

**VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ
UNIVERZITA OSTRAVA**

**Hornicko-geologická fakulta
Institut geoinformatiky**

**Implementace algoritmů prohledávání
dopravní sítě v prostředí PostgreSQL**

diplomová práce

Autor:

Jan Kolář

Vedoucí diplomové práce:

Ing. Jan Růžička, Ph.D.

Ostrava 2006

VŠB – TECHNICAL UNIVERSITY OF OSTRAVA

**Faculty of Mining and Geology
Institute of Geoinformatic**

**Implementation of searching algorithms in
traffic net with PostgreSQL**

thesis

Author:

Jan Kolář

Supervisor:

Ing. Jan Růžička, Ph.D.

Ostrava 2006

Anotace diplomové práce

Tato práce se zabývá vyhledávacími algoritmy v dopravní síti za využití databázového systému PostgreSQL/PostGIS. Cílem práce je implementace algoritmu hledání optimální cesty.

Práce je členěna do několika dílčích částí. V první části, 1. až 4. kapitola, je vymezena sledovaná problematika, jsou zde stanoveny cíle, postup práce a teoretické předpoklady.

Druhá část práce, 5 a 6. kapitola, popisuje a hodnotí možné datové zdroje a použité programové vybavení.

Třetí část práce, 7. kapitola, je praktická a zabývá se prací s databázovým systémem PostgreSQL a implementací řešení hledání optimální cesty s využitím JAVA knihovny JGraphT. Tato část popisuje návrh databáze, vnitřní funkci v PL/pgSQL pro přípravu dat a hledání cesty pomocí Dijkstraova algoritmu.

Poslední část, 9. kapitola a dále, je tvořena závěrem, zhodnocením výsledků diplomové práce a seznamem použitých zdrojů.

Klíčová slova: hledání optimální cesty, Dijkstrův algoritmus, PostgreSQL, JGraphT, StreetNet ČR

Annotation of thesis

This thesis is engaged in searching algorithms of traffic net with use of database system PostgreSQL. The main objective of the thesis is implementation of pathfinding algorithm.

The thesis is divided into several parts. In the first part, from first up to fourth chapter, objectives, working process and hypothesis are defined.

The second part, the fifth and sixth chapter, describes and evaluates possible data sources and used software.

The third part, the seventh chapter, is practical and concerns about work with database system PostgreSQL, and implementation of pathfinding with use of JAVA library JGraphT. This part also describes design of database, PL/pgSQL function for data preparation and pathfinding with Dijkstra's Algorithm.

The last part, the ninth chapter and further, contains conclusion, evaluation of thesis results and list of used information sources.

Keywords: pathfinding, Dijkstra's Algorithm, PostgreSQL, JGraphT, StreetNet ČR

Prohlášení

- *Celou diplomovou práci včetně příloh, jsem vypracoval samostatně a uvedl jsem všechny použité podklady a literaturu.*
- *Byl jsem seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č.121/2000 Sb. - autorský zákon, zejména § 35 – využití díla v rámci občanských a náboženských obřadů, v rámci školních představení a využití díla školního a § 60 – školní dílo.*
- *Beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou (resp. bakalářskou) práci užít (§ 35 odst. 3).*
- *Souhlasím s tím, že jeden výtisk diplomové práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o diplomové práci, obsažené v Záznamu o závěrečné práci, umístěném v příloze mé diplomové (resp. bakalářské) práce, budou zveřejněny v informačním systému VŠB-TUO.*
- *Rovněž souhlasím s tím, že kompletní text diplomové práce bude publikován v materiálech zajišťujících propagaci VŠB-TUO, vč. příloh časopisů, sborníků z konferencí, seminářů apod. Publikování textu práce bude provedeno v omezeném rozlišení, které bude vhodné pouze pro čtení a neumožní tedy případnou transformaci textu a dalších součástí práce do podoby potřebné pro jejich další elektronické zpracování.*
- *Bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst.4 autorského zákona.*
- *Bylo sjednáno, že užít své dílo – diplomovou (resp. bakalářskou) práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).*

V Ostravě dne 12.4.2005

Jan Kolář

Adresa trvalého pobytu diplomanta:

Pivovarská 3183
Mělník
27601

Obsah

1. ÚVOD	1
2. CÍLE A VÝCHODISKA	2
3. POSTUP PRÁCE	3
4. VYSVĚTLENÍ POJMŮ	4
4.1 HLEDÁNÍ OPTIMÁLNÍ CESTY	4
4.2 PATHFINDING	5
4.3 TEORIE GRAFŮ.....	5
4.4 SÍŤOVÉ ANALÝZY	8
4.5 DIJKSTRŮV ALGORITMUS	9
5. DATA	13
5.1 OBECNÉ POŽADAVKY NA DATA	13
5.2 PŘÍPRAVA DAT PRO SÍŤOVÉ ANALÝZY	14
5.3 MOŽNÉ ZDROJE DAT	15
5.3.1 ArcČR 500.....	15
5.3.2 Česká republika 1 : 50 000 - silniční síť (T-Mapy)	16
5.3.3 DMÚ-25.....	16
5.3.4 Silniční databanka Ostrava.....	17
5.3.5 StreetNet	18
5.4 VHODNOST DATOVÉ SADY STREETNET PRO SÍŤOVÉ ANALÝZY	19
6. PROGRAMOVÉ PROSTŘEDKY	22
6.1 POSTGRESQL 8.1	22
6.1.1 Procedurální jazyk pgSQL.....	23
6.1.2 PostGIS 1.1.0	23
6.2 ARCVIEW 9.0	24
6.3 JGRAPH-T 0.6.0.....	25
6.4 JBUILDER 2005 - FOUNDATION	26
6.5 QUANTUM GIS	26
7. NÁVRH ŘEŠENÍ	27
7.1 PRÁCE S DATABÁZOVÝM SYSTÉMEM POSTGRESQL.....	28
7.1.1 Návrh databáze.....	28
7.1.2 Import dat.....	29
7.2 PŘÍPRAVA DAT.....	30
7.2.1 Vnitřní funkce pro naplnění tabulky hran a uzlů	30
7.3 IMPLEMENTACE ŘEŠENÍ HLEDÁNÍ OPTIMÁLNÍ CESTY.....	33
7.3.1 Vytvoření grafu - verze první.....	33
7.3.2 Vytvoření grafu - verze druhá	34
7.3.3 Nalezení optimální cesty pomocí Dijkstraova algoritmu.....	36
7.3.4 UML diagramy.....	39
8. ERDIS	42
9. ZÁVĚR	43
SEZNAM POUŽITÝCH ZDROJŮ	45

SEZNAM ZKRATEK

České zkratky

ČR	Česká republika
DMÚ 25	Digitální model území
eRDIS	Redakční Dopravní Informační Systém
GIS	geografický informační systém
HGF	Hornicko-geologická fakulta
OS	operační systém
S-JTSK	Systém jednotné trigonometrické sítě katastrální
VŠB-TU	Vysoká škola báňská – Technická univerzita

Cizojazyčné zkratky

CAD	Computer Aided Design
CEDA	Central European Data Agency
ESRI	Environmental System Research Institute
J2SE	Java 2 Standard Edition
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
LGPL	Lesser General Public License
PL/pgSQL	Procedural Language/PostgreSQL Structured Query Language
RDBMS	Relational Database Management System
Shapefile	formát vektorových dat vytvořený společností ESRI
SHP	přípona souboru ESRI shapefile
SQL	Structured Query Language
UML	Unified Modeling Language
UMN	University of Minnesota
WGS84	World Geodetic System 1984

1. Úvod

Mezi nejpoužívanější služby internetových portálů patří aplikace umožňující nalezení optimální cesty. Ta se většinou vyhledává jako prostorově nejkratší či časově nejmíň náročná trasa mezi dvěma zastávkami. Výstupem pak bývá většinou rastrová mapa s vykreslenou cestou, někdy i doplněná o výpis seznamu zastávek.

V rámci této diplomové práce je navrženo vlastní řešení hledání optimální cesty vhodné pro využití v dopravním zpravodajství. Projekt byl zadán a v průběhu práce konzultován firmou CAD programy – Ing. Jan Vlčinský, tzn. výsledky najdou praktické uplatnění.

Oproti řešením dostupným na internetu je u koncové aplikace požadována rozšířená funkcionalita. Předpokládá se například možnost zadání úseku, kde projetí není možné, případně je změněno jeho ohodnocení na základě aktuálních informací o dopravní situaci.

Úkolem práce není vytvořit aplikaci určenou pro koncového uživatele. Výsledky, tzn. připravená data, návrh databáze a zdrojový kód algoritmu, budou využity v budoucích projektech firmy. Prvním projektem, který využil této práce, byl systém eRDIS - Redakční Dopravní Informační Systém.

Z toho, že projekt byl zadán soukromou firmou, vyplývala jistá omezení. Firma například jednoznačně preferovala řešení založené na Open source databázovém systému PostgreSQL, za vstupní data i přes jejich diskutabilní kvalitu byla zvolena datová sada Streetnet od společnosti CEDA. Důvodem byla znalost těchto produktů a vazby na stávající zákazníky.

2. Cíle a východiska

Mezi hlavní cíle této práce především patří:

- implementace algoritmu pro hledání optimální trasy na základě zadaných parametrů
- navrhnout databázi a vytvořit vnitřní funkce pro úpravu dat do podoby vhodné k vytvoření síťového grafu
- zhodnotit dostupné datové zdroje
- jako úložiště prostorových dat využít databázový systém PostgreSQL a jeho extenzi PostGIS.

Využití se předpokládá následující:

- zobrazení dopravních informací na zvolené trase
- nalezení optimální cesty na základě zadaných parametrů (např.: typ komunikace, dopravní události, doba jízdy, maximální délka trasy, a další)

Cílem projektu je zejména implementace algoritmu pro vyhledávání nejkratší cesty. K tomu byla schválena možnost využití opensource knihovny JGgraphT uvolněné pod licencí GNU LGPL. Zadavatelská firma má bohaté zkušenosti s PostgreSQL, jako úložiště dat bylo tedy podmíněno použít tento databázový systém.

3. Postup práce

Tento celoroční projekt, jenž je současně diplomovou prací, byl zadán ostravskou firmou CAD programy – Ing. Jan Vlčinský, s kterou byl další postup průběžně konzultován.

Nejdříve, v **říjnu 2005**, bylo na schůzce s představiteli firmy domluveno téma práce, cíle a prostředky, jež mají být využity pro vytvoření funkční aplikace.

V **průběhu listopadu 2005** bylo provedeno vyhodnocení dostupných dat vhodných pro síťové analýzy. Po konzultaci ve škole a ve firmě pak byla vybrána data, která budou využita pro vyhledávání. Při výběru dat byla rozhodující vazba na stávající zákazníky firmy CAD-programy.

V **prosinci 2005** byl vytvořen návrh databáze a proveden import dat do prostředí PostgreSQL. Dále byla vytvořena vnitřní funkce pro nezbytné úpravy dat.

V **průběhu ledna 2006** byl napsán JAVA kód pro vytvoření síťového grafu na základě dat z databáze a vyhledání optimální cesty.

Vlastní dokument diplomové práce byl vytvořen v **únoru až dubnu 2006**.

4. Vysvětlení pojmů

Pro řešení úlohy hledání optimální cesty se ve většině případů využívá teorie grafů, která je dnes samostatnou vědeckou disciplínou tvořenou souborem poznatků a metod, které vznikly při zkoumání praktických úloh a byly později zobecněny a doplněny.

K lepšímu pochopení tématu diplomové práce je vhodné vysvětlit některé pojmy z teorie grafů a objasnit aspoň některé algoritmy používané pro hledání optimální cesty.

4.1 Hledání optimální cesty

Za hledání optimální považujeme především [25]:

- § hledání libovolné cesty,
- § nejkratší cesty, tzn. cesty s nejmenším počtem hran případně s nejmenším počtem přesedání,
- § nejlevnější cesty, tzn. cesty s nejmenším součtem ohodnocení hran.

Je možné vyhledávat i větší počet cest splňujících zadaná kritéria. Může se jednat například o úlohu nalezení všech cest z místa A do B kratších než zadaná vzdálenost neprocházejících daným segmentem, kdy praktické využití může být například stanovení objízdne trasy.

Hledaná optimální cesta pak může být definována:

- § dvojicí zadaných uzlů,
- § ze zadaného uzlu do všech ostatních,
- § ze všech ostatních do zadaného koncového uzlu,
- § mezi všemi (uspořádanými) dvojicemi uzlů.

4.2 Pathfinding

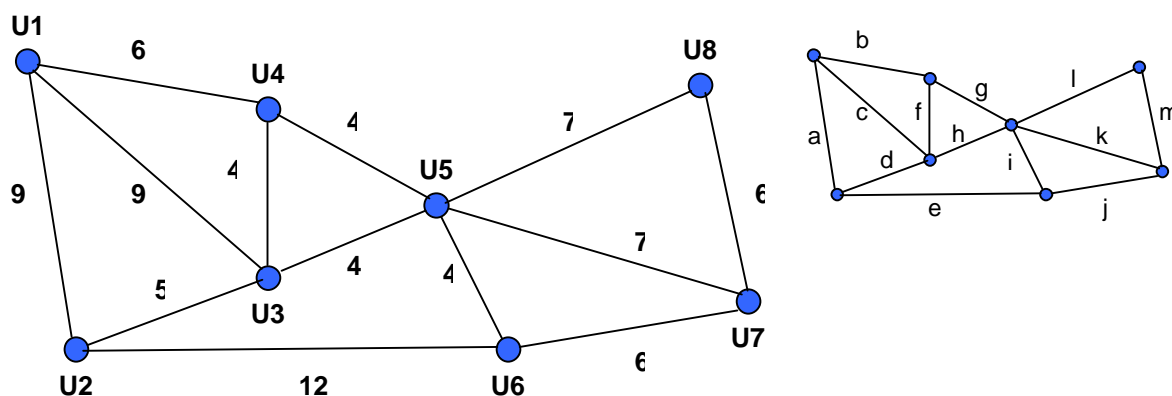
Na mezinárodní úrovni je poměrně často používaným pojmem Pathfinding, který lze chápat zhruba na stejné úrovni jako pojem hledání optimální cesty. Problémem hledání cesty (*pathfinding*) se rozumí úloha nalezení nejvhodnější cesty dle daných kritérií z místa A do místa B v definovaném prostoru (obvykle grafu). [30]

K řešení problému se využívají síťové analýzy grafů. Využití se uvádí v zejména v aplikacích pro hledání optimální cesty v dopravních sítích, ale také v oblasti počítačových her.

