



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Baigiamasis bakalauro darbas

Kelių sankryžų klasifikacija

Atliko:

Laimonas Suchockas

parašas

Vadovas:

doc. Rimvydas Krasauskas

Vilnius
2018

Turinys

Sutartinis terminų žodynas	3
Santrauka	4
Summary	5
Ivydas	6
1. Motyvacija	7
1.1. Sankryžos ir spūstys	7
1.2. Naujos sankryžos	7
2. Analizė	8
2.1. Sankryžų charakteristikos	8
2.2. Dobilų tipo sankryža	9
2.3. Žemėlapių formatai	9
2.4. GIS	10
2.5. Formos formato kelių žemėlapių analizė	11
3. Tikslas ir užduotys	11
4. Galimi sprendimo būdai	12
4.1. Pirmasis variantas	12
4.2. Antrasis variantas	12
4.3. Trečiasis variantas	13
4.4. Pasirinktas variantas	14
5. Algoritmo kūrimo procesas	15
5.1. Pirmasis prototipas	15
5.1.1. Pirmoji algoritmo dalis	15
5.1.2. Antroji algoritmo dalis	16
5.2. Antrasis prototipas	18
5.2.1. Nuvažiavimų radimas	18
5.2.2. Kelių segmentų padalinimas	18
5.2.3. Likusių sankryžos segmentų radimas	20
6. Pasiektas rezultatas	21
7. Iškilusios problemos	22
8. Tobulinimo galimybės	23
Išvados ir rekomendacijos	24
Literatūros šaltiniai	25

Sutartinis terminų žodynas

Formos formatas(ang. shapefile)- vienas populiariausių formatų, kaip saugoti geografinę informaciją koordinuotoje erdvėje.

GIS - Geografinė Informacinė Sistema. Sistema skirta geografinės informacijos atvaizdavimui bei darbui koordinuotoje erdvėje.

Kelių atkarpos - pavieniai formos formato įrašai.

Kelių segmentai - kelių atkarpų rinkiniai priklausantys tam pačiam keliui.

OSM - Open Street Map. Nemokamų pasaulio žemėlapių atviros licenzijos projektas.

Sankryžos - šiame darbe, kelių sankryžos, be šviesoforų, kuriose keliai vienas kitą kerta tiltais arba tuneliais.

Sankryžų jungtys - kelių atkarpos esančios jungties tipo. Dažniausiai nuvažiavimai nuo tiltų arba užvažiavimai ant jų.

Susikirtimo taškai(ang. weaving point) - taškai keliuose, kur kelių juostos susikerta, išsiskiria arba susijungia.

QGIS - šiame bakalauro darbe naudojama GIS.

Santrauka

Šio baigiamojo bakalauro darbo tikslas - iš atvirai prieinamų kelių žemėlapių išskirti sankryžas ir suskirstyti jas į atskiras klases, pagal jų esminius požymius. Tam, kad būtų pasiektas šis tikslas, yra privalu atlikti atitinkamas užduotis. Visų pirma, reikalinga atlikti išsamią jau atliktų mokslinių darbų bei prieinamų įrankių analizę. Atlikus išsamią analizę ir nusprendus, kokie įrankiai bus naudojami darbo metu reikalinga, nuskaityti kelių žemėlapius ir iš jų išrinkti kelių segmentus, kurie vaizduoja tiltus bei tunelius einančius virš ar po kitais keliais. Suradus visus šiuos tiltus bei tunelius, reikia surasti visus kelio segmentus priklausančius atitinkamam tiltui ar tuneliui ir apjungti juos į rinkinius, kas žymi, jog tai yra viena ir ta pati sankryža. Turint šiuo būdu išskirtas sankryžas, yra patikrinamos kiekvienos sankryžos charakteristikos pagal kurias yra nustatoma kokio tipo yra tikrinama sankryža. Būtent šis procesas, visi susiję apmąstymai, priimti sprendimai bei jų priežastys ir paaiškinimai yra aprašyti šiame bakalauro rašto darbe.

Summary

Classification of Road Interchanges

The purpose of this bachelor work is to find, separate and classify road interchanges from open source road maps. In order to achieve this aim certain tasks need to be completed. First of all it is required to do an analysis of currently available works on the subject as well as find out about the possible tools to be used in order to achieve the mentioned final result. After the analysis is finished the following steps are to read the available road maps and inside of these road maps find all the bridges and tunnels. The next thing is to combine find road segments that belong to the found bridge or tunnel. After all the related road segments are found and put to bundles it is required to check characteristics of each of the road segment bundles. Based on these characteristics road interchanges are classified as one or another type of road interchange. Whole process that revolves around making this algorithm as well as the thoughts that lead to certain actions and decisions are written in this bachelor thesis.

Iyadas

Kelių tiesimui ir priežiūrai investuojama labai daug. Viena svarbiausių vietų kelių tiesimo darbuose yra sankryžos, kuriose vienas iš kelių kerta kitą per tiltą ar tunelį ir kurios nėra reguliuojamos šviesoforais. Būtent tokiom sankryžom, kur yra daug viadukų skiriama labai daug lėšų. Tačiau nemažą dalį šių sankryžų galima supaprastinti ir tokiu būdu sumažinti reikiamų investicijų kiekį. Tačiau norint tai padaryti reikia žinoti, kokios sankryžos jau yra pastatytos. Būtent tam tikslui ir buvo pradėtas algoritmo kūrimas šio baigiamojo bakalauro darbo metu.

Siekiamas tikslas šio bakalauro metu buvo sukurti algoritmą, kuris iš kelių žemėlapių išrinktų ir suklasifikuotų kelių sankryžas pagal jų tipus. Sankryžos, šiame darbe, yra tos sankryžos, kurios nėra reguliuojamos šviesoforais ir yra sudarytos iš tiltų bei susijungiančių ar išsiskiriančių kelių. Atlikti darbą šia tema buvo pasirinkta, nes neegzistuoja jokio tipo duomenų sistema, kurioje galima pažiūrėti esamas sankryžas bei pakeisti jau esamą sankryžą į kitokio tipo sankryžą. Turint skaitmeninėje erdvėje laisvai redaguojamus sankryžų žemėlapius galima suprojektuoti pigesnę bei saugesnę sankryžą, nes pasaulyje yra sankryžų, kurių tiltų kiekis gali būti sumažintas nepadidinant konfliktinių taškų kiekio.

Šio baigiamojo bakalauro darbo metu buvo rašomas algoritmas, kuris QGIS (šio darbo metu naudota viena iš laisvai prieinamų geografinių informacinių sistemų) aplinkoje, išrenka kelių atkarpas, kurios laikomos viadukais ar tuneliais, iš pasirinkto formos formatu išsaugoto žemėlapiu parsiusito iš OSM (OpenStreetMap atviro kodo projekto) tinklalapio. Sekanti algoritmo dalies užduotis yra surasti sankryžos jungčių atkarpas, kurios yra sankryžų nuvažiavimai į kitus sankryžos kelius. Suradus sankryžos kelių jungtis algoritmas turi padalinti kitas kelio atkarpas į smulkesnes, nes naudojamuose kelių žemėlapiuose trūksta vientisumo. Sėkmingai sutvarkius kelių atkarpas tam, kad būtų tiksliau surastos sankryžos, algoritmas gali pagal ankstesnių algoritmo dalių išvesties failus surasti, kur iš tiesų yra sankryžos.

Šis kuriamas algoritmas dirba su labai dideliais kiekiais duomenų, dėl šios priežasties bakalauro kūrimo metu buvo pasirinkta testavimams naudoti tik Lietuvos kelių žemėlapi tam, kad sumažinti įrašų kiekį nuo milijardų iki milijonų. Tai žymiai sutrumpina algoritmo veikimo laiką testuojant, kas yra ypač naudinga kūrimo proceso metu, kai yra norima pamatyti ar tinkamai veikia algoritmas. Tačiau, kadangi didžioji dalis kelių vaizduojamų žemėlapyje, net ir sąlyginai nedidelėje šalyje, visiškai nesudaro įtakos sankryžų klasifikavimui, bakalauro darbo vykdymo pradžioje didelė dalis žemėlapyje esančių kelių įrašų buvo ištrinta dėl tos pačios priežasties, jog būtų sumažintas algoritmo veikimo laikas. Vieni iš ištrintų kelių tipų pavyzdžių būtų: dviračių takai, pėsčiųjų takai, miško keliai, žvyrkeliai bei kiti keliai, kurie neturi reikšmės kuriamam algoritmui. Ištrinus šiuos kelius Lietuvos kelių žemėlapiu įrašų kiekis sumažėjo vos iki kelių dešimčių tūkstančių įrašų, kas palyginus su pradiniu kiekiu duomenų yra labai nedaug ir būtent tai padėjo projekto metu sutaupyti nemažai laiko, kuris buvo skirtas algoritmo kūrimui.

Kuriant šį algoritmą iškilo nemažai sunkumų, kurie stabdė algoritmo kūrimą, nuo tokių, kurių buvo išvengta, nes jos buvo pastebėtos prieš šioms problemoms atsirandant, iki tokių, kurios buvo pastebėtos tik testuojant algoritmą. Iš anksto pavyko išvengti problemų, kurios neleistų sėkmingai atlikti sankryžų radimo. Iš anksto rastų problemų buvo išvengta pakeičiant algoritmo sprendimo būdus. O problemos, kurios iš anksto nebuvo numatytos, labai stipriai paveikė algoritmo sprendimą, tačiau sprendimas ir joms buvo sugalvotas, bet nesėkmingai įgyvendintas. Bakalauro darbo pabaigoje pagrindinė dalis tikslo buvo pasiekta, sankryžos iš naudojamų kelių žemėlapių yra randamos, belieka patikrinti šių sankryžų charakteristikas tam, kad jos būtų suklasifikuotos.

1. Motyvacija

Ši tema pasirinkta dėl to, jog nėra duomenų sistemos, kurioje būtų aprašytos pasaulyje esančios sankryžos. Vienas iš trūkumų, neegzistuojant tokiai duomenų sistemai, yra tai, jog žmogus sugalvojęs naują sankryžos tipą negali prisiimti nuopelnų, nes iš tiesų tokio tipo sankryža jau gali būti, kur nors sugalvota ir pastatyta. Tačiau dar didesnis ir žymiai svarbesnis privalumas esant tokiai duomenų sistemai ir žinant, kokios sankryžos egzistuoja yra tas, jog pažiūrėjus į jau egzistuojančią sankryžą galima ją supaprastinti sumažinus viadukų skaičių, o kelių realiai nekeičiant. Tai žymiai sumažintų išlaidas kelių statymui.

Surinkti informaciją apie egzistuojančias sankryžas būtų galima ir rankiniu būdu naudojantis šalių žemėlapiais, tačiau tai yra labai neefektyvus būdas atlikti darbą. Didžiausias minusas yra, kad atlikus šį darbą būtų labai sudėtinga sužiūrėti ir visada užpildyti šią sistemą, kai bet kurioje pasaulio vietoje yra pastatoma nauja sankryža. Šią problemą panaikintų automatinis sankryžų klasifikavimo algoritmas. Esant tokiam algoritmui, pastačius naują sankryžą ir jos informacijai atsiradus žemėlapyje tereiktų įjungti sankryžų klasifikavimo algoritmą ir jis surastų naująją sankryžą. Tačiau šiuo metu yra atlikta labai nedidelis kiekis su šia tema susijusių mokslinių darbų. Būtent ši priežastis ir yra viena pagrindinių motyvacinių paskatų, dėl kurių yra atliekamas šis baigiamasis bakalauro darbas.

1.1. Sankryžos ir spūstys

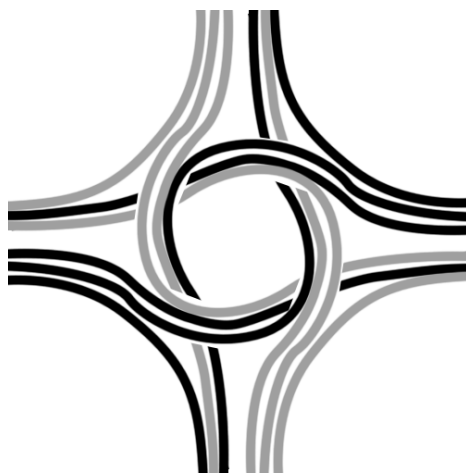
Šio baigiamojo bakalauro darbo metu yra kreipiamas dėmesys tik į sankryžas, kurios sudarytos iš kelių aukštų todėl, kad sankryžose, kuriose visi keliai susikerta viename aukštyje, eismas yra labai intensyvus. Kadangi transporto priemonių kiekis vis didėja miestuose ir visuose keliuose yra veiksminga įrengti kelių aukštų sankryžas tam, kad padidinti transporto priemonių pralaidumą ir pagerinti susisiekimą. Tačiau šių sankryžų įrengimas gali kainuoti milijonus eurų. Todėl yra svarbu, kad kelių aukštų sankryžos ne tik išspręstų didelių transporto spūsčių problemas, bet kad ir jų įrengimas kainuotų, kaip galima pigiau.

