

UNIVERZITA PARDUBICE
DOPRAVNÍ FAKULTA JANA PERNERA

**ALOKAČNÍ ÚLOHY V TURBULENTNÍM
PROSTŘEDÍ**

DISERTAČNÍ PRÁCE

2011

Ing. Filip Vízner

UNIVERSITY OF PARDUBICE

JAN PERNER TRANSPORT FACULTY

**THE ALLOCATION PROBLEM
IN THE TURBULENT ENVIRONMENT**

DOCTORAL DISSERTATION

2011

Ing. Filip Vízner

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 16.3.2011

Ing. Filip Vízner

ANOTACE

Práce řeší metodiku svozně-rozvozního problému s obsluhou hran ve vazbě na čištění komunikací a svoz komunálního odpadu ve větších městech a průmyslových aglomeracích. Problematika je řešena s využitím aparátu metod teorie grafů a lineárního programování s počítačovou podporou tvorby tras obslužných vozidel s aplikací v prostředí GIS. Metodika sestavování tras obslužných vozidel je vytvořena na základě analýzy a vyhodnocení současných trendů v oblasti výpočtů svozně-rozvozních problémů. Aplikační část práce je zaměřena na výpočet vícekriteriálních svozně-rozvozních úloh na vektorovém mapovém podkladu s možností vizualizace výsledných tras v GIS.

KLÍČOVÁ SLOVA

svozně-rozvozní problém s obsluhou hran, GIS, GPS, plány tras, software, operační výzkum, lineární programování, teorie grafů, čištění a údržba pozemních komunikací

TITLE

The Allocation Problem in the Turbulent Environment

ANNOTATION

The dissertation is focused on Arc Routing Problem in attachment to route sweeping and waste collection in towns and industrial agglomerations. The questions are solved with utilizing of computer supported route planning using GIS and also with application of linear programming and graph theory. The methodology of the route planning is constructed with regard to analysis and evaluation of actual trends in solving Arc Routing Problems. The application part of dissertation is focused on solving of multi-criterial Arc Routing Problem with the vector map as entering data with possibility to route visualization in the GIS.

KEYWORDS

Arc Routing Problem, GIS, GPS, Route Planning, Software, Operation Research, Linear Programming, Graph Theory, Street Sweeping

Poděkování:

V úvodu disertační práce bych chtěl vyslovit poděkování mému školiteli doc. Ing. Josefu Volkovi, CSc. za odborné vedení, cenné připomínky a rady, kterých se mi v průběhu celého studia dostávalo.

Autor

OBSAH

Úvod	11
1 Současný stav problému	12
1.1 Technologie ČUPK	13
1.2 Současná právní situace v oblasti ČUPK	14
2 Cíle disertační práce	19
3 Metody zkoumání	20
3.1 Teoretický aparát metod zkoumání	20
3.1.1 Aparát teorie grafů	20
3.1.2 Aparát matematického programování	25
3.1.3 Složitost algoritmů	28
3.2 Analýza metod pro výpočet svozně-rozvozních úloh s obsluhou hran	30
3.3 Problém čínského poštáka	35
3.3.1 Neorientovaný problém čínského poštáka (<i>UCPP – Undirected Chinese Postman Problem</i>)	35
3.3.2 Orientovaný problém čínského poštáka (<i>DCPP – Directed Chinese Postman Problem</i>)	40
3.3.3 Nesouměrný problém čínského poštáka (<i>WPP – Windy Postman Problem</i>)	43
3.3.4 Smíšený problém čínského poštáka (<i>M CPP – Mixed Chinese Postman Problem</i>)	44
3.3.5 Stupňovitý problém čínského poštáka (<i>H CPP – Hierarchical Chinese Postman Problem</i>)	54
3.4 Problém příměstského poštáka (<i>RPP – Rural Postman Problem</i>)	56
3.4.1 Neorientovaný problém příměstského poštáka (<i>URPP – Undirected Rural Postman Problem</i>)	56
3.4.2 Orientovaný problém příměstského poštáka (<i>DRPP – Directed Rural Postman Problem</i>)	60
3.4.3 Problém regálového nakladače (<i>SCP – Stacker Crane Problem</i>)	62
3.5 Kapacitně omezený svozně-rozvozní problém (<i>CARP – Capacited Arc Routing Problem</i>)	67
3.5.1 Orientovaný kapacitně omezený svozně-rozvozní problém (<i>DCARP – Directed Capacited Arc Routing Problem</i>)	68
3.5.2 Neorientovaný kapacitně omezený svozně-rozvozní problém (<i>UCARP – Undirected Capacited Arc Routing Problem</i>)	69

3.6	Smíšený kapacitně omezený svozně-rozvozní problém (MCARP – Mixed Capacited Arc Routing Problem)	71
3.6.1	<i>Matematický model MCARP</i>	72
3.6.2	<i>Identifikace podmínek v metodě sečných nadrovin pro MCARP</i>	75
4	Řešení úkolu a jeho výsledky	78
4.1	Přístup k řešení	78
4.2	Metodika tvorby tras obslužných vozidel provádějících ČUPK.....	78
4.2.1	<i>Konstruktivní heuristiky pro MCARP</i>	78
4.2.2	<i>Úprava heuristik řešících MCARP</i>	80
4.2.3	<i>Heuristika Path-Scanning (PS)</i>	83
4.2.4	<i>Heuristika Augment-Merge (AM)</i>	84
4.2.5	<i>Ulusoyova heuristika (Split procedura)</i>	87
4.2.6	<i>Genetické algoritmy a svozně-rozvozní úlohy s obsluhou hran</i>	89
4.2.7	<i>Memetický algoritmus</i>	91
4.2.8	<i>Upravený MA pro MCARP rozšířený o další omezení</i>	96
4.3	Řešení svozně-rozvozního problému s využitím GIS	98
4.3.1	<i>GIS software</i>	99
4.4	GIS data jako vstup pro výpočet metodami operačního výzkumu.....	100
4.4.1	<i>Zdroje vektorových dat</i>	100
4.4.2	<i>Rastrová data</i>	100
4.4.3	<i>Vektorová data jako vstup pro výpočet</i>	101
4.4.4	<i>Určování polohy míst obsluhy pomocí GPS a přenesení údajů do grafu sítě</i>	104
4.4.5	<i>NetOpt a existující GIS software pro řešení ARP</i>	105
4.5	Postup sestavení, verifikace a implementace metodiky	106
4.5.1	<i>Implementace metodiky do SW NetOpt</i>	107
4.5.2	<i>Verifikace navržené metodiky</i>	108
5	Vyhodnocení a diskuse výsledků	110
6	Návrhy na využití výsledků	111
Závěr	112
Seznam použitých informačních zdrojů	114
Publikační činnost autora	117
Seznam tabulek	118
Seznam obrázků		119
Seznam příloh		121

SEZNAM ZKRATEK

3-D	3-Dimensional (trojrozměrný)
AM	Augment-Merge heuristika (konstruktivní heuristika)
APRS	Automatic Position Reporting System (radioamatérská digitální komunikace)
ARP	Arc Routing Problem (svozně-rozvozní problém s obsluhou hran)
a.s.	akciová společnost
CAD	Computer Aided Design (typ aplikace pro vektorové kreslení)
CARP	Capacited Arc Routing Problem (kapacitně omezený svozně-rozvozní problém s obsluhou hran)
CARP-TW	Capacited Arc Routing Problem with Time Windows (kapacitně omezený svozně-rozvozní problém s obsluhou hran s časovými okny)
CCPP	Capacited Chinese Postman Problem (kapacitně omezený problém čínského pošťáka)
CPP	Chinese Postman Problem (problém čínského pošťáka)
ČR	Česká republika
ČUPK	Čištění a údržba pozemních komunikací
DCARP	Directed Capacited Arc Routing Problem (orientovaný kapacitně omezený svozně-rozvozní problém s obsluhou hran)
DCPP	Directed Chinese Postman Problem (orientovaný problém čínského pošťáka)
DPH	daň z přidané hodnoty
DRPP	Directed Rural Postman Problem (orientovaný problém příměstského pošťáka)
GA	Genetic algorithm (genetický algoritmus)
GIS	Geographic Information System (geografický informační systém)
GPS	Global Position System (globální polohový systém)
GSM	Global System for Mobile Communication (globální systém pro mobilní komunikaci)
HCPP	Hierarchical Chinese Postman Problem (stupňovitý problém čínského pošťáka)

ILP	Integer Linear Programming (<i>celočíselné lineární programování</i>)
IM	Improved Merge procedure (<i>vylepšená Merge heuristika</i>)
LP	Linear Programming (<i>lineární programování</i>)
LOX	Linear Order Crossover (<i>lineární křížení v genetickém algoritmu</i>)
MA	Memetic Algorithm (<i>memetický algoritmus</i>)
MCEOP	Minimum Cost Eulerian Orientation Problem (<i>problém minimální Eulerovské orientace</i>)
MCARP	Mixed Capacitated Arc Routing Problem (<i>smíšený kapacitně omezený svozně-rozvozní problém s obsluhou hran</i>)
MCPP	Mixed Chinese Postman Problem (<i>smíšený problém čínského pošťáka</i>)
MIP	Mixed Integer Programming (<i>smíšené celočíselné programování</i>)
MS	Microsoft (<i>americká softwarová společnost</i>)
OX	Order Crossover (<i>typ křížení v genetickém algoritmu</i>)
PC	Personal Computer (<i>osobní počítač</i>)
PDA	Personal Digital Assistant (<i>osobní digitální zařízení s oprečním systémem</i>)
PS	Path-Scanning algoritmus (<i>konstruktivní heuristika</i>)
RPP	Rural Postman Problem (<i>problém příměstského pošťáka</i>)
S-42	geodetický souřadnicový systém 1942
SCP	Stacker Crane Problem (<i>problém regálového zakladače</i>)
SHP	Shapefile (<i>přípona souboru vektorových GIS dat</i>)
S-JTSK	souřadnicový systém jednotné trigonometrické sítě katastrální
SW	software
TIFF	Tag Image File Format (<i>formát digitálního rastrového obrázku</i>)
TSP	Traveling Salesman Problem (<i>problém obchodního cestujícího</i>)

UCARP	Undirected Capacited Arc Routing Problem <i>(neorientovaný kapacitně omezený svozně-rozvozní problém s obsluhou hran)</i>
UCPP	Undirected Chinese Postman Problem <i>(neorientovaný problém čínského pošťáka)</i>
URPP	Undirected Rural Postman Problem <i>(neorientovaný problém příměstského pošťáka)</i>
USA	Spojené státy americké
VRP	Vehicle Routing Problem <i>(problém okružních jízd)</i>
WGS-84	World Geodetic System 1984 <i>(geodetický eliptický souřadnicový systém)</i>
WPP	Windy Postman Problem <i>(problém větrného pošťáka)</i>

ÚVOD

Řešení svozně-rozvozních úloh s obsluhou hran je jeden z nejnáročnějších kombinatorických optimalizačních problémů, který byl podrobněji zkoumán již před 50 lety. Problém spočívá v nalezení optimálního souboru tras pro vozový park, za účelem obsluhy souboru zákazníků. Svozně-rozvozní problém je úloha zkoumaná pro své časté praktické využití, stejně jako pro svou složitost. Existuje mnoho algoritmů, řešících svozně-rozvozní problém, kde jsou obsluhovanými prvky grafu dopravní sítě vrcholy. V jistých případech je vhodné řešit úlohu s obsluhou hran, například soz komunálního odpadu, čištění a údržbu pozemních komunikací, soz, rozvoz školním autobusem, doručování zásilek atd. Z toho je patrné, že svozně-rozvozní problém je často řešen právě na síti městských komunikací. To s sebou nese řadu komplikací. Graf sítě není homogenní, může obsahovat jak orientované hrany (jednosměrné komunikace), tak neorientované hrany (obousměrné komunikace), popřípadě kruhové objezdy, ostrůvky pro chodce atd. Tento fakt výrazně komplikuje řešení problému. Každý rok jsou na tyto služby vynakládány nemalé finanční prostředky, a to zejména z důvodu špatného plánování. Odborníci v oblasti operačního výzkumu tyto problémy zkoumají a navrhují realizovatelná řešení, která přinášejí značné úspory.

V této disertační práci je navržena metodika pro řešení svozně-rozvozního problému s obsluhou hran na síti městských komunikací s využitím vícekriteriálního modelu pro tvorbu tras vozidel s kapacitním a dojezdovým omezením, násobným umístěním deponovacích míst pro vykládku vozidel a jedním depem. Při analýze, sestavování a popisu modelu byla využita teorie grafů a metody celočíselného lineárního programování. Stanovení souboru optimálních tras vozidel je řešeno prostřednictvím upraveného genetického algoritmu, který ve svém jádru zahrnuje konstruktivní heuristické metody a metody lokálního prohledávání. Tento metaheuristický algoritmus je jeden z nejefektivnějších prostředků řešení svozně-rozvozního problému s ohledem na fakt, že řešení metodami exaktními, zpravidla matematickými modely lineárního programování, je omezeno obtížným a někdy i neúplným definováním omezujících podmínek modelu a rostoucím množstvím podmínek v závislosti na velikosti grafu dopravní sítě, respektive v závislosti na počtu obsluhovaných hran. Řešení získána touto metaheuristikou odpovídají ve většině případů optimálním řešením, získaným exaktními metodami, nebo se liší jen minimálně. Tato metodika je implementována do SW nástroje, který umožňuje sestavovat trasy obslužných vozidel provádějících čištění a údržbu pozemních komunikací ve větších městech a průmyslových aglomeracích.

1 SOUČASNÝ STAV PROBLÉMU

Ve větších městech a průmyslových aglomeracích v ČR, zajišťují čištění a údržbu pozemních komunikací (dále ČUPK) a dalších ploch smluvně firmy, ve kterých vlastní většinový podíl město, nebo soukromé firmy bez podílu města. V některých případech se o tuto činnost dělí více firem. Například v Praze provádějí ČUPK Pražské služby a.s., kde je většinovým vlastníkem město. V Pardubicích jsou to Služby města Pardubic a.s. Tyto firmy poskytují i další služby jako například péči o zeleň, instalaci a údržbu dopravního značení, správu veřejného osvětlení, opravu pozemních komunikací ve městě, provoz sběrných dvorů atd. Na ČUPK je každý rok z rozpočtu města vyhrazena nemalá částka v řádu desítek miliónů Kč, která je v mnoha případech překročena. Rozpočet se sestavuje na základě zkušeností z předchozích let a je přímo závislý na celkové délce čištěných komunikací a velikosti čištěných ploch. Jednotlivé položky rozpočtu jsou zpravidla rozděleny podle činností prováděných při čištění města. Ve statutárním městě Pardubice se jedná o činnosti uvedené v tabulce 1.1. Z této tabulky jsou patrné i náklady na tyto činnosti a výkon v podobě vyčištěných metrů a m².

Tabulka 1.1: Celkové náklady na ČUPK ve statutárním městě Pardubice za rok 2010

činnost	výkon [m], [m ²]	[Kč/m]	jednotka	náklady [Kč]
metení vozovek	5 343 946	0,43	m	2 297 897
metení chodníků	3 995 690	0,74	m	2 956 811
strojní čištění s dopravním značením	142 909	4,00	m	571 636
blokové čištění komunikací	431 824	8,00	m ²	3 454 592
mytí chodníků	611 980	0,50	m	305 990
čištění podchodů	479 100	1,25	m	598 875
čištění cyklostezek	1 413 000	0,74	m	1 045 620
likvidace plevelů	400 000	5,00	m ²	2 000 000
splachování vozovek	87 140	0,41	m	35 727
rezerva				1 000 000
náklady na ČUPK				14 267 148
náklady včetně DPH				16 977 906

Zdroj: data Služeb města Pardubic a.s.

Podle datových podkladů z roku 2010 poskytnutých Pražskými službami a.s. byla výše nákladů vynaložených na ČUPK např. v Praze-Jižní Město 685,3 mil. Kč. Snahou každého města je tyto náklady minimalizovat. Možným řešením je vytvoření vhodné metodiky pro optimalizaci tras vozidel provádějících ČUPK, při zachování požadovaného výkonu,

a to minimalizováním bezobslužných najetých km. Předmětem optimalizace je pouze strojní čištění pozemních komunikací.

V ČR se v současné době nepoužívají žádné SW optimalizační nástroje na tvorbu tras obslužných vozidel provádějících ČUPK. Ani hlavní město Praha a Pražské služby a.s. nedisponují žádným takovýmto nástrojem. Trasy vozidel se sestavují na základě zkušeností z předchozích let metodou pokusů a omylů. Výsledné trasy obslužných vozidel nemusejí být optimální. Podmínky pro ČUPK se každým rokem mění, je to dáno například požadavky města, obyvatel, technikou, ale i vlivem zainteresovaných subjektů. Proto je nutné sestavovat plány tras obslužných vozidel na každý rok. Firmy zajišťující ČUPK se snaží nalézt vhodné řešení v podobě SW nástrojů využívajících metodiku řešení svozně-rozvozních úloh, avšak žádný takovýto SW nástroj české produkce zatím není k dispozici. V USA jsou dostupné dva SW nástroje řešící tento problém. Jedním je FleetRoute od firmy CIVIX a druhým je RouteSmart od firmy Route Smart Technologies. Pouze první z nich je dostupný i v Evropě, ale jedná se o velice drahý SW nepřizpůsobený podmínkám ČUPK v ČR. Jeho adaptace pro použití v ČR by vyžadovala určitý čas a samozřejmě finanční prostředky. Disertační práce se zabývá návrhem vhodné metodiky pro ČUPK, která se liší od metodiky použité v amerických SW produktech, a její implementací do SW nástroje.

1.1 Technologie ČUPK

Čištěné úseky a plochy se dělí podle vlastnictví, druhu a způsobu čištění. Úseky a plochy mohou být ve vlastnictví města, správy a údržby silnic příslušného kraje, nebo Ředitelství silnic a dálnic ČR. Podle druhu jde o vozovky, chodníky, podchody, cyklostezky, pěší zóny a ostatní plochy. Podle způsobu čištění jde o metení, strojní čištění, mytí chodníků, likvidaci plevelů, splachování vozovek a blokové čištění pozemních komunikací, kde probíhá čištění komunikace i chodníku včetně zajištění odtahu vozidel, které se v dané oblasti nemají nacházet. Pro bezproblémové provedení služby je zapotřebí, aby v čištěné lokalitě nebyla zaparkována žádná vozidla a netvořila tak překážku samotnému čištění. Proto se 7 dní před výkonem rozmísťují dopravní značky s informací, jaký den a v kolik hodin bude čištění probíhat. Pro řidiče je toto informace, že mají svá vozidla přeparkovat, jinak riskují odtah. Dále se provádí čištění buď levé části, pravé části, nebo obou částí komunikace.

Strojní čištění pozemních komunikací zametacími vozidly je prováděno samosběrnými zametacími vozidly. Nečistota se ukládá tíhovou silou ve sběrné skříni

nástavby a smetky jsou ukládány na skládce (pro Pardubice je tato skládka umístěna v Dražkovicích).

Strojní metení chodníků a cyklistických stezek je prováděno chodníkovými zametacími vozy s vysáváním. Čištění je prováděno samosběrnými zametacími vozidly. Protiběžné zametací talířové kartáče smetají nečistoty přímo před jedoucím vozidlem a smetené nečistoty jsou následně odsávány přes sací hubici silným proudem vzduchu do sběrného zásobníku nástavby. Oběhový vodní systém zajišťuje stejnoměrné zvlhčování smetených nečistot a snižuje množství pevných částic ve vzduchu (prašnost). Smetky jsou ukládány na skládce.

Strojní čištění (splachování-mytí) pozemních komunikací a chodníků je prováděno kropícími vozy (viz příloha 1). K mytí se používá zpravidla užitková voda z řeky.

Úklid komunálním vysavačem je prováděn ručně, komunálním vysavačem (viz příloha 2). Ruční vysavač vysává odpadky a psí exkrementy z náměstí a chodníků v centrech měst a míst, která jsou těžko přístupná. Díky svým kompaktním rozměrům může být využíván také k průběžnému čištění pozemních komunikací a chodníků.

Pohotovostní úklid je prováděn například po dopravních nehodách, při likvidaci drobných ekologických havárií, odstraňování znečištění z pozemních komunikací a odstraňování důsledků živelných katastrof.

1.2 Současná právní situace v oblasti ČUPK

Pozemní komunikace je podle **zákona o pozemních komunikacích č. 13/1997 Sb. [1]** dopravní cesta určená k užití silničními a jinými vozidly a chodci, včetně pevných zařízení nutných pro zajištění tohoto užití a jeho bezpečnosti. Může mít charakter stavby (prakticky vždy u dálnic a silnic, ve většině případů u místních komunikací), která je podle současné české právní úpravy samostatnou nemovitou věcí nezapisovanou do katastru nemovitostí, nebo se může jednat o pozemek či jeho část (typické u účelových komunikací).

V České republice se pozemní komunikace dělí na tyto **kategorie**:

- **dálnice**, určená pro rychlou dálkovou a mezistátní dopravu silničními motorovými vozidly, která je budována bez úrovnových křížení, s oddělenými místy napojení pro vjezd a výjezd, a která má směrově oddělené jízdny pásy,
- **silnice**, kterou je veřejně přístupná pozemní komunikace určená k užití silničními a jinými vozidly a chodci. Jedná se o nejtypičtější kategorii pozemních komunikací, v běžném jazyce se pro pozemní komunikace používá označení silnice,

- **místní komunikace**, kterou je veřejně přístupná pozemní komunikace, která slouží převážně místní dopravě na území obce. Místní komunikací IV. třídy může být i samostatná pěší komunikace,
- **účelové komunikace**, která slouží ke spojení jednotlivých nemovitostí pro potřeby vlastníků těchto nemovitostí nebo ke spojení těchto nemovitostí s ostatními pozemními komunikacemi nebo k obhospodařování zemědělských a lesních pozemků. Dělí se na veřejně přístupné, které mají v některých ohledech obdobný režim jako místní komunikace nebo silnice, a veřejně nepřístupné. Vyústění účelové komunikace na jiný druh komunikace se nepovažuje za křižovatku.

V České republice rozlišujeme podle **zákona o pozemních komunikacích č. 13/1997 Sb. tři třídy silnic:**

- **silnice I. třídy** je určena zejména pro dálkovou a mezistátní dopravu. Označuje se jednomístným nebo dvojmístným číslem, před nímž se někdy uvádí ještě římské číslo I oddělené lomítkem. V současné době jsou čísla 1–71. Ve stejném systému jsou číslovány i rychlostní silnice a dálnice, před jejichž číslem se vkládá písmeno R nebo D. Silnice I. třídy vystavěná jako rychlostní silnice má obdobné stavebně technické vybavení a provozní podmínky jako dálnice. Zpravidla je z hlediska provozu označena jako silnice pro motorová vozidla,
- **silnice II. třídy** je určena pro dopravu mezi okresy. Označuje se trojmístným číslem, před nímž se někdy uvádí římské číslo II oddělené lomítkem. V současné době je jejich počet přibližně 450,
- **silnice III. třídy** je určena k vzájemnému spojení obcí nebo jejich napojení na ostatní pozemní komunikace. V terénu ani v mapách se zpravidla číslem neoznačují. V úředních dokumentech a specializovaných mapách se označují čtyřmístným nebo pětimístným číslem, před nímž se někdy uvádí římské číslo III oddělené lomítkem. Číslo obvykle vychází z některé silnice II. třídy, na kterou se napojuje (např. u silnice II/128 je III/12801), výjimečně z čísla silnice I. třídy (např. u silnice I/19 je III/01924).

Obdobně se rozlišují třídy (I. až IV.) a funkční třídy A–D, a to i u místních komunikací.

Vlastníkem dálnic a silnic I. třídy je stát. Vlastníkem silnic II. a III. třídy je kraj, na jehož území se silnice nacházejí. Vlastníkem místních komunikací je obec. Silnice a místní

komunikace jsou za stanovených podmínek veřejně přístupné na základě zákonem daného práva obecného užívání (§ 19 zákona č. 13/1997 Sb.). U většiny úseků dálnic je užívání podmíněno dálničním poplatkem.

Místní komunikace je v České republice podle § 6 zákona o pozemních komunikacích č. 13/1997 Sb. označení pro kategorii pozemní komunikace, do které má silniční správní úřad zařazovat veřejně přístupné pozemní komunikace, které slouží převážně místní dopravě na území obce. Vlastníkem je podle zákona obec.

Místní komunikace se podle § 6 zákona č. 13/1997 Sb. a § 3 vyhlášky č. 104/1997 Sb. [2] rozdělují podle dopravního významu, určení a stavebně technického vybavení do těchto tříd:

- **místní komunikace I. třídy**, kterou je zejména rychlostní místní komunikace, podle prováděcí vyhlášky též dopravně nejvýznamnější sběrné komunikace ve městech,
- **místní komunikace II. třídy**, kterou je dopravně významná sběrná komunikace s omezením přímého připojení sousedních nemovitostí, která spojuje části měst navzájem nebo napojuje město nebo jeho část na pozemní komunikaci vyšší třídy nebo kategorie,
- **místní komunikace III. třídy**, kterou je obslužná komunikace ve městě nebo jiné obci běžně přístupná provozu motorových vozidel a umožňující přímou dopravní obsluhu jednotlivých objektů,
- **místní komunikace IV. třídy**, kterou je komunikace nepřístupná provozu silničních motorových vozidel nebo na které je umožněn smíšený provoz, například samostatné chodníky, stezky pro pěší, cyklistické stezky, cesty v chatových oblastech, podchody, lávky, schody, pěšiny, zklidněné komunikace, obytné a pěší zóny apod.

O zařazení pozemní komunikace do kategorie dálnice, silnice nebo místní komunikace nebo o změně zařazení rozhoduje příslušný silniční správní úřad na základě jejího určení, dopravního významu a stavebně technického vybavení. O zařazení do kategorie místní komunikace nebo vyřazení z ní rozhoduje obec, na jejímž území se komunikace nachází.

Za místní komunikace se nepovažují tzv. **průjezdny úseky silnic a dálnic**, tedy úseky dálnic a silnic vedoucí zastavěným nebo zastavitelným územím obce. Průjezdny úseky jsou

vymezeny v územně plánovací dokumentaci nebo je určí příslušný stavební úřad na návrh příslušného silničního správního úřadu a po předchozím projednání s obcí. Kritéria stanoví **§ 8 odst. 3 zákona č. 13/1997 Sb.** a **§ 4 vyhlášky 104/1997 Sb.** Umístění dopravních značek *Obec* a *Konec obce* nemusí být totožné s hranicemi průjezdního úseku silnice. ČUPK průjezdních úseků silnic a dálnic je prováděno ve většině případů smluvně službami města.

Místními komunikacemi nejsou rozuměny účelové komunikace. Dopravní značky ani běžné mapy či plány pro veřejnost však neoznačují ani nerozlišují, které pozemní komunikace jsou místními komunikacemi a které účelovými komunikacemi. **Účelovou komunikací** je zpravidla taková komunikace, kterou nevlastní obec (ani městská část nebo městský obvod) ani nejde o silnici nebo dálnici. Účelové komunikace však, stejně jako místní komunikace IV. třídy, nepodléhají speciální evidenci podle **§ 5 vyhlášky č. 104/1997 Sb.**, a proto rozlišení mezi nimi nemusí být zřejmé ani z oficiálních mapových a evidenčních podkladů. Směrodatné by podle zákona mělo být, zda silniční správní úřad rozhodl o zařazení do kategorie místní komunikace – pokud o zařazení komunikace do kategorie nerozhodl, může být taková komunikace jen účelovou komunikací.

Zásadní praktický rozdíl mezi místní a účelovou komunikací je ten, že zatímco vyústění účelové komunikace na jinou komunikaci se nepovažuje za křižovatku, vyústění místní komunikace (např. pěší komunikace IV. kategorie – pěšiny či schodiště, pokud nejde o označenou pěší nebo obytnou zónu) podle **zákona č. 361/2000 Sb. [3]**, o provozu na pozemních komunikacích a o změnách některých zákonů (zákon o silničním provozu), křižovatkou je.

Chodník je část pozemní komunikace nebo samostatná pozemní komunikace, která slouží chodcům k přesunu po délce komunikace. Chodníkem zpravidla bývají vybaveny místní komunikace, případně průjezdní úseky silnic, v zastavěných částech obcí, a to buď po jedné, nebo po obou stranách.

Podle **§ 12 odst. 4 zákona č. 13/1997 Sb., o pozemních komunikacích**, jsou přilehlé chodníky, chodníky pod podloubími, podchody a zařízení pro zajištění a zabezpečení přechodů pro chodce součástí místních komunikací, pokud nejsou samostatnými místními komunikacemi. Jejich vlastníkem je vždy obec.

Podle **§ 34 zákona č. 128/2000 Sb. [4]**, o obcích, jsou veřejně přístupné chodníky veřejným prostranstvím bez ohledu na to, kdo je vlastníkem.

Podle **§ 12 odst. 4 zákona č. 13/1997 Sb., o pozemních komunikacích** je **stezka pro cyklisty** (cyklostezka, cyklistická stezka) pozemní komunikace nebo její jízdní pás,

vyhrazený dopravní značkou, pro jízdu na jízdním kole. Je určena pouze pro cyklistickou dopravu. Automobilová a motocyklová doprava je z ní vyloučena. Pravidla silničního provozu též povolují užití cyklostezky například jezdcům na kolečkových bruslích apod.

Pěší zóna je ulice uzpůsobená tak, že na ní nejsou jízdní pruhy, celá šířka pozemní komunikace je určena pro chodce a ulice je označena příslušnou dopravní značkou. Dopravní značka označující pěší zónu může povolit vjezd vybraným druhům vozidel, nebo dopravní provoz v omezeném období. Rychlost vozidel v pěší zóně nesmí překročit 20 km/h.

Termíny pro jarní a podzimní **čištění pozemních komunikací** I. až III. třídy jsou dány **vyhláškou č. 104/1997 Sb., § 47**, stejná ustanovení platí i pro místní komunikace. ČUPK jednotlivých místních komunikací (I.–IV. třídy) a ostatních ploch probíhá několikrát za dané období (dáno požadovanou četností), zpravidla od dubna do listopadu. V období od prosince do března je prováděna zimní údržba pozemních komunikací.

2 CÍLE DISERTAČNÍ PRÁCE

Základem práce je podrobná analýza současných metod řešení svozně-rozvozního problému s obsluhou hran, popis jednotlivých typů úloh, jejich časová a výpočetní složitost a vhodnost jejich použití k optimalizaci na městské infrastruktuře (kapitola 3).

Disertační práce se zabývá aplikací těchto metod v prostředí GIS a demonstrací obecných principů začlenění řešení svozně-rozvozních úloh do prostředí GIS (kapitola 4).

Cílem disertační práce je návrh modelu, na kterém jsou popsány principy začlenění metod řešení svozně-rozvozního problému do prostředí GIS a který bude nástrojem na jeho řešení. Model je aplikován na problém čištění pozemních komunikací a veřejných prostranství větších měst a průmyslových aglomerací, kde se doposud optimalizační metody operačního výzkumu používají minimálně. Druhým, neméně významným cílem disertační práce je vytvoření uceleného systému algoritmů optimalizace s využitím nejnovějších možností dopravní telematiky a GIS (kapitola 4).

Hypotéza: Navržení vhodné metodiky na řešení svozně-rozvozního problému s obsluhou hran a její aplikace v softwarovém nástroji s podporou GIS povede k racionalizaci stávajícího řešení problému čištění a údržby pozemních komunikací ve větších městech.

3 METODY ZKOUMÁNÍ

Návrhu vhodné metodiky řešící optimalizaci tras obslužných vozidel, které provádějí čištění a údržbu pozemních komunikací ve větších městech a průmyslových aglomeracích, předchází podrobná analýza současných metod řešení svozně-rozvozního problému s obsluhou hran. V analýze jsou popsány jednotlivé typy úloh a jejich časová složitost z pohledu teorie grafů a matematického programování. Matematické značení je možné nalézt v [5].

3.1 Teoretický aparát metod zkoumání

V této podkapitole je popsán teoretický aparát, který je využit v analýze metod řešících svozně-rozvozní úlohy s obsluhou hran (dále ARP – *Arc Routing Problem*) a v metodice sestavování tras obslužných vozidel provádějících ČUPK. Jedná se o metody teorie grafů, matematického programování a teorii složitosti.

3.1.1 Aparát teorie grafů

V tomto oddílu je citováno z [6]. ARP je obecně definován na **smíšeném grafu** $G = (V, E, p)$, $E = X \cup Y$, kde prvky množiny V nazýváme **vrcholy grafu** G , prvky množiny X **neorientovanými hranami** grafu G , prvky množiny Y **orientovanými hranami**, nebo také **or-hranami** grafu G a p **incidencí grafu** G . Pokud nebude řečeno jinak, budou hrany a polyhrany obecně označovány písmeny h a l , vrcholy písmeny v a u .

O grafu G hovoříme jako o **neorientovaném grafu** v případě, že $Y = \{\emptyset\}$, o **orientovaném grafu (digrafu)**, pokud $X = \{\emptyset\}$ a o smíšeném pokud $X \neq \{\emptyset\} \wedge Y \neq \{\emptyset\}$. Obvykle, pokud nebude řečeno jinak, počet vrcholů grafu budeme značit n . Počet hran grafu budeme značit m , počet neorientovaných hran v grafu m_x a počet orientovaných hran v grafu m_y .

Incidence p přiřazuje každé hraně grafu uspořádanou nebo neuspořádanou dvojici vrcholů. Pokud je hrana neorientovaná, jedná se o neuspořádanou dvojici, pokud je hrana orientovaná, jedná se uspořádanou dvojici. Platí-li pro incidenci hrany $h \in X$ $p(h) = (u, v)$, nebo $h \in Y$ $p[h] = [u, v]$, hovoříme, že hrana h inciduje s vrcholem u i s vrcholem v .

Vrcholy u, v nazýváme **přilehlými vrcholy**, pokud $\exists h \in X : p(h) = (u, v)$. Budeme říkat, že tyto **vrcholy** jsou **sousední**, nebo také že spolu sousedí. Hrany, incidující se stejným vrcholem se nazývají **přilehlé hrany (sousední hrany)**.

Vrcholy u, v nazýváme **krajními vrcholy** polyhrany h . Polyhranám (u, v) , resp. $[u, v]$, není incidence p přiřazována a o vrcholech u, v hovoříme pouze jako o krajních vrcholech polyhrany, resp. počátečním a koncovém vrcholu polyhrany.

Protože každá hrana v neorientovaném grafu může být vyjádřena jako neuspořádaná dvojice sousedních vrcholů, je zápis $p(h) = (u, v)$ ekvivalentní se zápisem $p(h) = (v, u)$. U or-hran tato ekvivalence neplatí.

Ve všeobecnosti incidence p nemusí být prostým zobrazením, ale může i více různým hranám $h_i \in X$, nebo $h_i \in Y$, $i = 1, 2, \dots, k$ přiřadit stejnou dvojici (u, v) , resp. $[u, v]$. Tyto hrany se nazývají **násobné hrany (multihrany)**. Pokud připustíme, že incidence přiřadí některé hraně h dvojici (u, u) , resp. $[u, u]$, potom takovéto hraně říkáme **smyčka**. Grafy, obsahující násobné hrany se nazývají **multigrafy**. **Nesouměrnou hranou** nazýváme hranu, která má v každém směru odlišné ohodnocení.

Graf $G = (V, E, p)$ nazýváme **vrcholově/hranově ohodnoceným grafem**, pokud existuje pro vrchol $v \in V$ funkce $c(v)$ /pro hranu $h \in E$ funkce $c(h)$, resp. $c(u, v)$, nebo pro or-hranu funkce $c[h]$, resp. $c[u, v]$ ve vztahu ke krajním bodům u, v hrany h , která přiřadí každému vrcholu $v \in V$ /hraně $h \in E$ nezáporné číslo vyjadřující určitou kvantitativní nebo kvalitativní vlastnost vrcholu/hrany. Toto nezáporné číslo budeme nazývat **ohodnocením vrcholu/hrany**. Grafy mohou být vrcholově i hranově ohodnocené. Pokud je ohodnocení hrany při obsluze jiné než při průchodu hranou, označíme ohodnocení při obsluze $c^R(h)$ a při průchodu $c(h)$.

Nechť pro dvojici vrcholů u, v grafu $G = (V, E, p)$ existuje střídavá posloupnost vrcholů a hran:

$$T = \{u_0, h_1, u_1, h_2, u_2, \dots, u_{n-1}, h_n, u_n\}, \quad (3.1)$$

kde $h \in E$, $p(h_i) = (u_{i-1}, u_i)$, resp. $p[h_i] = [u_{i-1}, u_i]$, pro $i = 1, 2, \dots, n$,

$u_i \in V$ pro $i = 1, 2, \dots, n$,

$u_0 = u, u_n = v$,

potom T nazýváme **sledem**, resp. **orientovaným sledem** grafu G mezi vrcholy u, v . Sled, resp. orientovaný sled, ve kterém se neopakuje žádná hrana, resp. or-hrana, nazýváme **tahem**, resp. orientovaným tahem. Tah, resp. orientovaný tah, ve kterém se neopakuje žádný vrchol, nazýváme **cestou**, resp. orientovanou cestou (dráhou). Uzavřenou cestu (tj. cestu, ve které počáteční a koncový vrchol jsou totožné) nazýváme **kružnicí**. Uzavřenou orientovanou cestu nazýváme **cyklus**.

Označme M množinu všech cest $m(u, v)$, resp. $m[u, v]$, z vrcholu u do vrcholu v grafu $G = (V, E, p)$.

Nejkratší (minimální) cestou, resp. **polyhranou**, mezi vrcholy u, v v grafu $G = (V, E, p)$ je cesta $m^*(u, v) \in M$, resp. $m^*[u, v] \in M$, pro kterou platí:

$$\sum_{h \in m^*(u, v)} c(h) = \min_{m(u, v) \in M} \left\{ \sum_{h \in m(u, v)} c(h) \right\}. \quad (3.2)$$

Délku minimální (nejkratší) cesty, resp. **ohodnocení polyhrany** mezi vrcholy u, v v grafu G budeme značit $d(u, v)$, resp. $d[u, v]$ a je rovna $d(u, v) = |m^*(u, v)|$, resp. $d[u, v] = |m^*[u, v]|$.

Množinu neorientovaných hran (u, v) incidujících s vrcholem u označujeme $X(u)$. **Množinu or-hran $[u, v]$, vycházejících z vrcholu u** označujeme $Y^+(u)$. **Množinu or-hran $[v, u]$ vcházejících do vrcholu u** označujeme $Y^-(u)$.

Počet hran incidujících s vrcholem v nazýváme **stupeň vrcholu** a označujeme $st(v)$.

Maticí minimálních vzdáleností, nebo také **distanční maticí** grafu $G = (V, E, p)$ rozumíme matici $D = (d_{ij})_{i, j=1}^n$, kde n je počet vrcholů grafu a jednotlivé prvky matice d_{ij} jsou rovny délce minimální cesty mezi vrcholy v_i a v_j , $d_{ij} = d(v_i, v_j)$. Prvky této matice na hlavní diagonále jsou vždy nulové. O **symetrické distanční matici** hovoříme, pokud jsou její prvky souměrné podle hlavní diagonály, tedy pokud jsou všechny hrany grafu neorientované.

Množinu R , $R \subseteq E$, nazýváme **množinou hran obsluhy**. Prvky této množiny tvoří hrany, které mají být obslouženy na grafu G a mají nezáporný celočíselný **požadavek obsluhy** $w(h)$. Ve vztahu k vrcholům incidujícím s h je požadavek označen w_{ij} , pro $p(h) = (v_i, v_j)$, $p[h] = [v_i, v_j]$. Množinu V_R nazýváme **množinou vrcholů obsluhy** na grafu G . Mohutnost množiny R budeme označovat m_R a mohutnost množiny V_R budeme

označovat n_R . Mohutnost **množiny neorientovaných hran obsluhy** X_R budeme označovat m_{XR} . Mohutnost **množiny or-hran obsluhy** Y_R budeme označovat m_{YR} . Přičemž platí $R = X_R \cup Y_R$.

Pokud pro hranu $h \in Y$ platí $p[h] = [u, v]$, potom vrchol u nazýváme **výchozí (počáteční) vrchol** a vrchol v **koncový vrchol** hrany h . Pokud nebude řečeno jinak, budou tyto vrcholy značeny $v_{poč}[h]$ a $v_{kon}[h]$.

I-tou trasou na smíšeném grafu $G = (V, E, p)$ rozumíme množinu hran T_i^R obsahující pouze hrany obsluhy, mezi kterými jsou implicitně předpokládány minimální cesty (jsou spojeny polyhranami). **I-tou trasou na neorientovaném grafu** $G = (V, X, p)$ rozumíme množinu hran $T_i^{X_R}$ obsahující pouze hrany obsluhy, mezi kterými jsou implicitně předpokládány minimální cesty. **Množinu** všech **tras** na grafu $G = (V, E, p)$ označme $T^R = \{T_1^R, \dots, T_{m_R}^R\}$, na grafu $G = (V, X, p)$ označme tuto množinu $T^{X_R} = \{T_1^{X_R}, \dots, T_{m_{XR}}^{X_R}\}$. **Úplnou trasou** T_1^R budeme nazývat trasu pokrývající všechny hrany obsluhy grafu $G = (V, E, p)$. **Délku i-té trasy** budeme označovat $c(T_i^R)$ nebo $c(T_i^{X_R})$.

Graf nazýváme **souvislým**, pokud mezi libovolnou dvojicí jeho vrcholů v_i, v_j existuje alespoň jedna cesta.

Strom je souvislý graf, který neobsahuje jako podgraf kružnici¹.

Kostra grafu T_G je podgraf grafu $G = (V, X, p)$, jehož množina vrcholů je totožná s množinou vrcholů grafu G , množina hran grafu je podmnožinou množiny hran grafu G , a tento podgraf je stromem².

Neorientovaný graf $G = (V, X, p)$, jehož každý vrchol má sudý stupeň, nazýváme **eulerovským grafem** (dále E -grafem), nebo také unicursálním grafem, E -graf lze vyjádřit jako sjednocení soustavy vzájemně hranově disjunktních kružnic:

$$G = \mathbf{U} K_i, K_i \mathbf{I} K_j = \{\emptyset\} \text{ pro } i \neq j. \quad (3.3)$$

Graf $G = (V, E, p)$ nazýváme **sudým grafem**, pokud pro každý vrchol grafu platí, že počet hran, které s tímto vrcholem incidují, je sudý. Dále jako **graf symetrický**, pokud pro

¹ Stromy je možné definovat také jako souvislé grafy, kde pro libovolnou dvojici vrcholů $v_i, v_j \in V$ existuje právě jedna cesta $m(v_i, v_j)$.

² Protože množina vrcholů kostry grafu je shodná s množinou vrcholů původního grafu, je kostra faktorovým podgrafem.

každý vrchol grafu platí, že počet vycházejících or-hran do vrcholu je roven počtu vycházejících or-hran z tohoto vrcholu. Pokud je graf zároveň sudý a symetrický nazýváme tento **graf vyváženým**.

