

Posudek oponenta bakalářské práce

Autor práce:	David Beinhauer
Název práce:	Optimalizace rozmístění stanic pro nabíjení elektrických vozidel
Studijní program:	Informatika
Studijní obor:	Obecná informatika
Rok odevzdání:	2022

Cílem práce je vytvořit simulátor dopravy a použít jej k optimalizaci rozmístění nabíjecích stanic pro elektromobily.

První kapitola popisuje algoritmy použité k optimalizaci. Druhá kapitola rozebírá, jak student získal z Open Street Maps silniční síť potřebnou k simulaci dopravy. Třetí kapitola nazvaná „Popis simulátoru“ podává definice pojmů používaných v práci. Čtvrtá rozebírá simulaci jízd vozidel a jejich nabíjení. Pátá popisuje optimalizační algoritmy pro rozmístění nabíjecích stanic. Poslední kapitola podává výsledky experimentů. V příloze textu práce je uživatelská a programátorská dokumentace. V softwarové příloze jsou zdrojové programy, testovací data a text práce.

Popis problému studovaného v práci vnímám jako těžko srozumitelný z následujících důvodů.

- Na začátku práce chybí souhrnný popis studovaného modelu a motivace jednotlivých částí, a proto jsem při čtení dost tápal, co simulace dopravy počítá. Například z názvu práce je zřejmé, že práce hledá rozmístění nabíjecích stanic, ale v práci není pořádně napsáno, zda se stanice může nacházet jen ve městech nebo i na křižovatkách či na libovolném místě silnice. Jedinou indicií je definice nabíjecí stanice na straně 25, kde jednou vlastností stanice je „ $v \in V$ - pozice stanice (vrchol v grafu dopravní sítě)“, z čehož není zřejmé, zda může být v libovolném vrcholu či jsou uvažovány nějaká omezení daná možnostmi výstavby nových stanic či využitím stávajících. Další otázkou je, co se student snaží optimalizací rozmístění stanic zlepšit. Je cílem minimalizovat počet stanic, náklady na jejich výstavbu a provoz, množství energie nutné k zajíždění na stanice, ztráty na přenosové síti nebo maximalizovat spokojenost řidičů? Až na straně 39 je definována ztrátová funkce, která je váženým součtem pěti hodnot. Dále na straně 21 se čtenář dozví, že se v modelu pracuje s různými typy silnic, na straně 23 se z nich počítá kapacita silnic, z čehož postupně plyne, že simulace uvažuje i hustotu dopravy.
- Přesnost definic popisuje sám autor na straně 19: „Dále je potřeba zmínit, že některé definice uvedené v následujícím textu, jsou zjednodušeny a jejich zápis nemusí být matematicky naprosto přesný. Jsou například vynechány nějaké předpoklady z definic, které zřejmě vyplývají z textu. K tomuto kroku bylo přistoupeno pod snahou zpřehlednit jednotlivé definice.“
- Bohužel u nepřesných definic obsahujících řadu faktických chyb (viz níže) není popsán jejich smysl ani účel.

Z těchto důvodů je možné, že jsem některým věcem z práce správně neporozuměl, z čehož můžou pramenit nepřesnosti v tomto posudku. Proto jsem se se studentem sešel, aby mi nejas-

nosti vysvětlil. Domluvili jsme se, že student opraví experimenty, což udělal, jak je podrobněji rozebráno níže. Proto některé části posudku mohou vycházet i z osobní diskuze a pozdějších oprav.

