany je z důvodu zamezit navštívení sebe sama.

Vstupní hodnoty v matici značí čas projetí mezi konkrétními uzly v daný časový interval  $m$ . Pro lepší názornost je značení v minutách. Následný převod na hodiny, které posléze používá model, jako výchozí jednotku času, se nachází na řádce 46.

Kód 7 Ukázka zápisu dat do matice T1

```

47  table T1(i,j,m) 'upper time bound for time interval m'
48      i1.m1  i2.m1  i3.m1  i1.m2  i2.m2  i3.m2
49  i1          6      6          8      8
50  i2          6          8
51  i3
52  ;
53  T1(i,j,m) = max(T1(i,j,m), T1(j,i,m));

```

Matice T1 nastavuje časové omezení pro každý interval ve výpočtu. Hodnoty jsou stejné pro všechny body v danou dobu. Zapisuje se pouze poslední přípustný čas pro daný úsek neboli pro číslo 6 konkrétní interval zahrnuje denní dobu od půlnoci do šesti hodin ráno. Při

překročení stanoveného času se úsek zvýší o jedna. Pro nový úsek vždy platí nový maximální přípustný čas, který musí být vyšší než minulý. Rovnice na posledním řádku zaručí pouze výběr indexů  $T1(i, j, m)$  díky neinicializování jejich převrácených protějšků  $T1(j, i, m)$ .

### 7.2.5 Zápis proměnných

Hodnoty proměnných v modelu jsou neznámé před začátkem kompilace. Nabývají kladných hodnot až při výpočtu a mohou se měnit každou prováděnou iterací. Vyjimku tvoří proměnné  $t\_overall$  a  $t\_final$ . Ty slouží pro uchování výsledků účelových funkcí a GAMS nedovoluje nijak omezit jejich rozsah. Proto mohou nabývat libovolných reálných čísel. Poslední uváděnou variantou je binární proměnná s rozsahem 0 a 1. Syntax a zápis sad je stejná jako u množin nebo parametrů.

Kód 8 Zápis proměnných v GAMSu

```
54  positive variables
55      u(i) number of visited nodes,
56      t(i) time of visit at node i,
57      w(i) capacity of visiting node
58  ;
59  variables
60      t_overall overall times of all nodes,
61      t_final total spent time on the route at hours
62  ;
63  binary variables
64      x(i, j, m) binary routing variable
65  ;
```

Binární proměnná  $x(i, j, m)$  vyjadřuje existenci hrany v okruhu pro číslici 1. Naopak pro 0 značí její absenci. Kladná proměnná  $u(i)$  vstupuje do rovnice pro zamezení dílčích okruhů a nabývá libovolných čísel,  $t(i)$  stanovuje čas návštěvy uzlu  $i$  a poslední  $w(i)$  určuje aktuální kapacitu při návštěvě uzlu  $i$ .

### 7.2.6 Zápis účelové funkce a omezení

Rovnice se deklarují v jedné sadě equations. To musí být provedeno dříve než jejich pozdější definice v modelu.

Kód 9 Deklarace rovnic

```
66 equations
67     start_constraint
68     end_constraint
69     source_node_constraint (i)
70     destination_node_constraint (j)
71     subtour_elimination_constraint (i,j,m)
72     constraint_t0(i)
73     time_constraint(i,j,m)
74     time_constraint1(i,j,m)
75     time_constraint2(i,j,m)
76     weight_constraint(i,j)
77     weight_constraint_w0(i)
78     weight_constraint1(i)
79     objective_function_t_overall
80     function_t_final
81 ;
```

Účelová funkce obsahuje slovní spojení objective\_function. Ostatní rovnice znázorňují omezení a označují se slovem constraint. Rovnice t\_final počítá čas svozu vozidel a značí se function.

Při definování rovnic je důležité dodržet správnou syntaxi. GAMS neumožňuje zápis účelových funkcí a omezení konvenčním matematickým způsobem. Název je shodný se sadou deklarací rovnic. Za názvem mohou následovat indexy množin a logické podmínky. Povinnou součástí jsou dva symboly teček ".." pro značení začátku algebry. Poslední součástí syntaxe rovnic je samotná jejich definice, která se zakončuje symbolem středníku. Zakončení středníkem ovšem není povinné a pouze zlepšuje přehlednost pro konec rovnice.

## Kód 10 Zápis rovnic

```

82  start_constraint.. sum((i2(j),m), x('%depot%',j,m)) =e= V;
83  end_constraint.. sum((i2(i),m), x(i,'%depot%',m)) =e= V;
84  source_node_constraint (i2(i)).. sum(e(i,j,m), x(i,j,m))
=e= 1;
85  destination_node_constraint (i2(j)).. sum(e(i,j,m),
x(i,j,m)) =e= 1;
86  subtour_elimination_constraint (e(i,j,m))$(i2(i) and
i2(j)).. u(i) - u(j) + n*x(i,j,m) =L= n-1;
87  constraint_t0(i).. t('%depot%') =e= St;
88  time_constraint(i,i2(j),m).. t(j) - t(i) - x(i,j,m)*M1 =G=
c_servis(j) + c(i,j,m) - M1;
89  time_constraint1(i,i2(j),m).. T1(i,j,m) + M2 =G= t(i) +
x(i,j,m) * M2;
90  time_constraint2(i,i2(j),m).. t(i) - x(i,j,m) * T1(i,j,m-1)
=G= 0;
91  weight_constraint(i,i2(j)).. d(j) - MaxW =L= w(j) - w(i) -
MaxW * sum(e(i,j,m), x(i,j,m));
92  weight_constraint_w0(i).. w('%depot%') =e= 0;
93  weight_constraint1(i).. w(i) =L= MaxW;
94  objective_function_t_overall.. t_overall =e= sum((i),t(i));
95  function_t_final.. t_final =e= sum(i_v(i),t(i) - St);

```

## 7.2.7 Omezení pro výpočet

U problémů, patřící do třídy složitosti problémů NP-úplných, je nalezení řešení časově náročnější pro každý přidávaný uzel a pro složitější modely mohou dojít výpočetní prostředky stroje. Ukázka v kódu 11 představuje možnosti regulace trvání doby výpočtů.

## Kód 11 Omezení pro výpočet

```

96  option optcr = 0.15;
97  option resLim = 100000;
98  option iterLim = 10000000;

```

Příkaz `optcr` akceptuje relativní toleranci od optimálního řešení. Využívá se v případech, kdy nalezení řešení je extrémně obtížné vyřešit a mnohdy potřebuje enormní množství zdrojů a času. V modelu je tolerance nastavená na 15 % a po dosažení se výpočet ukončí.

Časový limit `resLim` je ochrana před extrémně dlouhým výpočtem. Určuje maximální čas provádění výpočtu. Jednotkou jsou sekundy a výchozí nastavení je na dobu 1000 sekund. Nastává v situaci, kdy chod programu překročí povolenou dobu. V takovém případě dochází k přerušení a vypsání dosavadních výsledků.

Poslední `iterLim` nastavuje povolený počet prováděných iterací. Při překročení stanoveného limitu nastává stejný scénář jak u předešlých dvou příkazů. Dojde k přerušení svoz

### 7.2.8 Nastavení modelu

Použité nastavení níže minimalizuje proměnnou `t_overall` za použití smíšeného celočíselného programování. To pracuje pouze s binárními a celočíselnými proměnnými, nikoliv s nelineárními výrazy.

