

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Simonas Razminas

**Realaus laiko transporto maršrutų optimizavimo  
algoritmų tyrimas**

Magistro darbas

Darbo vadovas  
doc. dr. L. Nemuraitė

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Simonas Razminas

**Realaus laiko transporto maršrutų optimizavimo  
algoritmų tyrimas**

Magistro darbas

Kalbos konsultantė

Lietuvių kalbos katedros lektorė  
dr. Jurgita Mikelionienė  
2006-05

Darbo vadovas

doc. dr. L. Nemuraitė  
2006-05

Recenzentas

doc. V. Kiauleikis  
2006-05

Atliko

IFM-0/2 gr. stud.  
Simonas Razminas  
2006-05

Kaunas, 2006

# **Research of real time vehicle routing optimization algorithms**

## **Summary**

Nowadays traffic congestion is a big problem all over the world. To solve this problem governments build broader roads, establish more reasonable traffic rules. Dynamic routing is a good and efficient way to reduce traffic congestion. That is why research of real time vehicle routing optimization algorithms was made. Experiment showed that best performance of shortest path algorithm was Dijkstra algorithm.

Based on that, a software prototype was developed – optimized route search system. Driver can select shortest or fastest route to his destination. There was used roads length to evaluate shortest path and special coefficient to evaluate fastest route. This coefficient is calculated respectively to road load, length, speed limit, capacity. Performance of developed system is good so I conclude that the system is capable of routing vehicles in real time in complex traffic network.

# TURINYS

<b>SUMMARY .....</b>	<b>3</b>
<b>1. ĮVADAS .....</b>	<b>8</b>
<b>2. PROBLEMINĖS SRITIES TYRIMAS .....</b>	<b>11</b>
2.1. INTELEKTUALIŲJŲ TRANSPORTO SISTEMŲ (ITS) TYRIMAS .....	11
2.1.1. <i>Projektiniai sprendimai</i> .....	14
2.1.1.1. TelemArk .....	14
2.1.1.2. Actif .....	14
2.1.1.3. ARTIST .....	14
2.1.2. <i>Techniniai sprendimai</i> .....	15
2.1.2.1. Siemens sprendimas .....	15
2.1.2.2. Telvent sprendimas .....	15
2.2. MARŠRUTŲ OPTIMIZAVIMO SISTEMŲ TYRIMAS .....	16
2.2.1. <i>Statinės maršrutų sistemos</i> .....	16
2.2.2. <i>Dinaminės maršrutų sistemos</i> .....	16
2.3. TRUMPiausio KELIO RADIMO ALGORITMŲ TYRIMAS .....	17
2.3.1. <i>Trumpiausio kelio radimas svoriniame grafe</i> .....	18
2.3.2. <i>Grafo svorio tyrimas</i> .....	20
2.3.3. <i>Dijkstra algoritmo analizė</i> .....	20
2.3.4. <i>Bellman-Ford algoritmo analizė</i> .....	21
2.3.5. <i>Floyd-Warshall algoritmo analizė</i> .....	22
2.3.6. <i>Johnson algoritmo analizė</i> .....	22
2.3.7. <i>„Paieška į gylį“ algoritmo analizė</i> .....	23
2.3.8. <i>„Paieška į plotį“ algoritmo analizė</i> .....	23
<b>3. TRUMPiausio KELIO RADIMO ALGORITMŲ ĮVERTINIMAS.....</b>	<b>24</b>
3.1. TRUMPiausio KELIO RADIMO ALGORITMŲ ATRANKA .....	24
3.2. DIJKSTRA IR BELLMAN-FORD ALGORITMŲ Palyginimas .....	24
3.2.1. <i>Algoritmų palyginimas mažo dydžio grafuose</i> .....	25
3.2.2. <i>Algoritmų palyginimas didelio dydžio grafuose</i> .....	27
3.2.3. <i>Algoritmų palyginimo rezultatai</i> .....	28
<b>4. GREIČIAUSIO KELIO RADIMO SISTEMOS PROTOTIPAS.....</b>	<b>30</b>
4.1. PANAUDOJIMO ATVEJŲ VAIZDAS .....	31
4.1.1.1. Panaudojimo atvejų sąrašas .....	32
4.2. SISTEMOS STATINIS VAIZDAS .....	40
4.2.1. <i>Apžvalga</i> .....	40
4.2.2. <i>Paketų detalizavimas</i> .....	40
4.2.2.1. Paketas “Vairuotojo sąsaja” .....	40
4.2.2.2. Paketas “Administratoriaus sąsaja” .....	41

4.2.2.3.	Paketas “Kelionių planavimo paslaugos” .....	42
4.2.2.4.	Paketas “Duomenų paslaugos” .....	43
4.2.2.5.	Paketas “Transporto stebėjimo sistemos sąsaja” .....	44
4.3.	SISTEMOS DINAMINIS VAIZDAS .....	45
4.3.1.	<i>Būsenų diagramos</i> .....	45
4.3.1.1.	Prisijungimas .....	45
4.3.1.2.	Parametrų nustatymas bei kitų duomenų tvarkymas (taip pat ir duomenų atnaujinimas) .....	46
4.3.1.3.	Ataskaitų formavimas .....	47
4.3.1.4.	Maršruto radimas .....	48
4.3.2.	<i>Veiklos diagramos</i> .....	49
4.3.2.1.	Avarių peržiūra .....	49
4.3.3.	<i>Sekų ir bendradarbiavimo diagramos</i> .....	50
4.3.3.1.	Panaudojimo atvejis „Administratoriaus prisijungimas“ .....	50
4.3.3.2.	Panaudojimo atvejis „Parametrų nustatymas“ .....	51
4.3.3.3.	Panaudojimo atvejis „Stebėjimas ir koregavimas“ .....	52
4.3.3.4.	Panaudojimo atvejis „Ataskaitų formavimas“ .....	53
4.3.3.5.	Panaudojimo atvejis „Vairuotojo prisijungimas“ .....	53
4.3.3.6.	Panaudojimo atvejis „Maršruto radimas“ .....	54
4.3.3.7.	Panaudojimo atvejis „Avarių peržiūra“ .....	54
4.3.3.8.	Panaudojimo atvejis „Transporto srautų duomenų atnaujinimas“ .....	55
4.4.	IŠDĖSTYMO VAIZDAS.....	56
4.5.	DUOMENŲ VAIZDAS .....	57
4.6.	SUKURTOS SISTEMOS PROTOTIPO APRAŠYMAS .....	58
4.6.1.	<i>Prisijungimas prie sistemos</i> .....	58
4.6.2.	<i>Žemėlapių užkrovimas, išsaugojimas</i> .....	59
4.6.3.	<i>Pagrindinis langas</i> .....	60
4.6.4.	<i>Žemėlapių redagavimas</i> .....	61
4.6.5.	<i>Peržiūros režimai</i> .....	63
4.6.6.	<i>Greičiausio kelio paieška</i> .....	64
4.6.7.	<i>Ataskaitos</i> .....	66
4.6.7.1.	Sistemos sandara:.....	68
4.6.7.2.	Sistemos pranešimai .....	68
<b>5.</b>	<b>IŠVADOS</b> .....	<b>69</b>
	<b>LITERATŪRA</b> .....	<b>70</b>
	<b>TERMINŲ IR SANTRUMPŲ ŽODYNAS</b> .....	<b>73</b>

