

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

FAKULTA INFORMATIKY A STATISTIKY

Disertační práce

2006

Ing. Jan Fábry

Vysoká škola ekonomická v Praze
Fakulta informatiky a statistiky
katedra ekonometrie

Dynamické okružní a rozvozní úlohy

doktorská disertační práce

Doktorand : Ing. Jan Fábry
Školitel : prof. RNDr. Jan Pelikán, CSc.
Obor : Ekonometrie a operační výzkum

© 2006 Jan Fábry
fabry@vse.cz

při citaci uvádějte odkaz:

Fábry, J.: Dynamické okružní a rozvozní úlohy, disertační práce, VŠE-FIS, Praha, 2006

Praha, listopad, 2006

Prohlášení k disertační práci

Prohlašuji, že doktorskou práci na téma „Dynamické okružní a rozvozní úlohy“ jsem vypracoval samostatně. Použitou literaturu a podkladové materiály uvádím v příloženém seznamu literatury.

V Praze dne 13. listopadu 2006

.....
Podpis

Poděkování

Rád bych touto cestou poděkoval svému školiteli prof. Janu Pelikánovi za cenné rady a připomínky, které přispěly ke zkvalitnění předkládané disertační práce. Zároveň bych chtěl poděkovat prof. Josefu Jablonskému za jeho pomoc a významná doporučení při počítačovém zpracování modelů uvedených v práci.

Věnování

Tuto práci věnuji svým rodičům.

Abstract

Dynamic Vehicle Routing Problems

Existing competitive business forces distribution firms to offer immediate services to their customers. While a static version of routing problems does not accept additional requirements of customers, the dynamic version enables dispatcher to change the planned routes of the vehicles after an occurrence of on-line requests. We focus on Dynamic Traveling Salesman Problem and Dynamic Vehicle Routing Problem. In extensions of these problems we consider multiple vehicles available in a unique depot or multiple depots. Mathematical models are proposed to find the optimum solution of described dynamic problems. The objective is to minimize the total length of all routes or to minimize the service time. Heuristic insertion algorithm is used as an alternative approach to the dynamic problems. Time windows constraints are included in Traveling Salesman Problems together with the cost function considering the travel costs and lateness costs as the penalties in case time windows are violated. We discuss special dynamic distribution problems: Dynamic Messenger Service Problem, A Priori Dynamic Salesman Problem and Dynamic Split Delivery Vehicle Routing Problem. The solution to most of analyzed problems is demonstrated on generated data using LINGO as a solver and VBA in MS Excel as an interface and output environment. The results of computational experiments are presented.

Keywords: dynamic traveling salesman problem, dynamic vehicle routing problem, multiple vehicles, time windows, insertion algorithm

Abstrakt

Dynamisches Vehicle Routing Problem

Das heute vorhandene Wettbewerbsumfeld zwingt Logistikdienstleister dazu, ihren Kunden unmittelbare, möglichst rechtzeitige Bedienung anzubieten. Während ein statischer Ansatz zu Routing Problemen keine zusätzlichen Kundenbedürfnisse berücksichtigt, befähigt eine dynamische Version einen Dispatcher dazu die geplanten Fahrzeugrouten nach Eingang einer gestellten Anfrage zu verändern. Wir konzentrieren uns auf das dynamische Traveling Salesman Problem und das dynamische Vehicle Routing Problem. Als Erweiterung dieser Probleme stellen wir Überlegungen zu mehreren verfügbaren Transportmitteln in einem oder mehreren Depots an. Zur optimalen Lösung dieser dynamischen Probleme werden mathematische Modelle angeboten mit dem Ziel die gesamte Länge aller Routen oder die Bedienungszeit zu minimieren. Ein heuristisches Verfahren in Form eines Insertion Algorithmus wird als alternative Methode zur Lösung dieser dynamischen Probleme angewendet. Nebenbedingungen zu Zeitfenstern werden in Traveling Salesman Problemen im Zusammenhang mit Kostenfunktionen betrachtet, die Transportkosten sowie Kosten für Verspätungen als Konventionalstrafen im Falle einer Verletzung eines Zeitfensters einbezieht. Wir handeln von den speziellen dynamischen Vertriebsprobleme: Dynamische Messenger Service Probleme, dynamische Salesman Probleme mit den a priori Informationen und dynamische Split Delivery Vehicle Routing Probleme. Die Lösungen zu den meisten analysierten Problemen werden anhand generierter Daten mittels des Solvers LINGO und VBA in MS-Excel als Schnittstelle und Outputumgebung aufgezeigt. Die Lösung der kombinatorischen Experimente wird präsentiert.

Schlüsselwörter: dynamisches Traveling Salesman Problem, dynamisches Vehicle Routing Problem, mehrere Transportmittel, Zeitfenster, Insertion Algorithmus

Obsah

1. ÚVOD	8
2. DYNAMICKÉ ROZŠÍŘENÍ OKRUŽNÍCH A ROZVOZNÍCH ÚLOH	11
3. DYNAMICKÁ ÚLOHA OBCHODNÍHO CESTUJÍCÍHO	13
3.1 Standardní úloha obchodního cestujícího	13
3.2 Dynamizace úlohy obchodního cestujícího	14
3.2.1 RE-OPTIMALIZACE DYNAMICKÉ ÚLOHY OBCHODNÍHO CESTUJÍCÍHO	16
3.2.2 VKLÁDACÍ ALGORITMUS PRO DYNAMICKOU ÚLOHU OBCHODNÍHO CESTUJÍCÍHO	20
3.2.3 OKAMŽIKY PŘÍCHODU NOVÝCH POŽADAVKŮ V DYNAMICKÉ ÚLOZE OBCHODNÍHO CESTUJÍCÍHO	21
3.3 Dynamická úloha obchodního cestujícího s časovými okny	24
3.3.1 STATICKÁ ÚLOHA OBCHODNÍHO CESTUJÍCÍHO S ČASOVÝMI OKNY	25
3.3.1.1 Úloha s čekáním vozidla u zákazníka před jeho obsluhou	27
3.3.1.2 Úloha s čekáním vozidla u právě obsluženého zákazníka	28
3.3.2 DYNAMICKÁ ÚLOHA OBCHODNÍHO CESTUJÍCÍHO S ČASOVÝMI OKNY.....	30
3.3.2.1 Re-optimalizace trasy po přidání zákazníka	31
3.3.2.2 Zařazení nového zákazníka pomocí vkládacího algoritmu	32
3.3.3 DYNAMICKÁ ÚLOHA OBCHODNÍHO CESTUJÍCÍHO S ČASOVÝMI OKNY A PENALIZACÍ	33
3.3.3.1 Změna modelu statické úlohy	34
3.3.3.2 Změna modelu dynamické úlohy	35
3.3.4 DYNAMICKÁ ÚLOHA OBCHODNÍHO CESTUJÍCÍHO S APRIORNÍ INFORMACÍ	36
3.3.4.1 Možné strategie přesunu vozidla	38
3.3.4.2 Přesun volného vozidla	39
3.3.4.3 Výpočetní experimenty	40
3.4 Dynamická úloha kurýrní služby	41
3.4.1 STATICKÁ ÚLOHA KURÝRNÍ SLUŽBY BEZ ČASOVÝCH OKEN	41
3.4.2 STATICKÁ ÚLOHA KURÝRNÍ SLUŽBY S ČASOVÝMI OKNY	44
3.4.3 DYNAMICKÁ ÚLOHA KURÝRNÍ SLUŽBY	45
4. DYNAMICKÁ ÚLOHA S VÍCE OBCHODNÍMI CESTUJÍCÍMI	50
4.1 Úloha s více obchodními cestujícími	50
4.2 Aplikace úlohy s více obchodními cestujícími	51
4.3 Úloha s více obchodními cestujícími s minimalizací celkové ujeté vzdálenosti	52
4.3.1 ÚLOHA S JEDNÍM VÝCHOZÍM MÍSTEM	52
4.3.2 ÚLOHA S NĚKOLIKA VÝCHOZÍMI MÍSTY	54
4.4 Úloha s více obchodními cestujícími s minimalizací času potřebného k obsluze všech zákazníků	56
4.4.1 ÚLOHA S JEDNÍM VÝCHOZÍM MÍSTEM	57
4.4.2 ÚLOHA S NĚKOLIKA VÝCHOZÍMI MÍSTY	60
4.5 Úloha s více obchodními cestujícími s časovými okny	62
4.6 Dynamická úloha s více obchodními cestujícími	63
4.6.1 MINIMALIZACE CELKOVÉ VZDÁLENOSTI – VÍCE VÝCHOZÍCH MÍST	64
4.6.2 MINIMALIZACE ČASU POTŘEBNÉHO K OBSLOUŽENÍ VŠECH ZÁKAZNÍKŮ – JEDNO VÝCHOZÍ MÍSTO ...	71
4.6.3 MINIMALIZACE ČASU POTŘEBNÉHO K OBSLOUŽENÍ VŠECH ZÁKAZNÍKŮ – VÍCE VÝCHOZÍCH MÍST....	78
5. DYNAMICKÉ ROZVOZNÍ ÚLOHY	82
5.1 Rozvozní úloha s jedním vozidlem	83
5.1.1 STATICKÁ ÚLOHA	83
5.1.2 DYNAMICKÁ ÚLOHA	85

5.2 Rozvozní úloha s více vozidly v jednom výchozím místě.....	93
5.2.1 MINIMALIZACE CELKOVÉ UJETÉ VZDÁLENOSTI	93
5.2.1.1 Statická úloha	94
5.2.1.2 Dynamická úloha.....	95
5.2.2 MINIMALIZACE ČASU POTŘEBNÉHO K OBSLOUŽENÍ VŠECH ZÁKAZNÍKŮ	97
5.2.2.1 Statická úloha	97
5.2.2.2 Dynamická úloha.....	99
5.3 Rozvozní úloha s více vozidly v několika výchozích místech	100
5.4 Rozvozní úloha s dělenou dodávkou	101
5.4.1 STATICKÁ ÚLOHA	102
5.4.2 DYNAMICKÁ ÚLOHA	104
5.4.3 ROZLOŽITELNOST ÚLOHY S DĚLENOU DODÁVKOU	106
5.4.4 HEURISTICKÁ METODA PRO ÚLOHU S DĚLENOU DODÁVKOU.....	107
6. VÝPOČETNÍ EXPERIMENTY.....	109
6.1 Dynamická úloha obchodního cestujícího	109
6.2 Úloha s více obchodními cestujícími s minimalizací celkové vzdálenosti	110
6.2.1 ÚLOHA S JEDNÍM VÝCHOZÍM MÍSTEM	110
6.2.2 ÚLOHA S NĚKOLIKA VÝCHOZÍMI MÍSTY	111
6.3 Úloha s minimalizací času potřebného k obslužení všech zákazníků.....	113
6.4 Rozvozní úlohy	115
6.5 Shrnutí výpočetních experimentů	116
7. ZÁVĚR	118
PŘÍLOHY	121
Úloha obchodního cestujícího.....	122
Dynamická úloha obchodního cestujícího.....	123
Popis systému TSP. xls	124
Statická úloha s více obchodními cestujícími v jednom výchozím místě.....	129
Statická úloha s více obchodními cestujícími ve více výchozích místech.....	131
Statická úloha s více obchodními cestujícími v jednom výchozím místě.....	133
Statická úloha s více obchodními cestujícími v několika výchozích místech	137
Dynamická úloha s více obchodními cestujícími v několika výchozích místech.....	140
Rozvozní úloha s jedním vozidlem	142
Rozvozní úloha s dělenou dodávkou	147
Obsah příloženého CD	152
TERMINOLOGICKÝ SLOVNÍK.....	153
POUŽITÁ LITERATURA.....	155

1. Úvod

V posledních letech byl zaznamenán obrovský pokrok v oblasti komunikace a informačních technologií. Jedná se především o stále širší používání internetových služeb, rozvoj sítí mobilních operátorů, využití satelitních systémů apod. Současně s tím vznikají vyšší nároky na pružnější reakci firem zajišťujících logistické služby svozu a rozvozu zboží, materiálu či lidí, firem poskytujících kurýrní či opravárenské služby, aj. Úspěšnost a atraktivita firmy pro její zákazníky se v dnešní době odvíjejí nejen od ceny či kvality nabízených služeb, ale především od schopnosti flexibilně reagovat na zákaznickovy požadavky.

Distribuční úlohy, které jsou řešeny použitím standardních modelů a metod operačního výzkumu, používají tzv. **statický přístup**. Informace o všech zákaznících a jejich požadavcích jsou známy předem, tj. před tím, než je úloha předána analytikům k nalezení optimálního řešení. Může se jednat o deterministické či stochastické informace. V reálných situacích však firma musí reagovat i na požadavky, které přicházejí až po nalezení optimálního řešení, v případě okružních a rozvozních úloh optimálních tras. Předmětem tzv. **dynamického přístupu** se stává rozhodnutí o tom, kdy a kdo (v případě použití více vozidel) nově vzniklý požadavek obslouží.

Vzhledem k tomu, že většina podobných úloh patří do celočíselného, resp. bivalentního programování, naráží analytik na problém s výpočetním časem, který má k dispozici pro nalezení optimálního řešení. **Statické** okružní a rozvozní problémy patří do skupiny NP-obtížných úloh (Pelikán, 2001), a tudíž mezi tyto úlohy patří i jejich **dynamické** rozšíření. S rostoucím počtem zákazníků a omezení, která je v problému nutné respektovat, narůstá výpočetní čas neúměrně praktickým požadavkům na rychlou reakci firmy. Proto se v úlohách tohoto typu často používají heuristické a metaheuristické postupy poskytující řešení, které je z hlediska zadaného kritéria velice blízké optimálnímu řešení, a mající tu výhodu, že takové řešení poskytují v podstatě okamžitě, resp. s přípustným zpožděním po vzniku nového požadavku. V běžné praxi jsou analytici či řešitelé zastoupeni dispečery, kteří k nalezení výhodnějšího řešení využívají navržených postupů v podobě speciálních výpočetních systémů.

Tato práce je zaměřena na vybranou část distribučních úloh a jejich dynamické rozšíření, přičemž za základ bere **úlohu obchodního cestujícího** v mnoha variacích. Na tyto úlohy pak navazují **rozvozní úlohy**, ve kterých je nutné respektovat kapacitu vozidla, resp.

vozidel. Ve všech typech uvažovaných úloh jde především o formulaci matematického modelu, nikoli o nalezení nejlepší či nejvhodnější metody pro vyřešení problému. Proto je v práci u každé úlohy formulován optimální algoritmus, resp. **optimalizační matematický model**, vedle kterého je uveden i heuristický postup v podobě **vkładacího algoritmu** jako poměrně jednoduché východisko pro aplikaci modelů v reálné praxi.

Uvedené modely byly ověřeny na generovaných příkladech za použití optimalizačního systému **LINGO**¹, verze 9.0 a aplikace **MS Office Excel 2003**. U jednodušších modelů bylo pro úlohy vytvořeno uživatelské rozhraní spolu s grafickým výstupem řešení, tj. grafem znázorňujícím trasy či jejich dynamické změny. K tomuto účelu bylo využito modulu **VBA** v **MS Excel**. Cílem práce není v žádném případě vytvořit kompaktní výpočetní systém zahrnující veškeré úlohy uvedené v práci; jde spíše o naznačení směru, kterým by se mohla ubírat práce zkušeného programátora.

Druhá kapitola práce je **úvodem do problematiky okružních a rozvozních úloh**, přičemž jsou definovány pojmy statická a dynamická úloha. Čtenáři je předložen stručný přehled literatury týkající se zkoumaných distribučních úloh.

Třetí kapitola je věnována **standardní úloze obchodního cestujícího** a jejímu **dynamickému rozšíření**. V této části jsou formulovány úlohy bez časových oken i s časovými okny a případnou penalizací za jejich nedodržení. Speciální aplikací je úloha obchodního cestujícího s apriorní informací zahrnující známé pravděpodobnostní rozdělení místa vzniku nového požadavku. Na závěr kapitoly je řešena úloha kurýrní služby.

Čtvrtá kapitola se týká rozšíření úlohy **obchodního cestujícího**, ve které je uvažováno **více obsluhujících vozidel** umístěných v jednom výchozím místě či několika různých stanovištích. Uvedené matematické modely jsou formulovány s cílem minimalizovat celkovou ujetou vzdálenost nebo obsloužit všechny zákazníky v minimálním čase.

V **páté kapitole** jsou analyzovány **rozvozní úlohy**, v nichž je cílem splnit požadavky zákazníků na dovoz či odvoz určitého množství zboží, materiálu, odpadu apod. K dispozici je buď jedno nebo několik vozidel o kapacitě, která nesmí být překročena. Zvláštní pozornost je věnována rozvozním úlohám, ve kterých je možné rozdělit požadavek zákazníka mezi několik vozidel, což je v dynamickém prostředí velmi výhodné.

¹ Optimalizační systém LINGO je produktem firmy Lindo Systems Inc. (<http://www.lindo.com>).

Šestá kapitola obashuje vybrané výpočetní experimenty s modely a metodami pospanými v této práci. Při výpočetně náročnějších experimentech byl pro porovnání se systémem **LINGO** použit systém **XPRESS MP**², release 2005.

Práce je prvním uceleným odborným textem v českém jazyce, který se zabývá dynamickým pohledem na okružní a rozvozní úlohy v takovém rozsahu a zahrnuje přístup vyhovující potřebám praxe. Text navazuje na práce Brezina (2003) a Janáček (2003), jejichž autoři se dlouhodobě zabývají distribučními a logistickými úlohami.

Cíle předkládané práce lze shrnout do následujících bodů:

1. Podat přehled o základních typech okružních a rozvozních úloh z pohledu jejich dynamizace a přiblížit tak modelový přístup reálným požadavkům distribučních firem.
2. Navrhnout řešení dynamických distribučních úloh v podobě optimalizačních algoritmů. Pro účely praxe naznačit možnost řešení úloh pomocí heuristických metod.
3. Vytvořit základ softwarového systému pro řešení praktických aplikací i pro provádění simulačních experimentů s navrženými modely.
4. Naznačit další cesty analýzy distribučních úloh. Tuto práci lze brát jako východisko pro další odborné texty typu diplomových či disertačních prací. Jedná se především o vývoj rychlých heuristických postupů, které jsou nepostradatelné především v dynamickém prostředí, ve kterém čas hraje zřejmě nejdůležitější roli.

Jak bylo zmíněno, řada matematických modelů uvedených v práci není softwarově zpracována v požadované podobě z důvodu vysokých nároků na programátorskou práci. Sestavení dokonalého obecného softwarového systému, který by byl schopen pojmout většinu úloh analyzovaných v práci, není z praktického hlediska účelné. Tento fakt dokládá především rozmanitost distribučních firem a jejich zákazníků. Jinou otázkou je ovšem vytvoření podobného systému za účelem zkvalitnění výukového procesu pro snadnější představu a pochopení poměrně složitých přístupů k řešení úloh. Tento zajímavý úkol čeká na budoucí autory.

² XPRESS MP je produktem firmy Dash Optimization Inc. (<http://www.dashoptimization.com>).

2. Dynamické rozšíření okružních a rozvozních úloh

U statických okružních a rozvozních úloh předpokládáme, že všechny parametry úlohy jsou dány předem a nelze je během realizace naplánovaných tras měnit. Oproti tomu dynamické úlohy obsahují ve většině případů množinu parametrů známých do určitého časového okamžiku (začátku realizace trasy, resp. tras), k nimž postupně přibývají další parametry týkající se nově vzniklých požadavků.

Dynamické okružní a rozvozní úlohy se staly předmětem výzkumu především v posledních letech. Z hlediska pozdějšího zkoumání jsou významné především práce Psaraftis (1988, 1995), Powell et al. (1995) a Lund et al. (1996). Na dalším vývoji metod řešících dynamické rozvozní úlohy se podíleli Gendreau and Potvin (1998), kteří předložili přehled základních aplikací využívajících možnosti vzniku objednávek on-line. Gendreau et al. (1999) zkoumali dynamickou rozvozní úlohu s časovými okny na příkladu kurýrních služeb, zajišťujících požadavky zákazníků, které je zapotřebí navštívit během časového okna, vyzvednout připravenou zásilku a dovézt na požadované místo, případně do centra, kde se zásilky shromažďují pro další zpracování, případně hromadný rozvoz (např. služba DHL).

Jak bylo uvedeno v úvodu práce, dynamické okružní a rozvozní úlohy patří do třídy NP-obtížných úloh¹. Pro řešení rozsáhlejších úloh, časově velmi náročných, lze použít řadu heuristických postupů, které poskytují poměrně kvalitní řešení v akceptovatelném čase. V uvedené práci autorů Gendreau et al. (1999) byla použita paralelní metoda tabu search. Pro statickou úlohu popsali tento heuristický algoritmus Badeau et al. (1997) a Taillard et al. (1997). Montemanni et al. (2002) použili na dynamickou rozvozní úlohu metaheuristický algoritmus² založený na chování mravenčí kolonie (tzv. Ant Colony System). V citované práci autoři aplikovali algoritmus na množinu náhodně vygenerovaných 50 míst v městě Lugano ve Švýcarsku. Gambardella et al. (2003) představili dva softwarové produkty vycházející ze zmíněného algoritmu. Použitím genetického algoritmu (Genetic Algorithm) a metody založené na principu simulovaného žíhání (Simulated Annealing) v rozvozní úloze s časovými okny se zabývají Thangiah et al. (1994) a Bräysy and Gendreau (2005b). Aplikaci genetického algoritmu na tento typ úlohy popisují také Homberger and Gehring (1999), Potvin and Bengio (1996) a Čičková (2005).

¹ Definici NP-obtížných, resp. NP-těžkých úloh uvádí Pelikán (2001).

² Metaheuristické algoritmy jsou obecné heuristické postupy aplikovatelné na obecný optimalizační problém, zatímco algoritmy označované jako heuristické jsou speciální metody formulované pro konkrétní úlohu.

V dnešní době jsou předmětem zájmu alternativní optimalizační postupy pro řešení časově náročných úloh, založené na metodě větvení a řezů či generování sloupců. Metoda větvení a řezů³ pro statickou rozvozní úlohu je popsána v práci Bard et al. (2002) a Achuthan et al. (2003). Pro statickou rozvozní úlohu s časovými okny použili metodu generování sloupců⁴ Desrochers et al. (1992), pro dynamickou úlohu Chen and Xu (2006).

Na základě reálných aplikací vznikla celá řada speciálních modelů, které jsou rozšířením původní dynamické rozvozní úlohy s časovými okny, jako je například stochastická rozvozní úloha, tj. rozvozní úloha se stochastickou poptávkou, kterou rozpracovali Gendreau et al. (1996). Stejní autoři také navrhli další rozšíření metod na situace, kdy lze např. využít dostupné informace o budoucích požadavcích. Larsen et al. (2004) provedli řadu zajímavých simulačních experimentů pro úlohu obchodního cestujícího s apriorní informací v podobě pravděpodobnosti vzniku požadavku v určitém regionu.

Další rozsáhlý okruh v oblasti okružních úloh tvoří problémy s více obchodními cestujícími, kteří obsluhují zákazníky z jednoho či několika výchozích míst. Svě výpočetní zkušenosti shrnuli Svestka and Huckfeldt (1973). V počátcích výzkumu se zapsal výrazně Gavish (1976), významná je především pozdější práce autorů Gavish and Srikanth (1986), v níž navrhli metodu pro nalezení optimálního řešení pro rozsáhlé výše zmíněné úlohy. Podobně Laporte and Nobert (1980) navrhli pro řešení této úlohy algoritmus založený na metodě řezných nadrovin. Popis systému DRIVE⁵ založeného na metodě větvení a oceňování⁶ podali Savelsbergh and Sol (1998).

V reálných rozvozních úlohách velmi často existuje možnost rozdělení dodávky zákazníkovi mezi několik vozidel, resp. mezi více tras. Touto problematikou se zabývají Archetti et al. (2001), Dror and Trudeau (1989), Dror et al. (1994), Fábry (2005), aj. Metodu tabu search aplikovali na tuto úlohu Archetti et al. (2006).

Tato práce ve vztahu k vytčeným cílům je koncipována tak, že u každé úlohy je nejprve uvedena její statická verze a potom její dynamické rozšíření, přičemž je kladen důraz na logickou návaznost úvah a přehlednost textu. Definice dynamických modelů jsou ve většině případů původní, mají však svůj základ ve formulacích statických modelů v citovaných zdrojích.

³ Branch-and-Cut Algorithm.

⁴ Column Generation Algorithm.

⁵ DRIVE je zkratkou názvu Dynamic Routing of Independent VEHicles.

⁶ Metoda větvení a oceňování (Branch-and-Price Algorithm) využívá metody generování sloupců.

3. Dynamická úloha obchodního cestujícího

3.1 Standardní úloha obchodního cestujícího

Statická úloha obchodního cestujícího (TSP)¹ předpokládá znalost všech parametrů před započítáním optimalizace i před samotnou realizací okružní jízdy. Miler-Tucker-Zemlinova formulace matematického modelu (Pelikán, 2001) je následující:

$$\text{minimalizovat } z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}, \quad (3.1)$$

za podmínek

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n, \quad (3.2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n, \quad (3.3)$$

$$u_i - u_j + n x_{ij} \leq n - 1, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j, \quad (3.4)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n, \quad (3.5)$$

kde n je počet míst, která vozidlo musí projet (včetně výchozího místa označeného indexem 1) a c_{ij} představuje vzdálenost mezi místy i a j . Proměnná x_{ij} je bivalentní proměnná nabývající hodnoty 1 v případě, že vozidlo jede do místa j z místa i , resp. hodnoty 0 v opačném případě. Podmínky (3.2) a (3.3) zajišťují, že každé místo je navštíveno právě jednou. Soustava podmínek (3.4) s proměnnými u_i je opatřením proti vytváření parciálních cyklů.

V celé disertační práci budeme z důvodu zjednodušení všech matematických modelů předpokládat, že proměnné x_{ij} jsou definované i pro $i = j$. V opačném případě bychom ve všech účelových funkcích, omezeních a matematických výrazech museli připojit podmínku $i \neq j$. Tento postup by byl jistě korektní, ovšem na úkor přehlednosti.

Úloha obchodního cestujícího je nejjednodušší verzí okružních úloh. Cílem je navštívit každého zákazníka, a to právě jednou, aniž bychom se zabývali typem či velikostí požadavků jednotlivých zákazníků. Z tohoto důvodu není třeba řešit ani otázku kapacity

¹ Traveling Salesman Problem.

vozidla. Reálným rozšířením úlohy je zavedení časových oken, tedy časového intervalu, během kterého je nutné daného zákazníka navštívit a obsloužit. Tyto úlohy budou předmětem analýzy v další části práce.

3.2 Dynamizace úlohy obchodního cestujícího

Uvedený statický model úlohy obchodního cestujícího (3.1) - (3.5) předpokládá znalost všech parametrů před započítáním okružní jízdy. V průběhu jízdy nelze tyto hodnoty měnit, nelze tudíž ani přidávat či ubírat zákazníky. Naproti tomu v dynamické úloze obchodního cestujícího (DTSP)² může kdykoli během realizace jízdy přijít nový požadavek od zákazníka nacházejícího se kdekoli v předem ohraničeném území, resp. od kohokoli z předem známé množiny potenciálních zákazníků. V praxi to znamená, že dispečer přijme nový požadavek a rozhodne o jeho zařazení do předem naplánovaného okruhu, resp. zbývajících částí okruhu, kterou vozidlo ještě musí absolvovat.

Získání a zpracování informace o nových požadavcích nevyžaduje v podstatě žádné dodatečné náklady, zatímco přínos pro tímto způsobem fungující firmu může být velice významný, a to nejen z hlediska samotného získání dalších zákazníků zařazených do okruhu, ale především z hlediska celkové úspěšnosti firmy na konkurenčním trhu. Fábry (2006) popisuje dva základní přístupy k zařazení nově vzniklých požadavků do předem naplánované trasy vozidla.

Re-optimalizace

Jakmile se objeví nový zákazník, dispečer jej zařadí mezi zákazníky, které vozidlo ještě nenavštívilo, a nalezne optimální trasu. Jedná se o sofistikovaný přístup, jehož cílem je najít nejlepší řešení s využitím veškerých informací, které má dispečer v daný okamžik k dispozici. Tato metoda je samozřejmě výhodná vzhledem k minimalizaci účelové funkce představující celkovou ujetou vzdálenost potřebnou k obsluze všech zbývajících zákazníků. Jestliže je počet zákazníků vysoký, naráží dispečer opět na problém s výpočetní náročností úlohy. K podobným problémům může dojít i tehdy, bude-li frekvence vzniku nových požadavků tak vysoká, že častá re-optimalizace povede až k úplnému zahlcení systému, kdy během dne nebude možné uspokojit všechny požadavky.

Tento přístup využili Bell et al. (1983), Hill et al. (1988), Brown et al. (1987), Psaraftis et al. (1980, 1983), Powell et al. (1988), Dial (1995) a další.

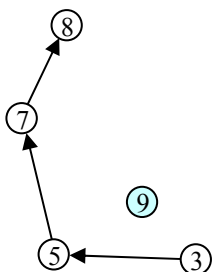
² Dynamic Traveling Salesman Problem.

Vkládací algoritmus

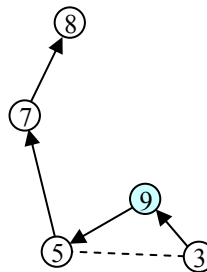
Pokud přijde nový požadavek, zákazník je zařazen do plánované trasy mezi dva po sobě následující zákazníky, kteří mají být vozidlem na trase navštíveni. Pro nalezení nejvhodnější dvojice zákazníků lze použít jako kritérium hodnotu prodloužení stávající trasy po zařazení nového zákazníka, která se bude minimalizovat:

$$\Delta z = c_{i_k, n+1} + c_{n+1, i_l} - c_{i_k, i_l}, \quad (3.6)$$

kde i_k a i_l jsou indexy všech bezprostředně po sobě následujících zákazníků, které má vozidlo navštívit podle plánované trasy, a $n+1$ je index nového zákazníka. Vkládací algoritmus je heuristickou metodou, která nemusí poskytovat optimální řešení. Na obr. 3.1 a 3.2 je uveden příklad vložení zákazníka č. 9 do trasy mezi zákazníky č. 3 a 5.



Obr. 3.1 – Vznik nového požadavku



Obr. 3.2 – Vložení nového požadavku

Vzhledem k tomu, že je možné určit na trase místo, kam má být nový zákazník vložen, prakticky okamžitě i při větším počtu zákazníků, lze tento postup aplikovat především v těch situacích, v nichž hraje významnou roli čas a firma se musí rozhodnout ihned po přijetí nového požadavku (Lund et al., 1996).

Vkládání zákazníků je výhodné zejména při vysoké frekvenci vzniku nových požadavků, ovšem pouze z hlediska času nutného pro nalezení nové trasy, nikoliv však již z hlediska jejího celkového prodloužení. Z tohoto důvodu je vhodné v praxi kombinovat oba přístupy, tj. po určité době, během níž se několikrát aplikoval vkládací algoritmus, se provede re-optimalizace trasy. Další možností je aplikovat re-optimalizační algoritmus na množinu několika nových požadavků, aniž by se pro každého z nich prováděl vkládací algoritmus.

Použití metod, případně jejich kombinace, závisí na charakteru konkrétní firmy. Jednotlivé přístupy lze samozřejmě měnit i během dne podle výkyvů frekvence vzniku nových požadavků.

Uvedené metody předpokládají dynamický, navíc i stochastický charakter vzniku požadavků, neuvažují však možnost předpovídat vlastnosti požadavku na základě pravděpodobnosti jeho vzniku z hlediska časového ani prostorového. Touto otázkou se zabývá část 3.3.4.

3.2.1 Re-optimalizace dynamické úlohy obchodního cestujícího

Jak bylo uvedeno v předchozí části, re-optimalizace trasy spočívá v nalezení optimálního řešení úlohy, ve kterém vozidlo po přijetí nového požadavku obslouží všechny dosud nenavštívené zákazníky a ujede přitom nejkratší vzdálenost. V tomto případě se již nejedná o klasickou úlohu obchodního cestujícího, neboť cílem není nalézt nejkratší okruh, ale nejkratší cestu ze stávajícího uzlu do výchozího uzlu č. 1 takovou, na níž bude každý zákazník navštíven právě jednou. Matematický model této úlohy lze pak formulovat následujícím způsobem:

$$\text{minimalizovat } z = \sum_{i \in U_N} \sum_{j \in U_N} c_{ij} x_{ij}, \quad (3.7)$$

za podmínek

$$\sum_{\substack{j \in U_N \\ j \neq j_{next}}} x_{ij} = 1, \quad i \in U_N - \{1\}, \quad (3.8)$$

$$\sum_{\substack{i \in U_N \\ i \neq 1}} x_{ij} = 1, \quad j \in U_N - \{j_{next}\}, \quad (3.9)$$

$$u_i - u_j + |U_N| \cdot x_{ij} \leq |U_N| - 1, \quad i \in U_N, \quad j \in U_N - \{j_{next}\}, \quad i \neq j, \quad (3.10)$$

$$x_{ij} \in \{0,1\}, \quad i, j \in U_N, \quad (3.11)$$

kde U_N je množina míst, která musí vozidlo ještě navštívit³, $|U_N|$ je jejich počet. Index $j_{next} \in U_N$ odpovídá místu, k němuž vozidlo směřuje v okamžiku přijetí nového požadavku, a které se tak stává výchozím místem pro následující trasu⁴. Pro řešení této úlohy lze použít i modifikovaný matematický model TSP:

$$\text{minimalizovat } z = \sum_{i \in U_N} \sum_{j \in U_N} c_{ij} x_{ij}, \quad (3.12)$$

³ Jako poslední bude navštíveno výchozí místo, tj. uzel č. 1.

⁴ Předpokládáme, že re-optimalizace bude dokončena dříve než bude obsloužen tento zákazník. V opačném případě by výchozím místem pro novou trasu byl až zákazník bezprostředně následující po j_{next} v původní trase.

za podmínek

$$\sum_{j \in U_N} x_{ij} = 1, \quad i \in U_N, \quad (3.13)$$

$$\sum_{i \in U_N} x_{ij} = 1, \quad j \in U_N, \quad (3.14)$$

$$u_i - u_j + |U_N| \cdot x_{ij} \leq |U_N| - 1, \quad i \in U_N, \quad j \in U_N - \{i\}, \quad i \neq j, \quad (3.15)$$

$$x_{1j_{next}} = 1, \quad (3.16)$$

$$x_{ij} \in \{0,1\}, \quad i, j \in U_N. \quad (3.17)$$

Tento model předpokládá, že v původní matici vzdáleností se provede změna vzdálenosti $c_{1j_{next}}$ mezi výchozím místem č. 1 a zákazníkem j_{next} , ke kterému vozidlo právě směřuje, na hodnotu odpovídající délce dosud absolvované trasy⁵ z výchozího místa do místa j_{next} .

Na situaci popsanou uvedeným modelem lze pohlížet tak, že se hledá nejkratší okruh z výchozího místa přes všechny dosud nenavštívené zákazníky, přičemž vozidlo jede zdánlivě z výchozího místa č. 1 do místa j_{next} , což zaručuje podmínka (3.16). Je zapotřebí si uvědomit, že tato změna je provedena jen pro účely výpočtu, nikoli pro samotnou realizaci trasy vozidla, které ve skutečnosti nejede do místa j_{next} z výchozího místa, ale z místa, které navštívilo jako poslední. V účelové funkci (3.12) je zahrnuta i konstanta $c_{1j_{next}}$, což nemá samozřejmě vliv na samotnou optimalizaci.

Příklad 3.1

Tabulka 3.1 obsahuje vygenerované kartézské souřadnice $[X, Y]$ výchozího místa (č. 1) a šesti zákazníků (č. 2-7). Hodnoty jsou uvedeny v kilometrech. Požadavky zákazníků v místech č. 2-6 byly známy předem, požadavek posledního zákazníka v místě č. 7 přišel v okamžiku, v němž již bylo obsluhující vozidlo na trase. Tabulka 3.2 obsahuje euklidovské vzdálenosti c_{ij} mezi každou dvojicí míst i, j .

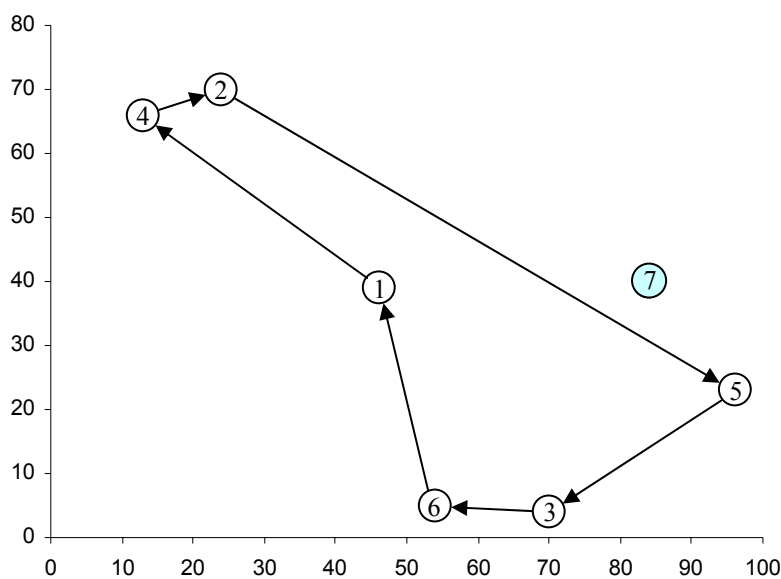
Nejprve řešíme statický model úlohy obchodního cestujícího TSP (3.1) - (3.5) pro všechny předem známé požadavky. Pro optimalizaci byl použit program LINGO; model zapsaný v tomto systému je uveden v příloze č. 1. Optimální trasa, jejíž délka činí přibližně 223,49 km, je znázorněna na obr. 3.3.

⁵ V této trase je zahrnutý i zákazník j_{next} .

Místo	X	Y
1	46	39
2	24	70
3	70	4
4	13	66
5	96	23
6	54	5
7	84	40

Tab. 3.1 – Generovaná úloha DTSP

c_{ij}	1	2	3	4	5	6	7
1	0	38,01	42,44	42,64	52,50	34,93	38,01
2	38,01	0	80,45	11,70	85,98	71,59	67,08
3	42,44	80,45	0	84,22	32,20	16,03	38,63
4	42,64	11,70	84,22	0	93,48	73,50	75,61
5	52,50	85,98	32,20	93,48	0	45,69	20,81
6	34,93	71,59	16,03	73,50	45,69	0	46,10
7	38,01	67,08	38,63	75,61	20,81	46,10	0

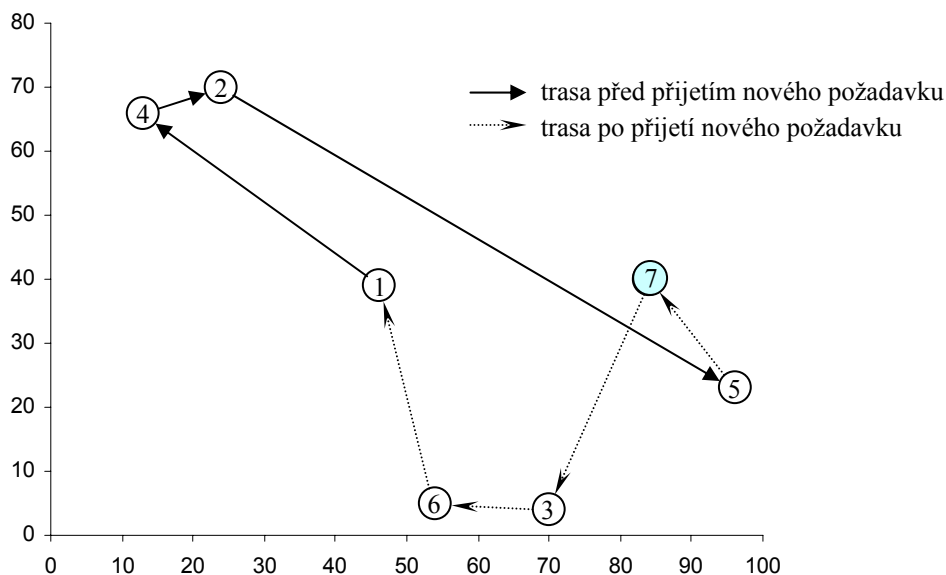
Tab. 3.2 – Symetrická matice vzdáleností⁶ (v km)

Obr. 3.3 – Optimální řešení statického modelu TSP

Pro zařazení nového zákazníka v místě č. 7 do stávající trasy je rozhodující, ve kterém okamžiku jeho požadavek přichází. Předpokládejme, že obsluhující vozidlo se právě nachází mezi místy č. 2 a 5. Množina míst, která mají být vozidlem ještě navštívena, je

⁶ Hodnoty jsou zaokrouhleny na dvě desetinná místa.

$U_N = \{1,3,5,6,7\}$, tj. $|U_N| = 5$. Než dojede vozidlo do místa č. 5, je vypočtena optimální trasa začínající v tomto místě a končící ve výchozím místě č. 1. Trasa vede postupně přes místa č. 7, 3 a 6, jak je patrné z obr. 3.4. Délka této trasy činí cca 110,39 km, délka trasy 1-4-2-5 je cca 140,33 km, vozidlo tedy ujede celkem přibližně 250,72 km.



Obr. 3.4 – Re-optimalizace trasy po přijetí požadavku zákazníka v místě č. 7

Zápis modelu (3.7) – (3.11) v systému LINGO pro tento příklad obsahuje příloha č. 2. Pro zajímavost uvedme ještě dva příklady.

Příklad 3.2

Pokud by dispečer znal požadavek zákazníka před výjezdem vozidla z výchozího místa č. 1, úloha by byla řešena za použití statického modelu. Délka optimální trasy 1-4-2-7-5-3-6-1 je přibližně 225,40 km, tj. trasa je o více než 25 km kratší než v příkladě 3.1. Stejný výsledek obdržíme, přijde-li nový požadavek v okamžiku, kdy je již vozidlo na trase, ale dosud nenavštívilo zákazníka v místě č. 2.

Příklad 3.3

Pokud by nový požadavek přišel později než při přejezdu vozidla z místa č. 2 do místa č. 5, dojde patrně k dalšímu prodloužení trasy. Lze se o tom snadno přesvědčit, pokud budeme předpokládat příchod požadavku v okamžiku, kdy vozidlo bude na cestě z místa č. 5 do místa č. 3. Trasa 1-4-2-5-3 má délku cca 172,53 km, trasa 3-6-7-1 délku cca 110,14 km, celková ujetá vzdálenost je pak přibližně 272,67 km.

Z uvedených příkladů vyplývá, že je výhodné znát co nejvíce požadavků před započítáním jízdy, případně co nejdříve po začátku jízdy. Re-optimalizace trasy je v tomto případě vysoce efektivní a vede k výraznému zkrácení trasy.

3.2.2 Vkládací algoritmus pro dynamickou úlohu obchodního cestujícího

Při vysokém počtu zákazníků, resp. vysoké frekvenci vzniku nových požadavků, je z hlediska reálného času prakticky nemožné opakovaně aplikovat re-optimalizační algoritmus popsany v předchozí části. V reálných situacích zřejmě postačí jednodušší a časově nenáročný heuristický algoritmus vkládání nových požadavků mezi nejvhodnější dvojici zákazníků, pro kterou nabývá výraz (3.6) svého minima.

Necht' $U_N = \{i_1, i_2, \dots, i_m\}$ je posloupnost m míst⁷ ($i_m = 1$), která mají být vozidlem ještě navštívena podle naplánované trasy. Označíme-li zákazníka s novým požadavkem indexem $r \notin U_N$, hodnotu prodloužení stávající trasy po jeho vložení mezi místa i_k a i_{k+1} lze určit jako

$$\Delta z_k = c_{i_k, r} + c_{r, i_{k+1}} - c_{i_k, i_{k+1}}, \quad k = 1, 2, \dots, m-1. \quad (3.18)$$

Cílem je nalézt takový index t , pro který platí:

$$\Delta z_t = \min_{k=1, 2, \dots, m-1} \Delta z_k. \quad (3.19)$$

Nový zákazník r bude navštíven bezprostředně po zákazníkovi i_t před zákazníkem i_{t+1} .

Příklad 3.4

Uvažujme stejnou situaci jako v příkladě 3.1, tj. příchod nového požadavku zákazníka č. 7 v okamžiku, kdy vozidlo jede po optimální trase z místa č. 2 do č. 5 (viz obr. 3.3). Pro posloupnost $U_N = \{5, 3, 6, 1\}$ vytvoříme následující tabulku:

i_k	i_{k+1}	$c_{i_k, 7}$	$c_{7, i_{k+1}}$	$c_{i_k, i_{k+1}}$	Δz_k
5	3	20,81	38,63	32,20	27,23
3	6	38,63	46,10	16,03	68,69
6	1	46,10	38,01	34,93	49,18

Tab. 3.3 – Prodloužení stávající trasy (v km)

⁷ Tato posloupnost může tvořit původní optimální trasu stanovenou před zahájením jízdy vozidla, případně trasu, která byla určena z této optimální trasy postupným vkládáním několika nových požadavků.

Z posledního sloupce tabulky vyplývá, že $t = 1$, tj. nový zákazník bude vložen do stávající trasy mezi zákazníky č. 5 a 3. Výsledek je shodný s optimálním řešením získaným re-optimalizací (viz obr. 3.4); prodloužení trasy oproti řešení bez zákazníka č. 7 činí přibližně 27,23 km.

Z uvedených příkladů je zcela zřejmé, že v okružních úlohách, a především pak v úlohách dynamických, hraje významnou roli čas. Jedná se především o okamžiky příchodu nových požadavků, ale také o časová okna, o která později rozšíříme jednoduché rozvozní úlohy.

3.2.3 Okamžiky příchodu nových požadavků v dynamické úloze obchodního cestujícího

Pro řešení úlohy DTSP je důležité, kdy a jak často vznikají nové požadavky. V příkladech uvedených v předchozích částech jsme předpokládali příchod jediného nového požadavku, a to v okamžiku, kdy se vozidlo nacházelo v určitém místě trasy. V reálných situacích se ovšem jedná většinou o náhodný příchod několika po sobě následujících požadavků.

Abychom byli schopni určit místo, kde se vozidlo nachází v okamžiku, kdy zavolá nový zákazník, musíme znát kromě vzdáleností mezi všemi místy i rychlost vozidla. Pro zjednodušení budeme předpokládat konstantní, resp. průměrnou rychlost vozidla 60 km.h^{-1} . Parametr c_{ij} představující vzdálenost (v kilometrech) mezi místy i a j se tak stává zároveň i časovým parametrem znamenajícím dobu (v minutách), za kterou se vozidlo přesune z místa i do místa j .

Zavedeme další zjednodušení problému týkající se doby obsluhy zákazníka. Bez újmy na obecnosti budeme pro všechny zákazníky považovat tuto dobu za nulovou⁸.

V okamžiku přijetí nového požadavku rozdělují Ichoua et al. (2000) trasu vozidla na tři základní části (obr. 3.5):

Dokončená část

Jedná se o zákazníky, kteří již byli vozidlem obslouženi. Na obr. 3.5 tato část začíná výchozím místem č. 1 a končí posledním navštíveným zákazníkem č. 3.

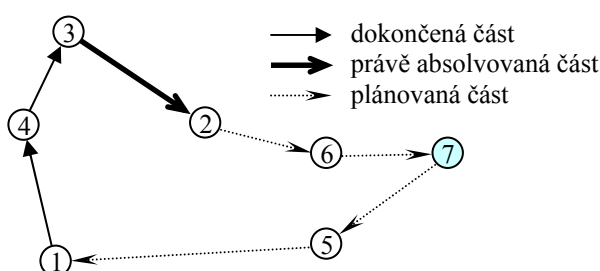
⁸ Model s nenulovou dobou obsluhy bude uveden později u úlohy obchodního cestujícího s časovými okny.

Právě absolvovaná část

Tento úsek trasy odpovídá přejezdu mezi posledním navštíveným zákazníkem č. 3 a bezprostředně následujícím zákazníkem č. 2. Existují různé strategie pro chování vozidla v této části trasy. Budeme předpokládat, že vozidlo v každém případě dokončí naplánovanou cestu k bezprostředně následujícímu zákazníkovi⁹.

Plánovaná část

Předtím, než vozidlo dojde k bezprostředně následujícímu zákazníkovi č. 2, zařadí se nový zákazník č. 7 mezi všechny dosud nenavštívené zákazníky a určí se nová trasa¹⁰, která začíná u zákazníka č. 2 a končí ve výchozím místě č. 1.



Obr. 3.5 – Trasa v okamžiku zařazení nového požadavku

Dispečerská služba přijímající ve firmě nové požadavky a řídící pohyb vozidla pracuje během dne způsobem, který lze popsat následujícím algoritmem:

1. krok – Optimalizace trasy pro všechny předem známé požadavky

Předpokládáme znalost matice vzdáleností mezi všemi místy představovanými výchozím místem a známými zákazníky. Dispečer řeší standardní úlohu obchodního cestujícího (3.1) - (3.5); výsledkem je optimální trasa¹¹, která začíná a končí ve výchozím místě č. 1. Označme počet známých míst (včetně výchozího místa) jako n_z . Délku výsledné trasy pak můžeme zapsat jako

$$T_Z = \sum_{i=1}^{n_z} \sum_{j=1}^{n_z} c_{ij} x_{ij}. \quad (3.20)$$

⁹ Tento předpoklad znamená, že se vozidlo neodkloní z plánované trasy, aby obsloužilo právě vzniklý požadavek, i kdyby se nacházel velice blízko trasy nebo dokonce přímo na trase.

¹⁰ Tuto trasu lze určit např. pomocí re-optimalizace nebo vkládacího algoritmu.

¹¹ V případě velkého počtu zákazníků můžeme použít místo optimalizačního algoritmu některou heuristickou metodu, např. výše popsaný vkládací algoritmus.

Protože c_{ij} představuje nejen vzdálenost mezi místy i a j , ale podle předpokladu o rychlosti vozidla i dobu potřebnou k jeho přejezdu z místa i do místa j , udává hodnota (3.20) zároveň dobu (v minutách), za kterou vozidlo obslouží všechny zákazníky a vrátí se zpět do výchozího místa.

V reálné situaci by měl dispečer porovnat tuto hodnotu s délkou pracovní doby T_{PD} a v případě jejího překročení rozhodnout o změně – např. odmítnutí některých zákazníků či odsunutí jejich obsluhy na další den, přidání dalšího vozidla¹² apod.

2. krok – Zahájení obsluhy zákazníků podle naplánované trasy

Dispečerská služba dá pokyn k výjezdu vozidla z výchozího místa k prvnímu zákazníkovi na trase. Ve stejném okamžiku nastaví celkovou dobu trvání na $T = 0$.

3. krok – Nastává jedna ze tří možných situací¹³:

(a) Vozidlo dorazí k zákazníkovi.

K dosavadní celkové době jízdy T přičteme dobu přejezdu vozidla z předešlého místa k danému zákazníkovi. Protože podle zjednodušujícího předpokladu neuvažujeme žádné zdržení vozidla u zákazníka, tj. doba trvání jeho obsluhy je nulová, vozidlo okamžitě odjíždí k dalšímu zákazníkovi na trase.

(b) Objeví se nový zákazník se svým požadavkem.

Dispečer zařadí zákazníka do seznamu dosud nenavštívených zákazníků a provede naplánování trasy řešením modelu (3.7) - (3.11), resp. (3.12) – (3.17) nebo na základě minimalizace výrazu (3.18). Tato trasa začíná u zákazníka, ke kterému vozidlo právě směřuje¹⁴. Jestliže je součet délky této trasy a celkové délky dosavadní trasy T nižší než je délka pracovní doby T_{PD} , pak je nový požadavek přijat a trasa je upravena podle provedeného výpočtu. V opačném případě dispečer zákazníka odmítne, případně jeho požadavek po dohodě s ním odloží na následující pracovní den.