Tačiau yra ir tokių sankryžų, kurios yra sudarytos iš keleto aukštų ir yra ganėtinai neoptimalios kainos prasme. Tokiose sankryžose galima sumažinti aukštų skaičių ar šiek tiek praplėsti pačią sankryžą realiai nepakeičiant kelių susikirtimo taškų ir realiai išlaikant tokią pačią sankryžos struktūrą. Statant būtent tokias sankryžas galima sutaupyti lėšas ir už tą pačią pinigų sumą pastatyti, kartais netgi dvi sankryžas skirtingose vietose ir susisiekimas keliuose ženkliai padidėtų, nes būtų sumažinamos transporto spūstys.

1.2. Naujos sankryžos

Esant sistemai, kurioje būtų suklasifikuotos pasaulio sankryžos, sugalvojus kažkokios sankryžos patobulinimą arba, visišką pakeitimą tam, kad būtų sumažintos transporto spūstys ar tam, kad būtų panaikinti kelių susikirtimo taškai ir padidintas kelių saugumas, būtų galima pakeisti visas šio tipo sankryžas nepraleidžiant nei vienos. Tokiu atveju būtų paprasta surasti ir patobulinti visas problematiškas sankryžas nesijaudinant, kad bus praleista nors viena keičiamo tipo sankryža.

Sankryžų, kurios yra labai mažai žinomos, iš tiesų yra pakankamai. O iš tų sankryžų, kurios sumažintų ar netgi visai panaikintų kelių juostų susijungimo taškus teoriškai



1 pav. Pinavijos sankryžos schema

sukurtų yra nemažai [6], tačiau jos yra dar kol kas nepastatytos arba yra pastatyta kažkur pasaulyje vienintelė. Pavyzdžiui, Pinavijos sankryža yra keturių kelių ir dviejų aukštų sankryža, kur pačioje sankryžoje nėra nei vieno kelio susikirtimo taško [5]. Tai yra pasiekama dėl unikalios kelių išsidėstymo, kai keliai vienas kitą pervažiuoja tiltu ar tuneliu, o visas eismas vyksta ratu. Turint sąrašą visų esančių sankryžų tipų bei jų vietų, galima ganėtinai nesunkiai surasti, kuriose vietose yra galimybė pritaikyti naujai sukurtas sankryžas, tokiu būdu palengvinant darbą ir panaikinant rankinį tinkamų vietų ieškojimą.

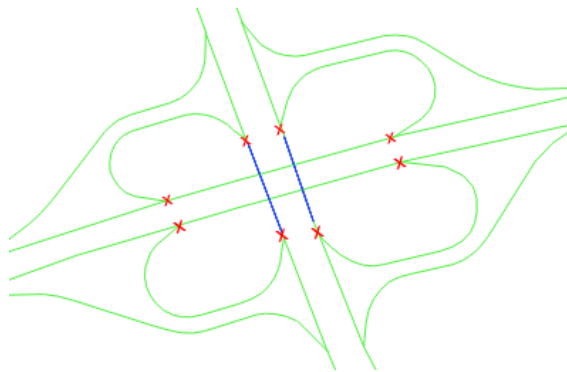
2. Analizė

Baigiamojo bakalauro darbo pradžioje atliktos analizės metu buvo tiriami moksliniai darbai, egzistuojančių sistemų dokumentacijos, vadovėliai, praktiškai analizuojamos esančios GIS. Tačiau, kaip jau ir minėta 1 skyriuje, yra labai nedidelis kiekis atliktų mokslinių darbų šia tema. Tačiau, nors ir ganėtinai nedaug, bet pakankamai plati atrasta informacija padėjo šio baigiamojo bakalauro darbo atlikimo metu. Šiame skyriuje yra pristatoma apie pačias sankryžas, jų charakteristikas, plačiau yra pristatoma dobilo tipo sankryža, nes į šio tipo sankryžos suradimą ir išskyrimą buvo fokusuojamasi šio baigiamojo darbo metu. Taip pat šiame analizės skyriuje yra pristatomos technologijos, jų veikimo principai, suteikiamos galimybės bei kokie pagrindiniai duomenys ir koku būdu yra saugojama informacija apie kelių žemėlapius.

2.1. Sankryžų charakteristikos

Šiame bakalauro darbo ataskaitos poskyriuje yra aprašoma apie pačias sankryžas bei pagal kokias charakteristikas viena sankryža skiriasi nuo kitos. Taigi, pirmiausia, visos sankryžos turi dvi charakteristikas, aukštų bei susikirtimo taškų kiekius. Sankryžoms esant kelių aukštų yra galimybė keliams kirsti vienas kitą nestabdant eismo einančio šiais keliais. Tai yra didelis privalumas, visų pirma, dėl to, jog važiuojantiems šiuo keliu yra sutaupomas laikas, tuo pačiu sumažinama gamtos tarša, nes nėra stovima vienoje vietoje ir deginama daugiau kuro nei reikalinga.

Tačiau didžiausias privalumas yra tas, jog automobilių važiavimo trajektorijoms nesikertant yra mažesnė tikimybė, jog įvyks eismo įvykis. Tačiau, vien tiltų kiekio padidinimas nepadaro sankryžos geresnė. Yra labai svarbus pačių tiltų pozicionavimas. Tiesiog pastačius tiltą, kad vienas kelias kirstų kitą atsiranda užduotis, pastatyti nuvažiavimus, kaip iš vieno kelio patekti į kitą. Vienas paprasčiausių variantų yra tiesiog padaryti nuvažiavimus, tačiau problema yra norint padaryti nuvažiuoti į kairiąją pusę, kai eismas vyksta dešiniąją kelio pusę. Paprasta išeitis yra tiesiog už tilto, kelių susikirtimo, padaryti nuvažiavimus, kurie įvažiuotų į kertamą kelią (žiūrėti 2 pav.). Tačiau iš to pačio taško kelyje į kurią įvažiuojama, kiti automobiliai turi atlikti tokį patį manevrą. Tai reiškia, jog tame pačiame taške yra ir įvažiavimas į kelią ir išvažiavimas ir tai sukelia konfliktines situacijas. Būtent šie konflikto taškai ir yra vieni iš susikirtimo taškų į kuriuos svarbu atsižvelgti sankryžose. Tačiau čia ne vienintelis galimas konfliktinio taško tipas. Be prieš tai paminėtojo konfliktinio susikirtimo taško, vietose, kur kelias išsiskiria į kelis kelius arba, kai keli keliai susijungia į vieną, irgi yra vadinami konfliktiniais ir susikirtimo taškai.



2 pav. Dobilo tipo sankryža

2.2. Dobilo tipo sankryža

Dobilo tipo sankryža (žiūrėti 2 pav.) yra sankryža, kuri yra sudaryta mažiausiai iš vieno tilto (dažniausiai dviejų) ir keturių susikirtimo taškų. Šis sankryžos tipas pasaulyje pasitaiko tikriausiai dažniausiai, nes ši sankryža suteikia galimybę dviem greitkeliams kirsti vienas kitą, nestabdant eismo. Taip pat didelis dobilo tipo sankryžos privalumas yra tai, kad šiose sankryžose nebūtina sumažinti važiavimo greitį iki ganėtinai nedidelio.

Tačiau, kaip ir kiekviena sankryža bei visi kiti išradimai, dobilo tipo sankryža turi savų trūkumų. Pirmiausia dobilo tipo sankryžos reikalauja ganėtinai didelio ploto, ypač jas norint pritaikyti didesniuose greitkeluose. Norint šią sankryžą pritaikyti didelio greičio greitkeliams, kelių nuvažiavimai turi būti ganėtinai didelio diametro tam, kad būtų galima juo važiuoti dideliu greičiu. Jeigu nuvažiavimai, bus siauro diametro, mašinos turės važiuoti lėčiau ir tai prives prie spūsčių greitkelyje. Tačiau dar didesnė charakterizuojanti vieta kiekvienoje sankryžoje, taip pat, kaip ir dobilo tipo sankryžoje yra susikirtimo taškai, kur dar yra vadinami konflikto taškais. Šiuose taškuose toje pačioje kelio juostoje susilieja įvažiuojantys ir išvažiuojantys srautai. Visų pirma šiuose susikirtimo taškuose yra didesnė tikimybė įvykti eismo įvykiui. Bet tai nėra vienintelis trūkumas, kurį suteikia susikirtimo taškai. Susikirtimo taškai apriboja galinčių pasukti juostų kiekį.

2.3. Žemėlapių formatai

Vienas populiariausių formatų, kurį pasinaudojus galima saugoti geolokacinę informaciją, yra formos formatai [2]. Šiuo formatu saugomi duomenys nėra saugomi viename faile. Tam, kad būtų galima pilnai atidaryti šį formatą, yra privaloma turėti bent 3 skirtingo plėtinio failus. Kiekviena iš šių failų laikoma skirtinga informacija apie tą patį vektorinį žemėlapi. Pagrindiniai formos formato failų papildiniai yra .shp, .shx ir .dbf. .shp yra pagrindinis failas nusakantis visus geometrinius žemėlapių objektų duomenis; .shx failas nusako geometrinių duomenų indeksavimą, kad būtų galima lengvai pereiti per duomenis; .dbf atributų failas, kuris yra aprašytas dBase IV formatu ir, kuriame yra saugomi duomenys apie kiekvieną geometrinę įrašą. Be šitų trijų failų formos formatai gali turėti ir daugiau failų, kurie aprašo skirtingus dalykus apie atvaizduojamą žemėlapi. Pavyzdžiui, .prj failas yra naudojamas koodrinačių sistemai ir jos informacijos saugojimui atviru tekstu gerai žinomo teksto (ang. well-known text) formatu.

Vienas iš pagrindinių atributų, kurie yra aprašyti formos formate, yra kelio tipas. Keli iš kelio

	id	osm_id	code	fclass	name	ref	oneway	maxspeed	layer	bridge	tunnel
4	7	4853642	5114	secondary	ventaragio g.	A3;A15;101	B	40	0	F	F
5	10	4869653	5114	secondary	Feliks Vaitkaus g.		F	40	0	F	F
6	12	4869656	5113	primary	Liepkalnio g.	A3;A15	B	50	0	F	F
7	13	4869753	5113	primary	Pylimo g.		F	50	0	F	F
8	17	4870069	5114	secondary	J. Jasinskio g.		B	0	0	F	F
9	18	4870096	5113	primary	Saltonio?ki? ?iedas		B	50	0	F	F
10	19	4870109	5112	trunk	Gele?inio Vilko g.	A2;A14	F	60	1	T	F
11	21	4870121	5113	primary			B	50	0	F	F
12	32	4870916	5132	trunk_link		A4;A16	F	40	0	F	F
13	33	4870955	5111	motorway		A1	F	120	0	F	F
14	34	4871038	5112	trunk	Gele?inio Vilko g.	A1;A4;A16	F	60	-1	F	T
15	35	4871059	5132	trunk_link			F	50	0	F	F

3 pav. Dalies formos formatu aprašytų įrašų atributų lentelė

tipo pavyzdžių galėtų būti, tarptautiniai keliai bei jungiamieji keliai. Tai ganėtinai trumpos kelio atkarpos, jungiančios kelius. Pavyzdžiui, kelio nuvažiavimas nuo tilto iki po tiltu einančio kelio yra jungiamasis kelias. Dar labai svarbus atributas aprašytas formos formate yra kelio sluoksnius. Ant žemės nutiesti keliai yra nuliniame sluoksnyje. Jeigu virš šio kelio yra nutiestas viadukas - tai viadukas formos formato žemėlapyje yra pažymėtas esantis pirmame sluoksnyje. Jeigu po keliu eina požeminis kelias, tai jis yra pažymėtas esantis minus pirmame sluoksnyje.