## Lentelių sąrašas

LENTELĖ 1.	TRUMPIAUSIO KELIO RADIMO ALGORITMAI .....	18
LENTELĖ 2.	ALGORITMŲ SAVYBIŲ PALYGINIMŲ LENTELĖ .....	24
LENTELĖ 3.	ALGORITMŲ PALYGINIMAS IEŠKANT GRAFUOSE SU DIDELIU VIRŠŪNIŲ KIEKIU (PIRMASIS EKSPERIMENTAS).....	27
LENTELĖ 4.	ALGORITMŲ PALYGINIMAS IEŠKANT GRAFUOSE SU DIDELIU VIRŠŪNIŲ KIEKIU (ANTRASIS EKSPERIMENTAS).....	28
LENTELĖ 5.	ARCHITEKTŪROS MODELIAVIMO ELEMENTAI .....	30
LENTELĖ 6.	ADMINISTRATORIAUS PRISIJUNGIMAS.....	32
LENTELĖ 7.	PARAMETRŲ NUSTATYMAS .....	33
LENTELĖ 8.	STEBĖJIMAS KOREGAVIMAS .....	34
LENTELĖ 9.	ATASKAITŲ FORMAVIMAS.....	34
LENTELĖ 10.	VAIRUOTOJO PRISIJUNGIMAS .....	35
LENTELĖ 11.	MARŠRUTO RADIMAS .....	36
LENTELĖ 12.	MARŠRUTO SURADIMAS ĮVERTINANT TRUKMĘ .....	37
LENTELĖ 13.	MARŠRUTO SURADIMAS ĮVERTINANT ATSTUMĄ.....	38
LENTELĖ 14.	AVARIJŲ PERŽIŪRA .....	39
LENTELĖ 15.	TRANSPORTO SRAUTŲ DUOMENŲ ATNAUJINIMAS .....	39
LENTELĖ 16.	GRAFINĖS APLINKOS PRANEŠIMAI .....	68

## Paveikslėlių sąrašas

PAVEIKSLAS 1.	KELIONIŲ PLANAVIMO SISTEMA ITS KONTEKSTE .....	13
PAVEIKSLAS 2.	DINAMINIS MARŠRUTŲ RADIMAS .....	17
PAVEIKSLAS 3.	SVORINIS KRYPTINIS GRAFAS SU NEIGIAMAIS SVORIAIS.....	19
PAVEIKSLAS 4.	PIRMASIS EKSPERIMENTO BANDYMAS (MAŽI GRAFAI) .....	25
PAVEIKSLAS 5.	ANTRASIS EKSPERIMENTO BANDYMAS (MAŽI GRAFAI) .....	26
PAVEIKSLAS 6.	PAIEŠKA DIDELIUOSE GRAFUOSE (PIRMASIS EKSPERIMENTAS) .....	27
PAVEIKSLAS 7.	PAIEŠKA DIDELIUOSE GRAFUOSE (ANTRASIS EKSPERIMENTAS).....	28
PAVEIKSLAS 8.	PANAUDOJIMO ATVEJŲ DIAGRAMA .....	31
PAVEIKSLAS 9.	PAKETŲ DIAGRAMA.....	40
PAVEIKSLAS 10.	VAIRUOTOJO PAKETO DETALIZACIJA.....	40
PAVEIKSLAS 11.	„VAIRUOTOJO SAŠAJOS“ PAKETO DETALIZACIJA .....	41
PAVEIKSLAS 12.	„KELIONIŲ PLANAVIMO PASLAUGŲ“ PAKETO DETALIZACIJA .....	42
PAVEIKSLAS 13.	„DUOMENŲ PASLAUGŲ“ PAKETO DETALIZACIJA .....	43
PAVEIKSLAS 14.	„TRANSPORTO STEBĖJIMO SISTEMOS SAŠAJOS“ PAKETO DETALIZACIJA .....	44
PAVEIKSLAS 15.	PRISIJUNGIMO BŪSENŲ DIAGRAMA.....	45
PAVEIKSLAS 16.	PARAMETRŲ NUSTATYMO BEI KITŲ DUOMENŲ TVARKYMO BŪSENŲ DIAGRAMA .....	46

PAVEIKSLAS 17.	ATASKAITOS FORMAVIMO BŪSENŲ DIAGRAMA .....	47
PAVEIKSLAS 18.	MARŠRUTO RADIMO BŪSENŲ DIAGRAMA .....	48
PAVEIKSLAS 19.	AVARIJŲ PERŽIŪROS VEIKLOS DIAGRAMA .....	49
PAVEIKSLAS 20.	PA „ADMINISTRATORIAUS PRISIJUNGIMAS“ SEKŲ DIAGRAMA .....	50
PAVEIKSLAS 21.	PA „ADMINISTRATORIAUS PRISIJUNGIMAS“ BENDRADARBIAVIMO DIAGRAMA .....	50
PAVEIKSLAS 22.	PA „PARAMETRŲ NUSTATYMO“ SEKŲ DIAGRAMA .....	51
PAVEIKSLAS 23.	PA „STEBĖJIMO IR KOREGAVIMO“ SEKŲ DIAGRAMA .....	52
PAVEIKSLAS 24.	PA „ATASKAITŲ FORMAVIMAS“ SEKŲ DIAGRAMA .....	53
PAVEIKSLAS 25.	PA „VAIRUOTOJO PRISIJUNGIMAS“ SEKŲ DIAGRAMA.....	54
PAVEIKSLAS 26.	PA „MARŠRUTO RADIMO“ SEKŲ DIAGRAMA .....	54
PAVEIKSLAS 27.	PA „AVARIJŲ PERŽIŪROS“ SEKŲ DIAGRAMA.....	54
PAVEIKSLAS 28.	PA „TRANSPORTO SRAUTŲ DUOMENŲ ATNAUJINIMAS“ SEKŲ DIAGRAMA .....	55
PAVEIKSLAS 29.	SISTEMOS PROTOTIPO IŠDĖSTYMO VAIZDAS .....	56
PAVEIKSLAS 30.	SISTEMOS PROTOTIPO DUOMENŲ VAIZDAS .....	57
PAVEIKSLAS 31.	KAIP PASIEKTI AUTENTIFIKACIJOS LANGĄ.....	58
PAVEIKSLAS 32.	AUTENTIFIKACIJOS LANGAS .....	58
PAVEIKSLAS 33.	ŽEMĖLAPIO PASIRINKIMAS .....	59
PAVEIKSLAS 34.	ŽEMĖLAPIŲ UŽKROVIMAS BEI SAUGOJIMAS Į FAILĄ .....	59
PAVEIKSLAS 35.	PAGRINDINIS LANGAS.....	60
PAVEIKSLAS 36.	PASIRINKTA SANKRYŽA.....	61
PAVEIKSLAS 37.	ŽEMĖLAPIO OBJEKTŲ REDAGAVIMAS .....	62
PAVEIKSLAS 38.	PAŽYMĖTAS KELIAS. ....	62
PAVEIKSLAS 39.	PERŽIŪROS REŽIMAI .....	63
PAVEIKSLAS 40.	PERŽIŪROS REŽIMAS „ADAPT STREETS“.....	64
PAVEIKSLAS 41.	GREIČIAUSIO KELIO MENIU PASIRINKIMAS .....	64
PAVEIKSLAS 42.	TRUMPIAUSIO KELIO ATVAIZDAVIMAS .....	65
PAVEIKSLAS 43.	KAMŠTIS.....	66
PAVEIKSLAS 44.	ATASKAITOS.....	67
PAVEIKSLAS 45.	GATVIŲ PAVADINIMŲ ATASKAITA .....	67

## 1. Įvadas

Augant transporto priemonių skaičiui, transporto kamščiai tapo kasdieniniu reiškiniu daugelyje pasaulio miestų. Kamščius gali sukelti autoavarijos, prasti orai, didelės šventės, gaisrai ir kitokie laikini transporto sistemos trukdžiai. Žmonėms kamščiai reiškia laiko praradimą, galimybių praradimą, asmeninių resursų švaistymą. Visuomenei transporto kamščiai reiškia mažesnę darbo produktyvumą, pavėluotus pristatymus ir padidėjusias investicijas.