Kód 12 Nastavení modelu

```
99  model zavada_model/all/;
100 file results /results.txt/;
101 put results;
102 solve zavada_model minimizing t_overall using mip;
103 distance_final = sum(e(i,j,m), x.l(i,j,m) * distance(i,j));
104 put distance_final; put "          ";
105 display distance_final,
x.l,t_overall.l,t_final.l,t.l,w.l,u.l;
```

Klíčové slovo `/all/` označí celý model pro následnou aplikaci smíšeného celočíselného programování. Řádky 100, 101 a 104 nastiňují vkládání výsledků `distance_final` do souboru `results.txt`. Ty jsou získány za pomoci sumarizační funkce obsažené na řádce 103. Poslední řádek slouží k pozdějšímu zobrazení vypsáných proměnných ve výsledcích výpočtu.

### 7.3 Problémy při implementaci a jejich řešení

Virtuální depa, zmíněné v předpokladu podkapitoly 6.3.5., jsou řešením situace, kdy program nezapočítával poslední projetí hranou cesty. S tím souvisí i nesprávný celkový čas a počet ujetých kilometrů. Jejich omezení je znázorněno v matici  $C$ , kdy virtuální uzel je přímo směřován do depa a kde také platí veškerá absence hran mezi všemi ostatními depy. Počet je rovný velikosti použitého vozového parku a do každého z nich cílí pouze jeden vůz.

Implementace proběhla v matici  $C$ , kde byly přidány mezi ostatní adresní body. Seznam všech virtuálních dep je obsažen ve množině  $I_v$ , která posléze vstupuje do rovnice. Jednotlivé časy vozů se poté sečtou v proměnné  $t_{final}$ . Program sice stále nezapočítává poslední projetou hranu mezi normálními a uměle vytvořenými depy, ale díky totožné lokalitě na mapě mají i totožné hrany a výsledný čas je správný.

## 8 VÝSLEDKY

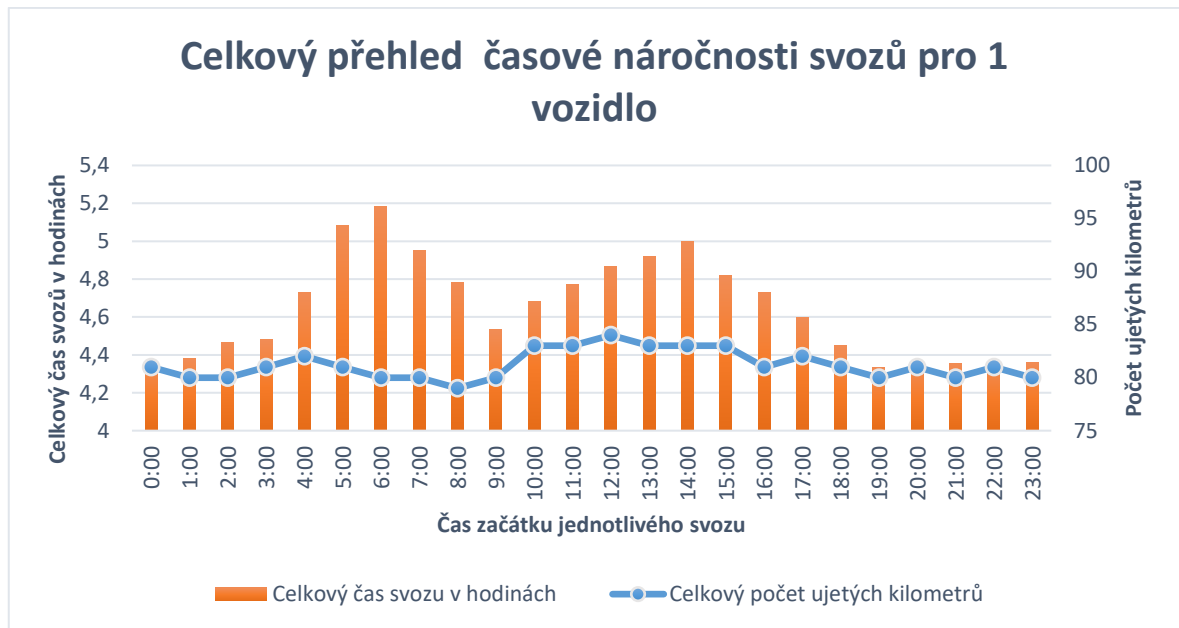
Výsledky z obou měření obsažené v Tabulkách 8 a 9. Ty jsou rozděleny na 2 části a prezentovány formou příložených tabulek, grafů a map. První polovina kapitoly znázorňuje trasu svozu, kdy celkový objem nádob nepřesáhl kapacitu vozu. Naopak druhá polovina se zabývá kombinací dvou vozidel, jelikož celkový objem přesahuje nákladní prostor jednoho vozu a trasa musela být patřičně rozdělena.

### 8.1 Měření pro 1 vozidlo

Tabulka 8 Výsledky výpočtů pro 1 vozidlo

$St$	$l^{final}$	$t^{final}$
0:00	81	4,367
1:00	80	4,383
2:00	80	4,467
3:00	81	4,483
4:00	82	4,727
5:00	81	5,083
6:00	80	5,183
7:00	80	4,95
8:00	79	4,783
9:00	80	4,533
10:00	83	4,683
11:00	83	4,769
12:00	84	4,867
13:00	83	4,917
14:00	83	5
15:00	83	4,817
16:00	81	4,727
17:00	82	4,6
18:00	81	4,45
19:00	80	4,333
20:00	81	4,349
21:00	80	4,352
22:00	81	4,355
23:00	80	4,361

První sloupec  $St$  značí denní čas začátku svozu. Ten byl pro každou hodinu optimalizován zvlášť.  $l^{final}$  představuje konečný počet uražených kilometrů vozidlem a poslední sloupec  $t^{final}$  určuje celkovou dobu trvání svozu dle denní doby. Z hodnot zanesených v tabulce bylo následně sestrojeno grafické znázornění pro lepší přehlednost, představené v Obrázku 8.



