Česká zemědělská univerzita v Praze
Provozně ekonomická fakulta
Katedra systémového inženýrství

Bakalářská práce

Automatický převod lokací na matici sazeb pro problém obchodního cestujícího

VYTVOŘIL MATYÁŠ ELICER

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Matyáš Elicer

Informatika

Název práce

Automatický převod lokací na matici sazeb pro problém obchodního cestujícího

Název anglicky

Automatic conversion of locations to a matrix of weights for the travelling salesman problem

---

Cíle práce

Cílem této bakalářské práce je vytvořit program, jehož výstupem bude matice sazeb pro modelování problému obchodního cestujícího. Matice bude vytvořena na základě předložené množiny lokací na mapě. Jednotlivými prvky matice budou sazby popisující délku tras mezi jednotlivými páry lokací, nebo čas nutný k jejich překonání; celá výstupní matice bude formátována pro následné využití v programu TSPkosa, nebo jako textový výstup v podobě modelu lineárního celočíselného programování.

Hlavním smyslem práce je zjednodušení a automatizace jinak zdlouhavé manuální práce pro řešitele zmíněného problému z různých oborů. Program bude v zájmu etiky a široké dostupnosti použitelný pro uživatele operačních systémů GNU a Microsoft Windows, včetně uživatelského rozhraní.

Metodika

Prvním krokem v rámci hlavní náplně práce bude průzkum dostupných zdrojů informací o zmapovaných trasách mezi lokacemi, a dostupných nástrojů pro získávání těchto informací, které by program mohl využívat. Zdroje a nástroje budou voleny tak, aby případné využití programu nebylo limitováno zvýšenými náklady. Za tím účelem bude na zřetel brána také jejich dlouhodobá použitelnost.

Po jejich zvolení bude vytvořen samotný program, jehož součástí bude přívětivé uživatelské rozhraní, a který bude doprovázen kompletním návodem k celému procesu od instalace přes případné nastavení a vedlejší kroky až po samotný převod lokací na matici. Ostatní technologie užité pro tvorbu programu budou voleny na základě způsobilosti splňovat vymezené technické požadavky v první řadě, a také na základě osobních zkušeností a preferencí autora práce v řadě druhé.

Doporučený rozsah práce

30-40 stran

Klíčová slova

Mapy, Matice sazeb, Problém obchodního cestujícího, Program, Software

---

Doporučené zdroje informací

APPLEGATE, D.L., BIXBY, R.E., CHVÁTAL, V. and COOK, W.J. The Traveling Salesman Problem – a Computational Study. Oxford: Princeton University Press. 2006. ISBN 978-0-691-12993-8

PELIKÁN, J. 1993. Praktikum z operačního výzkumu. 1.vyd. Praha: VŠE. 86 s. ISBN 80-7079-135-7

PELIKÁN, J. 2001. Diskrétní modely v operačním výzkumu. 1. vyd. Praha: Professional Publishing. 164 s. ISBN 80-86419-17-7

ŠUBRT, Tomáš a kolektiv, 2019. Ekonomicko-matematické metody. 3. vyd. Plzeň: Aleš Čeněk, s.r.o. ISBN 978-80-7380-762-7

Předběžný termín obhajoby

2022/23 LS – PEF

Vedoucí práce

doc. Ing. Igor Krejčí, Ph.D.

Garantující pracoviště

Katedra systémového inženýrství

Elektronicky schváleno dne 16. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 30. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 14. 03. 2023

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci „Automatický převod lokací na matici sazeb pro problém obchodního cestujícího“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15. března 2023 .....

Poděkování

Rád bych touto cestou poděkoval vedoucímu práce, doc. Ing. Igorovi KREJČÍMU, Ph.D., za odborné vedení, ochotnou pomoc a velkou trpělivost. Dále bych chtěl poděkovat své rodině a svým blízkým za podporu v době studia i mimo ni. A v neposlední řadě také komunitám projektu *OpenStreetMap* a svobodného softwaru obecně, na jejichž dlouhodobé práci tato práce stojí.

Automatický převod lokací na matici sazeb pro problém obchodního cestujícího

Abstrakt

Bakalářská práce se zabývá tvorbou matice sazeb pro jednookruhový okružní dopravní problém. Za hlavní cíl má vytvořit program, který by tuto činnost automatizoval. V první části je zkoumán samotný problém, a především pak dílčí úloha získání údajů odpovídajících skutečným dopravním sítím. Dále jsou zkoumány dostupné prostředky, které by v této úloze bylo možné využít, z nichž je jako nejvhodnější vybrán *OpenStreetMap*. V druhé části práce je pak přistoupeno k tvorbě a popisu samotného programu, které předchází výběr vhodných technologií, návrh programové struktury, a návrh jednoduchého uživatelského rozhraní. Zmíněný program je pak funkčním výstupem této práce.

Klíčová slova: Automatizace, Mapy, Matice sazeb, Okružní dopravní problém, OpenStreetMap, Otevřená data, Problém obchodního cestujícího, Program, Software

Automatic conversion of locations to a matrix of weights for the travelling salesman problem

Abstract

This bachelor's thesis is concerned with the forming of a matrix of weights for the travelling salesman problem. Its main goal is to create a program, which would automatise the task. The first part explores the problem itself, especially the partial task of acquiring the data corresponding to actual transportation networks. Disponible resources which could be used for this work are explored next, of which *OpenStreetMap* is chosen as the most suitable. In the second part, the program itself is created and described, after the suitable technologies are chosen, and the program structure and a simple user interface are designed. The aforementioned program is a functional result of this work.

Keywords: Automatisation, Maps, Matrix of weights, Open data, OpenStreetMap, Travelling salesman problem, Program, Software

Obsah

1 Úvod

Problém obchodního cestujícího je matematickým problémem, jehož podstata spočívá v nalezení optimální (v praxi tedy nejkratší nebo nejrychlejší) cesty několika místy. Vedle častého předmětu vědeckého zkoumání, v kterézto roli zůstává doposud zajímavým a aktuálním, je také problémem často řešeným v praxi, ať už školní či pracovní, a to zejména v oblasti dopravy. Konkrétně se tedy jedná o hledání cesty mezi několika skutečnými lokacemi.

Předpokladem pro úspěšné řešení takové úlohy je nabytí matice sazeb, které vyjadřují náročnost cest mezi jednotlivými lokacemi. To však může samo o sobě být poměrně náročné, pokud se všechny sazby získávají postupně, jedna po jedné, ruční prací; počet sazeb v matici totiž exponenciálně roste vůči počtu lokací.

Jedním z snad nejdostupnějších způsobů, jak sazby získat, je právě postupné vyhledávání cest za pomoci veřejně dostupných služeb. Opakovat tento úkon ručně neustále dokola je však náročné, a nadto nezáživné. Jedná se tedy o úkon velice vhodný k automatizaci, která by v tomto případě dokázala ušetřit řešitelům čas, popř. i peníze, neboť právě těmi bývá podmíněn přístup k již existujícím nástrojům, které jsou schopny tento problém alespoň částečně řešit.

Je zde tedy prostor pro nástroj, který by dovedl právě tyto úkony automatizovat, a zpřístupnit tak řešitelům této úlohy elegantní řešení podproblému nabytí matice sazeb, zachovávaje přitom lineární růst úsilí potřebného z jejich strany. Zaplnění tohoto prostoru bude tak hlavní náplní této práce.

2 Cíle práce a metodika

2.1 Cíle

Hlavním cílem práce je vytvoření multi-platformního počítačového programu vybaveného přívětivým uživatelským rozhraním, jehož hlavní (a v zásadě také jedinou) funkcí bude převod seznamu lokací na matici sazeb pro řešení problému obchodního cestujícího. Výstupní matice sazeb má být ve formátu, který může být zároveň vstupem programu *TSPkosa*.

2.1.1 Vlastnosti programu

Multi-platformnost

Multi-platformností programu je v tomto případě myšlena existence alespoň dvou spustitelných verzí programu, a to konkrétně jedné spustitelné na operačních systémech *Microsoft Windows* (dále jen Windows), a druhé spustitelné na operačním systému *GNU* s jádrem *Linux* (dále jen GNU/Linux), potažmo na jeho různých distribucích a odvozeninách. Smyslem této vlastnosti je předejít závislosti celého programu na konkrétní platformě, a jeho zpřístupnění většímu počtu uživatelů, včetně těch, kteří preferují svobodný software[1].

Nutno podotknout, že uživatelská základna systému GNU/Linux není ve srovnání se systémem Windows nikterak velká, ba naopak je několikanásobně menší, ale na druhou stranu je systém GNU/Linux cenově dostupnější, ba i zadarmo, a to na široké škále zařízení. Lidé, co nejsou uživateli tohoto systému, ale nemají přístup k systému Windows, se jimi tak relativně snadno[2] mohou stát.

Přívětivé uživatelské rozhraní

Přívětivostí uživatelského rozhraní je pak myšlena jeho jednoduchost a přímočarost. Jeho obsluha by měla být co nejvíce zřejmá a povědomá, a uživatel by k jeho obsluze neměl potřebovat jiné než zcela základní znalosti práce s osobním počítačem, a s prostředím, ve kterém program bude spouštěn. Program by měl být pokud možno samostatný; k jeho spuštění a obsluze by neměla být nutná složité nastavování či instalace podružných programů, ani znalost jiných specializovaných nástrojů.

To samé platí o znalostech z jiných oblastí; nepředpokládá se tedy, že by se uživatel hlouběji pohyboval v oblastech matematiky, ekonomie, či dopravy, se kterými je problém obchodního cestujícího spjat. To sice od uživatele vzhledem k povaze programu čekat lze, ale funkce programu se to netýká. Předpokládá se pouze chápání podstaty samotného problému.

2.1.2 Funkce programu

Samotná funkce programu má totiž spočívat právě v převodu lokací na matice sazeb; tuto funkci by program měl plnit dobře a spolehlivě, a další zpracování výsledných dat by pak měl přenechat programům k tomu způsobilým (jmenovitě pak programu *TSPkosa*).

S ohledem na právě zmíněné vlastnosti by samotný převod měl fungovat tím způsobem, že uživatel předá programu seznam matic, a ten uživateli vrátí matici sazeb jednotlivých cest mezi všemi myslitelnými páry z daných lokací. Na uživateli by měla být požadována pouze znalost běžných názvů jednotlivých lokací a jejich umístnění (t.j. adresa, název obce, poštovní směrovací číslo, kraj, stát, apod). Bude-li program vyžadovat přesnější či jednoznačnější údaje, jako např. souřadnice či jiné technické identifikátory daných lokací, měl by uživateli poskytnout jasné instrukce vedoucí k jejich nabytí a následnému použití v programu. Uživatel musí mít možnost údaje zadat prostým vepsáním, ať už do jednotlivých polí, nebo do jednoho pole s tím, že od sebe jednotlivé údaje oddělí.

Sazby jednotlivých cest mají pro každý pár lokací vyjadřovat náročnost cesty z lokace jedné do lokace druhé. Za nejspolehlivější způsob, jak vyjádřit náročnost cesty, se v tomto případě předpokládá vzdálenost, jež cesta pokrývá, popř. přímo i časová náročnost (která se od celkové vzdálenosti odvíjí).

2.1.3 Omezení

Nutno podotknout, že samotné řešení problému obchodního cestujícího matematickým výpočtem není jedním z cílů programu, a jeho velmi podrobný průzkum tedy není ani jednou ze součástí hlavní náplně této práce. Role programu v řešení dané úlohy spočívá pouze v převodu lokací na matici sazeb, kteráž bude následně zkoumána a zpracovávána vně programu; kompetence programu končí při odpovědnosti matici takovému zpracovávání uzpůsobit.

2.2 Metodika

V rámci práce byly z dostupné literatury a jiných zdrojů zkoumány různé části řešené problematiky. Zkoumán byl v první řadě samotný problém obchodního cestujícího, přičemž byla věnována zvýšená pozornost jeho výskytu a využití v praxi.

Dalším, náplni práce samotné snad i bližším předmětem zkoumání byla úloha hledání optimální cesty grafem, a též i její praktické aplikace. Byla tak odůvodněna potřeba map jakožto modelu skutečných dopravních sítí, a nadále byly zkoumány dostupné technologie pro jejich získání a následnou práci s nimi. Především se jednalo o služby *Google Maps*, *Bing Maps*, *Mapy.cz*, a projekt *OpenStreetMap*.