(c) Vozidlo dorazí do výchozího místa.

Požadavky všech zákazníků byly uspokojeny, vozidlo absolvovalo celou naplánovanou trasu. K dosavadní celkové době trvání trasy T přičteme dobu přejezdu vozidla

¹² Situace s více vozidly zajišťujícími obsluhu zákazníků budou řešeny v další části práce.

¹³ Situace (a) a (b), resp. (b) a (c) mohou ve zcela výjimečných případech nastat současně.

¹⁴ Pokud by vozidlo dorazilo k zákazníkovi, k němuž právě jede, dříve než za dobu nutnou k dokončení výpočtu, může vozidlo u právě obsluženého zákazníka počkat na určení trasy, případně navštívit dalšího zákazníka podle předem naplánované trasy a výpočet další trasy provést až od tohoto zákazníka. Tyto situace zde však uvažovat nebudeme, tj. předpokládáme, že novou trasu je možné určit okamžitě.

od posledního zákazníka do výchozího místa. Vozidlo pak čeká nevyužito na další požadavky, přičemž se hodnota T postupně zvyšuje o délku čekání. V případě, že přijde další požadavek od nového zákazníka označeného indexem r , dispečer posoudí, zda jej vozidlo stihne obsloužit do konce pracovní doby. Pokud

$$T + c_{1r} + c_{r1} \leq T_{PD}, \quad (3.21)$$

kde T je aktuální hodnota uplynulého času od začátku pracovního dne, T_{PD} je délka pracovní doby, c_{1r} a c_{r1} jsou vzdálenosti z výchozího místa k novému zákazníkovi a zpět, resp. doba potřebná pro obsluhu tohoto zákazníka, pak dispečer vydá pokyn k obsluze nového zákazníka. V opačném případě je zákazník odmítnut či jeho obsluha odsunuta na následující pracovní den a vozidlo čeká ve výchozím místě na dalšího nového zákazníka, jehož obsluhu by bylo ještě možné zajistit do konce pracovní doby, tj. pro nějž by byla splněna nerovnost (3.21).

Opakujeme 3. krok tak dlouho, dokud neskončí pracovní doba. Jinými slovy, pokud se dispečer rozhoduje podle uvedeného algoritmu, nemůže nastat jiný případ než ten, že pracovní doba skončí ve chvíli, kdy je vozidlo zpět ve výchozím místě, v ideálním případě přesně ve chvíli, kdy do výchozího místa dorazí.

3.3 Dynamická úloha obchodního cestujícího s časovými okny

Klasická úloha obchodního cestujícího s časovými okny (TSPTW)¹⁵ předpokládá podobně jako TSP znalost všech požadavků před zahájením jízdy. Pro každého zákazníka je navíc definován časový interval, ve kterém má být realizována jeho obsluha. Opět se jedná o standardní optimalizační okružní úlohu, ve které je cílem určit pořadí, v němž budou místa navštívena tak, aby byly splněny požadavky zákazníků a náklady¹⁶ spojené s rozvozem, resp. se svozem, byly minimální.

V dynamické úloze obchodního cestujícího s časovými okny (DTSPTW)¹⁷, na rozdíl od statické úlohy, jsou některé požadavky známy předem, zatímco další požadavky se objevují až při samotné realizaci okružní jízdy. Firma, která obsluhu zákazníků zajišťuje, musí dynamicky reagovat na on-line požadavky nových zákazníků. Jednou ze zajímavých reálných aplikací je svoz noční pošty či zásilek od pravidelných zákazníků, k nimž se během noci náhodně přidávají další zákazníci. Jako další aplikace úlohy DTSPTW lze uvést

¹⁵ Traveling Salesman Problem with Time Windows.

¹⁶ V našem případě celková ujetá vzdálenost či celkový čas nutný pro realizaci okružní jízdy.

¹⁷ Dynamic Traveling Salesman Problem with Time Windows.

taxislužbu, zajišťování servisních služeb jako jsou opravy telefonních stanic, instalatérské, elektrikářské a topenářské práce apod.

3.3.1 Statická úloha obchodního cestujícího s časovými okny

Nejprve uvedeme formulaci matematického modelu úlohy TSPTW, ve které jsou všichni zákazníci známi předem. Časové okno u i -tého zákazníka je definované intervalem mezi nejdříve možným začátkem obsluhy e_i a nejpozději přípustným začátkem obsluhy l_i . Okamžik, ve kterém začne reálná obsluha i -tého zákazníka, označíme τ_i . Předpokládejme omezení $\tau_i \geq e_i$, které znamená, že obsluha i -tého zákazníka nesmí začít před nejdříve možným termínem zahájení obsluhy. Pokud přijede vozidlo k zákazníkovi před nejdříve možným začátkem, musí čekat až do tohoto termínu. Pokud vozidlo nestihne přijet do nejpozději přípustného termínu, tj. pokud $\tau_i > l_i$, pak vzniká penále, tj. náklady spojené s nedodržením zákaznickova požadavku. Tato omezení jsou v literatuře často označována jako „soft“, tedy slabá omezení (Gendreau et al., 1999). Některé reálné aplikace však nepřipouštějí možnost obsluhy mimo interval představující časové okno a pro všechny zákazníky musí tudíž platit $e_i \leq \tau_i \leq l_i$. Taková omezení se pak označují jako „hard“, tj. silná omezení.

V následujícím matematickém modelu opět pro jednoduchost předpokládáme, bez újmy na obecnosti, nulovou dobu obsluhy všech zákazníků (Fábry, 2006):

$$\text{minimalizovat } z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}, \quad (3.22)$$

za podmínek

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n, \quad (3.23)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n, \quad (3.24)$$

$$e_i \leq \tau_i \leq l_i, \quad i = 2, 3, \dots, n, \quad (3.25)$$

$$\tau_i + t_{ij} - M(1 - x_{ij}) \leq \tau_j, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j, \quad (3.26)$$

$$\tau_1 = 0, \quad (3.27)$$

$$\tau_i \geq 0, \quad i = 2, 3, \dots, n, \quad (3.28)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n, \quad (3.29)$$

kde n je počet míst, která vozidlo musí projet (včetně výchozího místa označeného č. 1), c_{ij} představuje vzdálenost mezi místy i a j , t_{ij} dobu přejezdu mezi místy i a j . Hodnoty e_i a l_i představují nejdříve možný a nejpozději přípustný termín obsluhy zákazníka i , M představuje vysokou konstantu. Proměnná x_{ij} je bivalentní proměnná nabývající hodnoty 1 v případě, že vozidlo jede z místa i přímo do místa j , hodnoty 0 v opačném případě. Proměnná τ_i udává okamžik, ve kterém vozidlo navštíví místo i .

Účelová funkce (3.22) a podmínky (3.23) - (3.24) mají stejný význam jako v modelu standardní úlohy obchodního cestujícího TSP. Podmínky (3.25) stanovují, aby zákazníci byli obslouženi uvnitř časového okna. Omezení (3.26) zajišťují, že časový interval mezi návštěvou zákazníka j bezprostředně po zákazníkovi i má minimálně hodnotu t_{ij} . Pokud vozidlo nepojede od zákazníka i k zákazníkovi j , pak díky vysoké konstantě M je tato nerovnost vždy splněna. Desrosiers et al. (1995) uvádějí tyto podmínky také v následující podobě:

$$x_{ij}(\tau_i + t_{ij} - \tau_j) \leq 0, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j. \quad (3.26a)$$

Je zřejmé, že tato omezení jsou nelineární a pro praktické využití tudíž vyhovují lépe původní omezení (3.26). Rovnice (3.27) definuje nulový okamžik výjezdu vozidla z výchozího místa.

Nechť vozidlo po dokončení obsluhy zákazníka i má podle naplánované trasy odjet k zákazníkovi j . Vzhledem k tomu, že vozidlo musí respektovat časové okno zákazníka j , existují dvě strategie **čekání vozidla** u zákazníků:

1) Vozidlo odjede k zákazníkovi j bezprostředně po dokončení obsluhy zákazníka i . Pokud přijede k zákazníkovi j před otevřením časového okna, bude čekat do okamžiku e_j .

2) Vozidlo zůstane u zákazníka i a k zákazníkovi j odjede v takovém okamžiku, aby k němu dorazilo přesně ve chvíli otevření jeho časového okna, tj. vyjede v okamžiku $e_j - t_{ij}$. Výhoda této strategie spočívá v tom, že během čekání vozidla může přijít další požadavek, který ovlivní dříve naplánovanou trasu, tj. vozidlo nepojede z místa i k zákazníkovi j , ale k jinému zákazníkovi tak, aby zbývající trasa byla optimální.

3.3.1.1 Úloha s čekáním vozidla u zákazníka před jeho obsluhou

Přijmeme-li již dříve uvedený předpoklad o průměrné rychlosti vozidla 60 km.h^{-1} , pak $c_{ij} = t_{ij}$ ($i, j = 1, 2, \dots, n$)¹⁸. Zavedeme-li proměnné $W_j \geq 0$ ($j = 2, 3, \dots, n$) představující dobu čekání vozidla před obsluhou zákazníka j , pak účelovou funkci (3.22) lze přepsat do tvaru:

$$\text{minimalizovat } z = \sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij} + \sum_{j=2}^n W_j. \quad (3.22a)$$

Cílem je najít okruh s minimální celkovou dobou trvání. Místo soustavy nerovností (3.26) je v modelu zapotřebí uvést soustavu rovnic:

$$\tau_i + t_{ij} - M(1 - x_{ij}) + W_j + v_{ij} = \tau_j, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j. \quad (3.26b)$$

Proměnné v_{ij} musí respektovat následující omezení:

$$0 \leq v_{ij} \leq 2M(1 - x_{ij}), \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j. \quad (3.30)$$

Pokud vozidlo jede z místa i do místa j ($x_{ij} = 1$), pak nutně musí platit $v_{ij} = 0$. V opačném případě ($x_{ij} = 0$) proměnná v_{ij} funguje jako pomocná proměnná zajišťující přípustnost řešení vzhledem k časovému rozvrhu. V případě, že by v rovnicích (3.26b) tyto proměnné nebyly zavedeny, nebylo by možné zajistit splnění všech uvedených rovnic zároveň.

Pro úplnost ještě uvedme upravený model, ve kterém je kromě údajů o časovém okně u každého zákazníka navíc zadaný údaj o délce obsluhy $S_i \geq 0$ ($i = 2, 3, \dots, n$). Předpokládáme, že časové okno je definováno jako časový interval, ve kterém musí obsluha začít. Obsluha zákazníka i tudíž může skončit až po nejpozději přípustném začátku obsluhy l_i ¹⁹. V účelové funkci se tato změna projeví pouze v podobě konstanty udávající celkovou dobu, kterou vozidlo stráví obsluhou všech zákazníků²⁰:

$$\text{minimalizovat } z = \sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij} + \sum_{i=2}^n S_i + \sum_{j=2}^n W_j. \quad (3.22b)$$

¹⁸ Časové vzdálenosti jsou potom uvedeny v minutách.

¹⁹ Úloha může být definována i tak, že l_i představuje nejpozději přípustný konec obsluhy, nikoli její nejpozději přípustný začátek. Pak je ovšem nutné změnit či přidat některá omezení.

²⁰ Tato konstanta nemá vliv na samotnou optimalizaci, pouze na hodnotu účelové funkce, tj. celkovou dobu jízdy vozidla. Model se ovšem může zkomplikovat v situaci, kdy doba trvání obsluhy není určena deterministicky, ale je hodnotou náhodné veličiny, např. při opravě plynového sporáku či ledničky, kdy není předem znám typ a rozsah závady. Takové úlohy však v této práci uvažovat nebudeme.

Omezení (3.26b) představující časový rozvrh jízdy vozidla mezi zákazníky se pak změní následujícím způsobem:

$$\tau_i + S_i + t_{ij} - M(1 - x_{ij}) + W_j + v_{ij} = \tau_j, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j, \quad (3.26c)$$

přičemž je definováno $S_1 = 0$ a platí omezující podmínky (3.30).

3.3.1.2 Úloha s čekáním vozidla u právě obsluženého zákazníka

Jak bylo uvedeno dříve, může být z hlediska optimalizace trasy pro vozidlo výhodné po dokončení obsluhy zákazníka i u něj setrvat a odjet k následujícímu zákazníkovi j až po určité době tak, aby byl tento zákazník navštíven přesně v okamžiku otevření jeho časového okna, tj. v okamžiku nejdříve možného zahájení obsluhy e_j .

V případě nenulové doby obsluhy lze pak účelovou funkci, podobně jako v předchozí části, zapsat jako

$$\text{minimalizovat } z = \sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij} + \sum_{i=1}^n W'_i + \sum_{j=2}^n S_j, \quad (3.22c)$$

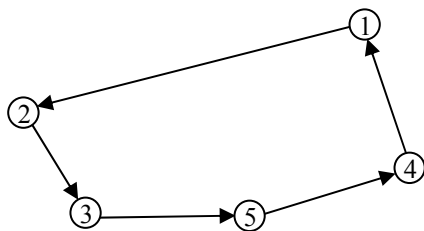
kde proměnné $W'_i \geq 0$ ($i = 2, 3, \dots, n$) představují dobu čekání vozidla po dokončení obsluhy zákazníka i před jeho odjezdem k dalšímu zákazníkovi, resp. zpět do výchozího místa. Hodnota proměnné W'_i odpovídá době, po kterou bude vozidlo čekat ve výchozím místě před tím než vyjede na trasu. Rovnice (3.26c) pak nahradí následující soustava rovnic:

$$\tau_i + W'_i + S_i + t_{ij} - M(1 - x_{ij}) + v_{ij} = \tau_j, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j. \quad (3.26d)$$

Obě strategie dávají evidentně stejný výsledek z hlediska optimální doby, po kterou je vozidlo na trase, a poskytují stejnou posloupnost navštívených míst. Řešení se ovšem liší v hodnotách a interpretaci dob čekání v jednotlivých místech W_i , resp. W'_i . Jak lze ukázat na jednoduchém příkladě, druhá strategie, kdy vozidlo čeká u právě obsluženého zákazníka, může být výhodnější v případě dynamické úlohy obchodního cestujícího s časovými okny, která bude předmětem následující části.

Příklad 3.5

Uvažujme výchozí místo č. 1 a známé zákazníky č. 2-5. Optimální trasa 1-2-3-5-4-1 je znázorněna na obr. 3.6.



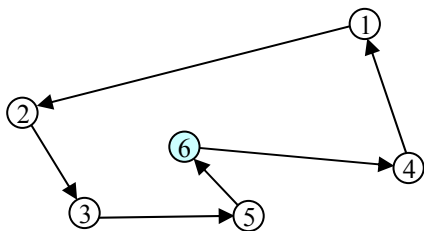
Obr. 3.6 – Optimální trasa

Pokud vozidlo používá **první strategii**, odjíždí od každého zákazníka ihned po dokončení jeho obsluhy k následujícímu zákazníkovi, u něhož případně čeká na otevření časového okna. Této strategii odpovídá obr. 3.7. Požadavek nového zákazníka č. 6 může vzniknout v okamžiku, kdy

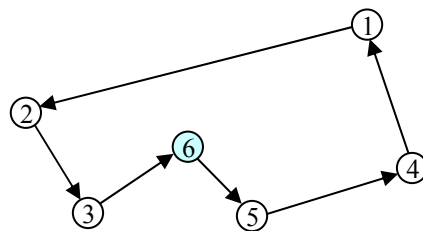
- vozidlo přejíždí od zákazníka č. 3 k zákazníkovi č. 5,
- vozidlo čeká u zákazníka č. 5 na otevření jeho časového okna,
- probíhá obsluha zákazníka č. 5.

Nový zákazník je zařazen do trasy za zákazníka č. 5.

Při použití **druhé strategie** (obr. 3.8) vozidlo čeká u zákazníka č. 3 a odjíždí v takovém okamžiku, aby přijel k zákazníkovi č. 5 přesně ve chvíli otevření jeho časového okna. Pokud tedy vznikne požadavek nového zákazníka č. 6 po dokončení obsluhy zákazníka č. 3 během čekání vozidla u tohoto zákazníka, má vozidlo možnost změnit dříve naplánovanou trasu ještě před tím, než odjede k zákazníkovi č. 5. Jestliže budou splněny všechny omezující podmínky, bude nový zákazník navštíven ještě před obslužením zákazníka č. 5, čímž získáme lepší řešení než v případě první strategie.



Obr. 3.7 – Čekání u zákazníka č. 5 před jeho obslužením



Obr. 3.8 – Čekání u zákazníka č. 3 po jeho obslužení

Uvedený příklad ukazuje, že může být výhodné čekat u obslužených zákazníků, zda nevznikne nový požadavek, a upravit pak co nejdříve zbývající část trasy²¹.

3.3.2 Dynamická úloha obchodního cestujícího s časovými okny

Jak bylo naznačeno v předešlé části textu, i v úloze s časovými okny často neznáme všechny zákazníky před zahájením jízdy vozidla. Další požadavky přicházejí od nových zákazníků postupně během jízdy vozidla; dispečer musí, pokud je to možné, nově vzniklé požadavky zařadit do zbývající části trasy. Celý proces zpracování požadavků zákazníků lze opět rozdělit do tří následujících kroků:

- Optimalizace trasy pro všechny předem známé požadavky.
- Zahájení obsluhy zákazníků podle naplánované trasy.
- Zakomponování nových požadavků do plánované trasy, případně jejich odmítnutí.

Podívejme se na jednotlivé kroky podrobně, přičemž pro zjednodušení úvah budeme opět předpokládat nulovou dobu obsluhy u všech zákazníků²².

Označíme-li počet známých míst (včetně výchozího místa) jako n_z , pak celková doba jízdy vozidla je dána účelovou funkcí

$$\text{minimalizovat } T_Z = \sum_{i=1}^{n_z} \sum_{j=1}^{n_z} t_{ij} x_{ij} + \sum_{j=2}^{n_z} W_j . \quad (3.31)$$

Při minimalizaci této funkce je nutné respektovat omezující podmínky (3.23) – (3.25), (3.26b) a (3.27) – (3.30), v nichž místo hodnoty n je použita hodnota n_z . Reálným výsledkem optimalizace je trasa, tj. posloupnost zákazníků, které má vozidlo postupně navštívit, dále pak čas, ve kterém vozidlo dojde k určitému zákazníkovi, a okamžik, ve kterém má začít jeho obsluha, resp. v našem případě okamžik, ve kterém se uskuteční obsluha nulové délky. U každého zákazníka mohou nastat dvě situace:

- 1) Vozidlo dojde k zákazníkovi j **před nejdříve možným začátkem obsluhy** e_j .

V takovém případě vozidlo čeká na zahájení obsluhy dobu $W_j > 0$.

²¹ Předpokladem tohoto ilustrativního příkladu je, že zařazení nového zákazníka splňuje časová okna všech zákazníků.

²² Z předešlých úprav je zřejmé, že zahrnutí délky obsluhy do modelu je poměrně jednoduchou záležitostí; na tomto místě by ovšem podobné úvahy byly provedeny na úkor přehlednosti.

2) Vozidlo přijede k zákazníkovi j **po okamžiku nejdříve možného začátku obsluhy** e_j . Vozidlo pak může ihned zákazníka obsloužit, neboli $W_j = 0$.

Jakmile dispečer zná všechny potřebné údaje, může dát pokyn řidiči vozidla k zahájení okružní jízdy.

3.3.2.1 Re-optimalizace trasy po přidání zákazníka

V jakémkoli okamžiku, kdy je vozidlo na trase, může přijít požadavek od nového zákazníka. Dispečer se pokusí zařadit takového zákazníka do zbývající části trasy, tj. do množiny dosud nenavštívených zákazníků. Po re-optimalizaci této trasy zjistí, zda po zařazení nového požadavku nedošlo k prodloužení jízdy vozidla za hranici pracovní doby²³. Aktualizovaná trasa je pak nahlášena řidiči vozidla. Re-optimalizaci lze založit na následujícím matematickém modelu (Fábry, 2006):

$$\text{minimalizovat } z = \sum_{i \in U_N} \sum_{j \in U_N} t_{ij} x_{ij} + \sum_{\substack{j \in U_N \\ j \neq 1}} W_j, \quad (3.32)$$

za podmínek

$$\sum_{j \in U_N} x_{ij} = 1, \quad i \in U_N, \quad (3.33)$$

$$\sum_{i \in U_N} x_{ij} = 1, \quad j \in U_N, \quad (3.34)$$

$$e_i \leq \tau_i \leq l_i, \quad i \in U_N - \{1\}, \quad (3.35)$$

$$\tau_i + t_{ij} - M(1 - x_{ij}) + W_j + v_{ij} = \tau_j, \quad i \in U_N, \quad j \in U_N - \{1\}, \quad i \neq j, \quad (3.36)$$

$$0 \leq v_{ij} \leq 2M(1 - x_{ij}), \quad i \in U_N, \quad j \in U_N - \{1\}, \quad i \neq j, \quad (3.37)$$

$$\tau_1 = 0, \quad (3.38)$$

$$\tau_j \geq 0, \quad j \in U_N - \{1\}, \quad (3.39)$$

$$x_{1j_{next}} = 1, \quad (3.40)$$

$$x_{ij} \in \{0,1\}, \quad i, j \in U_N, \quad (3.41)$$

kde U_N je množina míst (včetně sídla nově přidaného zákazníka), která mají být vozidlem ještě navštívena²⁴, $|U_N|$ je jejich počet. Index j_{next} odpovídá zákazníkovi, k němuž vozidlo

²³ Pokud by vozidlo nestihlo obsloužit zákazníky v rámci pracovní doby, je nový požadavek odmítnut, případně po dohodě se zákazníkem jeho obsluha přeložena na další pracovní den.

²⁴ Jako poslední bude navštíveno výchozí místo č. 1.

směřuje v okamžiku přijetí nového požadavku. Protože vozidlo nemůže změnit směr jízdy, musí být splněna podmínka (3.40).

Podobně jako v modelu (3.12) – (3.17) je před optimalizací upraven původní čas přejezdu $t_{1j_{next}}$ na hodnotu odpovídající době trvání jízdy vozidla z výchozího místa k zákazníkovi j_{next} . Pokud v praktické úloze sledujeme pracovní dobu, která nesmí být překročena, pak je nutné přidat omezení, které zakazuje účelové funkci (3.32) tuto hodnotu překročit.

Uvedený matematický model je formulován pro **úlohu s čekáním vozidla u zákazníka před jeho obsluhou**. Tento model lze snadno upravit pro úlohu s čekáním vozidla u právě obsluženého zákazníka.

3.3.2.2 Zařazení nového zákazníka pomocí vkládacího algoritmu

Podobně jako v dynamické úloze obchodního cestujícího bez uvažování časových oken DTSP, je i v úloze s časovými okny DTSPTW při vysokém počtu zákazníků výhodné místo výše uvedené re-optimalizace použít vkládací algoritmus (Fábry, 2006).

Nechť $U_N = \{i_1, i_2, \dots, i_m\}$ je posloupnost m míst ($i_m = 1$), která vozidlo má ještě navštívit podle plánované trasy. Označme nového zákazníka indexem $r \notin U_N$ a místa, mezi která jej vložíme, indexy i_k a i_{k+1} . Okamžik obsluhy zákazníka r lze určit jako

$$\tau_r = \max(\tau_{i_k} + t_{i_k, r}, e_r), \quad (3.42)$$

přičemž musí platit:

$$e_r \leq \tau_r \leq l_r. \quad (3.43)$$

Hodnota čekání na obsluhu zákazníka r je pak definována jako

$$W_r = \max(0, e_r - \tau_{i_k} - t_{i_k, r}). \quad (3.44)$$

Vzhledem k tomu, že vložením nového zákazníka dojde ke změně zbývajících částí trasy, je zapotřebí přepočítat hodnoty τ_i a W_i pro všechna dosud nenavštívená místa, počínaje tím, před které byl nový zákazník vložen:

$$\tau_{i_{k+1}} = \max(\tau_r + t_{r, i_{k+1}}, e_{i_{k+1}}), \quad (3.45)$$

$$W_{i_{k+1}} = \max(0, e_{i_{k+1}} - \tau_r - t_{r,i_{k+1}}), \quad (3.46)$$

$$\tau_{i_s} = \max(\tau_{i_{s-1}} + t_{i_{s-1},i_s}, e_{i_s}), \quad s = k+2, k+3, \dots, m, \quad (3.47)$$

$$W_{i_s} = \max(0, e_{i_s} - \tau_{i_{s-1}} - t_{i_{s-1},i_s}), \quad s = k+2, k+3, \dots, m. \quad (3.48)$$

Při použití uvedeného postupu je nutné hlídat přípustnost upravené trasy z hlediska časových oken, tj. musí platit:

$$e_{i_s} \leq \tau_{i_s} \leq l_{i_s}, \quad s = k+1, k+2, \dots, m. \quad (3.49)$$

Dobu trvání nové trasy po vložení zákazníka r mezi místa i_k a i_{k+1} lze psát jako

$$z_k = \sum_{s=1}^{k-1} (t_{i_s, i_{s+1}} + W_{i_{s+1}}) + (t_{i_k, r} + W_r) + (t_{r, i_{k+1}} + W_{i_{k+1}}) + \sum_{s=k+1}^{m-1} (t_{i_s, i_{s+1}} + W_{i_{s+1}}). \quad (3.50)$$

Cílem je nalézt takové v , pro které platí:

$$z_v = \min_{k=1,2,\dots,m-1} z_k. \quad (3.51)$$

Nového zákazníka pak vložíme mezi místa i_v a i_{v+1} .

3.3.3 Dynamická úloha obchodního cestujícího s časovými okny a penalizací

V reálných situacích, řešených jako problém obchodního cestujícího s časovými okny, je nutnost respektování dolní a horní hranice intervalu časových oken často příliš silným omezením, které prakticky znemožňuje firmám přijímat další požadavky. Při zařazení takových požadavků do naplánované trasy, ať již na základě re-optimalizace či např. pomocí vkládacího algoritmu, může dispečer snadno dospět k nepřipustnému řešení, což v konečném důsledku vede k odmítnutí zákazníka a přenechání jeho požadavku službám konkurenčních firem.

Jistým východiskem z této situace je uvolnění striktních omezení ve smyslu možnosti překročit horní hranici časového okna zákazníka i , tj. nejpozději přípustný termín začátku obsluhy l_i . Za toto překročení ovšem firma musí zaplatit zákazníkovi určité penále, které má často podobu paušální částky. Jeho výše může ovšem také záviset na době, o kterou je termín l_i překročen.

3.3.3.1 Změna modelu statické úlohy

Uvažujeme-li uvolnění zmíněných podmínek u statické úlohy popsané v části 3.3.1, pak účelovou funkci (3.22), resp. (3.22a) přepíšeme do tvaru nákladové funkce (Larsen et al., 2004):

$$\text{minimalizovat } z = \alpha \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} + \beta \sum_{i=2}^n \max\{0, (\tau_i - l_i)\}, \quad (3.52)$$

resp.

$$\text{minimalizovat } z = \alpha \left(\sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij} + \sum_{j=2}^n W_j \right) + \beta \sum_{i=2}^n \max\{0, (\tau_i - l_i)\}. \quad (3.53)$$

Parametr β je hodnota penále, které firma zaplatí za překročení nejpozději přípustného termínu začátku obsluhy o 1 časovou jednotku. Konstanta α koriguje řádové rozdíly mezi první částí nákladové funkce, tj. celkovými přepravními náklady plynoucími ze vzdáleností mezi jednotlivými zákazníky, resp. z časových hodnot, a druhou penalizační částí.

Pokud by velikost penalizace nedodržení časových oken byla závislá na individuálních smluvních podmínkách či dohodách se zákazníky, bude penalizační část nákladové funkce (3.52), resp. (3.53) nahrazena výrazem

$$\sum_{i=2}^n \beta_i \max\{0, (\tau_i - l_i)\}, \quad (3.54)$$

kde β_i je smluvní jednotkové penále, platné pro i -tého zákazníka.

Podmínky (3.25) v modelu statické úlohy se pak změň na následující nerovnosti:

$$e_i \leq \tau_i, \quad i = 2, 3, \dots, n. \quad (3.55)$$

Nelinearitu penalizační účelové funkce lze odstranit zavedením nových proměnných, pro které platí:

$$\tau_i + d_i^- - d_i^+ = l_i, \quad i = 2, 3, \dots, n. \quad (3.56)$$

Hodnota d_i^+ představuje velikost překročení nejpozději přípustného termínu l_i , hodnota d_i^- udává časovou rezervu do vypršení tohoto termínu. Účelové funkce (3.52) a (3.53) jsou přepsány následovně:

$$\text{minimalizovat } z = \alpha \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} + \beta \sum_{i=2}^n d_i^+, \quad (3.52a)$$

$$\text{minimalizovat } z = \alpha \left(\sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij} + \sum_{j=2}^n W_j \right) + \beta \sum_{i=2}^n d_i^+. \quad (3.53a)$$

3.3.3.2 Změna modelu dynamické úlohy

Optimální řešení úlohy obchodního cestujícího s časovými okny pro množinu předem známých zákazníků získáme na základě minimalizace statické úlohy s následující nákladovou účelovou funkcí:

$$\text{minimalizovat } z = \alpha \left(\sum_{i=1}^{n_z} \sum_{j=1}^{n_z} t_{ij} x_{ij} + \sum_{j=2}^{n_z} W_j \right) + \beta \sum_{i=2}^{n_z} d_i^+, \quad (3.57)$$

kde n_z je počet známých míst (včetně výchozího místa). Celková doba jízdy vozidla T_z je dána funkcí (3.31).

Pro **re-optimalizaci** bude použit matematický model formulovaný v části 3.3.2.1, ve kterém bude místo účelové funkce (3.32) použita následující nákladová funkce:

$$\text{minimalizovat } z = \alpha \left(\sum_{i \in U_N} \sum_{j \in U_N} t_{ij} x_{ij} + \sum_{\substack{j \in U_N \\ j \neq 1}} W_j \right) + \beta \sum_{\substack{j \in U_N \\ j \neq 1}} d_i^+, \quad (3.58)$$

kde U_N je množina míst (včetně sídla nově přidaného zákazníka a výchozího místa č. 1), která musí vozidlo ještě navštívit. Místo silných omezení (3.35) platí slabá omezení $e_i \leq t_i$, $i \in U_N - \{1\}$. Z důvodu linearizace původní účelové funkce jsou přidány podmínky $\tau_i + d_i^- - d_i^+ = l_i$, $i \in U_N - \{1\}$. Celková doba jízdy vozidla je určena hodnotou funkce (3.32), která se samozřejmě může výrazně lišit²⁵ od své původní optimální hodnoty v úloze bez uvolnění podmínek pro časová okna.

²⁵ Celková doba jízdy může být vyšší i nižší.

Při použití **vkładacího algoritmu** lze nákladovou funkci pro situaci, kdy vložíme zákazníka r mezi místa i_k a i_{k+1} , vyjádřit následovně:

$$\text{minimalizovat } z = \alpha z_k + \beta \left[d_r^+ + \sum_{s=k+1}^{m-1} d_s^+ \right], \quad (3.59)$$

kde z_k je doba trvání nové trasy, vypočtená podle (3.50). Podobně jako při re-optimalizaci budou i v tomto případě splněna slabá omezení, včetně $e_r \leq \tau_r$. Pro linearizaci účelové funkce je nutné zavést výše uvedené podmínky, včetně podmínky $\tau_r + d_r^- - d_r^+ = l_r$. Index v , pro který nabývá funkce (3.59) svého minima, určuje dvojici zákazníků i_v a i_{v+1} , mezi které bude vložen nový zákazník, podobně jako při použití kritéria (3.51).

3.3.4 Dynamická úloha obchodního cestujícího s apriorní informací

Ve všech doposud uvedených modelech jsme předpokládali, že distribuční firma není schopna předem odhadnout, kdy a v jakém místě se vyskytne nový požadavek. Tyto informace mají většinou stochastický charakter. Má-li firma k dispozici typ a parametry pravděpodobnostního rozdělení náhodných veličin vyskytujících se v modelu, může významně zvýšit kvalitu nalezeného řešení snížením celkových nákladů souvisejících s plánováním trasy a v neposlední řadě zvýšit atraktivitu firmy pro potenciální zákazníky.

Práce Larsen et al. (2004) se týká dynamické úlohy obchodního cestujícího s časovými okny s apriorní informací (ADTSPTW)²⁶ o pravděpodobnosti vzniku požadavku v určité oblasti. Pro další úvahy lze předpokládat, že všichni zákazníci se nacházejí v oblasti, kterou můžeme znázornit jako čtverec rozdělený do n_{REG} čtvercových regionů identické velikosti. V úloze je definována množina stanovišť, která slouží jako výchozí nebo čekací místa pro vozidlo, tj. jako místa, kde se vozidlo může nacházet v případě, že je volné. V některých regionech se může nacházet více stanovišť, zatímco v jiných regionech nemusí být stanoviště žádné. Bez újmy na obecnosti předpokládejme, že v každém regionu se nachází právě jedno stanoviště, tedy počet všech stanovišť je n_{REG} . U každého zákazníka je definované časové okno. Vozidlo může čekat jak v některém ze stanovišť, tak u některého ze zákazníků.

²⁶ A Priori Dynamic Traveling Salesman Problem with Time Windows.

Nejprve je na základě množiny známých zákazníků²⁷, jejichž počet je označen n_{AR} , určena optimální trasa. K tomuto účelu lze využít řešení statické úlohy obchodního cestujícího s časovými okny popsané v části 3.3.1. Během realizace okružní jízdy volají další zákazníci²⁸, jejichž počet n_{IR} je hodnotou náhodné veličiny. Předpokládejme, že počet požadavků n_j vzniklých v regionu j ($j = 1, 2, \dots, n_{REG}$) za jednotku času se řídí Poissonovým rozdělením se známou intenzitou λ_j . Okamžik přijetí požadavku i -tého zákazníka, který označíme T_i ($i = 1, 2, \dots, n_{IR}$), je hodnotou náhodné veličiny, jejíž pravděpodobnostní rozdělení není firmě známo. Hodnota T_0 představuje začátek pracovní doby, tj. okamžik výjezdu vozidla na trasu.

V úloze s apriorní informací je zapotřebí rozlišit časová okna pro předem známé zákazníky a pro nové zákazníky.

Časová okna známých zákazníků

Požadavky předem známých zákazníků vznikly před okamžikem T_0 . Časové okno i -tého zákazníka ($i = 1, 2, \dots, n_{AR}$) je definováno nejdříve možným začátkem obsluhy e_i a nejpozději přípustným začátkem obsluhy l_i (viz část 3.3.1). Předpokládejme, že pro skutečný okamžik začátku obsluhy τ_i platí slabé omezení, které striktně respektuje pouze nejdříve možný termín začátku obsluhy. Za zpoždění začátku obsluhy oproti nejpozději přípustnému termínu je nutné zaplatit penále.

Časová okna nových zákazníků

Přijde-li nový požadavek i -tého zákazníka v okamžiku T_i ($i = 1, 2, \dots, n_{IR}$), pak jeho časové okno může být otevřeno nejdříve v čase $e_i = T_i + TR$, kde TR je reakční doba po obdržení požadavku, tj. časový interval nutný pro zpracování nového požadavku a rozhodnutí o další strategii vozidla.

Postup při řešení úlohy obchodního cestujícího s apriorní informací se opírá o modely uvedené v předchozí části této kapitoly. Nejdříve je vyřešena statická úloha pro předem známé zákazníky, následně jsou přidáváni noví zákazníci s použitím re-optimalizačního algoritmu. Jako **účelová funkce** je použita některá z dříve formulovaných nákladových funkcí.

²⁷ Tito zákazníci jsou v práci Larsen et al. (2004) označeni jako Advance-Request Customers.

²⁸ Immediate-Request Customers (Larsen et al., 2004).

Následující text je zaměřen na využití apriorní informace za účelem pružnější reakce distribuční firmy na nově vzniklé požadavky.

3.3.4.1 Možné strategie přesunu vozidla

V případě neexistence apriorní informace o intenzitách vzniku požadavku mohou v okamžiku, kdy je ukončena obsluha určitého zákazníka, nastat tři situace:

- 1) Okno dalšího zákazníka v pořadí je již otevřené nebo bude otevřené před tím, než vozidlo přijede k tomuto zákazníkovi. Vozidlo odjíždí k dalšímu zákazníkovi bezprostředně po dokončení obsluhy předchozího zákazníka.
- 2) Okno dalšího zákazníka se otevře později než v okamžiku, kdy by k němu vozidlo přijelo, pokud by odjelo hned od zákazníka, jehož obsluhu právě ukončilo. Vozidlo setrvává u právě obslouženého zákazníka a čeká na vhodný okamžik odjezdu k dalšímu zákazníkovi, případně na vznik dalšího požadavku.
- 3) Žádný zákazník nečeká na obsluhu. V tomto případě vozidlo čeká u zákazníka, jehož obsluhu dokončilo, dokud se neobjeví nový požadavek.

Využije-li firma apriorní informaci o intenzitách vzniku požadavků, pak lze modifikovat strategie chování vozidla po dokončení obsluhy pro výše uvedenou druhou, resp. třetí situaci podle následujících kritérií:

(1) nejbližší stanoviště – po obsluze zákazníka vozidlo odjíždí na stanoviště, které je nejbližší pozici, na které se právě nachází,

(2) nejvytíženější stanoviště – po obsluze zákazníka přejíždí vozidlo na stanoviště do regionu s nejvyšší hodnotou λ_j ($j = 1, 2, \dots, n_{REG}$),

(3) nejatraktivnější stanoviště – pro přesun vozidla je vybráno stanoviště s nejvyšší hodnotou atraktivnosti, určenou podle následujícího vzorce:

$$AT_j = \lambda_j \Delta T_j, \quad j = 1, 2, \dots, n_{REG}, \quad (3.60)$$

kde ΔT_j je délka prostoje vozidla v případě, že bylo pro přesun vybráno j -té stanoviště, neboli časový interval mezi současným okamžikem a okamžikem, kdy vozidlo bude muset odjet

z j -tého stanoviště k dalšímu zákazníkovi. Hodnotu AT_j lze jednoduše interpretovat jako očekávaný počet požadavků vzniklých v j -tém regionu během intervalu ΔT_j .

3.3.4.2 Přesun volného vozidla

Pokud je na základě jednoho ze tří výše uvedených kritérií pro přesun vozidla vybráno j -té stanoviště, pak rozhodnutí o skutečném přesunu vozidla či jeho setrvání u právě obsluženého zákazníka lze řídit prostřednictvím tzv. **prahu výhodnosti přesunu** L_p . Přesun do j -tého stanoviště je výhodný v případě, že pravděpodobnost vzniku alespoň jednoho požadavku během intervalu ΔT_j je rovna minimálně hodnotě L_p . Vzhledem k tomu, že počet požadavků N vzniklých v j -tém regionu během ΔT_j je diskrétní náhodná veličina s Poissonovým rozdělením s intenzitou λ_j , pravděpodobnost vzniku k požadavků během tohoto intervalu lze určit jako

$$P\{N = k\} = \frac{1}{k!} (\lambda_j \Delta T_j)^k e^{-\lambda_j \Delta T_j}, \quad k = 0, 1, 2, \dots \quad (3.61)$$

Pro pravděpodobnost vzniku alespoň jednoho požadavku lze pak psát

$$P\{N \geq 1\} = 1 - e^{-\lambda_j \Delta T_j}. \quad (3.62)$$

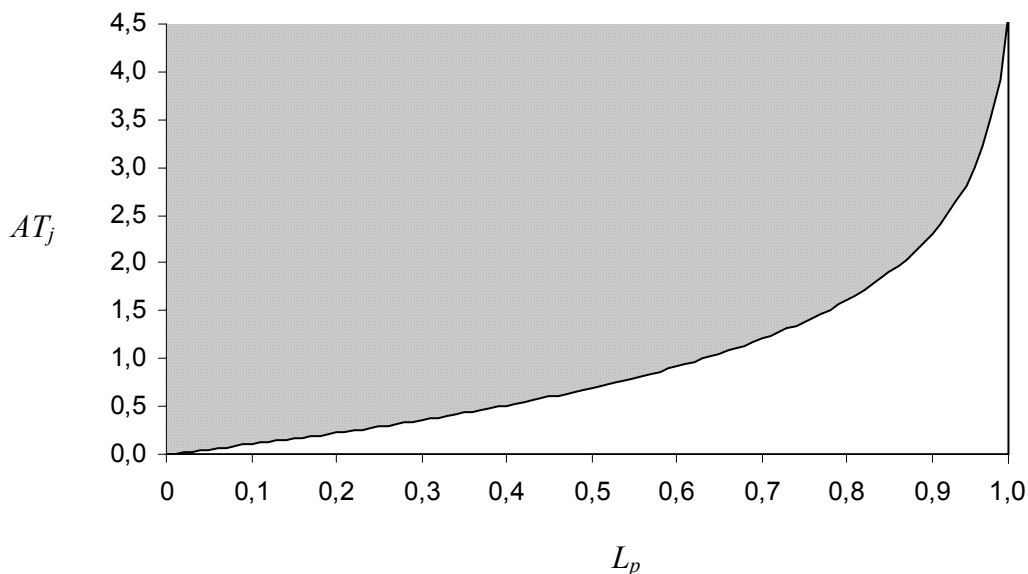
Jak bylo uvedeno, přesun do j -tého stanoviště bude uskutečněn v případě, že pravděpodobnost (3.62) bude rovna alespoň prahu výhodnosti přesunu L_p :

$$P\{N \geq 1\} \geq L_p. \quad (3.63)$$

Dosazením (3.63) do (3.62) a využitím vztahu (3.60) dostáváme po jednoduché úpravě výslednou podmínku pro přesun vozidla do j -tého stanoviště:

$$AT_j \geq -\ln(1 - L_p). \quad (3.64)$$

Uvedená podmínka dává do souvislosti atraktivnost vybraného j -tého stanoviště a práh výhodnosti přesunu. Vyšrafovaná oblast nad křivkou na obr. 3.9 odpovídá takovým kombinacím obou veličin, při kterých se přesun uskuteční, zatímco oblast pod křivkou znamená pro vozidlo, že má zůstat u právě obsluženého zákazníka.



Obr. 3.9 – Závislost atraktivnosti stanoviště na prahu výhodnosti přesunu

Je zřejmé, že čím vyšší je práh výhodnosti přesunu, tím vyšší musí být intenzita vzniku požadavků v regionu, do jehož stanoviště se hodlá vozidlo přesunout.

3.3.4.3 Výpočetní experimenty

Jak ukazují výsledky výpočetních experimentů (Larsen et al., 2004), přesun volného vozidla do atraktivních stanovišť je výhodný z hlediska většího počtu zákazníků, které je možné během dne uspokojit. Použití této strategie je však značně nevýhodné z hlediska celkové vzdálenosti, kterou vozidlo během dne ujede. Nejedná se samozřejmě o větší vzdálenost ujetou v důsledku vyššího počtu zákazníků, ale v důsledku umělých přejezdů vozidla do stanovišť na základě znalosti apriorní informace.

Důležitou roli hraje tzv. **stupeň dynamiky**²⁹, který charakterizuje konkrétní úlohu z hlediska relativního počtu nových zákazníků vzhledem k celkovému počtu všech zákazníků:

$$DOD = \frac{n_{IR}}{n_{AR} + n_{IR}} 100\%. \quad (3.65)$$

S rostoucím stupněm dynamiky roste i rozdíl mezi celkovou vzdáleností ujetou vozidlem, které přejíždí do atraktivnějších míst a vzdáleností ujetou vozidlem, které čeká u právě obsluženého zákazníka, případně přejíždí do nejbližšího stanoviště.

²⁹ Degree of Dynamism (Larsen et al., 2004).

Pokud jde o zpoždění obsluhy oproti nejpozději přípustným začátkům obsluhy, při nízkém stupni dynamiky dochází prakticky ke stejnému zpoždění při všech použitých strategiích³⁰. Při vyšších stupních dynamiky je častý přejezd vozidla příčinou mnohem většího zpoždění než v případě, že vozidlo nevyužívá apriorní informaci. Vhodným kompromisem je v tomto případě použití nejbližšího či nejvytíženějšího stanoviště.

Obecně lze říci, že dostupnost apriorní informace je výhodná z hlediska vyššího počtu obslužených zákazníků a tím souvisejících vyšších tržeb za poskytnuté služby. Nevýhodou je nárůst jak přepravních nákladů, tak i nákladů spojených se zpožděním obsluhy. Volba nejvhodnější strategie závisí na charakteru konkrétní firmy a finančních podmínkách, za kterých operuje na trhu. Jak bylo uvedeno, důležitou roli pro výběr optimální strategie hraje také stupeň dynamiky konkrétní aplikace.

3.4 Dynamická úloha kurýrní služby

Další modifikací úlohy obchodního cestujícího je poskytování kurýrních služeb, umožňujících zákazníkům doručit zásilku z jednoho místa do druhého (tzv. doručení ze stolu na stůl). Úloha je obdobou tzv. dial-a-ride problému, ve kterém jsou místo zásilek převáženi např. hendikepovaní lidé (tzv. přeprava od dveří ke dveřím). V takovém případě je zapotřebí uvažovat kapacitu vozidla a časová okna pro nástup a výstup každého zákazníka. Přesný algoritmus pro statickou úlohu s jedním vozidlem uvedl Psaraftis (1983). Pro dynamické rozšíření této úlohy Psaraftis (1980) využil dynamické programování. Cordeau (2006) popsal metodu větvení a řezů pro statickou úlohu, v níž je k dispozici několik vozidel umístěných v jednom výchozím místě. Heuristické algoritmy vyvinuli Toth and Vigo (1997), Jaw et al. (1986) a další.

V této práci se soustředíme na úlohu kurýrní služby, v níž je pro převoz zásilek použito jedno vozidlo s neomezenou kapacitou. Nejprve bude popsána statická úloha bez časových oken i s časovými okny. Následuje dynamické rozšíření této úlohy.

3.4.1 Statická úloha kurýrní služby bez časových oken

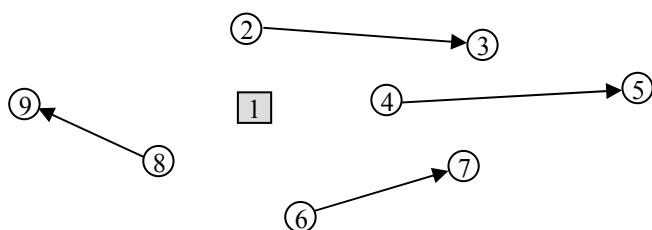
Přestože časová okna jsou v reálných aplikacích týkajících se kurýrních služeb definována téměř vždy, nejprve se zaměříme na jednodušší model, a sice bez uvažování časového

³⁰ Jako účelová funkce je použita celková hodnota zpoždění, která se minimalizuje.

intervalu, ve kterém má být zásilka vyzvednuta. Podobně neuvažujeme ani interval, ve kterém má být zásilka doručena.

Příklad 3.6

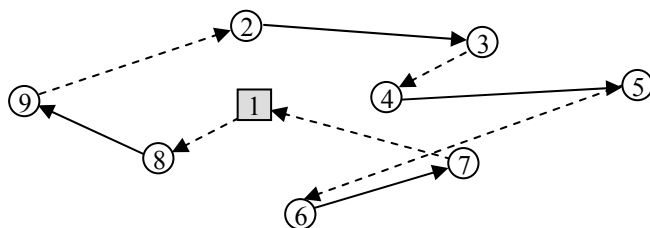
Na obr. 3.10 je zobrazeno výchozí místo firmy poskytující kurýrní služby (označené č. 1). Na začátku pracovní doby jsou známí 4 zákazníci. Zákazník č. 2 požaduje doručit zásilku ze svého sídla do místa č. 3, zákazník č. 4 do místa č. 5, zákazník č. 6 do místa č. 7 a zákazník č. 8 do místa č. 9. Zákazníci jsou očíslováni v pořadí, v jakém volali do firmy. Jak bude zřejmé z matematických modelů formulovaných v další části textu, způsob očíslování míst, ze kterých mají být zásilky vyzvednuty, a míst, do kterých mají být doručeny, je záměrný. Sudými indexy jsou označena sídla zákazníků, zatímco lichými indexy místa doručení zásilek. Místo doručení má vždy index o jedničku vyšší než je index sídla zákazníka, který do tohoto místa zásilku posílá. Orientované hrany na obr. 3.10 znázorňují směr doručení zásilky.



Obr. 3.10 – Požadavky zákazníků na doručení zásilek

Poznámka. Ve všech úlohách a modelech bude platit zjednodušující předpoklad, že sídlo žádného zákazníka nemůže být místem doručení pro jiného zákazníka a naopak žádné místo doručení nemůže být sídlem zákazníka. Navíc nebudou uvažovány situace, ve kterých by zákazník měl dvě či více zakázek se dvěma či více různými místy doručení, resp. by jedno místo bylo místem doručení zásilek pro dva či více zákazníků.

Při hledání optimální trasy je důležitá skutečnost, že řešení není omezeno časovými okny. Vozidlo také nemusí doručit zásilku bezprostředně po jejím vyzvednutí. Proto pokud vozidlo vyzvedne zásilku od určitého zákazníka, může při cestě do místa doručení této zásilky navštívit další zákazníky, případně doručit zásilky již předtím získané. Není tudíž nutné naplánovat trasu podobně jako na obr. 3.11.



Obr. 3.11 – Doručení zásilek bezprostředně po jejich vyzvednutí

V reálné praxi jsou samozřejmě uplatňována časová okna minimálně v podobě vyzvednutí zásilky po určitém okamžiku a jejím doručení v nejpozději přípustném termínu. Nejprve však uvedeme matematický model bez časových oken.

Předokládejme, že známe n sídel zákazníků a n míst doručení zásilek, celkem tedy $2n + 1$ míst (včetně výchozího místa). Dále známe vzdálenosti c_{ij} mezi místy i a j .

$$\text{minimalizovat } z = \sum_{i=1}^{2n+1} \sum_{j=1}^{2n+1} c_{ij} x_{ij}, \quad (3.66)$$

za podmínek

$$\sum_{j=1}^{2n+1} x_{ij} = 1, \quad i = 1, 2, \dots, 2n + 1, \quad (3.67)$$

$$\sum_{i=1}^{2n+1} x_{ij} = 1, \quad j = 1, 2, \dots, 2n + 1, \quad (3.68)$$

$$u_i - u_j + (2n + 1)x_{ij} \leq 2n, \quad i = 1, 2, \dots, 2n + 1, \quad j = 2, 3, \dots, 2n + 1, \quad i \neq j, \quad (3.69)$$

$$u_{2i} \leq u_{2i+1}, \quad i = 1, 2, \dots, n, \quad (3.70)$$

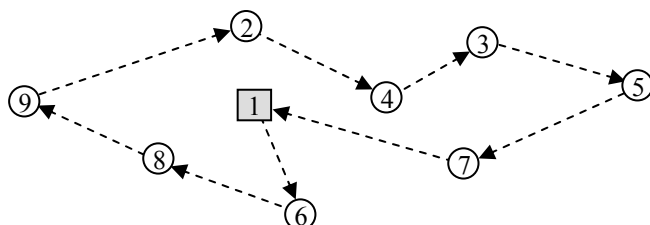
$$u_1 = 0, \quad (3.71)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, 2n + 1. \quad (3.72)$$

Proměnná x_{ij} je bivalentní proměnná nabývající hodnoty 1 v případě, že vozidlo jede do místa j z místa i , hodnoty 0 v opačném případě. Účelová funkce (3.66) představuje celkovou vzdálenost ujetou vozidlem. Soustava podmínek (3.67) a (3.68) zajišťuje, že každé místo je navštíveno právě jednou. Nerovnosti (3.69) s proměnnými u_i jsou opatřením proti vytváření parciálních cyklů. Vzhledem k tomu, že jednotlivá místa jsou záměrně označena tak, že sudé

indexy představují sídla zákazníků a liché indexy místa doručení zásilek³¹, je podmínkami (3.70) zaručeno, aby vozidlo nedoručilo zásilku dříve než ji vyzvedne.

