

Univerzita Pardubice  
Fakulta ekonomicko–správní

Umístění logistických center v území jako problém diskretní lokace

Bc. Magdaléna Burešová

Diplomová práce

2009

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Magdaléna BUREŠOVÁ**  
Studijní program: **N6209 Systémové inženýrství a informatika**  
Studijní obor: **Informatika ve veřejné správě**  
Název tématu: **Umístění logistických center v území jako problém  
diskrétní lokace**  
Zadávací katedra: **Ústav systémového inženýrství a informatiky**

### Z á s a d y p r o v y p r a c o v á n í :

Práce bude vycházet ze:

- a) Studia teoretických principů řešení problematiky v odborné literatuře
  - Studium bude zaměřeno na teorii lokační analýzy se zaměřením na řešení úloh diskrétní optimalizace lokace středisek v geografickém prostoru a na dokumentované výsledky řešení praktických úloh.
- b) Analýzy současného stavu řešení problematiky ve světě a v České republice
  - Analýza současného stavu řešení problematiky ve světě a v ČR bude zaměřena zejména na již existující a používané SW produkty, oblasti jejich využití a použitelnost těchto nástrojů v podmínkách ČR.
- c) Aplikace teorie lokační analýzy na řešení konkrétní úlohy
  - Vlastním přínosem diplomantky k řešení problematiky bude funkční interaktivní program umožňující automatický výpočet optimální lokace středisek obsluhy v území s alokací míst obsluhy do příslušných atrakčních obvodů.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

MARK S., D. Network and discrete location : models, algorithms, and applications. New York : John Wiley & Sons, c1995. 498 s.

PLESNÍK, J. Grafové algoritmy. Bratislava : Veda, 1983. 343 s.

NEŠETŘIL, J. Teorie grafů. Praha : Státní nakladatelství technické literatury, 1979. 316 s.

VOLEK, J. Operační výzkum I. Pardubice : Univerzita Pardubice, 2002. 111 s. ISBN 80-7194-410-6.

JABLONSKÝ, J. Operační výzkum. Praha : Professional Publishing, 2002.

DEMEL, J. Grafy a jejich aplikace. Praha : Academia, 2002.

CHRISTOFIDES, N. Graph Theory. New York : Academic Press, 1975.

LOLA (Library of Location Algorithms).


COYLE, John J., BARDI, Edward J., LANGLEY, John C. The management of business logistics. Saint Paul : West Publishing, 1988. 631 s. ISBN 0-314-65263-9.

Vedoucí diplomové práce:

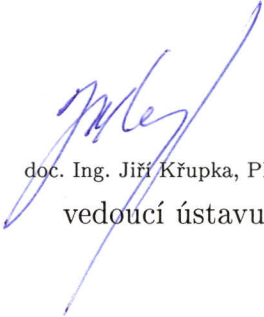
  
doc. Ing. Josef Volek, CSc.  
Katedra informatiky v dopravě

Datum zadání diplomové práce: **5. října 2009**

Termín odevzdání diplomové práce: **30. dubna 2010**

  
doc. Ing. Renáta Myšková, Ph.D.  
děkanka

L.S.

  
doc. Ing. Jiří Křupka, Ph.D.  
vedoucí ústavu

V Pardubicích dne 5. října 2009

Prohlašuji: Tuto práci jsem vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci použila, jsou uvedeny v seznamu literatury. Byla jsem seznámena a s tím, že se na diplomovou 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 této práce v Univerzitní knihovně.

V Pardubicích dne 30. 4. 2010

Bc. Magdaléna Burešová

**Anotace** Práce se zaměřuje na optimální lokaci logistického centra pomocí Hakimiho algoritmu. Tento algoritmus je v práci řešen manuálně a pak pomocí naprogramované aplikace v jazyku C++ s využitím knihovny Qt pro grafické uživatelské rozhraní. Práce se rovněž zabývá analýzou současného stavu řešení problematiky ve světě a v České republice.

**Klíčová slova** lokační problém, Hakimiho algoritmus, logistické centrum, lokace

**Title** Location of logistics centres in territory like problem of discrete location

**Annotation** The work focuses on the optimal location of the logistic center by Hakimiho algorithm. This algorithm is solved in the work manually and then by the help of the programming application C++ using the Qt library for graphical user interface. Solutions are based on the theory of locational study. This work also deals with the analysis of current solutions to the problems in the world and the Czech Republic.

**Keywords** location problem, Hakimi algorithm, logistic center, location

# Obsah

<b>1</b>	<b>Úvod</b>	<b>8</b>
<b>2</b>	<b>Logistika v lokačních problémech</b>	<b>9</b>
2.1	Hlavní cíle logistiky . . . . .	9
2.2	Vymezení pojmu logistické centrum . . . . .	10
2.3	Dělení dopravních sítí a přeprava zboží . . . . .	11
2.4	Rozšířenost dopravních sítí v České republice . . . . .	13
<b>3</b>	<b>Lokace obslužných středisek</b>	<b>14</b>
3.1	Historie lokačního problému . . . . .	14
3.2	Definice lokačního problému . . . . .	15
3.3	Vybrané metody řešení lokačního problému . . . . .	16
3.3.1	Torricelliho bod . . . . .	16
3.3.2	Varignonův rámeček . . . . .	17
3.3.3	Voroniovy diagramy . . . . .	18
3.4	Omezení lokačního problému . . . . .	18
3.5	Spojité pokrývající lokační problémy . . . . .	19
3.5.1	Vzdálenosti bodů . . . . .	19
3.5.2	Možnosti řešení spojitých lokačních problémů . . . . .	20
3.6	Diskrétní lokační problémy . . . . .	21
3.6.1	Základní modely diskrétní lokace . . . . .	21
3.6.2	P-mediánový problém . . . . .	22
<b>4</b>	<b>Lokační problém na síti</b>	<b>24</b>
4.1	Teorie grafů v lokačních úlohách . . . . .	24
4.1.1	Maticové vyjádření neorientovaných grafů . . . . .	25
4.1.2	Souvislost grafů . . . . .	26
4.1.3	Vysvětlení pojmů lokačních úloh pomocí teorie grafů . . . . .	27
4.1.4	Optimální umístění obslužných středisek v grafu . . . . .	28
4.2	Hakimiho algoritmus . . . . .	29
4.2.1	Postup Hakimiho algoritmu . . . . .	30
<b>5</b>	<b>Analýza současného stavu problematiky ve světě a v České republice</b>	<b>31</b>
5.1	Instituce zabývající se vývojem softwaru pro lokační teorii . . . . .	31

5.2	LoLA – Library of Location Algorithms . . . . .	32
5.2.1	LoLA v GIS . . . . .	33
5.3	INFORMS – Institute for Operations Research and the Management Sciences	34
5.3.1	SoLA – Section on Location Analysis . . . . .	36
5.4	EWGLA – EURO Working Group on Locational Analysis . . . . .	36
5.5	Řešení lokačních problémů v České republice . . . . .	37
5.5.1	Weiszfeldův algoritmus . . . . .	37
5.6	Způsoby alokace veřejných logistických center . . . . .	38
5.7	Indikátory pro místní udržitelný rozvoj . . . . .	39
<b>6</b>	<b>Lokace logistických center internetových obchodů</b>	<b>40</b>
6.1	Vymezení podmínek pro lokaci logistického centra . . . . .	41
6.2	Řešení lokace obslužného centra v síti . . . . .	42
6.3	Počítačová implementace Hakimiho algoritmu . . . . .	47
6.3.1	Knihovna Qt a programovací jazyk C++ . . . . .	47
6.3.2	Navrhnuté grafické prostředí . . . . .	49
6.3.3	Popis datových struktur aplikace . . . . .	53
<b>7</b>	<b>Závěr</b>	<b>58</b>
<b>8</b>	<b>Seznam zdrojů</b>	<b>60</b>
8.1	Seznam knižních zdrojů . . . . .	60
8.2	Seznam internetových zdrojů . . . . .	61

## Seznam obrázků

1	Torricelliho bod (zdroj: [4]) . . . . .	16
2	Torricelliho bod (zdroj: [4]) . . . . .	17
3	Varignonův rámeček (zdroj: [4]) . . . . .	17
4	Voronoiovy diagramy (zdroj: [autor]) . . . . .	18
5	Příklad komunikační sítě (zdroj: autor) . . . . .	25
6	Floydův diagram ve formě vývojového diagramu (zdroj:[13]) . . . . .	27
7	Vážená excentricita bodu $y$ hrany $(v_i, v_j)$ (zdroj: [15]) . . . . .	30
8	Spojení GIS a LoLA (zdroj: [4]) . . . . .	34
9	Síť použitá pro řešení lokace (zdroj: autor) . . . . .	42
10	Diagram hodnot pro hranu $h_4$ (zdroj: autor) . . . . .	44
11	Diagram hodnot pro hranu $h_5$ (zdroj: autor) . . . . .	45
12	Diagram hodnot pro hranu $h_6$ (zdroj: autor) . . . . .	46
13	Umístění absolutního depa v síti (zdroj: autor) . . . . .	46
14	Grafické uživatelské rozhraní vytvořené aplikace . . . . .	49
15	Okno zobrazující seznam vrcholů a hran (zdroj: autor) . . . . .	51
16	Okno pro vložení distanční matice (zdroj: autor) . . . . .	51
17	Diagram hodnot pro hranu 4 (zdroj: autor) . . . . .	52
18	Diagram hodnot pro hranu 5 (zdroj: autor) . . . . .	52
19	Diagram hodnot pro hranu 6 (zdroj: autor) . . . . .	53
20	Znázornění naprogramované datové struktury (zdroj: autor) . . . . .	56
21	Vzájemné vztahy mezi jednotlivými třídami (zdroj: autor) . . . . .	58

## Seznam tabulek

1	Přeprava zboží u jednotlivých druhů dopravy (zdroj: [20]) . . . . .	12
2	Infrastruktura jednotlivých druhů dopravy (zdroj: [16]) . . . . .	13
3	Klasifikační schéma pro lokační problém (zdroj: [4]) . . . . .	33
4	Matice vzdáleností (zdroj: autor) . . . . .	42
5	Výpočet horního ohraničení vážené excentricity absolutního depa (zdroj: autor) . . . . .	43



# 1 Úvod

V poslední době je patrná snaha řešit přepravní řetězce komplexně. Logistický přístup umožňuje optimalizovat řešení přepravního procesu jako celku. Logistické systémy řízení oběhu materiálu, jeho skladování, balení, překládání, distribuce a přepravy nemohou být realizovány bez stabilně fungujícího přepravního systému. Přeprava v takovém systému musí být integrujícím prvkem logistických systémů.

Logistická centra se zřizují v posledních letech zejména z důvodu zefektivnění a urychlení přepravy zboží od výrobce k zákazníkovi. Požadavky na přepravu musí splňovat určitý cíl, ať už to je minimalizace celkových nákladů nebo minimalizace času na přepravu. Proto se klade důraz na výběr vhodné lokality pro umístění obslužného zařízení. Pomocí lokačních úloh se hledají optimální polohy logistických center nejen v České republice ale i ve světě. Obecně je umísťováno několik obslužných center stejného typu a tato střediska budou využívána skupinou zákazníků rovněž stejného typu.

Cílem práce je vytvořit aplikaci, která řeší vybraný lokační problém. Vytvořená aplikace značně zefektivňuje řešení vybraného lokačního problému. Dalším cílem práce je analyzovat současný stav řešení problematiky ve světě a v České republice bude zaměřena zejména na již existující a používané metody, oblasti jejich využití a použitelnost těchto nástrojů v podmínkách České republiky.

## 2 Logistika v lokačních problémech

V současnosti je řešení lokačního problému předmětem výzkumu teoretických úvah i běžné manažerské praxe. V rozsáhlé oblasti s velkým počtem zákazníků není vhodné zásobování zákazníků přímo od výrobce nebo z jediného centrálního skladu. Efektivnější je vytvořit síť distribučních skladů, jejíž existence může snížit náklady na přepravu a čas potřebný k přepravě zboží nebo produktů. Cílem je najít cenový kompromis, který redukuje dopravní náklady (od výrobce přes distribuční sklady až k zákazníkovi) a fixní náklady spojené s činností skladů a udržováním zásob. [9]

### 2.1 Hlavní cíle logistiky

V silné konkurenci může být úspěšný jen ten výrobce, který dokáže uspokojovat čím dál náročnější potřeby zákazníků a poskytovat jim kvalitní nabídku zboží a služeb. Výrobce se musí postarat o to, aby jeho zboží bylo k dispozici ve správném množství, na správném místě, ve správném čase a to všechno s minimálními náklady. To znamená, že nabídka produktů musí být pružná a včasná, aby co nejlépe a nejrychleji uspokojila potřebu zákazníka.

K dosažení vysoké pružnosti je zapotřebí dostatečné technické vybavení s dokonalým řízením výrobních a zejména oběhových procesů, což je přirozeně nákladné. Obecně platí, že čím vyspělejší je země, tím větší podíl z ceny hotových výrobků tvoří náklady na oběh. Oběhem rozumíme souhrn všech předvýrobních a povýrobních hmotných procesů, jimiž prochází suroviny, materiál, díly, nedokončené a hotové výrobky určené k prodeji. Jedná se o procesy netechnické povahy složené z operací balení, tvorby manipulačních a přepravních operací, skladování, kontroly a další. Dále platí, že čím pomalejší je oběh, tím méně efektivním se stává. Proto je nutno oběh zrychlit, zpružnit a racionalizovat. Je potřeba řešit a koordinovat veškeré hmotné i nehmotné operace v rámci výrobních a oběhových procesů.

Tyto operace jsou vzájemně propojeny do řetězců, jejichž počátkem je zjištění potřeby zákazníka po určitém zboží a koncem je dodání tohoto zboží zákazníkovi. To vše s maximální pružností a hospodárností. Soubor těchto řetězců se označuje jako logistické řetězce. Disciplína, která vysvětluje zákonitosti působící při vytváření a dává návody k řízení těchto řetězců se nazývá logistika.

Logistika se někdy vysvětluje jako metoda řízení. Zabývá se pohybem zdrojů přes jejich zpracování a vytváření finálních výrobků včetně distribuce až ke koncovým zákazník-

kům z hlediska dopravy, zásobování, služeb, skladování, manipulace, balení, projektování a rozmísťování kapacit. [8]

## 2.2 Vymezení pojmu logistické centrum

Logistická centra se začala vytvářet z důvodu zefektivnění a urychlení vzájemných vztahů mezi dopravou a ostatními dílčími systémy. Logistické centrum je komplex zařízení v místě alespoň se dvěma druhy dopravy (např. silniční a železniční), v němž se poskytuje široké spektrum logistických služeb, jako jsou přepravní, skladovací, manipulační, třídící, celní, pojišťovací, informační, servisní, bezpečnostní a další služby.

Účelem logistických center je zvýšit kvalitu a efektivnost přepravy a služeb a organizovanost toku zboží. Tyto střediska chtějí maximálně využívat přednosti jednotlivých druhů doprav, vzájemně je provázet a optimalizovat.

Přírozenými lokalitami pro logistická centra jsou velké hospodářské aglomerace, dopravní uzly, přístavy, hraniční přechody atd., kde jsou předpoklady pro dostatečnou koncentraci materiálových toků.

Logistická centra mohou být klasifikována dle jejich účelu:

- Firemní logistická centra (centra slouží potřebám velké firmy nebo obchodního řetězce).
- Logistická centra logistických firem (střediska jsou provozovaná poskytovateli logistických služeb, která slouží vybraným zákazníkům).
- Logistické areály (zařízení poskytuje služby více logistických firem).
- Logistická centra sítě poskytovatelů kurýrních, expresních a balíkových služeb.
- Logistická centra internetových obchodů.

Logistická centra tvoří jeden z článků tzv. logistických řetězců tj. souboru hmotných a nehmotných toků, které probíhají v posloupnosti určitých navazujících článků, jejichž struktura je odvozena od potřeb konečného zákazníka.

Kritéria pro konkrétní umístění logistického centra jsou zejména:

- napojení logistického centra na dopravní síť,
- vzdálenost a časová dostupnost zákazníkům,
- majetkové a právní vztahy k pozemkům,

- vliv na životní prostředí,
- kapacita logistických center,
- velikost zastavěné plochy logistického centra a možnost dalšího rozšíření,
- a jiné okolnosti. [21]

## 2.3 Dělení dopravních sítí a přeprava zboží

Následující kapitola popisuje jednotlivé druhy dopravy a dále se zabývá velikostí přepravy zboží po těchto dopravních sítích. Logistické centrum je efektivněji využíváno, pokud je napojené na více druhů dopravních sítí.

Obslužnost dopravní sítě logistickým centrem závisí na dopravních prostředcích, které budou obsluhu vybraného území obstarávat. Tato kapitola se soustřeďuje na základní třídění dopravních prostředků. Ty bývají zpravidla děleny na:

- silniční,
- železniční,
- vodní,
- letecké a
- potrubní.

Silniční doprava se vyznačuje velkou dynamikou rozvoje a má značné výhody ve srovnání s ostatními typy dopravy. Silniční komunikace jsou v rámci Evropy nejrozšířenější. Výhodou tohoto typu dopravy je zajištění přepravy zboží "z domu do domu". Hlavním znakem silniční dopravy je vyšší rychlost a větší pohotovost. Má však i některé nevýhody. Rychlost této přepravy negativně ovlivňuje nehodovost, počasí, kvalita povrchu jednotlivých typů silnic a dálnic, ale i hmotnostní limity. Nutné je zmínit i negativní dopady na životní prostředí (hluk, výfukové plyny, otřesy aj.). Využitelná je zejména na střední a kratší vzdálenosti.

Železniční doprava se od roku 1822, kdy byla postavena první trať v Anglii, rozrostla a mnohokrát se modernizovala. Po druhé světové válce se délka železnic začala zkracovat a jejich význam se ve vyspělých státech světa snižuje. Způsobil to mimořádně prudký rozvoj silniční dopravy. Železniční doprava má mnoho předností, které jí za jistých okolností

zvýhodňují proti jiným druhům dopravy. K výhodám železniční dopravy ve srovnání se silniční dopravou patří nižší náročnost na spotřebu energie, nižší náklady a zejména větší šetrnost k životnímu prostředí. Železnice jsou v celosvětovém měřítku využívány především na přepravu nákladů na střední a dlouhé vzdálenosti. Nejvíce se přepravují těžké a objemné náklady (např. obilí, rudy, pevná paliva, dřevo, stavební materiál nebo umělá hnojiva).