Lietuvoje ši situacija juntama ne tik Vilniuje, bet ir Kaune bei Klaipėdoje. Laikui bėgant šis sąrašas tik plėsis. Valdžia bando išspręsti kamščių problemas įvairiais būdais: tiesiant naujus kelius, pertvarkant keliuose važiavimo apribojimus, tiesiant naujas viešojo transporto linijas. Vilniuje buvo mintis įvesti naują transporto rūšį – tramvajus, tačiau projektas nebuvo patvirtintas kaip efektyvus sprendimas. Žmonės bijo naujų kamščių tiesiant bėgius. Viešojo transporto vairuotojai bijo prarasti darbą. Šiuo metu kontruojamos žaliosios bangos, bei daromi kiti sprendimai. Kaip bebūtų, dėl įvairiausių politinių, finansinių ar aplinkos priežasčių, kai kurie iš šių sprendimų nėra efektyvūs arba neveikia visai. Vienintelis naujų kelių tiesimas yra visada veikianti priemonė šiai problemai gelbėti. Vis dėlto faktai yra tokie, kad transporto priemonių skaičius auga daug sparčiau nei naujų kelių.

Vienas variantas, kaip sumažinti kamščius neišleidžiant daugybės pinigų ir fizinės jėgos yra optimizuoti kelių apkrovimą. Optimalūs maršrutai gali būti atviras vyriausybės kelias į teisingą problemos sprendimo kelią. Renkant duomenis apie visus maršrutus, žmonės gali optimaliai išnaudoti eismo tinklą. Tada transporto priemonės vengtų kelių su kamščiais ir rinktųsi alternatyvius maršrutus. Jei transporto vairuotojai turės galimybę pasirinkti alternatyvų kelią naudojant tokio tipo maršrutų optimizavimo sistemą, eismo kamščiai bus sumažinti iki tam tikro laipsnio.

Daugybė įmonių jau siūlo navigacines paslaugas. Pagal duomenis, kuriais remiantis veikia sistemos, jas galima skirstyti į dvi kategorijas: statinės ir dinaminės. Statinėse sistemose kelionės laikas visame eismo tinkle tarp bet kokių taškų yra fiksuotas ir nesikeis. Trumpiausi keliai tarp bet kokių dviejų taškų `nekis ir todėl juos reikia paskaičiuoti tik vieną kartą. Tuo tarpu dinaminėms trumpiausias kelias gali pasikeisti bet kurią sekundę, todėl jį reikia pastoviai perskaičiuoti.

Dinaminės sistemos yra teisingas problemos sprendimo variantas, todėl šio **darbo tikslas** yra sukurti geriausio maršruto paieškos sistemos prototipą. Darbo tikslas yra prototipo kūrimas todėl, kad realiai jo ištestuoti kol kas yra neįmanoma. Duomenų bazės struktūra negali būti tiksli, dėl neegzistuojančios sistemos, todėl prototipas kuriamas taip, kad būtų



galima lengvai, greitai ir nesunkiai priderinti prie naujos struktūros. Lygiai tokia pati situacija su žemėlapių, bei maršrutų atvaizdavimu. Prototipe yra pateikiamas tik brėžinys-grafas, kadangi derinant prie kokios nors egzistuojančios žemėlapių sistemos, grafinis variklis jau turės egzistuoti. Taigi šio darbo **tyrimo sritis** apsiriboja transporto maršrutų valdymo sprendimais bei optimizavimo algoritmais, o **tyrimo objektas** tampa atskiro vairuotojo maršruto optimizavimo posistemis.

Kuo daugiau vairuotojų rinksis optimizuotus maršrutus, tuo labiau pasisikirstys transporto srantai. Kuo labiau pasiskirstys transporto srantai, tuo labiau sumažės kamščiai. Ypač tai turėtų pasijausti tvarkant gatves, įvykus avarijoms, kadangi ši sistema papildomai tarnaus kaip informacinė priemonė apie pravažiuojamus, naujus, bei mažiau žinomus mažus keliukus (dažniausiai apie remonto darbus suzinoma papuolus į kamštį). Turistai arba tiesiog miesto svečiai turės galimybę pasinaudoti ne tik pagrindinėmis gatvėmis bet ir mažesniais keliais.

Tikslui įgyvendinti reikia apžvelgti sukurtas ITS, surasti sprendimui reikalingus algoritmus, juos išanalizuoti, pritaikyti ir panaudoti. Šiam uždaviniui spręsti galima taikyti grafams pritaikytus algoritmus, kadangi kelių tinklą galima prilyginti grafui (kelio atkarpos – tai grafo kraštinės, sankryžos – grafo viršūnės). O kas yra „geriausias maršrutas“ vairuotojui? Atsakymas paprastas: geriausias maršrutas – tai toks maršrutas, kuris labiausiai patenkintų vairuotoją. Prototipas turi pasiūlyti vairuotojui tiek trumpiausią, tiek greičiausią maršrutą iš taško A į tašką B. Galutinis varototjas gali nuspręsti, kuriuo maršrutu jis sutaupys daugiausiai pinigų bei laiko. Taigi, vartotojas, pasileidęs sistemos prototipo programą, pasirenka kokio tipo maršruto jis norės ieškoti (trumpiausio arba greičiausio). Tada žemėlapyje (jį taip pat galima pasirinkti), nurodomi pradinis ir galutinis maršruto taškai, bei pasirenkama paieškos komandos meniu.

Probleminės srities tyrime apžvelgti ITS projektiniai bei techniniai sprendimai (Siemens, Telvent), palygintos dinaminės bei statinės sistemos. Nagrinėta Italijos, Prancūzijos, Suomijos nacionalinės Intelektualiosios Transporto Sistemos, bei daugumos tų projektų pagrindą sudarantis FRAME/KAREN projektas. Šiai daliai ištirti literatūros sąrašas yra tikrai didelis bei pati literatūra yra didelės apimties. KAREN projekto struktūroje galima rasti vidinę televiziją, tikralaikio keleivių informavimo sistemą, kintamų ženklų pranešimus, viešojo transporto valdymą ir dar galybę kitų sistemų. Ištirti trumpiausio kelio algoritmai, tokie kaip Dijkstra, Bellman-Ford, Floyd-Warshall ir kt. Iš šių algoritmų atmetimo būdu atsirinkti Dijkstra ir Bellman-Ford.

Atsirinktų algoritmų įvertinimui buvo aliktas eksperimentas (sukurta atskira programa). Iš pradžių buvo generuojami nedideli grafai (4-20 viršūnių), po to dideli (iki 10000 viršūnių).

Rezultatas toks pat – Dijkstra algoritmas vienareikšmiškai spartesnis, todėl būtent jis buvo pasirinktas kuriant geriausio maršruto paieškos sistemos prototipą. Nubrėžtos panaudojimo atvejų, paketų, būsenų, veiklos, sekos bei būsenų diagramos norint aiškiai apibrėžti kuriamos sistemos prototipą. Darbe pateikiami varotojo langų pavyzdžiai, bei gaunamos išvados.