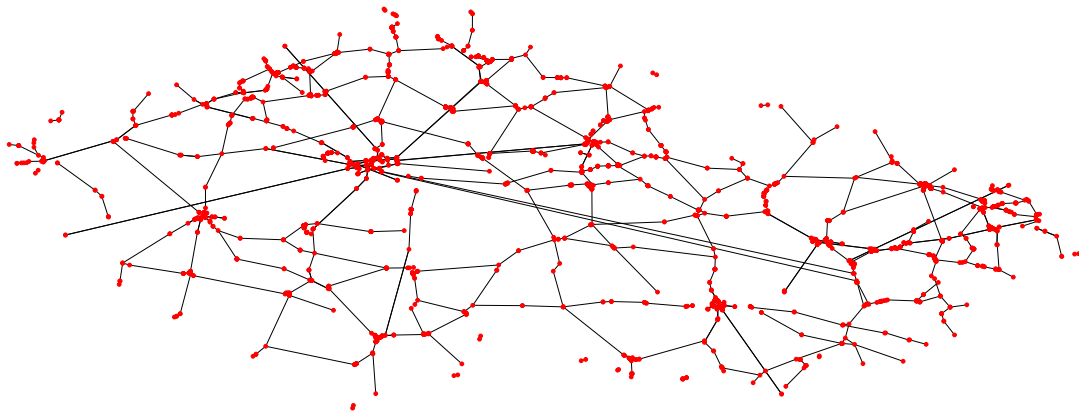
Nyní uvedu několik příkladů nepřesných definic.

- Na straně 12 je popsán algoritmus A^* využívající heuristickou funkci h , pro nichž monotónnost je vysvětlena slovy „hodnota funkce h roste po přechodu na další hranu“.
- Na straně 21 je uvedeno, že každé silnici je přiřazeno „ S – množina segmentů silnice, pro kterou je splněna podmínka: $(\forall s, r \in N; r < s)(s \in S \implies r \in S)$ “. To by znamenalo, že tedy $S = \{1, \dots, z\}$ pro nějaké z , což ve zbytku textu nedává smysl.
- Na straně 23 se kapacita silnice definuje vzorcem „ $c(r) := \alpha \cdot f(r.t) + \beta \cdot d \cdot |r.S|$ “, kde $f(r.t)$ je hodnota závislá na typu silnice a $d\Delta|r.S|$ odpovídá délce silnice. Dle vyjádření studenta se jedná o překlep, protože se tyto dvě hodnoty mají násobit, což dává větší smysl a doopravdy se tak ve zdrojových kódech děje. V tomto případě, ale nedává smysl součin násobit dvěma různými parametry α a β .
- Na straně 48 student uvádí: „Dále je vhodné zmínit, že volíme pevnou dobu úplného nabití vozidla na 5 minut, která je optimistickým předpokladem s ohledem na reálné hodnoty² a každá nabíjecí stanice má pouze 1 nabíjecí slot.“ Odkazovaný zdroj ale vůbec netvrdí, že baterii elektrických aut lze nabít za 5 minut. Dle vysvětlení studenta předpokládá, že stanice má ve skutečnosti více nabíjecích slotů a tak zvládne odbovit průměrně jedno auto za 5 minut a tím si zjednodušil model. Bohužel se tím ale nezapočítává doba nabíjení do celkové doby jízdy, která se v experimentech uvažuje. Tento fakt student po konzultaci opravil.
- Strana 70 a další: „segmentLength: Délka hrany grafu v kilometrech.“. Všechny silnice nemají stejnou délku, ale hodnota udává, na jak dlouhé úseky jsou silnice rozděleny.
- Strana 70: „carConsumption: Spotřeba baterie vozidla v procentech kapacity za minutu.“. Hodnota nemá být v procentech, ale udávat část z celku, tedy být v intervalu 0 až 1.

K porozumění textu jsou nejdůležitější definice v třetí kapitole, kde je definován graf dopravní sítě, zjednodušený graf dopravní sítě a další pojmy. Tyto definice mi přijdou zbytečně komplikované a mělo by stačit napsat, že máme graf, ve kterém vrcholy odpovídají křižovatkám, hrany silnicím, každá silnice je rozdělena na množiny úseků, a vyjmenovat hodnoty uložené ve vrcholech a hranách. Zavádění zjednodušeného grafu mi přijde zbytečné, protože slouží jen k popisu algoritmu A^* , který se spouští z koncových vrcholů hrany, na které se auto nachází.