Po poznání takových technologií a jejich vlastností byly uváženy prostředky autora a potřeby předpokládaných uživatelů programu. Z těch byly logicky odvozeny předpoklady, které by program a jím užívané technologie měli splňovat, aby výstup programu odpovídal zmíněným potřebám, a mohl tedy být využit k řešení praktických úloh. V kontextu daných předpokladů byly jednotlivé technologie posouzeny, aby byl zvolen nejvhodnější kandidát, jímž byl shledán projekt *OpenStreetMap* spolu s od něj odvozenými nástroji.

Pro účely samotného programování byly prozkoumány některé nástroje pro tvorbu grafických aplikací. Hlavními vlastnostmi, které u nich byly vyhledávány, byly: dostupnost,

spolehlivost, snadné užívání, a právě podpora cílených platforem. Průběžně byly vyzkoušeny různé knihovny, ale nakonec byla pro její rozšířenost a technické kvality zvolena multiplatformní sada nástrojů *Qt*.

S její pomocí, a za použití nástrojů a služeb spjatých s projektem *OpenStreetMap*, byl vytvořen program *TSPbrousek* v jazyce *C++*. Pro jednotlivé třídy byla vytvořena poměrně jednoduchá třídní hierarchie. Program byl následně testován na obou cílových platformách, kde byl v obou případech shledán plně funkčním. Výstup programu a výsledek práce jako takové byli posléze zhodnoceny.

3 Současný stav poznání řešené problematiky

3.1 Problém obchodního cestujícího

Problém obchodního cestujícího (obdobně též známý pod názvem „jednookruhový okružní dopravní problém“, popř. zkrátka „okružní dopravní problém“) je tzv. *NP-úplným*[3, s. 102–103] matematickým problémem; to v tomto případě znamená, že neexistuje žádný známý a efektivní algoritmus pro nalezení jeho matematického optima, a pro jeho řešení je tedy třeba prozkoumat všechny možné varianty.

Podstata problému spočívá v nalezení optimální (tzn. co nejkratší, potažmo nejrychlejší) cesty, jenž postupně prochází všemi danými místy, a každým z nich právě jednou, nikoliv vícekrát. Jeho asymptotická složitost je tedy $\mathcal{O}(n!)$. Úlohu lze také matematicky formulovat vzorcem.

Je-li dáno n míst, buď pro každou přímou cestu z místa i do místa j stanovena proměnná c_{ij}, která vyjadřuje sazbu (tedy ocenění) dané cesty, a bivalentní proměnná x_{ij}, jejíž hodnota je 1 pokud je daná cesta využita, nebo 0 v opačném případě. Úlohu pak lze vyjádřit lineární funkcí,[3, s. 102–103] u které bude hledáno minimum:

$$\sum_{i=j}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \rightarrow \mathrm{MIN}$$

$$x_{ij} \in \{0;1\} \qquad i=1,2,\ldots,n \qquad j=1,2,\ldots,n$$

Samotná tato formulace je však neúplná, neboť by připouštěla i „řešení“, kde je výsledek neúplný, nebo dokonce nulový. Je tedy třeba stanovit podmínky[3, s. 102–103], které zaručí, aby z každého místa vedla právě jedna cesta, a právě jedna cesta vedla také do něj:

$$\sum_{i=1}^{n} x_{ij}=1 \qquad j=1,2,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij}=1 \qquad i=1,2,\ldots,n$$

Tyto podmínky však nevyloučí „řešení“, která nevyjadřují jednu souvislou kružnici, nýbrž několik oddělených kružnic. Je-li tedy $n=6$, mohlo by optimálním řešením být např. ohodnocení $x_{12}=x_{23}=x_{31}=1=x_{45}=x_{56}=x_{64}$, což není žádoucí. Taková řešení je tedy dále třeba vyloučit, čehož lze dosáhnout použitím tzv. *smyčkových podmínek*[4], které dodržení jednoho okruhu zaručí. Existuje jich vícero, ale za příklad se zde dávají podmínky *Tuckerovy*, jejichž počet je roven počtu hran (což je méně než v případě ostatních podmínek). Využívají pomocné proměnné u_i, vyjadřující pořadí místa i. Jejich formulace je následovná:

$$u_i - u_j + n x_{ij} \leq n-1 \qquad i=1,2,\ldots,n \qquad j=2,3,\ldots n \qquad i \neq j$$

3.1.1 Historický kontext

Jako samostatný předmět intenzivního matematického zkoumání se problém obchodního cestujícího začal objevovat již v první polovině 20. století[5], ačkoliv v matematickém oboru se obdoba problému v určitých podobách objevuje již o století dříve. Zmínit lze například hlavolam publikovaný s. Williamem Rowanem HAMILTONEM, mezi jehož variacemi se vyskytuje i úloha nalezení hamiltonovské kružnice (pojmenované právě po věhlasném autorovi hlavolamu) mezi městy znázorněnými na dvanáctistěnu představujícím jakýsi model zeměkoule [6, s. 31–35]. To samé však nelze tvrdit o problému samotném, který je bezpochyby mnohem starší. Je tomu tak důsledkem jeho přirozeného výskytu v každodenním životě; dejme tomu při výkonu různých povolání, mezi které patří (popř. patřili) právě obchodní cestující, snažící se na své cestě navštívit různá města, aniž by museli procházet tím samým městem vícekrát za sebou; setkávali se s ním i úředníci při sčítání produkce v zemědělství[5].

3.1.2 Současný stav

Mezi dalšími lidmi, co se s tímto problémem (nebo jemu obdobným) mohou – v dnešní době snad ještě častěji – setkat, se ale najdou i turisté[5], kteří hodlají navštívit co nejvíce míst, aniž by museli podnikat (dosti možná nákladné) cesty do míst, která už viděli; to se úzce týká i průvodců, kteří jim v tom v rámci své práce dopomáhají. Dále mohou být jmenováni policisté při plánování hlídek či obchůzek, nebo řidiči při rozvozu materiálu, zboží, odpadků,[4] nebo pracující v hromadné dopravě.

Problém se rozhodně nevztahuje jen na dopravu lidí či materiálu na větší vzdálenosti po povrchu zemském. Setkat se s ním je možné i při výrobě a montáži elektronických součástek, stavbě turbín pro vzdušnou přepravu, nakládání materiálu ve skladech, či zkoumání vlastností krystalických látek[7]. Tato bakalářská práce má však blíže k dříve zmíněným výskytům problému obchodního cestujícího, neboť se zabývá převodem reálných lokací na mapách, kterých se přeci jen daleko více týká cestování, než úkony zcela běžně se odehrávající uvnitř jedné budovy.

Jen v předcházejících pěti letech bylo na Provozně ekonomické fakultě České zemědělské univerzity obhájeno 43 závěrečných prací[8] kde byl jako klíčové slovo uveden *problém obchodního cestujícího*, nebo slovo *okružní*, a všechny z nich se zabývaly právě dopravou (resp. logistikou). Je tedy zřejmé, že právě tento kontext je i pro tuto práci velmi relevantní, a program by mohl najít praktické využití.

3.1.3 Vztah problému k praktickému převodu lokací

Na samotný problém lze nahlížet také z pohledu teorie grafů, kdy se jednotlivými místy chápou uzly, které jsou propojeny hranami.[9, s. 17] V případě, že matice sazeb doposud nebyla nabyta, mohou k jejímu získání dopomoct i jiné problémy, kterými se teorie grafů zabývá. Jedná se zejména o problém nalezení optimální cesty grafem, který lze v rámci hlavního problému taktéž řešit při uvážení lokací jakožto uzlů grafu.

3.2 Hledání optimální cesty grafem

Nalezení optimální cesty grafem (t.j. takové cesty, kterou se lze co nejsnadněji dopravit z vybraného počátečního uzlu do vybraného koncového uzlu v porovnání s ostatními cestami) je úlohou velmi často řešenou při praktickém využití teorie grafů. Nejrozšířenější a nejčastěji používanou základní metodou využívanou k jejímu řešení je Dijkstrův algoritmus, v jehož rámci se kromě jedné kýžené cesty vedoucí do koncového uzlu nachází také cesty vedoucí do uzlů jiných, a to nikoliv na úkor eficience[3, s. 281] hledání té hlavní cesty mezi vybranými uzly.

Pro řešení problému obchodního cestujícího v praxi je právě tato úloha stěžejní, neboť je pro dosažení co nejoptimálnějšího výsledku nutné pracovat právě s optimálními cestami mezi jednotlivými uzly. Uvažujeme-li, že skutečnou dopravní síť lze znázornit jako orientovaný graf, lze i pro hledání optimálních cest v takové síti využít stejné či obdobné postupy z teorie grafů.

Poznámka 1. Třebaže je v českém jazyce zvykem hovořit v teorii grafů o nejkratší *cestě* grafem[3], právě pojem *cesta* lze chápat z hlediska zeměpisného také jako „*úsek, pruh terénu upravený pro chůzi, jízdu, dopravu*“[10], což je při práci právě s takovými skutečnými cestami také velmi relevantní definice. Pro zamezení matoucích záměn těchto dvou významů bude tedy ve zbytku práce užíván přednostně pojem *trasa*, vyjadřující právě cestu ve smyslu matematickém (kteráž může v sobě zahrnovat hned vícero cest ve smyslu zeměpisném).

3.3 Hledání cest mezi skutečnými lokacemi

Snad nejpřímočařejším způsobem hledání cesty mezi dvěma lokacemi je vlastní průzkum terénu a cestování: pokud se někdo sám vypraví z lokace jedné do lokace druhé, vcelku spolehlivě si tím ověří, že po zvolené trase lze mezi lokacemi skutečně cestovat, a jak dlouho taková trvá. Tato metoda má však hned několik zjevných nedostatků:

- Informace získané tímto způsobem mohou být zkreslené, tzn. že průběh podniknuté cesty se může významně lišit od běžného průběhu té samé cesty, v závislosti na okolnostech. Pro získání důvěryhodnějších informací by tedy tu samou cestu bylo nutné podniknout hned mnohokrát.
- Cestování je časově náročné i při jedné cestě, a jako způsob nabývání informací o velkém počtu různých (a dlouhých) tras je tedy značně nepraktické.
- Cestování může být (zvláště pokud má cestující za cíl omezit náročnost časovou) náročné i finančně.

Naštěstí k této metodě existuje dostupná alternativa v podobě map, jakožto polohopisných modelů skutečného světa, zejména pak jeho dopravních sítí (které mohou být – a mnohdy také jsou[11, s. 36] – tvořeny právě na základě poznatků vzešedších z cestování). Lze-li

z mapy vyčíst délku jednotlivých cest, jako i jejich částí, lze z ní také snadno odvodit délku celkové trasy. Pokud je známá i rychlost, jakou je cestující schopen se po trase pohybovat, lze z těchto údajů již celkem snadno vypočítat také čas, který cesta po dané trase zabere. (Toho lze na nějaké úrovni docílit např. za pomoci základního fyzikálního vzorce $t = \frac{s}{v}$.[12, s. 12]) Tímto lze tedy dosáhnout alespoň přibližného časového ohodnocení trasy.

Pro nabytí celé matice sazeb, kterou by bylo možné využít při řešení problému obchodního cestujícího, je třeba takto postupovat pro každý pár lokací zvlášť. Obzvláště důležitý je pak detail, že při práci s mapami s detailními údaji o skutečných cestách je třeba takto u každého páru postupovat hned dvakrát: jednou pro nalezení cesty z jedné lokace do druhé, a podruhé pro nalezení cesty zpět z druhé lokace do první. Skutečné cesty totiž ne vždy umožňují stejně rychlou cestu tam i zpět; při automobilové dopravě se můžeme setkat s jednosměrnými ulicemi, které by mohly cestu výrazně usnadnit, či naopak ztížit (podle toho, kterým směrem cestující právě cestuje); při chůzi bychom zase mohli předpokládat náročnější cestu do kopce, než by tomu tak bylo při chůzi z kopce. Jelikož tedy při každé cestě záleží na pořadí obou dvou lokací, výsledná matice nemusí nutně být symetrická, a pro dvě lokace mohou existovat dvě různé trasy, z nichž jedna je optimální v jednom směru, a ta druhá v tom druhém.