Orientovaný graf $G = (V, Y, p)$ nazýváme E -grafem, pokud je grafem sudým. Smíšený graf $G = (V, E, p)$ je E -grafem, pokud je vyvážený. Navíc pro libovolnou množinu vrcholů $S \subseteq V$ v tomto grafu musí platit, že rozdíl mezi počtem orientovaných hran z S do $V \setminus S$ a počtem orientovaných hran z $V \setminus S$ do S je menší nebo roven počtu neorientovaných hran spojujících S a $V \setminus S$. Tato podmínka je také někdy nazývána podmínkou „vyváženosti množin“.

Orientovaný graf, který neobsahuje cyklus, nazýváme **acyklický graf**.

Bipartitním grafem nazýváme graf $G = (V, E, p)$, jehož množinu vrcholů je možné rozdělit na dvě disjunktní množiny V_1, V_2 tak, že žádné dva vrcholy ze stejné množiny nejsou spojeny hranou, $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{\emptyset\}$, $\forall (u, v) \in E : u \in V_1, v \in V_2$.

Eulerovský tah (E -tah), který je výsledkem řešení problému čínského pošťáka, může, ale nemusí začínat a končit v témže vrcholu a obsahuje každou hranu právě jednou. Podle toho hovoříme o **otevřeném**, nebo **uzavřeném E -tahu**. Nutnou a zároveň postačující podmínkou k tomu, aby konečný souvislý graf $G = (V, X, p)$ mohl být sestrojen jedním uzavřeným E -tahem je, aby byl E -grafem. Uzavřený E -tah je popsán vektorem $(v_1, h_1, v_2, \dots, h_{n-1}, v_n)$, kde hrana $p(h_i) = (v_i, v_{i+1})$ náleží do množiny X , nebo hrana $p[h_i] = [v_i, v_{i+1}]$ náleží do množiny Y , pro $i = 1, \dots, n-1$ a $v_n = v_1$. Řešením problému čínského pošťáka na orientovaném, neorientovaném nebo smíšeném grafu je získán E -tah.

Podgrafy G_i , $i = 1, 2, \dots, k$ grafu G nazýváme **komponentami**, pokud jsou souvislé a platí:

$$\bigcup_{i=1}^k G_i = G, \quad (3.4)$$

$$G_i \cap G_j = \emptyset, i \neq j. \quad (3.5)$$

Mostem nazýváme hranu, jejímž odstraněním se graf G rozpadne na dvě komponenty.

Kompletním (úplným) grafem nazýváme graf, ve kterém je každý vrchol přilehlý k ostatním vrcholům grafu. Tyto grafy označujeme $K_{|n|}$, kde n je mohutnost množiny

vrcholů V . Kompletní graf je v řešení ARP používán k určení hran perfektního minimálního párování, kde množinu vrcholů V_L kompletního grafu $K_{|n_{vz}|}$ tvoří vrcholy lichého stupně grafu $G = (V, X, p)$, na kterém chceme určit E -tah, a množina hran X_L obsahuje hrany, jejichž ohodnocení je rovno délce minimální cesty v G mezi těmito vrcholy. Hrany určené minimálním párováním označujeme X_p .

Kapacitu s -tého vozidla, označujeme Q_s , $s = 1, \dots, k$. Pokud mají všechna vozidla vozového parku totožnou kapacitu, označujeme ji Q .

Maximální **dojezdovou vzdálenost vozidla** označujeme L .

Depem rozumíme místo na síti, kde je umístěno středisko obsluhy³. Pokud nebude řečeno jinak, bude vrchol s depem označován v_1 .

Deponovacím místem rozumíme místo na síti, kde dochází k vykládce vozidla.

3.1.2 Aparát matematického programování

Tvorbu tras obslužných vozidel v ARP lze řešit také jako úlohu matematického programování. V tomto oddílu je popsán základní aparát lineárního programování, který do matematického programování patří.

Lineární programování (dále LP – *Linear Programming*) je způsob řešení problémů, jehož základem je matematické modelování, využívající pro řešení modelů exaktní matematické postupy. Aby bylo možné ARP řešit metodami LP, musí být matematicky popsán. To znamená z verbální formulace problému sestavit matematický model, který zahrnuje kritérium (účelovou funkci) a soustavu omezení (omezující podmínky). Kritériem u ARP je minimalizace součtu ohodnocení hran (délek úseků pozemních komunikací) v trase obslužného vozidla. Omezením může být například kapacita vozidel, dojezdová vzdálenost vozidel, selekce hran obsluhy atd.

Matematický model úlohy lineárního programování je tvořen lineární **účelovou funkcí** (3.6) a soustavou lineárně nezávislých **omezujících podmínek** (lineárních rovnic nebo nerovnic (3.7)–(3.9)).

Matematické modely ARP mají následující obecný tvar (pro přehlednost jsou podmínky (3.7) a účelová funkce (3.6) rozepsány vpravo jako součet jednotlivých koeficientů a strukturních proměnných).

³ Střediskem obsluhy je například čerpací stanice pohonných hmot, skládka posypového materiálu atd.

$$\text{Min. } \sum_{(v_i, v_j) \in X} c_{ij} x_{ij} \quad \text{Min. } f(x) = c_{11}x_{11} + \dots + c_{1l}x_{1l} + c_{21}x_{21} + \dots + c_{2l}x_{2l} + \dots + c_{kl}x_{kl} \quad (3.6)$$

pro $i = 1, \dots, k$ a $j = 1, \dots, l$

Za podmíněk:

$$\sum_{(v_i, v_j) \in X} a_{ij} x_{ij} \left\{ \begin{array}{l} < \\ = \\ > \end{array} \right\} b \quad \left. \begin{array}{l} a_{11}x_{11} + a_{12}x_{12} + \dots + a_{1l}x_{1l} \left\{ \begin{array}{l} < \\ = \\ > \end{array} \right\} b \\ a_{21}x_{21} + a_{22}x_{22} + \dots + a_{2l}x_{2l} \left\{ \begin{array}{l} < \\ = \\ > \end{array} \right\} b \\ \mathbf{M} \\ a_{k1}x_{k1} + a_{k2}x_{k2} + \dots + a_{kl}x_{kl} \left\{ \begin{array}{l} < \\ = \\ > \end{array} \right\} b \end{array} \right\} ((v_i, v_j) \in X) \quad (3.7)$$

$$x_{ij} \geq 0 \text{ celočíselně} \quad x_{11}, x_{12}, \dots, x_{kl} \geq 0, x_{ij} \in N \quad ((v_i, v_j) \in X) \quad (3.8)$$

$$\text{nebo } x_{ij} \in \{0,1\} \quad x_{11} \equiv \{0,1\}, \dots, x_{kl} \equiv \{0,1\} \quad ((v_i, v_j) \in X) \quad (3.9)$$

V tomto obecném zápisu jsou x_{ij} nezávislé **strukturní proměnné** úlohy LP. V ARP představují strukturní proměnné počty průchodů vozidla hranou (v_i, v_j) v trase vozidla, nebo počet přidaných hran do grafu, aby byl graf E -grafem. **Koeficienty účelové funkce** c_{ij} představují ohodnocení hran (v_i, v_j) , které náleží do množiny hran grafu G . **Strukturní koeficienty** podmínek (3.7) jsou libovolná reálná čísla, $a_{ij} \in R$. Pravou stranu podmínek nazýváme **požadavkem** nebo **omezením**, b je libovolné nezáporné číslo, $b \in R^+$. Podmínky (3.7) jsou lineární rovnice nebo nerovnice a nazýváme je **základním** nebo **funkčním omezením**. **Podmínky nezápornosti** (3.8) definují proměnné x_{ij} na oboru celých nezáporných čísel, $x_{ij} \in N$, poté hovoříme o **úloze celočíselného LP**. Pokud jsou místo podmínek (3.8) v matematickém modelu úlohy LP **bivalentní podmínky** (3.9), proměnná x_{ij} může nabývat jedné ze dvou hodnot 0 a 1, hovoříme o **bivalentní úloze LP**. Vektor

$x = (x_{11}, x_{12}, \dots, x_{kl})$, jehož souřadnice splňují (3.7)–(3.9) nazýváme **přípustným řešením** úlohy LP.

Dále je v tomto oddílu citováno z [7]. Uspořádanou n -tici reálných čísel (a_1, a_2, \dots, a_n) budeme nazývat **n -rozměrným vektorem** a označovat písmenem a . Množinu všech n -rozměrných reálných vektorů spolu s obvyklými operacemi sčítání vektorů a násobení vektorů reálným číslem nazýváme **n -rozměrný reálný vektorový prostor** a označujeme E_n .

Konvexní kombinací vektorů a_1, a_2, \dots, a_n , náležejících vektorovému prostoru E_n , budeme nazývat takovou jejich lineární kombinaci $a_1 a_1 + a_2 a_2, \dots, a_n a_n$, pro kterou platí $a_i \geq 0$, kde $i = 1, \dots, n$ a $\sum_{i=1}^n a_i = 1$. Pokud jsou d a e dva body z E_n , poté množinu jejich lineárních kombinací $\{ad + (1-a)e : a \in \langle 0, 1 \rangle\}$ nazýváme úsečkou s koncovými body d a e .

Nechť U je nějaká podmnožina E_n . Množinu U budeme nazývat **konvexní množinou**, pokud pro každé dva body $d, e \in U$ náleží do U každá jejich konvexní kombinace. Bod konvexní množiny U , který nelze vyjádřit jako konvexní kombinaci dvou jiných různých bodů této množiny, nazýváme **krajním bodem** konvexní množiny U . Konvexní omezenou uzavřenou množinu U , $U \subset E_n$, mající pouze konečný počet krajních bodů, nazýváme **konvexní polyedr**. Každý bod obalu konvexního polyedru můžeme vyjádřit jako konvexní kombinaci jeho krajních bodů.

Množina přípustných řešení, kterou vymezují omezující podmínky, je v úloze LP konvexní množinou v E_n , popřípadě konvexním polyedrem. V jednom z krajních bodů této množiny se nachází optimální řešení úlohy LP. Tento bod je určen právě jedním společným bodem účelové funkce a konvexního obalu množiny přípustných řešení. Všechny body uvnitř konvexní množiny jsou přípustnými řešeními.

Řešit úlohu LP znamená hledat takový vektor x_o , který minimalizuje (3.6) a splňuje (3.7)–(3.9). Vektor x_o nazýváme **optimálním řešením** úlohy LP. V ARP je minimalizována celková suma délek hran v trase obslužného vozidla.

Bazickým řešením soustavy (3.7) nazýváme takový vektor x , který je řešením této soustavy a volené proměnné vektoru jsou nuly, přičemž vektory se strukturními koeficienty, odpovídající nenulovým proměnným, jsou lineárně nezávislé.

Simplexová metoda (G.B. Dantzig, 1947) je iterativní metoda pro nalezení optimálního řešení úlohy LP, která používá následující postup: nejprve je určeno výchozí

bazické řešení. V každé iteraci je zkoumáno, zdali je bazické řešení optimální. Pokud ano, algoritmus končí, pokud ne, nalezne nové bazické řešení, kterému odpovídá menší, nebo v krajním případě stejná hodnota účelové funkce než v předchozím případě, a postup se opakuje. Tím, že se metoda nevrací k horším bazickým řešením, než je právě zkoumané, se počet zkoumaných bazických řešení podstatně redukuje. Přejchod od jednoho bazického řešení k dalšímu se děje pomocí výměny jednoho bazického vektoru.

Celá řada praktických problémů týkajících se optimalizace může být modelována a řešena pomocí **celočíselného** nebo také **diskrétního lineárního programování** (ILP – *Integer Linear Programming*). Tato úloha se od úlohy běžného lineárního programování LP liší v omezení strukturních proměnných na celá čísla. Pokud mohou některé strukturní proměnné nabývat i neceločíselných hodnot, poté úlohu nazýváme **smíšeným celočíselným programováním** (MIP – *Mixed Integer Programming*). Pokud by byl výsledek MIP zaokrouhlen, nebylo by zaručeno, že výsledné řešení bude optimální a také přípustné. Hlavní nevýhodou ILP je časová složitost algoritmu řešícího tuto úlohu. Zatímco úloha LP je řešitelná v polynomiálním čase, úloha ILP je tzv. NP-těžká (*NP-hard*), tzn., není znám polynomiální algoritmus. Mezi nejznámější metody řešení obecné úlohy ILP patří **metody sečných nadrovin, výčtové metody a metody větví a mezí**.

Metodou sečných nadrovin nazýváme postup řešení, který je založen na opakovaném řešení úlohy LP a jejímž výstupem je přípustné celočíselné řešení úlohy ILP. Výpočet je prováděn iterativně tak, že je v každém kroku přidána další omezující podmínka zužující oblast přípustných řešení. Každá nová omezující podmínka je přidána, pokud:

- optimální řešení nalezené pomocí LP se stane nepřípustným,
- přidáním nové podmínky se žádné přípustné celočíselné řešení v předchozím kroku nestane nepřípustným.

Vzniklý ILP program je vždy znovu řešen jako úloha LP. Proces je opakován, dokud není nalezeno přípustné celočíselné řešení. Konvergence takového algoritmu potom závisí na způsobu přidávání omezujících podmínek. Mezi nejznámější metody patří **Gomoryho řezy** a Dantzigovy řezy.

3.1.3 Složitost algoritmů

Součástí této práce je analýza metod řešících APR. U každé z těchto metod je uvedena její časová složitost, která je jedním ze dvou hlavních parametrů určujících jejich vhodnost

pro řešení ARP. Druhým parametrem je kvalita výsledného řešení v podobě délek tras obslužných vozidel. V tomto oddílu jsou popsány pojmy z teorie složitosti, které jsou citovány z [8] a jsou součástí dalšího textu této práce.

Úlohu chápeme jako obecně zadaný problém obsahující parametry. **Instancí** úlohy rozumíme konkrétní zadání všech parametrů, které úloha obsahuje. **Řešení** je matematický objekt, který vyhovuje verbálnímu zadání úlohy. **Algoritmus** pro řešení úlohy je postup, který pro každou instanci dané úlohy najde její řešení. Říkáme, že algoritmus **řeší úlohu**, pokud se algoritmus po konečném počtu kroků zastaví na každé instanci úlohy.

Funkce $t_{worst}(n)$ udává čas, který stačí k vyřešení každé instance o velikosti n . Tento přístup nazýváme **analýzou nejhoršího případu** (*worst case analysis*), a to proto, že udává čas, který je potřebný i pro ty nejsložitější instance, a to i v případě, že nejsložitější instance může být velmi nepravděpodobná a řešení trvá kratší dobu. Funkce $t_{aver}(n)$ udává průměrný čas, který je třeba k vyřešení instance velikosti n , poté hovoříme o **průměrné složitosti** (*average complexity*), která říká, s jakou dobou můžeme v průměru počítat, chceme-li vyřešit instanci o velikosti n . Místo přesného počítání funkce $t_{worst}(n)$ popisujeme asymptotický růst funkce $t_{worst}(n)$, kterým odhadujeme chování této funkce pro velká n až na multiplikační konstantu.

Nechť f a g jsou funkce z množiny přirozených čísel N do množiny N . Říkáme, že funkce f je $O(g)$, pokud existuje $n_0 \in N$ a $c > 0$ tak, že pro všechna $n \geq n_0$ je $f(n) \leq c g(n)$. Intuitivně to znamená, že funkce f pro dostatečně velká n neroste rychleji než násobek funkce g .

Funkce f je $o(g)$, pokud pro každé $c \in N$ existuje $n_0 \in N$ tak, že pro všechna $n \geq n_0$ je $f(n) \leq \frac{1}{c} g(n)$. Toto tvrzení je silnější než předchozí. Vztah f je $o(g)$ intuitivně říká, že funkce f roste pomaleji než libovolně malý násobek funkce g .

Funkce f je $\Omega(g)$, pokud g je $O(f)$. Jinými slovy, funkce f je $\Omega(g)$, pokud existuje $n_0 \in N$ a $c > 0$ tak, že pro všechna $n \geq n_0$ je $f(n) \geq \frac{1}{c} g(n)$.

Funkce f je $\Theta(g)$, pokud je současně $O(g)$ a $\Omega(g)$.

Říkáme, že **algoritmus má časovou složitost** $O(f(n))$, pokud funkce $t_{worst}(n)$ je $O(f(n))$. Algoritmus má **průměrnou složitost** $O(f(n))$, pokud $t_{aver}(n)$ je $O(f(n))$.

Říkáme, že úloha má **horní odhad složitosti** $f(n)$, pokud existuje algoritmus, který řeší úlohu a má časovou složitost $O(f(n))$.

Říkáme, že **algoritmus** je **optimální** pro danou úlohu, pokud neexistuje algoritmus, který by řešil danou úlohu a použil by, v nejhorším případě, méně základních operací (ve smyslu asymptotického popisu růstu funkcí).

Nedeterministický algoritmus (stochastický) nazýváme algoritmus, který v některých krocích může volit z několika možných dalších kroků. Nedeterministický algoritmus může při stejném vstupu poskytovat rozdílné výsledky.

Říkáme, že **algoritmus** je **polynomiální**, pokud je jeho časová složitost $O(n^k)$ pro libovolnou konstantu $k \in \mathbb{N}$. Naproti tomu říkáme, že algoritmus má **exponenciální časovou složitost**, pokud jeho časová složitost není polynomiální. Příkladem jsou algoritmy s časovou složitostí $O(n!)$, $O(2^n)$ atd.

Třída P je třída všech rozhodovacích úloh, pro něž existuje polynomiální algoritmus, který řeší úlohu. Tento algoritmu má složitost $O(p(n))$ pro libovolný polynom $p(n)$. Říkáme, že **úloha** je **P-těžká**.

Ne všechny rozhodovací úlohy náleží do třídy P. Například pro kapacitně omezenou verzi ARP není dosud znám polynomiální algoritmus, který by řešil tuto úlohu.

Třída NP je třída rozhodovacích úloh, pro něž existuje nedeterministický algoritmus pracující v polynomiálním čase. Říkáme, že **úloha** je **NP-těžká**.

Rozhodovací úloha je **NP-úplná** (*NP-complete*), pokud je NP-úlohou. Každá NP-úloha se na NP-úplnou polynomiálně redukuje. Třidu všech NP-úplných úloh označujeme NPC. Tedy, NP-úplné jsou ty úlohy, které jsou nejtěžší mezi všemi NP-úlohami.

3.2 Analýza metod pro výpočet svozně-rozvozních úloh s obsluhou hran

V této podkapitole je provedena podrobná analýza současných metod a trendů řešení ARP. Na základě analýzy je v této práci navržena metodika pro určování optimálních tras vozidel provádějících ČUPK ve větších městech a průmyslových aglomeracích v podmínkách ČR.

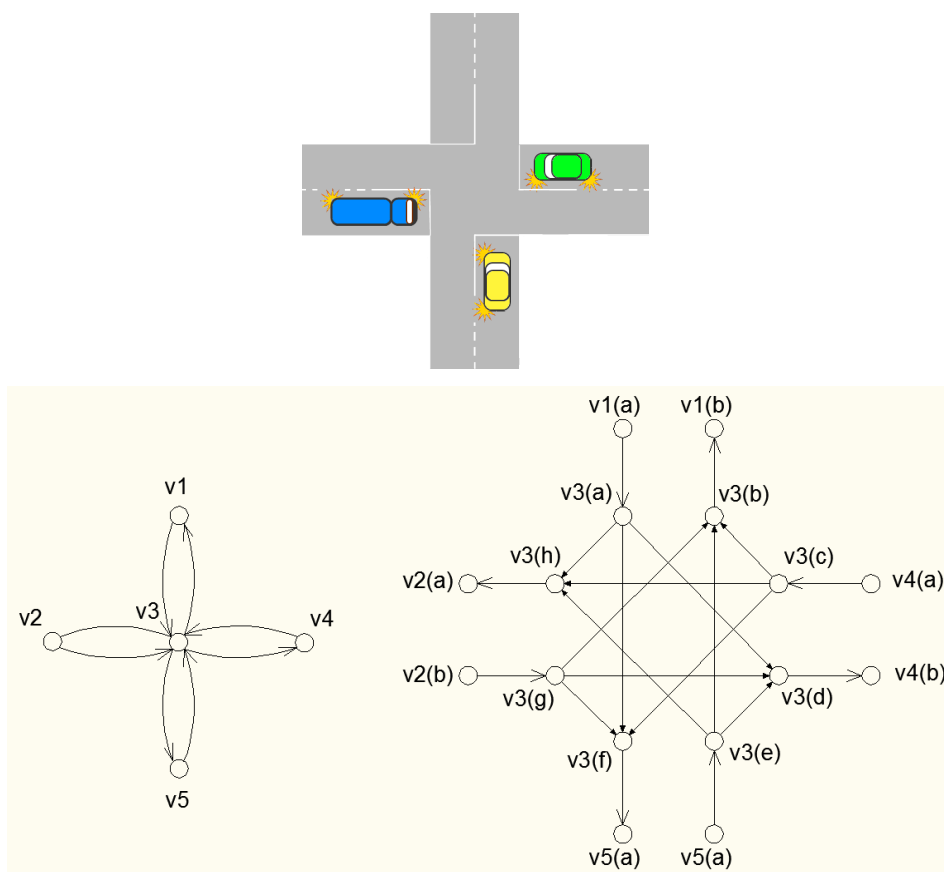
Řešení ARP je základem pro řešení mnoha praktických aplikací, ve kterých jsou komunikace či ulice obsluhovány z důvodu údržby a čištění komunikací, svozu odpadu, rozvozu zásilek, přemísťování zásilek, svozu školním autobusem, výběru parkovacích automatů, odečítání elektroměrů, provádění revizí plynových a tlakových zařízení atd. Řešením ARP se rozumí stanovení optimálních tras obslužných vozidel na síti pozemních

komunikací. Na realizaci svozně-rozvozních operací se každý rok utratí nemalé částky, přitom potenciál úspor získaných optimalizacemi je obrovský. Řada těchto aplikací nemůže být formulována jako čistý ARP, protože jsou definovány dalšími omezujícími podmínkami a mají své charakteristické rysy. Metodologie řešení je často uzpůsobována dané situaci, a pokud je aplikována v jiné souvislosti, vyžaduje si určitou modifikaci. Síť komunikací, na které je daný problém řešen, je ve zjednodušené formě reprezentována grafem sítě. Tento graf je jeden ze vstupů řešení. Zda je problém definován na orientovaném, neorientovaném, nebo smíšeném grafu záleží na topologii sítě pozemních komunikací, kterou reprezentují, a na operační politice zainteresovaných subjektů.

Jednosměrné komunikace jsou v grafu zpravidla reprezentovány orientovanými hranami a obousměrné komunikace neorientovanými hranami. Pokud je nutné ve stejný čas obsloužit obě strany komunikace, jako je tomu například u čištění komunikací nebo svozu odpadu, je vhodné zvolit reprezentaci městské dopravní sítě jako smíšeného grafu. V některých případech musejí být obě strany komunikace obsluhovány odděleně, což může být i případ jednosměrných komunikací. Or-hrany představující jednosměrné komunikace jsou duplikovány. Neorientované hrany představující komunikaci s oboustrannou obsluhou jsou nahrazeny dvěma or-hranami s opačnou orientací. Výsledný graf je poté orientovaný. Obecně je obsluha komunikace vozidlem náročnější, než pouze její průjezd. Z tohoto důvodu může mít hrana představující komunikaci jiné ohodnocení při obsluze a jiné při průjezdu.

Ve většině případů je do obsluhy zapojeno více typů obslužných vozidel, která se od sebe liší svými omezeními, jako například svou kapacitou, dojezdovou vzdáleností, rychlostí atd. V případě heterogenního vozového parku je řešení ARP značně omezeno a jen v málo případech se podaří vyřešit úlohu prostřednictvím známých exaktních metod. Běžnou strategií je před řešením dekomponovat originální graf na několik podgrafů (Chapleau, Ferland a Rousseau, 1985, Levy a Bodin, 1989, Bodin a Levy, 1991). Při řešení je nutné respektovat pravidla silničního provozu jako např. přednost v jízdě, či zákazy otáčení (Bodin a Kursch, 1979 a McBride, 1982). Souhrnně se jedná o pravidla chování vozidla při projíždění křižovatek. Jeden ze způsobů, jak ve výpočtu respektovat tato pravidla, je duplikace vrcholů křižovatek a propojení těchto vrcholů or-hranami tak, aby byla tato pravidla dodržena, jedná se o převod sítě komunikací do městského režimu, viz obrázek 3.1. Tím však vzroste počet vrcholů a hran grafu. Tento fakt vede k větší výpočetní složitosti problému. Ve výpočtu ARP heuristickými metodami lze tato pravidla zohlednit tzv. penalizacemi nebo zákazy otáčení, které jsou popsány v podkapitole 4.2.

Obecně lze metody řešení ARP rozdělit na lokální heuristické metody, metaheuristické metody a exaktní metody. Heuristické metody jsou metody, které neposkytují zpravidla optimální řešení, ale řešení, pro která neexistuje obecný důkaz korektnosti. K optimálnímu řešení se však mohou blížit. Jsou často používány pro řešení složitých problémů s množstvím omezujících podmínek a mnoha extrémů, kde exaktní metody s obecným důkazem selhávají. Lokální heuristické metody jsou oproti exaktním metodám rychlejší a jednodušší, ale mohou uvíznout v lokálním optimu. Lokální heuristiky lze v základě rozdělit na iterativní a konstruktivní. Konstruktivní heuristiky začínají od triviálního počátečního řešení a postupným přidáváním prvků se propracovávají ke kompletnímu řešení, ve kterém jsou splněny všechny omezující podmínky úlohy. Iterativní heuristiky začínají již na kompletním řešení a postupně ho vylepšují. Kombinací těchto dvou typů heuristik jsou dvoufázové heuristiky, které v první fázi použijí konstruktivní heuristiku a v druhé fázi vylepší řešení iterativní heuristikou.



Obrázek 3.1: Převod na graf městského režimu

Zdroj: Autor

Metaheuristické metody jsou takové metody, které kombinují a řídí některé konstruktivní heuristiky, a to z toho důvodu, aby v řešení nedošlo k uvíznutí v lokálním optimu. Buď konstruktivní heuristika začíná od nulového řešení a přidáváním prvků dojde k vytvoření kompletního řešení, anebo iterativní heuristika lokálně prohledává okolí některých prvků kompletního řešení a iteračně je modifikuje s cílem dosáhnout lepšího řešení. Díky schopnosti opuštění lokálního optima lze metaheuristickými metodami dosáhnout lepších výsledků, než při použití klasických heuristik.

Exaktní metody jsou metody procházející všechna lokální optima úlohy a poskytují optimální řešení. Mají však větší časovou a výpočetní složitost než předchozí metody a ne vždy je lze úspěšně použít, nebo matematicky popsat omezující podmínky úlohy. Mezi exaktní metody patří metody lineárního programování. S množstvím vozidel nebo hran obsluhy grafu roste počet podmínek matematického modelu úlohy ILP, tedy i náročnost a čas potřebný k výpočtu. Toto je největší negativum exaktních metod pro řešení ARP.

Při určování množiny optimálních tras vozidel zahrnujících všechny hrany obsluhy grafu s kladným celočíselným požadavkem na obsluhu $w(h) > 0$, nemusejí být do tahu, resp. sledu, zahrnuty všechny hrany z množiny $E = X \cup Y$, ale jen podmnožina hran obsluhy $R \subseteq E$, $R = X_R \cup Y_R$. Tato podmnožina je sjednocením neorientovaných hran obsluhy X_R a or-hran obsluhy Y_R . Je-li to nutné, do tahu, resp. sledu, jsou zahrnuty i některé bezobslužné hrany z množiny $E \setminus R$. V obecném svozně-rozvozním problému (dále GRP – *General Routing Problem*) je získán na grafu G tah, resp. sled, který obsahuje podmnožinu vrcholů obsluhy $V_R \subseteq V$ a podmnožinu hran obsluhy $R \subseteq E$ a je-li to nezbytné, i ostatní vrcholy a hrany grafu.

GRP lze rozdělit na dva hlavní typy úloh:

1. svozně-rozvozní problém s obsluhou hran (ARP), kde $V_R = \{\emptyset\}$ a $R \neq \{\emptyset\}$,
2. problém obchodního cestujícího (dále TSP – *Traveling Salesman Problem*), kde $V_R \neq \{\emptyset\}$ a $R = \{\emptyset\}$.

V problému obchodního cestujícího je úkolem nalézt minimální cestu přes všechny vrcholy grafu tak, aby byl každý vrchol navštíven pouze jednou a skončit opět ve výchozím vrcholu.

Z GPR mohou být odvozeny dva hlavní typy úloh ARP:

1. problém čínského pošťáka (dále CPP – *Chinese Postman Problem*), kde $V_R = \{\emptyset\}$ a $R = E$,
2. problém příměstského pošťáka (dále RPP – *Rural Postman Problem*), kde $V_R = \{\emptyset\}$ a $R \subseteq E$.

RPP je řešen například při doručování zásilek v příměstských oblastech, kde je zapotřebí, aby vozidlo rozvázející zásilky v určitém počtu obcí obsloužilo množinu hran (zákazníků) z množiny R a zároveň použilo množinu hran (bezobslužných komunikací) z $E \setminus R$, spojujících jednotlivé obce, na přemístění mezi obcemi. Naproti tomu CPP bývá řešen na městské infrastruktuře, kde jsou předmětem obsluhy všechny hrany grafu. Většina praktických aplikací ARP je typu RPP.

Problém obsluhy hran grafu omezený kapacitou vozidla (dále CARP – *Capacitated Arc Routing Problem*) je zobecněný RPP s homogenním vozovým parkem, kde má každé s -té vozidlo kapacitu Q_s , $s = 1, \dots, k$. Jeden vrchol v této úloze představuje depo, ze kterého bude uskutečňována obsluha a hrany obsluhy mají celočíselné, nezáporné ohodnocení, představující požadavek na obsluhu. Svozně-rozvozní trasa vozidla je akceptovatelná, pokud obsahuje depo a celková suma požadavků hran v trase, obsluhovaných jedním vozidlem, nepřekročí jeho kapacitu Q_s , popřípadě není překročena dojezdová vzdálenost vozidla. Úkolem je nalézt množinu svozně-rozvozních tras s minimálním celkovým ohodnocením tak, že každá hrana obsluhy je obsloužena právě jedním vozidlem. Počet vozidel k může být dopředu známý, nebo je v úloze zjišťován. RPP spadá do kategorie NP-těžkých problémů. CARP je také NP-těžký problém za předpokladu, že kapacita vozidla Q_s je větší nebo rovna celkovým požadavkům na obsluhu.

Speciálním případem CARP je problém obsluhy hran grafu omezený kapacitou vozidla a časovými okny (dále CARP-TW – *Capacitated Arc Routing Problem with Time Windows*). Jedná se o problém obsluhy množiny hran se stanovením intervalu začátku a konce doby obsluhy každé obsluhované hrany. Existuje také varianta této úlohy s tzv. měkkými časovými okny (*soft time windows*). V takovéto úloze je dovoleno časová okna porušovat a každé porušení je v řešení penalizováno. V případě, že je úloha tohoto typu rozsáhlejší nebo jsou širší časové intervaly, je řešení tohoto problému extrémně obtížné. Vazby mezi jednotlivými kategoriemi svozně-rozvozních úloh je možné nalézt v příloze 3.

3.3 Problém čínského pošťáka

Problém čínského pošťáka patří do podkategorie ARP. Předmětem obsluhy jsou všechny hrany grafu a je hledán E -tah na grafu, který je E -grafem. Tento problém je řešen na neorientovaném, orientovaném nebo smíšeném grafu. Speciálním případem je nalezení E -tahu na grafu, ve kterém mají hrany v každém směru různá ohodnocení, nebo na grafu, kde jsou hrany obsluhovány v určitém pořadí dle své priority.

3.3.1 Neorientovaný problém čínského pošťáka (UCPP – Undirected Chinese Postman Problem)

První známá zmínka týkající se UCPP je od švýcarského matematika Leonharda Eulera z roku 1736, který ukázal, že souvislý neorientovaný graf $G = (V, X, p)$ je E -grafem, pokud jsou všechny jeho vrcholy sudého stupně. Čínský matematik Mei-Ko Kwan v roce 1962 vypožoroval, že graf G má vždy sudý počet vrcholů lichého stupně, a že E -graf $G' = (V, X', p)$ může být vytvořen z G přidáním hran mezi vrcholy lichého stupně. Dokázal také, že nutnou a postačující podmínkou pro sestrojení optimálního E -tahu v E -grafu G' je fakt, že neobsahuje žádné nadbytečné prvky, tj. neobsahuje více jak dvě hrany spojující libovolnou dvojici vrcholů a všechny hrany $(v_i, v_j) \in X'$ přidané mezi vrcholy lichého stupně grafu splňují trojúhelníkovou nerovnost⁴. UCPP lze formulovat jako úlohu celočíselného lineárního programování (dále úloha ILP) o nalezení takového minimálního rozšíření grafu G na G' , aby všechny vrcholy G' byly sudého stupně.

V následující formulaci bivalentní úlohy LP představuje strukturní proměnná x_{ij} počet přidaných hran (v_i, v_j) do grafu G . V úloze je respektována relace $i < j$. Tímto vznikne graf G' . Kladné reálné číslo $c_{ij} \in R_0^+$ je ohodnocení hrany (v_i, v_j) v grafu G . Nechť je $X(v_i)$ množina hran (v_i, v_j) incidujících s vrcholem v_i , V_L je množina vrcholů lichého stupně z množiny V a X je množina hran (v_i, v_j) v grafu G , pak podle [9]:

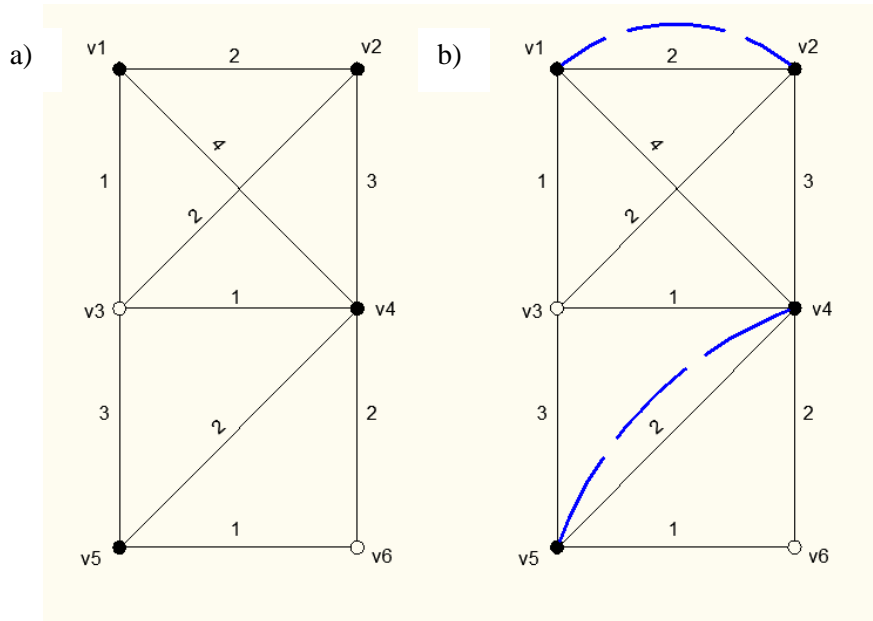
$$\text{Min} \sum_{(v_i, v_j) \in X} c_{ij} x_{ij} \quad (3.10)$$

⁴ Součet délek dvou stran trojúhelníku není menší než délka strany třetí. Obecněji, délka cesty z v_i do v_k a poté do v_j není kratší než délka cesty z v_i do v_j , $d(v_i, v_j) > d(v_i, v_k) + d(v_k, v_j)$.

Za podmíněk:

$$\sum_{(v_i, v_j) \in X(v_i)} x_{ij} = \begin{cases} 1 \pmod{2} & (v_i \in V_L) \\ 0 \pmod{2} & (v_i \in V \setminus V_L) \end{cases} \quad (3.11)$$

$$x_{ij} \in \{0,1\} \quad ((v_i, v_j) \in X) \quad (3.12)$$



Obrázek 3.2: Neorientovaný graf s vrcholy lichého stupně

Zdroj: Autor

Uvažujme neorientovaný graf na obrázku 3.2a). Vrcholy lichého stupně mají černou výplň, hranám je přiřazeno ohodnocení, které představuje jejich délku a přerušované hrany jsou rozšířením původního grafu na E -graf, 3.2b). Formulace bivalentní úlohy ILP pro graf 3.2a) je následující.

$$\text{Min } f(x) = 2x_{12} + x_{13} + 4x_{14} + 2x_{23} + 3x_{24} + x_{34} + 3x_{35} + 2x_{45} + 2x_{46} + x_{56}$$

$$x_0 = (1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0)$$

Za podmíněk:

$$v_1 : \quad x_{12} + x_{13} + x_{14} = 1$$

$$(1 \quad 0 \quad 0 \quad \quad \quad)$$

$$\begin{array}{rcl}
v_2 : & x_{12} & + x_{23} + x_{24} = 1 \\
& (1 & 0 \quad 0) \\
v_3 : & x_{13} & + x_{23} + x_{34} + x_{35} = 0 \\
& (0 & 0 \quad 0 \quad 0) \\
v_4 : & & x_{24} + x_{34} + x_{45} + x_{46} = 1 \\
& (& 0 \quad 0 \quad 0 \quad 1) \\
v_5 : & & x_{35} + x_{45} + x_{56} = 1 \\
& (& 0 \quad 0 \quad 1) \\
v_6 : & & x_{46} + x_{56} = 0 \\
& (& 0 \quad 0)
\end{array}$$

Strukturní proměnné x_{ij} mohou nabývat podle (3.11) a (3.12) tří hodnot: $x_{ij} = 1+1$, $x_{ij} = 0+1$ a $x_{ij} = 0+0$. Hrana je vždy vybrána dvakrát, jednou v jednom směru a podruhé ve směru opačném. Po celočíselném dělení složek výsledného vektoru dvěma je $x_{ij} = 1$, pro hrany incidující se dvěma vrcholy lichého stupně, $x_{ij} = 0$, pro hranu incidující s vrcholem sudého a lichého stupně a $x_{ij} = 0$, pro hranu incidující s vrcholy sudého stupně.

Tento problém může být řešen jako úloha perfektního minimálního párování na kompletním grafu, která je popsána v [6]. Necht' $K_{|V_L|} = (V_L, X_L, p)$ je kompletní graf, kde je množina hran grafu definována jako $X_L = \{(v_i, v_j) \in X_L : v_i, v_j \in V_L, i < j\}$, ohodnocení $c(h)$ je rovno délce minimální cesty z v_i do v_j v G . Graf G je poté rozšířen na G' přidáním hran z množiny X_p , $X_p \subset X_L$, které byly získány perfektním minimálním párováním na grafu $K_{|V_L|}$.

Jack Edmonds a Ellis L. Johnson navrhli algoritmus pro nalezení perfektního minimálního párování s časovou složitostí $O(|V|^3)$. Pro každou neprázdnou lichou podmnožinu vrcholů S , $S \subset V$, je definována řezová množina hran $X(S) = \{(v_i, v_j) \in X(S) : v_i \in S, v_j \in V \setminus S \text{ nebo } v_i \in V \setminus S, v_j \in S, i \neq j\}$. Nalezení perfektního

minimálního párování na grafu $G = (V, X, p)$ je definováno podle [9] jako úloha ILP, kde strukturální proměnná x_{ij} představuje počet kopií hran (v_i, v_j) v grafu G , aby vznikl E -graf.

$$\text{Min} \sum_{(v_i, v_j) \in X} c_{ij} x_{ij} \quad (3.13)$$

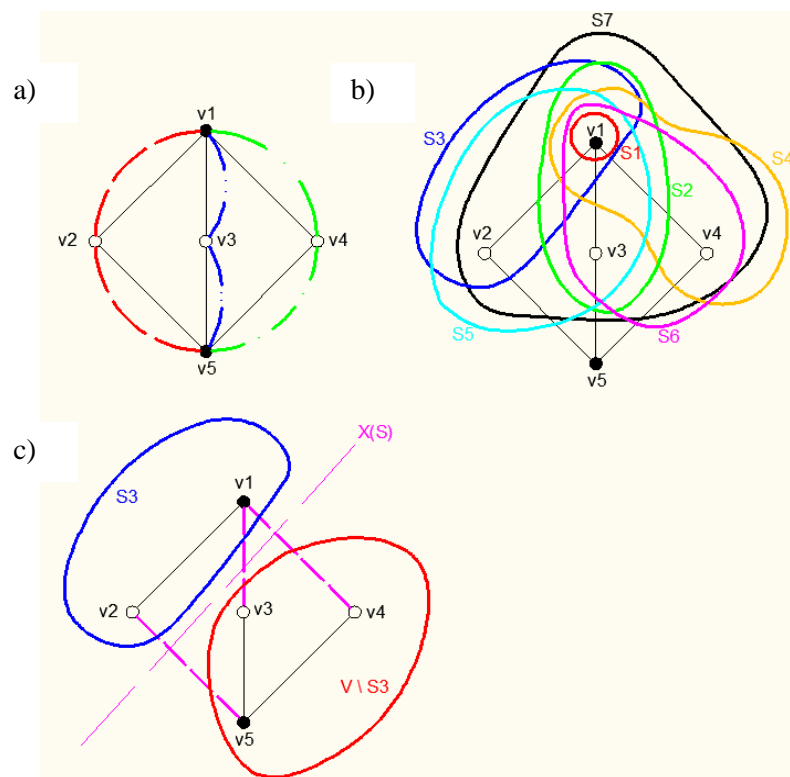
Za podmíněk:

$$\sum_{(v_i, v_j) \in X(S)} x_{ij} \geq 1 \quad (S \subset V, S \text{ je lichá množina}) \quad (3.14)$$

$$x_{ij} \geq 0 \quad ((v_i, v_j) \in X) \quad (3.15)$$

$$x_{ij} \text{ celočíselně} \quad ((v_i, v_j) \in X) \quad (3.16)$$

Podmínky (3.14) se nazývají „nerovnostmi rozvoje“ a jsou definovány pro každou lichou množinu S , tj. pro každou neprázdnou podmnožinu množiny V obsahující lichý počet lichých vrcholů (množina může obsahovat i vrcholy sudého stupně). Jakmile získáme E -graf přidáním hran perfektního minimálního párování, stanovíme na G' E -tah.



Obrázek 3.3: Princip podmínek nerovnosti rozvoje

Zdroj: Autor

Pro pochopení podmínek (3.14) uvažujme graf na obrázku 3.3a). Graf obsahuje dva vrcholy lichého stupně, které mají černou výplň, nejedná se tedy o E -graf. Graf na obrázku lze rozšířit na E -graf třemi způsoby přidání hran, hrany jsou zakresleny přerušovanou čarou. Na obrázku 3.3b) jsou znázorněny všechny možné liché množiny vrcholů S . Vycházíme z tvrzení, že každý souvislý graf obsahuje sudý počet vrcholů lichého stupně. Pokud množina S obsahuje lichý počet vrcholů lichého stupně je vždy spojena se svým doplňkem $V \setminus S$ lichým počtem hran z řezové množiny $X(S)$, obrázek 3.3c). Lichý počet vrcholů lichého stupně v množině S zajišťuje, že doplňková množina $V \setminus S$ bude obsahovat alespoň jeden vrchol lichého stupně. Podmínka (3.14) zajistí, že počet přidávaných hran mezi množiny S a $V \setminus S$ bude roven nejméně jedné, tzn., přes všechny množiny S jsou identifikovány všechny možné cesty z lichých vrcholů z množin S do lichých vrcholů z množin $V \setminus S$. Tímto je zaručeno, že nebudou spojeny minimální cestou dvojice totožných vrcholů lichého stupně. Minimalizační účelová funkce vybere právě minimální cesty. Tyto cesty obsahují kopie hran, které přidáním do grafu vytvoří E -graf.

