

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA STAVEBNÍ  
PROGRAM GEODÉZIE A KARTOGRAFIE  
OBOR GEOMATIKA



DIPLOMOVÁ PRÁCE  
ROZŠÍŘENÍ ZÁSUVNÉHO MODULU QGIS PRO  
ZPRACOVÁNÍ GTFS O VIZUALIZACI TARIFNÍCH PÁSEM  
PID

Vedoucí práce: Ing. Martin Landa, Ph.D.  
Katedra geomatiky

červen 2021

Bc. Martin KOUBA

## ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta stavební

Thákurova 7, 166 29 Praha 6



# ZADÁNÍ DIPLOMOVÉ PRÁCE

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení:	Kouba	Jméno:	Martin	Osobní číslo:	468232
Zadávající katedra:	Katedra geomatiky				
Studijní program:	Geodézie a kartografie				
Studijní obor:	Geomatika				

## II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:	Rozšíření zásuvného modulu QGIS pro zpracování dat GTFS o vizualizaci tarifních pásem PID		
Název diplomové práce anglicky:	QGIS GTFS Plugin extension with visualization of PID tariff zones		
Pokyny pro vypracování:	<p>Cílem diplomové práce je rozšířit zásuvný modul (plugin) platformy QGIS pro zpracování dat ve formátu GTFS, na jehož vývoji se autor spolupodílel v rámci semestrálního projektu předmětu Free Software GIS ve 2. semestru magisterského studia. Primárním úkolem je najít a vhodně implementovat způsob vizualizace tarifních pásem Pražské integrované dopravy (PID) generovaných automaticky z podkladových dat.</p>		
Seznam doporučené literatury:	<p>SHERMAN, Gary. The PyQGIS Programmer's Guide: Extending QGIS 3 with Python 3. Ilustrované vydání. Locate Press, 2018. ISBN 9780998547725. MARK, Pilgrim. Dive into Python 3. Apress, 2009. ISBN 978-1430224150.</p>		
Jméno vedoucího diplomové práce:	Ing. Martin Landa, Ph.D.		
Datum zadání diplomové práce:	10.2.2021	Termín odevzdání diplomové práce:	17.5.2021
Údaj uveďte v souladu s datem v časovém plánu příslušného ak. roku			
Podpis vedoucího práce		Podpis vedoucího katedry	

## III. PŘEVZETÍ ZADÁNÍ

<p><i>Beru na vědomí, že jsem povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutné uvést v diplomové práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.</i></p>	
Datum převzetí zadání	Podpis studenta(ky)

## **ABSTRAKT**

v českém jazyce

## **KLÍČOVÁ SLOVA**

Python, QGIS, zásuvný modul, GTFS, PyQGIS

## **ABSTRACT**

v anglickém jazyce

## **KEYWORDS**

Python, QGIS, plugin, GTFS, PyQGIS

## PROHLÁŠENÍ

Prohlašuji, že diplomovou práci na téma „Rozšíření zásuvného modulu QGIS pro zpracování GTFS o vizualizaci tarifních pásem PID “ jsem vypracoval samostatně. Použitou literaturu a podkladové materiály uvádím v seznamu zdrojů.

V Praze dne .....

.....

(podpis autora)

## PODĚKOVÁNÍ

Tímto bych chtěl poděkovat panu Ing. Martinu Landovi, Ph.D. za pomoc při  
..... Zvláště bych chtěl poděkovat celé své rodině  
.....

# Obsah

<b>Úvod</b>	<b>9</b>
<b>1 Rešerše</b>	<b>10</b>
<b>2 GTFS</b>	<b>11</b>
<b>3 Pražská Integrovaná doprava (PID)</b>	<b>13</b>
<b>4 Postup</b>	<b>14</b>
4.1 Aktualizace pluginu do verze 1.0.0 . . . . .	14
4.2 Postup 1 - pomocí nástrojů QGIS . . . . .	14
<b>Závěr</b>	<b>24</b>
<b>Odkazy</b>	<b>25</b>
<b>Seznam příloh</b>	<b>26</b>

# Seznam obrázků

2.1	Ukázka CSV formátu ze souboru shapes.txt . . . . .	11
2.2	Diagram GTFS datasetu . . . . .	12
3.1	Logo Pražské integrované dopravy . . . . .	13
4.1	ProgressBar . . . . .	14
4.2	Voroného polygony . . . . .	15
4.3	Voroného polygony pro všechny zastávky . . . . .	16
4.4	Vybrané Voroného polygony pro pásmo 2 . . . . .	17
4.5	Výsledek nástroje Dissolve pro pásmo 2 . . . . .	18
4.6	Výsledek nástroje Extract vertices pro pásmo 2 . . . . .	18
4.7	Spojené body třech vektorových vrstev pro pásmo 2 . . . . .	19
4.8	Konkávní obálka pro pásmo 2 . . . . .	20
4.9	Znázornění algoritmu „Douglas-Peucker“ . . . . .	21
4.10	Výsledek nástroje Simplify pro pásmo 2 . . . . .	22
4.11	Výsledek nástroje Smooth pro pásmo 2 . . . . .	23

# Seznam tabulek



# Úvod

Pro cestování po Praze a okolí je často nejlepší volbou použít veřejnou dopravu. Tu zahrnuje metro, tramvaje, železnici, městské a příměstské autobusové linky, lanovou dráhu na Petřín a také některé přívozy. Celý tento systém se nazývá Pražská integrovaná doprava, který je postupně integrován společnými přepravními a tarifními podmínkami a jednotným dopravním řešením včetně koordinace jízdních řádů.

Systém Pražské integrované dopravy je rozdělen do dvanácti tarifních pásem, pro které platí různé jízdní doklady. Rozlohou jsou po celé Praze, na většině Středočeského kraje a dokonce i na částech Plzeňského a Ústeckého kraje, či na Vysočině.

Tarifní pásma Pražské integrované dopravy jsou zatím modelována manuálně a nejsou nikterak zautomatizována. Cílem této diplomové práce bude vytvořit postup pro automatické vyhotovení tarifních pásem a co nejvíce se přiblížit k jejich oficiální podobě. Tento postup bude publikován jako pokračování vývoje zásuvného modulu v softwaru QGIS, u kterého byla započata tvorba v předmětu Free software GIS v 5. semestru magisterského studia.

