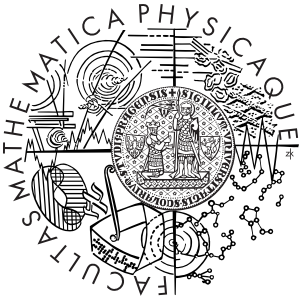
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Vladislav Martínek

Efektivní hledání nejkratších cest v sítích hromadné přepravy osob

Katedra softwarového inženýrství

Vedoucí diplomové práce: RNDr. Michal Žemlička, Ph.D.
Studijní program: Informatika, softwarové systémy

2010

Děkuji vedoucímu diplomové práce, RNDr. Michalu Žemličkovi, Ph.D., za obětavou spolupráci a cenné připomínky; Mgr. Pavlu Machkovi a Mgr. Martinu Marešovi, Ph.D., za odborné konzultace; Českému úřadu zeměměřickému a katastrálnímu za poskytnutí vektorových dat pěších cest;

Tato práce byla částečně podporována programem "Informační společnost" v rámci projektu 1ET100300517 a Grantovou agenturou České republiky grantem číslo 201/09/0983.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne Vladislav Martínek

Obsah

Název práce: Efektivní hledání nejkratších cest v sítích hromadné přepravy osob
Autor: Vladislav Martínek
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí diplomové práce: RNDr. Michal Žemlička, Ph.D.
e-mail vedoucího: Michal.Zemlicka@mff.cuni.cz

Abstrakt: Hledání nejkratší cesty je jeden z nejstudovanějších grafových problémů; má mnoho zajímavých aplikací v nejrůznějších odvětvích. Jedním z odvětví je hromadná přeprava osob, kde délka cesty závisí na jízdních řádech spojů realizujících cestu.

Cílem této práce je najít efektivní algoritmus pro hledání nejkratší cesty v sítích hromadné dopravy a implementovat jej v knihovně, která bude použitelná i na přenosných zařízeních. V průběhu implementace budou prozkoumány možnosti předzpracování jízdních řádů a využití heuristik pro urychlení hledání cesty.

Klíčová slova: síť hromadné dopravy, hledání nejkratší cesty, redukce grafu, pěší navigace

Title: Efficient shortest path search in the public transportation networks
Author: Vladislav Martínek
Department: Dept. of Software Engineering
Supervisor: RNDr. Michal Žemlička, Ph.D.
Supervisor's e-mail address: Michal.Zemlicka@mff.cuni.cz

Abstract: The search for the shortest path is one of the most studied graph problems with interesting applications in various fields. One such field is human mass transportation, where the path length depends on the time tables of the traffic relations, which implements the path.

Goal of this study is to find efficient algorithm for the shortest path search in human mass transportation network and implement it in the library, which will be also useable on portable devices. The possibilities of time tables preprocessing and use of heuristics on search acceleration will be explored during implementation.

Keywords: public transport network, shortest path search, network simplification, pedestrian navigation

Kapitola 1

Úvod

Systém hromadné přepravy osob bývá pro pohyb obyvatel ve velkých městech klíčový. V opravdu velkých metropolích může být tento systém natolik složitý, že pouhá znalost jízdních řádů nemusí cestujícímu stačit k naplánování efektivní cesty. K plánování efektivní cesty se používají aplikace, které nad databází jízdních řádů hledají cestu vyhovující zadaným parametrům. S rostoucí složitostí dopravního systému se výrazně zvyšuje výpočetní náročnost plánování cest nad tímto dopravním systémem. V závislosti na použitém algoritmu je složitost výpočtu typicky superlineární vzhledem k velikosti dopravní sítě.

1.1 Orientace na přenosná zařízení

Aplikace pro plánování cest v městské hromadné dopravě jsou široce rozšířené na platformách pro mobilní zařízení. PDA zařízení mají řadu specifik, která ovlivnila jejich vývoj z hlediska výpočetní a paměťové kapacity. Historie například ukázala, že Moorův zákon[1] u PDA příliš nefunguje. V případě PDA zařízení dochází z dlouhodobého hlediska k relativně mírnému rozvoji výpočetního výkonu. Jedním z důvodů může být přímá úměra mezi výpočetním výkonem a spotřebou energie. Napájecí zdroje přenosných zařízení bývají omezené a vývoj technologií pro úsporu energie prozatím nedokáže nárůst výkonu přijatelně kompenzovat.

Dokument "PDAzarizeni.doc" na přiloženém CD se vlastnostmi PDA a jejich historickým vývojem zabývá podrobněji. Prezentace "PDA.ppt" na přiloženém CD se ve svém úvodu zabývá právě Moorovým zákonem. Při celkovém návrhu aplikace pro přenosná zařízení lze vycházet z těchto přístupů:

"Thin-client" je druh klient-server architektury, kde většina výpočetní zátěže spočívá na straně serveru a klient většinou pouze zobrazuje výsledky.

Výpočetně náročné operace by v tomto případě prováděla centralizovaná služba. Funkčnost aplikace by tak byla plně závislá na připojení k poskytovateli dané služby. Objem přenesených dat by se odrážel na určité režii na straně poskytovatele připojení. Tento přístup může být úspornější z pohledu dodavatele služby. Jedním z důvodů volby této architektury může být například snazší vývoj aplikace.

[1] Moorův zákon je v podstatě empirické pozorování, které říká, že počet tranzistorů integrovaných do jednoho čipu se zdvojnásobí každých 18 měsíců. Existují i jiné varianty tohoto zákona.

“Thick-client” je druh klient-server architektury, kde je většina zátěže přesunuta na klientské zařízení.

Pokud bude možné provádět plánování cest přímo v zařízení, bude funkčnost aplikace nezávislá na jakémkoli připojení. Aktualizace dat lze provádět ve vybraných okamžicích, kdy je připojení k poskytovateli služby k dispozici. To může uživateli přinést snížení nákladů za provoz aplikace a zároveň snížení zátěže pro přenosové médium mezi uživatelem a poskytovatelem při udržení určité kvality poskytované služby.

Zejména ve druhém případě, tedy v případě návrhu aplikace “thick-client”, je výpočetní a prostorová složitost algoritmu pro plánování cest klíčovou otázkou.

1.2 Cíle práce

Cílem této práce je implementovat knihovnu pro efektivní plánování cest v síti hromadné přepravy osob. Přitom výsledná implementace by měla být nezávislá na zvolené architektuře a měla by být použitelná jak pro přenosná zařízení, tak pro stolní počítače.

Implementace základního řešení

Cílem je vytvořit základní verzi knihovny, která bude umožňovat vyhledávání spojení v jízdních řádech. Implementace by měla vycházet z dosavadních poznatků o fungování dopravní sítě. Dále by měla být použitelná v PDA zařízeních a zároveň na stolních počítačích.

Redukce sítě hromadné dopravy

Cílem je prozkoumat využití metod popsaných v [13] pro účely vyhledávání spojení v MHD. Nalezené postupy pak aplikovat na konkrétní síti hromadné dopravy a ověřit jejich použití na reálných datech.

Komplexní plánování cest

Cílem této části bude rozšířit implementaci plánování spojení o vyhledávání pěších cest. Tím se otevírají nové možnosti plánovaní, které je třeba prozkoumat. Implementace by měla být navržena vzhledem k použití na PDA tak, aby šetrně hospodařila s paměťovými a výpočetními prostředky při zachování určité kvality nalezených cest.

1.3 Předpokládané znalosti čtenáře

Předpokládá se základní znalost teorie grafů, zejména grafových algoritmů.

1.4 Terminologie

1.4.1 Zkratky

DPP - Dopravní podnik hlavního města Prahy, a.s.

DORIS - dopravní řídicí a informační systém. Telematický systém sledující v Praze provoz tramvají.

GPS - Global Positioning System - globální polohový systém – systém okamžitého určení zeměpisných souřadnic a nadmořské výšky.

GSM - Global System for Mobile Communications - standardizovaný systém pro používání mobilních služeb.

MHD - městská hromadná doprava.

PDA - Personal Digital Assistant - osobní digitální pomocník - druh přenosného zařízení, původně určený pro organizaci času a kontaktů. Rozšířeným pojmenováním je též "kapesní počítač", což lépe odráží současnou úroveň vyspělosti těchto zařízení.

ROPID - Regionální organizátor Pražské integrované dopravy.

SMS - Short Message Service - služba krátkých textových zpráv - služba GSM sítě umožňující výměnu krátkých textových zpráv mezi účastnickými zařízeními.

ZABAGED - Základní báze geografických dat. Digitální geografický model území České republiky, který poskytuje "Český úřade zeměměřický a katastrální".

1.4.2 Obecné pojmy

V obecném textu mohou zde uvedené pojmy nabývat různých významů, proto je nutné vymezit jejich význam v tomto textu.

Číslo linky je označení dané linky srozumitelné pro cestující. Typicky se jedná o číselné označení.

Linka je vymezení všech dopravních spojů, který jezdí pod daným číslem linky.

Trasa je posloupnost zastávek. Může se jednat i o okružní trasu.

Směr linky nebo spoje je dán koncovou stanicí dané trasy. Spoj může typicky jet po určité trase oběma směry.

Spoj je konkrétní vůz určité linky jedoucí v určitý čas po určité trase.

Spojení představuje možnost jízdy nějakým spojem hromadné dopravy mezi dvěma zastávkami.

Časová tabulka pro daný spoj je část jízdního řádu spoje určující dobu odjezdu spoje z dané zastávky.

Navazující spoj je spoj, který neodjede ze zastávky dříve než mu určuje jízdní řád a zároveň neodjede dříve než na zastávku přijede spoj, na který navazuje.

Převlékací linka je sada spojů, které shodně mění své číslo linky jinde než v konečné stanici.

Zastávka je označení stanice hromadné dopravy uvedené v jízdním řádu.

Zastávkový ostrůvek je místo určené pro nástup do nebo výstup cestujících ze spoje. Někdy také označované jako "stání". Jedna zastávka může mít více ostrůvků, například pro různé směry. Ostrůvky dané zastávky od sebe mohou být i značně vzdálené.

Polyline je lomená čára. Tvoří ji alespoň dva body, jejichž propojením vznikne lomená linie.

1.4.3 Nově zavedené pojmy

Komplexní navigační systém je aplikace pro plánování cest kombinující více druhů dopravních prostředků. Cílem je nabídnout uživateli co nejúplnější plán cesty.

Plán cesty nebo spojení je posloupnost pokynů. Jednotlivé pokyny by měly obsahovat dostatek informací k tomu, aby se cestující, který se jimi řídí, dostal z výchozího bodu cesty do cílového. Plán spojení si lze představit jako časovou osu a jednotlivé položky plánu jako bloky na této ose.

- **Pevný blok** v plánu cesty je taková část cesty, kterou nelze přesouvat v čase. Počáteční okamžik, kdy se cestující vydá na tento úsek cesty, je pevně daný. Typicky jde o jízdu spojem, který se řídí jízdním řádem. V tom případě jsou uživateli předány informace o době odjezdu ze zastávky, čísle linky, směru linky, době jízdy a případně další upřesňující detaily.
- **Volný blok** v plánu cesty je taková část cesty, kterou je možné posouvat v čase. Cestující má určitou volnost ve volbě okamžiku, kdy vyrazí na daný úsek cesty. Typicky jde o pěší úsek, kde nezáleží na okamžiku započetí chůze. V tom případě jsou uživateli předány podrobnosti o pěší trase, předpokládaný čas začátku chůze, celková doba chůze v tomto úseku a případně další upřesňující detaily.
- **Mezera mezi bloky** v plánu cesty představuje časovou prodlevu mezi přesuny cestujícího. Typicky jde o čekání na spoj. V tom případě je uživateli sdělena délka čekání, tento údaj je v podstatě jen orientační.

Sublinka je linka, která jezdí po pevně dané trase s pevně určeným časovým rozvrhem. Jízdní řád linky může pro různé časové okamžiky určovat různou trasu spoje. Každá z těchto tras představuje spolu s danou sadou okamžiků v jízdním řádu jednu sublinku.

1.4.4 Formální definice

Dále v textu je popisována práce s některými matematickými strukturami, které reprezentují reálný svět. Způsob reprezentace je blíže popsán v daných kapitolách, zde je uvedena základní definice.

Zastávkový graf reprezentuje síť zastávek hromadné dopravy. Jedná se o orientovaný multigraf.

- **Vrchol grafu** nechť reprezentuje zastávkový ostrůvek. Každá zastávka může mít více zastávkových ostrůvků.
- **Hrana** nechť reprezentuje přímé spojení mezi dvěma sousedními zastávkovými ostrůvky, které nebude přerušeno zastavením daného spoje na jiném zastávkovém ostrůvku. Orientace hrany je dána směrem jízdy spoje mezi zastávkovými ostrůvky, které hrana spojuje.

Pěší graf reprezentuje síť ulic, chodníků, pěšin a dalších pěších cest. Jedná se o orientovaný graf.

- **Vrchol grafu** nechť reprezentuje průsečík, začátek nebo konec pěší cesty.
- **Hrana** nechť reprezentuje pěší cestu, která není přerušena křížením s jinou pěší cestou.

1.5 Dostupná data

Před zahájením práce na samotném řešení zadané úlohy bylo nutné zvážit a promyslet, jaká data bude řešená úloha vyžadovat. Tento krok je poměrně důležitý pro úspěšné dokončení práce. Pokud by se až v průběhu řešení ukázalo, že aktuálně dostupná data nejsou vhodná, mohlo by to vést k nežádoucímu zdržení nebo k výraznému snížení kvality výsledné práce.

1.5.1 Síť MHD

Rozhodli jsme se zaměřit na hromadnou přepravu osob ve městech, kde je dopravní síť relativně hustá. Dopravní síť ve městech má relativně vysoký stupeň větvení a frekvence odjezdů spojů je ve většině případů také relativně vysoká. Z důvodu znalosti prostředí jsme zvolili město Praha.

Data jízdních řádů poslouží k tvorbě zastávkového grafu, proto by data měla obsahovat trasy jednotlivých linek. Dále bude na základě dat jízdních řádů vytvořena ohodnocovací funkce, která určuje dobu jízdy a dobu čekání, proto by vstupní data měla obsahovat časové tabulky.

Jízdní řády poskytované cestujícím obsahují dostatek informací potřebných pro tvorbu zastávkového grafu a ohodnocovací funkce. Proto jsme se rozhodli čerpat data z jízdních řádů, které poskytuje v elektronické podobě "Dopravní podnik hlavního města Prahy, a.s." (zkráceně DPP).

1.5.2 Síť pěších cest

Jelikož máme k dispozici jízdní řády pro město Praha, měla by data pěších cest co nejlépe pokrývat možnosti pěší přepravy v této oblasti. Existuje řada datových sad pro navigaci automobilů, ale ty jsou pro použití v rámci naší úlohy nevhodné. Pokud by v městském prostředí platilo, že chodníky se vyskytují na okraji každé ulice, pak by navigační data pro automobily mohla být za jistých okolností a po jistých úpravách použitelná.

ZABAGED

Navigační data přímo určená pro chodce jsme neměli k dispozici. Rozhodli jsme se použít pro tvorbu navigačních dat vektorové mapové podklady ZABAGED, které "Český úřad zeměměřický" poskytuje bezplatně pro studijní účely. Vektorová podoba dat zaručuje, že budeme schopni z mapových podkladů vytvořit síť pěších cest.

Mapové podklady ZABAGED jsou relativně podrobné co se týče pokrytí pěších cest. Výhodou je, že tyto podklady lze použít na zobrazení mapy pro uživatele. V datech však chybí některé podrobnosti o charakteru pěší cesty, jako je bariérovost úseku, druh povrchu, převýšení a další. V datech nejsou odlišeny přechody pro chodce. Bez těchto doplňujících údajů je možné se obejít, přestože se může stát, že v některých případech bude výsledek plánování nepřesný.

Závažnější problém, na který jsme narazili až při testování mapových funkcí, je absence mimoúrovňového křížení cest. Veškeré cesty jsou v mapách reprezentovány pomocí polyline. Kříží-li se dvě cesty, odpovídá tomu uzel v síti polyline. Je-li křížení mimoúrovňové, tedy jedna cesta je položena výše než druhá, ke křížení fakticky nedochází a uzel by zde být neměl. V datech ZABAGED jsme narazili nejméně na jeden případ křížení pěších cest, které nebyly na stejné úrovni.

1.5.3 Pozice zastávkových ostrůvků

Pro realizaci komplexního vyhledávání cest je potřeba propojit síť hromadné dopravy a síť pěších cest. Zastávkové ostrůvky jsou body, ve kterých se obě sítě stýkají. V případě dat, které máme k dispozici, nastávají dva zásadní problémy.

Zastávkové ostrůvky v mapě

Mapové podklady ZABAGED neobsahují zastávkové ostrůvky MHD. Bylo tedy nutné získat pozice zastávkových ostrůvků z jiného zdroje a doplnit je do mapových podkladů.

GPS pozice zastávkových ostrůvků jsme získali ze serveru "POI.cz". Kromě samotného zanesení zastávkových ostrůvků do mapových podkladů bude potřeba napojit nové body do stávající sítě pěších cest.

Mapování zastávek z jízdních řádů na zastávkové ostrůvky v mapě

Zastávkové ostrůvky v jízdních řádech nejsou v rámci jedné zastávky prakticky nijak odlišeny. V jízdních řádech bývají ostrůvky upřesněny typicky pouze názvem zastávky a spojem, pro který je daný jízdní řád určen. Aby bylo možné provázat vyhledávání pěších cest s vyhledáváním spojení v hromadné dopravě, je potřeba provést mapování mezi zastávkovými ostrůvky v mapě a zastávkami v jízdních řádech.

1.6 Existující aplikace

V současné době existují dva vyhraněné druhy aplikací pro přenosná zařízení, které umožňují chodcům plánování cesty. Dále pak existuje několik aplikací, které jsou schopné zajistit komplexní plánování cesty.

1.6.1 Navigátor pro chodce

V této skupině aplikací opomineme navigace, které neplánují cestu, ale plní pouze funkci kompasu a ukazují směr k cíli. Budeme uvažovat klasické navigace, kde si uživatel zvolí libovolná místa na mapě a aplikace najde vhodnou cestu mezi zvolenými cíli. Nalezenou cestu pak uživatel prochází a aplikace kontroluje jeho polohu a uděluje uživateli rady, kterým směrem má jít.

Aplikace pro své plné využití nutně potřebují GPS přijímač. V některých mobilních zařízeních je vestavěný i digitální kompas, který dokáže určit orientaci zařízení aniž by se pohybovalo[2]. Tyto aplikace vyžadují pro svůj provoz mapové podklady. Umožňují uživateli orientovat se i v naprosto neznámém terénu, jsou tedy vhodné pro pěší turistiku.

Nokia Maps

Tato aplikace je určena pro mobilní telefony. Umožňuje propracovanou navigaci chodců včetně hlasových instrukcí informujících o průběhu cesty. Pro plánování cest nelze přímo využít hromadné dopravy.

Google Maps

Tato aplikace je přístupné přes webové rozhraní. Umožňuje plánování cest mezi obecnými polohami. Dopravním prostředkem může být automobil nebo vlastní chůze. V případě vyhledávání pěší cesty je aplikace v současné době ve verzi beta a podle upozornění aplikace je možné, že na nalezené pěší cestě budou chybět chodníky nebo stezky pro chodce.

Aplikace je připravena pro plánování cest za použití prostředků hromadné dopravy a chůze. V současné době je toto plánování k dispozici jen v několika menších městech.

1.6.2 Vyhledávač spojení v hromadné dopravě

Druhou skupinou aplikací jsou nástroje pro vyhledávání spojení v hromadné dopravě. Často také poskytují další funkce související s prohlížením jízdních řádů. Tyto aplikace dovolují plánovat cestu typicky mezi dvěma zastávkami. Pro úspěšné naplánování cesty je nutná znalost názvu výchozí a cílové zastávky. Lze tedy předpokládat, že uživatel by měl alespoň zběžně znát prostředí, ve kterém se pohybuje, a měl by znát cestu vedoucí na výchozí zastávku. Od aplikace čeká spíše informaci o čase odjezdu a průběhu spojení v hromadné dopravě.

[2] U navigací pro automobily není kompas zapotřebí, orientace se určuje dynamicky z rozdílu GPS pozic během jízdy. Chůze je příliš pomalý pohyb na to, aby bylo možné s dostatečnou přesností rozlišit pohyb uživatele od nepřesnosti GPS.

Pro plánování cesty je zde chůze potřeba pouze v případě, že je nutné přestoupit mezi prostředky hromadné dopravy. Zmíněné přestupy jsou pevně nastavené. Pokud z nějakého důvodu spoj nejede na určitou zastávku a pěší přestup není nastavený, aplikace neumí na tuto zastávku nalézt cestu.

IDOS

Jedná se o celostátní systém jízdních řádů. K systému je možno přistupovat přes webové rozhraní. Existuje řada variant webových formulářů od různých dopravců, které jsou pouze front-endem pro IDOS.

Aplikace umožňuje základní funkce pro vyhledávání spojení mezi zadanými zastávkami. Navíc umožňuje vyhledávání na příjezd a dovoluje zadat přestupní místa. Výhodou tohoto systému je integrace různých dopravních sítí, což umožňuje kombinovat například vnitrostátní dopravu s městskou hromadnou dopravou.