Formos formato failuose gali būti saugomi 3 tipų geometrinių objektai. Daugiakampiais poligonais žymimi pastatai, miškai, teritorijos ir t.t. Taškais galima žymėti kažkokių tai konkrečių objektus. Tačiau svarbiausias šiam bakalauro darbui geometrinių objektas, kuriuo ir yra atvaizduojami keliai, yra vektorinės linijos. Šios linijos turi pradžios ir pabaigos taškus bei kryptį. Kelio krypties žinojimas yra labai naudingas kelių sankryžų nustatymui, nes pagal tai galima nustatyti ar kelių atkarpos tarp tilto ir pagrindinio kelio išilieja į pagrindinį kelią ar tai vis dėlto išvažiavimas. Pagal šiuos duomenis vienas iš esamų atributų nusako, kuria kryptimi vyksta eismas kelio atkarpoje. Šio atributo reikšmės gali būti „F“, „T“ ir „B“. „F“ nusako, jog eismas šia kelio atkarpa vyksta ta pačia kryptimi, kuria eina linija formos formato faile; „T“ reiškia, jog eismas vyksta priešinga kryptimi nei eina atkarpa; „B“ reiškia, jog eismas juda abiejomis kryptimis.

Šio projekto kūrimo proceso vykdymui iš pradžių buvo galvota iš karto testuoti kuriamas algoritmo dalis su viso pasaulio kelių žemėlapiu. Tačiau panagrinėjus šiek tiek išsamiau, buvo nuspręsta šiam bakalauro darbui naudoti tik Lietuvos kelių žemėlapi. Tai palengvina ir žymiai sutrumpina algoritmo veikimo laiką, o tai ypač algoritmo kūrimo metu yra labai svarbu norint efektyviai dirbti prie šio bakalauro darbo. Būtent Lietuvos žemėlapis buvo pasirinktas dėl to, jog šie keliai yra neblogai žinomi ir iškilus kažkokiems tai neaiškumams žemėlapyje, jeigu tokių atsirastų, būtų galima nesunkiai patikrinti, kaip viena ar kita sankryža atrodo iš tiesų. Sėkmingai sukūrus kelių sankryžų klasifikavimo algoritmą, jį būtų galima be problemų pritaikyti, bet kokiam kitam žemėlapiui. Tai ir yra didysis kuriamo algoritmo privalumas.

2.4. GIS

GIS (Geografinė Informacinė Sistema) yra informacinė sistema, kuri yra skirta skaitmeninių duomenų skaitymui ir atvaizdavimui koordinatų sistemoje. Pagrindinė bet kokios GIS funkcija yra geografinės informacijos vizualizacija. Dauguma šių informacinių sistemų ne tik atvaizduoja įvairių formatų žemėlapius, bet ir suteikia galimybę vartotojui patogiai redaguoti ar kurti žemėlapius. Kai kurios geografinės informacinės sistemos suteikia daug didesnę funkcionalumą. Kai kurios sistemos suteikia labai didelį pasirinkimą funkcijų informacijos apdorojimui. Pavyzdžiui, linijų susikirtimo taškų radimas. Ši funkcija yra viena naudingiausių funkcijų šio baigiamajam bakalauriniam darbui.

Tačiau viena naudingiausių ypatybių, kurias suteikia kai kurios GIS, yra galimybė kurti savo skriptus informacijos apdorojimui. Būtent kuriant šiuos skriptus ir buvo praleista didžioji dalis šio bakalauro darbo atlikimo laiko. Šio darbo pradžioje buvo svarbu pasirinkti tinkamą GIS, kurią naudojančią ir bus atliekamas šis bakalauro darbas. Yra labai nemažas pasirinkimas geografinių informacinių sistemų, kurias buvo galima rinktis šio bakalauro darbo atlikimui, pavyzdžiui, ArcGIS, Google Earth, Maptitude, GRASS, QGIS ir daugiau. Nemaža šių sistemų dalis yra sukurta komerciniais tikslais, būtent dėl šios priežasties buvo bakalauro darbas pradėtas daryti su GRASS GIS. Pradėjus gilintis į šią informacinę sistemą buvo suprata, jog šios sistemos suteikiamų funkcionalumų neužtenka šio bakalauro darbo atlikimui. Funkcija, kurios trūko buvo, kelių susikirtimų paieška. Dėl šios funkcijos ir buvo pereita prie QGIS naudojimo [1].

2.5. Formos formato kelių žemėlapių analizė

Šio baigiamojo bakalauro darbo atlikimo metu tam, kad būtų išsiaiškinta, kaip ir kokiais būdais reikia nagrinėti turimus duomenis, buvo nagrinėjami formos formatu saugomi kelių žemėlapiai, apie kuriuos plačiau aprašoma 2.3 poskyriuje. Vis tobulinant ir pildant kuriamą algoritmą buvo išsiaiškinami nauji dalykai, kurie padėjo dar labiau patobulinti algoritmą.

Pagrindiniai pastebėjimai, dėl formos formatu aprašytų kelių yra tokie, jog visų pirma kelių atkarpų aprašyme nėra vientisumo. Tai reiškia, jog, palyginimui paėmus kelią tarp dviejų sankryžų, kuris yra sudarytas iš dviejų vienpusio eismo kelių, vienas iš kelių gali būti sudarytas iš vienos atkarpos, o tuo pačiu metu, kitas gali būti sudarytas iš daugiau nei vienos atkarpos, nors abu galėtų būti sudaryti iš to paties skaičiaus atkarpų, nes jie yra identiški keliai. Toks duomenų nevientisumas stipriai apsunkina darbą. Pavyzdžiui, vietoj to, kad suradus atitinkamas atkarpas, būtų galima jas tiesiog įterpti į naują failą, kuriame yra aprašytos vien sankryžos, reikia netinkamas atkarpas padalinti į daugiau dalių, o kitas apjungti į vieną atkarpą tam, kad būtų vėliau aiškiau, kaip ištiesų yra sudarytos sankryžos.

3. Tikslas ir užduotys

Atlikus pradinę darbų analizę buvo suformuluotos šio bakalauro darbo tikslo pasiekimui reikalingos atlikti užduotys. Šio baigiamojo bakalauro darbo tikslas yra sukurti algoritmą, kuris perrenka kelių žemėlapius ir išrenka bei išrūšiuoja kelių sankryžas. Tam, kad būtų pasiektas šis tikslas, reikia atlikti šias pagrindines užduotis:

1. Išsiaiškinti, kaip yra saugomi kelių žemėlapiai.
2. Nuskaityti kelių žemėlapio duomenis.
3. Išrinkti kur yra kelio atkarpos einančios virš ar po kitomis kelio atkarpomis.
4. Apjungti išrinktas kelio atkarpas į rinkinius, sankryžas.
5. Sugrupuoti rinkinius pagal atitinkamą kelių išsidėstymą.

Baigiamajam bakalauro darbui einant į pabaigą tikslas buvo šiek tiek supaprastintas ir išsigrynintas į išrinkimą vieno tipo sankryžų. Šiuo atveju buvo pasirinkta dobilo tipo sankryžos, nes jos vienos populiariausios. Sprendimas supaprastinti darbo tikslą buvo priimtas dėl tos priežasties, jog buvo sugaišta daug daugiau laiko prototipavimui nei buvo sugaišta daugiau laiko nei buvo pradžioje numatyta, dėl vis išskylančių klaidų ar išsiaiškintos naujos informacijos apie naudojamus įrankius. Tačiau nors ir tikslas šiek tiek supaprastėjo, iš tiesų, jis išliko toks pat. Sėkmingai sukūrus algoritmą iš naudojamų žemėlapių surasti dobilo tipo sankryžą, pagrindiniai elementai ir funkcijos sankryžų atradimui bus sukurti ir norint surasti, kitas sankryžas. Tam reikia pritaikyti naudojamą algoritmą, kad būtų randami skirtingai išsidėsčiusios kelio atkarpos. Norint suklasifikuoti visas sankryžas reikėtų išnagrinėti bei algoritme aprašyti kiekvieną galimą sankryžą. Tai yra ganėtinai pasikartojantis darbas, nes realiai pačios sankryžos radimas ir išskyrimas nuo viso žemėlapio yra pagrindinė dalis šio bakalauro darbo. Tolesnis darbas sankryžų klasifikavimui pagal jų tipą yra kiekvienam sankryžos tipui pasikartojantis, nes vienintelis skirtumas tarp skirtingų sankryžų tipų yra algoritme aprašytos jų charakteristikos.

Darbo vykdymo metu buvo išsiaiškinta, jog naudojamuose OSM tipo kelių žemėlapiuose labai trūksta vientisumo. Pavyzdžiui, vietose, kur galėtų būti identiški kelio segmentai, iš tiesų, šie segmentai gan stipriai skiriasi. Viename segmente gali būti, jog jis sudarytas iš vienos kelio atkarpos, o kitas toks pat kelio segmentas esantis šalia yra sudarytas iš kelių atkarpų. Apie tai plačiau galima

pasiskaityti 2.5 poskyriuje. Būtent dėl šio vientisumo trūkumo atsirado nedidelė papildoma užduotis reikiamas atkarpos padalinti į smulkesnes, bei atrastoje sankryžoje atskiras atkarpos apjungti į mažiau tam, kad būtų aiškiau atvaizduojamos sankryžos.

4. Galimi sprendimo būdai

Šiame poskyriuje yra aprašomi keletas galimų būdų, kuriais yra galima pasiekti užsibrėžtą šio bakalauro darbo tikslą - iš kelių žemėlapių išrinkti ir suklasifikuoti kelių sankryžas pagal jų tipą. Taip pat šiame skyriuje aprašomi, šių galimų sprendimų privalumai, bei galimi trūkumai. Pristatomas pasirinktas sprendimo būdas, jo privalumai palyginus su kitais būdais ir keletas, numatytų problemų su kuriomis tikriausia bus susidurta.

4.1. Pirmasis variantas

Pats pirmasis sankryžų radimo būdas buvo sugalvotas vos išsiaiškinus, kaip yra saugomi kelių žemėlapiai formos formato failuose ir kokie duomenys aprašo šias kelių atkarpas. Šis sprendimo būdas yra labai primityvus, nes kuriant algoritmą pagal šį sprendimo būdą, algoritmas turėtų būti parašytas visiškai nuo nulio. Tai reiškia, jog bakalauro tikslas taptų ne sukurti algoritmą, kuris surastų sankryžas, o sukurti sistemą, kuri skaitytų, apdorotų bei atvaizduotų kelių žemėlapius. Tačiau tai yra visiškai nenaudinga, nes tokių sistemų yra daug įvairiausių, nuo nemokamai prieinamų iki komercinių, kurios yra naudojamos valstybinėse institucijose dirbančiose su valstybiniais keliais. Būtent tai ir yra didžiausias šio sprendimo trūkumas, tačiau nei vienas iš esamų įrankių nesuranda kelių sankryžų. Taigi naudojantis šiuo sprendimo būdu būtų tiesiog matematiškai brėžiami visi keliai ir ieškoma jų susikirtimų. Tam, kad paprasčiau būtų nuskaitomi šie keliai, formos formatu aprašyti kelių žemėlapiai turėtų būti paversti į .csv formato failus. Tai yra grynuoju tekstu užrašyti bei kableliais atskirti duomenys. Tai reiškia, jog kiekvienas kelio atkarpos įrašas yra aprašytas naujoje eilutėje, o duomenys apie kiekvieną atkarpą yra rašomi į vieną eilutę ir atskirti kableliais.

Kaip ir jau minėta pagrindinis ir didžiausias šio sprendimo būdo trūkumas yra tai, jog darant bakalauro darbą pagal šį variantą būtų atliekama labai daug nereikalingo darbo, kuris ir taip jau yra padarytas, nes vietoj to, kad pasinaudoti jau esamais įrankiais būtų stengiamasi iš naujo sukurti šiuos įrankius. Dar vienas ganėtinai didelis trūkumas kuriant algoritmą pagal šį sprendimo būdą yra tas, jog kelių duomenys yra aprašyti pagal Žemės geografines koordinatas. Tai yra didelis trūkumas, nes kelių koordinatės turi labai daug skaičių po kablelio. Skaičiuojant tokius didelius skaičius atsiranda paklaidos, kurios ieškant kelių sankryžų gali sukelti labai didelių netikslumų, nes dauguma sankryžų palyginus su Žemės koordinatinių sistema yra labai smulkios.