Vodní doprava se zaměřuje na přepravu objemných nákladů na velké vzdálenosti. Lodě převážejí ropu, uhlí, obilí, ocel, dřevo a umělá hnojiva. Přeprava po vodě je levnější a ekologicky šetrnější než doprava po silnici nebo po železnici. Je však pomalejší, takže se používá pro přepravu nákladů, u nichž rychlost nehraje podstatnou úlohu.

Letecká doprava se řadí mezi nejrychlejší typ dopravy. Uplatňuje se především při přepravě osob na velké a střední vzdálenosti. Tento typ dopravy se rovněž využívá k přepravě zboží podléhajícím rychlé zkáze (květiny, ovoce a zelenina aj.). Nevýhodou letecké dopravy je omezení přepravy velmi těžkých a objemných nákladů.

Potrubní doprava má nízké dopravní náklady a malý negativní vliv na životní prostředí. Využívá se na přepravu zejména tekutých a plyných látek na velké vzdálenosti (ropa, zemní plyn). [2]

Následující tabulka č. 1 ukazuje celkovou přepravu zboží od roku 1995 do roku 2008 v České republice. Hmotnosti jsou uvedeny v tisících tun za jednotlivé roky.

Tabulka 1: Přeprava zboží u jednotlivých druhů dopravy (zdroj: [20])

Rok	Silniční	Železniční	Vodní	Letecká	Potrubní
1995	X	108 871	4 441	X	7 083
1996	X	107 235	3 252	X	8 992
1997	521 481	111 379	1 831	X	9 217
1998	470 888	104 788	1 856	X	9 217
1999	448 300	90 734	1 890	X	8 050
2000	414 725	98 255	1 907	X	8 346
2001	438 683	97 218	1 910	X	8 674
2002	474 884	91 989	1 686	44,126	8 815
2003	447 956	93 297	1 277	53,485	8 962
2004	466 034	88 843	1 275	58,866	9 192
2005	461 144	85 612	1 956	56,85	11 305
2006	444 574	97 491	2 032	60,475	10 875
2007	453 537	99 777	2 242	61,208	10 131
2008	431 855	95 074	1 905	56,392	11 877

Speciálním druhem dopravy je doprava zpráv, informací a dat (tzv. spoje). Právě toto odvětví je klíčové pro rozvoj moderní společnosti. Je jedním z hlavních projevů i nástrojů současné globalizace světa. Základní složky tohoto druhu dopravy jsou doprava listovních zásilek, telefon, rozhlas a televize. Rychlé a kvalitní spojení umožňují telekomunikační družice. Progresivní vývoj informační techniky a technologie výrazně přispěl k rozvoji a využívání Internetu. [2]

## 2.4 Rozšířenost dopravních sítí v České republice

Následující kapitola je soustředěna na rozšířenost jednotlivých dopravních sítí v České republice. Podle rozšíření dopravních sítí lze stanovit, jaká bude nejoptimálnější kombinace dopravy napojené na logistické centrum.

V následující tabulce č. 2 jsou uvedeny délky jednotlivých dopravních sítí (v kilometrech) na území České republiky od roku 1995 do roku 2008. Délka silničních komunikací obsahuje dálnice, silnice různých tříd a evropské silniční sítě typu E. Délka sítě železničních tratí zahrnuje elektrifikované a neelektrifikované tratě. Síť vodní dopravy zahrnuje splavné řeky, jezera a uměle vytvořených kanály, včetně cest v nádržích a jezerech sloužících převážně k rekreační osobní dopravě a sportovní plavbě. Ve sloupci letecké dopravy, číslo udává celkový počet letišť v kombinaci veřejných, neveřejných, vnitrostátních a mezinárodních.

Tabulka 2: Infrastruktura jednotlivých druhů dopravy (zdroj: [16])

Rok	Silniční	Železniční	Vodní	Letecká	Potrubiční
1995	55 500	9 430	677	73	567
1996	55 511	9 430	677	67	747
1997	55 394	9 430	677	74	747
1998	55 394	9 430	664	85	747
1999	55 432	9 444	664	84	747
2000	55 408	9 444	664	86	675
2001	55 427	9 523	664	85	675
2002	55 422	9 600	664	85	675
2003	55 447	9 602	664	87	675
2004	55 500	9 612	664	87	675
2005	55 510	9 614	664	88	675
2006	55 585	9 597	664	89	675
2007	55 584	9 588	664	91	675
2008	55 654	9 586	664	91	675

## 3 Lokace obslužných středisek

Lokace obslužných center (výrobní závody, distribuční sklady, nákupní centra apod.) ovlivňuje materiálové toky ve stále rozsáhlejších dodavatelských řetězcích. Umístění těchto zařízení má vliv na náklady spojené s realizací materiálových toků.

Nutnost řešení lokace obslužného zařízení v území se poprvé objevuje již v 17. století. Lokační problém nabízí velký počet variací. Pro všechny varianty lokačního problému je potřeba znát základní data:

- kolik zařízení obsluhy je potřeba rozmístit,
- kde by měla tato zařízení být umístěná,
- jakou kapacitu by měla mít,
- které obslužné body by měly být k zařízení přiřazené,
- jakou kapacitu by měla mít cesta mezi obslužným centrem a zákazníkem,
- jaká by měla být kapacita obslužných center. [4]

Je složité vyvinout modely, které by odpověděly na všechny předcházející otázky současně. O tomto problému psalo mnoho odborníků z nejrůznějších vědních oborů. Do lokační analýzy patří například problémy: Fermatův problém, Steinerův problém, Torricelliho problém, Weberův problém, problém mediánu, problém centrálního mediánu, problém minimálního součtu atd.

Lokační teorie má tři hlavní směry:

- spojitý lokační problém,
- diskrétní lokační problém,
- lokační problém na síti. [7]

### 3.1 Historie lokačního problému

Následující kapitola se zabývá stručným výtahem historie problému hledající územní medián. (bod s minimálním součtem Euklidovské vzdálenosti). W. Kuhn (1967) navrhl výborný model tohoto problému. Jedním z jeho zdrojů byl článek od M. Zachariase (1913).

Mezi další zdroje, které poskytly detaily jeho řešení byly Pottage (1983), Honsberger (1973) a Dörrie (1965).

Kdo ale první poskytl řešení problému a v jaké formě, nebude pravděpodobně nikdy známo. Obvykle je tento problém připisován Pierru de Fermatovi (1601-1665), který navrhl základní formu problému s územním mediánem. Italský matematik Galileo, Evangelista Torricelli (1608-1647) našel údajně řešení problému, kde jsou na ploše umístěné tři body a hledá se lokace čtvrtého bodu tak, aby součet vzdáleností od daných bodů byl minimální. Ne všichni připisují Fermatovi řešení onoho problému. Jiní řešení tohoto problému přisuzují Battistovi Cavalierovi (1598-1647).

Do dvacátého století různí matematici popisovali jiné varianty tohoto problému. Mezi nejvýznamnější patří Thomas Simpson (1710-1761) a Jacob Steiner (1796-1863). V roce 1909 použil Alfred Weber (1868–1958) váženou verzi problému o třech bodech, která zobrazovala lokaci čtvrtého bodu minimalizující cenu dopravy (součet vzdáleností od daných bodů).

Ve dvacátém století bylo publikováno mnoho metod a postupů, které rozšiřovaly již z minulosti známé modely. Mezi matematiky tehdejší doby se řadí maďarský matematik Endre Vaszonyi Weiszfeld (1936), Miehle (1958), Kuhna a Kuenne (1962), Juel a Love (1986), Ostresh (1978) a mnoho dalších. [4]

### 3.2 Definice lokačního problému

Řešením lokačního problému je bod  $(x^*, y^*)$ , který minimalizuje součet vážených euklidovských vzdáleností mezi tímto bodem a  $n$  pevných bodů se souřadnicemi  $(a_i, b_i)$ . Váhy, které jsou k pevným bodům přiřazeny, se označují  $w_i$ . Bod  $(x^*, y^*)$  je umístěné obslužné středisko tak, že minimalizuje vzdálenost ke všem pevným bodům  $(a_i, b_i)$  (tzv. poptávkové body).

Primární lokační problém je definovaný následovně:

$$\min_{x,y} \left\{ W(x, y) = \sum_{i=1}^n w_i d_i(x, y) \right\},$$

kde  $d_i(x, y) = \sqrt{(x - a_i)^2 + (y - b_i)^2}$  je euklidovská vzdálenost mezi body  $(x^*, y^*)$  a  $(a_i, b_i)$ . [4]



### 3.3 Vybrané metody řešení lokačního problému

Tato kapitola se zaměřuje na některá řešení lokačních problémů.

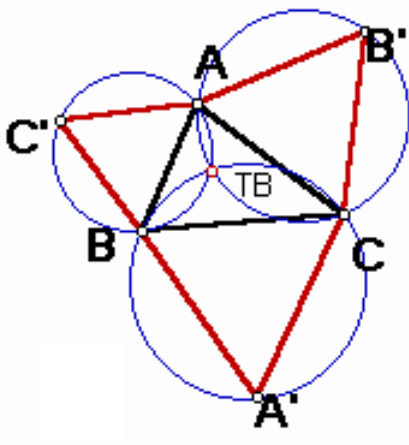
#### 3.3.1 Torricelliho bod

Grafické řešení problému pro  $n = 3$  jako první navrhl Evangelista Torricelli (1608-1647). Tři body vytváří trojúhelník  $\triangle ABC$ . Řešení spočívá v následujícím postupu:

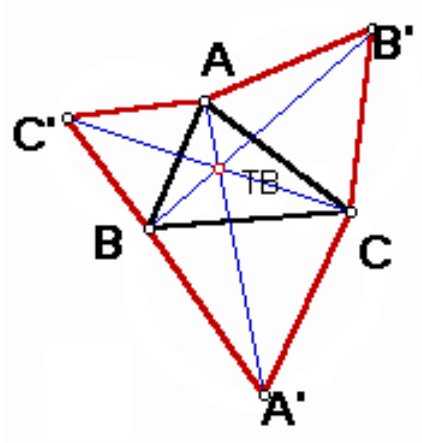
- Nad každou stranou trojúhelníka se vytyčí rovnostranný trojúhelník; trojúhelník nad stranou  $AB$  se označí  $ABC'$ , nad stranou  $AC$  se označí  $ACB'$ , a nad stranou  $BC$  se označí  $BCA'$  (viz obrázek číslo 1).
- V každém trojúhelníku se určí střed kružnice opsané a kružnici se sestrojí.
- Tři opsané kružnice se protínají v bodě, který se označuje jako TB (Torricelliho bod) nebo F (Fermatův bod).

Torricelliho bod minimalizuje součet délek úseček  $FA + FB + FC$  a při stejné váze všech vrcholů trojúhelníku  $w_A = w_B = w_C = 1$  je řešením lokační úlohy.

Alternativní postup vyhledání Fermatova bodu je konstrukce přímků spojující body  $C'$  s  $C$ ,  $B'$  s  $B$  a  $A'$  s  $A$ . Přímků se protínají v bodu F (viz obrázek číslo 2). Bod F je též označován jako první isogonický střed trojúhelníka. Označení vychází ze skutečnosti, že úseky  $AF$ ,  $BF$  a  $CF$  svírají vzájemně shodný úhel  $120^\circ$ . Označení isogonický pochází z řečtiny a má kořeny v označení stejného úhlu iso a gon. [4]



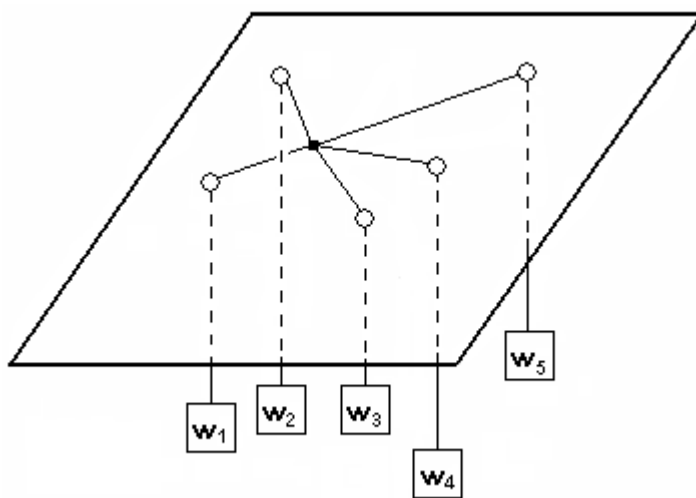
Obrázek 1: Torricelliho bod (zdroj: [4])



Obrázek 2: Torricelliho bod (zdroj: [4])

### 3.3.2 Varignonův rámeček

Varignon navrhnul mechanický analogový model, jehož schéma je znázorněno na obrázku číslo 3, který je nazýván Varignonův rámeček. Deska s  $n$  otvory odpovídá  $n$  bodům obsluhy. Každým otvorem prochází nit, na jejímž volném konci visí závaží s vahou (hmotností)  $w_i, i = 1, \dots, n$ . Opačné konce nití jsou svázaný v uzlu, jehož souřadnice po ustálení jsou  $(x^*, y^*)$  a představují optimální umístění střediska obsluhy. Minimalizují funkci  $W$ , která je popsána v kapitole 3.2, primární lokační problém. [4]

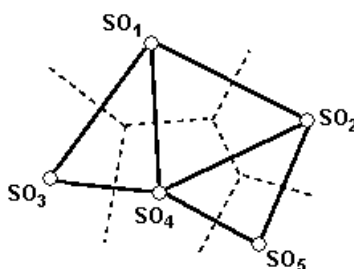


Obrázek 3: Varignonův rámeček (zdroj: [4])

### 3.3.3 Voronoiovy diagramy

Voronoiovy diagramy rozdělí obsluhovaný prostor mezi sousední střediska na dvě poloviny tak, aby zákazníci v každé polovině byli obsluhováni vždy z nejbližšího střediska. Existuje několik algoritmů, které rozdělí dané území mezi sousední střediska.

Jeden z možných přístupů je takový, že hranicí mezi dvěma polovinami je přímka, která prochází středem úsečky spojující obě sousední střediska  $SO_i$  a která je kolmá k této spojnici (základní úloha předpokládá euklidovské vzdálenosti). Tvar obsluhovaného prostoru je konvexní mnohoúhelník. Dělením každého konvexního mnohoúhelníku vznikají dva nové konvexní mnohoúhelníky (viz obrázek číslo 4). [4]



Obrázek 4: Voronoiovy diagramy (zdroj: [autor])

## 3.4 Omezení lokačního problému

Jelikož je základní lokační problém založený na geometrické reprezentaci, museli autoři lokačních modelů při začleňování do reálného území počítat s jistými omezeními. Typickým příkladem jsou zakázané oblasti.

V těchto regionech není lokace střediska dovolena. Příkladem jsou národní parky nebo jiné chráněné oblasti, hustě obydlené regiony, armádní oblasti, horské louky, jezera, nebo velké řeky. Lokační problém pro zakázané regiony je formulovaný:

$$\min_{(x,y) \in T} \left\{ B(x,y) = \sum_{i=1}^n w_i d_i(x,y) \right\}.$$

Zařízení obsluhy se umísťuje vně tyto oblasti a vzdálenost  $d_i(x,y)$  je definována jako nejkratší cesta mezi depem a poptávkovým bodem  $i$  a nesmí procházet tyto zakázané oblasti. Centrum obsluhy se umísťuje do vhodného regionu  $T$ . Počet požadavků v poptávkových bodech je  $w_i$ . [4]

### 3.5 Spojité pokrývající lokační problémy

Spojité lokační problém je popsán prostřednictvím spojitých veličin, obvykle souřadnicových. Daný prostor může být popsán dvěma souřadnicemi (tzn. že je rovinný). V případě, že je nezbytné vzít do úvahy výšku nebo hloubku (např. uvnitř budovy, pod vodou nebo ve vzduchu), je potřeba třetí souřadnice a lokační prostor je trojrozměrný.

Základní otázkou pro lokaci obslužných středisek je, kde se budou tato střediska umisťovat. Pro toto umístění je potřeba znát, jaká místa jsou k dispozici a na základě čeho se tato místa vybírají. Ve spojitých lokačních problémech se mohou obslužná zařízení umisťovat kdekoli v neobsazených místech (ve vrcholu, na hranách sítě i mimo síť). Středisko má svoji polohu popsanou souřadnicemi. Pro minimalizaci určité ceny, poškození či nepohodlí anebo pro maximalizaci zisku, kvality nebo duševní pohody je potřeba vhodně volit místo pro obslužné středisko. Tyto modely jsou označovány jako optimalizační.

Regiony, kde má být umístěné obslužné středisko, se popisují nepřiliš komplikovanou formou. Daný geografický prostor je většinou popisovaný jako polygon. Hranice tohoto území je sled rohových bodů spojené úsečkami (taková data jsou typická při používání GIS). Komplikovanější regiony vznikají v případě technických, ekonomických nebo politických omezení. V takovém případě se sestavují diskrétní modely popsané v kapitole 3.6. [4, 7]

#### 3.5.1 Vzdálenosti bodů

Základní vlastnost vzájemné polohy je vzdálenost mezi body (místy). Vzdálenost je matematický popis představy blízkosti, a proto může mít mnoho forem v závislosti na použití. Například v horské oblasti je vzdušná vzdálenost mezi dvěma vrcholy podstatně kratší než je skutečná vzdálenost po turistické trase. V dopravě se mluví o tzv. deviatilitě, což je odchylka dopravní cesty od přímé (ortodromické) vzdálenosti.