# 1 Rešerše

Tarifní pásma Pražské integrované dopravy jsou mimo jiné publikována jako shapefile<sup>1</sup> na portálu *opendata hlavního města Prahy* [2]. Tento shapefile obsahuje vektorové vrstvy polygonů.

První myšlenkou, jak se přiblížit k takovým polygonům, byl algoritmus Delaunay triangulace. Tento algoritmus vytváří liniové spojnice bodů mezi jednotlivými body, které si jsou sobě nejbližší, za pomoci opsaných kružnic. Delaunay triangulace má několik vlastností. Jednou z nich je například, že uvnitř kružnice k opsané libovolnému trojúhelníku neleží žádný jiný bod. Zároveň Delaunay triangulace je jednoznačná, pokud žádné čtyři body neleží na kružnici. [3]

Další nápadem byla aplikace Voronoi diagram algoritmu, což je způsob dekompozice metrického prostoru určený vzdálenostmi k dané diskrétní množině bodů v prostoru. Tento algoritmus má taky několik vlastností, jež budou prezentovány v dalších kapitolách (viz 4.2).

---

<sup>1</sup>formát vektorového datového úložiště Esri pro ukládání umístění, tvaru a atributů geografických prvků [1]

## 2 GTFS

General Transit Feed Specification, zkráceně GTFS, je dokument, který definuje obecný formát pro jízdní řády veřejné dopravy a příbuzné geografické informace. GTFS „feeds“ umožňují veřejným dopravním agenturám zveřejňovat svá přepravní data a vývojářům psát aplikace, které tato data spotřebovávají interoperabilním způsobem. [4]

GTFS dataset<sup>2</sup> obsahuje CSV soubory, jehož úplný seznam je v následující tabulce. Co je to CSV soubor?

CSV, *Comma-separated values*, v překladu *hodnoty oddělené čárkami*, je jednoduchý souborový formát určený pro výměnu tabulkových dat. Data jsou oddělována „oddělovačem“. Ačkoli podle definice by měl být formát oddělen čárkami, oddělovač může být prakticky cokoli. Nejčastějšími oddělovači jsou právě čárky, středníky nebo mezery. CSV soubor se může upravovat v jakémkoliv textovém editoru.

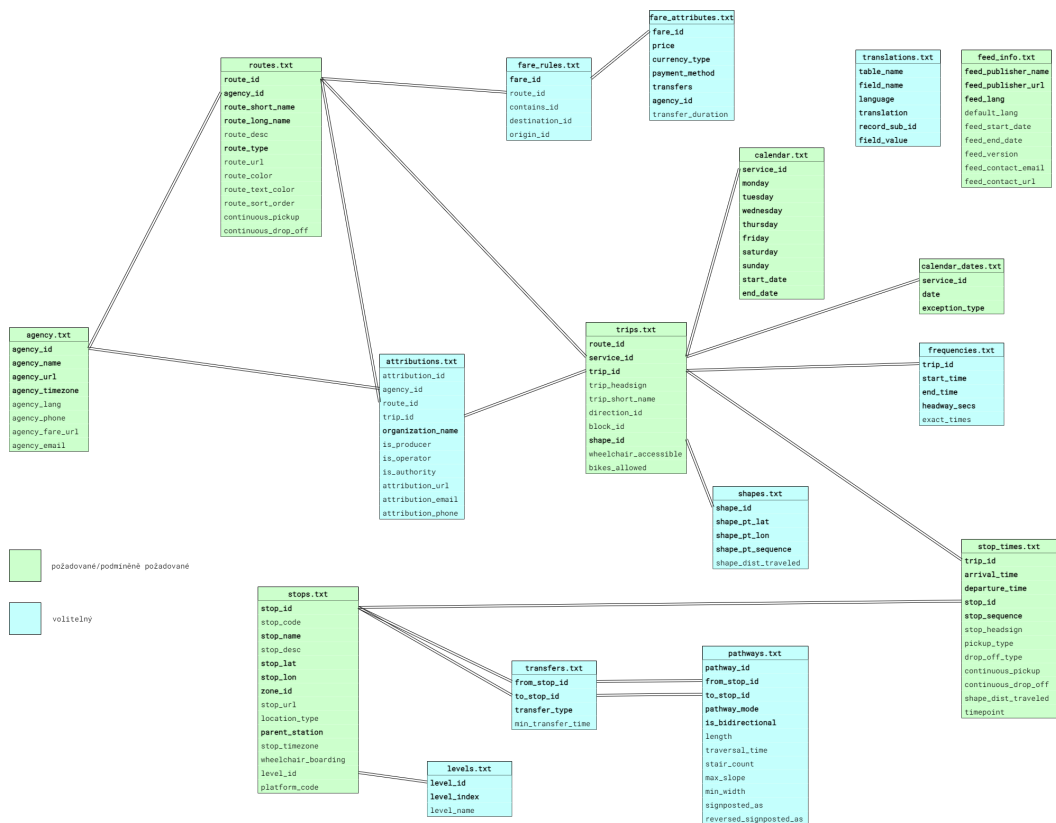
```
shape_id,shape_pt_lat,shape_pt_lon,shape_pt_sequence,shape_dist_traveled
L991V1,50.07553,14.51539,1,0.00000
L991V1,50.07371,14.51461,2,0.20965
L991V1,50.07361,14.51456,3,0.22173
L991V1,50.07351,14.51451,4,0.23382
L991V1,50.07340,14.51447,5,0.24552
L991V1,50.07329,14.51444,6,0.25817
L991V1,50.07319,14.51441,7,0.26988
L991V1,50.07308,14.51437,8,0.28158
L991V1,50.07298,14.51435,9,0.29298
L991V1,50.07288,14.51433,10,0.30438
L991V1,50.07278,14.51431,11,0.31579
L991V1,50.07269,14.51429,12,0.32623
```

Obrázek 2.1: Ukázka CSV formátu ze souboru shapes.txt

V GTFS datasetu je určité množství povinných a nepovinných CSV souborů v textové podobě.

Tučně zvýrazněné pole jsou požadované či podmíněně vyžadované. Ostatní pole jsou volitelné.