Optimální řešení³² úlohy uvedené v příkladu 3.6 je znázorněno na obr. 3.12. V tomto konkrétním příkladě bude zásilka zákazníka č. 6 vyzvednuta jako první a doručena do místa č. 7 jako poslední.



Obr. 3.12 – Optimální trasa pro doručení zásilek

3.4.2 Statická úloha kurýrní služby s časovými okny

Aplikace, která je bližší reálným požadavkům firem, zahrnuje pro každého zákazníka interval, během kterého je nutné zásilku vyzvednout, a interval, v němž je nutné tuto zásilku doručit do určeného místa. Kromě vzdáleností c_{ij} je nutné znát i dojezdové doby t_{ij} mezi všemi místy. Matematický model, v němž je použito podobné značení jako v modelu (3.66) – (3.72), lze pak zapsat následovně:

$$\text{minimalizovat } z = \sum_{i=1}^{2n+1} \sum_{j=1}^{2n+1} c_{ij} x_{ij}, \quad (3.73)$$

za podmínek

$$\sum_{j=1}^{2n+1} x_{ij} = 1, \quad i = 1, 2, \dots, 2n+1, \quad (3.74)$$

$$\sum_{i=1}^{2n+1} x_{ij} = 1, \quad j = 1, 2, \dots, 2n+1, \quad (3.75)$$

$$e_i \leq \tau_i \leq l_i, \quad i = 2, 3, \dots, 2n+1, \quad (3.76)$$

$$\tau_i + t_{ij} - M(1 - x_{ij}) \leq \tau_j, \quad i = 1, 2, \dots, 2n+1, \quad j = 2, 3, \dots, 2n+1, \quad i \neq j, \quad (3.77)$$

$$\tau_{2i} \leq \tau_{2i+1}, \quad i = 1, 2, \dots, n, \quad (3.78)$$

$$\tau_1 = 0, \quad (3.79)$$

³¹ Každé místo doručení má vždy o jedničku vyšší index než sídlo zákazníka, který do tohoto místa doručení posílá svou zásilku.

³² Pro přehlednost byly z grafu vypuštěny hrany označující dvojice sídla zákazníků a místa doručení zásilek.

$$\tau_i \geq 0, \quad i = 2, 3, \dots, 2n+1, \quad (3.80)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, 2n+1. \quad (3.81)$$

Proměnná τ_i udává okamžik, ve kterém vozidlo navštíví místo i . Účelová funkce (3.73) představuje celkovou vzdálenost ujetou vozidlem. Podmínky (3.74) a (3.75) zajišťují, že každé místo je navštíveno právě jednou. Hodnoty e_i a l_i v soustavě nerovností (3.76) představují nejdříve možný a nejpozději přípustný termín vyzvednutí zásilky (v případě sudých indexů), resp. jejího doručení (v případě lichých indexů). Nerovnosti (3.77) s proměnnými u_i jsou opatřením proti vytváření parciálních cyklů, parametr M představuje vysokou konstantu. Soustava podmínek (3.78) zaručuje, že každá zásilka bude doručena později než bude vyzvednuta.

I v tomto případě může vozidlo navštívit mezi vyzvednutím a doručením konkrétní zásilky několik jiných míst. Jak již bylo naznačeno, podmínky (3.76) týkající se časových oken mohou být v reálné praxi zadány i jiným způsobem. Jednou z možností je definovat nejdříve možný termín pro vyzvednutí zásilky a nejpozději přípustný termín pro její doručení:

$$e_{2i} \leq \tau_{2i}, \quad i = 1, 2, \dots, n, \quad (3.82)$$

$$\tau_{2i+1} \leq l_{2i+1}, \quad i = 1, 2, \dots, n. \quad (3.83)$$

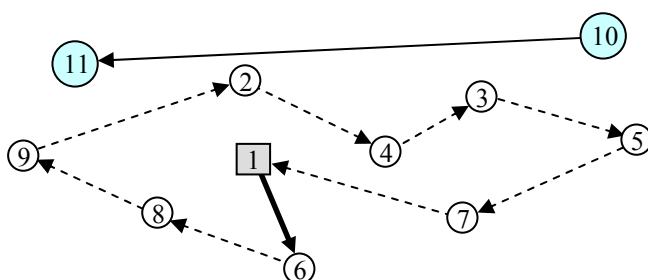
Podobně jako u standardní úlohy obchodního cestujícího s časovými okny popsané v předchozích kapitolách, lze i v případě kurýrní služby použít penalizaci za nedodržení odpovídajících termínů. Vzhledem k tomu, že penalizaci lze velice snadno zahrnout právě s použitím předchozího odvození, nebudeme tyto modely uvádět a soustředíme se na dynamické rozšíření úlohy kurýrní služby.

3.4.3 Dynamická úloha kurýrní služby

Výsledkem statické úlohy je optimální trasa vozidla, které postupně vyzvedává a doručuje zásilky na předem známých místech. Během jízdy vozidla se může vyskytnout požadavek nového zákazníka na vyzvednutí a doručení jeho zásilky. Tyto dvě operace jsou pak zařazeny do stávající trasy vozidla pomocí heuristického algoritmu (např. vkládací algoritmus) či re-optimalizace zbývajících částí trasy. Podívejme se podrobněji na dynamickou úlohu kurýrní služby bez časových oken.

Příklad 3.7

Na obr. 3.13 je znázorněna optimální trasa vozidla, které má za úkol doručit čtyři zásilky podle obr. 3.10. V okamžiku, kdy vozidlo směřuje k zákazníkovi č. 6, vznikne nový požadavek zákazníka č. 10 na doručení jeho zásilky do místa č. 11. Pro zařazení tohoto požadavku do trasy vozidla bude použit vkládací algoritmus a re-optimalizace zbývajících částí trasy.

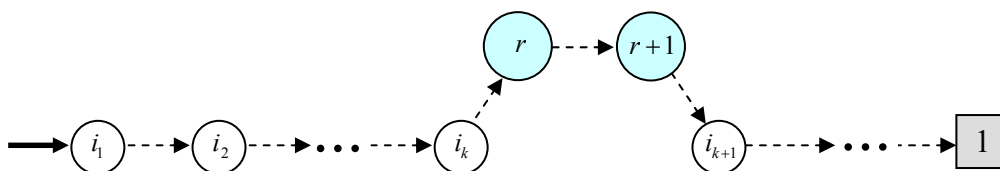


Obr. 3.13 – Nově vzniklý požadavek

Vkládací algoritmus

Při použití vkládacího algoritmu je nutné respektovat důležité omezení, které určuje, že místo doručení zásilky zákazníka lze vložit na trase až za jeho sídlo. Necht' $U_N = \{i_1, i_2, \dots, i_m\}$ je posloupnost m míst ($i_m = 1$), která vozidlo musí ještě navštívit podle naplánované trasy. Označme indexem r sídlo nového zákazníka a indexem $r+1$ místo, do kterého má být doručena jeho zásilka. Bude-li sídlo zákazníka vloženo mezi místa i_k a i_{k+1} , potom místo určení musí být vloženo do trasy za sídlo zákazníka. Mohou nastat dvě situace:

(1) Sídlo nového zákazníka r je vloženo mezi místa i_k a i_{k+1} a místo doručení bezprostředně za zákazníka r , tj. před místo i_{k+1} (obr. 3.14).



Obr. 3.14 – Vložení nového zákazníka a místa doručení zásilky bezprostředně za sebe

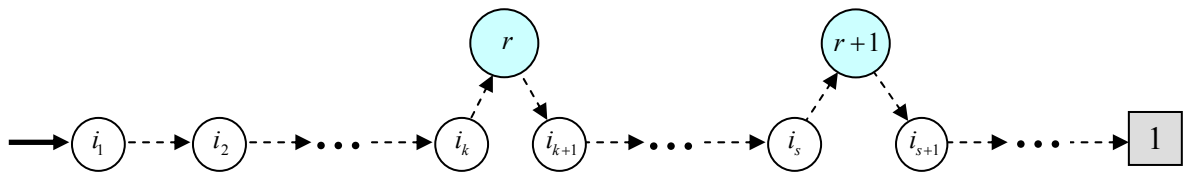
Prodloužení stávající trasy po vložení dvou nových míst bezprostředně za sebe lze pak vypočítat jako

$$\Delta z_k = c_{i_k, r} + c_{r, r+1} + c_{r+1, i_{k+1}} - c_{i_k, i_{k+1}}, \quad k = 1, 2, \dots, m-1. \quad (3.84)$$

Cílem je nalézt takový index t , pro který platí:

$$\Delta z_t^1 = \min_{k=1, 2, \dots, m-1} \Delta z_k. \quad (3.85)$$

(2) Sídlo nového zákazníka r je vloženo mezi místa i_k a i_{k+1} a místo doručení zásilky $r+1$ mezi místa i_s a i_{s+1} , přičemž prvním zákazníkem, za nějž může být místo doručení vloženo, je zákazník i_{k+1} (obr. 3.15). Místem i_k nemůže být místo i_{m-1} , neboť v tomto případě by se jednalo o předchozí situaci, kdy jsou místa r a $r+1$ vložena bezprostředně za sebe.



Obr. 3.15 – Vložení nového zákazníka a místa doručení zásilky mezi různá místa na trase

Prodloužení trasy po vložení sídla nového zákazníka a místa doručení zásilky je

$$\Delta z_{ks} = c_{i_k, r} + c_{r, i_{k+1}} - c_{i_k, i_{k+1}} + c_{i_s, r+1} + c_{r+1, i_{s+1}} - c_{i_s, i_{s+1}}, \quad (3.86)$$

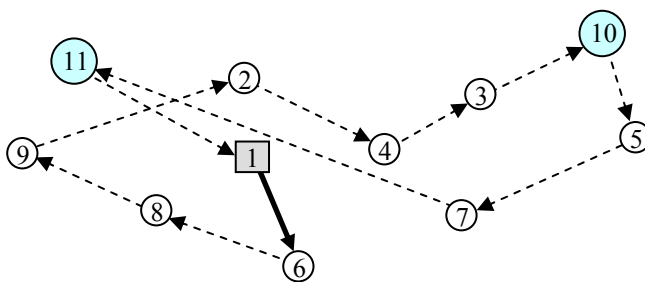
$$k = 1, 2, \dots, m-2, \quad s = k+1, k+2, \dots, m-1.$$

Cílem je nalézt takové indexy t a p , pro které platí:

$$\Delta z_{tp}^2 = \min_{\substack{k=1, 2, \dots, m-2 \\ s=k+1, k+2, \dots, m-1}} \Delta z_{ks}. \quad (3.87)$$

Výběr nižší z hodnot (3.85) a (3.87) určí nejvýhodnější strategii vložení nového zákazníka a místa doručení jeho zásilky.

Bude-li v příkladu 3.7 použit uvedený vkládací algoritmus, pak sídlo zákazníka č. 10 bude zařazeno do trasy mezi místa č. 3 a 5, místo doručení zásilky č. 11 mezi místo č. 7 a výchozí místo č. 1 (obr. 3.16).



Obr. 3.16 – Aplikace vkládacího algoritmu

Re-optimalizace

Re-optimalizační algoritmus spočívá v nalezení optimálního řešení úlohy, ve kterém vozidlo po přijetí nového požadavku na doručení zásilky navštíví všechna dosud nenavštívená místa, vrátí se do výchozího místa a ujede přitom minimální vzdálenost. Matematický model pro tuto úlohu lze formulovat následujícím způsobem:

$$\text{minimalizovat } z = \sum_{i \in U_N} \sum_{j \in U_N} c_{ij} x_{ij}, \quad (3.88)$$

za podmínek

$$\sum_{j \in U_N} x_{ij} = 1, \quad i \in U_N, \quad (3.89)$$

$$\sum_{i \in U_N} x_{ij} = 1, \quad j \in U_N, \quad (3.90)$$

$$u_i - u_j + |U_N| \cdot x_{ij} \leq |U_N| - 1, \quad i \in U_N, \quad j \in U_N - \{1\}, \quad i \neq j, \quad (3.91)$$

$$u_{i_{2k-1}} \leq u_{i_{2k}}, \quad i_{2k-1}, i_{2k} \in \omega, \quad k = 1, 2, \dots, \frac{|\omega|}{2} - 1, \quad (3.92)$$

$$u_1 = 0, \quad (3.93)$$

$$x_{1j_{next}} = 1, \quad (3.94)$$

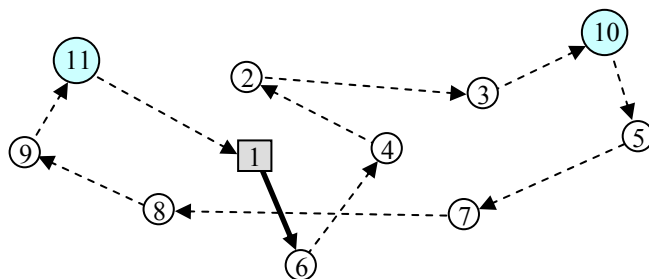
$$x_{ij} \in \{0, 1\}, \quad i, j \in U_N. \quad (3.95)$$

V modelu opravíme původní matici vzdáleností tak, že vzdálenost $c_{1j_{next}}$ mezi výchozím místem č. 1 a zákazníkem j_{next} , ke kterému vozidlo směřuje v okamžiku vzniku nového požadavku, změním na hodnotu odpovídající délce dosud absolvované trasy z výchozího místa k zákazníkovi j_{next} .

V množině U_N jsou místa, která musí vozidlo ještě navštívit, včetně místa j_{next} , výchozího místa č. 1, sídla nového zákazníka a místa doručení jeho zásilky. Počet všech míst

je dán hodnotou $|U_N|$. Tato množina obsahuje místa doručení dvojího typu: ty, jejichž zásilky nebyly vyzvednuty, a ty, které čekají na doručení zásilek vyzvednutých před tím, než vozidlo dorazí do místa j_{next} . Necht' ω je rostoucí posloupnost indexů všech sídel zákazníků a míst doručení obsažených v množině U_N , jejichž zásilky nebyly doposud vyzvednuty³³. Počet prvků této posloupnosti $|\omega|$ je vždy sudé číslo. Soustava nerovností (3.92) je formulována právě pro prvky této posloupnosti a zaručuje, že v optimální trase bude zásilka nejprve vyzvednuta a teprve potom doručena. Tyto podmínky mají stejný význam jako omezení (3.70) ve statické úloze.

V příkladu 3.7 je $U_N = \{1,2,\dots,11\}$, tj. množina obsahuje všechna místa. Index místa, do kterého vozidlo směřuje v okamžiku vzniku nového požadavku, je $j_{next} = 6$. Posloupnost míst, jejichž zásilky nebudou po návštěvě zákazníka č. 6 ve vozidle (kromě zásilky právě tohoto zákazníka), je $\omega = \{2,3,4,5,8,9,10,11\}$. Optimalizací modelu (3.88) – (3.95) získáme trasu znázorněnou na obr. 3.17.



Obr. 3.17 – Re-optimalizace trasy

Dynamická úloha kurýrní služby s časovými okny vychází z optimálního řešení statické úlohy popsané v části 3.4.2. Vzhledem k tomu, že vkládací algoritmus i re-optimalizace jsou velmi podobné postupům pro řešení dynamických úloh bez časových oken, nebudeme se dále věnovat jejich popisu. Při odvození těchto algoritmů lze využít předpokladů a postupů uvedených v předchozí kapitole.

³³ Pokud je místem, do kterého právě směřuje vozidlo, sídlo některého ze zákazníků, kde dojde k vyzvednutí zásilky, nebude toto místo v posloupnosti zahrnuto, neboť platí omezení (3.94).

4. Dynamická úloha s více obchodními cestujícími

V předchozích modelech jsme uvažovali situace, kdy pro obsluhu zákazníků bylo určeno jediné vozidlo. V reálných aplikacích ovšem obsluhu zákazníků zajišťuje většinou více vozidel, a proto je zapotřebí využít složitějších modelů, které se již významně přibližují potřebám praxe a reálnému chování firem zajišťujících plnou spokojenost svých zákazníků.

4.1 Úloha s více obchodními cestujícími

V literatuře lze najít celou řadu modifikací problému s více obchodními cestujícími (**mTSP**)¹. Základní klasifikaci těchto úloh předkládá Bektas (2006):

1. Jedno nebo více výchozích míst. Všechna vozidla mohou vyjždět ze stejného výchozího místa nebo z několika výchozích míst, jejichž vzájemná vzdálenost je předem známá. V každém z míst pak může být buď jedno nebo více vozidel. Vozidla se po obsluze všech zákazníků mohou vrátit do svého výchozího místa nebo do některého jiného z výchozích míst. V některých aplikacích je nutné, aby na konci dne byl počet vozidel v jednotlivých místech stejný jako na počátku dne, jiné aplikace tento stav nevyžadují.

2. Počet vozidel. Počet vozidel může být v některých úlohách omezenou proměnnou nebo být definován a-priori jako konstantní.

3. Fixní poplatky na vozidlo. Pokud je počet disponibilních vozidel v problému proměnný, pak je většinou použití vozidla spojeno s konkrétní fixní částkou, např. v podobě mzdy, kterou musí firma platit řidiči vozidla, pojištění vozidla apod. Počet použitých vozidel pro obsluhu všech zákazníků se pak může stát součástí minimalizované nákladové funkce.

4. Časová okna. Stejně jako v úloze jediného obchodního cestujícího, také v případě většího počtu vozidel hrají svoji roli časová okna definovaná pro jednotlivé zákazníky. Zařazení dalších vozidel tak může výrazně přispět k řešení úloh, které jsou v případě jediného vozidla neřešitelné.

5. Další speciální omezení. Tato omezení se mohou týkat maximálního počtu zákazníků, které smí jedno vozidlo navštívit, maximální nebo minimální vzdálenosti, kterou může či musí vozidlo ujet apod.

¹ Multiple Traveling Salesman Problem. Kromě zkratky mTSP lze v literatuře najít m-TSP, případně MTSP.

4.2 Aplikace úlohy s více obchodními cestujícími

Jak bylo uvedeno výše, zavedení většího počtu vozidel přibližuje model reálné situací, neboť lze jen těžko předpokládat, že jediné vozidlo je schopné uspokojit stále náročnější požadavky zákazníků, resp. nároky související s rozšiřující se klientelou. Zde zmíníme alespoň některé reálné aplikace uvedeného typu okružní úlohy.

Jednou z úspěšných aplikací je **svoz vkladů z poboček do centrální banky** (Svestka and Huckfeldt, 1973 in Bektas, 2006). Problém spočívá v určení tras jednotlivých posádek s cílem minimalizovat celkové náklady. Další dvě publikované aplikace (Lenstra and Rinnooy Kan, 1975 in Bektas, 2006) se týkají tras technických týmů zajišťujících funkčnost **telefonních budek** v severním Holandsku a obsluhy 200 **poštovních schránek** v Utrechtu. V obou případech bylo cílem minimalizovat počet použitých vozidel. Jinou aplikací tohoto typu byl případ rozvržení jízd **fotografů**, kteří měli za úkol navštívit základní a střední školy (Zhang, 1999 in Bektas, 2006).

Dalším typickým problémem je úloha (Angel et al., 1972 in Bektas, 2006), v níž je cílem získat takový rozpis **linek školních autobusů**, aby počet navržených tras byl minimální. Žádný z autobusů nesmí být přeplněn kvůli bezpečnosti cestujících žáků a v žádném případě nesmí dojít k překročení požadovaného rozpisu vyložení žáků na určených místech.

Ve většině úloh je cílem rozhodnout, **kolik vozidel** nasadit na zajištění požadavků tak, aby jejich počet byl minimální, stejně tak jako **celková délka všech uskutečněných tras**. Vzhledem k tomu, že se k základní formulaci úlohy často přidávají i specifické podmínky, ať již v podobě časových oken nebo požadavků na rychlost obsluhy aj., stávají se podobné úlohy stále žádanějšími adepty na úspěšné vyřešení.

Dalšími úlohami rozšiřujícími problém více obchodních cestujících je např. **vyrovnávání pracovní zátěže mezi jednotlivými vozidly** (Okonjo-Adigwe, 1988 in Bektas, 2006). V této úloze jsou přidána další omezení ve formě dolních a horních mezí doby jízdy a celkové hmotnosti jednotlivých vozidel. Jako poslední příklad uveďme problém **noční bezpečnostní služby** (Calvo and Cordone, 2003 in Bektas, 2006), spočívající v provádění pravidelných kontrol předem určených objektů.

4.3 Úloha s více obchodními cestujícími s minimalizací celkové ujeté vzdálenosti

4.3.1 Úloha s jedním výchozím místem

Vzhledem k zaměření této práce se soustředíme na následující formulaci matematického modelu (Bektas, 2006), která je založena na Miller-Tucker-Zemlinově formulaci² úlohy obchodního cestujícího TSP:

$$\text{minimalizovat } z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}, \quad (4.1)$$

za podmínek

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 2, 3, \dots, n, \quad (4.2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 2, 3, \dots, n, \quad (4.3)$$

$$\sum_{j=2}^n x_{1j} = K, \quad (4.4)$$

$$\sum_{j=2}^n x_{j1} = K, \quad (4.5)$$

$$u_i - u_j + p x_{ij} \leq p - 1, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j, \quad (4.6)$$

$$x_{ii} = 0, \quad i = 1, 2, \dots, n, \quad (4.7)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n, \quad (4.8)$$

kde K je počet vozidel³, n je počet míst, která vozidla musí projet (včetně výchozího místa označeného indexem 1) a c_{ij} představuje vzdálenost mezi místy i a j ; bivalentní proměnná x_{ij} nabývá hodnoty 1 v případě, že vozidlo jede do místa j z místa i , hodnoty 0 v opačném případě.

Cílem je minimalizovat délku všech tras ujetých všemi vozidly (4.1). Soustavy podmínek (4.2) a (4.3) zajišťují, že každé místo je navštíveno právě jednou, podmínky (4.4)

² Dalšími popsanými modely (Bektas, 2006) je Laporte-Norbertův model, dále modely založené na hledání centru grafu a toku na síti.

³ Úlohy s více obchodními cestujícími se v literatuře objevují také pod názvem m -Traveling Salesman Problem, ve kterém m označuje počet obchodních cestujících, resp. vozidel. V této práci je počet vozidel ve všech modelech označován jako K , neboť symbol m je v textu použit pro jiný účel.

a (4.5) určují, že z výchozího místa vyjede přesně K vozidel a stejný počet vozidel se opět do výchozího místa vrátí. Podmínky (4.7) zamezí u každého místa vytváření smyček. Soustava podmínek (4.6) s proměnnými u_i je opatřením proti vytváření parciálních cyklů, p určuje maximální počet zákazníků, které může obsloužit jedno vozidlo (Miller et al., 1960 in Bektas, 2006).

Pokud není parametr p znám, jinou možností, jak zamezit vytváření parciálních cyklů⁴, je použít následující soustavu omezujících podmínek (Gavish, 1976 in Bektas, 2006):

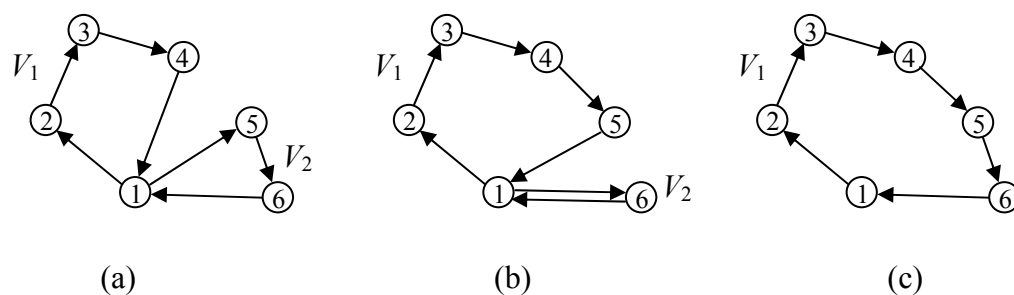
$$u_i - u_j + (n - K)x_{ij} \leq n - K - 1, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j. \quad (4.9)$$

V uvedeném matematickém modelu jsou zahrnuta silná omezení (4.4) a (4.5), která striktně požadují, aby byla využita všechna dostupná vozidla⁵. V případě uvolnění této podmínky v tom smyslu, že některá vozidla z disponibilního vozového parku mohou zůstat nevyužita, je optimálním řešením využití jen jediného vozidla. Tato skutečnost platí ovšem za předpokladu, že v úloze jsou splněny trojúhelníkové nerovnosti⁶:

$$c_{is} + c_{sj} \geq c_{ij}, \quad i, j, s = 1, 2, \dots, n, \quad i \neq j \neq s. \quad (4.10)$$

Příklad 4.1

Předpokládáme 2 vozidla V_1 a V_2 umístěná ve výchozím místě č. 1 (obr. 4.1).



Obr. 4.1 – Výjezd jednoho či dvou vozidel

Situace (a) a (b) jsou zjevně přípustnými řešeními matematického modelu (4.1) – (4.8). Případ (c) je z hlediska nejkratší celkové ujeté vzdálenosti jednoznačně nejlepším řešením,

⁴ Vytvářením parciálních cyklů u tohoto typu úloh rozumíme, že žádné vozidlo nesmí uskutečnit více než jeden okruh. V takto definovaném matematickém modelu je tedy počet tras roven přesně počtu vozidel, tj. hodnotě K .

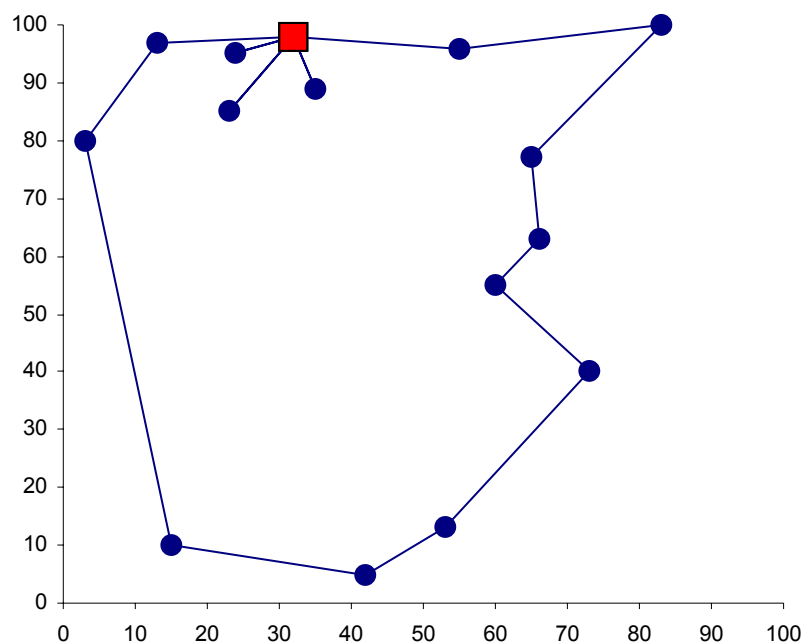
⁵ Uvedený matematický model lze pochopitelně aplikovat pouze na úlohy, ve kterých je $K \leq n$.

⁶ Trojúhelníkové nerovnosti (4.10) jsou v reálných úlohách většinou splněny. V této práci se případy, kdy tyto nerovnosti neplatí, zabývat nebudeme.

tj. optimální je, aby vozidlo V_2 vůbec nevyjíždělo na trasu; toto řešení je však nepřijatelným řešením uvedeného matematického modelu.

Příklad 4.2

V příloze č. 4 je uveden příklad se 14 zákazníky a 4 vozidly. Jako optimální bylo nalezeno řešení, ve kterém jedno z vozidel obsluhuje 11 zákazníků. Zbývající 3 zákazníci jsou pak obsluženi dalšími 3 vozidly (každé vozidlo obsluhuje právě jednoho zákazníka – viz obr. 4.2).



Obr. 4.2 – Čtyři vozidla vyjíždějící z jednoho výchozího místa

4.3.2 Úloha s několika výchozími místy

Při splnění uvedených nerovností (4.10) je zajímavým problémem úloha, kdy vozidla nevyjíždějí ze stejného výchozího místa, ale existuje několik výchozích míst či stanišť, ze kterých vozidla mohou, avšak nemusí vyjíždět. Cílem je stejně jako v předchozím modelu minimalizovat délku všech tras ujetých všemi vozidly:

$$\text{minimalizovat } z = \sum_{i=1}^{K+n} \sum_{j=1}^{K+n} c_{ij} x_{ij}, \quad (4.11)$$

za podmínek

$$\sum_{i=1}^{K+n} x_{ij} = 1, \quad j = K+1, K+2, \dots, K+n, \quad (4.12)$$

$$\sum_{i=1}^{K+n} x_{ji} = 1, \quad j = K+1, K+2, \dots, K+n, \quad (4.13)$$

$$\sum_{j=K+1}^{K+n} x_{ij} \leq 1, \quad i = 1, 2, \dots, K, \quad (4.14)$$

$$\sum_{j=K+1}^{K+n} x_{ij} = \sum_{j=K+1}^{K+n} x_{ji}, \quad i = 1, 2, \dots, K, \quad (4.15)$$

$$u_i - u_j + nx_{ij} \leq n - 1, \quad i, j = K+1, K+2, \dots, K+n, \quad i \neq j, \quad (4.16)$$

$$x_{ij} = 0, \quad i, j = 1, 2, \dots, K, \quad (4.17)$$

$$x_{ii} = 0, \quad i = K+1, K+2, \dots, K+n, \quad (4.18)$$

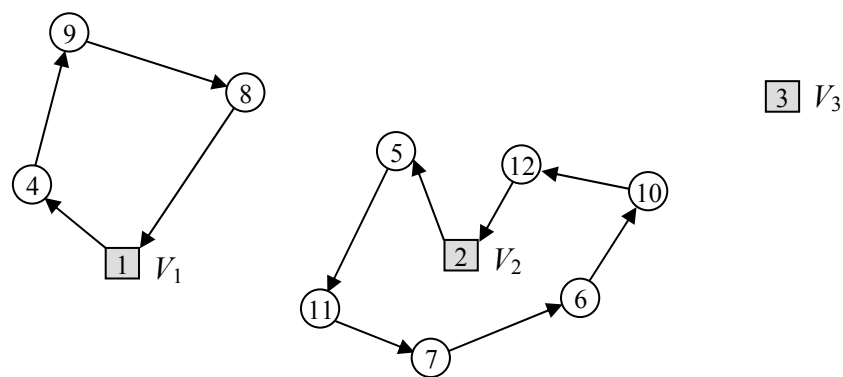
$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, K+n. \quad (4.19)$$

V uvedeném modelu předpokládáme, že existuje K stanovišť. V každém z nich je právě jedno vozidlo, které může, ale nemusí vyjet. Dále uvažujeme n zákazníků. Bivalentní proměnná x_{ij} nabývá hodnoty 1 v případě, že některé vozidlo pojede přímo z místa i do místa j , hodnoty 0 v opačném případě. Z jednotlivých podmínek je zřejmé, že existuje celkem $K+n$ míst, indexy $1, 2, \dots, K$ jsou zvoleny pro výchozí místa, indexy $K+1, K+2, \dots, K+n$ pro sídla zákazníků.

Soustavy rovnic (4.12) a (4.13) zajišťují, že každý zákazník bude navštíven právě jedním vozidlem. Podmínky (4.14) určují, že z každého výchozího místa vyjede maximálně jedno vozidlo. Rovnice (4.15) zaručují, že počet vozidel vyjíždějících z každého výchozího místa je roven počtu vozidel, které se do tohoto místa vracejí, v našem případě je tento počet roven 0 nebo 1. Soustava nerovností (4.16) tvoří smyčkové podmínky. Soustava rovnic (4.17) zakazuje pohyb vozidel mezi jednotlivými výchozími místy navzájem, podmínky (4.18) zabraňují u každého zákazníka vytváření smyček.

Příklad 4.3

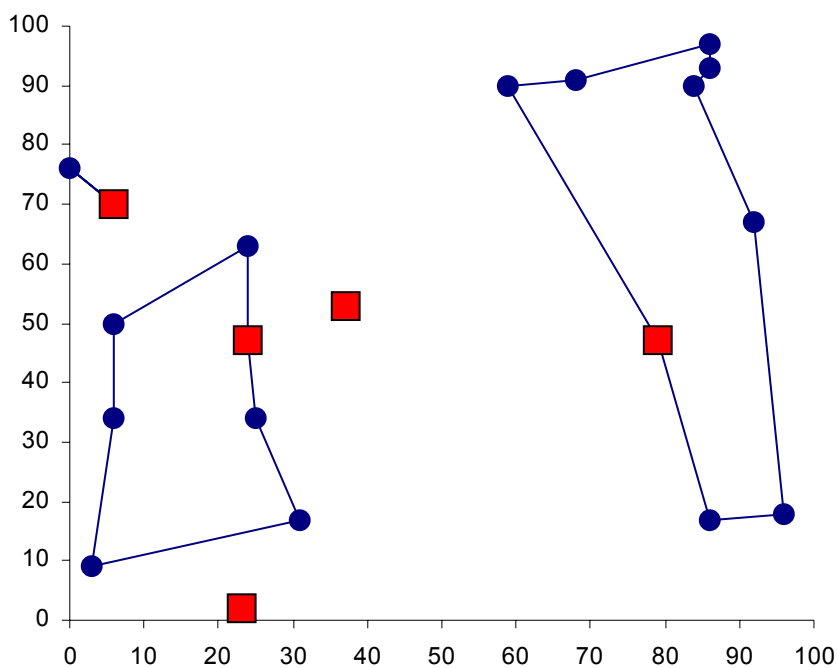
Na obr. 4.3 je uveden příklad se třemi výchozími místy s vozidly V_1 , V_2 a V_3 . Pokud je cílem minimalizovat celkovou vzdálenost, kterou všechna tři vozidla dohromady ujedou, pak optimálním řešením je zřejmě strategie, kdy vozidlo V_3 nebude pro obsluhu zákazníků vůbec použito.



Obr. 4.3 – Úloha se třemi výchozími místy

Příklad 4.4

V příloze č. 5 je vygenerováno 5 výchozích míst a 15 zákazníků. Optimální řešení této úlohy je graficky znázorněno na obr. 4.4. Obsluhu zákazníků budou zajišťovat tři vozidla, zbývající dvě vozidla nevyjedou.



Obr. 4.4 – Příklad s pěti výchozími místy

4.4 Úloha s více obchodními cestujícími s minimalizací času potřebného k obsluze všech zákazníků

Nyní předpokládejme reálnou situaci, kdy firma potřebuje obsloužit všechny zákazníky za minimální dobu. V následujících modelech bude do doby, za níž vozidlo obslouží jemu

přidělené zákazníky, započítávána i cesta od posledního zákazníka zpět do výchozího místa. Jako reálnou aplikaci lze uvést situaci, kdy je např. nutné svézt poštovní zásilky od všech zákazníků co nejdříve do výchozího místa pro jejich další zpracování či hromadnou distribuci.

4.4.1 Úloha s jedním výchozím místem

Nejprve předpokládejme situaci, v níž existuje jediné výchozí místo, ve kterém je umístěno K vozidel. Pro nalezení optimálního řešení úlohy formulujeme následující matematický model:

$$\text{minimalizovat } z = T, \quad (4.20)$$

za podmínek

$$\sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij}^k \leq T, \quad k = 1, 2, \dots, K, \quad (4.21)$$

$$\sum_{k=1}^K \sum_{j=1}^n x_{ij}^k = 1, \quad i = 2, 3, \dots, n, \quad (4.22)$$

$$\sum_{k=1}^K \sum_{i=1}^n x_{ij}^k = 1, \quad j = 2, 3, \dots, n, \quad (4.23)$$

$$\sum_{j=2}^n x_{1j}^k \leq 1, \quad k = 1, 2, \dots, K, \quad (4.24)$$

$$\sum_{i=1}^n x_{ij}^k = \sum_{i=1}^n x_{ji}^k, \quad j = 2, 3, \dots, n, \quad k = 1, 2, \dots, K, \quad (4.25)$$

$$u_i - u_j + n x_{ij}^k \leq n - 1, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j, \quad k = 1, 2, \dots, K, \quad (4.26)$$

$$x_{ii}^k = 0, \quad i = 1, 2, \dots, n, \quad k = 1, 2, \dots, K, \quad (4.27)$$

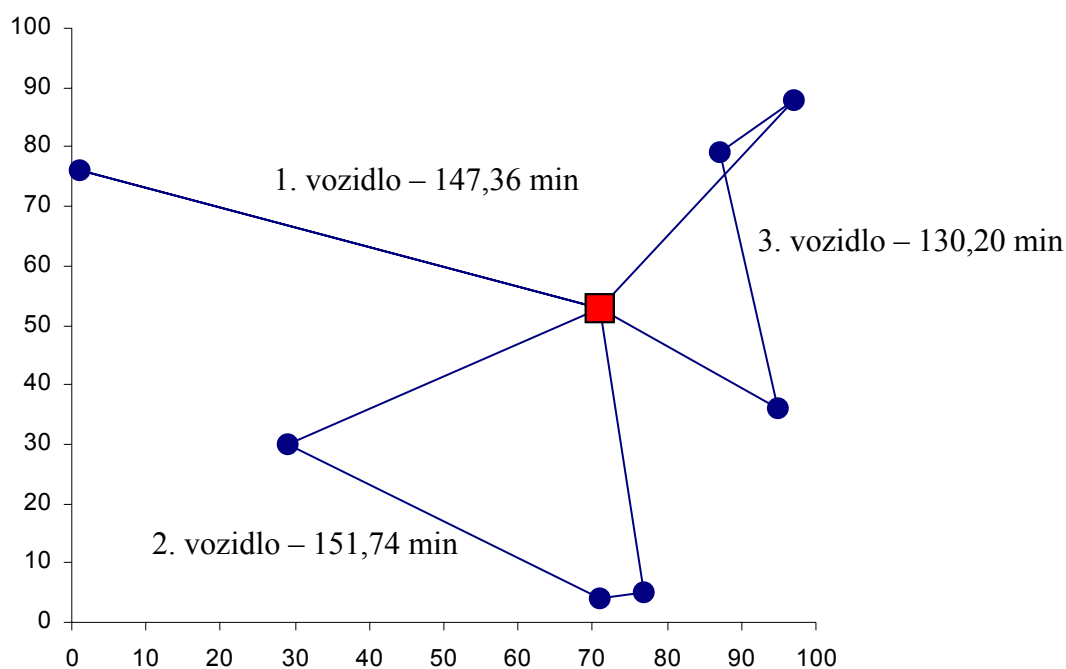
$$x_{ij}^k \in \{0, 1\}, \quad i, j = 1, 2, \dots, n, \quad k = 1, 2, \dots, K. \quad (4.28)$$

V modelu jsme místo vzdáleností c_{ij} mezi místy i a j použili parametry t_{ij} představující jejich časovou vzdálenost. Bivalentní proměnná x_{ij}^k nabývá hodnoty 1 v případě, že vozidlo k pojede přímo z místa i do místa j , hodnoty 0 v opačném případě. Proměnná T představuje horní mez doby trvání jízdy vozidel ve všech trasách; cílem je pak tuto hodnotu minimalizovat, tj. najít takový rozvrh tras, kdy doba nejdelší trasy bude rovna právě hodnotě T (4.20). Vztah mezi délkou konkrétní trasy a hodnotou T je určen podmínkami (4.21). Hodnota K představuje již zmíněnou velikost vozového parku.

Soustavy rovnic (4.22) a (4.23) zajišťují, že každé míso bude navštíveno právě jednou. Nerovnice (4.24) určují, že každé vozidlo odjede z výchozího místa nejvýše jednou, čímž je zajištěno, že některá vozidla nemusí vyjet. Soustava bilančních rovnic (4.25) zajišťuje, aby vozidlo, které přijede k danému zákazníkovi, od tohoto zákazníka také odjelo. Nerovnosti (4.26) jsou smyčkovými podmínkami. Omezení (4.27) zabraňují všem vozidlům u všech zákazníků vytváření smyček.

Příklad 4.5

V příloze č. 6 je vygenerováno 7 zákazníků, kteří mají být obslouženi 3 vozidly umístěnými v jednom výchozím místě. Optimální řešení modelu (4.20) – (4.28) aplikovaného na tuto úlohu je znázorněno na obr. 4.5. Minimální doba potřebná k obslužení všech zákazníků, včetně návratu vozidel do výchozího místa, je určena délkou trasy 2. vozidla, tj. hodnotou cca 151,74 min.



Obr. 4.5 – Minimalizace času potřebného k obslužení všech zákazníků

Je zřejmé, že pouze u trasy 2. vozidla je zaručeno, že její délka je optimální, zatímco u trasy 3. vozidla se evidentně nejedná o nejkratší trasu. Aby se firma chovala co nejefektivněji, lze provést dodatečnou optimalizaci jednotlivých tras. Jednou z možností je po určení tras, které optimalizují účelovou funkci (4.20), řešit zvlášť pro každou naplánovanou trasu standardní úlohu obchodního cestujícího. Přestože je počet zákazníků v každé trase výrazně nižší než v původní úloze a nenastává tudíž problém s řešitelností několika úloh

obchodních cestujících, je možné použít efektivnější metodu, která vychází z optimálního řešení úlohy (4.20) – (4.28).

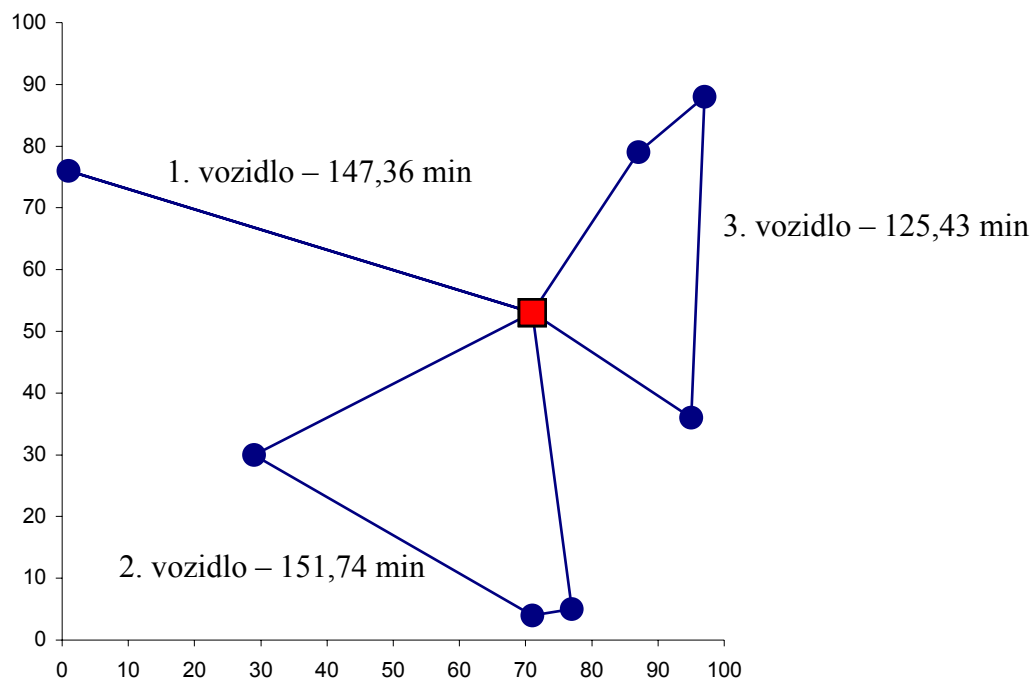
Nechť T_{\min} je optimální hodnota účelové funkce (4.20). Pak řešíme obdobnou úlohu jako (4.20) – (4.28), v níž je místo soustavy omezujících podmínek (4.21) definována tato soustava nerovností:

$$\sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij}^k \leq T_{\min}, \quad k = 1, 2, \dots, K, \quad (4.21a)$$

čímž zaručíme, že řešení zůstane optimální vzhledem k původní úloze s cílem nalézt takové trasy, které obslouží všechny zákazníky v minimálním čase. Abychom našli takové řešení, které bude optimální i v rámci trasy každého vozidla, použijeme následující účelovou funkci:

$$\text{minimalizovat } \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij}^k. \quad (4.20a)$$

V příkladě 4.5 je $T_{\min} = 151,74 \text{ min}^7$; řešení matematického modelu s uvedenými změnami je znázorněno na obr. 4.6.



Obr. 4.6 – Dodatečná optimalizace dílčích tras

⁷ V matematickém modelu musí být použita původní, tj. nezaokrouhlená hodnota.

Optimální délky tras jsou u 1. vozidla, resp. 2. vozidla stejné, tj. cca 147,36 min, resp. 151,74 min, zatímco trasa 3. vozidla se zkrátila z původních cca 130,20 min na 125,43 min.

4.4.2 Úloha s několika výchozími místy

Nyní předpokládejme, že existuje K výchozích míst; v každém z nich je právě jedno vozidlo. Matematický model úlohy, v níž je cílem minimalizovat čas potřebný k obsluze všech zákazníků, je založen na modelech (4.11) – (4.19) a (4.20) – (4.28):

$$\text{minimalizovat } z = T, \quad (4.29)$$

za podmínek

$$\sum_{i=1}^{K+n} \sum_{j=1}^{K+n} t_{ij} x_{ij}^k \leq T, \quad k = 1, 2, \dots, K, \quad (4.30)$$

$$\sum_{k=1}^K \sum_{j=1}^{K+n} x_{ij}^k = 1, \quad i = K + 1, K + 2, \dots, K + n, \quad (4.31)$$

$$\sum_{k=1}^K \sum_{i=1}^{K+n} x_{ij}^k = 1, \quad j = K + 1, K + 2, \dots, K + n, \quad (4.32)$$

$$\sum_{j=K+1}^{K+n} x_{ij}^k \leq 1, \quad k = 1, 2, \dots, K, \quad (4.33)$$

$$\sum_{i=1}^{K+n} x_{ij}^k = \sum_{i=1}^{K+n} x_{ji}^k, \quad j = 1, 2, \dots, K + n, \quad k = 1, 2, \dots, K, \quad (4.34)$$

$$u_i - u_j + n x_{ij}^k \leq n - 1, \quad i, j = K + 1, K + 2, \dots, K + n, \quad i \neq j, \quad k = 1, 2, \dots, K, \quad (4.35)$$

$$x_{ij}^k = 0, \quad i, j, k = 1, 2, \dots, K, \quad (4.36)$$

$$x_{ij}^k = 0, \quad i, k = 1, 2, \dots, K, \quad i \neq k, \quad j = K + 1, K + 2, \dots, K + n, \quad (4.37)$$

$$x_{ii}^k = 0, \quad i = K + 1, K + 2, \dots, K + n, \quad k = 1, 2, \dots, K, \quad (4.38)$$

$$x_{ij}^k \in \{0, 1\}, \quad i, j = 1, 2, \dots, K + n, \quad k = 1, 2, \dots, K. \quad (4.39)$$

Hodnota proměnné T v (4.29) a (4.30) opět představuje v optimálním řešení délku nejdelšího okruhu (z hlediska času). Soustavy rovnic (4.31) a (4.32) zaručují, že každý zákazník bude navštíven právě jednou, podmínky (4.34) zajišťují, že se tato návštěva uskuteční právě jedním vozidlem, tj. zákazník bude přiřazen k právě jedné trase⁸. Nerovnice (4.33) určují, že každé vozidlo odjede z každého výchozího místa nejvýše jednou, tzn. některá vozidla nemusí vyjet. Podmínky (4.35) a (4.38) jsou smyčkovými podmínkami. Rovnice (4.36) zakazují všem

⁸ Navíc je podmínkami (4.34) zaručeno, že vozidlo, které k určitému zákazníkovi přijede, od něj také odjede.

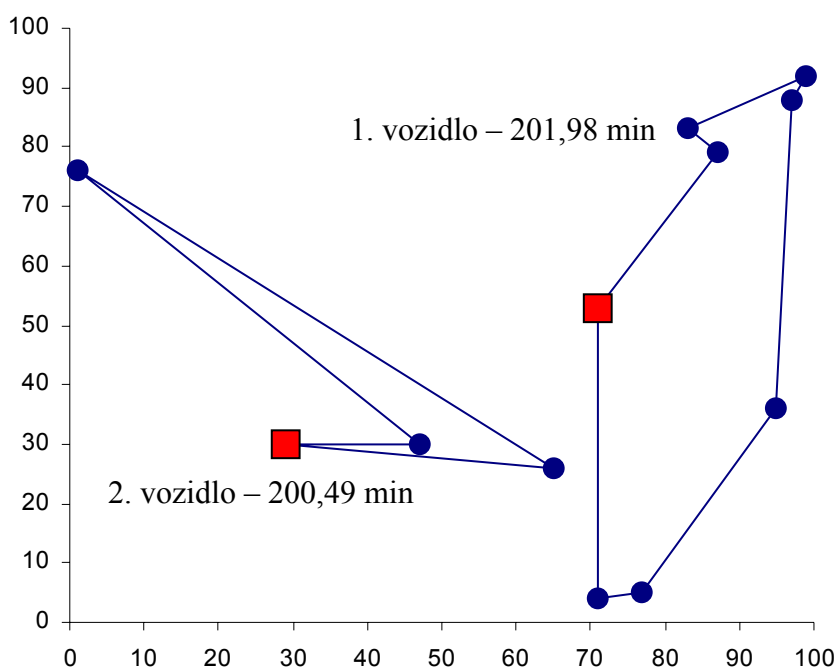
vozidlům navštěvovat výchozí místa jiných vozidel, omezení (4.37) nepřipouští možnost, aby vozidlo k vyjíždělo z jiného místa než k .

Příklad 4.6

V následujícím příkladě (viz příloha č. 7) uvažujeme 2 vozidla ve dvou různých výchozích místech, která mají obsloužit 10 zákazníků. Výsledek optimalizace matematického modelu (4.29) – (4.39) je graficky znázorněn na obr. 4.7. Také v tomto případě lze použít dodatečnou optimalizaci dílčích tras (obr. 4.8), v níž místo (4.21a) a (4.20a) bude definována následující soustava omezujících podmínek a účelová funkce:

$$\sum_{i=1}^{K+n} \sum_{j=1}^{K+n} t_{ij} x_{ij}^k \leq T_{\min}, \quad k = 1, 2, \dots, K, \quad (4.21b)$$

$$\text{minimalizovat } \sum_{k=1}^K \sum_{i=1}^{K+n} \sum_{j=1}^{K+n} t_{ij} x_{ij}^k. \quad (4.20b)$$

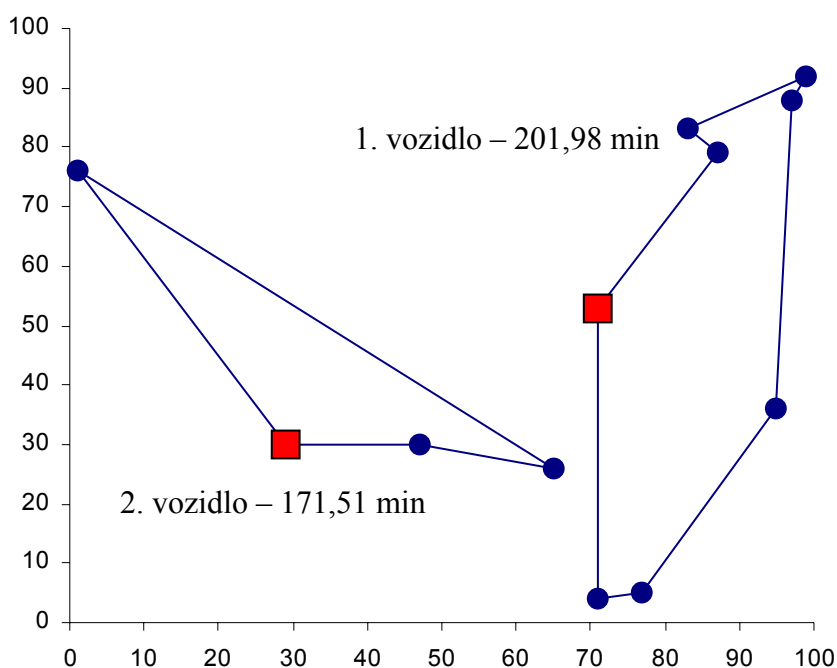


Obr. 4.7 – Minimalizace času potřebného k obslužení všech zákazníků

Minimální čas nutný k obslužení všech zákazníků je cca 201,98 min, což je hodnota, která je dosažena⁹ jako T_{\min} do nerovností (4.21b). Dodatečnou optimalizací došlo ke zkrácení trasy 2. vozidla z původní hodnoty cca 200,49 min na 171,51 min.