Znalost polohy dvou bodů (díky jejich souřadnicím) je nedostačující k vypočítání jejich vzdáleností. Je potřeba znát typ vzdálenostní míry. Těchto metrik existuje velmi mnoho, z nichž nejznámější je euklidovská vzdálenost vyměřující distanci mezi dvěma body. Jestliže jsou dané dva body  $X = (x, y)$  a  $P = (a, b)$  v ortonormálním souřadnicovém systému (tzn. osy  $x$  a  $y$  jsou na sebe kolmé a na osách se používá stejné měřítko), euklidovská vzdálenost je popsána:

$$d^{euc}(X, P) = \sqrt{(a - x)^2 + (b - y)^2}.$$

Dalším typem metriky je manhattanská vzdálenost. Vzdálenost mezi dvěma body  $X = (x, y)$  a  $P = (a, b)$  je dána součtem z absolutních rozdílů jednotlivých souřadnic. Manhattanská metrika je definována:

$$d^{rec}(X, P) = |a - x| + |b - y|.$$

Čebyševova vzdálenost mezi dvěma body  $X = (x, y)$  a  $P = (a, b)$  je dána vztahem:

$$d^{\max}(X, P) = \max\{|a - x|, |b - y|\}. [4]$$

### 3.5.2 Možnosti řešení spojitých lokačních problémů

Tato kapitola popisuje několik variant základního modelu spojitě lokace pro obslužná zařízení. Všechny modely jsou založené na představě pokrytí. Obslužné centra mají definovaný určitý akční rádius nebo určitou prahovou vzdálenost. Pokud je poptávkový bod uvnitř této definované oblasti, je považován za pokrytý obslužným střediskem. Takto pokrytý poptávkový bod je obsluhován příslušným obslužným zařízením.

Dané území se může pokrývat čtyřmi různými způsoby:

- Při *plném pokrytí* se usiluje o umístění zařízení (např. akustická siréna) tak, aby toto zařízení bylo použitelné pro největší okruh poptávkových bodů. Pro optimální řešení se využívá euklidovská vzdálenost a také se bere v úvahu, že přijímací kvalita se neustále snižuje se vzdáleností. Usiluje se o umístění zařízení s největším možným akčním rádiusem, který pokrývá všechny požadované body.
- Při *maximálním pokrytí* se využívá pevně daný rádius a je požadováno pokrytí co nejvíce poptávkových bodů.
- Při *prázdném pokrytí* se usiluje o umístění obslužného zařízení (např. sklad výbušnin, fotbalový stadion) s největším možným rádiusem tak, aby žádný z požadovaných bodů nebyl pokrytý.
- Při *minimální pokrytí* se snaží o lokaci zařízení s pevně daným akčním rádiusem tak, aby pokrýval co nejméně poptávkových bodů. [4]

## 3.6 Diskrétní lokační problémy

Diskrétní lokační modely se velmi často uplatňují v praktických případech. Zahrnují konečný soubor míst, ve kterých může být obslužné zařízení umístěno, a konečný soubor klientů, jejichž požadavky musí být splněné od přiděleného obslužného zařízení.

Do diskrétní lokace se zahrnují entity, jako jsou letiště, pobřežní přístavy, továrny, sklady, maloobchodní prodejny, školy, nemocnice, centra péče o děti, autobusové zastávky, podzemní stanice, terminály, záchranné zdravotnické služby a jiné. Rozhodování o umístění jednotlivých zařízení vedlo k zájmu o lokační problémy.

Za diskrétní problém se považují situace, které musí řešit umístění:

- obslužných zařízení, které jsou ve všech úrovních lidské organizace, od jedinců přes domácnosti k firmám, vládním agenturám a dokonce i mezinárodním úřadům;
- obslužných středisek, které zahrnují ekonomické externality, jako je znečištění, dopravní zácpy, ekonomický rozvoj regionu, zaměstnanost apod.;
- obslužného centra v území, které je velmi obtížné řešit pro jeho matematickou složitost (tyto modely se začaly řešit až s příchodem vysokorychlostních počítačů);
- obslužných zařízení, které jsou specifické pro jeho danou aplikaci (cíle, omezení).

V důsledku toho neexistuje obecný lokační model, který je vhodný pro všechny stávající aplikace.

Diskrétní síťové lokační modely přitahují pozornost výzkumníků a praktiků kvůli:

- praktickému významu těchto modelů v soukromém i veřejném sektoru,
- stále přítomné potřebě včlenit do modelu více omezení a cílů,
- schopnosti formulací důležitých rozhodovacích problémů. [4, 7]

### 3.6.1 Základní modely diskrétní lokace

Tato kapitola se zabývá rozdělením základních modelů diskrétní lokace. Ve všech těchto modelech je hlavním problémem umístění nového zařízení tak, aby co nejlépe optimalizoval zadané cíle.

Tato kapitola popisuje všech osm základních modelů diskrétního problému. Ve všech těchto modelech je daná poloha bodů, které mají být obsluhovány, a umístění obslužných středisek. První čtyři modely jsou založena na maximální vzdálenosti a zbývající modely na úplné (nebo průměrné) vzdálenosti. Mezi základní modely patří:

- stanovující pokrývající lokační model (cílem tohoto modelu je umístit minimální množství zařízení potřebných k pokrytí všech požadovaných uzlů);
- lokační model maximálního pokrytí (platí za předpokladu, že všechny požadované uzly jsou pokryté; cílem tohoto problému je lokalizovat předem dané množství zařízení  $p$  takovým způsobem, aby maximalizoval požadovanou potřebu tzn. pokryl většinu požadovaných uzlů);
- $p$ -centrální model (tento model se snaží minimalizovat maximální vzdálenosti tím, že je známa velikost požadavku od poptávkových bodů a je známa poloha obslužných zařízení);
- $p$ -vzdálenostní model (tento model maximalizuje minimální vzdálenost mezi libovolnou dvojicí obslužných středisek);
- $p$ -mediánový problém (cílem tohoto problému je najít místo pro  $p$  obslužných zařízení tak, aby se minimalizovala určitá míra dopravních nákladů mezi depem a poptávkovými body);
- fixní cena lokačního problému (cílem tohoto problému je minimalizovat počet obslužných center a dopravní náklady; poptávkový bod se vždy nepřirazuje k tomu nejbližšímu obslužnému centru);
- hub lokační problém (tento model je navržen k tomu, aby poskytoval větší kapacitu a rychlejší přepravu k cílovému bodu; následně tento model snižuje průměrnou dopravní cenu za jeden kilometr nebo čas potřebný k přepravě);
- problém maximálního součtu (tento model se snaží umístit nežádoucí zařízení tak, aby vzdálenost mezi poptávkovými body s přidělenými obslužnými středisky, byla co největší). [4]

### 3.6.2 P-mediánový problém

Jeden ze základních modelů v oblasti diskrétní lokace je  $p$ -mediánový model. Nalézá umístění  $p$  obslužných zařízení tak, že minimalizuje určitou míru dopravních nákladů (např. vzdálenost, cestovní doba) mezi umístěnými obslužnými středisky a poptávkovými body, které jsou přiřazeny k určitému obslužnému zařízení. Poptávkové body se přidělují k nejbližšímu obslužnému centru. Problém je používaný ve veřejných i soukromých sektorech. Nejprve se definují následující vstupy a množiny:

$I$  = množina poptávkových uzlů, označovány  $i$ ,

$J$  = množina umístěných obslužných zařízení, označovány  $j$ ,

$d_{ij}$  = vzdálenost mezi poptávkovým uzlem  $i$  a obslužným střediskem  $j$ ,

$h_i$  = požadavek v uzlu  $i$ ,

$p$  = počet zařízení k umístění,

$x_j = \begin{cases} 1, & \text{jestli se umísťuje obslužné středisko v místě } j, \\ 0, & \text{jestli se obslužné centrum neumísťuje.} \end{cases}$

$y_{ij} = \begin{cases} 1, & \text{když požadovaný uzel } i \text{ je přiřazený k zařízení v uzlu } j, \\ 0, & \text{když není.} \end{cases}$

Model je formulovaný následovně:

$$\min \sum_{i \in I} \sum_{j \in J} h_i d_{ij} y_{ij},$$

za předpokladů:

$$\sum_{j \in J} x_j = p,$$

$$\sum_{j \in J} y_{ij} = 1 \quad \forall i \in I,$$

$$y_{ij} - x_j \leq 0 \quad \forall i \in I, j \in J,$$

$$x_j \in \{0, 1\} \quad \forall j \in J,$$

$$y_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J.$$

Účelová funkce minimalizuje celkovou vzdálenost přepravy. První omezení stanovuje počet umístěných obslužných středisek  $p$ . Druhý požadavek vyžaduje, aby každý poptávkový uzel byl přiřazený právě jednomu zařízení. Třetí postulát omezuje poptávkové uzly, aby byly obslouženy jen z otevřených center. Předposlední nastavení stanovuje dvojité prostorové uspořádání proměnných. A poslední předpoklad určuje, že požadavek v uzlu má být přiřazený pouze k jednomu obslužnému centru. Dále se může nahradit  $y_{ij} \geq 0$  pro  $\forall i \in I, j \in J$ , protože třetí omezení garantuje že  $y_{ij} \leq 1$ . Pokud je nějaké  $y_{ij}$  zanedbatelné, může se jednoduše přiřadit uzel  $i$  k tomu nejbližšímu otevřenému obslužnému centru. [4]



## 4 Lokační problém na síti

Jedno z nejdůležitějších a propracovaných odvětví v lokační teorii je analýza lokačních problémů na síti. Síť může představovat např. stát, vybrané území, město nebo rozmístění výrobních linek ve výrobě. Síť vhodně interpretuje různé lokační modely. Tato kapitola bude popisovat teoretický aparát potřebný k vybranému problému.

### 4.1 Teorie grafů v lokačních úlohách

Bude se uvažovat neorientovaná síť (také označováno jako graf) a tím se rozumí uspořádaná trojice  $G = (V, X, p)$ . Prvky množiny  $V$  se nazývají vrcholy grafu  $G$ , prvky množiny  $X$  jsou hranami grafu  $G$  a  $p$  je incidencí grafu  $G$ . Incidence  $p$  grafu  $G$  přiřazuje každé jeho hraně neuspořádanou dvojici vrcholů. Pokud incidence hrany  $h \in X : p(h) = (u, v)$  rozumíme tím, že hrana  $h$  má společnou část (inciduje) s vrcholy  $u$  a  $v$  a vrcholy  $u, v \in V$  jsou krajními body hrany  $h$ . V neorientovaném grafu nezáleží na pořadí vrcholů  $u$  a  $v$ .

Mohutnost množiny vrcholů je označena  $n = |V|$  a mohutnost množiny hran je  $q = |X|$ . Počet hran incidujících s vrcholem  $v \in V$  je značeno jako stupeň vrcholu  $st(v)$ . Následující vztah platí pro každý graf udávající vztah mezi počtem hran a stupni vrcholů:

$$\sum_{v_i \in V} st(v_i) = 2 \cdot q.$$

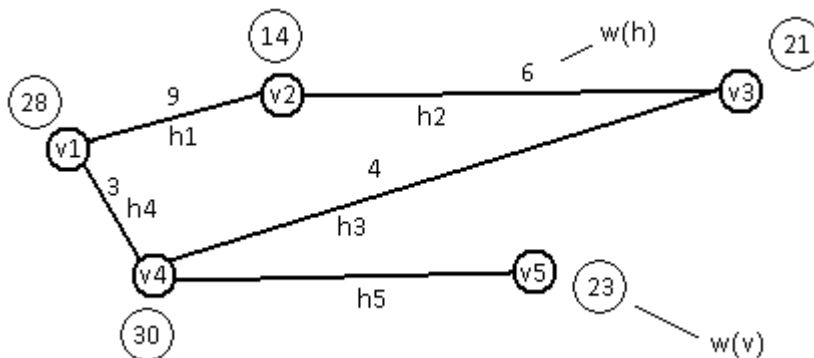
Literatura popisuje různé druhy grafů:

- multigraf (může obsahovat násobné hrany (= rovnoběžné hrany, hrany mají stejné krajní vrcholy), smyčky a izolované vrcholy),
- prostý graf (může obsahovat smyčky, neobsahuje násobné hrany),
- obyčejný graf (neobsahuje násobné hrany a smyčky),
- triviální graf (skládá se pouze z jednoho vrcholu, množina hran je prázdná),
- diskrétní graf (je tvořen izolovanými vrcholy a množina hran je prázdná),
- prázdný graf (neobsahuje žádnou množinu hran ani vrcholů).

Graf v této práci bude vyjadřovat běžnou komunikační síť:

- vrchol  $v$  (např. křižovatka komunikací, obec),

- hrana  $h$  (spojení nebo úsek mezi vrcholy),
- váha vrcholu  $w(v)$  (požadavek uzlu  $v$  na odsluhu),
- ohodnocení hrany  $w(h)$  (kvantitativní ohodnocení úseku – např. délka úseku, kvalitativní váha úseku – např. třída, kvalita komunikace). [3,13]



Obrázek 5: Příklad komunikační sítě (zdroj: autor)

#### 4.1.1 Maticové vyjádření neorientovaných grafů

Maticové vyjádření neorientovaného grafu je používáno ve výpočtech. Vyjadřuje důležité vlastnosti grafů.

*Matice sousednosti (přilehlosti)* grafu  $G = (V, X, p)$ , kde  $|V| = n$ , je označována jako matice  $V = (v_{ij})_{i,j=1}^n$  definovanou následovně:

- $v_{ij} = 0$ , když neexistuje hrana  $(v_i, v_j)$ ,
- $v_{ij} = 1$ , když hrana  $(v_i, v_j)$  existuje.

Vlastnosti matice přilehlosti:

- pro řádkové součty platí  $\sum_{j=1}^n v_{ij} = st(v_i)$ , pro  $i = 1, \dots, n$ , který vyjadřuje stupeň vrcholu  $v_i$ ,
- pro sloupcové součty je definován vztah  $\sum_{i=1}^n v_{ij} = st(v_j)$ , pro  $j = 1, \dots, n$ , který vyjadřuje stupeň vrcholu  $v_j$ .

Pokud jsou hrany grafu označeny jako  $h \in X$ , kde je  $h_1, h_2, \dots, h_m$ , může se graf vyjádřit pomocí *matice incidence*  $B = (b_{ij})_{i,j=1}^{n,m}$ , která je definována následovně:

- $b_{ij} = 0$ , jestliže  $v_i \notin p(h_j)$ ,
- $b_{ij} = 0$ , jestliže  $v_i \in p(h_j)$ .

Vlastnosti matice incidence:

- pro řádkové součty matice  $\sum_{j=1}^m b_{ij} = st(v_i)$  pro  $i = 1, \dots, n$  vyjadřující stupeň vrcholu  $v_i$ ,
- pro sloupcové součty matice  $\sum_{i=1}^n b_{ij} = 2$  pro  $j = 1, \dots, m$ , kde každá hrana  $h_j$  inciduje se dvěma vrcholy.

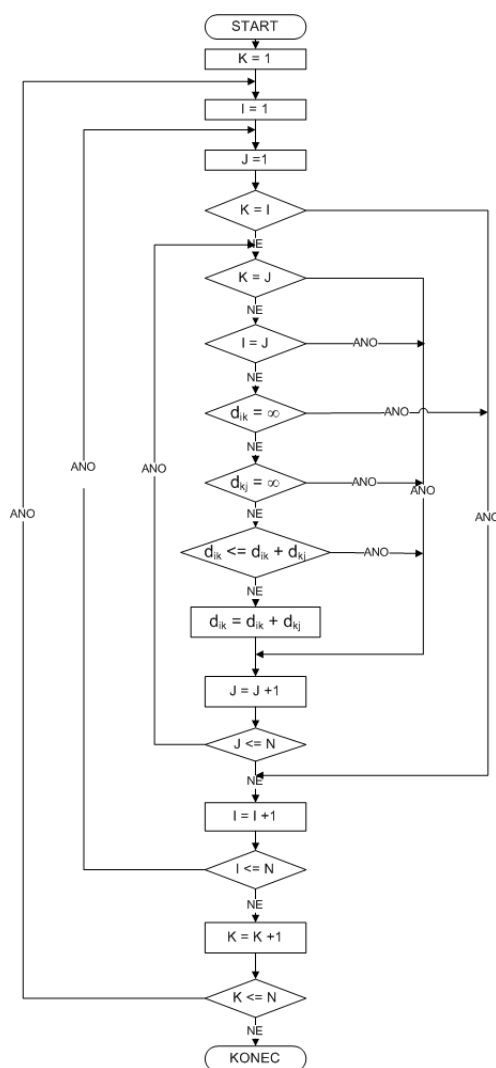
*Matici přímých vzdáleností*  $D = (d_{ij})_{i,j=1}^n$  v ohodnoceném grafu  $G = (V, X, p)$ , ve kterém je každé hraně  $h \in X$  přiřazena hodnota  $o(h)$  vyjadřující její délku (může se jednat o kvalitativní ohodnocení hrany, ne jen kvantitativní). Prvky matice přímých vzdáleností jsou definovány následovně:

- $d_{ij} = o(h)$  jestliže  $\exists h \in X$ , pro kterou  $p(h) = (v_i, v_j)$ ,  $i \neq j$ ,
- $d_{ij} = \infty$  jestliže  $\nexists h \in X$ , pro kterou  $p(h) = (v_i, v_j)$ ,  $i \neq j$ ,
- $d_{ij} = 0$  pro  $i = j$ . [3, 13]

Matice přímých vzdáleností se pomocí Floydova algoritmu (autor je R. W. Floyd), viz obrázek číslo 6, přepočítá na *matici vzdáleností*. Výsledkem algoritmu je matice, která obsahuje nejkratší cestu mezi vrcholy.

#### 4.1.2 Souvislost grafů

Je dán graf  $G = (V, X)$  a dvojice vrcholů  $u, v \in V$ . *Sled* je označení pro střídavou posloupnost vrcholů a hran mezi vrcholy  $u$  a  $v$ . Tyto vrcholy jsou krajními body sledu, kde  $u$  je počátečním a  $v$  je koncovým vrcholem. Sled, ve kterém se neopakuje žádná hrana je nazýván *tahem*. A tah, ve kterém se neopakuje žádný vrchol je nazýván *cestou*. Cesta mezi dvěma vrcholy  $u, v \in V$  je označována  $m(u, v)$ . Uzavřený sled, tah nebo cesta je, pokud se počáteční vrchol shoduje s koncovým vrcholem. Uzavřená cesta je nazývána *kružnicí*. Graf je označován jako souvislý, pokud mezi libovolnou dvojicí jeho vrcholů  $u, v \in V$  existuje alespoň jedna cesta. Sled, tah i cesta mají svou délku, která je dána součtem ohodnocení hran, které obsahuje. Speciální případem je tzv. smyčka, ve které incidujícími vrcholy hrany je jeden vrchol, a izolovaný vrchol, se kterým neinciduje žádná hrana. [3, 13]



Obrázek 6: Floydův diagram ve formě vývojového diagramu (zdroj:[13])

### 4.1.3 Vysvětlení pojmů lokačních úloh pomocí teorie grafů

Atrakční obvod  $A(v)$  obslužného střediska  $v \in D_k$  je množina vrcholů a hran sítě, pro které platí:

- $u \in A(v)$ , pokud  $\nexists w \in D_k$ , pro které  $d(w, u) < d(v, u)$ ,
- $h \in A(v)$ , pokud  $\nexists w \in D_k$ , pro které  $d(w, h) < d(v, h)$ .