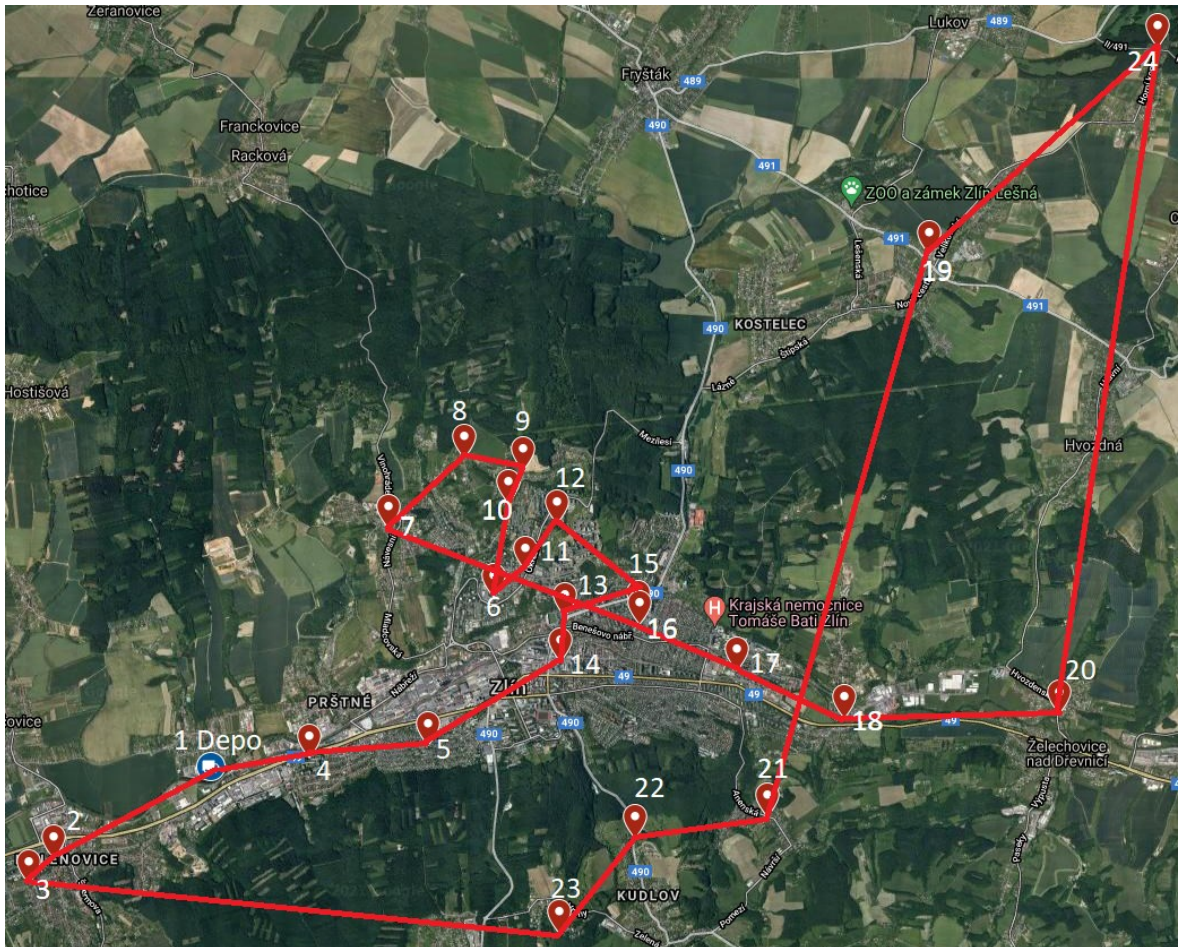
Obrázek 8 Graf přehledu časové náročnosti svozů pro 1 vozidlo

Osa  $x$  je znázorňuje průběh celého dne. Posun je daný vždy přičtením jedné hodiny a značí začátek svozu  $St$ . Vlevo na ose  $y$  je znázorněna časová náročnost svozu v hodinách  $t^{final}$ . Vpravo je umístěn počet ujetých kilometrů za doby trvání svozu  $l^{final}$ .

Z celodenního měření je jasně patrný postupný růst časové náročnosti pro brzké ranní hodiny, způsobeno zhoršenou dopravou a narůstající hustotou řidičů směřující do práce. Následuje strmý pokles, kde se situace uklidňuje. Poslední zvýšení nastává v časech od 10:00 do 14:00, zapříčiněné pozvolným vracením řidičů zpět do domovů. Od 14:00 se časová náročnost svozu snižuje až dosahuje svého minima.

Rozdíl nejnižších a nejvyšších hodnot je 51 minut. To znázorňuje dobu strávenou čekáním, zapříčiněné různými dopravními omezeními, jako je zpomalení dopravní rychlosti či stáních v kolonách. Model vytváří pro daný čas vždy nejvhodnější trasu, aby zdržení a nepříznivou situaci nejvíce eliminoval. Křivka celkového počtu kilometrů není přímo úměrná celkovému času  $t^{final}$  a až na pár odchylek si drží svého optima. Drobné výchylky jsou zapříčiněné různorodostí trasy, kde jde názorně vidět její přizpůsobivost pro různé denní doby je názorně vidět v nadcházejících mapách.