Další částí práce studenta byla příprava dat obsahující silniční síť k experimentům. Zdrojem dat je Open Street Map, ze které student získal seznam měst a silnic v ČR, ze kterého vynechal silnice nižších tříd. Následně si student našel místa, kde se kříží alespoň tři silnice, která prohlásil za vrcholy, které propojil silnicemi.

- Tento postup je v zásadě rozumný a správný, ale student si bohužel výsledný graf nevykreslil (viz Obrázek 1), aby odhalil podstatné chyby mající zásadní vliv na experimenty.
- Sám student udává na straně 16: „V našem případě je graf rozdělen do 75 komponent souvislosti, což se jeví jako poměrně vysoké číslo a vyvolává pochybnosti, zda je naše volba typů silnic vhodná. Vysoký počet komponent si odůvodňujeme výběrem



Obrázek 1: Studentem připravená silniční síť ČR. Červené tečky jsou vrcholy grafu. Většinu silnic není vidět, protože 1416 z 2726 silnic má délku méně než 100 metrů.

mapy pokrývající pouze výřez reálné oblasti, což může vézt k rozpadu silniční sítě na více komponent souvislosti především v hraničních oblastech.“ To je pravda, ale dalším důvodem je skutečnost, že při výběru typů silnic student vynechal nájezdy na dálnice, čímž se silniční síť rozpadla a například D1 vede z Prahy až k Přerovu bez křížení s jinými silnicemi.

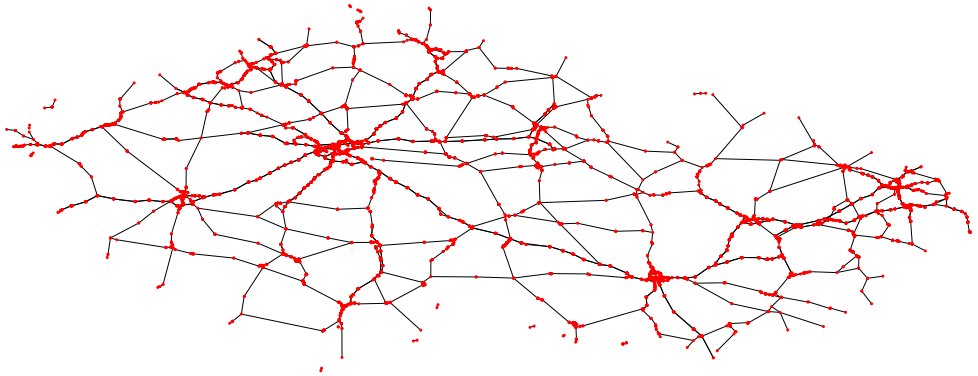
- Student na řadě míst neuvádí s jakými jednotkami pracuje, což se mu vymstilo v kapitole A.3.2, kde popisuje formát souboru reprezentující silnice. Program předpokládá, že délky silnic jsou v kilometrech, ale v souboru `prepared_edges.txt` jsou hodnoty v metrech, což je jeden z důvodů, proč výsledky experimentů nedávají smysl.

Po konzultaci student uvedené chyby opravil a výsledná mapa je na Obrázku 2.

Je sice pochopitelné, že se student snažil napsat obecný skript, který dokáže připravit silniční síť pro libovolnou oblast, ale z vlastních zkušeností vím, že takový úkol je velmi náročný a vzhledem k různým národním a místním specifikám v obecnosti nemožný. Proto by mi přišlo lepší kombinací malých skriptů a ručních úprav připravit jednu nebo několik málo kvalitních testovacích dat než spoléhat na data obsahující spousty chyb připravená obecným programem. Pro tento typ práce bych doporučoval naučit se pracovat s nějakým nástrojem pro Geografický informační systém.

Dále student musel nastavit různé parametry simulace, ke kterým mám tyto poznámky.