4.3 Teorie grafů

Teorie grafů představuje dnes již samostatně rozvinutou matematickou disciplínu. Její terminologie není všeobecně známá ani jednotná, je lepší uvést a vysvětlit některé pojmy.

Teorie grafů zkoumá vlastnosti struktur, zvaných grafy, které se definují jako uspořádaná dvojice $G = (V, E)$, kde V je množina vrcholů a E je množina hran, což je množina vybraných dvouprvkových podmnožin množiny vrcholů. Grafy se znázorňují obvykle jako množina bodů (vrcholy) spojených liniemi (hrany). [31]



Obr.1 – souvislý, neorientovaný, ohodnocený graf

Pomocí grafů lze reprezentovat struktury a úlohy z nejrůznějších oborů. Struktura grafu může být rozšířena o ohodnocení hran nebo vrcholů. Takové ohodnocení se často označuje jako váha a může reprezentovat délku, náklady na přesun, průchodnost a podobně. Výsledkem jsou modely reálné sítě, které se používají pro analýzu dopravy (např. právě pro hledání optimální cesty) nebo počítačových sítí.

Tab. č.1 – Vysvětlení pojmů z teorie grafů

Pojem	Popis
Graf	uspořádaná dvojice $G = (V,E)$, kde V je množina vrcholů a E je množina hran
Hrana	množina vybraných dvouprvkových podmnožin množiny vrcholů
Neorientovaná hrana	hrana bez vyznačení směru (bez orientace)
Orientovaná hrana	hrana s vyznačením směru
Smyčka	hrana, která vychází a končí ve stejném vrcholu
Sled	uspořádaná posloupnost vrcholů a hran
Tah	sled, ve kterém se každá hrana vyskytuje nejvýše 1x
Cesta	tah, ve kterém se každý uzel vyskytuje nejvýše 1x
Kružnice	uzavřená cesta
Orientovaný graf	obsahuje pouze orientované hrany.
Ohodnocený graf	hranám jsou přiděleny hodnoty větší než nula
Souvislý graf	mezi libovolnými uzly existuje sled
Multigraf	Mezi dvěma vrcholy lze vést libovolný počet hran, některé mohou být orientované. Může obsahovat smyčky.
Síť	orientovaný graf s ohodnocenými hranami a vrcholy

Pro popis grafů a optimalizaci v grafech se používají maticové nebo relační struktury. Především jsou to: [8]

- a) matice sousednosti** (někdy i matice spojitosti) zapisuje počet hran mezi 2 různými vrcholy grafu. Prvky této matice jsou čísla 0 nebo 1 u grafu a obecně celá čísla u multigrafu. Prvek matice a_{ij} je roven počtu hran vedoucích

z vrcholu i do vrcholu j grafu. Matice je čtvercová, řádky a sloupce odpovídají vrcholům grafu.

Tab. č.2 – Matice sousednosti (graf viz obr. 1)

		UZLY							
		1	2	3	4	5	6	7	8
U	1	0							
	2	1	0						
Z	3	1	1	0					
	4	1	2	1	0				
L	5	2	2	1	1	0			
	6	3	1	2	2	1	0		
Y	7	3	2	2	2	1	1	0	
	8	3	3	2	2	1	2	1	0

b) matice sousednosti hran. V této matici řádky a sloupce odpovídají hranám grafu a prvky matice a_{ij} popisují počet vrcholů, které jsou incidentní hranám i a j .

c) matice sousednosti vrcholů a hran (matice incidence). Do této matice se zapisuje existence spojení mezi uzly a hranami (incidence). V této matici řádky odpovídají vrcholům grafu a sloupce hranám. Prvky matice nabývají hodnot 0 nebo 1. Prvek je roven jedné, když odpovídající vrchol je incidentní odpovídající hraně.

Tab. č.3 – Matice incidence (graf viz obr. 1)

		HRANY												
		a	b	c	d	e	f	g	h	i	j	k	l	m
U	1	1	1	1	0	0	0	0	0	0	0	0	0	0
	2	1	0	0	1	1	0	0	0	0	0	0	0	0
Z	3	0	0	0	1	1	1	0	0	0	0	0	0	0
L	4	0	1	0	0	0	1	1	0	0	0	0	0	0
Y	5	0	0	0	0	0	0	1	1	1	0	1	1	0
	6	0	0	0	0	1	0	0	0	1	1	0	0	0
	7	0	0	0	0	0	0	0	0	0	1	1	0	1
	8	0	0	0	0	0	0	0	0	0	0	0	1	1

d) matice vzdáleností vrcholů grafu. Do modelu umožňuje zaznamenat další doplňitelné informace. Jednou z těchto informací může být vzdálenost

mezi vrcholy, kterou získáme pomocí ohodnocení každé hrany grafu číselnou hodnotu, např. její délkou.

e) **matice dosažitelnosti** obsahuje prvky a_{ij} , které nabývají hodnoty 1, pokud existuje sled mezi uzly U_i a U_j .

f) **odbočovací matice** u rozsáhlých grafů (např. silniční síť) může být vhodné znát, zda se lze volně pohybovat mezi jednotlivými hranami nebo zda existují bariéry pro přechod. Odbočovací matice popisuje možnost spojení (odbočení) mezi hranami nebo mezi hranami a uzly.

4.4 Síťové analýzy

Síťové analýzy jsou analytické operace prováděné nad liniovou vrstvou v rastrové či vektorové podobě. Takováto vrstva nejčastěji představuje model dopravní sítě. Vektorový model dopravní sítě (označovaný jako síť, graf) tvoří určité množství prvků sítě: segmenty sítě, uzly sítě, zastávky, centra, odbočky.

Segmenty sítě (hrany) jsou spojnice jejich počátečního a koncového uzlu. Na této spojnici se mohou nacházet i další body, označované jako vrcholy. Segmenty dopravní sítě můžeme v prostředí GIS chápat jako znázornění objektů reálného světa, například silniční a uliční úseky, železniční sítě, inženýrské sítě a jiné. Zastávky jsou taková místa v dopravní síti, přes které musí povinně procházet hledaná cesta.

Nejběžnějšími síťovými analýzami užívanými pro zjišťování dopravní dostupnosti území jsou úlohy nazvané hledání cesty a alokace zdrojů. Cílem úlohy hledání cesty je nalézt cestu mezi dvěma místy. Pro hledání cesty v dopravních sítích se využívají algoritmy jako např.: [1]

- § prohledávání do šířky
- § prohledávání do hloubky
- § Dijkstrův algoritmus
- § prohledávání čistě podle heuristiky
- § Dijkstrův algoritmus s heuristikou (A^*)

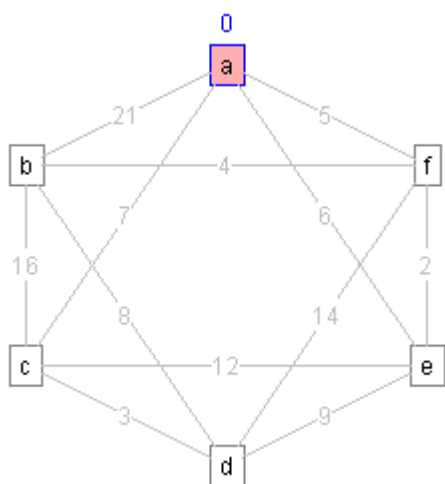
4.5 Dijkstrův algoritmus

Nejrozšířenějším algoritmem pro hledání optimální cesty je zřejmě Dijkstrův algoritmus, a to zejména díky menší časové náročnosti vyhledávání a jeho jednoduchosti. Charakteristické pro algoritmus je, že postupuje v každém směru tak rychle jak obtížný je terén, přitom jsou ale prohledávány i oblasti odvrácené od cíle.

Algoritmus rozděluje vrcholy grafu do dvou skupin, na vrcholy s trvalým ohodnocením a dočasným ohodnocením. Ohodnocením vrcholu zde rozumíme délku cesty (součet ohodnocení hran) od počátku k danému vrcholu. Trvalé ohodnocení je takové, které už nebudeme měnit, protože lepšího ohodnocení nemůžeme dosáhnout. Postup algoritmu můžeme shrnout do tří kroků : [2]

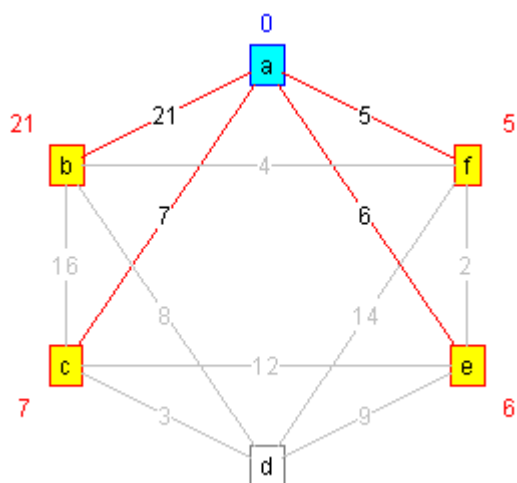
- 1) nalezení vrcholu X s minimálním dočasným ohodnocením
- 2) prohlášení vrcholu X za trvalý
- 3) změna ohodnocení sousedů tak, že $|S| = \min(|S|, |X| + \text{ohodnocení hrany z X do S})$, kde S je soused X.

Na začátku jsou všechny vrcholy označeny jako dočasné, jejich hodnota je nekonečno s výjimkou počátečního vrcholu, jenž má hodnotu nula. Algoritmus dále vstupuje do cyklu, který prochází body 1,2 a 3 do doby, než mají všechny vrcholy trvalé ohodnocení nebo do doby, kdy je jako minimální dočasný vrchol nalezen takový, který má dočasné ohodnocení nekonečno (tzn. z počátku do cíle cesta neexistuje).



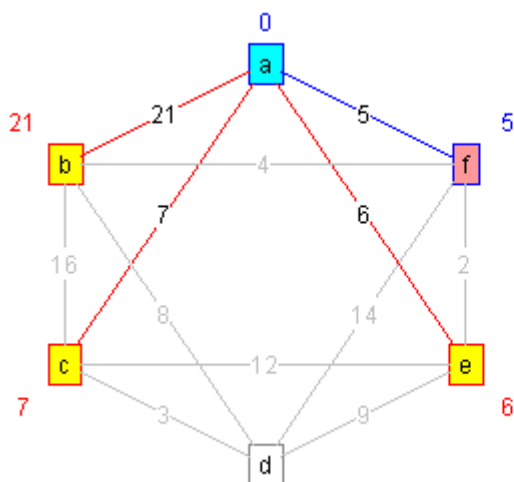
V prvním a druhém kroku je nalezen počáteční vrchol (hodnota nula) a prohlášen za trvalý. Poté jsou nalezeni všichni jeho sousedé. Ti jsou dočasní (hodnota nekonečno) a jejich nová hodnota se rovná dle třetího bodu délce hrany do nich vedoucí a začíná druhá iterace cyklu.

Obr. 2 – začátek algoritmu (1. krok)

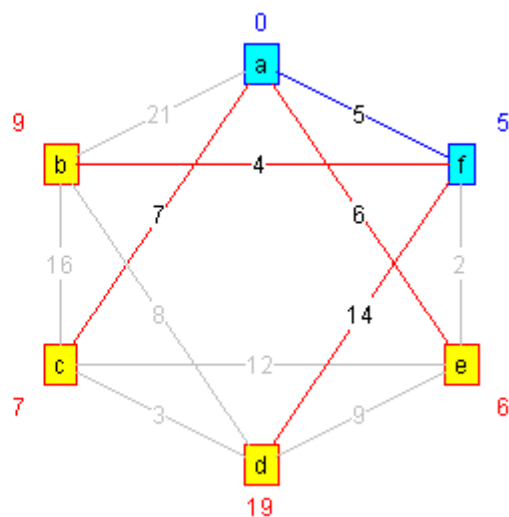


Obr. 3 – začátek algoritmu (2. a 3. krok)

Algoritmus najde vrchol s minimálním dočasným ohodnocením (jeden z právě přečíslovaných vrcholů, protože ostatní mají hodnotu nekonečno) a prohlásí ho za trvalý (druhý krok).