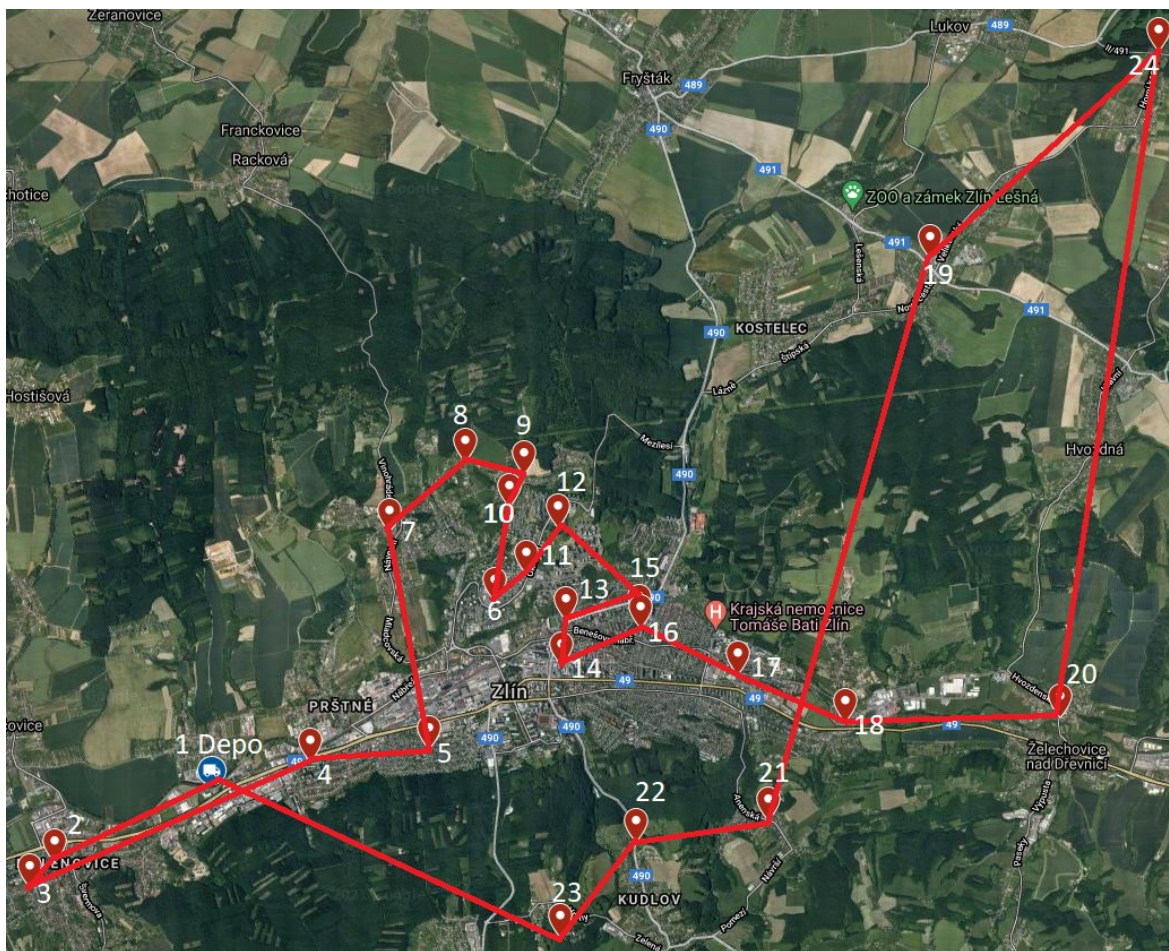




Obrázek 9 Trasa vozidla začínající ve 3:00

Obrázek 9 nastiňuje možnou cestu svozu pro začátek ve 3:00. Začátek se nachází v depu (1), ze kterého vozidlo směřuje tř. Tomáše Bati směrem do města (4, 5). Poté projíždí centrem (14, 13 a 15) a následuje sběr kontejnerů na Jižních svazích (12, 11 a 6). Sjíždí z hlavní ulice a pokračuje okrajovými částmi (10, 9 a 8) směrem na Mladcovou (7). Posléze projíždí střed města (16) směřujíc k nemocnici (17) a na Příluky (18), načež vjíždí do Želechovic (20), projíždí přílehlými vesnicemi: Velikovou (24), Štípou (19), Jaroslavicemi (21) a nakonec Kudlovem (22, 23). Cestu zakončuje projetím Malenovicemi (3, 2) opět v depu nacházejíc se v areálu technických služeb Zlín. Výsledný čas svozu jsou 4 hodiny a 29 minut.

Jelikož ve 3:00 jsou dopravní omezení zanedbatelné, vozidlo směřuje první do centra města, aby po změně časového intervalu v 6:00 projíždělo pouze okrajovými částmi a okolními vesnicemi, kde je časové zdržení zanedbatelné i pro interval s dopravní špičkou.

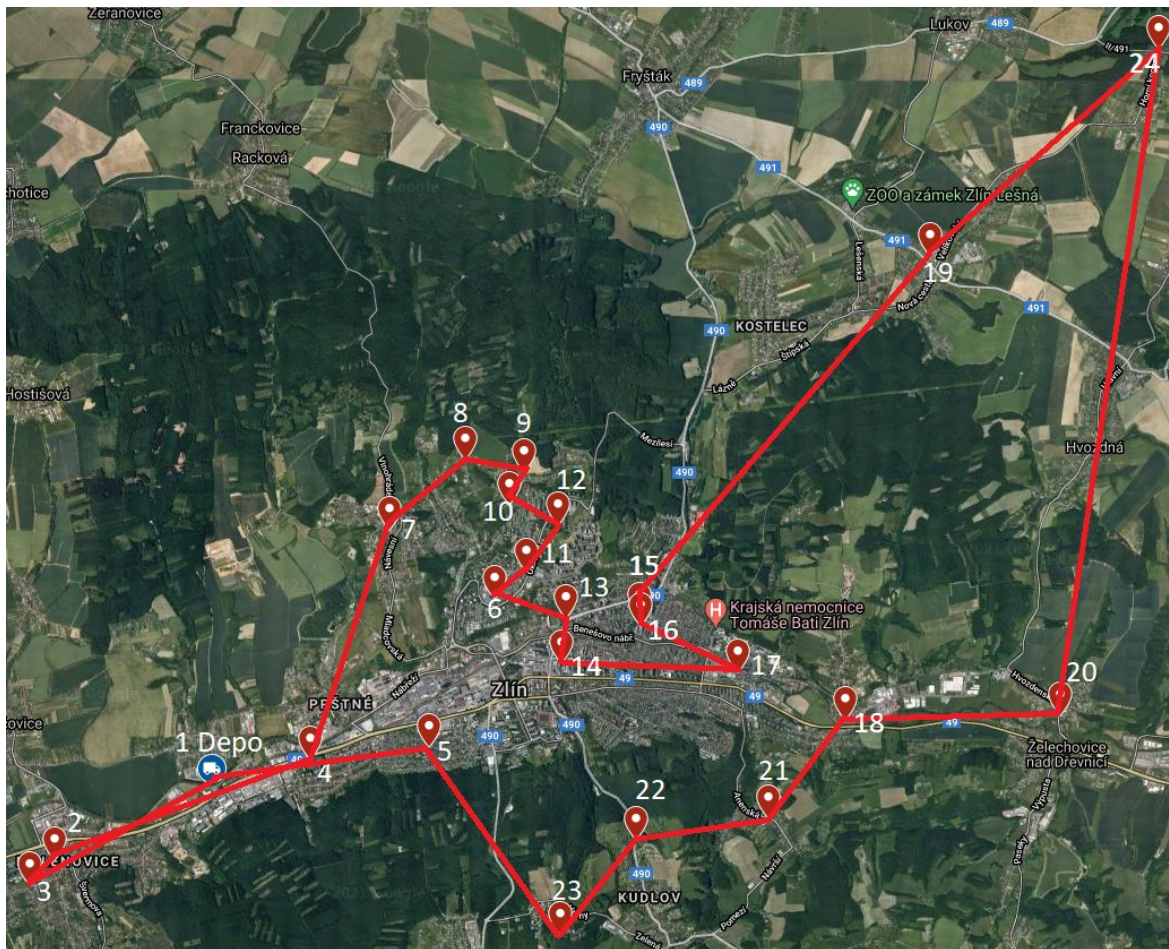


Obrázek 10 Trasa vozidla začínající v 6:00

Hlavní změnou předešlé trasy je změna intervalu. Od 6:00 do 8:00 město sužuje častá dopravní zdržení doprovázející mnohdy dopravními kolonami.

V Obrázku 10 vozidlo z depa (1) jede směrem na Malenovice (2, 3). Po obslužení kontejnerů se vrací po tř. Tomáše Bati (4, 5), načež se odklání od centra a směřuje na Mladcovou (7), projíždí okrajové části (8, 9 a 10) směrem k Jižním svahům (6, 11 a 12). Zde nastává změna časového intervalu, kdy se dopravní situace zlepšuje. Po změně vozidlo pokračuje směrem do centra města (15, 13, 14 a 16), mívá nemocnici (17) a Příluky (18). Sjíždí z města směrem na okolní vesnice: Želechovice (20), Veliková (24), Štípa (19), Jaroslavice (21), Kudlov (22, 23) zakončujíc cestu znovu v depu. Výsledný čas svozu je 5 hodin a 11 minut.

Příklad trasy, kdy vozidlo reaguje na nepříznivé dopravní omezení vyhýbáním se centra města do doby, kdy proběhne změna podmínek a dopravních omezení k lepšímu.