⁹ Dosažena musí být nezaokrouhlená hodnota.

V praktických aplikacích se lze často setkat s úlohou, v níž je cílem minimalizovat celkovou ujetou vzdálenost a přitom dodržet předem stanovenou dobu, za kterou musí být všichni zákazníci obslouženi a vozidla se vrátit do výchozího místa, resp. výchozích míst. Jedná se např. o problém svozu zboží, zásilek, bankovních příkazů, pošty apod. do centrálních míst, v nichž dochází k dalšímu zpracování, třídění, případně následné distribuci. Požadavkem je realizovat svoz do určitého časového okamžiku. Pro tuto úlohu lze použít matematický model (4.1) – (4.8), resp. (4.11) – (4.19), přičemž je nutné respektovat omezující podmínky (4.21a), resp. (4.21b), v nichž T_{\min} odpovídá stanovenému nejpozději přípustnému termínu¹⁰ návratu všech vozidel do výchozího místa, resp. míst.



Obr. 4.8 – Dodatečná optimalizace trasy 2. vozidla

4.5 Úloha s více obchodními cestujícími s časovými okny

Uvedené matematické modely úlohy s více obchodními cestujícími jsou z hlediska hledání optimálního řešení velice složité a časově náročné především díky třetímu indexu u proměnné a velkému počtu omezení. Přestože z hlediska praktického využití je přidání časových oken velmi užitečné a žádoucí, v této práci nám jde především o demonstraci základních přístupů k dané problematice než o kompletnost a důslednost z hlediska praxe. Uvažování časových oken souvisí s přidáním další proměnné τ_i^k , kde i představuje index zákazníka, resp. místa

¹⁰ Předpokládáme výjezd vozidel na trasy v nulovém okamžiku.

a k index vozidla. Jestliže vozidlo k navštíví i -tého zákazníka, pak hodnota proměnné τ_i^k určuje okamžik, ve kterém bude návštěva uskutečněna. Spolu s proměnnou τ_i^k je nutné přidat omezení typu (3.25) a (3.26), resp. další omezení související s pohybem vozidel po jednotlivých trasách.

Z praktického hlediska má zavedení časových oken smysl jen v případě, že cílem analýzy je minimalizovat celkovou vzdálenost ujetou všemi vozidly, tj. v úlohách popsanych v části 4.3. V úlohách uvedených v části 4.4, v nichž je cílem minimalizovat čas nezbytný k obslužení všech zákazníků, pozbývá zavedení časových oken výrazný smysl, neboť jejich respektování samo o sobě určuje odhad nejkratší možné doby. Tuto hodnotu lze jednoduše určit následujícím způsobem:

$$T = e_p = \max_i e_i, \quad (4.40)$$

kde e_i je okamžik nejdříve možného začátku obsluhy zákazníka i . Maxima této hodnoty je dosaženo pro zákazníka p ¹¹.

4.6 Dynamická úloha s více obchodními cestujícími

Předpokládejme, že dispečer určil optimální trasy vozidel pro obsluhu všech dosud známých zákazníků. Jak bylo uvedeno u jednotlivých modelů¹², některá vozidla vyjždějí ze svých stanišť, zatímco jiná zůstávají ve výchozím místě nevyužita. Pro další úvahy rozdělíme množinu všech vozidel $\{V_1, V_2, \dots, V_K\}$ na dvě podmnožiny:

- (a) Ω_0 - množina vozidel, která nevyjždějí ze svého staniště,
- (b) Ω_1 - množina vozidel, která vyjždějí ze svého staniště, tj. pro něž je naplánována nějaká trasa.

Zřejmě platí: $\Omega_0 \cup \Omega_1 = \{V_1, V_2, \dots, V_K\}$ a současně $\Omega_0 \cap \Omega_1 = \emptyset$.

V okamžiku, kdy se objeví nový požadavek, musí dispečer určit, které vozidlo pojedě k zákazníkovi a současně stanovit pro toto vozidlo novou trasu. Jak bylo uvedeno v části 4.3.1, úloha s vozidly nacházejícími se v jediném výchozím místě, jejímž cílem je

¹¹ K hodnotě (4.40) je nutné přičíst dobu návratu vozidla od zákazníka p do výchozího místa č. 1.

¹² Kromě modelu (4.1) – (4.8), ve kterém musí vyjet všechna vozidla.

optimalizovat celkovou ujetou vzdálenost, má triviální řešení, ve kterém budou všichni zákazníci obslouženi jediným vozidlem, zatímco ostatní vozidla zůstanou nevyužita. Množina Ω_1 je jednoprvková, množina Ω_0 obsahuje $K-1$ vozidel. Také v případě dynamické úlohy je optimálním řešením situace, kdy nového zákazníka bude obsluhovat právě vozidlo, které je na trase¹³. Dynamický problém je tedy zcela totožný s problémem popsáným ve třetí kapitole, kdy se ve výchozím místě nachází pouze jedno vozidlo; ostatní vozidla zůstávají vždy nevyužita.

Dynamické úlohy s více vozidly představují v každém případě pestrou paletu rozhodovacích problémů. V této práci se soustředíme na následující úlohy, které vycházejí z dříve popsáných rozvozních problémů¹⁴.

Minimalizace celkové ujeté vzdálenosti

- Dynamická úloha s více vozidly ve více výchozích místech.

Minimalizace času potřebného k obslužení všech zákazníků

- Dynamická úloha s více vozidly v jednom výchozím místě.
- Dynamická úloha s více vozidly ve více výchozích místech.

4.6.1 Minimalizace celkové vzdálenosti – více výchozích míst

Optimální řešení statické úlohy odpovídá situaci, kdy množina Ω_1 obsahuje vozidla, která jsou na trase a množina Ω_0 vozidla, která ze svých stanovišť nevyjela.

Vkládací algoritmus

Aplikace vkládacího algoritmu spočívá v rozhodnutí, zda k novému zákazníkovi pojedou některé volné vozidlo či k jeho obslužení bude využito jedno z vozidel, která jsou na trase. V tom případě je zapotřebí určit okamžik, ve kterém bude nový požadavek obslužen, resp. dvojici předem určených zákazníků, mezi kterými bude provedena návštěva nového zákazníka.

¹³ Za předpokladu, že platí trojúhelníkové nerovnosti.

¹⁴ Ve všech modelech připouštíme možnost, že po optimalizaci statické úlohy se známými zákazníky, zůstávají některá vozidla nevyužita, tj. neuvažujeme model (4.1) – (4.8) popsáný v části 4.3.1.

Předpokládejme, že obsluhujícímu vozidlu $V_k \in \Omega_1$ zbývá podle plánu navštívit posloupnost míst $U_k = \{i_1^k, i_2^k, \dots, i_{m_k}^k\}$, kde $i_{m_k}^k = 1$. Prodloužení stávající trasy po vložení nového zákazníka $r \notin U_k$ ($\forall V_k \in \Omega_1$) mezi místa i_s^k a i_{s+1}^k lze vypočítat jako

$$\Delta z_s^k = c_{i_s^k, r} + c_{r, i_{s+1}^k} - c_{i_s^k, i_{s+1}^k}, \quad s = 1, 2, \dots, m_k - 1. \quad (4.41)$$

Nechť výraz (4.41) nabývá svého minima pro zákazníka i_t^q navštíveného vozidlem $V_q \in \Omega_1$:

$$\Delta z_t^q = \min_{\substack{s=1, 2, \dots, m_k-1 \\ \forall V_k \in \Omega_1}} \Delta z_s^k. \quad (4.42)$$

Zbývá rozhodnout, zda není výhodnější kvůli novému zákazníkovi vypravit další vozidlo z některého výchozího místa. Pro každé vozidlo, které není na trase, určíme prodloužení trasy¹⁵ v případě, že toto vozidlo bude zajišťovat obsluhu nového zákazníka, a z těchto hodnot pak najdeme minimum:

$$\Delta z^p = \min_{\forall V_k \in \Omega_0} 2c_{kr}. \quad (4.43)$$

Výběr efektivní strategie je pak proveden na základě porovnání hodnot (4.42) a (4.43):

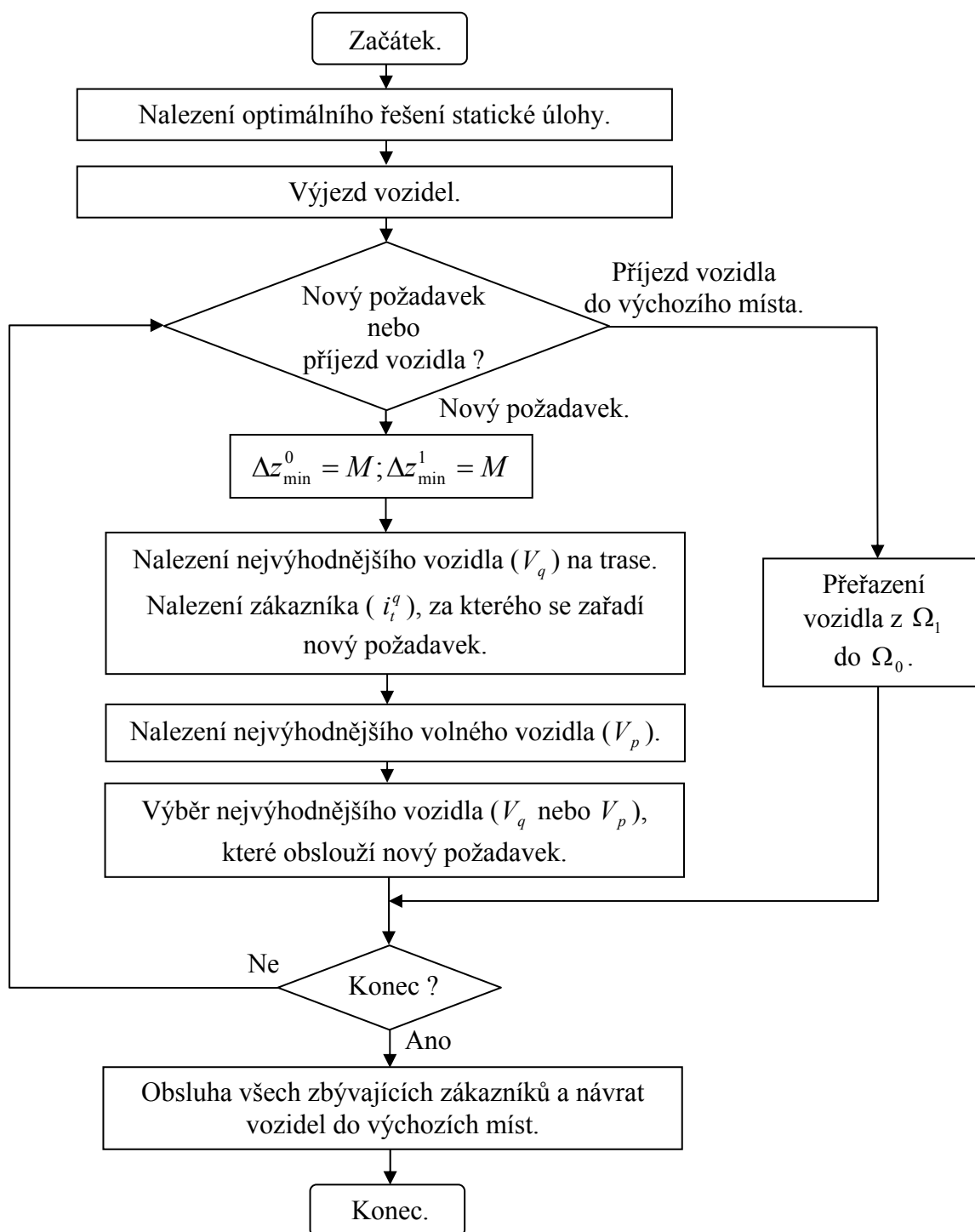
- a) $\Delta z_t^q \leq \Delta z^p$ nebo $\Omega_0 = \emptyset \Rightarrow$ nový zákazník r bude navštíven vozidlem $V_q \in \Omega_1$ bezprostředně po zákazníkovi i_t^q před zákazníkem i_{t+1}^q ,
- b) $\Delta z_t^q > \Delta z^p$ ($\Omega_0 \neq \emptyset$) \Rightarrow k obsluze nového zákazníka bude použito vozidlo $V_p \in \Omega_0$.

Tento postup lze opakovat pro další nově příchozí požadavky. Je zřejmé, že se volná vozidla, pokud je to efektivní, postupně zapojují do obsluhy nových zákazníků, neboli přecházejí z množiny Ω_0 do množiny Ω_1 . Stejně tak ovšem vozidlo, které se po dokončení své plánované trasy vrátí zpět do svého výchozího místa, přechází z množiny Ω_1 do množiny Ω_0 .

Implementace výše uvedeného postupu v praxi je schematicky znázorněna na obr. 4.9 – 4.11. V první fázi je provedena optimalizace statické úlohy, jejímž výsledkem je rozvrh tras všech vozidel, která budou obsluhovat zákazníky. Po jejich výjezdu dispečer čeká

¹⁵ Uvažujeme symetrický problém.

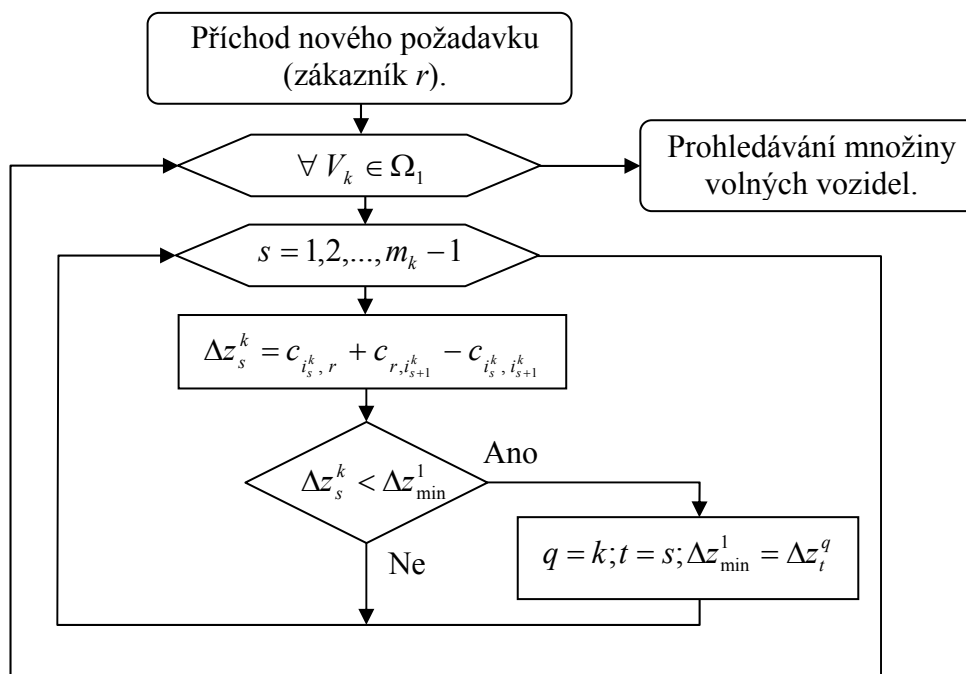
na jednu z následujících událostí: buď zavolá nový zákazník nebo řidič některého z vozidel oznámí, že dorazil zpět do svého výchozího místa¹⁶.



Obr. 4.9 – Vkládací algoritmus pro minimalizaci celkové vzdálenosti

¹⁶ Návrat vozidel dokáže samozřejmě odhadnout i samotný dispečer podle rozpisu návštěv jednotlivých zákazníků, který byl získán na základě uvedené optimalizace. Do jaké míry se tento rozpis liší od skutečné realizace jízd, není předmětem zkoumání v této práci.

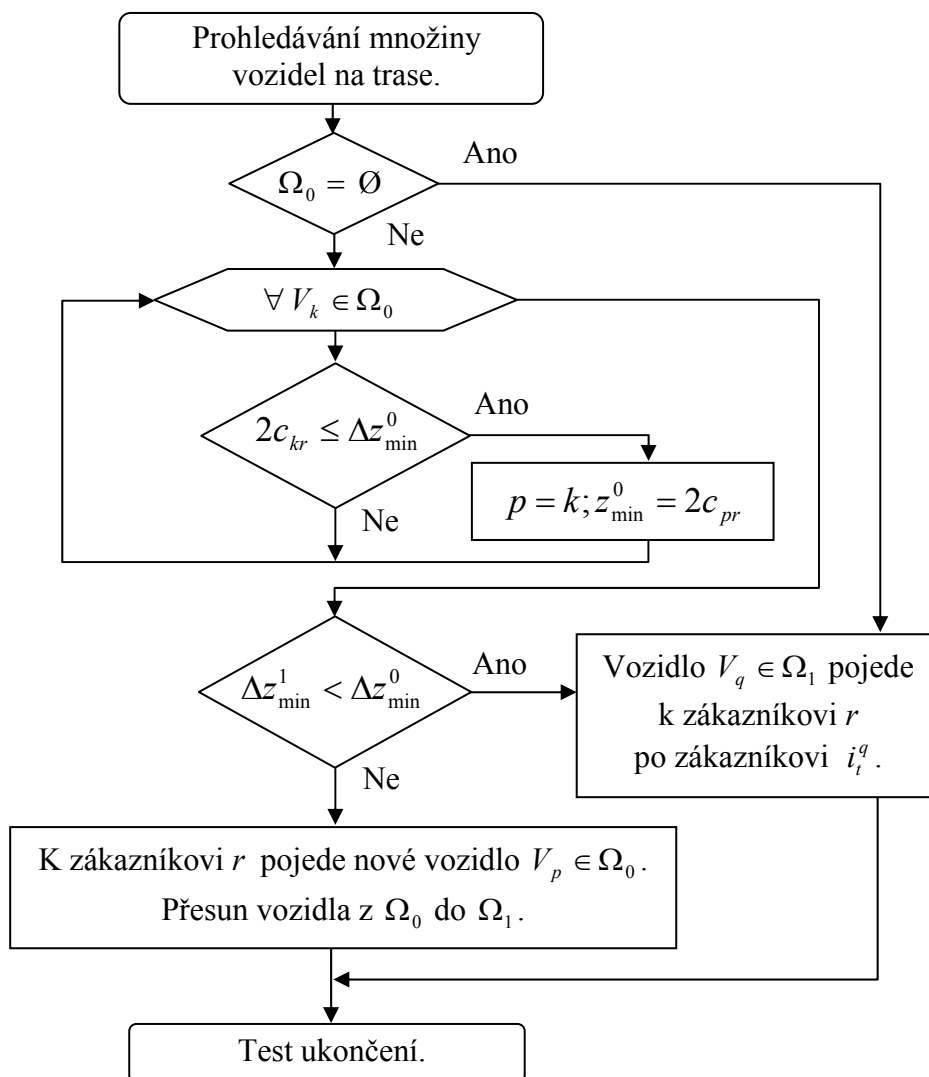
Pokud je následující událostí vznik nového požadavku, algoritmus nejprve prohledává množinu vozidel, která jsou již na trase, s cílem určit vozidlo, které je z hlediska minimálního prodloužení celkové délky všech tras nejvýhodnější (obr. 4.10). Zároveň je určen zákazník, po jehož obsluhu bude bezprostředně následovat obsluha nového zákazníka.



Obr. 4.10 – Prohledávání množiny vozidel na trase

Dále je nutné zjistit, zda by nebylo výhodnější obsloužit nového zákazníka některým z volných vozidel, která se právě nacházejí ve svých stanovištích (obr. 4.11). V případě, že není žádné vozidlo volné, obsluží nového zákazníka již předtím nalezené nejvýhodnější vozidlo na trase. Pokud existuje více volných vozidel, vybere se z nich takové vozidlo, které obsluhu nového zákazníka zajistí s minimálním prodloužením celkové délky všech tras. Následuje krok, ve kterém je nutné rozhodnout, zda je výhodnější, aby nového zákazníka obsloužilo nejvýhodnější volné vozidlo či nejvýhodnější vozidlo na trase. Jakmile dispečer rozhodne, které vozidlo zajistí obsluhu nově vzniklého požadavku, oznámí tuto změnu řidiči příslušného vozidla.

Celý proces končí ve chvíli, kdy je splněna určitá podmínka, kterou může být např. dosažení konce pracovní doby, resp. okamžiku, do kterého je možné přijímat další požadavky zákazníků; v rámci pracovní doby musí být ještě obslouženi zbývající zákazníci, na jejichž obsluhu se podílí všechna vozidla, která jsou na trase. Po obsloužení těchto zákazníků se všechna vozidla vrací do svých výchozích míst.

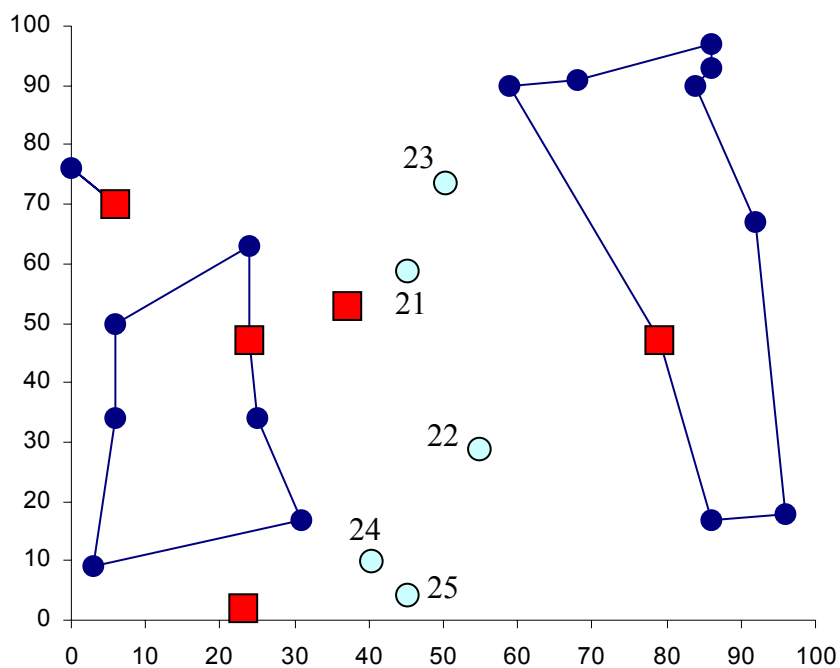


Obr. 4.11 – Prohledávání množiny volných vozidel a určení vozidla, které obslouží nového zákazníka

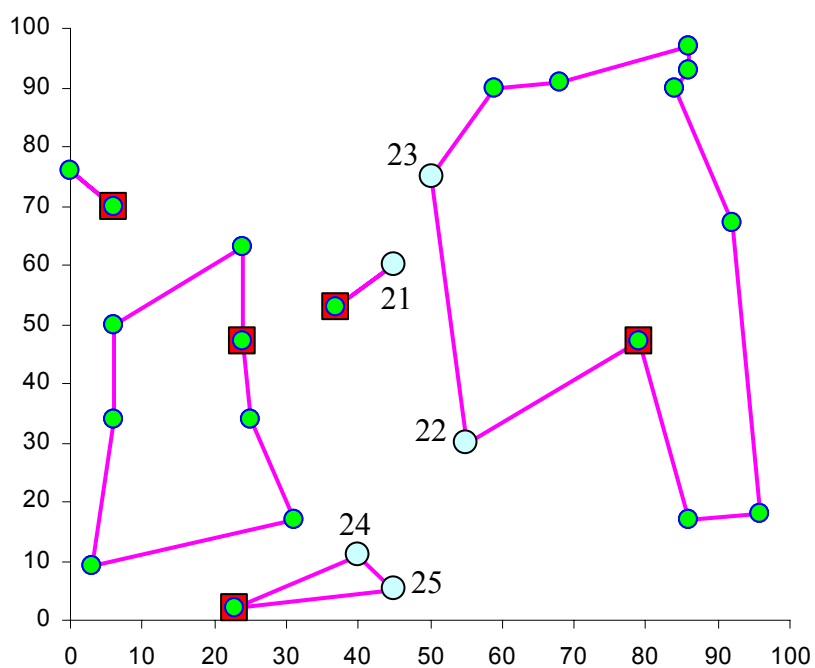
Příklad 4.7

Uvažujeme vygenerovanou úlohu s 5 výchozími místy a 15 zákazníky. V příloze č. 5 je uvedeno optimální řešení této úlohy, kdy se na obsluze zákazníků podílejí tři z pěti vozidel. Celková délka všech tří tras je přibližně 353,97 km.

V příloze č. 8 je vygenerováno 5 zákazníků (č. 21-25), jejichž požadavky přišly po výjezdu vozidel (obr. 4.12). Na obr. 4.13 jsou znázorněny všechny trasy, které byly navrženy použitím vkládacího algoritmu. Do obsluhy nových zákazníků jsou zapojena i původně volná vozidla. Celková vzdálenost, kterou ujedou všechna vozidla, je přibližně 469,24 km.



Obr. 4.12 – Požadavky nových zákazníků



Obr. 4.13 – Aplikace vkládacího algoritmu

Re-optimalizace

Po vzniku nového požadavku je možné místo aplikace vkládacího algoritmu provést re-optimalizaci naplánovaných tras. Pro minimalizaci celkové vzdálenosti ujeté všemi vozidly lze použít následující matematický model:

$$\text{minimalizovat } z = \sum_{k \in \omega} \sum_{i \in U_N} \sum_{j \in U_N} c_{ij} x_{ij}^k, \quad (4.44)$$

za podmínek

$$\sum_{k \in \omega} \sum_{i \in U_N} x_{ij}^k = 1, \quad j \in U_N, \quad j \notin \omega, \quad (4.45)$$

$$\sum_{k \in \omega} \sum_{i \in U_N} x_{ji}^k = 1, \quad j \in U_N, \quad j \notin \omega, \quad (4.46)$$

$$\sum_{j \in U_N} x_{kj}^k \leq 1, \quad k \in \omega, \quad (4.47)$$

$$\sum_{i \in U_N} x_{ij}^k = \sum_{i \in U_N} x_{ji}^k, \quad j \in U_N, \quad k \in \omega, \quad (4.48)$$

$$u_i - u_j + |U_N| \cdot x_{ij}^k \leq |U_N| - 1, \quad i, j \in U_N, \quad i, j \notin \omega, \quad i \neq j, \quad k \in \omega, \quad (4.49)$$

$$x_{k, next_k}^k = 1, \quad k \in \omega_1, \quad (4.50)$$

$$x_{ij}^k = 0, \quad i, j, k \in \omega, \quad (4.51)$$

$$x_{ii}^k = 0, \quad i \in U_N, \quad k \in \omega, \quad (4.52)$$

$$x_{ij}^k = 0, \quad i, k \in \omega, \quad i \neq k, \quad j \in U_N, \quad j \notin \omega, \quad (4.53)$$

$$x_{ij}^k \in \{0, 1\}, \quad i, j \in U_N, \quad k \in \omega. \quad (4.54)$$

Množinu U_N tvoří zákazníci, které musí vozidla ještě navštívit na svých plánovaných trasách. Kromě těchto zákazníků jsou v množině zahrnuta také všechna výchozí místa a nový zákazník. Hodnota $|U_N|$ udává počet prvků množiny U_N .

V množině ω jsou indexy všech vozidel, resp. výchozích míst, tj. $\omega = \{1, 2, \dots, K\}$. Množina $\omega_1 \subseteq \omega$ obsahuje indexy vozidel, která jsou v okamžiku vzniku nového požadavku na trase, přičemž místo, do kterého právě směřují, není výchozím místem¹⁷. Označme místo, do kterého směřuje vozidlo $k \in \omega_1$, indexem $next_k$. Uvedený matematický model vychází z předpokladu, že na základě právě ujetých tras některými, případně všemi vozidly, upravíme matici vzdáleností následujícím způsobem. Vzdálenosti $c_{k, next_k}$ mezi výchozím místem k a místem $next_k$ nahradíme délkou dosavadní trasy ujeté vozidlem k , tj. trasy, která začíná ve výchozím místě k a končí v místě $next_k$.

¹⁷ V opačném případě se totiž takové vozidlo stává pro další plánování tras volným vozidlem.

Účelová funkce (4.44) představuje celkovou délku všech tras. Podmínky (4.45), resp. (4.46) zajišťují, aby z každého místa vyjelo, resp. do každého místa přijelo právě jedno vozidlo. Nerovnosti (4.47) definují, že z každého výchozího místa může vyjet nejvýše jedno vozidlo. Soustava rovnic (4.48) zajišťuje, že každé místo bude navštíveno právě jedním vozidlem, tj. pokud do něj přijede nějaké vozidlo, toto vozidlo z něj také odjede. Omezení (4.49) a (4.52) představují smyčkové podmínky.

Pokud v okamžiku vzniku nového požadavku vozidla směřují do určitých míst, pak rovnicemi (4.50) budou odpovídající proměnné nastaveny na hodnotu 1. To ovšem neplatí pro situaci, kdy místem, do kterého vozidlo směřuje, je výchozí místo, tj. kdy se vozidlo vrací zpět do svého stanoviště. V takovém případě je na dané výchozí místo pro další analýzu nahlíženo podobně jako na výchozí místa, ve kterých vozidla čekají na případný výjezd. Podmínky (4.51) zakazují jednotlivým vozidlům vzájemně navštěvovat jednotlivá výchozí místa. Omezení (4.53) nepřipouštějí, aby vozidlo k vyjíždělo z jiného výchozího místa než k .

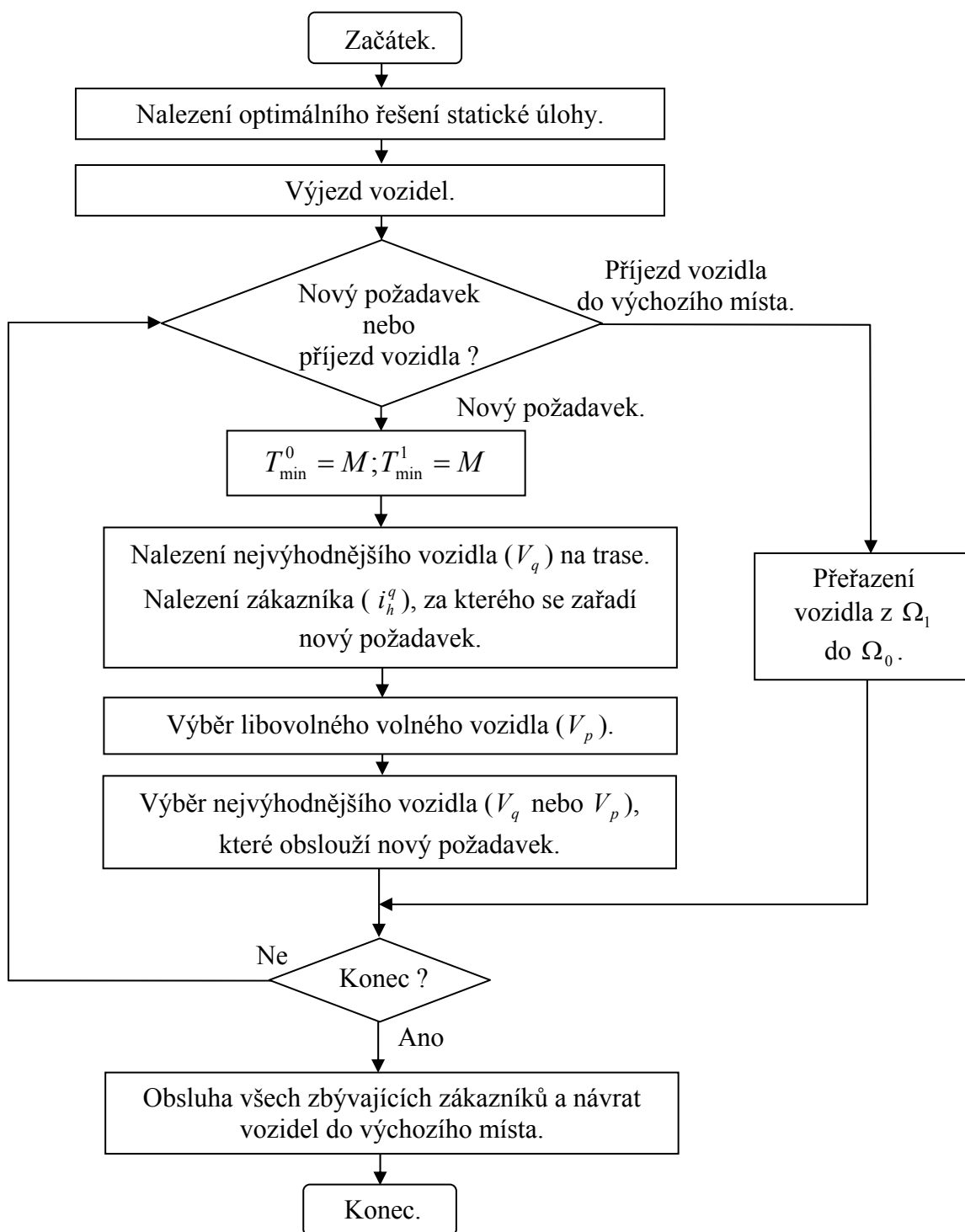
4.6.2 Minimalizace času potřebného k obslužení všech zákazníků – jedno výchozí místo

V případě jednoho výchozího místa, v němž je umístěno několik vozidel, je pro řešení statické úlohy s cílem obsloužit všechny známé zákazníky v minimálním čase použitý matematický model (4.20) – (4.28), případně model pro dodatečnou optimalizaci tras¹⁸. Do minimální doby potřebné k obslužení všech zákazníků je zahrnuta i doba nutná k návratu všech vozidel do výchozího místa. Po optimalizaci uvedené statické úlohy je množina vozidel rozdělena na množinu Ω_1 obsahující vozidla na trase a množinu Ω_0 obsahující vozidla, která z výchozího místa nevyjela.

Vkládací algoritmus

Podobně jako v dynamické úloze, jejímž cílem je minimalizovat celkovou ujetou vzdálenost, i v případě minimalizace času potřebného k obslužení všech zákazníků může být nový požadavek vložen do zbývajících částí trasy některého jedoucího vozidla nebo jeho obsluhu může zajistit volné vozidlo, čekající ve výchozím místě. Cílem pochopitelně není nalézt minimální prodloužení celkové ujeté vzdálenosti, ale zařadit nového zákazníka tak, aby jeho obsluha byla uskutečněna do naplánovaného termínu, případně aby byl tento termín posunut minimálně. Celý postup lze znázornit pomocí diagramu na obr. 4.14 – 4.16.

¹⁸ V tomto modelu je zahrnuto omezení (4.21a) a minimalizována účelová funkce (4.20a).



Obr. 4.14 – Vkládací algoritmus pro minimalizaci času potřebného k obslužení všech zákazníků

Opět předpokládáme, že obsluhující vozidlo $V_k \in \Omega_1$ má podle plánu navštívit posloupnost míst $U_k = \{i_1^k, i_2^k, \dots, i_{m_k}^k\}$, kde $i_{m_k}^k = 1$. Čas potřebný k absolvování této trasy lze vyjádřit výrazem

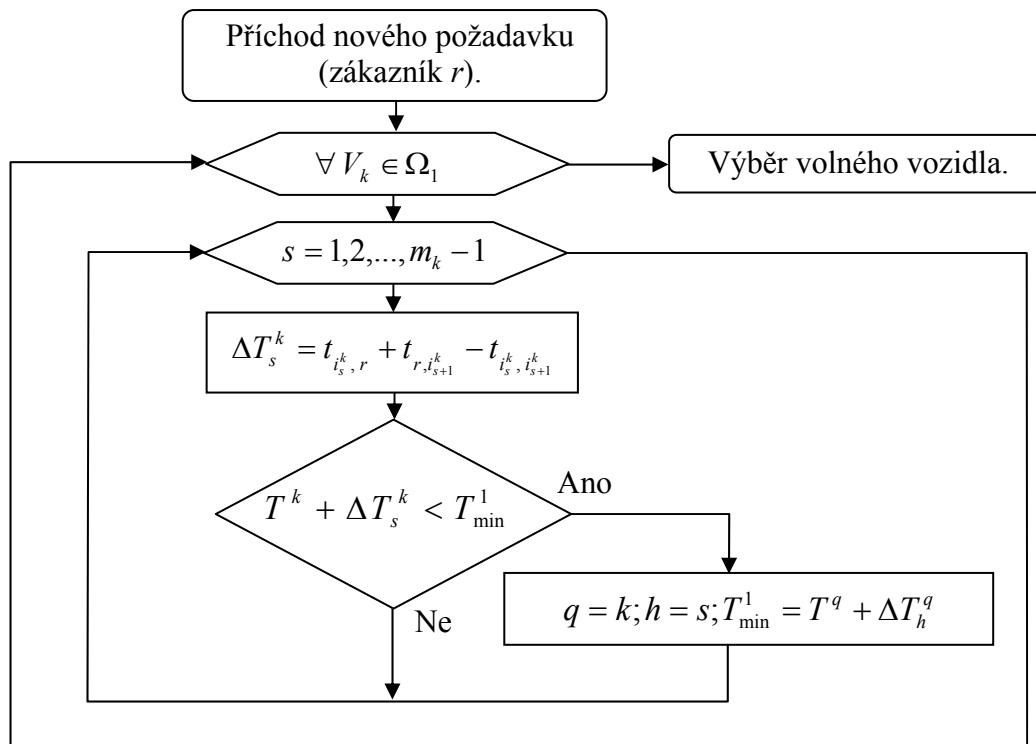
$$T^k = \sum_{s=1}^{m_k-1} t_{i_s^k, i_{s+1}^k}, \quad V_k \in \Omega_1. \quad (4.55)$$

Prodloužení stávající trasy vozidla V_k po vložení nového zákazníka $r \notin U_k$ ($\forall V_k \in \Omega_1$) mezi místa i_s^k a i_{s+1}^k lze vypočítat jako

$$\Delta T_s^k = t_{i_s^k, r} + t_{r, i_{s+1}^k} - t_{i_s^k, i_{s+1}^k}, \quad s = 1, 2, \dots, m_k - 1. \quad (4.56)$$

Cílem je nalézt zákazníka i_h^q navštíveného vozidlem $V_q \in \Omega_1$, pro kterého je délka prodloužené trasy minimální:

$$T^q + \Delta T_h^q = \min_{\substack{s=1,2,\dots,m_k-1 \\ \forall V_k \in \Omega_1}} (T^k + \Delta T_s^k). \quad (4.57)$$

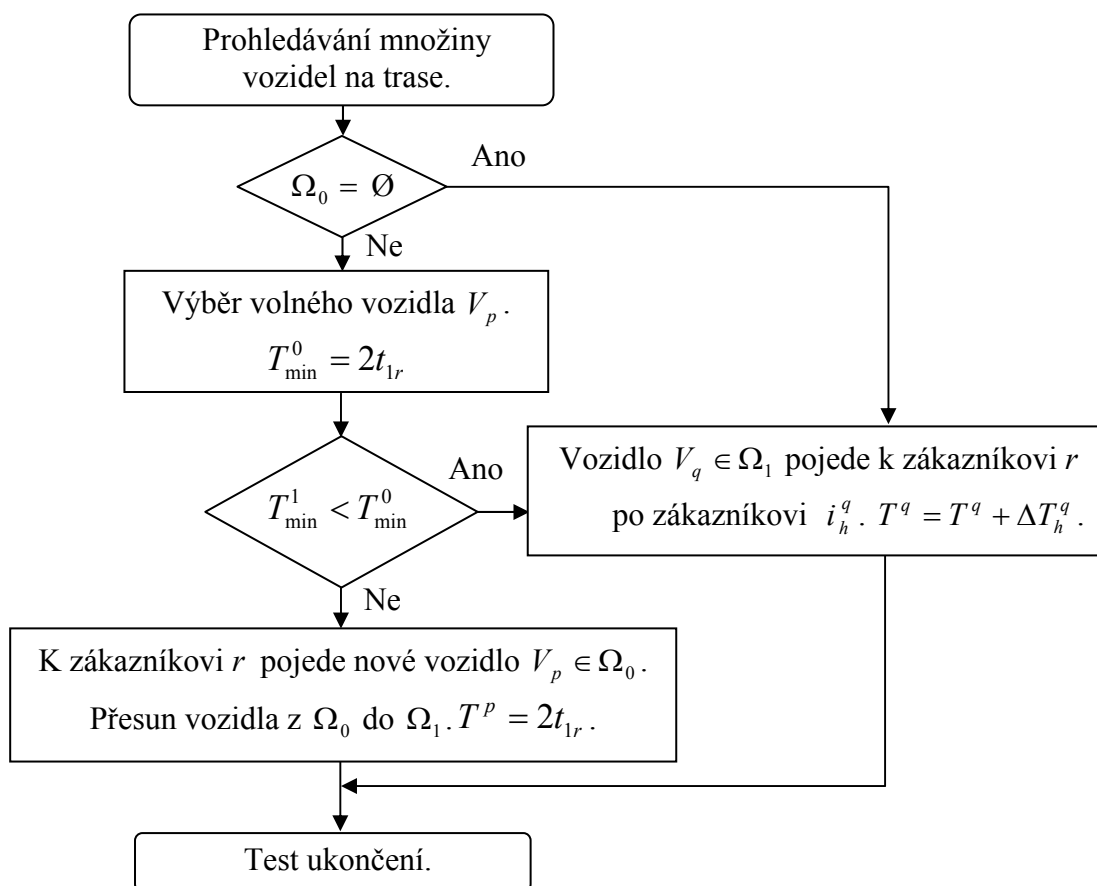


Obr. 4.15 – Prohledávání množiny vozidel na trase

Pro libovolné volné vozidlo (značené V_p), určíme prodloužení trasy¹⁹ v případě, že toto vozidlo bude zajišťovat obsluhu nového zákazníka:

$$T^p = 2t_{1r}. \quad (4.58)$$

Na základě porovnání hodnot (4.57) a (4.58) pak rozhodneme o tom, které vozidlo zajistí obsluhu nového zákazníka.



Obr. 4.16 – Výběr libovolného volného vozidla a určení vozidla, které obslouží nového zákazníka

V uvedeném postupu jsme se z důvodu zjednodušení úvah dopustili určité nepřesnosti, kterou lze však jednoduchým způsobem opravit. Přestože vozidlo nemůže v okamžiku vzniku nového požadavku měnit směr a musí dojet do místa, ke kterému směřuje, je z hlediska uvažované úlohy důležitý i čas, za který do tohoto místa dojede.

¹⁹ Uvažujeme symetrický problém.

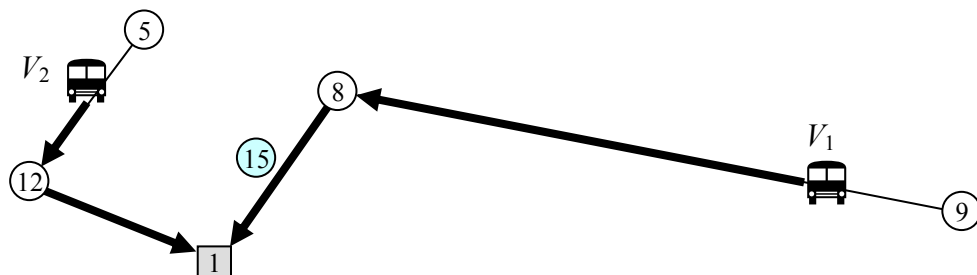
Příklad 4.8

Na obr. 4.17 jsou znázorněny trasy dvou vozidel V_1 a V_2 . V okamžiku vzniku nového požadavku č. 15 se vozidlo V_1 nachází mezi místy č. 9 a 8, vozidlo V_2 směřuje z místa č. 5 do místa č. 12. Pokud dispečer nezná okamžitou polohu vozidel, za výchozí místa nových tras vozidel bere místa č. 8 a 12. V tomto případě jsou při analýze použity vzorce (4.55) – (4.57), případně i vzorec (4.58), jestliže existuje ještě nějaké volné vozidlo. Dispečer pak zřejmě doporučí vozidlu V_1 , aby obsloužilo zákazníka č. 15 bezprostředně po obsloužení zákazníka č. 8.

Pokud ovšem dispečer zná okamžitou polohu vozidel, pak (4.55) lze upravit následovně:

$$T^k = t_{0i_1^k} + \sum_{s=1}^{m_k-1} t_{i_s^k, i_{s+1}^k}, \quad V_k \in \Omega_1, \quad (4.59)$$

kde $t_{0i_1^k}$ je doba, za kterou vozidlo V_k dorazí do místa i_1^k . V tomto případě by byl zákazník č. 15 obsloužen vozidlem V_2 , případně dalším volným vozidlem.



Obr. 4.17 – Známa okamžitá poloha vozidel na trase

Na závěr uvedeme možnost zlepšení získaného řešení z hlediska celkové ujeté vzdálenosti. Často lze na základě uvedeného postupu dospět k řešení, které po zařazení nového zákazníka nezmění původní hodnotu času potřebného k obsluze všech zákazníků. Tato situace nastane v případě, kdy nového zákazníka obslouží jiné vozidlo než to, které má z hlediska času nejdelší trasu. Přitom musí být splněna další podmínka – po zařazení nového zákazníka do trasy jiného vozidla nesmí dojít k překročení hodnoty nejdelší trasy. Matematicky lze jednoduše zapsat tyto dvě podmínky takto:

$$\min (T^q + \Delta T_h^q, T^p) < \max_{V_k \in \Omega_1} T^k, \quad (4.60)$$

kde hodnoty T^k jsou vypočteny podle (4.55) nebo (4.59). Podmínka (4.60) je podmínkou nutnou, nikoli však postačující pro možnost zlepšení řešení z hlediska celkové ujeté vzdálenosti. Další nutnou podmínkou je existence alespoň dvou vozidel²⁰, u nichž nedojde, po zařazení nového zákazníka do jejich trasy, k překročení²¹ výrazu na pravé straně (4.60).

Konečně postačující podmínkou pro zlepšení řešení je existence takového vozidla V_l , pro které platí následující nerovnost²²:

$$\Delta T_j^l < \Delta T_h^q, \quad (4.61)$$

kde ΔT_h^q je minimální prodloužení trasy, tj. minimum hodnot vypočtených podle (4.56) pro vozidlo V_q , které bylo vybráno na základě algoritmu na obr. 4.14 – 4.16, a ΔT_j^l je minimální prodloužení trasy pro vozidlo V_l . Pokud existuje více vozidel splňujících všechny uvedené nutné podmínky a podmínku (4.61), pak bude zřejmě vybráno vozidlo s minimálním prodloužením trasy.

Předmětem dalšího zkoumání může být otázka, zda opakovaným²³ zlepšováním řešení s cílem zkrátit ujetou vzdálenost nedojde k dlouhodobému zhoršení řešení z hlediska původního cíle, tj. minimalizace času potřebného k obslužení všech zákazníků. Nabízí se možnost porovnání výsledků simulačních experimentů na stejných datových souborech s použitím a bez použití uvedeného postupu.

Re-optimalizace

Pro optimální zařazení nového požadavku do zbývajících částí tras lze použít následující matematický model:

$$\text{minimalizovat } z = T, \quad (4.62)$$

za podmínek

$$\sum_{i \in U_N} \sum_{j \in U_N} t_{ij} x_{ij}^k \leq T, \quad k \in \omega, \quad (4.63)$$

$$\sum_{k \in \omega} \sum_{i \in U_N} x_{ij}^k = 1, \quad j \in U_N - \{1\}, \quad (4.64)$$

²⁰ Existuje-li pouze jediné takové vozidlo, je výběr jednoznačný.

²¹ U obou vozidel musí být podmínka splněna jako ostrá nerovnost.

²² Předpokládáme, že obě vozidla jsou na trase. Podobné nerovnosti lze odvodit i pro volná vozidla, případně dvojici vozidel, z níž jedno je na trase a druhé volné.

²³ Postup je aplikován opakovaně i pro další nové zákazníky.

$$\sum_{k \in \omega} \sum_{i \in U_N} x_{ji}^k = 1, \quad j \in U_N - \{1\}, \quad (4.65)$$

$$\sum_{\substack{j \in U_N \\ j \neq 1}} x_{1j}^k \leq 1, \quad k \in \omega, \quad (4.66)$$

$$\sum_{i \in U_N} x_{ij}^k = \sum_{i \in U_N} x_{ji}^k, \quad j \in U_N - \{1\}, \quad k \in \omega, \quad (4.67)$$

$$u_i - u_j + |U_N| \cdot x_{ij}^k \leq |U_N| - 1, \quad i \in U_N, \quad j \in U_N - \{1\}, \quad i \neq j, \quad k \in \omega, \quad (4.68)$$

$$x_{1next_k}^k = 1, \quad k \in \omega_1, \quad (4.69)$$

$$x_{ii}^k = 0, \quad i \in U_N, \quad k \in \omega, \quad (4.70)$$

$$x_{ij}^k \in \{0,1\}, \quad i, j \in U_N, \quad k \in \omega. \quad (4.71)$$

Podobně jako v modelu (4.44) – (4.54) jsou v množině U_N místa, která musí vozidla ještě navštívit na svých plánovaných trasách, včetně nového zákazníka. Navíc je v této množině zahrnuto i výchozí místo č. 1. Hodnota $|U_N|$ představuje počet prvků množiny U_N . Množina ω obsahuje indexy všech vozidel, tj. $\omega = \{1, 2, \dots, K\}$, v množině $\omega_1 \subseteq \omega$ jsou indexy vozidel, která jsou v okamžiku vzniku nového požadavku na trase, index $next_k$ označuje místo²⁴, do něhož v tomto okamžiku směřuje vozidlo $k \in \omega_1$.

V uvedeném matematickém modelu je pro všechna vozidla na trase původní časová vzdálenost t_{1next_k} změněna na dobu, za kterou vozidlo k dorazí do místa $next_k$ z místa, ve kterém se nachází v okamžiku vzniku nového požadavku. Pokud dispečer není schopen určit či alespoň odhadnout okamžitou polohu vozidel, pak změní tuto hodnotu na 0, čímž se ovšem dopouští výrazného zkreslení²⁵, a nalezené optimální řešení modelu nemusí být optimálním řešením původního problému.

Hodnota proměnné T v (4.62) a (4.63) představuje v optimálním řešení délku nejdelší trasy (z hlediska času). Rovnice (4.64) a (4.65) zajišťují, že každé místo bude navštíveno právě jednou. Podmínky (4.67) zaručují, že návštěva každého místa se uskuteční právě jedním vozidlem. Nerovnice (4.66) určují, že každé vozidlo odjede z výchozího místa nejvýše jednou, tzn. některá vozidla nemusí vyjet. Podmínky (4.68) a (4.70) jsou smyčkovými

²⁴ Jestliže vozidlo směřuje do výchozího místa, je v modelu považováno za volné vozidlo.

²⁵ Pokud dispečer nezná okamžitou polohu vozidel, dopouští se podobného zkreslení samozřejmě i v případě použití vkládacího algoritmu.

podmínkami. Rovnice (4.69) nedovolují vozidlům změnit směr jízdy v okamžiku vzniku nového požadavku.