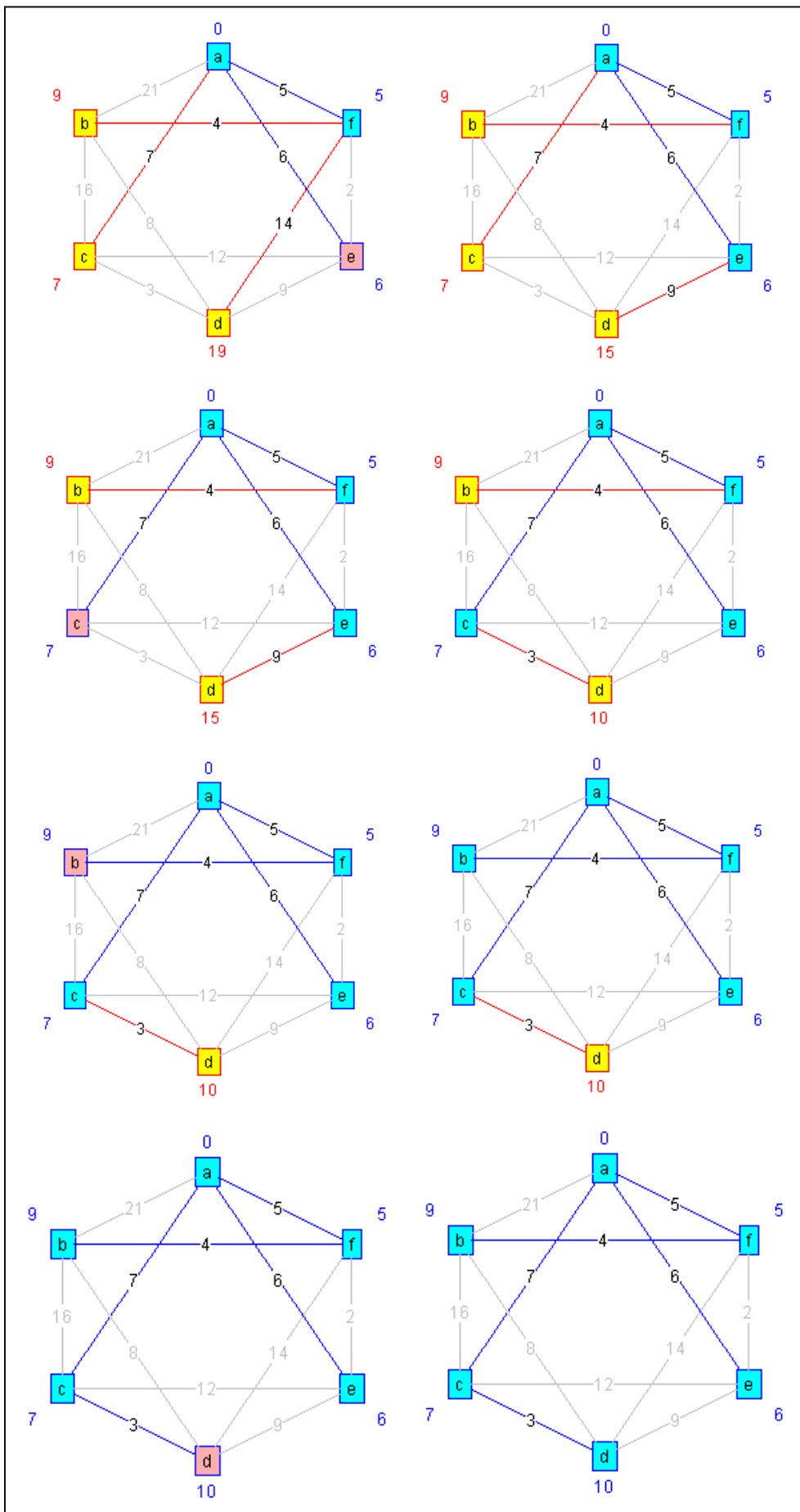


Obr.4 – 1. krok algoritmu



Obr.5 – 2. a 3. krok algoritmu

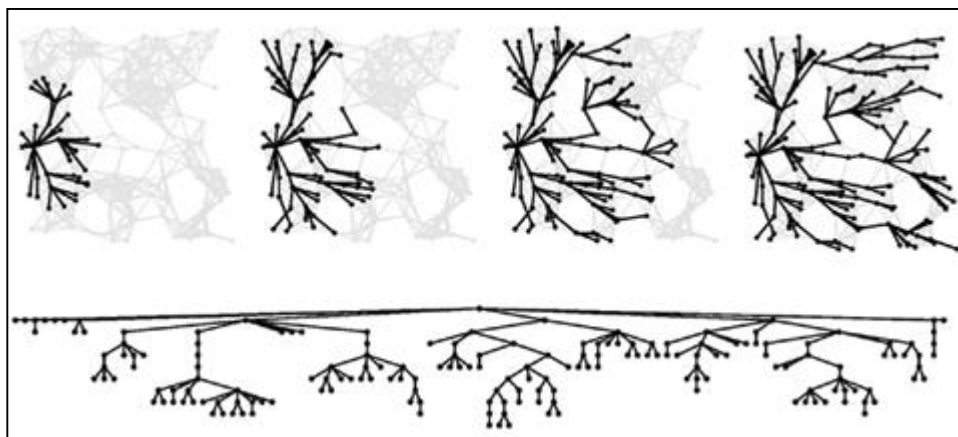
Jeho sousední vrcholy se v případě nutnosti (tj. podle bodu 3) přehodnotí, jelikož v některé iteraci se může stát, že do sousedního vrcholu se dá dostat po cestě s menším ohodnocením (již je taková cesta nalezena), než by byla cesta vedoucí přes vrchol, který je právě teď minimální. Takto algoritmus postupuje dále.



Obr. 6 – dokončení algoritmu a nalezení délky nejkratší cesty do vrcholu d

Dijkstrův algoritmus je vždy konečný. Označíme-li počet vrcholů grafu jako N , je celkový počet kroků potřebných k nalezení cesty z počátku do cíle zhruba N^2 . Při lepší implementaci úschovy nezpracovaných vrcholů lze na menších grafech dosáhnout i rychlejšího běhu algoritmu a čas je pak zhruba úměrný počtu hran grafu.

Výstupem Dijkstrova algoritmu je pouze délka nejkratší cesty. Samotnou cestu lze získat zpětným průchodem z pomocného pole s indexovanými vrcholy, kde pro každý vrchol je uloženo číslo vrcholu, ze kterého byla nalezena prozatímní (nebo trvalá) minimální cesta. S každou změnou hodnoty vrcholu se v průběhu algoritmu mění i předchůdce na nejkratší cestě. Celou cestu pak nalezneme průchodem tímto polem od vrcholu cílového.



Obr. 7 – postup výpočtu Dijkstrova algoritmu v orientovaném ohodnoceném grafu

5. Data

K vývoji aplikace pro hledání optimální cesty jsou nezbytná kvalitní vektorová data silniční sítě ve vhodném měřítku s přiřazenými popisnými informacemi pro identifikaci komunikací. V rámci této práce byla vyhledána a vyhodnocena data vhodná pro takové účely. Pro konečný výběr dat použitých v projektu byla však nakonec rozhodující vazba zadavatelské firmy na stávající zákazníky, kteří využívají datovou sadu Streetnet.

5.1 Obecné požadavky na data

Požadavky na data pro provádění síťových analýz se liší dle použitého programového vybavení a bývají většinou popsány v manuálu příslušné aplikace. V případě vlastní implementace síťových analýz vyhledávání nejkratší cesty jsou obecné požadavky zhruba následující:

- spojitost na sebe navazujících linií
- atributová přesnost dat
- možnost převedení dat do podoby síťového grafu

Zásadní podmínkou pro provádění síťových analýz je spojitost jednotlivých na sebe navazujících linií, které by na sebe měly navazovat v uzlech. Pro urychlení vyhledávání je také vhodné zjednodušit síť komunikací spojením jednotlivých částí sítě do delších souvislých celků.

Kvalita a přesnost ohodnocení modelu dopravní sítě průměrnými rychlostmi je jeden z faktorů, který má největší vliv na analýzy hledání nejkratší cesty. Proto je důležitá atributová přesnost dat. Typickým příkladem atributových nepřesností je výskyt chybných hodnot kódů vyjadřujících typ komunikace.

Data musí být převést do podoby síťového grafu, který bývá nejčastěji vyjádřen pomocí incidenční matice. Pro hledání nejkratší cesty je zásadním požadavkem možnost určení sousedících uzlů v síti a nalezení hran mezi nimi.

5.2 Příprava dat pro síťové analýzy

Před samotným prováděním síťových analýz je třeba vhodně připravit data, vytvořit síť. Použitá liniová vrstva musí být topologicky čistá. Je-li třeba, provádí se také spojování mapových listů. Důležitá je kontrola spojitosti sítě, která spočívá v kontrole spojitosti na sebe navazujících linií.

Dále je nutné do sítě přiřadit pomocí specifických atributů pravidla, která určují pohyb v síti. Ta se dělí na uzlová pravidla a hranová pravidla. Uzlová pravidla, tzv. odbočení, definují možnost a směr dalšího pokračování cesty v uzlu. Hranová pravidla definují rychlost a směr pohybu po jednotlivých segmentech dopravní sítě.

Klíčovým momentem přípravy modelu dopravní sítě je ohodnocení hran (segmentů) hodnotami, které budou ovlivňovat modelovaný pohyb v síti. Základním parametrem vyjadřujícím odpor proti pohybu v síti je délka úseku sítě. V případě detailních znalostí parametrů, které ovlivňují rychlost pohybu v síti, je možné provést ohodnocení úseků sítě hodnotami takových parametrů. Jedná se zpravidla o průměrnou rychlost, za kterou je daný úsek při pohybu v síti překonán, nebo čas po který je úsek překonáván.

Průměrná rychlost se nejčastěji stanovuje na základě znalosti kategorie silničního úseku a na základě toho, zda je silniční úsek hlavním průjezdem obce. Přiřazení průměrné rychlosti dle kategorie silnice představuje poměrně citlivou záležitost, neboť špatně zvolené rychlosti mohou ovlivnit výsledek síťové analýzy.

Tab. č.4 – návrh průměrných rychlostí dle kategorie silnice [4]

Typ komunikace	Průměrná rychlost [km/hod]
dálnice, silnice dálničního typu	85
silnice 1. kategorie	75
silnice 2. kategorie	55
silnice 3. kategorie	55
hlavní průjezd – 4 proudé silnice	60
hlavní průjezd -ostatní	40

5.3 Možné zdroje dat

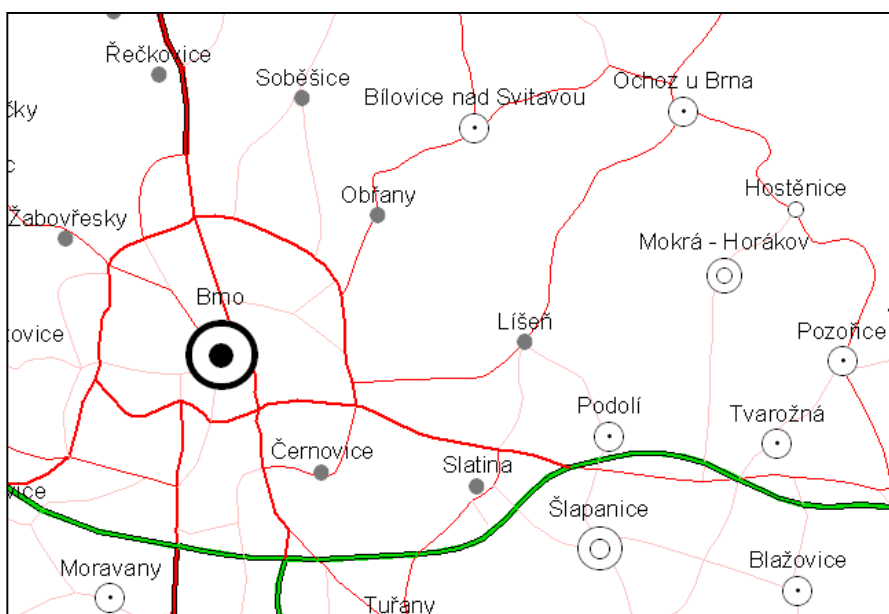
Jako možné zdroje dat vhodné pro hledání optimální cesty je možné použít několik datových sad obsahující vrstvu komunikací ČR.

5.3.1 ArcČR 500

Vektorová geografická databáze (aktuální verze 2.0) od firmy ARCDATA Praha a.s. v měřítku 1:500 000, dostupná v souřadnicovém systému JTSK, S42 a WGS84. Vektorová data jsou dostupná ve formátu ARC/INFO Coverage a ESRI Shapefile. Připojená atributová data jsou uchována jako soubory INFO nebo DBF.

Podle informací z webové prezentace firmy ARCDATA jsou data vhodná pro provádění různých druhů analýz. Hodí se například pro plánování tras rozvozu zboží. Vzhledem k rozlišení jednotlivých úseků komunikací se hodí pro dopravní analýzy na území velikosti kraje nebo celé republiky. [9]

Pro hledání optimální cesty je podstatná vrstva silnic, aktualizovaná k roku 2001. Komunikace jsou děleny na dálnice, rychlostní komunikace, komunikace I. třídy, II. třídy a ostatní komunikace. Chybí však údaje o směru dopravního provozu. Jako zastávky je možné využít samostatné bodové vrstvy sídel.



Obr. 8 – ArcČR 500, vrstva silnic a vrstva sídel (zastávky)

5.3.2 Česká republika 1 : 50 000 - silniční síť (T-Mapy)

Pro dopravní analýzy se dají využít i vektorové data silniční sítě od firmy T-Mapy. Na webové prezentaci firmy jsou data popsána jako vhodný podklad pro využití v oblasti dopravních a logistických analýz a v oblasti sledování dopravy. Data obsahují podrobnou silniční síť do úrovně III. třídy. Součástí dat jsou i databázové informace (druh komunikace a číslo). Hodí se spíše pro dopravní analýzy na území kraje či celé republiky.

5.3.3 DMÚ-25

Významným zdrojem topografických dat je vojenské mapové dílo DMÚ-25 v měřítku 1:25000. Správcem je Vojenský topografický ústav (VTOPU) se sídlem v Dobrušce. Autorská práva k tomuto dílu spravuje Generální štáb Armády České republiky.

Obsahuje topografické rozdělení do 7 tematických vrstev - vodstvo, sídla, komunikace, vedení sítí, hranice a ohrady, rostlinný a půdní kryt a terénní reliéf. Data jsou dostupná ve formátech ArcInfo coverage, ArcInfo Library a ESRI shapefile a v souřadnicových systémech S-JTSK, S-42 i WGS 84. Polohová přesnost dat je udávána v rozmezí 0,5 m - 20 m podle třídy objektu. Aktualizace databáze je prováděna plošně a její frekvence je deklarována 1× za 5 let. Jedná se o bezešvý digitální model celého území České republiky s mírným přesahem přes státní hranici.

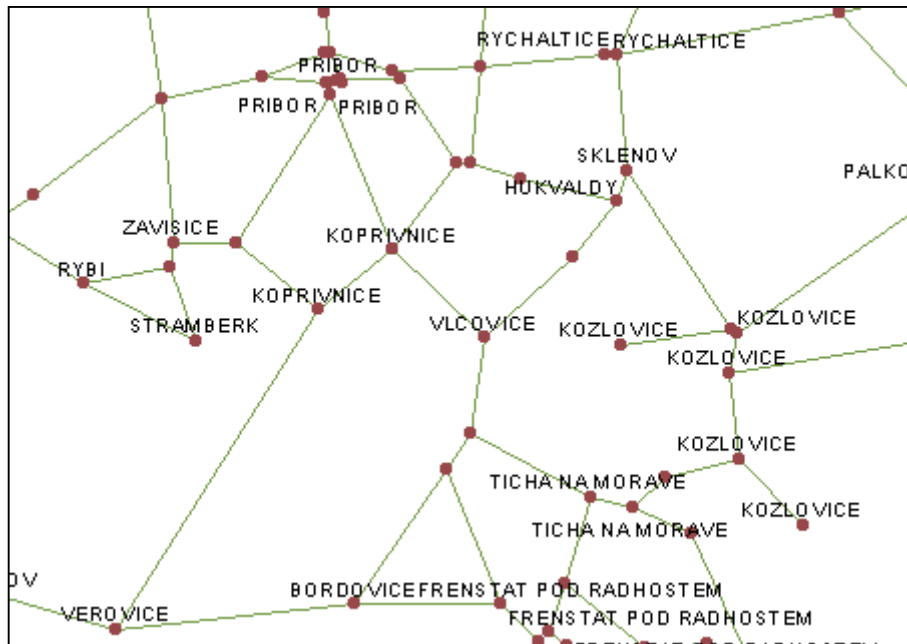
DMÚ-25 umožňuje rozdělit komunikace na silnice dálničního typu, silnice I. a II. třídy, ulice, hlavní průjezdy obcí, účelové komunikace a zpevněné cesty. Vzhledem k úrovni rozlišení se data hodí pro dopravní analýzy na území města nebo okresu. Pro území kraje či celé republiky je zřejmě vhodnější databáze DMÚ-200.