4.2. Antrasis variantas

Išanalizavus šiek tiek daugiau informacijos apie esamus įrankius, jų panaudojimą bei galimybes, buvo sugalvotas kita sprendimo būdas, kaip galima pasiekti užsibrėžtą šio darbo tikslą. Pagal šį sprendimo būdą reikia pasirinkti kažkokį tai kintamąjį, kuris nusakytų įsivaizduojamo apskritimo spindulį. Kintamasis gali keistis priklausomai nuo tikrinamos vietovės dydžio, nes sankryžos gali būti įvairiausių dydžių. Turint šį kintamąjį visas tikrinamas žemėlapis iš eilės yra patikrinamas atskiromis dalimis šių įsivaizduojamų apskritimų ribose. Kiekviename iš įsivaizduojamų apskritimų algoritmą turėtų eiti apskritimo lanko ribomis, iki tol, kol apskritimo lankas susikerta su kažkokiu žemėlapyje esančiu keliu. Kai yra pasiekiamas susikertantis kelias, algoritmas tikrina šį kelią, einant tikrinamo kelio žemėlapyje nutiesta linija, iki tol, kol šis kelias vėl susikerta su įsivaizduojamo apskritimo lanku. Jeigu tikrinamas kelias nors kurioje vietoje susikirto su kitu keliu, tai reiškia, jog šio įsivaizduojamo apskritimo ribose turėtų būti sankryža, tačiau čia kelių tikrinimas dar nesibaigia. Šitokiu būdu yra stengiamasi praeiti pro visą įsivaizduojamo apskritimo lanko ilgį ir tokiu

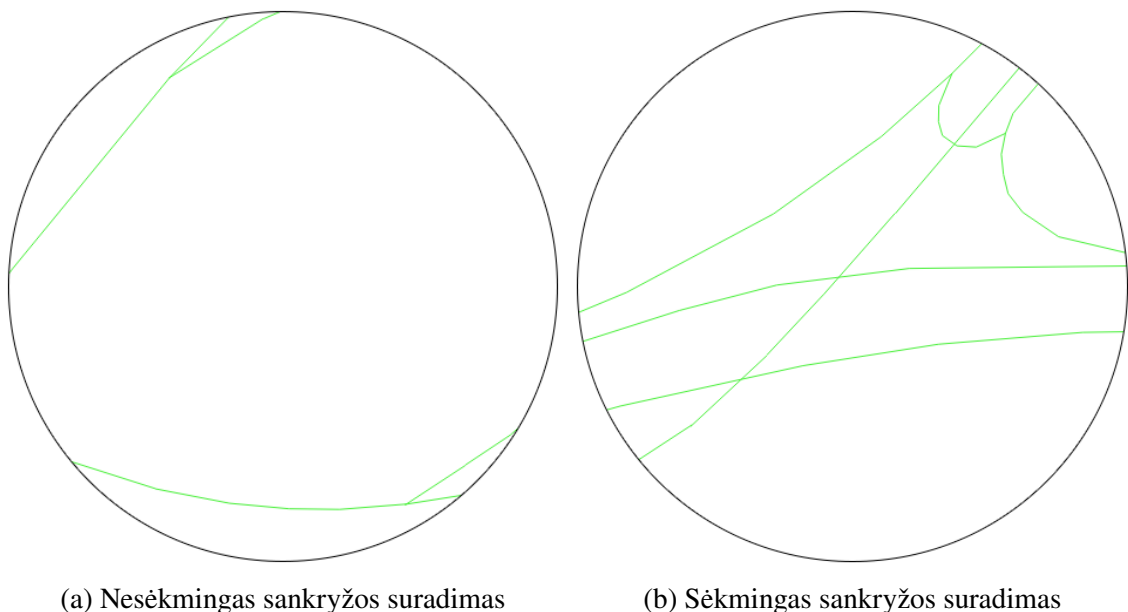
būdu patikrinti visus apskritimo ribose kelius. Tokiu būdu atskiromis dalimis yra patikrinamas visas kelių žemėlapis ir yra surandamos visos kelių sankryžos.

Didžiausias šio sprendimo būdo privalumas yra tai, jog algoritmas ne labai priklauso nuo to, kaip yra sudaryti tikrinami keliai bei kokie duomenys yra saugomi šiose kelių atkarpose. Tikrinamos kelio atkarpos gali būti sudarytos iš daugelio atkarpų ar iš vienos ir šis algoritmo sprendimo būdas turėtų sėkmingai surasti kelių sankryžas. Tačiau šis sprendimo būdas turi ir ganėtinai didelių trūkumų. Pirmiausia tai, jog šis būdas yra dalinai sėkmingas dėl tos priežasties, kad jis tik suranda sankryžas, bet tam, kad patikrinti kiekvienos sankryžos tipą reikalinga naudoti kažkokį papildomą algoritmą, kuris jau būtų ne šio sprendimo dalis. Tačiau didžiausios šio sprendimo problemos yra tos, jog tokio pačio tipo sankryžos gali būti įvairiausių dydžių, kas sukelia problemų parenkant apskritimo spindulį. Jeigu bus paimtas per mažas spindulys tai nebus surasta sankryža arba bus užregistruota, jog sankryža yra mažesnė nei iš tiesų yra. O jeigu spindulys bus per didelis, tai apskritime esančios kelios sankryžos gali būti paimtos, kaip viena ir ta pati sankryža. Su tuo yra susijusi ir antroji didelė šio sprendimo būdo problema. Kaip atskirti, kur sankryža apskritimo ribose prasideda bei užsibaigia ir kur yra sankryžos ribos. Norint tai patikrinti vis tiek reikalinga patikrinti, kaip ir kokie keliai yra atrastoje sankryžoje. Tokiu atveju didysis šio sprendimo būdo privalumas, kaip ir išnyksta, nes vis tiek yra privaloma patikrinti, iš kokių dalių yra sudaryta surasta sankryža ir kokie duomenys yra saugomi šiose kelių atkarpose.

4.3. Trečiasis variantas

Dar vienas galimas sprendimo būdas, kurį naudojant galima pasiekti šio bakalauro darbo tikslą, iš dalies panaudoja aukščiau paminėto sprendimo idėją, jog reikia tam tikrose ribose patikrinti kelių geometrines charakteristikas. Naudojant šį sprendimo būdą, algoritmas pirmiausia surastų besikertančius kelius, kur vienas iš kelių kerta kitą tiltu arba tuneliu. Naudojant šį surastą susikirtimo tašką, kuris būtų panaudotas išivaizduojamo apskritimo nubrėžimui, kurio centras būtų šis kelių susikirtimas. Šio apskritimo ribose būtų tikrinami keliai, bet priešingai, nei prieš tai aprašytame sprendime, algoritmas tikrintų kiekvieną kelią einantį nuo atrasto tilto ar tunelio. Jeigu tikrinama kelio atkarpa išivaizduojamo apskritimo ribose geometriškai apsisuka ir kerta centre esantį tiltą ar tunelį, galima pasakyti, jog šioje vietoje yra sankryža.

Naudojant šį sprendimo būdą būtų iš dalies panaikinta problema, su kuria būtų susidurta įgy-



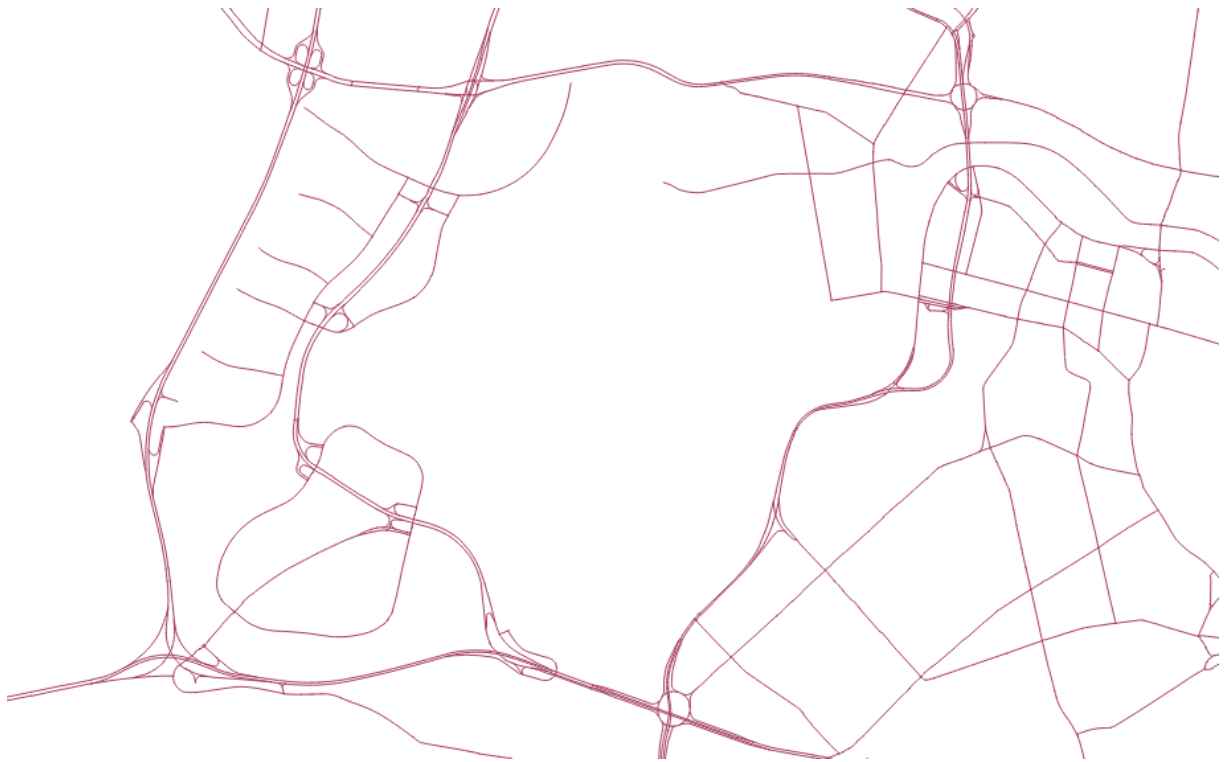
4 pav. Antrojo sprendimo būdo pavyzdžiai

vendinant aukščiau aprašytąjį sprendimo būdą. Tai yra, būtent kuriose vietose tikrinti kelių išsidėstymą, kas sutrumpintų algoritmo veikimo laikotarpį, nes nereikia patikrinti visiškai viso žemėlapių. Taip pat šis sprendimas iš dalies panaikintų problemą, jog yra sudėtinga nustatyti, kurios kelio atkarpos priklauso nagrinėjamai sankryžai. Naudojant šį sprendimo būdą visi algoritmo skaičiavimai yra pradami sankryžos ribose, nes tiltai ir tuneliai yra dalis iš sankryžos nusakančių charakteristikų, kas reiškia, jog algoritmui telieka nustatyti tik, kurioje vietoje pasibaigia sankryža. Visos kelio atkarpos nuo tikrinamo tilto ar tunelio, jeigu tikrinamoje vietoje yra sankryža, priklauso šiai sankryžai. Vienas didžiausių trūkumų, kurį turi šis sprendimo būdas, yra, kaip ir prieš tai aprašytame sprendime, tai, jog nėra tikslaus būdo, kaip nustatyti reikiamo dydžio ribas, kuriose yra tikrinamos kelių atkarpos, o šios ribos yra reikalingos tam, kad algoritmas teisingai surastų tik sankryžas, kaip atskiras sankryžas. Jeigu nebūtų ribų, tai algoritmas galėtų sankryžai priskirti daugiau kelio atkarpų bei vingių, nei iš tiesų sudaro sankryžą.

4.4. Pasirinktas variantas

Tačiau nei pagal vieną iš aukščiau aprašytų variantų nebuvo nuspręsta šio bakalauro darbo metu kurti algoritmą, kuris suranda ir suklasifikuoja kelių sankryžas. Naudojant naują sprendimo būdą sankryžos yra randamos nagrinėjant kiekvieną kelio atkarpą ir apie ją saugomus duomenis. Algoritmas parašytas pagal šį sprendimo būdą pirmiausia suranda kelio atkarpas, kurios kertasi su kitomis kelio atkarpomis ir yra nutiestos skirtinguose aukštuose. Tai reiškia, jog atrasta atkarpa yra tiltas arba tunelis. Nuo šios kelio atkarpos algoritmas patikrina, kokio tipo atkarpos susikerta su tikrinamu tiltu ar tuneliu ir kokio tipo atkarpos susikerta su šiomis atkarpomis. Pagal duomenis, kuriais yra aprašytos kelių atkarpos, galima nusakyti labai daug apie kelią, kuris yra sudarytas iš nagrinėjamų atkarpų.