Po nalezení optimálního řešení matematického modelu (4.62) – (4.71) lze dodatečně změnit navržené trasy tak, aby řešení bylo optimální i z hlediska celkové ujeté vzdálenosti. K tomuto účelu lze využít postupu doporučeného v části 4.4.1, přičemž je zapotřebí vhodně upravit účelovou funkci (4.20a) a soustavu omezujících podmínek (4.21a).

4.6.3 Minimalizace času potřebného k obslužení všech zákazníků – více výchozích míst

Řešením matematického modelu (4.29) – (4.39) je získán optimální rozvrh tras vozidel, která jsou umístěna v několika výchozích místech (v každém je umístěno právě jedno vozidlo). Pro zařazení nově vzniklých požadavků do tras jednotlivých vozidel budou upraveny oba algoritmy popsané v předchozí části.

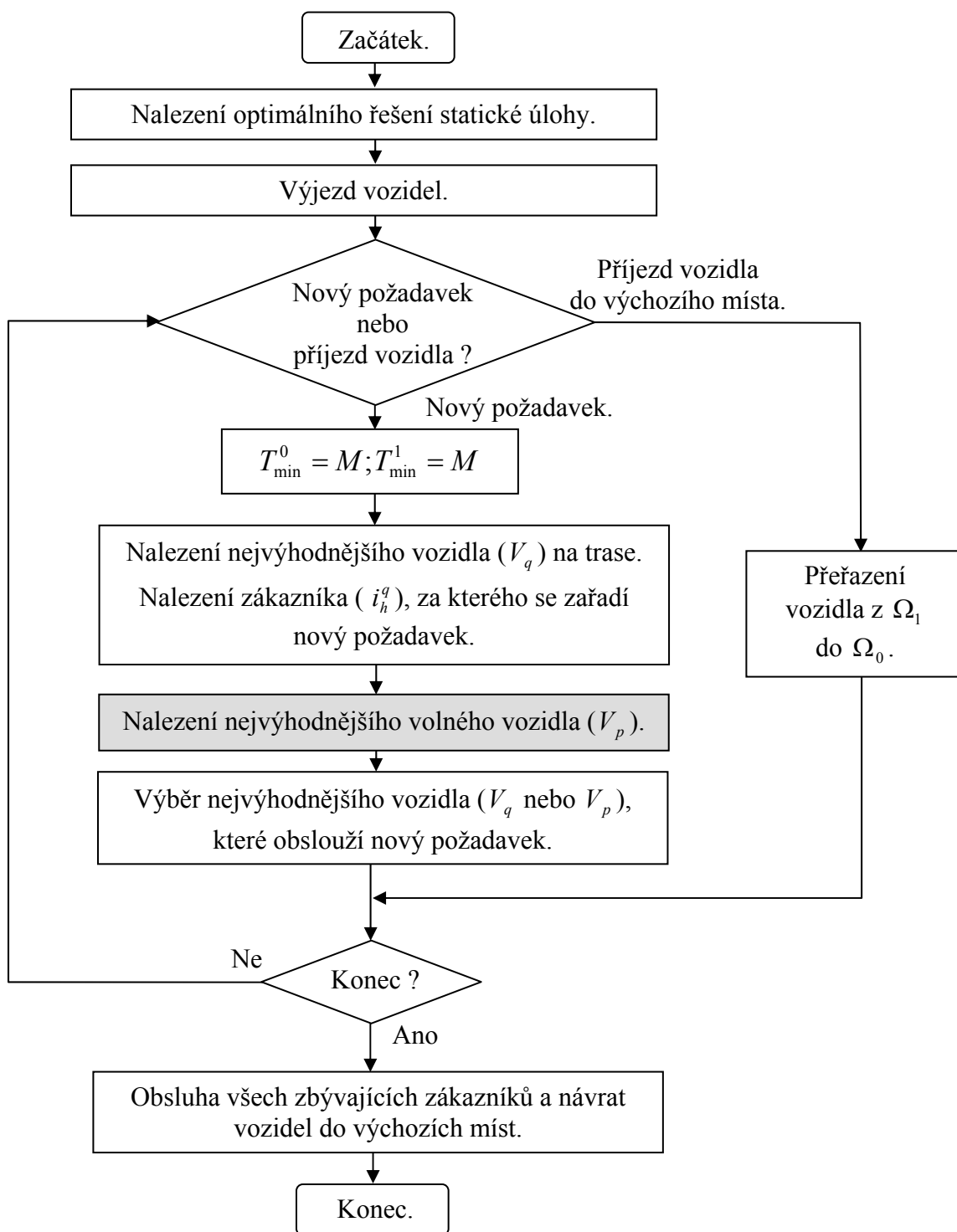
Vkládací algoritmus

Úloha s více výchozími místy se jen nepatrně liší od úlohy s vozidly umístěnými v jednom výchozím místě. Bude proto účelné pouze upozornit na změny, které je zapotřebí v postupu provést²⁶. Základní schéma algoritmu je znázorněno na obr. 4.18.

Obsluhující vozidlo $V_k \in \Omega_1$ má podle plánu navštívit posloupnost míst $U_k = \{i_1^k, i_2^k, \dots, i_{m_k}^k\}$, kde $i_{m_k}^k = k$, tj. vozidlo se musí vrátit do svého výchozího místa. Předpokládejme, že dispečer dokáže určit či odhadnout okamžitou polohu každého vozidla, resp. dobu, za kterou vozidlo dorazí do místa, do kterého směřuje v okamžiku vzniku nového požadavku. Doba, za kterou se vozidlo vrátí do svého výchozího místa, je vypočtena podle (4.59).

Pro výběr nejvhodnějšího vozidla na trase (obr. 4.15), které obslouží nově vzniklý požadavek, lze použít vzorce (4.56) a (4.57). Výsledkem je nalezení takového vozidla $V_q \in \Omega_1$ a takového místa i_h^q , po němž vozidlo navštíví nového zákazníka, aby čas nutný pro návštěvu všech doposud nenavštívených míst byl minimální.

²⁶ Ve schématech znázorňujících algoritmus jsou tyto změny zvýrazněny stínováním.

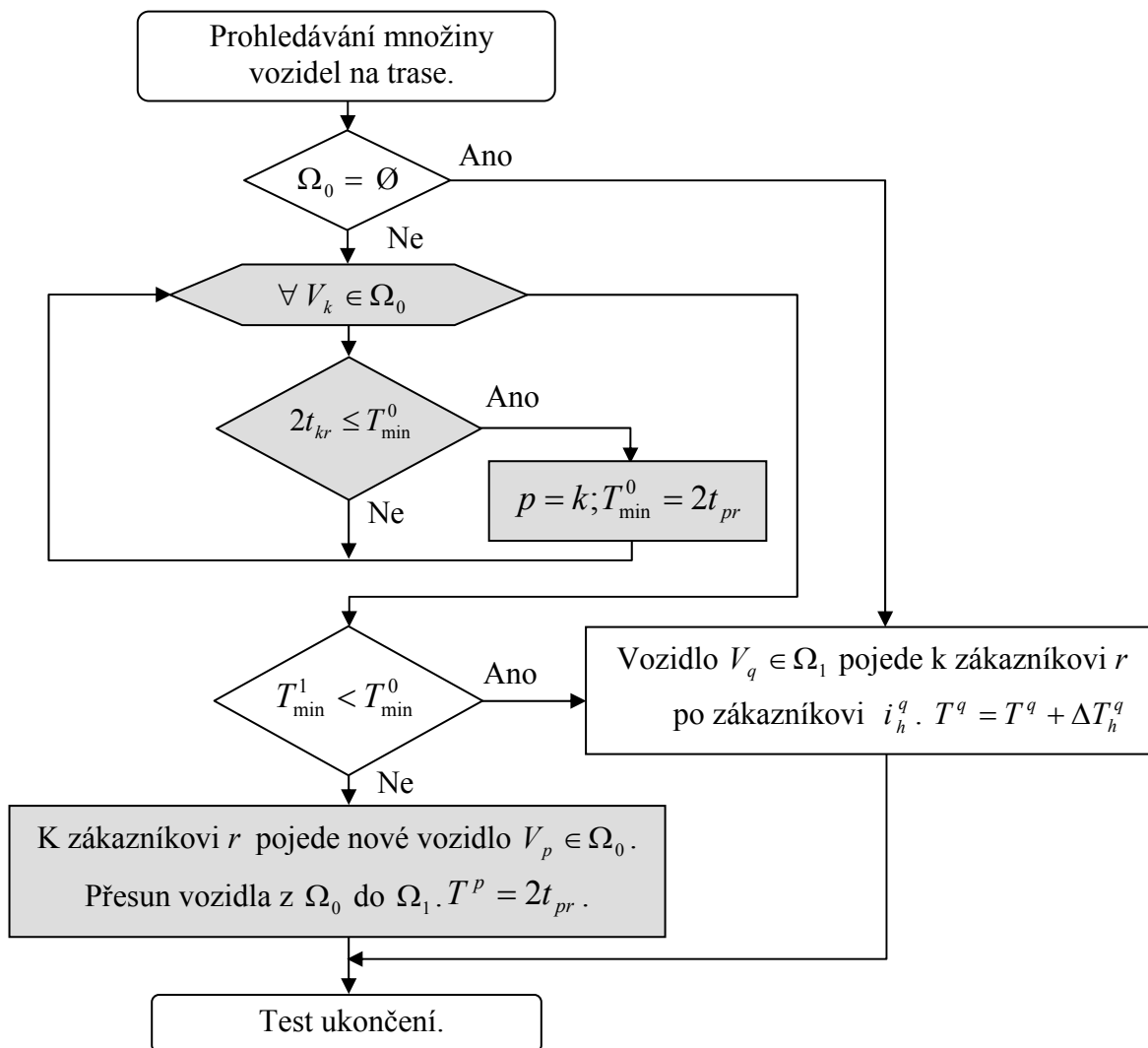


Obr. 4.18 – Vkládací algoritmus pro minimalizaci času potřebného k obslužení všech zákazníků vozidly v několika výchozích místech

Protože volná vozidla se nacházejí v různých výchozích místech, určíme pro každé vozidlo, které není na trase, prodloužení trasy²⁷ v případě, že toto vozidlo bude zajišťovat obsluhu nového zákazníka. Z těchto hodnot najdeme minimum:

$$T^p = \min_{\forall V_k \in \Omega_0} 2t_{kr}. \quad (4.72)$$

Příslušným způsobem je pak upraveno schéma algoritmu (obr. 4.19).



Obr. 4.19 – Prohledávání množiny volných vozidel a určení vozidla, které obslouží nového zákazníka

²⁷ Uvažujeme symetrický problém.

Re-optimalizace

Matematický model pro optimální zařazení nově vzniklého požadavku je založený na modelech (4.44) – (4.54) a (4.62) – (4.71). V množině U_N jsou kromě dosud nenavštívených zákazníků (včetně nového zákazníka) také všechna výchozí místa. Množina ω obsahuje indexy všech vozidel, resp. výchozích míst, tj. $\omega = \{1, 2, \dots, K\}$, množina $\omega_1 \subseteq \omega$ obsahuje indexy vozidel na trase. V matematickém modelu je pro všechna vozidla na trase původní časová vzdálenost $t_{k, next_k}$ změněna na dobu, za kterou vozidlo k dorazí do místa $next_k$ z místa, ve kterém se nachází v okamžiku vzniku nového požadavku.

$$\text{minimalizovat } z = T, \quad (4.73)$$

za podmínek

$$\sum_{i \in U_N} \sum_{j \in U_N} t_{ij} x_{ij}^k \leq T, \quad k \in \omega, \quad (4.74)$$

$$\sum_{k \in \omega} \sum_{i \in U_N} x_{ij}^k = 1, \quad j \in U_N, \quad j \notin \omega, \quad (4.75)$$

$$\sum_{k \in \omega} \sum_{i \in U_N} x_{ji}^k = 1, \quad j \in U_N, \quad j \notin \omega, \quad (4.76)$$

$$\sum_{j \in U_N} x_{kj}^k \leq 1, \quad k \in \omega, \quad (4.77)$$

$$\sum_{i \in U_N} x_{ij}^k = \sum_{i \in U_N} x_{ji}^k, \quad j \in U_N, \quad k \in \omega, \quad (4.78)$$

$$u_i - u_j + |U_N| \cdot x_{ij}^k \leq |U_N| - 1, \quad i, j \in U_N, \quad i, j \notin \omega, \quad i \neq j, \quad k \in \omega, \quad (4.79)$$

$$x_{k, next_k}^k = 1, \quad k \in \omega_1, \quad (4.80)$$

$$x_{ij}^k = 0, \quad i, j, k \in \omega, \quad (4.81)$$

$$x_{ii}^k = 0, \quad i \in U_N, \quad k \in \omega, \quad (4.82)$$

$$x_{ij}^k = 0, \quad i, k \in \omega, \quad i \neq k, \quad j \in U_N, \quad j \notin \omega, \quad (4.83)$$

$$x_{ij}^k \in \{0, 1\}, \quad i, j \in U_N, \quad k \in \omega. \quad (4.84)$$

Význam všech částí uvedeného matematického modelu je stejný jako u re-optimalizačních modelů popsaných v částech 4.6.1 a 4.6.2.

5. Dynamické rozvozní úlohy

V předchozích kapitolách byly definovány okružní úlohy, u nichž není limitujícím faktorem kapacita vozidel. Jedná se o problémy, které mají základ v úloze obchodního cestujícího. Typické reálné aplikace lze najít u firem zajišťujících opravárenské služby. V takových úlohách vozidlo většinou nepřeváží žádné zboží (kromě náhradních dílů potřebných při opravě) a tudíž kapacita vozidla je nevýznamná.

Existuje ovšem celá řada aplikací, ve kterých zákazník požaduje, aby mu bylo z nějakého centrálního distribučního místa dodáno určité množství zboží. Takové úlohy se nazývají **rozvozní úlohy (VRP)**¹. Do této třídy úloh patří i problémy, ve kterých naopak zákazník požaduje odvoz zboží, materiálu, odpadu apod. Desrosiers et al. (1995) uvádějí třídu úloh VRP jako speciální podskupinu úloh třídy **pick-up and delivery problems**. Další rozdělení těchto aplikací na úlohy svozu a úlohy rozvozu je důležité především z hlediska samotné ekonomické definice úlohy, nikoli však z hlediska jejího řešení.

Pro nalezení optimálního řešení rozvozních úloh lze použít metodu větvení a hranic², metodu větvení a řezů³, případně metodu větvení a oceňování⁴. Tyto metody popisují Toth and Vigo (1998), Achuthan et al. (2003), Desrochers et al. (1992), Kolen et al. (1987) aj. Přibližné řešení poskytují heuristické či metaheuristické postupy, založené většinou na metodách tabu search, genetických algoritmech apod. Jejich použitím se zabývají Čičková (2005), Potvin and Bengio (1996), Potvin et al. (1996), Bräysy and Gendreau (2005a, 2005b), Gendreau et al. (1994), Russel (1995), Taillard et al. (1997), Golden et al. (1998) aj. Většina uvedených prací se týká rozvozních úloh s časovými okny (**VRPTW**)⁵.

Tato práce je zaměřena na dynamické problémy, v nichž požadavek některých zákazníků vzniká až v okamžiku, kdy je vozidlo na trase. Touto problematikou se zabývají Gendreau et al. (1999), Gendreau and Potvin (1998), Savelsbergh and Sol (1998), Powell et al. (1995), Ichoua et al. (2000), Chen and Xu (2006) aj.

¹ Vehicle Routing Problem.

² Branch-and-Bound Algorithm.

³ Branch-and-Cut Algorithm.

⁴ Branch-and-Price Algorithm. Tento postup je založen na metodě generování sloupců, jejíž použití v celočíselném programování uvádějí Fábry (2005) a Fábry and Pelikán (2004).

⁵ Vehicle Routing Problem with Time Windows.

Při rozvozu zboží⁶ musí mít vozidlo v okamžiku výjezdu na trasu dostatečnou zásobu pro uspokojení nově vzniklých požadavků. V opačném případě se bude vozidlo neustále vracet do svého výchozího místa, čímž samozřejmě neúměrně vzrostou přepravní náklady, nehledě na časovou ztrátu. Naopak při svozu zboží bude jistě výhodné neplnit vozidlo na trase zcela, ale ponechat určité místo pro zboží, které bude neplánovaně naloženo u nových zákazníků. Tato strategie hraje významnou roli především v případech, kdy je možné odhadnout výši nově vzniklých požadavků, tj. distribuční firma zná pravděpodobnostní rozdělení velikosti poptávky, resp. nabídky potenciálních zákazníků.

Pro zjednodušení budeme ve všech modelech předpokládat, že cílem je svézt zboží od zákazníků do jednoho, případně několika stanovišť.

5.1 Rozvozní úloha s jedním vozidlem

5.1.1 Statická úloha

V této úloze existuje jediné výchozí místo označené č. 1, v němž je umístěno jediné vozidlo. Toto vozidlo o známé kapacitě V má zajistit svoz materiálu zboží či materiálu od $n-1$ zákazníků, u nichž je známá velikost požadavku $0 < q_i \leq V$ ($i = 2, 3, \dots, n$). Je definována matice nejkratších vzdáleností mezi všemi zákazníky a nejkratších vzdáleností mezi výchozím místem a všemi zákazníky. Vzdálenost mezi místy i a j je označena c_{ij} ($i, j = 1, 2, \dots, n$).

Matematický model úlohy, v níž je cílem minimalizovat celkovou vzdálenost ujetou vozidlem, splnit požadavky zákazníků a nepřekročit kapacitu vozidla, lze pak zapsat následujícím způsobem (Pelikán, 2001):

$$\text{minimalizovat } z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}, \quad (5.1)$$

za podmínek

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 2, 3, \dots, n, \quad (5.2)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 2, 3, \dots, n, \quad (5.3)$$

$$u_i + q_j - V(1 - x_{ij}) \leq u_j, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j, \quad (5.4)$$

⁶ Uvažujeme homogenní zboží.

$$q_i \leq u_i \leq V, \quad i = 2, 3, \dots, n, \quad (5.5)$$

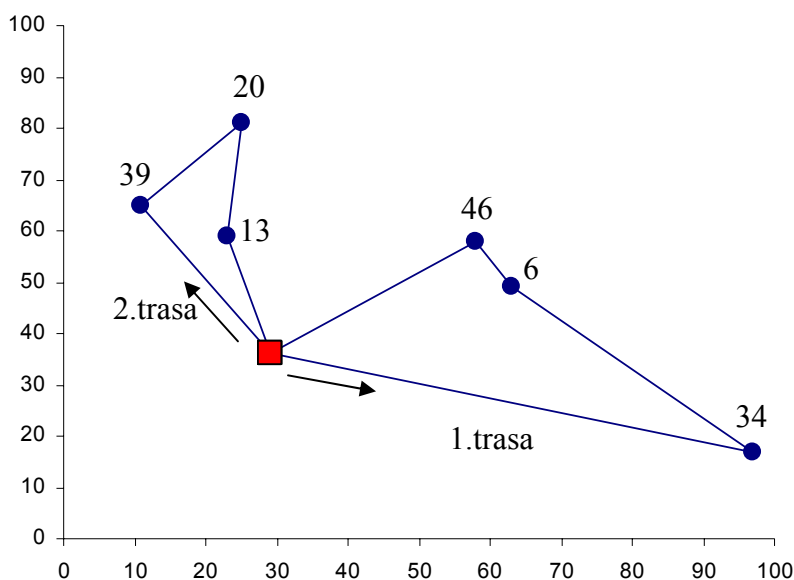
$$u_1 = 0, \quad (5.6)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n. \quad (5.7)$$

Proměnná x_{ij} je bivalentní proměnná nabývající hodnoty 1 v případě, že vozidlo jede do místa j z místa i , hodnoty 0 v opačném případě. Účelová funkce (5.1) představuje celkovou vzdálenost ujetou vozidlem. Omezující podmínky (5.2) a (5.3) stanovují, že každé místo bude navštíveno právě jednou. Nerovnosti (5.4) jsou smyčkovými podmínkami zamezujícími vytváření parciálních cyklů, tj. cyklů neobsahujících výchozí místo. Zároveň představují bilanci nákladu vozidla. Podmínky (5.5) nepovolují překročení kapacity vozidla V po návštěvě libovolného zákazníka, navíc bilancují požadavek zákazníka a náklad vozidla. Jestliže požadavek některého ze zákazníků převyšuje kapacitu vozidla, je nutné úlohu řešit jako rozvozní problém s dělenou dodávkou, resp. nakládkou (tato úloha bude popsána v další části disertační práce). Rovnice (5.6) definuje nulový náklad vozidla ve výchozím místě.

Příklad 5.1

V příloze č. 9 je uveden příklad se 6 zákazníky, kteří požadují odvézt určitý počet jednotek do výchozího místa. K dispozici je jediné vozidlo o kapacitě 100 jednotek.



Obr. 5.1 – Svážení jednotek od 6 zákazníků

Na obr. 5.1 je znázorněno optimální řešení, ve kterém vozidlo realizuje svoz ve dvou trasách. Čísla u jednotlivých zákazníků představují požadavky zákazníků, tj. počty jednotek, které

chtějí odvézt. Délka první trasy je přibližně 163,99 km, délka druhé trasy přibližně 101,25 km. Na první trase vozidlo sveze 86 jednotek, na druhé 72 jednotek.

5.1.2 Dynamická úloha

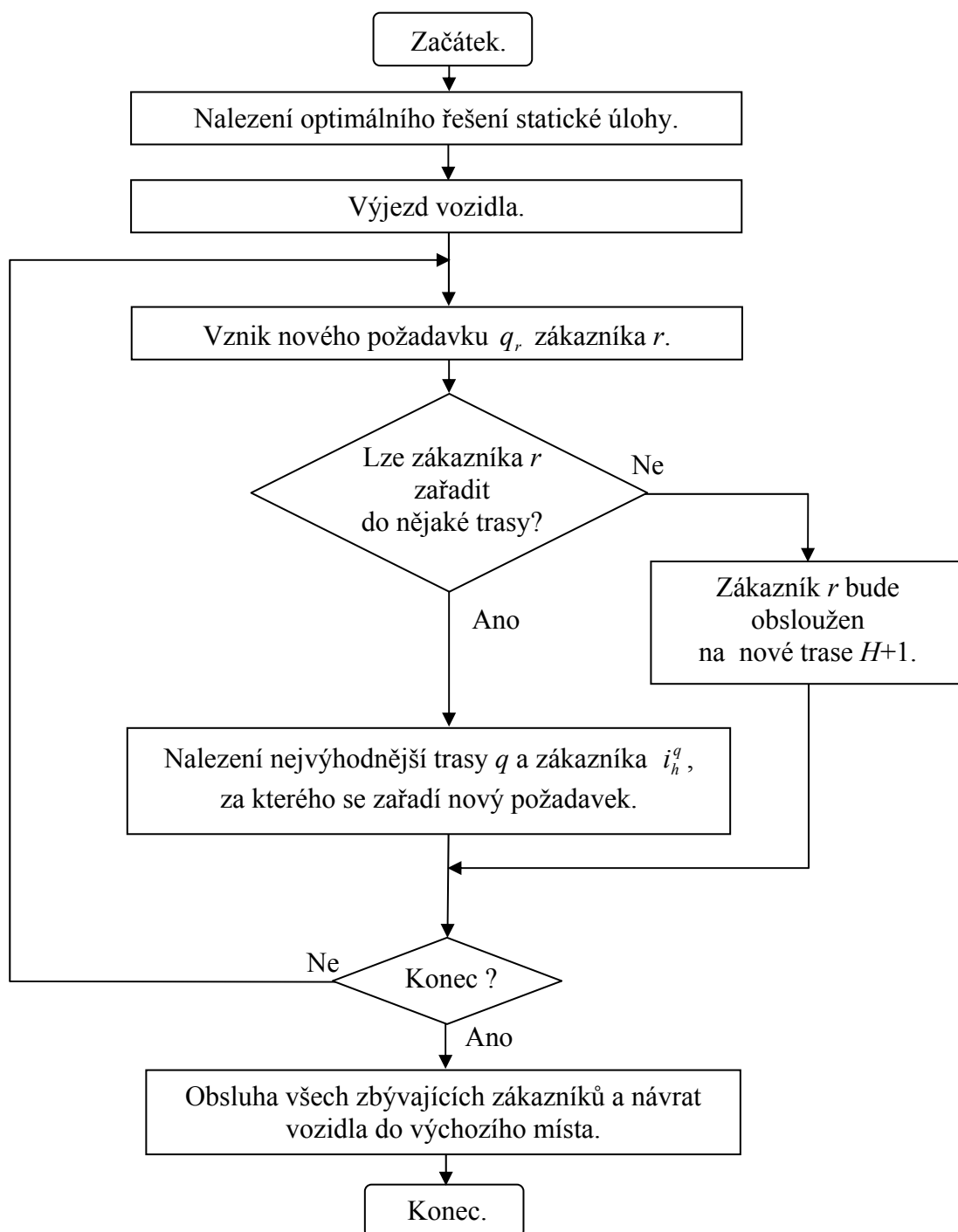
Podobně jako v úloze obchodního cestujícího předpokládáme, že během jízdy vozidla může vzniknout nový požadavek. Zařazení nového zákazníka lze provést na základě heuristických algoritmů nebo re-optimalizací zbývajících částí trasy, resp. tras.

Vkládací algoritmus

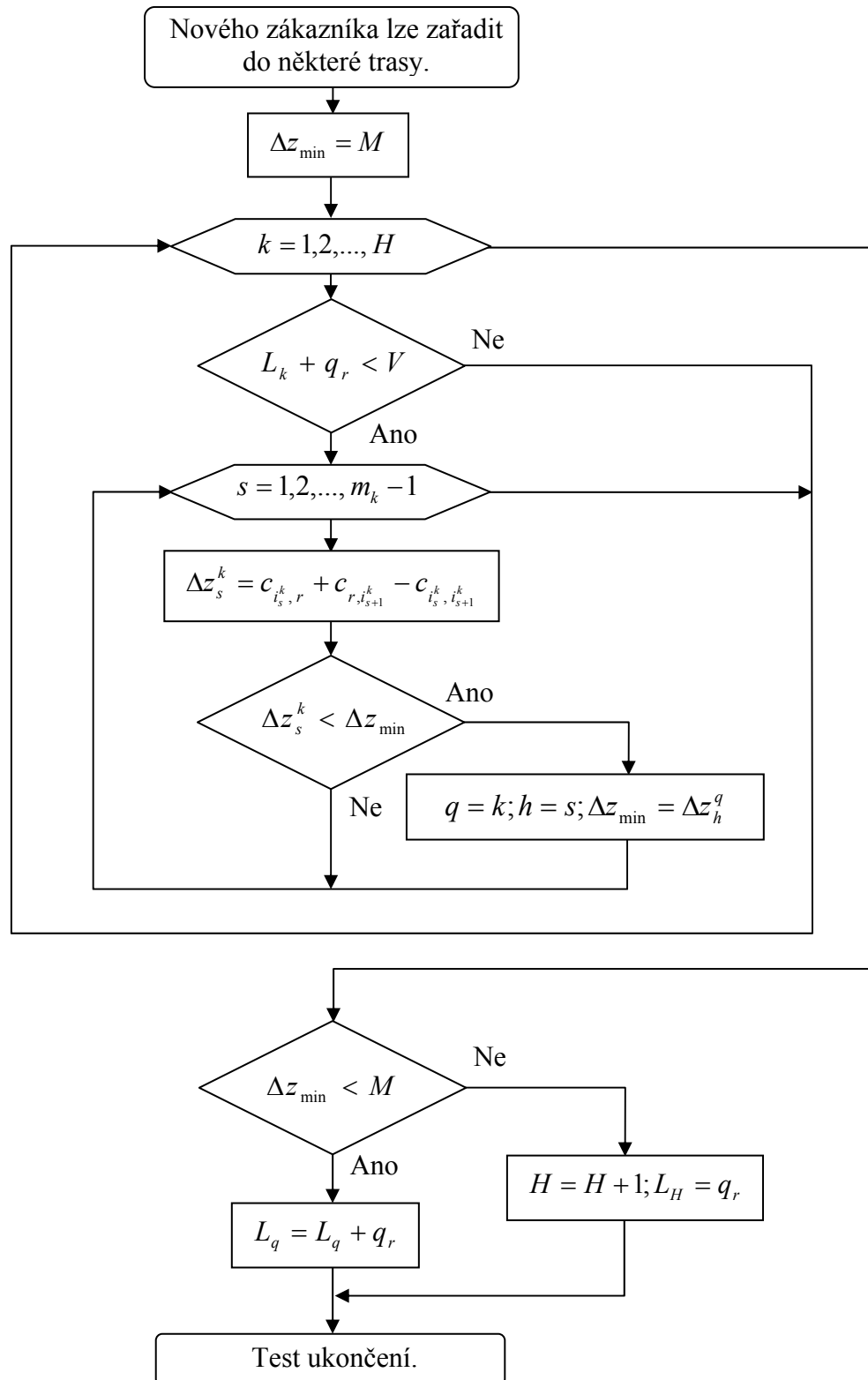
Hrubé schéma algoritmu je znázorněno na obr. 5.2. V okamžiku vzniku nového požadavku se vozidlo nachází na jedné z tras, které musí ještě absolvovat podle optimálního rozvrhu, resp. již upraveného rozvrhu. Jejich počet označme H , číslem 1 označme nedokončenou trasu, na které se vozidlo právě nachází. Posloupnost $U_k = \{i_1^k, i_2^k, \dots, i_{m_k}^k\}$ je tvořena místy, která má vozidlo navštívit na k -té trase ($k = 1, 2, \dots, H$). Zřejmě platí $i_{m_1}^1 = 1$ a $i_1^k = i_{m_k}^k = 1$ pro $k > 1$. Dále označme $L_k \leq V$ předpokládaný náklad, které bude mít na k -té trase vozidlo při návratu do výchozího místa.

Po vzniku nového požadavku je nutné otestovat, zda je možné pro jeho splnění využít vozidla v rámci některé z navržených tras. Naložení nákladu r -tého zákazníka q_r nesmí mít v žádném případě za následek překročení kapacity vozidla V na trase. Cílem je nalézt nejvýhodnější trasu a zákazníka, za kterého bude vložen nový zákazník tak, aby prodloužení trasy bylo minimální. Pokud by kapacita vozidla byla překročena u všech tras, pak musí být nový zákazník obsloužen na nové trase.

Výběr trasy, resp. místa na trase, za které bude vložen nový zákazník, je podrobně znázorněn na obr. 5.3. Parametr M v uvedeném schématu představuje vysokou konstantu. Celý proces končí v okamžiku, do kterého je možné přijímat další požadavky zákazníků, aby nedošlo k překročení pracovní doby distribuční firmy. Vozidlo dokončí obsluhu všech zbývajících zákazníků a vrací se do výchozího místa.



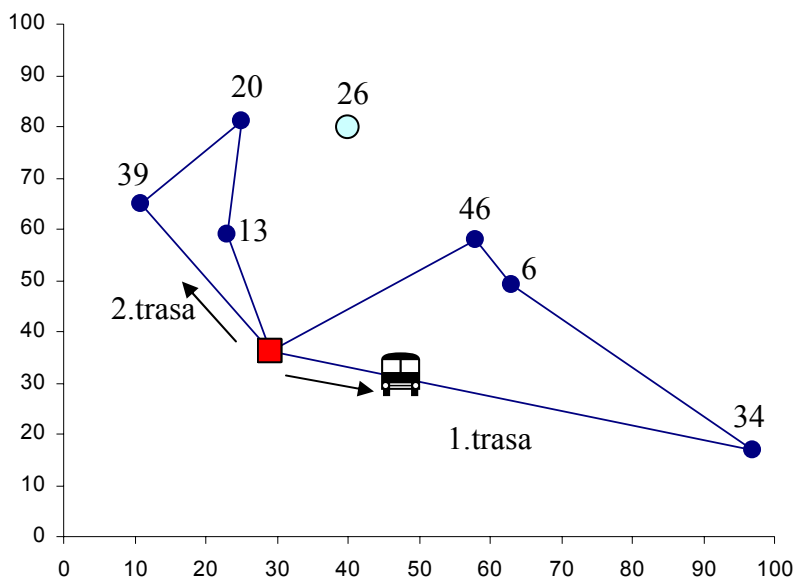
Obr. 5.2 – Vkládací algoritmus pro dynamickou rozvozní úlohu



Obr. 5.3 – Výběr trasy, na které bude obslužen nový zákazník

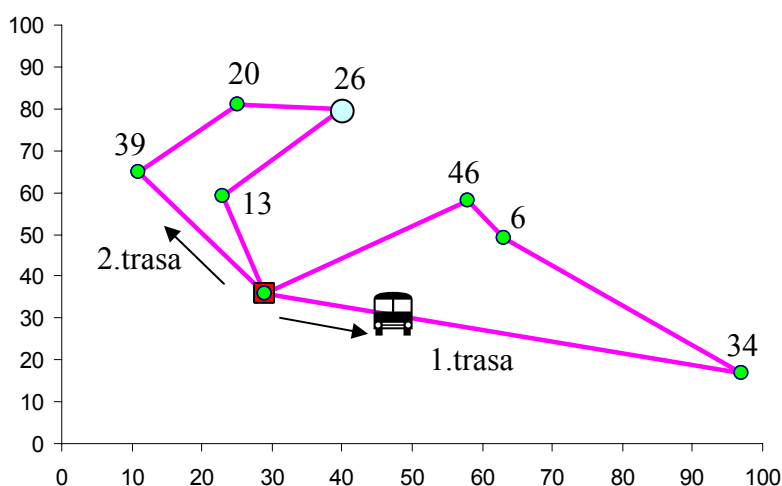
Příklad 5.2

K zákazníkům v příkladě 5.1 přidáme nového zákazníka, jehož požadavek na odvoz 26 jednotek přišel v okamžiku, kdy vozidlo vyjelo na 1. trasu (obr. 5.4).



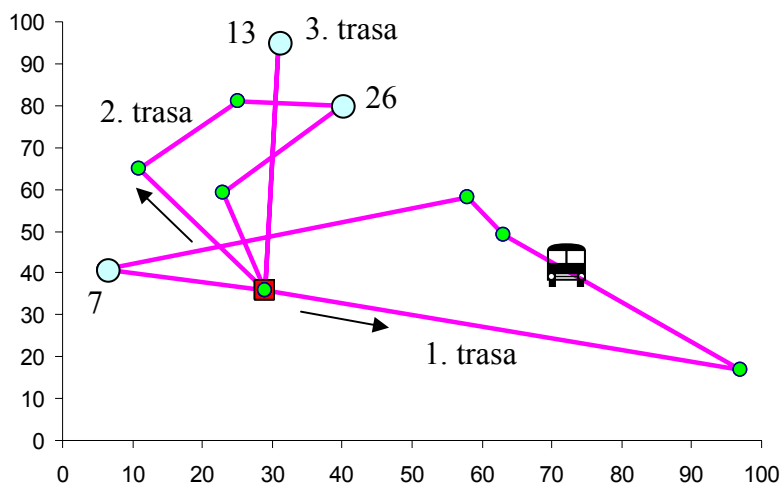
Obr. 5.4 – Nový zákazník s požadavkem 26 jednotek

Aplikujeme-li vkládací algoritmus na tuto úlohu, vozidlo nemůže zařadit nového zákazníka do 1. trasy, neboť by náklad převyšil jeho kapacitu o 12 jednotek. Po zařazení zákazníka do 2. trasy bude náklad činit $72 + 26 = 98$ jednotek, což je přípustná hodnota. Z hlediska minimálního prodloužení této trasy bude zákazník obslužen bezprostředně po zákazníkovi s 20 jednotkami (obr. 5.5). Vozidlo ujede celkem přibližně 285,21 km.



Obr. 5.5 – Zařazení nového zákazníka vkládacím algoritmem

V příloze č. 9 úloha pokračuje vygenerováním dalších 2 zákazníků. Jejich výsledné zařazení znázorňuje obr. 5.6. Zákazník s požadavkem 7 jednotek bude vložen do 1. trasy; náklad vozidla bude $86 + 7 = 93$ jednotek. Poslední zákazník s požadavkem 13 jednotek nemůže být zařazen do žádné ze dvou uvedených tras, neboť by došlo k překročení kapacity vozidla 100 jednotek. Proto bude pro obsluhu zákazníka naplánována 3. trasa. Po zařazení nových zákazníků činí délka všech tras přibližně 445,12 km.



Obr. 5.6 – Zařazení tří nových zákazníků vkládacím algoritmem

Re-optimalizace

Při popisu re-optimalizačního algoritmu lze vyjít z postupů uvedených v předchozích kapitolách. Formulace matematického modelu vyžaduje znalost následujících informací:

- místo, do něhož vozidlo směřuje v okamžiku přijetí nového požadavku, jeho index označíme j_{next} ,
- množinu U_N obsahující dosud nenavštívená místa, včetně sídla nového zákazníka označeného r ; v množině je i výchozí místo č. 1 a místo, výše označené indexem j_{next} ,
- objem nákladu $L \leq V$, který bude mít vozidlo po návštěvě místa j_{next} .

Re-optimalizační model lze pak zapsat následujícím způsobem:

$$\text{minimalizovat } z = \sum_{i \in U_N} \sum_{j \in U_N} c_{ij} x_{ij}, \quad (5.8)$$

za podmínek

$$\sum_{j \in U_N} x_{ij} = 1, \quad i \in U_N - \{1\}, \quad (5.9)$$

$$\sum_{i \in U_N} x_{ij} = 1, \quad j \in U_N - \{1\}, \quad (5.10)$$

$$u_i + q_j - V(1 - x_{ij}) \leq u_j, \quad i \in U_N, \quad j \in U_N - \{1\}, \quad i \neq j, \quad (5.11)$$

$$q_i \leq u_i \leq V, \quad i \in U_N - \{1\}, \quad (5.12)$$

$$u_1 = 0, \quad (5.13)$$

$$u_{j_{next}} = L, \quad (5.14)$$

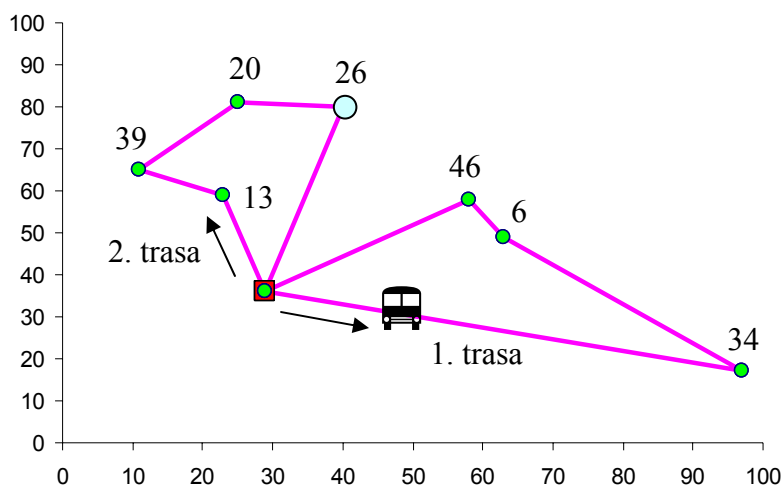
$$x_{1j_{next}} = 1, \quad (5.15)$$

$$x_{ij} \in \{0,1\}, \quad i, j \in U_N. \quad (5.16)$$

Při řešení úlohy předpokládáme, že původní vzdálenost $c_{1j_{next}}$ je změněna na hodnotu odpovídající dosud ujeté vzdálenosti⁷.

Příklad 5.3

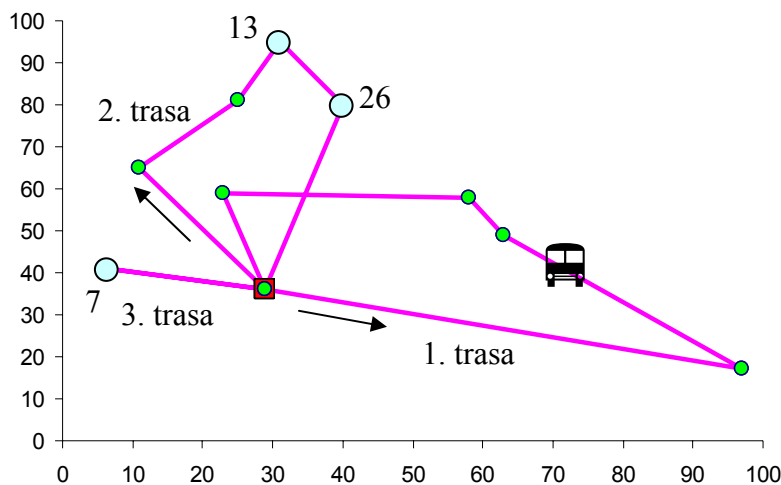
Na obr. 5.7 jsou znázorněny re-optimalizované trasy po přidání nového zákazníka s požadavkem 26 jednotek. Zatímco 1. trasa je shodná s odpovídající trasou naplánovanou pomocí vkládacího algoritmu, 2. trasa se liší. Vozidlo na obou trasách ujede přibližně 282,83 km, což je o 2,38 km méně než při aplikaci vkládacího algoritmu.



Obr. 5.7 – Zařazení nového zákazníka po re-optimalizaci tras

⁷ Případně je tato hodnota nastavena na 0. V případě, že vozidlo směřuje do výchozího místa, podmínky (5.14) a (5.15) nebudou v matematickém modelu uvedeny.

Po přidání dalších dvou zákazníků (viz příloha č. 9) za použití re-optimalizačního algoritmu bude vozidlo absolvovat trasy znázorněné na obr. 5.8, které se podstatně liší od tras získaných vkládacím algoritmem (obr. 5.6). Vozidlo ujede přibližně 366,92 km, což je výrazné zkrácení⁸ oproti hodnotě 445,12 km získané vkládacím algoritmem.



Obr. 5.8 – Zařazení tří nových zákazníků na základě re-optimalizace tras

Řešení uvedených příkladů jasně ukazuje, že se volná kapacita vozidla na trase snižuje zařazováním nových požadavků. Již po optimalizaci statické úlohy může být vozidlo na každé trase prakticky zcela zaplněno a zařazení dalších zákazníků je vyloučeno. Jejich obsluhu je pak nutné zajistit v nově vytvořených trasách. Tato strategie je nevýhodná především v situacích, kdy vozidlo jede plně naloženo kolem zákazníka velmi vzdáleného od výchozího místa a musí se k němu vrátit na některé z příštích tras.

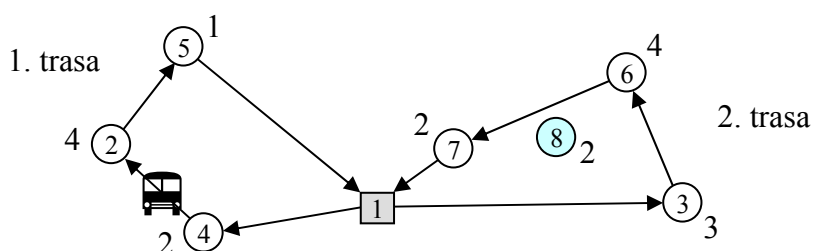
Nabízí se možnost umělého vytváření rezerv v kapacitě vozidla pro podobné případy. Z dlouhodobějšího hlediska (např. vzhledem k pracovní době firmy) může být podobná strategie výhodná ve smyslu minimalizace celkové vzdálenosti i při větším počtu tras. V popsáných modelech lze provést jednoduchou změnu tak, že se postupně zvyšuje parametr V , představující kapacitu vozidla, např. z poloviční hodnoty až na skutečnou hodnotu. Tyto úvahy, které jsou podmíněny charakterem reálné firmy a jejích zákazníků, si jistě zaslouží důkladné rozpracování v podobě vyhodnocení simulačních experimentů. Sofistikovaný přístup samozřejmě předpokládá znalost pravděpodobnostních rozdělení náhodných veličin, týkajících se vzniku nových požadavků.

⁸ Zkrácení představuje přibližně 17,5 %.

Lze se snadno přesvědčit, že vkládací algoritmus není pro rozvozní úlohy příliš vhodnou heuristickou metodou. Mnohem vhodnějším postupem je např. změna pořadí obsluhy zákazníků v rámci jedné trasy, případně výměna zákazníků obslužených v rámci různých tras, jak demonstruje následující příklad.

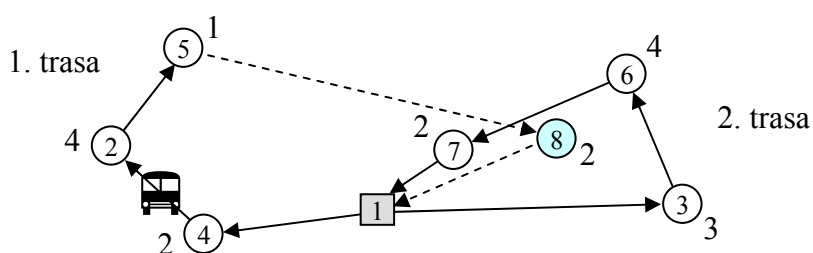
Příklad 5.4

Vozidlo o kapacitě 10 tun má svézt odpad od šesti zákazníků znázorněných na obr. 5.9 (požadavky jsou uvedeny u příslušných uzlů). Optimalizací byly určeny dvě trasy: 1-4-2-5-1 s celkovým nákladem 7 tun a 1-3-6-7-1 s nákladem 9 tun.



Obr. 5.9 – Optimální trasy svozu odpadu

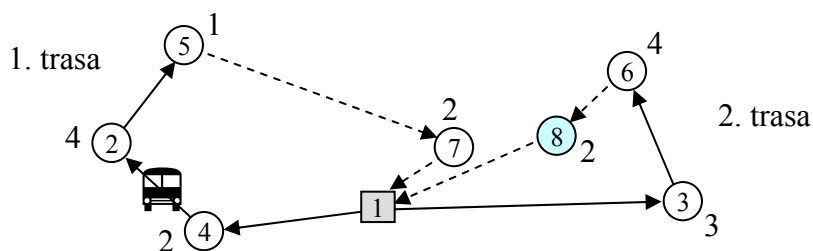
Nový zákazník č. 8, jehož požadavek o velikosti 2 tun přijde v okamžiku, kdy je vozidlo na 1. trase mezi zákazníky č. 4 a 2, nemůže být zařazen do 2. trasy, protože by došlo k překročení kapacity vozidla o 1 tunu. Vkládacím algoritmem bude tedy tento zákazník vložen do 1. trasy mezi zákazníka č. 5 a výchozí místo č. 1 (obr. 5.10). Náklad vozidla bude mít na obou trasách hodnotu 9 tun.



Obr. 5.10 – Zařazení nového zákazníka vkládacím algoritmem

Pokud bude použit výše zmíněný postup výměny zákazníků mezi jednotlivými trasami, pak zákazník č. 7 bude obslužen v rámci 1. trasy a nový zákazník č. 8 v rámci druhé trasy (obr. 5.11). Náklad vozidla bude mít na obou trasách opět hodnotu 9 tun. Tento výsledek je zřejmě shodný s výsledkem získaným na základě re-optimalizace obou tras. Je důležité poznamenat, že vložením nového zákazníka do některé z tras může dojít k překročení

kapacity (před výměnou zákazníků); ta musí být dodržena až po výměně zákazníků mezi trasami.



Obr. 5.11 – Výměna zákazníků

Navíc může být vyměněn jiný než nový zákazník, což samozřejmě při větším počtu zákazníků klade mnohem větší nároky na čas, nutný pro nalezení nejlepší výměny zákazníků. Savelsbergh (1992) uvádí možnost výměny hran, resp. i větších částí tras při řešení statických verzí okružních úloh. Tento postup lze snadno modifikovat pro řešení dynamických úloh.

5.2 Rozvozní úloha s více vozidly v jednom výchozím místě

5.2.1 Minimalizace celkové ujeté vzdálenosti

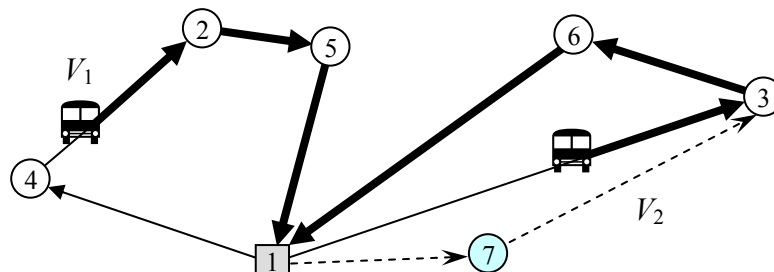
V předchozí kapitole byla popsána úloha s jedním vozidlem, které obsluhuje všechny předem známé i nové zákazníky. Vzhledem k tomu, že je omezena kapacita vozidla, bude vozidlo v takových úlohách s velkou pravděpodobností absolvovat několik tras. Někteří zákazníci mohou tak na obsluhu čekat příliš dlouho. Východiskem se stává možnost použít ke svozu více vozidel. Je dobré si uvědomit dvě základní, i když zcela zřejmé, skutečnosti:

- pokud neuvažujeme kapacitu vozidla, bude optimální, aby obsluhu zákazníků zajistilo jediné vozidlo (viz část 4.3.1),
- pokud uvažujeme kapacitu vozidla a ke svozu bude použito více vozidel, pak z důvodu, že vozidla mohou vyjet z výchozího místa současně, budou výsledné trasy po zařazení nových zákazníků odlišné od tras, které bude postupně absolvovat jedno vozidlo; význam této skutečnosti si doložíme na následujícím příkladě.

Příklad 5.5

Na obr. 5.12 jsou zákazníci č. 4, 2 a 5 obsluhováni vozidlem V_1 , zákazníci č. 3 a 6 vozidlem V_2 . Přijde-li nový požadavek zákazníka č. 7 v okamžiku, kdy jsou obě vozidla na trase (znázorněno symbolem), bude zřejmě obsloužen vozidlem V_2 po návštěvě zákazníka č. 6.

Pokud by existovalo jediné vozidlo V_1 , byl by nový zákazník tímto vozidlem obslužen bezprostředně po jeho návratu do výchozího místa ze své první trasy (znázorněno přerušovanými hranami)⁹. Rozdíl v prodloužení tras je zřejmý.



Obr. 5.12 – Obsluha zákazníků dvěma vozidly

Uvedený příklad ukazuje, že ne vždy je výhodné zařadit více vozidel do obsluhy zákazníků. Jak však bylo výše naznačeno, zákazník, který zavolá v 9 hodin ráno a je obslužen v 5 hodin odpoledne, může upustit od další spolupráce s nepružnou distribuční firmou. V případě velkého množství zákazníků či velkých vzdáleností mezi nimi samozřejmě jediné vozidlo nedokáže požadovanou obsluhu zajistit vůbec. Dalším problémem hodným důkladné analýzy je určení optimálního počtu vozidel nasazených na určitou směnu.

V této práci je předpokládán konstantní vozový park, tj. předem daný a neměnný počet vozidel s předem známou kapacitou. Dalším předpokladem je, že tato vozidla mohou, ale nemusí být využita, tj. na obsluze zákazníků se může podílet jen část vozového parku.