Předzpracování DMÚ-25 pro síťové analýzy již bylo provedeno na institutu geoinformatiky VŠB-TUO.

5.3.4 Silniční databanka Ostrava

Jedná se o vektorová data silniční sítě poskytovaná úsekem výstavby Ředitelství silnic a dálnic České republiky, se sídlem v Ostravě (odbor silniční databanky). Databanka obsahuje silnice I., II. a III. třídy. Obsahuje podrobný popis komunikací jako například: šířkové uspořádání komunikací, znázornění dopravních směrů, počet jízdnic pruhů, omezení rychlosti, vybavení komunikace (parkoviště, odpočívadla, čerpací stanice, zastávky MHD, motoresty a další). Dále pak obsahuje registr objektů - mosty, podjezdy, železniční přejezdy, přívozy, brody, tunely. Jejich aktualizace se provádí dvakrát za rok.

Největší výhodou těchto dat je, že jsou poskytována zdarma a jsou již předzpracována pro dopravní analýzy. Jejich součástí je také bodová vrstva zastávek s přiřazenými atributovými údaji, kdy geometrie jednotlivých bodů zastávek odpovídá uzlům liniové vrstvy komunikací. Vzhledem k jejich rozlišení jsou však tyto data vhodná pouze pro dopravní analýzy na území kraje nebo celé republiky.



Obr. 9 – Silniční databanka Ostrava, vrstva silnic a zastávek

5.3.5 StreetNet

Datová sada od firmy CEDA (Central European Data Agency, a.s.) dostupná ve formátu ESRI Shapefile nebo MapInfo MIF/MID a v souřadnicovém systému S-JTSK, WGS84 a S-42. Měřítko mapových podkladů je v intravilánu 1 : 10 000 a v extravilánu 1 : 25 000.

Obsahuje kompletní silniční síť České republiky až do úrovně ulic a místních komunikací. Obsahuje také vrstvy doplňující mapovou sadu o vrstvy využití území a zemního pokryvu (zástavba, zeleň, lesy, vody atd.).

Data jsou určena pro přesné řešení dopravních úloh a problémů. Uvádí se například vhodnost pro řešení svozu, rozvozu zboží a dopravní analýzy v prostředí geografických informačních systémů. Umožňuje určování tras mezi městy až do úrovně uličních úseků s vysokou přesností v místech mimoúrovňových křížení, dvouproudých silnic apod.

Popisná data obsahují obecné navigační informace (zákazy vjezdu, informace o směru dopravního provozu, a další). Komunikace mají v attributech přiřazeny informace o číslu, třídě a typu silnice, typu povrchu, informace o městě a názvu ulice. Je možné zjistit i zda se jedná o úsek s poplatkem. Struktura datového modelu odpovídá specifikaci GDF 4.0. Označení atributů a použité kódy vycházejí z této specifikace.

Mapovou sadu lze kombinovat s dalšími daty z nabídky CEDA, jako například s adresními body pro přesnou lokalizaci ve městech, či s lokalizační databází pro lokalizování dopravních událostí.

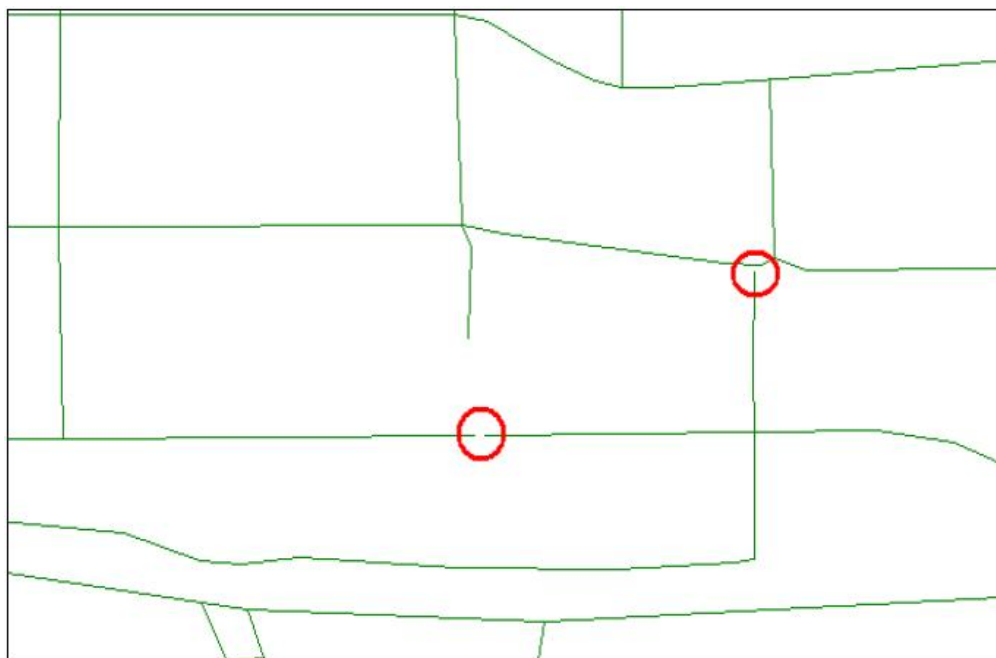
Vzhledem k velké detailnosti datové sady Streetnet se jedná o data vhodná pro dopravní analýzy především na území města či okresu. Pro použití na území celé republiky by bylo třeba provést jejich předzpracování, tzn. sloučení navazujících segmentů a definice zástavek v dopravní síti. Problémem je především obtížná identifikace zastávek v dopravní síti.

Je možné také použít verzi StreetNet Lite, která obsahuje pouze komunikace I. až III. třídy a pro dopravní analýzy na celém území republiky nebo kraje postačuje. Cena sady StreetNet je však poměrně vysoká, dle ceníku z 1.1.2006 činila 500.000,- Kč, u StreetNet Lite pak 200.000,- Kč.

5.4 Vhodnost datové sady Streetnet pro síťové analýzy

Po konzultacích se zadavatelskou firmou CAD programy – Ing. Jan Vlčinský byla pro tento projekt vybrána data od firmy StreetNet, a to zejména z důvodu existence vazby na stávající zákazníky společnosti. Nicméně tomuto rozhodnutí předcházela analýza datové sady StreetNet, jež měla za úkol posoudit zda bude tato datová sada vůbec použitelná pro aplikace hledání optimální cesty. Hodnocena byla verze uvolněná k 20.6.2003.

Mezi hlavní nedostatky patří především nespojitosti na sebe navazujících liniích, které mohou být způsobeny nepřesnou digitalizací (nedotažené linie či přetahy) nebo chybami v atributové tabulce. Problém se vyskytl zejména u komunikací nižších kategorií. Následuje ukázka nespojitosti komunikací nalezená na území hlavního města Prahy.



Obr. 10 – Streetnet, ukázka nespojitosti

Nalezené nespojitosti však nemusí být nutně chybou, každý nález je třeba konzultovat s poskytovatelem dat firmou CEDA, která může zkontrolovat nálezy se svými podklady.

Pro hledání časově nejkratší cesty je často používaným údajem průměrná rychlost, která se odvozuje od kategorie komunikace, návrh průměrných rychlostí dle kategorie komunikace viz kapitola 5.2. Atributová data nedovolují rozeznat silnice 1.třídy od silnic třídy druhé. Návrh atributu FC popisujícího kategorie silnic v následující tabulce.

Tab. č.5 – kategorizace silnic v datové sadě Streetnet

Kategorizace silnic	Atribut - FC	
		0 -dálnice
		1 -silnice 1. tř. s mezinárodním značením
		2 -ostatní silnice 1.tř. a významné silnice 2.tř.
		3 -ostatní silnice 2.tř.
		4 -silnice 3. tř.
		5 -významné silnice v rámci sídel;
		6 -ostatní významné silnice v rámci sídel
		7 -silnice vedoucí z hlavní silniční sítěna konkrétní adresu
		8 -ostatní /lesní a polní cesty, chodníky pro pěší,.../

Při stanovení průměrné rychlosti pro silnice s hodnotou atributu FC=2 (ostatní silnice 1.třídy. a významné silnice 2.třídy) nelze zaručit správnost výsledků například při vyhledávání nejkratší cesty podle časového kritéria. Pro stanovení průměrných rychlostí dle stávajícího dělení komunikací by bylo třeba provést srovnání s výsledky hledání cest poskytované veřejnými mapovými portály.

Jednotlivé komunikace jsou často rozděleny do více linií (segmentů) na základě příslušnosti k jednotlivým administrativním celkům. Toto dělení může být sice užitečné, ale pro urychlení hledání nejkratší cesty je vhodné takové úseky sjednotit. Na následujícím obrázku je komunikace rozdělena na dvě linie dle příslušnosti k dané obci.



Obr. 11 – ukázka rozdělení komunikace na dvě linie

Problémem může být také definice zastávek v silniční síti. Uzly odpovídající koncům linií vrstvy komunikací nemají přiřazené atributové údaje. Možností je využití bodové vrstvy sídel a na základě prostorových dotazů přiřadit jednotlivá sídla k uzlům v dopravní síti.

6. Programové prostředky

6.1 PostgreSQL 8.1

PostgreSQL je relační databázový systém, vyvíjený komunitou jako Open source software. Vzhledem k jeho funkcím se jedná o jeden z nejvyspělejších databázových systémů vůbec. Jeho přednostmi je podpora všech moderních operačních systémů, včetně OS Linux (součástí většiny distribucí), os/2, Novell a od verze 8.0.1 i Microsoft Windows 2000, XP. Je šířen pod BSD licenci umožňující vlastní úpravy a šíření binárního kódu.

Původní název databázového systému byl Postgres. Po přidání jazyka SQL se název změnil na Postgres95. Koncem roku 1996 byl RDBMS přejmenován na PostgreSQL. Zdrojový kód, ze kterého současný PostgreSQL vychází, je výsledkem úsilí mnoha studentů a programátorů pracujících pod vedením prof. Michaela Stonebraker z University of California v Berkley.

PostgreSQL je systém klient - server podporující SQL92 a SQL99 normu a mnoho dalších moderních rysů jako:

- § komplexní dotazy
- § cizí klíče (foreign keys)
- § spouště (triggers)
- § pohledy (views)
- § transakce
- § vlastní datové typy
- § agregační funkce
- § uložené procedury

Server je objektově-relační a vysoce rozšiřitelný díky možnosti definovat vlastní datové typy a uložené procedury. Z hlediska programátora je výborná možnost přístupu z mnoha jazyků C/C++ (knihovny libpq a libpq++), skriptovacích jazyků Perl (pgsql_perl5), Python (PyGreSQL), ale i třeba Javy (přístup přes rozhraní JDBC nebo využití PL/Java) a PHP.

Ve srovnání s jiným také velmi rozšířeným databázovým systémem MySQL nabízí více pokročilých funkcí, naproti tomu se uvádí, že je nepoměrně pomalejší.

6.1.1 Procedurální jazyk pgSQL

PostgreSQL umožňuje navrhovat a používat tzv. uložené procedury, tzn. kód, který je uložen a spouštěn SQL serverem. Pro PostgreSQL můžeme uložené procedury psát v některém z následujících programovacích jazyků: SQL, Perl, Python, TCL a PL/pgSQL.

Ačkoliv je PL/pgSQL pouze jedním z jazyků, který můžeme použít při návrhu uložených procedur, je patrně nejpoužívanější. Jedná se o jednoduchý programovací jazyk navržený pouze pro psaní uložených procedur RDBMS PostgreSQL. Nezavádí nové typy a vlastní funkce, obojí sdílí s databázovým systémem. Funkce mohou obsahovat většinu parametrizovaných SQL příkazů: pro správu tabulek, databází i jednotlivých záznamů. PL/pgSQL má konstrukci pro iteraci napříč množinou záznamů specifikovanou příkazem SELECT, jazyk umožňuje vytvářet SQL příkazy a pak je nechat provádět. Autoři PL/pgSQL se inspirovali jazykem PL/SQL, což je nativní programovací jazyk pro RDBMS Oracle, a tak je poměrně snadné konvertovat uložené procedury z Oracle do PostgreSQL a naopak.

6.1.2 PostGIS 1.1.0

PostGIS je nadstavbou nad databázovým systémem PostgreSQL. Umožňuje ukládat do databáze i prostorové objekty běžně používané v GIS (2D, 3D, 4D) a zároveň poskytuje rozšiřující funkce pro jednoduchou správu a manipulaci s těmito objekty.

PostGIS přidává do PostgreSQL nové datové typy, nové operátory (&& - průnik geometrií, @ - kompletně obsažen, a další), nové funkce pro práci s prostorovými daty (výpočet vzdálenosti, obsahu, a další) a nové tabulky obsahující metadata.

Data uložená v systému PostGIS lze následně zobrazit v celé řadě programových produktů jako Quantum GIS, Jump, UMN Mapserver a další. Je možné vytvořit vlastní funkce pro práci s geodaty v procedurálním jazyku PL/pgSQL. PostGIS podporuje následující typy geometrických objektů:

- § POINT(0 0)
- § LINESTRING(0 0,1 1,1 2)
- § POLYGON((0 0,4 0,4 4,0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))
- § MULTIPOINT(0 0,1 2)
- § MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))
- § MULTIPOLYGON(((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 2,1 1)), ((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1)))
- § GEOMETRYCOLLECTION(POINT(2 3),LINESTRING((2 3,3 4)))

Podporované geometrické objekty vychází ze specifikace OGC “Simple Features for SQL“. PostGIS podporuje veškeré funkce a objekty definované v této specifikaci a zároveň přidává podporu pro 3D a 4D objekty.

6.2 ArcView 9.0

ArcView je produkt firmy ESRI, který patří do kategorie desktop GIS. Jedná se o první ze tří úrovní řady ArcGIS Desktop. Balík ArcView 9 tvoří sada aplikací: ArcMap, ArcCatalog, ArcToolbox a ModelBuilder. ArcView je nástroj pro interaktivní tvorbu map, umožňuje definovat mapy na základě vektorových, rastrových a vrstev webových služeb. Umožňuje vytvoření profesionálních tématických map včetně popisků, grafiky, legendy, měřítka, komplexní knihovny symbolů i tisk map velkých formátů.