Obrázek 11 Trasa vozidla začínající v 9:00

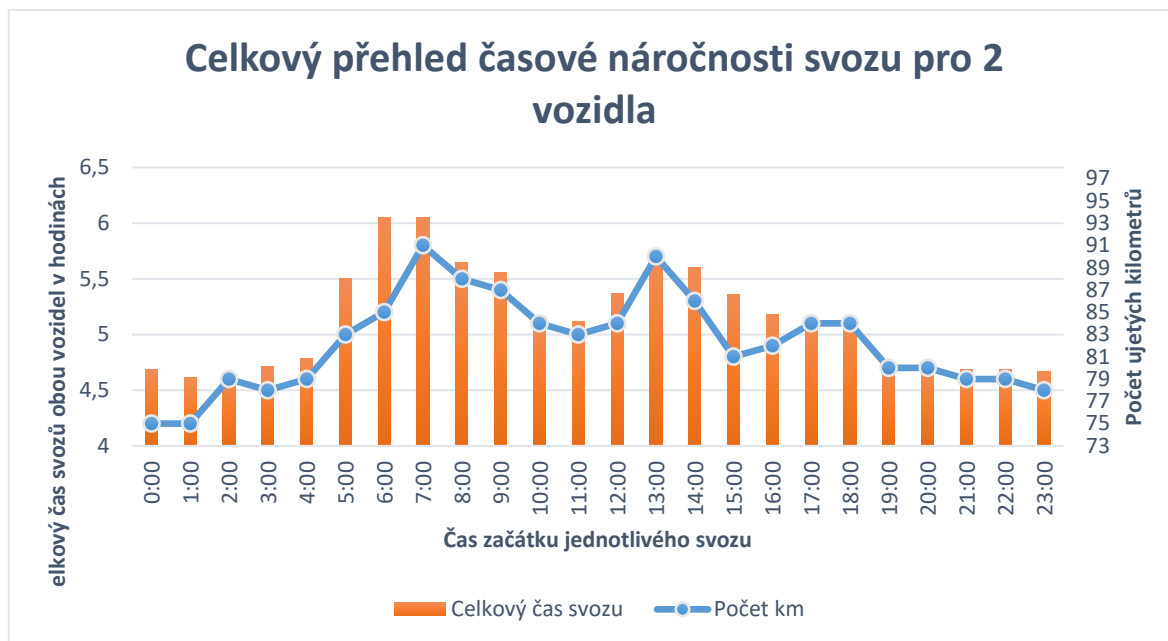
Vozidlo v Obrázku 11 se nachází zlomu 3. a 4. intervalu, to představuje mírnou dopravu. Trasa začínající v depu (1) pokračuje směr Malenovice (3, 2), posléze jako obvykle projíždí tř. Tomáše Bati (4), opouští město směřujíc na Mladcovou (7), okrajové části (8, 9, a 10) pokračujíc Jižní svahy (12, 11 a 6), středu města (13, 14), nemocnice (17). Posléze vyjíždí ulicí Sokolská (16, 15) do okolních vesnic: Štípa (19), Velíková (24), Želechovice (20). Odtud vede jediná cesta pro dokončení zbylých míst skrz Příkladky (18), Jaroslavičice (21), Kudlov (22, 23). Poslední bod před návratem do depa se nachází v obytné části (5). Výsledný čas jsou 4 hodiny 32 minut.

## 8.2 Měření pro 2 vozidla

Tabulka 9 Výsledky výpočtů pro 2 vozidla

<b><i>St</i></b>	<b><i>ř<sub>final</sub></i></b>	<b><i>t<sub>final</sub></i></b>
0:00	75	4,683
1:00	75	4,617
2:00	76	4,65
3:00	78	4,717
4:00	79	4,783
5:00	83	5,5
6:00	85	6,05
7:00	91	6,05
8:00	88	5,65
9:00	87	5,557
10:00	84	5,133
11:00	83	5,117
12:00	84	5,367
13:00	90	5,7
14:00	86	5,6
15:00	81	5,357
16:00	82	5,183
17:00	84	5,133
18:00	84	5,15
19:00	80	4,717
20:00	80	4,696
21:00	79	4,689
22:00	79	4,688
23:00	78	4,67

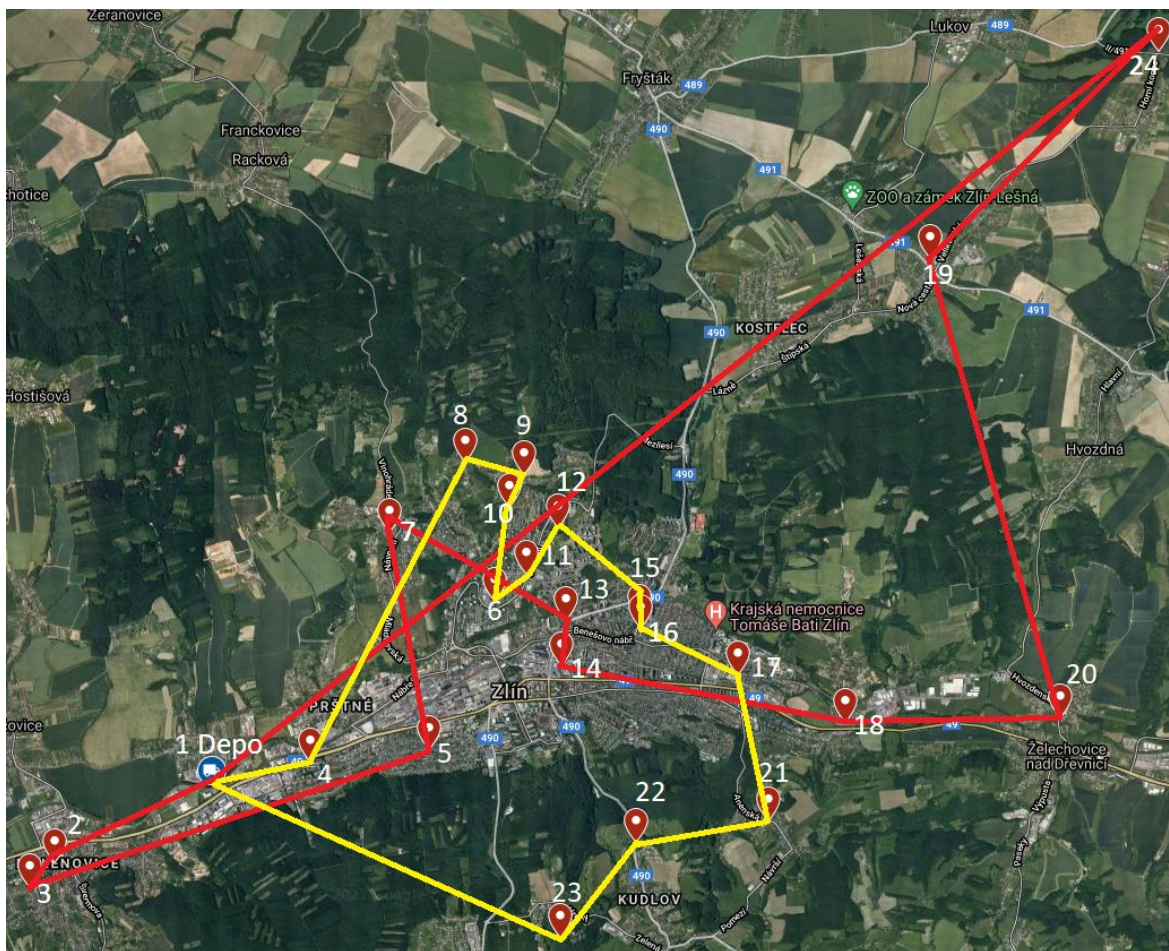
Vstupní parametry byly pro všechny měření nastaveny stejně. Jediný rozdíl činil objem zaplněných kontejnerů. Ten byl v tomto případě zvýšen, kvůli simulaci použití dvou vozidel. Matice hran byla konstruována tak, aby u každého adresního bodu existovalo přímé spojení se všemi ostatními. Tudiž není žádné omezení dopravní infrastruktury a díky velké hustotě dopravních spojení se rapidně zvyšuje výpočetní složitost, ale zároveň se zvyšuje šance k nalezení nejpříhodnějších výsledků.