Smartřády

Jedná se o off-line aplikaci pro platformu Windows Mobile. Data jízdních řádů pocházejí ze systému IDOS, jejich užívání je zpoplatněné. Aplikace poskytuje příjemnější uživatelské rozhraní než webové formuláře pro IDOS, avšak základní funkce jsou stejné. Výhodou aplikace je nezávislost na připojení.

1.6.3 Komplexní vyhledávače

Existují aplikace, které kombinují vyhledávání pěších cest a plánování spojení pomocí hromadné dopravy nebo jiných prostředků. Často se tyto aplikace snaží vytvořit co nejpodrobnější plán cesty tak, aby vyhověla nejrůznějším požadavkům uživatele. Délka cesty se pak stává pouze jedním z požadavků na podobu cesty.

Navitime

Jedná se o on-line aplikaci pro mobilní zařízení rozšířenou především v Japonsku. Aplikace dokáže sestavovat plán cesty mezi obecnými polohami za použití široké škály prostředků od chůze až po leteckou přepravu. Navíc poskytuje cestujícímu užitečné podrobnosti o průběhu cesty[3].

[3]Například které dveře soupravy metra jsou nejblíže k výstupu.

Kapitola 2

Analýza úlohy

Podívejme se na naší úlohu zcela obecně. Vstupem jsou jízdní řády a požadavky uživatele. Výstupem by měl být plán časově nejkratší cesty vyhovující zadání uživatele. Uživatel typicky mění pouze své požadavky na podobu plánované cesty – tedy počáteční, cílovou zastávku, čas odjezdu a různá nastavení. Naopak sada jízdních řádů se mění mnohem méně často. Navíc data jízdních řádů by měla být pro všechny uživatele aplikace, která bude naší úlohu řešit, shodná. Až na rozdíly vzniklé aktualizací jízdních řádů.

Základní otázkou naší úlohy je nalezení časově nejvýhodnějšího spojení za pomoci prostředků hromadné přepravy osob a chůze. Zvolíme-li vhodnou grafovou reprezentaci sítě hromadné dopravy, lze na tuto část úlohy pohlížet jako na grafovou úlohu nalezení nejkratší cesty.

Úlohu lze řešit dvěma základními přístupy. První možností je nalézt cesty mezi všemi vrcholy a při dotazu jen vrátit výsledek. V sítích hromadné dopravy je situace komplikována tím, že hodnota hrany není konstantní, ale závisí na aktuálním čase. Předvypočítat výsledky by tedy znamenalo nalézt cesty mezi všemi vrcholy pro určitý časový interval. Tento přístup je vhodný pokud dotazů je relativně mnoho a je k dispozici dostatečná paměťová a výpočetní kapacita k tomu, aby bylo možné dostatečně rychle reagovat na změny v datech.

Druhou možností je hledat nejkratší cestu přímo podle zadaných parametrů. Tento přístup je vhodný, je-li dotazů relativně málo a je-li možné najít odpověď v přijatelném čase. Výhodou proti předchozímu případu je, že proměnné ohodnocení hran nepředstavuje tak výraznou komplikaci.

Při vyhledávání nejkratší cesty v dopravních sítích se často používá nějaký druh prohledávání do šířky. Při implementaci je potřeba se vyrovnat s vlastnostmi sítě hromadné dopravy. Více v kapitole 3.

Složitost algoritmů používaných pro vyhledávání nejkratších cest v grafu je typicky superlineární vzhledem k počtu vrcholů a hran. Pokud se nám podaří snížit mohutnost vstupního grafu, velmi výrazně se to projeví na rychlosti přímého výpočtu.

V oblasti hromadné přepravy osob narážíme na řadu jevů, které mohou mít zásadní vliv na volbu reprezentace grafu a způsob řešení úlohy nalezení nejkratší cesty ve zvolené grafové reprezentaci. Tyto jevy se budeme snažit popsat a analyzovat jejich vliv na řešení zadané úlohy. Jsme dokonce přesvědčeni, že některé specifické vlastnosti hromadné přepravy osob lze využít pro optimalizaci vyhledávání nejkratších cest v sítích hromadné dopravy. Více v kapitole 4.

Při přestupech mezi jednotlivými prostředky hromadné dopravy se cestující po-

hybují pěšky. Plánování spojení lze rozšířit o vyhledávání pěších cest vedoucích mezi zastávkami. Dále lze plánování rozšířit na vyhledávání cest vedoucích z obecné výchozí pozice na zastávku hromadné dopravy, případně ze zastávky do cíle. Vyhledávání spojení rozšířené o pěší cesty zpřesní plánování přestupů a přinese nové možnosti pro vyhledávání spojení. Což může znamenat i nalezení nových, rychlejších spojení. Tím se dostáváme do oblasti komplexních navigací a vyvstává otázka, jak propojit síť hromadné dopravy se sítí pěších cest, aby bylo vyhledávání spojení v MHD stále dostatečně efektivní. Více v kapitole 5.

2.1 Problematika vyhledávání spojení v jízdních řádech

Nahlížejme na problém nalezení spojení v síti hromadné přepravy osob jako na grafovou úlohu. Síť hromadné přepravy osob lze reprezentovat zastávkovým grafem. Aby bylo možné mezi všemi možnými spojeními vybrat to nejvýhodnější, je nutné zvolit jejich ohodnocení. Pro jednoduchost nechť je tímto ohodnocením doba potřebná pro dosažení cíle.

Závislost na aktuálním čase

Plánování spojení prostředky hromadné dopravy je typicky závislé na aktuálním čase. Spoje by měli odjíždět ze zastávky v předem dané časové okamžiky. Cestující může ale na zastávku přijít obecně kdykoli, a potom stráví určitý čas čekáním na příjezd požadovaného spoje. Tuto dobu lze určit na základě času plánovaného příchodu cestujícího na zastávku a času odjezdu spoje. Z toho plyne, že časovou délku daného úseku cesty, jsme schopni určit až v okamžiku, kdy je tento úsek naplánován jako součást nějakého spojení. Celková délka nalezeného spojení tedy závisí na výchozím čase a na konkrétní podobě cesty.

Dobu potřebnou k čekání na spoj budeme dynamicky připočítávat k hodnotě hrany, která tvoří plánované spojení.

Doba jízdy spojem

Kromě doby čekání na spoj je potřeba dynamicky určovat i samotnou hodnotu hrany. Hodnota hrany grafu by měla představovat dobu potřebnou pro jízdu spojem v daném úseku trasy spoje. Tato hodnota se může měnit v závislosti na denní době. Při plánování možných cest je potřeba udržovat pojem o aktuálním čase v rámci sestavovaného plánu každé jednotlivé cesty.

Hodnotu hrany budeme určovat na základě aktuálního času v rámci plánovaného spojení a jízdního řádu platného v daný okamžik.

Jízdní řád po půlnoci

Jízdní řád bývá typicky různý pro jednotlivé dny v týdnu. Pokud probíhá vyhledávání cesty kolem půlnoci, je otázka, zda se po překročení půlnoci řídit novým jízdním řádem, pokud je jízdní řád pro nový den v týdnu odlišný. Navíc s novým dnem se může časový údaj dostat mimo platnost jízdního řádu.

Prozatím jsme nenalezli případ, kdy by byl jízdní řád po půlnoci odlišný od následujícího dne v týdnu. Budeme se tedy orientovat jízdním řádem, podle kterého spoj vyjel ze své počáteční zastávky.

Posun začátku spojení

Plán nejkratší nalezené cesty lze ve výsledku dále upravovat, aby byl pro uživatele přijatelnější. Vyskytuje-li se například na první zastávce v plánu cesty nenulové čekání na první spoj, lze výchozí čas plánu posunout a čekání tak eliminovat. Tím se ale změní celková délka cesty.

Uživateli nabídneme možnost provádění zmíněné úpravy výsledného plánu spojení. Za nejkratší nalezené spojení budeme nadále považovat takové spojení, pomocí kterého se cestující dostane do cíle nejdříve.

Závislost podoby plánovaného spojení na čase

Závislost podoby výsledné cesty na výchozím okamžiku přináší jeden zajímavý jev. Pokud budeme hledat nejkratší cestu pro dva relativně blízké okamžiky, výsledné trasy mohou mít značně odlišný průběh. Praktický důsledek pro cestujícího je ten, že pokud cestující zmešká nějaký spoj v itineráři plánované cesty, je potřeba zbývající část cesty přeplánovat. Nový plán může vést jinudy nebo může používat jiné spoje než plán původní. Problém nastává, pokud nová cesta začíná z jiného zastávkového ostrůvku, než na kterém se cestující právě nachází. V tom případě by totiž musel cestující přejít na vzdálený ostrůvek a není zaručeno, že by spojení z tohoto ostrůvku stihl. Tento problém by nastal, pokud bychom v zastávkovém grafu nerozlišovali jednotlivé ostrůvky v rámci zastávky, ale pouze celé zastávky.

Tuto situaci řešíme při dostatečně podrobném zadání od uživatele – na vstupu obdržíme geografickou polohu nebo přímo ostrůvek, na kterém se uživatel nachází. Pokud uživatel zadá pouze název zastávky bez bližší specifikace ostrůvku, není možné problému předejít.

Obrázek 2.1: Proměnlivost plánu cesty
Zadání zvýrazněných plánů cest se liší pouze ve výchozím čase – o jednu minutu.

Obrázek 2.1 ukazuje dva různé plány cesty mezi stejnými místy. Jediný rozdíl v požadavcích na oba plány byl počáteční čas spojení. První plán cesty začíná o jedinou minutu dříve než druhý plán. Situace ukazuje, jak výrazně se může lišit průběh volby spojení v plánu výsledné cesty v závislosti na aktuálním čase.

Počet cílů plánované cesty

Spojení je běžně vyhledáváno mezi dvěma zastávkami – počáteční a koncovou. Aplikace zajišťující vyhledávání typicky dovolují zadat jednu nebo více zastávek, přes které má spojení vést. Za určitých podmínek lze úlohu rozložit na hledání nejkratších cest v uživatelem stanovených úsecích. Předpokládejme, že cestující nebude ze spoje vystupovat a specifikovaným cílem chce jen projíždět. Při plánování každého úseku cesty zvlášť bude pro každý úsek plánováno spojení odděleně. Může stát, že bude výhodnější zvolit odlišné spojení, pokud bychom si udržovali přehled nad všemi úseky cesty.

Předpokládejme, že v každém ze zadaných cílů je potřeba ze spoje vystoupit a čekat na zadaném místě určitou nenulovou dobu. Pak můžeme považovat jednotlivé úseky cesty za oddělené, a tak k nim také přistupovat při plánování. Počet cílů v zadání nebude z naší strany omezen. Umožníme uživateli definovat časovou prodlevu mezi jednotlivými úseky cesty.

Přestupy

Aby bylo možné v zastávkovém grafu vyhledávat spojení kombinující všechny přípustné linky, je potřeba doplnit přestupové hrany. Jsou to hrany reprezentující pěší přesuny mezi jednotlivými ostrůvky uvnitř zastávky, případně mezi ostrůvky jiné zastávky. Podle nároků uživatele na přesnost plánu vyhledávaného spojení, lze pěší přestupy buďto odhadovat, nebo přesně určovat na základě pěšího grafu. Více v kapitole 2.2 o propojení pěší navigace a systému hromadné dopravy osob.

Přestupové hrany budou do grafu zastávek doplněny tak, aby umožňovaly vyhledávání efektivních cest. Hodnota hran a jejich přesná podoba se bude odvíjet od použitých dat.

Nespolehlivý přestup

Jízdní řád určuje pouze předpokládaný okamžik odjezdu ze zastávky. V reálné dopravě dochází k odchylkám od jízdního řádu. Problém může nastat, dojde-li k naplánování přestupu mezi dvěma spoji tak, že nezbývá žádná rezerva pro případně zpoždění prvního spoje. Pokud není dopravcem zaručena návaznost spojů, může dojít k tomu, že uživatel nestihne navazující spoj.

Situace je podobná, dochází-li k přestupu mezi dvěma linkami v rámci jednoho ostrůvku. Přijedou-li dva spoje na stejnou zastávku ve stejný okamžik podle jízdního řádu, nelze zaručit jaká bude reálná situace a který spoj přijede jako první. Pokud bychom uvažovali přestup mezi těmito spoji ve stejnou minutu, přestup ze spoje, který přijede jako první, by byl možný, ale již nelze zaručit přestup ze spoje, který přijede na zastávku jako druhý ve stejnou minutu. Z jízdního řádu nelze vyčíst pořadí příjezdu spojů. Může se i stát, že vejde-li se na daný ostrůvek jen jeden ze spojů[1].

[1] Ať už z důvodu malého ostrůvku nebo dlouhého vozidla daného spoje.

Tyto přestupy ošetříme tak, že budeme počítat s jistou rezervou potřebnou pro výstup cestujícího ze spoje. Tím zamezíme plánování přestupu mezi spoji, které odjíždějí z daného ostrůvku dle jízdního řádu ve stejný okamžik.

Délka nástupiště

Doposud jsme uvažovali zastávkové ostrůvky jako jednoduché body. Nástupní plocha ostrůvku může však být značně dlouhá. Pokud je v rámci plánované cesty nutné přejít dlouhé nástupiště, může to znamenat až několikaminutový rozdíl v délce cesty.

Problém je komplikovaný tím, že je potřeba rozhodnout, zda je vůbec vhodné celé nástupiště přecházet. To je v případě, že je pro cestujícího výhodné vystupovat na opačném konci soupravy než na jakém nastoupil. Tím, že přejde na druhý konec nástupiště, ušetří čas po výstupu ze soupravy a může stihnout jiný navazující spoj, než kdyby musel po výstupu ještě přecházet celé nástupiště.

Daný problém má smysl jen v případě, že cestující přijde na nástupiště přesně v okamžik odjezdu spoje. Pokud přijde dříve, je schopen nástupiště přejít v rámci doby čekání na spoj. Stejně tak pokud by cestující čekal na navazující spoj, lze tuto dobu zkrátit o přechod nástupiště po výstupu z předchozího spoje na cestě.

Pokud bychom znali podrobné parametry nástupiště a dopravního prostředku, mohli bychom cestujícímu poskytnout cennou radu, ohledně místa nástupu do vozidla v rámci nástupní plochy. Takto podrobná data nemáme k dispozici, a proto považujeme řešení této problematiky nad rámec naší úlohy.

Odchylky od jízdních řádů

Jízdní řády určují předepsané časy odjezdu jednotlivých spojů, které se mohou od skutečných časů odjezdu lišit. Mohou zpravidla nastávat dva druhy odchylek. První mohou nastávat relativně často a mohou být relativně malé. Tyto mohou být způsobeny aktuální hustotou dopravy, počasím, stavem vozovky nebo jinými relativně předvídatelnými vlivy.

Druhé jsou výjimečné události většího dopadu. Tyto nelze předvídat a způsobují relativně velké odchylky od jízdních řádů. Typicky mohou vyvolat výpadek části přepravní sítě a z toho plynoucí dočasný zásah dopravce do jízdních řádů. Například zavedení náhradních spojů nebo změnu trasy spojů, kterých se nastalá událost dotýká.

Pro odhad spolehlivosti jednotlivých spojů by bylo potřeba statisticky vyhodnotit data o průběhu jízd daných spojů. Plánování cest s ohledem na spolehlivost spojení by vyžadovalo použití vícekriteriálního vyhledávání. Posouváme toto rozšíření úlohy nad rámec současného zadání. Při výběru spojení budeme uvažovat jen jediné kritérium – délku cesty.

2.2 Problematika navigace chodců za použití prostředků MHD

Při plánování spojení za pomoci prostředků MHD je nutné počítat s pěšími přestupy mezi zastávkovými ostrůvky. Pokud budeme tyto přestupy odhadovat, plánování nikdy nebude zcela přesné, protože jak je zmíněno v 2.1 i malá časová odchylka může mít výrazný dopad na výsledný plán cesty.

Naopak pokud budeme mít podrobné informace o možnostech pěšího přesunu mezi zastávkovými ostrůvky, lze kromě přesnějšího plánování docílit i vyšší efektivity využití dopravního systému. Přidáním možnosti přechodu na větší vzdálenost otevřeme možnosti plánování nových spojení. Lze nalézt případy, kdy pěší přesun na vzdálenější zastávku umožňuje nalezení časově kratšího spojení, než jakého bychom dosáhli při čekání na původní zastávce.

Pokud navíc umožníme zadávání výchozího nebo cílového bodu jako obecné geografické polohy, lze při vyhledávání nejkratší cesty brát v úvahu spojení ze všech zastávek, na které je cestující schopen dojít. Dostáváme tak mnohem více možností, jak výslednou cestu naplánovat, což znamená, že výsledný plán nebude horší než při zadání jen jediné výchozí zastávky.

Na úlohu budeme stále pohlížet jako na grafovou úlohu, ale v tomto případě je potřeba si uvědomit, že kombinujeme dva grafy. Respektive máme zastávkový graf reprezentující síť MHD a máme pěší graf reprezentující veškeré pěší cesty v daném městě. Hrany pěšího grafu nechť jsou ohodnoceny dobou potřebnou k překonání daného úseku cesty, čímž je zajištěna určitá kompatibilita s ohodnocením zastávkového grafu.

Propojení vyhledávacích sítí

Jak síť hromadné dopravy, tak síť pěších cest mohou být značně velké. Při jejich kombinaci může celková velikost prohledávané sítě narůst nad výpočetně únosnou mez. Efektivnost celkového plánování tedy silně závisí na šetrnosti zvoleného řešení.

Základní otázkou naší úlohy je nalézt takovou metodu napojení obou sítí, aby výpočetní ani paměťová náročnost vyhledávání nepřekročila možnosti PDA. Současně by nemělo dojít k poklesu kvality výsledků plánování nejkratšího spojení.

Závislost na aktuálním čase

Při plánování pěší trasy není většinou potřeba brát příliš ohled na aktuální čas, protože chodec může typicky na daný úsek cesty vyrazit kdykoli. Některé úseky pěší cesty ale mohou být průchozí jen v určitou denní dobu. V tom případě zde aktuální čas hraje roli stejně jako v hromadné dopravě osob. Lze nalézt i složitější případy, kdy průchod některými úseky může být výrazně obtížnější a pomalejší v určitých periodicky se opakujících okamžicích.

Pokud budeme mít k dispozici odpovídající data, lze určovat hodnotu hran dynamicky podobně jako u sítě hromadné dopravy. Pokud data nebudou k dispozici, lze si vystačit se statickým ohodnocení hran pěšího grafu.

Pevné a volné bloky cesty

Při plánování kombinované cesty je potřeba rozlišovat úseky cesty, které jsou pevně ukotvené v čase a volné úseky. Jízda spojem hromadné dopravy musí být typicky započata v určitý okamžik, který je pevně daný v čase příslušným jízdním řádem. Oproti tomu pěší přesun může uživatel zahájit většinou v libovolný okamžik. Z hlediska časového plánu cesty mohou mezi jednotlivými úseky vznikat mezery, které mohou odpovídat například čekání na spoj hromadné dopravy. Vhodným posunutím volných úseků lze tyto mezery minimalizovat nebo přesunout podle potřeb uživatele. Aby bylo možné správně kombinovat jednotlivé bloky cesty, je potřeba

znát jejich délku. Navíc u pevných bloků je daný jejich počátek a nelze s nimi hýbat.

Posunem jednotlivých bloků v rámci plánu cesty se budeme zabývat pouze okrajově.

Dispozice uživatele

Při plánování kombinované cesty je potřeba určit délku každého bloku předtím, než je naplánován do cesty. V případě hromadné dopravy se doba určuje na základě aktuálního času a platného jízdního řádu spoje, který realizuje danou část cesty. V případě chůze se doba určuje na základě doby, kterou bude uživatel potřebovat pro překonání daného úseku. V obou případech budou výraznou roli hrát parametry uživatele. Tyto parametry budou ovlivňovat volbu úseků cesty a v případě chůze mohou ovlivňovat i délku jednotlivých úseků.

Ve složitějších úsecích pěší cesty, jako jsou různé bariéry nebo úseky s převýšením, se neobejdeme bez dodatečné informace o charakteru úseku. Pro uživatele s omezenými možnostmi pohybu jsou tyto informace zcela zásadní pro naplánování schůdné cesty. Podobné informace mohou hrát roli i při výběru dopravního prostředku. Například spoje, které jsou realizovány prostředky určenými pro přepravu osob se sníženou pohyblivostí, bývají v jízdních řádech označeny. Pokud uživatel vyžaduje nalezení bezbariérové cesty, je nutné naplánovat cestu pouze za pomoci odpovídajících spojů. V souvislosti s tímto požadavkem uživatele je nutné vybírat také odpovídající pěší cesty pro přestupy tak, aby celá cesta byla bezbariérová.

Požadavek uživatele nemusí být striktní, cestující může bezbariérové cesty pouze upřednostňovat. V takovém případě je třeba předem určit do jaké míry jsou tyto úseky upřednostňovány na úkor ostatních parametrů cesty. Praktickým příkladem může být cestující s kočárkem, který raději zvolí nízkopodlažní spoj, ovšem pouze pokud na takový spoj nebude čekat příliš dlouho nebo pokud nedojde k nějakému nevhodnému přeplánování cesty.