3.4 Možnost využití počítačových programů

Výše popsaný postup lze automatizovat za pomoci počítačových programů. Je k tomu však třeba mapa uložená v číslicové formě, jejíž cesty jsou již osazeny vzdálenostmi, nebo z jejichž dat lze vzdálenosti vypočítat. (Je-li, dejme tomu, cesta znázorněna křivkou, lze její délku odvodit z jejích rozměrů.)

Toto řešení je oproti vlastnímu výzkumu a ručnímu výpočtu jistě mnohem rychlejší i pohodlnější, ale je podmíněno ne méně než třemi předpoklady:

1. Kvalita číslicových údajů musí být na dostatečně vysoké úrovni, čili musí alespoň přibližně odpovídat realitě.
2. Program, který s danými údaji pracuje, musí být s to nalézt mezi cestami obsaženými v mapě optimální trasu mezi dvěma lokacemi.
3. Program i údaje, se kterými pracuje, musí být dostupné uživatelům, a to takovým způsobem, aby je mohli využít k dosažení kýžených výpočtů.

Pokud jde o kvalitu číslicových údajů, je především důležité, aby odpovídaly skutečnosti. V případě, že se poloha lokací a tvar i rozměry jednotlivých cest významně liší od polohy skutečných lokací a tvaru či rozměrů skutečných cest, může být mapa sice užitečná pro teoretické bádání pracující s hypotetickými daty, ale jako zástupce skutečné dopravní struktury se stává zkreslenou, a pro praktické účely sloužící právě k navigaci ve skutečných dopravních strukturách se stává méně vhodnou. (Dále je také nutno brát v potaz, že různé úlohy mohou

klást různé požadavky na přesnost dat, a i nepřesná data tedy mohou být v určitých případech užitečná.)

V případě hledání optimální trasy by se neměl vyskytnout problém ohledně proveditelnosti úkonu, neboť ten je, jak již bylo zmíněno, známý a v praxi dokonce častý.[3, s. 281] Ohled je však nutno brát na to, aby byl algoritmus uzpůsoben údajům, potažmo údaje algoritmu. Při velkém počtu cest (a tudíž i hran v grafu) se totiž problém stává výpočetně náročnějším, a může být tedy na místě náročnost snížit, nebo zvolit výkonnější technické vybavení, které dosáhne výsledku v uspokojivém čase.

Dále si pak velkou pozornost si zaslouží právě dostupnost daného řešení. Ta totiž může být podmíněna různými faktory a okolnostmi, z nichž některé zde byly již zmíněny. Je kupříkladu zvláště příhodné, aby zvolené řešení bylo finančně dostupné, potažmo v tom nejlepším případě zcela zdarma. Řešení by také nemělo být časově omezené, nýbrž trvale funkční; jinými slovy by nemělo docházet k tomu, že v jeden čas bude využitelné, a v jiný čas nikoliv. S oběma problémy se lze hypoteticky setkat např. u řešení poskytovaná formou služeb (viz níže), kdy dostupnost uživateli zcela závisí na vůli a momentální schopnosti poskytovatele službu poskytovat, a na jeho uvážení, zda za službu požadovat náhradu, či nikoliv.

Mezi další možné překážky pak lze řadit třeba právě onu zmíněnou výpočetní náročnost. V případě, že si výpočet optimální trasy žádá mnoho času či jiných prostředků (kterými mohou být např. místo v paměti nebo na pevném disku), řešení se stává méně použitelným, ba dokonce i nepoužitelným na strojích, které jeho požadavky nesplňují. Je tedy v zájmu uživatele (a úspěšnosti celého řešení), aby zvolené řešení bylo výpočetně nenáročné, a proveditelné nejlépe v jakémkoliv čase, bez ohledu na vnější vlivy a okolnosti.

3.5 Dostupná řešení a částečná řešení

Během posledních dvou až tří desetiletí docházelo ve světe k určitému přerodu zeměpisných informací přístupných veřejnosti, a šířeji také zeměpisných informačních systémů. Z dříve poněkud uzavřeného odvětví, které zaznamenávalo aktivitu především od odborníků v oboru, vedenou v prvních řadách za odbornými účely, se s postupem času a výskytem nových technologií stalo odvětví otevřené širší veřejnosti[13].

Kolem začátku 21. století se začaly objevovat webová rozhraní a webové aplikace poskytující přístup k detailním číslicovým mapám, a zároveň usnadňující práci s nimi. To vedlo k velmi rychlému nárůstu jejich popularity, a velkému rozšíření číslicových map (a tedy i jejich využívání a zájmu o ně) mezi dříve netknutými skupinami. Tento „novozeměpis“ (*néogéographie*)[14], jak je tomuto fenoménu některými autory přezdíváno, tedy vedl k zakotvení svých hlavních činitelů jako významných zdrojů zeměpisných informací na světovém poli. Jedná se kupř. o služby jakými jsou *Google Maps*, *Bing Maps* (provozované společností *Microsoft*), či projekt *OpenStreetMap*. Jako místní pak lze uvést např. *Mapy.cz* (provozované společností *Seznam*).

Tyto entity lze považovat za možná částečná řešení zkoumané problematiky, neboť poskytují přístup k číslicovým zeměpisným údajům (ať už přímo či skrze vlastní nástroje), a nástroje k jejich zpracování, přičemž pro úspěšné řešení je požadován právě nástroj pro nalezení optimální trasy mezi dvěma lokacemi. Jsou to tedy vhodní kandidáti k bližšímu zkoumání při důkladném zvážení všech třech výše uvedených předpokladů.

3.5.1 Google Maps

Mapy od společnosti *Google* lze považovat za jeden z větších veřejně přístupných zdrojů číslicových map, jejichž kvalita by se dala považovat za uspokojivě vysokou, a jejichž forma je vhodně uzpůsobena hledání tras (což je zjevné z přidružených služeb, které v nich trasy velmi zdařile hledají[15]).

Za hlavní nedostatek tohoto řešení však lze považovat právě omezenou dostupnost. Přístup k nástrojům přes programovací rozhraní (dále jen API) webové služby již ze své podstaty vyžaduje přístup k internetu, znemožňuje tak jeho využívání bez internetového připojení. Některé nástroje jsou navíc zpoplatněné, což se týká i potřebného nástroje na hledání optimálních cest. (V současné době se cena pohybuje na pěti amerických dolarech za tisíc požadavků.) Zajímavou sice může být přítomnost nástroje na spočtení matice vzdáleností, který by celý proces vnitřně velmi zjednodušil, ale i ten je zpoplatněný[16].

Finanční kompenzace společnosti *Google* by nadto mohla být vnímána o to hůře, jelikož je společnost z etického či morálního hlediska (především pak v kontextu technologických otázek a záležitostí s ním spjatých, soukromí člověka nevyjímaje) některými hodnocena velmi kriticky,[17] a i z tohoto důvodu tedy může být schůdnější se jí vyvarovat.

3.5.2 Bing Maps

Mapy od společnosti *Microsoft* jsou na tom obdobně jako mapy od společnosti *Google*; je jistě možné srovnávat jejich kvalitu detailněji, a případně v ní najít významné rozdíly; podstatný problém nepřístupnosti nástrojů potřebných k hledání optimálních tras (a nemožnost jejich hledání právě bez daných nástrojů) však přetrvává[18] rovněž i zde. Z této příčiny je tedy přednost ke zvážení přenechána jiným částečným řešením.

3.5.3 Mapy.cz

Od služby *Mapy.cz*, provozovanou společností *Seznam*, by bylo možné očekávat vyšší kvalitu dat alespoň pro oblast České republiky, neboť (jak je už z názvu služby patrné) právě v České republice má služba své kořeny i zaměření. Toto hypotetické očekávání se alespoň v minulosti ukazovalo jako oprávněné, kdy v porovnání s *Google Maps* byly nalezeny významné rozdíly[11] právě ve prospěch služby *Mapy.cz*. Nutno však dodat, že podle aktuálních vyjádření společnosti využívá[19] služba pro mapy mimo Českou republiku právě datový soubor *OpenStreetMap*.

Služba sice dává k dispozici bezplatné rozhraní, ale jen pro omezený výčet funkcí; funkce

hledání optimálních tras není v současné době zahrnuta[20] ve stabilní bezplatné verzi (pouze ve vyvíjené placené). Z tohoto hlediska tedy není pro účely této práce ideální volbou.

3.5.4 OpenStreetMap

Projekt *OpenStreetMap* se od ostatních odlišuje zejména po stránce přístupové. Nejedná se totiž primárně o službu (poskytovanou za účelem výdělku), nýbrž o projekt vzniklý za účelem vytvoření souboru zeměpisných údajů celé Země. Projekt na té nejzákladnější úrovni funguje na principu dobrovolnosti; je otevřený příspěvkům a spolupráci nadšenců a široké veřejnosti[21], přičemž údaje, kterými je do souboru přispíváno, jsou kladeny pod svobodnou licenci (konkrétně se jedná *Open Database License*, dále jen ODbL) a volně šířeny. ODbL dává uživatelům daného souboru (mimo jiné) možnost údaje v něm obsažené svobodně využívat, šířit, a upravovat[22]. Pro účely této práce je stěžejní právě bezplatná dostupnost dat a možnost s nimi pracovat dle vlastního uvážení.

Co se kvality shromážděných údajů týče, vyvození jednoznačných závěrů se jeví jako nemožné, vzhledem k nedostatku dostatečně úplných a snadno veřejně dohledatelných odborných posudků, zejména pak v oblasti České republiky. Na základě několika zkoumání zveřejněných do roku 2013 (mimo jiné v Anglii, Francii, Německu a Spojených státech) lze říci, že údaje z projektu OpenStreetMap se vyznačují určitou nesourodostí (tzn. že v různých oblastech se kvalita údajů různí, a to např. vyšší kvalitou údajů ve městech v Evropě, kdežto naopak srovnatelně vyšší kvalitou na venkově v Americe, alespoň v rámci zkoumaných míst); obsahují nezanedbatelný počet nedostatků, a jsou tedy do určité míry chybové, kterážto míra se zdá s přibývajícím časem a přibývajícími příspěvky do souboru klesat[23]. Za pozitivní činitel lze považovat i existenci aktivní české komunity v podobě zapsaného spolku *OpenStreetMap Česká republika*[24].

Je však na místě nebrat v potaz jen zkoumání vedená čistě za účelem posouzení kvality polohopisných údajů, nýbrž vzít v něj také jejich dosavadní využití v praxi. Údaje obsažené v souboru *OpenStreetMap* jsou totiž hojně využívány velkým množstvím služeb, aplikací, a organizací. K samotným údajům poskytuje přístup hned několik služeb; mezi nimi lze jmenovat *Geofabrik*, *Overpass*, nebo *Mapzen*.

Projekt *OpenStreetMap* byl též využíván v řadě akcí a prací v oblasti výzkumu a vzdělávání, jmenovat lze kupř. akce *Mapping and the Citizen Sensor* a *ENERGIC*, či soubor vzdělávacích prostředků *TeachOSM*. Nezisková organizace *Humanitarian OpenStreetMap Team* organizuje kolektivní mapování při živelných pohromách; projekt *The Missing Maps* (založený mimo jiné humanitárními organizacemi jako jsou *Červený kříž* nebo *Doktoři bez hranic*) se zase zasazuje o zmapování zranitelných míst v nevyspělých částech světa. Mezi významné společnosti stavící na projektu OpenStreetMap patří již zmíněné *Geofabrik* a *Mapzen*, ale také *Mapbox*, *MapQuest*, *Stamen*, nebo *CampToCamp*. Mezi uživatele náleží i vládní organizace některých států.[25]

I přes nemožnost objektivně a úplně posoudit kvalitu a úplnost obsažených údajů lze tedy brát dosavadní výsledky zkoumání a důvěru poměrně velkého počtu uskupení jako určitou záruku důvěryhodnosti projektu, a především jeho vhodnosti k praktickému využívání.

Lze tedy předpokládat, že kvalita i dostupnost číslicových údajů jsou pro účely této práce na dostatečně dobré úrovni. Posledním předpokladem, který je třeba splnit, zůstává tedy nalezitelnost optimální trasy. K tomu jsou údaje naštěstí svou strukturou uzpůsobeny[26], a k tomuto úkonu existují i volně dostupné nástroje[25], jejichž průzkum a následné vyhodnocení tvoří jednu z částí vlastní práce.