Vzdálenost hrany  $h$  od zařízení  $v \in D_k$  je definována:

- $d(v, h) = \min\{d(v, r), d(v, s)\}$ , kde incidence hrany  $h \in X$  je  $p(h) = (r, s)$ .

Vzdálenost vrcholu  $u \in V$  od obslužného zařízení  $v \in D_k$  je definováno jako délka minimální cesty:

- $d(u, v) = \min_{m(u,v) \in M} \left\{ \sum_{h \in m(u,v)} o(h) \right\}$ , kde  $M$  je množina všech cest mezi  $u$  a  $v$ .

Prvotním atrakčním obvodem  $A'(v)$  depa  $v \in D_k$  je množina vrcholů a hran grafu (sítě), pro které platí:

- $u \in A'(v)$ , pokud  $\nexists w \in D_k$ , pro které  $d(w, u) \leq d(v, u)$
- $h \in A'(v)$ , pokud  $\nexists w \in D_k$ , pro které  $d(w, h) \leq d(v, h)$

Přiděleným atrakčním obvodem  $A^*(v)$  zařízení  $v \in D_k$  je množina vrcholů a hran grafu splňující vztahy:

- $A'(v) \subseteq A^*(v) \subseteq A(v)$  pro každé obslužné středisko  $v \in D_k$ ,
- $\bigcup_{v \in D_k} A^*(v) = X \cup V$ ,
- $A^*(v) \cap A^*(u) = \emptyset$  pro  $u \neq v; u, v \in D_k$ . [13]

#### 4.1.4 Optimální umístění obslužných středisek v grafu

Obslužná zařízení se mohou umísťovat buď do vrcholů sítě, nebo na její hrany. Proto se odděleně uvažuje obsluha vrcholů a hran sítě. Následují kritéria pro optimální lokaci středisek na grafu.

**Obsluha vrcholů grafu** Množina zařízení  $D_k$ , kde  $|D_k| = k$  je vrcholově optimálním umístěním  $k$  středisek na grafu  $G = (V, X)$ , pokud platí:

- $f(D_k) = \min_{D'_k} \left\{ f(D'_k) \right\}$ ,
- kde  $f(D'_k) = \sum_{v \in D'_k} \sum_{u \in A^*(v)} 2 \cdot d(u, v) \cdot w(u)$

**Obsluha hran grafu** Množina zařízení  $D_k$ , kde  $|D_k| = k$  je hranově optimálním umístěním  $k$  středisek na grafu  $G = (V, X)$ , pokud platí:

- $g(D_k) = \min_{D'_k} \{g(D'_k)\}$ ,
- kde  $g(D'_k) = \sum_{v \in D'_k} \sum_{h \in A^*(v)} 2 \cdot (d(v, h) + o(h)) \cdot w(h)$

$D'_k$  jsou všechny  $k$  prvkové podmnožiny  $V$ . [13]

## 4.2 Hakimiho algoritmus

Počátek zrodu Hakimiho studie lokačních problémů spadá mezi roky 1964 a 1965. Hakimi problém aplikoval na poli telekomunikací, přesněji v lokaci přepínacích center v komunikačních sítích, a tím představil p-mediánový a p-centrový problém. Hakimi neformuloval p-mediánovou úlohu jako celočíselný programovací problém. To jako první udělal ReVelle a Swain roku 1970 nezávisle na Hakimim. Až do dnešní doby vznikají stále nové alternativy p-mediového problému, který má široké uplatnění.

Hakimiho algoritmus řeší umístování obslužných středisek na hranách sítě tak, aby minimalizoval určitou míru dopravních nákladů. Algoritmus vyhledává na každé hraně grafu místo (resp. místa) s minimální váženou excentricitou. Z těchto nalezených míst vybereme místo, kde bude vážená excentricita minimální, a v tomto místě bude umístěno depo. Jedná se o p-mediánový problém. K tomu, aby se mohl problém řešit, je potřeba zavést následující pojmy. [4]

*Vážená excentricita vrcholu*  $v \in V$  je definována:

- $ec(v) = \max_{u \in V} \{d(u, v) \cdot w(u)\}$ , kde  $w(u)$  je váha vrcholu.

*Vzdálenostně optimálním umístěním zařízení* na síti  $G = (V, X)$  je vrchol  $v^* \in V$ , pro který platí:

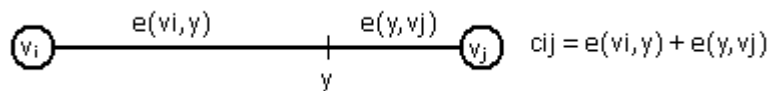
- $ec(v^*) = \min_{v \in V} \{ec(v)\}$ .

*Vážená excentricita bodu y hrany*  $(v_i, v_j)$  je definována vztahem:

- $ec(y) = \max_{u \in V} \{d(y, u) \cdot w(u)\}$ , kde  $d(y, u) = \min [e(v_i, y) + d(v_i, u)], [e(y, v_j) + d(v_j, u)]$ ,
- $c_{ij} = e(v_i, y) + e(y, v_j)$ , kde  $c_{ij}$  je délka hrany  $(v_i, v_j)$  (viz obrázek 7).

*Úplně vzdálenostně optimálním umístěním zařízení* je bod  $y^* \in G$ , pro který platí:

- $ec(y^*) = \min_{y \in G} \{ec(y)\}$ .

Obrázek 7: Vážená excentricita bodu  $y$  hrany  $(v_i, v_j)$  (zdroj: [15])

### 4.2.1 Postup Hakimiho algoritmu

Na grafu se určí pro všechny hrany  $h_k \in X$  bod nebo body  $y_k \in h_k$  s nejmenší váženou excentricitou. Na hraně nebo hranách může existovat více bodů se stejnou minimální váženou excentricitou. V tomto případě se vybere jedno místo s minimální váženou excentricitou.

Ze všech určených bodů  $y_k$ , kde  $k = 1, 2, \dots, m$ ,  $m \geq q$  a  $q = |X|$ , se určí bod  $y^* \in G$ , který má nejmenší váženou excentricitu. Tento bod je absolutním obslužným střediskem.

Praktický postup je velmi pracný, protože se jedná o grafickou metodu, kde se pro každou hranu sítě vytváří diagram hodnot. Počet diagramů se může omezit pomocí následujících vťahů:

- Absolutní zařízení na hraně  $(v_i, v_j)$  je spojeno s hodnotou  $p_{ij} = \max_{v_t \in V} \{w(v_t) \cdot \min\{d(v_i, v_t), d(v_j, v_t)\}\}$ , kde  $v_t$  jsou zbývající vrcholy v síti (odlišné od vrcholů  $v_i, v_j$ ). Hodnota  $p_{ij}$  je dolním ohraničením vážené excentricity absolutního depa ležícího na hraně  $(v_i, v_j)$ .
- Dolní ohraničení absolutního obslužného střediska na grafu  $G = (V, X)$  je dáno  $p = \min_{(v_i, v_j) \in X} \{p_{ij}\}$ .
- Pokud je absolutní zařízení na hraně  $(v_i, v_j)$ , potom hodnota  $h_{ij} = p_{ij} + \frac{1}{2} \cdot c_{ij} \cdot w(v_r)$ , kde  $w(v_r)$  je váha vrcholu, pro který je  $w(v_t) \cdot \min\{d(v_i, v_t), d(v_j, v_t)\}$  největší. Hodnota  $h_{ij}$  je horním ohraničením vážené excentricity absolutního depa ležícího na hraně  $(v_i, v_j)$ .
- Horní ohraničení absolutního obslužného střediska na grafu  $G = (V, X)$  je dáno  $H = \min_{(v_i, v_j) \in X} \{h_{ij}\}$ .
- Pro každou hranu  $(v_i, v_j) \in X$ , kde  $p_{ij} \geq H$ , platí, že může být postavena mimo množinu hran, pro které se sestavují diagramy hodnot.

V závislosti na  $e$  se pro každou hranu, teď už pouze pro hrany, které nesplnily výše uvedenou podmínku, sestavují diagramy hodnot, které jsou dány vztahem:

- $T_i = w(v_i) \cdot (e + d(v_b, v_i))$ ,
- $T_i' = w(v_i) \cdot (c_{ab} - e + d(v_a, v_i))$ , kde  $c_{ab}$  je délka hrany  $(v_a, v_b)$ . Pro určení vzdálenosti mezi jednotlivými vrcholy je vypočítat distanční matici (viz kapitola 5.1.1).

Pro všechny vrcholy  $v_i \in V$  se určí dolní ohraničení lomených čar  $T_i, T_i'$ . Dále se určí nejmenší horní ohraničení všech dříve určených dolních ohraničení. Minimum ze vzniklého ohraničení je  $ec(y_k)$ . [13]

## 5 Analýza současného stavu problematiky ve světě a v České republice

Následující kapitola nejprve popisuje instituce zabývající se vývojem softwaru pro lokační teorii. Další část kapitoly analyzuje současný stav lokace a alokace logistických center v České republice a možné způsoby využití lokačních modelů.

### 5.1 Instituce zabývající se vývojem softwaru pro lokační teorii

Tato kapitola se soustřeďuje na instituce, které se zabývají vývojem lokační teorie, zejména vytváření aplikací, které řeší lokační problémy. Existují však také instituce, které se zaměřují na podporu výzkumné činnosti (např. Deutsche Forschungsgemeinschaft, SAP Advanced planner and optimizer), ale nepodílí se na vývoji softwaru.

Výzkumná komunita lokační teorie se skládá z lidí z velmi různorodých oblastí (např. matematici, pracovníci v informatice, operační výzkumníci, průmyslový inženýři, marketingoví výzkumníci, regionální plánovači, stejně jako odborníci z různých odvětví pracujících na lokačních problémech). Každá skupina lidí dává důraz na jiné aspekty. Lokační problémy se řeší pomocí metod z různých oborů. V důsledku toho se nepoužívá společný jazyk při řešení problémů. Při řešení určitého problému, se může stát, že problém byl již vyřešen, ale pod úplně jiným názvem.

Ve snaze vyrovnat se s tímto problémem, jsou navrženy klasifikační systémy pro lokační modely. Z praktického hlediska je velmi obtížné stanovit převod mezi různými skupinami, které se zabývají lokačními problémy. Například matematici mají úplně jiný způsob popisu jejich řešících postupů než regionální plánovači.

Možný způsob řešení tohoto problému, bylo vyvinout software pro lokační modely a zpřístupnit ho pro komunitu zabývající se lokační teorií. Existuje několik skupin uživatelů tohoto softwaru, kteří mají odlišné požadavky:



**Učitelé** si přejí software s dobrým uživatelským prostředím a implementací všech algoritmů, které jsou popsány v jejich kursových učebnicích, ukáže ilustrační příklady a bude levný.

**Studenti** by používali software, který jim lépe pomáhá rozumět teoretickému materiálu a také je podporuje v tvorbě jejich domácích úkolů. Rádi by obsluhovali tento software na jejich domácích počítačích. Tato aplikace by měla být bezplatná.

**Výzkumníci** by užívali software s tolika implementovanými algoritmy, jak jen je to možné. Potřebují konstruovat příklady a protipříklady. Navíc potřebují software, který jim poskytne knihovnu, která může být použita pro rychlejší zavádění nových algoritmů. Program by měl být bezplatný a zdrojový kód volně dostupný.

**Praktici** by pracovali s programem, který může ovládat přesně určenou třídu problému, kterou potřebují řešit. Měl by být schopný vyřešit rozsáhlý problém v poměrně krátkém čase. Nalezení optimálního řešení není nutné, ve většině situací jim postačí dobré řešení. Software by měl mít vhodné rozhraní pro importování a exportování dat. Cena není důležitá.

**Softwarové společnosti** si přejí vyvinout program, který je snadno konfigurovatelný a ovládá několik tříd. Měl by umět zpracovat uživatelské specifické problémy. Software by měl mít moderní design a poskytovat možnost propojení se třetí stranou programového vybavení. Cena není důležitá. [4]

## 5.2 LoLA – Library of Location Algorithms

Hlavní motivací pro vyvíjení LoLA bylo budování rozsáhlé sbírky algoritmů pro lokační teorii zahrnující rovinné, síťové a diskrétní problémy a použít tyto algoritmy pro rychlé a snadné řešení různých problémů. LoLA lze bezplatně stáhnout z internetové adresy <http://www.mathematik.uni-kl.de/loLA>. Navíc jsou na této adrese i všechny zdrojové kódy. Pro zjištění vhodného algoritmu na daný lokační problém (tzn. odhalení, zde byl daný problém řešený v literatuře nebo systémem LoLA), se použije jednotný jazyk (tzv. klasifikační systém pro snadný přístup k implementovaným algoritmům). Tento jednotný jazyk (schéma) se skládá z následujících pěti pozic pos1/pos2/pos3/pos4/pos5. Význam každé pozice spolu s příklady je v tabulce 3.

Tabulka 3: Klasifikační schéma pro lokační problém (zdroj: [4])

Pozice	Význam	Značení	Příklady
1	číslo nového střediska		
2	typ problému	P D G	rovinný problém diskrétní problém problém na neorient. grafu
3	zvláštní předpoklady a omezení	$w_m = 1$	všechny váhy jsou si rovny
4	typ vzdálenostní funkce	$\gamma$	hlavní míra
5	typ účelové funkce	$\sum$ max	mediánová problém centrální problém

LoLA je navrhnutý softwarový systém obsahující algoritmické metody známé v lokačním plánování. Skládá se z GUI (grafické uživatelské rozhraní), textového rozhraní a programovacího rozhraní, jehož cílem je umožnit uživatelům napsat vlastní C++ program pomocí algoritmů z LOLA-knihoven. [4, 19]

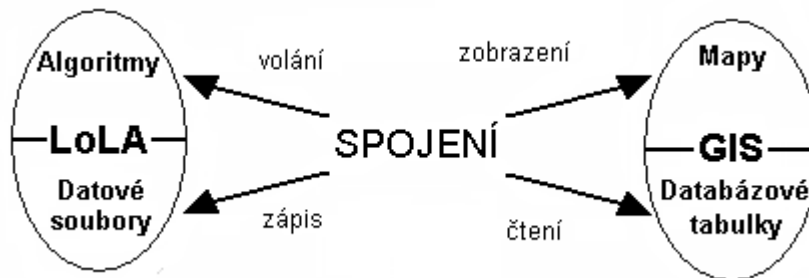
### 5.2.1 LoLA v GIS

Pro řešení skutečných světových problémů, je zapotřebí dostat do LoLA skutečná data. Pokud existuje velká množina zařízení obsluhy, jejichž lokality jsou uloženy v určité databázové tabulce (nebo v něčem podobném). Existují dva způsoby, které se můžou použít pro vložení do LoLA, a tím vytvořit vstupní soubor pro práci s daty. Na jedné straně se vstupní soubor může vytvořit ručně, to jest zkopírovat a vložit, a na druhé straně se může využít LoLA grafické rozhraní. Obě zmiňované cesty jsou velmi zdoluhavé a neúčinné, proto se vyvinul třetí způsob, který pomocí konvertoru z databázové tabulky vytvoří vstupní soubor.

Geografické informační systémy (GIS) jsou určeny pro vizualizaci skutečných světových dat zemí, států, měst, atd. s mapovými daty a dalšími informacemi, které dovolí najít rozhodovateli řešení problému jako je umístění střediska, směřování, atd.

Zdá se to být přirozený nápad spojit oba systémy, GIS a LoLA. LoLA potřebuje skutečná světová data a snadný a rychlý způsob jak vytvářet vstupní data, zatímco GIS potřebuje důmyslnější algoritmy k vytváření lokačních rozhodnutí. Jeden ze základních úkolů je transformovat data z GIS do LoLA a naopak. GIS by měl být schopen pracovat s algoritmy z LoLA, získat řešení lokačního problému a zobrazit ho na mapě s danými daty. Obrázek č. 8 zobrazuje schéma funkčnosti správného spojení.

Velmi užitečný nástroj GIS je ArcView (první ze tří úrovní ArcGIS Desktop, vyvinutý společností ESRI a dalšími organizacemi). V ArcGIS je implementovaný jazyk zvaný



Obrázek 8: Spojení GIS a LoLA (zdroj: [4])

Avenue. Tento jazyk je uživatelsky snadno ovladatelný a má schopnost kontrolovat každý proces v ArcView. A to je přesně to pojítko mezi LoLA a GIS. Implementují se skripty, které jsou čtené z GIS databáze, kde jsou všechny data uložena, a konvertují se do LoLA vstupního souboru. Stejně tak obráceně se mohou dostat vlastní řešení z LoLA zpět do ArcView. A proto ArcView společně s Avenue plní nezbytné podmínky pro spojení s LoLA. Pomocí jazyka Avenue je vytvořeno rozhraní mezi ArcView a LoLA. [4]

### 5.3 INFORMS – Institute for Operations Research and the Management Sciences

Historie a tradice výboru INFORMS podporuje a usnadňuje shromažďování a uchovávání materiálů týkající se operačního výzkumu a managementu. Příslušné materiály jsou v podobě dokumentů, audio a video nahrávek a dalších artefaktů. Podporuje vytváření, šíření nových a historických materiálů ve společnosti. Mezi tyto materiály se řadí knihy, sympózia, zasedání a celostátní setkání, elektronické média a další prostředky.

Operační výzkum s použitím pokročilých analytických metod se využívá k lepšímu rozhodování. Poskytuje trvale významnou hodnotu pro organizace a vedoucí pracovníky, kteří ji využívají. Zde jsou uvedeny některé výhody vyplývající z operačního výzkumu:

- různý obchodní pohled,
- zvýšení obchodní výkonnosti,
- snížení nákladů,
- efektivnější rozhodování,
- prognózování,

- lepší plánování,
- zvýšení rychlosti a snížení zpoždění,
- zlepšování kvality,
- zvýšení výnosů,
- zvýšení produktivity,
- odhalování rizik,
- větší využití zdrojů a další.