Jedna ze základních dispozic uživatele, kterou se budeme v naší úloze řídit, je rychlost chůze. Z datových podkladů typicky známe délku a průběh úseku pěší cesty. Na základě rychlosti chůze a charakteru cesty se pak dopočítává doba potřebná pro překonání úseku.

Plánování cesty na základě preferencí uživatele vyžaduje víceparametrické vyhledávání, což prozatím posouváme nad rámec naší úlohy.

2.3 Data

V této části textu se budeme snažit rozebrat, jaká data budeme pro naši úlohu potřebovat a jaké transformace bude třeba s danými daty provést.

2.3.1 Data MHD

Základní datovou sadou pro vytvoření grafu zastávek jsou jízdní řády. Z těchto dat by mělo být možné získat množinu zastávkových ostrůvků, které budou reprezentovány vrcholy zastávkového grafu. Dále by mělo být možné získat z dat hrany, které reprezentují možnost jízdy spojem mezi zastávkovými ostrůvky. Navíc bude

zapotřebí mít přehled o jednotlivých linkách a časovém itineráři jejich cesty, abychom mohli přiřazovat hranám dobu jízdy spojem jako základní výchozí hodnotu hrany. A v neposlední řadě je potřeba získat z dat časové tabulky, podle kterých budeme určovat dynamickou část ohodnocení hrany - dobu čekání na příjezd spoje.

Všechny tyto základní údaje lze typicky získat z jízdních řádů, které jsou k dispozici každému cestujícímu na označníku zastávky. Jediný problém nastává při rozlišování zastávkových ostrůvků, protože v jízdních řádech jsou typicky jen názvy zastávek, více v 2.3.3.

Nepravidelnosti v trasách linek

Jízdní trasa linky se může v určité vybrané okamžiky měnit. Tyto změny bývají zachycené v jízdním řádu, pokud je dopravce zná dopředu a plánoval tyto změny. Abychom byli schopni tyto změny pokrýt, bylo by potřeba dynamicky upravovat trasu linky podle výluk uvedených v jízdním řádu.

Jednou z cest, jak tyto odchylky v trase linky ošetřit, je zavádět sublinky pro každou jednotlivou variantu trasy dané linky. Problém je, že takových sublinek může vzniknout relativně mnoho. Na druhou stranu jízdní řády sublinek, které reprezentují neobvyklé trasy dané linky, budou velice řídké. Vyhledávací algoritmus je potřeba přizpůsobit tomuto jevu tak, aby nedocházelo k nežádoucímu zvýšení výpočetní náročnosti zavedením sublinek.

Předem známé výluky

V dopravní síti mohou nastávat výjimečné situace. Pokud je lze předvídat nebo je dopravce plánuje, pak by měly být zahrnuty v jízdních řádech. Takové výluky bývají často dlouhodobého charakteru. Dopravce může zavést výlukové jízdní řády s dobou platnosti odpovídající trvání výluky. Původní jízdní řád po tuto dobu pozbývá platnosti.

Jelikož se vyhledávání spojení řídí aktuálně platným jízdním řádem, není potřeba těmto výlukám věnovat zvláštní pozornost.

Nenadálé výluky

Mohou však nastat i výjimečné situace, které předvídat nelze a které vedou k okamžitým změnám v dopravní síti. Teprve až dopravce detekuje výjimečnou situaci, je schopen na ni reagovat a změnit aktuální jízdní řád. Až poté je aplikace schopna, pokud má tato data k dispozici, plánovat spojení s ohledem na výjimečnou situaci v dopravní síti.

Změny v dopravní síti vyvolané nepředvídanými situacemi bývají krátkodobého charakteru. Navíc je nutné se spolehnout na včasnou reakci dopravce. Informace o jednotlivých dotčených spojích nebývají k dispozici. Je proto vhodné dát uživateli možnost zásahu do vyhledávání tak, aby mohl vyloučit určité prostředky nebo část dopravní sítě. Přeplánování trasy pak uživateli nabídne alternativní spojení.

Aktualizace

Jízdní řády se relativně často mění v čase. Pokud data jízdních řádů nejsou spravována centrálně, je na uživateli, aby zajistil aktualizaci dat. V případě plánovaných

změn v dopravě je tento úkon zcela prostý. Problém nastává u neplánovaných změn. Jedná se typicky o drobné úpravy jízdních řádů, dočasné výluky spojů a podobně.

Problém je, jak tyto neplánované změny propagovat směrem k uživateli v co nejkratším čase od jejich vzniku. Jelikož se typicky jedná o lokální změny jízdních řádů, není nutné měnit celou datovou sadu, ale je vhodné zvolit mechanismus rozdílových aktualizací.

Vyhledávání bez záruky

Hromadná přeprava osob je typicky předem plánována a jízdní řády mají určitou předem danou platnost. Problém nastává pokud aplikace nemá pro požadované vyhledávání k dispozici platný jízdní řád. Logická reakce je nedovolit použití spoje, pro který nemáme platný jízdní řád k dispozici. To odpovídá situaci, kdy spoj přestane jezdit po vypršení doby platnosti.

Předpokládejme, že cestující ze zkušenosti ví, že spoj jezdí i nadále, ale aplikace nemá jeho aktuální jízdní řád. Lze dále předpokládat, že nový jízdní řád spoje, který nemáme k dispozici, bude mít alespoň přibližně podobnou charakteristiku jako starý jízdní řád. Pak má smysl poskytnout uživateli možnost vyhledávání na základě neplatného jízdního řádu s tím, že nalezené spojení je bez záruky.

Spolehlivost cesty

Významným parametrem, který by kromě délky cesty mohl uživatele zajímat, je spolehlivost cesty. Na první pohled se v tomto ohledu neobejdeme bez dodatečných dat od dopravce. Na druhou stranu se můžeme na spolehlivost spojení dívat jako na frekvenci jízd dané linky.

Některé pevné bloky v plánu cesty se mohou v relativně krátkých časových intervalech opakovat a mohou se tak chovat podobně jako volné bloky. Tomuto chování může odpovídat například metro ve špičce, které jezdí velmi často. Případné nestihnutí takového spoje nemusí znamenat výrazné přeplánování celé cesty.

Zajímavou výzvou může být naplánování takové cesty, kde nevadí nestihnutí některého nebo všech spojů. Respektive nestihnutí určitého spoje povede k minimálnímu zdržení vzhledem k ostatním plánovaným cestám. Tuto problematiku posouváme mimo rámec naší úlohy.

Zpoždění spoje

Díky propracovaným monitorovacím systémům mívají dopravci velice přesný přehled o poloze svých spojů. Vědí tedy s poměrně vysokou přesností, jaké mají jejich spoje aktuální zpoždění[2].

Pokud bychom byli schopni tato data od dopravce přímo propagovat až k uživateli, mohlo by to pomoci k reálnějšímu plánování spojení. Nemuseli bychom se opírat jen o předpokládanou dobu odjezdu spojů v jízdních řádech, ale mohli bychom přímo odhadnout reálnou dobu odjezdu.

Na základě statistického sběru těchto dat by bylo možné získat informace o pravděpodobnosti, že daná linka v daném úseku cesty v danou denní dobu nabere

[2]Z neoficiálních zdrojů víme, že například pro řidiče tramvají platí, že je daleko větší prohřešek, když odjede ze zastávky dříve než mu určuje jízdní řád, než když odjede později.

určité zpoždění[3].

Tato data nemáme bohužel momentálně k dispozici. Zmíněné problematice se bude věnovat výzkum navazující na tuto práci.

Návaznost spojů

V jízdních řádech se může objevit informace, že daný spoj čeká na nějaký jiný spoj v jedné ze zastávek na své trase. Znamená to, že daný spoj navazuje na jiný spoj. Pokud existuje mezi dvěma spoji návaznost, pak by měli mít cestující zaručeno, že v případě zpoždění jim navazující spoj neujede. Pokud budou navazující spoje zaneseny v jízdních řádech, budeme se snažit je zohlednit při plánování cest.

2.3.2 Mapové podklady

Pro tvorbu pěšího grafu a jeho napojení na zastávkový graf potřebujeme dostatečně podrobné mapové podklady. Aby bylo možné vytvořit snadno graf na základě mapy, je vhodné vycházet z vektorového formátu mapy. Pro účely vyhledávání cest nás zajímají pouze ty mapové objekty, které představují pěší cesty.

Jedním z výrazných problémů je nedostupnost mapových podkladů vhodných pro plánování pěších cest. Většina existujících mapových podkladů není dostatečně podrobná vůči chodcům. V první řadě je nutné, aby mapová data obsahovala skutečně data pro chodce - tedy chodníky, přechody, pěší zóny, pěšiny a jiné cesty použitelné pro chodce. Dále je potřeba, aby data obsahovala informace o různých typech překážek.

Přechody pro chodce

Bylo by vhodné, aby mapové podklady obsahovaly také údaje o přechodech, případně přechodech se signalizačním zařízením, na kterých je potřeba počítat se specifickým intervalem pro překonání daného úseku.

Pro legální přecházení silnice mají přechody pro chodce zásadní význam. Ve většině zemí není dovoleno chodcům přecházet silnici v místech, kde se do určité vzdálenosti od nich nachází přechod pro chodce. Situace bývá dále ztížena, pokud je chodník ohraničený například zábradlím nebo jinou těžko překonatelnou bariérou.

Určit místo přechodu silnice mimo přechod je samostatný problém. Pokud však mapové podklady nebudou dostatečně podrobné, není možné se jím vůbec zabývat.

Prostorové mapy

Dále by bylo vhodné, aby mapové podklady byly prostorové, tedy obsahovaly údaje o převýšení jednotlivých úseků cesty. Vzhledem k odlišným dispozicím chodců by bylo také vhodné, aby mapové podklady obsahovaly údaje o bariérovosti jednotlivých úseků cesty[4].

[3]Mohou existovat i úseky, kde lze nabrané zpoždění naopak eliminovat.

[4]Důležité je zejména odlišit vysoké obrubníky chodníků nebo schody, které pro různé uživatele mohou a nemusejí být překážkou. Mohou znamenat například jen změnu rychlosti pohybu.

Mimoúrovňové křížení

Další důležitou informací je odlišení mimoúrovňového křížení pěších cest[5]. V případech, kdy nelze volně přecházet mezi úrovněmi cesty, by nesprávně označené křížení cest mohlo vést k chybám v plánování cest. Jednotlivé úrovně cest je potřeba rozlišit také při zadávání výchozího bodu.

2.3.3 Propojení pěší sítě a sítě hromadné dopravy

Problémem úzce souvisejícím s výchozími daty je propojení mapových podkladů pro chodce a sítě hromadné dopravy. Propojení je potřeba provést v obou směrech a namapovat na sebe příslušné zastávkové ostrůvky.

Zastávkové ostrůvky v mapě

Je potřeba přesně určit uzly v pěších cestách, odkud se lze napojit na síť hromadné dopravy. Ideální je situace, kdy jsou v mapových podkladech přímo zastávkové ostrůvky napojené na síť chodníků. Pokud ostrůvky nejsou v mapových podkladech, je potřeba je doplnit i s cestami, po kterých se tam chodec může dostat z okolní sítě pěších cest. Propojení jednotlivých ostrůvků se sítí pěších cest je důležité i z hlediska hledání přestupů mezi spoji.

Mapování zastávek v síti hromadné dopravy na ostrůvky

Síť hromadné dopravy se typicky vytváří na základě tras jednotlivých linek. Každá zastávka v itineráři dané linky je identifikována názvem ostrůvku, na kterém spoj zastavuje. Pokud existuje více ostrůvků stejného názvu, vzniká problém jak vytvořit jednoznačné mapování mezi názvy ostrůvků v jízdním řádu a ostrůvky v mapě. Tento problém nenastane, pokud jsou zastávky v jízdních řádech identifikovány zeměpisnou souřadnicí[6].

[5] například v případě mostů a podchodů

[6] Dopravce by měl znát pozice ostrůvků na kterých jeho spoje zastavují.

Kapitola 3

Základní řešení

Základem celého řešení je návrh a implementace knihovny pro efektivní vyhledávání spojení v síti hromadné přepravy osob. Na základní řešení bude navazovat další práce, jejímž předmětem bude hlubší optimalizace výpočtu a prozkoumání přínosů komplexní navigace.

Vyhledávací knihovna je vyvíjena pro prototyp aplikace pro PDA, což umožňuje její testování přímo na cílové skupině zařízení. Navíc máme pro testování k dispozici relevantní sadu dat, což podstatně zvyšuje kvalitu celé práce. Na základě zkušeností z vývoje jsou následně volena taková řešení, která co nejlépe splňují zadanou úlohu. To nám dává možnost rozvíjet práci jak v teoretické, tak praktické rovině.

3.1 Analýza

Návrh vyhledávací knihovny by měl zohledňovat omezenou výpočetní a paměťovou kapacitu přenosných zařízení. Z tohoto pohledu je důležitý návrh datových struktur uchovávajících data jízdních řádů, který ovlivní jak paměťové tak výpočetní nároky celé aplikace. Důležitá je také samotná volba vyhledávacího algoritmu. Vyhledávací mechanismus by mělo být možné přizpůsobit specifickým vlastnostem sítě hromadné dopravy.

3.1.1 Reprezentace sítě MHD

Na systém hromadné dopravy lze pohlížet jako na orientovaný multigraf s dynamicky ohodnocenými hranami. Přirozený a často používaný přístup je zvolit jako vrcholy jednotlivé zastávky MHD, hrany pak představují přímé spojení mezi zastávkami.

Zcela jiný přístup může být reprezentovat samostatným vrcholem každý odjezd spoje. Tento přístup je příliš robustní pro naší úlohu a podle [12] vykazuje znatelně vyšší obtížnost prohledávání.

Jeden ze zajímavých přístupů může být zvolit jako vrcholy jednotlivé trasy linek a hrany pak představují možnost přestupu mezi linkami. Tento přístup má své opodstatnění a přináší řadu výhod a nových možností. Pro účely naší úlohy, zejména pozdější redukce grafu a napojení na pěší síť, by byl tento přístup neobratný.

Definujme zastávkový graf pro účely naší úlohy takto:

Zastávkový graf

Zastávkový graf reprezentuje síť zastávek hromadné dopravy. Jedná se o orientovaný multigraf. Definujeme ho následujícím způsobem.

Vrchol grafu nechť reprezentuje zastávkový ostrůvek. Každá zastávka může mít více zastávkových ostrůvků.

Hrana nechť reprezentuje přímé spojení mezi dvěma sousedními zastávkovými ostrůvky, které nebude přerušeno zastavením daného spoje na jiném zastávkovém ostrůvku. Orientace hrany je dána směrem jízdy spoje mezi zastávkovými ostrůvky, které hrana spojuje.

Ohodnocení hrany je předpokládaná doba jízdy pseudolinky mezi zastávkami, které hrana spojuje. Doba jízdy se může lišit podle denní doby[1].

3.1.2 Algoritmy

Existuje řada algoritmů pro hledání nejkratší cesty v grafu. Některé algoritmy si kladou podmínky na podobu grafu a vlastnosti ohodnocení hran v grafu.

V našem případe se jedná a graf reprezentující síť hromadné dopravy. Hrany v našem grafu jsou ohodnoceny časem. Tudíž ohodnocení hran je nezáporné a můžeme vyloučit záporné cykly. Těmito předpoklady se budeme řídit při výběru algoritmů. Nemá smysl uvažovat obecnější algoritmy, které například zvládají záporné ohodnocení, jelikož zpravidla nedosáhneme nižší výpočetní složitosti ani snazší implementace.

Při výběru algoritmu je potřeba brát v úvahu dynamické ohodnocení hran grafu. Implementace zvoleného algoritmu se stane základem mechanismu pro vyhledávání spojení v MHD.

Dijkstrův algoritmus

Dijkstrův algoritmus [3] vyžaduje, aby vstupní orientovaný graf měl nezáporné ohodnocení hran. Zastávkový graf je ohodnocen časem a nezáporné hrany tedy má. Mějme graf $G = (V, E)$. Následující text je převzatý z [4, strana 86].

Nejprve si vysvětlíme význam proměnných, které budeme používat. Množina $T \subseteq V$ je množina trvalých vrcholů, to je těch, do kterých známe nejkratší cestu. Hodnota $\mathrm{d}[v]$ je délka zatím známé cesty z s do v, a proto je horním odhadem na vzdálenost z s do v. Do pole odkud$[v]$ si ukládáme předchůdce na nejkratší cestě z s do v (odkud jsme do v přišli po nejkratší cestě). Začneme s prázdnou množinou T, vzdálenost do startu $\mathrm{d}[s]$ nastavíme na 0 a odhady vzdáleností do ostatních vrcholu na nekonečno. Postupně budeme odebírat netrvalé vrcholy s minimálním $\mathrm{d}[v]$ a prohlašovat je za

[1] Například v pracovní dny je pro tramvaje zavedena "zkrácená doba jízdy" - B a normální doba jízdy - A podle času odjezdu ze zastávky: 0:00-6:59 - B, 7:00-18:59 - A, 19:00-23:59 - B.

trvalé. Během prohlašování musíme aktualizovat odhady d[w] u ostatních vrcholů w, protože jsme se do nich mohli dostat zkratkou z nově prohlášeného trvalého vrcholu v.

$T := \emptyset$
d[s] = 0 a $\forall v \in V \setminus s$: d[v] = ∞
$\forall v \in V$: odkud[v] = nil

vytvoř prioritní frontu H z vrcholů V s prioritami d[v]
while fronta H neprázdná do
    v = DELETE_MIN(H) {vyber netrvalý vrchol s nejmenším d[v]}
    $T := T \cup v$
    for each $vw \in E$do
        if d[w] > d[v] + $c(vw)$ then
        d[w] = d[v] + $c(vw)$
        odkud[w] = v
        DECREASE_KEY(H, w) {aktualizuj haldu}

Časová složitost Dijkstrova algoritmu závisí na čase potřebném k provedení operací n × DELETE_MIN a m × DECREASE_KEY, kde n je počet vrcholů a m je počet hran v grafu. Implementujeme-li prioritní frontu pomocí binární haldy, dostaneme časovou složitost $O((n+m)\log n)$.

Floyd-Warshallův algoritmus

Floyd-Warshallův algoritmus vyžaduje, aby vstupní orientovaný graf neobsahoval záporné cykly. Zastávkový graf je ohodnocen časem a tedy má nezáporné hrany, což vylučuje existenci záporného cyklu v grafu. Nechť n je počet vrcholů a m je počet hran v grafu. Následující text je převzatý z [4, strana 88].

Vrcholy grafu očíslujeme čísly od jedničky do n. Všechny dvojice vzdáleností si nejsnáze uložíme do matice $n \times n$. Celý trik Floyd-Warshallova algoritmu spočívá v tom, že vzdálenosti nepočítáme přímo, ale v n iteracích.

V i-té iteraci spočítáme matici D^i. Hodnota $D^i[u,v]$ je délka nejkratší cesty z u do v, která smí procházet pouze přes vrcholy $\{1,2,...,i\}$. Jinými slovy $D^i[u,v]$ je délka nejkratší cesty v podgrafu indukovaném vrcholy $\{1,2,...,i\}$. V nulté iteraci začneme s maticí D^0. Hodnota $D^0[u,v]$ je délka hrany uv, pokud z u vede hrana do v, nula na diagonále a nekonečno jinak. Matice D^0 je tedy matice vzdáleností (upravená matice sousednosti, která místo jedniček obsahuje délky hran a místo nul mimo diagonálu nekonečna). V poslední iteraci skončíme s maticí D^n, která už bude obsahovat hledané vzdálenosti, protože cesty mezi u a v smí procházet přes všechny vrcholy.

...

$D := D^0$ {matice vzdáleností}
for $i = 1$ to n do
  for $u = 1$ to n do
  for $v = 1$ to n do
    if $D[u, v] < D[u, i] + D[i, v]$ then
      $D[u, v] = D[u, i] + D[i, v]$

Časová složitost Floyd-Warshallova algoritmu je $O(n^3)$. Průběh algoritmu se podobá násobení matic. Lze proto algoritmus urychlit podobně jako Strassenův algoritmus zrychluje násobení matic. Výsledná časová složitost by pak byla $O(n^{\log_2 7})$.

Volba

Floyd-Warshallův algoritmus je relativně snadný na implementaci. Jeho výhodou je, že vyhledává nejkratší cesty mezi všemi vrcholy grafu najednou. Algoritmus je hojně používán v silničních navigacích, kde je ohodnocení hran pevné. V našem případě jsou však hrany ohodnocovány dynamicky. Hodnota každého úseku cesty může být určena až na základě hodnoty předchozího úseku. Proto je použití tohoto algoritmu pro naši úlohu nevhodné.

V uvedené variantě Dijkstrova algoritmu se počítá s pevnou hodnotou hran. Nic však nebrání tomu, abychom hodnotu hran určovali dynamicky. Při plánování hrany do cesty, je její hodnota závislá na hodnotě počátečního úseku cesty. V okamžiku přidávání nové hrany do cesty známe tuto hodnotu a jsme tedy schopni určit i dynamickou hodnotu právě přidávané hrany. Znalost délky počátečního úseku cesty, kterým se k dané hraně dostaneme, je pro plánování spojení v MHD velmi důležité.