Francouzský matematik Fleury navrhl koncem 19. století heuristický algoritmus pro nalezení E -tahu na neorientovaném E -grafu. Algoritmus je popsán níže.

Fleuryho algoritmus pro určení E -tahu na neorientovaném E -grafu [6]

Krok 1: Konstrukci E -tahu začneme v libovolném vrcholu grafu. Vybereme libovolnou hranu incidující s tímto vrcholem a projdeme jí. Prošlou hranu označíme.

Krok 2: Při příchodu do vrcholu $v_i \in V$ grafu nikdy nepoužijeme hranu, která je v dané situaci mostem, jehož odstraněním by se graf složený z dosud neoznačených hran rozpadl na:

- netriviální komponenty,
- netriviální komponentu a vrchol, ve kterém tah začíná.

Pokud použijeme algoritmus správně, skončíme ve vrcholu, ve kterém E -tah začíná. Navzdory zjevné jednoduchosti může být tento algoritmus časově náročný z důvodu obtížného určování, zda je hrana mostem, v každém jeho kroku. Z tohoto důvodu Edmonds a Johnson navrhli jiný algoritmus s časovou složitostí $O(|V|)$. Jejich algoritmus koncového párování je popsán níže.

Algoritmus koncového párování pro určení E -tahu v neorientovaném E -grafu [10]

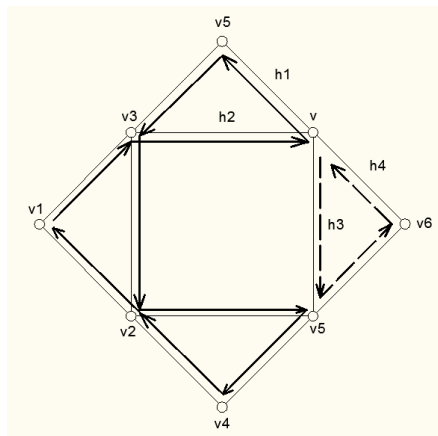
Krok 1: Nalezneme jednoduchý libovolný tah, který nemusí obsahovat všechny vrcholy grafu. Pokud jsou v tahu obsaženy všechny vrcholy grafu, skončíme.

Krok 2: Uvažujme libovolný vrchol v v tahu, který inciduje s hranou, která není součástí tahu. Vytvoříme druhý tah začínající ve v , aniž by se hranově překrýval s prvním.

Krok 3: Necht' h_1 a h_2 jsou dvě hrany incidující s vrcholem v v prvním tahu, h_3 a h_4 hrany incidující s vrcholem v v druhém tahu. Sloučíme oba tahy do jednoho. Například začneme ve vrcholu v a hranou h_3 , pokračujeme druhým tahem, dokud neskončíme opět ve v prostřednictvím hrany h_4 , poté pokračujeme z vrcholu v prvním tahem a začneme hranou h_1 dokud neskončíme opět ve v prostřednictvím hrany h_2 .

Pokud jsme prošli všechny hrany, skončíme, jinak pokračujeme krokem 2.

Ukázka algoritmu je na obrázku 3.4.



Obrázek 3.4: Ukázka algoritmu koncového párování

Zdroj: Autor

3.3.2 Orientovaný problém čínského pošťáka (DCPP – Directed Chinese Postman Problem)

Nutnou, ne však postačující podmínkou vytvoření E -grafu z grafu $G = (V, Y, p)$ je, aby byl graf G silně souvislý, tedy aby mezi všemi dvojicemi vrcholů grafu existovala orientovaná cesta. V orientovaném případě lze určit minimální E -graf z G pomocí nalezení nejlevnějšího toku s podmínkou, že dolní hranice kapacit or-hran bude rovna jedné, horní hranice nebude omezena a tok na každé or-hraně bude větší nebo roven jedné. Ohodnocení vrcholů je rovno rozdílu počtu or-hran vcházejících do vrcholu a z vrcholu vycházejících.

Řešením nejlevnějšího toku jsou získány hodnoty toku na jednotlivých or-hranách, které v optimálním řešení DCCP představují počet kopií or-hran, které je nutné přidat do G , aby vznikl E -graf.

Algoritmus pro nalezení nejlevnějšího toku na orientovaném grafu [11]

Krok 1: Ohodnotíme vrcholy $v_i \in A$, $A = \{v_i \in A : |Y^-(v_i)| > |Y^+(v_i)|, A \subset V\}$, hodnotou $r(v_i) = |Y^-(v_i)| - |Y^+(v_i)|$ a vrcholy $v_j \in B$, $B = \{v_j \in B : |Y^-(v_j)| < |Y^+(v_j)|, B \subset V\}$, hodnotou $s(v_j) = |Y^+(v_j)| - |Y^-(v_j)|$. Položíme ohodnocení $r'(v_i) = r(v_i)$ a $s'(v_j) = s(v_j)$, pro $i = 1, \dots, |A|$, $j = 1, \dots, |B|$, a tok or-hranami $y[h] = 1$ pro všechny $h \in Y$.

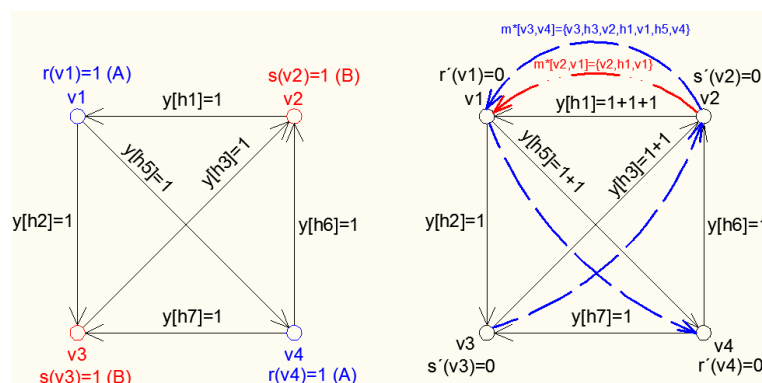
Krok 2: Pokud platí $B \neq \{\emptyset\}$, pokračujeme krokem 5, jinak vybereme vrcholy $v_u \in A$ a $v_z \in B$, mezi kterými existuje na grafu G orientovaná cesta $m[v_z, v_u]$. Pokud takové dva vrcholy neexistují, algoritmus končí (graf G není silně souvislý, nejlevnější tok nebyl nalezen).

Krok 3: V grafu G nalezneme minimální orientovanou cestu $m^*[v_z, v_u]$.

Krok 4: Vypočteme $g = \min \left\{ \min_{h \in m^*[v_z, v_u]} (y[h]), r'(v_u), s'(v_z) \right\}$.

Položíme $r'(v_u) = r'(v_u) - g$ a $s'(v_z) = s'(v_z) - g$. Pokud je $r'(v_u) = 0$, vyřadíme vrchol v_u z množiny A , pokud $s'(v_z) = 0$, vyřadíme vrchol v_z z množiny B . V grafu G přiřadíme každé hraně h tok $y[h] = (y[h] = y[h] + g : h \in m^*[v_z, v_u], h \in Y)$. Pokračujeme krokem 2.

Krok 5: Tok $y = \sum_{h \in Y} y[h]$ je nejlevnějším tokem na grafu G .



Obrázek 3.5: Ukázka algoritmu pro nalezení nejlevnějšího toku

Zdroj: Autor

Na obrázku 3.5 je příklad nalezení nejlevnějšího toku na orientovaném grafu, kde mají or-hrany jednotkové délky a počáteční tok na každé hraně je roven jedné (každá hrana musí být v E -tahu projita). Dva vrcholy náležejí do množiny vrcholů A a dva vrcholy náležejí do množiny B , jejich ohodnocení je dáno proměnnými $r(v_i)$ a $s(v_j)$. Přerušovanou a čerchovanou čarou jsou vyznačeny minimální orientované cesty nalezené mezi vrcholy s nenulovým ohodnocením z množiny A a B . Hodnota nejlevnějšího toku na hranách udává počet kopií hran v grafu, aby vznikl E -graf.

Minimální E -graf lze také sestavit pomocí řešení dopravní úlohy, která je příbuznou úlohou k úloze nalezení nejlevnějšího toku. Obdobně, necht' je A množina vrcholů v_i , pro které platí, že počet hran vcházejících do vrcholu je větší jak počet hran z vrcholu vycházejících, rozdíl těchto počtů představuje proměnná $r_i = |Y^-(v_i)| - |Y^+(v_i)|$. B je množina vrcholů v_j , pro které platí, že počet hran vycházejících z vrcholu je větší než počet hran do vrcholu vcházejících. Rozdíl těchto počtů představuje proměnná $s_j = |Y^+(v_j)| - |Y^-(v_j)|$. Takto může být proměnná r_i definována jako nabídka (3.18) a s_j jako poptávka (3.19). Koeficienty účelové funkce c_{ij} představují délku minimální cesty z v_i do v_j . Strukturální proměnná x_{ij} vyjadřuje počet přidávaných or-hran $[v_i, v_j]$ do grafu G . Úlohu lze poté podle [9] zapsat následovně:

$$\text{Min} \sum_{v_i \in A} \sum_{v_j \in B} c_{ij} x_{ij} \quad (3.17)$$

Za podmínek:

$$\sum_{v_j \in B} x_{ij} = r_i \quad (v_i \in A) \quad (3.18)$$

$$\sum_{v_i \in A} x_{ij} = s_j \quad (v_j \in B) \quad (3.19)$$

$$x_{ij} \geq 0 \quad (v_i \in A, v_j \in B) \quad (3.20)$$

Po přidání or-hran $[v_i, v_j]$ do grafu G bude výsledný graf symetrický. Jakmile je určen E -graf, E -tah může být jednoduše nalezen upraveným Fleuryho algoritmem pro orientované grafy.

3.3.3 Nesouměrný problém čínského pošťáka (WPP – Windy Postman Problem)

WPP je NP-těžký problém, který v sobě zahrnuje neorientovaný, orientovaný a smíšený případ CPP. Tento problém může být v některých případech vyřešen v polynomiálním čase. V WPP má každá neorientovaná hrana grafu $G = (V, X, p)$ odlišné ohodnocení, $c_1(h)$ v jednom směru a $c_2(h)$ v opačném směru. Mei-Ko Kwan uvedl postačující podmínku pro polynomiální řešení WPP. Každý cyklus v grafu G , obsahující všechny hrany obsluhy, musí mít v obou svých směrech stejnou délku. Poté lze minimální WPP cestu na grafu G určit následovně. Z grafu G je sestroyen graf $G' = (V, X', p)$ s ohodnocením hran $c(h') = (c_1(h) + c_2(h)) / 2$, $h \in X, h' \in X'$. Úloha je poté řešena jako CPP s určením E -tahu na G' . Výsledný E -tah je zároveň řešením nesouměrného problému čínského pošťáka na G . Tento algoritmus je polynomiální, ale vyžaduje kontrolu, zda mají všechny cykly v grafu G , v obou svých směrech, stejnou délku. Další dostačující podmínkou pro polynomiální řešení je, aby byl graf G E -grafem.

Nechť $G = (V, X, p)$ je výchozí graf a $X(v_i)$ je množina hran incidujících s vrcholem v_i . Strukturální proměnná x_{ij} bude představovat počet průchodů hranou (v_i, v_j) z v_i do v_j v optimálním řešení WPP. Koeficienty c_{ij} představují ohodnocení hrany (v_i, v_j) v jednom směru a c_{ji} ohodnocení hrany (v_i, v_j) ve směru druhém. Problém lze podle [12] formulovat jako úlohu ILP:

$$\text{Min} \sum_{(v_i, v_j) \in X} (c_{ij} x_{ij} + c_{ji} x_{ji}) \quad (3.21)$$

Za podmínek:

$$x_{ij} + x_{ji} \geq 1 \quad ((v_i, v_j) \in X) \quad (3.22)$$

$$\sum_{(v_i, v_j) \in X(v_i)} (x_{ij} - x_{ji}) = 0 \quad (v_i \in V) \quad (3.23)$$

$$x_{ij}, x_{ji} \geq 0 \quad ((v_i, v_j) \in X) \quad (3.24)$$

$$x_{ij}, x_{ji} \text{ celočíselně} \quad ((v_i, v_j) \in X) \quad (3.25)$$

Podmínky (3.22) zajistí, že každá hrana bude procházena alespoň jednou. Podmínky (3.23) zajistí, že počet příchodů do každého vrcholu bude roven počtu odchodů z tohoto vrcholu.

Příbuzným problémem k WPP je problém minimální Eulerovské orientace (dále MCEOP – *Minimum Cost Eulerian Orientation Problem*) [13]. Zde je $G = (V, X, p)$ neorientovaný E -graf a každá hrana h má dvě ohodnocení $c_1(h) = c_{ij}$ a $c_2(h) = c_{ji}$, $p(h) = (v_i, v_j)$. Orientace G je získána nahrazením neorientované hrany h or-hranou \bar{h} , $p[\bar{h}] = [v_i, v_j]$, nebo $p[\bar{h}] = [v_j, v_i]$. V MCEOP je hledána minimální Eulerovská orientace grafu G . Při řešení MCEOP je na rozdíl od WPP každá hrana v optimálním řešení procházena pouze jednou a toto řešení má časovou složitost $O(|V||X|^2)$.

3.3.4 Smíšený problém čínského pošťáka (MCP – Mixed Chinese Postman Problem)

Pro lepší popis problému bude předpokládáno, že MCP je definován na silně souvislém grafu $G = (V, E, p)$. Distanční matice $D = (d_{ij})_{i,j=1}^n$ je definována na množině hran $E = X \cup Y$. Řešením CPP je takové minimální rozšíření grafu G o kopie orientovaných a neorientovaných hran, aby byly splněny podmínky existence E -grafu. Pokud je graf sudý a symetrický, je zároveň vyvážený, ale jeho symetrie není nezbytnou podmínkou, aby byl E -grafem. Smíšený, silně souvislý graf je E -grafem, pokud je vyvážený. Pokud je graf E -grafem, lze na tomto grafu nalézt E -tah. E -tah lze nalézt podle [14] ve třech krocích:

Krok 1: Přiřadíme orientaci některým neorientovaným hranám tak, aby z grafu G vznikl symetrický graf.

Krok 2: Přiřadíme směry ostatním neorientovaným hranám v grafu.

Krok 3: Nalezneme E -tah na symetrickém grafu G .

Pro sestrojení symetrického grafu z G lze použít následující proceduru navrženou Fordem a Fulkersonem.

Ford-Fulkersonova procedura transformace smíšeného grafu na symetrický graf [15]

Krok 1: Nahradíme každou neorientovanou hranu v grafu $G = (V, E, p)$ párem or-hran, tím získáme orientovaný graf $G' = (V, Y', p)$. Každé hraně z $Y' \cap Y$ přiřadíme dolní

ohodnocení 1 (hodnotu minimálního toku protékajícího hranou), každé hraně $Y' \setminus Y$ dolní ohodnocení 0 a každé hraně z Y' horní ohodnocení 1 (hodnotu maximálního toku protékajícího hranou).

Krok 2: Použijeme algoritmus pro nalezení přípustného toku na intervalově ohodnoceném grafu G' , který navrhli také Ford a Fulkerson (popis algoritmu lze nalézt v [6]). Necht' y_{ij} představuje tok na hraně (v_i, v_j) .

Krok 3: Orientujeme hrany $(v_i, v_j) \in X$ v grafu G následovně: pokud $y_{ij} = 1$ a zároveň $y_{ji} = 0$, orientujeme hranu (v_i, v_j) z v_i do v_j .

Nyní je graf symetrický a zbývající neorientované hrany jsou orientovány podle následující procedury, která má časovou složitost $O(|E|)$.

Procedura kompletní orientace symetrického grafu [14]

Krok 1: Pokud jsou všechny hrany v grafu orientované, algoritmus končí.

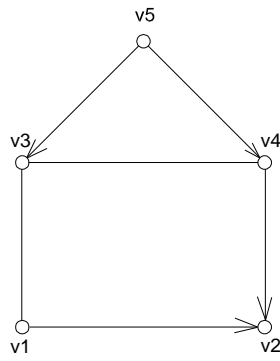
Krok 2: Necht' je v vrcholem symetrického grafu, se kterým inciduje alespoň jedna neorientovaná hrana (v, w) . Položíme $v_1 = v$ a $v_2 = w$.

Krok 3: Orientujeme hranu (v_1, v_2) z v_1 do v_2 . Pokud $v_2 = v$ pokračujeme krokem 1.

Krok 4: Položíme $v_1 = v_2$ a určíme hranu (v_1, v_2) incidující s v_1 . Pokračujeme krokem 3.

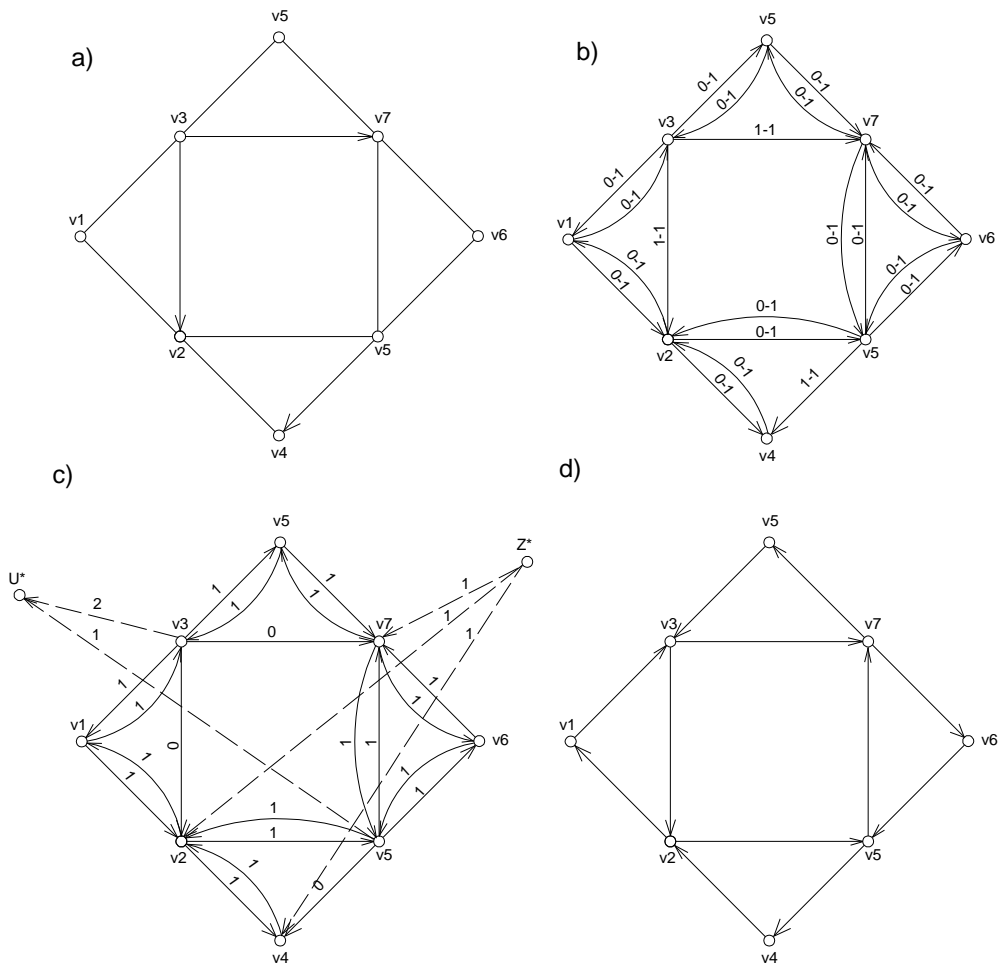
Na obrázku 3.7 je příklad transformace smíšeného grafu na symetrický graf. Obrázek 3.7a) je původní smíšený graf. Graf na obrázku 3.7b) vznikl po 1. kroku Ford-Fulkersonovy procedury. Ohodnocení hran představuje přípustný interval toku hranami. Na intervalově ohodnoceném grafu 3.7c) je hledán přípustný tok. Graf po 3. kroku Ford-Fulkersonovy procedury a po aplikaci procedury kompletní orientace symetrického grafu, na kterém lze nalézt E -tah, je na obrázku 3.7d).

Algoritmus nalezení přípustného toku lze aplikovat na sudý graf, aniž by bylo předem známo, zda je E -grafem. Pokud algoritmus nalezne přípustný tok, je graf E -grafem, v opačném případě graf není E -grafem. Pokud graf G není E -grafem, lze ho určit minimálním rozšířením grafu G , tedy přidáním orientovaných a neorientovaných hran tak, aby vznikl E -graf. Avšak existují grafy, které nemohou být na E -graf rozšířeny. Takovýto graf je na obrázku 3.6.



Obrázek 3.6: Příklad smíšeného grafu, z kterého nelze sestrojít E -graf

Zdroj: Autor



Obrázek 3.7: Příklad transformace smíšeného grafu na symetrický graf

Zdroj: Autor

Pro nalezení minimálního rozšíření smíšeného grafu lze použít matematického modelu ILP, který je řešen metodou sečných nadrovin, ve které jsou zpočátku relaxovány podmínky celočíselnosti. Úloha je řešena iterativně. V každé iteraci je identifikováno porušení podmínek celočíselnosti, sudosti (každý vrchol grafu musí být sudého stupně) a vyváženosti. Do matematického modelu jsou přidány dodatečné podmínky, které eliminují tato porušení, nevyplývají z lineárních kombinací ostatních podmínek úlohy LP, zachovávají množinu přípustných celočíselných řešení a zároveň zužují celou oblast přípustných řešení. Po určitém počtu iterací je získáno přípustné celočíselné řešení. Z toho je patrný exponenciální růst počtu podmínek úlohy LP.

Zde je popis matematického modelu úlohy ILP.

Nechť $Y^+(v_k) = \{[v_i, v_j] \in Y : v_i = v_k\}$ a $Y^-(v_k) = \{[v_i, v_j] \in Y : v_j = v_k\}$ jsou množiny or-hran vycházejících a vcházejících z (do) vrcholu v_k . Nechť strukturní proměnná x_{ij} představuje počet přidaných hran (v_i, v_j) a strukturní proměnná y_{ij} počet přidaných or-hran $[v_i, v_j]$ do grafu, aby vznikl E -graf a z_k je celočíselná proměnná. V této formulaci je ke každé neorientované hraně z X přiřazena vždy jen jedna proměnná x_{ij} . Celočíslné řešení LP tedy neurčí orientaci hran.

Dále, necht' p_k je binární konstanta, která je rovna 1, pokud je stupeň vrcholu v_k lichý a rovna 0, pokud je stupeň vrcholu v_k sudý.

Definujeme pro každou podmnožinu S , $S \subset V$, následující množiny:

$Y^+(S) = \{[v_i, v_j] \in Y : v_i \in S, v_j \in V \setminus S\}$, množina or-hran vycházejících z množiny S a vcházejících do množiny $V \setminus S$, $Y^-(S) = \{[v_i, v_j] \in Y : v_i \in V \setminus S, v_j \in S\}$, množina or-hran vcházejících do množiny S a vycházejících z množiny $V \setminus S$.

$X(S) = \{(v_i, v_j) \in X : v_i \in S, v_j \in V \setminus S \text{ nebo } v_i \in V \setminus S, v_j \in S\}$, řezová množina hran spojující množinu S a množinu $V \setminus S$, $E(S) = X(S) \cup Y^+(S) \cup Y^-(S)$, řezová množina všech hran a or-hran spojující množinu S a množinu $V \setminus S$, a konstanta $u(S) = |Y^+(S)| - |Y^-(S)| - |X(S)|$, vyjadřující nevyváženost množiny S .

Pokud je množina $S = \{v_k\}$ jednoprvková, lze psát $Y^+(S) = Y^+(v_k)$, $Y^-(S) = Y^-(v_k)$ a $X(S) = X(v_k)$.

Formulace úlohy ILP je podle [11] následující.

$$\text{Min} \sum_{(v_i, v_j) \in X} c_{ij} x_{ij} + \sum_{[v_i, v_j] \in Y} c_{ij} y_{ij} \quad (3.26)$$

Za podmíněk:

$$\sum_{(v_i, v_j) \in X(v_k)} x_{ij} + \sum_{\substack{[v_i, v_j] \in Y^+(v_k) \\ \text{nebo } [v_j, v_i] \in Y^-(v_k)}} y_{ij} + p_k = 2z_k \quad (v_k \in V, p_k \in \{0,1\}) \quad (3.27)$$

$$\sum_{(v_i, v_j) \in X(S)} x_{ij} - \sum_{[v_i, v_j] \in Y^+(S)} y_{ij} + \sum_{[v_i, v_j] \in Y^-(S)} y_{ij} \geq u(S) \quad (S \subset V, S \neq \{\emptyset\}) \quad (3.28)$$

$$z_k, x_{ij}, y_{ij} \geq 0 \text{ celočíselně} \quad (v_k \in V) \quad (3.29)$$

Podmínky (3.27) souvisí s podmínkami sudosti. Ve smíšeném grafu je stupeň vrcholu jednoduše roven počtu hran a or-hran, které s tímto vrcholem incidují. Tyto podmínky zajistí, že bude do grafu přidán lichý (sudý) počet hran incidujících s vrcholem lichého (sudého) stupně v_k , tak aby byl tento vrchol sudého stupně. Pravé strany podmínek odpovídají počtu přidanych hran do grafu incidujících s vrcholem v_k a levé strany podmínek jsou sudá čísla.

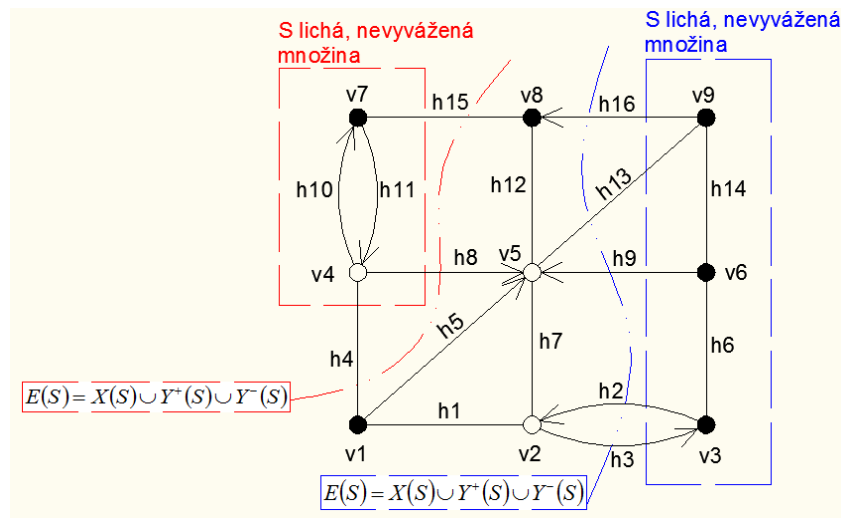
Podmínka (3.28) zajistí, aby byla každá neprázdná množina S vyvážená. Pokud je $u(S) > 0$, počet or-hran $|Y^+(S)|$ vycházejících z množiny S a vcházejících do množiny $V \setminus S$ překročí o hodnotu $u(S)$ počet hran spojujících S a $V \setminus S$, a or-hran vcházejících do S a vycházejících z $V \setminus S$, $|Y^-(S)| + |X(S)|$. Pokud má být dodrženo, aby každá hrana grafu byla procházena nejméně jednou, musí být do grafu přidáno nejméně $u(S)$ hran nebo or-hran spojujících S a $V \setminus S$. Tímto je odstraněna nesymetričnost $u(S)$ z množiny S .

V souvislosti s předchozími podmínkami lze podle [10] do matematického modelu přidat Edmondsovy zobecněné podmínky nerovnosti rozvoje pro MCPP, kde S je lichá množina, tj. množina s lichým počtem vrcholů lichého stupně, která může obsahovat i vrcholy sudého stupně, obdobně jako v podmínkách (3.10):

$$\sum_{[v_i, v_j] \in Y^+(S)} y_{ij} + \sum_{[v_i, v_j] \in Y^-(S)} y_{ij} + \sum_{(v_i, v_j) \in X(S)} x_{ij} \geq 1 \quad (S \subset V, S \text{ lichá množina}) \quad (3.30)$$

Lze dokázat, že množina S je lichá tehdy a jen tehdy, pokud je počet hran a or-hran $|E(S)| = |X(S)| + |Y^-(S)| + |X(S)|$ také lichý. Podmínka (3.30) zajistí, že mezi lichou množinou S a množinou $V \setminus S$ bude přidána alespoň jedna hrana, tím pomáhá posílit fakt, že v každém

E -tahu na grafu G musí být počet průchodů mezi množinou S a $V \setminus S$ sudý. Podmínky nerovnosti rozvoje jsou vzhledem k podmínkám (3.27) a (3.29) nadbytečné, ale pomáhají posílit LP relaxaci v metodě sečných nadrovin. Na obrázku 3.8 je ukázka smíšeného grafu s dvěma libovolně zvolenými lichými množinami vrcholů S , které indukují řezovou množinu hran $E(S)$ spojující vrcholy z S a $V \setminus S$. Množiny S jsou označeny obdélníky s přerušovanou čarou. Řezové množiny jsou křivky s čerchovanou čarou. Vrcholy s černou výplní jsou vrcholy lichého stupně.



Obrázek 3.8: Smíšený graf s lichými množinami vrcholů S

Zdroj: Autor

Podmínky nerovnosti rozvoje pro liché množiny S na obrázku 3.8:

$$x_{78} + x_{45} + x_{41} \geq 1 \quad (S = \{v_4, v_7\})$$

$$x_{98} + x_{95} + x_{65} + x_{32} + x_{23} \geq 1 \quad (S = \{v_3, v_6, v_9\})$$

A podmínky vyváženosti:

$$-x_{45} + x_{78} + x_{41} \leq 1 \quad (S = \{v_4, v_7\}) \quad u(S) = 1 - 0 - 2 = -1$$

$$x_{98} - x_{95} + x_{65} + x_{32} - x_{23} \geq 1 \quad (S = \{v_3, v_6, v_9\}) \quad u(S) = 3 - 1 - 1 = 1$$

Nyní bude podle [10] stručně popsána metoda sečných nadrovin. V první iteraci metody sečných nadrovin je relaxován matematický model úlohy ILP. Obsahuje účelovou funkci, podmínky nezápornosti bez celočíselného omezení, podmínky vyváženosti (3.28)

korespondující s nevyváženými vrcholy a nejvíce nevyváženými množinami S ($\max_{S \subset V} \{u(S)\}$) z G a zobecněné podmínky nerovnosti rozvoje (3.30), které jsou asociovány s vrcholy lichého stupně. Necht' P_k je relaxace matematického modelu P v k -té iteraci, $f(x_0, y_0)$ je optimální hodnota účelové funkce matematického modelu P , m_k je počet podmínek v P_k , vyjma podmínek nezápornosti, a vektor (x_0^k, y_0^k) , $x_0^k, y_0^k \in R^{X \cup Y}$, je vektorem optimálního řešení P_k .

Krok 1: (konstrukce počáteční relaxace) Sestavíme matematický model úlohy LP s proměnnými x, y :

Účelové funkce ve vztahu (3.26),

za podmínek:

- podmínky nerovnosti rozvoje (3.30), které jsou asociovány s vrcholy lichého stupně grafu G ,
- podmínky vyváženosti (3.28), které jsou asociovány s nevyváženými vrcholy a nejvíce nevyváženými množinami S grafu G ,
- podmínky nezápornosti.

Položíme m_0 rovno počtu podmínek v matematickém modelu P_0 mimo podmínek nezápornosti a proměnnou $k = 0$. Pokud je $m_0 = 0$ pokračujeme krokem 6.

Krok 2: (výpočet simplexovou metodou) Vypočteme P_k simplexovou metodou a získáme vektor optimálního řešení (x_0^k, y_0^k) a hodnotu účelové funkce $f^k(x_0^k, y_0^k)$.

Krok 3: (indikace porušení podmínek vyváženosti) Pokud je graf G_{x^k, y^k} vyvážený, položíme $e_1 = 0$ a pokračujeme krokem 4. Přidáme do P_k podmínky vyváženosti, které byly porušeny v (x_0^k, y_0^k) . Necht' e_1 je počet přidanych podmínek vyváženosti.

Krok 4: (indikace porušení podmínek sudosti) Přidáme do P_k podmínky nerovnosti rozvoje, které byly porušeny v (x_0^k, y_0^k) , a které nejsou zahrnuty v podmínkách přidanych v kroku 3. Necht' e_2 je počet přidanych podmínek nerovnosti rozvoje. Pokud $e = e_1 + e_2 = 0$ pokračujeme krokem 5, jinak položíme $k = k + 1$ a $m_k = m_{k-1} + e$, pokračujeme krokem 2.

Krok 5: (indikace porušení celočíselnosti proměnných) Pokud není výsledný vektor (x_0^k, y_0^k) celočíselný, přidáme do P_k jeden Gomoryho řez⁵, položíme $k = k + 1$ a $m_k = m_{k-1} + 1$, pokračujeme krokem 2. Pokud se v grafu G_{x^k, y^k} nachází vrchol lichého stupně v_i a vektor (x_0^k, y_0^k) je celočíselný, vygenerujeme podmínku (3.27) korespondující s vrcholem v_i , přidáme Gomoryho řez s podmínkou (3.27) do P_k , položíme $k = k + 1$ a $m_k = m_{k-1} + 2$, pokračujeme krokem 2.

Krok 6: (určení E -tahu) Položíme $f(x_0, y_0) = f^k(x_0^k, y_0^k)$ a určíme E -tah na grafu G_{x^k, y^k} .

Efektivita této metody závisí obecně na dvou faktorech. Za prvé, v kroku 2 musí být dostatečně rychle vyřešeny matematické modely, tj. řešení proběhne s menším počtem iterací. Pokud je celkový počet přidávaných podmínek do úlohy LP malý, simplexová metoda vypočte úlohu efektivněji. Za druhé, podmínky definující optimální řešení původní úlohy LP musejí být identifikovány po malém počtu iterací. Pokud by tomu tak nebylo, dojde k přidání většího počtu podmínek a řešení simplexovou metodou by zahrnovalo více iterací.

Heuristické řešení MCPP bylo navrženo Edmondsnem a Johnsonem a vylepšeno Fredericksonem a Christofidem. Tato heuristika je schopná nalézt dobré řešení, při splnění nutných a postačujících podmínek E -grafu. Greg N. Frederickson navrhl dvě heuristiky, MCPP1 a MCPP2, každou s časovou složitostí $O(\max\{|V|^3, |Y|(\max\{|Y|, |X|\})^2\})$. Zde je nástin těchto dvou algoritmů.

Heuristika MCPP1 pro MCPP [16]

Krok 1: V grafu $G = (V, E, p)$, $E = X \cup Y$, určíme vrcholy lichého stupně. Zanedbáme orientaci hran a sestrojíme kompletní graf s vrcholy lichého stupně a hranami s ohodnocením rovným délce minimální cesty mezi těmito vrcholy v G . Na kompletním grafu určíme perfektní minimální párování. Rozšíříme graf G přidáním orientovaných a neorientovaných hran získaných minimálním párováním. Vznikne sudý graf.

Krok 2: Použijeme modifikovaný algoritmus pro nalezení nejlevnějšího toku, abychom získali symetrický graf. Modifikace spočívá v přiřazení nulového toku

⁵ Způsob přidání nové podmínky v metodě sečných nadrovin. Gomoryho řez odřezává prostor množiny přípustných řešení úlohy LP, která neobsahuje žádný celočíselný bod (x_0^k, y_0^k) . Některý z celočíselných bodů se přitom posune do krajního vrcholu množiny přípustných řešení, nebo alespoň na její hranu.

neorientovaným hranám. Necht' je graf $G' = (V, E', p)$, $E' = X' \cup Y'$, výsledným grafem. Pokud není výsledný graf sudý, pokračujeme krokem 3, jinak algoritmus končí.

Krok 3: Určíme množinu vrcholů lichého stupně v grafu G' . Bez ohledu na orientaci hran nalezneme speciální cyklus, který je tvořen střídavou posloupností dvou typů cest s počátečním a koncovým vrcholem lichého stupně. V těchto vrcholech na sebe vždy tyto dva typy cest navazují. První typ cesty obsahuje pouze neorientované hrany z množiny $Y' \setminus Y$ (orientace or-hran z této množiny je dočasně zanedbána), druhý typ cesty obsahuje pouze neorientované hrany z množiny X' . Všechny tyto hrany musejí mít krajní vrcholy lichého stupně. Pokud je cyklus nalezen, přiřadíme cyklu libovolnou orientaci. Tímto jsou v cyklu orientovány neorientované hrany z množiny X' . Or-hrany, které tvoří součást cyklu, jsou z množiny $Y' \setminus Y$. Pokud mají tyto or-hrany s cyklem shodnou orientaci, jsou duplikovány, v opačném případě jsou vypuštěny z grafu. Pokud se v grafu nacházejí vrcholy lichého stupně, pokračujeme krokem 1, jinak je nalezen symetrický a sudý graf.

Krok 3 je nutný vzhledem ke skutečnosti, že v kroku 2 nemusí nutně vzniknout sudý graf. Pokud nejsou všechny hrany v grafu orientované, dokončíme proces použitím procedury kompletní orientace symetrického grafu. Nyní je graf orientovaný, sudý a symetrický, je tedy E -grafem.

Heuristika MCPP2 pro MCPP [16]

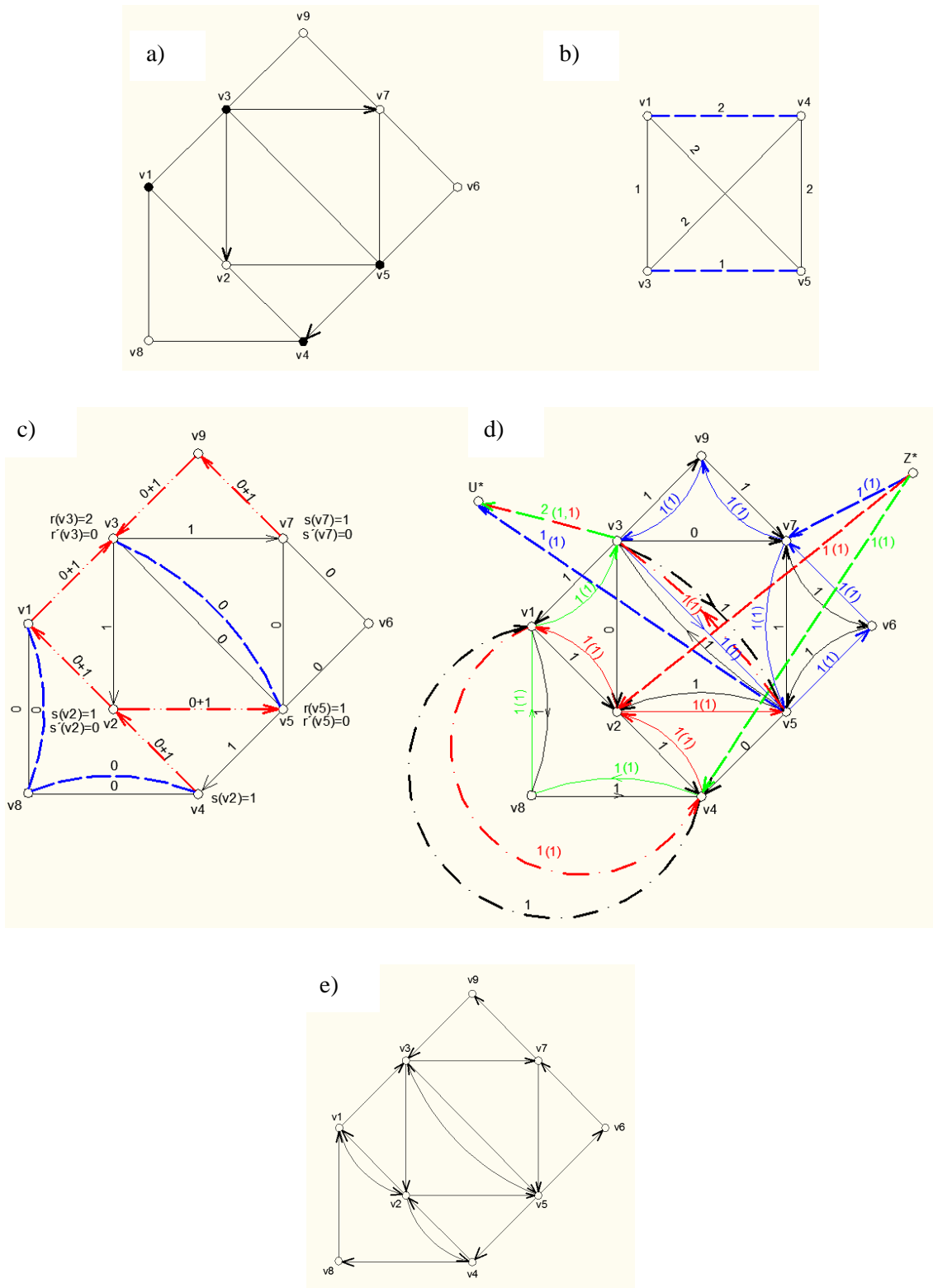
Krok 1: Tento krok je shodný s krokem 2 v heuristice MCPP1. Necht' je $G' = (V, E', p)$ výsledným grafem a $E' = X' \cup Y'$.

Krok 2: V grafu $G'' = (V, X', p)$ nalezneme vrcholy lichého stupně. V grafu $G''' = (V, X, p)$ nalezneme mezi těmito vrcholy minimální cesty. Určíme minimální párování lichých vrcholů v grafu G''' . Vložíme neorientované hrany určené minimálním párováním do množiny X' . Nalezneme E -tah přes or-hrany z množiny Y' a hrany z množiny X' .

Každé takovéto řešení je možné zjednodušit. Pokud ohodnocení or-hrany c_{ij} přesáhne délku minimální cesty z v_i do v_j , je vhodné nahradit or-hranu (v_i, v_j) touto cestou. Tím se sníží počet proměnných.

Na obrázku 3.9 je uveden příklad řešení heuristikou MCPP1. Na 3.9a) je původní smíšený graf s vrcholy lichého stupně, které jsou označeny černou výplní. Na 3.9b) je kompletní graf s vrcholy lichého stupně ze 3.9a) a zvýrazněnými hranami určenými minimálním párováním. Tyto hrany jsou přidány do grafu a vznikne sudý graf. Na 3.9c) je graf s nejlevnějším tokem po kroku 2 a na 3.9d) je výsledný E -graf, ve kterém byly

neorientované hrany z grafu na 3.9c) orientovány procedurou kompletní orientace grafu. Na obrázku 3.9d) je 2. krok algoritmu, který byl modifikován autorem této práce.