4 Vlastní práce

V praktické části práce byl vytvořen program *TSPbrousek*, který převádí seznam lokací na matici sazeb. Při jeho tvorbě byly použity nástroje zvolené dle své způsobilosti k dosažení stanovených cílů, a popř. i dalších kritérií stanovených níže. Program je k práci přiložen pouze elektronicky.

4.1 Volba knihovny grafického uživatelského rozhraní

Jednou z překážek pro vývoj multi-platformních programů je nesourodost technologií využívanými či podporovanými jednotlivými platformami. Její podstatou je fakt, že každá z platforem má své vlastní nástroje k tvorbě a zobrazování grafických rozhraní, a nelze tedy pracovat se všemi z nich za pomoci jednotného kódu.

Pro vytvoření jednotné grafické multi-platformní aplikace (tzn. takové, která bude stejně fungovat na různých platformách bez výrazných změn či obměn zdrojového kódu) může být vhodné využití knihovny, která tuto funkci zprostředkuje; tedy takové, která dává k dispozici vlastní sadu ovládacích prvků, které budou všude vypadat a fungovat obdobně, přičemž o jejich překlad do forem srozumitelným jednotlivým platformám se postará sama knihovna.

V rámci práce bylo zváženo několik dostupných knihoven s oficiální podporou systémů Windows a GNU/Linux. Nejdůležitějšími kritérii byly (v pořadí od nejdůležitějšího po nejméně důležité):

1. Snadná dostupnost a využitelnost (opět především z právního a spořivého hlediska).
2. Vnímaná robustnost a celková spolehlivost.
3. Osobní preference, vzhledem k:
    - Struktuře a návrhu jednotlivých knihoven.
    - Jejich podpoře v již známých programovacích jazycích.

4.1.1 Žádané vlastnosti užitého programovacího jazyka

Při zvažování dostupných programovacích jazyků (a na ně vázaných nástrojů) pro práci s danou knihovnou lze kromě vlastních návyků a osobního dojmu přihlédnout také ke způsobilosti jazyka (a jeho vázání k té které knihovně) k vytváření a následné distribuci spustitelných verzí programu. Toto hledisko odkazuje mimo jiné na možné rozdělení programovacích jazyků na jazyky *překládané* a jazyky *kompilované*.

Jazyky interpretované (např. *Lua*, *Perl*, *Python*) se zpravidla spouštějí za pomoci zvláštního programu, který je s to instrukce v jazyce zapsané přečíst, pochopit (tedy „interpretovat“), a následně vykonat. Nevýhodou tohoto přístupu je právě potřeba onoho zvláštního programu. V případě využití takového jazyka by tedy bylo zapotřebí spolu s programem hlavním šířit i takovýto program vedlejší, a instruovat uživatele k jejich instalaci a spuštění (nebo takový

postup nějak automatizovat). Takový přístup by sice mohl být přijatelný, ale přesto poněkud nežádoucí, neboť by mohl komplikovat užívání programu, což by se protivilo stanoveným cílům této práce.

Na druhé straně *jazyky kompilované* (např. *Ada*, *C*, *C#*) se překládají pro určitý systém z daného programovacího jazyka ještě před spuštěním, a to zpravidla do instrukcí ve formě binárního kódu. Daný systém by pak měl být schopen takto přeložené a předložené instrukce interpretovat sám, bez zvláštně dodaného pomocného programu. I takto přeložený program může případně vyžadovat ke spuštění přítomnost určitých knihoven, na kterých jeho funkce závisí, ale i v tom případě mohou být jeho instalace a spuštění o poznání méně komplikované. Z hlediska přívětivosti programu lze tedy pro tento typ programu obecně považovat kompilované jazyky za vhodnější (alespoň co se jednoduchosti implementace týče).

Poznámka 2. Toto rozdělení není nutně vrozené ani absolutní, a je zde prezentováno ve zjednodušené formě, pouze pro objasnění výběrového procesu, ve kterém jsou brány v potaz nástroje, za pomoci kterých se dané jazyky běžně používají.

S přihlédnutím také k vlastním preferencím a zkušenostem byla tedy upřednostňována řešení dostupná pro jazyky *C*, *C++*, *C#*, a *Scheme*. (Konkrétně se jedná o implementaci *Chicken Scheme*, která umožňuje kompilaci kódu v jazyce *Scheme*, konkrétně do kódu jazyka *C*,[27] třebaže bývá tento jazyk běžně interpretován.)

4.1.2 IUP

Knihovna IUP[28] je multi-platformní sadou nástrojů pro vývoj grafických aplikací. Nabízí rozhraní v jazycích *C* a *Lua*, jako i ve vlastním jazyce *LED*. Knihovna je dlouhodobě vyvíjena a aktualizována, a je dostupná pod svobodnou permisivní licencí *MIT/Expat*. Podporovanými systémy jsou jen Windows a GNU/Linux (kde funguje prostřednictvím dalších knihoven, jmenovitě *GTK* nebo *Motif*), což je postačující.

Tato knihovna však není v porovnání s ostatními hojně využívána, a její dostupnost v jednotlivých distribucích GNU/Linuxu je také na horší úrovni[29]. Zajímavými byly shledány vazby[30] k jazyku *Scheme*, které jsou však neoficiální a ne zcela hladce zprovoznitelné; přednost tedy byla dána knihovnám jiným.

4.1.3 WebView

Další knihovnou, která byla zvažována, je *WebView*. Ta funguje jen jako pojící prvek mezi prohlížečem webových stránek (který je vestavěný do nástrojových sad užívaných každou z cílených platforem, mezi které patří i *macOS X*), a celé grafické rozhraní je tedy vyvíjeno jako webová stránka[31] (která ovšem může být dostupná off-line, tedy lokálně). I v případě této knihovny je dostupná upravená verze[32] pro jazyk *Scheme*. Knihovna je svými technickými vlastnostmi vyhovující, avšak pro určitou neobvyklost a ne zcela hladký průběh jejího

testování byl upřednostněn následující kandidát.

4.1.4 Qt

Jednou z šířeji využívaných sadou nástrojů pro tvorbu multi-platformních grafických rozhraní je bezesporu *Qt*[33]. Příkladem jejího využití může být hned celé desktopové prostředí *KDE*[34] (využívané zejména v různých distribucích operačního systému GNU/Linux), mobilní operační systém *Ubuntu Touch*[35] (dříve vyvíjený britskou společností *Canonical*, dnes dobrovolnickou komunitou), nebo multi-platformní multimediální přehrávač *VLC*[36].

Hlavní poznanou výhodou knihoven *Qt* je právě jejich vyspělost a rozšířenost; v porovnání s ostatními, méně často užívanými (a snad by se dalo říct „exotičtějšími“) knihovnami, je pro knihovny *Qt* dostupný větší počet aplikací a neoficiální dokumentace (tedy návodů, veřejných diskuzí, a jinak popsaných řešení dílčích problémů knihoven se týkajících), díky kterým lze nejrůznější překážky vzcházející během vývoje mnohem snáze překlenout.

Další pozoruhodnou vlastností knihovny je skutečnost, že je vyvíjena komerčně. To samo o sobě nelze vnímat jako naprostou záruku kvality, ale ve spojení s její dlouhodobou existencí a využíváním v praxi to bylo vnímáno jako pozitivum. V tomto kontextu je vhodné zmínit, že knihovny jsou dostupné jednak pod komerčními licencemi, tedy za úplatu, ale zároveň s tím je většina z nich dostupná také pod tzv. *weak copyleft* licencí *GNU LGPLv3*; ta umožňuje jejich bezplatné využívání i v nesvobodných aplikacích, ale pouze za předpokladu, že text a licence samotných knihoven budou zachovány, aby z nich případně mohli těžit i koncový uživatelé. To pro tuto práci nebylo vnímáno jako problém.

Poslední významnou výhodou je obsáhlost knihoven, které zahrnují i knihovnu *Qt Network*[37] pro komunikaci se vzdálenými servery, což sytí potřebu vyhledávat k tomuto účelu jiné knihovny. Po zvážení všech zmíněných vlastností, a nepříliš hladké práci s ostatními zmíněnými knihovnami, byla pro vytvoření programu použita právě sada knihoven *Qt*.

4.2 Převodníky

Samotný převod lokací je v programu realizován pomocí „převodníků“. Jedná se o abstrakci, resp. *souhrnné pojmenování* dostupných nástrojů (a to jak v podobě místních programů, tak v podobě webových služeb), které jsou s to vyhledat optimální trasu mezi lokacemi, a lze tedy pomocí nich *převádět* souřadnice na sazby. Název byl zvolen tak, aby byl pokud možno výstižný (a to více než třeba poněkud obecný „nástroj“), a jasně komunikoval účel nástrojů v rámci programu i těm uživatelům, kteří nejsou uvyklí technologické hantýrce (což se neočekává od termínů jako je třeba „knihovna“).

Služeb a knihoven, které pracují s údaji projektu *OpenStreetMap*, je sice poměrně mnoho[38], ale jen některé jsou svými parametry zvláště vhodné k účelům této práce. Právě takové byly – mimo jiné pro svou dostupnost a snadnou integraci v programu – využity k vytvoření celkem tří převodníků. Ty jsou podrobněji popsány níže.

4.2.1 OpenRouteService

OpenRouteService[39] je, jak již její název napovídá, webová služba, kterou provozuje *Heidelberg Institut for Geoinformation Technology* při *Ruprechto-Karlově univerzitě v Heidelbergu*. Veřejnosti je zpřístupněna vcelku snadno užitelným aplikačním rozhraním, které mimo jiné umožňuje přímo i výpočet matice. Zejména toho důvodu byla vyhodnocena velmi vhodná pro využití v této práci. Výstupem jsou trasy pokládané za nejrychlejší [40].

Přístup k API je však omezen kvótou, a podmíněn využitím unikátního klíče. Pro nabytí klíče je třeba registrace, která je bezplatná. Pro účely této práce byl tedy získán klíč k API, a s uvážením skutečnosti, že kvóta by mohla být častým vypočítáváním velkých matic naplněna, není tato služba ve výsledném programu prezentována jako výchozí převodník, čímž by k jejímu užívání mělo docházet méně často, a kvóta by tak měla být naplněna později, nebo by neměla být naplněna vůbec.

4.2.2 Open Source Routing Machine

Open Source Routing Machine (dále jen *OSRM*) je programem vytvořeným pro rychlé nacházení optimálních tras mezi lokacemi[41], napsaným v jazyce *C++*. Je šířen pod velmi permisivní licencí *BSD-2-Clause License*[42]. Z dostupných zdrojů a tvrzení samotných autorů vyplývá, že program je schopen hledat cesty velmi rychle[43], avšak jeho výpočetní náročnost co se paměti týče je dosti vysoká[44, 45]. Z toho důvodu byla pro místní výpočty dána přednost knihovnám s nižšími výkonnostními požadavky.

Vedle samotného programu pro hledání tras je však dostupná i webová služba, která funkce programu zpřístupňuje dálkově, a je s to poskytnout uživateli sazby na požádání. Pro nenáročnost a zároveň jednoduchost implementace ve srovnání s místním využitím knihovny byl přednostně zvolen právě tento postup. Převodník tedy vrací matici sazeb pro nejrychlejší trasy mezi lokacemi[46].

4.2.3 RoutingKit

RoutingKit je knihovnou taktéž vyvinutou v jazyce *C++*, tentokrát týmem prof. Dorothey WAGNEROVÉ v Institutu technologie v Karlsruhe.

> *„Hlavním účelem vývoje je zprostředkování plodů čerstvého výzkumu lidem vyvíjejícím aplikace pro plánování tras. Klíčovým prvkem je rozhraní, které nabízí dobrý kompromis mezi výkonem a snadným užíváním.“*[47, volně přeloženo]

Výrazným prvkem knihovny je také využití datové struktury přizpůsobitelných kontrakčních hierarchií (*customisable contraction hierarchies*), které dále zrychlují hledání nejkratších cest grafem[48] v porovnání s běžnými kontrakčními hierarchiemi[49]. Vítanou vlastností knihovny (která bezpochyby svědčí o naplnění cíle snadného užívání) je pak její schopnost

pracovat přímo se soubory ve formátu *PBF*, bez nutnosti mapové údaje zvlášť překládat do jiného formátu. Tato vlastnost, společně se snadnou implementací podpory, činí z knihovny velice vhodného kandidáta pro místní (tedy off-line) hledání tras, a proto byla do programu začleněna.