5.2.1.1 Statická úloha

Nejprve uvedeme matematický model úlohy, ve které je k obsluze $n-1$ předem známých zákazníků připraveno ve výchozím místě K vozidel.

$$\text{minimalizovat } z = \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^k, \quad (5.17)$$

za podmínek

$$\sum_{k=1}^K \sum_{j=1}^n x_{ij}^k = 1, \quad i = 2, 3, \dots, n, \quad (5.18)$$

$$\sum_{k=1}^K \sum_{i=1}^n x_{ij}^k = 1, \quad j = 2, 3, \dots, n, \quad (5.19)$$

⁹ U obou vozidel předpokládáme, že zařazením požadavku nového zákazníka nebude překročena jejich kapacita.

$$\sum_{j=2}^n x_{1j}^k \leq 1, \quad k = 1, 2, \dots, K, \quad (5.20)$$

$$\sum_{i=1}^n x_{ij}^k = \sum_{i=1}^n x_{ji}^k, \quad j = 2, 3, \dots, n, \quad k = 1, 2, \dots, K, \quad (5.21)$$

$$u_i + q_j - V(1 - x_{ij}^k) \leq u_j, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j, \quad k = 1, 2, \dots, K, \quad (5.22)$$

$$u_1 = 0, \quad (5.23)$$

$$q_i \leq u_i \leq V, \quad i = 2, 3, \dots, n, \quad k = 1, 2, \dots, K, \quad (5.24)$$

$$x_{ii}^k = 0, \quad i = 1, 2, \dots, n, \quad k = 1, 2, \dots, K, \quad (5.25)$$

$$x_{ij}^k \in \{0, 1\}, \quad i, j = 1, 2, \dots, n, \quad k = 1, 2, \dots, K. \quad (5.26)$$

Formulace vychází z modelů již dříve uvedených, není tedy zapotřebí uvádět podrobný popis jeho částí. Je ovšem nutné upozornit na jednu podstatnou skutečnost. Nutnou, nikoli však postačující (viz následující příklad), podmínkou pro existenci přípustného řešení uvedeného matematického modelu je následující nerovnost:

$$\sum_{i=2}^n q_i \leq K \cdot V, \quad (5.27)$$

tj. součet požadavků všech zákazníků nesmí překročit celkovou kapacitu všech vozidel.

Příklad 5.6

Ve výchozím místě jsou umístěna 2 vozidla o stejné kapacitě $V = 10$ tun, která mají svézt odpad od 3 zákazníků s požadavky 5, 6 a 7 tun. Je zcela zřejmé, že pro svoz je zapotřebí 3 vozidel, neboť součet žádných dvou požadavků nesplňuje kapacitu vozidla. Pokud nelze navýšit počet vozidel, bude jedno z vozidel absolvovat 2 trasy.

Jednou z možností, jak upravit uvedený matematický model pro nalezení přípustného řešení, je vhodně zvýšit hodnotu K představující počet vozidel. V takovém případě bude tento parametr představovat horní mez pro počet absolvovaných tras. Ve všech následujících úlohách budeme předpokládat, že počet vozidel K postačuje pro obsluhu zákazníků.

5.2.1.2 Dynamická úloha

Výsledkem optimalizace modelu (5.17) – (5.26) jsou naplánované trasy některých vozidel. Zbývající vozidla zůstávají ve výchozím místě. Pro zajištění obsluhy nově vznikajících požadavků je rozhodující, podobně jako v případě jediného obsluhujícího vozidla, zda po

zařazení zákazníků do trasy nedojde k překročení kapacity vozidla. Při splnění této podmínky je pak nový zákazník zařazen tak, aby prodloužení trasy bylo minimální.

Vkládací algoritmus pro zařazení nově vzniklých požadavků do existujících tras, případně pro vytvoření nové trasy, je velice podobný algoritmům popsaným u předchozích úloh. Dříve vytvořená schémata mohou sloužit jako návod pro čtenáře k případné algoritmizaci řešení dalších úloh tohoto typu. Jak bylo navíc demonstrováno v příkladu 5.4, je použití této heuristické metody pro rozvozní úlohy nevhodné. Existují jiné, mnohem vhodnější, heuristické či metaheuristické postupy, které však z důvodu zaměření této práce zkoumat nebudeme. V dalším textu se tedy soustředíme jen na možnost re-optimalizace tras.

Přestože také matematický model pro **re-optimalizaci tras** je v mnohém podobný modelům formulovaným pro předchozí úlohy uvedené v této práci, uvedeme jeho úplnou formulaci. Z množiny všech vozidel (množina jejich indexů je označen ω) vyčleníme vozidla, která jsou na trase (množina indexů je označena ω_1). Dále je zapotřebí znát indexy míst, do kterých směřují vozidla v okamžiku vzniku nového požadavku) a upravit původní vzdálenosti mezi výchozím místem a těmito místy na hodnoty rovnající se dosud ujeté vzdálenosti příslušným vozidlem, případně na nulové hodnoty. Podobně jako v modelu (5.8) – (5.16) je nutné znát velikost nákladu L_k , který bude mít k -té vozidlo po návštěvě místa, do kterého právě směřuje. Formulace matematického modelu je pak následující:

$$\text{minimalizovat } z = \sum_{k \in \omega} \sum_{i \in U_N} \sum_{j \in U_N} c_{ij} x_{ij}^k, \quad (5.28)$$

za podmínek

$$\sum_{k \in \omega} \sum_{j \in U_N} x_{ij}^k = 1, \quad i \in U_N - \{1\}, \quad (5.29)$$

$$\sum_{k \in \omega} \sum_{i \in U_N} x_{ij}^k = 1, \quad j \in U_N - \{1\}, \quad (5.30)$$

$$\sum_{\substack{j \in U_N \\ j \neq 1}} x_{1j}^k \leq 1, \quad k \in \omega, \quad (5.31)$$

$$\sum_{i \in U_N} x_{ij}^k = \sum_{i \in U_N} x_{ji}^k, \quad j \in U_N - \{1\}, \quad k \in \omega, \quad (5.32)$$

$$u_i + q_j - V(1 - x_{ij}^k) \leq u_j, \quad i \in U_N, \quad j \in U_N - \{1\}, \quad i \neq j, \quad k \in \omega, \quad (5.33)$$

$$q_i \leq u_i \leq V, \quad i \in U_N - \{1\}, \quad k \in \omega, \quad (5.34)$$

$$u_1 = 0, \quad (5.35)$$

$$u_{next_k} = L_k, \quad k \in \omega_1, \quad (5.36)$$

$$x_{1next_k}^k = 1, \quad k \in \omega_1, \quad (5.37)$$

$$x_{ii}^k = 0, \quad i \in U_N, \quad k \in \omega, \quad (5.38)$$

$$x_{ij}^k \in \{0,1\}, \quad i, j \in U_N, \quad k \in \omega. \quad (5.39)$$

Význam jednotlivých částí matematického modelu je zcela zřejmý podle jejich popisu u předchozích matematických modelů.

5.2.2 Minimalizace času potřebného k obslužení všech zákazníků

Tato úloha je velmi podobná úloze s více obchodními cestujícími umístěnými v jednom výchozím místě. Jediný rozdíl je v zavedení omezené kapacity vozidel, kterou je nutné na trasách respektovat.

5.2.2.1 Statická úloha

Pro formulaci matematického modelu lze využít model popsany v části 4.4.1. Místo omezujících podmínek (4.26) budou v modelu definovány následující podmínky:

$$u_i + q_j - V(1 - x_{ij}^k) \leq u_j, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j, \quad k = 1, 2, \dots, K. \quad (5.40)$$

Navíc budou připojeny podmínky týkající se nákladu vozidla po návštěvě určitého místa a kapacity vozidla:

$$q_i \leq u_i \leq V, \quad i = 2, 3, \dots, n. \quad (5.41)$$

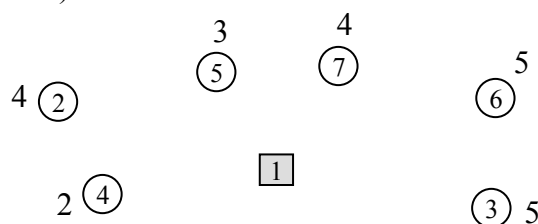
Výsledkem optimalizace jsou trasy, na nichž nedojde k překročení kapacity vozidel a při jejichž dodržení bude čas nutný ke svozu zboží od všech zákazníků minimální. Jak bylo demonstrováno na příkladě 4.5, je vhodné provést dodatečnou optimalizaci dílčích tras z hlediska minimalizace vzdálenosti, aniž by byl porušen původní požadavek na obsluhu všech zákazníků v minimálním čase.

Zvláště v této úloze je zapotřebí striktně předpokládat, že počet vozidel postačuje k obsluze všech zákazníků. Pokud by byl počet vozidel příliš nízký, bylo by nutné využít některá vozidla na více trasách. V takovém případě ale nelze uměle navýšit hodnotu K podobně jako v úloze s minimalizací celkové vzdálenosti popsané v části 5.2.1.1 (viz příklad 5.6). V této úloze je totiž lhostejné, zda obsluhu zajistí více vozidel, přičemž

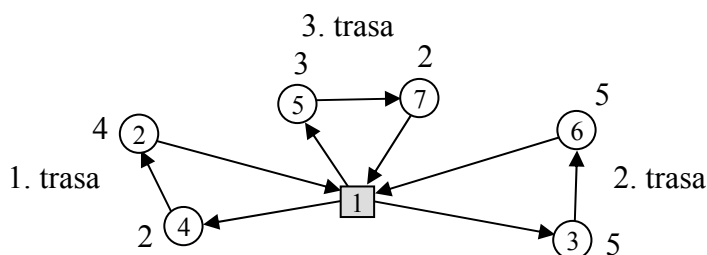
každé vozidlo absolvuje právě jednu trasu, nebo bude použito méně vozidel, z nichž některá vyjedou vícekrát. Naopak v úloze, ve které minimalizujeme čas potřebný k obslužení všech zákazníků, je výhodné použít více vozidel absolvujících právě jednu trasu než méně vozidel vyjíždějících vícekrát. Umělé navýšení počtu vozidel je však při aplikaci výše uvedeného modelu nepřipustné.

Příklad 5.7

Pro obsluhu šesti zákazníků jsou určena dvě vozidla s kapacitou 10 tun umístěná v jednom výchozím místě (obr. 5.13). Pokud bude pro řešení úlohy využito postupu naznačeného v úvodu této části, tj. matematického modelu pospaného v části 4.4.1 se změnami (5.40) a (5.41), pak úloha nebude mít přípustné řešení. Všechny zákazníky nelze obsloužit v rámci jedné či dvou tras, neboť součet všech požadavků je 21 tun, což je o 1 tunu více než je součet kapacit obou vozidel. Zvýšíme-li hodnotu K a použijeme-li stejný model, pak lze získat jeho optimální řešení (obr. 5.14).



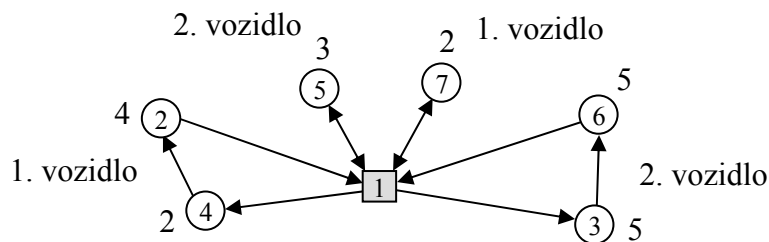
Obr. 5.13 – Výchozí místo se dvěma vozidly a 6 zákazníků



Obr. 5.14 – Optimální řešení pro tři vozidla

Minimální čas potřebný k obslužení všech zákazníků je zřejmě roven času, po který bude vozidlo na 2. trase. Vzhledem k tomu, že nejsou k dispozici tři vozidla, která mohou vyjet současně, ale pouze dvě vozidla, musí 3. trasu absolvovat 1. vozidlo po svém návratu z 1. trasy. Použití tohoto modelu je tedy z hlediska původní úlohy nesprávné, protože byl získán nereálný výsledek.

Optimální řešení původní úlohy je znázorněno na obr. 5.15. Obě vozidla absolvují po dvou trasách: 1. vozidlo trasy 1-4-2-1 a 1-7-1, 2. vozidlo trasy 1-3-6-1 a 1-5-1.



Obr. 5.15 – Optimální řešení pro dvě vozidla

Shrneme-li předchozí úvahy, výše zmíněný matematický model lze bez problémů použít v následujících situacích:

- ve výchozím místě je k dispozici libovolný (nekonečný) počet vozidel,
- ve výchozím místě je k dispozici omezený počet vozidel K a existuje přípustné řešení matematického modelu.

Naopak, uvedený matematický model nelze použít, pokud

- není splněna podmínka (5.27),
- je splněna podmínka (5.27), ale neexistuje přípustné řešení modelu (viz příklad 5.6).

5.2.2.2 Dynamická úloha

Také v dynamické úloze je možné využít úvahy týkající se úlohy s více obchodními cestujícími nacházejícími se v jednom výchozím místě, která byla popsána v části 4.6.2, ovšem za právě odvozených předpokladů.

Vzhledem k výše uvedeným poznámkám učiněným na adresu vkládacího algoritmu, bude pozornost soustředěna pouze na **re-optimalizaci tras**. Tentokrát bude upraven model (4.62) – (4.71). Místo podmínek (4.68) budou použity podmínky (5.33). Dále budou připojena omezení (5.34) a (5.36). I v tomto případě lze dodatečně optimalizovat navržené trasy z hlediska celkové ujeté vzdálenosti.

5.3 Rozvozní úloha s více vozidly v několika výchozích místech

Krátce se zmíníme i o rozvozní úloze, ve které je pro svoz určeno více výchozích míst, v nichž jsou umístěna vozidla o stejné kapacitě¹⁰. Úlohami tohoto typu se zabývají např. Laporte et al. (1988) a Savelsbergh and Sol (1998). Podobně jako v případě úlohy s více vozidly umístěnými v jednom výchozím místě lze i v úloze s několika výchozími místy minimalizovat celkovou ujetou vzdálenost nebo čas potřebný k obsluze všech zákazníků. Typickou aplikací je svoz pošty do třídících center či svoz platebních příkazů z poboček bank do centrálních míst. V těchto úlohách jsou však většinou všichni zákazníci předem známi a pro řešení je použito matematického modelu pro statickou rozvozní úlohu.

Příkladem použití dynamického modelu je svoz odpadu od zákazníků, z nichž někteří zavolají před zahájením jízd vozidel, jiní volají až během dne, kdy jsou vozidla na trase. Speciální úlohou tohoto typu, na které lze využít i pravděpodobnosti vzniku požadavku na určitém předem známém místě, je svoz tříděného odpadu z kontejnerů umístěných na známých stanovištích. Lze provádět jejich pravidelnou kontrolu a požadavek na vyprázdnění vznést až při jejich naplnění. V případě, že známe pravděpodobnosti naplnění kontejneru, je možné vytvářet záměrně rezervu kapacity vozidel, případně i odvézt obsah ne zcela zaplněného kontejneru, pokud je možné tento obsah odhadnout.

Na základě dříve uvedených matematických modelů a metod lze poměrně snadno odvodit postupy pro řešení rozvozní úlohy s více vozidly ve více výchozích místech. Pokud jde o minimalizaci celkové vzdálenosti, jedná se především o modely v částech 4.3.2 (statická úloha s více obchodními cestujícími v několika výchozích místech), 4.6.1 (dynamická úloha s více obchodními cestujícími v několika výchozích místech) a části 5.2.1 (statická a dynamická rozvozní úloha s více vozidly v jednom výchozím místě).

V případě minimalizace času potřebného k obsluze všech zákazníků lze využít části 4.4.2 (statická úloha s více obchodními cestujícími v několika výchozích místech), 4.6.3 (dynamická úloha s více obchodními cestujícími v několika výchozích místech) a část 5.2.2 (statická a dynamická rozvozní úloha s více vozidly v jednom výchozím místě).

¹⁰ V každém výchozím místě je právě jedno vozidlo.

5.4 Rozvozní úloha s dělenou dodávkou

Ve všech rozvozních úlohách dosud uvedených v této práci byl zaveden předpoklad, že dodávka určitému zákazníkovi nesmí být rozdělena do více částí. V úlohách, ve kterých se jedná o svoz zboží, materiálu nebo odpadu do jednoho či více míst, je celý požadavek každého zákazníka splněn najednou, tj. vozidlo, které zákazníka navštíví, naloží a odveze vše, co zákazník odvézt chce. Z tohoto předpokladu vyplývá, resp. tento předpoklad vyžaduje, aby v matematickém modelu byly formulovány podmínky zakazující navštívit některého zákazníka více než jednou.

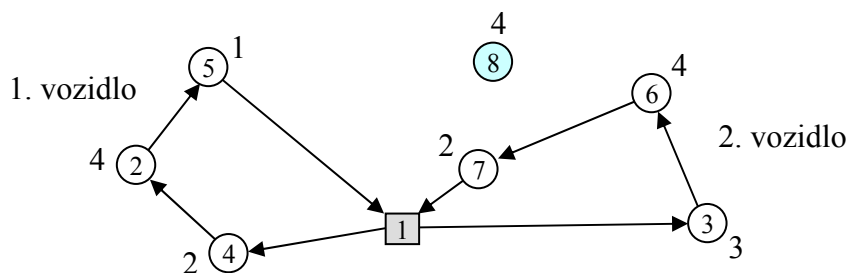
Jak je zřejmé z některých dříve uvedených příkladů, vzhledem k omezené kapacitě vozidel se může snadno vyskytnout situace, kdy nový požadavek nemůže být obsloužen žádným z vozidel na trase, protože by došlo k překročení jeho kapacity. Kdyby ovšem mohl být tento požadavek rozložen do několika částí a tyto části pak byly naloženy na různá vozidla, může dojít k výraznému snížení přepravních nákladů, resp. zkrácení celkové vzdálenosti ujeté všemi vozidly. Statickou verzi tohoto problému lze v literatuře nalézt pod názvem rozvozní úloha s dělenou dodávkou (**SDVRP**)¹¹. Pro řešení této úlohy byl navržen algoritmus tabu search (Archetti et al, 2006). Otázkou řešitelnosti a rozložitelnosti úlohy s dělenou dodávkou se zabývali Archetti et al. (2001, 2005).

Následující příklad demonstruje výhodu, kterou tento přístup k řešení reálných úloh nabízí. Analýzu úspor vzniklých díky možnosti rozdělení dodávky mezi více vozidel analyzovali Dror and Trudeau (1989), kteří zároveň prezentovali heuristický algoritmus pro řešení úloh tohoto typu.

Příklad 5.8

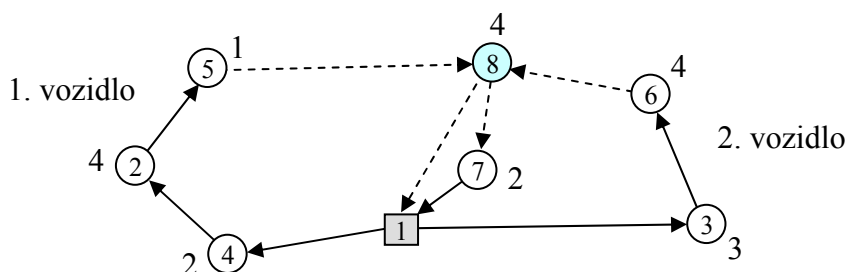
Příklad vychází ze situace uvedené v příkladě 5.4. Dvě vozidla o kapacitě 10 tun mají za úkol svézt odpad od 6 zákazníků (viz obr. 5.16); požadavky zákazníků jsou uvedeny u příslušných uzlů. Optimalizací byly určeny dvě trasy: 1-4-2-5-1 s nákladem 7 tun a 1-3-6-7-1 s nákladem 9 tun. Vznikne-li nový požadavek zákazníka č. 8 o velikosti 4 tun, nemůže jej zařadit do své trasy ani jedno z vozidel z důvodu překročení své kapacity. V takovém případě bude muset některé z vozidel nového zákazníka obsloužit po návratu do výchozího místa.

¹¹ Split Delivery Vehicle Routing Problem.



Obr. 5.16 – Optimální trasy dvou vozidel svážejících odpad

Existuje-li však možnost rozdělit náklad mezi obě vozidla, pak lze najít optimální trasy¹² (obr. 5.17), kdy 1. vozidlo odebere 3 tuny odpadu a 2. vozidlo 1 tunu odpadu. Obě vozidla tak budou zcela naplněna.



Obr. 5.17 – Optimální rozdělení nového požadavku

V této práci se zaměříme na rozvozní úlohu, jejímž cílem je minimalizovat celkovou vzdálenost. Budeme předpokládat existenci jediného výchozího místa, ve kterém je umístěno několik vozidel určených ke svozu zboží. Jak bylo naznačeno dříve, je poměrně snadné tuto úlohu rozšířit i na jiné typy úloh týkajících se minimalizace času potřebného k obsluze všech zákazníků, případně na úlohy, v nichž existuje více výchozích míst.

5.4.1 Statická úloha

Pro řešení statické úlohy s K vozidly umístěnými v jednom výchozím místě, která jsou určena pro svoz zboží od $n-1$ zákazníků, lze použít následující matematický model (Fábry, 2005):

$$\text{minimalizovat } z = \sum_{k=1}^K \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^k, \quad (5.42)$$

za podmínek

$$\sum_{j=2}^n x_{1j}^k \leq 1, \quad k = 1, 2, \dots, K, \quad (5.43)$$

¹² Optimální z hlediska celkové ujeté vzdálenosti.

$$\sum_{i=1}^n x_{ij}^k = \sum_{i=1}^n x_{ji}^k, \quad j = 2, 3, \dots, n, \quad k = 1, 2, \dots, K, \quad (5.44)$$

$$u_i^k + q_j^k - V(1 - x_{ij}^k) \leq u_j^k, \quad i = 1, 2, \dots, n, \quad j = 2, 3, \dots, n, \quad i \neq j, \quad k = 1, 2, \dots, K, \quad (5.45)$$

$$u_1^k = 0, \quad k = 1, 2, \dots, K, \quad (5.46)$$

$$q_i^k \leq u_i^k \leq V, \quad i = 2, 3, \dots, n, \quad k = 1, 2, \dots, K, \quad (5.47)$$

$$\sum_{k=1}^K q_i^k = q_i, \quad i = 2, 3, \dots, n, \quad (5.48)$$

$$0 \leq q_i^k \leq q_i \sum_{j=1}^n x_{ij}^k, \quad i = 2, 3, \dots, n, \quad k = 1, 2, \dots, K, \quad (5.49)$$

$$x_{ii}^k = 0, \quad i = 1, 2, \dots, n, \quad k = 1, 2, \dots, K, \quad (5.50)$$

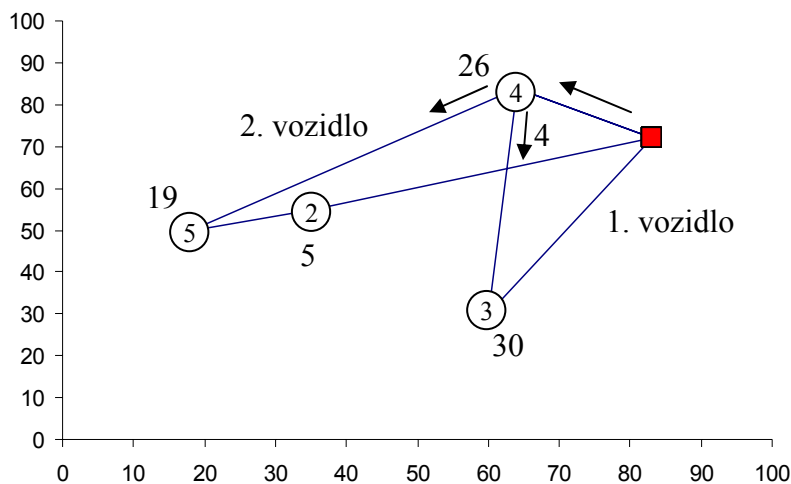
$$x_{ij}^k \in \{0, 1\}, \quad i, j = 1, 2, \dots, n, \quad k = 1, 2, \dots, K. \quad (5.51)$$

Proměnná q_i^k v modelu představuje část požadavku i -tého zákazníka q_i , která bude odvezena vozidlem na trase k . Podmínky (5.48) zajišťují, aby vozidla tímto způsobem odvezla od každého zákazníka požadované množství. Nerovnosti (5.49) mají dvojí význam. V případě, že k -tá trasa nevede přes i -tého zákazníka, tj. $x_{ij}^k = 0$ pro $j = 1, 2, \dots, n$, pak se na této trase od i -tého zákazníka nic neodváží, tj. $q_i^k = 0$. Pokud se na k -té trase od i -tého zákazníka odváží část požadavku $q_i^k > 0$, pak musí existovat alespoň jedno místo, do kterého vozidlo pojedje v rámci této trasy přímo od i -tého zákazníka, tj. $x_{ij}^k = 1$ pro některý index j .

Uvedený model je samozřejmě nutné použít v případě, že pro požadavek alespoň jednoho ze zákazníků platí, že $q_s > V$. Pak totiž musí být požadavek s -tého zákazníka odvezen na více trasách.

Příklad 5.9

V příloze č. 10 je uveden demonstrační příklad rozvozní úlohy s dělenou dodávkou. Jedná se o svoz materiálu, pro který jsou k dispozici 3 vozidla o stejné kapacitě 50 jednotek. Před zahájením jízdy vozidel jsou známé požadavky 4 zákazníků. Na obr. 5.18 jsou znázorněny optimální trasy dvou vozidel: 1. vozidlo postupně obslouží zákazníky č. 4 a 3, 2. vozidlo zákazníky č. 4, 5 a 2. Poslední vozidlo zůstane nevyužito ve výchozím místě. Požadavek zákazníka č. 4 o velikosti 30 jednotek je z části odvezen 1. vozidlem (4 jednotky) a z části 2. vozidlem (26 jednotek). Obě vozidla ujedou celkem přibližně 272,82 km.



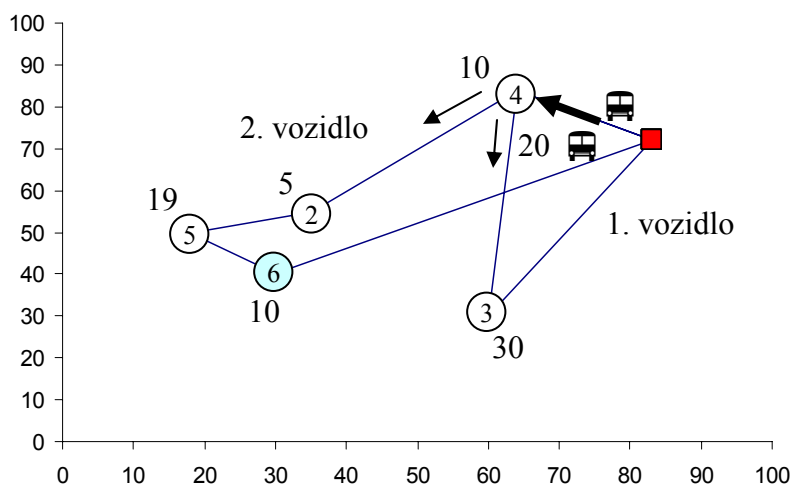
Obr. 5.18 – Optimální svoz materiálu od 4 zákazníků

5.4.2 Dynamická úloha

Při zařazování nově vzniklých požadavků v rozvozní úloze s dělenou dodávkou bude pozornost soustředěna na re-optimalizační algoritmus, jehož použití bude nejprve demonstrováno na několika příkladech, které jsou pokračováním příkladu 5.9.

Příklad 5.10

Nový požadavek zákazníka č. 6 na odvoz 10 jednotek vznikne v okamžiku, kdy obě vozidla směřují z výchozího místa k zákazníkovi č. 4 (obr. 5.19). Podle předpokladu o zákazu změny směru jízdy musí obě vozidla k tomuto zákazníkovi dorazit.



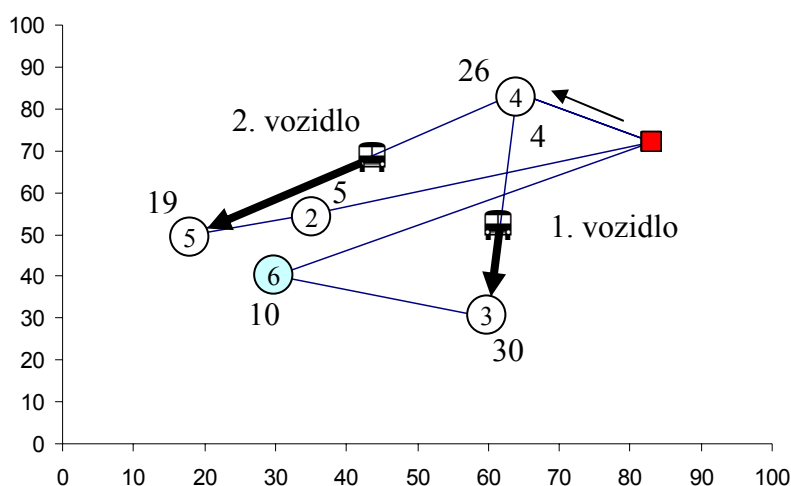
Obr. 5.19 – Vznik nového požadavku v okamžiku jízdy vozidel k zákazníkovi č. 4

Na základě optimalizace statické úlohy bylo naplánováno, že 1. vozidlo odebere v tomto místě 4 jednotky, 2. vozidlo 26 jednotek.

Re-optimalizací zbývajících částí tras se však tato strategie změní na 20 a 10 jednotek, což umožní 2. vozidlu zařadit do své trasy zákazníka č. 6, a to na místo, které je z hlediska minimalizace celkové vzdálenosti ujeté oběma vozidly optimální. Na této trase vozidlo místo původních 50 jednotek svezde do výchozího místa pouze 44 jednotek, zatímco 1. vozidlo, jehož trasa se nezmění co do pořadí navštívených zákazníků, místo původních 34 jednotek svezde 50 jednotek, čímž bude plně využito. Vozidla ujedou přibližně 283,24 km.

Příklad 5.11

Nový požadavek zákazníka č. 6 vznikne v okamžiku, kdy 1. vozidlo jede od zákazníka č. 4 k zákazníkovi č. 3 a 2. vozidlo jede od zákazníka č. 4 k zákazníkovi č. 5 (obr. 5.20). Obě vozidla odvezla od zákazníka č. 4 původně plánované počty jednotek: 4 a 26. Nový zákazník bude zařazen do trasy 1. vozidla, trasa 2. vozidla ani jeho celkový náklad se nezmění. Celková vzdálenost činí přibližně 318,47 km.



Obr. 5.20 – Vznik nového požadavku v okamžiku jízdy vozidel od zákazníka č. 4

Re-optimalizační algoritmus pro rozvozní úlohu s dělenou dodávkou lze odvodit z matematických modelů (5.42) – (5.51) a (5.28) – (5.39). Demonstrační příklady 5.10 a 5.11 ukazují, že pro formulaci re-optimalizačního modelu je nutné znát nejen indexy míst, do kterých jednotlivá vozidla směřují v okamžiku vzniku nového požadavku, ale také místa, z kterých právě jedou. Důvodem je skutečnost, že vozidlo nemůže měnit svůj současný náklad (po návštěvě posledního místa), ale může být provedena změna plánovaného nákladu po návštěvě dalších míst. Přitom je samozřejmě nutné respektovat již zmíněný předpoklad, že vozidlo musí dorazit do místa, do kterého právě směřuje.

Původní matice vzdáleností bude tudíž upravena tak, že se změní vzdálenosti mezi výchozím místem a místy právě navštívenými vozidly. V modelu pak budou zahrnuty následující podmínky:

$$q_{prev_k}^k = L_k, \quad k \in \omega_1, \quad (5.52)$$

$$x_{1prev_k}^k = 1, \quad k \in \omega_1, \quad (5.53)$$

$$x_{prev_k,next_k}^k = 1, \quad k \in \omega_1, \quad (5.54)$$

kde ω_1 je množina indexů vozidel, která jsou na trase a L_k je současný náklad k -tého vozidla, tj. jeho náklad po obslužení předchozího zákazníka s indexem $prev_k$. Zbývající části modelu zde neuvádíme, neboť je poměrně snadné na základě předešlého textu všechny podmínky odvodit.

5.4.3 Rozložitelnost úlohy s dělenou dodávkou

Archetti et al. (2001, 2005) se zabývali výpočetní složitostí a rozložitelností úlohy s dělenou dodávkou. Vymezili speciální úlohu rozvozu kontejnerů (SDP)¹³ vozidly o kapacitě $V = 2$, kterou lze řešit v polynomiálním čase; obecná úloha SDVRP s vozidly o kapacitě $V \geq 3$ patří do třídy NP-obtížných úloh.

Výše uvedení autoři definují rozložitelnost úlohy s dělenou dodávkou následujícím způsobem. Úloha je rozložitelná, jestliže existuje její optimální řešení takové, že každý zákazník požadující alespoň množství V je obslužen maximálním možným počtem přímých jízd z výchozího místa s plně využitou kapacitou V . Zbývající požadavek každého zákazníka je pak nižší než V .

Předpokládejme, že všechna vozidla mají stejnou kapacitu, tj. $V_k = V$ ($k = 1, 2, \dots, K$), kde K je počet vozidel. Pro požadavek q_i ($i = 2, 3, \dots, n$) takový, že $q_i > V$, lze definovat $q_i = n_i V + q'_i$, kde n_i počet přímých jízd z výchozího místa k zákazníkovi (trasa 1- i -1) a $q'_i \in (0, V)$ je zbývající množství, které je rozvezeno dodatečně. Tento dodatečný rozvoz lze realizovat na základě řešení redukované rozvozní úlohy s dělenou dodávkou pro poptávku q'_i ($i = 2, 3, \dots, n$). Je zřejmé, že v takové úloze nejsou z původní množiny zákazníků ti zákazníci, pro něž je $q'_i = 0$.

¹³ Skip Delivery Problem.

Archetti et al. (2001) dokázali, že symetrická rozvozní úloha SDP, pro kterou platí trojúhelníkové nerovnosti, je rozložitelná. Symetrická rozvozní úloha SDVRP s vozidly o kapacitě $V = 3$ je rozložitelná v případě, že platí zesílené trojúhelníkové nerovnosti

$$c_{ij} \leq \alpha(c_{ik} + c_{kj}), \quad i, j, k = 1, 2, \dots, n; \quad i \neq j \neq k; \quad \alpha \in (0, 1) \quad (5.55)$$

pro $\alpha = \frac{2}{3}$.

Pro rozvozní úlohy SDVRP s vozidly o kapacitě $V > 3$ lze samozřejmě uměle zkonstruovat příklady, které je možné rozložit, obecná pravidla vedoucí na rozložitelnost jakékoli úlohy však neexistují a jsou předmětem dalšího výzkumu.

5.4.4 Heuristická metoda pro úlohu s dělenou dodávkou

Vzhledem k tomu, že existence optimálního řešení na základě rozložitelnosti rozvozní úlohy s dělenou dodávkou je prokázána jen za určitých výše uvedených podmínek, jsou v reálných aplikacích používány heuristické postupy nalezení suboptimálního řešení. Jako ukázkou uvedeme postup, který popsali Dror and Trudeau (1989).

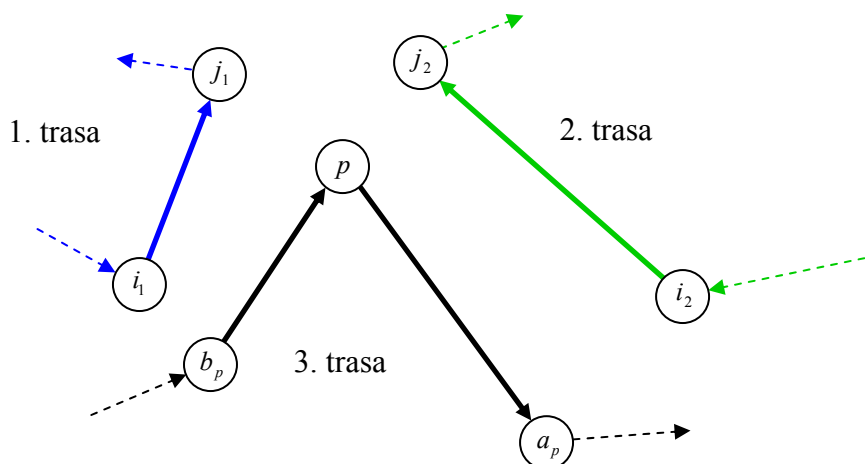
Na začátku vyjděme z nějakého přípustného řešení dané rozvozní úlohy. Předpokládejme, že existují tři trasy; vozidla na první a druhé trase mají volné kapacity $v_1, v_2 > 0$. Necht' na třetí trase existuje zákazník p s dodávkou q_p realizovanou právě na této trase tak, že platí:

$$v_1 + v_2 \geq q_p, \quad (5.56)$$

tj. tuto dodávku lze naložit na vozidla jedoucí v první a druhé trase.

Na trase 1 označme po sobě jdoucí uzly i_1, j_1 , na trase 2 pak podobně uzly i_2, j_2 a na trase 3 označme jako b_p uzel bezprostředně předcházející uzlu p a jako a_p uzel bezprostředně následující po uzlu p (obr. 5.21). Součet vzdáleností, týkající se zmíněných uzlů na všech třech trasách lze vyjádřit takto:

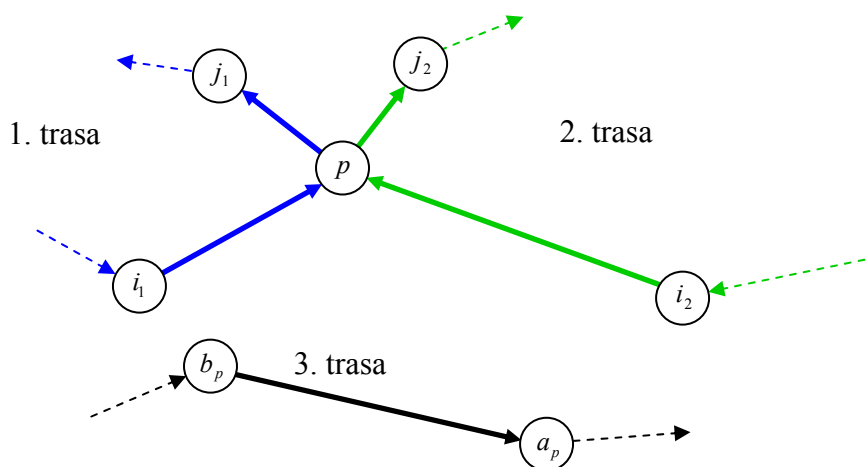
$$z = c_{i_1 j_1} + c_{i_2 j_2} + c_{b_p p} + c_{p a_p}. \quad (5.57)$$



Obr. 5.21 – Výchozí přípustné řešení

Jestliže při respektování omezení (5.56) rozdělíme původní dodávku q_p do uzlu p na trasu 3 na vozidla jedoucí po trasách 1 a 2, získáme nový součet vzdáleností:

$$z' = c_{i_1 p} + c_{p j_1} + c_{i_2 p} + c_{p j_2} + c_{b_p a_p}. \quad (5.58)$$



Obr. 5.22 – Nové řešení

Hodnota $\Delta z = (z - z') > 0$ je pak úspora vzdálenosti, které bylo dosaženo tím, že původní dodávka do uzlu p byla přiřazena k trasám 1 a 2, čímž byl uzel p z trasy 3 zcela vyloučen (obr. 5.22). Podobným způsobem lze dodávku do určitého uzlu rozdělit do více než 2 tras, jak uvádějí Dror and Trudeau (1989).

6. Výpočetní experimenty

V této části práce budou prezentovány výsledky experimentů¹ s některými modely uvedenými v předchozích kapitolách. Pro testování jsem mohl využít reálná data ve formě matice vzdáleností mezi reálnými místy. Vzhledem k tomu, že jsem neměl k dispozici žádné reálné údaje o nově vzniklých požadavcích (tj. místo a okamžik vzniku), které by bylo nutné vygenerovat, rozhodl jsem se opět využít pouze vygenerovaná data. Důvodem pro toto rozhodnutí byla i skutečnost, že nebylo zapotřebí vytvářet speciální prostředí pro jednotlivé algoritmy a mohl jsem využít veškeré prostředky, které jsem v této práci použil pro demonstraci jednotlivých modelů a metod. Pro všechny demonstrační úlohy byl jako řešitel zvolen optimalizační systém LINGO, verze 9.0, v některých experimentech byl pro porovnání z hlediska výpočetního času vybrán softwarový produkt XPRESS MP, release 2005.

6.1 Dynamická úloha obchodního cestujícího

Pro výpočetní experimenty s modely dynamické úlohy obchodního cestujícího byl použit systém *TSP.xls*, který je popsán v příloze č. 3. V tabulce 6.1 jsou uvedeny výsledky pro dva různé typy experimentů. V prvním z nich označeném Exp 1 se nový požadavek objevuje bezprostředně po zahájení jízdy, v experimentu Exp 2 vzniká tento požadavek až ve druhé polovině trasy.

Dynamická úloha obchodního cestujícího								
Počet míst	Exp 1				Exp 2			
	Reoptimalizace			Vkládací algoritmus	Reoptimalizace			Vkládací algoritmus
	Délka trasy		Doba výpočtu	Délka trasy	Délka trasy		Doba výpočtu	Délka trasy
	OPT	1 min			OPT	1 min		
5	151,21	-	0	151,21	168,00	-	0	168,00
10	319,42	-	0	343,70	348,45	-	0	357,56
15	354,43	-	16	354,43	424,88	-	4	424,88
20	402,87	-	18	402,87	403,35	-	1	407,95
25	457,49	466,49	73	457,49	464,63	-	10	464,63
30	426,10	-	48	426,10	514,69	-	1	514,86

Tab. 6.1 – Výsledky experimentů s dynamickým modelem úlohy obchodního cestujícího

¹ Pro počítačové experimenty bylo použito počítače s procesorem Intel Pentium 2,5 GHz, 512 MB RAM.

Oba typy experimentů byly postupně provedeny s daty vygenerovanými pro statickou úlohu obchodního cestujícího s příslušným počtem míst² uvedeným v prvním sloupci tabulky. U každého datového souboru byl pak v rámci dynamické úlohy vygenerován nový zákazník a byla určena nová trasa na základě re-optimalizace a pomocí vkládacího algoritmu. Celkovou délku trasy po re-optimalizaci lze najít v tabulce ve sloupci OPT. V případě, že doba výpočtu (v tabulce uvedená v sekundách) při re-optimalizaci byla delší než 1 minutu (pro 25 míst), byl algoritmus pro srovnání s optimální délkou přerušen po 1 minutě (příslušná délka trasy je ve sloupci označeném 1 min). U vkládacího algoritmu je doba výpočtu zanedbatelná.

Výsledky jsou v souladu s předpokladem, že prodloužení trasy je přímo úměrné době, která uplyne od výjezdu vozidla, tj. čím dříve se o novém zákazníkovi řidič vozidla dozví, tím kratší vzdálenost bude muset ujet. Doba re-optimalizace se zkracuje s klesajícím počtem míst, která musí vozidlo ještě navštívit, tj. pokud se nový požadavek objeví bezprostředně po výjezdu vozidla, výpočet trvá přibližně stejnou dobu jako výpočet původní statické úlohy.

Vkládací algoritmus je vhodné použít při vysokém počtu míst především v případě potřeby rychlé reakce na nově vzniklé požadavky. Předčasné ukončení re-optimalizace závisí na typu konkrétní firmy a vzdálenostech mezi postupně navštěvovanými místy, tj. na čase, který má dispečer k dispozici pro určení nové trasy. Dobu výpočtu může dispečer aktuálně měnit podle okamžité polohy vozidla.

6.2 Úloha s více obchodními cestujícími s minimalizací celkové vzdálenosti

6.2.1 Úloha s jedním výchozím místem

Tabulky 6.2 a 6.3 shrnují výpočetní experimenty s matematickým modelem (4.1) – (4.8) formulovaným v části 4.3.1 pro statickou úlohu. Tento model předpokládá, že pro obsluhu zákazníků musí být použita všechna vozidla, což při rostoucím počtu vozidel znamená zvyšující se celkovou vzdálenost (tabulka 6.2). Tabulka 6.3 obsahuje údaje o době výpočtu (v sekundách) v závislosti na počtu vygenerovaných míst a počtu vozidel, které je nutné k obsluze zákazníků použít.

² V rámci počtu míst je zahrnuto i výchozí místo.

Statická úloha s více obchodními cestujícími - 1 výchozí místo									
Počet míst	Počet vozidel								
	2	3	4	5	6	7	8	9	10
5	286,99	334,48	402,19	-	-	-	-	-	-
10	345,70	385,25	430,10	483,95	544,63	607,76	675,47	766,76	-
15	386,94	411,09	437,24	478,20	524,59	582,09	641,11	701,18	769,62
20	429,08	492,45	557,61	623,36	689,12	758,27	829,22	912,61	1007,04
25	391,64	474,92	558,71	642,57	736,42	839,52	942,70	1049,20	1157,59
30	509,84	540,12	577,15	618,11	669,96	722,98	777,36	832,87	892,81

Tab. 6.2 – Délka trasy v závislosti na počtu míst a vozidel

Statická úloha s více obchodními cestujícími - 1 výchozí místo									
Počet míst	Počet vozidel								
	2	3	4	5	6	7	8	9	10
5	0	0	0	-	-	-	-	-	-
10	0	0	0	0	0	0	0	0	-
15	1	1	1	1	1	1	1	1	1
20	38	35	28	38	17	23	11	17	7
25	127	62	44	41	38	78	47	94	37
30	373	106	125	138	37	47	35	48	34

Tab. 6.3 – Doba výpočtu (v sekundách) v závislosti na počtu míst a vozidel

6.2.2 Úloha s několika výchozími místy

Část 4.3.2 se týká statické úlohy s více obchodními cestujícími v několika výchozích místech, část 4.6.1 je věnována dynamickému rozšíření této úlohy. V následujících tabulkách jsou uvedeny příslušné experimenty, ve kterých byl použit vedle optimalizačního systému LINGO i řešitel systému XPRESS MP³. Tabulky obsahují údaje zjištěné pro 2, 3, 5 a 10 výchozích míst, resp. vozidel umístěných v těchto místech⁴. U statické úlohy byly vypočteny optimální délky tras oběma uvedenými řešiteli. Pokud zjištěná doba výpočtu (v tabulkách je zapsána v sekundách) přesáhla u systému LINGO 1 minutu, byl pro srovnání s optimálním řešením výpočet přerušen po 1 minutě (délka tras je uvedena ve sloupci nadepsaném 1 min).

³ Matematický model zapsaný v systému LINGO byl exportován do formátu MPS a ten byl pak spuštěn v systému XPRESS MP.

⁴ Čísla uvedená ve sloupci Počet míst zahrnují jak zákazníky, tak výchozí místa.

Zařazení nového zákazníka do naplánovaných tras vozidel bylo provedeno pomocí vkládacího algoritmu. Výsledky jsou uvedeny v tabulkách v části nazvané Dynamická úloha. Podobně jako v části 6.1 byl i u této úlohy zkoumán vliv okamžiku vzniku nového požadavku na prodloužení tras. Experiment Exp 1 předpokládá vznik tohoto požadavku bezprostředně po vyjetí vozidel na jejich trasy naplánované ve statické úloze, v experimentu Exp 2 se nový požadavek objevuje až v okamžiku, kdy jsou vozidla již delší dobu na trasách, resp. některá vozidla jsou již zpět ve svých výchozích místech. Údaj o délce tras ve sloupci OPT byl získán na základě vložení nového zákazníka do optimálních tras příslušné statické úlohy. Délka tras ve sloupci 1 min odpovídá vložení nového zákazníka do tras naplánovaných po přerušení optimalizace statické úlohy po 1 minutě.

Úloha s více obchodními cestujícími ve více výchozích místech								
2 vozidla								
Počet míst	Statická úloha				Dynamická úloha			
	Délka tras		Doba výpočtu		Exp 1		Exp 2	
	OPT	1 min	LINGO	XPRESS	OPT	1 min	OPT	1 min
5	181,33	-	0	0	247,49	-	313,35	-
10	299,77	-	2	0	300,28	-	364,28	-
15	295,85	295,85	231	1	296,85	296,85	328,65	328,65
20	416,37	432,05	411	2	422,38	438,06	469,40	484,26
25	425,49	-	57	3	429,90	-	503,52	-
30	467,86	511,81	2061	22	475,34	536,39	598,00	641,94

Tab. 6.4 – 2 vozidla ve 2 výchozích místech

Úloha s více obchodními cestujícími ve více výchozích místech								
3 vozidla								
Počet míst	Statická úloha				Dynamická úloha			
	Délka tras		Doba výpočtu		Exp 1		Exp 2	
	OPT	1 min	LINGO	XPRESS	OPT	1 min	OPT	1 min
5	186,95	-	0	0	222,95	-	222,95	-
10	258,57	-	19	1	274,74	-	321,06	-
15	332,28	332,28	94	4	333,26	333,26	337,84	337,84
20	382,02	-	4	1	397,63	-	401,83	-
25	392,92	-	31	4	421,92	-		-
30	470,68	470,68	80	10	473,54	473,54	493,72	493,72

Tab. 6.5 – 3 vozidla ve 3 výchozích místech

Úloha s více obchodními cestujícími ve více výchozích místech								
5 vozidel								
Počet míst	Statická úloha				Dynamická úloha			
	Délka tras		Doba výpočtu		Exp 1		Exp 2	
	OPT	1 min	LINGO	XPRESS	OPT	1 min	OPT	1 min
10	207,65	-	1	0	262,83	-	262,83	-
20	373,00	-	31	2	374,26	-	406,53	-
30	463,67	468,33	99	15	479,79	484,45	479,79	484,45

Tab. 6.6 – 5 vozidel v 5 výchozích místech

Úloha s více obchodními cestujícími ve více výchozích místech								
10 vozidel								
Počet míst	Statická úloha				Dynamická úloha			
	Délka tras		Doba výpočtu		Exp 1		Exp 2	
	OPT	1 min	LINGO	XPRESS	OPT	1 min	OPT	1 min
20	356,17	-	7	1	376,57	-	389,16	-
30	433,40	-	30	3	438,73	-	492,87	-

Tab. 6.7 – 10 vozidel v 10 výchozích místech

Vkládací algoritmus se ukazuje jako vhodná metoda pro tento typ dynamických distribučních úloh. Pro určení optimálních tras ve statické úloze postačuje systém LINGO, pro větší počet zákazníků je možné využít systém XPRESS MP, a to především v aplikacích, kdy zákazníci jsou známi těsně před vyjetím vozidel na trasy a je tedy nutné výpočet urychlit. Z údajů o době výpočtu uvedených ve sloupci XPRESS lze předpokládat, že tento systém bude vhodným nástrojem i pro re-optimalizaci tras na základě modelu (4.44) – (4.54) v části 4.6.1.

6.3 Úloha s minimalizací času potřebného k obslužení všech zákazníků

V části 4.4 jsou popsány statické úlohy s více obchodními cestujícími v jednom či více výchozích místech s cílem minimalizovat čas potřebný k obslužení všech známých zákazníků a k navrácení všech vozidel zpět do výchozího místa, resp. výchozích míst. Výpočetní experimenty ukazují, že optimalizace těchto úloh i jejich dynamického rozšíření je z výpočetního hlediska velice náročná pro vysoký počet bivalentních proměnných

a omezujících podmínek i při nízkém počtu zákazníků. V tabulce 6.8 jsou uvedeny výsledky optimalizace pro 10 míst a pro vozidla, která se nacházejí v jednom výchozím místě. Tabulka obsahuje i údaje o dodatečné optimalizaci dílčích tras z hlediska jejich celkové délky při zachování vypočteného minimálního času potřebného k obslužení všech zákazníků.

S rostoucím počtem vozidel roste počet bivalentních proměnných a doba výpočtu se úměrně prodlužuje. Přestože systém XPRESS MP poskytuje optimální výsledky mnohem rychleji než LINGO, je zřejmé, že při větším rozsahu úlohy je získání optimálního řešení nereálné. Optimalizační model lze však úspěšně použít v situaci, kdy firmě postačuje přibližné řešení. Dispečer může přerušit optimalizační algoritmus podle času, který má k dispozici pro navržení trasy. Již po několika desítkách sekund poskytují oba systémy velmi dobré řešení a často i řešení optimální⁵. Např. minimální čas 226,00 minut s délkou všech tras 766,53 km pro úlohu se 4 vozidly určil XPRESS MP již za 38 sekund, zatímco celá optimalizace trvala 2886 sekund. Podobně i dodatečná optimalizace tras mohla být přerušena po 90 sekundách, kdy již bylo dosaženo optimální délky všech tras 763,02 km.