<sup>2</sup>kolekce dat, která by měla být publikována na permanentní URL adrese



Obrázek 2.2: Diagram GTFS datasetu

### 3 Pražská Integrovaná doprava (PID)

Pražská integrovaná doprava (PID) je dopravní systém, zahrnující metro, tramvaje, železnici, městské a příměstské autobusové linky, lanovou dráhu na Petřín a některé přívozy. Tento systém je postupně integrován společnými přepravními a tarifními podmínkami a jednotným dopravním řešením včetně koordinace jízdních řádů. [5]



Obrázek 3.1: Logo Pražské integrované dopravy [5]

## 4 Postup

### 4.1 Aktualizace pluginu do verze 1.0.0

Po dokončení pluginu v předmětu Free Software GIS měl plugin veškerou základní funkcionalitu, kterou měl mít. Tou bylo načítání GTFS ZIP souboru do QGISu, rozbalení ZIP souboru, načtení CSV souborů do geodatabázového kontajneru GeoPackage, vytvoření vektorových vrstev pro soubory *stops.txt* a *shapes.txt*, obarvení vektorové vrty *shapes.txt* a vložení do CSV souborů a vrstev do *layer tree*.

Avšak pro uvedení do "ostrého" provozu musel být plugin více uživatelsky přívětivější. Proto byl veškerý proces přesunut na pozadí, aby celý QGIS program "nezamrzal" a mohla se při jeho výpočtech provádět i jiné akce. To se provedlo díky python třídě *QgsTask* a jejím metodám, které byly zděděny z této třídy. [6]

Pro zobrazování procesu během výpočtu byla použita třída *QProgressBar* a její metody. Zobrazení postupu bylo implementováno do lišty zpráv QGISu spolu s chybovými hláškami.



Obrázek 4.1: ProgressBar v liště QGISu

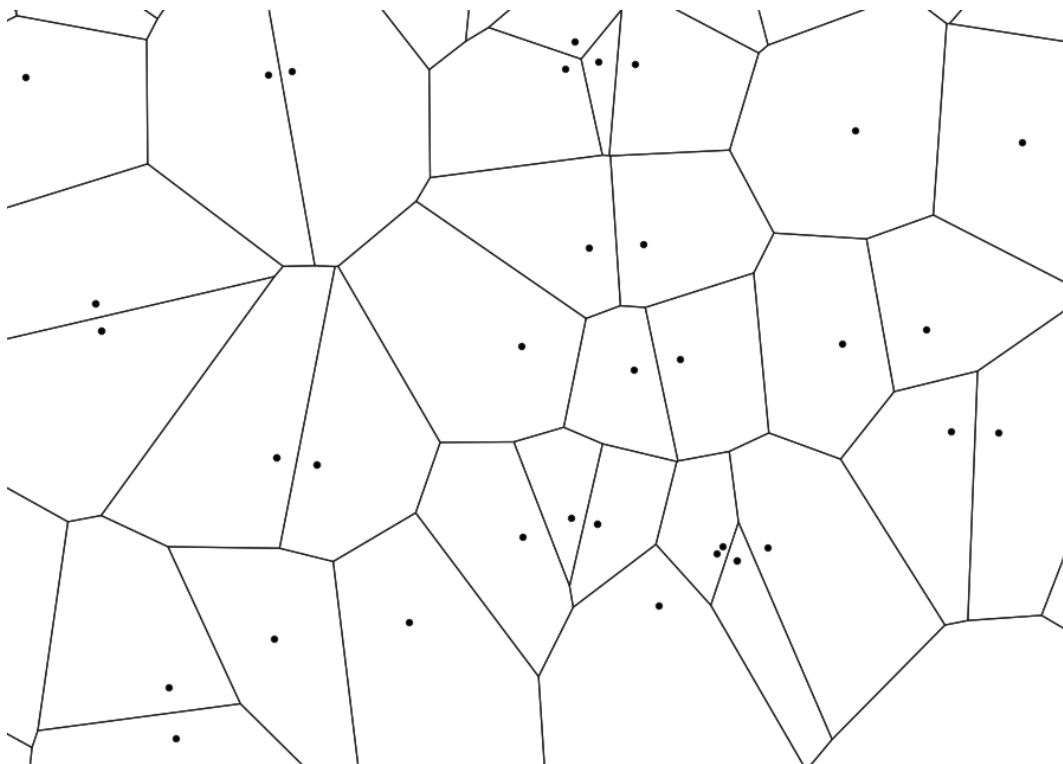
### 4.2 Postup 1 - pomocí nástrojů QGIS

GTFS obsahuje povinný CSV soubor *stops.txt*. Tento soubor obsahuje mimo jiných polí také pole *zone\_id*, které bylo při tvorbě tarifních pásem klíčové. Pole *zone\_id* má datový typ *string* a znamená, ve kterém tarifním pásmu daná zastávka leží. Verze GTFS Loaderu 1.0.0 soubor *stops.txt* převádí do vektorové vrstvy ve formě bodů. Z této bodové vrstvy byla vytvořena vektorová vrstva Voroného diagramů nástrojem *Voroného polygony* v programu QGIS.

Pro použití QGIS nástrojů v pythoním skriptu je potřeba importovat modul *processing*, který má funkci *run*, do které se vkládají dva parametry. První parametr je ID nástroje ve formě *stringu* a druhý je *slovník* vstupních parametrů. Vstupní parametry se lze dozvědět z QGIS dokumentace. [7]

Vstupem do tohoto nástroje je vektorová vrstva *stops* a výstupem jsou právě Voroného polygony.

Voroného diagramy, někdy pod názvy jako Voroného teselace, Voroného dekompozice, Thiessenovy polygony nebo Dirichletova teselace, podle definice představují rozklad množiny bodů  $P$  na  $n$  uzavřených či otevřených oblastí  $V(p) = \{V(p_1), V(p_2), \dots, V(p_n)\}$  takových, že každý bod  $q$  náležící množině  $V(p_i)$  je blíže k bodu  $p_i$  než k jakémukoliv bodu  $p_j$  náležící množině  $P$ . [8]



Obrázek 4.2: Voroného polygony

### Vlastnosti Voroného diagramů

- Voronoi diagram  $V(P)$  je planárním grafem.
- Vrchol  $q$  Voronoi buňky  $\{V(p_i)\}$  je průnikem 3 hran, právě když je  $V(P)$  nede degenerovaný.
- Pokud  $p_i$  náležící  $H(P)$ , pak je  $V(p_i)$  otevřený.
- Pro každý bod  $p_i$  náležící  $P$  je  $V(p_i)$  konvexní.
- Bod  $p_i$  je nejbližším bodem bodu  $p$  jestliže  $p$  náleží  $\{V(p_i)\}$ .
- Každá strana  $q_i q_j$ ,  $i$  se nerovná  $j$ , je sdílena právě dvěma sousedními buňkami  $V(p_i)$ .