Problém s dynamickým ohodnocením hran by mohl nastat v případě, že by bylo možné, aby jedna linka jela mezi sousedními zastávkami různě dlouhou dobu. Spoje dané linky by se pak mohli navzájem předjíždět. Tuto situaci Dijkstrův algoritmus při dané reprezentaci nepokrývá.

V našem případě se díky zavádění sublinek nemění doba jízdy spoje mezi zastávkovými ostrůvky. Hodnotu hrany mění pouze připočítávaná doba čekání. Nemůže tak dojít k vzájemnému předjetí spojů jedné sublinky.

Pro řešení naší úlohy tedy nechť je zvolen Dijkstrův algoritmus. Při jeho implementaci bude potřeba provést řadu úprav tak, aby výsledný mechanismus vyhledávání co nejlépe zohledňoval specifika hromadné přepravy osob.

3.2 Datové struktury

Pro efektivní vyhledávání v síti MHD je potřeba zvolit vhodnou reprezentaci dat a použít takové datové struktury, které budou vyhledávání co nejvíce usnadňovat. Avšak ne na úkor jiných parametrů, zejména paměťové kapacity. Volba datových struktur je důležitá zejména u těch dat, ke kterým vyhledávací algoritmus přistupuje během výpočtu relativně často.

Je vhodné doplnit některé pojmy související s reprezentací reálné sítě hromadné dopravy.

Sublinka je linka, která jezdí po pevně dané trase s pevně určeným časovým rozvrhem. Jízdní řád linky může pro různé časové okamžiky určovat různou trasu spoje. Každá z těchto tras představuje spolu s danou sadou okamžiků v jízdním řádu jednu sublinku.

Pseudolinka je sublinka nebo realizace pěšího úseku.

Doba čekání je předpokládaná doba mezi příchodem cestujícího na zastávkový ostrůvek a odjezdem cestujícího z daného ostrůvku některým spojem. Tato hodnota se přičítá k hodnotě hrany v případě přestupu mezi spoji. Určuje se na základě aktuálního času a platného jízdního řádu pseudolinky, na kterou cestující čeká.

3.2.1 Zastávkový graf

Graf pro vyhledávání spojení je tvořen na základě jízdních řádů. Jízdní řád každého spoje obsahuje trasu spoje, tedy posloupnost vrcholů a hran.

Zastávkový graf vytvoříme tak, že do prázdného grafu postupně vložíme cesty představující trasy všech sublinek a přidáme hrany představující přestupy mezi spoji. Graf bude tedy obsahovat trasy všech pseudolinek.

Výjimky v jízdních řádech

Vývěska jízdního řádu může obsahovat výjimky, které mění v některých časech vlastnosti spoje. Tuto situaci lze řešit zavedením zcela nové sublinky s novým jízdním řádem, kam přeneseme všechny spoje se stejnou změnou proti původní trase linky. Takto vytvořená sublinka ponese pro uživatele stejné označení jako linka původní. Z jedné linky je možné vytvořit několik sublinek. Každá sublinka má vlastní trasu a tedy je reprezentována posloupností vrcholů a hran.

Díky tomuto postupu jsou výjimky zanesené přímo ve vstupních datech a jsou pro vyhledávání neodlišitelné od standardních linek. Není proto potřeba nijak upravovat vyhledávací algoritmus.

Přestupy

Pro realizaci pěších přesunů je vhodné zavést nové pseudolinky. Jedná se o zvláštní druh linky bez označení a jízdního řádu. Zavedení pseudolinek lze použít k rozlišení různých druhů pěších cest, například schodiště, eskalátory, výtahy a podobně.

Pěší přestupy jsou doplněny poloautomaticky nebo ručně jen na základě odhadu a znalosti prostředí. Pro základní řešení nebylo použito mapových podkladů a nebyla tedy k dispozici síť pěších cest. Ta byla využita pro doplnění přestupních tras až později, více v kapitole 5.

Mapování zastávek v síti hromadné dopravy na zastávkové ostrůvky

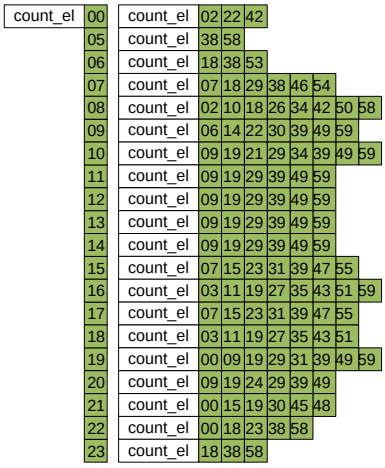
Transformace jízdních řádů na zastávkový graf je ve skutečnosti komplikovanější. Vrcholy zastávkového grafu jsou tvořeny zastávkovými ostrůvky, kdežto v jízdních řádech jsou pouze názvy zastávek. Je potřeba provést mapování názvů zastávek na zastávkové ostrůvky podle spoje, v jehož jízdním řádu se název zastávky nachází.

3.2.2 Časové tabulky

Právě časové tabulky jsou ta data, ke kterým vyhledávací algoritmus potřebuje přistupovat relativně často. Na základě časových tabulek je totiž vytvořena funkce ohodnocující hrany zastávkového grafu. Vyhledávací algoritmus k nim přistupuje prakticky pokaždé, když určuje délku hrany. K časové tabulce není potřeba přistupovat jen v případě hrany, kdy vzhledem k předchozí hraně na cestě nedochází k přestupu mezi spoji.

Vhodným návrhem datové struktury udržující časové tabulky lze výrazně snížit reálnou časovou náročnost výpočtu. Nejčastější operace nad časovými tabulkami je "zjištění doby odjezdu spoje". Vstupem je aktuální čas a výstupem by měl být čas odjezdu nejbližšího spoje, případně doba čekání nebo informace, zda spoj v určitém limitu vůbec pojede.

Reprezentace spojovým seznamem



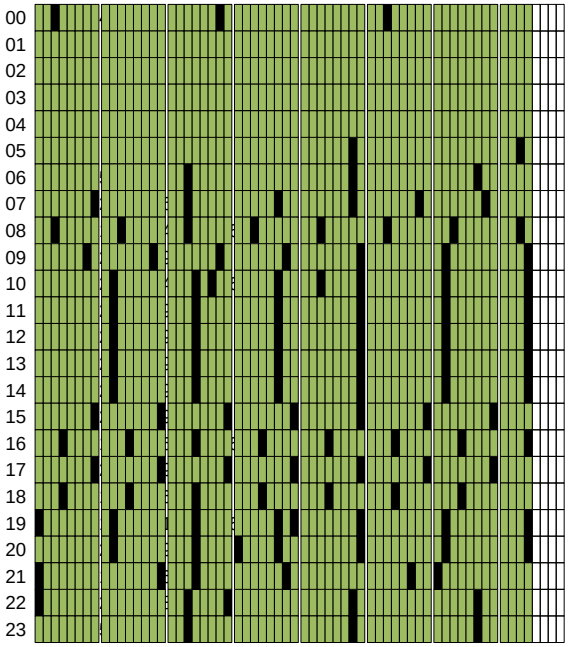
Obrázek 3.1: Reprezentace časové tabulky spojovým seznamem

Schema 3.1 znázorňuje reprezentaci časové tabulky jako spojového seznamu hodinových záznamů. Každý hodinový záznam je tvořen hodnotou dané hodiny a spojovým seznamem minutových záznamů. Minutový záznam je tvořen hodnotou minuty, kdy spoj podle jízdního řádu odjíždí v rámci dané hodiny. Minutové záznamy jsou v rámci hodinového záznamu setříděné. Hodinové záznamy jsou setříděné.

Dotaz na čas odjezdu příštího spoje je vyhodnocován postupným průchodem spojových seznamů.

Reprezentace bitmapou

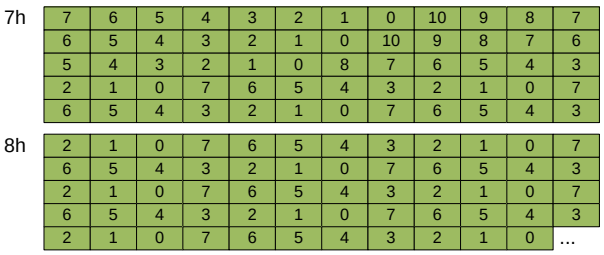
Schema 3.2 znázorňuje reprezentaci časové tabulky jako bitové mapy. Ta je tvořena 24-mi bitovými vektory představující hodinové záznamy. Každý hodinový vektor obsahuje hodnotu 1 na místě reprezentující minutu odjezdu spoje v dané hodině, hodnotu 0 jinde. Pro snazší implementaci byla jako hodinové vektory zvolena 64-bitová celá čísla.



Obrázek 3.2: Reprezentace časové tabulky bitovou mapou

Dotaz na čas odjezdu příštího spoje je vyhodnocován přímým výběrem hodinového záznamu a následným postupným průchodem bitového vektoru.

Reprezentace mapou výsledků



Obrázek 3.3: Reprezentace časové tabulky mapou výsledků

Schema 3.3 znázorňuje pouze část reprezentace časové tabulky jako mapy výsledků. Jedná se o pole 24 hodinových záznamů. Každý hodinový záznam je tvořen polem 60-ti minutových záznamů. Minutový záznam přímo uchovává dobu čekání na odjezd příštího spoje.

Maximální hodnota výsledku je limitována rozsahem zvolené číselné reprezentace minutových záznamů. Je potřeba rezervovat speciální hodnotu pro informaci, že doba čekání na daný spoj překračuje maximální hodnotu výsledku.

Dotaz na čas odjezdu příštího spoje je vyhodnocován přímým výběrem výsledku z pole.

Volba

Rychlost nalezení doby odjezdu příštího spoje je rozhodující pro volbu reprezentace časových tabulek. Na druhou stranu je potřeba brát ohled na ostatní parametry

uvažované datové struktury.

Reprezentace pomocí mapy výsledků zabírá obrovské množství paměti. Její použití pro danou úlohu naráží na několik problémů. Na druhou stranu je nejrychlejší vzhledem k dotazu, který nás nad daty zajímá nejvíce. S ohledem na paměťovou kapacitu přenosných zařízení nelze tuto reprezentaci doporučit.

Reprezentace pomocí bitmapy je výhodná v situaci, kdy časové tabulky jsou relativně husté. Tedy pro linky s vysokou frekvencí jízd. Vzhledem k ošetření výjimek zaváděním sublinek s řídkou časovou tabulkou není tato reprezentace zcela ideální. Na druhou stranu pro účely slévání jízdních řádů[2] je ideální. Slévání však není tak kritická operace jako zjištění doby odjezdu příštího spoje.

Reprezentace pomocí spojového seznamu se zdá být nejvýhodnější jak z hlediska rychlosti dotazu, tak z hlediska paměťových nároků. V řídkých časových tabulkách budou její výhody velice markantní.

Všechny uvedené typy tabulek jsou v knihovně implementované v podobě generických tříd, včetně přístupových metod. Bude-li v budoucnu některá implementace časových tabulek shledána jako výhodnější, lze v rámci knihovny velice snadno implementace zaměnit. Stejně tak lze doimplementovat případné nové reprezentace časových tabulek.

3.2.3 Redukce jízdních řádů

Uchovávat časovou tabulku jízdního řádu pro každý označník na zastávce by bylo paměťově náročné. Proto jsme hledali úspornější řešení. Jako slibný přístup se nám jeví uchovávat časové tabulky jen pro počáteční zastávku a na ostatních zastávkách časy odjezdu dopočítávat. V obecné podobě je tento postup popsán v [5]. Zejména je brán ohled na nepravidelnosti v časovém plánu spojů plynoucí z reálného světa.

Pro určení času odjezdu stačí znát dobu jízdy z počáteční zastávky. To však naráží na jeden reálný problém a tím je proměnlivá doba jízdy z počáteční zastávky. Dalším problémem může být zaokrouhlovací chyba, pokud jsou jízdní řády tvořeny s větší přesností než s jakou jsou publikovány.

Tento problém je řešen zaváděním více linek se shodným označením, pokud se odlišuje časový rozvrh průjezdu trasou. Toto řešení však není konečné, protože při velkém počtu dodatečně zavedených sublinek naroste výpočetní složitost vyhledávání nejkratšího spojení.

3.2.4 Kalendář svátků

Časové tabulky v jízdním řádu se mohou lišit pro jednotlivé dny v týdnu. Rozlišení se týká také státních svátků, kdy spoje typicky jezdí podle jízdního řádu pro neděli. Informaci o dni v týdnu, který odpovídá určitému datu, typicky udržuje systémový kalendář. Státní svátky je však potřeba udržovat samostatně. Proto je součástí vstupních dat.

[2] Slévání je technika používaná při redukci grafu, více kapitola 4.

3.3 Vyhledávací mechanismus

Vyjdeme z Dijkstrova algoritmu, jak je popsán v kapitole 3.1.2. Implementace zvoleného algoritmu pro nalezení nejkratší cesty v nezáporně ohodnoceném grafu by měla být provedena tak, aby odpovídala vyhledávání nejkratšího spojení v síti hromadné přepravy osob. Zvláště dynamické ohodnocení hran bude mít výrazný vliv na návrh vyhledávacího mechanismu. Hodnota hran závisí na aktuálním čase, a proto bude potřeba provádět plánování spojení postupně, abychom v každém okamžiku znali délku aktuálně plánovaného úseku cesty.

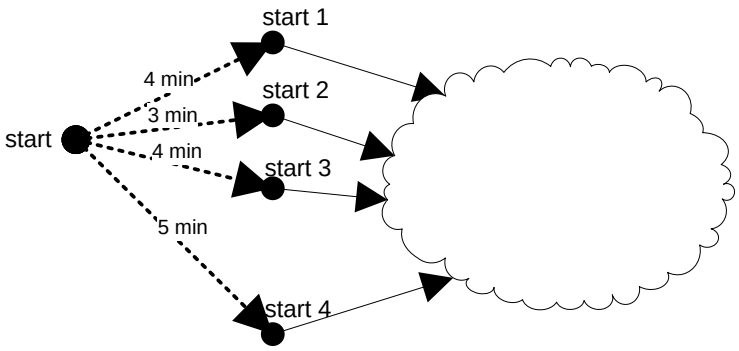
3.3.1 Délka cesty

Když mluvíme o nalezení nejkratší cesty, respektive nejkratšího spojení, uvažujeme plán takového spojení, kterým se uživatel dostane do cíle nejdříve.

3.3.2 Výchozí vrcholy

Před zahájením vlastního procházení grafu je nutné provést inicializaci. V základní variantě zvoleného algoritmu se počítá pouze s jedním výchozím vrcholem. Ten je při inicializaci jako jediný přidán do datové struktury udržující množinu trvalých vrcholů[3].

Vyhledávání je možné rozšířit tak, aby výchozích vrcholů bylo více. Tyto výchozí vrcholy navíc mohou mít různý počáteční odhad délky cesty. Původní algoritmus se tím nenaruší. Rozšíření si lze představit tak, že do vstupního grafu přidáme nový vrchol a připojíme ho hranami o příslušných hodnotách k výchozím vrcholům. Nový vrchol pak použijeme jako výchozí vrchol pro hledání nejkratší cesty. Situaci znázorňuje obrázek 3.4.



Obrázek 3.4: Více výchozích vrcholů
Vrchol "start" je uměle přidaný výchozí vrchol, přes který vyhledávací mechanismus přejde ke skutečným výchozím vrcholům.

Zmíněné rozšíření v praxi představuje situaci, kdy může cestující zahájit cestu z několika různých zastávkových ostrůvků. Nalezená nejkratší cesta bude vždy začínat na jednom z výchozích ostrůvků.

[3]Tedy vrcholů s nejlepším možným odhadem, tento odhad už nelze zlepšit, je trvalý.

Analogickou úpravu lze provést s cílovým vrcholem. Tyto úpravy umožní zadávat více výchozích i cílových vrcholů, mezi kterými má být nalezena jediná nejkratší cesta.

3.3.3 Nalezení výsledné cesty

Výsledná cesta se rekonstruuje zpětně až po dosažení cílového vrcholu, tedy až po dokončení vyhledávání. V každém vrcholu se ukládá ukazatel na vrchol, ze kterého byl daný vrchol dosažen. V multigrafu tato informace k rekonstrukci nalezené cesty nestačí a potřebujeme znát přesně hranu, kterou jsme daného vrcholu dosáhli. V případě sítě hromadné dopravy je potřeba si vedle předchozího vrcholu také pamatovat pseudolinku, kterou se cestující dopravil do aktuálního vrcholu.

3.3.4 Dynamické ohodnocení hran

Jak je popsáno v základním variantě algoritmu v kapitole 3.1.2, hodnota hran grafu je dána ohodnocovací funkcí. Lze předpokládat, že tato ohodnocovací funkce může být rozšířena o další proměnnou, kterou může být čas. V tom případě by jedna hrana mohla nabývat různé hodnoty pro různé časové okamžiky. Pokud zůstanou zachovány vlastnosti funkce[4], pak by měla být zachována korektnost algoritmu. Tato vlastnost by měla být zachována vzhledem k tomu, že ohodnocovací funkce je definována časovými tabulkami jízdních řádů.

Ohodnocení hrany tvoří pevná složka, která je určena dobou jízdy spoje mezi vrcholy dané hrany. Při hledání nejkratší cesty v síti hromadné dopravy je nutné započítat do délky cesty nejen hodnotu hrany, ale také dobu čekání na spoj, pokud přestupujeme nebo nastupujeme do nějakého dopravního prostředku. Aby bylo možné detekovat přestupy, je potřeba si v každém probraném vrcholu pamatovat, jakou pseudolinkou jsme daného vrcholu dosáhli. Pokud se do stávající cesty přidává nová hrana, ale tato hrana je realizována jinou pseudolinkou, než kterou jsme dorazili do posledního vrcholu ve stávající cestě, připočítává se k hodnotě nové hrany i doba čekání.

Přestupy mezi spoji mohou znamenat přestup z jednoho spoje na jiný v rámci jednoho nástupiště nebo mezi dvěma od sebe vzdálenými nástupními ostrůvky. V prvním případě uživatel pouze vystoupí z jednoho spoje a zůstane čekat na nástupním ostrůvku, než přijede požadovaný spoj. Ve druhém případě uživatel vystoupí ze spoje, přechází pěšky na jiný ostrůvek a až tam čeká na svůj spoj. V obou případech může být situace komplikována momentální dopravní situací, jak je popsáno v 3.3.5.

3.3.5 Minuta navíc při vystoupení ze spoje

Časy v jízdních řádech nejsou doby příjezdu na zastávku. Je proto vhodné počítat s časovou rezervou na výstup ze spoje. Může se totiž stát, že spoj, kterým bychom měli pokračovat v jízdě, následuje těsně za spojem, ze kterého jsme vystoupili.

Například autobusy se mohou navzájem předjíždět. Může tak nastat situace, kdy spoj, kterým jsme měli pokračovat v cestě, se dostane před spoj, ze kterého právě vystupujeme. Přestup pak nebude možný.

[4]tedy pro danou hranu a daný časový okamžik bude funkce vracet jedinou hodnotu

U vozidel bez možnosti předjetí, jako jsou například tramvaje, tato konkrétní situace nehrozí. Obecně ale může nastat stejný problém při příliš těsné návaznosti spoje po přestupu.

Abychom situaci předešli, je vyhledávací mechanismus modifikován tak, že vždy počítá s minimální dobou čekání na další spoj. Tato doba je ve vyhledávací knihovně implicitně nastavená na 1 minutu.

3.3.6 Vyhledávání na příjezd

Typické zadání pro vyhledávání spojení je čas odjezdu. Očekávaný výsledek je trasa, která vede k cílovému bodu v co nejkratším čase. Alternativní zadání je čas příjezdu. Očekávaný výsledek je potom trasa, kterou se dostaneme do cíle dříve než ve zvolený čas příjezdu nebo alespoň ve stejný okamžik.

Jeden z přístupů k problému je vytvořit odhad délky cesty, podle něj pak odhadnout potřebnou dobu odjezdu a zkusit vyhledat cestu podle tohoto odhadu. Výsledná cesta by nemusela splňovat podmínku, že dorazíme do cíle dříve než v danou dobu příjezdu.

Zvolili jsme jiný přístup, který spočívá ve využití zpětných hran. Každý vrchol zastávkového grafu, obsahuje seznam dopředných hran a navíc i seznam zpětných hran. Dopředná hrana reprezentuje jízdu spojem **ze** zastávky, zpětná hrana reprezentuje jízdu spojem **do** zastávky. Díky těmto hranám jsme schopni procházet graf zastávek proti směru jízdy a zajišťovat tak skutečné hledání nejkratší cesty.

Komplikace

Toto řešení sebou přináší kromě vyšší spotřeby paměti i další komplikace. Prohledávání grafu v protisměru není přirozené pro principy fungující v hromadné dopravě. Bylo nutné opravit dobu jízdy spoje z počáteční stanice, podle které se určují časy odjezdů. Abychom zachovali použitelnost zvoleného algoritmu pro vyhledávání, bylo nutné počítat odhad cesty v kladných číslech. Místo abychom hledali čas příštího spojení, je nutné hledat čas předchozího spojení.