Obrázek 3.9: Heuristika MCPP1

Zdroj: Autor

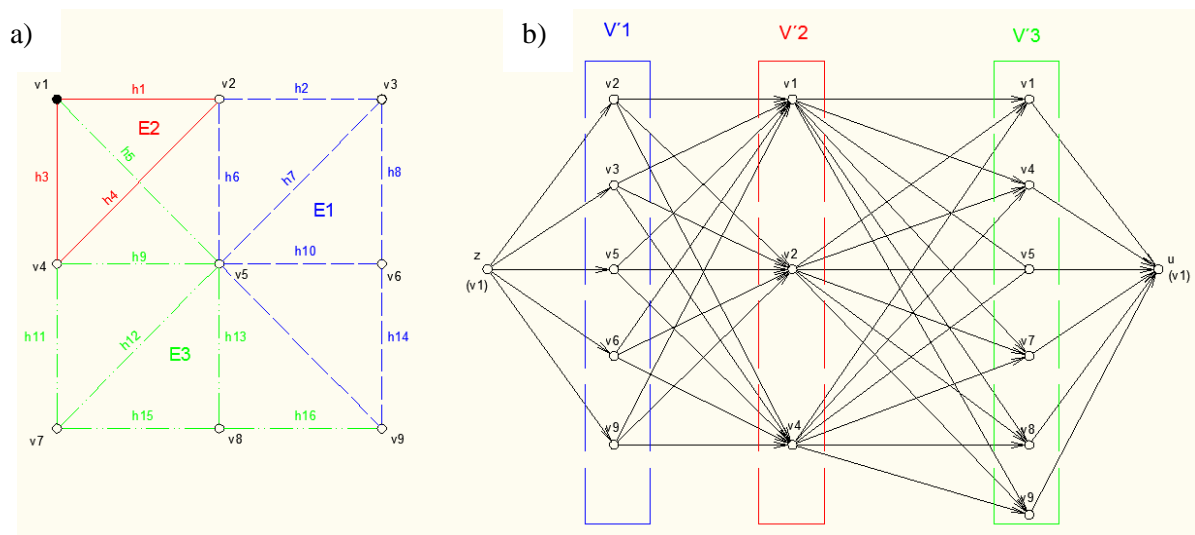
Jedná se o nahrazení algoritmu nalezení nejlevnějšího toku algoritmem nalezení maximálního toku na intervalově ohodnocené síti. Jak se ukázalo, tato modifikace dosahuje stejných výsledků jako původní algoritmus, s tím rozdílem, že po dokončení algoritmu dochází k méně častému použití procedury kompletní orientace grafu.

3.3.5 *Stupňovitý problém čínského pošťáka (HCPP – Hierarchical Chinese Postman Problem)*

HCPP je definován na neorientovaném nebo orientovaném grafu $G = (V, E, p)$. Tento problém se v praxi vyskytuje například u zimní údržby pozemních komunikací, kde množina hran E odpovídá komunikacím s různou prioritou obsluhy. Dále se jedná například o čištění komunikací a svoz komunálního odpadu. Množina vrcholů V obsahuje vrchol v_1 , který je depem. Hrany z množiny E jsou podle své priority rozděleny do množin $\{E_1, \dots, E_k\}$. Mezi těmito množinami existuje vzájemná relace $<$, která reprezentuje prioritu obsluhy hran. V HCPP se na grafu G určuje minimální cyklus mezi obsahující vrchol v_1 a hrany obsluhy z množin $\{E_1, \dots, E_k\}$ takovým způsobem, aby platila vzájemná relace $E_p < E_q$, $p \neq q$. Poté jsou všechny hrany z množiny E_p obsluhovány před hranami z množiny E_q .

Podle [17] jsou uvažovány podgrafy $G_p = (V_p, E_p, p)$ indukované množinami hran E_p . Pokud je každý podgraf souvislý a je definováno pořadí obsluhy hran $E_1 < E_2 \dots < E_k$, úloha může být řešena v čase $O(k|V|^5)$. Nejprve je následujícím způsobem sestrojen orientovaný, acyklický graf $G' = (V', Y', p)$. Množina vrcholů V' bude obsahovat vrchol z , který je nazýván zdrojem, a vrchol u , který je nazýván ústím. Množina vrcholů V'_p bude obsahovat jednu kopii každého vrcholu, který inciduje s hranou z množiny E_p , poté $V' = \{u, z\} \cup V'_1 \cup \dots \cup V'_k$. Množina hran Y' bude obsahovat všechny or-hrany vycházející z vrcholu z a ústící do vrcholu z množiny V'_1 . Tyto hrany budou představovat minimální cesty z vrcholu v_1 do vrcholů V'_1 v původním grafu G . Množina hran Y' bude obsahovat všechny or-hrany vycházející z vrcholů z množiny V'_k a ústící do vrcholu u . Tyto hrany budou představovat minimální cesty z vrcholů V'_1 do vrcholu v_1 v původním grafu G . Dále bude množina hran Y' obsahovat všechny or-hrany vycházející z vrcholů množiny V'_p a ústící do vrcholů množiny V'_{p+1} , $p = 1, \dots, k-1$. Ohodnocení $c[h]$ hran $p[h] = [v_i, v_j]$ z množiny Y'

je dáno následovně. Pokud $v_i = z$ a vrchol $v_j \in V'_1$, potom je $c[h]$ rovno délce minimální cesty z v_i do v_j . Pokud $v_i \in V'_p$ a $v_j \in V'_{p+1}$ pro $p = 1, \dots, k$ a $V'_{k+1} = \{u\}$, potom je $c[h]$ rovno délce otevřeného nebo uzavřeného E -tahu začínajícího ve vrcholu v_i a končícího ve vrcholu v_j , který pokrývá všechny hrany z E_p . HCPP je poté řešen určením minimální cesty z vrcholu z do vrcholu u na grafu $G' = (V', Y', p)$.



Obrázek 3.10: Příklad řešení HCPP

Zdroj: Autor

Na obrázku 3.10a) je graf $G = (V, E, p)$, ve kterém jsou tři množiny hran podle priority obsluhy $E_1 < E_2 < E_3$. Hrany z jednotlivých množin jsou rozlišeny barvou a typem čáry. Na obrázku 3.10b) je graf $G' = (V', Y', p)$ s vrcholy z, u , které představují vrchol s depem v_1 v grafu G . Na grafu G' je hledána minimální cesta z vrcholu z do vrcholu u obsahující E -tahy na množinách E_1, E_2, E_3 .

Složitost této procedury je dána výpočtem ohodnocení or-hran v $G' = (V', Y', p)$. Poté může každý vrchol z množiny V' náležet až k množinám V_p a časová složitost procedury bude $O(k|V|^2)$. Existují algoritmy, které jsou schopné vypočítat ohodnocení hran v čase $O(|V|^3)$, ať je graf G orientovaný, či neorientovaný. V neorientovaném případě může být menší časové složitosti dosaženo použitím rychlejšího algoritmu párování. Pokud netvoří podgrafy G_p souvislý graf, nebo je relační uspořádání množin jen částečné, problém se stává NP-těžkým.

Souhrn metod řešení ARP je uveden v tabulce 3.1.

Tabulka 3.1: Metody řešení CPP

problém	exaktní algoritmy	heuristické algoritmy
UCPP	Polynomiální. Algoritmus perfektního minimálního párování.	Fleuryho algoritmus.
DCPP	Polynomiální. Algoritmus určení nejlevnějšího a maximálního toku na grafu.	-
WCPP	NP-těžký. Polynomiální, pokud jde o E -graf. Metoda sečných nadrovin.	Zaokrouhlování zlomkových proměnných v LP relaxaci.
MCPP	NP-těžký. Metoda větví a řezů.	MCPP1 a MCPP2.
HCPP	NP-těžký. Může být řešen v čase $O(k V ^5)$.	-

Zdroj: Autor na podkladu podkapitoly 3.3

3.4 Problém příměstského poštáka (RPP – Rural Postman Problem)

Jak bylo již uvedeno, RPP je příbuzným problémem k CPP a liší se od CPP v předmětu obsluhy hran. Předmětem obsluhy v RPP nejsou všechny hrany grafu, ale pouze jejich podmnožina. Stejně jako u CPP existují u RPP varianty úloh na orientovaném, neorientovaném a smíšeném grafu. Další úlohy RPP jsou omezené kapacitou vozidla. Dále budou popsány jednotlivé typy úloh řešící RPP.

3.4.1 Neorientovaný problém příměstského poštáka (URPP – Undirected Rural Postman Problem)

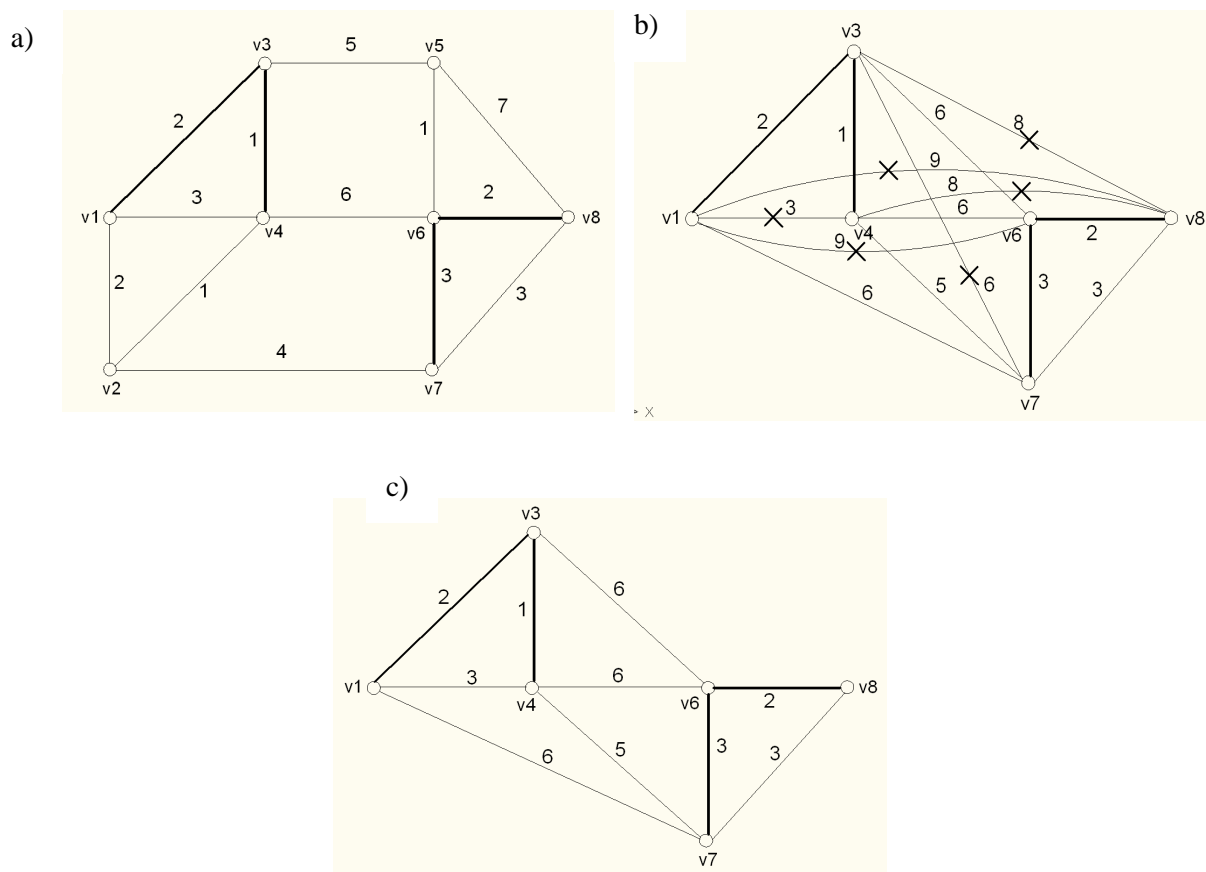
V neorientovaném RPP jsou všechny hrany grafu $G = (V, X, p)$ neorientované. Pokud je graf $\bar{G} = (V, X_R, p)$ spojitý a množina X_R je množinou hran obsluhy grafu G , může být problém řešen jako neorientovaný CPP (Eiselt, Gendreau, Laporte, 1995) s výpočtem minimálních cest mezi vrcholy lichého stupně (některé cesty mohou obsahovat hrany z $X \setminus X_R$, tj. hrany, které nejsou předmětem obsluhy) v grafu G .

Aby byla zmenšena časová složitost tohoto problému, je RPP obecně řešen na upraveném grafu $G' = (V', X', p)$, který je definován následovně.

Pro množinu vrcholů platí $V' = \{v_i \in V : (v_i, v_j) \in X_R\}$, $v_j \in V$. Tento graf obsahuje menší nebo stejný počet hran a vrcholů a v řešení RPP je ekvivalentní s původním grafem G . Množina hran X' je získána:

- přidáním hran $p(h) = (v_i, v_j)$ do X , pro každý vrchol $v_i, v_j \in V'$, jejichž ohodnocení $c(h) = d(v_i, v_j)$ je rovno délce minimální cesty mezi v_i a v_j ,
- vypuštěním všech hran $(v_i, v_j) \in X \setminus X_R$, pro které platí trojúhelníková nerovnost $d(v_i, v_j) > d(v_i, v_k) + d(v_k, v_j)$, $v_k \in V$,
- vypuštěním multihran se stejným ohodnocením.

Tato úprava je nazývána procedurou předběžného zpracování grafu RPP. Pro názornou ukázkou uvažujme dva grafy na obrázku 3.11, kde jsou hrany obsluhy vyznačeny tučně a mají číselné nezáporné ohodnocení.



Obrázek 3.11: Procedura předběžného zpracování grafu

Zdroj: Autor

Na obrázku 3.11a) je původní graf G se silně označenými hranami z množiny X_R . Ohodnocení hran představuje jejich délku. Na obrázku 3.11b) je graf, který vznikl přidáním hran do, G a na obrázku 3.11c) nově vzniklý graf G' , po vypuštění hran $(v_i, v_j) \in X \setminus X_R$, které splňují $d(v_i, v_j) > d(v_i, v_k) + d(v_k, v_j)$, a multihran s totožným ohodnocením. Proceduru předběžného zpracování lze také aplikovat na smíšený graf.

V upraveném grafu G' indukuje množina hran obsluhy X_R' spojitě komponenty grafu G_1', \dots, G_p' s množinami vrcholů $V_1', \dots, V_p' \subset V'$. Pokud všechna ohodnocení hran c_{ij} splňují podmínku trojúhelníkové nerovnosti, může být použita heuristika pro neorientovaný RPP, která je založena na heuristice symetrického TSP (Frederickson, 1979).

Heuristika pro neorientovaný RPP [18]

Krok 1: (nalezení minimální kostry grafu) Sestrojíme minimální kostru T_G spojující podgrafy G_1', \dots, G_p' . Necht' $c(T_G)$ je délka minimální kostry. Označíme $c(X_R)$ jako sumu ohodnocení všech hran z množiny X_R a $c(T)$ jako délku optimálního řešení RPP, představovaného sledem T . Potom platí $c(T_G) + c(X_R) \leq c(T)$.

Krok 2: (minimální párování) Nalezneme minimální párování mezi všemi vrcholy lichého stupně v grafu indukovaném množinou $X_R \cup T_G$. Necht' je X_P množinou hran získaných minimálním párováním a $c(X_P)$ suma ohodnocení hran z množiny X_P .

Krok 3: (E -tah) Řešení RPP je získáno nalezením E -tahu na grafu indukovaném množinou $X_R \cup T_G \cup X_P$, za předpokladu, že všechna ohodnocení hran $c(v_i, v_j)$, $v_i, v_j \in X_R \cup T_G \cup X_P$, splňují trojúhelníkovou nerovnost, což platí, pokud $c(X_P) \leq \frac{c(T)}{2}$.

Poté délka sledu $c(T_G) + c(X_R) + c(X_P)$ nepřesáhne hodnotu $\frac{3c(T)}{2}$.

Pro řešení RPP existují dvě formulace úlohy ILP. V první formulaci úlohy ILP je strukturní proměnná x_{ij} definována následovně. Pokud $(v_i, v_j) \in X_R$, proměnná x_{ij} představuje počet průchodů hranou obsluhy (v_i, v_j) , tedy hranu (v_i, v_j) projdeme $(1 + x_{ij})$ krát. Pokud $(v_i, v_j) \in X' \setminus X_R$, x_{ij} představuje počet průchodů bezobslužné hrany (v_i, v_j) , hranu projdeme x_{ij} krát. Formulace úlohy ILP je podle [19] následující:

Problém URPP1

$$\text{Min. } \sum_{(v_i, v_j) \in X_R} c_{ij} (1 + x_{ij}) + \sum_{(v_i, v_j) \in X' \setminus X_R} c_{ij} x_{ij} \quad (3.31)$$

Za podmíněk:

$$\sum_{\substack{(v_i, v_j) \in X_R \\ j > i}} (1 + x_{ij}) + \sum_{\substack{(v_i, v_j) \in X' \setminus X_R \\ j > i}} x_{ij} = \sum_{\substack{(v_i, v_j) \in X_R \\ j < i}} (1 + x_{ji}) + \sum_{\substack{(v_i, v_j) \in X' \setminus X_R \\ j < i}} x_{ji} \pmod{2} \quad (v_i \in V') \quad (3.32)$$

$$\sum_{v_i \in S, v_j \in \bar{S}} x_{ij} \geq 1 \quad \left(S = \bigcup_{k \in P} V_k, \bar{S} = \left(\bigcup_{k=1}^p V_k \right) \setminus S, P \subset \{1, \dots, p\} \right) \quad (3.33)$$

$$x_{ij} \geq 0 \text{ celočíselně} \quad ((v_i, v_j) \in X') \quad (3.34)$$

Podmínka (3.32) zajistí, že všechny vrcholy budou sudého stupně, podmínka (3.33) zajistí, že všech p spojitých komponent G_1', \dots, G_p' bude vzájemně propojeno. Je relativně snadné dokázat, že strukturní proměnná x_{ij} může být shora ohraničená hodnotou 1, pokud $(v_i, v_j) \in X_R$, a hodnotou 2, pokud $(v_i, v_j) \in X' \setminus X_R$. Rovněž podmínku (3.32) lze nahradit podmínkou (3.32'), je-li pravá strana rovnice rovna hodnotě $2d_i$, přičemž:

$$d_i \geq 0 \text{ a je celočíselné} \quad (v_i \in V') \quad (3.35)$$

Obdobnou formulaci navrhli Corberain a Sanchis (1991). Necht' $X(v_i) = \{(v_i, v_j) \in X'\}$ je množina hran, ve které všechny hrany incidují s vrcholy v_i v upraveném grafu G' . Vrchol v_i nazveme X_R -sudý (resp. X_R -lichý), pokud inciduje se sudým (resp. lichým) počtem hran z množiny X_R . Strukturní proměnné x_{ij} jsou definovány stejným způsobem jako v předchozím případě. Úlohu lze podle [20] formulovat následovně:

Problém URPP2

$$\text{Min. } \sum_{(v_i, v_j) \in R} c_{ij} (1 + x_{ij}) + \sum_{(v_i, v_j) \in X' \setminus R} c_{ij} x_{ij} \quad (3.36)$$

Za podmíněk:

$$\sum_{(v_i, v_j) \in X(v_i)} x_{ij} = 0 \quad (\text{mod } 2) \quad (v_i \in V', v_i \text{ je } X_R - \text{sudý}) \quad (3.37)$$

$$\sum_{(v_i, v_j) \in X(v_i)} x_{ij} = 1 \quad (\text{mod } 1) \quad (v_i \in V', v_i \text{ je } X_R - \text{lichý}) \quad (3.38)$$

$$\sum_{\substack{v_i \in S, v_j \in \bar{S} \\ \text{nebo } v_j \in S, v_i \in \bar{S}}} x_{ij} \geq 2 \quad \left(S = \bigcup_{k \in P} V_k, \bar{S} = \left(\bigcup_{k=1}^p V_k \right) \setminus S, P \subset \{1, \dots, p\} \right) \quad (3.39)$$

$$x_{ij} \geq 0 \text{ celočíselně} \quad ((v_i, v_j) \in X') \quad (3.40)$$

Sanchis a Corberain (1991) určili několik tříd stěn polytopů⁶ konvexního obalu⁷ přípustných řešení definovaných podmínkami (3.37)–(3.40). Dále zjistili, že všechny stěny polytopů indukované podmínkami ILP pro TSP jsou také stěnami polytopů v neorientovaném RPP. Tuto úlohu lze tedy řešit metodou větví a hranic (*Branch and Bound Method*) jako úlohu TSP.

3.4.2 Orientovaný problém příměstského pošťáka (DRPP – Directed Rural Postman Problem)

Orientovaný RPP je definován na grafu $G = (V, Y, p)$, kde Y je množina or-hran grafu. Pokud je graf $\bar{G} = (V, Y_R, p)$ spojitý, může být tento problém redukován na orientovaný CPP. Obecně je problém řešen na upraveném grafu $G' = (V', Y', p)$ zkonstruovaném obdobně jako v neorientovaném případě, ale hrany $[v_i, v_j]$ jsou nyní orientované a mají délku c_{ij} , která je rovna délce minimální cesty z v_i do v_j , $c_{ij} = d[v_i, v_j]$. Spojité komponenty G'_1, \dots, G'_p s příslušnými množinami vrcholů jsou definovány výše. Christifides a kol. navrhli pro orientovaný RPP následující heuristiku.

⁶ Polytop je konvexním obalem konečně mnoha bodů v n -rozměrném vektorovém prostoru E_n (jedná se o uzavřenou konvexní množinu, navíc vždy omezenou).

⁷ Konvexním obalem $\text{Conv}(A)$ množiny $A \subseteq E_n$ je průnik všech konvexních množin obsahujících prvky množiny A , tj. množina všech bodů, které lze získat z bodů A (násobnými) konvexními kombinacemi.

Heuristika pro orientovaný RPP [21]

Krok 1: (nalezení minimální kostry orientovaného grafu) Sestrojíme minimální kostru orientovaného grafu (Edmonds, 1967) s libovolným počátečním vrcholem, která ve visících vrcholech spojuje komponenty G'_1, \dots, G'_p . Necht' je \tilde{G} výsledným grafem.

Krok 2: (dopravní problém) Stejně jako v případě orientovaného CPP získáme symetrický graf \tilde{G} přidáním or-hran nejkratších délek do grafu tak, aby se ve všech vrcholech počet vycházejících hran rovnal počtu hran vcházejících.

Krok 3: (E -tah) Na takto rozšířeném grafu určíme E -tah.

Proceduru je možné opakovat postupnou volbou všech vrcholů za vrchol počáteční a poté vybrat nejlepší řešení. Christofides formuloval tento problém v [21] jako úlohu ILP. Zde je uvedena zjednodušená formulace. Proměnná x_{ij} představuje počet opakování or-hrany $[v_i, v_j] \in Y_R$, v optimálním řešení RPP. Strukturální proměnná x_{ij} představuje počet průchodů or-hranou $[v_i, v_j] \in Y' \setminus Y_R$. Dále jsou definovány konstanty:

$$b_{ij} = \begin{cases} 1 & \text{pokud } [v_i, v_j] \in Y_R \wedge v_i, v_j \in V' \\ 0 & \text{jinak} \end{cases}$$

$$\bar{b}_{ij} = \begin{cases} 1 & \text{pokud } (v_i, v_j) \in Y' \setminus Y_R \wedge v_i, v_j \in V' \\ 0 & \text{jinak} \end{cases}$$

Poté lze problém podle [21] formulovat jako úlohu ILP s ohledem na jednotlivé množiny vrcholů $\bar{V} \in \{V'_1, \dots, V'_p\}$ následovně:

Problém DRPP

$$\text{Min.} \quad \sum_{(v_i, v_j) \notin Y_R} c_{ij} (1 + x_{ij}) + \sum_{(v_i, v_j) \in Y' \setminus Y_R} c_{ij} x_{ij} \quad (3.41)$$

Za podmínek:

$$\sum_{[v_i, v_j] \in Y_R} (1 + x_{ij}) b_{ij} + \sum_{[v_i, v_j] \in Y' \setminus Y_R} x_{ij} \bar{b}_{ij} = \sum_{[v_i, v_j] \in Y_R} (1 + x_{ji}) b_{ji} + \sum_{[v_i, v_j] \in Y' \setminus Y_R} x_{ji} \bar{b}_{ji} \quad (v_i \in V') \quad (3.42)$$

$$\sum_{v_i \in S, v_j \in \bar{S}} x_{ij} \geq 1 \quad \left(S = \bigcup_{k \in P} V_k, \bar{S} = \left(\bigcup_{k=1}^p V_k \right) \setminus S, P \subset \{1, \dots, p\}, \bar{V} \subseteq S \right) \quad (3.43)$$

$$x_{ij} \geq 0 \text{ celočíselně} \quad ((v_i, v_j) \in X') \quad (3.44)$$

Podmínky vyváženosti (3.42) zajistí pro každý vrchol, že počet vcházejících hran do vrcholu bude roven počtu vycházejících hran z téhož vrcholu. Podmínky (3.43) zajistí, že řešení vytvoří orientovanou kostru grafu obsahující visící vrchol, který je součástí každé komponenty grafu G_1, \dots, G_p , indukované příslušnou množinou V_1', \dots, V_p' .

3.4.3 Problém regálového nakladače (SCP - Stacker Crane Problem)

Problém regálového zakladače je definován na smíšeném grafu $G = (V, E, p)$, kde X je množina neorientovaných hran a Y množina or-hran a platí $E = X \cup Y$. Problém spočívá v určení nejkratšího uzavřeného sledu, obsahujícího všechny or-hrany z množiny Y právě jednou. Tyto or-hrany představují jednotlivé pohyby regálového nakladače v daných směrech. Or-hrany mohou také představovat přemísťování zásilek dopravním prostředkem, poté se jedná o minimalizaci délky sledu představujícího trasu vozidla. Přemísťováním je myšleno, že vozidlo vyjede z depa a navštíví vrchol, kde naloží zásilku a přemístí ji do dalšího vrcholu, jede do jiného vrcholu, kde naloží další zásilku a přemístí ji do jiného vrcholu. Toto provede několikrát za sebou a vrátí se do depa. Pokud je ohodnocení $c[v_i, v_j]$ všech or-hran $[v_i, v_j] \in Y$ nulové, je úloha SCP ekvivalentní s úlohou TSP. Z tohoto vyplývá, že SCP je NP-těžkým problémem. Frederickson, Hecht a Kim navrhli pro řešení tohoto problému v [18] dvě heuristiky, LARGEARCS a SMALLARCS, ve kterých je nutné, aby graf G splňoval následující podmínky:

- 1) každý vrchol grafu musí incidovat alespoň s jednou or-hranou z množiny Y ,
- 2) ohodnocení $c[v_i, v_j]$ neorientovaných hran grafu musí splňovat trojúhelníkovou nerovnost.

Pokud graf G nespĺňuje tyto dvě podmínky, může být procedurou předběžného zpracování transformován na graf G' , který by je splňoval. Pokud jsou podmínky splněny, je problém řešen na grafu G' , a řešení může být interpretováno na původním grafu G . Dále bude předpokládáno, že podmínky 1) a 2) graf G splňuje. Pokud je celková suma ohodnocení

or-hran větší než délka optimálního sledu na G , poskytuje heuristika LARGEARCS lepší řešení, než když je tomu naopak.

Algoritmus LARGEARCS [18]

Krok 1: (Bipartitní graf) Z grafu G vybereme množinu Y všech or-hran (obrázek 3.12a). Sestrojíme bipartitní graf $G' = (V_{poč} \cup V_{kon}, X, p)$, který bude obsahovat dvě množiny vrcholů. První množina vrcholů $V_{poč}$ bude obsahovat počáteční vrcholy, ze kterých or-hrana vychází, a druhá množina V_{kon} koncové vrcholy, do kterých or-hrana vchází. Mezi těmito dvěma množinami vrcholů sestrojíme neorientované hrany vyjma dvojic vrcholů, které tvořily krajní vrcholy or-hran z množiny Y . Tyto hrany ohodnotíme minimálními vzdálenostmi mezi dvojicemi vrcholů v G , které s hranami incidují (obrázek 3.12b).

Krok 2: (párování) Na grafu G' vyřešíme problém minimálního bipartitního párování.

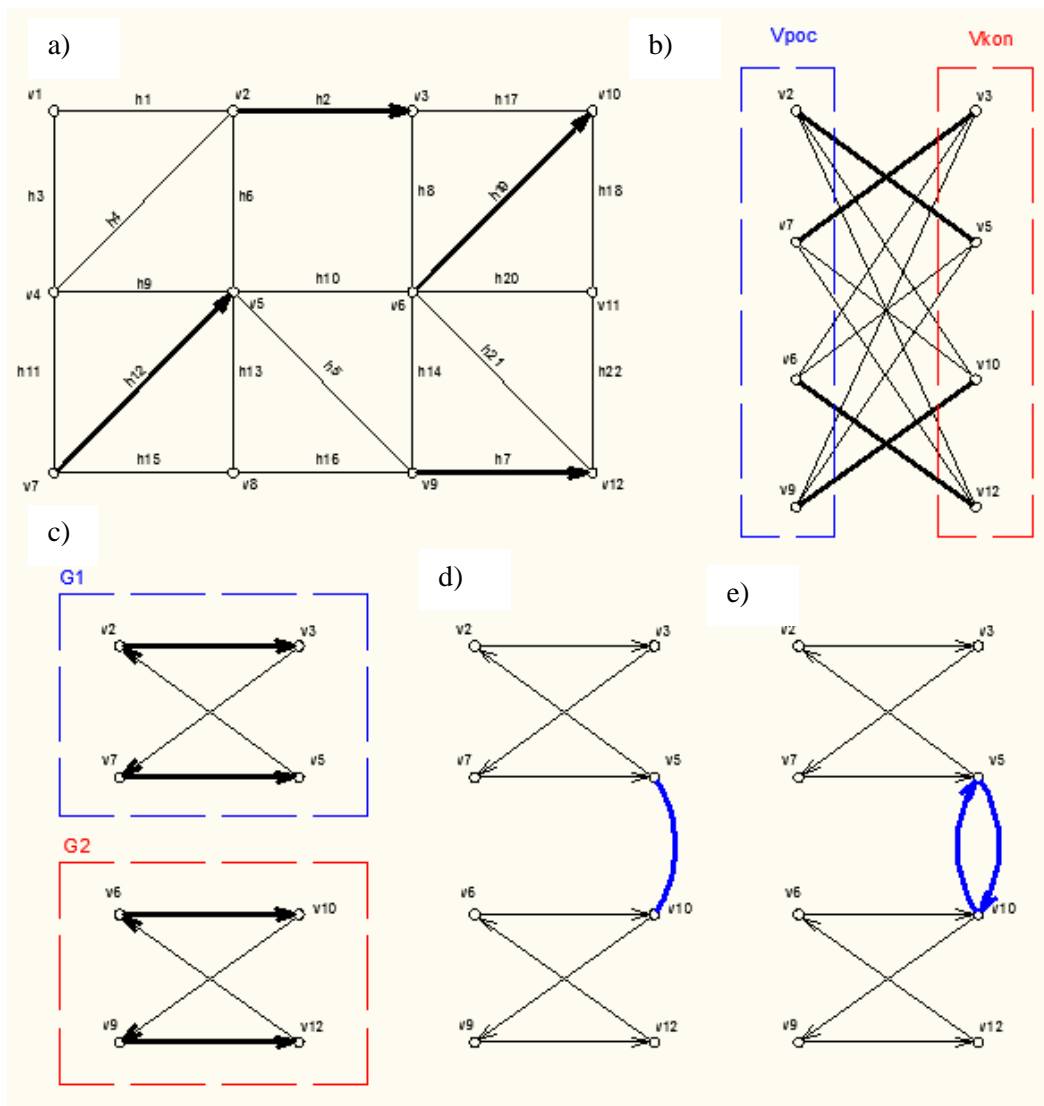
Minimální bipartitní párování určuje množinu neorientovaných hran M_B , kde každý vrchol z množiny $V_{poč} \cup V_{kon}$ inciduje nejvýše s jednou hranou z množiny M_B a suma ohodnocení hran $c(v_i, v_j)$ je minimální, $(v_i, v_j) \in M_B$. Pokud vrchol neinciduje s žádnou hranou z množiny M_B , jde o tzv. nepokrytý (nespárovaný) vrchol, $|V_{poč}| \neq |V_{kon}|$. Párování je tzv. perfektní, pokud jsou všechny vrcholy $V_{poč} \cup V_{kon}$ pokryty, $|V_{poč}| = |V_{kon}|$.

Sestrojíme orientovaný graf G'' . Množina hran tohoto grafu bude odpovídat množině Y a všem neorientovaným hranám nalezeným minimálním bipartitním párováním, které orientujeme z koncového do počátečního vrcholu (obrázek 3.12c).

Krok 3: (minimální kostra) Graf G'' nyní obsahuje několik spojitých komponent. Nalezneme minimální kostru spojující tyto komponenty s ohledem na původní ohodnocení grafu (obrázek 14d).

Krok 4: (E -tah) Sestrojíme orientovaný graf \bar{G} , jehož množina hran odpovídá množině hran grafu G'' a dvojícím opačně orientovaných hran, kterými nahradíme neorientované hrany minimální kostry (obrázek 3.12e). Tento graf je Eulerovský a řešení SCP získáme řešením orientovaného CPP na \bar{G} .

Časová složitost algoritmu LARGEARCS je $O(\max\{|V|^3, |Y|^3\})$.



Obrázek 3.12: Algoritmus LARCEARCS. a) původní množina or-hran Y , b) bipartitní graf G'' , c) graf G'' , d) minimální kostra spojující komponenty grafu G_1'' a G_2'' , e) orientovaný graf \bar{G}

Zdroj: Autor

Pokud distanční matice splňuje trojúhelníkovou nerovnost, algoritmus nalezne sled $T_{LARCEARCS}$, délky $c(T_{LARCEARCS})$, splňující nerovnost

$$c(T_{LARCEARCS}) \leq 3c(T) - 2c(Y), \quad (3.45)$$

kde $c(T)$ je délka sledu optimálního řešení SCP na G . $c(Y)$ je suma ohodnocení hran z množiny Y .

Druhý algoritmus, SMALLARCS, řeší převod smíšeného grafu na E -graf, který by měl být sudý, symetrický a vyvážený.

Algoritmus SMALLARCS [18]

Krok 1: (redukovaný graf) Grafu G redukuje na neorientovaný graf G^* . Or-hrany z množiny Y nahradíme vrcholy, které budou množinou vrcholů grafu G^* . Mezi tyto vrcholy vložíme neorientované hrany a vznikne kompletní graf. Ohodnocení těchto hran, odpovídající dvěma or-hranám $[v_i, v_j]$ a $[v_k, v_l]$, je rovno $\min \{c[v_i, v_k], c[v_i, v_l], c[v_j, v_k], c[v_j, v_l]\}$, obrázek 3.13a). Určíme nejkratší řetězce hran mezi všemi dvojicemi vrcholů grafu G^* a těmi nahradíme stávající neorientované hrany. Vznikne graf G^{**} , obrázek 3.13b).

Krok 2: (minimální kostra) Na grafu G^{**} určíme minimální kostru, obrázek 3.13c).

Krok 3: (minimální párování) Na minimální kostře grafu G^{**} nalezneme minimální párování mezi vrcholy lichého stupně, obrázek 3.13c).

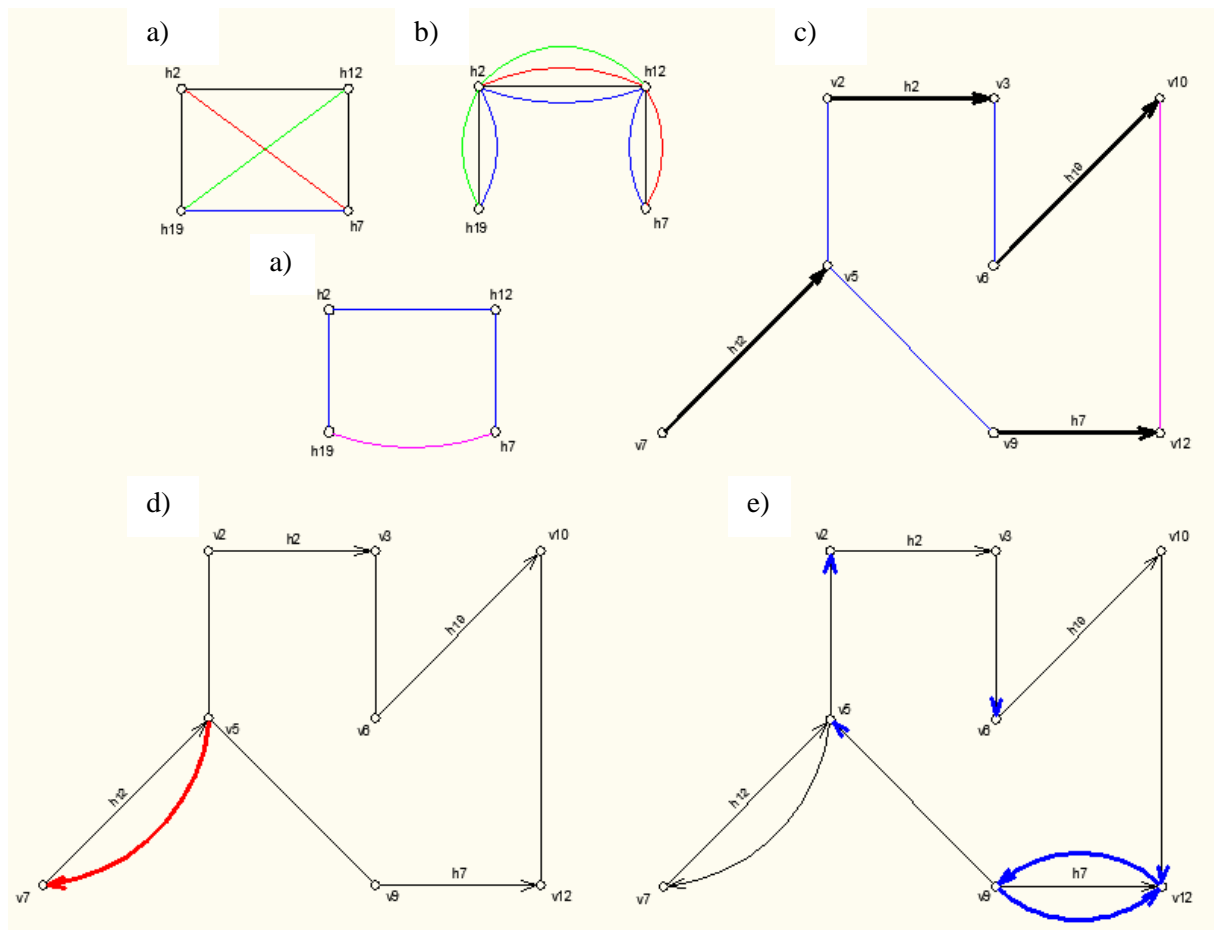
Krok 4: (zpětné rozšíření grafu) Všechny vrcholy nahradíme zpětně or-hranami z množiny Y a každou neorientovanou hranu vzniklou párováním, nebo hranu která je součástí minimální kostry, nahradíme odpovídajícím nejkratším řetězcem hran z kroku 1. Získáme smíšený graf \hat{G} , obrázek 3.13d). Pokud je or-hrana v \hat{G} or-hranou $[v_i, v_j]$ grafu G , tak inciduje s vrcholy v_i a v_j stejného stupně (sudá or-hrana). Pokud jsou vrcholy v_i a v_j lichého stupně, přidáme mezi tyto vrcholy neorientovanou hranu (v_i, v_j) a dočasně ji orientujeme z v_j do v_i . Nechť je G^{***} výsledným grafem, obrázek 3.13e).

Krok 5: (první E -tah) V G^{***} nalezneme E -tah ignorující orientaci sudých or-hran. Pokud délka sudých hran, které jsou procházeny opačným směrem, přesáhne $\frac{c(Y^{***})}{2}$, otočíme směr sledu.

Krok 6: (přidání or-hran) Každé neorientované hraně přiřadíme orientaci sledu. Pokud je sudá or-hrana $[v_i, v_j]$ procházena proti své orientaci, přidáme do grafu dvě or-hrany $[v_i, v_j]$ a $[v_j, v_i]$. Nechť je \tilde{G} výsledný graf, obrázek 3.13f).

Krok 7: (konečný E -tah) Určíme E -tah na \tilde{G} .

Algoritmus SMALLARCS má časovou složitost $O(\max\{|V|^3, |Y|^3\})$.



Obrázek 3.13: Algoritmus SMALLARCS. a) kompletní graf G^* , b) nahrazení hran minimálními cestami, G^{**} , c) minimální kostra grafu a minimální párování, d) smíšený graf \hat{G} , e) smíšený graf G^{***} , f) orientovaný graf \tilde{G}

Zdroj: Autor

Pokud distanční matice splňuje trojúhelníkovou nerovnost, algoritmus nalezne E -tah $T_{SMALLARCS}$, délky $c(T_{SMALLARCS})$, splňující nerovnost

$$c(T_{SMALLARCS}) \leq \frac{1}{2}(3c(T) + c(Y)) . \quad (3.46)$$

Lepších výsledků lze dosáhnout, pokud jsou pro výpočet použity oba algoritmy, LARGEARCS i SMALLARCS. Necht' $c(T_{LSARCS}) = \min\{c(T_{LARGEARCS}), c(T_{SMALLARCS})\}$ a je splněna trojúhelníková nerovnost, pak $c(T_{LSARCS}) \leq \frac{9c(T)}{5}$.

Platnost tohoto tvrzení závisí na skutečnosti, zda $c(T_{SMALLARCS}) \leq c(T_{LARGEARCS})$ a $c(Y) \leq \frac{3c(T)}{5}$. Pokud je $c(Y) \geq \frac{3c(T)}{5}$, může být nerovnost (3.45) nahrazena $c(T_{LARGEARCS}) \leq \frac{9c(T)}{5}$. Pokud je $c(Y) \leq \frac{3c(T)}{5}$, nerovnost (3.46) může být nahrazena $c(T_{SMALLARCS}) \leq \frac{9c(T)}{5}$.

Lukka a Salminen (1987) přizpůsobili TSP heuristiku vkládání or-hran pro řešení SCP, kterou navrhli Rosenkrantz, Stearns a Lewis (1977). Tato heuristika má časovou složitost $O(|Y|^2, \log|Y|)$. Nicméně, na náhodně generovaných úlohách kvalita řešení vkládacího algoritmu spadá někde do rozmezí $c(T_{LARGEARCS})$ a $c(T_{SMALLARCS})$ a je blíže k lepší z těchto dvou hodnot. Jeden z hlavních rysů této vkládací heuristiky spočívá v její flexibilitě. Může být snadno upravena pro více typů vozidel a rozšířena o dodatečné omezující podmínky. Frederickson, Hecht a Kim navrhli verzi vkládací heuristiky SCP pro s -vozidel, kde je hledána minimální délka nejdelšího sledu. Tato verze má časovou složitost $O(\max\{|V|^3, |Y|^3\})$.