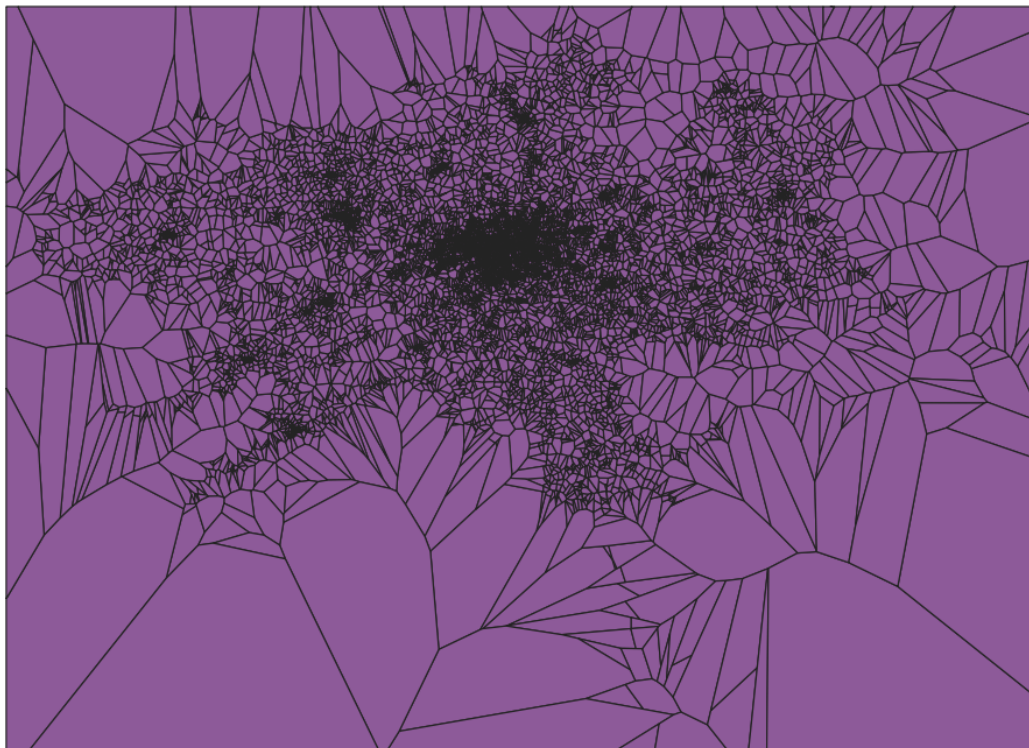
- Bod  $q$  je vrcholem  $V(p)$ , pokud existuje kružnice  $k(q, r)$  procházející třemi nebo více generátory  $p_i, p_j, p_k$ , a neobsahuje žádný další bod  $P$  (spojitost s  $DT(P)$ ).
- Kružnici  $k(q, r)$  označujeme jako největší prázdnou kružnici ze všech prázdných kružnic se středem v bodě  $q$ .
- Průměrné množství Voronoi hran ve Voronoi polygonu nepřekročí hodnotu 6.
- Vztah mezi počtem bodů  $n$ , počtem hran  $n_h$  a počtem trojúhelníků  $n_t$  teselace  $V(P)$ :

$$n_h \leq 3n - 6$$

$$n_t \leq 2n - 5$$

- Voronoi diagram  $V(P)$  představuje ortografickou projekci stěn mnohostěnu tvořeného průsečnicemi všech polorovin  $A_i$  do roviny  $xy$ .
- Necht bod  $p_i^*$  představuje ortografický průmět bodu  $p_i$  na povrch paraboloidu daného rovnicí:

$$z = x^2 + y^2$$



Obrázek 4.3: Voroného polygony pro všechny zastávky

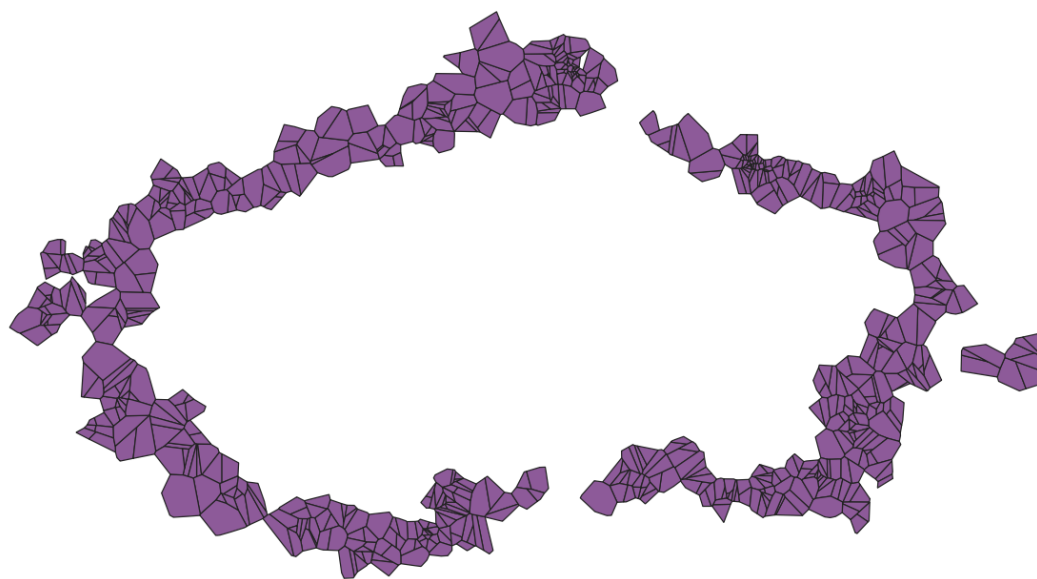


Pro každé tarifní pásmo byly vybrány zastávky pomocí třídy *QgsVectorLayer* a její metody *selectByExpression*, která vybírá prvky podle zadaného výrazu ve formě *string*. Znění zadaného výrazu je následující:

*zone\_id in "tarifní pásmo" and location\_type = 0*

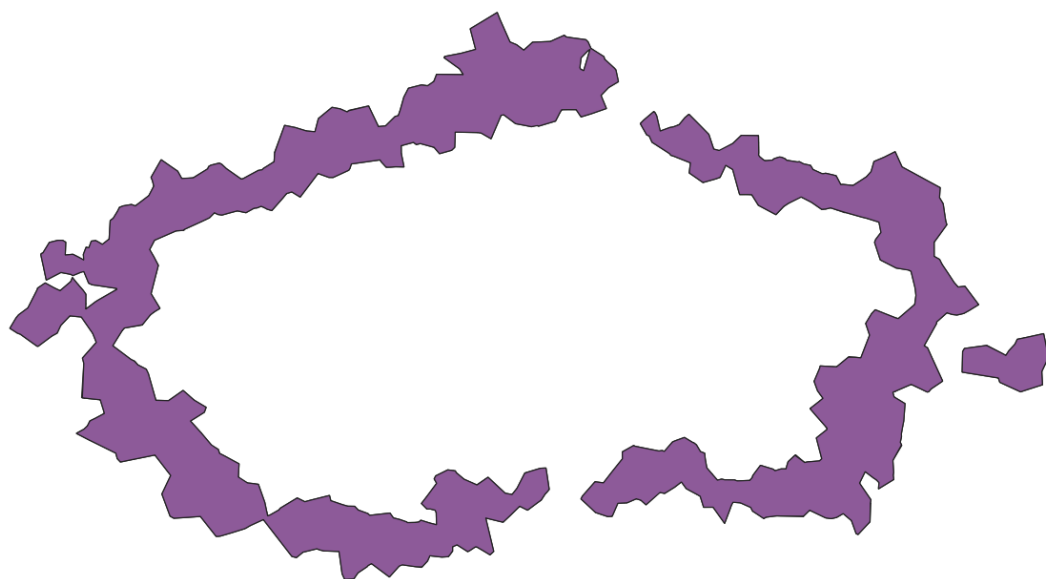
V zadávaném výrazu figuruje taktéž údaj o poli *location\_type*, což je typ lokace. Hodnota nuly (nebo prázdná hodnota) je právě lokace zastávky. Třída *QgsVectorLayer* představuje zároveň vektorovou vrstvu, která spravuje vektorové datové sady a která může být považována jako vstup do nástroje QGIS.

Pro vybrané zástavky bylo potřeba vybrat ty Voroného polygony, které svou pozicí dané zastávky protínaly. To bylo provedeno nástrojem *Select by location*, do kterého vstupovala vrstva vybraných zastávek a vrstva Voroného polygonů. Výsledkem tohoto nástroje byla vektorová vrstva vybraných Voroného polygonů. Jako příklad v obrázku zde budu uvádět tarifní pásmo 2.



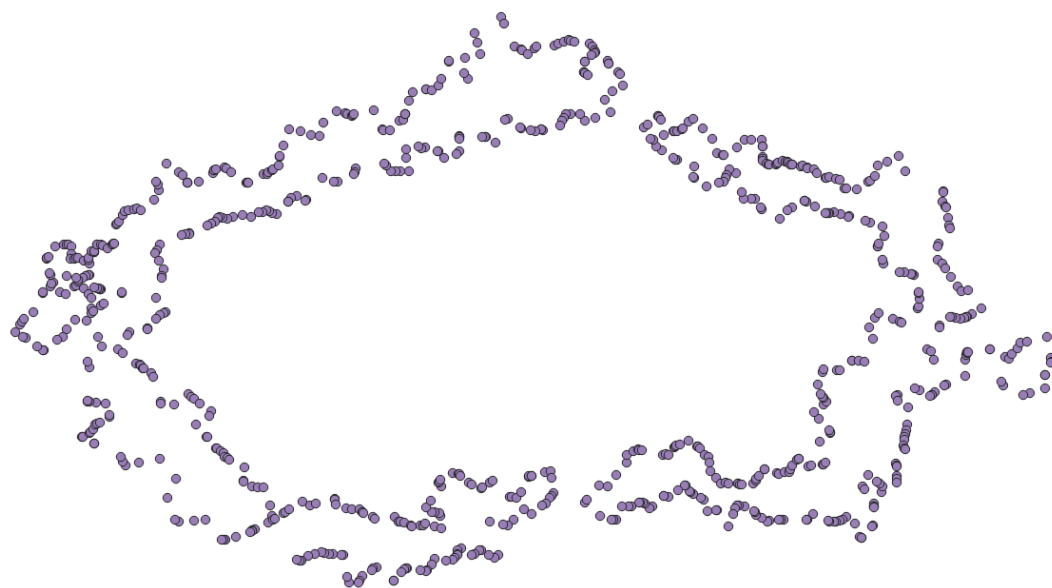
Obrázek 4.4: Vybrané Voroného polygony pro pásmo 2

Tyto polygony byly následně nástrojem *Dissolve* spojeny do jednoho společného polygonu. Vstupem tohoto nástroje byl výstup nástroje *Select by location* a výstupem byl polygon s jednou geometrií.