Obrázek 12 Graf přehledu časové náročnosti svozů pro 1 vozidlo

Graf má podobný průběh jako předešlý v Obrázku 8. Zdržení je nejvyšší pro interval od 5:00 do 9:00 a následně pro polední hodiny od 12:00 do 15:00. Jak již bylo zmíněno, tyto zdržení způsobuje prudký nárůst řidičů v ranních a odpoledních hodinách. Od 16:00 se situace uklidňuje a nastává pozvolný pokles.

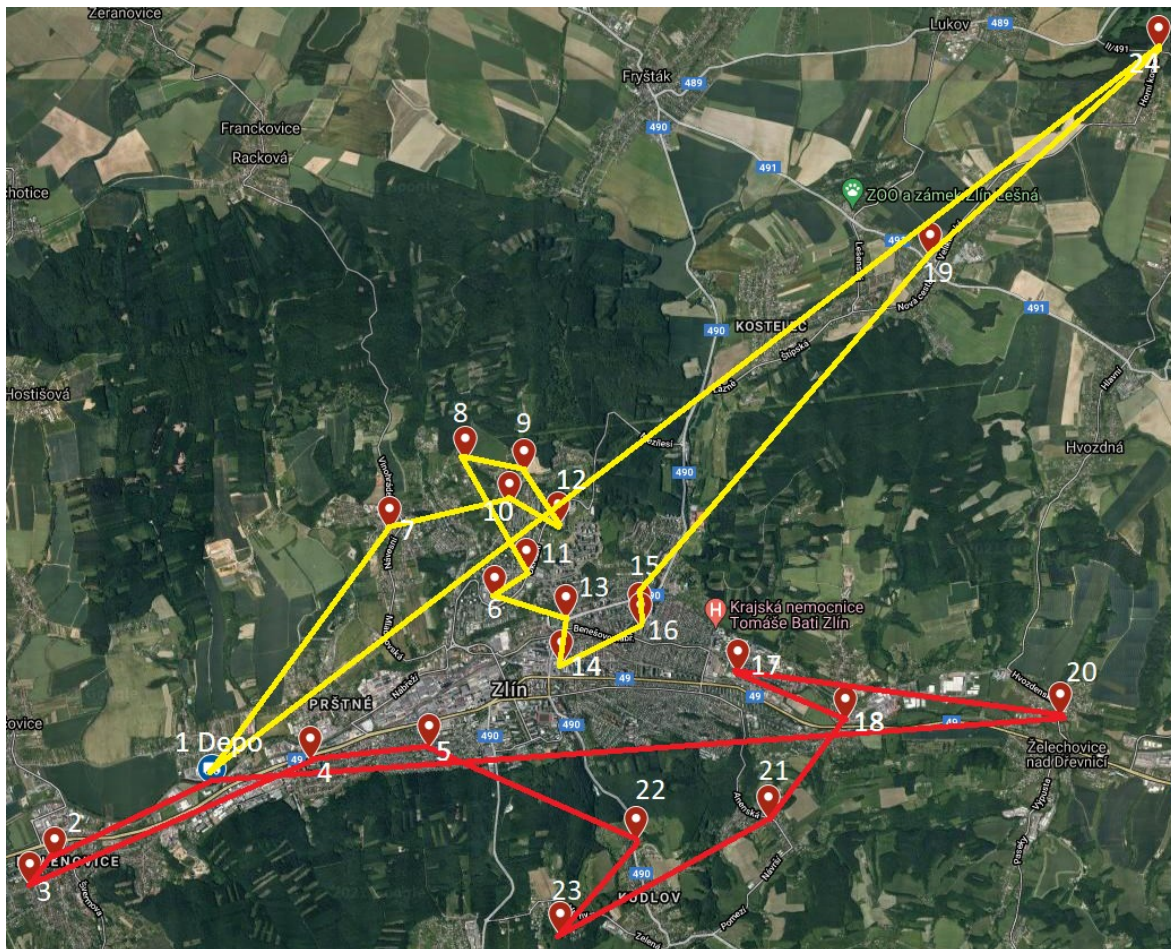
Hlavní rozdíl je ve křivce celkového počtu ujetých kilometrů. Ta je v tomto případě přímo úměrná celkovému času svozu. Rozdíl oproti prvnímu grafu je v samotné výběru cesty. Pro 1 vozidlo trasa mnohdy odpovídala okruhu kolem města Zlína, kdežto rozdělením trasy mezi 2 vozidla vznikají zcela odlišné trasy v závislosti na dopravní situaci v každé hraně mezi příslušnými uzly. Aby bylo docíleno minimalizaci času, model směřuje vozidlo první do okrajových částí města a až poté, po zlepšení dopravy, míří obslužit uzly v centru, jak lze vidět v následujících mapách.



Obrázek 13 Trasa dvou vozidel začínající ve 4:00

Mapa v Obrázku 13 je ukázkou trasy 2 svozových vozidel, odlišené červenou a žlutou barvou. Jelikož výjezd z depa (1) je stanoven na 4:00, podmínky na cestách jsou optimální takřka bez zdržení, žluté vozidlo první směřuje tř. Tomáše Bati (4) do obrátových částech (8, 9 a 10), přejíždějí Jižní Svahy (6, 11 a 12). Následně obchvatem se vrací zpět do města skrz obytnou čtvrť (15, 16 a 17), načez následně projíždí okolní vesnice: Jaroslavice (21) a Kudlov (22, 23), končí v depu.

Červené vozidlo zprvu vjíždí do Malenovic (2, 3), následuje skrz tř. Tomáše Bati do obytné čtvrti (5), projíždí Mladcovou (7), centrem města (13, 14), pokračuje tř. Tomáše Bati na Příluky (18), objíždí okolní vesnice: Želechovice (20), Štípu (19), Velíkovou (24) a končí taktéž v areálu technických služeb Zlín neboli v depu. Výsledný celkový čas obou vozidel jsou 4 hodiny 47 minut.



Obrázek 14 Trasa dvou vozidel začínající v 10:00

Oproti předchozí trase, kdy se vozidla míjela a jejich cesty byly propletené, mapa v Obrázku 14 ukazuje svoz kombinací dvou vozidel, které se v žádném okamžiku cesty neseťkají.

Žluté vozidlo hned z depa (1) vjíždí na Mladcovou (7), směřuje do okrajových částí (10, 12, 9 a 8), načež projíždí Jižními svahy (11, 6), centrem města (13, 14, 16 a 15), objíždí okolní vesnice: Štípu (19) a Velíkovou (24) a trasu zakončuje v depu.

Červené vozidlo naopak směřuje opačným směrem do Malenovic (2, 3), poté se vrací po tř. Tomáše Bati do města (4, 5). Posléze objíždí okolní vesnice: Kudlov (22, 23) a Jaroslavice (21), projíždí Příluky (18) směrem k nemocnici (17) a posledním bodem zůstávají Želechovice (20), načež končí v depu. Výsledný celkový čas obou vozidel je 5 hodin 8 minut.