Management je interdisciplinární obor aplikované matematiky, strojírenství a přírodních věd, který používá různé vědecké výzkumné zásady, strategie a analytické metody včetně matematického modelování, statistiky a algoritmy pro zlepšení schopnosti organizace, pro přijetí racionálního a smysluplného rozhodování managementu. Některé oblasti, které jsou zahrnuty v řízení vědy, jsou:

- data mining,
- teorie her,
- logistika,
- matematické modelování,
- průmyslové inženýrství,
- strojírenství,
- rozhodovací analýza,
- prognóza,
- optimalizace,
- pravděpodobnost a statistika,
- projektové řízení,
- simulace,

- sociální sítě,
- doprava prognostických modelů,
- zásobování a mnoho dalších.

Tato organizace má rozsáhlé pole působnosti. Podílí se na vydávání knih a časopisů. Představenstvo společnosti jsou vysoce vzdělaní lidé v různých oborech a z různých koutů světa. INFORMS spolupracuje se světově známými podniky (např. IBM, SAS). Členství v této komunitě není zadarmo. Platí se i vstup do různých sekcí a manipulace s jednotlivými materiály. [14]

### 5.3.1 SoLA – Section on Location Analysis

Sekce výboru INFORMS podporuje rozvoj analytických metod, technik a nástrojů v tradičním operačním výzkumu a managementu pro studium lokačních problémů. Umístění objektu je definováno jako výběr místa nebo míst pro zařízení, na které jsou kladeny určité požadavky. Tyto nároky jsou formulovány tak, aby střediska poskytovaly služby s cílem optimalizovat účinnost, spravedlnost, míru efektivity a to vše s omezenými zdroji.

Aplikačními oblastmi jsou zdravotní služby, životní prostředí a přírodní zdroje, doprava a logistika, výroba a řízení zásob v městském a veřejném sektoru. SoLA se zabývá matematickým programováním, multikriteriálními rozhodovacími modely, stochastickým modelováním a ekonomii. SoLA se pokouší posílit vazby mezi jednotlivými sekcemi INFORMS, jejichž zájmy se překrývají v aplikačních oblastech. [23]

## 5.4 EWGLA – EURO Working Group on Locational Analysis

EWGLA je Evropská asociace Národní společnosti operačního výzkumu. Vznikla na prvním letním samitu EURO v Bruselu mezi červencem a srpnem roku 1984. Tato společnost je zaměřená na lokační teorii. Zpočátku bylo hlavním cílem vytvořit síť lidí, kteří budou pracovat dohromady, a udržet kontinuitu při výměně informací mezi účastníky. Od té doby tato pracovní skupina roste a rozšiřuje svoji působnost.

Zájmy skupiny se soustřeďují se na optimální volbu lokalit pro jeden nebo více objektů v jakémkoli systému (diskrétním nebo spojitým prostorem). Zabývá se i dalšími oblastmi jako je doprava, směřování problémů, statistika, shluková analýza, matematické (lineární, celočíselné, nelineární, konvexní, globální) programování, vícekriteriální analýza, teorie aproximace, výpočetní geometrie atd.

EWGLA se stala prostředkem komunikace mezi členy, zabývající se operačním výzkumem, v Evropě i v zahraničí. Hlavním cílem je podporovat výměnu informací a spolupráci v oblasti výzkumu a aplikací, podporovat myšlenky a metody vyvinuté v rámci oboru.

Skupina pořádá každoroční setkání, které svolává jeden ze členů. Zájemci o prezentaci své práce se mohou připojit ke skupině na jejích schůzích. Teoretické příspěvky nebo aplikovaný výzkum, prezentace softwaru, jsou ceněny. Recenzované jednání každého zasedání jsou pro účastníky zdarma a jsou zveřejňovány obvykle jako speciální vydání publikace *Studií v lokační analýze*. [22]

## 5.5 Řešení lokačních problémů v České republice

Tato kapitola analyzuje současný stav řešení lokace a alokace (činnost, ve které se poptávkové body přiřazují k určitému logistickému centru) obslužných center v České republice. Lokace obslužných zařízení v České republice je řešena pomocí lokačních úloh. Může se jednat buď o lokaci ve spojitém, nebo v diskrétním prostoru.

Spojité prostoru je kontinuální a střediska obsluhy je možné umístit kamkoliv do určitého geometrického prostoru. Jedná se o lokační problém  $W$  definovaný v kapitole 3.2. K řešení vhodného umístění obslužných středisek se používá například Weiszfeldův algoritmus (také implementovaný v knihovně LoLA).

V případě řešení diskrétní lokace je předem určená množina možných umístění obslužných center. Existuje tedy jen několik přijatelných variant, mezi kterými se rozhoduje. Pro řešení optimální lokace se nabízí multikriteriální analýza. Rozhodování probíhá pomocí různých kritérií, která jsou většinou protichůdná. Na základě ohodnocení těchto kritérií se určuje optimální varianta problému.

V případě lokace na dopravní síti lze použít kterýkoli vhodný grafový algoritmus. Pracuje se s modelem dopravní sítě, který je interpretovaný jako graf. Je však nutné posoudit, zda výhody z řešení převažují nad úsilím spojeným s jeho konstrukcí. [5]

### 5.5.1 Weiszfeldův algoritmus

Mezi  $p$  středisky obsluhy  $X = (x, y)$  a poptávkovými body  $P = (a, b)$  se pomocí euklidovské metriky (viz kapitola 3.5.1) vypočítá jejich vzájemná vzdálenost. Náklady na obsluhu  $i$ -tého objektu se určí jako součin této vzdálenosti a váhy (požadavek v uzlu)  $w_i$ , celkové

náklady na obsluhu všech objektu pak jako:

$$f(x, y) = \sum_{i=1}^p w_i \cdot \sqrt{(x-a)^2 + (y-b)^2}.$$

Samotný algoritmus pracuje na principu transformace  $W(x, y)$ , která převádí souřadnice bodu  $(x, y)$  do souřadnic bližších optimu. Výchozí souřadnice mohou být zvoleny libovolně, pomocí transformace jsou vypočítány nové souřadnice z původních. Tato posloupnost se blíží (tedy konverguje) k optimálnímu řešení problému.

Samotná transformace  $W(x, y)$  je odvozena na základě parciálních derivací funkce  $f(x, y)$ . Následující parciální derivace se položí rovny nule a vyjádří se z nich souřadnice  $x$  a  $y$ . [5]

$$\frac{\partial f(x, y)}{\partial x} = \sum_{i=1}^p \frac{w_i}{\sqrt{(x-a)^2 + (y-b)^2}} \cdot (x-a)$$

$$\frac{\partial f(x, y)}{\partial y} = \sum_{i=1}^p \frac{w_i}{\sqrt{(x-a)^2 + (y-b)^2}} \cdot (y-b)$$

## 5.6 Způsoby alokace veřejných logistických center

Veřejné logistické centrum má všechny vlastnosti jako obyčejné logistické středisko s tím rozdílem, že vznik a rozvoj sítě těchto center není možný bez jednotné koncepce s možností podpory ze státního rozpočtu nebo veřejných fondů včetně zdrojů Evropské Unie.

Při řešení alokace veřejných logistických center jsou použity tři způsoby řešení:

- *Analýza výroby a spotřeby* Výroba je posuzována podle rozmístění výrobních podniků v kategorii podle velikosti. Velikost podniku je dána počtem zaměstnanců. Vhodnost podniků byla posuzována podle oboru činnosti dle statistického sledování (Odvětvová klasifikace ekonomických činností – OKEČ). Jednotlivé obory byly posuzovány podle náročnosti na přepravu. Analýza spotřeby je daná počtem obyvatel a jejich kupní silou. Roli zde sehrává i cestovní ruch.
- *Analýza na základě přepravních proudů* tj. vyhodnocení současných přepravních proudů.
- *Analýza na základě geografických vah* Stanovení lokalit na základě geografických dat, jehož výsledkem je stanovení souřadnic pro alokaci veřejného logistického centra jak na úrovni krajů, tak na úrovni celostátní. [18]

## 5.7 Indikátory pro místní udržitelný rozvoj

Každá obec, aby se dále vyvíjela, by se měla snažit o trvale udržitelný rozvoj. Udržitelný rozvoj je proces, který splňuje stávající potřeby lidské společnosti, aniž by se dotýkal schopnosti budoucích generací uspokojovat své potřeby. V České republice dala popud k činnosti evropská kampaň *Sustainable Cities* (Udržitelná města), z projektu Evropské komise, *European Common Indicators* (ECI, Společné evropské indikátory).

Výsledkem této iniciativy vznikla v roce 2005 v České republice společnost *Týmová iniciativa pro místní udržitelný rozvoj* (TIMUR), což je občanské sdružení, jehož posláním je podporovat udržitelný rozvoj měst, obcí a jejich sdružení v České republice zejména zaváděním místních indikátorů udržitelného rozvoje. Indikátory představují ukazatele vývoje určitého vybraného jevu získané průběžným sledováním, zaznamenáváním a vyhodnocováním souboru přesně stanovených údajů. Doposud v České republice podobná společnost neexistovala. Základní prvky udržitelnosti, které tvoří podklady pro výběr indikátorů jsou následující:

- Rovnost a společenské začlenění (dostupnost přiměřených a cenově přijatelných základních služeb pro všechny, např. vzdělání, zaměstnání, energie, zdraví, bydlení, školení, doprava);
- místní správa/pravomoci/demokracie (účast všech skupin místní komunity na místních plánovacích a rozhodovacích procesech);
- lokální/globální vztahy (uspokojovat místní potřeby místně, od výroby po spotřebu a odpad; uspokojovat ty potřeby, které nelze uspokojit lokálně, udržitelnějším způsobem);
- místní hospodářství (sladit místní dovednosti a potřeby s dostupností zaměstnání a dalšími zařízeními tak, aby co nejméně ohrožovaly přírodní zdroje a životní prostředí);
- ochrana životního prostředí (přijmout přístup vytváření ekosystémů; minimalizovat využívání přírodních zdrojů a půdy, tvorbu odpadů a znečišťujících emisí, chránit biologickou rozmanitost);
- kulturní dědictví/kvalita vybudovaného životního prostředí (ochrana, zachování a rehabilitace historických, kulturních a architektonických hodnot, včetně budov, památníků, událostí; rozšiřovat a chránit přitažlivost a funkčnost prostor a budov).



Obecné sdružení TIMUR poskytne na základě analýzy obce a vybraných indikátorů podklad pro nápravu špatné situace a návrh vhodných opatření, které budou respektovat požadavky těch, kterých se týkají.

Analýza těchto indikátorů může řešit také lokační problém v obci. Výsledkem této analýzy může být například to, že:

- obci je umístěno nedostatek škol, veřejných prostranstvích a služeb apod.,
- cesta dětí do školy vede skrz nebezpečná místa,
- část obce s vysokou hustotou obyvatel je položena v prostředí zatížené hlukem nebo znečištěným ovzduším apod.

Při řešení používá spojitý nebo diskrétní prostor. Řešit se může například pomocí plného, maximálního, prázdného a minimálního pokrytí (viz kapitola 3.5.2). Dále se může při řešení využívat všech osm základních diskrétních problémů (kapitola 3.6.1). Existuje několik sad indikátorů udržitelného rozvoje na místní úrovni, ale nejsou implementovány v České republice. [24]

## 6 Lokace logistických center internetových obchodů

Neusteustále rostoucí poptávka po internetovém zboží, donutila internetové obchody zřízovat logistická centra. Tyto střediska obsluhy umožňují zabalit více zásilek a logistická centra mají blíže k zákazníkům a ke zboží (zákazníci si mohou pro zboží sami dojít).

Internetový obchod, také nazývaný jako e-shop, je aplikace využívající komerci v prostředí Internetu. E-shop je založený na databázovém a webovém serveru. Prostřednictvím elektronického obchodu lze prodávat a kupovat zboží, digitální služby a informace. Obchodování na Internetu označuje obchodní a marketingové aktivity či transakce mezi spotřebitelem (customer, consumer) a podnikem (business). Součástí je i placení přes Internet (e-payment).

Internetový obchod je součástí e-komerce, což je též obchodování na Internetu mezi spotřebitelem a firmou, ale s rozšířenou působností. Do této skupiny patří např. rezervace letenek, prodej různého zboží, her. Dále se sem řadí služby jako je minilab on-line (digilab), aukce on-line, elektronické bankovníctví (e-banking) apod. Do oblasti čistě Internetového byznysu (digibusiness) patří nákup shareware, jeho stažení a registrace, vzdělávání na Internetu (e-learning), informační služby, portály, živé hudební vysílání, fotobanky, elektronická tržiště, což je virtuální místo pro obchodování mezi odběrateli a dodavateli.

Logistický proces v internetovém obchodě je složitý a je nutno učinit mnoho kroků, ovšem za co nejkratší dobu a v deklarované kvalitě.

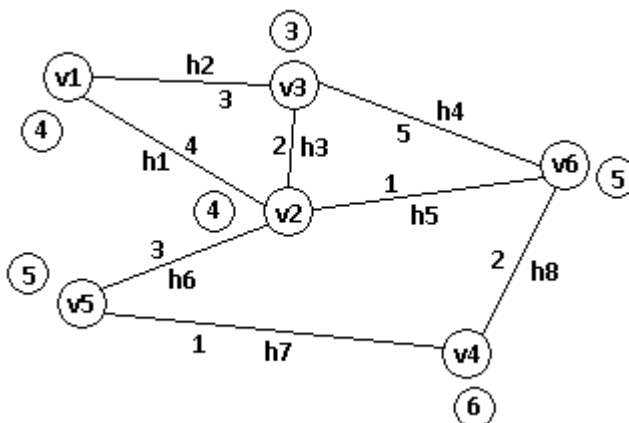
Na začátku všeho stojí objednávka, kterou zákazník po výběru příslušného typu a množství zboží vyplní a odešle přes webové stránky. Po přijetí objednávky obchodem systém ověří dostupnost zboží a odešle zákazníkovi e-mail s potvrzením objednávky. Pokud je požadované zboží přímo na skladě, ihned je systém zablokuje. Pokud je dostupné u dodavatele, vyšle objednávku. Dalším krokem je tedy vyskladnění, zajištění distribuce a fakturace. Následně přichází na řadu přeprava zboží dle zákaznickova výběru. Některé e-shopy zákazníkům umožňují nahlédnout pod pokličku a díky plně zabezpečenému zpřístupnění dílčích částí informačního systému sledovat celý proces vyřízení jejich objednávky. [6, 12]

## 6.1 Vymezení podmínek pro lokaci logistického centra

Diplomová práce se zaměřuje na řešení lokace logistických center internetových obchodů, které si doručení zásilky ke koncovému zákazníkovi vykonávají samy. Aby doručování bylo efektivní a rychlé, je potřebné zřídit logistická centra.

Při řešení optimální lokace obslužného střediska se použije ohodnocená síť zobrazená na obrázku číslo 9, jedná se o obyčejný graf (viz kapitola 4.1). V síti jsou ohodnocené hrany a vrcholy. Ohodnocení hran představuje čas přepravy po komunikaci a počet požadavků ve vrcholech určuje jejich ohodnocení. V síti se bude umísťovat jedno logistické centrum a to na hraně sítě a obsluhovat se budou vrcholy grafu. Na optimální umístění se použije Hakimiho algoritmus (viz kapitola 4.2).

Lokace na spojnicích mezi vrcholy může mít různé příčiny, zejména ekonomické (např. drahé pozemky), politické, technické a ekologické. V řešení se zanedbávají všechny obchodní procesy, marketing, řešení e-shopu, webmining apod. Práce řeší umístění logistického centra v území tak, že minimalizuje náklady na dodání zboží k zákazníkovi pomocí logistického centra případně podporu zákazníka (např. servis). Je zvoleno určité území, které bude obsluhováno a na kterém není žádné omezení. Doplnění a udržování fyzických zásob ve středisku práce neřeší.



Obrázek 9: Síť použitá pro řešení lokace (zdroj: autor)

## 6.2 Řešení lokace obslužného centra v síti

Postup řešení je popsán v kapitole 4.2 Hakimiho algoritmus. Nejprve se určí matice vzdáleností (viz kapitola 4.1.1), která je znázorněna v tabulce 4.

Tabulka 4: Matice vzdáleností (zdroj: autor)

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$
$v_1$	0	4	3	7	7	5
$v_2$	4	0	2	3	3	1
$v_3$	3	2	0	5	5	3
$v_4$	7	3	5	0	1	2
$v_5$	7	3	5	1	0	3
$v_6$	5	1	3	2	3	0
$w_i$	4	4	3	6	5	5

Pokračuje se výpočtem dolního ohraničení vážené excentricity absolutní depa. Vzorec je uveden v kapitole 4.2 Hakimiho algoritmus. Uvedený výpočet se vztahuje jen na existující hrany grafu.

$$p_{12} = \max\{3 \cdot \min\{3, 2\}, 6 \cdot \min\{7, 3\}, 5 \cdot \min\{7, 3\}, 5 \cdot \min\{5, 1\}\} = 18$$

$$p_{13} = \max\{4 \cdot \min\{4, 2\}, 6 \cdot \min\{7, 5\}, 5 \cdot \min\{7, 5\}, 5 \cdot \min\{5, 3\}\} = 30$$

$$p_{23} = \max\{4 \cdot \min\{4, 3\}, 6 \cdot \min\{3, 5\}, 5 \cdot \min\{3, 5\}, 5 \cdot \min\{1, 3\}\} = 18$$

$$p_{25} = \max\{4 \cdot \min\{4, 7\}, 3 \cdot \min\{2, 5\}, 6 \cdot \min\{3, 1\}, 5 \cdot \min\{1, 3\}\} = 16$$

$$p_{26} = \max\{4 \cdot \min\{4, 5\}, 3 \cdot \min\{2, 3\}, 6 \cdot \min\{3, 2\}, 5 \cdot \min\{3, 3\}\} = 16$$

$$p_{36} = \max\{4 \cdot \min\{3, 5\}, 4 \cdot \min\{2, 1\}, 6 \cdot \min\{5, 2\}, 5 \cdot \min\{5, 3\}\} = 15$$

$$p_{45} = \max\{4 \cdot \min\{7, 7\}, 4 \cdot \min\{3, 3\}, 3 \cdot \min\{5, 5\}, 5 \cdot \min\{2, 3\}\} = 28$$

$$p_{46} = \max\{4 \cdot \min\{7, 5\}, 4 \cdot \min\{3, 1\}, 3 \cdot \min\{5, 3\}, 5 \cdot \min\{1, 3\}\} = 20$$

Nyní jsou spočítány dolní ohraničení vážené excentricity absolutního depa všech hran. Určí se  $p = \min\{p_{ij}\}$ , což je dolní ohraničení absolutního depa na síti. Jako další krok je výpočet horního ohraničení vážené excentricity absolutního depa  $h_{ij}$  (viz tabulka 5). Vzorec pro výpočet horního ohraničení  $h_{ij}$  je uveden v kapitole 4.2 Hakimiho algoritmus.