Obrázek 4.5: Výsledek nástroje Dissolve pro pásma 2

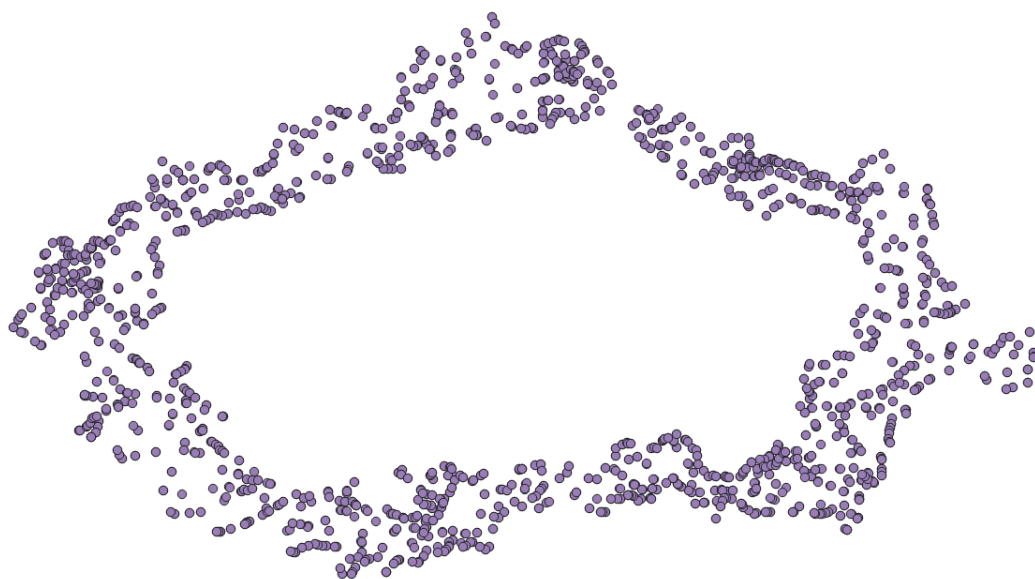
Ze spojených polygonů byla poté vygenerována vektorová vrstva bodů nástrojem *Extract vertices*, která představovala vrcholy spojených polygonů. Pro tuto vrstvu bylo vstupem výstup nástroje *Dissolve* a výstupem byl vektorová vrstva bodů doplněná o pole (mimo původních polí z vektorové vrstvy *stops*) jako *vertex\_index*, *vertex\_part*, *vertex\_part\_ring*, *distance* a *angle*. Hodnoty těchto polí avšak nebyly využity v dalším výpočtu.



Obrázek 4.6: Výsledek nástroje Extract vertices pro pásma 2

Dále byly s pomocí třídy *QgsVectorLayer* a její metody *selectByExpression* vybrány z původní vektorové vrstvy *stops* ty zastávky, které obsahovaly v poli *zone\_id* hodnotu tarifního pásma 1,2 nebo 2,3. Takové zastávky ležely na hranici pásma a polygon tarifního pásma měl skrz ně vézt hranici. Z vybraných zastávek byla vytvořena vlastní vektorová vrstva.

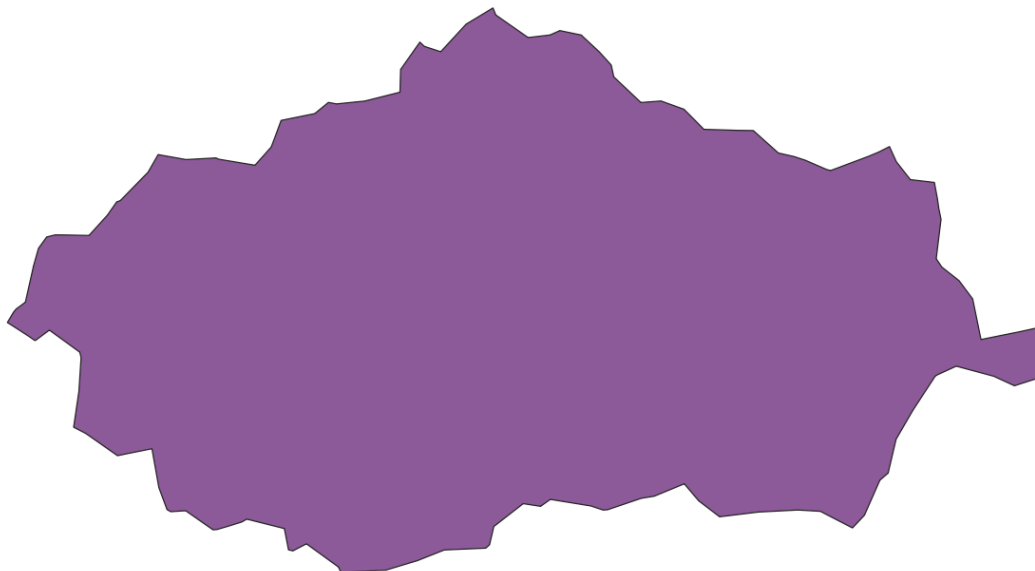
Následně byly spojeny vektorové vrstvy hraniční zastávek, zastávek uvnitř tarifního pásma 2 a bodů z výstupu nástroje *Extract vertices* nástrojem *Merge vector layers*. Tyto tři zmiňované vektorové vrstvy byly vstupem do tohoto nástroje a výstupem byla vektorová vrstva spojených bodů.



Obrázek 4.7: Spojené body třech vektorových vrstev pro pásmo 2

Ze spojených bodů byla vytvořena konkávní obálka. To bylo provedeno nástrojem Concave hull (alpha shapes), který počítá konkávní obálku z vstupních bodů. Vstupem tedy byla vrstva spojených bodů, což byl první parametr nástroje. Dalším parametrem byl *Práh (Threshold)* s datovým typem *číslo*, který byl volen od 0 do 1, kdy 0 znamenala maximum konkávní obálky a 1 konvexní obálky. Po několika testovacích spuštěních byla určena hodnota 0,09 jako nejlepší hodnotou pro tvorbu tarifních pásem. Dalším parametrem bylo *Povolení děr (Allow holes)* s datovým typem *boolean*, který byl nastaven na *False*. Posledním parametrem bylo Rozdělit