3.5 Kapacitně omezený svozně-rozvozní problém (CARP – Capacited Arc Routing Problem)

V CARP má každá neorientovaná hrana (v_i, v_j) , nebo or-hrana $[v_i, v_j]$, nezáporné ohodnocení a požadavek obsluhy w_{ij} . Je předpokládáno, že homogenní vozový park s s -vozidly a kapacitou vozidla Q_s , $s = 1, \dots, k$, je umístěn do depa ve vrcholu v_1 . Řešení CARP spočívá v určení uzavřeného sledu obsahujícího všechny hrany obsluhy tak, že celkové požadavky hran obsluhy obsluhované vozidly nepřekročí jejich kapacity Q_s . Vozidlo může projet hranou obsluhy, aniž by ji obsloužilo, a všechny hrany obsluhy musejí být obslouženy právě jednou. CARP, kde všechny mají všechny hrany (v_i, v_j) , nebo or-hrany $[v_i, v_j]$, požadavek na obsluhu $w_{ij} > 0$, je zobecněným kapacitně omezeným problémem čínského pošťáka (CCPP – *Capacited Chinese Postman Problem*).

3.5.1 Orientovaný kapacitně omezený svozně-rozvozní problém (DCARP – Directed Capacitated Arc Routing Problem)

Opět budeme předpokládat, že graf G byl procedurou předběžného zpracování transformován na graf G' . V případě orientovaného CARP jsou binární proměnné x_{ijk} rovny 1, pokud vozidlo k prochází hranu $[v_i, v_j]$ z v_i do v_j . Binární proměnné x_{ijk}^R jsou rovny 1, pokud vozidlo k obsluhuje hranu $[v_i, v_j]$ z v_i do v_j . Proměnná x_{ijk} je shora ohraničená hodnotou 1, což zaručí, že vozidlo projede hranu v daném směru maximálně jedenkrát. Všechny hrany $[v_i, v_j]$ s požadavkem $w_{ij} > 0$ mají být obslouženy a zbývající hrany pouze projety. Množina vrcholů S je podmnožinou libovolných vrcholů grafu G' , které jsou součástí podcyklu, indukovaného těmito vrcholy.

Formulace úlohy ILP je podle [22] následující:

Problém DCARP

$$\text{Min} \sum_{k=1}^s \sum_{[v_i, v_j] \in Y'} c_{ij} x_{ijk} \quad (3.47)$$

Za podmínek

$$\sum_{[v_j, v_i] \in Y'} x_{jik} - \sum_{[v_i, v_j] \in Y'} x_{ijk} = 0 \quad (v_i \in V, k = 1, \dots, s) \quad (3.48)$$

$$\sum_{k=1}^s (x_{ijk}^R + x_{jik}^R) = \begin{cases} 0 & \text{pokud } w_{ij} = 0 \\ 1 & \text{pokud } w_{ij} > 0 \end{cases} \quad ([v_i, v_j] \in Y') \quad (3.49)$$

$$x_{ijk} \geq x_{ijk}^R \quad ([v_i, v_j] \in Y', k = 1, \dots, s) \quad (3.50)$$

$$\sum_{[v_i, v_j] \in Y'} q_{ij} x_{ijk}^R \leq Q \quad (k = 1, \dots, s) \quad (3.51)$$

$$\left. \begin{aligned} \sum_{v_i, v_j \in S} x_{ijk} &\leq |S| - 1 + n^2 u_k^S \\ \sum_{v_j \in S, v_i \notin S} x_{ijk} &\geq 1 - r_p^S \\ u_k^S + r_k^S &\leq 1 \\ u_k^S, r_k^S &\in \{0, 1\} \end{aligned} \right\} (S \subseteq V \setminus \{v_1\}, S \neq \{\emptyset\}, k = 1, \dots, s) \quad (3.52)$$

$$x_{ijk}, x_{ijk}^R \in \{0,1\} \quad (k = 1, \dots, s) \quad (3.53)$$

Podmínky (3.48) zajistí, že každé k -té vozidlo navštíví vrchol a opět ho opustí. Podmínky (3.49) zajistí, že budou obslouženy pouze or-hrany obsluhy. Podmínky (3.50) zajistí, že or-hrana bude obsloužena pouze vozidlem, které bude touto or-hranou projíždět. Podmínky (3.51) zajistí, že nebude překročena kapacita vozidla. Podmínky (3.52) zajistí, že výsledné řešení bude tvořeno jedním cyklem, začínajícím a končím v depu, a nebude obsahovat podcykly. Abychom pochopili, jak tyto podmínky fungují, všimněme si, že pro vozidlo k a množinu S může nabývat hodnoty 1 pouze jedna binární proměnná, u_k^S nebo r_k^S . Každý podcyklus indukovaný množinou vrcholů S a or-hranami, které v tomto podcyklu projede k -té vozidlo, musí být v trase vozidla spojen s množinou $V \setminus S$ (a s depem v_1), poté:

$$\sum_{v_i, v_j \in S} x_{ijk} > |S| - 1 \Rightarrow u_k^S = 1 \Rightarrow r_k^S = 0 \Rightarrow \sum_{v_i \in S} \sum_{v_j \notin S} x_{ijk} \geq 1.$$

Tato úloha ILP vede k řešení pomocí metody sečných nadrovin.

3.5.2 Neorientovaný kapacitně omezený svozně-rozvozní problém (UCARP – Undirected Capacited Arc Routing Problem)

Ve formulaci neorientovaného případu CARP navržené Belenguerem a Benaventem jsou strukturní proměnné x_{ijk} a x_{ijk}^R definovány pro relaci $i < j$. Nyní x_{ijk} představuje počet projetí hrany (v_i, v_j) k -tým vozidlem bez obslužení této hrany. Necht' je $X_R(S)$ množinou hran obsluhy, které spojují množiny S a $V \setminus S$, $X_R(S) = X(S) \cap \{(v_i, v_j) \in X_R : w_{ij} > 0\}$. Necht' $X_R^S(S)$ je množinou hran obsluhy, které mají oba incidující vrcholy v množině S . Indexem R je označena každá množina hran nebo or-hran, které jsou předmětem obsluhy. Formulace úlohy ILP je podle [23] následující:

Problém UCARP

$$\text{Min} \sum_{k=1}^s \sum_{(v_i, v_j) \in X'} c_{ij} (x_{ijk} + x_{ijk}^R) \quad (3.54)$$

Za podmíněk

$$\sum_{k=1}^s x_{ijk}^R = 1 \quad ((v_i, v_j) \in X' \wedge w_{ij} > 0) \quad (3.55)$$

$$\sum_{(v_i, v_j) \in X'} w_{ij} x_{ijk}^R \leq Q \quad (k = 1, \dots, s) \quad (3.56)$$

$$\sum_{(v_i, v_j) \in X(S)} x_{ijk} + \sum_{(v_i, v_j) \in X_R(S)} x_{ijk}^R \geq 2x_{hik}^R \quad (3.57)$$

$$(S \subseteq V \setminus \{v_i\}, S \neq \{\emptyset\}, k = 1, \dots, s, (v_h, v_l) \in X_R^S)$$

$$\sum_{(v_i, v_j) \in X(S)} x_{ijk} + \sum_{(v_i, v_j) \in X_R(S)} x_{ijk}^R \geq 2z_k^S \quad (S \subseteq V \setminus \{v_1\}, S \neq \{\emptyset\}, k = 1, \dots, s) \quad (3.58)$$

$$x_{ijk} \geq 0 \text{ celočíselně a } z_k^S \geq 0 \text{ celočíselně} \quad (3.59)$$

$$x_{ijk}^R \in \{0, 1\} \quad ((v_i, v_j) \in X', k = 1, \dots, s) \quad (3.60)$$

Podmínky (3.55) zajistí, že všechny hrany s kladným požadavkem jsou obslouženy vozidlem právě jednou. Kapacitní podmínky (3.56) zajistí, že nebude překročena kapacita vozidla. Podmínky (3.57) jsou obdobné jako podmínky (3.52) a zajistí, že pokud množina S obsahuje vrcholy hrany obsluhy (v_h, v_l) obsluhované vozidlem k , musí být množina S spojena s její doplňkovou množinou $S \setminus V$ jízdou stejným vozidlem. Podmínky (3.58) zajistí, že každá neprázdná množina S neobsahující depo musí být spojena se svým doplňkem $S \setminus V$ sudým počtem jízd vozidla. Bohužel neexistuje žádný způsob vyjádření těchto podmínek pouze prostřednictvím proměnných x_{ijk} a x_{ijk}^R . Necht' množina X_R^L je lichou podmnožinou množiny $X_R(S)$, $X_R^L \subseteq X_R(S)$. Následující přípustné podmínky liché řezové množiny, které jsou odvozeny z podmínky sudosti, zajistí ve většině případů sudý stupeň všech vrcholů grafu:

$$\sum_{(v_i, v_j) \in X(S)} x_{ijk} + \sum_{(v_i, v_j) \in X_R(S) \setminus X_R^L} x_{ijk}^R - \sum_{(v_i, v_j) \in X_R^L} x_{ijk}^R \geq 1 - |X_R^L|$$

$$(S \subseteq V \setminus \{v_1\}, S \neq \{\emptyset\}, X_R^L \subseteq X_R(S), |X_R^L| \text{ liché číslo}) \quad (3.58')$$

Tyto podmínky jsou přípustné, protože $\sum_{(v_i, v_j) \in X_R^L(S)} x_{ijk}^R \leq |X_R^L|$. Pokud je $\sum_{(v_i, v_j) \in X_R^L(S)} x_{ijk}^R = |X_R^L|$, musí být $\sum_{(v_i, v_j) \in X(S)} x_{ijk} + \sum_{(v_i, v_j) \in X_R(S) \setminus X_R^L} x_{ijk}^R \geq 1$, aby byla splněna podmínka sudosti, protože $|X_R^L|$ je liché číslo. Tato úloha ILP vede opět k řešení pomocí metody sečných nadrovin.

Tabulka 3.2: Metody řešení RPP

problém	exaktní algoritmy	heuristické algoritmy
URPP	NP-těžký. ILP algoritmus na bázi metody větví a mezí.	Heuristika (Frederickson).
DRPP	NP-těžký. ILP algoritmus na bázi metody větví a mezí.	Heuristika (Christofides).
SCP	NP-těžký. Není znám žádný exaktní algoritmus.	Largearcs a Smallarcs.
CARP	NP-těžký. Metoda větví a mezí (Belenguer a Benavert).	Augment-Insert (Pearn).

Zdroj: Autor na podkladu podkapitoly 3.4 a 3.5

3.6 Smíšený kapacitně omezený svozně-rozvozní problém (MCARP – Mixed Capacited Arc Routing Problem)

CARP je modelován na neorientovaném grafu. Jedná se o NP-těžký problém. Exaktní metody řešení jsou stále omezeny počtem obsluhovaných hran, které zvyšují počet podmínek v úlohách ILP. Tato skutečnost vysvětluje význam metod heuristických. Mezi dobré heuristiky poskytující přijatelná řešení patří Path-Scanning, Augment-Merge a Ulusoyova heuristika. Mnohem lepší výsledky lze získat řešením pomocí metaheuristik. Mezi tyto metaheuristiky patří například Tabu-Search, memetický algoritmus, Tabu-Scatter Search a Guided Local Search. Srovnání s metodami, které určují dolní hranice řešení (*lower bounds*), ukázalo, že většinu úloh s počtem hran obsluhy do 100, lze metaheuristikami optimálně vyřešit.

Ve skutečnosti je CARP příliš jednoduchý model na to, aby modeloval reálné sítě městských komunikací, na kterých je řešen například problém čištění komunikací. Neorientovaný graf může modelovat síť městských komunikací pouze tak, že jsou obě strany

obousměrných komunikací obsluhovány v libovolném směru paralelně. Tento model je běžný v oblastech s menšími dopravními proudy. Skutečné dopravní sítě obsahují jednosměrné komunikace, a obousměrné komunikace, kde jsou obě dvě strany obsluhovány nezávisle na sobě. Aby byl při řešení tento fakt respektován, řeší se CARP na smíšené dopravní síti a vzniká nový problém, který se nazývá smíšeným CARP (MCARP).

3.6.1 *Matematický model MCARP*

MCARP je definován na smíšeném grafu $G = (V, E, p)$, $E = X \cup Y$, kde V je množina vrcholů, X je množina neorientovaných hran a Y je množina or-hran. Vozový park čítající k identických vozidel s omezenou kapacitou Q je umístěn ve vrcholu v_1 , který se nazývá depem. Mohutnost množiny $X \cup Y$ je označena m . Každá hrana bez obsluhy (v_i, v_j) , nebo $[v_i, v_j]$, má ohodnocení c_{ij} . Podmnožina hran obsluhy $R = X_R \cup Y_R$ musí být vždy obsloužena vozidlem, kde $m_R = |R|$ a $R \subseteq E$. Tato podmnožina se skládá z podmnožiny neorientovaných hran obsluhy $X_R \in X$ a or-hran obsluhy $Y_R \in Y$, kde $m_{XR} = |X_R|$ a $m_{YR} = |Y_R|$. Každá hrana obsluhy (v_i, v_j) , nebo $[v_i, v_j]$, má požadavek na obsluhu w_{ij} a ohodnocení c_{ij}^R . Všechny požadavky a ohodnocení jsou nezáporná celá čísla.

Cílem je určit množinu minimálních tras vozidel, kde každá trasa začíná a končí ve vrcholu s depem, všechny hrany obsluhy jsou obslouženy v rámci jedné trasy (a to pouze jednou) a celkové požadavky na trase jsou omezeny kapacitou vozidla. MCARP je NP-těžký, protože v sobě zahrnuje několik obdobných úloh, které jsou také NP-těžké: UCARP, DCARP, MRPP (pokud je $Q = \infty$) a MCPP (pokud $Q = \infty$ a všechny hrany jsou předmětem obsluhy).

Matematický model MCARP je založen na formulaci modelu UCARP a MCPP na smíšeném grafu, který je popsán v oddílu 3.3.4 této práce. V úloze ILP pro MCARP jsou stejně jako v případě řešení MCPP obsaženy podmínky sudosti, a podmínky vyváženosti, které každé orientované a neorientované hraně přiřadí pouze jednu proměnnou. Tyto podmínky pomohou blížeji a těsněji formulovat úlohu na smíšeném grafu. Připomeňme tyto podmínky:

- *podmínky sudosti*, stupeň každého vrcholu je sudý,

- *podmínky vyváženosti.* Pro každou neprázdnou množinu vrcholů S je počet or-hran vcházejících do S minus počet or-hran vycházejících z S menší nebo roven počtu hran mezi S a $V \setminus S$.

Je dána libovolná podmnožina vrcholů $S \subseteq V \setminus \{v_1\}$ grafu $G = (V, E, p)$, $S \neq \{\emptyset\}$. Stejně jako ve formulaci úlohy ILP MCPP je $Y^+(S)$ množina or-hran vycházejících z S , $Y^-(S)$ množina or-hran vcházející do S a $X(S)$ množina hran spojující S a $V \setminus S$. Řezová množina indukovaná množinou S je definována stejně jako v MCPP $E(S) = Y^+(S) \cup Y^-(S) \cup X(S)$. Necht' $E_R^S(S)$ je množinou hran a or-hran obsluhy, které mají oba incidující vrcholy v množině S . $Y_R^+(S)$ je množinou or-hran obsluhy, které vycházejí z vrcholu z množiny S , a $Y_R^-(S)$ je množinou or-hran obsluhy, které vcházejí do vrcholu z množiny S . Jako $w(S)$ jsou označeny celkové požadavky hran a or-hran obsluhy z množiny $E_R(S) \cup E_R^S(S)$. K obsluze hran a or-hran z množiny $E_R(S) \cup E_R^S(S)$ je zapotřebí nejméně $k(S)$ vozidel, $k(S) = \frac{w(S)}{Q}$. Nevyváženost řezové množiny $E(S)$ je definována stejně jako v MCPP $u(S) = |Y^+(S)| - |Y^-(S)| - |X(S)|$.

Řešení MCARP je dáno množinou tras vozidel, kde proměnná x_{ij} vyjadřuje celkový počet bezobslužných průchodů hranami $(v_i, v_j) \in X$ a $[v_i, v_j] \in Y$. Přidáním x_{ij} kopií hran do grafu $G_R = (V, R, p)$, $R = X_R \cup Y_R$, je vytvořen E -graf odpovídající řešení MCARP. Necht' množina E_R^L je lichou podmnožinou množiny $E_R(S)$, $E_R^L \subseteq E_R(S)$. Podle [24] lze úlohu ILP zjednodušeně, za pomoci množiny hran E , formulovat následovně:

$$\text{Min} \sum_{\substack{(v_i, v_j) \in R \\ \text{nebo } [v_i, v_j] \in R}} c_{ij}^R + \sum_{\substack{(v_i, v_j) \in E \\ \text{nebo } [v_i, v_j] \in E}} c_{ij} x_{ij} \quad (3.59)$$

Za podmínek:

$$\sum_{(v_i, v_j) \in X(S)} x_{ij} - \sum_{[v_i, v_j] \in Y^+(S)} x_{ij} + \sum_{[v_i, v_j] \in Y^-(S)} x_{ij} \geq u(S) \quad (S \subset V, S \neq \{\emptyset\}) \quad (3.60)$$

$$\sum_{\substack{(v_i, v_j) \in E(S) \\ \text{nebo } [v_i, v_j] \in E(S)}} x_{ij} \geq 2k(S) - |E_R(S)| \quad (S \subset V \setminus \{v_1\}, S \neq \{\emptyset\}) \quad (3.61)$$

$$\sum_{\substack{(v_i, v_j) \in E(S) \\ \text{nebo } [v_i, v_j] \in E(S)}} x_{ij} \geq 1 \quad (3.62)$$

$$(S \subset V \setminus \{v_1\}, S \neq \{\emptyset\}, E_R^L \subseteq E_R(S), |E_R^L(S)| \text{ liché číslo})$$

$$x_{ij} \geq 0 \text{ a celočíselně} \quad ((v_i, v_j) \in E \text{ nebo } [v_i, v_j] \in E) \quad (3.63)$$

Podmínka vyváženosti (3.60) zajistí vyváženost množin hran indukovaných množinou S ve smíšeném E -grafu. Podmínka (3.61) je nazývána kapacitní podmínkou a může být vysvětlena následujícím způsobem. Jak je uvedeno výše, k obsluze celkových požadavků hran obsluhy z množiny $E_R(S) \cup E_R^S(S)$ je potřebných nejméně $k(S)$ vozidel.

Každé vozidlo, které obslouží hranu z $E_R(S) \cup E_R^S(S)$, projde řezovou množinou $E(S)$ dvakrát, jednou z S do $V \setminus S$ a podruhé z $V \setminus S$ do S . Celkový počet průchodů hran bez obsluhy náležejících do řezové množiny $E(S)$ je tedy roven $2k(S) - |E_R(S)|$. Samozřejmě, že předchozí tvrzení platí jen za předpokladu, že $2k(S) - |E_R(S)| > 0$. Podmínka (3.62) je nazývána podmínkou liché řezové množiny a je přímo odvozena z podmínky sudosti. Tato podmínka je ekvivalentem podmínky (3.30) v matematickém modelu úlohy ILP pro MCPP.

Kapacitní podmínky (3.61) je možné rozepsat i v orientované verzi. Celkový počet průchodů řezovou množinou z $V \setminus S$ do S je roven přinejmenším $k(S)$, poté lze zapsat orientovanou kapacitní podmínku:

$$\sum_{[v_i, v_j] \in Y^-(S)} x_{ij} + \sum_{(v_i, v_j) \in X(S)} x_{ij} \geq k(S) - |Y_R^-(S)| - |X_R(S)| \quad (3.64)$$

Celkový počet průchodů řezovou množinou z S do $V \setminus S$ je roven také přinejmenším $k(S)$, pro opačný směr tedy platí:

$$\sum_{[v_i, v_j] \in Y^+(S)} x_{ij} + \sum_{(v_i, v_j) \in X(S)} x_{ij} \geq k(S) - |Y_R^+(S)| - |X_R(S)| \quad (3.65)$$

Nicméně, podmínky (3.64) a (3.65) nebudou použity, protože jak lze dokázat, jsou zahrnuty v podmínkách (3.60) a (3.61).

Výše uvedená matematická formulace problému MCARP není kompletní, protože může poskytovat celočíselná řešení, které neodpovídají přípustnému řešení MCARP. Není

známa žádná jiná matematická formulace používající výhradně jedinou proměnnou x_{ij} . Z existence jediné proměnné x_{ij} v úloze ILP vyplývá fakt, že poskytuje informaci pouze o bezobslužných průchodech hran na množině tras vozidel, ne však v jednotlivých trasách.

Lze dokázat, že každá podmínka ve formulaci UCARP může být obecně převedena na platnou podmínku pro MCARP. Formulace UCARP používá stejné proměnné x_{ij} , které jsou asociovány pouze s neorientovanými hranami. Uvažujme MCARP definovaný na smíšeném grafu $G = (V, E, p)$, kde $E = X \cup Y$. Necht' $G' = (V, X \cup X', p)$ je neorientovaný graf, získaný nahrazením všech or-hran z množiny Y neorientovanou hranou, která je zařazena do množiny X' a která má shodné ohodnocení a požadavek jako or-hrana. Necht' je or-hrana $h \in Y$ hranou, z které vznikla neorientovaná hrana $h' \in X'$. Všimněme si, že každé přípustné řešení MCARP je také přípustným řešením UCARP definovaném na grafu G' (kde lze projít hranou v obou směrech). Necht' budou obecně koeficienty $a \in R^{X \cup X'}$ a omezení $b_0 \in R$ taková, že obecně zapsaná podmínka $\sum_{h' \in X \cup X'} a_{h'} x_{h'} \geq b_0$ je přípustnou podmínkou pro UCARP, definovaném na grafu G' . Pro MCARP definovaném na grafu G lze poté tuto podmínku obecně zapsat jako $\sum_{h' \in X} a_{h'} x_{h'} + \sum_{h' \in X'} a_{h'} x_{h'} \geq b_0$.

3.6.2 Identifikace podmínek v metodě sečných nadrovin pro MCARP

Metoda sečných nadrovin je schopna nalézt dolní hranici řešení MCARP a je založena na základě výše popsané formulace MCARP. Tato metoda již byla popsána v řešení MCPP. V následujícím textu bude podle [25] popsána pouze samotná identifikace porušených podmínek úlohy ILP. Obdobně jako v případě MCPP je v každé iteraci řešena úloha LP, která obsahuje: účelovou funkci (3.59), nezáporné podmínky (3.60) a přípustné podmínky (3.61)–(3.63). Optimální řešení úlohy ILP je dolní hranicí řešení MCARP. I když je řešení úlohy LP celočíselné, nemusí odpovídat optimálnímu řešení MCARP, protože její formulace není kompletní. Stejně jako v případě MCPP je v každé iteraci metody sečných nadrovin hledán soubor podmínek, které byly porušeny v optimálním řešení ILP, a které jsou poté přidány do matematického modelu ILP. Algoritmus končí, pokud již nebyla nalezena žádná podmínka, která by byla v optimálním řešení porušena.

Soubor dodatečných podmínek pro úlohu ILP lze získat následovně. Začneme ve vrcholu $v \in V \setminus \{v_1\}$, určíme množiny vrcholů W_i takové, že $W_0 = \{v\}$ a pro $i \geq 1$ jsou do těchto množin přidávány vrcholy následovně: W_{i+1} je získána přidáním vrcholů, které incidují

v grafu G alespoň s jedním vrcholem z množiny W_i (mimo vrcholu v_1 , který je depem). Tato procedura končí, pokud množiny vrcholů obsahují všechny vrcholy mimo depa. Do úlohy ILP jsou poté přidány podmínky (3.60) korespondující s každou množinou W_i a $V \setminus (W_i \cup \{v_1\})$, ale jen v případě, že jsou jejich pravé strany kladné, a také podmínka (3.62) v případě, že $|E_R(W_i)|$ je liché číslo.

V každé iteraci je použito několik identifikačních procedur, které tyto podmínky generují. Necht' je vektor $x_0^k \in R^{X \cup Y}$ optimálním řešením úlohy ILP v k -té iteraci a $x_0^k(h)$ složka vektoru x_0^k vztahující se k hraně h .

Dodatečné podmínky (3.60) mohou být identifikovány následovně. V grafu G jsou nově ohodnoceny hrany a vznikne graf $G^k = (V, E, p, c^k(h))$. Necht' má každá hrana obsluhy $h \in X_R \cup Y_R$ ohodnocení $c^k(h) = x_0^k(h) + 1$ a $c^k(h) = x_0^k(h)$ pro $h \notin X_R \cup Y_R$. Je patrné, že podmínka vyváženosti (3.60), korespondující s množinou $S \subseteq V$, je porušena vektorem x_0^k v případě, že $\sum_{h \in X(S)} h - \sum_{h \in Y^+(S)} h + \sum_{h \in Y^-(S)} h < 0$. Množina S , pro kterou platí tato nerovnost, je nevyváženou množinou. Na grafu G^k se pomocí úlohy určení maximálního toku na grafu s $|V| + 2$ vrcholy hledá nejvíce nevyvážená množina S ($\max_{S \subseteq V} \{u(S)\}$). Poté jsou do úlohy LP přidány podmínky, které tuto nevyváženost odstraní.

Porušené podmínky (3.61) a (3.62) mohou být identifikovány na grafu, kde jsou or-hrany považovány za neorientované hrany. Identifikační procedura bude stručně popsána níže.

Necht' $G^{k*} = (V, X^{k*}, p, c_0(h))$, je neorientovaný graf indukovaný hranami $h \in E$, $E = X \cup Y$, které jsou v k -té iteraci procházeny vozidlem, tedy $x_0(h) > 0$. Množina hran X^{k*} je tvořena neorientovanými hranami X a or-hranami Y , u kterých je zanedbána orientace. Identifikaci porušených podmínek lichého řezu (3.62) lze dosáhnout v polynomiálním čase pomocí Padberg-Raova algoritmu minimálního lichého řezu.

Bohužel není znám žádný polynomiální algoritmus, který dokáže identifikovat porušené kapacitní podmínky (3.61).

Aby bylo možné identifikovat porušené podmínky (3.61) lze použít dvou heuristik, které vygenerují několik množin vrcholů $S \subseteq V \setminus \{v_1\}$, na kterých bude toto porušení patrné. Zároveň budou na každé této množině vrcholů testovány podmínky (3.60) a (3.62). První heuristika je založena na nalezení spojitých komponent grafu G^{k*} . Druhá heuristika je

založena na procentuálním navýšení jednotlivých požadavků na hranách a nalezení minimálního řezu v pomocném grafu. Procedura identifikuje relaxovanou verzi kapacitních podmínek nazývanou zlomkovými kapacitními omezeními a je aplikována na několik různých hodnot procentuálního navýšení vah vrcholů.

Předešlé identifikační algoritmy jsou kombinovány podle následující strategie: první je použita heuristika, která generuje množiny vrcholů $S \subseteq V \setminus \{v_1\}$. V případě, že tyto heuristiky nenaleznou žádné porušené podmínky typu (3.60)–(3.62), je použita exaktní metoda identifikace pro podmínky (3.60) a (3.62).

4 ŘEŠENÍ ÚKOLU A JEHO VÝSLEDKY

Řešení úkolu je rozděleno na dvě části – na část teoretickou, popisující metodiku tvorby tras obslužných vozidel provádějících čištění a údržbu pozemních komunikací, která byla sestavena na základě analýzy v kapitole 3, a na část aplikační, ve které je popsáno začlenění této metodiky do SW nástroje NetOpt. Části teoretická i aplikační jsou vypracovány v obecné rovině tak, aby byla metodika v SW nástroji použitelná i pro řešení ostatních problémů z oblasti svozně-rozvozních úloh s obsluhou hran (např. svoz komunálního odpadu, zimní údržba pozemních komunikací atd.). Postup při sestavování metodiky je popsán vývojovým diagramem v příloze 4.

4.1 Přístup k řešení

Výběr vhodné kategorie metod řešících svozně-rozvozní úlohy s obsluhou hran je zdůvodněn v podkapitole 4.5. Nejlepších řešení na náhodně generovaných instancích s větším objemem vstupních dat dosahovaly metaheuristické algoritmy řešící MCARP. Objem vstupních dat je dán mohutností množiny hran a vrcholů smíšeného grafu reprezentujícího síť městských komunikací. Použití tohoto typu algoritmu se oproti jiným metodám osvědčilo pro instance větších rozměrů zejména v čase potřebném pro výpočet a uspokojivém řešení, které se blíží optimálnímu.

SW nástroj s implementovanou metodikou byl navržen tak, aby byl schopen pracovat s reálnými daty sítě městských komunikací ve vektorovém formátu a výstupy v podobě navržených tras bylo možné zobrazit v kterékoliv GIS aplikaci, i na GPS zařízení.

4.2 Metodika tvorby tras obslužných vozidel provádějících ČUPK

V této podkapitole jsou popsány principy metaheuristického algoritmu, spadajícího do kategorie MCARP, řešícího optimalizaci tras obslužných vozidel provádějících čištění a údržbu pozemních komunikací. Nejprve budou popsány konstruktivní heuristické algoritmy, které poskytují dobré počáteční řešení, a dále genetický algoritmus a metody lokálního prohledávání, které toto řešení dokáží vylepšit. Vzájemnou vazbou a spojením těchto typů algoritmů v jeden celek vzniká výsledná metaheuristika.

4.2.1 Konstruktivní heuristiky pro MCARP

Pro řešení MCARP budou dále zobecněny tři heuristiky: konstruktivní Path Scanning, konstruktivní Augment-Merge a dvoufázová Ulusoyova heuristika. Původní vlastnosti těchto

heuristik jsou: na síti existuje jediné depo, vozový park je homogenní a není dovoleno rozdělovat požadavky na hranách. Počet vozidel je předem známý. Přípustné řešení lze získat pouze tehdy, pokud není překročena kapacita vozidel a je stanovena jejich dojezdová vzdálenost tak, aby bylo možné každým vozidlem dosáhnout jakékoliv hrany obsluhy, obsloužit ji a vrátit se zpět do depa. Ohodnocení trasy je součtem ohodnocení hran v trase. Cílem je nalézt množinu minimálních tras, které pokrývají všechny hrany obsluhy.

Dále je zde popsáno rozšíření heuristik o dalších pět vlastností, která řešení více přiblíží reálným podmínkám:

- problém lze řešit na smíšeném multigrafu, ve kterém může mezi dvěma vrcholy existovat více paralelních hran,
- zákaz otáčení (U-turns) a penalizaci při odbočení vozidla v křižovatce,
- limitování dojezdové vzdálenosti proměnnou L (maximální délka jakékoliv trasy v řešení CARP),
- v grafu mohou existovat nesouměrné hrany, které mají v každém směru různá ohodnocení, a hrany s různým ohodnocením při obsluze a průchodu bez obsluhy,
- přítomnost více vrcholů reprezentujících deponovací místa pro vyprazdňování, resp. vykládku vozidel.

Tyto dodatečné vlastnosti hrají klíčovou roli při řešení praktických aplikací. Například při obsluze komunikace, která je paralelní s jinou komunikací, se problém řeší lépe na multigrafu. Také musejí být brány v úvahu zákazy otáčení a pravidla pro chování dopravního prostředku v křižovatce. Při odbočování vozidla na světelné křižovatce dochází k časovým ztrátám, které lze vyjádřit penalizací. Limitací dojezdové vzdálenosti lze korigovat dojezd vozidla nebo maximální jízdní dobu. Komunikace s klesáním, či stoupáním může být ve výpočtu uvažována jako nesouměrná hrana (neorientovaná hrana, která má v každém směru jiné ohodnocení). V praktické aplikaci může být na síti komunikací umístěno více deponovacích míst pro vykládku vozidel před návratem do depa.

V heuristikách rozšířených o tyto vlastnosti platí stejná pravidla jako v jejich základních verzích.

Reálnou síť městských komunikací lze dobře modelovat na smíšeném grafu. Bezobslužné komunikace jsou v grafu reprezentovány buď jednou or-hranou, pokud se jedná o jednosměrnou komunikaci, nebo dvěma opačně orientovanými hranami, pokud se jedná

o obousměrnou komunikaci. Komunikace, které jsou předmětem obsluhy, lze reprezentovat v grafu následovně: obousměrné komunikace, které lze obsloužit v libovolném směru, jsou reprezentovány jednou neorientovanou hranou, obousměrné komunikace, které je nutné obsluhovat v každém směru zvlášť, jsou reprezentovány dvěma opačně orientovanými hranami a jednosměrné komunikace jsou reprezentovány jedinou or-hranou. Ve smíšeném multigrafu lze navíc modelovat jednosměrné komunikace, které jsou příliš široké, nemohou být obslouženy oboustranně a obsluha si vyžaduje více jak jeden průjezd vozidla. Tyto komunikace jsou reprezentovány orientovanou multihranou.

Aby bylo možné lépe popsat reálnou síť městských komunikací za předpokladu, že jednotlivé komunikace jsou obsluhovány paralelně, je smíšený graf uvažován jako orientovaný. Každá neorientovaná hrana je nahrazena dvojicí opačně orientovaných hran. V přípustném řešení je obsloužena pouze jedna z těchto or-hran. Aby byla tato podmínka dodržena, je oběma or-hranám přiřazen ukazatel, který zajistí v případě výběru jednoho ze směrů obsluhy, že obě tyto or-hrany budou označeny za obsloužené. Pokud jsou komunikace obsluhovány v obou směrech, není tento ukazatel použit.

4.2.2 Úprava heuristik řešících MCARP

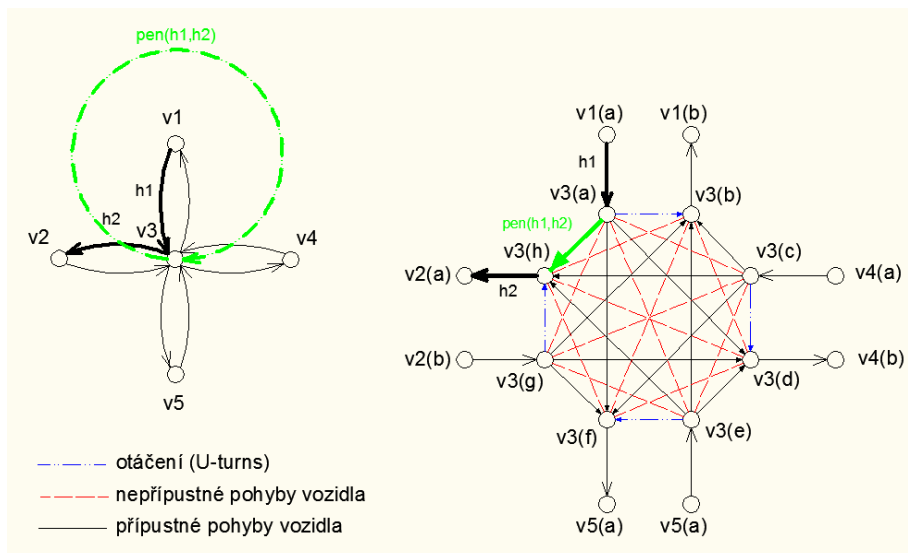
V následujících heuristikách je smíšený multigraf $G = (V, E, p)$, $E = X \cup Y$, převeden na orientovaný graf $G^* = (V, Y^*, p)$ nahrazením každé neorientované hrany dvěma opačně orientovanými hranami a přidáním fiktivní smyčky s nulovým ohodnocením pro depo a deponovací místa (úloha těchto smyček bude vysvětlena v dalším textu). Y^* je množina or-hran h_i , označených indexy $i = 1, \dots, m_{Y^*}$. Označení hran indexy, na rozdíl od identifikace hran přilehlými vrcholy, od sebe odliší multihrany. Každá or-hrana $p(h) = [v_{poc}, v_{kon}]$ má počáteční vrchol $v_{poc}(h)$, koncový vrchol $v_{kon}(h)$ a bezobslužné ohodnocení $c(h)$.

Počet hran obsluhy grafu G je $m_R = m_{XR} + m_{YR}$. Hrany obsluhy grafu G odpovídají v grafu G^* podmnožině or-hran $Y_R^* \subseteq Y^*$ a platí $m_{YR}^* = |Y_R^*| = 2m_{XR} + m_{YR}$. V grafu G^* má každá or-hrana obsluhy požadavek $w(h)$, ohodnocení $c^R(h)$ a ukazatel $opc(h)$, který představuje hranu h , procházenou opačným směrem, $p(opc(h)) = [v_{kon}, v_{poc}]$. Každá or-hrana z množiny Y_R je reprezentována v množině Y_R^* jednou or-hranou h s ukazatelem $opc(h) = 0$. Každá neorientovaná hrana z množiny X_R přiřadí do množiny or-hran Y_R^* dvě

opačně orientované hrany h a l , s ukazateli $opc(h)=l$ a $opc(l)=h$, shodnými požadavky $w(h)=w(l)$ a pokud je hrana nesouměrná, tak s odlišnými ohodnoceními v obou směrech.

Problém zákazu otáčení a penalizace při odbočení vozidla je řešen následovně. V grafu G^* je každé or-hraně h přiřazena množina $w(h)$ obsahující or-hrany l , pro které platí $l \in w(h)$, $v_{kon}(h)=v_{poc}(l)$, a je povolen pohyb vozidla (h,l) . Definujme přípustný sled od or-hrany h k or-hraně l jako posloupnost or-hran s obsluhou a bez obsluhy $T = \{h = h_1, h_2, \dots, h_k = l\}$. Ve sledu T platí $h_{i+1} \in w(h_i)$ a $i = 1, \dots, k-1$. Dvojici or-hran (h_i, h_{i+1}) je přiřazena penalizace otáčení, resp. odbočení, $pen(h_i, h_{i+1}) \geq 0$. Délka sledu T je poté definována vztahem (4.1). Z obrázku 4.1 jsou patrné zakázané pohyby vozidla, přípustné pohyby vozidla a jejich penalizace v křižovatce. Pomocí upraveného Dijkstrova algoritmu pro nalezení nejkratší cesty je získána distanční matice D mezi or-hranami o rozměru $m_Y^* \times m_Y^*$, ve které $d(h,l)$ představuje délku minimální cesty mezi or-hranami h a l . Aby byly usnadněny operace vypouštění a vkládání hran ve sledu T , není v $d(h,l)$ zahrnuto ohodnocení $c(h)$ a $c(l)$. Pokud je mezi or-hrany h a l vložena or-hrana obsluhy h_{vlz} , změní se $d(h,l)$ v distanční matici D jednoduše o hodnotu $d(h, h_{vlz}) + c^R(h_{vlz}) + d(h_{vlz}, l) - d(h, l)$.

$$c(T) = pen(h, h_2) + \sum_{i=2}^{k-1} (c(h_i) + pen(h_i, h_{i+1})) \quad (4.1)$$



Obrázek 4.1: Zakázané a přípustné pohyby vozidla v křižovatce, otáčení v křižovatce a jejich penalizace

Zdroj: Autor

Nechť je Λ množina smyček deponovacích míst. Pro snazší výpočet vzdáleností mezi jednou or-hranou a depem v_1 nebo jednou or-hranou a deponovacím vrcholem je do množiny or-hran Y^* přidána jedna fiktivní smyčka s nulovým ohodnocením S , $p(S) = [v_1, v_1]$, pro depo a jedna fiktivní smyčka s nulovým ohodnocením J , $p(J) = [v_{depon}, v_{depon}]$, pro každý deponovací vrchol. Tyto smyčky nahrazují vrchol s depem nebo deponovacím místem. Poté $|Y^*| = m_Y^* = 2|X| + |Y| + |P| + 1$. Nejvhodnější deponovací smyčku $dpn(h)$, která bude v trase končící or-hranou obsluhy h následovat (deponovací vrchol v_{depon}), lze určit podle vztahu (4.2), kde Λ_Y je množina deponovacích smyček a I je ohodnocení přiřazené deponovacím operacím (např. čas potřebný k vykládce). Minimální sumu ohodnocení or-hran $kon(h)$, které jsou potřebné k dokončení trasy za or-hranou h , lze vypočítat podle (4.3).

$$dpn(h) = J : \arg \min \{d(h, J) + I + d(J, S) : J \in \Lambda_Y\} \quad (4.2)$$

$$kon(h) = d(h, dpn(h)) + I + d(dpn(h), S) \quad (4.3)$$

Řešení MCARP je dáno množinou tras $T^R = \{T_1^R, \dots, T_{|T^R|}^R\}$, potřebných k obslužení všech hran grafu. Počet prvků množiny T^R je roven $|T^R|$ a T_i^R je v pořadí i -tý prvek množiny. Trasa $T_i^R = \{h_1, \dots, h_{m_R}\}$ je uspořádanou podmnožinou or-hran obsluhy z množiny Y_R^* . Mezi těmito or-hranami obsluhy jsou implicitně předpokládány minimální bezobslužné cesty. Množství nakládky na trase, $nak(T_i^R)$ je suma požadavků, která nesmí překročit kapacitu vozidla Q (4.4). Celková suma ohodnocení trasy $c(T_i^R)$ zahrnuje ohodnocení hran obsluhy, ohodnocení minimálních cest spojujících tyto hrany a minimální sumu ohodnocení hran pro dokončení trasy. $c(T_i^R)$ nesmí překročit předem stanovenou hodnotu L , představující např. omezení dojezdové vzdálenosti vozidla (4.5). Hodnota řešení je dána sumou ohodnocení jednotlivých tras z množiny T^R .

$$nak(T_i^R) = \sum_{j=1}^{m_R} w(h_j) \leq Q \quad (h_j \in T_i^R, i = 1, \dots, |T^R|) \quad (4.4)$$

$$c(T_i^R) = d(S, h_1) + \sum_{j=1}^{m_R-1} (c(h_j) + d(h_j, h_{j+1})) + c(h_{m_R}) + kon(h_{m_R}) \leq L \quad (h_j \in T_i^R, i = 1, \dots, |T^R|) \quad (4.5)$$

4.2.3 Heuristika Path-Scanning (PS)

PS je konstruktivní sekvenční heuristika, odvozená z heuristiky *nejbližšího souseda* pro řešení TSP, která ve své relaxované verzi nalezne jedinou trasu obsahující všechny hrany obsluhy. Jediná trasa T_1^R obsahující všechny hrany obsluhy je nazývána *úplnou trasou*, množina tras pro MCARP je tedy $T^R = \{T_1^R\}$. Hledaná trasa je na svém konci prodlužována o nejbližší hranu obsluhy. V ARP je ve většině případů vzdálenost od poslední obsloužené hrany h k dalším ještě neobslouženým hranám l_i totožná (např. na hranu obsluhy h navazuje i -tý počet hran obsluhy l_i). Poté je podle [26] nutné uskutečnit výběr další hrany obsluhy l minimalizováním, resp. maximalizováním jednoho z pěti následujících kritérií:

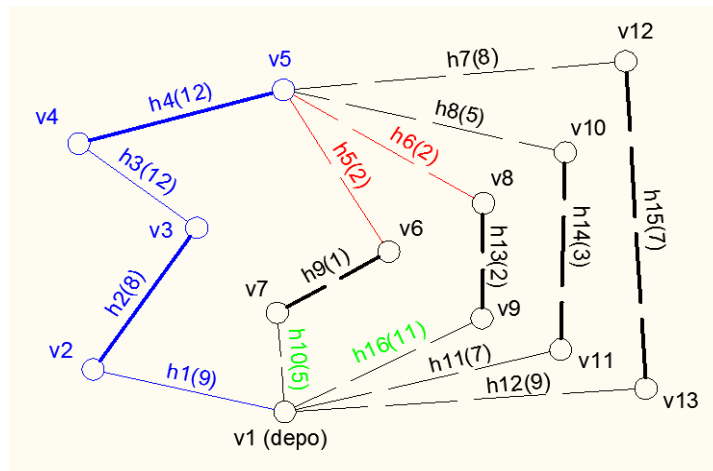
- $krit_1(l)$: $\min(kon(l))$, minimalizace ohodnocení trasy za hranou l do depa,
- $krit_2(l)$: $\max(kon(l))$, maximalizace ohodnocení trasy za hranou l do depa,
- $krit_3(l)$: $\min\left(\frac{w(l)}{c^R(l)}\right)$, minimalizace poměru vyjadřujícího „produktivitu“ hrany l ,
- $krit_4(l)$: $\max\left(\frac{w(l)}{c^R(l)}\right)$, maximalizace poměru vyjadřujícího „produktivitu“ hrany l ,
- $krit_5(l) = krit_2(l)$, pokud je $nak(T_1^R) \leq \frac{Q}{2}$, jinak $krit_5(l) = krit_1(l)$.