- Strana 48–49: „... volíme pevnou dobu úplného nabití vozidla na 5 minut ... každá nabíjecí stanice má pouze 1 nabíjecí slot. ... parametrem je doba simulace. Tu nastavujeme na 1000 minut. ... celkový počet nabíjecích stanic, který nastavujeme na hodnotu 300.“ Tedy za celou simulaci je možné nabít nejvýše 60 tisíc aut, ale počet vybitých aut se v experimentech pohybuje okolo 600 tisíc.
- Parametr `carConsumption` udává, jak velká část baterie se vybité za minutu. Tedy při vyšší rychlosti má auto na stejnou vzdálenost menší spotřebu.
- Testované hodnoty parametru `carConsumption` jsou 0.0001 a 0.001, tedy více než 16 hodin, resp. téměř 7 dní. Největší dojezdová vzdálenost uvedená ve zdroji, který student v práci používá, je 383 km. Navíc připomeňme, že doba simulace je 1000 minut.



Obrázek 2: Opravená mapa po konzultaci.

- Hodnota parametru `carBatteryMean` na straně 48 udávající „Střední hodnota počáteční hladiny baterie“ je 0.7, což v kombinaci s parametrem `carConsumption` znamená, že vozidlo může z počátečního místa jet 700 minut. Pro srovnání dle serveru `mapy.cz` doba jízdy z Aše do Jablunkova je 6 hodin a 8 minut, což znamená, že by auto měla být schopná přejet celou ČR bez nabíjení.
- Testované hodnoty parametru `carVelocity` na straně 49 udávající „Průměrná rychlost všech vozidel v kilometrech za minutu“ jsou 1, 1.5, 2, 3 a 5. Maximální povolená rychlost v ČR je 130 km/h.
- Strana 51 udává váhy v cílové funkci takto: `runDownParameter` („Parametr počtu vozidel, kterým se vybila baterie, pro výpočet ztrátové funkce“) má hodnotu 100 a `durationParameter` („Parametr průměrné doby cestování v minutách pro výpočet ztrátové funkce“) má hodnotu 1. Jelikož student v práci simuluje cesty na dlouhé vzdálenosti, by auta měla jezdit delší dobu než 100 minut. V cílové funkci, kterou se optimalizační algoritmy snaží minimalizovat, je výhodnější nechat auto co nejdříve vybit a započítat jen 100 než celou dobu jízdy vozidla až do cíle.
- Na straně 20 je uvedeno: „V zájmu výpočetní efektivity simulátoru a zkoumaného problému opomíjíme většinu cest na krátkou vzdálenost a naopak nadhodnocujeme počet dálkových tras vozidel.“ Souhlasím, že při optimalizaci rozmístění stanic je vhodné vynechat jízdy, které je možno ujet bez dobíjení. V této situaci ale nerozumím nastavení hodnot pro generování vzdálenosti jízd, který je přesněji rozebrán v zaslané opravě. Na straně 7 je uvedeno „Spotřebu vozidla volíme jako 0.002 z celkové hladiny baterie za minutu a rychlost vozidla volíme jako 1 km/min (60 km/hod). Tedy dojezd vozidla s maximální kapacitou je 500 km. Střední hodnotu normální distribuce popisující počáteční hladinu baterie jsme zvolili jako 0.9 s rozptylem 0.2.“. Na straně 10: „průměrná vzdálenost do cíle je 200 km“. Jestliže baterie jsou v průměru nabitě na 450 km a průměrná délka trasy je 200 km, tak téměř všechna vozidla nepotřebují dobíjení.
-

Ze své praxe vím, že získat správná data a realistické hodnoty parametrů je často velmi náročný úkol a někdy není možné vhodné hodnoty získat. Ale myslím si, že v uvedených

příkladech šlo hodnoty zvolit lépe. Student ve své práci mohl uvažovat futuristický model, ve kterém se auta stihnou nabít za 5 minut, můžou jet 300 km/h a zvládnout objet celou zeměkouli na jedno nabití. V tomto případě by bylo vhodné na tuto skutečnost jasně upozornit a nastavit parametry tak, aby simulace dávala smysl (např. aby se auta stihla vybit za dobu simulace a nemohla na počáteční nabití přejet mezi libovolnými místy testované silniční sítě).