Didelis privalumas ieškant sankryžų šiuo būdu yra tas, jog pakankamai išsamiai išnagrinėjus sankryžas yra įmanoma sėkmingai surasti sankryžas netikrinant kelio geometrinių charakteristikų, tai yra, netikrinant, kokia kreive eina kelias. Dar vienas šio sprendimo būdo privalumas yra tas, jog nereikia tikrinti visų kelio atkarpų tam tikrose ribose, kurių diametrą sunku nusakyti. Užtenka, kaip jau prieš tai minėta, patikrinti nuo tilto ar tunelio per tam tikrą kiekį atkarpų nutolusias kelių atkarpas. Teoriškai, net jeigu tokio pačio tipo sankryža yra didesnė už kitą tokio pačio tipo sankryžą, jų sudėtis vis tiek turi būti struktūriškai tokia pati. Tačiau šio algoritmo kūrimo metu buvo atrasta problemų, kurios žymiai sulėtino darbą. Tai vėliau įrodė, jog šis būdas gal ir nėra pats geriausias ir galima dar patobulinti, tačiau šio bakalauro darbo vykdymo metu jau nebuvo likę tam pakankamai laiko. Tačiau kuriant algoritmą pagal šį sprendimo būdą atsirandančios problemos yra vis tiek smulkesnės, nei prieš tai aprašytuose sprendimuose. Pagrindinė šio sprendimo problema yra ta, jog šis sprendimas yra priklausomas nuo žemėlapiuose vaizduojamų kelių vientisumo. Tačiau tai nėra tokia didelė problema palyginus su prieš tai minėtais sprendimais. Naudojantis šiuo sprendimu sankryžos yra vis tiek surandamos sėkmingai, o nežinant, kokio dydžio vietovėje ieškoti sankryžos, yra žymiai sunkiau, o tuo pačiu, kelių nevientisumą galima lengviau ištaisyti.



5 pav. Dalis Vilniaus miesto kelių

5. Algoritmo kūrimo procesas

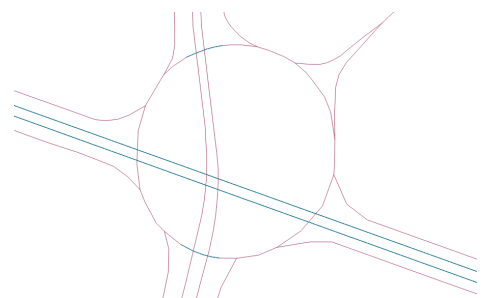
5.1. Pirmasis prototipas

5.1.1. Pirmoji algoritmo dalis

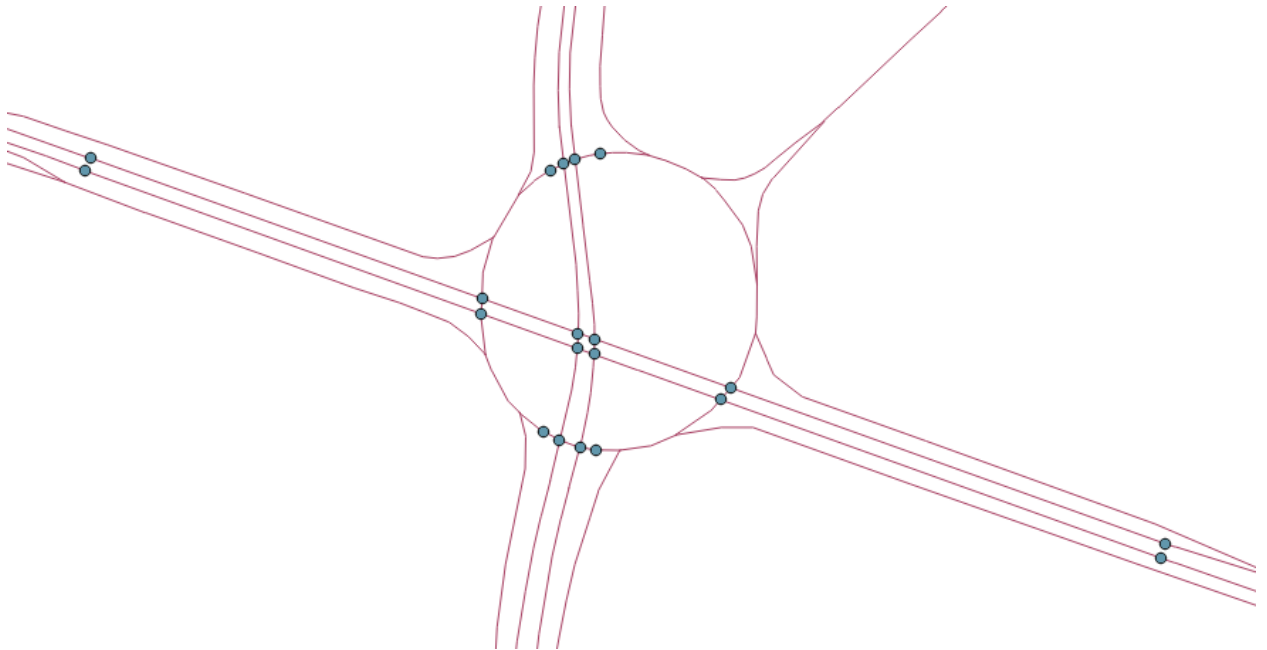
Pirmosios algoritmo dalies tikslas yra išrinkti kelių atkarpas, tiltus bei tunelius, kurie eina virš arba po kitomis kelio atkarpomis. Suradus tinkamą geografinės informacijos sistemą ir išnagrinėjus turimą informaciją apie kelius, buvo pradėtas šio algoritmo dalies kūrimo procesas. Pirmasis šios užduoties sprendimo būdas buvo išskirti visus kelių susikirtimo taškus perduodant naujiems taškams abiejų susikirtusių kelių, linijų atributus, tokiu būdu būtų lengvai galima nustatyti, kurie susikirtimo taškai žymi viadukus ar tunelius. Tačiau naudojama QGIS funkcija neleido to padaryti nerašant atskiros skripto informacijos perdavimui naujai kuriamam taškų sluoksniui. Ši funkcija su numatytais nustatymais sugeba tik suteikti vieną iš atributų stulpelių linijų susikirtimo taškams. Dėl šios priežasties buvo nuspręsta suteikti vieno iš linijų kelio sluoksnio atributą, o vėliau turint visus susikirtimo taškus iš taškų sluoksnio buvo panaikinti visi įrašai, kurie buvo pažymėti esantys nuliniame kelių sluoksnyje.

Tai padarius visi likę taškai žymėjo tik arba tiltus arba tunelius. Tačiau problema buvo ta, jog viename tilte buvo net keletas taškų, nes formos formatu saugomi žemėlapiai yra aprašyti nedideliais segmentais ir tilto pradžia bei pabaiga, taip pat yra skaitoma, kaip linijų susikirtimas. Tame pačiame koordinatų taške yra ir tiltų vaizduojančios linijos segmento dalis, ir su tuo tiltu sujungto kelio atkarpos dalis. Dėl šios priežasties reikėjo apjungti taškus į rinkinius tam, kad vieną tiltą žymėtų tik vienas įrašas. Tačiau čia ir iškilo pirmoji didelė problema.

Šių pavienių taškų apjungimui buvo pradėtas rašyti



6 pav. Tiltai žiedinėje sankryžoje



7 pav. Taškai, kuriuose susikerta kelių atkarpos, esančios skirtinguose aukštuose

skriptas, kuris išrinktų visų pirma visus pasikartojančius taškus. Kadangi su numatytaisiais nustatymais panaudotas linijų sankirtų funkcija sukuria po du taškus viename susikirtime. Šioje funkcijoje pagal numatytuosius nustatymus nėra žymimos koordinatės, kuriose jau yra užfiksuotas linijų susikirtimas [4]. Šitai buvo nuspręsta atlikti todėl, kad tolimesnė užduoties vykdymo sprendimo idėja buvo ta, jog kiekvienas taškas bus sulyginamas su visais kitais taškais. Bet ir šitos idėjos buvo atsisakyta, nes buvo sugalvotas mažiau resursų reikalaujantis sprendimo būdas.

Vietoj to, kad atrinkti kelių susikirtimo taškus ir juos paskui apjungti į kelius, yra tiesiog patogiau perrinkti visas kelių atkarpas ir iškart patikrinti ar tikrinama kelio atkarpa kertasi su kitu keliu ir ar jų kelių sluoksniai yra skirtingi. Šioms sąlygoms atitikus, pirmoji kelio atkarpa kartu su savo duomenimis yra nukopijuojama į naujai kuriamą atvaizdavimo sluoksnį. Tokiu būdu yra supaprastinamas darbas kompiuteriui, sutaupomas laikas bei galutiniai rezultatai išlaiko visus reikiamus duomenis apie save, kuriuos vėliau galima panaudoti tolimesnėse algoritmo dalyse.

Atrinktos kelių atkarpos yra išsaugojamos naujame formos formato faile su geometriniais duomenimis bei iš pradinių duomenų nukopijuotais identifikacijos duomenimis ir kelių sankryžai svarbiais atributais. Tokiu būdu yra paprasta naudoti šiuos duomenis norint vėliau juos panaudoti apjungiant reikiamas kelių atkarpas į sankryžas.

1 algoritmas. Pirmosios algoritmo dalies prototipo pseudo-kodas

Ivestis: *roadSegments* - formos formato failas (Kelių žemėlapis)

Išvestis: *newFile* - formos formato failas

- 1: Kiekvienas įrašas iš *roadSegmentMap* patikrinamas su visais kitais susikertančiais įrašais
 - 2: **if** Tikrinami įrašai nėra tame pačiame sluoksnyje **IR** tikrinamas įrašas nėra 0 sluoksnyje **then**
 - 3: pirmasis tikrinamas įrašas įrašomas į *newFile*
 - 4: **end if**
-

5.1.2. Antroji algoritmo dalis

Antroji algoritmo dalis sulygina kiekvieną įrašą iš pirmajame algoritme atrinktų kelio atkarpų su pradiniais duomenimis. Ši algoritmo dalis patikrina kiekvieną kelio atkarpą, kuri susikerta ar liečiasi su išskirtu tiltu ar tuneliu. Taip pat patikrina ar ši kelio atkarpa susikerta ar susiliečia su

kita kelio atkarpa, kuri yra pažymėta, kaip kelio jungtis. Jeigu kelio atkarpa yra kelio jungtis, tai reiškia, jog šis kelias yra nuvažiavimas. Tai galima teigti todėl, kad šis kelio segmentas yra šalia tilto ar tunelio. Tačiau vien šitos informacijos neužtenka tam, kad būtų galima pasakyti, jog šis kelio nuvažiavimas ir tiltas ar tunelis, su kuriuo tikriname, priklauso tai pačiai kelio sankryžai. Būtent dėl šios priežasties tolimesnė algoritmo užduotis yra patikrinti ar atrasta kelio jungtis susikerta ar liečiasi su kita kelio atkarpa, kuri eina po išskirtu tiltu ar virš tunelio, su kuriuo yra tikrinamos kelio atkarpos, ir ar ši kelio atkarpa yra būtent ta atkarpa, kuri yra pažymėta, einanti per, tiltą, su kuriuo yra tikrinamos visos kelio atkarpos. Jeigu ši paskutinioji kelio atkarpa atitinka šiuos kriterijus tai reiškia, jog visos šiame tikrinime naudotos kelio atkarpos, su tiltu ar viaduku susikertanti ar susiliečianti kelio atkarpa, ją liečiantis kelio jungtis bei kelio atkarpa einanti virš ar po tiltu ar viaduku priklauso tai pačiai sankryžai. Šių kelio segmentų geometriniai duomenys yra nukopijuojami į naują formos formato failą, kartu su savo identifikavimo duomenimis bei kitais aprašančiais kelio atkarpą duomenimis.