3.3.7 Posunutí doby odjezdu

Pokud uživatel není právě na cestě, ocení, pokud je začátek cesty odsunut na dobu, kdy by měl vyrazit, aby bez čekání nastoupil do prvního spoje. Vyhledávací mechanismus sám o sobě nemění danou dobu počátku cesty. Ve výsledné cestě se tedy objevuje čekání na první zastávce. Pokud bychom tuto dobu čekání přesunuli na začátek cesty, mohli bychom ji zcela vypustit. Tento posun aplikace provádí, přičemž zároveň opravuje dobu odjezdu.

3.3.8 Aktualizace dat

Aktualizační data mají podobu rozdílových aktualizací, kdy se provádí přidávání a odebírání položek tak, abychom ve výsledku dostali aktuální datovou sadu. Soubory s aktualizacemi se stahují ze serveru v archivu, aby se ušetřil objem přenášených dat.

Aktualizace binárních datových souborů uživatelské aplikace je proces, při kterém se výchozí datová sada změní na cílovou datovou sadu. Tyto postupy by měly vést ke shodné cílové datové sadě:

1. Aktualizovat výchozí datovou sadu na cílovou.
2. Vygenerovat cílovou datovou sadu.

V uživatelském prostředí se rozlišují dva druhy aktualizací. V tomto textu budeme za "aktualizaci" považovat vždy první z nich, tedy "rozdílovou aktualizaci".

Rozdílová aktualizace je doplnění binárních souborů výchozí datové sady tak, aby odpovídaly cílové datové sadě.

Úplná aktualizace je nahrazení původní datové sady cílovou sadou. V tomto případě se binární datové soubory nijak neupravují, pouze se soubory původní datové sady přepíší soubory cílové datové sady.

Pro účely rozpoznání o jakou datovou sadu se jedná, jsou datové sady označeny verzí. Verze obsahuje datum sestavení jízdních řádů a zastávkového grafu. Prozatím nepředpokládáme častější aktualizace než jedenkrát denně. Pokud by bylo potřeba provádět aktualizace častěji, stačí odpovídajícím verzi o jemnější určení času než je prosté datum.

Údaj verze v datovém souboru nese informaci o dni sestavení datové sady, identifikaci mapových podkladů, ke kterým se data vztahují, a další údaje. Aktualizace je vždy určena pro konkrétní verzi výchozích dat. Pokud verze výchozích dat neodpovídá, aktualizace se neprovede. Musí se shodovat celá verze, tedy i identifikace mapové sady a identifikace zdroje dat.

Kapitola 4

Redukce grafu

Jak již bylo zmíněno v kapitole 1.1, přenosná zařízení mají řadu specifik. Pro naši úlohu je omezujícím faktorem výpočetní a paměťová kapacita. Na stolních počítačích jde rozvoj těchto kapacit víceméně v souladu s tak zvaným Moorovým zákonem. Na přenosných zařízeních je rozvoj výrazně pomalejší, zejména v oblasti výpočetního výkonu a kapacity primární paměti. Na druhou stranu v oblasti sekundární paměti lze pozorovat u přenosných zařízení rozvoj odpovídající Moorovu zákonu.

Základní řešení zajišťuje na současných zařízeních dostatečně efektivní vyhledávání nejkratšího spojení pro dopravní sítě, které jsou hustotou srovnatelné s MHD v Praze. Pro řádově složitější sítě by uvedené řešení nemuselo být v prostředí přenosných zařízení schůdné, především z hlediska času potřebného pro výpočet. Je proto vhodné hledat metody, jak dále snížit výpočetní složitost vyhledávání spojení.

Výsledky použité redukce byly publikovány v článku [9].

4.1 Analýza

Zajímavá metoda redukce železniční sítě je uvedena v [13]. Za určitých podmínek, lze tuto metodu aplikovat na síť hromadné dopravy osob a výrazně tak snížit výpočetní náročnost vyhledávání nejkratšího spojení.

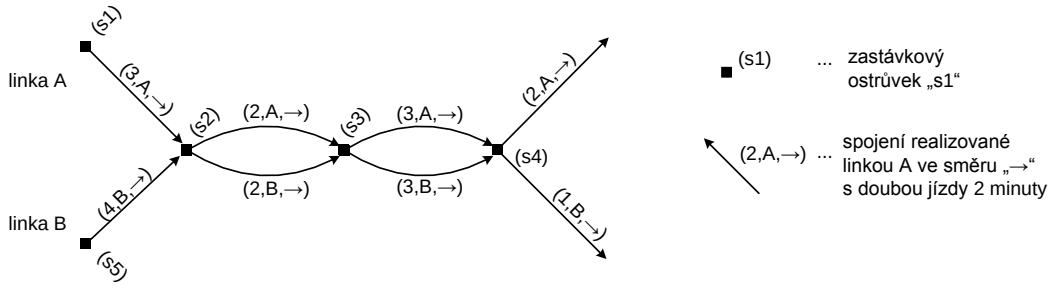
Tato redukce je v obecném případě NP-těžká úloha, jak je popsáno v [7]. Naštěstí existuje heuristika popsaná v [8], pomocí které jsme schopni nalézt uspokojivé řešení v přijatelném čase.

4.2 Řešení

Abychom snížili velikost vstupních dat, budeme se snažit vhodnými úpravami redukovat určité hrany a vrcholy ve vstupním grafu. Redukce hran je dosaženo tím, že nahrazujeme vždy několik původních hran jedinou novou hranu. Nová hrana přitom plně zastupuje všechny původní hrany při hledání nejkratší cesty. K redukci vrcholů dojde tak, že po redukci hran mohou zůstat v grafu izolované vrcholy, které nemá smysl pro hledání nejkratší cesty udržovat.

Důležité je, aby při nalezení cesty v redukovaném grafu bylo možné snadno rekonstruovat cestu v grafu původním. Pokud by byla zpětná rekonstrukce cesty příliš složitá, mohlo by to eliminovat výhodu prohledávání menšího grafu. Navíc musí být

zachovány váhy na hranách, jinak by mohlo dojít k porušení podmínky nejkratší cesty.



Obrázek 4.1: Příklad původního zastávkového grafu

Následující úpravy navazují jedna na druhou tak, jak jsou popsány. Samostatné použití úprav je možné, ale se slabším účinkem. Výsledek použití v opačném pořadí je nejistý. Dosažené výsledky oproti předchozí úpravě jsou uvedeny v tabulce pod každou z úprav.

4.2.1 Slučování hran

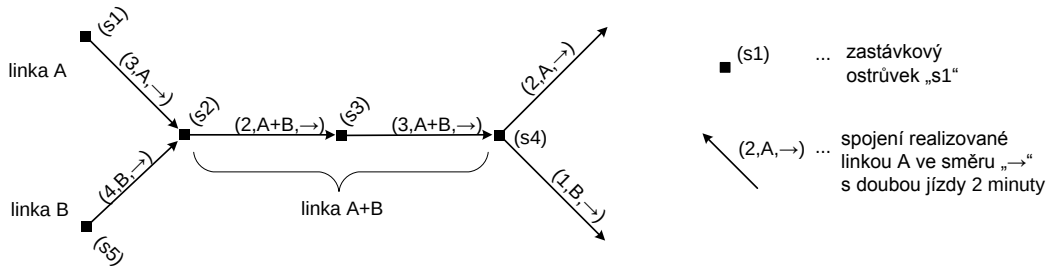
První z úprav vychází z následujícího pozorování. Chceme-li se dostat z jedné zastávky na následující v co nejkratším čase, pak pojedeme spojem, který nás na danou zastávku dopraví nejdříve. Pokud mezi těmito zastávkami jede několik různých spojů **stejně dlouhou dobu**, pak zvolíme ten, na který budeme čekat nejkratší čas. Abychom zjistili, jak dlouho budeme na jednotlivé spoje čekat, jsme nuceni prohlédnout jízdní řády všech těchto spojů.

Výše popsaná situace se v grafu projeví tak, že dva vrcholy jsou spojeny několika hranami se stejnou hodnotou. Tyto hrany se liší pouze v pseudolince. Vytvoříme novou pseudolinku, která bude kombinací všech pseudolinek na zmíněných hranách. Tyto hrany nahradíme jedinou novou hranou. Nová hrana bude spojovat stejné dva vrcholy jako původní hrany, bude mít stejné ohodnocení jako původní hrany, ale bude identifikována novou pseudolinkou:

Multi-linka je nová pseudolinka, která vznikne sloučením několika původních pseudolinek. Řídí se vlastním jízdním řádem, který je kombinací jízdních řádů původních pseudolinek. Spoj multi-linky odjíždí z dané zastávky v daný čas, pokud z ní v tento čas odjíždí některá z původních pseudolinek. Pro účely zpětné rekonstrukce cesty je potřeba si pamatovat z jakých pseudolinek byla multi-linka vytvořena.

Zpětná rekonstrukce cesty

Vrcholy nového grafu přímo odpovídají vrcholům grafu původního. Problém nastává v případě hran, kdy díky sloučeným hranám ztrácíme přehled o tom, jakou pseudolinku jsme použili k cestě mezi zastávkami. Abychom byli schopni určit, kterou



Obrázek 4.2: Slučování hran se stejnou dobou jízdy

hranu v původním grafu použít, je nutné si při hledání cesty pamatovat identifikaci původní pseudolinky, kterou právě jedeme. Odjíždí-li ze zastávkové ostrůvků více pseudolinek ve stejný čas, pak se tyto časy mohou slít do jednoho času pro multi-linku. V tom případě může být pseudolinek, kterými může cestující v danou chvíli jet, i více. Budeme si tedy pamatovat jejich seznam. Následujícím způsobem také změníme detekci přestupu při vyhledávání v novém grafu:

Nástup Při nástupu do multi-linky zjistíme, jakou původní pseudolinkou cestující pojede. Nejprve najdeme ve sloučeném jízdním řádu čas odjezdu spoje. Dostaneme tak čas odjezdu multi-linky. Nyní potřebujeme zjistit, která z původních pseudolinek tvořících multi-linku, jede v tomto čase. Je tedy nutné prohlédnout jízdní řády původních pseudolinek a zapamatovat si spoje, kterými může cestující v čas odjezdu multi-linky jet[1].

Přestup jen mezi multi-linkami Je-li hrana, kterou se chystáme přidat do plánované cesty, realizována jinou multi-linkou než poslední hrana na cestě, nemusí nutně docházet k přestupu mezi spoji. K přestupu nedochází, je-li průnik seznamu pseudolinek, kterými cestující může aktuálně jet, a seznamu pseudolinek, kterými je tvořena nově uvažovaná multi-linka, **neprázdný**. V takovém případě může cestující pokračovat libovolnou pseudolinkou z průniku, aniž by byl nucen přestoupit. Tento průnik bude nadále seznamem pseudolinek, kterými cestující může aktuálně jet[2]. V tomto případě se nezapočítává doba potřebná na přestup ani čekání na další spoj.

Skutečný přestup Ke skutečnému přestupu dochází, je-li průnik seznamu pseudolinek, kterými cestující může aktuálně jet, a seznamu pseudolinek, kterými je tvořena nově uvažovaná multi-linka, **prázdný**. V tomto případě se jedná o skutečný přestup a cestující musí z aktuálního spoje vystoupit. Vyhledávací mechanismus následně pokračuje obvyklým způsobem, tedy detekuje přestup a při volbě nového spojení uvažuje dobu na přestup a čekání na nový spoj.

[1] Tato situace je jednodušší než v původním grafu. Čas, kdy jede příští spoj, hledáme jen jednou - ve sloučeném jízdním řádu. Pak už jízdní řád neprocházíme, ale podíváme se jen na konkrétní jeden záznam podle již nalezeného času.

[2] Můžeme si dovolit ignorovat ty pseudolinky tvořící multi-linku nové hrany, které nejsou v průniku, protože by v jejich případě docházelo ke zdržení při přestupu. Z vlastností multi-linky máme zaručeno, že by časový odhad cesty do cílového vrcholu nové hrany byl větší než při použití libovolné pseudolinky z průniku.

	# vrcholů	# hran	spotřeba paměti	čas na prohledání celého grafu
původní síť	1096	8473	429 721B	4s
po 1. úpravě	1096	2927	1 182 226B	1.70s
rozdíl	pokles o 0%	pokles o 65%	nárůst o 175%	pokles o 58%

Tabulka 4.1: Porovnání testů vyhledávání nad původní sítí Pražské MHD a vyhledávání nad sítí se sloučenými hranami.

Při přestupu na novou multi-linku je potřeba opět zjistit, jakou původní pseudolinkou cestující pojede. Tedy provést stejnou akci jako při nástupu do multilinky.

Tento postup zajistí, že při nalezení cesty v novém grafu budu mít stejnou informaci jako při nalezení cesty v grafu původním. Identifikace vrcholů se shoduje s původním grafem a identifikace původních pseudolinek na hranách se udržuje po celou dobu hledání cesty. Hodnota hran se shoduje s hodnotou původních hran a dodatečná doba potřebná pro přestup je započítávána na stejných místech, jako v původním grafu.

4.2.2 Slučování cest

Druhá z úprav navazuje na pozorování z té první. Stačí uvažovat místo přímého spojení mezi dvěma zastávkami delší úsek několika po sobě následujících zastávek. Podmínkou je, aby na tomto úseku jezdily pouze určité spoje a žádné jiné spoje se do tohoto úseku nenapojovaly ani z něj neodbočovaly. Takovou posloupnost zastávek můžeme sloučit do jakéhosi "potrubí".

Potrubí je sloučení pseudolinek v rozsahu určitého úseku, ve kterém mají dané pseudolinky společnou trasu a shodnou dobu jízdy. Z hlediska zachování korektnosti plánování nejkratší cesty je důležité, aby pořadí odjezdů spojů ze začátku "potrubí" bylo stejné jako pořadí příjezdu spojů na konec "potrubí". "Potrubí" by tedy mělo zachovávat pořadí spojů. Aby mohla být tato vlastnosti zaručena, potřebujeme, aby spoje, ze kterých je "potrubí" postaveno, se uvnitř "potrubí" navzájem nepředjížděly. U metra, tramvají nebo trolejbusů by tato situace neměla nastat. U autobusů může ke změnám pořadí docházet.

Máme-li graf, na kterém byla již první úprava provedena, máme výchozí situaci ulehčenu. V následujícím textu budeme provedení první úpravy předpokládat, a pokud budeme mluvit o pseudolinkách, budeme tím chápat pseudolinky vytvořené předchozí úpravou, tedy multi-linky. Druhou úpravu lze provádět ve dvou fázích.

První fáze

V první fázi potřebujeme odlišit vrcholy uvnitř "potrubí" a vrcholy na jeho koncích. "Potrubí" budou hrany nového grafu. Vrcholy, které představují konce potrubí, budou tvořit vrcholy nového grafu – "uzly". Pro vnitřní vrcholy musí být splněny následující dvě podmínky:

I. Vnitřní vrchol nesmí mít více jak dva různé sousední vrcholy[3].

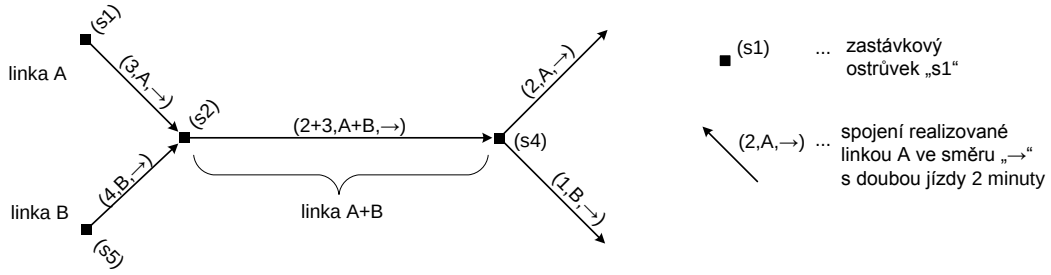
[3]Mezi sousedy se zde počítají i vrcholy dostupné v protisměru orientované hrany.

II. Hrany, které do vnitřního vrcholu vchází z něj musejí také vycházet. Sobě odpovídající vstupní a výstupní hrana musí nést stejnou identifikaci pseudolinky. Vnitřní vrchol nemůže být koncovým zastávkovým ostrůvkem na trase pseudolinky.

Všechny ostatní vrcholy budou tvořit konce "potrubí", vytvoříme z nich vrcholy nového grafu – "uzly".

Druhá fáze

Ve druhé fázi budeme postupně probírat "uzly" a doplňovat k nim hrany. Hrana, tedy "potrubí", mezi dvěma "uzly" odpovídá posloupnosti vrcholů v původním grafu. Tato posloupnost musí být tvořena hranami jedné stejné pseudolinky a nesmí být přerušena jiným "uzlem". Takovou posloupnost nahradíme v novém grafu jedinou hranou. Hodnota nové hrany je rovna součtu hodnot původních hran v posloupnosti. Pseudolinka nové hrany je pseudolinka původních hran.



Obrázek 4.3: Tvorba "potrubí" mezi "uzlovými" zastávkami

Napojení při hledání cesty

V novém grafu dojde k redukci vrcholů. To znamená, že pokud hledáme cestu z nějakého vrcholu, který není v novém grafu "uzlem", potřebujeme najít nejprve cestu do nejbližších "uzlů" a teprve pak můžeme hledat cestu v novém grafu. Tuto cestu najdeme v původním grafu poměrně snadno, protože z vlastností vrcholů uvnitř "potrubí" plyne, že cesty vedou nejvýše do dvou směrů. Narazíme tak po velmi krátkém hledání na nejvýše dva okrajové "uzly". Oba okrajové uzly budeme brát jako výchozí body pro hledání v novém grafu. Počáteční hodnotu odhadu cesty u výchozích bodů nastavíme na délku cesty z původního vrcholu do nejbližšího "uzlu". Analogicky postupujeme, není-li cílový vrchol "uzlem".

Zpětná rekonstrukce cesty

Jelikož tato druhá úprava grafu vychází z první, potřebujeme použít při vyhledávání stejný postup detekce přestupů jako v první úpravě. Cesta nalezená v tomto novém grafu je tvořená "uzly" a "potrubím".

Došlo k redukci vrcholů, proto potřebujeme doplnit do cesty ty vrcholy, které jsou uvnitř "potrubí". Došlo také k redukci hran, ze kterých je vytvořeno "potrubí", takže potřebujeme doplnit i tyto hrany. Abychom to mohli provést, je potřeba si pro každé "potrubí" pamatovat posloupnost vrcholů a hran, ze kterých bylo vytvořeno. Pak lze

	# vrcholů	# hran	spotřeba paměti	čas na prohledání celého grafu
původní síť	1096	2927	1 182 226B	1.70 s
po 2. úpravě	549	1987	1 225 810B	1.03 s
rozdíl	pokles o 50%	pokles o 32%	nárůst o 4%	pokles o 39%

Tabulka 4.2: Porovnání testů vyhledávání nad sítí Pražské MHD po první úpravě a vyhledávání nad síti po provedení obou úprav

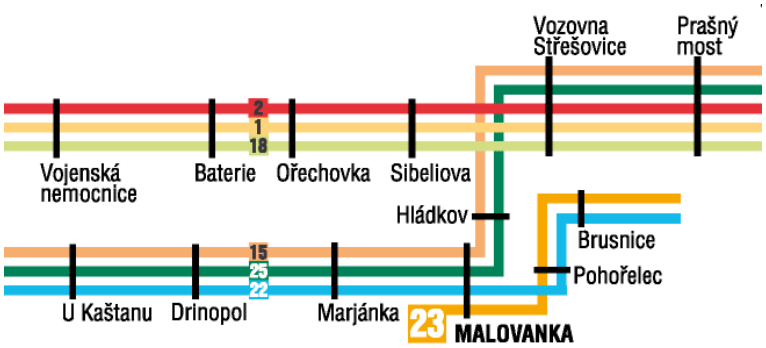
snadno nahradit jednotlivá "potrubí" v nalezené cestě odpovídajícími posloupnostmi vrcholů a hran.

Nakonec zbývá do výsledné cesty doplnit už jen počáteční úseky, kterými jsme se dostali z počátečních vrcholů do počátečních "uzlů".

4.3 Ukázka redukce nad pražskou MHD

V době psaní této práce je k dispozici pouze datová sada Pražské hromadné dopravy, proto bude ukázková úprava provedena právě na části těchto dat.

V tabulkách je uveden orientační čas potřebný pro nalezení nejkratší cesty. Aby tento čas nebyl ovlivněn délkou nalezené cesty, byla z vyhledávacího algoritmu vyjmuta ukončující podmínka. Takto modifikovaný algoritmus pokračuje v hledání, dokud neprojde celý graf.



Obrázek 4.4: Vzorek pražské MHD

Obrázek 4.4 ukazuje výřez z linkového schématu tramvajové dopravy získaný z [1]. Každá z vyobrazených linek jezdí oběma směry.