### 8.3 Nedostatky zvoleného přístupu

Hlavním nedostatkem je neumožnění vozidlu vrátit se do depa a následně pokračovat ve zbývajícím trase. Použitá metoda prikazuje pouze jeden návrat pro každý použitý dopravní prostředek. Při překročení kapacity nákladového prostoru dochází ke zvyšování počtu vozidel, které si trasu náležitě rozdělí. Začínají svoz ve stejný okamžik a poté nezávisle na sobě pokračují v činnosti.

Parametr  $MaxW$  nastavuje maximální kapacitu vozidla. Při použití stejných typů svážejících prostředků je zvolený přístup dostačující a všem je přiřazena stejná hodnota. Komplikace přichází, když vozový park obsahuje různé typy vozidel. Alternativní přístup by mohla být množina, která by obsahovala různé hodnoty  $MaxW$  a umožnila použití víc druhů nákladních aut.

Posledním nedostatkem je použití virtuálních dep. Při větším počtu svozových vozidel jsou data nepřehledná a matoucí. S tím souvisí i zvýšení výpočetní složitosti kvůli zahrnutí více uzlů do množiny  $I$ .

### 8.4 Směry pro budoucí výzkum

Současné data zahrnují časy průjezdů mezi body jednoho konkrétního dne. Výrazné zlepšení by mohlo zaručit zahrnutí všech dní v týdnu, kde by výrazný rozdíl byl mezi všedními dny a víkendem. Zohlednění ročních období, hlavně zimních a letních měsíců by také mohlo přispět k přesnějším výsledkům, protože změna počasí nebo turistické sezóny hrají v dopravě velkou roli.

Dalším znatelným zlepšení modelu by bylo rozdělení dne do více časových intervalů s maximální velikostí do dvou hodin. Současné dělení po 2,4 a 6 hodinách nemusí vždy odpovídat konkrétní situaci, jelikož hodnoty v intervalech jsou vždy průměr ze všech hodin v nich obsažené.

Avšak hlavní budoucí směr je implementace zvoleného přístupu na reálnou aplikaci svozu odpadů za použití reálných dat. S tím souvisí i zahrnutí dalších nákladů do modelu a použití naměřených dat ze stávajících metod. Cílem by bylo porovnat zvolené metody a dosáhnout snížení nákladů za použití vytvořeného přístupu.



## ZÁVĚR

Cílem této bakalářské práce byla minimalizace celkového času svozu prostřednictvím optimalizačního softwaru GAMS za použití správného výběru dat a vhodně zvolených omezení.

Optimalizační metody byly aplikovány na zvolené úloze rozmístěných kontejnerů na elektro odpad ve městě Zlín. Data (časové informace a kilometry mezi uzly) byly získány pomocí Google Maps. Zaplnění kontejnerů bylo navrženo za účelem pokrytí dvou možných situací, které mohou při obsluze kontejnerů nastat. První situací je svoz odpadu jedním vozidlem, kdy celková kapacita je nižší než nákladní prostor vozu. Naopak v situaci, kdy odpad přesáhne celkovou kapacitu bylo potřeba použití více vozidel.

U případu použití 1 vozidla pro svoz odpadu je optimální čas začátku svozu ve 3:00 (Obrázek 9) a 9:00 (Obrázek 11). Pro čas 3:00 je průměrná rychlost je vyšší o 10 km/h oproti času 6:00 a dochází ke snížení celkového času o 42 minut. Naopak pro 9:00 dochází ke zvýšení průměrné rychlosti o 9 km/h, díky tomu se celkový čas svozu se sníží o 39 minut.

V případě zapojení 2 vozidel do svozu odpadu nastává optimální čas svážení u časů 4:00 (Obrázek 13) a 10:00 (Obrázek 14). V prvním případě hodnota průměrné rychlosti oproti vrcholům v 6:00 a 7:00 vzroste o 8 km/h a dojde ke snížení celkového času o 1 hodinu 16 minut. V případě druhého času pro optimální svoz, tedy 10:00 se průměrná rychlost zvýší o 6 km/h a celková doba svozu se sníží o hodnotu 55 minut.

Zvýšení průměrné rychlosti a minimalizaci času na cestách přispívá k nižší spotřebě paliva, poklesu vyprodukovaných emisí a také díky správné optimalizaci trasy svozová vozidla nezatěžují okolní dopravu na hlavních či rušných silnicích. To je zapříčiněno optimálním rozhodování vozidla, ukazujíc Obrázek 10, kdy svozový vůz zprvu směřuje do okrajových částí měst nebo do přilehlých vesnic a do centra města vjíždí až po změně časového intervalu, kdy dochází ke zlepšení dopravní situace a redukování dopravních kolon a jiných nežádoucích omezení.

Pozdější správná implementace reálnými daty pro kapacitu zaplněnosti kontejnerů a s následnými vylepšeními modelu, popsané v kapitole 8.4, model může sloužit k ušetření celkových nákladů za svoz ve městech či v přilehlých obcích.

**SEZNAM POUŽITÉ LITERATURY**

- [1] DAS, Swapan a Bidyut Kr. BHATTACHARYYA. Optimization of municipal solid waste collection and transportation routes. Waste Management [online]. 2015, 43, 9-18 [cit. 2021-4-27]. ISSN 0956053X. Dostupné z: doi:10.1016/j.wasman.2015.06.033
- [2] PIRES, Ana, Graça Martinho, Susana RODRIGUES a Maria Isabel GOMES. Sustainable solid waste collection and management. Springer, 2019. ISBN: 978-3-319-93199-9. Dostupné z doi:10.1007/978-3-319-93200-2
- [3] GHIANI, Gianpaolo, Gilbert LAPORTE a Roberto MUSMANNO. Introduction to logistics systems planning and control. Hoboken, NJ, USA: J. Wiley, c2004. ISBN isbn:0-470-84917-7.
- [4] BOYD, Stephen P. a Lieven VANDENBERGHE. Convex optimization. Cambridge: Cambridge University Press, 2004. ISBN 978-0-521-83378-3.
- [5] SCHRIJVER, Alexander. Theory of linear and integer programming. Chichester: Wiley, c1986. ISBN 978-0-471-98232-6.
- [6] LUENBERGER, David G. a Yinyu YE. Linear and Nonlinear Programming [online]. Cham: Springer International Publishing, 2016 [cit. 2021-4-[24] International Series in Operations Research & Management Science. ISBN 978-3-319-18841-6. Dostupné z: doi:10.1007/978-3-319-18842-3
- [7] GOLDREICH, Oded. P, NP, and NP-completeness: the basics of computational complexity. New York: Cambridge University Press, c2010. ISBN 978-0-521-19248-4.
- [8] DU, Dingzhu a Ker-I KO. Theory of computational complexity. Second edition. Hoboken, New Jersey: John Wiley, 2014. ISBN 978-1-118-30608-6.
- [9] DE FIGUEIREDO, Celina M.H. The P versus NP–complete dichotomy of some challenging problems in graph theory. Discrete Applied Mathematics [online]. 2012, 160(18), 2681-2693 [cit. 2021-4-26]. ISSN 0166218X. Dostupné z: doi:10.1016/j.dam.2010.12.014
- [10] DEDOV, Florian. The Bible of Algorithms and Data Structures: A Complex Subject Simply Explained, 2020 [cit. 28.04.2021]. ISBN 9798678017192.
- [11] ARORA, Sanjeev a Boaz BARAK. Computational complexity: a modern approach. Cambridge: Cambridge University Press, c2009. ISBN 9780521424264.