Obsahuje nástroje pro práci s mapou jako geokódování adres, identifikace geoprvků, připojování a editaci atributových tabulek, tvorba grafů a zpráv a průvodce prostorovými operacemi.

Vývojářům poskytuje komplexní objektový model (ArcObjects). Rozšiřitelnost je možná pomocí jazyků VB, VC++ a jazyků platformy .NET.

6.3 JGraphT 0.6.0

Pro vyhledání nejkratší cesty byl využit Open Source projekt JGraphT implementující objekty a algoritmy teorie grafů.

JGraphT podporuje mnoho různých druhů grafů:

- § orientované (neorientované) grafy
- § ohodnocené (neohodnocené) grafy
- § grafy s vlastní implementací hran
- § grafy s libovolnou násobností hran (prostý graf nebo multigraf)
- § grafy s režimem pouze pro čtení
- § grafy podporující události (změna vnitřní struktury vyvolá událost)

I přes pokročilé funkce, je JGraphT navržen s ohledem na co největší jednoduchost použití. Podporuje vytváření vlastní podoby grafů založených na řetězcích, URL adresách či XML dokumentech. Knihovna je určena pro vytváření rozsáhlých datových a výpočetních modelů a dokáže zpracovat grafy s několika miliony uzlů a hran. Je popsána pomocí API rozhraní přístupného na internetu.

Jedná se o JAVA knihovnu, jež je uvolněna pod licencí GNU-LGPL, která umožňuje poskytnout svobody zaručené licencí GPL (tedy neomezené užívání, přístup ke zdrojovému kódu, možnost modifikace, a další) a současně umožnit použití v odlišně licencovaných programech. Důležité je, že možnost použití v programech bez licence GPL je omezena na dynamické přilinkování knihovny. Pokud by se kód knihovny do programu přilinkoval staticky, musely by se na tento program vztahovat podmínky uložené licencí GPL. [11]

Důležitou vlastností této knihovny je její optimalizace na rychlost, kdy implementované algoritmy pro hledání optimální cesty jsou schopny pracovat dostatečně rychle i s velmi rozsáhlými grafy.

6.4 JBuilder 2005 - Foundation

JBuilder 2005 je sada nástrojů pro vývojáře v jazyce Java poskytující integrované vývojové prostředí (IDE) pro zjednodušení vývoje Java aplikací nezávislých na operačním systému. Verze Foundation je odlehčená verze poskytovaná zdarma i pro komerční účely. Představuje základní výbavu pro rychlé kódování a ladění prostřednictvím integrovaného, výkonného editoru zdrojových kódů, grafického ladicího programu, překladače, školicích návodů a příkladů aplikací. Existuje ve verzích pro platformy Microsoft® Windows®, Linux® a Solaris™. Plně podporuje Java Development Kit (JDK) 1.4 a vyšší.

Mezi klíčové funkce produktu patří prohlížeč aplikace (AppBrowser) zjednodušující správu kódu, nástroj CodeInsight™, který hlídá syntaktické chyby a zrychluje psaní kódu, grafický ladicí nástroj pro platformu Java, dvoucestné nástroje pro čistý jazyk Java, podpora Ant a návrhář Swing. Uživatelé si mohou přizpůsobit a rozšířit prostředí při vývoji podle jejich vlastních potřeb použitím nástrojů Open Tools API, které jim umožní snadnější integraci nástrojů a komponent třetích stran.

6.5 Quantum GIS

Quantum GIS (QGIS) je uživatelsky přívětivý Open Source GIS, dostupný pro platformy Unix, Mac OSX a Windows. Umožňuje zobrazení a editaci rozličných vektorových a rastrových formátů, z nichž nejzajímavější je zřejmě podpora ESRI Shapefile, Mapinfo MIF/MID. Význačná je pak především podpora databázového systému PostgreSQL s extenzí PostGIS.

7. Návrh řešení

Na základě nastudování vhodných informačních zdrojů bylo navrženo řešení pro hledání optimální cesty. Jako vstupní data byla vybrána vrstva komunikací z datové sady Streetnet od firmy CEDA, která byla posléze importována do databáze PostgreSQL. Pro vytvoření grafu a vyhledání optimální cesty je použita aplikace napsaná v jazyce JAVA využívající knihovnu JGraphT, která poskytuje vlastní implementaci Dijkstraova algoritmu i vlastní podobu grafu.

Celé řešení hledání optimální cesty je možné shrnout do několika kroků:

1. návrh databáze a import dat
2. příprava dat do podoby vhodné pro vytvoření síťového grafu
3. vytvoření síťového grafu (třída JGraphT DirectedWeightedMultigraph)
4. nalezení optimální cesty

Pro hledání optimální cesty byly navrženy dva alternativní postupy, které se liší v druhém a třetím kroku.

První verze je založena na tabulkách hran a uzlů naplněných pomocí funkce napsané v procedurálním jazyce pgSQL na základě záznamů z importované tabulky komunikací. Záznamy v těchto tabulkách lze chápat jako pomocný graf posléze využitý pro vytvoření vlastní podoby grafu JGraphT. Celý proces přípravy pomocného grafu je časově poměrně náročný, pro 100 tisíc záznamů v tabulce komunikací se jedná o úlohu zabírající několik hodin.

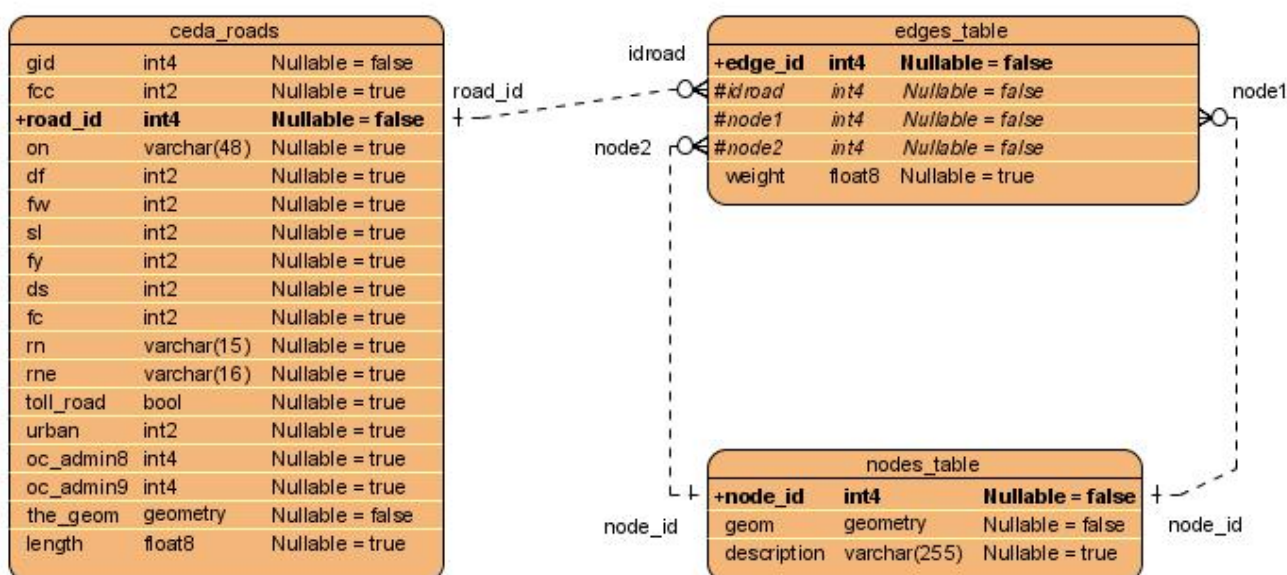
Druhá verze využívá pro vytvoření grafu pouze tabulky komunikací. Přímou v JAVA aplikaci jsou generovány uzly a hrany a následně vkládány do grafu, který je možno serializovat na disk nebo rovnou použít pro hledání optimální cesty. Tento postup je mnohem rychlejší, ale má značné paměťové nároky (min. 1 GB operační paměti). Prakticky je tento postup zajištěn pouze překrytím metody pro generování grafu.

7.1 Práce s databázovým systémem PostgreSQL

7.1.1 Návrh databáze

Databáze obsahuje celkem tři tabulky: tabulku komunikací (ceda_roads), tabulku hran (edges_table) a tabulku uzlů (nodes_table). První byla vytvořena importem dat z atributové tabulky vstupní vrstvy komunikací s přidáním sloupců pro geometrii (typ MULTILINESTRING) a délku linie. Tabulka uzlů pak obsahuje sloupce pro geometrii uzlů (typ POINT), identifikátor uzlu a sloupec s popisem kvůli možnosti definice zastávek.

Podstatná je především tabulka hran, která vstupuje do algoritmu pro vyhledávání nejkratší cesty. Relační schéma databáze viz níže:



Obr. 12 – relační schéma databáze

Ze schématu je patrné, že tabulka hran obsahuje sloupec s identifikátorem komunikace zapsané v tabulce ceda_roads, sloupce s identifikátory počátečního a koncového uzlu z tabulky nodes_table (cizí klíče), a sloupec obsahující váhu hrany. Definice tabulek v jazyce SQL viz příloha.

Pro urychlení prostorového vyhledání byl vytvořen index typu GiST (Generalized Search Tree) pro sloupce s geometrií. Ukázka SQL příkazu pro vytvoření indexu pro tabulku ceda_roads:

```
CREATE INDEX mygist ON ceda_roads USING gist (the_geom);
```

Dále byly vytvořeny indexy typu B-Tree pro sloupce s jedinečnými číselnými identifikátory, u nichž se předpokládá využití v SQL dotazech. Ukázka SQL příkazu:

```
CREATE UNIQUE INDEX roads_btree ON ceda_roads USING btree  
(road_id);
```

7.1.2 Import dat

Nejdříve bylo třeba vytvořit databázi, do které měla být importována data ze zdrojového SHP souboru obsahující vrstvu komunikací ČR. Pro databázi bylo třeba nastavit kódování WIN 1250, v němž jsou uložena atributová data vrstvy komunikací:

```
CREATE DATABASE streetnet WITH OWNER = postgres  
ENCODING = 'WIN1250'  
TABLESPACE = pg_default;
```

Data byla převedena z SHP souboru do textové podoby SQL příkazů pomocí utility shp2pgsql:

```
..\shp2pgsql -D -s 32633 roads.shp ceda_roads >  
ceda_roads.sql
```

Vlastní import byl proveden v klientovi psql:

```
..\psql -d streetnet -f ceda_roads.sql -U postgres
```

7.2 Příprava dat

Vytvoření grafu spočívá v konverzi uzlů sítě na vrcholy grafu a úseků sítě na hrany grafu, kdy ohodnocení hran se počítá se na základě délky úseku (délka komunikace). Délka úseků je většinou součástí atributových údajů. Dílčím úkolem je také definice zastávek v síti.

Aby bylo možné vytvořit graf silniční sítě, bylo třeba vstupní data vhodně předpřipravit. Jelikož použitá data neobsahovala informace o délce linie, byl do importované tabulky komunikací (ceda_roads) přidán sloupec pro délku linie důležitý pro odvození ohodnocení hran. Pro výpočet hodnot byl použit příkaz SQL, který využívá funkci extenze PostGIS pro výpočet délky:

```
UPDATE ceda_roads SET length = length(the_geom)*100;
```

Obtížnější bylo navržení procedury pro definování zastávek v dopravní síti. Zastávky bývají většinou dodávány společně s vrstvou komunikací v samostatné bodové vrstvě s přiřazenými atributovými údaji. Jednotlivé body zastávek geometricky odpovídají koncovým bodům linií (uzlům) reprezentující jednotlivé komunikace.

V případě, že vrstva se zastávkami dostupná není, lze s využitím dostupné bodové vrstvy sídel přiřadit zastávkám atributové údaje například pomocí prostorových dotazů. Správné přiřazení sídla k zastávce záleží také na přesnosti dat, nelze tedy zaručit stoprocentní úspěšnost. Použitá datová sada StreetNet samostatnou vrstvou zastávek neobsahuje, nicméně pro splnění zadání diplomové práce nebylo třeba řešit chybějící atributové údaje.

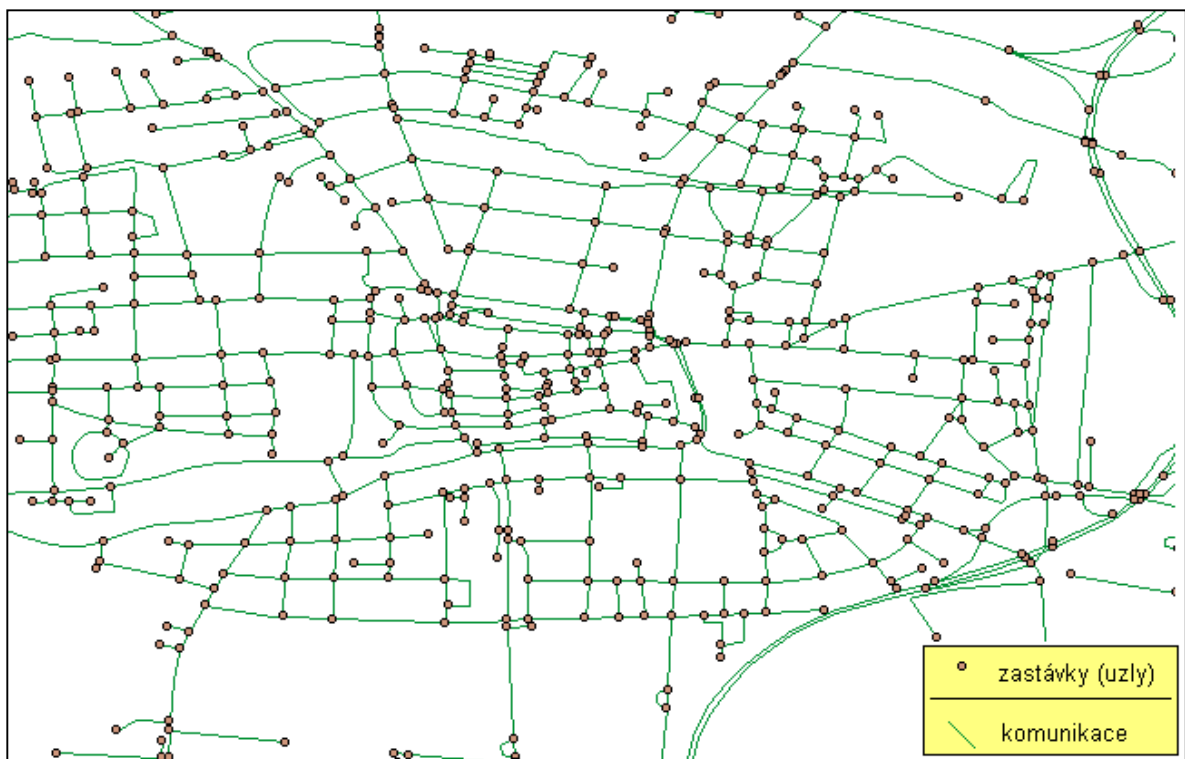
7.2.1 Vnitřní funkce pro naplnění tabulky hran a uzlů

Uzly grafu (zastávky) byly v případě první verze řešení vygenerovány jako záznamy v tabulce uzlů (nodes_table) pomocí vnitřní funkce napsané v jazyce PL/pgSQL, která zároveň vkládá záznamy i do tabulky hran (edges_table).