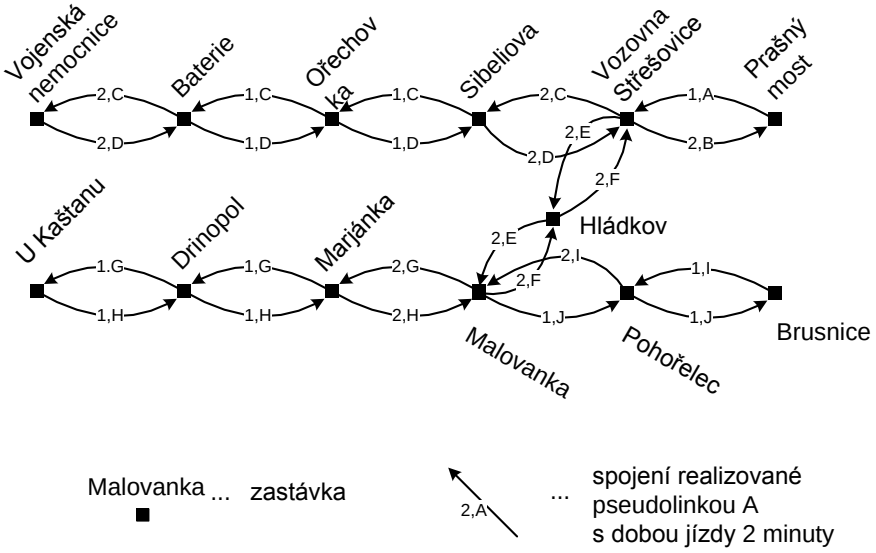
4.3.1 Sloučení hran

Při provádění první úpravy budou nově zavedeny tyto pseudolinky:

A slučuje linky číslo 2, 1, 18, 15, 25 od zastávky "Prašný most" po zastávku "Vozovna Střešovice"

B slučuje stejné linky jako A, ve stejném úseku, ale v opačném směru

C slučuje linky číslo 2, 1, 18 od zastávky "Vozovna Střešovice" po zastávku "Vojenská nemocnice"



Obrázek 4.5: Vzorek pražské MHD po první úpravě

D slučuje stejné linky jako C, ve stejném úseku, ale v opačném směru

E slučuje linky číslo 15, 25 od zastávky "Vozovna Střešovice" po zastávku "Malovanka"

F slučuje stejné linky jako E, ve stejném úseku, ale v opačném směru

G slučuje linky číslo 15, 25, 22 od zastávky "Malovanka" po zastávku "U Kaštanu"

H slučuje stejné linky jako G, ve stejném úseku, ale v opačném směru

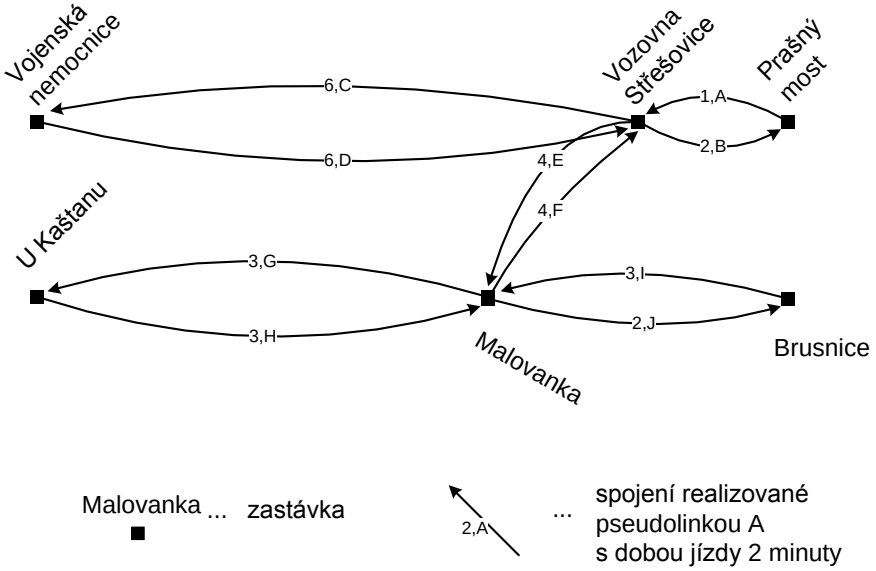
I slučuje linky číslo 15, 25, 22 od zastávky "Brusnice" po zastávku "Malovanka"

J slučuje stejné linky jako I, ve stejném úseku, ale v opačném směru

V původním grafu je 68 hran, pokud počítáme pro každou linku dva odlišné směry a nepočítáme výjimky. Při první úpravě došlo k agregaci hran, které jsou nyní realizovány sloučenými pseudolinkami. Nových hran je pouze 24, což je téměř třikrát méně oproti původnímu grafu. Počet vrcholů se nezměnil, graf má 13 vrcholů.

4.3.2 Sloučení cest

Provádění druhé úpravy již nevyžaduje tvorbu nových pseudolinek. Došlo k agregaci cest, které jsou realizovány shodnou pseudolinkou a vedou mezi uzly. Jako uzly byly rozpoznány zastávky "Prašný most", "Vozovna Střešovice", "Vojenská nemocnice", "Brusnice", "Malovanka", "U Kaštanu". Počet hran klesl na 10, což je více jak dvojnásobný pokles oproti první úpravě. Počet vrcholů se snížil na 6, což je více jak dvojnásobný pokles.



Obrázek 4.6: Vzorek pražské MHD po provedení obou úprav

	# vrcholů	# hran	spotřeba paměti	čas na prohledání celého grafu
původní síť	1096	8473	429 721B	4s
po 1. úpravě	1096	2927	1 182 226B	1.70s
po 2. úpravě	549	1987	1 225 810B	1.03s
celkový rozdíl	pokles o 50%	pokles o 77%	nárůst o 185%	pokles o 74%

Tabulka 4.3: Porovnání testů vyhledávání nad původní sítí Pražské MHD, nad sítí po provedení první úpravy a nad sítí po provedení obou úprav

4.3.3 Výsledek ukázky

V ukázkovém příkladu se díky úpravám počet vrcholů snížil více jak dvakrát a počet hran téměř sedmkrát. V celkovém měřítku brání takto výrazné redukci hustota křížení jednotlivých linek a nepravidelné uspořádání zastávkových ostrůvků.

4.4 Výsledky

Tabulka 4.3 shrnuje výsledky dosažené při testování popsané redukce sítě Pražské hromadné dopravy na reálném zařízení. K testování redukce bylo přistoupeno v rané fázi vývoje knihovny pro projekt JRGPS, kdy bylo potřeba hledat metody pro snížení doby výpočtu. Více o projektu v kapitole 7.4.

Při testování byla z vyhledávacího algoritmu odstraněna ukončovací podmínka. Vyhledávací mechanismus tak byl donucen prohledávat celý graf. Díky tomu byla eliminována závislost na konkrétní podobě zadání, která by mohla dosažené výsledky zkreslovat.

Podařilo se nám snížit dobu potřebnou pro výpočet, avšak za cenu vyšších paměťových nároků. Stále však únosných pro přenosná zařízení. V pozdější fázi

vývoje knihovny bylo dosaženo dostatečného zrychlení vhodnou implementací a ořezáváním výpočtu. Z tohoto důvodu nebyla redukce nasazena ve výsledné implementaci knihovny.

Nicméně u opravdu rozsáhlých dopravních sítí by redukce vyhledávacího grafu byla zcela jistě na místě. Její velkou výhodou je ortogonalita k většině optimalizací vyhledávacího mechanismu.

Kapitola 5

Komplexní systém navigace

Komplexní navigační systémy kombinují různé dopravní prostředky a snaží se tak uživateli nabídnout co možná nejúplnější plán cesty.

Aplikace, které nabízejí plánování cest pro určitý druh dopravního prostředku, se často opírají o nepřesné informace, jako jsou například odhady přestupní doby v hromadné dopravě. Je-li odhad doby příliš nízký, může to vést k nepoužitelnosti navrženého plánu. Je-li odhad doby naopak příliš nadsazený, může to vést ke zbytečným prodlevám a nesprávnému plánování nejkratšího spojení. Podaří-li se nám na základě podrobnějších informací zpřesnit použité odhady, pak bude výsledný plán cesty přesnější a spolehlivější.

Komplexní navigační systém může být navíc schopen vyhledávat efektivnější cesty než jednoúčelová aplikace. Je to dáno tím, že vyhledávací algoritmus má více možností, jak výslednou cestu sestavit. Komplexní navigační systémy používají mnohem rozsáhlejší vyhledávací sítě, což výrazně zvyšuje výpočetní náročnost těchto aplikací. V navigačních systémech, které kombinují více dopravních sítí, je důležitým faktorem způsob kombinace vyhledávacích sítí a celková výpočetní náročnost vyhledávání cest v kombinovaném grafu.

Zkušenosti z vývoje komplexní navigace byly publikovány v článku [10].

5.1 Analýza

Základní řešení je postaveno tak, že umožňuje vyhledávat spojení pouze prostředky hromadné dopravy osob. K tomu používá síť hromadné přepravy osob. Abychom mohli rozšířit plánování cest o pěší přesuny, je potřeba vhodně zkombinovat síť hromadné dopravy a síť pěších cest. Navíc je potřeba navrhnout takový vyhledávací mechanismus, který bude schopen plánovat nejkratší cesty v kombinaci obou sítí.

5.1.1 Reprezentace pěší sítě

Na síť pěších cest lze pohlížet jako na neorientovaný graf. Vrcholy mohou představovat místa, kde se jednotlivé cesty kříží a hrany jsou úseky cest bez křížení. Hrany je možné nezáporně ohodnotit dobou potřebnou k překonání daného úseku.

Vzhledem k ohodnocení hran dobou chůze je lepší použít k reprezentaci sítě pěších cest orientovaný graf. Doba chůze může totiž záviset na směru průchodu hranou. Typickým příkladem může být chůze do kopce, do schodů apod.

Pěší graf

Pěší graf reprezentuje síť ulic, chodníků, pěšin a dalších pěších cest. Jedná se o orientovaný graf.

Vrchol grafu nechť reprezentuje průsečík, začátek nebo konec pěší cesty.

Hrana nechť reprezentuje pěší cestu, která není přerušena křížením s jinou pěší cestou.

5.1.2 Dálniční hierarchie

Dálniční hierarchie [2] je heuristika inspirovaná silniční sítí. Spočívá v tom, že hustou síť rozdělíme na oblasti, ze kterých vytvoříme hierarchicky nadřazenou vrstvu sítě. Vyhledávání se pak realizuje nad každou sítí zvlášť, přičemž v hierarchicky nižších sítích se vždy hledá vstup do hierarchicky nadřazené sítě.

Myšlenka použití této heuristiky v naší úloze vychází z podobnosti vztahu mezi silnicemi různé třídy v silniční síti a různými dopravními prostředky v síti MHD. Síť pěších cest je výrazně hustší, ale také pomalejší v porovnání se sítí dopravních prostředků. Podobně jako síť místních komunikací v porovnání s dálnicemi. Najet nebo sjet z dálnice není možné kdekoli, ale jen na vybraných místech. Stejně jako do prostředků hromadné dopravy lze nastoupit nebo z nich vystoupit jen na zastávkách.

Vezměme si úsek mezi dvěma obecnými body. Cestu, která je propojuje, můžeme rozdělit na tři úseky: pěší cestu na zastávku – spojení hromadnou dopravou – pěší cestu ze zastávky. První a poslední úsek bude realizován chůzí. Prostřední úsek cesty je realizován městskou hromadnou dopravou, do které započítáváme i pěší přestupy mezi zastávkami.

Základním předpokladem, který ospravedlňuje uvedené rozdělení, je, že prostředky realizující střední úsek jsou mnohem rychlejší než prostředky realizující oba krajní úseky. Což je v našem případě typicky splněno. V případě nesplnění této podmínky by bylo potřeba hledat jiné řešení.

5.1.3 Kombinace dopravních sítí

Způsobů, jak provádět plánování cest nad různými dopravními sítěmi zároveň, je více. Jsou-li si dopravní sítě navzájem podobné (například dva různí dopravci v jednom regionu), pak lze tyto sítě v určitých bodech propojit a jednoduše je sjednotit.

Častěji je však potřeba kombinovat sítě různé hierarchie (například městkou hromadnou dopravu a vnitrostátní dopravu). Pokud bychom sítě jednoduše sjednotili, dostali bychom rozsáhlou a nadbytečně podrobnou síť. Časová složitost vyhledávání v takové síti by byla zbytečně vysoká.

Je-li možné vhodně propojit obě sítě v relativně malém počtu vrcholů, pak lze s úspěchem použít princip dálniční hierarchie. Ta umožňuje provádět vyhledávání cest v oddělených sítích a nalezené úseky vhodně spojit v přechodových vrcholech. Předpokladem pro úspěšné použití dálniční hierarchie je podstatně vyšší rychlost pohybu v nadřazené síti. Porovnáme-li rychlost chůze a rychlost přepravy prostředky MHD, pak je tento předpoklad pro naši úlohu splněn.

5.1.4 Volba vyhledávacího mechanismu

Díky dálniční hierarchii je z hlediska výpočtu odděleno vyhledávání pěších cest a spojení v hromadné dopravě. Celkové řešení bude vycházet ze základního řešení, které je popsané v kapitole 3. Je potřeba doimplementovat vyhledávací mechanismus pro plánování pěších úseků cesty a vhodně jej propojit s vyhledávacím mechanismem v základním řešení.

Síť pěších cest tvoří orientovaný graf. Je vhodné zvolit dobu chůze jako ohodnocení hran pěšího grafu. Toto ohodnocení je nezáporné a lze ho vhodně kombinovat s ohodnocením zastávkového grafu. Na rozdíl od hromadné dopravy bývá ohodnocení pěších hran statické. Lze však najít řadu výjimek, kdy ohodnocení statické není a průchodnost pěší hrany závisí na aktuálním čase. Navíc lze ohodnocení měnit parametry uživatele, například rychlostí chůze.

Při výběru vyhledávacího algoritmu můžeme vycházet ze stejné třídy algoritmů jako v základním řešení, kapitola 3.1.2. Při volbě algoritmu je potřeba si uvědomit způsob jeho použití ve vyhledávacím mechanismu. Bude potřeba hledat nejkratší cesty nejen mezi dvěma vrcholy, ale hlavně mezi výchozím vrcholem a množinou vrcholů v určité vzdálenosti. Což je přesně případ, kdy máme zadané výchozí místo v síti pěších cest a cílem je jakýkoliv zastávkový ostrůvek hromadné dopravy v určitém okolí.

Floyd-Warshallův algoritmus by mohl být vhodný, protože vyhledává cesty mezi všemi vrcholy najednou, což je dokonce více než potřebujeme. Při jeho implementaci by bylo možné se omezit pouze na podgraf určité velikosti a hledat tak cesty z výchozího místa na zastávkové ostrůvky v okolí. Stále však přetrvává problém s potřebou dynamického ohodnocení hran.

Dijkstrův algoritmus vyhledává cesty z výchozího vrcholu do všech vrcholů v grafu, pokud vyhledávání neukončíme při prohlášení cílového vrcholu za trvalý. Ukončovací podmínku lze změnit tak, že algoritmus skončí při dosažení určité vzdálenosti od výchozího vrcholu. To přesně odpovídá nalezení nejkratších cest z výchozího místa do zastávkových ostrůvků v daném okolí.

Zvolili jsme Dijkstrův algoritmus, protože lépe splňuje specifické požadavky na řešení dané úlohy. Navíc lze při jeho implementaci v pěší síti vycházet z implementace vyhledávacího mechanismu v základní úloze.

5.2 Reprezentace dat

5.2.1 Síť pěších cest

Mapové podklady, které byly k dispozici, jsou tvořeny sadou polyline, reprezentující různé druhy cest. Každá polyline obsahuje informaci o druhu cesty a podrobný průběh cesty. Pro účely vyhledávání není průběh cesty důležitý. Na základě polyline reprezentující pěší cesty byl vytvořen vyhledávací graf pěších cest.

Vrcholy tohoto grafu jsou křížení nebo zakončení polyline. Specifickými vrcholy pěšího grafu jsou zastávkové ostrůvky, které nesou identifikaci odpovídajícího zastávkového ostrůvku v síti hromadné dopravy. Díky tomu, lze po dosažení zastávkového ostrůvku plynule přejít do vyhledávacího grafu hromadné dopravy.

Každá polyline je reprezentována dvěma orientovanými hranami v navzájem opačných směrech. Pro hledání nejkratší cesty v grafu je potřeba přiřadit hranám

ohodnocení, podle kterého se bude délka cesty počítat. Jelikož ohodnocení hran v síti hromadné dopravy představuje dobu jízdy, je výhodné ohodnotit hrany dobou chůze. Z mapových podkladů známe průběh pěší cesty a jsme schopni na základě vzdálenosti a rychlosti chůze dopočítat dobu potřebnou k překonání hrany. Tato doba by navíc měla být parametrizovatelná dle dispozic uživatele.

Problém nastává v případě úseků s převýšením[1] nebo nějakým druhem bariéry a na přechodech[2]. V dostupných datech nebyly tyto údaje k dispozici.

5.2.2 Redukce pěšího grafu

Pro účely vyhledávání nemá smysl uchovávat v grafu vrcholy mohutnosti 2. V tomto vrcholu nedochází k větvení, vyhledávací mechanismus se pouze posune o jednu hranu dále. Takový vrchol lze vypustit a prodloužit hranu, která jím byla v podstatě rozdělena. Tuto jednoduchou redukci lze provést už při převodu vektorových dat na vnitřní reprezentaci grafu.

5.2.3 Propojení pěší sítě a sítě hromadné dopravy

Aby bylo možné volně přecházet ve vyhledávání ze sítě hromadné dopravy do sítě pěších cest a zpět, je nutné provést vzájemné mapování přechodových uzlů. Těmito přechodovými uzly jsou zastávkové ostrůvky, na kterých cestující vystupují a nastupují do spojů hromadné dopravy.

Zastávkové ostrůvky v mapě

V mapových podkladech, které jsme měli k dispozici, nebyly zaneseny zastávkové ostrůvky. Pozice zastávkových ostrůvků jsme měli z jiného zdroje. Bylo proto nutné provést jednak korekci souřadnic a hlavně bylo nutné napojit zastávkové ostrůvky do stávající sítě pěších cest. Napojení jsme prováděli tak, že jsme přidali polyline vedoucí ze zastávkového ostrůvku na nejbližší polyline v síti pěších cest. Tento postup jsme prováděli automaticky a v některých složitějších případech bylo třeba provést dodatečnou ruční korekci.

Předvýpočet přestupů

Abychom nemuseli přímo kombinovat vyhledávání v obou sítích, provedli jsme předvýpočet pěších přestupů a přidali do sítě hromadné dopravy zvláštní hrany reprezentující tyto pěší přestupy. Díky tomu nemusíme při plánování cesty neustále opouštět vyhledávací graf pro hromadnou dopravu a výrazně tak klesne celkové větvení výpočtu. Předvýpočet provádíme tak, že probíráme jednotlivé zastávkové ostrůvky a hledáme všechny pěší cesty na zastávkové ostrůvky v určitém okolí. Což přesně odpovídá předchozímu uvedenému případu.

Protože se jedná o předvýpočet, je nutné zvolit maximální délku pěšího přestupu mezi spoji už při sestavování datové sady. Pokud bychom prováděli předvýpočet s předpokladem libovolně dlouhého přestupu, vedlo by to k neúnosnému nárůstu větvení vyhledávacího grafu hromadné dopravy[3]. Pokud bychom naopak zvolili příliš

[1] hodnota hran se může lišit v závislosti na směru

[2] hodnota se může měnit v závislosti na aktuálním čase

[3] Jak z hlediska výpočetních tak paměťových nároků

krátkou maximální dobu pro přestup, omezovalo by to uživatele a mohlo by to mít nepříznivý vliv na podobu výsledné cesty. Některé vhodné přestupy by se díky předvýpočtu nedostaly do vyhledávacího grafu. Krajní dobu pro přestup určujeme při sestavování dat, většinou na základě zkušeností a testů.

Předvypočítané hrany jsou doplněny do datové sady hromadné dopravy a následně převedeny do vnitřní reprezentace zastávkového grafu. V zásadě odlišujeme hrany dvou typů.

Pěší hrany s pevnou hodnotou

Prvním typem jsou hrany, na jejichž hodnotu nemá vliv rychlost chůze. Jedná se zpravidla o výstupy z metra, kde výraznou část trasy tvoří například jízda po eskalátorech. Pokud uživatel změní rychlost chůze, hodnota těchto hran se nezmění. Tyto hrany byly přidány do grafu ručně a jejich hodnota je určena odhadem nebo změřením doby průchodu trasy přímo v terénu.

Obecně by mělo být možné měnit hodnoty všech hran na základě dispozic uživatele. navíc určení hodnoty některých pěších přesunů nemusí být jednoduché – například výtahy, přechody. Situace je komplikovanější a jedno z řešení by mohlo souviset s víceparametrickým vyhledávání. Více v kapitole 6.3.

Pěší hrany s proměnnou hodnotou

Druhým typem jsou hrany, jejichž hodnotu lze měnit v závislosti na rychlosti chůze. Jedná se zpravidla o hrany, reprezentující povrchový přestup mezi zastávkovými ostrůvky. Tyto hrany byly vytvořeny automaticky a byla jim přiřazena hodnota v minutách odpovídající době chůze při rychlosti 5 km/h. Výpočet doby je prováděn s jemností na vteřiny, ve výsledku je převeden na minuty a zaokrouhlen nahoru.