Dokud je kapacita vozidla využita maximálně na polovinu, kritérium $krit_5$ maximalizuje sumu ohodnocení hran potřebných pro dokončení trasy, poté tuto sumu minimalizuje.

PS heuristika řeší úlohu celkem pětkrát, pokaždé pro jiné kritérium $krit_1, \dots, krit_5$. Jako finální je vybráno nejlepší z těchto pěti řešení.

Na obrázku 4.2 je příklad volby následující hrany obsluhy kritériem $krit_1$. Hrany obsluhy jsou zde vyznačeny silně, tenká hrana odpovídá minimálním bezobslužným cestám mezi hranami obsluhy a depo v_1 je tomto případě zároveň deponovacím místem. Trasa končí hranou h_4 . Minimální vzdálenost k hranám obsluhy, které nebyly ještě obslouženy, je rovna hodnotě 2 a množina kandidátů na další hranu trasy je $Z = \{h_5, h_6\}$. Pro výběr další hrany

obsluhy trasy je použito kritérium $krit_1$, které minimalizuje zpáteční cestu do depa $\min\{kon(h_9), kon(h_{13})\} = \min\{5, 11\}$, a trasa je prodloužena o hranu h_9 .



Obrázek 4.2: Iterace Path-Scanning heuristiky s kritériem $krit_1$

Zdroj: Autor

Pro jedno zvolené kritérium $krit \in \{krit_1, krit_2, krit_3, krit_4, krit_5\}$ lze získat řešení v čase $O(m_R)$, tj. v $O(n^2)$ pro reálnou síť městských komunikací s $m_R \leq m^* \approx 4n$.

Náhodným výběrem další hrany obsluhy l na trase lze získat více jak pět variant řešení a v průměru zlepšit hodnotu řešení. Existují dvě možnosti, jak tento výběr provést. První možností je v každé iteraci náhodně vybrat jedno z pěti kritérií $krit_1, \dots, krit_5$, každé se stejnou pravděpodobností, které určí další hranu obsluhy l . Druhou možností je ignorování kritérií a místo výběru hrany minimalizující kritérium $krit$ na množině Z provést náhodný výběr hrany z množiny nejbližších hran Z .

4.2.4 Heuristika Augment-Merge (AM)

Nejprve bude podle [22] stručně popsána AM pro UCARP. Počátečním řešením AM je množina triviálních řešení T^{X_R} , kde každá trasa $T_1^{X_R}, \dots, T_{m_{XR}}^{X_R}$ obsahuje pouze jednu hranu obsluhy h . Tyto hrany jsou na začátku seřazeny do klesající posloupnosti podle svého ohodnocení $c(h)$. Pokud již není součástí jiné trasy, je pro každou hranu obsluhy $p(h) = (v_i, v_j)$, $h \in X_R$, nalezena triviální trasa s ohodnocením $c(T^{X_R}) = \{d_{\min}(v_1, v_i), h, d_{\min}(v_j, v_1)\}$, kde vrchol v_1 je depem. Necht' je $T^{X_R} = \{T_1^{X_R}, \dots, T_{m_{XR}}^{X_R}\}$

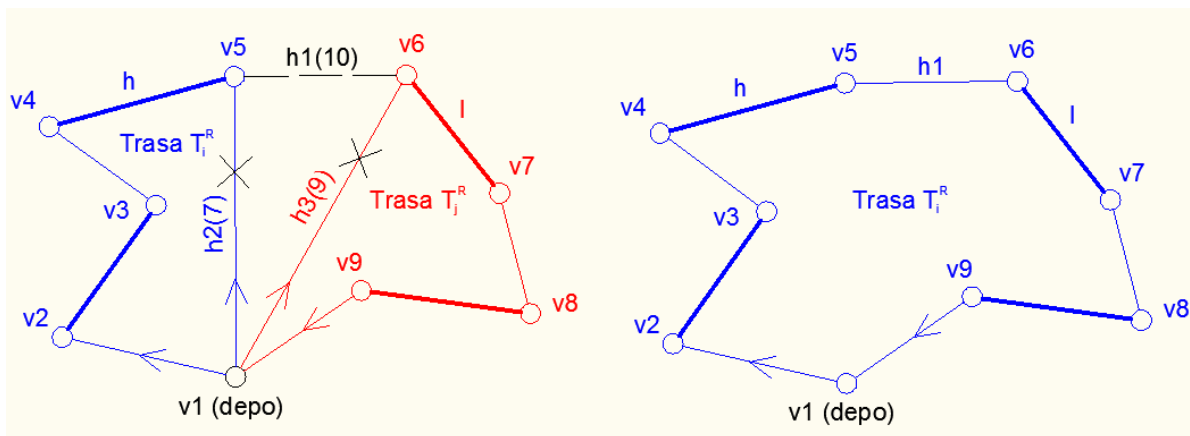
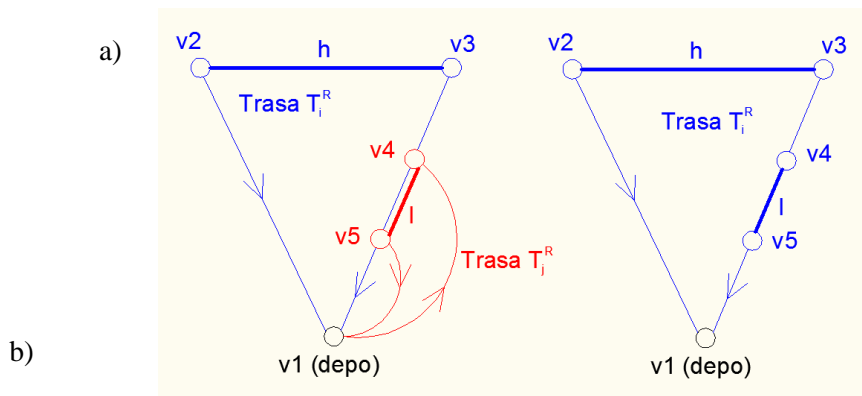
výsledná uspořádaná množina tras, které jsou seřazeny podle svého ohodnocení $c(T_1^{X_R}) \geq c(T_2^{X_R}) \geq \dots \geq c(T_{m_{X_R}}^{X_R})$.

V první fázi, zvané *augmentace*, je pro každou trasu $T_i^{X_R}$, $i = 1, \dots, m_{X_R} - 1$, ve které minimální cesty spojující hrany obsluhy obsahují hranu obsluhy nějaké další trasy z množiny T^{X_R} , prohledávána podmnožina tras $T_j^{X_R}$, $j = i + 1, i + 2, \dots, m_{X_R}$. Pokud existuje na trase $T_i^{X_R}$ v libovolné minimální cestě spojující hrany obsluhy hrana obsluhy h , která náleží také do trasy $T_j^{X_R}$, a platí $nak(T_i) + c^R(h) \leq Q$, je trasa $T_j^{X_R}$ vyloučena z množiny, viz obrázek 4.3a). Délka trasy $T_i^{X_R}$ se nezmění, protože v UCARP je ohodnocení hrany při obsluze rovno ohodnocení téže hrany při průchodu bez obsluhy $c^R(h) = c(h)$, ale celkové ohodnocení tras z množiny T^{X_R} je sníženo o $c(T_j^{X_R})$.

V druhé fázi, zvané *sloučení (merge)*, dochází ke slučování dvou tras $\{T_i^{X_R}, T_j^{X_R}\}$ tak, aby nebyla překročena kapacita vozidla Q . Trasy $T_i^{X_R}$ a $T_j^{X_R}$ se slučují tak, aby došlo k maximálním kladným úsporám v ohodnocení sloučené trasy, a to dokud není již další sloučení možné. Například sloučení tras $T_i^{X_R}$ a $T_j^{X_R}$ v tomto pořadí na obrázku 4.3b) přinese úsporu $7 + 9 - 10 = 6$. Existuje osm variant, jak sloučit trasy: $T_i^{X_R}$ a $T_j^{X_R}$ v tomto pořadí, $T_j^{X_R}$ a $T_i^{X_R}$, s možností změny směru každé trasy, tj. $T_i^{X_R}$ a $T_j^{X_R}$, $zpt(T_i^R)$ a $T_j^{X_R}$, $T_i^{X_R}$ a $zpt(T_j^R)$, $zpt(T_i^R)$ a $zpt(T_j^R)$, $T_j^{X_R}$ a $T_i^{X_R}$, $zpt(T_j^R)$ a $T_i^{X_R}$, $T_j^{X_R}$ a $zpt(T_i^R)$, $zpt(T_j^R)$ a $zpt(T_i^R)$, kde je obecně $zpt(T_k^{X_R})$ trasou $T_k^{X_R}$ procházenou v opačném směru. Ve skutečnosti však existují jen čtyři varianty, protože například sloučení $T_i^{X_R}$ a $T_j^{X_R}$ přinese stejné úspory jako sloučení $zpt(T_j^R)$ a $zpt(T_i^R)$.

Dále bude procedura AM popsána v souvislosti s řešením MCARP. V MCARP je hodnota řešení ovlivněna vícenásobným umístěním deponovacích míst, možným rozdílem mezi ohodnocením hran pokud je hrana obsluhována, či ne, a skutečností, že trasa v opačném směru nemusí mít stejné ohodnocení, pokud je distanční matice D asymetrická.

V MCARP nemusí být distanční matice symetrická, proto lze dobrého počátečního řešení ve tvaru „květu“ dosáhnout výběrem vhodné orientace hran obsluhy tak, aby trasa obsahující hranu obsluhy h měla minimální délku podle $\min\{d(s, h) + c^R(h) + kon(h), d(s, opc(h)) + c^R(opc(h)) + kon(opc(h))\}$.



Obrázek 4.3: AM heuristika, a) augmentace, b) slučování

Zdroj: Autor

V MCARP mohou být ohodnocení hran obsluhy při průchodu a obsluze různá $c(h) \neq c^R(h)$. V tomto případě jsou úspory získané *augmentací* tras T_i^R (trasa obsahuje hranu obsluhy h) a T_j^R (trasa obsahuje hranu obsluhy l), v tomto pořadí, rovny hodnotě $c(T_j^R) + kon(h) - d(h, l) - c^R(l) - kon(l)$. Tyto úspory mohou být vyjádřeny také jako $c(T_j^R) + c(l) - c^R(l)$. Hrana obsluhy l je od této chvíle koncovou součástí trasy T_i^R od hrany h do vrcholu s depem. Délka této koncové části je $kon(h) = d(h, l) + c(l) + kon(l)$. Nicméně, kromě kapacitní podmínky musí být splněna i podmínka dojezdová $c(T_i^R) + c^R(l) - c(l) \leq L$.

Při slučování se nabízí celkem osm variant, jak sloučit dvě trasy. Necht' $pos(T_i^R)$ je poslední hranou obsluhy v trase T_i^R a $zpt(T_i^R)$ je trasa T_i^R s opačným směrem. V trase $zpt(T_i^R)$ jsou všechny hrany obsluhy obráceny opačným směrem, tj. pokud $T_i^R = \{h_1, h_2\}$, tak $zpt(T_i^R) = \{opc(h_2), opc(h_1)\}$. Sloučení na obrázku 4.3b) přináší úsporu

$kon(h)+d(s,l)-d(h,l)$, zatímco sloučení T_i^R a $zpt(T_i^R)$ by přineslo úsporu $c(T_j^R)+kon(h)-c(zpt(T_j^R))+d(s,opc(pos(T_j^R)))-d(h,opc(pos(T_j^R)))$. Ve skutečnosti mohou být některá sloučení předem vyloučena, protože některé trasy obsahující or-hrany obsluhy nelze obrátit opačným směrem. Osm možností sloučení bylo popsáno výše. Algoritmus AM má časovou náročnost $O(m_R^2 \cdot \log m_R^2)$, tj. $O(n^2 \cdot \log n)$ pro reálnou síť městských komunikací s $m_R \leq m^* \approx 4n$.

Podle [22] jsou možná dvě vylepšení algoritmu AM. Pokud jsou všechny hrany hranami obsluhy, vše je v pořádku, protože hrany, které jsou zahrnuty do trasy, bývají hranami sousedními. Pokud tomu tak není, bezobslužné cesty vzniklé mezi hranami zahrnutými do trasy augmentací nemohou být během slučovací fáze obnoveny a tato skutečnost vede k horšímu řešení. Pro tento případ byla navržena verze AM bez augmentace nazvaná M.

Některé z dvojic tras mohou mít ve fázi slučování stejné úspory. V testech s verzí M došlo k vylepšení řešení, pokud byly sloučeny dvojice tras $\{T_i^R, T_j^R\}$ s největšími rozdíly celkových požadavků na obsluhu takových, že $|nak(T_i^R)-nak(T_j^R)|$ je maximální. Tato verze je nazývána IM (*improved M*). Po M a IM může následovat fáze augmentace, vzniknou tak další verze algoritmu M, AIM a IM.

4.2.5 Uluoyova heuristika (*Split procedura*)

Podle [27] spočívá princip původní UCARP dvoufázové heuristiky v nalezení úplné trasy $T_1^{X_R}$ pokrývající všechny hrany obsluhy (bez ohledu na kapacitu vozidla) a jejím rozdělení na kapacitně přípustné trasy za použití procedury nazývané *Split*, která bude popsána níže. Po rozdělení úplné trasy nemusí nutně vzniknout optimální kapacitně přípustné trasy. Stanovení úplné trasy, může být provedeno libovolnou heuristikou, například AM.

Na obrázku 4.5a) je ukázka procedury *Split*, která je aplikována na neorientovanou úplnou trasu $T_1^{X_R} = \{s, h_{40}, h_1, h_2, h_{14}, h_{19}, h_9, h_{13}, h_{39}, h_{32}, h_7, h_{27}, s\}$, kde číselné hodnoty v závorkách u jednotlivých hran představují požadavek na hraně obsluhy a ostatní číselné hodnoty ohodnocení hrany. Kapacita vozidla je stanovena na $Q = 5$. V proceduře *Split* je vytvořen pomocný acyklický graf H s $m_R + 1$ vrcholy indexovanými od 0 vzestupně, viz obrázek 4.5b). Každá přípustná trasa z podmnožiny $\{T_{1i}^{X_R}, \dots, T_{1j}^{X_R}\} \subseteq T_1^{X_R}$ úplné trasy je v grafu H reprezentována jedinou or-hranou $[i-1, j]$ s ohodnocením délky trasy. Trasa je

přípustná, pokud není překročena kapacita vozidla Q a dojezdová vzdálenost L . Množství nakládky na trase a její ohodnocení lze vypočítat podle (4.4) a (4.5). Minimální cesta z vrcholu s indexem 0 do vrcholu s indexem m_R v grafu H určuje nejlepší rozdělení úplné trasy. Na obrázku 4.5 je tato cesta vyznačena silnou čarou. Rozdělením vzniknou tři trasy s celkovým ohodnocením 288. Procedura *Split* v tomto případě našla tři optimální řešení UCARP s ohledem na posloupnost hran v původní trase, viz obrázek 4.5c).

Minimální cestu lze nalézt pomocí Bellman-Fordova algoritmu (R. Bellman, L. R. Ford, 1958) pro orientovaný acyklický graf, jehož časová náročnost je lineárně závislá na počtu or-hran grafu. Pro graf H je tento počet nejvýše $\frac{m_R \cdot (m_R - 1)}{2}$, časová náročnost je tedy $O(m_R^2)$. Časová náročnost pro reálnou síť městských komunikací s $m_R \leq m^* \approx 4n$ je $O(n^2)$.

Menší časové náročnosti lze dosáhnout, pokud je dostatečně velký požadavek w_{\min} , poté trasa zahrnuje nejvýše $\frac{Q}{w_{\min}}$ hran obsluhy a graf H obsahuje $\left(m_R \cdot \frac{Q}{w_{\min}}\right)$ or-hran. Tato verze má časovou náročnost $O(m_R)$.

Ulusoyova heuristika může být vylepšena rozdělením několika úplných tras a vybráním nejlepšího řešení MCARP, získaného tímto rozdělením. Při zanedbání kapacity vozidla Q a dojezdové vzdálenosti L lze získat úplné trasy PS algoritmem nebo jeho dvěma upravenými variantami popsanými v oddílu 4.2.3. PS vytvoří pět takovýchto tras (jednu pro každé kritérium), zatímco upravené varianty PS vytvoří více jak pět tras.

Posledního vylepšení lze dosáhnout záměnou pořadí hran v trase na grafu H . Na obrázku 4.5b) odpovídá or-hrana $[v_0, v_2]$ trase $\{h_{40}, h_1\}$, která má ohodnocení 90. Vzdálenost od hrany h_1 k hraně h_{40} je rovna 22. Pokud bude trasa začínat hranou h_1 a bude obsahovat cyklicky původní hrany ve stejném pořadí, je získána nová trasa $\{h_1, h_{40}\}$ s ohodnocením 82. Lepších výsledků lze tedy dosáhnout, pokud je každé or-hraně na grafu H přiřazeno ohodnocení vzniklé nejlepší cyklickou záměnou pořadí hran v trase.

V další části bude nejprve obecně popsán genetický algoritmus, a poté jeho varianta se zpracováním heuristik PS, AM a Ulusoyovy heuristiky, který je schopen velice efektivně řešit CARP. Tato varianta genetického algoritmu je známa jako *memetický algoritmus*.

4.2.6 *Genetické algoritmy a svozně-rozvozní úlohy s obsluhou hran*

Základní myšlenky těchto algoritmů vycházejí z Darwinovy teorie o vývoji druhů. Obecně se genetické algoritmy používají pro řešení optimalizačních problémů. Jsou založeny na principech genetiky a mechanismech přirozeného výběru. Na rozdíl od matematických optimalizačních metod jsou genetické algoritmy ve svém základu velmi jednoduché. Dále je v tomto oddílu citováno z [28].

Mezi základní pojmy, které popisují část genetických algoritmů, kde jsou neseny informace a hodnocena jejich kvalita, patří chromozóm, gen, populace a fitness hodnota.

Chromozóm je řetězec informací, který v sobě nese vlastnosti a chování každého jedince. Nejčastěji se jedná o řetězec nul a jedniček, kterým je zakódována pozice jedince v prostoru možných řešení. Nemusí se však jednat jen o binární řetězec, který představuje celé číslo. V případě řešení MCARP bude chromozóm obsahovat řetězec indexů hran obsluhy, které tvoří trasu vozidla.

Gen je nejmenší část chromozómu, která už je dále při aplikaci algoritmů nedělitelná. V MCARP se jedná o hranu obsluhy.

Populací je nazývána skupina jedinců popsaných svými chromozómy v rámci jedné generace.

Fitness hodnota je číselné vyjádření kvality každého jedince. Obvykle jde o reálné číslo v rozsahu od 0 do 1, ale může to být číslo z libovolného intervalu. V MCARP je to suma ohodnocení hran obsluhy jedné trasy. Pro každý problém je nutné sestavit tzv. **fitness funkci**, která bude jako svůj výsledek poskytovat požadovanou číselnou hodnotu.

Činnost genetických algoritmů je velmi jednoduchá. Používají tři základní operace: selekce (výběr), křížení a mutace. Tyto operace se vždy v dané generaci aplikují nad celou populací a výsledkem je nová generace. Tento proces je opakován do té doby, než se v nově vytvořené generaci vyskytne jeden nebo více jedinců s požadovanými vlastnostmi.

Selekcí nazveme proces, při kterém dochází k výběru jedinců z populace. Tito jedinci se mohou stát rodiči. Výběr lze uskutečnit několika způsoby. Hlavním ukazatelem výběru je kvalita jedince, tedy fitness hodnota jeho chromozómu.

Vážená ruleta je jedna z prvních používaných metod. Každý jedinec získá na pomyslné ruletě takový podíl, jaký odpovídá jeho fitness hodnotě. Tento podíl lze procentuálně vyjádřit jako (4.6). Nechť je p_i pravděpodobnost, s jakou bude i -tý jedinec vybrán, f_i označme fitness hodnotu i -tého jedince a n_c jako počet jedinců v populaci.

$$p_i = \frac{f_i}{\sum_{i=1}^{n_c} f_i} \quad (4.6)$$

Podle (4.6) je vytvořen kruh rulety rozdělený na n_c částí, které jsou úměrné hodnotám p_i . S takto vytvořenou ruletou je poté provedeno losování pro výběr dalšího rodiče.

Turnajová metoda náhodně vybírá skupinu jedinců z populace. Tato skupina musí obsahovat minimálně dva jedince. Vítězem turnaje ve skupině se stává jedinec s nejvyšší fitness hodnotou.

Při ořezávání, což je další metoda selekce, jsou všichni jedinci seřazeni vzestupně podle své fitness hodnoty. Tato řada je poté podle libovolně zvoleného parametru rozdělena na dvě části. Z první části řady s nízkými fitness hodnotami není rodič vybírán, z druhé části řady lze vybírat podle jakéhokoliv deterministického či stochastického pravidla.

Náhodný výběr je nejjednodušší metoda selekce, která žádným způsobem nezohledňuje kvality jedinců. Jako rodiče jsou náhodně vybíráni jedinci z celé populace.

Křížení je operace, která navazuje na selekci. Stejně jako v přírodě si rodiče vymění část svého genetického kódu, tedy část chromozómu. Nejjednodušší metodou je jednobodové křížení, kdy je náhodně zvolen bod v chromozómu rodičů. Tato hranice rozdělí chromozóm na dvě části a ty se mezi potomky vymění. Příklad jednoduchého jednobodového křížení dvou rodičů P1 a P2 s chromozómy:

P1: a b c d e f | g h i j

P2: c b a g h i | j f d e

Křížením vzniknou dva potomci C1 a C2 s chromozómy:

C1: a b c d e f | j f d e

C2: c b a g h i | g h i j

Existuje mnoho variant křížení, v následující metaheuristice řešící MCARP jsou použity varianty OX a LOX. Ukázky těchto křížení jsou zde:

Křížení OX (*Order Crossover*):

P1: a b c | d e f | g h i j

P2: c b a | g h i | j f d e

C1: e f j | g h i | a b c d

C2: h i j | d e f | c b a g

Křížení LOX (*Linear Order Crossover*):

P1: a b c | d e f | g h i j

C1: a b c | g h i | d e f j

P2: c b a | g h i | j f d e

C2: c b a | d e f | g h i j

Dále následuje volba, zda umožnit oběma novým jedincům postoupit do další generace, nebo náhodně vybrat pouze jednoho z nich. Může být také zvoleno i vícebodové křížení a kód potomka může vznikat různými kombinacemi z více jak jednoho rodiče. Je možné, aby do další generace postoupili i někteří rodiče, a to dokonce aniž by se zúčastnili křížení. Tato možnost se ovšem používá jen s velmi malou pravděpodobností.

Mutace je poslední operace genetického algoritmu. U každého jedince z nové generace je procházen celý chromozóm a s velmi malou pravděpodobností jsou měněny hodnoty některých genů. Význam mutace spočívá v možném objevení vlastnosti, kterou doposud žádný jedinec neměl a nemohl ji tedy předat potomkům.

Činnost genetického algoritmu lze stručně popsat následujícími kroky:

Krok 1: (Inicializace) Stanovení velikosti populace a náhodné vygenerování chromozómů všech jedinců. Tím je vytvořena 1. generace, ve které je stanovena fitness hodnota jedinců.

Krok 2: Selektce.

Krok 3: Křížení.

Krok 4: Mutace.

Krok 5: Vyhodnocení nově vzniklé generace. Pokud vznikl jedinec splňující požadované vlastnosti, algoritmus končí.

Krok 6: Náhrada stávající generace nově vygenerovanou.

Krok 7: Pokračování krokem 2.

4.2.7 *Memetický algoritmus*

Memetický algoritmus (MA) je genetický algoritmus s iterační heuristikou lokálního prohledávání. Lacomme v [29] navrhl MA pro řešení CARP se zákazem otáčení ve vrcholech. Algoritmus byl rozšířen o penalizaci při otáčení a odbočení vozidla, multihrany a omezení dojezdové vzdálenosti. Průměrná odchylka od optimálního řešení, nalezeného určením dolní hranice řešení, je vylepšena krátkými restarty algoritmu, založenými na částečném nahrazení populace. Oba algoritmy byly aplikovány pouze na klasické úlohy UCARP.

Dále bude podle [29] popsána poslední verze, která je jednou z nejefektivnějších metod řešení CARP. Chromozóm je tvořen posloupností genů, které reprezentují jednotlivé hrany obsluhy úplné trasy. Chromozóm tedy představuje relaxované řešení UCARP nebo MCARP pro vozidlo s neomezenou kapacitou a dojezdem. Každý gen je označen indexem hrany obsluhy, který podle jednoduchého pravidla rozlišuje mezi oběma směry hrany v trase. Mezi hranami obsluhy jsou implicitně uvažovány minimální cesty, které jsou reprezentovány jedinou hranou. Fitness funkce je získána procedurou *Split*, která optimálně rozdělí úplnou trasu na více přípustných tras, ve kterých je respektována kapacita vozidla a jeho dojezdová vzdálenost. Hodnota Fitness funkce je tedy ohodnocením přípustného řešení CARP. Počáteční populace (n_c chromozómů) zahrnuje tři počáteční řešení získaná základními verzemi heuristik PS, AM a Ulusoyovy heuristiky a dokončená náhodnými permutacemi. Klony (totožné chromozómy) jsou zakázané. V každé iteraci jsou v turnajové metodě vybráni dva nejlepší rodiče, kteří jsou kříženi upravenou verzí klasického křížení pořadí genů Order Crossover (OX) nebo Linear Order Crossover (LOX). Jeden potomek je náhodně vybrán a druhý je vyřazen.

Chromozómy tras bez kapacitního a dojezdového omezení mohou být klasicky kříženy za vzniku jejich dalších permutací - potomků. Výslední potomci jsou po křížení zpracovány procedurou *Split*. LOX je navrženo pro lineární chromozómy (v chromozómech jsou zakódovány trasy, které začínají a končí v odlišných vrcholech, jako otevřená cesta), zatímco OX se týká spíše permutací cyklických chromozómů (počáteční a koncový vrchol trasy je totožný, jako v TSP). V tomto případě je lepší volbou OX, protože chromozómy mohou být před rozdělením procedurou *Split* chápány jako cyklický objekt (úplná trasa, obsahující všechny hrany obsluhy a začínající a končící v depu).

Tabulka 4.1: Ukázka křížení chromozómů v memetickém algoritmu

Index genu chromozómu	1	2	3	4	5	6	7	8	9	10	11	Fitness fce	
Dělicí geny chrom. r1 a r2						r1=6		r2=8				(Split)	
Rodič P1 (chromozóm č.1)	40	1	2	14	19		9	13	39		32	7	27
Rodič P2 (chromozóm č.2)	<u>17</u>	18	<u>31</u>	2	29	27	23	<u>35</u>	19	36	32		
Potomek C1 (chrom. č.3)	18	2	29	27	23		9	13	39		19	36	32
Aplikace Split pro P1	{40,	1	2	14	19}	{9}	{13,	39	32	7	27}	f(P1)=288	
Aplikace Split pro P2	{17,	18	31}	{2,	29	27}	{23,	35	19	36	32}	f(P2)=286	
Aplikace Split pro C1	{18,	2	29	27}	{23,	9}	{13,	39	19	36	32}	f(C1)=276	

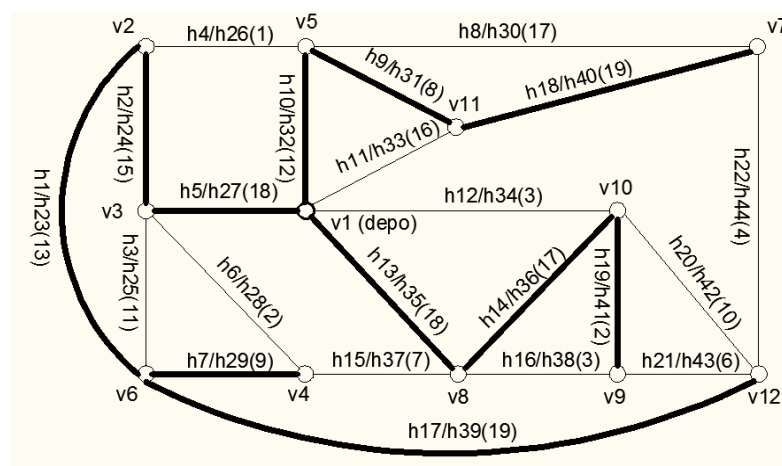
Zdroj: Autor

Jsou dáni dva rodiče P1 a P2, každý reprezentován chromozómem s m_R geny. V obou případech křížení jsou v chromozómu určeny dva dělicí body r_1 a r_2 , platí $1 \leq r_1 \leq r_2 \leq m_R$. Aby byl získán potomek C1, LOX zkopíruje sekvenci genů $P1(r_1), \dots, P1(r_2)$ do $C1(r_1), \dots, C1(r_2)$. Chromozóm rodiče P2 je poté prohledáván zleva doprava a geny, které nejsou obsaženy v C1, jsou kopírovány, také zleva doprava, na prázdné pozice $C1(1), \dots, C1(r_1 - 1)$ a poté na $C1(r_2 + 1), \dots, C1(m_R)$. V chromozómu se nemohou vyskytovat dva stejné geny. Geny v chromozómu představují hrany obsluhy a jejich směry jsou odlišeny indexem (kurzívou psané hodnoty v tabulce 4.1, řádek P2, sloupec s indexem genu 3: or-hrana s indexem 31 odpovídá opačně orientované hraně $31-22=9$ atd.). Pokud má být do C1 kopírován gen reprezentující or-hranu obsluhy a v C1 již tato or-hrana opačného směru existuje, gen do C1 zkopírován není.

V klasickém OX je získán potomek C1 zkopírováním sekvence genů $P1(r_1), \dots, P1(r_2)$ do $C1(r_1), \dots, C1(r_2)$ následované sekvencí $P2(1), \dots, P2(r_1 - 1)$ a $P2(r_2 + 1), \dots, P2(m_R)$ do $C1(r_2 + 1), \dots, C1(m_R)$ a poté do $C1(1), \dots, C1(r_1 - 1)$ s podmínkou, že geny z chromozómu rodiče P2 jsou zkopírováni jen v případě, pokud nejsou obsaženy v chromozómu C1. Potomek C1 je reprezentován cyklickým chromozómem.

V testech s křížením OX autor provedl úpravu, která poskytuje lepší hodnoty fitness funkce, viz příloha 5. Úprava spočívá v záměně pořadí následovaných sekvencí v tomto pořadí: $P2(r_2 + 1), \dots, P2(m_R)$ a $P2(1), \dots, P2(r_1 - 1)$ do $C1(r_1), \dots, C1(r_2)$.

V obou případech křížení LOX a OX lze získat také druhého potomka C2, jednoduchou záměnou rodičů P1 a P2.



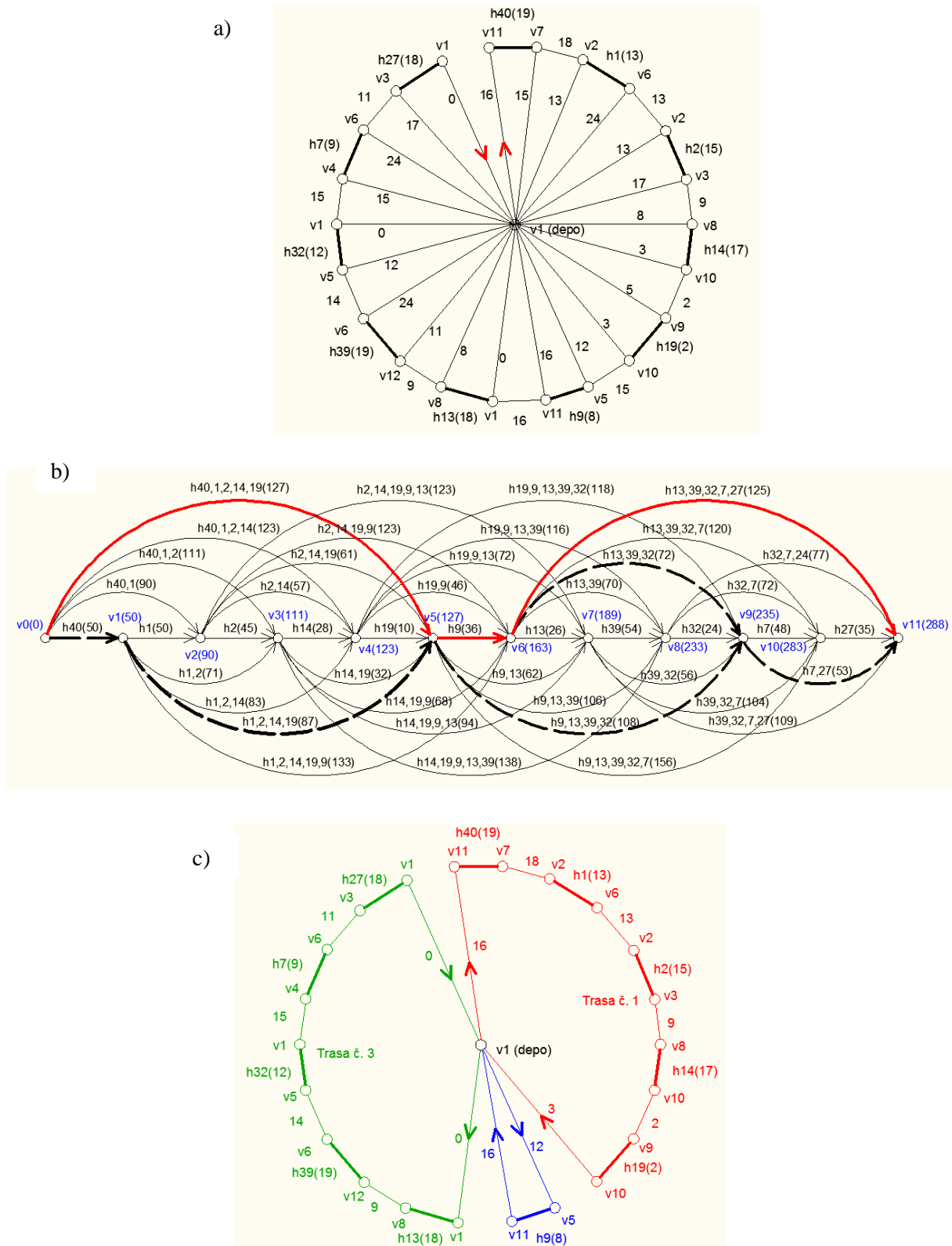
Obrázek 4.4: Neorientovaný graf, na kterém je ukázán princip memetického algoritmu

Zdroj: Autor

V tabulce 4.1 je ukázka křížení LOX. Je vycházeno z neorientovaného grafu na obrázku 4.4. Jedná se o klasickou úlohu CARP s depem ve vrcholu v_1 , kapacitou vozidla $Q = 5$, počtem hran grafu $m = 22$ a počtem hran obsluhy $m_R = 11$, které mají jednotkový požadavek obsluhy. Každá hrana má index u , číslo v závorce představuje ohodnocení hrany. Hrany obsluhy jsou zvýrazněny silnou čarou, ostatní hrany reprezentují minimální cestu mezi těmito hranami. Upravený orientovaný graf G s $m = 44$, kde je každá neorientovaná hrana nahrazena dvojicí orientovaných hran opačného směru, není z obrázku patrný, ale každá or-hrana je dána uspořádanou dvojicí $[i, j]$, v opačném směru $[j, i]$, pro $i < j$. Například or-hrana $[v_5, v_7]$ má index $idx = 9$ a or-hrana $[v_7, v_5]$ má index $m + idx = 22 + 9 = 31$. Počáteční řešení (dvojice úplných tras) je dáno dvěma chromozómy rodičů P1 a P2 a je získáno pomocí heuristik AM nebo PS, viz obrázek 4.5a). LOX křížením rodičů P1 a P2 vznikne potomek C1. Dále je použita procedura *Split*, která rozdělí relaxované úplné trasy na podmnožinu přípustných tras tak, aby byla splněna dojezdová a kapacitní podmínka. V tomto případě je každá úplná trasa rozdělena na trojici dílčích tras, viz obrázek 4.5c). Při rozdělování na dílčí trasy je brán zřetel na orientaci hran obsluhy, tj. hrana (v_7, v_{11}) je v P2 uvažována jako or-hrana $[v_7, v_{11}]$ s indexem 18 a v P1 jako or-hrana $[v_{11}, v_7]$ s indexem 40. Fitness funkce, jejichž hodnoty jsou rovny součtu ohodnocení dílčích tras získaných procedurou *Split* v jednotlivých chromozómech, jsou v tabulce 4.1 označeny jako $f(P1)$, $f(P2)$ a $f(C1)$.

Mutace je v genetickém algoritmu nahrazena iterační heuristikou lokálního prohledávání volanou s pravděpodobností p_m , která nepracuje na úplné trase, ale na individuálních trasách získaných *Split* procedurou. Procedury lokálního prohledávání spočívají ve vyjmutí jedné nebo dvou po sobě následujících hran obsluhy z trasy a jejich opětovném umístění na jinou pozici v trase, dále v záměně dvou hran obsluhy trasy a v *2-opt* záměnách.

Tyto záměny mohou probíhat na jedné nebo dvou trasách a při opětovném umístění jsou testovány oba směry hrany obsluhy. Každá iterace prochází v čase $O(m_R^2)$ všechny varianty záměn, dokud není nalezena záměna vylepšující řešení. Lokální prohledávání končí, pokud již není nalezeno žádné vylepšení. Trasy jsou nyní uspořádány do chromozómů (úplné trasy), které jsou nově ohodnoceny procedurou *Split*.



Obrázek 4.5: Procedura Split. a) $T_1^{X_R} = \{S, h_{40}, h_1, h_2, h_{14}, h_{19}, h_9, h_{13}, h_{39}, h_{32}, h_7, h_{27}, S\}$ počáteční trasa, b) pomocný graf H s minimálními cestami (vrcholy s indexem mají číselné ohodnocení délky trasy), c) výsledné trasy

Zdroj: Autor

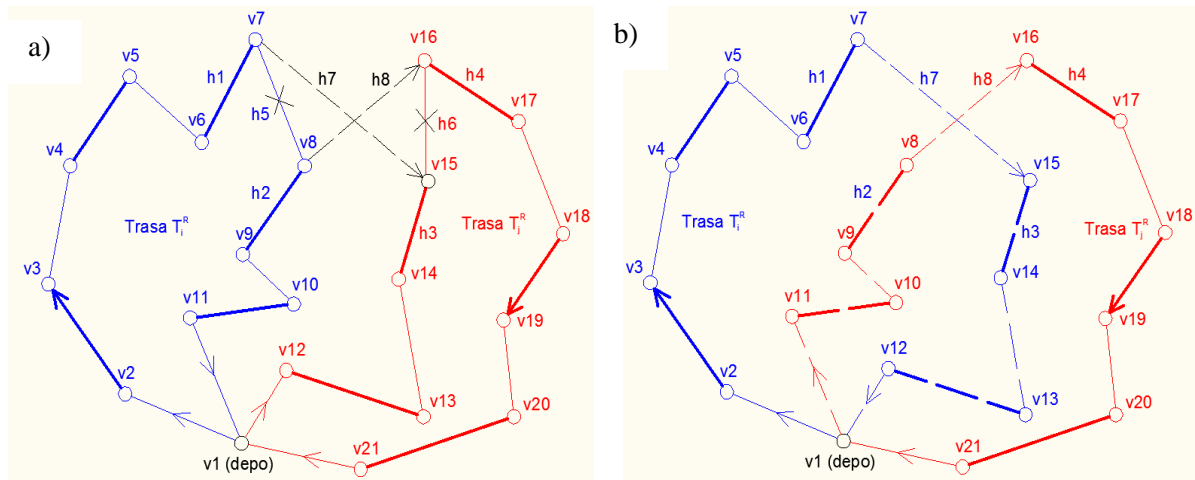
Výsledný potomek nahradí jeden z $\frac{n_c}{2}$ nejhorších chromozómů, který je náhodně vybrán z populace tak, aby nahrazením nevznikl žádný klon rodiče. MA končí po daném počtu iterací n_i , nebo po předem určené stabilizované době dané n_s iteracemi, pokud nedojde ke zlepšení prozatím nejlepšího řešení, nebo dosažením dolní hranice intervalu řešení. Pokud není dosaženo dolní hranice řešení, je vykonáno n_r krátkých restartů s intenzivnějším lokálním prohledáváním (s pravděpodobností volání $p_{mr} > p_m$). Každý restart začíná procedurou, která nahrazuje n_{cr} chromozómů jedním novým ($n_{cr} < n_c$) při zachování nejlepšího řešení. MA končí, pokud je dosaženo dolní hranice řešení, nebo po daném počtu iterací n_{ir} ($n_{ir} < n_i$).

4.2.8 Upravený MA pro MCARP rozšířený o další omezení

Standardní verze PS, AM a Ulusoyovy heuristiky, které do počáteční populace poskytnou tři uspokojivá řešení, jsou nahrazeny lepšími variantami popsány v oddílech 4.2.3, 4.2.4 a 4.2.5. Jsou přidána vícenásobná umístění deponovacích míst a nesouměrné hrany, které ovlivní výpočet pouze v proceduře *Split* a v proceduře lokálního prohledávání. Aby byla zachována nízká časová náročnost $O(m_R^2)$, původní verze MA testovala možné *2-opt* záměny pouze na trasách, které měly v opačném směru stejné ohodnocení, tj. neobsahovaly or-hrany obsluhy. Hlavní úprava v novém MA spočívá v prohledání *2-opt* záměn na všech trasách při dodržení časové náročnosti $O(m_R^2)$.

Na obrázku 4.6 je příklad procedury *2-opt* pro dvě trasy T_i^R a T_j^R . Hrany obsluhy jsou zvýrazněny silně. Silné hrany s šipkami jsou or-hranami obsluhy. Minimální cesty jsou reprezentovány tenkou čarou. Cesta mezi hranami obsluhy h a l je označena jako $h \rightarrow l$. Na obrázku 4.6a) je použita procedura *2-opt* pro záměnu dvou bezobslužných cest h_5 a h_6 za h_7 a h_8 : $h_1 \rightarrow h_5 \rightarrow h_2$ a $h_3 \rightarrow h_6 \rightarrow h_4$ za $h_1 \rightarrow h_7 \rightarrow opc(h_3)$ a $opc(h_2) \rightarrow h_8 \rightarrow h_4$. Dvě části tras jsou procházeny opačným směrem: část konce trasy T_i^R začínající hranou obsluhy h_2 a část začátku trasy T_j^R po hranu obsluhy h_3 včetně. Na obrázku 4.6b) jsou tyto hrany přerušované. Tato záměna je přípustná, protože obě dvě části trasy s opačným směrem neobsahují žádné or-hrany obsluhy, které by byly procházeny v protisměru. Protože jsou hrany obsluhy uvažovány jako nesouměrné a je uvažováno vícenásobné umístění

deponovacích míst, mají tyto dvě části trasy po záměně odlišné ohodnocení. Po aplikaci procedury *2-opt* vzniknou dvě nové trasy, které jsou na obrázku 4.6b).