4.3 Grafické uživatelské rozhraní

Po zvážení a zvolení dostupných nástrojů k dosažení stanovených cílů bylo možné postoupit k samotnému návrhu uživatelského grafického rozhraní. Teprve tehdy totiž nabyl pracovní postup při používání programu jasnějších obrysů, ze kterých bylo možné patřičný návrh odvodit.

Celý proces opakovaného užívání programu (pomíjeje tedy povětšinou jednorázový krok seznamování s programem) je poměrně jednoduchý a přímočarý; lze jej rozdělit do tří kroků:

1. Zadávání lokací do programu.
2. Volba vhodného převodníku.
   - S tímto krokem mohou být spojeny další, dílčí kroky, podle toho, které úkony si daný převodník žádá. (Může se jednat např. o výběr souboru s mapovými údaji, popř. o zadání klíče k API.) Tyto se však typicky týkají úkonů vně programu samotného, a jejich zásah do práce s programem je tedy minimální, a do rozhraní snadno začlenitelný bez výrazné změny jeho struktury.
3. Nabytí a následné vytažení matice sazeb.

Následně lze tedy i celé grafické rozhraní rozčlenit do tří na sebe navazujících částí, z nichž každá bude zaměřena právě na jeden dílčí krok. V případě, že složitost jednotlivých částí dovolí tak učinit, bylo by velmi příhodné zobrazit všechny části pohromadě, jednu vedle druhé. Takové rozvržení nabízí nadhled nad celým procesem, a zároveň umožňuje uživateli se v něm volně pohybovat; nedochází tedy ke změnám mezi vícero kontexty, neboť celá aplikace prezentuje jen jeden ucelený.

4.3.1 Zadávání lokací

Cílem práce je nabídnout co nejjednodušší rozhraní, celkově i pro dílčí části, tedy i pro zadávání lokací. V lidské řeči (alespoň co se jazyka českého týče) se lokace (ať už přesné, jako třeba ulice, nebo nepřesné, jako třeba města) zpravidla nazývají krátkými jmény (např. *Lomnice nad Popelkou*), pro upřesnění pak celými adresami, jejichž jsou taková jména zpravidla součástí (např. *Lomnice nad Popelkou, Liberecký kraj, Česká republika*).

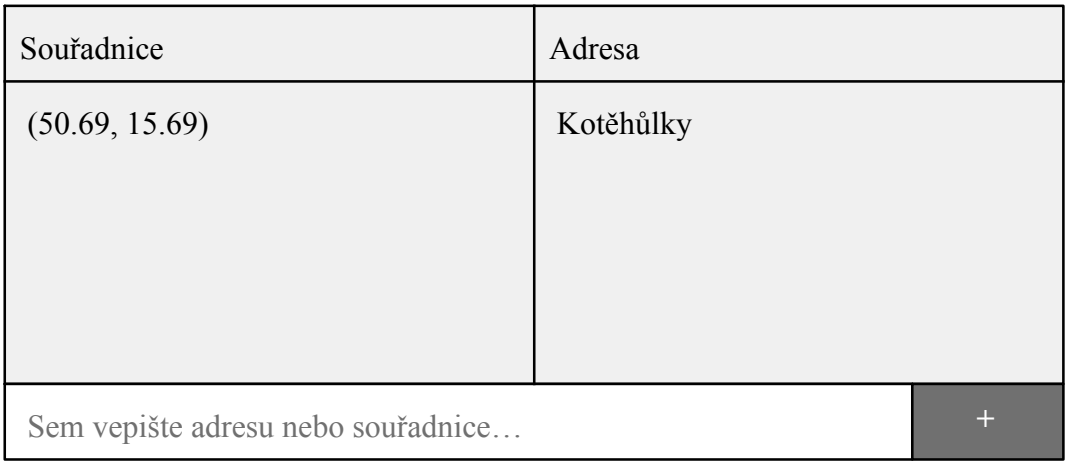
Naproti tomu v užitých nástrojích jsou hlavním označením lokací souřadnice, které neoznačují místa z hlediska lidského vnímání, nýbrž z hlediska zeměpisné polohy s určitou přesností. Výhodou souřadnic je jejich jednoznačnost: dostatečně přesné souřadnice odkazují právě na jedno místo na mapě, a na žádné jiné odkazovat nemohou. (V případě nepřesných

souřadnic se pak jedná o místo přibližné.) Naproti tomu jména užívaná v běžné řeči mohou být mnohoznačná, zvláště jsou-li pak nepřesná. Příkladem může být název obce *Nová Ves*, kterých se v České republice objevuje hned 62[50]. Nejjednodušším způsobem, jak docílit jednoznačného označení lokace je tedy použití souřadnic, které jsou tedy jak pro správnou funkci programu, tak pro jednoduchost implementace vhodným řešením.

Nevýhodou souřadnic je však právě jejich technická povaha, resp. neobvyklost v mluvené řeči, a tedy obecný nezvyk označovat jimi lokace. To v případě jejich využití může být překážkou právě pro zadávání lokací, neboť vzniká potřeba nejprve převést označení lokací z tvaru známého a obvyklého (tedy jména) do tvaru neznámého a neobvyklého (tedy souřadnice).

Pro odstranění (nebo alespoň zmenšení) této nevýhody je v programu implementováno jak přímé zadávání souřadnic (se kterými program pracuje primárně), tak i adres lokací, které jsou na souřadnice při zadání převáděny pomocí webové služby *OpenRouteService*. Výsledný návrh příslušné části viz níže (obrázek 1).

Obrázek 1. Návrh části k zadávání lokací.



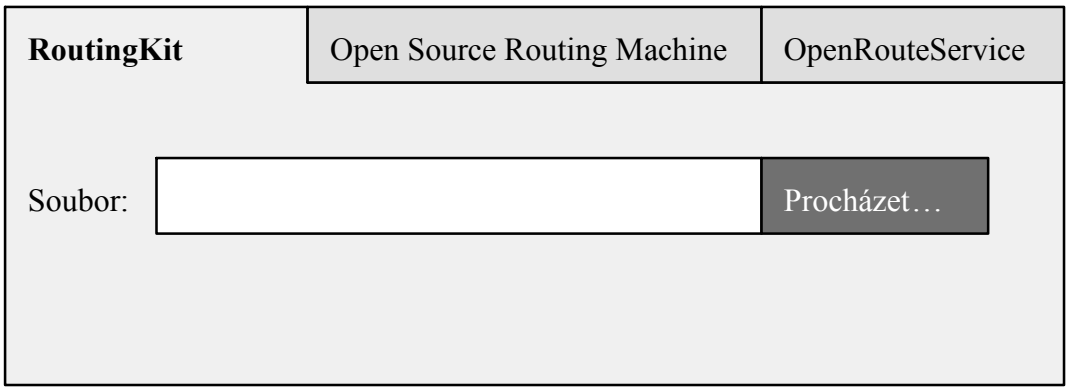
Zdroj: Vlastní kresba.

4.3.2 Výběr převodníku

Jak již bylo nastíněno, k dispozici jsou celkem tři převodníky, mezi kterými si uživatel může vybrat. Asi zcela nejjednodušším způsobem výběru by bylo použití přepínače, kde by uživatel pouze označil žádaný převodník v seznamu.

Jednotlivé převodníky však disponují různými parametry, kterými lze funkci uzpůsobit svým potřebám. Zobrazení nastavení všech převodníků najednou by zabralo mnoho místa, a mohlo by nadto působit poněkud přehlceně. Místo přepínače jsou tedy převodníky uživateli prezentovány ve formě panelů (resp. karet), pod kterými se vždy zobrazí nastavení právě vybraného panelu (a tedy jím představovaného převodníku). Návrh části viz níže (obrázek 2).

Obrázek 2. Návrh části výběru převodníku.



Zdroj: Vlastní kresba.

RoutingKit

Nastavení převodníku *RoutingKit* je velmi prosté: převodník vyžaduje pouze soubor ve formátu *PBF*, ze kterého by mohl číst údaje. Žádná cesta k souboru v programu není přednastavena, neboť program takový soubor neobsahuje, a nemůže tedy předpokládat jeho přítomnost. Nelze totiž s jistotou předvídat, jaké údaje (resp. jaké mapy) bude uživatel přesně potřebovat, a šíření velkých a nadbytečných souborů spolu s programem by jeho sdílení zbytečně ztížilo (i nehledě na to, že takto přiložené mapy by zanedlouho již nebyly aktuální). Návod k obsluze (jako i nápověda v samotném programu) tedy dává uživateli k dispozici alespoň odkazy na zdroje, ze kterých je údaje možné čerpat.

Open Source Routing Machine

Nastavení převodníku *OSRM* je taktéž velmi prosté: převodník vyžaduje pouze odkaz na webovou službu, kterou bude žádat o výpočty sazeb. Přednastaven je odkaz na již zmíněnou službu, a uživatel tedy nemusí ke zprovoznění převodníku nic měnit. V případě, že by se však někdy změnila adresa služby, nebo by uživatel chtěl využít alternativní (případně vlastní) služby využívající stejný software, má možnost zadat adresu jinou.

OpenRouteService

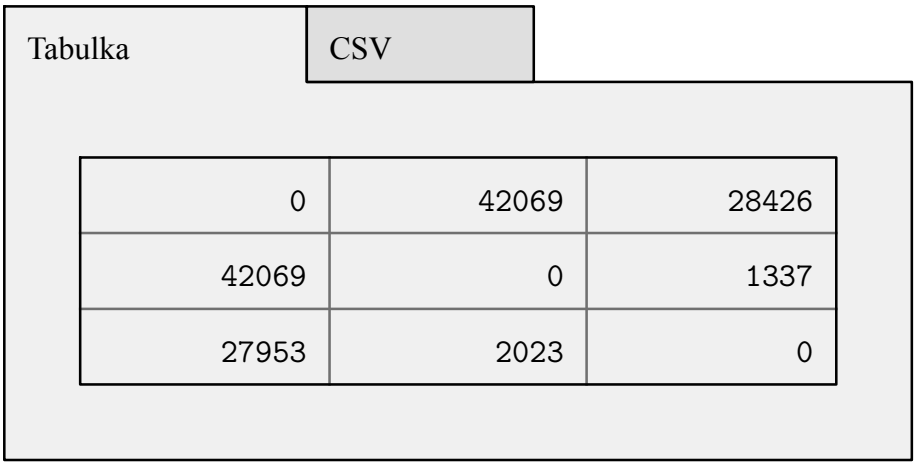
Tento převodník má nejvíce parametrů. Stejně jako u služby *OSRM*, je i zde možné zadat jinou adresu, než je ta přednastavená. Stejně tak je možné i zadat jiný klíč k API dané služby. Předpokládá se, že k tomuto kroku se většina uživatelů nebude uchylovat, ale pokud by došlo k naplnění kvóty přednastaveného klíče k API, a uživatel by přesto chtěl využít tohoto převodníku, má možnost se dle instrukcí obsažených v návodu u služby sám zaregistrovat, a získat tak klíč vlastní.

Právě tento převodník je navíc implicitně využíván k převodu adres na souřadnice. Pro úplnost je tedy zahrnuta také možnost tuto funkci vypnout, počemž bude možné zadávat adresy pouze ve formě souřadnic.

4.3.3 Zobrazení matice sazeb

Neboť celý výpočet probíhá automaticky, a uživatel do něj v jeho průběhu nikterak nezasahuje, může být dalším krokem hned zobrazení výstupu programu, tedy samotné matice sazeb. Sazby jsou udávány v metrech, a samotná matice na sebe bere podobu tabulky, jelikož v takové formě bude také následně zpracovávána. Návrh výstupu viz níže (obrázek 3).

Obrázek 3. Návrh části výstupu ve tvaru matice.



Zdroj: Vlastní kresba.

Kromě úhledně vyhlížející tabulky je uživateli předložen také alternativní panel, ve kterém je matice dostupná ve snadno kopírovatelném formátu *CSV*.