## 2. Probleminės srities tyrimas

Maršrutų optimizavimo sistema yra tik viena iš Intelektualiųjų Transporto Sistemų sudedamųjų dalių, todėl dažniausiai yra siūloma kompleksas paslaugų.

### 2.1. Intelektualiųjų Transporto Sistemų (ITS) tyrimas

Intelektualiąsias transporto sistemas sudaro [1]:

- automatizuota eismo valdymo sistema
- pagalba specializuotų tarnybų operacijoms
- keliautojų informacinės paslaugos
- krovinių gabenimo ir skrydžių valdymo sistemos
- elektroninės apmokėjimo paslaugos
- pažengusios sistemos transporte
- ir kt.

Intelektualiosios transporto sistemos yra architektūrų, sistemų infrastruktūra, kuri siekia užtikrinti transporto funkcionalumą, saugumą, susisiekimo kokybę ir šios infrastruktūros vartotojų poreikius.

Daugybė įmonių ir institucijų bus(arba yra) suinteresuotos naudotis šia infrastruktūra. Visų pirma tai būtų miestų savivaldybės (moderniosios transporto sistemos). Iš esmės jau yra padaryta išvada, jog tiesiog tiesiant vis daugiau kelių yra retai sprendimas išskylančiai universaliausiai problemai – transporto kamščiams. Eismas turi būti valdomas efektyviau jau esančiuose keliuose. Eismo valdymas, turima galvoje ne tik kelių, bet ir traukinių, jūrų, upių, oro. Incidentų atpažinimo sistemos, kurios automatiškai praneša eismo valdymo centrams ir tuo pat metu nedelsiant praneša vairuotojams. Didelis žingsnis būtų padarytas keliautojų informacijų sistemose. ITS leistų informaciją gauti įvairiausiai būdais tiek prieš kelionę tiek per: kioskuose, per Internetą, WAP telefonais ir t.t. Keliautojai daug lengviau galės nuspręsti kaip jiems keliauti, koki keliavimo būdą pasirinkti, kokį maršrutą susidaryti. Pavyzdžiui vairuotojai būtų išpėti apie priekyje esantį transporto kamštį ir pasiūlyti alternatyvūs maršrutai link tikslo. Viešojo transporto naudotojai būtų informuoti apie atvykimo laiką. Konkretūs šiandienos egzistuojantys pavyzdžiai:

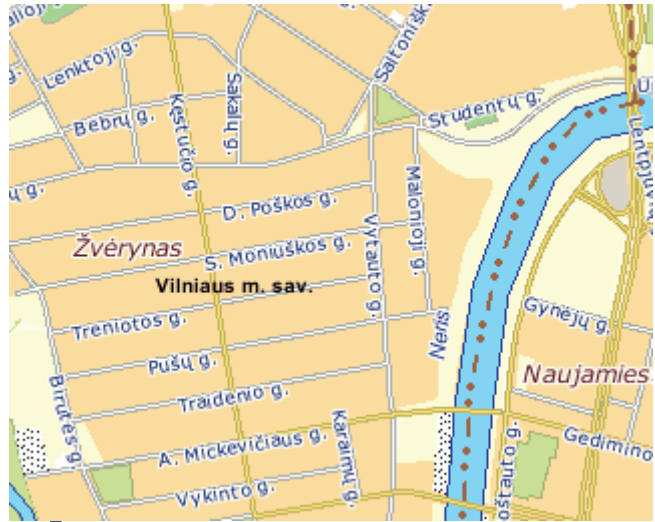
- ITS sistemos siūlo pilną kelionių paslaugų kompleksą: nuo kelionių planavimo ir maršruto konsultacijų iki bilietų užsakymo. Nuorodos su turistų adresais siūlo papildomas paslaugas, tokias kaip viešbučių užsakymas (ir visa tai viename puslapyje).

- Eismo, oro ir kelių sąlygų stebėjimas kelionės metu ir duodami patarimai keisti maršrutus ar keliavimo būdą priklausomai nuo oro sąlygų.
- Stebėjimas keliautojo lokacijos, lyginimas su planuotu maršrutu ir reikalui esant krypčių pateikimas.
- Viešojo transporto atvykimo laiko informacijos pateikimas realiu laiku sustojimuose, per mobilų telefoną ar Internetu.

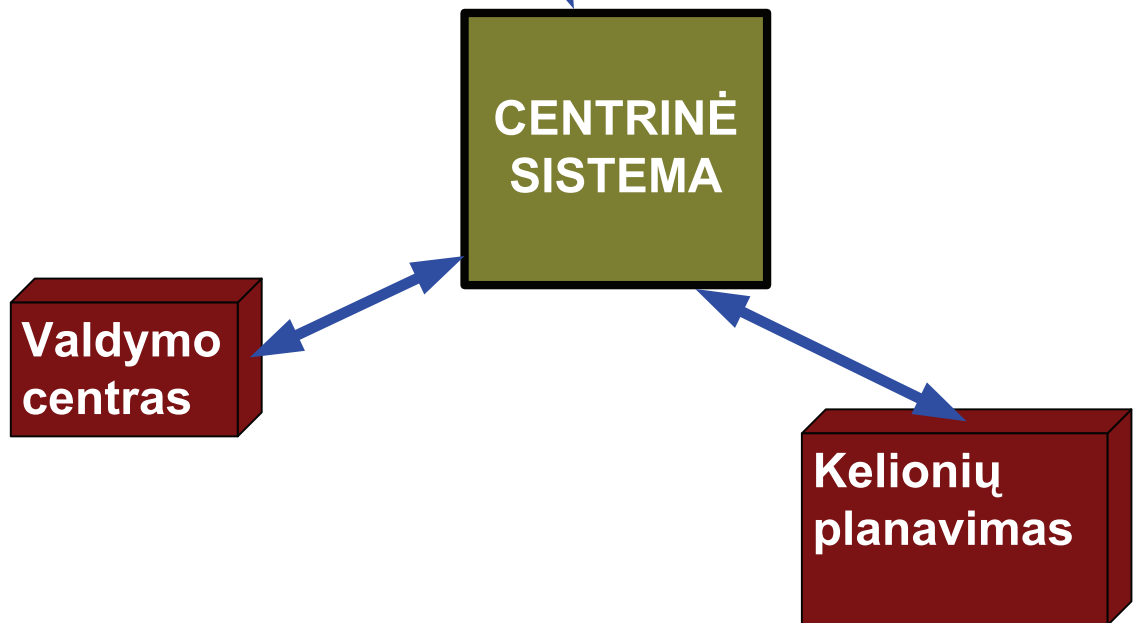
Galybė didelių įmonių turi logistikos skyrius, taip pat atskiros įmonės užsiimančios tik logistikos paslaugomis bus suinteresuotos prisijungti prie šios sistemos. Yra aibė programų, kurios padeda skrydžių operatoriams ir viešojo transporto paslaugų, komercinių krovinių gabenimo, tiek tolimojo reiso tiek miesto ribose pristatymus gabenantiems vairuotojams. Pavyzdžiui:

- transporto priemonių ir vairuotojų grafikų sudarymas, automatinė kelionių ataskaitų kompiliacija.
- optimalaus maršruto parinkimas, standartiniams ir nestandartiniais (didelių išmatavimų), transporto priemonėms.
- prekių lokacijos stebėjimas jų gabenimo metu ir jų fizinė būklė jei jos yra greitai gendančios.
- automatizacija komercinių ir kontrolės dokumentų, kurie turi lydėti krovinį ir transporto priemones.