Případá mi, že student zvolil zbytečně komplikovaný model, který v některých případech nejen ztěžuje nastavování parametrů a reálnost spíš jen zhoršuje. Například na straně 20 student uvádí: „Při volbě velikosti simulované oblasti jsme předpokládali, že většina majitelů vozidel má možnost dobít své vozy v počáteční a cílové destinaci (např. mají nabíjecí sloty v místě bydliště, či na pracovišti).“ S tímto souhlasím, a proto by bylo reálně možné předpokládat, že všechna auta budou při výjezdu plně nabitá, čímž by vypadly parametry `carBatteryMean`, `carBatteryDeviation`, `carStartBatteryBottomLimit`. Dále bych generování jízdy zjednodušil tak, že by se cílové místo volilo stejně jako počáteční a případně bych vynechával jízdy nevyžadující dobíjení, čímž bych si ušetřil parametry `exponentialLambdaCities` a `endCityRatio`.

Experimenty vzhledem k nevhodně nastaveným parametrům a chybám v testované silniční síti nedávají užitečné informace a většina vozidel se v simulaci vybila. Student sice zkouší zvětšovat počet nabíjecích stanic, ale ani tím počet vybitých vozidel podstatně neklesá. V opravené verzi se studentovi podařilo některé chyby opravit a „Neúspěšných cest do cílové destinace je tedy přibližně 3.2%“ (strana 10), ale vzhledem k průměrné délce trasy a dojezdové vzdálenosti mi tato neúspěšnost připadá příliš velká a bylo by vhodné podrobněji analyzovat, co se v simulaci děje. V této situaci bych například doporučoval nastavit všechny váhy ztrátové funkce kromě `runDownParameter`, aby nevnašeli do výsledků šum a chyby se tak dalo lépe odhalit. V experimentech nejsou analyzovány hodnoty ztrátové funkce, kterou evoluční algoritmy optimalizují. Pokud správně rozumím hladovému algoritmu, tak se v něm ztrátová funkce neřeší, takže výsledky hladového a evolučního algoritmu nejsou srovnatelné. V této kapitole by bylo užitečné též uvést velikost grafu a jak dlouho trvá jedna simulace a dobu výpočtu jednotlivých algoritmů.

Zpracování dat napsáno v jazyce Python a simulace s optimalizací v C++. Tato kombinace je pro daný problém vhodně zvolená.

Program se mi podařilo zkompileovat po přidání řádku `set(CMAKE_CXX_STANDARD 14)` do souboru `CMakeLists.txt`. Při spuštění dle návodu program spadl na `Segmentation fault`. Při konzultaci se ukázalo, že jsem program spouštěl z adresáře, do kterého se vytvořit spustitelný program místo adresáře s daty. V tomto případě by bylo lepší ohlásit, že nebyly nalezeny soubory s testovacími daty.

Požadavky ke spuštění programu neodpovídají skutečnosti. K překladu nestačí kompilátor C++11, ale je nutný C++14. Na druhou stranu GIT není potřebný, když v práci ani není odkaz na žádný gitový repozitář.

Zdrojové kódy jsou přehledné, funkce a proměnné srozumitelně pojmenované a pěkně komentované. Celý program je dobře strukturovaný. Při spuštění ze správného adresáře je stabilní a vypisuje užitečné informace.

V práci student prokázal, že umí programovat v jazycích Python i C++ a úspěšně zvládl používat knihovnu Boost. Student porozuměl evolučním algoritmům a zvládl je implementovat. Programátorsky je práce výborně zvládnutá. Zpracování dat z Open Street Map student s pomocí vedoucího i oponenta nakonec též zvládl. Bohužel odborné vyjadřování je nedo-

statečné a nastavení experimentů nedává smysl.

Na základě uvedených skutečností doporučuji, aby komise zvážila, zda u studenta bakalářského programu Informatika se specializací Umělá inteligence na MFF UK je důležitější programování nebo odborné vyjadřování a matematické porozumění studovaného problému.

Praha, 22. 8. 2022

Jiří Fink
KTIML MFF UK