2 algoritmas. Antrosios algoritmo dalies prototipo pseudo-kodas

Įvestis: *roadSegments* - formos formato failas (Kelių žemėlapis), *bridges* - formos formato failas (Tiltų ir viadukų žemėlapis iš pirmojo algoritmo)

Išvestis: *newFile* - formos formato failas

- 1: kiekvieną įrašą iš *bridges* patikrinama su kiekvienu besikertančiu įrašu iš *roadSegments*
 - 2: antrasis tikrinamas įrašas yra patikrinamas su visais besikertančiais *roadSegments* įrašais
 - 3: **if** šis tikrinamas įrašas yra *jungtis* IR kerta kitą *roadSegments* įrašą, kuris kerta pirmąjį tiltą **then**
 - 4: į *newFile* įrašomos visos minėtosios kelio atkarpos
 - 5: **end if**
-

Turint šį naują failą, kuriame yra dalis kelio atkarpų priklausančių tai pačiai sankryžai, galima dirbti tik su tais keliais, kurie būtent ir priklauso nagrinėjamai sankryžai. Tačiau yra kelios problemos atlikus algoritmus iki šios ribos. Pirmiausia, tai, kad šiame naujajame formos formato faile nėra vis dėlto visos atkarpos priklausančios nagrinėjamoms sankryžoms. Tačiau tam, kad išspręsti šią problemą reikalinga šiek tiek sudėtingesnė algoritmų logika. Antroji problema, kurią galima išspręsti paprasčiau ir be itin sudėtingų algoritmų yra ta, jog naujajame formos formato faile, kuriame yra didžioji dalis sankryžos kelio atkarpų, yra daugiau nei po vieną identiškų įrašų. Tai reiškia, jog, toje pačioje geometrinėje linijoje yra keletas identiškų atributus turinčių įrašų. Tokia situacija susidaro dėl to, kad sankryžose dažnai nuo vienos kelio atkarpos prie susikertančios atkarpos, nepaisant eismo judėjimo krypties, galima prieiti daugiau nei vienu keliu. Tokiu būdu, vykdant anksčiau aprašytą algoritmą, kiekvieno ciklo metu, suradus skirtingus kelius, kai kurios atkarpos yra pridamos keletą kartų.

Tam, kad būtų išspręsta ši problema, buvo bandyta patobulinti anksčiau aprašytąjį algoritmą, kad jis patikrintų ar naujai kuriamame faile nėra jau pridėta identiško įrašo. Tačiau po keleto nesėkmingų bandymų šis variantas buvo atmestas. Kitas variantas išsprendžiantis identiškų įrašų buvimą buvo tai, kad ištiesų atrinkinėti, kad nebūtų identiškų variantų, net ir realiai nereikia. Tam galima visus identiškus įrašus apjungti į vieną, ir kartu apjungti į vieną įrašą visas besiliečiančias atkarpas, kurios ir taip priklauso vienai ir tai pačiai atkarpai. Toks ganėtinai kraštutinis variantas buvo priimtas todėl, kad buvo išsiaiškinta, jog formos formato failuose esantys duomenys nėra pakankamai vientisi. Pavyzdžiui, kai kuriose vietose kelio atkarpa nuo tilto tęsiasi iki artimiausio nuvažiavimo, tačiau, kai kuriose panašiose vietose, vietoj to, kad atkarpa užsibaigtų ties pirmuoju ar net antruoju nuvažiavimu, ši atkarpa tęsiasi žymiai toliau, nei iš tiesų užsibaigia sankryža. Būtent dėl šitos priežasties buvo nuspręsta, jog kelio atkarpas reikia perkirsti nustatytose vietose, kur prasideda sankryža. Tačiau variantas apjungti visas kelio atkarpas į vieną įrašą, sukelia ganėtinai nemažai nepatogumų stengiantis apskaičiuoti kelių atkarpų priklausomybes vieną nuo kitos. Daug

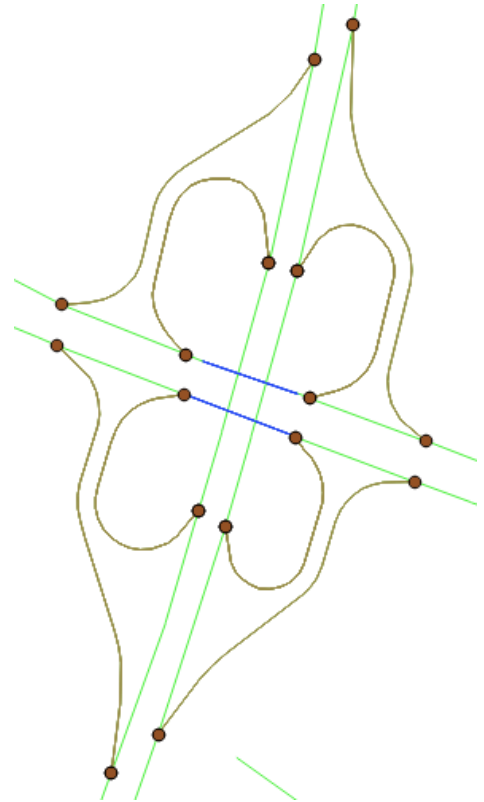
patogiau padaryti, kai šios turi atskirus aprašymus, nors ir reik iš naujo sukurti naujas kelio atkarpas, kurios neužimtų už sankryžos ribų, bei tokiu būdu suteiktų galimybę lengvai pakeisti vieną sankryžą kita.

5.2. Antrasis prototipas

5.2.1. Nuvažiavimų radimas

Antrąjį prototipą buvo nuspręsta kurti dėl tos priežasties, jog pirmojo prototipo kūrimo metu buvo surastos loginės klaidos, kurios neleistų vėliau pasiekti norimą rezultatą, šiuo atveju bakalauro darbo tikslą: išrinkti ir išrūšiuoti sankryžas. Pirmojo prototipo kūrimo metu buvo išsiaiškinta, jog formos formato failuose dobilo tipo sankryžų tiltus bei toliausiai nuo jų nutolusius, bet tai pačiai sankryžai priklausančius, kelių nuvažiavimus skiria dvejų kelių atkarpos.

Būtent šia ypatybe ir yra remiamasi antrajame prototipe. Šiame algoritme, panašiai, kaip ir pirmojo prototipo pirmojoje algoritmo dalyje, iš formos formato kelių žemėlapių yra surandamos kelių atkarpos, kurios yra tiltai arba tuneliai, kertantys kitas kelių atkarpas. Suradus šias atkarpas yra patikrinamos visos susikertančios bei susiliečiančios atkarpos. Jeigu šios atkarpos susikerta su atkarpomis, kurios yra pažymėtos kelių jungties atkarpa, susikirtimo taškas, kuriame susikerta tiltą ar viaduką kertanti ar liečianti ir jungties atkarpos, yra įrašomas į naują formos formato failą. Taip pat šiam taškui yra priskiriamas tilto ar tunelio, nuo kurio buvo pradėtas algoritmas, identifikacinis numeris tam, kad būtų galima surasti taškus, kurie priklauso tai pačiai sankryžai. Be naujojo susikirtimo taškų formos failo yra sukuriamas ir naujas kelių atkarpų failas. Ši algoritmo dalis į naująjį kelių atkarpų failą įrašo kelių atkarpas, kurios yra pažymėtos esančios jungties atkarpomis. Taip pat, kaip ir naujai sukurtame susikirtimo taškų faile, taip ir kelio jungčių faile, kiekviena atkarpa turi atributą, kuris nusako, kuriam tiltui ar tuneliui priklauso ši jungties atkarpa.



8 pav. Surastos sankryžos jungties atkarpos ir jų susikirtimo taškai su kitomis sankryžos atkarpomis

5.2.2. Kelių segmentų padalinimas

Kelio atkarpų padalinimas į smulkesnes atkarpas šio algoritmo kūrimo procese tikriausiai yra pati sudėtingiausia dalis, kurią privalu atlikti. Tai yra būtina atlikti dėl tos priežasties, jog kelių žemėlapiuose, kurie yra naudojami, trūksta vientisumo. Šis vientisumo trūkumas kelia įvairių problemų, nuo tiesiog neaiškaus grafinio sankryžos atvaizdavimo iki algoritmo pritaikymo problemų, kurioms esant algoritmo kūrimas tampa žymiai sudėtingesnis ir reikalauja visiškai kitokio sprendimo norint, jog būtų sėkmingai pasiektas šio darbo tikslas. Vienas iš galimų algoritmo veikimo būdų, kuriuo būtų galima surasti sankryžas, esant naudojamų žemėlapių nevientisumui, yra geometrinis kelių patikrinimas. Tai reiškia, jog pagal pasirinktą kažkokį faktorių patikrinti, tam tikru skersmeniu aplink kiekvieną tiltą bei tunelį, nuo jo einančius kelius. Jeigu esant šio apskritimo ribose, kur tiltas arba tunelis yra apskritimo centre, tikrinant kelią einantį nuo šio tilto ar tunelio

3 algoritmas. Sankryžos nuvažiavimo atkarpų radimas

Ivestis: *roadSegments* - formos formato failas (Kelių žemėlapis)

Išvestis: *newLinesFile* - formos formato failas (Jungties kelio atkarpos), *newPointsFile* - formos formato failas (Jungčių susikirtimo taškai su kitomis kelio atkarpomis)

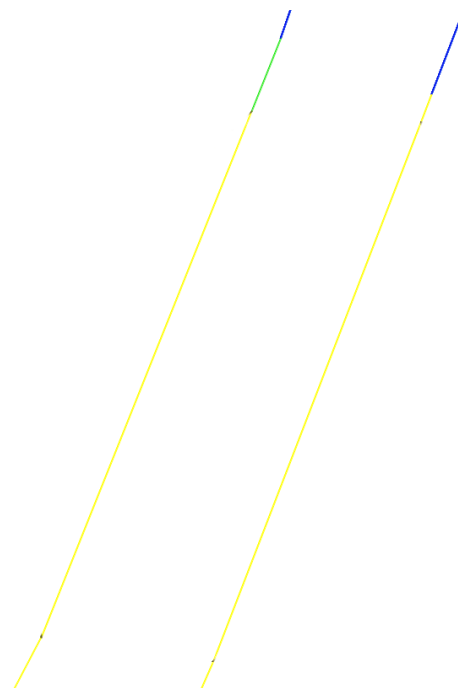
```
1: roadSegments faile yra surandamos tiltus ar tunelius žyminčios kelio atkarpos
2: surandamos atkarpos firstLine, kurios kertasi su rastu tiltu ar tuneliu
3: if firstLine kertasi su atkarpa, kuri yra jungtis then
4:   atkarpa, kuri yra jungtis yra įrašoma į newLinesFile
5:   firstLine susikirtimo taškas su atkarpa, kuri yra jungtis, įrašomas į newPointsFile
6: else
7:   rasta atkarpa, kuri nėra jungtis laikoma esanti secondLine, kurios kertasi su firstLine
8:   if secondLine kertasi su atkarpa, kuri yra jungtis then
9:     atkarpa, kuri yra jungtis yra įrašoma į newLinesFile
10:    secondLine susikirtimo taškas su atkarpa, kuri yra jungtis, įrašomas į newPointsFile
11:   end if
12: end if
```

galima vienu arba keliais kitais keliais sugrįžti atgal prie tikrinamo tilto ar tunelio. Iš to galima spręsti, jog šio apskritimo ribose yra sankryža.

Dar vienas iš galimų būdų, kurią galima būtų naudoti, norint sukurti sėkmingą sankryžas kelių žemėlapiuose surandantį algoritmą, galėtų būti labai panašus į prieš tai minėtąjį, kuriame yra tikrinami kelių geometriniai duomenys. Tačiau šiuo antruoju būdu būtų tikrinami keliai einantys ne nuo atrastų tiltų ar tunelių, bet būtų pasirinktas kažkoks kintamasis, kuris reikštų, kokio spindulio apskritimo ribose būtų ieškoma sankryžų. Naudojantis šį sprendimo būdą būtų nusibrėžiamas įsivaizduojamas apskritimas ir einama jo ribomis iki tol, kol yra sutinkama kažkokia kelio atkarpa einanti į apskritimo vidų. Tada algoritmas tikrina, kur eina ši kelio atkarpa iki tol, kol yra įsivaizduojamo apskritimo viduje. Kai kelio atkarpa susikerta su apskritimo ribomis, algoritmas toliau eina pagal apskritimo ribas, kol vėl yra sutinkama kita kelio atkarpa. Algoritmas pabaigia kelių tikrinimą, kai tokiu būdu grįžta prie pradinio taško, kuriame buvo pradėtos tikrinti kelio atkarpos. Jeigu patikrinti keliai apskritimo viduje nesusikirto nei vieno karto, tai reiškia, jog sankryžos tikrinamoje vietoje nėra. Bet, jeigu keliai susikerta nors vieną kartą tai reiškia, jog šiame tikriname apskritime yra mažiausiai viena sankryža.