Hrany byly vytvořeny tak, že pro každý zastávkový ostrůvek byly nalezeny cesty k okolním zastávkovým ostrůvkům. Velikost okolí je omezena maximální dobou chůze. Základní rychlost chůze jsme zvolili 5 km/h a hodnoty hran jsme předvypočítali s jemností na sekundy. Uživatel může rychlost chůze změnit. Doba chůze se pak přenásobí koeficientem odpovídající uživatelské změně. Tento předvýpočet zajišťuje doprovodná aplikace “viewer” popsaná v kapitole 7.5.

5.3 Řešení

Snažili jsme se optimálně zkombinovat použití prostředků hromadné dopravy a chůze. Pěší síť je velmi rozsáhlá, na druhou stranu pěší cesta se typicky hledá na malou vzdálenost. Navíc pěší síť, kterou máme k dispozici, typicky nemá dynamickou hodnotu hran[4]jako síť hromadné dopravy. Lze tedy velkou část výpočetní náročnosti snížit vhodným předzpracováním dat.

5.3.1 Plánování cest

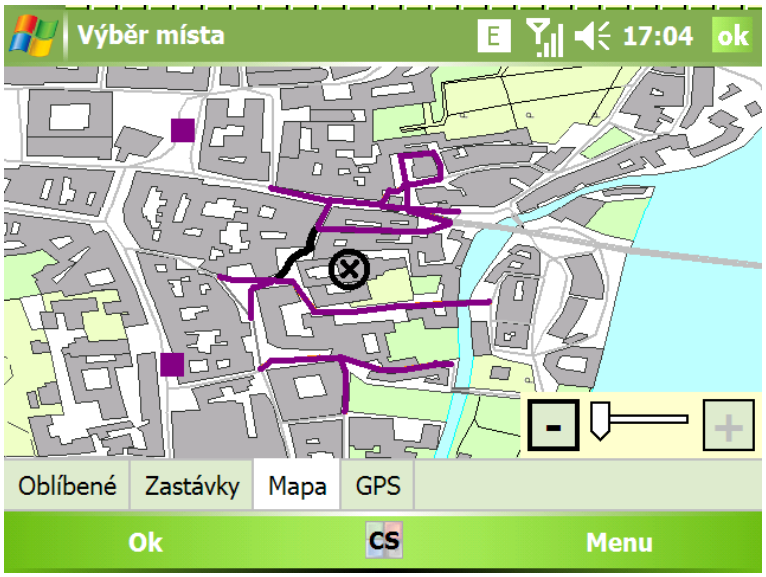
Vyhledávací mechanismus pro nalezení nejkratšího spojení v síti hromadné dopravy je již popsán v kapitole 3. Propojením chůze a prostředků hromadné dopravy nám

[4]Obecně by mělo být ohodnocení pěších hran určováno dynamicky, avšak vzhledem k aktuálně dostupným datům to není nezbytné.

dává možnost plánovat s poměrně vysokou přesností pohyb uživatele mezi zcela libovolnými místy. Uvažujme úlohu v obecné podobě, tedy máme zadanou výchozí a cílovou pozici v mapě a chceme najít cestu, která je spojuje. Při plánování budeme postupovat tak, že nejprve naplánujeme pěší cesty z výchozí pozice do zastávek hromadné dopravy v určitém okolí. Obdobně naplánujeme pěší cesty i pro cílovou pozici[5]. Tím se úloha zjednoduší na hledání cesty mezi zastávkami hromadné dopravy. Při přestupech mezi spoji však stále potřebujeme používat pěší graf k určení doby přestupu.

Zadávání pozice v mapě

Při zadávání výchozí nebo cílové pozice v mapě je potřeba přesně určit pěší cestu, ke které je uživatel nejblíže. Příklad ukazuje obrázek 5.1. Průběh cesty z obecně zadaného bodu k nejbližší pěší cestě neznáme, proto tuto cestu aproximujeme vzdušnou čarou. Zvolení nejbližší pěší cesty přenecháváme na uživateli. Tato volba může být komplikovaná a pokud selže prvotní automatický odhad, uživatel má okamžitou možnost korekce na základě vlastních zkušeností v konkrétní lokalitě. V nejhorším případě se uživatel jednoduše spolehne na automatickou volbu.



Obrázek 5.1: Výběr výchozího bodu
Výchozí bod je označen křížkem. Pěší cesty v okolí jsou zvýrazněny fialovou barvou. Vybraný úsek cesty, který bodu nejblíže a je označen černě.

Vyhledávání pěší cesty na zastávku

Při vyhledávání pěší cesty z výchozí pozice nás zajímají všechny zastávkové ostrůvky v určitém okolí. Z těchto zastávkových ostrůvků může cestující pokračovat hromadnou dopravou. Velikost okolí si může uživatel nastavit podle toho, jak daleko je ochoten jít pěšky. Díky tomuto omezení není nutné prohledávat celý pěší graf, ale jen jeho část.

[5]Pěší cesty z cílové pozice je potřeba hledat v opačném směru.

Díky prohledávání do šířky jsme schopni nalézt všechny cesty z výchozí pozice do všech zastávkových ostrůvků v daném okolí najednou. V dalším vyhledávání se pokračuje už v grafu hromadné dopravy s tím, že výchozí pozice jsou všechny dosažené ostrůvky a jejich počáteční odhad délky cesty je nastaven na délku nalezené pěší cesty z daného výchozího bodu.

5.3.2 Vyhledávání spojení v hromadné dopravě

Množina zastávkových ostrůvků dosažitelných z výchozího vrcholu pěšího grafu se stává množinou výchozích vrcholů zastávkového grafu pro hledání spojení v hromadné dopravě. Množina zastávkových ostrůvků dosažitelných z cílového vrcholu pěšího grafu se stává množinou cílových vrcholů zastávkového grafu.

Množiny výchozích a cílových zastávkových ostrůvků jsou do této fáze vyhledávání předány spolu s informací o době chůze z počátečního respektive cílového vrcholu pěšího grafu. Vyhledávací mechanismus pro hromadnou dopravu je postaven tak, aby tento vstup dokázal zpracovat. Více v kapitole 3.3.2. Informace o odhadu počáteční pěší cesty se tak předává mezi jednotlivými fázemi vyhledávání a plánovací mechanismus z této informace vychází při určování nejkratšího spojení.

Při samotném vyhledávání cesty je potřeba zohledňovat parametry nastavené uživatelem. Například maximální délku pěšího úseku[6], rychlost chůze a další. Předvypočítané pěší přestupy mezi spoji MHD jsou sice součástí zastávkového grafu, ale stále reprezentují pěší pohyb. Na ten se vztahují parametry nastavené uživatelem pro pěší graf.

5.3.3 Sloučení výsledků

Pro nalezení cesty je potřeba realizovat tři různá vyhledávání. Dvakrát se hledají okolní zastávky v pěším grafu a jednou se hledá spojení v grafu hromadné dopravy. Tím, že jsme do grafu hromadné dopravy přidali předvypočítané hrany, jsme umožnili používat chůzi tam, kde spoje nejezdí dostatečně často nebo dojde v výluce a spoj nejede vůbec.

Nalezená cesta tedy není tvořena třemi ostře oddělenými částmi, ale jedinou souvislou cestou, kde se prolíná chůze s jízdou prostředky hromadné dopravy. Pěší přechody v nalezené cestě nesou pouze předvypočítaný číselný údaj o délce přechodu. Pro zobrazení pěšího úseku cesty v mapě je potřeba danou cestu znovu nalézt, protože uchovávání průběhu každé takové cesty by bylo paměťově příliš náročné.

5.3.4 Uživatelské předvolby

Uživatelé mohou mít velice různé dispozice určující jejich pohyb. To se může výrazně projevit na podobě naplánované cesty. Snažili jsme se zohlednit alespoň základní parametry, které si uživatel může nastavit.

Rychlost chůze

Veškeré pěší hrany ve vyhledávacích grafech jsou ohodnoceny časem nikoliv délkou. Až na speciální hrany, je touto hodnotou doba chůze. Ta je určena na základě délky

[6] Uživatel je ochoten jít pěšky v kuse pouze úsek určité délky.

pěšího úseku, který hrana reprezentuje, a předvolené rychlosti chůze 5km/h. Pokud uživatel změní přednastavenou rychlost chůze, provádíme příslušnou korekci přímo při plánování cest.

Speciální hrany jsou odlišeny dodatečným údajem o charakteru úseku, který daná hrana reprezentuje. Tyto hrany nepodléhají korekci rychlosti chůze.

Maximální délka souvislého pěšího úseku

Tento parametr je přidán zejména kvůli předvypočítaným pěším přestupům, které teoreticky limitují možnosti pěšího přestupu. Praktický význam pro uživatele má v případě, že si nastaví limit na hodnotu nižší nebo rovnou limitu použitému při předvýpočtu. Zároveň tento parametr omezuje velikost okolí, které se prochází při vyhledávání cest k nejbližším zastávkám.

Může nastat případ, kdy nastavený limit zabrání naplánování výhodné cesty. Například v situaci, kdy intervaly mezi odjezdy spojů jsou relativně dlouhé. Pro cestujícího může být výhodnější překonat pěšky relativně velkou vzdálenost, než čekat na příjezd spoje. V tomto případě by přednastavený limit souvislé pěší chůze znamenal omezení možností plánování. Problém lze řešit tím, že uživatel zvýší limit souvislé pěší chůze.

5.3.5 Uživatelská místa

Při každodenním používání naší navigace se v zadání mohou často opakovat místa, která uživatel běžně používá jako výchozí nebo cílové body. Jsou to místa jako domov, práce, škola apod. V těchto případech většinou uživatel již velmi dobře ví, kudy vede pěší cesta, a zná čas, který mu cesta zabere na zastávky v okolí. Proto jsme do aplikace zavedli možnost si tato místa předdefinovat včetně časů na okolní zastávky hromadné dopravy. Takto předdefinovaná místa zkracují čas potřebný pro zadání cesty a zvyšují uživatelské pohodlí celé aplikace.

Realizace uživatelských míst není součástí knihovny, ale uživatelského rozhraní. Je však nutné, aby knihovna takové zadání podporovala, což je popsáno v kapitole 3.3.2.

5.3.6 Spolehlivost cesty

Během cesty může dojít k nenadálému omezení dopravní sítě například z důvodu technické závady na trase nebo spoji. V tomto případě dosavadní plán cesty neplatí a je potřeba jej přepočítat na základě nových údajů. Nový plán cesty se může od původního výrazně lišit, nastane podobná situace jako je vidět na obrázku 2.1.

Vyloučení linky

V naší aplikaci dovolujeme uživateli reagovat na podobnou situaci vyloučením konkrétní linky, která je výpadkem zasažena. Vyloučit lze přitom libovolný počet linek. Následně bude cesta naplánována s použitím jiných linek.

Vyloučení úseku

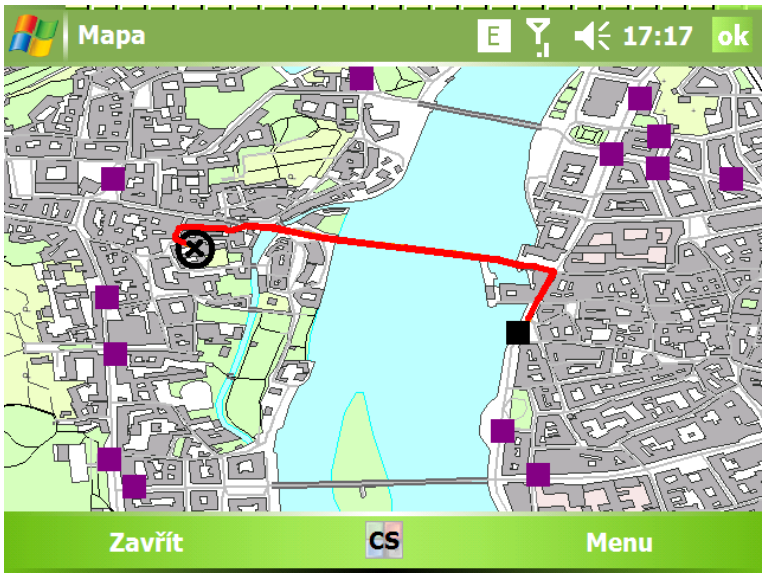
Výpadek určité části sítě může postihnout jak hromadnou dopravu, tak pěší síť. V obou případech by bylo potřeba umožnit uživateli označit zasaženou část sítě a

přeplánovat cestu jinudy. Tento úkon může vyžadovat určité nestandardní znalosti na straně uživatele a v naší aplikaci jej neřešíme.

5.3.7 Příklad výhodného pěšího přestupu

Výhody plánování kombinace chůze a hromadné dopravy se projeví v situacích, kdy je vzhledem k celkové délce cesty výhodnější jet spojem z relativně odlehlé zastávky. Obrázek 5.2 ukazuje počáteční pěší úsek plánu cesty, který vede na relativně vzdálenou zastávku.

V případě omezení doby přestupu na příliš nízkou hodnotu by mohlo dojít k ignorování cesty vedoucí ze vzdálené zastávky. Jako výsledná cesta by pak mohla být zvolena jiná, delší, cesta.



Obrázek 5.2: Výhodný pěší přístup

Z hlediska výběru nejkratší cesty je opodstatněné vybírat spojení ze všech zastávek v určitém okolí. Někdy může vést dlouhý počáteční pěší úsek na zastávku, odkud jede výhodné spojení. Tato situace může být častá v případech, kdy je část dopravní sítě oddělena bariérou jakou je například řeka.

Kapitola 6

Další vývoj

6.1 Využití redukce sítě

Redukt dopravní sítě, který jsme navrhli v kapitole 4, vznikl relativně přirozenou cestou, tedy v souladu s vypozorovaným chováním dopravní sítě. Pokud bychom měli od provozovatele dopravní sítě příslušná data, bylo by možné sledovat chování jednotlivých "potrubí" například z hlediska spolehlivosti. Bylo by pak možné vysledovat úseky dopravní sítě, kde například pravidelně dochází ke zpoždění, a tyto poznatky promítnout do plánovacího mechanismu. Tyto informace by samozřejmě byly zajímavé nejen pro cestující, ale pro samotného provozovatele dopravní sítě.

6.2 Dálniční hierarchie

Jedním ze zajímavých postupů, který by mohl navazovat na uvedené úpravy je "dálniční hierarchie". Tuto heuristiku jsme již použili pro kombinaci sítě hromadné dopravy a pěší sítě, jak je popsáno v kapitole 5.1.2. Tento přístup však lze použít i samostatně v síti hromadné dopravy.

Aby aplikace tohoto postupu přinesla znatelné zlepšení, musíme být schopni v zastávkovém grafu nalézt specifické oblasti. Tyto oblasti by měly mít relativně málo linek, které je spojují s ostatními oblastmi v relativně malém počtu vrcholů. Oproti tomu uvnitř oblasti by měla být doprava relativně hustá.

Tato metoda by mohla být použita dvěma způsoby. Mohli bychom dále redukovat síť hromadné dopravy z druhé úpravy a dosáhnout tak dalšího zrychlení. Problémem zůstává v nalezení takových oblastí, které by vyhovovaly "dálniční hierarchii". V případě městské hromadné dopravy považujeme nalezení vyhovujících oblastí tak, aby přinášely znatelné zlepšení za velmi obtížné.

Tento přístup lze mnohem lépe aplikovat na propojení například městské hromadné dopravy a dopravy vnitrostátní, případně mezistátní.

6.3 Víceparametrické vyhledávání cesty

Doposud jsme se zabývali pouze nalezením časově nejkratší cesty s několika omezujícími podmínkami. Požadavky uživatele na výslednou podobu cesty mohou být značně různorodé a nemusí se vždy jednat o striktní podmínky. Cestující může zajímat například čas strávený v dopravních prostředcích nebo při čekání, cena spojení,

počet a náročnost přestupů, spolehlivost nalezeného spojení, bezpečí, pohodlí během cesty apod.

Abychom byli schopni zohlednit preference cestujícího vůči parametrům plánované cesty, bude potřeba provádět multikriteriální vyhledávání nad kombinací obou sítí, což může být výpočetně velmi náročné. Jako slibný přístup k této problematice se v kontextu železniční sítě ukazuje řešení uvedené v [11].

6.3.1 Předčasný odjezd

Situace může nastat v současné implementaci vyhledávací knihovny, následují-li na cestě za sebou dvě linky, z nichž první jezdí velmi často a druhá relativně málo často. Vyhledávací algoritmus použije první linku a bude se snažit vytvořit co nejnižší odhad. Pojede tedy hned příštím spojem této linky. Před nástupem do druhé linky však bude dlouho čekat, protože tato linka jezdí málo často.

Pokud bychom dopředu věděli, kdy jede druhý spoj, počkali bychom s použitím prvního spoje určitou dobu. Takto ušetřenou dobu lze považovat za zkrácení cesty. Doba příjezdu do cíle se však nemění. Nad daty, které jsme měli k dispozici, jsme tento problém nezaznamenali a proto jsme jeho řešení ponechali otevřené.

Problém lze částečně řešit vyhledáváním v opačném směru. Při vyhledávání v opačném směru dojde nejprve k výběru spojení linkou, která jezdí málo často. Teprve potom se vybere spoj linky, která jezdí relativně často. Do plánu cesty tak nebude zařazeno zbytečně dlouhé čekání a cesta tím bude pro uživatele příjemnější.

Vyhledávací mechanismus umí hledat cestu na základě dané doby příjezdu do cíle, což je vyhledávání v opačném směru. V takovém případě se zmíněnému problému částečně vyhneme. Analogický problém může však nastat i při tomto vyhledávání.

Úplného řešení by mělo být dosaženo při použití vícekriteriálního vyhledávání, kde lze zmíněnou situaci popsat kritérii na dobu čekání nebo pohodlí cestujícího.

6.3.2 Zvýšený počet přestupů

Pokud máme dvě linky, jejichž trasa je v určitém úseku souběžná, pak může vyhledávací algoritmus, který použitý v současné implementaci knihovny, vygenerovat zbytečný přestup. Dojde k tomu tak, že první linka jede dříve než druhá. Algoritmus mechanicky zvolí první linku a vytvoří co nejlepší odhad na příští zastávce. Tato linka však nevede k cíli cesty, ale odbočuje po několika společných zastávkách z trasy druhé linky. Druhá linka vede přímo k cíli. Algoritmus tedy provede přestup na druhou linku a pokračuje do cíle.

Pokud by algoritmus již na začátku zvolil druhou linku, k přestupu by nemuselo dojít. Podmínkou pro toto rozhodnutí by musela být dopředná znalost cesty, kterou linka povede. To by opět bylo splněno při vyhledávání v opačném směru. Avšak s tím, že zde může vyvstat analogický problém.

Tento problém by měl být zcela vyřešen při použití vícekriteriálního vyhledávání, kde by jedním z kritérií na podobu cesty byl právě počet přestupů.

6.3.3 Spolehlivost nalezené cesty

Jedním z charakteristických rysů plánování cest v hromadné dopravě je fakt, že jízdní řády jsou pouze předpis, jak mají spoje jezdit. V reálných případech mohou

mít spoje zpoždění nebo výpadky. V případě, že se tato situace opakuje, lze počítat s jistou pravděpodobností, zda spoj přijede včas nebo bude mít určité zpoždění. Pokud bychom znali tyto pravděpodobnosti, můžeme při plánování cesty zohlednit spolehlivost spoje. Případně při požadavku uživatele na nalezení spolehlivé cesty můžeme přizpůsobit vyhledávání tak, abychom zaručili ošetření nejpravděpodobnějších zpoždění.

Navíc může dojít k takovéto situaci. V naplánované cestě se vyskytuje spoj s řídkými intervaly, jehož případné nestihnutí by znamenalo velkou časovou ztrátu pro uživatele. V tom případě je vhodné naplánovat cestu tak, aby i při nepříznivých dopravních podmínkách, kdy dochází ke zpoždění spojů, bylo možné s vysokou pravděpodobností zaručit stihnutí kritického spoje.

Na spolehlivost cesty je možné mít i jiný pohled. Například můžeme chtít minimalizovat závislost trasy cesty na aktuálním čase. Tento jev je popsaný v kapitole 2.1. Úkolem by bylo najít takovou cestu, která by se v průběhu času příliš neměnila a byla přitom pro uživatele výhodná. Takový plán cesty je užitečný například v případě, že cestující neví zcela přesně, kdy započne cestu, nebo je pro něj cestování neznámými cestami problematické.

Pro plánování spojení v nespolehlivé dopravní síti je vhodné využít aproximativní přístup. Dvě takové metody jsou porovnány v [6].

6.3.4 Points of interest

Podobně jako je tomu u běžných turistických navigací, by i v této navigaci bylo možné doplnit body zájmu. Plán cesty by pak mohl být přizpůsoben podmínce navštívit konkrétní bod zájmu nebo nějakou kategorii bodů zájmu, která se nachází co nejblíže směru plánované cesty. Tyto úvahy vycházejí z předpokladu využití multikriteriálního plánování cesty.