Vytvořená vnitřní funkce využívá extenze PostGIS. Jedná se o iteraci přes všechny záznamy tabulky komunikací, která je jediným podkladem pro generování záznamů v tabulce uzlů a hran.

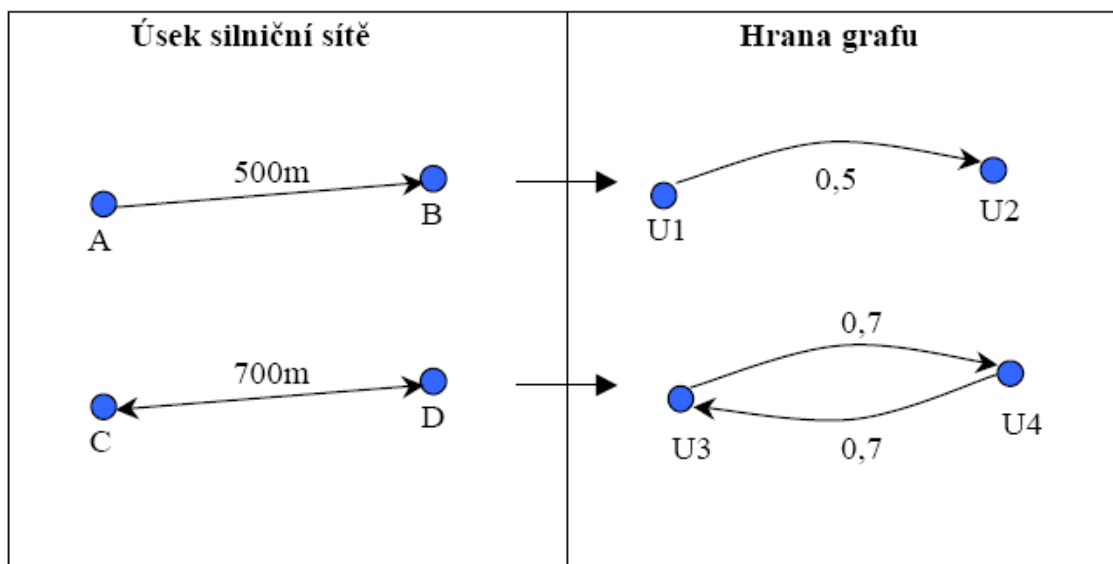
V iteraci se pomocí funkcí PostGIS zjišťuje pro každou linii, představující konkrétní komunikaci, počáteční a koncový bod (uzly). Pro linii se pak vkládají maximálně 2 záznamy do tabulky uzlů, kdy se pomocí souřadnic kontroluje zda již uzel v tabulce je uložený či nikoliv. Každý uzel tak má přiřazenou jedinečnou dvojici souřadnic X, Y.

Vytvořenou tabulku uzlů je možné přímo zobrazit včetně podkladové vrstvy komunikací například v programovém produktu Quantum GIS a provést tak vizuální kontrolu výstupu, viz následující obrázek.



Obr. 13 – vrstva uzlů a komunikací

Nakonec se v iteraci vkládá záznam pro hranu, skládající se z identifikátorů uzlů, identifikátoru komunikace a váhy hrany. Záznamy v tabulce hran jsou generovány z tabulky komunikací na základě principu znázorněném na následujícím obrázku.



Obr. 14 – generování záznamů v tabulce hran

Je zřejmé, že pro jeden záznam v tabulce komunikací jsou v závislosti na jednosměrnosti/obousměrnosti komunikace generovány 1 nebo 2 záznamy v tabulce hran. Zároveň je hraně přiřazeno ohodnocení na základě atributu délky linie.

Zdrojový kód vnitřní funkce viz příloha.

7.3 Implementace řešení hledání optimální cesty

Pro řešení úlohy hledání optimální cesty byly vytvořeny vlastní třídy a sestavena aplikace napsaná v jazyce JAVA (J2SE 1.5.0), která vypisuje identifikátory jednotlivých komunikací na nalezené optimální cestě. Celý proces nalezení cesty mezi uzly A-B lze rozložit do těchto kroků:

1. vytvoření grafu
2. vyhledání cesty pomocí Dijkstrova algoritmu

Nejsložitější byla zřejmě úloha vytvoření grafu, pro kterou byla navržena vlastní třída PostgreGraphMaker. Třída poskytuje metody pro vytvoření grafu na základě údajů z databáze.

Jelikož poskytované třídy grafů JGraphT nepodporují serializaci, bylo třeba přidat i metody pro serializaci grafu na disk, které využívají možnosti, že každý graf lze rozložit na kolekci hran a uzlů, které již lze samostatně serializovat do jednoho souboru.

Pro vytvoření grafu byly použity dva přístupy, prakticky realizované pomocí překrytí metody makeGraph(). Oba využívají jiné tabulky pro generování uzlů a hran grafu, dále se také liší rychlostí celého procesu a paměťovými nároky. Výstupem obou metod je třída implementující rozhraní Graph definované v knihovně JGraphT.

7.3.1 Vytvoření grafu - verze první

První verze využívá tabulku hran, viz návrh databáze kapitola 7.1.1, vygenerované pomocí vnitřní funkce napsané v PL/pgSQL, kapitola 7.2.1.

Každá hrana je definovaná pomocí identifikátoru počátečního a koncového uzlu, svojí váhy a identifikátoru komunikace. Všechny údaje jsou získány z databáze, počet vytvořených hran v grafu odpovídá počtu záznamů v tabulce hran. V případě využití jiného kritéria pro stanovení váhy hrany než délky komunikace lze přidat sloupec do databáze s vypočtenou váhou nebo provést výpočet přímo v aplikaci.

Vstupními parametry do metody pro generování grafu jsou názvy:

- § tabulky hran
- § sloupce s identifikátorem hrany
- § sloupce s identifikátorem komunikace (cizí klíč)
- § sloupce s identifikátorem počátečního a koncového uzlu hrany

Celý proces generování grafu je poměrně rychlý, na základě záznamů v tabulce hran se vytváří hrany a uzly, která se vkládají do grafu. Hrany jsou označeny identifikátorem komunikace pro pozdější rekonstrukci optimální cesty.

Navržený postup má jednu zásadní nevýhodu a tou je doba generování záznamů v tabulce hran a uzlů pomocí navržené vnitřní funkce. Při využití počítače s procesorem AMD Barton 2600+ s 512 MB operační paměti trval celý proces pro 100 tisíc záznamů tabulky komunikací (asi 1/6 všech komunikací v datové sadě Streetnet) téměř pět hodin. Na druhou stranu samotné generování grafu z takto předpřipravených dat již proběhlo za pouhých 10 sekund.

Využití tohoto postupu tedy přichází v úvahu pouze za předpokladu, že se vstupní data nebudou často měnit. Při menších změnách je možné upravit pouze související záznamy komunikací a hran, není třeba provádět celé generování od začátku. Při změně dopravní situace, např. objížďky, je pak možné odpovídající hranu z grafu dočasně vyjmout nebo změnit její váhu.

7.3.2 Vytvoření grafu - verze druhá

Druhá verze byla vytvořena na základě připomínek zadavatele. Pro vytvoření grafu používá záznamy z tabulky komunikací vzniklé importem originálních podkladů doplněné o sloupec s délkou linie.

V iteraci přes všechny záznamy tabulky komunikací jsou generovány uzly a hrany grafu a vkládány do grafu. U uzlů, počáteční a koncový bod linie, se kontroluje pomocí souřadnic, zda byl již do grafu vložen či ne, tzn. stejný postup jako u vnitřní funkce pro generování tabulky uzlů a hran, tzn. identifikátorem uzlu jsou jeho souřadnice. Hrana je opět definována pomocí uzlů, váhy a

identifikátoru komunikace. Vytvoření hrany nejlépe objasňuje ukázka zdrojového kódu:

```
edge = new StreetnetEdge(coord2.toString(), coord1.toString());  
edge.setWeight(lineString.length());  
edge.setRoadID(roadID);
```

Pro vytvoření grafu slouží metoda `makeGraph()` třídy `PostgreGraphMaker`. Jejími vstupními parametry jsou názvy:

- § tabulky komunikací
- § sloupce s identifikátorem komunikace (primární klíč)
- § sloupce s geometrií
- § sloupce s dopravními směry
- § sloupce s GID (identifikátor geoprisku)
- § sloupce s popisem a označením komunikace (volitelné)

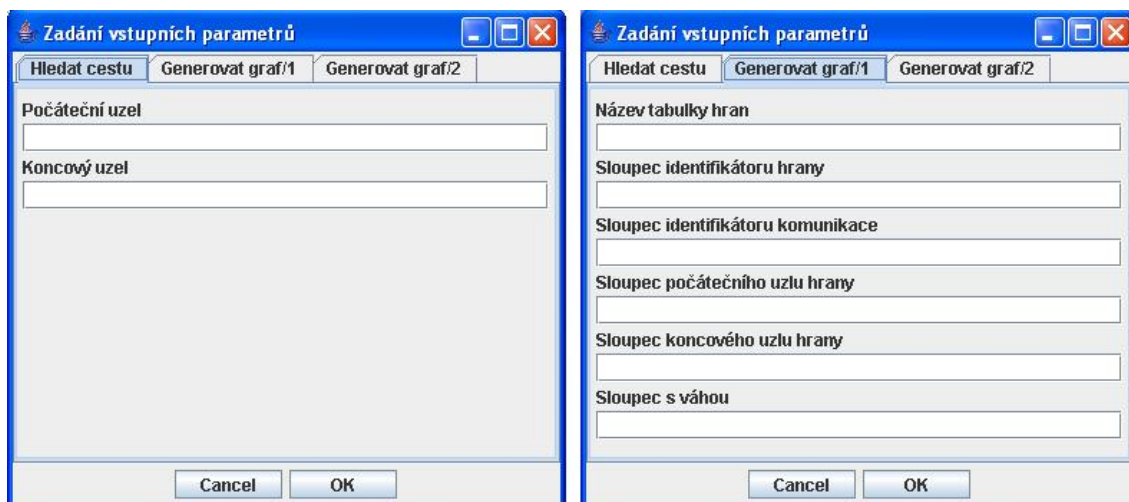
Výhodou tohoto přístupu je celková rychlost vytvoření grafu, neboť není třeba předpřipravovat data. Při 100 tisících záznamech v tabulce komunikací byl graf vygenerován za 27 sekund, konfigurace počítače opět AMD Barton 2600+ s 512 MB paměti.

Paměťové nároky tohoto přístupu jsou však značné a nastavení horního limitu paměti JVM bylo třeba nastavit aspoň na 128 MB, pro 400 tisíc záznamů již bylo třeba alokovat minimálně 450 MB paměti. Dalším problémem může být speciálně u datové sady `StreetNet` zadávání počátečního a koncového uzlu cesty (není dostupná vrstva uzlů s přiřazenými atributovými údaji), neboť uzly jsou v grafu definovány pomocí souřadnic. Alternativní možností je zadat počáteční a koncovou hranu cesty.

7.3.3 Nalezení optimální cesty pomocí Dijkstrova algoritmu

Vytvořený graf je použit jako vstupní parametr pro metodu třídy DijkstraPathFinder využívající implementaci Dijkstrova algoritmu JGraphT. Povinným vstupním parametrem je také identifikátor počátečního a koncového uzlu hledané cesty. Dále může být specifikován maximální poloměr kružnice pro hledání optimální cesty a seznam hran, přes které nalezená cesta nesmí vést. Odebrání hran je pouze dočasné a nemá vliv na graf, při práci s vlákny by však bylo třeba provést menší úpravy.

Pro možnost otestování rychlosti aplikace, bylo přidáno základní grafické rozhraní umožňující zadat parametry pro vyhledávání, případně parametry nutné pro vytvoření grafu.