Kuriama sistema yra tik tam tikra dalis visos ITS. Sistema turi padėti vairuotojams surasti kuo geresnį kelią (Paveikslas 1).



Keliai, davikliai, vaizdo kameros



Paveikslas 1. Kelionių planavimo sistema ITS kontekste

## **2.1.1. Projektiniai sprendimai**

### **2.1.1.1. TelemArk**

TelemArk – nacionalinė Suomijos sistemų architektūra sukurta 2000 m., apimanti 11 ITS paslaugų vystymą. Kadangi TelemArk buvo pradėta vystyti lygiagrečiai su KAREN, todėl yra didelių skirtumų tarp šių dviejų architektūrų. Kita vertus, KAREN taip pat buvo naudojama Suomijoje, ypač nustatant vartotojų poreikius, reikalingas funkcijas ir subfunkcijas ir duomenų srautus konkreitiems ITS panaudojimams. Transporto ir komunikacijos ministerija nusprendė paruošti abiejų sistemų panaudojimo gaires, išmesti persidengiančias vietas iš TelemArk ir pakeisti funkcijas į atitinkančias FRAME [7].

### **2.1.1.2. Actif**

Actif (Prancūzija) – naudoja Europos ITS architektūrą kaip atspirties tašką atsižvelgiant į Prancūzijos specifiškumus bei nacionalinius ir internacionalinius standartus. Actif kuriamos architektūros[8]:

- Elektroniniai apmokėjimai
- Avarių valdymas
- Eismo valdymas
- Viešasis transportas
- Pagalba vairuotojams
- Kelionių informacija
- Teisės aktų atitikimo užtikrinimas
- Krovinių gabenimo valdymas
- Archyvinių duomenų prieiga

### **2.1.1.3. ARTIST**

ARTIST (Italija) – architektūra, suderinta ir besiderinanti prie Europos ITS Framework architektūros ir glaudžiai susijusi su Prancūzijos nacionaline architektūra Actif. ARTIST susideda iš 7 pagrindinių elementų, nusakančių projekto kūrimo fazes:

- Šiuo metu kuriamų ITS sistemų tyrimo
- Vartotojų poreikių analizės
- Loginės architektūros
- Fizinės architektūros
- Organizacinės architektūros
- Navigacijos įrankio

- Specialiųjų terminų žodyno

## **2.1.2. Techniniai sprendimai**

### **2.1.2.1. Siemens sprendimas**

Siemens skelbia, jog eismo kontrolės sistemose jie yra pionieriai (nuo 1924m) ir internacionalinės rinkos lyderiai. Jie siūlo savo sistemą SITRAFFIC. Į ją įeina valdiklis SITRAFFIC C800V[16]. Jis naudojamas tiek paprastoms tiek sudėtingoms transporto valdymo sistemoms. Panaudotas 200MHz dažnio AK procesorius. Šis CPU yra 32 bitų procesorius, kuris gali valdyti iki 48 signalų grupių. Atmintis yra didelė ir gali būti praplėsta iki 9,5 Mb. C800V galima naudoti su pažangiomis vaizdo atpažinimo sistemomis. Tuo pačiu siūlomos kelios modifikacijos:

C800VK (C800V kompakt) modelis yra C800V valdiklis, kuris skirtas smulkiems taškams. Šis modelis gali būti sukonfigūruotas valdyti iki 8 signalų grupių.

C800VF (C800V pėstysis) modelis skirtas pėsčiųjų šviesoforams valdyti (iki 4 signalų grupių).

C800VP – su PLUS technologija, skirtas valdyti šalutinius ir pagrindinius šviesoforus.

Šiuose modeliuose yra SITRAFFIC Control – duomenų priėmimo programa.

SITRAFFIC Servise – tai įrankių krūva testavimams, komunikacijai, operacijoms, diagnostikai. SITRAFFIC Language (TL) – eismo inžinerijos problemoms spręsti skirta unikali programavimo sistema, turinti krūvelę modulių, skirtų spręsti atskiras dalines problemas. Siūlomos modernios eismo valdymo programos, tokios kaip SITRAFFIC MOTION arba SITRAFFIC SCOOT optimizuoja signalus kelių tinkle tiesiogiai. Viešajam transportui pirmenybę teikianti programa yra SITRAFFIC PDM.

Dar vienas siūlomas Siemens sprendimas miesto eismo valdymo kompiuteriai yra SITRAFFIC CENTRAL, ACTRA ir ICONS. Tai šakniniai metodai, kurie yra atsakingi už šviesoforų valdymą visame mieste. Operatoriams ir eismo planuotojams kontrolės centre pateikiamas aiškiai suprantamas vaizdas „iš viršaus“.

### **2.1.2.2. Telvent sprendimas**

Kaip ir Siemens, Telvent teigia yra pirmaujantys savo (energija, eismas, transportas ir aplinka) rinkoje Amerikoje, Ispanijoje ir Kinijoje. Ši įmonė jau virš 40 metų ir joje dirba virš 2300 profesionalų. Sprendimų objektai yra: sumažinti kamščius mieste, padidinti

tarpmiestinių kelių galimybes, padidinti naudotojų saugumą ir aprūpinti naudotojus reikalingais servisais. Telvent transporto valdymo sistemą sudaro[14]:

- Itaca (Intelektuali miesto transporto kontrolės sistema)
- RedEye (Raudonos šviesos pažeidimų kontrolės sistema)
- MoviSmart (Integruota eismo ir transportavimo sistema)
- Odyssey (ITS, autostrados ir miesto kelių naudojimo sistema)
- Sicutie (tunelių naudojimo ir valdymo sistema)

Miesto eismo bazinė kontrolės sistema yra OPTIMUS, kuri operuoja signalus pagal užprogramuotus taimerius. Sudėtingesnis dinaminis valdymas yra realizuotas ITACA sistemoje. ITACA – tai realiu laiku adaptuojanti sistema, kuri reguliuoja parametrus kiekvienoje sankryžoje, priklausomai nuo eismo sąlygų. Telvent sukūrė daugybę eismo valdiklių, kurie komunikuoja per varinius laidus arba optiką naudojant TCP/IP protokolą.