Obrázek 4.6: Ukázka *2-opt* záměny na dvojici tras T_i^R a T_j^R

Zdroj: Autor

Cesta mezi hranami obsluhy h a l je označena jako $h \rightarrow l$. Na obrázku 4.6a) je použita procedura *2-opt* pro záměnu dvou bezobslužných cest h_5 a h_6 za h_7 a h_8 : $h_1 \rightarrow h_5 \rightarrow h_2$ a $h_3 \rightarrow h_6 \rightarrow h_4$ za $h_1 \rightarrow h_7 \rightarrow \text{opc}(h_3)$ a $\text{opc}(h_2) \rightarrow h_8 \rightarrow h_4$. Dvě části tras jsou procházeny opačným směrem: část konce trasy T_i^R začínající hranou obsluhy h_2 a část začátku trasy T_j^R po hranu obsluhy h_3 včetně. Na obrázku 4.6b) jsou tyto hrany přerušované. Tato záměna je přípustná, protože obě dvě části trasy s opačným směrem neobsahují žádné or-hrany obsluhy, které by byly procházeny v protisměru. Protože jsou hrany obsluhy uvažovány jako nesouměrné a je uvažováno vícenásobné umístění deponovacích míst, mají tyto dvě části trasy po záměně odlišné ohodnocení. Po aplikaci procedury *2-opt* vzniknou dvě nové trasy, které jsou na obrázku 4.6b).

Nechť je $zpt(T_i^R)$ trasa T_i^R s opačným směrem, $podco(T_i^R, h)$ počet or-hran obsluhy od začátku trasy k hraně obsluhy h včetně a $podod(T_i^R, h)$ počet or-hran obsluhy od hrany obsluhy h , včetně, do konce trasy. Obdobně, necht' jsou $nakdo(T_i^R, h)$ celkové požadavky k hraně obsluhy h , $nakod(T_i^R, h)$ celkové požadavky od hrany h do konce trasy, $cdo(T_i^R, h)$ celkové ohodnocení k hraně obsluhy h a $cod(T_i^R, h)$ celkové ohodnocení od hrany h do konce trasy.

Záměna hran na obrázku 4.6a) je možná, pokud $pocod(T_i^R, h_2) = pocdo(T_j^R, h_3) = 0$ (v částech trasy s opačným směrem nejsou obslužné or-hrany) a pokud $nakdo(T_i^R, h_1) + nakdo(T_j^R, h_3) \leq Q$ a $nakod(T_i^R, h_2) + nakod(T_j^R, h_4) \leq Q$ (v nově vzniklých trasách je respektována kapacita vozidla). Poté je ohodnocení záměny dáno $o(h_7) + o(h_8)$, kde $o(h_7) = d(h_1, opc(h_3)) - d(h_1, h_2) + cdo(zpt(T_i^R), opc(h_2)) - cod(T_j^R, h_2)$ a $o(h_8) = d(opc(h_2), h_4) - d(h_3, h_4) + cod(zpt(T_j^R), opc(h_3)) - cdo(T_j^R, h_3)$. Výsledné trasy jsou na obrázku 4.6b).

V porovnání s UCARP jsou tyto výpočty více komplikované. Nicméně pro stávající řešení lze nalézt vylepšení heuristikou *2-opt* také v čase $O(m_R^2)$. Tato implementace prochází všechny páry tras $\{T_i^R, T_j^R\}$, každou hranu obsluhy h_1 v T_i^R , a každou hranu obsluhy h_3 v T_j^R . Na začátku každého prohledávání jsou pro každou trasu inicializovány hodnoty *pocod*, *poddo*, *nakod*, *nakdo*, *cod* a *cdo*. Časová složitost tohoto předběžného zpracování je $O(m_R)$. Pokud dojde během prohledávání k záměně hran, je tato změna provedena v čase $O(1)$.

4.3 Řešení svozně-rozvozního problému s využitím GIS

V této podkapitole jsou popsány obecné principy začlenění metod svozně-rozvozních úloh do prostředí GIS. Výpočetní složitost svozně-rozvozních úloh s obsluhou hran strmě roste s množstvím vstupních dat. V tomto případě s množstvím obsluhovaných komunikací dopravní sítě a při obousměrné obsluze komunikací.

Je proto nutné nalézt kompromis mezi přesností výsledného řešení a časem, který je třeba k výpočtu. Aby výsledná řešení byla přesná, nestačí pouze vhodně zvolit výpočtový algoritmus, ale také naplnit daný model přesnými daty. To mimo jiné znamená vytvoření grafu sítě co nejvíce odpovídajícího skutečnosti. V tomto případě jsou nejvhodnější volbou data GIS. Výhoda GIS nespočívá pouze v přesnosti a velké datové základně, ale také v přiřazených attributech jednotlivým entitám mapových hladin a možnosti zpětné vizualizace výsledků na jakémkoliv běžném GIS software. K datům lze také přistupovat online přes mapové servery (dále WMS – *Web Map Service*).

Výsledky v podobě jednotlivých tras obslužných vozidel lze také využít přímo v terénu řidiči vozidel, kteří mohou být vybaveni běžně dostupnou GPS navigací, s uloženou naplánovanou trasou, a v průběhu směny jsou přesně navigováni, viz příloha 6. Dále může být v průběhu směny přesně monitorována poloha obslužného vozidla a informace o poloze

přenášeny pomocí bezdrátové sítě na řídicí dispečink. Tato skutečnost umožňuje operativní řízení obslužných vozidel přímo v reálném čase. Vstupy pro výpočty tvoří GIS data a výstupy v podobě obslužných tras lze vizualizovat na běžném komerčním GIS software nebo na WMS. Po podrobné analýze současných trendů a postupů řešení ARP byla zvolena jako hlavní výpočetní metoda metaheuristika memetický algoritmus.

4.3.1 GIS software

GIS software je systém pro sběr, ukládání, správu, dotazování, analyzování a prezentaci geografických informací. Jde o systém programových prostředků, který slouží k naplnění funkcí GIS.

V současné době existuje na trhu s GIS software přibližně stovka produktů. Mezi největší světové distributory GIS software patří společnosti ESRI, Autodesk, Intergraph, MapInfo, GE Smallworld a další. Z českých GIS společností to jsou Berit, DIGIS, Foresta SG, GEPRO, T-Mapy, Xanadu. Na základě funkcionality lze rozlišit následující skupiny GIS software [30]:

- **profesionální GIS** – plně funkční systém sloužící pro pořizování dat, jejich editaci a administraci databází. Je rozšířen o nástroje prostorových analýz a další speciální nástroje. Jako příklad lze uvést ESRI ArcInfo nebo Smallworld GIS,
- **desktop GIS** – nejvíce se rozšiřující GIS v posledních letech, jsou též nazývány desktop mapovací systémy. Pokrývají většinový podíl uživatelů GIS. Zaměření jejich funkcí je zejména na práci s daty než na jejich pořizování. Obsahují výtečné nástroje pro tvorbu map, grafů a dalších výstupů. Příklady desktop GIS jsou Autodesk World, Autodesk map 3D, ESRI ArcView, Intergraph GeoMedia a další,
- **příruční GIS** – software uzpůsobený pro miniaturizovaný hardware s mobilním využitím v terénu (GIS pro ruční počítače PDA, GSM zařízení). Příkladem je ArcPad (ESRI),
- **GIS prohlížeče** – software zpravidla obsahující pouze funkce umožňující prohlížení dat a dotazování. Příkladem je ArcExplorer (ESRI), GeoMedia Viewer (Intergraph), ProViewer (MapInfo),
- **Internet GIS** – GIS produkty s potenciálně nejvyšším počtem uživatelů na základě nízké pořizovací ceny. Stoupající využívání těchto aplikací je stimulováno širokým rozšířením Internetu a poptávkou po geografických informacích. Znakem Internet GIS je jednoduché zobrazování dat a zjednodušené pokládání dotazů. Příkladem software

je ArcIMS (ESRI) nebo GeoMedia Web Map (Intergraph). OpenSource a freeware příkladem produktu je MapServer vyvíjený na Minnesotské univerzitě.

4.4 GIS data jako vstup pro výpočet metodami operačního výzkumu

Zcela klíčovou částí GIS jsou data. Jejich vysoká hodnota je podložena časovou náročností procesu jejich tvorby i nároky na jejich aktuálnost a přesnost. Data se podle typu rozdělují v základě na analogová a digitální. Analogová data jsou papírové mapy, náčrty, fotografie. Pro jejich využití v GIS je nezbytný jejich převod do digitální podoby. Tento proces se nazývá digitalizace.

Výhoda digitálních dat spočívá v možnosti jejich snadného zobrazení, úsporného uložení a analyzování. Digitální data využívaná v GIS lze rozdělit na rastrová a vektorová. Proces digitalizace za vzniku vektorových dat bývá označován slovem vektorizace. K vektorizaci je využíván digitizér. Naopak skenováním papírové mapy do digitální podoby jsou získána data rastrová. Tento proces se nazývá rasterizace.

4.4.1 Zdroje vektorových dat

Získání digitálních geografických dat může být významným problémem, a to jak z důvodu finančního, tak i z důvodu jejich vhodnosti jako jednoho ze vstupů pro výpočty metodami operačního výzkumu. Data by měla být:

- cenově dostupná,
- v dostatečné podrobnosti (měřítku),
- k jednotlivým entitám by mělo existovat dostatečné množství přiřazených atributů,
- úplná a korektní.

4.4.2 Rastrová data

Data v rastrovém formátu jsou charakterizována souborem pixelů o přesně daném počtu. Bod je dán jediným pixelem o celočíselných souřadnicích x a y s počátkem v $[0,0]$, které nepředstavují geografickou polohu. Rastrová data jsou vhodná pro prostorové analýzy a pro tvorbu 3-D modelů území. Rastrová data mohou také sloužit jako podklad k vektorizaci a získání vektorových dat. Vznikají skenováním povrchu Země senzory umístěnými na družicích (satelitní snímky), v letadlech (letecké snímky) metodami dálkového průzkumu

Země nebo skenování papírových podkladů. Příkladem rastrového formátu dat může být TIFF, ale existuje celá řada dalších formátů.

4.4.3 Vektorová data jako vstup pro výpočet

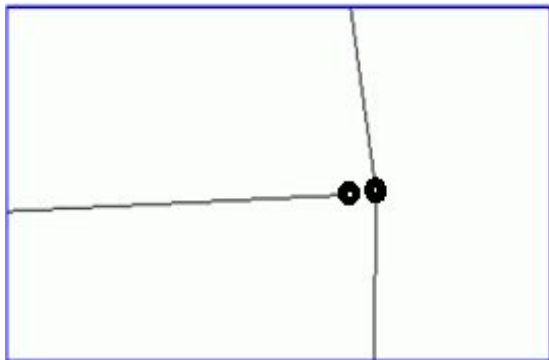
Bod je v případě vektorových dat dán plošnými souřadnicemi x a y nebo eliptickými souřadnicemi zeměpisné šířky a zeměpisné délky ve stupních, minutách a sekundách. Linie lze definovat jako soubor bodů mezi těmito souřadnicemi a označujeme je jako hrany. Tyto body označujeme jako vrcholy, které reprezentují počátek a konec hrany, popřípadě mezilehlé vrcholy. Prostor mezi vrcholy je dopočítáván předem definovaným způsobem jako nejkratší spojnice dvou vrcholů nebo jako křivka s danými parametry.

Primárním zdrojem vektorových dat (tedy dat získaných přímo měřením v terénu) mohou být údaje poskytnuté zařízením GPS. Druhotně lze získat vektorová data digitalizací (vektorizací) topografických a tematických map, leteckých a družicových snímků.

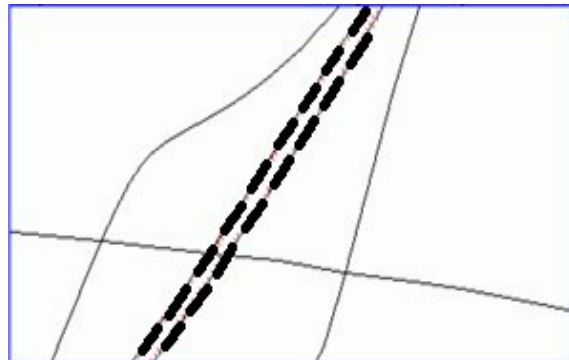
GIS data jsou převážně uchovávána na WMS, kde k nim přistupují jednotliví uživatelé. V malém měřítku jsou umístěny přímo na lokálních PC, kde se s nimi pracuje offline.

GIS pracuje s celou řadou formátů vektorových i rastrových dat. Snahou distributorů GIS software je vytvářet prostředky pro převod mezi jednotlivými formáty dat. Ve světě a v České republice je obzvláště rozšířeným formátem vektorových dat takzvaný shapefile, nativní formát software ESRI. Tyto soubory mají příponu SHP. Tato vektorová data GIS se skládají ze dvou základních částí. První částí jsou vektorová data rozdělená do takzvaných entit – body, úsečky, polygony, křivky reprezentující v tomto případě komunikace, křižovatky, budovy a tak dále, kde každá entita má v souboru SHP vlastní hladinu. Druhou část tvoří databáze ve formátu dBASE IV popisující část první, to znamená, každé entitě vektorových dat jsou přiřazeny určité atributy (název, typ, souřadnice a jiné). Aby bylo možné data použít jako vstup pro výpočet ARP, je nutná jejich transformace do korektního grafu dopravní sítě. Zde může nastat několik problémů s nekorektností geografických vektorových dat. Jedná se o duplicitu, nespojitost, chybějící a přebývající vrcholy. V případě duplicity se jedná o zdvojení hrany či vrcholu vektorových dat. Hrany na sebe nemusejí navazovat, i když při vizualizaci mapy není tato chyba patrná. Častým jevem je nepřítomnost vrcholu v místě dotyku hran a naopak přítomnost vrcholu v místě, kde není vyžadován (například v mimoúrovňovém křížení komunikací). Tyto nekorektnosti je třeba při transformaci na graf sítě odstranit. Ukázky nekorektnosti vektorových dat jsou na obrázku 4.7.

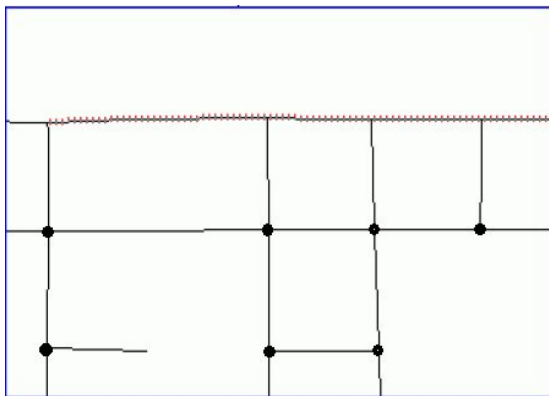
Problém nespojitosti GIS dat



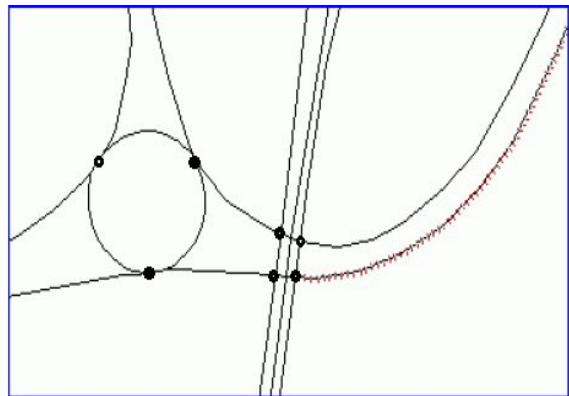
Problém duplicity dat (duplicita hrany)



Problém s odbočením (nepřítomnost vrcholu v oblasti dotyku hran v oblasti dotyku hran)



Problém v mimoúrovňovém křížení hran (přítomnost vrcholu v oblasti dotyku hran)

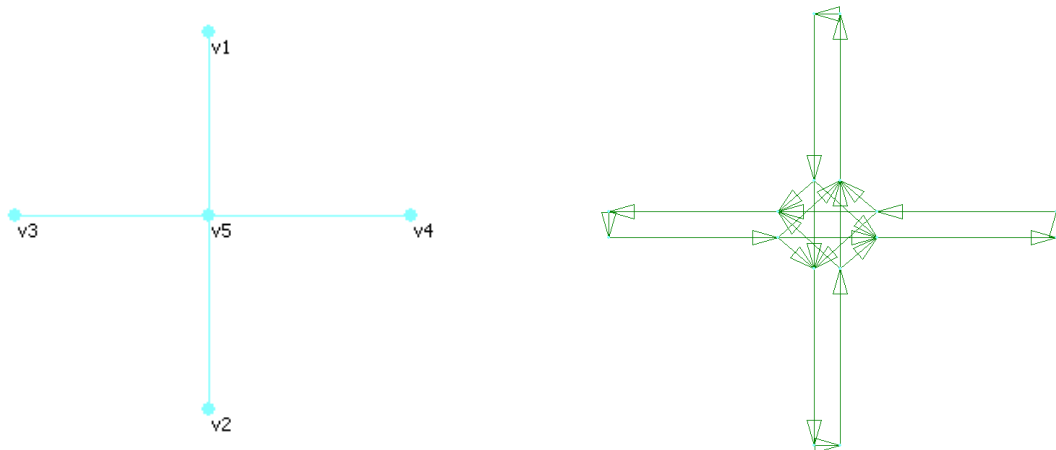


Obrázek 4.7: Problém nekorektnosti GIS dat

Zdroj: Autor

Aby bylo možné provést vlastní výpočet optimální trasy ARP, je nutné v grafu dopravní sítě zohlednit určitá pravidla chování dopravního prostředku na reálné dopravní infrastruktuře. Pokud bude místo obsluhy určeno vrcholem, který leží na příslušné hraně, a jedná se o křižovatku, je nutné při výpočtu optimální trasy zabránit otočení se vozidla v tomto vrcholu a zpětnému průchodu touto hranou. Problém při určování optimální cesty může vzniknout i s přednostmi v jízdě (kruhové objezdy) a takzvanými U-turns (vozidlo se nesmí otočit v křižovatce). To znamená, že reprezentace křižovatky jako jednoho vrcholu je při výpočtu vyloučena. V tomto případě je nutné graf sítě přizpůsobit městskému provozu přechodem na orientovaný graf, respektive smíšený graf sítě, s větším počtem vrcholů a hran. Při řešení rozsáhlejšího ARP mimo město je tento problém zanedbatelný a ARP lze řešit na neorientovaném grafu.

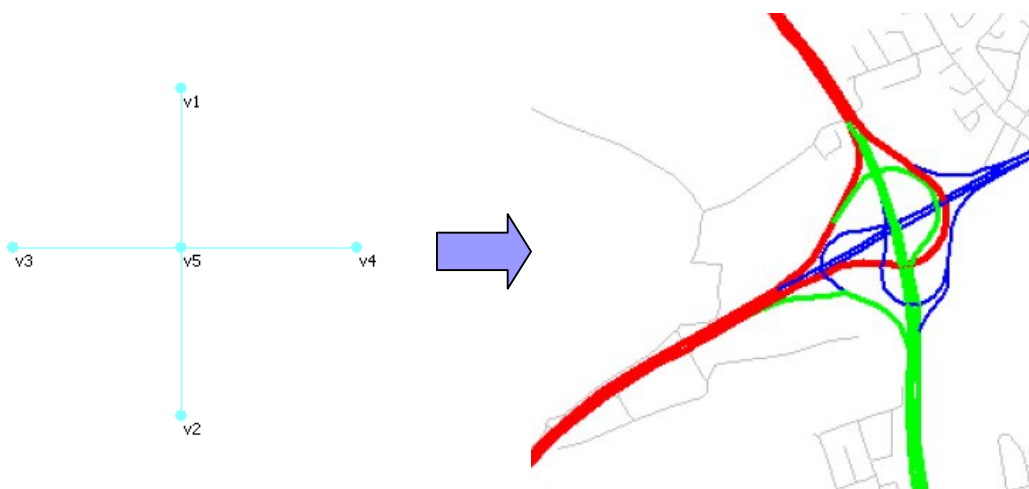
Po převodu grafu na graf sítě městských komunikací, kde jsou zohledněna pravidla silničního provozu, a definici množiny hran obsluhy je možné aplikovat algoritmus řešící MCARP, tedy memetický algoritmus. Ukázka převodu do městského režimu je na obrázku 4.8.



Obrázek 4.8: Ukázka převodu síťového grafu pomocí software NetOpt do městského režimu (pravidla přednosti v jízdě, kruhové objezdy, zákaz otáčení v křižovatce a tak dále)

Zdroj: Autor

Vlastní výpočet vygeneruje vektor (chromozóm), obsahující hrany obsluhy v pořadí, ve kterém mají být projížďeny obslužným vozidlem. Jde tedy o výslednou trasu vozidla. Atributy GIS dat v sobě nesou mimo jiné informaci o názvech a typech komunikací. Výsledná trasa je tedy popsána posloupností obslužných i bezobslužných hran, resp. jejich názvy. Délka trasy a jednotlivé polohy vozidla jsou určeny s přesností řádově na metry a celkový čas potřebný k projetí trasy i s obsluhou komunikací lze přesně určit pomocí zvolených parametrů – rychlosti vozidla na jednotlivých typech komunikací a penalizacemi při odbočování a stání v křižovatce. Trasu lze uložit jako jmenný seznam komunikací a křižovatek s údaji o času průjezdu a kilometrovníku, který je poté k dispozici řidiči obslužného vozidla. Dále lze trasu zpětně uložit jako další hladinu do vektorového mapového podkladu GIS pro pozdější použití, nebo jako trasu pro zařízení GSP. Řidič obslužného vozidla vybaveného zařízením GPS může tedy v reálném čase sledovat průjezd naplánované trasy. Ukázka vizualizace trasy v prostředí Google maps je v příloze 7. Zpětným převodem výsledných dat do formátu GIS lze výsledné řešení vizualizovat v běžném GIS software nebo WMS. Ukázka převodu je na obrázku 4.9.



Obrázek 4.9: Zpětná transformace výsledného řešení do prostředí GIS

Zdroj: Autor

4.4.4 *Určování polohy míst obsluhy pomocí GPS a přenesení údajů do grafu sítě*

Při obsluze vrcholů, jež představují jednotlivé zákazníky, nebývají tyto vrcholy součástí geografických dat (například svoz odpadu a umístění kontejnerů). Pro zjištění polohy těchto vrcholů lze použít zařízení GPS. Nejvýhodnějším řešením získání dat je pomocí běžně dostupného přístroje GSM, který obsahuje operační systém s funkčním příručním GIS softwarem a zařízení GPS s dostatečnou přesností. Například SmartPhone s operačním systémem Windows Mobile, aplikací ArcPad od firmy ESRI, který je možné v případě nepřesného interního zařízení GPS doplnit o přesnější externí GPS zařízení s rozhraním Bluetooth⁸. Na rozdíl od klasického PDA je přístroj GSM schopen přenášet data online na WMS. Při zanášení údajů o poloze do mapového podkladu ve vektorovém formátu je nutné, aby byly vyjádřeny ve stejném souřadnicovém systému, ve kterém je mapa. Souřadnicový systém České republiky existuje ve formátu S-JTSK (plošný souřadnicový systém), který je používán v civilní geodetické službě. Druhým, méně používaným systémem, je vojenský souřadnicový systém S-42. Zařízení GPS poskytuje hodnoty v eliptickém souřadnicovém systému WGS84. Tyto eliptické souřadnice je tedy třeba transformovat na kartézské. Některé GIS softwary tuto transformaci umožňují pomocí speciálních pluginů. Software NetOpt umožňuje oboustranný převod mezi těmito dvěma souřadnicovými systémy. Ostatně matematický model převodu je dobře znám a je dostatečně popsán v příslušné literatuře [31].

Sběr dat pomocí GPS může probíhat v terénu dvěma způsoby. První způsob je offline, to znamená, že získaná data se pomocí příslušného zařízení shromažďují na přenosné datové

⁸ Bezdrátová komunikační technologie sloužící k výměně digitálních dat na krátké vzdálenosti.

médium a do GIS mapy se zanesou bez použití vzdáleného přístupu k této mapě. Druhý způsob je online, který umožňuje, například pomocí sítě GSM, získaná data v reálném čase v přijatelných intervalech zanášet do vektorové mapy na WMS. Druhý způsob je vhodným řešením pro zajištění operability svozně-rozvozní firmy.

Pokud jsou zákazníci umístěni podél komunikace, jsou místa obsluhy reprezentována vrcholy v příslušné hladině souboru SHP. Při transformaci na graf sítě a řešení ARP je nutné těmto vrcholům přiřadit hrany obsluhy. Toto lze zajistit jednoduchým způsobem pomocí zjištění délky kolmice z vrcholu k okolním hranám. Nejkratší kolmice k příslušné hraně vybere tuto hranu jako hranu obsluhy. Tímto získáme množinu hran obsluhy.

Při obsluze zákazníků rozmístěných podél komunikace je z hlediska dostupného počtu a menší složitosti algoritmů vhodnější řešení svozně-rozvozního problému s obsluhou vrcholů (VRP – *Vehicle Routing Problem*), avšak při velkém množství vrcholů obsluhy a následném převodu grafu na graf městské sítě komunikací se počet vrcholů a hran grafu může zvýšit do té míry, že je na výkonných PC časově náročné vyřešit úlohu i heuristicky. Při určování optimální trasy vozidla je kladen důraz na přesnost výsledného řešení, ale i na výpočetní čas. Z tohoto důvodu je vhodné řešit problém jako ARP. V řadě svozně-rozvozních problémů nejsou zákazníci určeni vrcholy, ale přímo hranami. Zde již není o vhodnosti volby řešení ARP pochyb (například ČUPK, zimní údržba pozemních komunikací atd.). VRP je vhodné použít v případě, že mezi jednotlivými zákazníky existuje více hran, například při rozvozu zboží na větší vzdálenosti.

4.4.5 *NetOpt a existující GIS software pro řešení ARP*

GIS software, který umožňuje řešit ARP, již existuje. Jedná se však o velice drahou záležitost ve většině případů řešící pouze VRP, nikoliv ARP. Profesionální software je vyvíjen několik let desítkami lidí. Cílem vývoje SW nástroje NetOpt je snaha alespoň se přiblížit úrovni profesionálních produktů a demonstrovat principy řešení svozně-rozvozních úloh v prostředí GIS. Prostředí tohoto SW nástroje je v příloze 8. Předchozí upravené verze byly úspěšně použity i v praxi na výpočet lokace servisních středisek firmy MUZO a.s., která v rámci České republiky instaluje, spravuje a provádí opravy bankomatů. Další použití tohoto softwaru v praxi byl výpočet tras vozidel organizace Centrum sociálních služeb a pomoci o.p.s. se sídlem v Hradci Králové, která rozváží chlazenou stravu sociálně slabším, zdravotně postiženým a nemocným občanům. Dále byl použit pro počítačovou podporu

tvorby spojů firmy Connex Východní Čechy a.s. v rámci zajištění dopravní obslužnosti mikroregionu Heřmanův Městec.

Příklady existujících profesionálních SW nástrojů:

- FleetRoute (řešení ARP) od firmy CIVIX,
- RouteSmart (řešení ARP) od firmy RouteSmart Technologies,
- TransCAD (řešení VRP, síťová analýza, dopravní analýza, plánování v dopravě) od firmy Caliper Corporation,
- StreetSync (řešení VRP) od firmy Route Solution, Inc.,
- Single Depot Route Planning Software (řešení VRP) od firmy Paragon,
- TourSolver (řešení VRP) od firmy Magellan Ingénierie,
- PlanTour (řešení VRP) od firmy PASS Logistics Solutions atd.

4.5 Postup sestavení, verifikace a implementace metodiky

Při sestavování metodiky tvorby tras obslužných vozidel provádějících ČUPK bylo třeba nejprve stanovit požadavky na tuto metodiku. Metodika by měla v praktické aplikaci v SW nástroji NetOpt sestavovat trasy obslužných vozidel v co nejkratším čase s minimální odchylkou od exaktního řešení a měla by být schopna pracovat s velkými objemy vstupních dat ve formě prvků grafu sítě městských komunikací. Ve větších městech a průmyslových aglomeracích jde o grafy s řádově tisíci hranami a vrcholy. Mezi těmito požadavky existuje vzájemná vazba. Pokud by byl řešen tento NP-těžký problém v co nejkratším čase, bylo by dosaženo horších výsledků, a naopak. Velikost vstupních dat ovlivňuje oba předchozí požadavky. S rostoucím objemem dat roste čas potřebný na výpočet a klesá přesnost řešení. Navržená metodika by měla tedy pracovat s velkým objemem vstupních dat a za přijatelný čas poskytovat co nejpřesnější řešení v podobě minimálních délek tras obslužných vozidel.

Z analýzy metod ARP vyplynulo, že vhodnou podkategorií tohoto problému pro sestavení metodiky je RPP, konkrétně metody typu CARP, ve kterých je uvažována kapacita vozidla. Problém je řešen na smíšeném grafu, který lépe popisuje síť městských komunikací. Důvodem je fakt, že předmětem obsluhy při ČUPK nejsou všechny komunikace ve městě (části města) současně, ale obsluhují se vždy jen některé. Byla tedy vyloučena podkategorie CPP, kde jsou předmětem obsluhy všechny hrany grafu sítě městských komunikací. Dostupné metody řešící CARP v sobě nezahrnují tyto vlastnosti: graf, na kterém je problém řešen, obsahuje orientované hrany, neorientované hrany a multihrany. V křižovatce jsou zakázány

nepřípustné pohyby vozidla (U-turnus, jízda do protisměru). Je uvažována časová penalizace při průjezdu křižovatkou. V grafu mohou existovat nesouměrné hrany, které mají v každém směru různá ohodnocení. Graf může obsahovat více vrcholů s deponovacími místy pro vyprazdňování vozidel.

Aby byly v navržené metodice splněny tyto požadavky a vlastnosti, bylo nutné na základě analýzy metod ARP vyloučit metody exaktní. Vzhledem k časové složitosti nelze do těchto metod včlenit dodatečné vlastnosti úlohy takovým způsobem, aby byly instance řešeny v přijatelném čase. V tomto případě by bylo obtížné i samotné včlenění dodatečných vlastností.

Při sestavování metodiky tvorby tras obslužných vozidel provádějících ČUPK byly použity konstruktivní heuristiky PS, AM a Ulusoyova dvoufázová heuristika, spadající do kategorie CARP, které byly upraveny pro MCARP o dodatečné vlastnosti zmíněné v oddílu 4.2.1. Původní vlastnosti těchto heuristik jsou uvedeny také v oddílu 4.2.1. Úpravy konstruktivních heuristik pro MCARP s dodatečnými vlastnostmi jsou popsány v oddílu 4.2.2. Takto upravené heuristiky byly včleněny do memetického algoritmu. Princip včlenění je uveden v oddílu 4.2.7. Tímto vznikla metaheuristika, která je popsána v oddílu 4.2.8. Vývojový diagram algoritmu metaheuristiky je v příloze 9. Výsledná metodika je implementací této metaheuristiky do SW nástroje NetOpt, který pracuje s vektorovými GIS daty (vektorovými mapovými podklady). Tento SW nástroj optimalizuje trasy obslužných vozidel provádějících ČUPK.

4.5.1 Implementace metodiky do SW NetOpt

V tomto oddílu je popsána aplikační část metodiky. SW nástroj NetOpt je naprogramován v prostředí Borland Delphi. Jedná se o aplikaci s CAD/GIS prostředím, ve které je možné na vektorovém mapovém podkladu řešit úlohy typu VRP nebo ARP, popřípadě v upravené verzi lokační úlohy.

Aplikace se skládá ze dvou hlavních oken *Graf sítě* (CAD prostředí) a *GIS* (GIS prostředí). V okně *GIS* je zobrazována vektorová mapa sítě městských komunikací. Tato mapa může být tvořena více hladinami, tzn. kromě komunikací lze zobrazit i budovy, vodní toky, železniční tratě atd. Mapy se zobrazují v plošném souřadnicovém systému S-JTSK. Na mapě lze vybrat oblast, na které má být provedeno ČUPK. Vybraná vektorová data včetně svých atributů (názvy ulic, třídy pozemních komunikací atd.) se transformují na graf sítě městských komunikací, kde je provedeno odstranění nekorektnosti dat (viz oddíl 4.4.3). Graf

sítě městských komunikací je zobrazen v okně *Graf sítě* a lze jej dále upravovat – odstraňovat, vkládat, posouvat vrcholy a hrany, editovat atributy vrcholů a hran. Dále je možné v oblastech, kde nejsou dostupná vektorová data, vložit rastrový obrázek s mapou a vektorizovat jej. Pokud je graf korektní, může být proveden výpočet tras obslužných vozidel provádějících ČUPK. Nejprve je nutné přiřadit hranám požadavky obsluhy, implicitně je předpokládána hodnota 1, tzn. každá část komunikace je obslužena jednou. Jsou nastaveny penalizace při průjezdu křižovatkami, stanovena kapacita obslužných vozidel a jednotlivým hranám přiřazeny rychlosti, kterými budou projížděny obslužnými vozidly. Dále je možné rozhodnout, zda bude pozemní komunikace obsluhována pouze v jednom směru nebo obousměrně, popřípadě zda bude zakázán průjezd některými úseky pozemních komunikací. Pokud se jedná o obousměrnou obsluhu komunikací, je graf sítě převeden do městského režimu (popsáno v oddílu 4.4.3). Následuje samotný výpočet tras. Trasy jsou zobrazeny v tabelárním seznamu spolu s údaji o své délce a čase potřebném na obsluhu. Při výběru trasy z tohoto seznamu je trasa zobrazena na grafu sítě. Trasy lze poté uložit ve formátu aplikace NetOpt, formátu GIS nebo formátu pro zařízení GPS. Každé trase je možné ze seznamu přiřadit typ vozidla a řidiče. Výstupem může být uspořádaný tištěný seznam ulic a křižovatek pro řidiče, nebo lze trasu importovat do zařízení GPS umístěném v obslužném vozidle. Řidič obslužného vozidla může sledovat na zařízení GPS vlastní polohu a zároveň je mu zobrazována naplánovaná trasa.

4.5.2 Verifikace navržené metodiky

Navržená metodika byla verifikována v SW nástroji NetOpt na deseti náhodně generovaných instancích⁹. Na vektorové mapě statutárního města Pardubice byly podle počtu obousměrně obsluhovaných pozemních komunikací vybrány oblasti, kde bude provedeno ČUPK. Počet hran obsluhy byl z intervalu 50–500. V těchto oblastech byl postupně proveden výpočet tras obslužných vozidel upravenými metodami AM, PS, Ulusoyovou heuristikou a metaheuristikou MA. Aby bylo možné zjistit odchylky řešení od optima, byla pro každou instanci vypočtena celková délka hran obsluhy, která představuje dolní hranici řešení. Hodnota optimálního řešení je vždy větší nebo totožná s hodnotou dolní hranice řešení. Jednotlivé délky tras byly s touto dolní hranicí porovnávány. Ukázalo se, že nejlepších výsledků v přijatelném čase dosáhla upravená metaheuristika MA, která se lišila od dolní hranice řešení v průměru o 2,3 %. U upravených konstruktivních heuristik AM a PS byl

⁹ Náhodnou instancí se zde rozumí soubor komunikací v náhodně zvolené části města, které mají být obsluženy vozidlem.

tento rozdíl 5,5 % a 8,3 %. Upravená dvoufázová Ulusoyova heuristika se lišila od dolní hranice řešení o 4,5 %. Čas potřebný pro výpočet jednotlivými metodami byl v průměru u konstruktivních heuristik AM a PS 2,6 s a 2,2 s, u dvoufázové Ulusoyovy heuristiky 3 s a u metaheuristiky MA 3,7 s. Zde je patrná závislost přesnosti řešení na výpočetním čase. U algoritmů, které poskytly horší řešení, byl výpočetní čas menší a naopak.

V příloze 10 je porovnání časové náročnosti a přesnosti výpočtu MCARP upravenými konstruktivními heuristickými metodami a metaheuristikou. Tučně jsou zvýrazněny nejlepší dosažené hodnoty. Celý výpočet proběhl na PC s konfigurací: procesor Intel Core 2,2 GHz, 4 GB RAM. Příklad náhodné instance ve statutárním městě Pardubice – části Židov je v příloze 12. V příloze 12a) je graf sítě pozemních komunikací této oblasti, v příloze 12b) je tento graf převeden do městského režimu. Výpočty probíhaly na grafu v příloze 12b). Výsledné hodnoty pro tuto instanci jsou v příloze 10, poslední řádek tabulky. Doba výpočtu trasy upravenou metaheuristikou MA na této instanci o 500 hranách obsluhovaných v obou směrech byla 8 s a výsledné řešení se od dolní hranice řešení lišilo o 4,5 %. Také bylo dosaženo zlepšení řešení úpravou metaheuristiky MA ve fázi křížení chromozómů. Tato úprava je popsána v oddílu 4.2.7. V příloze 5 jsou uvedena řešení dosažená křížením chromozómů LOX a upraveným OX, tučně jsou uvedeny nejlepší hodnoty. Metaheuristika MA s upraveným křížením OX poskytovala v průměru o 0,9 % lepší výsledky než s křížením LOX.

5 VYHODNOCENÍ A DISKUSE VÝSLEDKŮ

Firmy provádějící čištění a údržbu pozemních komunikací ve větších městech a průmyslových aglomeracích jsou nuceny v konkurenčním prostředí snižovat ceny za své služby. Nemohou však slevit z kvality služeb. Aby zachovaly své přiměřené zisky, musejí snižovat své náklady. Efektivní cestou k dosažení tohoto cíle je optimalizace tras obslužných vozidel. Tímto způsobem lze omezit náklady na pohonné hmoty. V případě fixních nákladů se jedná o omezení nákladů na nákup obslužných vozidel a mzdy řidičů. Obslužná vozidla jsou lépe využívána a jejich počet lze tedy snížit, stejně tak i počet řidičů.

Firma Služby města Pardubic a.s., která provádí čištění a údržbu pozemních komunikací ve statutárním městě Pardubice, dává k dispozici řidičům obslužných vozidel pouze jmenný seznam ulic, které má dané vozidlo během směny obsloužit. V seznamu není určeno pořadí, v jakém se mají ulice obsluhovat, viz příloha 11. Pořadí obsluhy ulic záleží na vlastní úvaze řidiče vozidla. Neexistuje přesný záznam, jakou trasu obslužné vozidlo projelo. Ze statistických údajů jsou k dispozici pouze délky obsluhovaných ulic, četnost jejich obsluhy a výše spotřeby pohonných hmot za rok. Z tohoto důvodu není možné porovnání vypočtených tras SW nástrojem NetOpt se skutečnými trasami vozidel Služeb města Pardubic a.s. Pokud se výsledná řešení dosažená metodikou, která byla v této práci sestavena, pohybovala v průměru o 2,3 % nad dolní hranici řešení, lze tvrdit, že výsledné trasy se blíží optimálním, popřípadě jsou optimálními trasami. Z toho lze usoudit, že ve Službách města Pardubic a.s. by optimalizací mohlo dojít ke snížení nákladů, ale vzhledem k faktu, že nejsou k dispozici záznamy tras obslužných vozidel, nelze tuto úsporu vyčíslit. Jednou z možností je, aby Služby města Pardubic a.s. použily řešení získané SW nástrojem NetOpt a po dobu jednoho roku obslužná vozidla jezdila podle vypočtených tras. Následovalo by porovnání spotřeby pohonných hmot se spotřebou za minulá období.

Ze statistických údajů firmy Minerva a.s. [27], která distribuuje SW nástroj optimalizující trasy vozidel přepravujících zásilky metodami teorie grafů a matematického programování, dochází optimalizací ke snížení nákladů na přepravu v rozmezí 15–30 % (problém VRP). Vzhledem k faktu, že náklady na ČUPK Služeb města Pardubic a.s. dosáhly za rok 2010 výše 17 mil. Kč (viz tabulka 1), předpokládané úspory v rozmezí 15–30 % by znamenaly významné snížení nákladů.

6 NÁVRHY NA VYUŽITÍ VÝSLEDKŮ

Metodika tvorby tras obslužných vozidel provádějících ČUPK je navržena v obecné rovině tak, aby mohla být použita i pro řešení ostatních svozně-rozvozních úloh s obsluhou hran. S čistotou a pořádkem ve větších městech souvisí i svoz komunálního odpadu. Nádoby pro sběr odpadu a kontejnery jsou rozmístěny na okrajích pozemních komunikací. Komunální vozidlo vždy zastaví a nádobu vyprázdí, jedná se tedy o obsluhu vrcholů. Takovýto problém je řešen metodami VRP. Avšak velký počet odpadních nádob rozmístěných podél úseku pozemní komunikace zvyšuje objem vstupních dat pro výpočet. Graf obsahuje mimo vrcholů reprezentujících křižovatky také vrcholy s umístěním odpadních nádob. Problém může být převeden na ARP. Každé hraně v grafu je přiřazeno ohodnocení, které představuje počet obsluhovaných odpadních nádob na této hraně. Hrana se stane hranou obsluhy a z grafu jsou vypuštěny vrcholy s odpadními nádobami. Objem vstupních dat se značně redukuje a problém lze řešit metodikou navrženou v této práci.

Může nastat situace, kdy je obslužné vozidlo nuceno přehodnotit naplánovanou trasu. Navrženou metodiku by bylo možné využít při operativním řízení obslužných vozidel provádějících ČUPK. Vzhledem k výpočetní rychlosti metodiky v SW nástroji NetOpt by mohla být vozidlům vybavených zařízením GPS a digitálním komunikačním datovým zařízením (např. GSM zařízení) z dispečinku operativně měněna naplánovaná trasa. Výpočetní rychlost metodiky je patrná z přílohy 10.

Datové přenosy, uskutečňované prostřednictvím GSM sítě mezi vozidlem a dispečinkem, jsou zpoplatněny. Alternativou pro sběr a přenos polohových dat je systém APRS (*Automatic Position Reporting System*). Jedná se o obousměrný radiový přenos informací o poloze mezi vozidlem vybaveným radiostanicí s GPS modulem a dalšími radiostanicemi s možností přenosu dat na WMS. Pro testovací účely je tento systém velice výhodný pro bezplatný provoz. Jeho nevýhody spočívají v menší frekvenci zasílání dat o poloze, nutnosti vlastnit radioamatérskou licenci (koncesi) a zcela veřejnému přenosu dat na dané frekvenci, která je pro tento účel vyhrazena (v Evropě 144,8 MHz).