Obr. 15 – grafické rozhraní

Výstupem je vlastní třída ReturnedPath obsahující seznam hran a délku nalezené optimální cesty. Výsledný JAVA kód pro vyhledání nejkratší cesty včetně vytvoření grafu je záležitostí několika řádek:

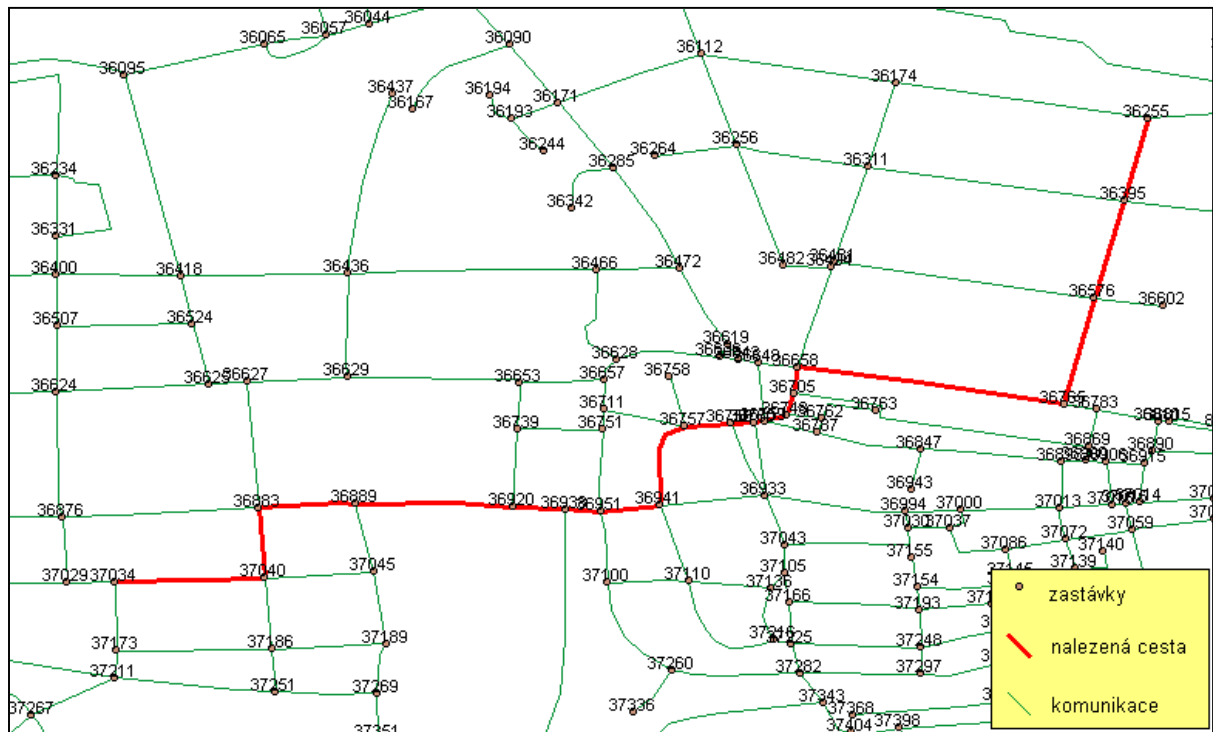
```
DBLinker dblinker = new
DBLinker("jdbc:postgresql://localhost:5432/streetnet","postgres","postgres");
PostgreGraphMaker graphMaker = new PostgreGraphMaker(dblinker);
DirectedWeightedMultigraph newGraph = new DirectedWeightedMultigraph();
GraphMaker.makeGraph("roads100","road_id","the_geom","df","gid",newGraph);
ReturnedPath rp = DijkstraPathFinder.findPath(newGraph, startNode, endNode,
Double.POSITIVE_INFINITY);
```

V předchozí ukázce byly nastaveny parametry připojení k databázi, vytvořen nový graf, který byl dále předán odkazem metodě `makeGraph`, což umožňuje použití metody pro jakýkoliv typ grafu implementujícího rozhraní `Graph`. Dále byly vloženy informace o názvech tabulky komunikací a potřebných sloupcích. Nakonec byla volána metoda třídy `DijkstraPathFinder` vracející výsledky vyhledávání. Je možné získat seznam uzlů i seznam hran, přes které nalezená cesta vede. Výpis z aplikace pak může být následující:

```
Pocet hran nalezene cesty: 18
Ohodnocení cesty: 0.02016
idRoad: 568332 label: Havlíčkova
idRoad: 568902 label: Havlíčkova
idRoad: 569423 label: Havlíčkova
idRoad: 569424 label: Vápenice
idRoad: 569410 label: Školní
idRoad: 569411 label: Pernštýnské nám.
idRoad: 569403 label: Skálovo nám.
....
```

Dijkstraův algoritmus implementovaný v knihovně `JGraphT` dosahuje výborných výsledků, průměrná doba hledání cesty byla v rozsáhlém grafu se 187 tisíci hranami přibližně 0.05 sekundy.

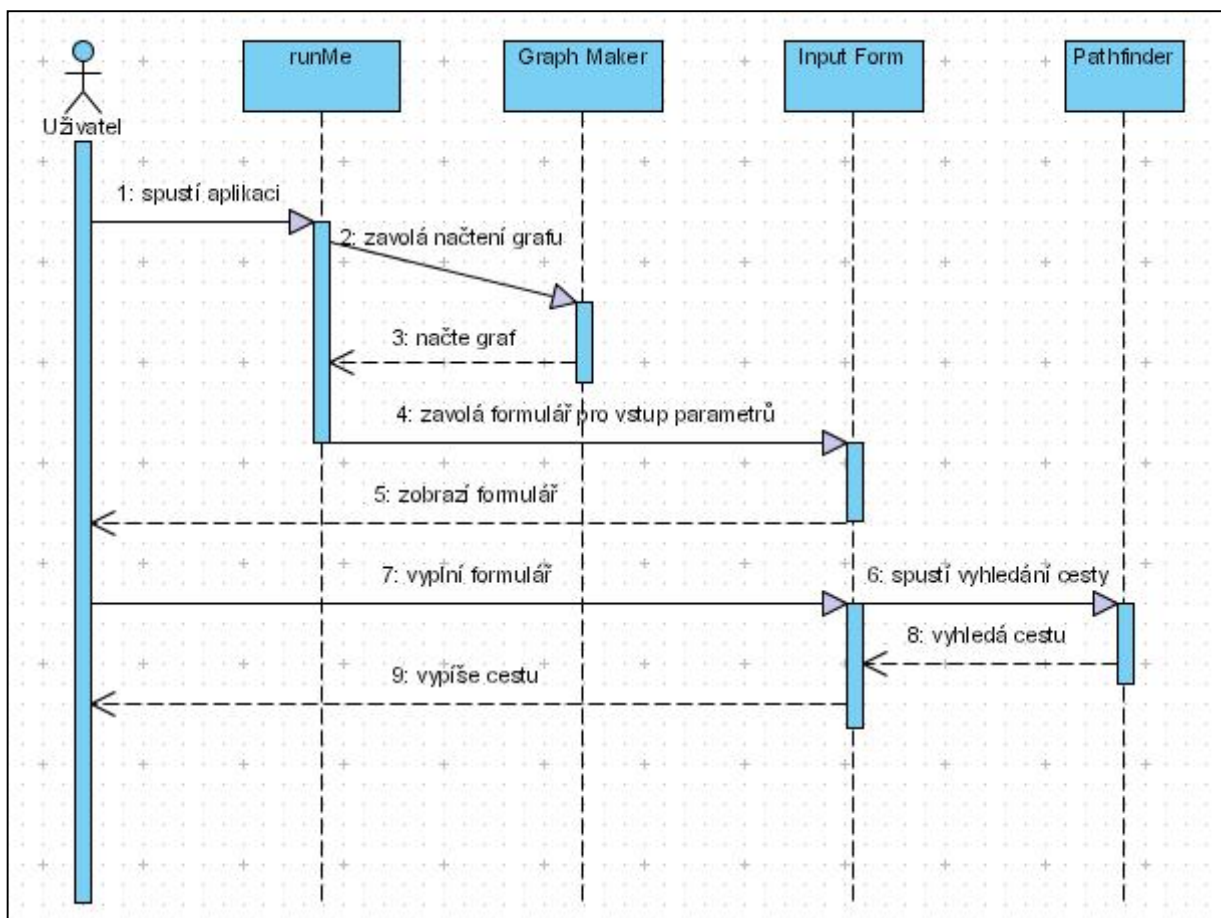
Vzhledem k tomu, že uzly generované na základě vrstvy komunikací datové sady `StreetNet` zatím nemají přiřazené atributové údaje, nelze vypsát mezi kterými zastávkami cesta vede. Pro kontrolu výsledku vyhledávání je však možné použít identifikátory komunikací a zobrazit výsledek pomocí dotazu. Nalezená nejkratší cesta mezi uzly s označením 37034 a 36255 vykreslena na následujícím obrázku.



Obr. 16 – zobrazení výsledku hledání nejkratší cesty

7.3.4 UML diagramy

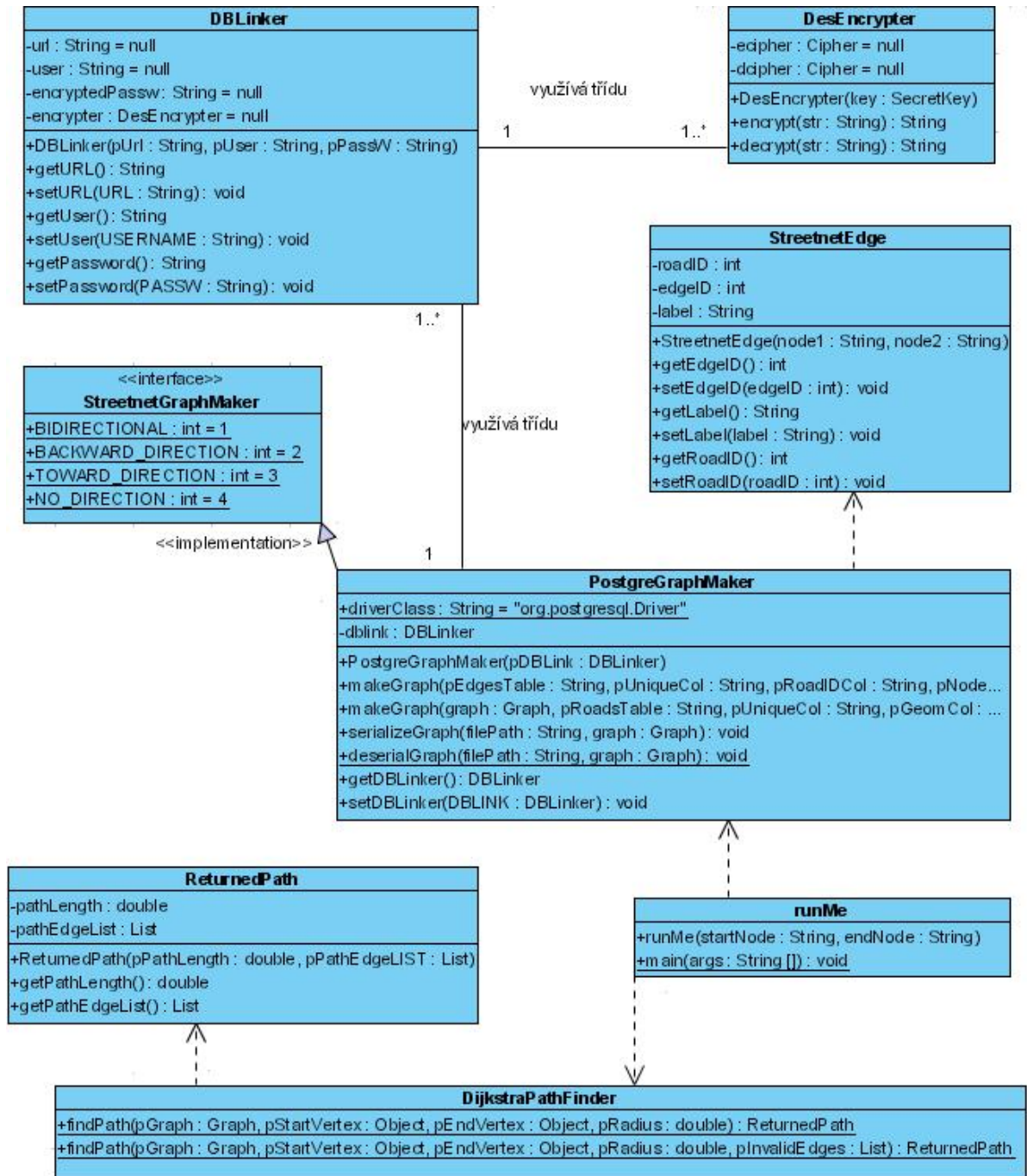
Popis statické struktury systému a jeho chování byl popsán pomocí CASE nástroje. Následuje UML sekvenční diagram, diagram tříd a diagram aktivit. Sekvenční diagram zobrazuje komunikaci (zasílání zpráv) mezi objekty. U každého objektu je definována časová osa.



Obr. 17 – sekvenční diagram

Diagram tříd se používá k zobrazení statické struktury systému prostřednictvím tříd a vztahů mezi nimi. Vytváří se ve fázi návrhu projektovaného systému a slouží jako podklad pro návrh a tvorbu programového kódu. Je možné ho vytvořit i reverzně na základě programového kódu.

Obr. 18 – diagram tříd



Obr. 19 – diagram aktivit

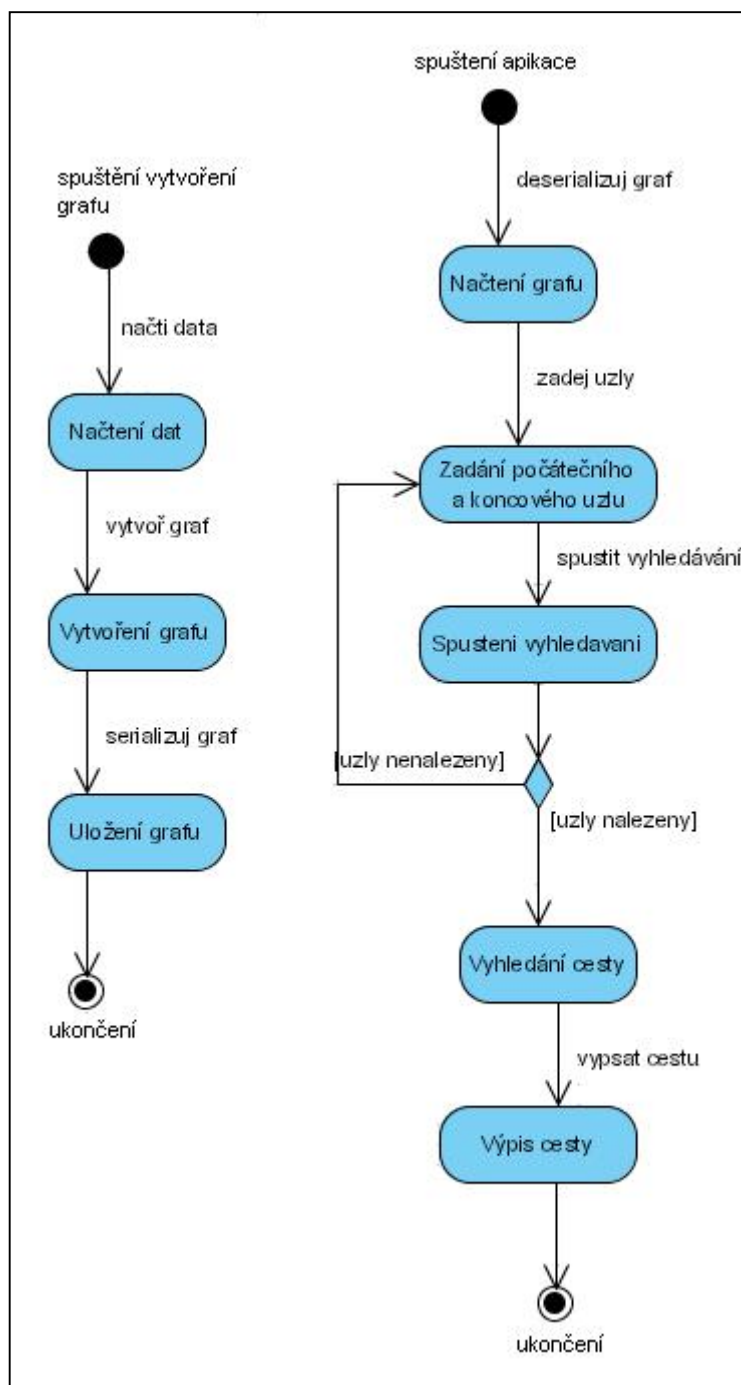


Diagram aktivit slouží k reprezentaci dynamiky počítačových a organizačních procesů v systému. Zaměřuje se především na jeho vnitřní chování. Využívá se k zobrazení řídicích toků mezi akcemi od počátečního bodu do konce. Důraz se klade na zobrazení pořadí aktivit.