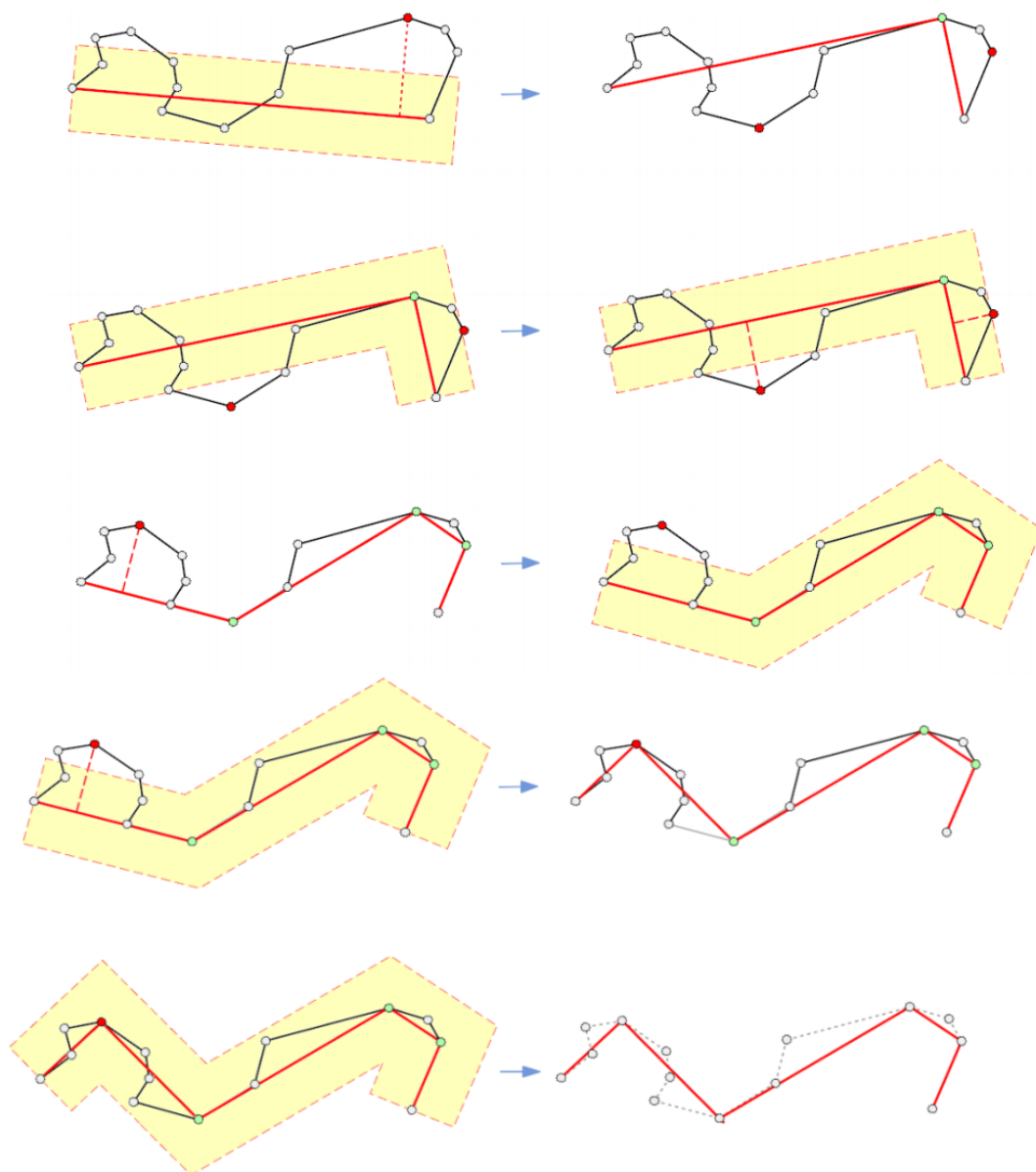
vícedílnou geometrii na jednotlivé části (Split multipart geometry into singlepart geometries) taktéž s datovým typem *boolean*, který byl nastaven na *True*. Výstupem byla polygonová vrstva konkávní obálky.



Obrázek 4.8: Konkávní obálka pro pásmo 2

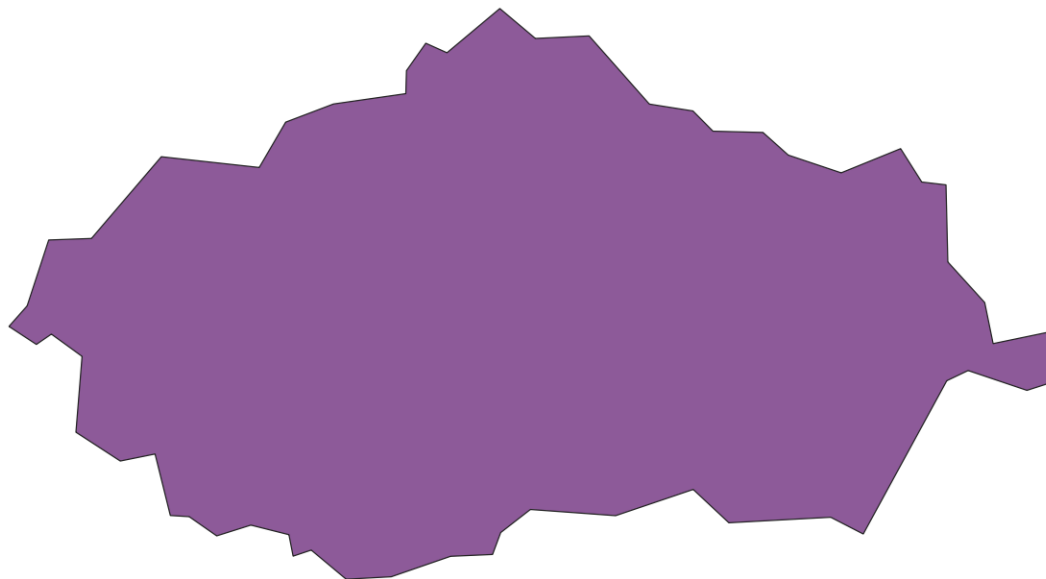
Z konkávní obálky byla pomocí nástroje Simplify "zjednodušena" geometrie polygonové vrstvy. Tento nástroj využívá tři druhy zjednodušení vektorové vrstvy: založené na vzdálenosti (algoritmus „Douglas-Peucker“), založené na ploše (algoritmus „Visvalingam“) a přichytávání geometrií k mřížce. Pro můj postup byl zvolen první volba zjednodušení, pomocí algoritmu „Douglas-Peucker“.

Algoritmus „Douglas-Peucker“ na rozdíl od ostatních algoritmů neodstraňuje vrcholy nesplňující geometrickou podmínku, ale přidává do ní postupně kritické body, které podmínku splňují. Tento algoritmus je jeden z nejlepších generalizačních algoritmů a je velmi často implementován v GIS software. [9]



Obrázek 4.9: Znázornění algoritmu „Douglas-Peucker“ [9]

Do nástroje Simplify byl vstupem výsledek nástroje Concave hull (alpha shapes), parametrem byl výběr algoritmu a nastavení hodnoty tolerance vzdálenosti. Výstupem byla polygonová vektorová vrstva.