6.3.5 Navigace pro cyklisty

Kombinovat hromadnou dopravu lze i s jiným typem uživatelů než jsou jen chodci. Dobrým příkladem jsou cyklisté. Jejich rychlost pohybu je značně vysoká. Délka úseku, který bude výhodnější jet na kole místo hromadnou dopravou, poroste. Na druhou stranu cyklisté mají oproti chodcům omezené možnosti v přepravě hromadnou dopravou. V síti tras použitelných pro cyklisty může velkou roli hrát bezpečnost, čímž se opět dostáváme k multikriteriálnímu vyhledávání.

6.3.6 Napojení na autonavigaci

Pohyb osobním automobilem v centru některých velkých měst může být nevýhodný z hlediska spolehlivosti, ceny, parkovacích možností v centru města atd. Řada měst používá parkoviště typu "P+R", která jsou určena pro krátkodobé návštěvníky města. Tato parkoviště jsou umístěna přímo u zastávek hromadné dopravy. Nabízí se tak myšlenka propojit plánování cesty osobním automobilem s možnostmi plánování cesty hromadnou dopravou. Opět by se zvýraznil ekologický efekt hromadné dopravy.

Kapitola 7

Závěr

Cílem této práce bylo navrhnout a implementovat knihovnu pro efektivní vyhledávání spojení v hromadné přepravě osob. Za tímto účelem byla provedena důkladná analýza specifik sítě hromadné přepravy osob. Na základě získaných zkušeností byla navržena základní verze vyhledávací knihovny tak, aby v rámci možností pokrývala danou problematiku.

V navazujícím vývoji byly zkoumány možnosti redukce vstupních dat jako nástroje pro zvýšení efektivity vyhledávání. Účinnost metody založené na redukci dopravní sítě byla následně otestována na dostupných datech.

Vyhledávací možnosti knihovny byly významně posíleny implementací komplexního vyhledávání, které dovoluje kombinovat pěší navigaci a vyhledávač spojení v hromadné dopravě. Plánování spojení v hromadné dopravě se tím výrazně zkvalitnilo.

Vyhledávací knihovna byla od počátku dimenzována pro použití v přenosných zařízeních typu PDA. Výsledná implementace knihovny je šetrná vůči paměti i výpočetní kapacitě. Její použitelnost byla ověřena nasazením knihovny v projektu JRGPS, který byl úspěšně obhájen v září 2009[1].

7.1 Základní řešení

Na dostupných datech jsme prokázali, že přímé vyhledávání spojení v hromadné dopravě není omezeno ani paměťovými ani výpočetními možnostmi přenosných zařízení. Navíc k zajištění aktuálnosti dat postačí i omezené možnosti připojení. Pro většinu evropských měst by parametry vyhledávání měly být srovnatelné, až na řádově větší města jako je Paříž, Londýn, Moskva a další.

V základním řešení bylo aplikováno několik postupů, které významně přispěli k celkové efektivitě vyhledávání. Výsledná implementace je díky tomu použitelná jak pro stolní počítače tak pro přenosná zařízení bez výrazných omezení.

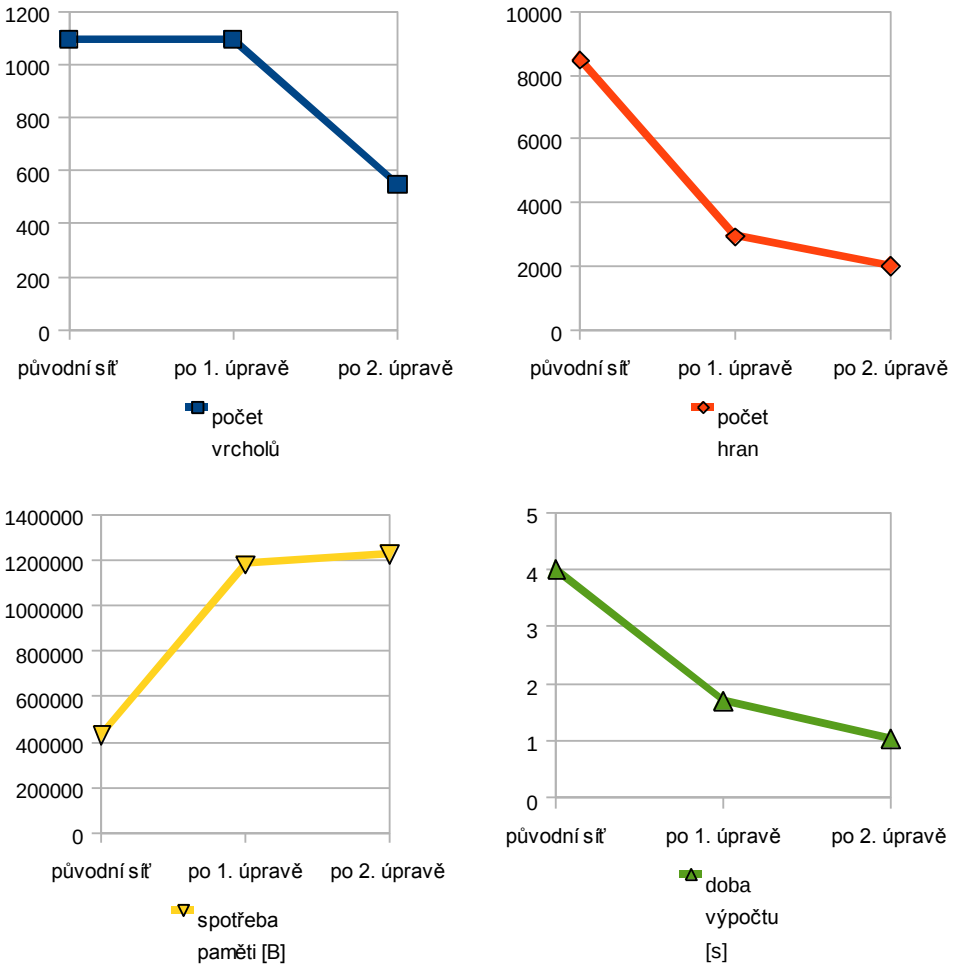
Vnitřní návrh knihovny je proveden tak, aby bylo možné dodatečně rozšiřovat implementaci a relativně snadno ji přizpůsobovat změnám. Vnější rozhraní umožňuje využití knihovny nezávisle na celkovém návrhu aplikace.

[1] Projektu bylo přiděleno mimořádné ohodnocení.

7.2 Redukce dopravní sítě

Na dostupných datech se nám podařilo ověřit proveditelnost redukce sítě MHD. Z provedených testů je vidět, že metoda redukce vstupní sítě může přinést podstatné zrychlení vyhledávání. Navíc tento přístup nebrání použití jiných metod vedoucích k dalšímu zvýšení efektivity vyhledávání.

Jak je vidět z obrázku 7.1, zrychlení výpočtu je dosaženo na úkor vyšší spotřeby paměti. Na druhou stranu celková spotřeba paměti je na rozumné úrovni. Navíc je možné přenést část zátěže na sekundární paměť přenosných zařízení. Tato paměť má typicky příznivé vlastnosti a její kapacita je výrazně méně omezena než výpočetní kapacita přenosných zařízení.



Obrázek 7.1: Posun v počtu vrcholů, hran, paměťových nároků a výpočetní rychlosti

7.2.1 Spotřeba paměti

Celkem jsou tři úrovně grafu – původní graf, po první a po druhé úpravě. Pro samotné vyhledávání potřebujeme udržovat v paměti dvě nejvyšší úrovně. Ve druhé úrovni se hledají počáteční úseky, pokud počáteční vrchol není "uzel". Ve třetí úrovni se hledá cesta mezi "uzly". Původní graf není potřeba mít v hlavní paměti. Velikost

jednotlivých úrovní grafu přitom klesá. Z toho plyne, že paměť potřebná na samotné grafy by neměla překročit dvojnásobek původní hodnoty.

Kvůli detekci přestupů v upraveném grafu potřebujeme data původních jízdních řádů. Dále potřebujeme sloučené jízdní řády k určení doby čekání. Výsledná spotřeba paměti zde záleží na způsobu reprezentace jízdních řádů. Pokud budeme udržovat časové tabulky pro každou zastávku zvlášť, pak sloučené časové tabulky budou zabírat mnohem méně paměti, než tabulky původní. Tím se opět dostaneme na hodnotu výrazně menší než dvojnásobek paměti. Pokud budeme udržovat časové tabulky pouze pro počáteční zastávku dané pseudolinky, pak počet sloučených časových tabulek bude záviset na počtu "potrubí". Každé potrubí má vlastní pseudolinku.

V současné implementaci jsou všechny tři úrovně grafu udržovány v paměti. To odráží i uvedená spotřeba paměti. Naším cílem do budoucna je zvolit vhodnou reprezentaci časových tabulek a minimalizovat paměť potřebnou paměť pro ukládání pomocných struktur.

7.2.2 Použitý hardware

Uvedené rychlosti výpočtu byly naměřeny na zařízení "HTC X7500" s procesorem Intel XScale 624MHz a 128MB operační paměti, z toho 65MB volné. Přenosná zařízení, pro která je současná implementace především určena, bývají často vybavena sekundární pamětí typu Flash. Tento typ pamětí je specifický tím, že zápis je až několikanásobně pomalejší než čtení. Také proto je výhodné, že aplikovaná zlepšení nevyžadují častý zápis do sekundární paměti. Další specifickou vlastností je, že přístup na různá místa v paměti není tak problematický jako například u pevných disků. Proto je možné udržovat původní graf v sekundární paměti, aniž by uživatel zaznamenal znatelné zpomalení.

7.2.3 Ortogonalita řešení

Výhoda úprav uvedených v této práci je, že nijak nemění podstatu problému. Pouze redukují velikost vstupních dat. Tím nijak nebrání aplikaci dalších postupů pro redukci složitosti vyhledávání.

7.3 Komplexní navigace

Prokázali jsme, že v současné době mají přenosná zařízení dostatečnou výpočetní i paměťovou kapacitu, aby bylo možné provádět základní plánování kombinované trasy přímo v zařízení. Díky tomu může být aplikace nezávislá na aktuální dostupnosti připojení. Aktualizace jízdních řádů mohou být prováděny až v okamžiku, kdy je připojení dostupné, není-li k dispozici jiný mechanismus pro aktualizaci.

Klíčovým prvkem je vzájemná kombinace vyhledávacích sítí. Podařilo se nám realizovat kombinaci sítě hromadné dopravy a pěší sítě pomocí předvýpočtu tak, aby náročnost vyhledávání nepřekročila možnosti přenosných zařízení. Přitom míra předvypočítaných dat by neměla omezovat možnosti plánování kombinované cesty. Výsledný plán by tedy měl být ve většině případů shodný s plánem, který bychom získali bez použití předvýpočtu, tedy nad sloučenou sítí hromadné dopravy a pěších cest.

7.3.1 Výhody kombinace dvou odlišných druhů navigace

Efektivita cesty

Propojením obou sítí získáváme mnohem větší možnosti plánování, než kdybychom používali jen jeden druh navigace. Navíc máme pro pěší úseky daleko podrobnější informace než kdybychom používali čistě navigaci pro hromadnou dopravu. Výsledný plán cesty tak nemusí počítat s odhady přestupů a je mnohem přesnější. To nám umožňuje naplánovat efektivnější a spolehlivější cestu.

Ekologický aspekt

Snažíme se nabídnout širokému okruhu uživatelů pohodlné a přesné plánování cest ve městě s využitím hromadné dopravy. Tím zlepšujeme komfort využívání hromadné dopravy a úroveň služeb s dopravou souvisejících. Čím více uživatelů bude používat hromadnou dopravu na úkor méně ekologických alternativ, tím menší bude dopad městské dopravy na životní prostředí.

7.3.2 Nevýhody kombinace dvou odlišných druhů navigace

Odlišné plánování

Snažíme se propojit dvě velice odlišné sítě. V každé z nich platí jiná pravidla a heuristiky, které lze úspěšně aplikovat v jedné síti nemusí vůbec platit v té druhé. Je tedy potřeba vyhledávání oddělit. Na druhou stranu plán celkové cesty by měl naplňovat společná kritéria. K jejich dosažení je však často potřeba odlišný mechanismus.

Různé zdroje dat

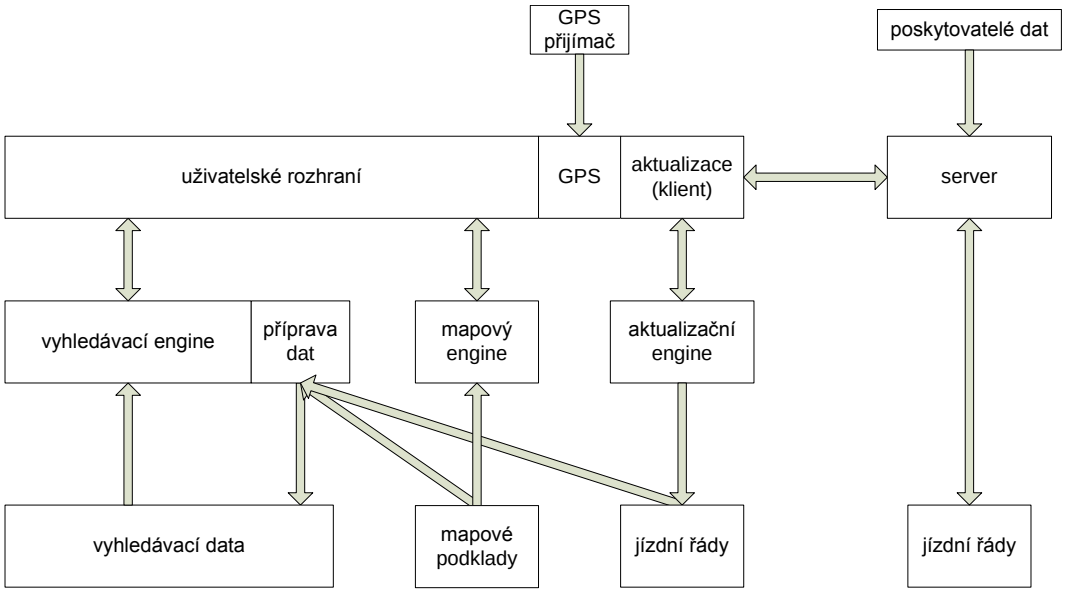
Pro síť hromadné dopravy potřebujeme data jízdních řádů a data o rozmístění zastávkových ostrůvků. Pro pěší síť potřebujeme mapové podklady včetně podrobností potřebných pro navigaci chodců. Pro provoz aplikace tedy potřebujeme data od několika různých subjektů. V případě komerčního nasazení aplikace vyvstává řada otázek souvisejících se vzájemnými vztahy mezi zúčastněnými subjekty. Například jak rozdělit zodpovědnost za konzistenci dat?

7.4 Projekt JRGPS

Vyhledávací knihovna vyvíjená v rámci této práce se stala základním kamenem softwarového projektu JRGPS. Jeho cílem se stalo vytvořit prototyp aplikace pro plánování cest kombinující hromadnou dopravu a chůzi.

Aplikace byla vyvíjena pro platformu Windows Mobile. Výpočetně náročné procedury jsou zapouzdřeny v dynamicky linkovaných knihovnách. Díky tomu je efektivně využita výpočetní i paměťová kapacita přenosných zařízení. Knihovny jsou psány v přenositelném jazyce a je proto možné je relativně snadno použít pro další platformy. Pouze uživatelské rozhraní je do jisté míry systémově závislé.

Práce na projektu byla striktně rozdělena podle jednotlivých modulů aplikace. Vzájemné propojení modulů je znázorněno na obrázku 7.2. Rozhraní mezi jednotlivými částmi je popsáno v programátorské dokumentaci.



Obrázek 7.2: Schema aplikace
Vyhledávací knihovna obsahuje "vyhledávací engine", "aktualizační engine" a dále metody pro "přípravu dat".

Členové týmu a jejich úlohy v rámci projektu:

RNDr. Michal Žemlička, Ph.D. – vedoucí projektu

Vladislav Martínek – vyhledávací knihovna

Tomáš Kučera – uživatelské rozhraní

Zdeněk Reischig – server

Dominik Malý – mapová knihovna

Při vzniku projektu byla podstatná část vyhledávací knihovny již implementována. Z důvodu vyšší kompatibility mezi moduly byly šablony datových tříd pro vyhledávací knihovnu použity také pro mapovou knihovnu.

Vyhledávací knihovna byla implementována bez přispění ostatních členů týmu podílejících se na vývoji projektu. Uvedené části programátorské a administrátorské dokumentace jsou dílem autora této diplomové práce, stejně tak doprovodná aplikace "Viewer" a generátory datové sady.

7.4.1 Uživatelská dokumentace

Uživatelská dokumentace k projektu je obsažena v "user_manual.pdf" na přiloženém CD.

7.4.2 Administrátorská dokumentace

Administrátorská dokumentace k projektu popisuje údržbu datové sady a její generování pomocí příslušných nástrojů. Je obsažena v "doc_admin.pdf" na přiloženém CD.

Vyhledávací knihovna implementuje veškeré datové třídy a metody potřebné pro generování a modifikaci vnitřní datové sady. Nástroje pro generování datové sady jsou pouze uživatelská rozhraní pro třídy a metody obsažené ve vyhledávací knihovně. Díky tomu je zaručena snadná udržovatelnost vnitřní reprezentace dat.

Formát vstupních dat pro generování je podrobně popsán v administrátorské dokumentaci – "doc_admin.pdf", strany 30–53. Formát aktualizačních souborů je popsán v administrátorské dokumentaci – "doc_admin.pdf", strany 54–66.

7.4.3 Programátorská dokumentace

Programátorská dokumentace k projektu je obsažena v "doc_prog.pdf" na přiloženém CD.

Rozhraní knihovny a chybové kódy jsou popsané v programátorské dokumentaci – "doc_prog.pdf", strany 16–18. Základní části vyhledávací knihovny jsou popsané v programátorské dokumentaci – "doc_prog.pdf", strany 20–23. Vnitřní reprezentace dat a formát binárních souborů je podrobně popsán v programátorské dokumentaci – "doc_prog.pdf", strany 82–117.

7.5 Viewer

"Viewer" je doprovodná aplikace pro stolní počítač, která slouží k ladění vyhledávací knihovny, generátorů dat a pro předvýpočet některých částí datové sady. Tato aplikace byla vytvořena jako ladící nástroj pro projektové knihovny a jejich rozhraní. Slouží také k ručnímu ověřování správnosti vygenerovaných dat. Až na hlavní formulář sdílí veškeré zdrojové soubory s uživatelskou aplikací pro PDA.

Literatura

[1] Dopravní podnik hlavního města Prahy, a.s.: (Užitečná dopravní schémata) `http://www.dpp.cz/uzitecna-dopravni-schemata/`.

[2] Delling, D., Sanders, P., Schultes, D., Wagner, D.: Highway Hierarchies Star. Technical report, ARRIVAL Project (2006) work presented at 9th DIMACS Challenge on Shortest Paths.

[3] Edsger. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik 1 (1959), 269–271.* Amsterdam.

[4] J. Černý, "Nejkratší cesta v grafu," in *Základní grafové algoritmy*, 2008, pp. 90–109. Publikace dostupná on-line(naposledy ověřeno 8.4.2010): `http://kam.mff.cuni.cz/~kuba/ka/ka.pdf`.

[5] R. Kasperovics, M. H. Böhlen, and J. Gamper, "Representing public transport schedules as repeating trips," in *TIME '08: Proceedings of the 2008 15th International Symposium on Temporal Representation and Reasoning.* Washington, DC, USA: IEEE Computer Society, 2008, pp. 54–58.

[6] J. Koszelew, "Two methods of quasi-optimal routes generation in public transportation network," in *CISIM '08: Proceedings of the 2008 7th Computer Information Systems and Industrial Management Applications.* Washington, DC, USA: IEEE Computer Society, 2008, pp. 231–236.

[7] A. Liebers, D. Wagner and K. Weihe, "On the hardness of recognizing bundles in time table graphs," in *In Proceedings 25th International Workshop on Graph-Theoretic Concepts in Computer Science.* Springer Lecture, 1999, pp. 325–337.

[8] A. Liebers and K. Weihe, "Recognizing bundles in time table graphs - a structural approach," in *Algorithm Engineering*, ser. Lecture Notes in Computer Science, S. Näher and D. Wagner, Eds., vol. 1982. Springer, 2000, pp. 87–98.

[9] V. Martínek and M. Žemlička, "Speeding up shortest path search in public transport networks," in *DATESO 2009*, K. Richta, J. Pokorný, and V. Snášel, Eds. Prague, Czech Republic: Czech Technical University in Prague, 2009, pp. 1–12.

[10] V. Martínek and M. Žemlička, "Some issues and solutions for complex navigation systems: Experience from JRGPS project," in *2010 Fifth International Conference on Systems.* IEEE CS Press, 2010, pp. 92–98.

[11] M. Müller-Hannemann and K. Weihe, "On the cardinality of the pareto set in bicriteria shortest path problems." *Annals OR*, vol. 147, no. 1, pp. 269–286, 2006.

[12] E. Pyrga, F. Schulz, D. Wagner, and C. Zaroliagis, "Efficient models for timetable information in public transportation systems," *J. Exp. Algorithmics*, vol. 12, pp. 1–39, 2008.

[13] K. Weihe, "Covering trains by stations or the power of data reduction," in *Proceedings of "Algorithms and Experiments" (ALEX98)*, R. Battiti and A. A. Bertossi, Eds., 1998, pp. 1–8.