Tabulka 5: Výpočet horního ohraničení vážené excentricity absolutního depa (zdroj: autor)

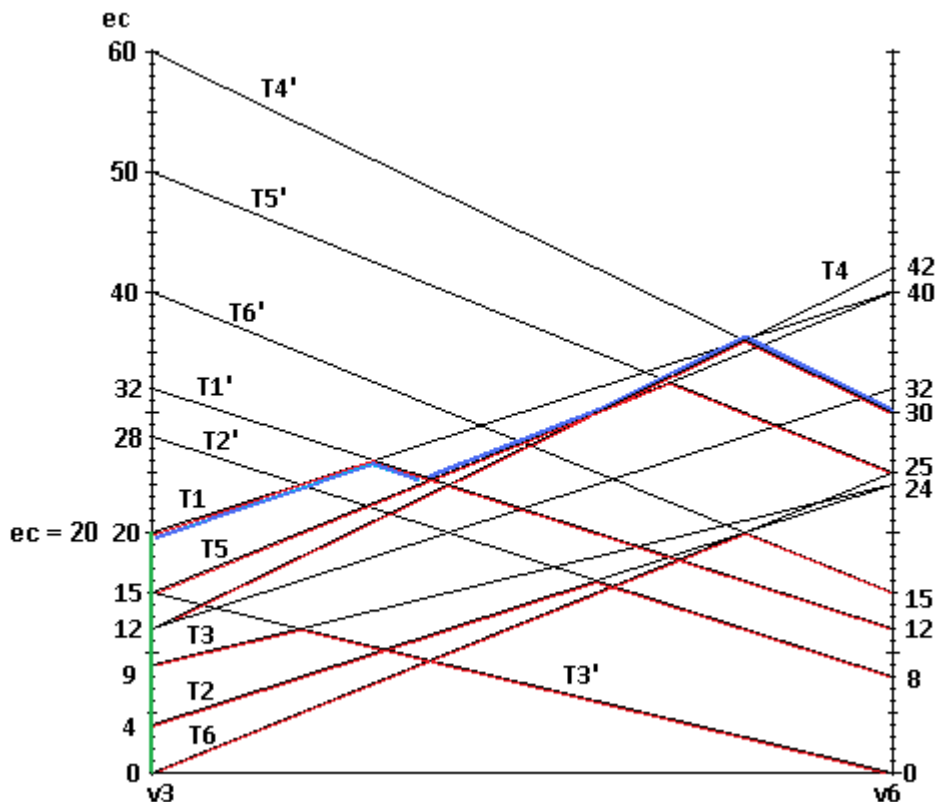
$(v_i, v_j)$	$p_{ij}$	$c_{ij}$	$w(v_r)$	$h_{ij}$
$(v_1, v_2)$	18	4	6 ( $v_4$ )	30
$(v_1, v_3)$	30	3	6 ( $v_4$ )	39
$(v_2, v_3)$	18	2	6 ( $v_4$ )	24
$(v_2, v_5)$	16	3	4 ( $v_1$ )	22
$(v_2, v_6)$	16	1	4 ( $v_1$ )	18
$(v_3, v_6)$	15	5	5 ( $v_5$ )	27,5
$(v_4, v_5)$	28	1	4 ( $v_1$ )	30
$(v_4, v_6)$	20	2	4 ( $v_1$ )	24

Horní ohraničení absolutního depa  $H$  je rovno  $\min\{h_{ij}\}$ . Nyní se vynechají hrany, které splnily podmínku  $p_{ij} \geq H$ , což znamená  $p_{ij} \geq 18$ . Podmínku nesplnily hrany, které mají  $p_{ij}$  menší než 18, jedná se o hrany:  $(v_2, v_5)$ ,  $(v_2, v_6)$ ,  $(v_3, v_6)$ . Pro tyto tři hrany, se budou sestrojovat diagramy hodnot.

Podle diagramů hodnot se určí nejprve všechny dolní ohraničení a poté horní ohraničení z těchto dolních ohraničení. Absolutní depo  $y_k^*$  bude umístěno v minimálním bodě horního ohraničení.

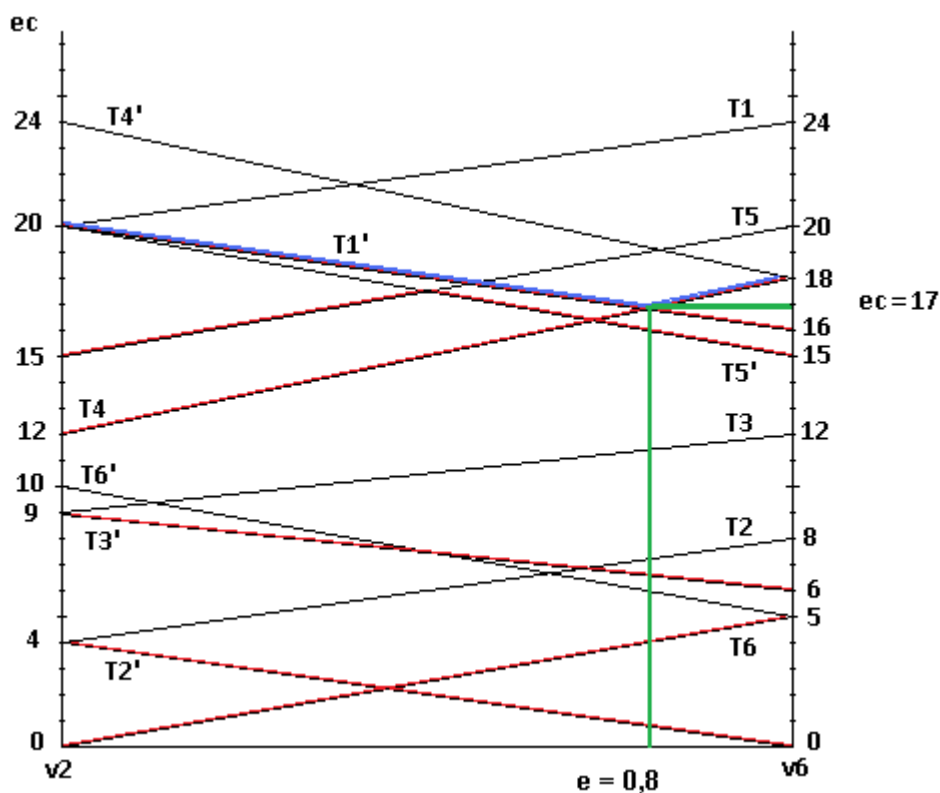
$$\begin{array}{l|l}
 p(h_4) = (v_3, v_6) & e = d(v_3, v_6) = 5 \\
 T_1 = 4 \cdot (e + 5) = 4e + 20 & T_1' = 4 \cdot (3 - e + 3) = 32 - 4e \\
 T_2 = 4 \cdot (e + 1) = 4e + 4 & T_2' = 4 \cdot (3 - e + 2) = 28 - 4e \\
 T_3 = 3 \cdot (e + 3) = 3e + 9 & T_3' = 3 \cdot (3 - e + 0) = 15 - 3e \\
 T_4 = 6 \cdot (e + 2) = 6e + 12 & T_4' = 6 \cdot (3 - e + 5) = 60 - 6e \\
 T_5 = 5 \cdot (e + 3) = 5e + 15 & T_5' = 5 \cdot (3 - e + 5) = 50 - 5e \\
 T_6 = 5 \cdot (e + 0) = 5e & T_6' = 5 \cdot (3 - e + 3) = 40 - 5e
 \end{array}$$

Minimum horního ohraničení leží na průsečíku přímek  $T_5$  a  $T_1'$  (viz obrázek 10). Z toho vyplývá, že  $5e + 15 = 32 - 4e$  a  $e = \frac{17}{9}$ . Vážená excentricita absolutního depa  $ec = e \cdot d(v_3, v_6) = 5 \cdot \frac{17}{9} = \frac{75}{9} = 9,44$ .

Obrázek 10: Diagram hodnot pro hranu  $h_4$  (zdroj: autor)

$$\begin{array}{l|l}
 p(h_5) = (v_2, v_6) & e = d(v_2, v_6) = 1 \\
 T_1 = 4 \cdot (e + 5) = 4e + 20 & T_1' = 4 \cdot (1 - e + 4) = 20 - 4e \\
 T_2 = 4 \cdot (e + 1) = 4e + 4 & T_2' = 4 \cdot (1 - e + 0) = 4 - 4e \\
 T_3 = 3 \cdot (e + 3) = 3e + 9 & T_3' = 3 \cdot (1 - e + 2) = 9 - 3e \\
 T_4 = 6 \cdot (e + 2) = 6e + 12 & T_4' = 6 \cdot (1 - e + 3) = 24 - 6e \\
 T_5 = 5 \cdot (e + 3) = 5e + 15 & T_5' = 5 \cdot (1 - e + 3) = 20 - 5e \\
 T_6 = 5 \cdot (e + 0) = 5e & T_6' = 5 \cdot (1 - e + 1) = 10 - 5e
 \end{array}$$

Minimum horního ohraničení leží na průsečíku přímek  $T_4$  a  $T_1'$  (viz obrázek 11). Z toho vyplývá, že  $6e + 12 = 20 - 4e$  a  $e = \frac{4}{5}$ . Vážená excentricita absolutního depa  $ec = e \cdot d(v_2, v_6) = 1 \cdot \frac{4}{5} = \frac{4}{5} = 0,8$ .

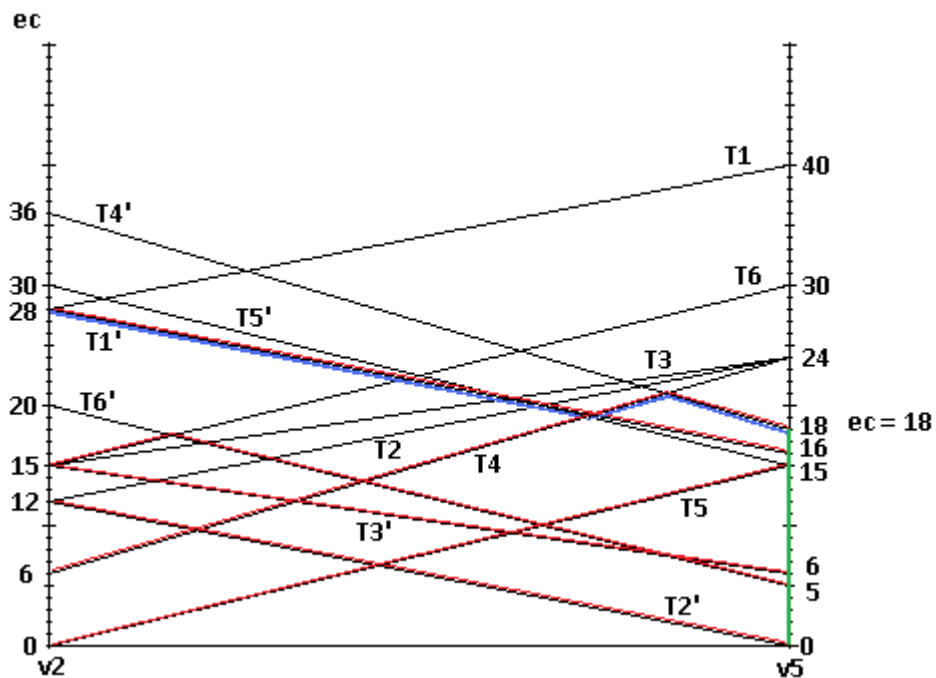
Obrázek 11: Diagram hodnot pro hranu  $h_5$  (zdroj: autor)

$$\begin{array}{l|l}
 p(h_6) = (v_2, v_5) & e = d(v_2, v_5) = 3 \\
 T_1 = 4 \cdot (e + 7) = 4e + 28 & T_1' = 4 \cdot (3 - e + 4) = 28 - 4e \\
 T_2 = 4 \cdot (e + 3) = 4e + 12 & T_2' = 4 \cdot (3 - e + 0) = 12 - 4e \\
 T_3 = 3 \cdot (e + 5) = 3e + 15 & T_3' = 3 \cdot (3 - e + 2) = 15 - 3e \\
 T_4 = 6 \cdot (e + 1) = 6e + 6 & T_4' = 6 \cdot (3 - e + 3) = 36 - 6e \\
 T_5 = 5 \cdot (e + 0) = 5e & T_5' = 5 \cdot (3 - e + 3) = 30 - 5e \\
 T_6 = 5 \cdot (e + 3) = 5e + 15 & T_6' = 5 \cdot (3 - e + 1) = 20 - 5e
 \end{array}$$

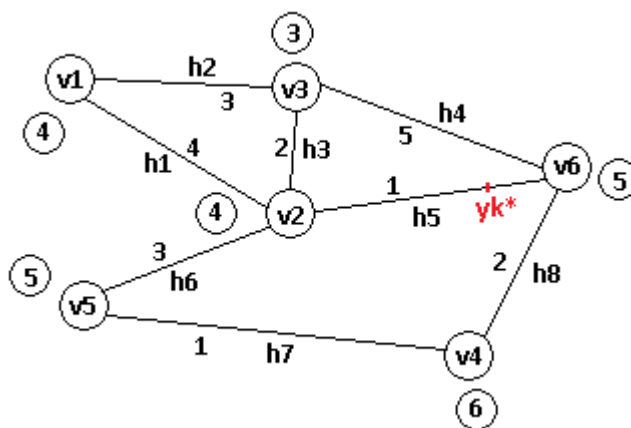
Minimum horního ohraničení leží na průsečíku přímek  $T_4$  a  $T_1'$  (viz obrázek 12). Z toho vyplývá, že  $6e + 6 = 28 - 4e$  a  $e = \frac{11}{5}$ . Vážená excentricita absolutního depa  $ec = e \cdot d(v_2, v_5) = 3 \cdot \frac{11}{5} = \frac{33}{5} = 6,6$ .

Obrázky 10, 11, 12 ukazují diagramy hodnot hran, které nesplnily podmínku  $p_{ij} \geq H$ . Absolutní depo internetového obchodu  $y_k^*$  v grafu bude umístěno na hraně  $h_5 = (v_2, v_6)$  ( $ec$  je zde nejmenší) ve vzdálenosti 0,8 od vrcholu  $v_2$  (viz obrázek 13).

Jak je vidět na tomto jednoduchém příkladu, normální způsob řešení je příliš pracný. Proto se další část práce se zabývá racionalizací optimální lokace logistického centra.



Obrázek 12: Diagram hodnot pro hranu  $h_6$  (zdroj: autor)



Obrázek 13: Umístění absolutního depa v síti (zdroj: autor)

Následující kapitola popisuje vlastní počítačovou implementaci řešení Hakimiho algoritmu s pomocí programovacího jazyka C++.

### 6.3 Počítačová implementace Hakimiho algoritmu

Jelikož je ruční výpočet optimálního umístění logistického centra pracný, zvláště velké sítě, je pro racionální řešení použita výpočetní technika. Vytvořená aplikace řeší lokaci absolutního depa v neorientované ohodnocené síti, která neobsahuje smyčky ani násobné hrany (obyčejný graf, viz kapitola 5.1).

Pomocí aplikace je řešen stejný příklad, který je uveden v předchozí kapitole. Obě metody se shodují ve výsledcích.

Velikost sítě je omezena:

- počet ohodnocených vrcholů nesmí být větší než deset,
- počet ohodnocených hran nesmí být větší než patnáct.

Dané omezení je z důvodu složitosti řešení, aplikace je přizpůsobená tomuto omezení. V síti se lokalizuje jedno středisko obsluhy. Vytvořená aplikace je naprogramovaná v jazyce C++ a pomocí knihovny Qt je vytvořeno grafické uživatelské rozhraní.

Následující podkapitola popisuje strukturu a použité vlastnosti jazyka C++. Vysvětlení použitých metod a parametrů je popsáno při výkladu aplikace (popis je vyznačen kurzívou).

#### 6.3.1 Knihovna Qt a programovací jazyk C++

Knihovna Qt je grafické uživatelské rozhraní (GUI) programovacího jazyku C++ a první zájem o její vývoj měla norská společnost Trolltech už v roce 1988. Vytvořený systém je podporovaný platformami Unix, Macintosh a Windows. Multiplatformová knihovna C++ se stala veřejně dostupná v květnu 1995. Knihovna využívá standardní C/C++ třídy a funkce. Signály, sloty a návrh aplikace patří ke klíčovým myšlenkám knihovny. Signály a sloty slouží pro komunikaci mezi objekty Qt knihovny a řeší nekompatibilitu komunikace mezi objekty na jednotlivých platformách. [1, 15]

Jazyk C++ je objektově orientovaný programovací jazyk. Byl vyvíjen od roku 1983 autorem Bjarnem Stroustrupem a dále se dokončoval v Bellových laboratořích AT&T jako rozšíření jazyka C. První oficiální forma jazyka C++ byla přijata až v roce 1998.



Podporuje několik programovacích stylů neboli paradigmat jako je procedurální programování, objektově orientované programování a generické programování. V současné době patří jazyk C++ k nejrozšířenějším na světě.

Mezi základní pojmy objektově orientovaného programování (OOP) patří *objekt*, což je entita, která má svou identitu a své vlastnosti. Každý objekt lze jednoznačně odlišit od objektu jiného. Každý objekt má nějaký vnitřní stav a nějak se chová vůči svému okolí. Každý objekt, který má být užitečný, musí poskytovat svému okolí služby. Uživatel objektu k jednotlivým objektům přistupuje jako k tak zvaným "černým skříňkám". Nezajímá se, jakým způsobem objekty služby poskytují, ale zajímá se jen, jaké služby poskytují.