8. eRDIS

Výsledky práce byly využity zadavatelskou firmou v systému eRDIS (Redakční Dopravní Informační Systém) pro hledání nejkratší cesty. V systému eRDIS se zpracovávají dopravní informace z řady zdrojů (telefonáty řidičů, importy z jiných informačních systémů, faxy, e-mail). Výstupem je dopravní informace určená k distribuci navazujícími systémy. Praktické úlohy řešené pomocí eRDIS mohou být: hledání obcí a ulic, náhled na několik nalezených objektů, nalezení bodu na silnici dle kilometráže, hledání cesty.

Jedná se o třívrstvou server-klient aplikaci pro tvorbu a správu dopravních informací, založenou na freeware a Open Source řešeních. Na straně serveru běží aplikační server JBoss a databázový systém PostgreSQL. Klient je reprezentován internetovým prohlížečem s podporou JavaScriptu.

Kód v jazyce JAVA posloužil jako podklad pro vytvoření programových komponent JAVA Bean pro generování grafu a hledání cesty, vystavených v aplikačním serveru. Hledání cesty na klientovi probíhá zadáním počáteční a koncové komunikace. Pro zadávání a vykreslování nalezené trasy je použit JavaScript.



The screenshot displays the eRDIS web interface. At the top, there is a search bar with the text 'Nová dopr. inf. | Přejít na ... | Vyhledat | Hledat'. Below the search bar is a map showing a network of roads with a highlighted route in pink. The map includes a scale bar for 400m and various map controls. Below the map, there is a navigation menu with options like 'mapa: nízká | střední | vysoká | externí mapa © | Zobrazit LT'. A search results section is visible, showing a table of traffic incidents.

Typ	Editace	Vytvořeno	Od	Do	Urg	Událost	Místo
▲	zobíli	03/04 20:54	04/04 20:52	13/04 20:52		silnice uzavřena, 1 nehod; olej na vozovce	Markovická, délka 0,2km
▲	zobíli	14/02 12:16	01/03 06:00	30/06 19:00		práce na silnici, zúžení vozovky z 3 pruhů na dva jízdní pruhy	Jesenická / Za státními / Na vinobraní / Jablňovská / U Zahradního města / Hlohová, délka 0,9km dálniční křižovatka; Exit 66, 150 -

Obr. 20 – hledání nejkratší cesty v eRDIS

9. Závěr

Hlavním cílem tohoto projektu bylo implementovat vlastní řešení hledání nejkratší cesty mezi dvěma uzly silniční sítě. Výstupem měl být textový popis nalezené cesty.

K tomu bylo třeba provést několik kroků. Nejdříve byla provedena analýza možných zdrojů dat. I přes jisté nedostatky byla vybrána datová sada Streetnet od firmy CEDA, a to především kvůli vazbě na koncové zákazníky zadavatelské firmy. Nejzávažnějším problémem u těchto dat je nedostatečné atributová přesnost, jež může ovlivnit výsledek hledání cesty, a problémy s identifikací zástavek v síti.

V dalším kroku bylo třeba připravit databázi. V zadání byla specifikována Open source databázový systém PostgreSQL s extenzí PostGIS. Zde bylo třeba importovat liniovou vrstvu komunikací ve formátu SHP do databáze a vytvořit vnitřní funkci pro generování údajů na základě importované tabulky. Funkce vkládá záznamy do tabulky uzlů a hran, kdy uzly jsou získány jako první a poslední vrchol linie. Tabulka hran pak obsahuje identifikátory linie, uzlů a svoji váhu. Všechny potřebné údaje jsou tedy uloženy a případně i generovány přímo v databázi (např. váhy).

V posledním kroku byl napsán JAVA kód, který řeší vytvoření grafu a vyhledání nejkratší cesty. K tomu je využita knihovna JGraphT, která celou úlohu značně zjednodušila a zároveň umožnila dosáhnout velmi dobrých výsledků z hlediska časové náročnosti hledání nejkratší cesty.

Pro vytvoření grafu byly použity dva přístupy. První využívá pro naplnění grafu tabulku hran naplněnou pomocí vnitřní funkce v PL/pgSQL. Proces vytvoření grafu je velmi rychlý a nemá velké paměťové nároky. Na druhou stranu naplnění tabulky hran trvá až několik hodin v závislosti na množství dat. Druhý přístup využívá pouze importované tabulky komunikací. Hrany a uzly jsou generovány přímo v aplikaci pomocí tříd pro práci s prostorovými objekty PostGIS. Tento postup je mnohem rychlejší, ale má značné paměťové nároky.

Výsledky práce byli prezentovány v zadavatelské firmě a schváleny. Vytvořené podklady byly předány a budou vyžity pro projekty firmy CAD programy.

Seznam použitých zdrojů

Elektronické články

- 1) HILDEBRAND, A. *Pathfinding algoritmy* [online]. Aktualizace 8.9.2000 [cit. 2005-10-25]. Dostupné na www: <<http://pathlib.hildebrand.cz/doc/Referat/pathref.html>>
- 2) HOKSZA, D. *Nejkratší cesta v ohodnoceném grafu* [online]. Aktualizace 2000 [cit. 2005-01-08]. Dostupné na www: <<http://www.reboot.cz/index.phtml?id=223>>
- 3) PEŇÁZ, T. , Horák. J. *Využití DMÚ 25 pro prostorovou analýzu nezaměstnanosti na území okresu Nový Jičín* [online]. Aktualizace 2000 [cit. 2006-01-18]. Dostupné na www: <http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2000/Sbornik/Penaz/Referat.htm>
- 4) PEŇÁZ, T. , Horák. J., Horáková. B. *Analýza územní dostupnosti významných firem na území okresu Nový Jičín* [online]. Aktualizace 1999 [cit. 2005-11-05]. Dostupné na www: <http://gis.vsb.cz/gacr_pan/Clanky/sec2.html>
- 5) ŠVEC, P. *Teorie grafů* [online]. Aktualizace 2004 [cit. 2006-04-02]. Dostupné na www: <http://zorro.fme.vutbr.cz/graphs/graphs_flat.html#id395190>
- 6) VLČINSKÝ, J. eRDIS [online]. Aktualizace 4.4.2006 [cit. 2006-04-08]. Dostupné na www: <www.issc.cz/archiv/2006/download/prezentace/vlcinsky_ewb.ppt>

Knihy a sborníky

- 7) BRŮHA, L. *Java – hotová řešení*. 1.vyd. Brno: Computer Press, 2003. ISBN 80-251-0072-3.
- 8) FUKS, Petr. *Aplikace pro plánování rozvozu zboží*. Diplomová práce. Ostrava: VŠB-TU Ostrava, Hornicko-geologická fakulta, 2005.
- 9) PEŇÁZ, T. *Porovnání vhodnosti vektorových databází DMÚ 25, DMÚ 200 a ArcČR 500 pro model dopravní sítě v prostředí GIS*. Ostrava: VŠB-TU Ostrava, Hornicko-geologická fakulta, 2000. ISBN 80-7078-853-4.
- 10) ZELENÝ, J. , NOŽIČKA, J. *COM+, CORBA, EJB*. 1. vyd. Praha: BEN – technická literatura, 2002. ISBN 80-7300-057-1.

WWW stránky

- 11) ABC linuxu: *GNU LGPL* [online]. Aktualizace 1.11.2005 [cit.2005-12-02]. Dostupné na www: <<http://www.abclinuxu.cz/slovník/gnu-lgpl>>
- 12) ARCDATA Praha: *ArcČR 500 – základní informace* [online]. Aktualizace 2006 [cit.2006-04-03]. Dostupné na www: <<http://www.arcdata.cz/data/arccr>>
- 13) Berka Milan: *Teorie grafů a úlohy na grafech* [online]. Aktualizace 2002 [cit.2005-12-12]. Dostupné na www: <<http://home.eunet.cz/berka/o/grafy.htm>>
- 14) BORLAND: *JBuilder* [online]. Aktualizace 2005 [cit.2005-12-20]. Dostupné na www: <<http://www.borland.cz/products/jbuilder/>>
- 15) CEDA: *Silniční a uliční sítě* [online]. Aktualizace 2004 [cit.2005-12-01]. Dostupné na www: <<http://www.ceda.cz/article.asp?nArticleID=191&nDepartmentID=152&nLanguageID=1>>
- 16) ČVUT: *Jemný úvod do jazyka PL/pgSQL PostgreSQL* [online]. Aktualizace 2006 [cit. 2006-01-12]. Dostupné na www: <<http://postgresql.ok.cz/doc/plpgsql.html>>

- 17) EASTWOOD Bohemia: *Produkty* [online]. Aktualizace 2006 [cit. 2006-03-12]. Dostupné na www: <<http://www.eastwood.cz/?art=produkty>>
- 18) JGraphT: *About JGraphT* [online]. Aktualizace 2005 [cit.2005-11-20]. Dostupné na www: <<http://jgrapht.sourceforge.net/>>
- 19) JGraphT: *JGraphT Documentation* [online]. Aktualizace 2005 [cit.2005-12-18]. Dostupné na www: <<http://jgrapht.sourceforge.net/javadoc/>>
- 20) Mathematical Programming: Dijkstra's Algorithm [online]. Aktualizace 7.6.2000 [cit. 2005-01-11]. Dostupné na www: <<http://www-b2.is.tokushima-u.ac.jp/~ikedasuuri/dijkstra/Dijkstra.shtml>>
- 21) Ochodková Eliška: *Grafové algoritmy* [online]. Aktualizace 2005 [cit.2005-10-16]. Dostupné na www: <<http://www.cs.vsb.cz/ochodkova/>>
- 22) PostGIS: *PostGIS Documentation* [online]. Aktualizace 2005 [cit. 2006-01-22]. Dostupné na www: <<http://postgis.refractions.net/documentation/>>
- 23) PostgreSQL: *PostgreSQL Documentation* [online]. Aktualizace 2005 [cit. 2005-12-28]. Dostupné na www: <<http://www.postgresql.org/docs/manuals/>>
- 24) PostgreSQL: *PostgreSQL FAQ* [online]. Aktualizace 23.6.2004 [cit. 2005-12-09]. Dostupné na www: <http://www.postgresql.org/docs/faqs.FAQ_czech.html/>
- 25) Prostorová analýza dat: *Teorie grafů* [online]. Aktualizace 8.12.2002 [cit.2005-11-08]. Dostupné na www: <http://gis.vsb.cz/pad/Kap_5/kap__5_2.htm >
- 26) Quantum GIS: *About QGIS* [online]. Aktualizace 2006 [cit. 2006-03-30]. Dostupné na www: <<http://www.qgis.org/>>
- 27) Ředitelství silnic a dálnic ČR: *Silniční databanka* [online]. Aktualizace 2005 [cit.2005-11-02]. Dostupné na www: <<http://www.rsd.cz/>>
- 28) Sun: *Java™ 2 Platform Standard Edition 5.0 API Specification* [online]. Aktualizace 2005 [cit. 2006-01-08]. Dostupné na www: <<http://java.sun.com/j2se/1.5.0/docs/api/> >
- 29) T-Mapy: *Geografická data* [online]. Aktualizace 10.9.2003 [cit. 2006-04-04]. Dostupné na www: <http://www.tmapy.cz/public/tmapy/cz/_aktualne/*clanky/geograficka_data.html>
- 30) Wikipedia: *Pathfinding* [online]. Aktualizace 22.3.2006 [cit. 2006-03-28]. Dostupné na www: <<http://en.wikipedia.org/wiki/Pathfinding>>
- 31) Wikipedia: *Teorie grafů* [online]. Aktualizace 2006 [cit. 2006-01-20]. Dostupné na www: <http://cs.wikipedia.org/wiki/Teorie_graf%C5%AF>

SEZNAM POUŽITÝCH OBRÁZKŮ

Obr.1 – souvislý, neorientovaný, ohodnocený graf	5
Obr. 2 – začátek algoritmu (1. krok)	9
Obr. 3 – začátek algoritmu (2. a 3. krok)	10
Obr.4 – 1. krok algoritmu	10
Obr.5 – 2. a 3. krok algoritmu	10
Obr. 6 – dokončení algoritmu a nalezení délky nejkratší cesty do vrcholu d	11
Obr. 7 – postup výpočtu Dijkstrova algoritmu v orientovaném ohodnoceném grafu	12
Obr. 8 – ArcČR 500, vrstva silnic a vrstva sídel (zastávky)	15
Obr. 9 – Silniční databanka Ostrava, vrstva silnic a zastávek	17
Obr. 10 – Streetnet, ukázka nespojitosti	19
Obr. 11 –StreetNet, ukázka rozdělení komunikace na dvě linie	21
Obr. 12 – relační schéma databáze	28
Obr. 13 – vrstva uzlů a komunikací	31
Obr. 14 – generování záznamů v tabulce hran	32
Obr. 15 – grafické rozhraní	36
Obr. 16 – zobrazení výsledku hledání nejkratší cesty	38
Obr. 17 – sekvenční diagram	39
Obr. 18 – diagram tříd	40
Obr. 19 – diagram aktivit	41
Obr. 20 – hledání nejkratší cesty v eRDIS	42

SEZNAM POUŽITÝCH TABULEK

Tab. č.1 – Vysvětlení pojmů z teorie grafů	6
Tab. č.2 – Matice sousednosti	7
Tab. č.3 – Matice incidence	7
Tab. č.4 – návrh průměrných rychlostí dle kategorie silnice	14
Tab. č.5 – kategorizace silnic v datové sadě Streetnet	20