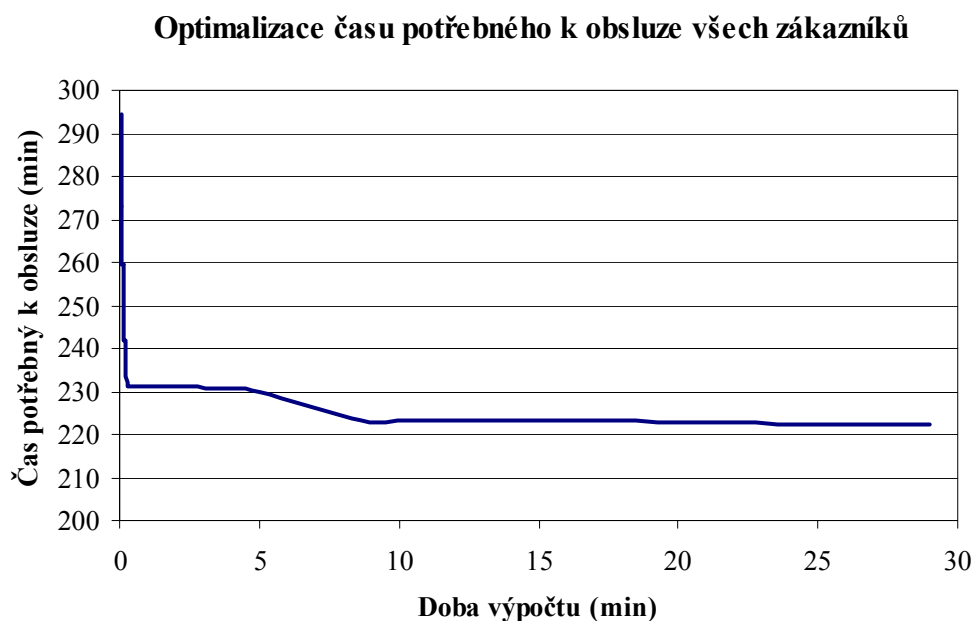
Statická úloha s více obchodními cestujícími v jednom výchozím místě							
Počet vozidel	Minimalizace času				Dodatečná optimalizace		
	Minimální čas	Délka všech tras	Doba výpočtu		Délka všech tras	Doba výpočtu	
			LINGO	XPRESS		LINGO	XPRESS
2	188,41	367,19	30	3	367,19	29	2
3	161,11	466,57	127	39	466,57	763	14
4	226,00	766,53	4260	2886	763,02	9277	1028

Tab. 6.8 – Statická úloha s 1 výchozím místem a 9 zákazníky

Graf na obr. 6.1 je příkladem průběhu optimalizace statické úlohy s více obchodními cestujícími ve více výchozích místech, v níž je cílem minimalizovat čas potřebný k obslužení všech známých zákazníků a k navrácení všech vozidel zpět do výchozích míst. Optimalizace v XPRESS MP trvala cca 29 minut. Optimálního řešení s minimální časem pro obsluhu 222,53 minut bylo dosaženo cca za 24 minut. Za cca 9 minut po zahájení optimalizace bylo získáno plně postačující řešení s hodnotou účelové funkce 223,36 minut.

⁵ Zbývající čas do ukončení optimalizace pak algoritmus pouze prohledává množinu celočíselných řešení, aniž by našel lepší řešení.

Pokud by firma přijala i řešení s hodnotou účelové funkce 231,24 minut, takové řešení je získáno po 30 sekundách výpočtu.



Obr. 6.1 – Příklad závislosti hodnoty účelové funkce na době výpočtu

6.4 Rozvozní úlohy

Experimenty s matematickými modely formulovanými pro rozvozní úlohy nabízejí mnohem větší variabilitu než experimenty pro okružní úlohy. Kromě experimentování s počtem zákazníků a vozidel lze experimentovat i s velikostí požadavků zákazníků a s kapacitou vozidel. Vzhledem k vymezeným cílům této práce mají provedené experimenty ilustrativní charakter. Soustředíme se jen na dynamickou rozvozní úlohu s jedním vozidlem uvedenou v části 5.1.2.

V tabulce 6.9 jsou uvedeny výsledky experimentu s jedním vozidlem o kapacitě 100 jednotek. Dynamická úloha vychází z řešení statické úlohy pro daný počet míst⁶. Ve všech experimentech byl uvažován nový zákazník, jehož požadavek o velikosti 10 jednotek vznikl bezprostředně po výjezdu vozidla z výchozího místa. Zařazení nového zákazníka bylo provedeno na základě re-optimalizace naplánované trasy a pomocí vkládacího algoritmu. Hodnoty představující délku trasy, resp. tras jsou u obou metod shodné, což platí ovšem jen pro tyto konkrétní experimenty; pro jiné experimenty, resp. jiná data se tyto hodnoty mohou

⁶ Údaj ve sloupci Počet míst zahrnuje i výchozí místo.

výrazně lišit. Lze očekávat, že pro vyšší hodnoty požadavků nových zákazníků, které se budou svou velikostí přibližovat kapacitě vozidla, bude výhodnější re-optimalizace. Ověření této hypotézy ovšem vyžaduje důkladné statistické vyhodnocení velkého počtu experimentů.

Dynamická rozvozní úloha					
Počet míst	Reoptimalizace				Vkládací algoritmus
	Délka trasy		Doba výpočtu		Délka trasy
	OPT	1 min	LINGO	XPRESS	
5	256,50	-	0	0	256,50
10	353,03	-	14	5	353,03
15	582,76	589,13	1844	212	582,76

Tab. 6.9 – Experimenty s modely dynamické rozvozní úlohy

Pokud jde o dobu výpočtu, jehož prostřednictvím je nový zákazník zařazen mezi doposud nenavštívené zákazníky, je systém XPRESS MP podle očekávání výhodnější než LINGO i v případě tohoto typu rozvozních úloh. Při větším počtu zákazníků je opět možné po určité době přerušit re-optimalizační algoritmus a přijmout dosud nejlepší nalezené řešení. V tabulce je pro 15 míst uvedena délka trasy 589,13 km, která je získána po 1 minutě výpočtu v systému LINGO. Tato hodnota je velmi blízká optimální hodnotě 582,76 km.

6.5 Shrnutí výpočetních experimentů

Uvedené experimenty nejsou v žádném případě statistickou analýzou použití jednotlivých modelů a metod pro řešení okružních a rozvozních úloh. Jejich cílem je spíše demonstrovat možnosti chování složitějších algoritmů pro rozsáhlejší úlohy. Některé experimenty zcela jasně podporují oprávněnost používání heuristických metod pro většinu reálných aplikací, neboť optimalizační model není schopen v reálném čase poskytnout optimální řešení. Jak se ukazuje, i v případě rozsáhlých úloh lze však často použít optimalizační metody pro získání poměrně kvalitního řešení, přeruší-li řešitel výpočet na konci doby, kterou má pro určení trasy k dispozici. Takto získané řešení může být mj. i dobrým výchozím řešením pro některé heuristické algoritmy založené na zlepšování nalezeného řešení v krátkém čase.

V některých dynamických rozvozních úlohách je použití vkládacího algoritmu velice rozumným kompromisem z hlediska kvality získaného řešení a doby výpočtu. V praxi lze

kombinovat či střídat re-optimalizační postupy s vkládacím algoritmem či jinou heuristickou metodou podle délky doby, za kterou je nutné naplánovat upravenou trasu.

Z hlediska optimalizace je klíčovou otázkou výběr optimalizačního systému, resp. řešitele, který bude pro výpočet v praxi používán. V experimentech byl porovnán systém LINGO se systémem XPRESS MP, který prakticky ve všech experimentech poskytoval optimální řešení významně rychleji. I v tomto ohledu by bylo zapotřebí provést sofistikované statistické experimenty ať již na vygenerovaných nebo na reálných datech.

7. Závěr

Při zpracování práce Dynamické okružní a rozvozní úlohy se nabízelo několik základních přístupů. Prvním z nich je zmapování známých distribučních problémů a popis jejich dynamizace, a to především s důrazem na formulaci matematických modelů uvedených úloh. Další přístup k dané problematice se týká metod používaných k řešení distribučních úloh. Kromě tradičních optimalizačních postupů založených především na metodě větvení a mezi se v dnešní době uplatňují vysoce efektivní algoritmy¹, které mají význam zvláště u rozsáhlých praktických úloh. Navíc se stále používají rychlé heuristické a metaheuristické algoritmy poskytující ve velmi krátké době plně vyhovující řešení úlohy. Veškeré metody lze aplikovat nejen na statické úlohy, ale i na jejich dynamické rozšíření. Dalším přístupem, který byl při vypracování textu zvažován, byla konkrétní aplikace s reálnými daty. Velice zajímavou se také ukazovala možnost vyvinout v rámci práce efektivní softwarový produkt, který by umožnil řešit základní typy studovaných úloh.

Při rozhodování o způsobu zpracování zvoleného tématu hrála významnou roli skutečnost, že v českém jazyce dosud neexistuje ucelená publikace, která by se hlouběji zabývala problematikou dynamických okružních a rozvozních úloh. V zahraniční literatuře lze najít celou řadu textů týkajících se jak statických, tak dynamických distribučních úloh, většina z nich je však orientována na heuristické postupy typu tabu search či genetických algoritmů. Prostředí distribučních firem nabízí ohromnou škálu úloh a jejich dynamizací se tento prostor ještě více rozšiřuje. Proto bylo za hlavní cíl práce zvoleno zmapování základních typů okružních a rozvozních úloh s naznačením dalších možných aplikací.

Po metodické stránce se práce soustředí především na optimalizační matematické modely, které se mohou stát východiskem pro další výzkum týkající se uvedených úloh. Text má tudíž jistý význam i z hlediska výuky rozvozních úloh na vysokých školách, která se doposud soustředila především na aplikace ze statického pohledu, jenž je pro některé typy firem zcela nevhodný.

Veškeré příklady uvedené v disertační práci jsou založeny na vygenerovaných datech a slouží především pro demonstraci a snadnější pochopení textu. Při zkoumání matematických modelů úloh byla zvažována i možnost reálné aplikace z prostředí některé distribuční firmy.

¹ Metoda větvení a řezů, metoda větvení a oceňování, aj.

Vzhledem k zaměření práce byl tento záměr přesunut do některé z potenciálních diplomových prací, které budou brát tuto disertační práci jako jedno z teoretických východisek. V rámci ověřování funkčnosti matematických modelů uvedených v této práci byly vytvořeny softwarové nástroje, které lze poměrně snadno upravit pro potřeby praktických výpočtů či dalšího teoretického výzkumu. V závěru práce byly provedeny výpočetní experimenty, které dokládají význam heuristických metod v některých praktických aplikacích. Navíc ukazují na důležitost správné volby optimalizačních softwarových nástrojů pro potřeby firmy.

Vlastní přínos práce spatřuji především v následujících bodech:

1) Jak bylo výše uvedeno, jedná se o první text v českém jazyce takového rozsahu zabývající se dynamickým pohledem na okružní a rozvozní úlohy. V práci je podán podrobný přehled úloh tohoto typu. Parciální problémy, zpracované ve světové literatuře, se přitom často od sebe diametrálně liší nejen označením, ale i zavedenými předpoklady. Proto je z globálního pohledu na tento problém důležitá také určitá unifikace pojmů a jejich jasné vytýčení.

2) Matematické modely dynamických úloh v podobě re-optimalizačních algoritmů i heuristické vkládací metody jsou z větší části původní. Zvláštní pozornost zasluhuje úloha kurýrní služby s jejím dynamickým rozšířením v části 3.4. Úloha obchodního cestujícího s časovými okny je rozšířena o možnost čekání u právě obsluženého zákazníka či u zákazníka, k němuž vozidlo přijíždí před otevřením jeho časového okna (části 3.3.1 a 3.3.2). Velmi zajímavé jsou okružní a rozvozní problémy, v nichž je cílem minimalizovat čas potřebný k obsluze všech zákazníků. Vlastní definicí matematických modelů těchto úloh se otevřel široký prostor pro další výzkum ve spolupráci s reálnými distribučními firmami. Nezanedbatelným přínosem je vlastní definice matematických modelů dynamických úloh s několika vozidly umístěnými v jednom či více výchozích místech. Dynamický pohled významně posouvá reálné využití matematického modelování při reálném rozhodování dispečera distribuční firmy.

3) Veškeré typy úloh jsou prezentovány na názorných demonstračních příkladech, což umožňuje čtenáři velmi pohodlně a příjemně proniknout do občas poněkud složitějších úvah. I z tohoto důvodu byla připojena schémata algoritmů v částech 4.6 a 5.1. Velká pozornost je věnována heuristickým výpočetním postupům, z nichž vkládací metody jsou dobrým základem pro manažerské rozhodnutí při změně okružní, resp. rozvozní trasy ve velmi krátkém čase.

4) Uvědomuji si, že bylo možné se podrobněji věnovat výpočetním experimentům; vzhledem k zaměření práce jsem však tuto část odsunul do oblasti své další vědecké práce, a to především z důvodu možnosti porovnání přesných optimalizačních metod s heuristickými a metaheuristickými algoritmy, nejlépe pak na konkrétních reálných datech. Jsem přesvědčený, že tyto experimenty potvrdí vhodnost využití optimalizačních a heuristických přístupů v různých rozhodovacích procesech.

5) Vzhledem k vytýčeným cílům a přínosům považuji tuto práci za text, který se může stát výchozí literaturou pro výuku dané problematiky na vysokých školách či fakultách dopravního či ekonomického zaměření ve specializovaných kursech diskrétních modelů či logistického managementu.

V samotné práci je naznačena celá řada konkrétních kroků či směrů, kterými by se mohly ubírat další aktivity týkající se problematiky dynamických distribučních úloh. Lze je shrnout do následujících bodů:

1) Zmapování metod pro řešení statických okružních a rozvozních úloh a jejich modifikace pro dynamické úlohy – především metoda větvení a řezů, metoda větvení a oceňování, heuristické a metaheuristické algoritmy.

2) Analýza využívání optimalizačních metod u firem zabývajících se rozvozem či svozem zboží, materiálu, lidí, pošty, odpadu apod. Formulace konkrétních úloh v těchto firmách.

3) Řešení reálné dynamické okružní či rozvozní úlohy v některé z distribučních firem.

4) Vytvoření kompaktního softwarového modulu pro řešení základních statických a dynamických okružních a rozvozních úloh pro účely výukového procesu.

5) Provedení simulačních experimentů s modely týkajícími se okružních a rozvozních úloh s apriorní informací. Výzkum vlivu velikosti rezervy v nákladu vozidla u dynamických rozvozních úloh na finanční náklady firmy.

6) Použití optimalizačních systémů pro řešení dynamických okružních a rozvozních úloh. Jedná se především o produkty softwarových firem Dash Optimization Inc., ILOG Inc., Paragon Decision Technology Inc., aj.

7) Prohloubení a statistické vyhodnocení výpočetních experimentů prezentovaných či naznačených v této práci.

Přílohy

- Příloha č. 1: Úloha obchodního cestujícího
- Příloha č. 2: Dynamická úloha obchodního cestujícího
- Příloha č. 3: Popis systému TSP. xls
- Příloha č. 4: Statická úloha s více obchodními cestujícími v jednom výchozím místě (minimalizace celkové vzdálenosti)
- Příloha č. 5: Statická úloha s více obchodními cestujícími ve více výchozích místech (minimalizace celkové vzdálenosti)
- Příloha č. 6: Statická úloha s více obchodními cestujícími v jednom výchozím místě (minimalizace času potřebného k obloužení všech zákazníků)
- Příloha č. 7: Statická úloha s více obchodními cestujícími v několika výchozích místech (minimalizace času potřebného k obloužení všech zákazníků)
- Příloha č. 8: Dynamická úloha s více obchodními cestujícími v několika výchozích místech (minimalizace celkové vzdálenosti)
- Příloha č. 9: Rozvozní úloha s jedním vozidlem
- Příloha č. 10: Rozvozní úloha s dělenou dodávkou
- Příloha č. 11: Obsah přiloženého CD

Příloha č. 1

Úloha obchodního cestujícího

Statický model úlohy obchodního cestujícího v programu LINGO (příklad 3.1)

Zápis matematického modelu v optimalizačním systému LINGO:

```
MODEL:
SETS:
  Mesto:u;
  Prirazeni (Mesto,Mesto) :KM,X;
ENDSETS
DATA:
  N=6;
  Mesto,KM=@OLE('TSP6.xls');
  @OLE('TSP6.xls','Vysledek')=X;
ENDDATA
  MIN=@SUM(Prirazeni:KM*X);
@FOR(Mesto(I):X(I,I)=0);
@FOR(Mesto(I):@SUM(Mesto(J):X(I,J))=1);
@FOR(Mesto(J):@SUM(Mesto(I):X(I,J))=1);
U(1)=0;
@FOR(Mesto(I):@FOR(Mesto(J)|J#NE#1:U(I)-U(J)+1-N*(1-X(I,J))<=0));
FOR(Prirazeni:@BIN(X));
END
```

Lokalizace souborů na CD: **Kapitola 3 / TSP6.lg4**

Kapitola 3 / TSP6.xls

Příloha č. 2

Dynamická úloha obchodního cestujícího

Re-optimalizace trasy (příklad 3.1) po přijetí požadavku zákazníka č. 7

Zápis matematického modelu v systému LINGO:

```
MODEL:
SETS:
  Mesto:u;
  Prirazeni (Mesto,Mesto) :KM,X;
ENDSETS
DATA:
  UN=4;
  Mesto,KM=@OLE('TSP7.xls');
  @OLE('TSP7.xls','Vysledek')=X;
ENDDATA
  MIN=@SUM(Prirazeni:KM*X);
@FOR(Mesto(I):X(I,I)=0);
@FOR(Mesto(I)|I#NE#1:@SUM(Mesto(J)|J#NE#3:X(I,J)=1));
@FOR(Mesto(J)|J#NE#3:@SUM(Mesto(I):X(I,J)=1));
U(3)=0;
@FOR(Mesto(I):@FOR(Mesto(J)|J#NE#3 #and# J#NE#I:
  U(I)-U(J)+1-UN*(1-(I,J))<=0);
FOR(Prirazeni:@BIN(X));
END
```

Lokalizace souborů na CD: **Kapitola 3 / TSP7.lg4**

Kapitola 3 / TSP7.xls

Příloha č. 3

Popis systému TSP. xls

Tento systém, který byl vytvořen v tabulkovém procesoru **MS Excel**, integruje dynamické modely úlohy obchodního cestujícího, jež byly uvedeny v kapitole 3. Při jeho návrhu bylo využito prostředí **VBA v MS Excel** a možnosti propojení **MS Excel** s optimalizačním systémem **LINGO**. Výpočty byly provedeny na základě vygenerovaných údajů. Systém lze však poměrně jednoduše upravit pro potřeby konkrétní firmy na základě jejich požadavků tak, aby dispečer mohl snadno aktualizovat známé údaje a na základě nich pak provádět kalkulace, které povedou k efektivnímu rozhodování.

Hlavní list s názvem **DTSP** dává uživateli možnost zvolit počet zákazníků známých před zahájením trasy a celkový počet zákazníků¹.

Dynamická úloha obchodního cestujícího

Počet známých zákazníků	5	Generování zákazníků
Celkový počet zákazníků	10	

Optimální trasa pro známé zákazníky

Délka trasy statické úlohy	216,90	Optimalizace	Trasa
----------------------------	--------	--------------	-------

Přidávání zákazníků: vkládací metoda

Délka trasy dynamické úlohy	361,66	Postupně	Vše
-----------------------------	--------	----------	-----

Přidávání zákazníků: re-optimalizace

Délka trasy dynamické úlohy	319,45	Postupně	Vše
-----------------------------	--------	----------	-----

Obr. P3.1 - List *DTSP*

Použitím tlačítka **Generování zákazníků** se vygeneruje celkem 50 míst², a sice jejich souřadnice $[X,Y]$ a časový interval mezi požadavkem daného zákazníka a požadavkem bezprostředně předchozího zákazníka. Pro souřadnice bylo použito diskrétní rovnoměrné

¹ Tento počet zahrnuje i výchozí místo.

² Z vygenerovaných 50 míst se pro danou úlohu vyberou zákazníci podle počtu známých zákazníků, resp. celkového počtu zákazníků zadaného uživatelem.

rozdělení na intervalu $\langle 0,100 \rangle^3$, pro dobu mezi vznikem dvou po sobě následujících požadavků exponenciální rozdělení se střední hodnotou 20 minut. Seznam všech vygenerovaných zákazníků je uveden v listu **SEZNAM**. V listu **VZDALENOSTI** jsou pak vypočítány euklidovské vzdálenosti mezi všemi 50 místy.

Podle algoritmu uvedeného v části 3.2.3 je nejprve nutné určit optimální trasu pro zákazníky, jejichž požadavky jsou známy, tj. vyřešit statickou úlohu obchodního cestujícího. Optimalizační systém **LINGO** umožňuje spustit výpočet přímo z prostředí **MS Excel**. Kód modelu musí být zapsán v samostatném listu, v našem případě s názvem **MODEL TSP**. Po kliknutí na tlačítko **Optimalizace** v listu **DTSP** je nalezeno optimální řešení statické úlohy, které je uvedeno v listu **VYSLEDEK TSP**.

Z praktického hlediska je užitečné připravit seznam zákazníků v pořadí, ve kterém mají být vozidlem navštíveni. Tento seznam, vytvořený pomocí tlačítka **Trasa**, je uveden v listu **TRASATSP**, kde lze kromě čísla zákazníka a jeho souřadnic nalézt také průběžnou délku trasy, resp. čas (v minutách), v němž bude zákazník vozidlem navštíven⁴. Předpokládáme, že vozidlo opustí výchozí místo v čase 0

Zákazník	X	Y	Délka
1	69	56	0,00
5	34	56	35,00
2	4	58	65,07
4	24	7	119,85
3	71	6	166,86
1	69	56	216,90

Tab. P3.1 - Optimální trasa

Na základě těchto údajů je znázorněna výsledná trasa i graficky (Obr. P3.2).

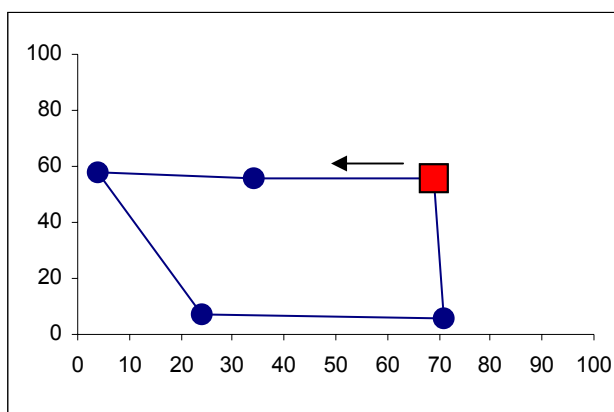
Z uvedených příkladů lze vyčíst, že optimální trasa povede postupně přes zákazníky č. 5, 2, 4 a 3 a její délka bude cca 216,90 km, resp. bude trvat cca 216,90 minut. Tato hodnota je také uvedena v listu **DTSP** jako **Délka trasy statické úlohy**.

Jakmile je určena optimální trasa pro všechny předem známé zákazníky, dispečer dá řidiči vozidla pokyn, v jakém pořadí má tyto zákazníky navštívit. Zároveň je připraven přijímat další požadavky od nových zákazníků. Časové intervaly mezi příchody požadavků jsou

³ Hodnoty jsou ve všech úlohách v km.

⁴ Rychlost vozidla, podle předpokladu zavedeného v kapitole 3, činí 60 km.h⁻¹.

uvedeny v listu **SEZNAM**. Hodnota u prvního nového zákazníka (v našem příkladě označeného č. 6) znamená interval od výjezdu vozidla na trasu.



Obr. P3.2 – Grafické znázornění optimální trasy statické úlohy

Nejprve využijeme vkládací algoritmus, který je uveden v části 3.2.2. Přejde-li nový požadavek, zařadí se do trasy mezi dva doposud nenavštívené zákazníky, pro něž bude prodloužení původní trasy minimální. Přitom je zapotřebí kontrolovat, aby jako první zákazník na nové trase byl ten, ke kterému vozidlo právě směřuje v okamžiku příchodu nového požadavku.

V průběhu aktualizace trasy podle příchodu dalších požadavků může dojít ke dvěma krajním situacím:

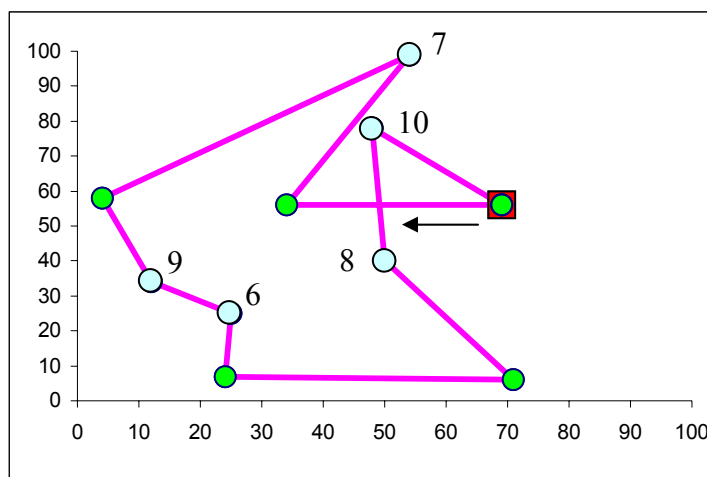
1. Po vložení nového požadavku by došlo k překročení pracovní doby. Takový požadavek je pak odmítnut a dispečer čeká na požadavek dalšího zákazníka. Pro praktické účely lze toto omezení samozřejmě libovolně měnit, např. zavedením směnného provozu apod.
2. Nový požadavek přijde až po dokončení trasy vozidlem, tj. příjezdu vozidla zpět do výchozího místa. V takovém případě se začne vytvářet nová trasa, případně více tras, které vozidlo postupně absolvuje.

System založený na simulaci příchodů nových požadavků umožňuje uživateli dvojitý způsob vkládání vygenerovaných zákazníků. Obě možnosti jsou umístěny v listu **DTSP** pod označením **Přidávání zákazníků: vkládací metoda**. Použitím tlačítka **Vše** lze vytvořit trasu pro všechny nové zákazníky ve zrychleném čase. Výsledná trasa je pak uvedena v tabulce

v listu *TRASADTSP*, stejně jako graf znázorňující tuto trasu. Délka trasy má hodnotu cca 361,66 km, resp. doba jejího trvání je 361,66 minut.

Zákazník	X	Y	Délka
1	69	56	0,00
5	34	56	35,00
7	54	99	82,42
2	4	58	147,08
9	12	34	172,38
6	25	25	188,19
4	24	7	206,22
3	71	6	253,23
8	50	40	293,19
10	48	78	331,25
1	69	56	361,66

Tab. P3.2 - Seznam všech zákazníků včetně okamžiku obsluhy



Obr. P3.3 - Grafické znázornění trasy po přidání 5 zákazníků vkládacím algoritmem

Druhou možností je použití tlačítka *Postupně*, které přesune ovládání algoritmu přímo do listu *TRASADTSP*, kde uživatel po zvolení tlačítka *Přidat zákazníka* přidává jednotlivé zákazníky postupně a má možnost průběžně sledovat aktualizaci trasy, a to jak v tabulce, tak i v grafu. Barevně je odlišena již dokončená část trasy, tj. již obslužení zákazníci, část trasy, ve které se právě nachází vozidlo a zbývající část trasy, kterou je možné ještě měnit. Tento přístup je vhodný pro skutečnou práci dispečera; systém je zapotřebí pouze doplnit o vstupní formulář, ve kterém by dispečer zadával polohu zákazníka, případně modul, který by na základě údajů o zákazníkovi jeho polohu vypočítal⁵. Výsledná délka trasy, která je za uvedených předpokladů současně dobou, během níž vozidlo obslouží všechny zákazníky

⁵ Práce na této části systému přesahuje vymezený rámec této práce, proto byl vstup údajů nahrazen vygenerovanými hodnotami.

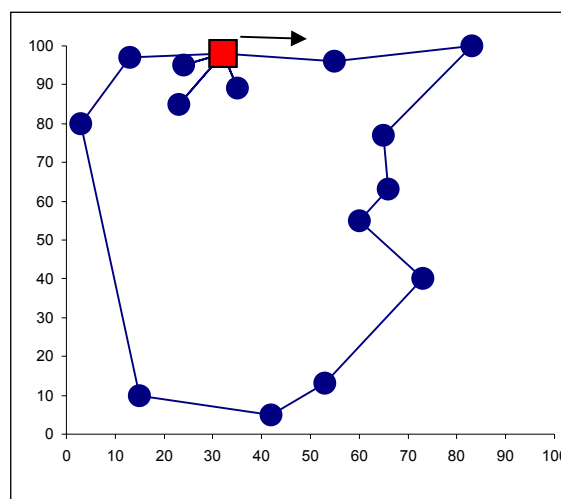
Příloha č. 4

Statická úloha s více obchodními cestujícími v jednom výchozím místě

Minimalizace celkové vzdálenosti

Následující úloha obsahuje 14 vygenerovaných zákazníků, k jejichž obsluze jsou ve výchozím místě označeném č. 1 připravena 4 vozidla. V tabulce P4.1 jsou uvedeny optimální trasy pro každé vozidlo. Jak je patrné i z následujícího grafu na obr. P4.1, 1. vozidlo navštíví zákazníka č. 4 a vrátí se zpět do výchozího místa, podobně jako 3. vozidlo a 4. vozidlo navštíví zákazníky č. 10 a 11. Oproti tomu 2. vozidlo navštíví všechny ostatní zákazníky.

Zákazník	X	Y	Délka
1	32	98	0,00
4	24	95	8,54
1	32	98	17,09
1	32	98	0,00
9	55	96	23,09
13	83	100	51,37
7	65	77	80,58
14	66	63	94,61
6	60	55	104,61
5	73	40	124,46
2	53	13	158,06
15	42	5	171,66
8	15	10	199,12
3	3	80	270,14
12	13	97	289,87
1	32	98	308,89
1	32	98	0,00
10	35	89	9,49
1	32	98	18,97
1	32	98	0,00
11	23	85	15,81
1	32	98	31,62



Tab. P4.1 – Optimální trasy 4 vozidel

Obr. P4.1 – Optimální trasy 4 vozidel

Celková vzdálenost, která odpovídá uvedenému optimálnímu řešení, je

$$T = 17,09 + 308,89 + 18,97 + 31,62 = 376,57 \text{ km.}$$

Při řešení úlohy bylo použito optimalizačního systému LINGO a následujícího zápisu matematického modelu:

```

MODEL:
SETS:
  Mesto:u;
  Prirazeni (Mesto,Mesto):KM,X;
ENDSETS
DATA:
  Mesto,KM,K=@OLE('MTSP.xls');
  @OLE('MTSP.xls','Vysledek')=X;
ENDDATA
N=@SIZE(Mesto);
[T]MIN=@SUM(Prirazeni:KM*X);
@FOR(Mesto(I):X(I,I)=0);
@FOR(Mesto(I)|I#NE#1:@SUM(Mesto(J):X(I,J))=1);
@FOR(Mesto(J)|J#NE#1:@SUM(Mesto(I):X(I,J))=1);
@SUM(Mesto(J)|J#NE#1:X(1,J))=K;
@SUM(Mesto(J)|J#NE#1:X(J,1))=K;
@SUM(Mesto(J)|J#NE#1:X(J,1))-@SUM(Mesto(J)|J#NE#1:X(1,J))=0;
U(1)=0;
@FOR(Mesto(I):@FOR(Mesto(J)|J#NE#1:U(I)-U(J)+1-(N-K)*(1-X(I,J))<=0));
@FOR(Prirazeni:@BIN(X));
DATA:
  @OLE('MTSP.xls','T')=T;
ENDDATA
END

```

Lokalizace souboru na CD: **Kapitola 4 / Část 4.3.1 / MTSP.xls**

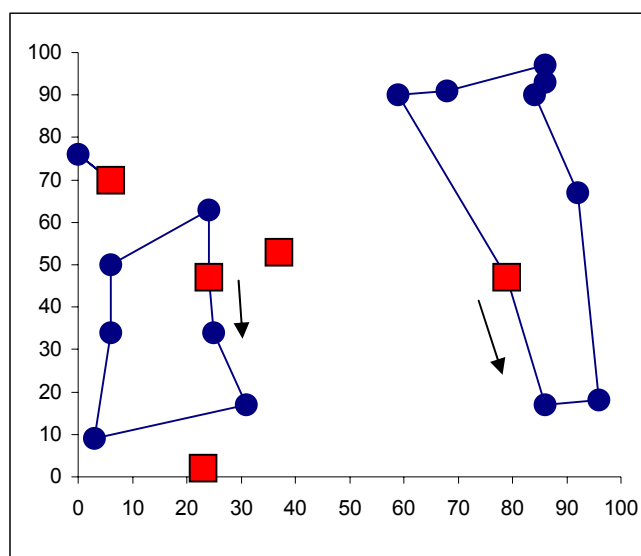
Příloha č. 5

Statická úloha s více obchodními cestujícími ve více výchozích místech

Minimalizace celkové vzdálenosti

Bylo vygenerováno 5 výchozích míst a 15 zákazníků. Pokud je cílem minimalizovat celkovou vzdálenost ujetou všemi vozidly, pak optimálním řešením je rozvrh, kdy 2. a 3. vozidlo vůbec nevyjíždí (tabulka P5.1 a obr. P5.1).

Zákazník	X	Y	Délka
1	79	47	0,00
8	86	17	30,81
12	96	18	40,86
17	92	67	90,02
6	84	90	114,37
11	86	93	117,98
13	86	97	121,98
16	68	91	140,95
20	59	90	150,00
1	79	47	197,43
2	23	2	0,00
3	37	53	0,00
4	6	70	0,00
10	0	76	8,49
4	6	70	16,97
5	24	47	0,00
7	25	34	13,04
9	31	17	31,07
15	3	9	60,19
14	6	34	85,37
19	6	50	101,37
18	24	63	123,57
5	24	47	139,57



Tab. P5.1 – Optimální trasy 5 vozidel

Obr. P5.1 – Optimální trasy 5 vozidel

Hodnota minimální vzdálenosti, kterou dohromady ujedou všechna tři vozidla, je

$$T = 197,43 + 16,97 + 139,57 = 353,97 \text{ km.}$$

Zápis modelu v systému LINGO:

```

MODEL:
SETS:
  Mesto:u;
  Prirazeni (Mesto,Mesto):KM,X;
ENDSETS
DATA:
  Mesto,KM,K=@OLE('MTSP.xls');
  @OLE('MTSP.xls','Vysledek')=X;
ENDDATA
N=@SIZE(Mesto)-K;
[T] MIN=@SUM(Prirazeni:KM*X);
@FOR(Mesto(J)|J#GE#K+1:@SUM(Mesto(I):X(I,J))=1);
@FOR(Mesto(J)|J#GE#K+1:@SUM(Mesto(I):X(J,I))=1);
@FOR(Mesto(I)|I#LE#K:@SUM(Mesto(J)|J#GE#K+1:X(I,J))<=1);
@FOR(Mesto(I)|I#LE#K:@SUM(Mesto(J)|J#GE#K+1:X(I,J))-
@SUM(Mesto(J)|J#GE#K+1:X(J,I))=0);
@FOR(Mesto(I)|I#GE#K+1:@FOR(Mesto(J)|J#GE#K+1#AND#I#NE#J:
  U(I)-U(J)+1-N*(1-X(I,J))<=0);
@FOR(Mesto(I)|I#LE#K:@FOR(Mesto(J)|J#LE#K:X(I,J)=0));
@FOR(Mesto(I)|I#GE#K+1:X(I,I)=0);
U(1)=0;
@FOR(Prirazeni:@BIN(X));
DATA:
  @OLE('MTSP.xls','T')=T;
ENDDATA
END

```

Lokalizace souboru na CD: **Kapitola 4 / Části 4.3.2 a 4.6.1 / MTSP.xls**

Příloha č. 6

Statická úloha s více obchodními cestujícími v jednom výchozím místě

Minimalizace času potřebného k obslužení všech zákazníků

V úloze uvažujeme 7 vygenerovaných zákazníků a 1 výchozí místo se 3 vozidly. Cílem je minimalizovat čas potřebný k obsluze všech zákazníků s tím, že se do tohoto času započítává i návrat vozidel do výchozího místa. Nejprve aplikujeme model (4.20) – (4.28), v němž je jako účelová funkce zvolena doba, ve které budou dokončeny všechny trasy. Tato doba je minimalizována. Zde je model přepsaný do kódu LINGO.

```

MODEL:
DATA:
  K=@OLE('MTSP.xls');
ENDDATA
SETS:
  Mesto:u;
  Vozidlo /1..K/;
  Vzdalenost(Mesto,Mesto):KM;
  Trasa(Vzdalenost,Vozidlo):X;
ENDSETS
DATA:
  Mesto,KM=@OLE('MTSP.xls');
  @OLE('MTSP.xls','Vysledek')=X;
ENDDATA
N=@SIZE(Mesto);
[T] MIN=TT;
@FOR(Vozidlo(KK):@SUM(Vzdalenost(I,J):KM(I,J)*X(I,J,KK))<=TT);
@FOR(Mesto(I)|I#GT#1:@SUM(Vozidlo(KK):@SUM(Mesto(J):X(I,J,KK)))=1);
@FOR(Mesto(J)|J#GT#1:@SUM(Vozidlo(KK):@SUM(Mesto(I):X(I,J,KK)))=1);
@FOR(Vozidlo(KK):@SUM(Mesto(J)|J#GT#1:X(I,J,KK))<=1);
@FOR(Vozidlo(KK):@FOR(Mesto(J)|J#GT#1:@SUM(Mesto(I):X(I,J,KK))-
  @SUM(Mesto(I):X(J,I,KK))=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I):@FOR(Mesto(J)|J#NE#1 #AND# J#NE#I:U(I)-
  U(J)+1-N*(1-X(I,J,KK))<=0));
@FOR(Vozidlo(KK):@FOR(Vzdalenost(I,I):X(I,I,KK)=0));
@FOR(Trasa:@BIN(X));
DATA:
  @OLE('MTSP.xls','T')=T;
ENDDATA
END

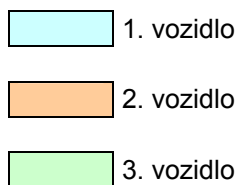
```

Výsledkem je rozvržení obsluhy zákazníků do tří následujících tras (tabulka P6.1):

- 1. vozidlo:** 1-3-1 s celkovou dobou 147,36 min,
- 2. vozidlo:** 1-2-4-8-1 s celkovou dobou 151,74 min,
- 3. vozidlo:** 1-7-5-6-1 s celkovou dobou 130,20 min.

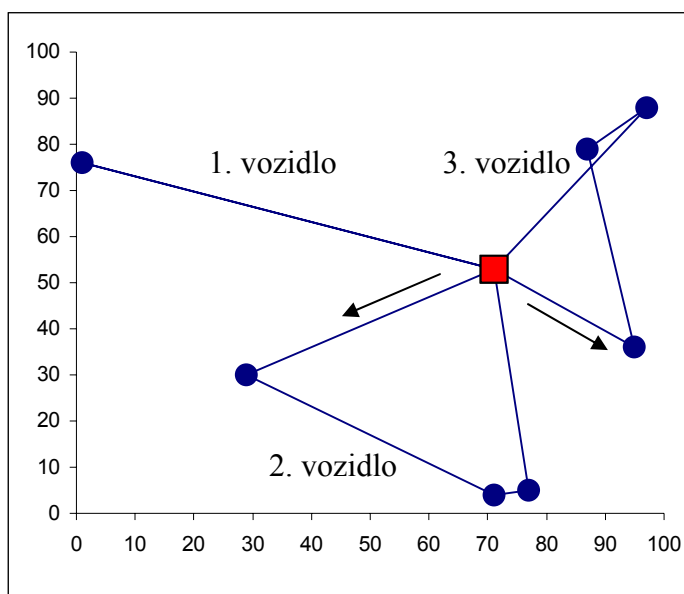
Optimální hodnota účelové funkce je dána délkou trasy druhého vozidla, tedy 151,74 min.

Zákazník	X	Y	Čas
1	71	53	0,00
3	1	76	73,68
1	71	53	147,36
1	71	53	0,00
2	29	30	47,89
4	71	4	97,28
8	77	5	103,36
1	71	53	151,74
1	71	53	0,00
7	95	36	29,41
5	87	79	73,15
6	97	88	86,60
1	71	53	130,20



Tab. P6.1 – Minimalizace času potřebného k obslužení 7 zákazníků 3 vozidly

Graficky je optimální rozvržení tras znázorněno na obr. P6.1



Obr. P6.1 – Optimální trasy 3 vozidel

Z uvedeného grafu je patrné, že 3. vozidlo nejede po optimální trase, tj. dodatečně lze zefektivnit obsluhu zákazníků, čili zkrátit délku tras jednotlivých vozidel.

V úloze byla provedena dodatečná optimalizace dílčích tras tak, aby optimální řešení původní úlohy zůstalo zachováno. Délky jednotlivých tras jsou pak optimalizovány pomocí modelu (4.20) – (4.28), v němž je použito omezení (4.21a) ve tvaru

$$\sum_{i=1}^n \sum_{j=1}^n t_{ij} x_{ij}^k \leq 151,74, \quad k = 1, 2, \dots, K$$

a účelová funkce (4.20a)⁶.

V následujícím kódu modelu zapsaného v systému LINGO představuje vstupující hodnota T hodnotu pravých stran výše uvedených omezení, hodnota $TMIN$ je pak součtem délek všech tras.

```

MODEL
DATA:
    K=@OLE('MTSP.xls');
ENDDATA
SETS:
    Mesto:u;
    Vozidlo /1..K/;
    Vzdalenost(Mesto,Mesto):KM;
    Trasa(Vzdalenost,Vozidlo):X;
ENDSETS
DATA:
    Mesto,KM,T=@OLE('MTSP.xls');
    @OLE('MTSP.xls','Vysledeko')=X;
ENDDATA
N=@SIZE(Mesto);
[TMIN] MIN=@SUM(Vozidlo(KK):@SUM(Vzdalenost(I,J):KM(I,J)*X(I,J,KK)));
@FOR(Vozidlo(KK):@SUM(Vzdalenost(I,J):KM(I,J)*X(I,J,KK))<=T);
@FOR(Mesto(I)|I#GT#1:@SUM(Vozidlo(KK):@SUM(Mesto(J):X(I,J,KK)))=1);
@FOR(Mesto(J)|J#GT#1:@SUM(Vozidlo(KK):@SUM(Mesto(I):X(I,J,KK)))=1);
@FOR(Vozidlo(KK):@SUM(Mesto(J)|J#GT#1:X(1,J,KK))<=1);
@FOR(Vozidlo(KK):@FOR(Mesto(J)|J#GT#1:@SUM(Mesto(I):X(I,J,KK))-
    @SUM(Mesto(I):X(J,I,KK))=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I):@FOR(Mesto(J)|J#NE#1 #AND# J#NE#I:U(I)-
    U(J)+1-N*(1-X(I,J,KK))<=0));
@FOR(Vozidlo(KK):@FOR(Vzdalenost(I,I):X(I,I,KK)=0));
@FOR(Trasa:@BIN(X));
DATA:
    @OLE('MTSP.xls','TMIN')=TMIN;
ENDDATA
END

```

⁶ Hodnota pravé strany omezujících podmínek odpovídá nejdelší trase v předešlé optimalizaci, je tedy převzata přímo ze systému LINGO, resp. MS Excel, tedy v nezaokrouhleném tvaru, resp. tvaru zaokrouhleném na odpovídající počet desetinných míst; zde je pro přehlednost provedeno zaokrouhlení pouze na 2 desetinná místa.

Na základě dodatečné optimalizace dojde ke zkrácení trasy 3. vozidla díky záměně pořadí zákazníků označených č. 5 a 6:

1. vozidlo: 1-3-1 s celkovou dobou 147,36 min,
2. vozidlo: 1-2-4-8-1 s celkovou dobou 151,74 min,
3. vozidlo: 1-7-6-5-1 s celkovou dobou 125,43 min.

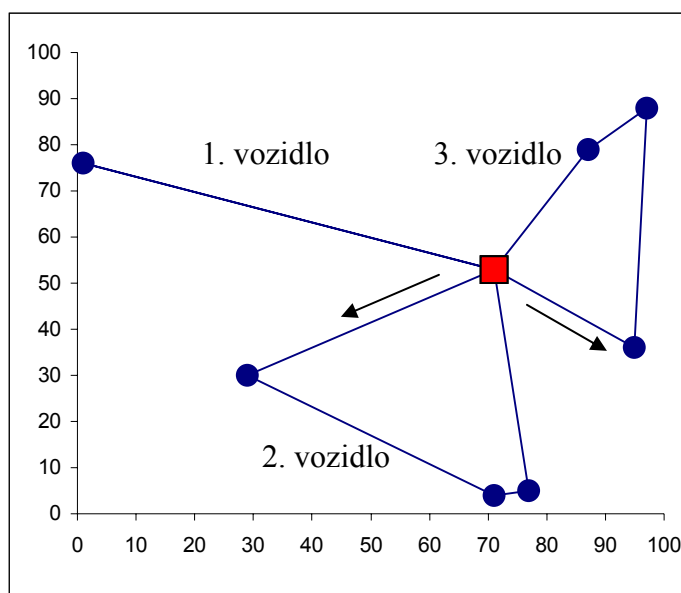
Zákazník	X	Y	Čas
1	71	53	0,00
3	1	76	73,68
1	71	53	147,36
1	71	53	0,00
2	29	30	47,89
4	71	4	97,28
8	77	5	103,36
1	71	53	151,74
1	71	53	0,00
7	95	36	29,41
6	97	88	81,45
5	87	79	94,90
1	71	53	125,43

1. vozidlo

2. vozidlo

3. vozidlo

Tab. P6.2 – Dodatečná optimalizace dílčích tras



Obr. P6.2 – Zkrácení trasy 3. vozidla pomocí dodatečné optimalizace

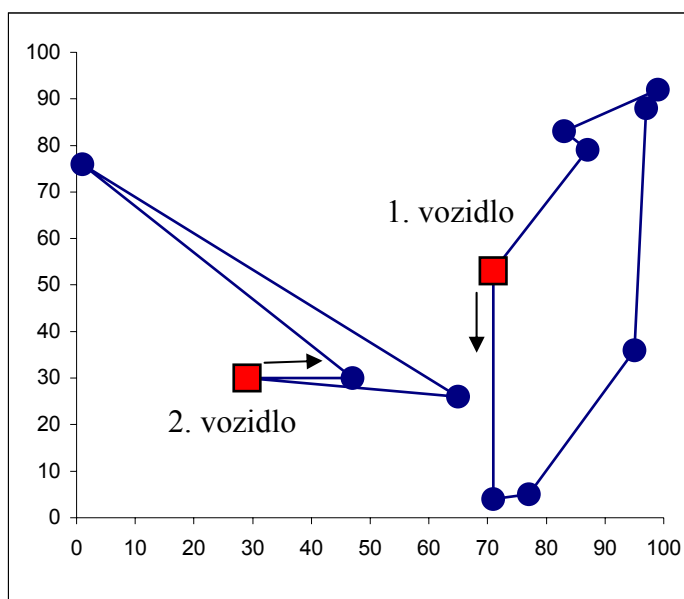
Lokalizace souboru na CD: **Kapitola 4 / Část 4.4.1 / MTSP.xls**

Příloha č. 7

Statická úloha s více obchodními cestujícími v několika výchozích místech

Minimalizace času potřebného k obslužení všech zákazníků

V následující úloze jsou definována 2 výchozí místa se 2 vozidly určenými k obsluze 10 zákazníků (obr. P7.1). Minimální čas⁷ potřebný k jejich obsluze činí 201,98 min, což je doba, po kterou bude na trase 1. vozidlo; délka trasy 2. vozidla pak bude 200,49 min (tabulka P7.1).



Obr. P7.1 – Optimální trasy 2 vozidel v různých výchozích místech

Zákazník	X	Y	Čas
1	71	53	0,00
4	71	4	49,00
8	77	5	55,08
7	95	36	90,93
6	97	88	142,97
12	99	92	147,44
11	83	83	165,80
5	87	79	171,45
1	71	53	201,98

Zákazník	X	Y	Čas
2	29	30	0,00
9	47	30	18,00
3	1	76	83,05
10	65	26	164,27
2	29	30	200,49

1. vozidlo

2. vozidlo

Tab. P7.1 – Optimální trasy 2 vozidel

⁷ Do této doby je započítána i doba návratu vozidla do výchozího místa.

Uvedený výsledek byl získán na základě optimalizace, při které byl použitý matematický model (4.29) – (4.39) formulovaný v části 4.4.2, jehož zápis v systému LINGO je následující:

```


MODEL:
DATA:
  K=@OLE('MTSP.xls');
ENDDATA
SETS:
  Mesto:u;
  Vozidlo /1..K/;
  Vzdalenost(Mesto,Mesto):KM;
  Trasa(Vzdalenost,Vozidlo):X;
ENDSETS
DATA:
  Mesto,KM=@OLE('MTSP.xls');
  @OLE('MTSP.xls','Vysledek')=X;
ENDDATA
N=@SIZE(Mesto)-K;
[T] MIN=TT;
@FOR(Vozidlo(KK):@SUM(Vzdalenost(I,J):KM(I,J)*X(I,J,KK))<=TT);
@FOR(Mesto(I)|I#GE#K+1:@SUM(Vozidlo(KK):@SUM(Mesto(J):X(I,J,KK)))=1);
@FOR(Mesto(J)|J#GE#K+1:@SUM(Vozidlo(KK):@SUM(Mesto(I):X(I,J,KK)))=1);
@FOR(Vozidlo(KK):@SUM(Mesto(J)|J#GE#K+1:X(KK,J,KK))<=1);
@FOR(Vozidlo(KK):@FOR(Mesto(J):@SUM(Mesto(I):X(I,J,KK))-
  @SUM(Mesto(I):X(J,I,KK))=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GE#K+1:@FOR(Mesto(J)|J#NE#I #AND#
  J#GE#K+1:U(I)-U(J)+1-N*(1-X(I,J,KK))<=0));
@FOR(Vozidlo(KK):@FOR(Vzdalenost(I,J)|I#LE#K+1 #AND#
  J#LE#K+1:X(I,I,KK)=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#LE#K #AND#
  I#NE#K:@FOR(Mesto(J)|J#GE#K+1:X(I,J,KK)=0));
@FOR(Vozidlo(KK):@FOR(Vzdalenost(I,I)|I#GE#K+1:X(I,I,KK)=0));
@FOR(Trasa:@BIN(X));
DATA:
  @OLE('MTSP.xls','T')=T;
ENDDATA
END

```

Provedením dodatečné optimalizace trasy 2. vozidla dojde k jejímu poměrně výraznému zkrácení na 171,51 min (tabulka P7.2 a obr. P7.2).