Vazbou SW nástroje na WMS by bylo možné sledovat polohu obslužných vozidel přes Internetové rozhraní a trasy i s časovými údaji na tento server ukládat pro pozdější použití. Tím by se zefektivnila a usnadnila práce plánovacím dispečerům a zkvalitnilo by se řízení rozhodovacích a logistických procesů.

ZÁVĚR

Návrhy a výsledky dosažené v disertační práci naplňují cíle stanovené v kapitole 2. Na základě podrobné analýzy současných trendů v oblasti řešení svozně-rozvozních úloh s obsluhou hran byla navržena metodika pro výpočet tras obslužných vozidel provádějících ČUPK. Při sestavování metodiky byl kladen důraz na malou časovou složitost výpočtu a na kvalitu výstupu v podobě délek navržených tras, resp. minimalizaci bezobslužných km. Po testech na náhodně vygenerovaných instancích, v SW nástroji NetOpt, byly předem vyloučeny metody řešící VRP a metody exaktní, z důvodu časové náročnosti výpočtu a v některých případech i obtížného matematického vyjádření instance. Ze zbývajících heuristických a metaheuristických metod poskytovaly přesnější výsledky metody metaheuristické při zachování přibližně stejné časové náročnosti. Na náhodně vygenerovaných instancích větších rozměrů, v SW nástroji NetOpt, byla vzhledem k současné situaci sestavování plánů tras obslužných vozidel provádějících ČUPK v ČR ověřena hypotéza, že navržení vhodné metodiky pro řešení svozně-rozvozního problému s obsluhou hran a její aplikace v softwarovém nástroji s podporou GIS povede k racionalizaci stávajícího řešení problému čištění a údržby pozemních komunikací ve větších městech. Tyto plány byly sestavovány pracovníky firem za minimálního využití optimalizačních nástrojů a aparátu metod operačního výzkumu. Za pomoci SW nástroje, lze tyto trasy sestavovat v řádu sekund nebo minut, oproti několika dnům v případě, že trasy sestavují pracovníci firmy např. v aplikaci MS Excel s využitím dat z minulých let (viz příloha 13). Výpočet tras je založen na matematickém aparátu a při větším objemu vstupních dat jsou navržené trasy optimální, nebo se k optimu blíží. Navrženou metodiku lze s jistými úpravami použít i pro sestavování tras obslužných vozidel provádějících svoz odpadu, zimní údržbu pozemních komunikací, rozvoz zásilek po městě atd.

Metodika byla implementována a verifikována v SW nástroji na bázi aplikací GIS, pracujících s reálnými digitálními daty pozemních komunikací. Zde byla také demonstrována rychlost výpočtu. Výsledné trasy lze zobrazovat na každém zařízení, které podporuje načítání tras v některém z formátů pro GPS. Dále lze trasy zobrazovat ve všech aplikacích na bázi GIS. Součástí výstupu jsou i tabulkové seznamy tras s názvy ulic, křižovatek a časem průjezdu vozidla. Každé trase lze z databáze přiřadit vozidlo a řidiče. Vypočtené trasy lze také dotačně ručně upravovat.

Jako neméně významný přínos je rovněž možno zmínit sestavení uceleného systému algoritmů řešících ARP, kde jsou ke každému problému uváděny jak metody heuristické, tak

matematické modely úloh LP. Vedlejším přínosem je demonstrace obecných principů začlenění řešení svozně-rozvozních úloh do prostředí GIS. Dále je to rozšíření konstruktivních heuristik PS, AM a Ulusoyovy dvoufázové heuristiky o další vlastnosti potřebné pro stanovení tras obslužných vozidel provádějících ČUPK. A také úpravu genetického algoritmu ve fázi křížení.

Řešení problému přineslo i řadu otázek na budoucí rozvoj výzkumu v této oblasti. Některé typy úloh z oblasti ARP nemají dosud svá polynomiální řešení. Vzhledem k náročnosti sestavování matematických modelů úloh LP nejsou některé vícekriteriální ARP matematicky popsány. Stejně tak je možné zkoumat implementaci dalších možných kritérií do řešení jako např. ekonomické ukazatele, technologické provozní ukazatele atd. Sledování obslužných vozidel v reálném čase a operativní změny tras těchto vozidel je další široká oblast výzkumu.

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

- [1] Zákon č. 13/1997 Sb., o pozemních komunikacích, ve znění pozdějších předpisů.
- [2] Vyhláška č. 104/1997 Sb., kterou se provádí zákon o pozemních komunikacích.
- [3] Zákon č. 361/2000 Sb., o provozu na pozemních komunikacích a o změnách některých zákonů (zákon o silničním provozu), ve znění pozdějších předpisů.
- [4] Zákon č. 128/2000 Sb., o obcích, ve znění pozdějších předpisů.
- [5] **BARTSCH H. J.: *Matematické vzorce* Praha: Academia, 2008. 834 s. ISBN 80-200-1448-9.**
- [6] **VOLEK, J.: *Operační výzkum I.* Pardubice: Univerzita Pardubice, 2002. 111 s. ISBN 80-7194-410-6.**
- [7] **VOLEK, J.; LINDA, B.: *Lineární programování.* Pardubice: Univerzita Pardubice, 2007. 140 s. ISBN 978-80-7395-038-5.**
- [8] **JAN VAN LEEUWEN: *Algorithms and Complexity Volume A.* Cambridge: MIT Press, 1990. 254 s. ISBN 978-0-444-88071-0.**
- [9] **KOZAN, E.; OHUCHI, A.: *Operation research, management science at work.* Dordrecht: Kluwer Academic Publishers Group, 2002. 440 s. ISBN 0-7923-7588-2.**
- [10] EDMONDS, J.; JOHNSON, E. L.: *Matching, Euler tours and the Chinese postman.* Mathematical Programming Journal, Vol. 5, 1973. s. 88–124. ISSN 0025-5610.
- [11] **AHUJA, R.K.; MAGNANTI, T.L.; ORLIN, J.B.: *Network Flows : Theory, Algorithms, and Applications.* New Jersey: Prentice Hall, 1993. 864 s. ISBN 978-0-136-17549-0.**
- [12] RAGHAVAFARI, B.; VEERASAMY, J.: *A 3/2-approximation algorithm for the Windy Postman Problem* [online]. Dallas: The University of Texas at Dallas, c2009 [cit. 2010-1-5]. Dostupné z <<http://www.utdallas.edu/library/>>.

- [13] GUGUAN, M.; PULLEYBLANK, W.: *Eulerian Orientations and Circulations*. Algebraic and Discrete Methods Journal, Vol. 6, Issue 4, 1985, s. 657–664. ISSN 0196-5212.
- [14] BRUCKER, P.: *The Chinese Postman Problem for Mixed Graphs*. Lecture notes in Computer Science, Vol. 100, 1981. s. 354–366. ISSN 0302-9743.
- [15] *Vehicle Routing Problems* [online]. Genova: Paolucci M., c2006 [cit. 2010-1-15]. Dostupné z <<http://www.discovery.dist.unige.it/didattica/LS/VRP.pdf>>.
- [16] RAGHAVAFARI, B.; VEERASAMY, J.: *Approximation Algorithms for Mixed Postman Problem* [online]. c2007 [cit. 2010-3-5]. Dostupné z <<http://citeseerx.ist.psu.edu/>>.
- [17] DROR, M.; STERN, H.; TREDEAU, P.: *Postman tour on a graph with precedence relation on arcs*. Networks – An International Journal, Vol. 17, 1987. s. 283–294. ISSN 1097-0037.
- [18] FREDERICKSON, G. N.; HECHT, M. S.; KIM, C. E.: *Approximation Algorithms for Some Postman Problems*. Journal of the ACM, Vol. 7, 1978. s. 538–554. ISSN 0004-5411.
- [19] CHRISTOFIDES, N.; CAMPOS, V.; CORBERAN, A.; MOTA, E.: *An Algorithm for the Rural Postman Problem*. In: Report IC.O.R.85, London: Imperial College, 1981. ISBN není.
- [20] CORBERAN, A.; SANCHIS, J. M.: *A Polyhedral Approach to the Rural Postman Problem*. European Journal of Operation Research, Vol. 79, 1994. s. 95–114. ISSN 0377-2217.
- [21] CHRISTOFIDES, N.; CAMPOS, V.; CORBERAN, A.; MOTA, E.: *An Algorithm for the Rural Postman Problem on a Directed Graph*. Mathematical Programmig Studies, Vol. 26., 1986. s. 155–166. ISSN 0025-5610.
- [22] GOLDEN, B. L.; WONG, R. T.: *Capacitated Arc Routing Problems*. Networks An International Journal, Vol. 11, 1981. s. 305–315. ISSN: 0028-3045.

- [23] BELENGUER, J. M.; BENAVENT, E.: *Polyhedral Results on the Capacitated Arc Routing Problem*. In: Working Paper, Departamento de Estadística e Investigación Operativa, Universidad de Valencia, 1991. 92 s. ISBN není.
- [24] BELENGUER, J. M.; BENAVENT, E.: *Cutting plane algorithm for the capacitated arc routing problem*. Computers and Operations Research Journal, Vol. 30, 2003. s. 705–728. ISSN 0305-0548.
- [25] PADBERG, M. W.; RAO, M. R.: *Odd minimum cut-sets and b-matchings*. Mathematics of Operations Research Journal, Vol. 7, 1982. s. 64–80. ISSN 1526-5471.
- [26] GOLDEN, B.L; DEARMON, J.S.; BAKER, E.K.: *Computational experiments with algorithms for a class of routing problems*. Oxford: Computers and Operations Research journal, Vol. 10, 1983. s. 47–59. ISSN 0305-0548.
- [27] ULUSOY, G.: *The Fleet Size and Mixed Problem for Capacitated Arc Routing*. Elsevier Science, European Journal of Operational Research, Vol. 22, 1985. s. 329–337. ISSN 0377-2217.
- [28] **HYNEK J.: *Genetické algoritmy a genetické programování*. Praha : Grada, 2008. 200 s. ISBN 978-80-247-2695-3.**
- [29] LACOMME, P.; PRINS, C.; RAMDANE-CHERIF, W.: *Competitive Memetic Algorithms for Arc Routing Problems*. Annals of Operations Research, Vol. 131, 2004. s. 159–185. ISSN 1572-9338.
- [30] *Typy GIS software* [online]. Liberec: Technická univerzita v Liberci, c2007 [cit. 2010-1-3]. Dostupné z <<http://gisdoskol.fp.tul.cz/>>.
- [31] HRDINA, Z.: *Transformace souřadnic ze systému WGS84 do systému S-JTSK*. Skriptum, Praha: ČVUT, 1997. 21 s. ISBN není.
- [32] *Statistický údaj firmy Minerva a.s.* [online]. ČR: oficiální stránky firmy Minerva a.s., c2011 [cit. 2011-3-4]. Dostupné z <<http://www.minerva-is.cz/optimalizace-a-planovani-tras.html>>

Poznámka: **tučně** jsou uvedeny knižní publikace.

PUBLIKAČNÍ ČINNOST AUTORA

- [1] VÍZNER, F.: *Optimalizace rozložení servisních středisek MUZO a.s.* In: Dopravní systémy, Pardubice, 29.11.–30.11.2005, s. 350–355, ISBN 80-7194-805-5.
- [2] VÍZNER, F.: *Vylepšení suboptimálních řešení výpočtu Hamiltonovského cyklu.* In: Sborník příspěvků čtvrté mezinárodní konference „Nové výzvy pro dopravu a spoje“, Pardubice, září 2006, s. 1025–1029, ISBN 978-80-7395-138-2.
- [3] VÍZNER, F.: *Řešení svozně-rozvozních úloh (VRP) s využitím GIS.* In: Úlohy diskrétní optimalizace v dopravní praxi – Lokace středisek s negativními vlivy na okolí, Pardubice, 13.11.–14.11.2008, s. 77–81, ISBN 978-80-7395-138-2.
- [4] VÍZNER, F.: *Metodika řešení svozně-rozvozních úloh.* In: Úlohy diskrétní optimalizace v dopravní praxi – Telematika v distribučních a svozných úlohách, Pardubice, 26.11.–27.11.2009, s. 108–118, ISBN 978-80-7395-193-1.
- [5] VÍZNER, F.: *Řešení svozně-rozvozního problému s obsluhou hran.* In: Úlohy diskrétní optimalizace v dopravní praxi – Řešení distribučních a svozových úloh, díl III., Pardubice, 8.6.–9.6.2009, s. 101–107, ISBN 80-7194-880-2.
- [6] VÍZNER, F.: *NetOpt.* Softwarová aplikace pod MS Windows, ID kód ODB: NetOpt v.1., Pardubice, 2009.
- [7] VÍZNER, F.: *Základní metody řešení problému čínského poštáka.* In: Úlohy diskrétní optimalizace v dopravní praxi – Kvantitativní metody optimalizace v dopravních a logistických systémech, Pardubice, 23.4.–30.4.2010, s. 193–199, ISBN 978-80-7395-297-6.
- [8] VÍZNER, F.: *The methodics of computing of optimal waste collection routes in towns and industrial agglomerations.* In: Sborník příspěvků konference LOGI 2010, Pardubice, 19.11.2010, s. 380–392, ISBN 978-80-7399-205-7.

SEZNAM TABULEK

Tabulka 1.1: Celkové náklady na ČUPK ve statutárním městě Pardubice za rok 2010	12
Tabulka 3.1: Metody řešení CPP	56
Tabulka 3.2: Metody řešení RPP	71
Tabulka 4.1: Ukázka křížení chromozómů v memetickém algoritmu	92

SEZNAM OBRÁZKŮ

Obrázek 3.1:	Převod na graf městského režimu	32
Obrázek 3.2:	Neorientovaný graf s vrcholy lichého stupně	36
Obrázek 3.3:	Princip podmínek nerovnosti rozvoje	38
Obrázek 3.4:	Ukázka algoritmu koncového párování	40
Obrázek 3.5:	Ukázka algoritmu pro nalezení nejlevnějšího toku	41
Obrázek 3.6:	Příklad smíšeného grafu, z kterého nelze sestrojít E -graf	46
Obrázek 3.7:	Příklad transformace smíšeného grafu na symetrický graf	46
Obrázek 3.8:	Smíšený graf s lichými množinami vrcholů S	49
Obrázek 3.9:	Heuristika MCPP1	53
Obrázek 3.10:	Příklad řešení HCPP	55
Obrázek 3.11:	Procedura předběžného zpracování grafu	57
Obrázek 3.12:	Algoritmus LARCEARCS	64
Obrázek 3.13:	Algoritmus SMALLARCS	66
Obrázek 4.1:	Zakázané a přípustné pohyby vozidla v křižovatce, otáčení v křižovatce a jejich penalizace	81
Obrázek 4.2:	Iterace Path-Scanning heuristiky s kritériem $krit_1$	84
Obrázek 4.3:	AM heuristika	86
Obrázek 4.4:	Neorientovaný graf, na kterém je ukázán princip memetického algoritmu	93
Obrázek 4.5:	Procedura Split	95
Obrázek 4.6:	Ukázka 2-opt záměny na dvojici tras T_i^R a T_j^R	97

Obrázek 4.7: Problém nekorektnosti GIS dat	102
Obrázek 4.8: Ukázka převodu síťového grafu pomocí software NetOpt do městského režimu	103
Obrázek 4.9: Zpětná transformace výsledného řešení do prostředí GIS	104

SEZNAM PŘÍLOH

Příloha 1:	Kropící vůz Služeb města Pardubic a.s.	I
Příloha 2:	Komunální vysavač Glutton Služeb města Pardubic a.s.	I
Příloha 3:	Vztahy mezi kategoriemi svozně-rozvozních úloh	II
Příloha 4:	Vývojový diagram znázorňující postup při sestavování metodiky pro tvorbu tras obslužných vozidel provádějících ČUPK	III
Příloha 5:	Výsledky poskytované memetickým algoritmem MCARP – křížení LOX a upravené OX	IV
Příloha 6:	Zobrazení polohy obslužného vozidla, jedoucího na trase stanovené SW nástrojem NetOpt, na zařízení GPS	IV
Příloha 7:	Vizualizace tras v Google maps	V
Příloha 8:	Prostředí SW nástroje NetOpt	VI
Příloha 9:	Vývojový diagram metaheuristiky řešící tvorbu tras obslužných vozidel provádějících ČUPK	VII
Příloha 10:	Časová náročnost a výsledky poskytované algoritmy MCARP	VIII
Příloha 11:	Jmenný seznam ulic určený pro řidiče obslužného vozidla Služeb města Pardubic a.s.	VIII
Příloha 12:	Náhodně vygenerovaná instance ARP v SW NetOpt – soubor komunikací Pardubice-Židov	IX
Příloha 13:	Současný stav sestavování plánů tras obslužných vozidel provádějících ČUPK v Pardubicích v aplikaci MS Excel, rok 2010	X

PŘÍLOHY

Příloha 1: Kropící vůz Služeb města Pardubic a.s.



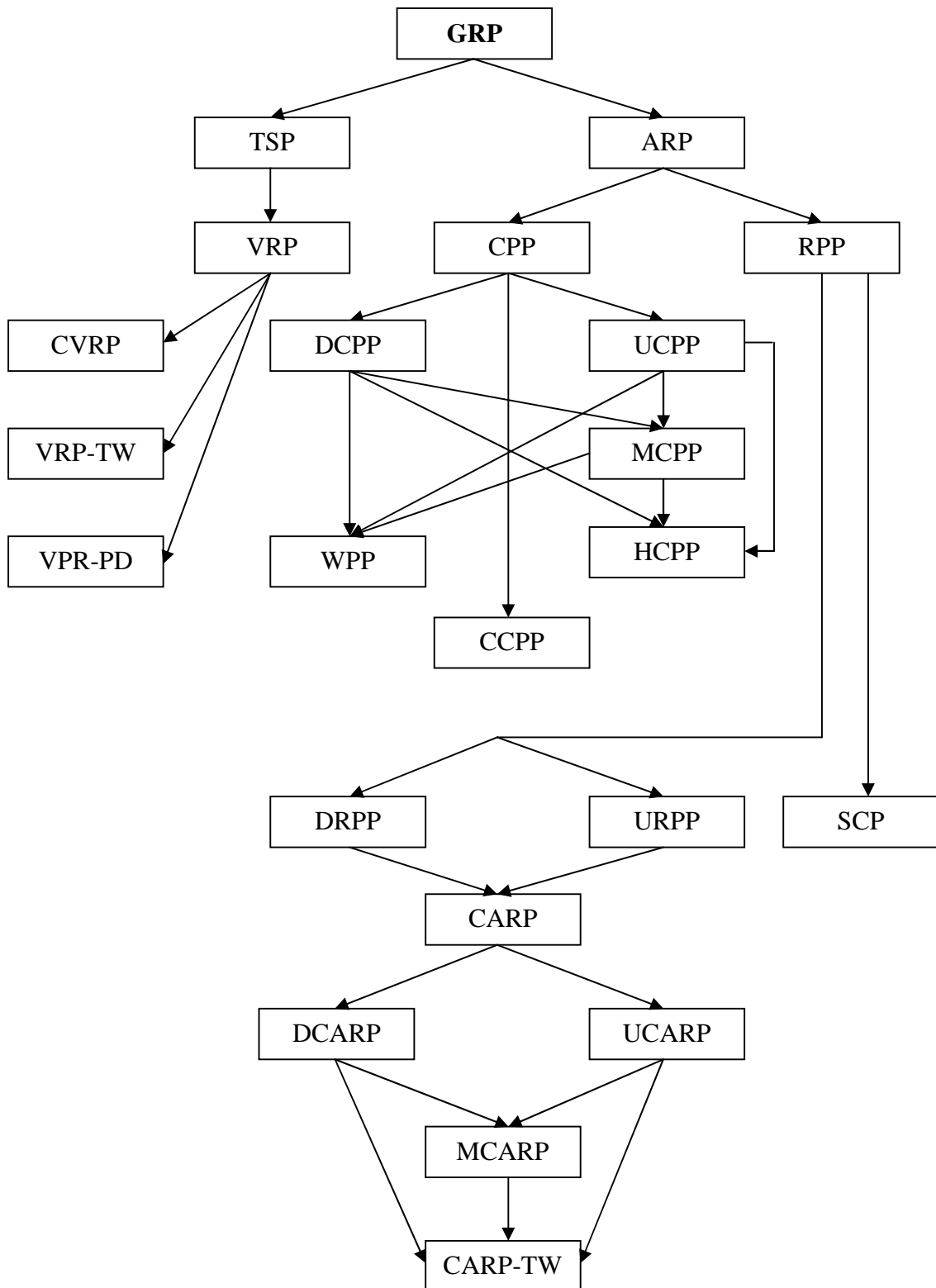
Zdroj: Oficiální stránky Služeb města Pardubic a.s., dostupné z <<http://www.smp-pce.cz/>>

Příloha 2: Komunální vysavač Glutton Služeb města Pardubic a.s.



Zdroj: Oficiální stránky Služeb města Pardubic a.s., dostupné z <<http://www.smp-pce.cz/>>

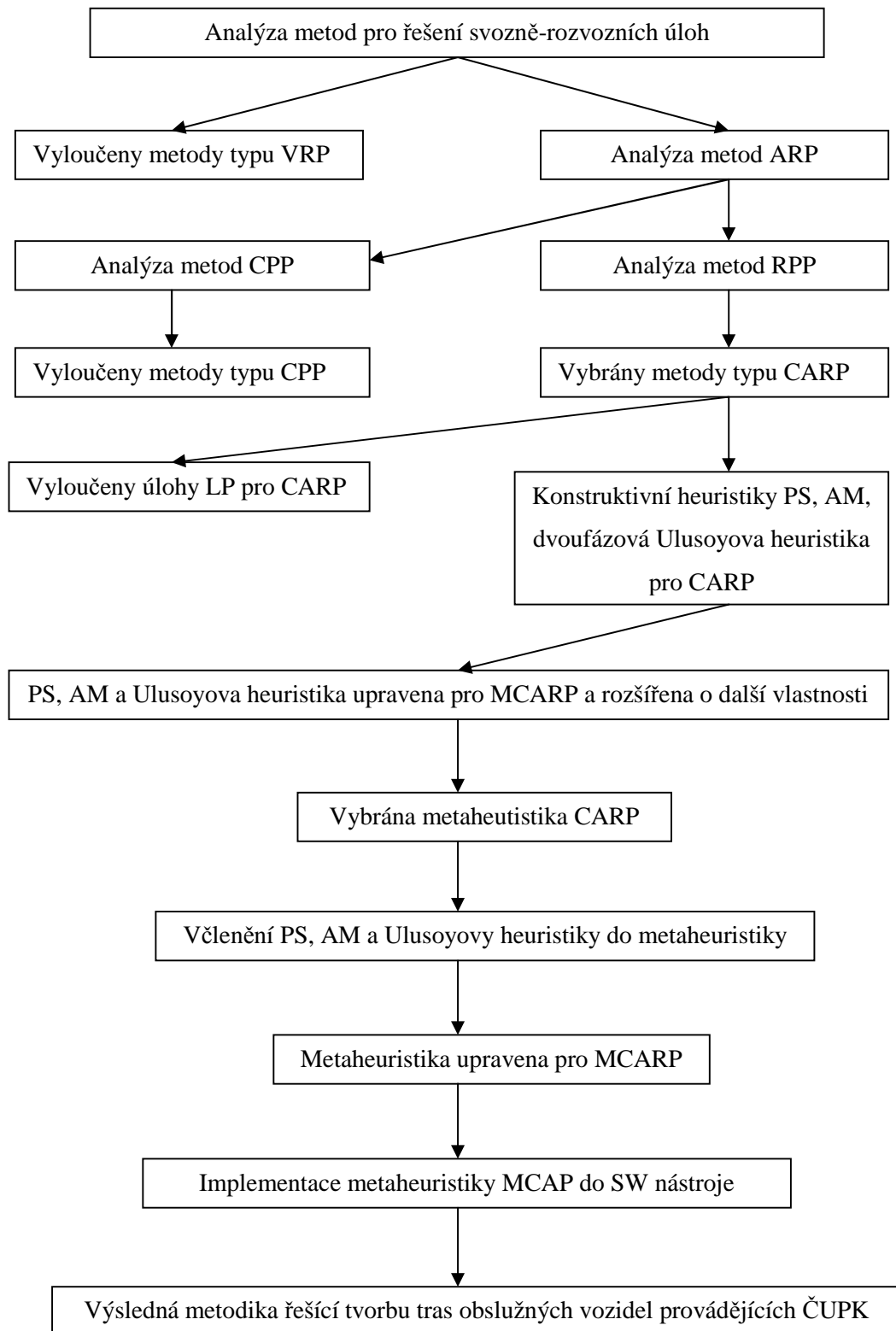
Příloha 3: Vztahy mezi kategoriemi svozně-rozvozních úloh



Poznámka: VRP-PD – Vehicle Routing Problem with Pick-Up and Delivery, CVRP – Capacitated Vehicle Routing Problem, VRP-TW – Vehicle Routing Problem with Time Windows

Zdroj: Autor

Příloha 4: Vývojový diagram znázorňující postup při sestavování metodiky pro tvorbu tras obslužných vozidel provádějících ČUPK



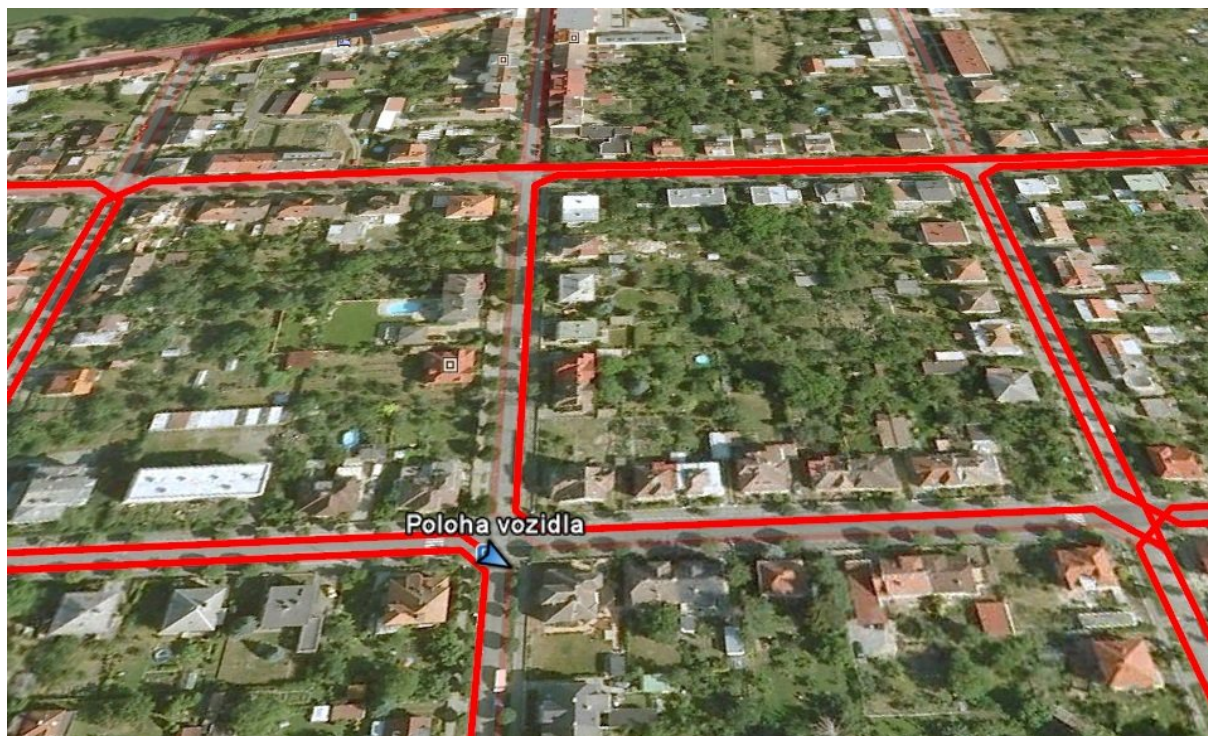
Zdroj: Autor

Příloha 5: Výsledky poskytované memetickým algoritmem MCARP – křížení LOX a upravené OX

náhodně generované instance		memetický algoritmus	
		křížení LOX	upravené křížení OX
<i>celková délka hran obsluhy [10 x m]</i>	<i>počet hran obsluhy [ks]</i>	<i>délka trasy [10 x m]</i>	<i>délka trasy [10 x m]</i>
368	50	396	383
748	100	774	774
1057	150	1079	1079
1351	200	1443	1430
1724	250	1712	1682
2174	300	2209	2199
2391	350	2441	2423
2757	400	2789	2789
3115	450	3174	3160
3536	500	3738	3698

Zdroj: Autor (údaje stanovené výpočtem v rámci disertační práce – SW nástroj NetOpt)

Příloha 6: Zobrazení polohy obslužného vozidla, jedoucího na trase stanovené SW nástrojem NetOpt, na zařízení GPS



Zdroj: Autor

Příloha 7: Vizualizace tras v Google maps



Zdroj: Autor (mapové podklady Google maps)

Příloha 8: Prostorové SW nástroje NetOpt

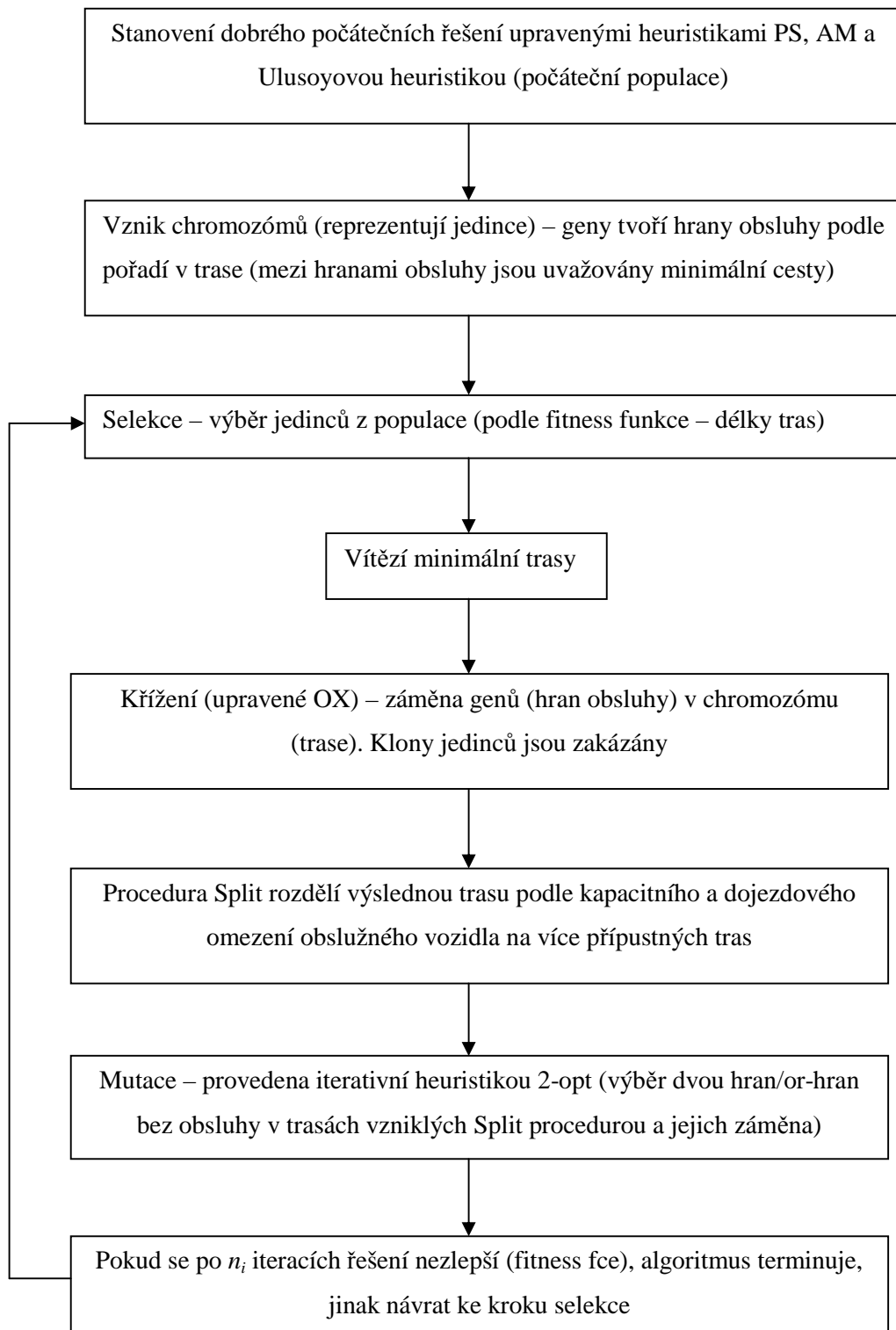
The screenshot displays the NetOpt software interface. The main window shows a network map with purple lines and green nodes. A toolbar on the left contains various editing and analysis tools. The top right shows a map view with a scale of 0,20 km. The bottom right features a data table with columns for node name, type, distance, and coordinates. Below the table are several control panels for route calculation and display options.

Název uzlu	Typ uzlu	Poloha X	Poloha Y
v137.1	krížovka	644	1333
v137.2	krížovka	643	1339
v3.1	krížovka	859	1210
v3.2	krížovka	859	1216
v135.1	krížovka	865	1389
v135.2	krížovka	863	1383
v135.3	krížovka	845	1389
v135.4	krížovka	853	1402
v135.5	krížovka	865	1395
v135.6	krížovka	868	1383

Název uzlu	Typ uzlu	Příjezd	Náležet trasu	Penalizace
v232.2	krížovka	0	0000	Ano
v71.5	krížovka	0,24	0001	Ano
v71.4	krížovka	0,24	0001	Ano
v5.5	krížovka	0,31	0001	Ano
v5.6	krížovka	0,31	0001	Ano
v72.4	krížovka	0,47	0002	Ano
v72.5	krížovka	0,47	0002	Ano
v81.5	krížovka	0,55	0002	Ano

Zdroj: Autor

Příloha 9: Vývojový diagram metaheuristiky řešící tvorbu tras obslužných vozidel provádějících ČUPK



Zdroj: Autor

Příloha 10: Časová náročnost a výsledky poskytované algoritmy MCARP

náhodně generované instance		memetický algoritmus MA (metaheuristika)		Path Scanning PS (heuristika)		Augment Merge AM (heuristika)		Ulusoyova heuristika UH (heuristika)	
celková délka hran obsluhy [10 × m]	počet hran obsluhy [ks]	čas výpoč. [sec.]	délka trasy [10 × m]	čas výpoč. [sec.]	délka trasy [10 × m]	čas výpoč. [sec.]	délka trasy [10 × m]	čas výp. [sec.]	délka trasy [10 × m]
368	50	1	383	1	444	1	422	1	415
748	100	1	774	1	842	1	827	1	790
1057	150	2	1079	1	1166	2	1109	2	1107
1351	200	2	1430	1	1514	2	1452	2	1462
1724	250	3	1682	2	1784	2	1715	2	1709
2174	300	4	2199	2	2303	3	2216	3	2229
2391	350	5	2423	3	2530	3	2491	4	2463
2757	400	5	2789	3	2891	3	2851	4	2826
3115	450	6	3160	4	3308	4	3259	5	3182
3536	500*	8	3698	4	3888	5	3842	6	3787

Poznámka: Instance s počtem hran obsluhy 500 je zobrazena v příloze 12. Výpočet proběhl na PC s konfigurací – procesor Intel Core 2, 2 GHz, 4 GB RAM.

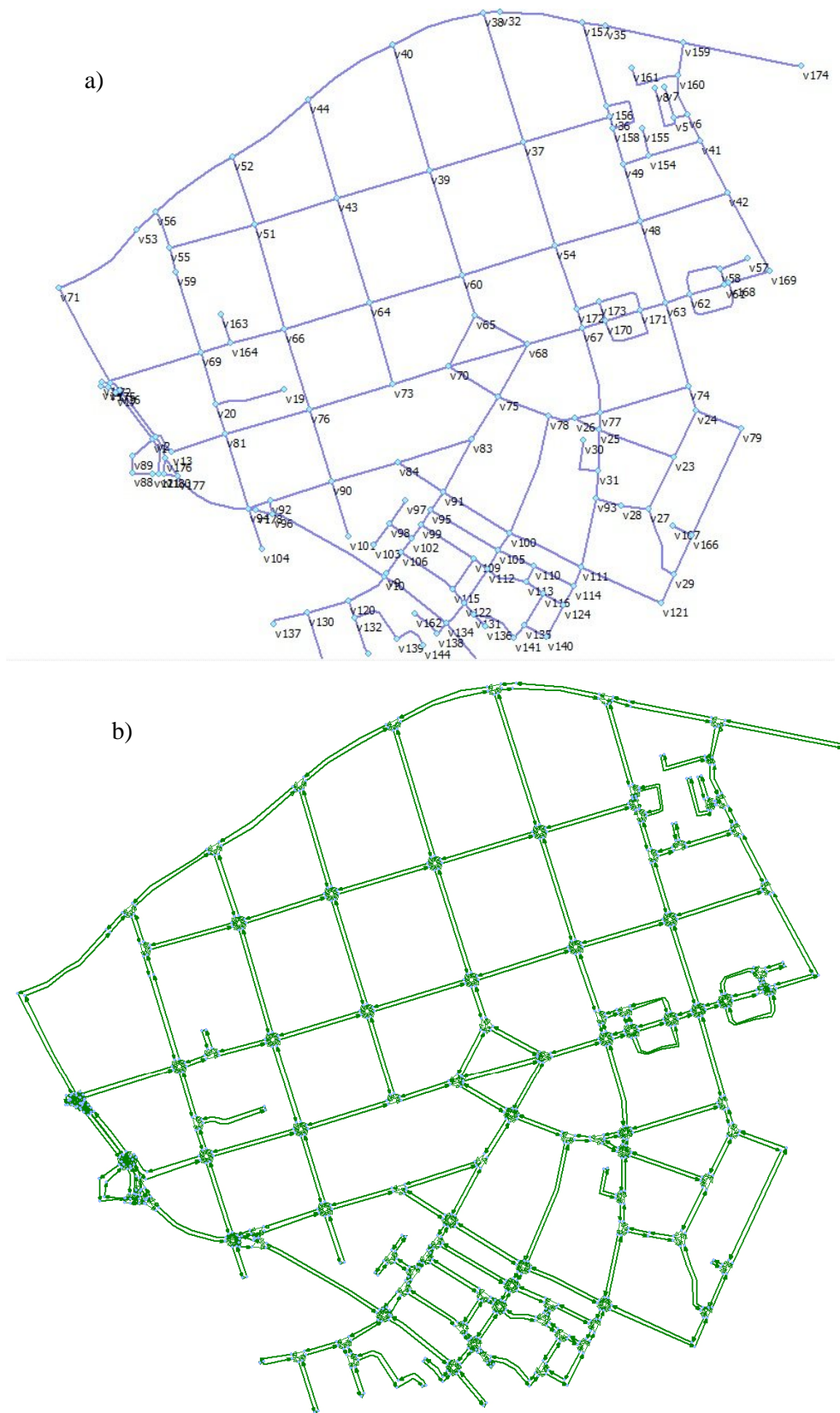
Zdroj: Autor (údaje stanovené výpočtem v rámci disertační práce – SW nástroj NetOpt)

Příloha 11: Jmenný seznam ulic určený pro řidiče obslužného vozidla Služeb města Pardubic a.s.

33. týden	
9.8 P	-Češkova od ul. Gorkého k ul. Kašparova - P strana -Jiránkova od ul. A Krause k ul. Wolkerova - P strana -A. Krause od ul. Čs. Armády k ul. Sokolovská - P strana
10.8. Ú	-Češkova od ul. Kašparova k ul. Gorkého - P strana -Jiránkova od ul. Wolkerova k ul. A Krause - P strana -A. Krause od ul. Sokolovská k ul. Čs. Armády - P strana
11.8. S	-Boženy Němcové od ul. Češkova k ul. Čs. Armády - O strany -Bocháčkova od ul. Češkova k ul. V Ráji - O strany
12.8. Č	-Češkova od ul. Teplého k ul. Milheimova - O strany -Češkova od ul. Demokratické mládeže k ul. Široká - O strany -Terezy Novákové od ul. Na Záboří k ul. Čacké - O strany
13.8. P	-Mikulovická od ul. Chrudimská k ul. Nemošická - P strana -úklid oblouků křižovatek dle potřeby

Zdroj: data Služeb města Pardubic a.s., rok 2010

Příloha 12: Náhodně vygenerovaná instance ARP v SW NetOpt – soubor komunikací Pardubice-Židov



Zdroj: Autor

Příloha 13: Současný stav sestavování plánů tras obslužných vozidel provádějících ČUPK v Pardubicích v aplikaci MS Excel, rok 2010

Optimalizace čištění v Pardubicích												čištění vozovek a chodníků				
Legenda :																
modře jsou označeny silnice I a II třídy, které jsou v majetku či správě SÚS																
červeně jsou označeny komunikace, které nelze strojně čistit, nebo nebyly nalezeny																
ve sloupci metry (m ²) je uvedena čistěná délka, tedy v řadě případů úkon na obou stranách vozovky																
ve sloupci četnost je uveden počet příslušného úkonu za rok																
ve sloupci m/rok je součin celkové uklizených metrů za rok																
Abeecední seznam ulic	Městská část	Strojní čištění vozovek	Σ SÚS	Str.čt. vozovek s odtahem	Splachování vozovek	Blokové čištění komunikací	Strojní čištění chodníků	Splachování chodníků								
		metry	četnost	m/rok	metry	četnost	m/rok	metry	četnost	m/rok	metry	četnost				
17. listopadu	Pardubice I	980	4	3 920	1 800	2	3 600	1 800	2	3 600	580	35	20 300	1 000	64	64 000
22. července	Pardubice III	440	0	0				400	1	400						
28. října	Pardubice IV	700	18	12 600												
5. května	Pardubice IV	1 830	18	32 940												
8. května	Pardubice VI	560	18	10 080												
Adolfův Opáček	Pardubice VII															
Arabská	Pardubice IV															
Alešova	Pardubice VII	250	18	4 500												
Anenská	Pardubice I	440	72	31 680							1 120	36	40 320			
Arbesova	Pardubice VII	500	18	9 000												
Arnošta z Pardubic	Pardubice I				480	2	960									
Artura Krause	Pardubice V	520	18	9 360	520	3	1 560									
Bacháčkova	Pardubice V	320	18	5 760												
Bartolomějská	Pardubice I															
Bartoňova	Pardubice III	1 070	18	19 260	1 070	1	1 070				340	36	12 240			
Bělehradská	Pardubice II	2 250	72	162 000							2 030	72	146 160			
Bělehradská	Pardubice VII	500	72	36 000							1 520	72	109 440			
IMK Bělehrad - Hradecká																
Bělobranská dubina	Pardubice III															
Bělobranské náměstí	Pardubice I															
Benediktino	Pardubice I	160	18	2 880	160	2	320									
Bezejmenná	Pardubice VII	500	18	9 000												
Benešovo náměstí	Pardubice V	480	18	8 640												
Bezručova	Pardubice I				892	2	1 784				320	36	11 520			
Bezručova	Pardubice III				500	2	1 000									
Blahoutova	Pardubice III	1 050	18	18 900							1 100	36	39 600			

Zdroj: data Služeb města Pardubic a.s., rok 2010