Tačiau šio bakalauro darbo vykdymo metu buvo nuspręsta naudoti, kitą sprendimo būdą. Tai yra, panaikinti žemėlapiuose esantį nevientisumą. Kadangi, kai kurie keliai yra sudaryti iš vienos kelio atkarpos, o kiti beveik identiški keliai yra sudaryti iš keleto atkarpų, tai yra patogu tiesiog padalinti ilgąsias kelio atkarpas į smulkesnes kelio atkarpas tose vietose, kur sankryžoms priklausančios kelius jungiantys nuvažiavimai, kurie yra žymimi, kaip kelio jungtys, susikerta su kelio atkarpa, kuri šiame susikirtimo taške neprasideda ar nepasibaigia.

Pirmoji algoritmo prototipo iteracija veikia taip, jog algoritmas ieškodamas kelio segmentų,



9 pav. Vientisumo trūkumo pavyzdys. Skirtingomis spalvomis pavaizduotos skirtingos atkarpos.

4 algoritmas. Kelių segmentų padalinimas

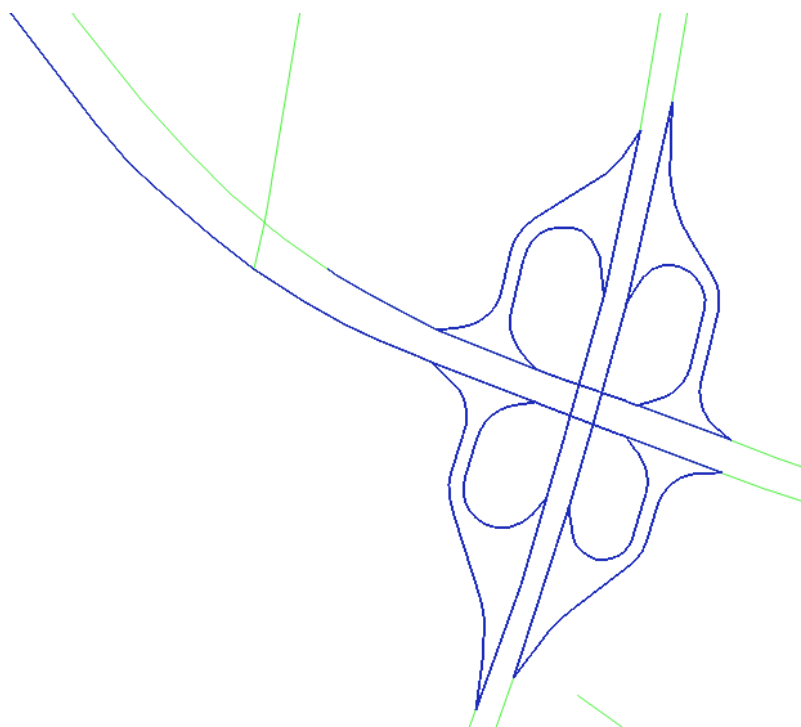
Ivestis: *roadSegments* - formos formato failas (Kelių žemėlapis), *linkPoints* - formos formato failas (Sankryžos jungčių susikirtimo su kitomis sankryžos atkarpomis taškai)

- 1: *roadSegments* faile yra surandamos tiltus ar tunelius žyminčios kelio atkarpos
 - 2: surandamos atkarpos *firstLine*, kurios kertasi su rastu tiltu ar tuneliu
 - 3: **if** tikrinamas *linkPoints* nėra nei pradžios nei pabaigos taškas *firstLine* atkarpoje **then**
 - 4: *firstLine* padalinama taške *linkPoints*
 - 5: **else**
 - 6: rasta atkarpa, kuri nėra *jungtis* laikoma esanti *secondLine*, kurios kertasi su *firstLine*
 - 7: **if** tikrinamas *linkPoints* nėra nei pradžios nei pabaigos taškas *secondLine* atkarpoje **then**
 - 8: *secondLine* padalinama taške *linkPoints*
 - 9: **end if**
 - 10: **end if**
-

kurie yra pažymėti esantys kelio jungtimis, tuo pačiu patikrina ir taškus, kuriuose šios jungties susikerta su kitomis kelio atkarpomis. Tikrinantis algoritmas sulygina besikertančios kelio atkarpos pradžios ir pabaigos taškus su kelio atkarpų susikirtimo tašku. Jeigu nei pradžios taškas nei pabaigos taškas nesutampa su jungties ir tikrinamos kelio atkarpos susikirtimo tašku, tai tikrinama atkarpa yra padalinama į dvi atkarpas.

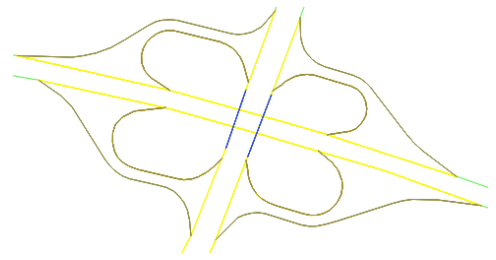
5.2.3. Likusių sankryžos segmentų radimas

Kai yra surastos pagrindinės sankryžos kelių atkarpos, kurios charakterizuoja sankryžas, belieka surasti visas likusias kelių atkarpas, kurios priklauso nagrinėjamai sankryžai. Šiuo atveju, dobilo tipo sankryžoje, suradus tiltus ar tunelius bei kelio nuvažiavimus, kurie yra žymimi, kaip kelio jungtimis, reikia surasti, kelio atkarpas, kurios sujungia kelio jungtis su sankryžos tiltais. Būtent šios kelių atkarpos ir yra pagrindinės sankryžų kelio atkarpos, nes būtent šiomis kelio



10 pav. Dobilo tipo sankryža surasta nekeičiant pradinių duomenų

atkarpomis sudarytų kelių ir yra pastatytos sankryžos. Tam, kad būtų surastos šios kelio atkarpos, yra patikrinamos visos kelių atkarpos esančios pradiniam formos formato kelių žemėlapyje ir jos yra sulyginamos su turimomis sankryžų tiltų bei tunelių atkarpomis. Sutampančios atkarpos yra patikrinamos su taškų, žyminčių kelio jungčių bei kitų sankryžos kelio atkarpų susikirtimus, koordinatėmis. Jeigu kelio atkarpa eina per šį tašką ir kerta arba liečiasi su sankryžos tiltu, kurio identifikacijos numeris yra tas pats, kaip ir jungties taško duomenyse aprašytame, kuriam tiltui ar tuneliui priklauso jungtis, tokiu atveju, kelio atkarpa jungianti jungties tašką ir tiltą ar tunelį yra įrašoma į naują failą, kuriame saugomos likusios kelių atkarpos, kurios nėra nei tiltai nei jungtys. Tačiau, dauguma dobilo tipo sankryžų turi po dvi kelio jungtis visomis keturiomis kryptimis ir tolimiausia kelio jungtis yra nutolusi nuo tilto ar tunelio per dvi kelio atkarpas. Būtent dėl šios priežasties jungties susikirtimo su kitomis sankryžos kelio jungtimis taškų duomenyse yra atributas nusakantis ar šis taškas priklauso tolimajai kelio jungčiai. Jeigu tikrinamas taškas yra pažymėtas, jog priklauso tolimesniajai jungčiai, tai į naują failą yra įrašomos dvi kelio atkarpos, kurios susikerta viena su kita ir sujungia sankryžos tiltą ar tunelį su nagrinėjamu jungties tašku.



11 pav. Šviesiai geltona pažymėtos likusios sankryžos dalys

5 algoritmas. Likusių sankryžos atkarpų radimas

Ivestis: *roadSegments* - formos formato failas (Kelių žemėlapis), *bridges* - formos formato failas (Tiltų ir viadukų žemėlapis iš pirmojo algoritmo), *linkPoints* - formos formato failas (Sankryžos jungčių susikirtimo su kitomis sankryžos atkarpomis taškai)

Išvestis: *newFile* - formos formato failas, kuriame žymimos likusios sankryžos kelio atkarpos

- 1: kiekvienam *bridges* patikrinami visi *linkPoints*
 - 2: kiekvienam *linkPoints* yra patikrinami visi *roadSegments*
 - 3: **if** *roadSegments* kertasi su *bridges* ir su *linkPoints* **IR** *roadSegments* yra pažymėtas, jog priklauso tikrinamam *bridges* **IR** tikrinamas *linkPoints* yra pažymėtas, jog nepriklauso antrajai jungčiai **then**
 - 4: *newFile* faile išsaugomas tikrinamas *roadSegments*
 - 5: **else if** tikrinamas *linkPoints* yra pažymėtas, jog priklauso antrajai jungčiai **then**
 - 6: tikrinamam *roadSegments* yra patikrinami visi *roadSegments*
 - 7: **if** abu tikrinami *roadSegments* kertasi **IR** antrasis *roadSegments* priklauso tikrinamam *bridges* **IR** antrasis *roadSegments* kertasi su *linkPoints* **then**
 - 8: abu tikrinami *roadSegments* yra įrašomi į *newFile*
 - 9: **end if**
 - 10: **end if**
-

6. Pasiektas rezultatas

Sukurtas algoritmas yra sudarytas iš kelių dalių, kurias atlikus atitinkama eilės tvarka QGIS sistemos aplinkoje, iš pasirinkto formos formatu aprašyto žemėlapio yra išrenkamos kelių sankryžos. Šios kelių sankryžos yra išrenkamos visos pagal dobilo tipo sankryžos charakteristikas, nes šio bakalauro darbo metu didžiausias dėmesys buvo kreipiamas į dobilo tipo sankryžas, kurios yra vienos populiariausių pasaulyje. Pilnai ir sėkmingai sukūrus algoritmą, kuris išrinktų dobilo tipo sankryžas, pagal jį būtų galima sukurti ir kitus algoritmus ar patobulinti esamą taip, kad būtų išrenkamos kitokio tipo sankryžos, nes jos nuo dobilo tipo sankryžos skiriasi tik charakteristikomis.

Šio bakalauro darbo metu buvo dirbama su Lietuvos kelių žemėlapiu parsisiųstu iš OSM projekto tinklalapio formos formatu [3]. Tai yra vienas populiariausių vektorinių žemėlapių formatų, kuriuose taip pat yra ir įrašą aprašantys atributai. Būtent šitus duomenis ir naudoja šio bakalauro darbo metu sukurtas algoritmas, kad atskirtų pirmajame algoritmo dalies skaičiavime visas kelių atkarpas, kurios nėra nutiestos ant žemės ir eina virš ar po kitomis kelio atkarpomis. Algoritmui atrinkus šias kelių atkarpas, jos yra patalpinamos į naują formos formato žemėlapi su atributais nusakančiais identifikacijos numerius tilto ar tunelio atkarpos bei kelio atkarpos su kuria yra rastas tilto ar tunelio susikirtimas.

Antroji algoritmo dalis yra skirta sankryžos nuvažiavimų radimui ir sankryžai priklausančių kelio atkarpų padalinimui į smulkesnes atkarpas. Šis atkarpų padalinimas į smulkesnes yra reikalingas tam, kad naudojami duomenys būtų vientisesni ir tokiu būdu sankryžos radimo algoritmas veiktų tiksliau bei rastų sankryžų atvaizdavimas būtų aiškesnis. Tačiau sankryžos kelio atkarpų padalinimas į smulkesnes neveikia ir dėl šios priežasties antroji algoritmo dalis tik suranda kelių atkarpas priklausančias sankryžai, kurios yra pažymėtos esančios jungtimis, bei taškus, kuriuose šios jungtys susijungia su kitomis sankryžai priklausančiomis kelio atkarpomis.

Trečioji ir paskutinioji sukurtą algoritmo dalis panaudoja išvestis iš praeitų dalių, kaip įvestį. Ši dalis iš naujo perrenka kelių atkarpas, ir suranda tas atkarpas, kurios priklauso kelių sankryžoms. Šios algoritmo dalies išvestis yra naujas formos formato failas, kuriame yra išsaugotos kelių sankryžas sudarančios atkarpos, kurių vienas iš atributų yra priklausomybė vienam iš tiltų ar tunelių nusakantis identifikacijos numeris. Turint šį atributą galima nusakyti, kurios kelio atkarpos yra tarpusavyje susijusios ir sudaro sankryžas.