- [12] LAW, Edward Ko Wah a Winco Kam Chuen YUNG. Optimizing control dynamic complexity and production schedule. *International Journal on Interactive Design and Manufacturing (IJIDeM)* [online]. 2019, 13(1), 47-58 [cit. 2021-4-26]. ISSN 1955-2513. Dostupné z: doi:10.1007/s12008-018-0488-2
- [13] FORTNOW, Lance. The status of the P versus NP problem. *Communications of the ACM* [online]. 2009, 52(9), 78-86 [cit. 2021-4-25]. ISSN 0001-0782. Dostupné z: doi:10.1145/1562164.1562186
- [14] HASLE, Geir, Knut-Andreas LIE a Ewald QUAK, ed. *Geometric Modelling, Numerical Simulation, and Optimization* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007 [cit. 2021-4-26]. ISBN 978-3-540-68782-5. Dostupné z: doi:10.1007/978-3-540-68783-2
- [15] George B Dantzig and John H Ramser. 1959. The truck dispatching problem. *Management science* 6, 1 (1959), 80--91.
- [16] TOTH, Paolo a Daniele VIGO. *The Vehicle Routing Problem* [online]. Philadelphia: Society for Industrial and Applied Mathematics, 2002 [cit. 2021-4-26]. ISBN 978-0-89871-498-2. Dostupné z: doi:10.1137/1.9780898718515
- [17] MUGAYSKIKH, Alexander V., Victor V. ZAKHAROV a Tero TUOVINEN. Time-Dependent Multiple Depot Vehicle Routing Problem on Megapolis Network under Wardrop's Traffic Flow Assignment. In: 2018 22nd Conference of Open Innovations Association (FRUCT) [online]. IEEE, 2018, 2018, s. 173-178 [cit. 2021-4-24]. ISBN 978-952-68653-4-8. Dostupné z: doi:10.23919/FRUCT.2018.8468273
- [18] BEN TICHA, Hamza, Nabil ABSI, Dominique FEILLET a Alain QUILLIOT. Vehicle routing problems with road-network information: State of the art. *Networks* [online]. 2018, 72(3), 393-406 [cit. 2021-4-29]. ISSN 00283045. Dostupné z: doi:10.1002/net.21808
- [19] Chryssi Malandraki, Mark S. Daskin *Time Dependent Vehicle Routing Problems: Formulations, Properties and Heuristic Algorithms*. *Transportation Science* 26 (3) 185-200
- [20] TOTH, Paolo a Daniele VIGO. *Vehicle routing: problems, methods, and applications*. Second edition. Philadelphia: Mathematical Optimization Society, [2014]. ISBN 978-1-611973-58-7.

- [21] ASEKOL » Červené kontejnery. ASEKOL [online]. Copyright © by ASEKOL a.s. 2021. All rights reserved. [cit. 21.04.2021]. Dostupné z: <https://www.asekol.cz/cervene-kontejnery/>
- [22] QIU, Yuzhuo, Liang WANG, Xuanjing FANG, Panos M. PARDALOS a Boris GOLDENGORIN. Formulations and branch-and-cut algorithms for production routing problems with time windows. *Transportmetrica A: Transport Science* [online]. 2018, 14(8), 669-690 [cit. 2021-5-2]. ISSN 2324-9935. Dostupné z: [doi:10.1080/23249935.2018.1427157](https://doi.org/10.1080/23249935.2018.1427157)
- [23] GAMS - Cutting Edge Modeling. GAMS - Cutting Edge Modeling [online]. Dostupné z: <https://www.gams.com/>
- [24] CPLEX Optimizer | IBM. [online]. Copyright © Copyright IBM Corporation 2021 [cit. 21.04.2021]. Dostupné z: <https://www.ibm.com/analytics/cplex-optimizer>
- [25] Data Entry: Parameters, Scalars and Tables. GAMS - Cutting Edge Modeling [online]. Copyright © [cit. 21.04.2021]. Dostupné z: [https://www.gams.com/33/docs/UG\\_DataEntry.html](https://www.gams.com/33/docs/UG_DataEntry.html)

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

LP	Lineární programování
MIP	Mixed integer programming (Smíšené celočíselné programování)
NLP	Nelineární programování
P	Polynomiální
NC	Nick's class
NP	Nedeterministicky polynomiální
VRP	Vehicle routing problem
TDVRP	Time-dependent vehicle routing problem
CVRP	Capacitated vehicle routing problem
GAMS	General Algebraic Modeling System

**SEZNAM OBRÁZKŮ**

Obrázek 1 Vztah množin P, NP a NP – Úplné [12].....	16
Obrázek 2 Příklad řešení trasy pomocí VRP .....	18
Obrázek 3 Přehled závislosti doby jízdy na příslušném času [19] .....	21
Obrázek 4 Graf přehledu zdržení v časových intervalech .....	26
Obrázek 5 Mapa města Zlína s vyznačenými kontejnery a depem.....	27
Obrázek 6 Graf závislosti počtu uzlů na vozidlech .....	30
Obrázek 7 Ukázka prostředí GAMSu .....	36
Obrázek 8 Graf přehledu časové náročnosti svozů pro 1 vozidlo .....	48
Obrázek 9 Trasa vozidla začínající ve 3:00 .....	49
Obrázek 10 Trasa vozidla začínající v 6:00.....	50
Obrázek 11 Trasa vozidla začínající v 9:00.....	51
Obrázek 12 Graf přehledu časové náročnosti svozů pro 1 vozidlo .....	53
Obrázek 13 Trasa dvou vozidel začínající ve 4:00 .....	54
Obrázek 14 Trasa dvou vozidel začínající v 10:00.....	55

**SEZNAM TABULEK**

Tabulka 1 Příklad časové složitosti algoritmů.....	15
Tabulka 2 Rešerše dosavadní vědecké činnosti.....	23
Tabulka 3 Rozdělení časových intervalů.....	25
Tabulka 4 Popis množin.....	28
Tabulka 5 Popis proměnných.....	28
Tabulka 6 Popis binární proměnné.....	28
Tabulka 7 Popis parametrů.....	29
Tabulka 8 Výsledky výpočtů pro 1 vozidlo.....	47
Tabulka 9 Výsledky výpočtů pro 2 vozidla.....	52

## SEZNAM PŘÍLOH

Příloha P I: Obsah CD



## **PŘÍLOHA P I: OBSAH CD**

Obsah CD:

- Bakalářská práce ve formátu .pdf
- Složka GAMS:
  - o Zdrojový kód pro 1 vozidlo – one\_vehicle.gms
  - o Zdrojový kód pro 2 vozidla – two\_vehicles.gms
  - o Projekt pro spouštění zdrojových kódů – project.gpr
  - o Data obsažená v modelu v souborech – BPSvoz.xlsx a parametry.txt