Dalším pojmem OOP je *zpráva*, což je požadavek na provedení služby. Objekt nikdy neexistuje sám v nějakém vakuu. Objekt je vždy obklopen jinými objekty, kterým poskytuje svoje služby a po kterých služby požaduje. Zašle-li se objektu zpráva, nastane jeden ze dvou případů: Objekt zprávě rozumí (je schopen ji přijmout) a zareaguje na ni nějakou svou metodou. Objekt zprávě nerozumí (není schopen zprávu přijmout) a nijak nezareaguje, případně jako svou reakci oznámí uživateli, že není schopen zprávu přijmout.

Dalším důležitým pojmem v OOP je *polymorfismus* (mnohotvářost), což je vlastnost objektů, která umožňuje pojmenovat metodu jedním jménem a tato metoda může být společná pro různé objekty ve stromové hierarchii, i když pro každý objekt v této hierarchii se bude chovat různě. Jinak řečeno: polymorfismus umožňuje, aby každý objekt ve stromové hierarchii objektů volal metodu se stejným jménem. Přitom může mít každý objekt tuto metodu jinak definovanou. Výsledkem je, že pro každý objekt ve stromové hierarchii bude mít volání metody jinou odezvu, avšak pro každý objekt vždy tu správnou.

*Třída objektů* je abstrakce množiny podobných objektů. Je zřejmé, že některé objekty mají mnoho společných vlastností – to jsou stejné třídy. Každý objekt vypadá jinak, přesto jsou si nějak podobné. Všechny objekty jsou stejné třídy. Každý objekt je nějaké třídy. Je-li objekt nějaké třídy, nazývá se instancí této třídy. Každá třída může mít libovolný počet instancí. Pro popisy vztahů mezi třídami existuje mnoho druhů třídních diagramů.

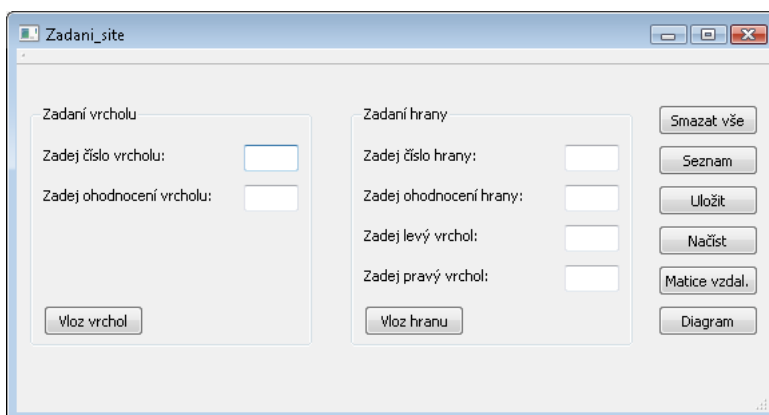
**Agregace** Jestliže třída A agreguje třídu B, potom instance třídy A v sobě obsahují instance třídy B.

**Asociace** Jestliže Třída A asociuje třídu B, potom instance třídy A nějakým způsobem musejí "vědět" o instancích třídy B.

**Dědičnost** Třída A dědí ze třídy B, jestliže třída A je podmnožinou třídy B. Třída A se nazývá podtřídou třídy B, a třída B se nazývá nadtřídou třídy A. V tomto případě instance třídy A jsou zároveň instancemi třídy B, naopak to však nemusí platit. [11, 17]

### 6.3.2 Navrhnuté grafické prostředí

Pro snadné vkládání struktury grafu a následné použití jakýchkoli grafových algoritmů je zapotřebí, uložit vrcholy, hrany a jejich ohodnocení do nějakého datového kontejneru. Pod pojmem datový kontejner se představuje něco, co má schopnost "zapamatovat si" nějaké data tzv. shromaždiště dat. V jazyku C++ existují standardní knihovny šablon (tzv. STL). Dřív než se začnu zabývat popisem použití těchto struktur, představím grafické uživatelské rozhraní, které ukládá data do datových kontejnerů. Obrázek 14 ukazuje mnou navrhnuté grafické uživatelské prostředí (GUI).



Obrázek 14: Grafické uživatelské rozhraní vytvořené aplikace

První group box s titulem *Zadání vrcholů* ohraničuje vkládání čísla vrcholů a jejich ohodnocení. Index a velikost vrcholu se vpisují do line edit. Druhý group box s titulem *Zadání hran* slouží k oddělení vkládání hrany a to indexu (čísla) hrany, její ohodnocení a vrcholů, který s ní incidují.

Dále jsou umístěna v GUI tlačítka, které spouštějí dané algoritmy. Tyto a další objekty se v knihovně Qt nazývají widgety. Každý z těchto widgetů má své přednosti a je jen na programátorovi, které si vybere. Aby bylo možné s těmito widgety pracovat, je potřeba je vhodně pojmenovat a tím jednoznačně specifikovat v kódu programu. Následuje popis widgetů line edit a tlačítek push button, které se v GUI vyskytují.

Do line editu označený labelem *Zadej číslo vrcholu*: se zadává index vrcholů. Vkládaný znak musí být celé dvojciferné číslo od 1 do 10 a vrchol nesmí již existovat, jinak aplikace upozorní na chybu.

Do dalšího line editu s popiskem *Zadej ohodnocení vrcholu*: se zadává ohodnocení vrcholu. Vkládaná hodnota musí být celé číslo a maximálně dvojciferná, jinak aplikace upozorní na chybu.

Další widget slouží ke vkládání indexu hrany, je označen popiskem *Zadej číslo hrany*:. Vkládaný znak musí být celé dvojciferné číslo od 1 do 15 a hrana nesmí již existovat, jinak aplikace upozorní na chybu.

Line edit s popiskem *Zadej ohodnocení hrany*: editu se zadává ohodnocení hrany. Vpisovaná hodnota musí být celé číslo a maximálně dvojciferná, jinak aplikace upozorní na chybu.

Vkládání celých čísel se získá tak, že se line editům nastaví vlastnost `InputMask` a jako atribut se zvolí `99`, což znamená dvou ciferné číslo, které obsahuje pouze čísla 0–9.

Widget s labelem *Zadej levý vrchol*: se zadává jeden z vrcholů incidující s hranou. Vpisovaný index vrcholu musí existovat. Stejně jako u předchozího line editu nyní s popiskem *Zadej pravý vrchol*:. Kombinace vrcholů hrany se nesmí opakovat, jinak aplikace upozorní na chybu.

Tlačítko s názvem *Vlož vrchol* umožňuje vkládat vrchol se zadanými parametry do datové struktury.

Widget s názvem *Vlož hranu* umožňuje vkládat hranu se zadanými parametry do datového kontejneru.

Push button s názvem *Smazat vše* smaže celou strukturu vrcholů a hran, nejdříve se ale uživatele zeptá, jestli chce opravdu vše smazat.

Po kliknutí na tlačítko *Seznam* se otevře nové okno, ve kterém budou zobrazeny v levé části indexy vložených vrcholů a jejich ohodnocení. Pokud se vybere určitý vrchol v levé části (označí se celý řádek), v pravé části se zobrazí vložené hrany, které obsahují vybraný vrchol (incidující hrany s tímto vrcholem), a jejich ohodnocení. Toto dialogové okno slouží k náhledu uložených vrcholů a hran (viz obrázek 15). Pro znázornění seznamu vrcholů a incidujících hran byly do okna vloženy dva table widgety. Do nich se plnily všechny vrcholy a potom i jednotlivé hrany. Názvy v záhlaví tabulky, byly vytvořeny při vkládání widgetů.

Pokud se vytvořená struktura vrcholů a hran má uložit do nějakého souboru, aby se s ní mohlo dále pracovat, může se použít widget s názvem *Uložit*. Uloží aktuálně vytvořenou strukturu do binárního souboru, který není čitelný pro jiné aplikace. Při uložení struktury

	Vrchol	Velikost
1	1	4
2	2	4
3	3	3
4	4	6
5	5	5
6	6	5

	Hrana	Velikost	Vrchol levý	Vrchol pravý
1	4	5	3	6
2	5	1	2	6
3	8	2	4	6

Obrázek 15: Okno zobrazující seznam vrcholů a hran (zdroj: autor)

se vždy přepíše již dříve uložený soubor. Pro opětovné načtení uložené struktury vrcholů a hran slouží push button s názvem *Načíst*.

K zadání matice vzdálenosti slouží tlačítko s názvem *Matice vzdal.*, který po kliknutí otevře nové okno (viz obrázek číslo 16), kam se zadává matice přímé vzdálenosti (viz kapitola 4.1.1). Pro vytvoření matice přímé vzdálenosti není použit žádný algoritmus, matice se zadává a vypočítává manuálně. Hodnoty matice se zadávají do line editů a jsou odděleny čárkou. Před hodnoty matice je zapotřebí napsat číslo příslušného vrcholu. Dialogové okno obsahuje navíc tlačítko *Ulož*, které slouží k uložení napsaných hodnot matice, a tlačítko *Načti*, které uložené hodnoty znova načte. Může se ukládat několik matic za sebou.

Zadání matice přímých vzdá...

Zadání vrcholu 1 1,0,4,3,7,7,5

Zadání vrcholu 2 2,4,0,2,3,3,1

Zadání vrcholu 3 3,3,2,0,5,5,3

Zadání vrcholu 4 4,7,3,5,0,1,2

Zadání vrcholu 5 5,7,3,5,1,0,3

Zadání vrcholu 6 6,5,1,3,2,3,0

Zadání vrcholu 7

Zadání vrcholu 8

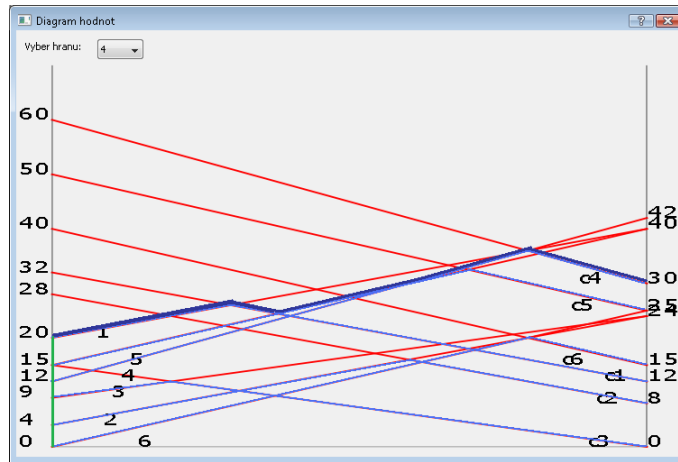
Zadání vrcholu 9

Zadání vrcholu 10

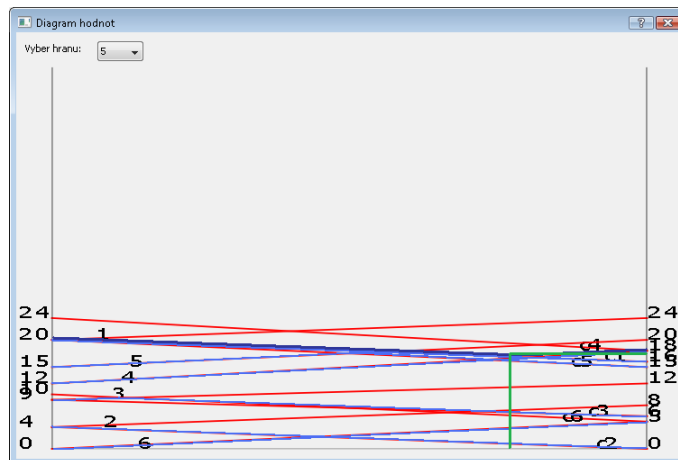
Načti Ulož Vlož hodnoty

Obrázek 16: Okno pro vložení distanční matice (zdroj: autor)

Po stisknutí tlačítka s názvem *Diagram* otevře nové okno, kde budou zobrazeny diagramy hodnot pro jednotlivé hrany grafu. Hrany lze přepínat v horní části combo boxem. Na levé i pravé svislé ose jsou napsány aktuální hodnoty bodů úseček. Jednotlivé přímky



Obrázek 17: Diagram hodnot pro hranu 4 (zdroj: autor)

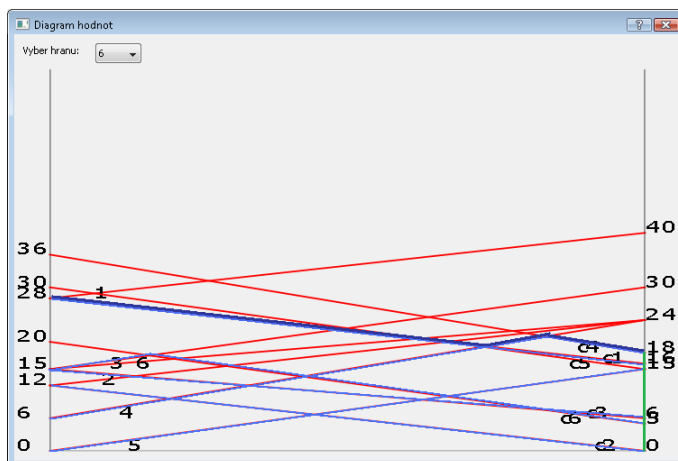


Obrázek 18: Diagram hodnot pro hranu 5 (zdroj: autor)

jsou pojmenovány. Do tohoto dialogového okna se může libovolně kreslit. Tahem myši zleva doprava se kreslí jednotlivé body, které mohou vytvářet libovolnou křivku. Opačným směrem po stejné trajektorii se jednotlivé body mažou.

Na následujících obrázcích 17, 18 a 19 jsou zobrazeny diagramy hodnot pro hranu číslo 4, 5 a 6. V diagramech jsou zvýrazněny dolní ohraničení a z těchto dolních ohraničení horní ohraničení. Jsou vybrány pouze ty hrany, které byly vybrány v předchozí kapitole. Aplikace jinak řeší všechny hrany grafu.

Do okna nebyl vložen žádný widget, kreslí se přímo na dialogové okno. Bylo zapotřebí nastavit logické souřadnice, šířku a výšku kreslicího okna. Okno má tu vlastnost, že při změně velikosti okna se mění i její nakreslené prvky (zoom).



Obrázek 19: Diagram hodnot pro hranu 6 (zdroj: autor)

### 6.3.3 Popis datových struktur aplikace

Nejdříve bylo potřeba vytvořit datovou strukturu pro ukládání vrcholů a hran. Dále byly vytvořeny struktury a třída pro zpracování vložených hodnot.

*Organizace programu C++ pro aplikace se skládá ze dvou typů souborů:*

- soubor s koncovkou *.h*, kde se deklarují struktury a třídy (u tříd se jedná pouze o deklaraci slotů, proměnných a metod), jedná se o hlavičkový soubor
- soubor s koncovkou *.cpp*, kde jsou sloty a metody definovány. Aby soubory o sobě věděly, musí se použít direktiva `#include`.

*Aplikace C++, kde se využívá knihovna Qt, existuje ještě třetí typ souboru s koncovkou *.ui*, což je grafické uživatelské rozhraní, ve kterém je definován vzhled QT aplikace (QT Designer). Tento typ souboru musí být inkudován v hlavičkovém souboru.*

*Další specifikace knihovny QT jsou následující:*

- soubory knihovny QT začínají na písmeno *q* (`#include <qapplication.h>`),
- názvy tříd (objektů) knihovny QT začínají na písmeno *Q* (*QApplication*, *QString*, *QPainter*, *QMessageBox* atd.),
- všechny třídy obsahující signály a sloty musí mít definované makro *Q\_OBJECT*,
- vstupní bod programu je standardní funkce *main()*,
- program obsahuje pouze a právě jednu instanci objektu *QApplication*, která musí být vytvořena z QT objektu jako první. [10, 17]

Knihovna QT jazyka C++ obsahuje mnoho předem naprogramovaných tříd. Tyto třídy se využívají v různých situacích a usnadňují práci programátorovi. Pokud má aplikace využívat kreslení, musí se inkludivat objekt *QPainter*. Při ukládání a načítání souborů, se musí začleňovat třída *QFileDialog*. Pokud nás aplikace informuje o chybě nebo provedení určitého algoritmu musí se inkludivat třída *QMessageBox* apod.

Pro každé nové okno, které se otevře po stisknutí jednoho z tlačítek *Seznam*, *Matice vzdal*. a *Diagram*, existují opět dva soubory, jeden s koncovkou *.h* (má v sobě inkludivané grafické prostředí *.ui*) a druhý s koncovkou *.cpp* (má v sobě inkludivaný hlavičkový soubor příslušného hlavičkového souboru).

Oddělení deklarace třídy od implementace metod chrání před nepříjemnými chybami a zaručuje efektivnost programování. Jinak by se celý kód mohl napsat do jednoho souboru *main*, což by rozsáhlejší kód činilo nepřehledným.

Signály jsou vysílány QT objekty, jsou používány při změně vnitřního stavu daného objektu a jsou vysílány, aby se příslušný QT objekt (objekty) o změně stavu dozvěděly. Signály mohou vysílat pouze třídy (a jejich podtřídy) ve kterých jsou signály definovány. Po vyvolání signálu, je slot na něj připojený vyvolán právě tak rychle, jako obyčejná funkce. Mechanismus slot/signál je absolutně nezávislý na GUI událostní smyčce. Jestliže je na jeden signál připojeno více slotů, jsou tyto spouštěny jeden po druhém, v závislosti na tom, kdy je signál vyvolán. Signály nesmí mít návratové hodnoty.

Sloty jsou přijímány QT objekty. Slot je volán, jestliže je vyvolán k němu připojený signál. Sloty jsou obyčejné C++ funkce a mohou být stejně jako funkce vyvolány (ne pouze na základě signálů). Jejich speciální vlastností je, že k nim může být připojen signál. Parametry slotů nemohou mít defoltní hodnoty. Sloty mohou být definovány virtuálně. [1, 16]

Nejdůležitější soubor, který byl vytvořen je *vrchhra.h*. V tomto hlavičkovém souboru se deklaroval datový kontejner, který v sobě ukládá vložené vrcholy, hrany a jejich ohodnocení.

První vytvořená struktura s názvem *Hrana* inicializuje a vytváří parametry, které hrana musí mít, aby se s ní mohlo dále pracovat. Mezi takové parametry patří její číslo, ohodnocení a incidující vrcholy. Dále je ve struktuře umístěn konstruktor, kopykonstruktor a inicializace proměnných ve struktuře. Ve struktuře jazyka C++ se jedná o skupinu proměnných různých typů, se kterou zacházíme jako s celkem.