## **2.2. Maršrutų optimizavimo sistemų tyrimas**

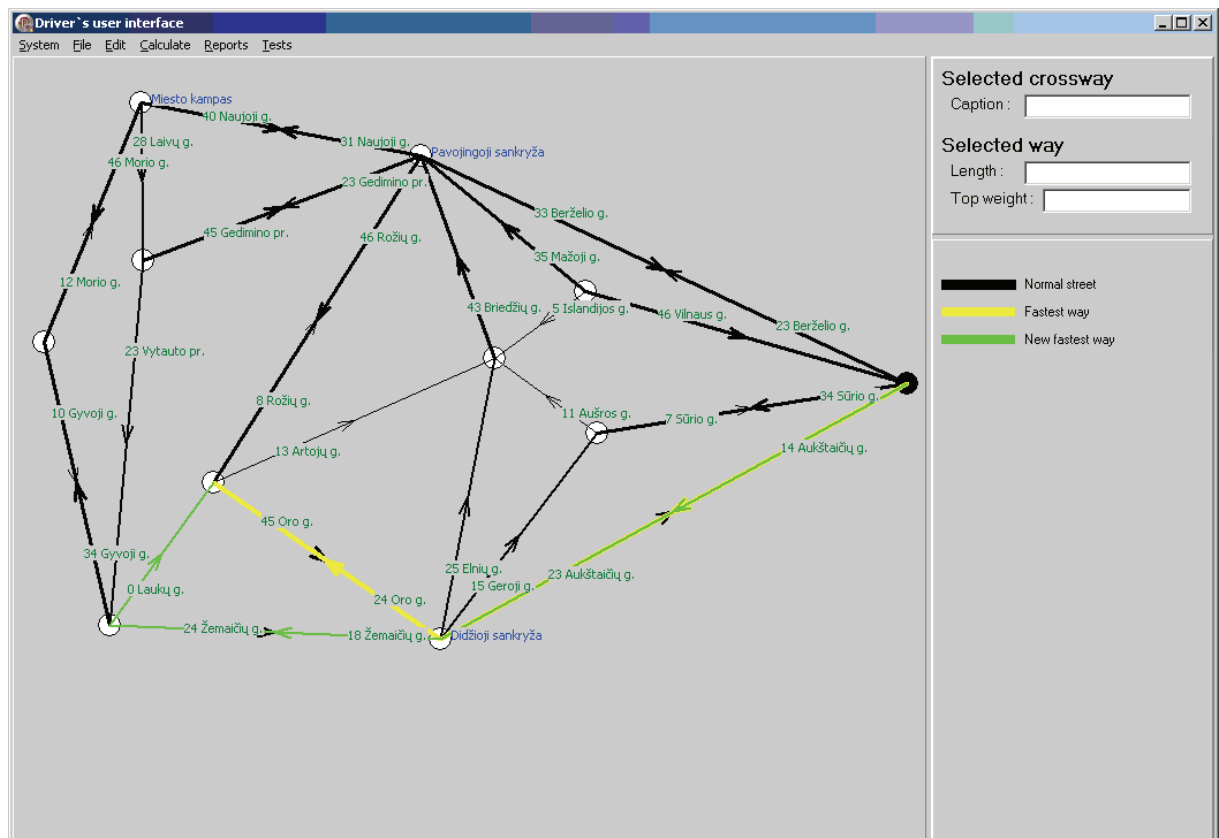
### **2.2.1. Statinės maršrutų sistemos**

Statinės maršrutų sistemos nepriklauso nuo vartotojo bei esamo momentinio kelių apkrovimo. Sistemos pateikiami maršrutai yra pagrįsti spėjimu, o ne realiu kelių apkrovimu. Šiais laikais daugybė įmonių yra pasirinkusios būtent tokį principą. Akivaizdus tokių maršrutų nustatymo trūkumas yra neįmanomas kamščių išvengimas dėl realaus (momentinio) transporto eismo būsenos informacijos trūkumo. Tokio tipo sistemos gali būti laikomos kaip pagalba randant kelią tarp pradinio ir galutinio kelionės taškų. Optimalaus kelio laike paieškos paskirtis atkrenta.

### **2.2.2. Dinaminės maršrutų sistemos**

Dinaminės maršrutų sistemos parenka maršrutus remiantis esama kelių apkrautumo būseną. Apkrautumas gali būti nuspėjamas ar paskaičiuojamas koku nors būdu, tačiau maršrutas paskaičiuojamas remiantis tikra, turima informacija. Dinaminio maršruto radimo pavyzdys pateiktas pirmame paveiksle. Čia tam tikrame fragmente greičiausias maršrutas iš pradinio taško į galutinį pažymėtas geltona linija. Po kelių sekundžių atrastas naujas maršrutas (žalia linija).





Paveikslas 2. Dinaminis maršrutų radimas

### 2.3. Trumpiausio kelio radimo algoritmų tyrimas

Greičiausio maršruto radimą galima prilyginti trumpiausio kelio radimo problemai svoriniame kryptiniame grafe. Viršūnės atitiktų sankryžas, kraštinės – kelius. Trumpiausio kelio algoritmui spręsti jau yra sukurta gerų algoritmų. Juos galime rasti pirmoje lentelėje[19]. Trumpiausiam keliui rasti algoritmai skaičiavimams naudoja kraštinių svorį. Jei šį svorį pakeistume atitinkamu koeficientu, algoritmas skaičiuos ne trumpiausią, bet greičiausią kelią. Koeficientas aptartas 2.3.1. skyriuje.

Numeris	Pavadinimas	Trumpas apibūdinimas
1	Dijkstra	Naudojamas rasti kelią tarp dviejų viršūnių tokiuose grafuose, kur kiekvienos briaunos svoris ne mažesnis už nulį. Algoritmu galima pakankamai efektyviai paskaičiuoti trumpiausią kelią nuo pradinės iki bet kurios kitos viršūnės.
2	Bellman-Ford	Randa trumpiausius kelius tokiuose grafuose, kurių

		briaunų svoris gali būti neigiamas
3	Floyd-Warshall	Randa trumpiausius kelius tarp kiekvienos viršūnių poros
4	Johnson	Randa trumpiausius kelius tarp kiekvienos viršūnių poros retame svoriniame kryptiniame grafe
5	DFS	Paieška į gylį nesvoriniame grafe
6	BFS	Paieška į plotį nesvoriniame grafe

Lentelė 1. Trumpiausio kelio radimo algoritmai

Algoritmai plačiau aptarti sekančiuose skyriuose. Įvairios algoritmų modifikacijos nenagrinėjamos, nes tai užimtų daugybę laiko.

### 2.3.1. Trumpiausio kelio radimas svoriniame grafe

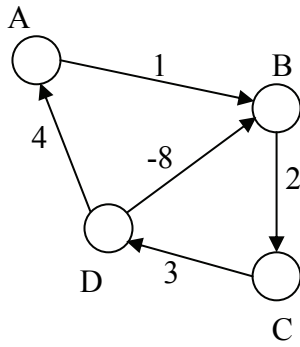
Grafe su svoriais kelio ilgį tarp viršūnių apibūdina kraštinių svorių suma.

„Trumpiausiojo kelio“ uždavinių yra keletas:

- rasti trumpiausiąjį kelią tarp 2 grafo viršūnių;
- rasti trumpiausiuosius kelius nuo vienos viršūnės iki visų likusių (*single source shortest path*);
- rasti trumpiausiuosius kelius iki duotosios viršūnės nuo visų likusių viršūnių (*single destination shortest path*);
- rasti trumpiausiuosius kelius tarp visų viršūnių porų (*all pairs shortest path*).

Įrodoma, kad trumpiausiojo kelio tarp dviejų viršūnių radimas grafe su svoriais yra tokio paties sudėtingumo uždavinys, kaip ir trumpiausiojo kelio radimas nuo vienos viršūnės iki visų likusių. Todėl trumpiausiojo kelio algoritmai paprastai kuriami antrajam, platesnės apimties uždaviniui. Jei grafo viršūnių svoriai visi teigiami, tai akivaizdu, kad joks trumpiausias kelias tarp dviejų viršūnių neturės ciklą.

Patyrinėkime atvejį, kai yra neigiamų svorių.