7. Iškilusios problemos

Šiame skyriuje yra aprašomos bakalauro darbo vykdymo metu iškilusios problemos, jų priežastys, kokius sunkumus jos sukėlė, galimi bei pasirinkti sprendimo būdai. Iškilusios problemos, kurios nebuvo išspręstos plačiau yra aprašytos 8 skyriuje, o išsamus aprašymas apie visas problemas ir jų sprendimus yra aprašytas 5 skyriuje.

Tikriausiai didžiausia problema, su kuria buvo susidurta šio algoritmo kūrimo metu yra ta, jog naudojamuose kelių žemėlapiuose trūksta vientisumo. Tai yra, jog palyginus du beveik identiškus kelių fragmentus, juos sudarančių atkarpų kiekis gali skirtis. Šis vientisumo nebuvimas sukelia įvairių problemų, pavyzdžiui, neaiškų grafinį sankryžos atvaizdavimą. Į jį pažiūrėjus yra sunku pasakyti, kur prasideda sankryža, ir tai, jog yra sudėtingiau pritaikyti algoritmą sankryžų radimui, nes sankryžą sudarančių kelio atkarpų radimui jau negalima panaudoti atkarpų skaičiaus skaičiuojant nuo tiltų žyminčios atkarpos. Šią problemą buvo stengiamasi išspręsti ilgesnes atkarpas padalinant į kelias smulkesnes tam, kad būtų paprasčiau atlikti algoritmus. Tačiau ši problema iki galo nebuvo sėkmingai išspręsta.

Dar viena didelė problema, kurios nebuvo spėta laiku pilnai ištaisyti yra ta, jog sukurtas algoritmas į naujus sukurtus failus įrašo keletą identiškų įrašų. Tai yra, naujame faile, kuriame yra saugomos surastos kelių sankryžos yra įrašų, kurie yra su tomis pačiomis koordinatėmis, kuriose yra ir kiti įrašai su tokiais pačiais atributais. Tai yra dėl tos priežasties, jog dvi viena su kita nesikertančios atkarpos viena kitą gali pasiekti keliais skirtingais būdais, nes visi keliai yra sujungti kitomis kelio atkarpomis. Tai ypač pasireiškia, kai sankryžą sudaro daugiau nei du galimus būdus, kaip sujungti dvi kelio atkarpas. Būtent dėl šios priežasties tikrinant visus galimus kelius, šių galimybių staigiai daugėja. Šią problemą galima išspręsti keliais būdais. Pirmasis būtų sukurti atskirą algoritmą, kuris jau turint visus duomenis naujame faile išrinktų pasikartojančius, tačiau, tai nėra efektyvus variantas, nes atsiranda papildomo laiko sąnaudų bei kompiuteriui yra žymiai daugiau skaičiavimo, nes reikia duomenis įrašyti, o paskui juos ištrinti. Būtent dėl to geresnis sprendimo būdas būtų teisingai aprašyti patikrinimą ar duotas įrašas jau yra įrašytas. Šį sprendimo būdą ir

buvo bandyta įgyvendinti, tačiau, dėl neaiškos priežasties, nesėkmingai.

Tačiau, ne visos susidurtos problemos nebuvo išspręstos. Kai kurios problemos su kuriomis buvo susidurta, buvo išspręstos darbo vykdymo metu. Pirmiausia aukščiau aprašytų problemų sprendimui buvo užtrunkta labai daug laiko tam, kad būtų sugalvotas būdas, kaip galima išspręsti problemas tokias, kaip vientisumo nebuvimas naudojamuose kelių žemėlapiuose. Kadangi didžioji dalis dėmesio šiame bakalauro darbe buvo skirta turimų duomenų analizei bei galimų sprendimo būdų ieškojimui, čia daugiausia laiko buvo praleista ir sutikta daugiausia problemų. Hipotetinių problemų, kurios vienaip ar kitaip trukdytų produktyviai pasiekti darbo tikslą. Pavyzdžiui, pagal vieną iš pirmųjų sugalvotų sprendimo būdų sankryžų reikėtų ieškoti kelių tikrinant įsivaizduojamų apskritimų ribose, tačiau, pasirinkti atitinkamą apskritimo vietą bei jo spindulį tam, kad sėkmingai būtų surastos sankryžos, yra problematiška. Nes neteisingai pasirinkus apskritimo duomenis sankryžos gali būti nerastos arba surastos nepilnos. Tam, kad būtų išvengta susidūrimo su šia problema, buvo praleista nemažai laiko, jog būtų surastas būdas pagal ką nustatyti tikrinimui reikalingo apskiravimo duomenys, tačiau, galiausiai buvo priimtas sprendimas, jog sankryžų reikia ieškoti kitoku būdu. 5 skyriuje yra aprašyta visi apmąstymai, kokios problemos gali iškilti bei kokių priemonių buvo imtasi, kad išvengti ar ištaisyti jas.

8. Tobulinimo galimybės

Šiame skyriuje yra aprašoma, kokiomis funkcijomis galima patobulinti šio bakalauro darbo metu kurtą sankryžų radimo ir klasifikavimo algoritmą bei kokios funkcijos nebuvo pilnai įgyvendintos dėl laiko trūkumo ar kitokių iškilusių problemų.

Tikriausiai svarbiausias funkcionalumas, kaip galima patobulinti sukurtą algoritmą, yra sėkmingai įgyvendinti kelių atkarpų padalinimą, kuris reikalingas tam, jog būtų tiksliau surandamos sankryžos bei būtų aiškesnis jų atvaizdavimas. Šis funkcionalumas yra reikalingas tam, kad būtų tiksliau surandamos kelių atkarpos priklausančios sankryžoms. Šis funkcionalumas algoritmo kūrimui skirtas laiko pabaigoje, deja, nėra veikiantis, nežinia dėl kokios priežasties. Tačiau, būdas, kaip tai yra bandoma padaryti, kuris turėtų atnešti sėkmingus rezultatus yra toks, jog kelio jungčių ir kitų sankryžos atkarpų susikirtimo vietas žymintys taškai yra patikrinami su pradiniais kelio atkarpų duomenimis ir yra patikrinama ar kelio atkarpos pradžios ir pabaigos taškai yra toje pačioje koordinatėje, kaip ir tikrinamas jungties susikirtimo taškas. Jeigu tikrinamos atkarpos nei vienas iš pradžios ar pabaigos taškų nėra tame pačiame taške, kaip ir jungties susikirtimo taškas, yra panaudojama QGIS funkcija, kuri padalina kelius į smulkesnius, perkirsdama nurodytoje vietoje, tačiau čia tikriausiai ir yra problema, dėl kurios ir neveikia kelio atkarpų padalinimas.

Dar vienas svarbus funkcionalumas, kuriuo būtų galima patobulinti šio bakalauro metu kurtą algoritmą, yra tai, jog iš visų atrinktų sankryžų reiktų atrinkti dobilo tipo sankryžas. Tai būtų galima padaryti, kai sankryžos neturi pasikartojančių įrašų. Jeigu nėra pasikartojančių įrašų, dobilo tipo sankryžas galim išskirti patikrinant, iš kokių kelio atkarpų yra sudaryta sankryža, ir kaip jos yra išsidėsčiusios. Pilnos dobilo tipo sankryžos turi du tiltus bei aštuonias kelio jungtis. Keturios kelio jungtys yra nutolusios per vieną kelio atkarpą nuo tiltų, o kitos keturios jungtys yra nutolusios per dvi kelio atkarpas nuo tiltų.

Sėkmingai patobulinus algoritmą aukščiau aprašytais būdais, vienintelis likęs dalykas, kuriuo galima patobulinti šį algoritmą, tai yra jo pritaikymas kito tipo sankryžoms. Tai reiškia, jog kiekvieną norimą surasti sankryžos tipą reikia išnagrinėti, kokios kelio atkarpos, ir kaip jos yra išsidėsčiusios. Išnagrinėjus norimą surasti sankryžos tipą, telieka aprašyti sąlygos algoritmus, kurie išrenka atitinkamas sankryžas. Įgyvendinus šias pagrindines funkcijas, kuriomis galima patobulinti šiuo metu turimą algoritmą, būtų sėkmingai pabaigtas sankryžų klasifikavimo algoritmas.

Išvados ir rekomendacijos

Baigamojo bakalauro metu tam, kad būtų sukurta duomenų sistema, kurią naudojant galima būtų pakeisti sankryžas į saugesnes išleidžiant mažesnes sumas pinigų, buvo stengiamasi sukurti algoritmą, kuris iš kelių žemėlapių surastų kelių sankryžas ir jas suklasifikuotų. Tam, kad būtų pasiektas šis tikslas, buvo naudojami geografinių duomenų analizavimo bei apdorojimo įrankiai su kurių pagalba ir buvo atliekamos užduotys, kurias reikalinga atlikti norint pasiekti prieš tai aprašytą tikslą. Pirmiausia, surasti tiltus ar tunelius, kurie eina per, virš ar po kitais kelio segmentais. Paskui, patikrinant pasirinktu sprendimo būdu, aplinkines kelio atkarpas, surandamos atkarpos priklausančios nagrinėjamai sankryžai. Tokiu būdu išrinkus visas sankryžas galima jas suklasifikuoti patikrinus, kaip atitinkamos sankryžos kelių atkarpos yra išsidėsčiusios ir kokio tipo šios atkarpos yra. Būtent tai ir yra dalykai nusakantys sankryžų charakteristikas pagal kurias skirtingi sankryžų tipai išsiskiria nuo kitų sankryžų tipų.

Kuriant algoritmą šio baigamojo bakalauro metu buvo susidurta su nemažai problemų, tarp kurių buvo tokių, kurias pavyko ganėtinai paprastai išspręsti, pavyzdžiui, perrenkant duomenis. Naudojami kelių žemėlapiai yra labai detalūs ir dėlto juose yra labai didelė dalis duomenų, kurie yra visiškai nereikalingi sankryžų paieškai. Tokios kelių atkarpos, kaip žvyrkeliai ar gyvenamųjų vietovių keliai, negali sudaryti sankryžų, kurias ieško šis algoritmas. Tačiau buvo susidurta ir su tokiomis problemomis, kurių sprendimo būdas buvo sugalvotas, bet jo nepavyko sėkmingai įgyvendinti dėl neaiškių priežasčių. Šių problemų sėkmingam įgyvendinimui neužteko laiko, tačiau šios neišspręstos problemos nėra tokios, dėl kurių visiškai neveikia kurtas algoritmas. Dauguma problemų su kuriomis buvo susidurta buvo sėkmingai ištaisytos.

Šio bakalauro metu pabaigoje užsibrėžtų tikslų pagrindinė dalis buvo pasiekta. Tam, kad būtų patobulintas sukurtas algoritmas reikėtų sukurti sėkmingą kelio atkarpų padalinimą į mažesnes atkarpas, ties sankryžoms priklausančių kelio nuvažiavimų susijungimais su kelio atkarpomis, kurias stengiamasi padalinti. Tokiu būdu rastos sankryžos būtų tikslesnės ir aiškesnės. Dar vienas būdas, kuriuo būtų galima patobulinti sukurtą algoritmą, yra surastų sankryžų charakteristikų patikrinimas ir, pagal išsiaiškintus duomenis, sankryžų suklasifikavimas. Šiomis funkcijomis papildžius sankryžų radimo ir klasifikavimo algoritmą, darbo tikslas būtų pilnai pasiektas.

Literatūros šaltiniai

- [1] Documentation for qgis. <https://docs.qgis.org/2.18/en/docs/>. Žiūrėta: 2018-01-05.
- [2] Esri shapefile technical description. <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. Žiūrėta: 2018-01-06.
- [3] Official wiki of openstreetmap. https://wiki.openstreetmap.org/wiki/Main_Page. Žiūrėta: 2018-01-06.
- [4] Qgis api documentation. <http://qgis.org/api/>. Žiūrėta: 2018-01-05.
- [5] R. Krasauskas A. Juozapavičius, S. Buteliauskas. Improvement possibilities of city transportation system by using pinavia interchange. *Environmental Research, Engineering and Management*, 3(53):36–42, 2010.
- [6] Hantao Zhong. Free-flow parclo interchange vs. the cloverleaf interchange with c-d roads and the all-directional four-level interchange: A comparison of geometrics, construction cost, and right of way requirements. ScM Thesis. North Carolina State University, 2012. <https://repository.lib.ncsu.edu/bitstream/handle/1840.16/8006/etd.pdf?sequence=2>.