Všechny proměnné a tedy i instance objektových typů by se měli hned při vytvoření inicializovat. K inicializaci dynamicky alokované paměti si používá speciální metoda, tzv. konstruktor. Také při zániku instance je potřeba uvolnit dynamicky alokovanou paměť pro

její složky. O to se také stará speciální metoda, tzv. destruktorka. Ve srovnání s ostatními metodami mají konstruktory a destruktory některá specifika. Kopy konstruktor dokáže inicializovat objekt podle vzoru realizovaného jiným, již existujícím objektem téže třídy.

Každá metoda v C++ je volána s tak zvaným implicitním parametrem *this*. Tento parametr je ukazatelem na instanci třídy, jejíž položkou metoda je. Tento ukazatel ukazuje právě na instanci, pro kterou je volán. [11]

Struktura *Hrana* je vložena do datového kontejneru knihovny STL vector (vektor). Jedná se o datový kontejner z knihovny STL jazyka C++. Vektor je šablona jednorozměrného pole. Na rozdíl od "klasického" pole má vektor, mnoho užitečných vlastností a služeb. Dále je nadefinován iterátor nad tímto vektorem. Iterátor v C++ je obdoba ukazatelů pro kontejnery, tzv. zobecněný ukazatel. Na následující prvek v kontejneru se dostane pomocí operátoru ++ u iterátoru. Pro různé kontejnery jsou iterátory různé (různě implementovány), mají ale všechny stejné rozhraní (se všemi se pracuje stejně). V kontejnerech jsou definované metody *begin()* a *end()*. Metoda *begin()* vrátí iterátor na první prvek. Metoda *end()* vrátí iterátor za poslední prvek. Prvek, na který ukazuje iterátor vrácený metodou *end()*, je neplatný (nealokován). Všechny datové kontejnery a jejich iterátory se deklarují pomocí direktivy #define.

Struktura s názvem *Vrchol* obsahuje index vrcholu a její ohodnocení. Dále je zde definován vektor, který byl definován nad strukturou *Hrana*. Nechybí konstruktor, kopy-konstruktor a inicializace proměnných ve struktuře.

Celá struktura *Vrchol* je vložena do datového kontejneru knihovny STL map. Jedná se o asociativní pole, kde jsou uloženy hodnoty ve tvaru (klíč, hodnota). Klíč je indexem prvku a může to být libovolný objekt.

Kontejner *map* má svůj vnitřní typ, jedná se o šablonu *pair*. Šablona struktury *pair* je struktura parametrizovaná dvěma typy – klíčem a hodnotou. K vytvoření "páru" se může použít konstruktor této šablony, nebo pomocí šablony funkce *make\_pair*. Ke každé hodnotě v šabloně *map* se přistupuje pomocí klíče. *Map* umožňuje mít pro jeden klíč pouze jednu hodnotu. [10, 17]

Následující obrázek 20 zobrazuje naprogramovanou datovou strukturu. Jako příklad byla použita síť v předcházející kapitole. Červenou barvou v prvním sloupci je znázorněn index vrcholu. Modrou barvou ve druhém sloupci je znázorněno ohodnocení vrcholu. V posledním sloupci jsou popsány incidující hrany s vrcholem. Červenou barvou je zobrazen index hrany, modrou barvou její ohodnocení a dvě černé číslice určují indexy incidujících vrcholů. Je patrné, že některý vrchol z těchto incidujících vrcholů je shodný s indexem vrcholu z prvního sloupce.



v	o(h)	incidující hrany
1	4	1, 4, 1, 2; 2, 3, 1, 3;
2	4	1, 4, 1, 2; 3, 2, 2, 3; 5, 1, 2, 6; 6, 3, 2, 5;
3	3	2, 3, 1, 3; 3, 2, 2, 3; 4, 5, 3, 6;
4	6	7, 1, 4, 7; 8, 2, 4, 6;
5	5	6, 3, 2, 5; 7, 1, 4, 7;
6	5	4, 5, 3, 6; 5, 1, 2, 6; 8, 2, 4, 6;

Obrázek 20: Znárodnění naprogramované datové struktury (zdroj: autor)

*Položky třídy jsou proměnné, nebo metody. Pojmu metoda se v C++ také někdy říká členská funkce (member function). Metoda je vlastně funkce, která není globální, ale je volitelná jen v souvislosti s instancemi dané třídy. Položky mohou být soukromé (private), chráněné (protected) a veřejné (public). Soukromé položky mohou být proměnné, ke kterým mají přístup pouze metody dané třídy, nebo metody, které mohou být volány jen jinými metodami dané třídy. Naopak veřejné položky může využívat kdokoli. Veřejné položky tvoří rozhraní objektu dané třídy. Soukromé položky jsou zapouzdřeny, a přístup k nim se provádí jen přes veřejné rozhraní. Jako veřejné položky by neměli být proměnné, protože by docházelo k porušení zapouzdření. Chráněné položky jsou vlastně soukromé položky, které se někdy chovají jako veřejné. Jako veřejné se chovají k tak zvaným přátelským funkcím a vůči instancím tak zvaných přátelských tříd. Použití chráněných položek je vlastně také porušení zapouzdření. [10, 17]*

Další definovaný objekt je třída *VrchHraMgr*. Tato vytvořená třída obsahuje metody, které vhodně pracují s vloženými daty. Určují, zda byly vloženy správné hodnoty, zda vrcholy existují apod.

Metoda pro přidání vrcholu *AddVrchol* je vlastně funkce, která vrací logickou hodnotu (true nebo false). Jejimi vstupními parametry jsou celočíselné konstanty *iCislo* a *iVelikost*. První konstanta vyjadřuje číslo vrcholu a druhá jeho ohodnocení.

Metoda *AddVrchol* v souboru .cpp přidává vrchol do datového kontejneru STL map. Argumenty, které přijdou do metody, musí být větší než 0 a menší nebo rovno 10. Nově vkládaný vrchol nesmí již ve struktuře existovat. Následuje vložení vrcholu do datového kontejneru STL map s danými parametry.

Logická metoda pro přidání hrany *AddHrana*. Jedná se o logickou funkci, do které vstupují celočíselné konstanty *iCislo*, *iVelikost*, *iVrchol1* a *iVrchol2*. Význam prvních dvou konstant je už znám a zbývající konstanty jsou incidující vrcholy hrany.

Metoda *AddHrana* v souboru .cpp přidává hranu do datového kontejnetu STL vector. Správnost přijatých argumentů se ověří v podmínce. Tato podmínky nepřijme parametry, které mají číslo vrcholu a jeho ohodnocení menší nebo rovno nule, incidující vrcholy jsou menší nebo rovno nule a číslo hrany je větší než patnáct. Následují podmínky a členské funkce, které ověřují, zda číslo hrany existuje, nebo zda existují stejné incidující vrcholy.

Následují metody *GetFirstVrchol*, která vždy ukazuje na první vložený vrchol, a *GetNextVrchol*, která ukazuje vždy na následující vrchol. Obě metody mají shodné vstupní argumenty a to *riCislo* a *riVelikost*. První argument je číslo vrcholu a druhý jeho ohodnocení.

Další logická metoda *GetFirstHrana* má vstupní argumenty *iVrcholCislo*, *riCislo*, *riVelikost*, *riVrchol1* a *riVrchol2*. První argument vrací první hranu požadovaného vrcholu, ostatní argumenty jsou známy.

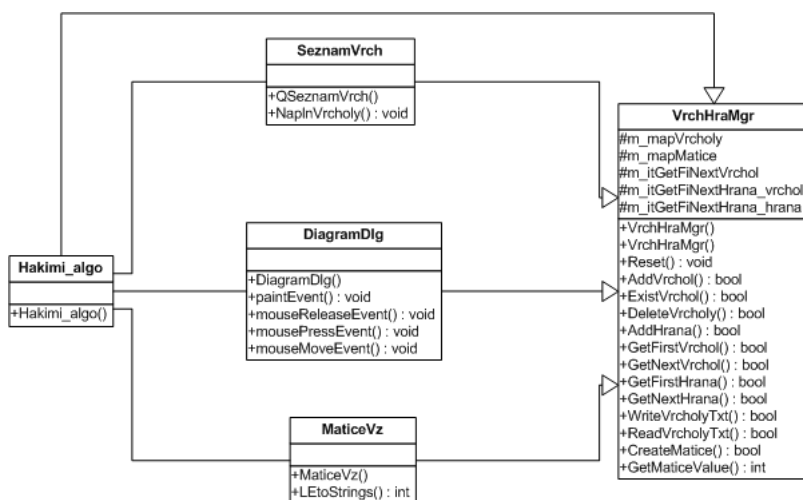
Následující logická metoda *GetNextHrana* se vstupními argumenty *riCislo*, *riVelikost*, *riVrchol1* a *riVrchol2* poskytuje následující hranu požadovaného vrcholu.

Logická metoda *WriteVrcholyTxt* ukládá námi vytvořenou strukturu do binárního souboru, který je v jiné aplikaci nečitelný. Logická metoda *ReadVrcholyTxt* načítá uloženou strukturu zpátky do aplikace. Vstup do těchto dvou metod je zadaná cesta a název souboru, do kterého se má struktura vrcholů a hran uložit. Cesta v aplikaci není žádná a název souboru je *vrcholy.bin*. Soubor se bude ukládat do adresáře, kde jsou uloženy všechny soubory aplikace.

Následující logická metoda *CreateMatice* vytvoří čtvercovou matici, ve které budou nejkratší cesty mezi jednotlivými vrcholy (viz kapitola 4.1.1, Matice vzdálenosti). Metoda nemá žádné vstupní parametry.

Poslední metoda *GetMaticeValue* má tři vstupní parametry. Vrací celočíselné ohodnocení (číslo) (*riDelka*) mezi dvěma zadanými vrcholy *iVrchol\_X*, *iVrchol\_Y*. Metoda využívá zadané matice vzdálenosti.

Obrázek číslo 21 zobrazuje dědičnost deklarovaných tříd mezi sebou. Dále ukazuje argumenty a metody, které třídy obsahují. Diagram neukazuje vstupní proměnné do jednotlivých metod, ty jsou popsány výše. Lze si představit, že hlavičkový soubor .h, soubor s definicí metod a dialogové okno jsou interpretovaný jako jedna třída. Z obrázku je patrné, že všechny třídy dědí od třídy *VrchHraMgr*. Hlavní dialogové okno je *Hakimi\_algo*, který je spojené s ostatními dialogovými okny.



Obrázek 21: Vzájemné vztahy mezi jednotlivými třídami (zdroj: autor)

## 7 Závěr

Cíle práce byly naplněny. Práce je věnována nejprve studiu lokační teorie. Zaměřuje se na obecný lokační problém a jeho některé řešící procedury. Vysvětluje spojitý lokační problém a diskrétní lokační problém, kde se práce věnuje nejpoužívanějšímu p-medián modelu. Dalším bodem práce je analýza institucí, zabývajících se vývojem nástrojů pro lokační teorii. Následuje teoretický aparát lokačního problému na síti. Jsou zde shrnuty poznatky potřebné v následující kapitole. V kapitole X je formulovaný praktický příklad. Úloha je nejprve vypočítána manuálně a následně pomocí programované aplikace.

Pro práci s vytvořenou aplikací se předpokládá znalost teorie grafů a znalost Hakimiho algoritmu. Po přečtení této práce budou čtenáři známy všechny okolnosti pro správnou práci s aplikací a řešení daného problému.

Při srovnání těchto dvou metod řešení (ruční a naprogramované řešení) je efektivnější použít vytvořenou aplikaci z následujících důvodů:

- řešení je časově rychlejší,
- neobsahuje náhodné chyby a omyly,
- grafické zobrazení je přesnější,
- možnost řešit složitější úlohy.

Další velkou výhodou aplikace je, že se naprogramovaný datový kontejner (shromážděně vrcholů a hran) dá použít na jiné grafové algoritmy a lokační úlohy. Těmito úlohami může být například:

- hledání nejkratší cesty,
- hledání nejspolehlivější cesty,
- hledání cesty s maximální kapacitou,
- lokační úloha s umístěním depa ve vrcholu, atd.

Možné zlepšení práce může být například takové, že se vymyslí a aplikuje algoritmus na vytvoření matice přímé vzdálenosti. Nyní je aplikace přizpůsobená na ruční vložení matice přímé vzdálenosti, neboť je algoritmus pro vytvoření této matice značně složitý.

Ve vytvořených diagramech hodnot si musí uživatel sám vyznačit dolní ohraničení, z vybraného dolního ohraničení stanovit horní ohraničení a z nich stanovit minimum pro určení umístění absolutního depa. Vytvořená aplikace je tedy interaktivní. Možné zlepšení by spočívalo v tom, že by aplikace automaticky vyhodnotila umístění absolutního depa.

Grafické uživatelské rozhraní vytvořené aplikace by se mohlo učinit pro uživatele lépe ovladatelným. V diagramech hodnot by se mohly automaticky přizpůsobovat maximální hodnoty na vertikálních osách.

## 8 Seznam zdrojů

### 8.1 Seznam knižních zdrojů

1. BLANCHETTE, Jasmin; SUMMERFIELD, Mark. *C++ GUI Programming with Qt 4*. Sebastopol : Safari Books Online, 2006. 537 s. ISBN 0-13-187249-4.
2. BIČÍK, Ivan, JÁNSKÝ, Bohumír. *Příroda a lidé Země*. Praha : Nakladatelství České geografické společnosti, s. r. o., 2004. 135 s. ISBN 80-86034-45-3.
3. DEMEL, Jiří. *Grafy a jejich aplikace*. Praha : Academia, 2002. 257 s. ISBN 80-200-0990-6.
4. DREZNER, Zvi, HAMACHER, Horst W. *Facility location : applications and theory*. Berlin : Springer, 2001. 457 s. ISBN 3-540-21345-7.
5. KOLÍNSKÝ, Bohuslav. *Lokace intermodálních terminálů v ČR v návaznosti na evropské dopravní síť*. Pardubice, 2009. 145 s. Diplomová práce. Univerzita Pardubice.
6. KOSIUR David, *Elektronická komerce : principy a praxe*. Brno : Computer Press, 1998. 267 s. ISBN 80-7226-097-9.
7. NICKEL, Stefan, PUERTO, Justo. *Location theory : a unified approach*. Berlin : Springer, 2005. 437 s. ISBN 3-540-24321-6.
8. PERNICA, Petr. *Logistika : vymezení a teoretické základy*. Praha : Vysoká škola ekonomická, 1994. 210 s. ISBN 80-7079-820-3.
9. PERNICA, Petr. *Logistika : aktivní prvky*. Praha : Vysoká škola ekonomická, 1994. 345 s. ISBN 80-7079-808-4.
10. PRATA, Stephen. *Mistrovství v C++*. Brno : Computer Press, 2007. 1120 s. ISBN: 978-80-251-1749-1.
11. VIRIUS, Miroslav. *Programování v C++*. Praha : ČVUT, 2001. 364 s. ISBN: 80-01-01874-1.
12. VITOVSKÝ, Antonín. *Moderní slovník softwaru*. Praha : AV software, 2006. 588 s. ISBN 80-901428-8-5.
13. VOLEK, Josef. *Operační výzkum I*. Pardubice : Univerzita Pardubice, 2005. 111 s. ISBN 80-7194-410-6.

## 8.2 Seznam internetových zdrojů

- 14 *About INFORMS* [online]. 2010 [cit. 2010-02-19]. INFORMS online. Dostupný z WWW: <<http://www.informs.org/>>.
- 15 EZUST, Alan, EZUST, Paul. *Introduction to Design Patterns in C++ with Qt4* [online]. 2009 [cit. 2010-02-19]. Introduction to Design Patterns in C++ with Qt4. Dostupný z WWW: <<http://cartan.cas.suffolk.edu/oopdocbook/opensource/index.html>>.
- 16 *Dopravní infrastruktura – časové řady* [online]. 2008 [cit. 2010-02-23]. Český statistický úřad. Dostupný z WWW: <[http://www.czso.cz/csu/redakce.nsf/i/dopravni\\_infrastruktura\\_casove\\_rady](http://www.czso.cz/csu/redakce.nsf/i/dopravni_infrastruktura_casove_rady)>.
- 17 DOSTÁL, Radim. *Builder* [online]. 2002 [cit. 2010-03-02]. Informační server o programování. Dostupné z WWW: <<http://www.builder.cz/serial24.html>>.
- 18 *Koncepce veřejných logistických center*. Logistic News [online]. 2009, 5, [cit. 2010-04-22]. Dostupný z WWW: <[www.logisticnews.cz/files/uploaded/UserFiles/pdf2009/05\\_2009/str\\_2\\_7.pdf](http://www.logisticnews.cz/files/uploaded/UserFiles/pdf2009/05_2009/str_2_7.pdf)>.
- 19 *LoLA* [online]. 2001 [cit. 2010-02-19]. LoLA. Dostupný z WWW: <<http://www.mathematik.uni-kl.de/~lola/>>.
- 20 *Nákladní doprava – časové řady* [online]. 2008 [cit. 2010-02-23]. Český statistický úřad. Dostupný z WWW: <[http://www.czso.cz/csu/redakce.nsf/i/nakladni\\_doprava\\_casove\\_rady](http://www.czso.cz/csu/redakce.nsf/i/nakladni_doprava_casove_rady)>.
- 21 PASTOR, Otto; WANSLEY, Michaela. *Logistická centra* [online]. 1991 [cit. 2010-02-28]. ČVUT - Fakulta dopravní. Dostupné z WWW: <[http://studium.fd.cvut.cz/html/logisticka\\_centra.html](http://studium.fd.cvut.cz/html/logisticka_centra.html)>.
- 22 *Scope and purpose* [online]. 2005 [cit. 2010-02-19]. EURO Working Group on Locational Analysis. Dostupný z WWW: <<http://www.vub.ac.be/EWGLA/homepage.htm>>.
- 23 *SoLA Online* [online]. 2007 [cit. 2010-02-23]. Dostupný z WWW: <<http://location.section.informs.org/>>.
- 24 *Timur* [online]. 2010 [cit. 2010-04-12]. Dostupný z WWW: <<http://www.timur.cz/>>.