4.3.4 Ostatní části grafického rozhraní

Grafické rozhraní programu zahrnuje i jiné části, a to konkrétně jiná okna, která se zobrazí při neobvyklých situacích, kupř. dojde-li k chybě, nebo bude-li třeba upřesnit vstupní údaje programu. Tyto části grafického rozhraní zde nejsou podrobněji rozváděny, neboť je každá z nich zaměřena právě na jeden samostatný prvek, a nejsou součástí běžného provozu; jejich uspořádání nebo začlenění do celku tedy nehraje významnou roli.

4.4 Vlastní implementace

Ve zdrojovém kódu programu jsou převodníky zobecněny abstraktní třídou `Router`, kde jsou definovány základní funkce zprostředkovávající převod. V abstraktní podtřídě `NetRouter` jsou pak definovány společné funkce pro převodníky, které využívají ke svému fungování webové služby.

Zvláštní abstraktní třídou je `NetAddresser`, která slouží k převodu názvů na souřadnice. Ten, jak název třídy napovídá, je implementován požadavkem webové službě. Třída využívá vlastní instanci třídy `Downloader` k přenosu dat, stejně jako třída `NetRouter`.

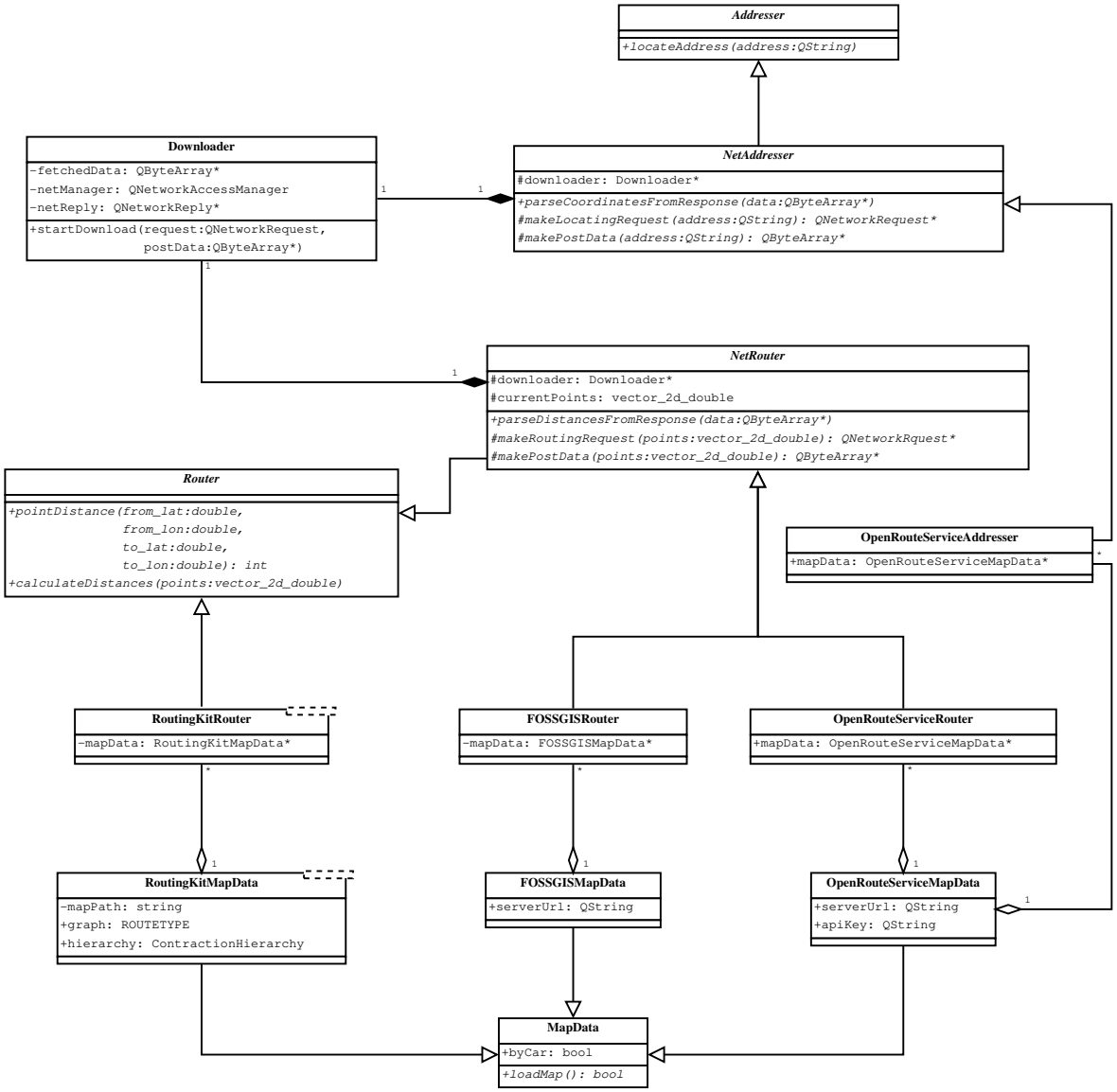
Každý převodník je pro účely převodu lokací implementován vlastní třídou (třídami), z které teprve vznikne instance. Třída `FOSSGISRouter` implementuje komunikaci s *OSRM*,

zatímco třída `OpenRouteServiceRouter` komunikaci s *OpenRouteService*. Obě třídy jsou potomky `NetRouter`, neboť potřebují komunikovat se vzdáleným serverem. Pro účely převodu adres na souřadnice je však implementována třída `OpenRouteServiceAddresser`, spolu s nadtřídami určenými k přehlednější organizaci a možnému rozšíření programu.

Třída `OpenRouteServiceRouter` navíc dědí i z třídy `NetAddresser`, protože jako jediná umožňuje překlad adres na souřadnice. Ostatní převodníky to neumožňují, ale pro budoucí rozšíření je dostupná i nadtřída `Addresser` pro převod lokální.

Zdrojový kód programu je dostupný ve formě elektronické přílohy, a je i řádně okomentován (v angličtině, pro zachování jazykové jednoty mezi komentáři a kódem samotným). Pro přehlednější znázornění všech vztahů viz lehce zjednodušený diagram v grafickém jazyce UML níže (obrázek 4).

Obrázek 4. UML diagram hlavní třídní struktury programu.



Zdroj: Vlastní kresba.

5 Zhodnocení výsledků

Konečným výsledkem této práce je počítačový program *TSPbrousek*, jehož výstupem je matice sazeb pro problém obchodního cestujícího, vypočítaná v metrech. Matice je dostupná ve formátu *CSV*, který je jednoduchý a srozumitelný jak pro člověka, tak i pro různé tabulkové procesory (jakými jsou *Microsoft Excel*, *LibreOffice Calc*, nebo *Gnumeric*)[51, 52, 53], a vyhovuje tedy také programu *TSPkosa*.

Program lze bez dodatečných úprav zdrojového kódu (který je jednotlivým cílovým systémům již uzpůsoben) spustit na operačních systémech *Windows* a *GNU/Linux*, a na obou z nich funguje dle očekávání; za multi-platformní jej tedy považovat lze.

Program je vybaven jednoduchým grafickým rozhraním, které je sestaveno z běžných a všeobecně známých prvků, a tak by nemělo představovat překážku ani pro technicky nezdatné uživatele. Zjevnost se nepředpokládá pouze u části určené k výběru převodníku, kde by se však – vzhledem ke specifičnosti dané části – ani dosti dobře předpokládat nedala. Práce s ní je tedy alespoň objasněna dokumentací v návodu k programu, a také zjednodušena vhodně zvoleným výchozím nastavením.

Zadávání lokací je možné jak souřadnicemi, tak i vlastními názvy lokací, přičemž byla vyvinuta snaha zamezit jejich nechtěné záměně s nositeli stejných názvů. Tento účel program alespoň pro zkoumané lokace splňuje. Jeho fungování se tedy zdá být v souladu se stanovenými cíli.

5.1 Spolehlivost a trvanlivost

Zvýšenou pozornost si zaslouží otázka, jestli program bude s to úlohu řešit spolehlivě a dlouhodobě. Jelikož byla při vývoji poznána nutnost jakéhosi kompromisu mezi dlouhodobou spolehlivostí, kterou poskytuje využití lokálních dat a nástrojů, a rychlostí výpočtu spolu s nízkými technickými požadavky, které splňuje využití externích služeb, byla zahrnuta obě řešení.

Dostupny jsou tedy dvě externí služby: *OSRM* a *OpenRouteService*. Služba *OSRM* nestanovuje žádná konkrétní omezení na počet požadavků (pouze na jejich frekvenci), a vzhledem k povaze programu se tedy nepředpokládá, že by tedy programu náhle přestala odpovídat z důvodu provádění mnoha výpočtů. Nelze však vyloučit možnost, že by taková omezení byla někdy zavedena, nebo že služba přestane být (ať už dočasně nebo trvale) dostupná, a z toho důvodu je k dispozici také alternativní služba *OpenRouteService*. Ta je zároveň jedinou, kterou program využívá k převodu adres na matice. Tato služba je omezena přístupovým klíčem, jehož kvóta může být naplněna[54], ale pro takový případ jsou uživateli dodány instrukce k nabytí vlastního klíče.

Jako lokální nástroj je pak dána k dispozici knihovna *RoutingKit*, jejíž fungování se ukazuje jako pomalejší, ale která s sebou naproti tomu ne nese vůbec žádná umělá omezení. Po

získání mapy, kterou je daný počítač schopen zpracovat, je tento převodník možno využívat zcela neomezeně.

5.2 Porovnání výstupu převodníků

Pro ověření funkčnosti a srovnání odlišností mezi sazbami jednotlivých převodníků byla provedena jednoduchá zkouška, při které byla spočítána matice sazeb pro 6 obcí v České republice při cestě autem; jejich sazby jsou dány v metrech:

1. Hřibojedy, Královéhradecký kraj (50,39206; 15,83473)
2. Máslojedy, Královéhradecký kraj (50,29693; 15,75882)
3. Mlékojedy, Středočeský kraj (50,25941; 14,53244)
4. Kozojedy, Královéhradecký kraj (50,31643; 15,37478)
5. Srnojedy, Pardubický kraj (50,03911; 15,70262)
6. Masojedy, Středočeský kraj (50,02598; 14,77788)

Výpočet převodníku *RoutingKit* trval 2 minuty a 20 sekund, zabíraje k tomu nejvýše 5,8 GiB paměti. Použita byla aktuální mapa České republiky ze 7. března 2023 o velikosti 782,8 MiB. Výstupní matice:

$$\begin{pmatrix} 0 & 14697 & 108475 & 40634 & 50393 & 98449 \\ 14697 & 0 & 100050 & 33069 & 39043 & 86601 \\ 108475 & 100050 & 0 & 71781 & 99735 & 37462 \\ 40634 & 33069 & 71781 & 0 & 50575 & 64284 \\ 50791 & 39345 & 99724 & 50580 & 0 & 74960 \\ 98551 & 86703 & 37478 & 64405 & 74625 & 0 \end{pmatrix}$$

Výpočet převodníku *OSRM* byl okamžitý. Výstupní matice:

$$\begin{pmatrix} 0 & 14904 & 134184 & 47997 & 58043 & 110963 \\ 14904 & 0 & 119694 & 34850 & 43554 & 96474 \\ 134640 & 120472 & 0 & 91937 & 125625 & 40084 \\ 48079 & 34850 & 91251 & 0 & 78991 & 68031 \\ 57138 & 42971 & 125220 & 78985 & 0 & 101999 \\ 110843 & 96676 & 40059 & 68140 & 101829 & 0 \end{pmatrix}$$

Výpočet převodníku *OpenRouteService* byl okamžitý. Výstupní matice:

$$\begin{pmatrix} 0 & 14906 & 133919 & 44842 & 56222 & 110975 \\ 14906 & 0 & 119450 & 34771 & 41753 & 96506 \\ 134374 & 120417 & 0 & 81154 & 111104 & 39682 \\ 44842 & 34771 & 80843 & 0 & 59787 & 71533 \\ 55854 & 41896 & 111080 & 60162 & 0 & 88136 \\ 110599 & 96641 & 39682 & 71779 & 87328 & 0 \end{pmatrix}$$

Ze srovnání jsou patrné rozdíly, které jsou místy poměrně značné (někdy se jedná o metry, jindy až o desítky kilometrů). Pro přesné určení příčiny všech rozdílů by byla třeba detailní

analýza celého výpočetního procesu spolu s mapovými údaji, ale logicky se právě v těch nějaké odlišnosti nacházet mohou.