Paveikslas 3. Svorinis kryptinis grafas su neigiamais svoriais

Jei pradėsime suktis ciklu BCD, tai kelio ilgis tarp viršūnių A ir B artės į minus begalybę. Problemos trumpiausiojo kelio uždavinyje iškyla tuo atveju, kai yra neigiamų ciklų. Šiam uždaviniui spręsti naudojamas Bellman-Ford algoritmas, kuris aptartas vėlesniame skyriuje. Šis algoritmas dar moka pranešti, apie neigiamo ciklo buvimą grafe.

Vieno šaltinio trumpiausiojo kelio uždavinio sprendinys yra medžio struktūros pradinio grafo poaibis. Teiginys nėra pilnai akivaizdus, todėl jis bus įrodytas.

**Teorema.** Trumpiausio kelio daliniai keliai grafe yra trumpiausieji keliai. Tarkime duotas orientuotas grafas  $G=(V,L)$  su svoriais  $w:L \rightarrow R$ . Tarkime kelias  $p=(v_1, v_2, \dots, v_k)$  yra trumpiausias kelias tarp viršūnių  $v_1$  ir  $v_k$ . Bet kuriems  $i, j, 1 \leq i \leq j \leq k$  sukurkime kelią  $p_{ij}$  ir pavadinkime jį kelio  $p$  daliniu keliu tarp viršūnių  $v_i$  ir  $v_j$ . Tada dalinis kelias  $p_{ij}$  bus trumpiausias kelias tarp viršūnių  $v_i$  ir  $v_j$ .

**Įrodymas.** Padalinkime kelią  $p$  į dalis:

$$v_1 \xrightarrow{p_{1i}} v_i \xrightarrow{p_{ij}} v_j \xrightarrow{p_{jk}} v_k$$

Tuomet kelio ilgis bus

$$w(p) = w(p_{1i}) + w(p_{ij}) + w(p_{jk}).$$

Tarkime egzistuoja kelias  $p'_{ij}$  nuo  $v_i$  iki  $v_j$ , toks, kad  $w(p'_{ij}) < w(p_{ij})$ . Tada kelio  $p'$

$$v_1 \xrightarrow{p_{1i}} v_i \xrightarrow{p'_{ij}} v_j \xrightarrow{p_{jk}} v_k$$

ilgis  $w(p') = w(p_{1i}) + w(p'_{ij}) + w(p_{jk}) < w(p)$ . Tai prieštarauja prielaidai, kad kelias  $p$  yra trumpiausias kelias tarp viršūnių  $v_1$  ir  $v_k$ .

**Įrodymo pabaiga.**

Įrodžius teoremą, tampa akivaizdu, kad trumpiausiais keliais gali būti medžio tipo struktūra, prasidedanti nuo pradinės viršūnės. Jei ieškosime kelio tarp viršūnių  $s$  ir  $v$ , pavadinkime jį  $p_{sv}$ , tai kelias susidės iš  $p_{sx}$  ir lanko  $(x,v)$ , kur  $x$  – kažkuri viršūnė, o  $p_{sx}$  – trumpiausias kelias nuo viršūnės  $s$  iki viršūnės  $x$ .

### 2.3.2. Grafo svorio tyrimas

Keliamas uždavinys yra greičiausio kelio radimas. Kadangi ši problema sprendžiama realybėje, vadinasi šį skaičių įtakoja ne tik automobilių skaičius, kelių ilgis, plotis, juostų skaičius, bet ir psichologiniai faktoriai (pvz. pastovus, bet lėtas judėjimas žmogui atrodo kur kas spartesnis, nei ilgesnė pauzė ir greitesnis judėjimas, nors pastaruoju atveju nukeliamas atstumas yra ilgesnis) ir kiti panašūs dalykai. Iš viso to išplaukia išvada, jog koeficiento radimas yra labai didelis uždavinys, reikalaujantis, turbūt didesnės apimties nei magistro darbas tyrimo.

Šiuo atveju koeficientas bus įvertintas projektiniais faktoriais, kurie nesunkiai paskaičiuojami. Tokie parametrai yra: kelio ilgis, plotis, apkrovimas, maksimalus greitis. Įvertinus šiuos parametrus išplaukia sekanti formulė:

$$\text{Svoris} = (\text{kelio apkrovimas}) * (\text{kelio ilgis} / \text{maksimalus greitis}) / (\text{kelio maksimalus apkrovimas})$$

### 2.3.3. Dijkstra algoritmo analizė

Dijkstra - algoritmas, kurį sukūrė kompiuterių specialistas *Edgar Dijkstra*; sprendžia vieno šaltinio trumpiausių kelių problemą kryptiniame grafe su ne neigiamais kraštinių svoriais.

Pavyzdžiui, jei grafo viršūnės vaizduoja miestus ir kraštinių svoriai vaizduoja atstumą tarp tų miestų, sujungtą tiesioginiu keliu, Dijkstra algoritmas naudojamas surasti trumpiausius kelius tarp tų miestų.

Algoritmo pradiniai duomenys yra kryptinis grafas  $G$  su kraštinių svoriais ir šaltinių viršūne  $S$  grafe  $G$ . Mes pavadinsime  $V$  visų viršūnių rinkinį grafe  $G$ . Kiekviena grafo kraštinė yra sutvarkyta viršūnių pora  $(u,v)$ , parodanti ryšį tarp viršūnių  $u$  ir  $v$ . Visų kraštinių rinkinys yra  $E$ . Kraštinių svoriai yra užduodami pagal svorių funkciją  $w:E$ ; čia  $w(u,v)$  yra ne neigiama kaina tiesiogiai einant nuo viršūnės  $U$  iki viršūnės  $V$ . Kraštinės kaina yra laikoma atstumu tarp šitų dviejų viršūnių. Kelio kaina tarp dviejų viršūnių yra kraštinių kainų suma tame kelyje.

Duotai porai viršūnių  $s$  ir  $t$  visų viršūnių rinkinyje  $V$  algoritmas suranda kelią nuo  $s$  į  $t$  su mažiausia kaina (trumpiausias kelias). Jis taip pat gali būti naudojamas rasti trumpiausių kelių kainai nuo vienos viršūnės  $s$  iki kitų viršūnių grafe.

Algoritmas dirba surasdamas kiekvienai viršūnei trumpiausio kelio kainą  $d[v]$  rastą tame kelyje tarp  $s$  ir  $v$ . Iš pradžių ši vertė yra 0 šaltinio viršūnei  $s$  ( $d(s)=0$ ) ir begalybė kitoms viršūnėms, pripažįstant faktą, kad mes nežinome jokių kelių iki tų viršūnių ( $d[v]=$  begalybė kiekvienam  $v$  iš  $V$ , išskyrus  $s$ ). Kai algoritmas baigsis,  $d[v]$  bus trumpiausias kelias nuo  $s$  iki  $v$ , ar begalybė, jei toks kelias neegzistuoja.