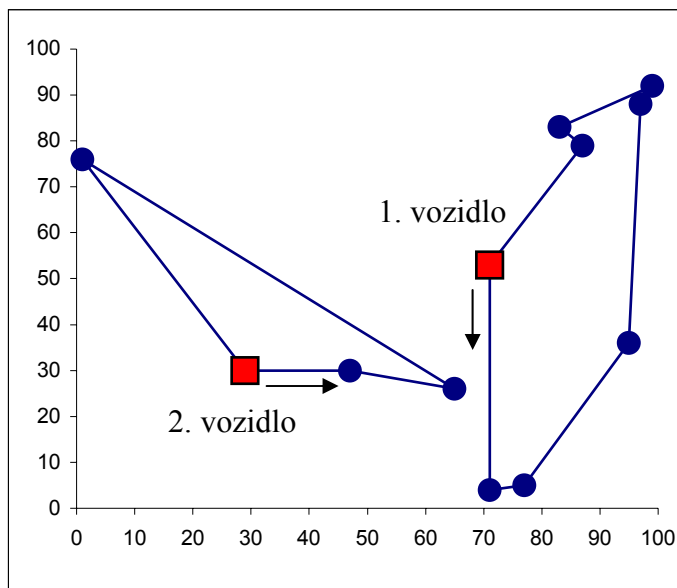
Zákazník	X	Y	Čas
1	71	53	0,00
4	71	4	49,00
8	77	5	55,08
7	95	36	90,93
6	97	88	142,97
12	99	92	147,44
11	83	83	165,80
5	87	79	171,45
1	71	53	201,98

Zákazník	X	Y	Čas
2	29	30	0,00
9	47	30	18,00
10	65	26	36,44
3	1	76	117,65
2	29	30	171,51

 1. vozidlo

 2. vozidlo

Tab. P7.2 – Dodatečná optimalizace trasy 2. vozidla



Obr. P7.2 – Dodatečná optimalizace trasy 2. vozidla

Pro úplnost ještě uveďme model zapsaný v LINGO:

```

MODEL:
DATA:
    K=@OLE('MTSP.xls');
ENDDATA
SETS:
    Mesto:u;
    Vozidlo /1..K/;
    Vzdalenost(Mesto,Mesto):KM;
    Trasa(Vzdalenost,Vozidlo):X;
ENDSETS
DATA:
    Mesto,KM,T=@OLE('MTSP.xls');
    @OLE('MTSP.xls','Vysledeko')=X;
ENDDATA
N=@SIZE(Mesto)-K;
[TMIN] MIN=@SUM(Vozidlo(KK):@SUM(Vzdalenost(I,J):KM(I,J)*X(I,J,KK)));
@FOR(Vozidlo(KK):@SUM(Vzdalenost(I,J):KM(I,J)*X(I,J,KK))<=T);
@FOR(Mesto(I)|I#GE#K+1:@SUM(Vozidlo(KK):@SUM(Mesto(J):X(I,J,KK)))=1);
@FOR(Mesto(J)|J#GE#K+1:@SUM(Vozidlo(KK):@SUM(Mesto(I):X(I,J,KK)))=1);
@FOR(Vozidlo(KK):@SUM(Mesto(J)|J#GE#K+1:X(KK,J,KK))<=1);
@FOR(Vozidlo(KK):@FOR(Mesto(J):@SUM(Mesto(I):X(I,J,KK))-
    @SUM(Mesto(I):X(J,I,KK))=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GE#K+1:@FOR(Mesto(J)|J#GE#K+1 #AND#
    J#NE#I:U(I)-U(J)+1-N*(1-X(I,J,KK))<=0));
@FOR(Vozidlo(KK):@FOR(Vzdalenost(I,J)|I#LE#K+1 #AND#
    J#LE#K+1:X(I,I,KK)=0));
@FOR(Vozidlo(KK):@FOR(Vzdalenost(I,I)|I#GE#K+1:X(I,I,KK)=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#LE#K #AND#
    I#NE#KK:@FOR(Mesto(J)|J#GE#K+1:X(I,J,KK)=0));
@FOR(Trasa:@BIN(X));
DATA:
    @OLE('MTSP.xls','TMIN')=TMIN;
ENDDATA
END

```

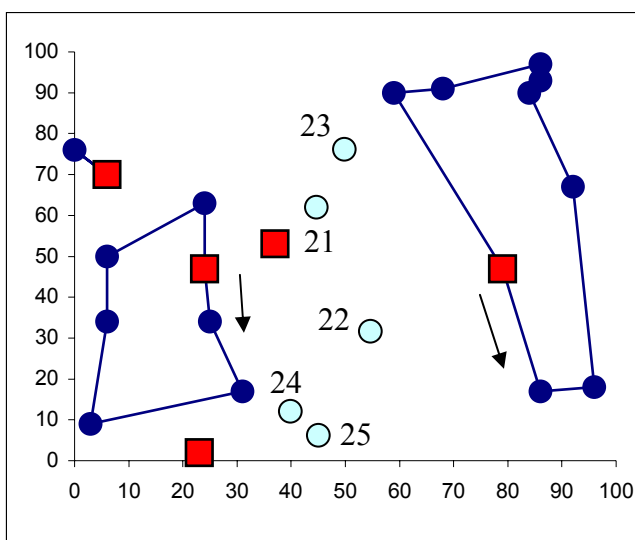
Lokalizace souboru na CD: **Kapitola 4 / Část 4.4.2 / MTSP.xls**

Příloha č. 8

Dynamická úloha s více obchodními cestujícími v několika výchozích místech

Minimalizace celkové vzdálenosti

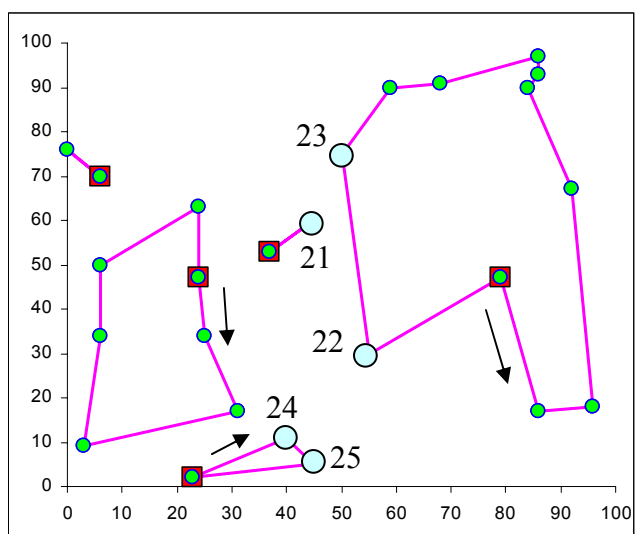
V příloze č. 5 je uvedeno optimální řešení příkladu s vygenerovanými 5 výchozími místy a 15 zákazníky. Obsluhu zajišťují pouze 3 vozidla, zbývající 2 vozidla zůstávají ve svých výchozích místech (obr. P8.1). Celková délka všech 3 tras je přibližně 353,97 km. Předpokládejme, že po výjezdu vozidel ze svých stanovišť přicházejí postupně požadavky dalších zákazníků č. 21-25 (tabulka P8.1).



Zákazník	X	Y
21	45	60
22	55	30
23	50	75
24	40	11
25	45	5

Tab. P8.1 – Noví zákazníci

Obr. P8.1 – Optimální trasy 3 vozidel (2 vozidla nevyjždějí)



Obr. P8.2 – Vkládací algoritmus

Na obr. P8.2 jsou znázorněny trasy vozidel, je-li pro zařazení nových zákazníků použit vkládací algoritmus. K zákazníkovi č. 21 vyjíždí ze svého stanoviště volné vozidlo, podobně jako jiné vozidlo vyjíždí k zákazníkům č. 24 a 25. Zákazníci č. 22 a 23 jsou obslouženi vozidlem, které je na trase. Délka všech tras činí přibližně 469,24 km.

Lokalizace souboru na CD: **Kapitola 4 / Části 4.3.2 a 4.6.1 / MTSP.xls**

Příloha č. 9

Rozvozní úloha s jedním vozidlem

Pro řešení dynamických rozvozních úloh je připraven soubor *VVRP.xls* s hlavním listem *DVRP* (obr. P9.1), ve kterém uživatel musí zadat kromě počtu známých zákazníků a celkového počtu zákazníků také kapacitu vozidla.

Dynamická rozvozní úloha

Počet známých zákazníků	7	Generování zákazníků
Celkový počet zákazníků	10	
Kapacita vozidla	100,00	

Optimální trasa pro známé zákazníky

Délka trasy statické úlohy	265,24	Optimalizace	Trasa
----------------------------	--------	--------------	-------

Přidávání zákazníků: vkládací metoda

Délka trasy dynamické úlohy	445,12	Postupně
-----------------------------	--------	----------

Přidávání zákazníků: re-optimalizace

Délka trasy dynamické úlohy	366,92	Postupně
-----------------------------	--------	----------

Obr. P9.1 – List *DVRP***Statická úloha**

V úloze je vygenerováno 6 zákazníků⁸:

Zákazník	X	Y	Požadavek
1	29	36	-
2	23	59	13
3	58	58	46
4	25	81	20
5	63	49	6
6	97	17	34
7	11	65	39

Tab. P9.1 – Zákazníci a objem jejich požadavků

⁸ Zákazník č. 1 v tabulce P9.1 je výchozím místem.

Ve sloupci Požadavek jsou uvedeny počty jednotek, které se mají svézt do výchozího místa. Ke svozu je určeno jedno vozidlo o kapacitě 100 jednotek. Cílem je minimalizovat celkovou vzdálenost ujetou vozidlem.

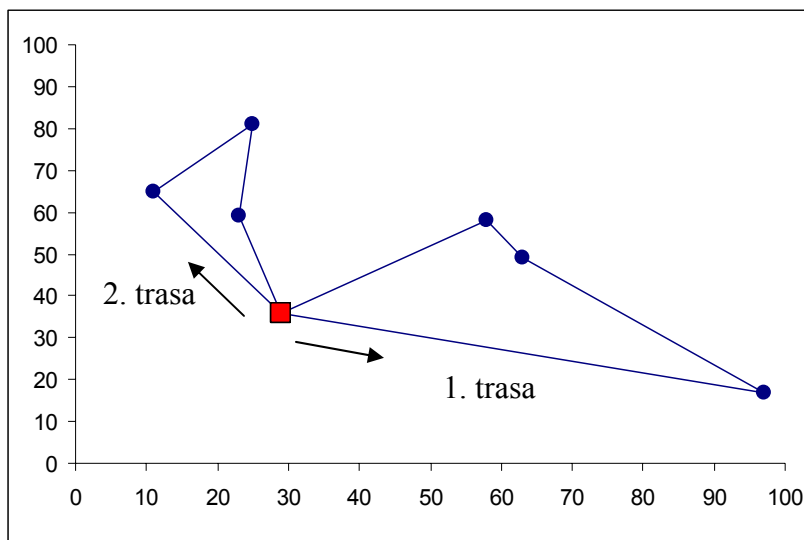
Matematický model (5.1) – (5.7) zapsaný v systému LINGO:

```

MODEL:
SETS:
  Mesto:u,Q;
  Prirazeni(Mesto,Mesto):KM,X;
ENDSETS
DATA:
  Mesto,Q,KM,V=@OLE('VRP.xls');
  @OLE('VRP.xls','Vysledek')=X;
ENDDATA
N=@SIZE(Mesto);
[T]MIN=@SUM(Prirazeni:KM*X);
@FOR(Mesto(I):X(I,I)=0);
@FOR(Mesto(I)|I#NE#1:@SUM(Mesto(J):X(I,J)=1);
@FOR(Mesto(J)|J#NE#1:@SUM(Mesto(I):X(I,J)=1);
U(1)=0;
@FOR(Mesto(I):@FOR(Mesto(J)|J#NE#1 #AND# J#NE#I:U(I)-U(J)+Q(J)-
  V*(1-X(I,J))<=0);
@FOR(Mesto(I)|I#NE#1:Q(I) - U(I)<=0);
@FOR(Mesto(I)|I#NE#1:U(I) - V <= 0);
@FOR(Prirazeni:@BIN(X));
DATA:
  @OLE('VRP.xls','T')=T;
ENDDATA
END

```

Optimální trasy jsou znázorněny na obr. P9.2. Je zřejmé, že svoz jednotek by mohla zajišťovat 2 vozidla, která mohou vyjet zároveň. Pokud je pro svoz určeno jediné vozidlo, vyjede na druhou trasu až po složení prvního nákladu ve výchozím místě.



Obr. P9.2 – Optimální trasy svozu jednotek 1 vozidlem

Informace o délce trasy a nákladu vozidla jsou uvedeny v tabulce P9.2. Na první trase ujede vozidlo přibližně 163,99 km, přičemž svezí 86 jednotek⁹. Na druhé trase vozidlo svezí 72 jednotek. Celkem vozidlo ujede přibližně 265,24 km, délka druhé trasy tedy činí přibližně 101,25 km.

Zákazník	X	Y	Délka	Objem
1	29	36	0,00	0
6	97	17	70,60	34
5	63	49	117,30	40
3	58	58	127,59	86
1	29	36	163,99	86
1	29	36	163,99	0
7	11	65	198,12	39
4	25	81	219,38	59
2	23	59	241,47	72
1	29	36	265,24	72

1. trasa
 2. trasa

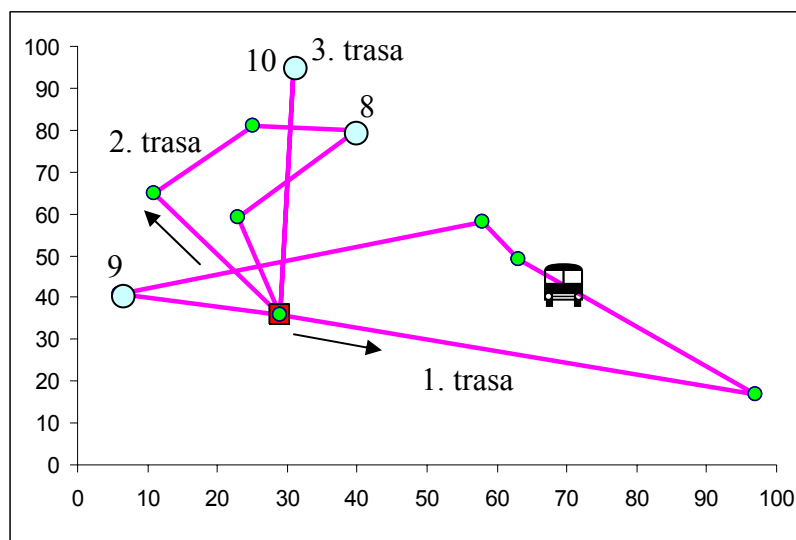
Tab. P9.2 – Délka optimálních tras a objem nákladu vozidla

Dynamická úloha: vkládací algoritmus

Během jízdy vozidla vznikly 3 nové požadavky:

Zákazník	X	Y	Požadavek
8	40	80	26
9	6	41	7
10	31	95	13

Tab. P9.3 – Noví zákazníci



Obr. P9.3 – Zařazení nových zákazníků vkládacím algoritmem

⁹ Údaje ve sloupci Objem představují kumulativní součty požadavků zákazníků na trase po naložení příslušného počtu jednotek u zákazníka, tj. aktuální objem nákladu vozidla.

Vložení nových zákazníků znázorňuje obr. P9.3 a tabulka P9.4. Celková délka všech tras je přibližně 445,12 km.

Zákazník	X	Y	Délka	Objem
1	29	36	0,00	0
6	97	17	70,60	34
5	63	49	117,30	40
3	58	58	127,59	86
9	6	41	182,30	93
1	29	36	205,84	93

Zákazník	X	Y	Délka	Objem
1	29	36	205,84	0
7	11	65	239,97	39
4	25	81	261,23	59
8	40	80	276,26	85
2	23	59	303,28	98
1	29	36	327,05	98

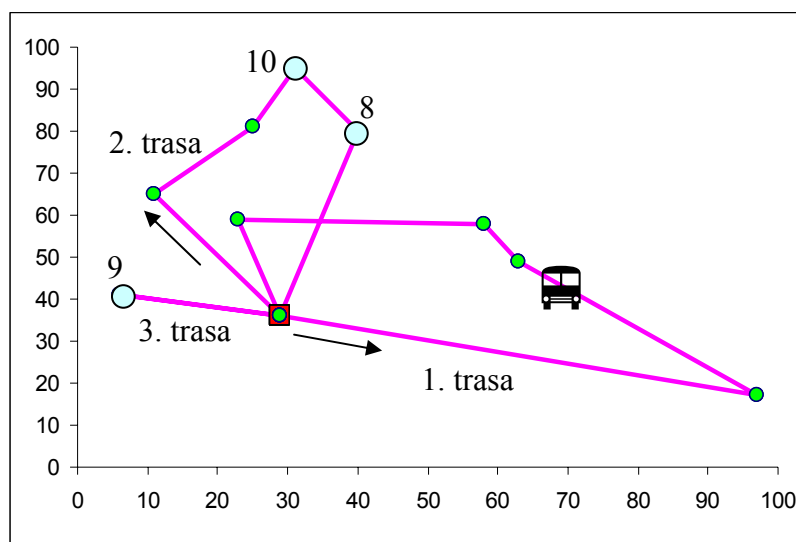
Zákazník	X	Y	Délka	Objem
1	29	36	327,05	0
10	31	95	386,08	13
1	29	36	445,12	13

	1. trasa
	2. trasa
	3. trasa

Tab. P9.4 – Trasy po zařazení nových zákazníků vkládacím algoritmem

Dynamická úloha: re-optimalizace

Zařazení nových zákazníků uvedených v tabulce P9.3 na základě re-optimalizačního algoritmu je znázorněno na obr. P9.4 a v tabulce P9.5.



Obr. P9.4 – Zařazení nových zákazníků na základě re-optimalizace tras

Celková délka všech tras je přibližně 366,92 km, zatímco v případě použití vkládacího algoritmu celková délka činí 445,12 km.

Zákazník	X	Y	Délka	Objem
1	29	36	0,00	0
6	97	17	70,60	34
5	63	49	117,30	40
3	58	58	127,59	86
2	23	59	162,60	99
1	29	36	186,37	99

Zákazník	X	Y	Délka	Objem
1	29	36	186,37	0
7	11	65	220,51	39
4	25	81	241,77	59
10	31	95	257,00	72
8	40	80	274,49	98
1	29	36	319,85	98

Zákazník	X	Y	Délka	Objem
1	29	36	319,85	0
9	6	41	343,38	7
1	29	36	366,92	7

1. trasa
 2. trasa
 3. trasa

Tab. P9.5 – Trasy po zařazení nových zákazníků re-optimalizačním algoritmem

Matematický model (5.8) – (5.16) zapsaný v systému LINGO:

```

MODEL:
SETS:
  MestoR:u,QR;
  Prirazeni (MestoR,MestoR):KMR,X;
ENDSETS
DATA:
  MestoR,QR,KMR,V,JNEXT_INDEX,L=@OLE('VRP.xls');
  @OLE('VRP.xls','VysledekR')=X;
ENDDATA
UN=@SIZE(MestoR);
[TR]MIN=@SUM(Prirazeni:KMR*X);
@FOR(MestoR(I):X(I,I)=0);
@FOR(MestoR(I)|I#NE#1:@SUM(MestoR(J):X(I,J))=1);
@FOR(MestoR(J)|J#NE#1:@SUM(MestoR(I):X(I,J))=1);
@FOR(MestoR(I):@FOR(MestoR(J)|J#NE#1 #AND# J#NE#I:U(I)-U(J)+QR(J)-
  V*(1-X(I,J))<=0);
@FOR(MestoR(I)|I#NE#1:QR(I)-U(I)<=0);
@FOR(MestoR(I)|I#NE#1:U(I)-V<=0);
U(1)=0;
U(JNEXT_INDEX)=L;
X(1,JNEXT_INDEX)=1;
@FOR(Prirazeni:@BIN(X));
DATA:
  @OLE('VRP.xls','TR')=TR;
ENDDATA
END

```

Lokalizace souboru na CD: **Kapitola 5 / Část 5.1 / VRP.xls**

Příloha č. 10

Rozvozní úloha s dělenou dodávkou**Statická úloha**

Ve výchozím místě jsou umístěna 3 vozidla, která mají zajistit svoz materiálu od 4 zákazníků č. 2-5:

Zákazník	X	Y	Požadavek
1	83	72	-
2	35	55	5
3	60	30	30
4	64	84	30
5	18	50	19

Tab. P10.1 – Požadavky zákazníků na svoz materiálu

Kapacita každého vozidla je 50 jednotek. Cílem je minimalizovat celkovou vzdálenost ujetou vozidly.

Matematický model (5.42) – (5.51) zapsaný v systému LINGO:

```

MODEL:
DATA:
    K, V=@OLE('VRP.xls');
ENDDATA
SETS:
    Mesto:Q;
    Vozidlo /1..K/;
    Naklad(Mesto, Vozidlo):u, qq;
    Vzdalenost(Mesto, Mesto):KM;
    Trasa(Vzdalenost, Vozidlo):X;
ENDSETS
DATA:
    Mesto, KM, Q=@OLE('VRP.xls');
    @OLE('VRP.xls', 'Vysledek')=X;
    @OLE('VRP.xls', 'Vysledekq')=qq;
ENDDATA
N=@SIZE(Mesto);
[T]MIN=@SUM(Vozidlo(KK):@SUM(Vzdalenost(I, J):KM(I, J)*X(I, J, KK)));
@FOR(Vozidlo(KK):@SUM(Mesto(J)|J#GT#1:X(1, J, KK))<=1);
@FOR(Vozidlo(KK):@FOR(Mesto(J)|J#GT#1:@SUM(Mesto(I):X(I, J, KK))-
@SUM(Mesto(I):X(J, I, KK))=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I):@FOR(Mesto(J)|J#NE#1 #AND# J#NE#I:
    U(I, KK)-U(J, KK)+qq(J, KK)-V*(1-X(I, J, KK))<=0));
@FOR(Vozidlo(KK):u(1, KK)=0);
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GT#1:u(I, KK)-V<=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GT#1:qq(I, KK)-u(I, KK)<=0));
@FOR(Mesto(I)|I#GT#1:@SUM(Vozidlo(KK):qq(I, KK))-Q(I)=0);
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GT#1:qq(I, KK)-Q(I)*@SUM(Mesto(J):
    X(I, J, KK))<=0));
@FOR(Vozidlo(KK):@FOR(Vzdalenost(I, I):X(I, I, KK)=0));

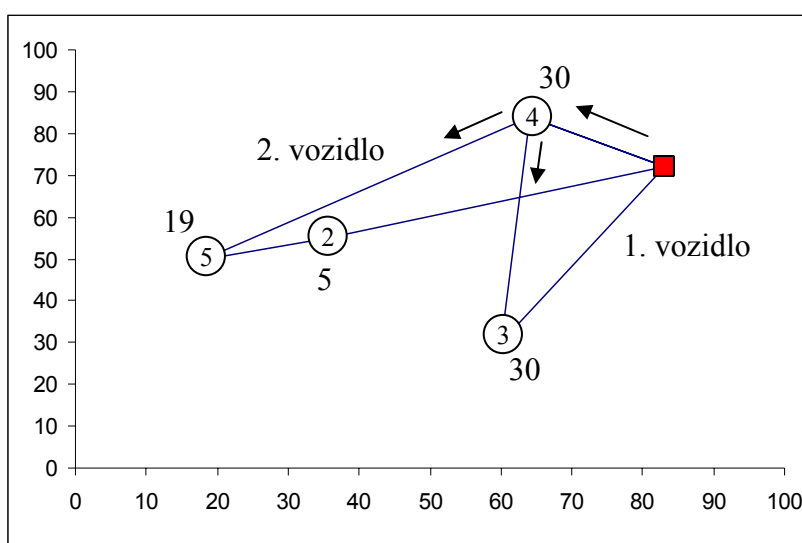
```

```

@FOR(Trasa:@BIN(X));
DATA:
@OLE('VRP.xls','T')=T;
ENDDATA
END

```

Na obr. P10.1 jsou znázorněny trasy 2 vozidel, které budou ke svozu použity. Zákazník č. 4 bude jako jediný obslužen oběma vozidly. Z tabulky P10.2, v níž je uveden průběžný náklad obou vozidel, lze snadno zjistit, že 1. vozidlo odveze od zákazníka č. 4 pouze 4 jednotky, zatímco 2. vozidlo od něj odveze zbývajících 26 jednotek. Celkový náklad 1. vozidla činí 34 jednotek, náklad 2. vozidla 50 jednotek, tj. toto vozidlo je plně využito. Obě vozidla ujedou celkem přibližně 272,82 km.



Obr. P10.1 – Optimální trasy svozu materiálu 2 vozidly

Zákazník	X	Y	Délka	Objem
1	83	72	0,00	0
4	64	84	22,47	4
3	60	30	76,62	34
1	83	72	124,51	34

1. vozidlo

2. vozidlo

Zákazník	X	Y	Délka	Objem
1	83	72	0,00	0
4	64	84	22,47	26
5	18	50	79,67	45
2	35	55	97,39	50
1	83	72	148,32	50

Tab. P10.2 – Optimální trasy vozidel

Lokalizace souboru na CD: **Kapitola 5 / Část 5.4.1 / VRP.xls**

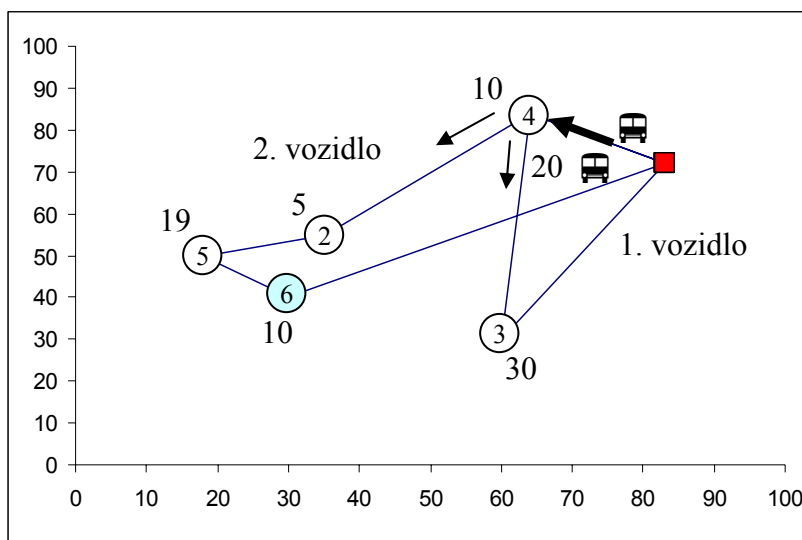
Dynamická úloha: re-optimalizace

Zařazení nového zákazníka uvedeného v tabulce P10.3 je provedeno na základě re-optimalizačního algoritmu.

Zákazník	X	Y	Požadavek
6	30	40	10

Tab. P10.3 – Nový zákazník

Obr. P10.2 znázorňuje trasy po re-optimalizaci tras, pokud požadavek nového zákazníka vznikne v okamžiku, kdy obě vozidla jedou z výchozího místa k zákazníkovi č. 4.



Obr. P10.2 – Vznik nového požadavku v okamžiku jízdy vozidel k zákazníkovi č. 4

Pro optimalizaci byl použitý matematický model statické úlohy upravený pro tuto konkrétní úlohu:

```

MODEL:
DATA:
  K, V=@OLE('VRP.xls');
ENDDATA
SETS:
  Mesto:Q;
  Vozidlo /1..K/;
  Naklad(Mesto, Vozidlo) :u, qq;
  Vzdalenost(Mesto, Mesto) :KM;
  Trasa(Vzdalenost, Vozidlo) :X;
ENDSETS
DATA:
  Mesto, KM, Q=@OLE('VRP.xls');
  @OLE('VRP.xls', 'Vysledek')=X;
  @OLE('VRP.xls', 'Vysledekq')=qq;
ENDDATA

```

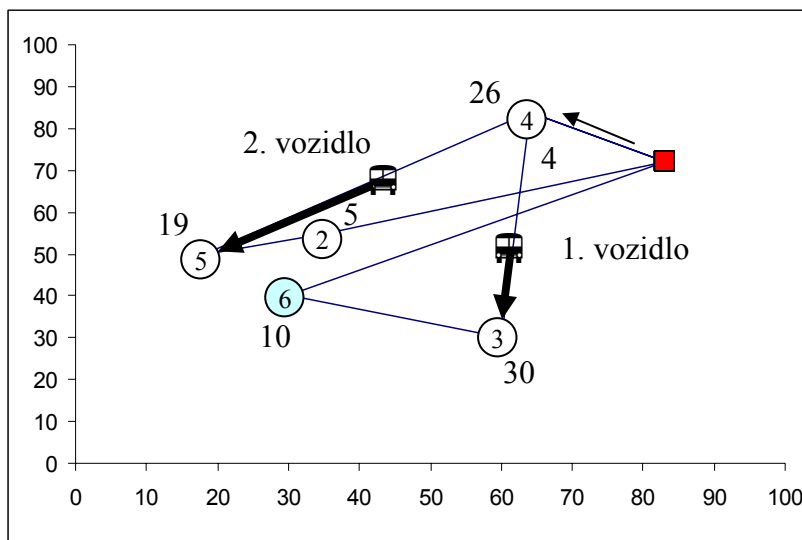
```

N=@SIZE(Mesto);
[T]MIN=@SUM(Vozidlo(KK):@SUM(Vzdalenost(I,J):KM(I,J)*X(I,J,KK)));
@FOR(Vozidlo(KK):@SUM(Mesto(J)|J#GT#1:X(1,J,KK))<=1);
@FOR(Vozidlo(KK):@FOR(Mesto(J)|J#GT#1:@SUM(Mesto(I):
X(I,J,KK)-@SUM(Mesto(I):X(J,I,KK))=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I):@FOR(Mesto(J)|J#NE#1 #AND# J#NE#I:
U(I,KK)-U(J,KK)+qq(J,KK)-V*(1-X(I,J,KK))<=0));
@FOR(Vozidlo(KK):u(1,KK)=0);
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GT#1:u(I,KK)-V<=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GT#1:qq(I,KK)-u(I,KK)<=0));
@FOR(Mesto(I)|I#GT#1:@SUM(Vozidlo(KK):qq(I,KK))-Q(I)=0);
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GT#1:qq(I,KK)-
Q(I)*@SUM(Mesto(J):X(I,J,KK))<=0));
@FOR(Vozidlo(KK):@FOR(Vzdalenost(I,I):X(I,I,KK)=0));
X(1,4,1)=1;
X(1,4,2)=1;
@FOR(Trasa:@BIN(X));
DATA:
@OLE('VRP.xls','T')=T;
ENDDATA
END

```

Lokalizace souboru na CD: **Kapitola 5 / Část 5.4.2 / Příklad 5.9 / VRP.xls**

Na obr. P10.3 je výsledek re-optimalizace tras pro situaci, v níž nový požadavek vznikne v okamžiku, kdy 1. vozidlo jede od zákazníka č. 4 k zákazníkovi č. 3 a 2. vozidlo jede od zákazníka č. 4 k zákazníkovi č. 5.



Obr. P10.3 – Vznik nového požadavku v okamžiku jízdy vozidel od zákazníka č. 4

Upravený model zapsaný v systému LINGO:

```

MODEL:
DATA:
  K,V=@OLE('VRP.xls');
ENDDATA
SETS:
  Mesto:Q;
  Vozidlo /1..K/;
  Naklad(Mesto,Vozidlo):u,qq;
  Vzdalenost(Mesto,Mesto):KM;
  Trasa(Vzdalenost,Vozidlo):X;
ENDSETS
DATA:
  Mesto,KM,Q=@OLE('VRP.xls');
  @OLE('VRP.xls','Vysledek')=X;
  @OLE('VRP.xls','Vysledekq')=qq;
ENDDATA
N=@SIZE(Mesto);
[T]MIN=@SUM(Vozidlo(KK):@SUM(Vzdalenost(I,J):KM(I,J)*X(I,J,KK)));
@FOR(Vozidlo(KK):@SUM(Mesto(J)|J#GT#1:X(1,J,KK))<=1);
@FOR(Vozidlo(KK):@FOR(Mesto(J)|J#GT#1:@SUM(Mesto(I):X(I,J,KK))-
@SUM(Mesto(I):X(J,I,KK))=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I):@FOR(Mesto(J)|J#NE#1 #AND# J#NE#I:U(I,KK)-
U(J,KK)+qq(J,KK)-V*(1-X(I,J,KK))<=0));
@FOR(Vozidlo(KK):u(1,KK)=0);

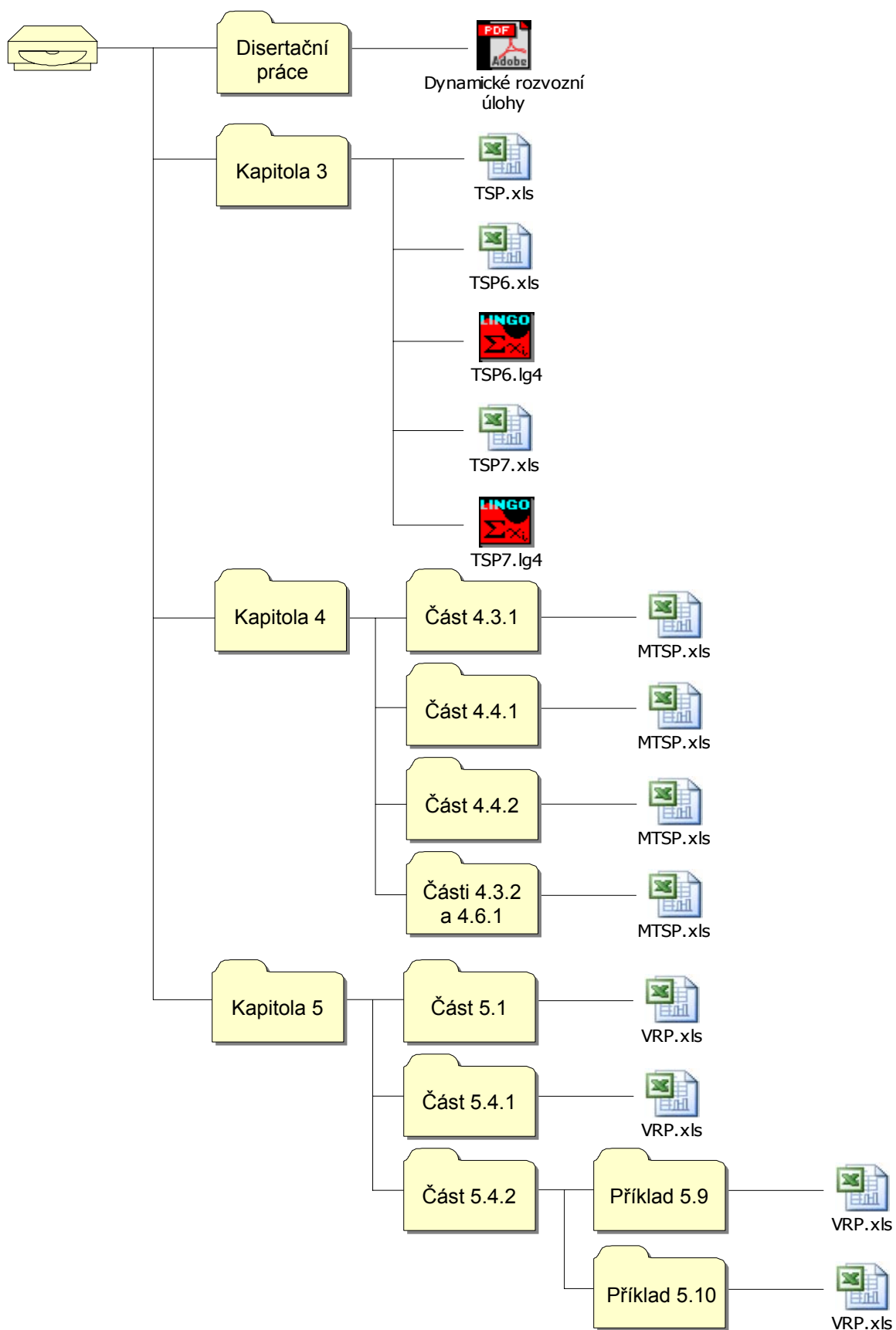
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GT#1:u(I,KK)-V<=0));
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GT#1:qq(I,KK)-u(I,KK)<=0));
@FOR(Mesto(I)|I#GT#1:@SUM(Vozidlo(KK):qq(I,KK))-Q(I)=0);
@FOR(Vozidlo(KK):@FOR(Mesto(I)|I#GT#1:qq(I,KK)-
Q(I)*@SUM(Mesto(J):X(I,J,KK))<=0));
@FOR(Vozidlo(KK):@FOR(Vzdalenost(I,I):X(I,I,KK)=0));
X(1,4,1)=1;
X(1,4,2)=1;
X(4,5,2)=1;
X(4,3,1)=1;
qq(4,1)=4;
qq(4,2)=26;
@FOR(Trasa:@BIN(X));
DATA:
@OLE('VRP.xls','T')=T;
ENDDATA
END

```

Lokalizace souboru na CD: **Kapitola 5 / Část 5.4.2 / Příklad 5.10 / VRP.xls**

V prvním případě obě vozidla ujedou přibližně 283,24 km, ve druhém případě 318,47 km. Z uvedených obrázků lze snadno zjistit, jaký náklad budou mít vozidla v průběhu svých jízd.

Příloha č. 11

Obsah přiloženého CD

Terminologický slovník

Termín	Význam
TSP	Traveling Salesman/Salesperson Problem, úloha obchodního cestujícího. Všichni zákazníci jsou známi před zahájením okružní jízdy jediného vozidla.
DTSP	Dynamic Traveling Salesman/Salesperson Problem, dynamická úloha obchodního cestujícího. V okamžiku, kdy je obsluhující vozidlo na trase, vznikají další požadavky, které jsou zařazeny do zbývajících částí trasy.
TSPTW	Traveling Salesman/Salesperson Problem with Time Windows, úloha obchodního cestujícího s časovými okny. Všichni zákazníci jsou známi před zahájením okružní jízdy jediného vozidla. U každého zákazníka je definován časový interval, ve kterém má být obslužen.
DTSPTW	Dynamic Traveling Salesman/Salesperson Problem with Time Windows, dynamická úloha obchodního cestujícího s časovými okny. V okamžiku, kdy je obsluhující vozidlo na trase, vznikají další požadavky, které jsou zařazeny do zbývajících částí trasy. Kromě definice své lokality každý zákazník oznámí časový interval, ve kterém má být obslužen.
ADTSPTW	A Priori Dynamic Traveling Salesman/Salesperson Problem with Time Windows, dynamická úloha obchodního cestujícího s časovými okny a s apriorní informací. Speciální případ úlohy DTSPTW, v níž má firma pravděpodobnostní informace o místě vzniku nových požadavků.
mTSP m-TSP MTSP	Multiple Traveling Salesman/Salesperson Problem, úloha s více obchodními cestujícími. Obsluhu zákazníků zajišťuje více vozidel umístěných v jediném či několika výchozích místech. Všichni zákazníci jsou známi před zahájením okružních jízd vozidel.
VRP	Vehicle Routing Problem, rozvozní úloha. Cílem je splnit požadavky zákazníků na rozvoz či svoz požadovaného množství zboží, materiálu apod. Všichni zákazníci jsou známi před zahájením rozvozu či svozu realizovaného většinou několika vozidly.
VRPTW	Vehicle Routing Problem with Time Windows, rozvozní úloha s časovými okny. Cílem je splnit požadavky zákazníků na rozvoz či svoz požadovaného množství zboží, materiálu apod. Všichni zákazníci jsou známi před zahájením rozvozu či svozu realizovaného většinou několika vozidly. U každého zákazníka je definován časový interval, ve kterém má být obslužen.

Termín	Význam
SDVRP	Split Delivery Vehicle Routing Problem, rozvozní úloha s dělenou dodávkou. Požadavek každého zákazníka lze realizovat ve více trasách, tj. dodávka může být rozložena do několika subdodávek.
SDP	Skip Delivery Problem, úloha rozvozu kontejnerů vozidly o kapacitě 2. Jedná se o speciální typ úlohy s dělenou dodávkou.

Použitá literatura

- [1] Archetti, C., Speranza, M., Hertz, A.: A Tabu Search Algorithm for the Split Delivery Vehicle Routing Problem. *Transportation Science*. 40: 64-73. 2006.
- [2] Archetti, C., Mansini, R., Speranza, M. G.: Complexity and Reducibility of the Skip Delivery Problem. *Transportation Science*. 39: 182-187. 2005.
- [3] Achuthan, N. R., Caccetta, L., Hill, S. P.: An Improved Branch-and-Cut Algorithm for the Capacited Vehicle Routing Problem. *Transportation Science*. 37: 153-169. 2003.
- [4] Bard, J. F., Kontoravdis, G., Yu, G.: A Branch-and-Cut Procedure for the Vehicle Routing Problem with Time Windows. *Transportation Science*. 36: 250-269. 2002.
- [5] Badeau, P., Gendreau, M., Guertin, F., et al.: A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows. *Transportation Science*. 5C: 109-122. 1997.
- [6] Bektas, T.: The Multiple Traveling Salesman Problem: an Overview of Formulations and Solution Procedures. *Omega*. 34: 209-219. 2006.
- [7] Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., Macj, R. G., Prutzman, P. J.: Improving the Distribution of Industrial Gases with On-Line Computerized Routing and Scheduling Optimizer. *Interfaces*. 13: 4-23. 1983.
- [8] Bräysy, O., Gendreau, M.: Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science*. 39: 104-118. 2005a.
- [9] Bräysy, O., Gendreau, M.: Vehicle Routing Problem with Time Windows, Part II: Metaheuristics. *Transportation Science*. 39: 119-139. 2005b.
- [10] Brezina, I.: Kvantitatívne metódy v logistike. EKONÓM. Bratislava. 2003.
- [11] Brown, G. G., Ellis, C. J., Lenn, G., Graves, W., Ronen, D.: Real Time, Wide Area Dispatch of Mobil Tank Trucks. *Interfaces*. 17: 107-120. 1987.
- [12] Cordeau, J.-F.: A Branch-and-Cut Algorithm for the Dial-a-Ride Problem. *Operations research*. 54: 573-586. 2006.

- [13] Cordeau, J.-F., Laporte, G.: A Tabu Search Heuristic for the Static Multi-Vehicle Dial-a-Ride Problem. *Transportation Research Part B*. 37: 579-594. 2003.
- [14] Čičková, Z.: Aplikácia evolučných algoritmov na riešenie úlohy obchodného cestujúceho. Dizertačná práca. KOVE FHI, EU, Bratislava. 2005.
- [15] Desrochers, M., Desrosiers, J., Solomon, M.: A New Algorithm for the Vehicle Routing Problem with Time Windows. *Operations Research*. 40: 342-354. 1992.
- [16] Desrosiers, J., Dumas, Y., Solomon, M. M., Soumis, F.: Time Constrained Routing and Scheduling. In Ball, M. O., Magnanti, T. L., Monma, C. L., Nemhauser, G. L. (eds.): *Network Routing, Handbooks in Operations Research and Management Science*. 8: 35-139. Elsevier Science. North-Holland, Amsterdam. 1995.
- [17] Dial, R. B.: Autonomous Dial-a-Ride Transit Introductory Overview. *Transportation Science*. 3C: 261-275. 1995.
- [18] Dror, M., Trudeau, P.: Savings by Split Delivery Routing. *Transportation Science*. 23: 141-145. 1989.
- [19] Dror, M., Laporte, G., Trudeau, P.: Vehicle Routing with Split Deliveries. *Discrete Applied Mathematics*. 50: 239-254. 1994.
- [20] Fábry, J.: Dynamic Traveling Salesman Problem. In *Mathematical Methods in Economics 2006 (CD-ROM)*. Plzeň. 137-146. 2006.
- [21] Fábry, J.: Rozvozní úloha s dělenou dodávkou. In *Progressive Methods and Tools of Management and Economics of Companies (CD-ROM)*. 7 stran. Brno. 2005.
- [22] Fábry, J.: Metoda generování sloupců a její použití v celočíselném programování. In Hronová, S. (ed.). *Sborník prací účastníků vědeckého semináře doktorandského studia FIS VŠE v Praze*. Oeconomica. Praha. 152-163. 2005.
- [23] Fábry, J., Pelikán, J.: Method of Column Generation Used in Integer Programming. In Bauer, L. (ed.). *Mathematical Methods in Economics 2004*. Brno. 242-247. 2004.
- [24] Gavish, B., Srikanth, K.: An Optimal Solution Method for Large-Scale Multiple Traveling Salesmen Problems. *Operations Research*. 34: 698-717. 1986.
- [25] Gendreau, M., Hertz, A., Laporte, G.: A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science*. 40: 1276-1290. 1994.

- [26] Gendreau, M., Laporte, G., Séguin, R.: Stochastic Vehicle Routing. *European Journal of Operational Research*. 88: 3–12. 1996.
- [27] Gendreau, M., Potvin, J.-Y.: Dynamic Vehicle Routing and Dispatching. In Crainic, T., Laporte, G. (eds.): *Fleet Management and Logistics*. 115–126. Kluwer Academic Publisher. Boston, MA. 1998.
- [28] Gendreau, M., Guertin, F., Potvin, J.-Y., Taillard, É.: Paralel Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*. 33: 381-390. 1999.
- [29] Golden, B. L., Wasil, E. A., Kelly, J. P., Chao, I.-M.: Metaheuristics in Vehicle Routing. In Crainic, T., Laporte, G. (eds.): *Fleet Management and Logistics*. 33–56. Kluwer Academic Publisher. Boston, MA. 1998.
- [30] Hill, A., Mabert, V., Montgomery, D.: A Decision Support System for the Courier Vehicle Scheduling Problem. *Omega - International Journal of Management Science*. 16: 333-345. 1988.
- [31] Homberger, J., Gehring, H.: Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows. *Information Systems and Operational Research*. 37: 297-318. 1999.
- [32] Chen, Z.-L., Xu, H.: Dynamic Column Generation for Dynamic Vehicle Routing with Time Windows. *Transportation Science*. 40: 74-88. 2006.
- [33] Ichoua, S., Gendreau, M., Potvin, J.-Y.: Diversion Issues in Real-Time Vehicle Dispatching. *Transportation Science*. 34: 426-438. 2000.
- [34] Janáček, J.: *Optimalizace na dopravních sítích*. EDIS. Žilina. 2003.
- [35] Jaw, J., Odoni, A. R., Psaraftis, H. N., Wilson, N. H. M.: A Heuristic Algorithm for the Multi-Vehicle Advance-Request Dial-a-Ride Problem with Time Windows. *Transportation Research Part B*. 20: 243-257. 1986.
- [36] Kolen, A. W. J., Rinnooy Kan, A. H. G., Trienekens, H. W. J. M.: Vehicle Routing with Time Windows. *Operations Research*. 35: 266-273. 1987.
- [37] Laporte, G., Nobert, Y.: A Cutting Planes Algorithm for the M-Salesman Problem. *Operational Research Quarterly*. 31:1017-1023. 1980.

- [38] Laporte, G., Nobert, Y., Taillifer, S.: Solving a Family of Multi-Depot Vehicle Routing and Location-Routing Problems. *Transportation Science*. 22: 161-172. 1988.
- [39] Larsen, A., Madsen, O. B. G., Solomon, M. M.: The A Priori Dynamic Salesman Problem with Time Windows. *Transportation Science*. 38: 459-472. 2004.
- [40] Lund, K., Madsen, O., Rygaard, J.: Vehicle Routing Problem with Varying Degrees of Dynamics. Technical report. IMM-REP-1996-1, Institute of Mathematical Modeling, Technical University of Denmark. Lyngby, Denmark. 1996.
- [41] Pelikán, J.: Diskrétní modely v operačním výzkumu. Professional Publishing. Praha. 2001.
- [42] Potvin, J.-Y., Kervahut, T., Garcia, B.-L., Rousseau, J.-M.: The Vehicle Routing Problem with Time Windows, Part I: Tabu Search. *INFORMS Journal on Computing*. 8: 158-164. 1996.
- [43] Potvin, J.-Y., Bengio, S.: The Vehicle Routing Problem with Time Windows, Part II: Genetic Search. *INFORMS Journal on Computing*. 8: 165-172. 1996.
- [44] Powell, W., Sheffi, Y., Nickerson, K. S., Butterbaugh, K., Atherton, S.: Maximizing Profits for North American Van Lines Truckload Division: A New Framework for Pricing and Operations. *Interfaces*. 18: 21–41. 1988.
- [45] Powell, W., Jaillet, P., Odoni, A.: Stochastic and Dynamic Networks and Routing. In Ball, M. O., Magnanti, T. L., Monma, C. L., Nemhauser, G. L. (eds.): *Network Routing, Handbooks in Operations Research and Management Science*. 8: 141–295. Elsevier Science. Amsterdam, The Netherlands. 1995.
- [46] Psaraftis, H. N.: A Dynamic Programming Solution to the Single-Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem. *Transportation Science*. 14: 130-154. 1980.
- [47] Psaraftis, H. N.: An Exact Algorithm for the Single Vehicle Many-to-Many Dial-a-Ride Problem with Time Windows. *Transportation Research*. 17: 351-357. 1983.
- [48] Psaraftis, H. N.: Vehicle Routing: Methods and Studies. B. Golden, A. Assad, eds. *Dynamic Vehicle Routing Problems*. North Holland. 223–248. Amsterdam, The Netherlands. 1988.

- [49] Psaraftis, H. N.: Dynamic Vehicle Routing: Status and Prospects. *Annals of Operations Research*. 61: 143-64. 1995.
- [50] Russel, R. A.: Hybrid Heuristics for the Vehicle Routing Problem with Time Windows. *Transportation Science*. 29: 156-166. 1995.
- [51] Savelsbergh, M. W. P.: The Vehicle Routing Problem with Time Windows: Minimizing Route Duration. *ORSA Journal on Computing*. 4: 146-154. 1992.
- [52] Savelsbergh, M., Sol, M.: DRIVE: Dynamic Routing of Independent vehicles. *Operations Research*. 46: 474-490. 1998.
- [53] Taillard, É. D., Badeau, P., Gendreau, M, Guertin, F., Potvin, J.-Y.: A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows. *Transportation Science*. 31: 170-186. 1997.
- [54] Toth, P., Vigo, D.: Exact Solution of the Vehicle Routing Problem. In Crainic, T., Laporte, G. (eds.): *Fleet Management and Logistics*. 1–31. Kluwer Academic Publisher. Boston, MA. 1998.
- [55] Toth, P., Vigo, D.: Heuristic Algorithms for the Handicapped Persons Transportation Problem. *Transportation Science*. 31: 60-71. 1997.

Internetové zdroje a důležité odkazy:

- [56] <http://fausto.eco.unibs.it/dmq/ricerca/quaderni/201.pdf>
Archetti, C., Mansini, R., Speranza, M. G.: The Split Delivery Vehicle Routing Problem with Small Capacity. 2001. (31.7.2006).
- [57] http://www.idsia.ch/~luca/MAS2003_18.pdf
Gambardella, L. M., Rizzoli, A. E., Oliverio, F., Casagrande, N., Donati, A. V., Montemanni, R., Lucibello, E.: Ant Colony Optimization for Vehicle Routing in Advanced Logistics Systems. *IDSIA*. 2003. (7.9.2006).
- [58] <ftp://ftp.idsia.ch/pub/techrep/IDSIA-23-02.pdf.gz>
Montemanni, R., Gambardella, L. M., Rizzoli, A. E., Donati, A. V.: A New Algorithm for a Dynamic Vehicle Routing Problem Based on Ant Colony System. Technical Report *IDSIA-23-02*, *IDSIA*. 2002. (7.9.2006).

- [59] <http://osiris.tuwien.ac.at/~wgarn/VehicleRouting/neo/data/articles/vrptw4.pdf>
Thangiah, S. R., Osman, I. H., Sun, T.: Hybrid Genetic Algorithm, Simulated Annealing and Tabu Search Methods for Vehicle Routing Problems with Time Windows. Working Paper UKC/IMS/OR94/4, University of Kent, Canterbury. 1994 (7.9.2006).
- [60] <http://www.lindo.com/>
Internetové stránky společnosti Lindo Systems, Inc. (31.7.2006), jejíž produktem je optimalizační systém LINGO.
- [61] <http://www.dashoptimization.com/>
Internetové stránky společnosti Dash Optimization, Inc. (31.7.2006), jejíž produktem je systém XPRESS MP.