Obrázek 4.10: Výsledek nástroje Simplify pro pásmo 2

Výstup nástroje Simplify byl vstupem do nástroje Smooth. Tento nástroj vyhlazuje geometrie liniové nebo polygonové vrstvy pomocí Chaikinova algoritmu.

Chaikin využil fixních poměrů na odříznutí rohů, takže byly všechny rozřezány stejně. Při matematickém zápisu postupuje Chaikinova metoda následovně: Dostaneme kontrolní polygon  $\{P_0, P_1, \dots, P_n\}$ , tento kontrolní polygon vylepšíme vygenerováním nové posloupnosti řídicích bodů

$$\{Q_0, R_0, Q_1, R_1, \dots, Q_{n-1}, R_{n-1}\}$$

kde každá nová dvojice bodů  $Q_i, R_i$  je třeba brát v poměru  $\frac{1}{4}$  a  $\frac{3}{4}$  mezi koncovými body segmentu čáry  $\overline{P_i P_{i+1}}$ .

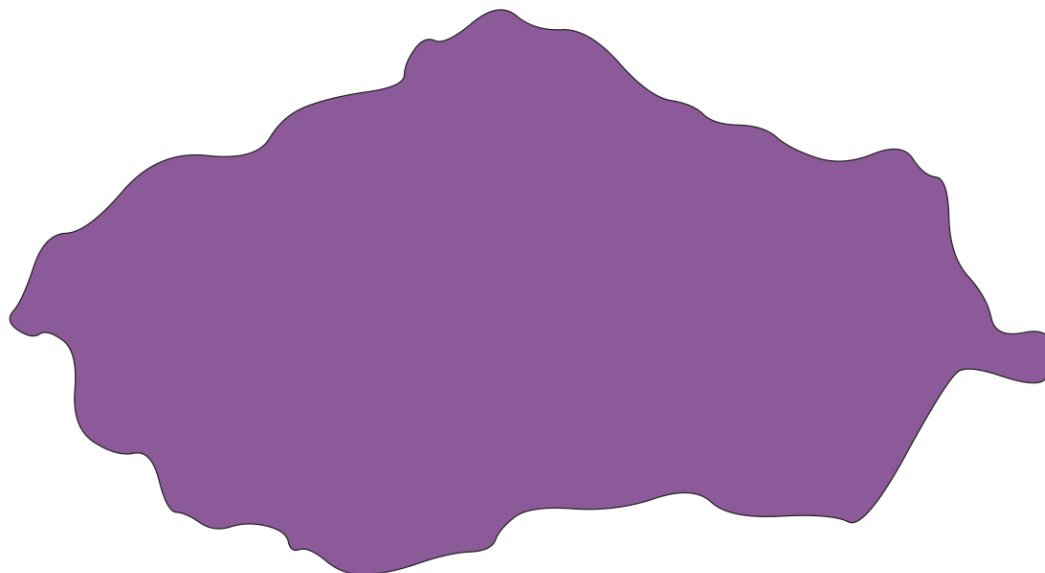
$$Q_i = \frac{3}{4}P_i + \frac{1}{4}P_{i+1}$$

$$R_i = \frac{1}{4}P_i + \frac{3}{4}P_{i+1}$$

Tyto 2 nové body lze považovat za nový řídicí polygon - vylepšení původního řídicího polygonu. [10]

U nástroje Smooth byly zvoleny tři parametry - počet iterací, offset a maximální úhel. Počet iterací znamená, kolik vyhlazovacích iterací bude použito pro každou geometrii. Hodnota počtu iterací byla nastavena na 10, což byla maximální volitelná hodnota (pro co největší hladkost). Parametr offset znamená, jak "těsně" vyhlazené

geometrie sledují původní geometrie. Zde byla ponechána výchozí hodnota 0,25. A poslední parametr maximálního úhlu lze použít k zabránění vyhlazení uzlů s velkými úhly. Zde byla také ponechána výchozí hodnota  $180^\circ$ . Výstupem nástroje byla polygonová vektorová vrstva.



Obrázek 4.11: Výsledek nástroje Smooth pro pásmo 2

# Závěr



## Odkazy

- [1] Esri. Shapefiles. [online], [cit. 2021-04-12]. Dostupné z: <https://doc.arcgis.com/en/arcgis-online/reference/shapefiles.htm>.
- [2] Opendata Praha. Pražská integrovaná doprava - tarifní pásma. [online], [cit. 2021-04-12]. Dostupné z: <https://opendata.praha.eu/dataset/ipr-prazska-integrovana-doprava--tarifni-pasma>.
- [3] Tomáš Bayer. Rovinné triangulace a jejich využití. [online], [cit. 2021-04-12]. Dostupné z: <https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk5.pdf>.
- [4] Google Developers. Gtfs static overview. [online], [cit. 2021-03-11]. Dostupné z: <https://developers.google.com/transit/gtfs/>.
- [5] Pražská integrovaná doprava. O systému|pražská integrovaná doprava. [online], [cit. 2021-03-11]. Dostupné z: <https://pid.cz/o-systemu/>.
- [6] QGIS. Třída qgtask. [online], [cit. 2021-03-25]. Dostupné z: <https://qgis.org/pyqgis/3.16/core/QgsTask.html#qgis.core.QgsTask>.
- [7] QGIS. Qgis dokumentace. [online], [cit. 2021-03-29]. Dostupné z: <https://docs.qgis.org/3.16/en/docs/index.html>.
- [8] Tomáš Bayer. Voronoi diagram. vlastnosti, použití, konstrukce. zobecněné voronoi diagramy. [online], [cit. 2021-03-23]. Dostupné z: <https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk6.pdf>.
- [9] Tomáš Bayer. Kartografické generalizační algoritmy generace bodových prvků/skupin. generalizace liniových prvků/skupin. generalizace plošných prvků/skupin. [online], [cit. 2021-04-08]. Dostupné z: <https://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk8.pdf>.
- [10] Kenneth I. Joy. Chaikin's algorithms for curves. [online], [cit. 2021-04-11]. Dostupné z: <https://www.cs.unc.edu/~dm/UNC/COMP258/LECTURES/Chaikins-Algorithm.pdf>.

## Seznam příloh