V první řadě je třeba brát v potaz skutečnost, že mapové údaje užívané k danému datu různými službami se od sebe mohou lišit. Například na stránkách služby *OSRM* se uvádí:

> *„Jsou využívány tři servery; jeden z nich neustále připravuje trasovací graf a stahuje nejnovější data* OpenStreetMap*, zatímco dva další poskytují trasy získané z těchto překalkulovanýma grafů. (…) Trasovací údaje jsou denně aktualizovány. Ve vnitřním zpracovávání a tvoření trasovacího diagramu se může vyskytnout prodleva.“*[55, volně přeloženo]

Naproti tomu v sekci „Často kladené otázky“ se na stránkách služby *OpenRouteService* lze dočíst toto:

> *„Náš proces aktualizace je spuštěn tehdy, kdy se objeví nový soubor planety. Nahrazení staré sítě trvává asi tak 1–2 týdny. (…)“*[56, volně přeloženo]

Z těchto sdělení lze tedy usoudit, že obě služby mohou ke stejnému datu pracovat s odlišnými verzemi map. Vzhledem k tomu, že mapy jsou aktualizovány a vylepšovány prakticky neustále, můžou se mezi verzemi vyskytnout změny (dříve nezmapované cesty nebo informace o nich, nebo změny ve skutečné dopravní síti) které se promítnou i do vypočítané trasy. Dokladem tohoto tvrzení může být matice získaná od služby *OSRM* jen tři dny po té předchozí:

$$\begin{pmatrix} 0 & 14904 & 134184 & 47997 & 57497 & 110963 \\ 14904 & 0 & 119694 & 34850 & 43008 & 96474 \\ 134640 & 120472 & 0 & 91937 & 125080 & 40084 \\ 48082 & 34850 & 91251 & 0 & 78445 & 68031 \\ 57138 & 42971 & 125220 & 78985 & 0 & 101999 \\ 110843 & 96676 & 40059 & 68140 & 101283 & 0 \end{pmatrix}$$

Jak vidno, došlo tu k určitým změnám (v řádu stovek metrů), a to zejména v pátém sloupci, tedy při cestách do Srnojed. Naproti tomu u služby *OpenRouteService* byla matice zcela totožná, a tedy lze předpokládat, že ke změnám nedošlo.

Dalším důvodem může být rozdíl v samotných programech, které výpočet provádějí, ať už se jedná o rozdíly v algoritmech, heuristice, nebo přímo ve výpočtu. Ukázat to lze na procházce Zahrádkou, v kraji Vysočina, okolo rybníka Vesník. Všechny převodníky byly postupně využity k výpočtu vzdálenosti souřadnic (49,2447123; 16,0958073) a (49,2437774; 16,101292) po optimální trase, a to jak pro cestu pěšky, tak pro cestu autem. Spočítaná vzdálenost byla v obou směrech stejná, ale mírně se lišila. Prezentována je níže, v tabulce 1.

Tabulka 1. Procházka Zahrádkou.

Převodník	Pěšky	Autem
RoutingKit	445 m	445 m
OSRM	449 m	449 m
OpenRouteService	610 m	448 m

Zdroj: Výstup programu *TSPbrousek*.

Jak je patrné, rozdíly v řádu metrů napovídají, že trasa byla zvolena stejná, ale v jejích poznaných parametrech mohlo při různých výpočtech dojít k drobným rozdílům. Pozoruhodný je pak rozdíl v řádech stovek metrů u služby *OpenRouteService* – ten již svědčí o tom, že byla zvolena trasa jiná, službou vyhodnocena jako vhodnější pro pěší cestující. Ostatní převodníky upřednostnili stejnou trasu jak pro pěší, tak pro automobilisty.

Jednotlivé převodníky tedy vracejí uživateli různé matice, které se od sebe liší v závislosti na zvláštnostech užitých nástrojů. Vzhledem k vrozené nedokonalosti map jakožto modelů skutečné dopravní sítě, a různících se přístupů k řešení dané problematiky, lze tento stav považovat za uspokojivý. Uživatel by ovšem měl být o možných rozdílech zpraven, čehož se autor ujímá v návodu k použití. (Viz přílohu č. 1.)

6 Závěr

Problém obchodního cestujícího je dnes již dobře známým a často řešeným problémem. V praxi se sním často setkávají různí řešitelé. Mezi ty, kteří hledají optimální trasu mezi skutečnými lokacemi, bezpochyby patří dopravci i někteří studenti. Prvním krokem k řešení takové úlohy je však získání matice sazeb, třeba v podobě vzdáleností mezi jednotlivými lokacemi, které může být dosti pracné.

Právě tuto dílčí úlohu automaticky řeší program *TSPbrousek*. K řešení využívá tří různých nástrojů, které také představují tři různé přístupy k řešení problému. Program dokáže úlohu řešit velmi rychle za použití veřejně dostupných internetových služeb. Pokud si však uživatel opatří mapové údaje (které jsou také veřejně dostupné, a to ze zdrojů šířených spolu s programem) a souřadnice jednotlivých lokací, je program schopen funkce i bez připojení a bez závislosti na jakýchkoliv dalších službách. Jeho obsluha je přitom velice jednoduchá, a funguje na operačních systémech *Windows* a *GNU/Linux*.

Program neřeší samotnou dopravní úlohu, k níž hotoví matici. Tu lze řešit pomocí jiných, již existujících programů, např. právě *TSPkosy*, pro kterou je výstupní matice *TSPbrousku* primárně určena. Nabízí se také možnost řešit úlohu jako celočíselné programování v programu *OpenSolver*, který také funguje při tabulkovém procesoru.

Přesto by však funkce programu mohla být dále zdokonalena. K dispozici jsou další služby a knihovny, které *TSPbrousek* v současné době nevyužívá, a mohl by tak být rozšířen o další převodníky (čemuž je strukturálně uzpůsoben). Převod adres navíc zprostředkovává pouze převodník *OpenRouteService*, který je omezen množstvím požadavků; jako další zlepšení se tedy nabízí implementace této funkce za použití jiné služby nebo knihovny. Zprovoznit lokální převod adres pomocí knihovny *RoutingKit* se v rámci práce nepodařilo.

Navázat na dosavadní práci tedy bude možné třeba právě odstraněním některých z výše zmíněných nedostatků, nebo rozšířením výstupu programu o další formáty, lépe uzpůsobené jednotlivým aplikacím; nabízí se třeba výstup do formátu *XLSX* (*Office Open XML Workbook*), spolu s připravenou a vhodně formátovanou soustavou rovnic dle Tuckerových podmínek. Velmi užitečnou by mohla být i podpora dalších populárních platforem, jako je třeba *macOS*, nebo převod do formy webové aplikace.

7 Seznam použitých zdrojů

[1] STALLMAN, Richard M. 2015. *Free Software, Free Society*. 3. vydání. Boston: Free Software Foundation. 305 s. ISBN 978-0-9831592-5-4.

[2] Ubuntu CZ/SK. *Ubuntu CZ/SK* [online]. [cit. 2023-03-06]. Dostupné z: https://www.ubuntu.cz/.

[3] ŠUBRT, T. 2019. *Ekonomicko-matematické metody*. 3. upravené a rozšířené vydání. Plzeň: Vydavatelství a nakladatelství Aleš Čeněk, s.r.o. ISBN 978-80-7380-762-7.

[4] PELIKÁN, J. 2001. *Diskrétní Modely v Operačním Výzkumu*. 1. vydání. Praha: Professional Publishing. 163 s. ISBN 978-80-86419-17-6.

[5] APPLEGATE, D. L. 2006. *The Traveling Salesman Problem: A Computational Study*. Princeton: Princeton University Press. 596 s. ISBN 978-0-691-12993-8.

[6] BIGGS, N. 1976. *Graph Theory 1736-1936*. Oxford: Clarendon Press. 266 s. ISBN 978-0-19-853901-8.

[7] MATAI, R., SINGH, S., LAL, M. 2010. *Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches*. V: DAVENDRA, D. *Traveling Salesman Problem, Theory and Applications*. InTech. ISBN 978-953-307-426-9. Dostupné z: https://doi.org/10.5772/12909.

[8] ČZU v Praze. 2023. *Závěrečné práce* [online]. [cit. 2023-03-04]. Dostupné z: https://is.czu.cz/auth/zp/portal_zp.pl.

[9] KUČERA, P. 2009. *Metodologie řešení okružního dopravního problému*. Praha. 122 s. Disertační práce. ČZU v Praze. HAVLÍČEK, J.

[10] FILIPEC, J., MEJSTŘIK, V. 2005. *Slovník spisovné češtiny pro školu a veřejnost*. 4. vydání. Praha: Academia. Studentská edice. 647 s. ISBN 978-80-200-1347-7.

[11] PAVLÍČKOVÁ, L. 2015. *Volně šiřitelné digitální mapy – Projekt OpenStreetMap*. České Budějovice. 51 s. Bakalářská práce. Jihočeská univerzita v Českých Budějovicích. PAVEL, M.

[12] BERNARD, W. 2012. *Fiziko, baza kurso, unua parto*. Jižní Tiroly: Esperanto-klubo Sudtirolo. 99 s. Dostupné z: https://walter.bernard.im/fiziklibro/.

[13] JOLIVEAU, T. 2010. La géographie et la géomatique au crible de la néogéographie. *Tracés*. Revue de Sciences humaines. 10, s. 227–239. ISSN 1763-0061.

[14] JOLIVEAU, T. 2011. Le géoweb, un nouveau défi pour les bases de données géographiques. *Espace géographique*. 40, s. 154–163. ISSN 0046-2497.

[15] Google. *Mapy Google* [online]. [cit. 2023-03-03]. Dostupné z: https://www.google.cz/maps/.

[16] Google. *Platform Pricing & API Costs* [online]. [cit. 2023-03-02]. Dostupné z: https://mapsplatform.google.com/pricing/.

[17] STALLMAN, R. M. *Reasons not to use Google* [online]. [cit. 2023-03-05]. Dostupné

z: https://stallman.org/google.html.

[18] *Bing Maps Dev Center* [online]. [cit. 2023-03-02]. Dostupné z: https://www.bingmapsportal.com/.

[19] Seznam.cz, a.s. *Základní mapový podklad*. Seznam Nápověda [online]. [cit. 2023-03-06]. Dostupné z: https://napoveda.seznam.cz/cz/mapy/mapove-podklady/zakladni-mapovy-podklad/.

[20] Seznam.cz, a.s. *API Mapy.Cz* [online]. [cit. 2023-03-02]. Dostupné z: https://api.mapy.cz/.

[21] OpenStreetMap. *What is the history of OSM?* [online]. [cit. 2023-03-05]. Dostupné z: https://welcome.openstreetmap.org/about-osm-community/history-of-osm/.

[22] Open Knowledge Foundation. *Open Data Commons Open Database License (ODbL)*. Open Data Commons: legal tools for open data [online]. [cit. 2023-03-05]. Dostupné z: https://opendatacommons.org/licenses/odbl/.

[23] SEHRA, S. S., SINGH, J., RAI, H. S. Assessment of OpenStreetMap Data – A Review. *International Journal of Computer Applications*. 76(16), s. 17–20. ISSN 09758887.

[24] *OpenStreetMap.cz* [online]. [cit 2023-03-06]. Dostupné z: https://openstreetmap.cz/.

[25] MOONEY, P., MINGHINI, M. 2017. A Review of OpenStreetMap Data. FOODY, G., SEE, L., FRITZ, S., MOONEY, P., OLTEANU-RAIMOND, A-M., FONTE, C. C., ANTONIOU, V. *Mapping and the Citizen Sensor*. London: Ubiquity Press. s. 37–59. Dostupné z: https://doi.org/10.5334/bbf.c.

[26] *Elements*. OpenStreetMap Wiki [online]. [cit. 2023-02-26]. Dostupné z: https://wiki.openstreetmap.org/wiki/Elements.

[27] BAKER, H. 1995. CONS should not CONS its arguments, part II: Cheney on the M.T.A. *Sigplan Notices – SIGPLAN*. 30, s. 17–20.

[28] Tecgraf/PUC-Rio. *IUP – Portable User Interface* [online]. [cit. 2023-03-07]. Dostupné z: https://www.tecgraf.puc-rio.br/iup/.

[29] Repology. *Iup package versions* [online]. [cit. 2023-03-09]. Dostupné z: https://repology.org/project/iup/versions.

[30] CHUST, T. *Iup*. The CHICKEN Scheme wiki [online]. [cit. 2023-03-07]. Dostupné z: https://wiki.call-cc.org/eggref/5/iup.

[31] *Webview* [online]. [cit. 2023-03-07]. Dostupné z: https://wiki.call-cc.org/eggref/5/iup.

[32] CHUST, T. *Webview*. The CHICKEN Scheme wiki [online]. [cit. 2023-03-07]. Dostupné z: https://wiki.call-cc.org/eggref/5/webview.

[33] The Qt Company. *Qt | Cross-platform Software Design and Development Tools* [online]. [cit. 2023-03-07]. Dostupné z: https://www.qt.io/.

[34] KDE e.V. *KDE* [online]. [cit. 2023-03-07]. Dostupné z: https://kde.org/.

[35] UBports Foundation. *Ubuntu Touch* [online]. [cit. 2023-03-08]. Dostupné z: https://ubuntu-touch.io/.

[36] VideoLAN. *VLC media player* [online]. [cit. 2023-03-08]. Dostupné z: https://www.videolan.org/vlc/.

[37] The Qt Company. *Qt Network 6.4.2*. Qt Documentation [online]. [cit. 2023-03-09]. Dostupné z: https://doc.qt.io/qt-6/qtnetwork-index.html.

[38] Přispěvatelé projektu OpenStreetMap. *Routing*. OpenStreetMap Wiki [online]. [cit. 2023-03-11]. Dostupné z: https://wiki.openstreetmap.org/wiki/Routing.

[39] openrouteservice. *Openrouteservice* [online]. [cit. 2023-03-08]. Dostupné z: https://openrouteservice.org/.

[40] SCHNELL, J. 2022. Matrix distance and time calculations. *openrouteservice ask forum* [online]. 2022-12-21 [cit. 2023-03-11]. Dostupné z: https://ask.openrouteservice.org/t/matrix-distance-and-time-calculations/4417/3.

[41] LUXEN, D., VETTER, C. 2011. Real-time routing with OpenStreetMap data. V: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Chicago Illinois: ACM. s. 513–516. ISBN 978-1-4503-1031-4. Dostupné z: https://doi.org/10.1145/2093973.2094062.

[42] *The 2-Clause BSD License*. Open Source Initiative [online]. [cit. 2023-03-04]. Dostupné z: https://opensource.org/license/bsd-2-clause/.

[43] NEIS, P. 2011. Comparison of (OSM) routing-engines – Reloaded. *Neis One!* [online]. [cit. 2023-03-08]. Dostupné z: https://neis-one.org/2011/07/comparison-reloaded/.

[44] Geoapify. 2022. *OpenStreetMap Routing: Pros And Cons* [online]. [cit. 2023-03-04]. Dostupné z: https://www.geoapify.com/openstreetmap-routing/.

[45] NOLDE, N. 2020. *Open Source Routing Engines And Algorithms – An Overview* [online]. GIS•OPS. [cit. 2023-03-08]. Dostupné z: https://gis-ops.com/open-source-routing-engines-and-algorithms-an-overview/.

[46] OSRM. *OSRM API Documentation* [online]. [cit. 2023-03-11]. Dostupné z: https://project-osrm.org/docs/v5.24.0/api/.

[47] RoutingKit. *RoutingKit* [online]. [cit. 2023-02-25]. Dostupné z: https://github.com/RoutingKit/RoutingKit.

[48] RoutingKit. *RoutingKit Documentation* [online]. [cit. 2023-03-11]. Dostupné z: https://github.com/RoutingKit/RoutingKit/blob/13bc2f49f41d2e8a750a299487705ca7217d634d/doc/ContractionHierarchy.md

[49] DIBBELT, J., STRASSER, B., WAGNER, D. 2015. *Customizable Contraction Hierar-*

chies.

[50] Český statistický úřad. *Zajímavosti názvů obcí v České republice* [online]. [cit. 2023-03-08]. Dostupné z: https://www.czso.cz/csu/czso/zajimavosti_nazvu_obci_v_ceske_republice.

[51] Microsoft. *Formáty souborů podporované v Excelu*. Podpora Microsoftu [online]. [cit. 2023-03-12]. Dostupné z: https://support.microsoft.com/cs-cz/office/form%C3%A1ty-soubor%C5%AF-podporovan%C3%A9-v-excelu-0943ff2c-6014-4e8d-aaea-b83d51d46247.

[52] Tým pro dokumentaci LibreOffice. 2023. *Příručka aplikace Calc*. 560 s. Dostupné z: https://documentation.libreoffice.org/assets/Uploads/Documentation/cs/CG74-CalcGuide-CS.pdf.

[53] Přispěvatelé softwaru Gnumeric. *The Gnumeric Manual*. v1.12. Dostoupné z: https://help.gnome.org/users/gnumeric/stable/index.html.en.

[54] openrouteservice. *Plans*. Openrouteservice [online]. [cit. 2023-03-12]. Dostupné z: https://openrouteservice.org/plans/.

[55] OSRM. *OSRM routing server using OpenStreetMap data* [online]. [cit. 2023-03-10]. Dostupné z: https://map.project-osrm.org/about.html.

[56] openrouteservice. *FAQ*. Openrouteservice [online]. [cit. 2023-03-11]. Dostupné z: https://openrouteservice.org/faq/.

8 Seznam obrázků, tabulek, grafů a zkratek

Seznam obrázků

Seznam tabulek

Seznam zkratek

API. Application Programming Interface (Programovací rozhraní aplikace)

CSV. Comma-Separated Values (Hodnoty oddělené čárkou)

GNU. GNU's Not Unix

LGPL. Lesser General Public License

MIT. Massachusetts Institute of Technology (Institut technologie v Massachusetts)

ODbL. Open Database License

OSRM. Open Source Routing Machine

PBF. Protocolbuffer Binary Format

UML. Unified Modelling Language

XML. eXtensible Markup Language

Přílohy

Příloha č. 1 – Návod k programu *TSPbrousek*

TSPbrousek je jednoduchý program, s jehož pomocí lze převést seznam lokací na mapě na matici sazeb pro problému obchodního cestujícího. Toto je návod k jeho užívání.

Obsah návodu

1. Instalace
2. Obsluha
    - Zadávání lokací
    - Převodníky
    - Zpracování výstupu
3. Kde získat soubory s mapami

1 Instalace

Program není třeba instalovat; lze jej spustit bez instalace.

- Pokud používáte systém *Windows*, spusťte soubor `tspbrousek.exe`.
- Pokud používáte systém *GNU/Linux* (třeba *Debian*, *Ubuntu*, *Fedoru*, *SUSE*), spusťte soubor `tspbrousek.x86_64.AppImage`.

Nepřesouvejte prosím žádné soubory z adresáře `windows`; program by pak na systémech *Windows* nemusel fungovat. Pokud chcete program přesunout, přesuňte celý adresář `Windows`. Adresář můžete klidně přejmenovat.

Pokud vám spustitelné soubory nefungují a stavba programů vám není zcela cizí, můžete si program sami zkompilovat. Instrukce naleznete v souboru `Kompilace.txt`.

2 Obsluha

Do programu nejprve musíte zadat lokace, pro které chcete vypočíst matici. Poté je třeba zvolit převodník, který má matici spočítat. Pak jen zbývá potvrdit váš výběr a zkopírovat data pro další zpracování.

Zadávání lokací

Lokace můžete zadávat buďto formou souřadnic, nebo jako adresy. V případě zadávání souřadnic pište souřadnice s desetinnou tečkou a oddělené čárkou, následovně: `50.1305712, 14.3732870`. V případě zadávání adres vepište údaje jako je ulice, město, či země, oddělené čárkami. Pokud bude nalezeno vícero lokací s podobnými adresami, budete vyzváni, abyste si právě jednu zvolili. Adresa bude následně převedena na souřadnice.

Převodníky

Převodník je nástroj, který převádí lokace na matici sazeb. Každý z nich vnitřně funguje trochu jinak, následkem čehož má i trochu jiný výstup. Pokud tedy hodláte porovnávat několik různých matic, raději je všechny vypočítejte stejným převodníkem. K dispozici jsou převodníky celkem tři:

RoutingKit. Tento převodník hledá co nejkratší cesty. Funguje bez připojení k internetu, ale je třeba mu poskytnout soubor s mapou. Navíc pro jeho zpracování potřebuje poměrně dost paměti. Pokud jí váš počítač má alespoň 8 GiB, měl by být schopen zpracovat mapu České republiky. Výpočet může trvat několik minut.

Open Source Routing Machine. Tento převodník hledá co nejrychlejší cesty. Využívá veřejně dostupné internetové služby, a vyžaduje tedy internetové připojení. Služba není téměř nijak omezena. Výpočet bývá velmi rychlý.

OpenRouteService. Také tento převodník hledá co nejrychlejší cesty, a využívá internetové služby. Je navíc zodpovědný za hledání adres. Služba vyžaduje přístupový klíč k API, který má pouze omezený počet požadavků denně. V programu je přednastaven společný klíč pro všechny uživatele. Pokud by snad přestal fungovat, nebo byste měli v plánu počítat stovky matic, můžete se zdarma registrovat na adrese `https://openrouteservice.org/plans/` a získat tak vlastní klíč. Výpočet bývá velmi rychlý.

Zpracování výstupu

Po zvolení přechodníku stiskněte tlačítko v pravém dolním rohu, a vyčkejte, dokud se vám nezobrazí matice.

Výstupní matice vyjadřuje délku zvolených tras v metrech. Ve výchozím zobrazení je prezentována v tabulce, ale lze ji zobrazit i ve formátu *CSV* (což jsou jednoduše hodnoty oddělené od sebe čárkami). Pro zkopírování tabulky ve formátu *CSV* použijte k tomu určené tlačítko, nebo položku menu `Úpravy` → `Zkopírovat CSV`. Tabulku v tomto formátu můžete vložit do libovolného tabulkového procesoru (jako je *Excel* nebo *Calc*).

Pro výpočet další matice nemusíte program restartovat; můžete odstranit všechny lokace pomocí menu `Úpravy` → `Odstranit všechny lokace` a celý proces zopakovat. Pokud již máte nahraný soubor s mapou, ušetří vám to čas.

3 Kde získat soubory s mapami

Převodník *RoutingKit* pracuje s mapami ve formátu *PBF*. Ty se dají získat z různých zdrojů, ať už pro celou zeměkouli, nebo jen pro její části. Pokud se rozhodnete pro velmi rozsáhlou mapu, pak vězte, že běžný osobní počítač dost možná nebude dostatečně výkonný, aby ji zpracoval. Zde jsou některé odkazy:

Vysoké učení technické v Brně. Poskytuje aktuální mapu České republiky na adrese:

https://osm.fit.vutbr.cz/extracts/czech_republic/

Geofabrik. Poskytuje mapy různých celé zeměkoule i dílčích částí světa na adrese: https://download.geofabrik.de/

OpenStreetMap.fr. Poskytuje mapy některých světadílů, zemí, i krajů, na adrese: https://download.openstreetmap.fr/extracts/

Protomaps. Umožňuje vyříznutí libovolné části mapy; vhodné v případě přeshraniční dopravy: https://app.protomaps.com/downloads/osm

TSPbrousek používá mapy projektu *OpenStreetMap*, které jsou volně dostupné široké veřejnosti, a do jejichž tvorby se může zapojit kdokoliv – i vy! Stránky místní komunity s informacemi o projektu v češtině naleznete zde: https://openstreetmap.cz/. Hlavní stránky projektu jsou pak na adrese: https://www.openstreetmap.org/.

Tento program byl vytvořen Matyášem ELICEREM v rámci jeho bakalářské práce na České zemědělské univerzitě v Praze.