Pagrindinė Dijkstra algoritmo operacija yra kraštinių atlaisvinimas, jei yra kraštinė nuo  $u$  iki  $v$ , tai trumpiausias žinomas kelias nuo  $s$  iki  $u$  ( $d[u]$ ) gali būti išplėstas į kelią nuo  $s$  iki  $v$ , pridėdam kraštinę  $(u, v)$  prie galo. Toks kelias turės ilgį  $d[u]+w(u,v)$ . Jei tai yra mažiau negu esamas  $d[v]$ , mes galime pakeisti esamą vertę  $d[v]$  nauja verte. Kraštinių atlaisvinimas yra taikomas, kol visos vertės  $d[v]$  atitinka trumpiausio kelio kainą nuo  $s$  iki  $v$ . Algoritmas yra organizuojamas taip, kad kiekviena kraštinė yra atlaisvinama tik vieną kartą, kai  $d[u]$  pasiekia savo galutinę vertę. Taigi algoritmo pseudokodas atrodytu štai taip:

```
function Dijkstra(G, w, s)
  for each v in V[G] // Inicializacija. Kiekvienai viršūnei v..
    d[v] := begalybė
    paskutinė[v] := nil //nenustatytas
  d[s] := 0
  S := tuščia aibė
  Q := V[G]
  while Q nėra tuščia aibė // pats algoritmas
    u := Extract_Min(Q)
    S := S union {u} // S ← S ∪ {u}
    for each (u,v) išeinanti iš u
      if d[u] + w(u,v) < d[v] // relaksacija
        d[v] := d[u] + w(u,v)
        paskutinė[v] := u
```

[vykdę Dijkstra algoritmą, trumpiausią kelią rasti galime taip:

```
S := tuščia seka
u := t
while turi reikšmę paskutinė[u]
  insert u to the beginning of S // įterpiame į S pradžia
  u := paskutinė[u]
```

Taigi  $S$  yra mūsų ieškomas trumpiausias kelias iš viršūnės  $s$  į  $t$ . Jei  $S$  yra tuščia seka, vadinasi kelias tarp šių viršūnių neegzistuoja.

### 2.3.4. Bellman-Ford algoritmo analizė

Šis algoritmas suranda trumpiausią kelią svoriniame grafe, kurio kraštinių svoriai gali būti neigiami. Dijkstra algoritmas išsprendžia tą pačią problemą greičiau, tačiau kraštinės

svoriai turi būti ne mažesni už nulį[21]. Paprastai Bellman-Ford algoritmas naudojamas grafuose su neigiamais svoriais. Bellman-Ford algoritmo pseudokodas yra labai panašus į Dijkstra ir atrodo šitaip:

```
// apibrėžiame duomenų tipus
record vertex { // viršūnė
    list edges // kraštinių sąrašas
    real distance // atstumas
    vertex predecessor // prieš taiėjusi viršūnė (toliau - tėvas)
}
record edge { // kraštinė
    node source // pradinė viršūnė
    node destination // galutinė viršūnė
    real weight // svoris
}

// pati funkcija
function BellmanFord(list vertices, list edges, vertex source)

    // grafo inicializacija
    for each vertex v in vertices:
        if v is source then v.distance = 0
        else v.distance := begalybė
        v.predecessor := nil // tėvas nenustatytas

    // trumpiausio kelio paieška
    for i from 1 to size(vertices):
        for each edge uv in edges:
            u := uv.source
            v := uv.destination // uv yra kraštinė iš u į v
            if v.distance > u.distance + uv.weight
                v.distance := u.distance + uv.weight
                v.predecessor := u

    // neigiamų ciklų paieška
    for each edge uv in edges:
        u := uv.source
        v := uv.destination
        if v.distance > u.distance + uv.weight
            error "Surastas neigiamas ciklas"
```

### 2.3.5. Floyd-Warshall algoritmo analizė

Floyd-Warshall algoritmas (dar žinomas kaip Roy-Floyd arba Warshall algoritmas) yra skirtas surasti trumpiausius kelius tarp visų viršūnių porų svoriniame kryptiniame grafe.

Šiam algoritmui reikalingi duomenys yra svorinis kryptinis grafas  $(V, E)$ . Grafo kraštinės  $E$ , gali turėti neigiamus svorius, tačiau grafe negali būti neigiamo svorio ciklų.

### 2.3.6. Johnson algoritmo analizė

Johnson algoritmas skirtas surasti trumpiausius kelius tarp visų reto svorinio kryptinio grafo viršūnių. Pirmiausia pridėdama nauja viršūnė iš kurios į visas kitas eina nulinio svorio viršūnės ir įvykdomas Bellman-Ford algoritmas (patikrinama ar nėra neigiamo svorio ciklų).

Taip surandamas mažiausio svorio kelias  $h(v)$  iš naujosios viršūnės į viršūnę  $v$ . Visų kraštinių svoriai sumažinami dydžiu  $h(v)$  ir kiekvienai viršūnei įvykdomas Dijkstros algoritmas.

### 2.3.7. „Paieška į gylį“ algoritmo analizė

Paieška į gylį (angl. *Depth-First Search* arba DFS) – tai tokia paieška, kurios pagrindinis principas yra toks: pasirenkama pradinė viršūnė ir einama grafu kiek įmanoma giliau; tada grįžtama iki artimiausios neaplankytos briaunos ir vėl ieškoma kuo giliau. Šis būdas priešpastatomas paieškai į plotį, kur esant viršūnėje  $v$  pirma aplankomos visos artimos  $v$  viršūnės (sujungtos su  $v$  briauna), tik tada pereinama prie kitos viršūnės.

Kadangi šis algoritmas skirtas nesvoriniam grafui, jo gilesnė analizė neatliekama.

### 2.3.8. „Paieška į plotį“ algoritmo analizė

Paieška į plotį (angl. *Breadth-first search* arba BFS) – paieškos algoritmas. Pagrindinis principas – aplankomos kaimyninės viršūnės. Tada aplankoma kiekvienos tos atrastos viršūnės kaimyninės viršūnės ir t.t., kol atrandama reikiama viršūnė. Šis algoritmas nėra optimalus, tačiau atlikti gali pilną paiešką. Deja, tai yra ne svoriniam grafui pritaikytas algoritmas, todėl jo gilesnė analizė neatliekama.

### 3. Trumpiausio kelio radimo algoritmų įvertinimas

Uždaviniui spręsti tinka tik keli algoritmai. Šiame skyriuje juos ir nustatysime.

#### 3.1. Trumpiausio kelio radimo algoritmų atranka

Tikslo uždavinyje sudaromas kryptinis svorinis neneigiamų svorių grafikas. Antroje lentelėje matosi trumpiausio kelio radimo algoritmų savybės. Remiantis ta lentelė iš kart atkrinta DFS, BFS algoritmai, dėl grafo tipo skirtumo. Johnson algoritmas taip pat atmetamas dėl neefektyvumo, nes yra vykdomi yra du papildomi algoritmai. Kadangi naudojamos sistemos vartotojas ieškos vieno ar kelių maršrutų, jam nereikia paieškos rezultatų, kuriuos siūlo trečiasis algoritmas. Lieka galimi pirmi du algoritmai. Pagal sąlyga neigiamo svorio kraštinių neturėtų būti, tačiau modifikuojant ateityje formulę iš 2.3.1. skyriaus, gali atsirasti ir neigiamų svorių. Taigi tirsime Dijkstra ir Bellman-Ford algoritmus.

Numeris	Pavadinimas	Grafo tipas	Papildomos pastabos
1	Dijkstra	Svorinis, kryptinis	Kraštinės turi būti tik teigiamo svorio
2	Bellman-Ford	Svorinis, kryptinis	Kraštinės gali būti neigiamo svorio
3	Floyd-Warshall	Svorinis, kryptinis	Paieška tarp visų kraštinių, negali būti neigiamų svorio ciklą
4	Johnson	Svorinis, kryptinis, retas	Naudoja papildomai Dijkstra ir Bellman-Ford algoritmus
5	DFS	Nesvorinis	Nėra
6	BFS	Nesvorinis	Nėra

Lentelė 2. Algoritmų savybių palyginimų lentelė

#### 3.2. Dijkstra ir Bellman-Ford algoritmų palyginimas

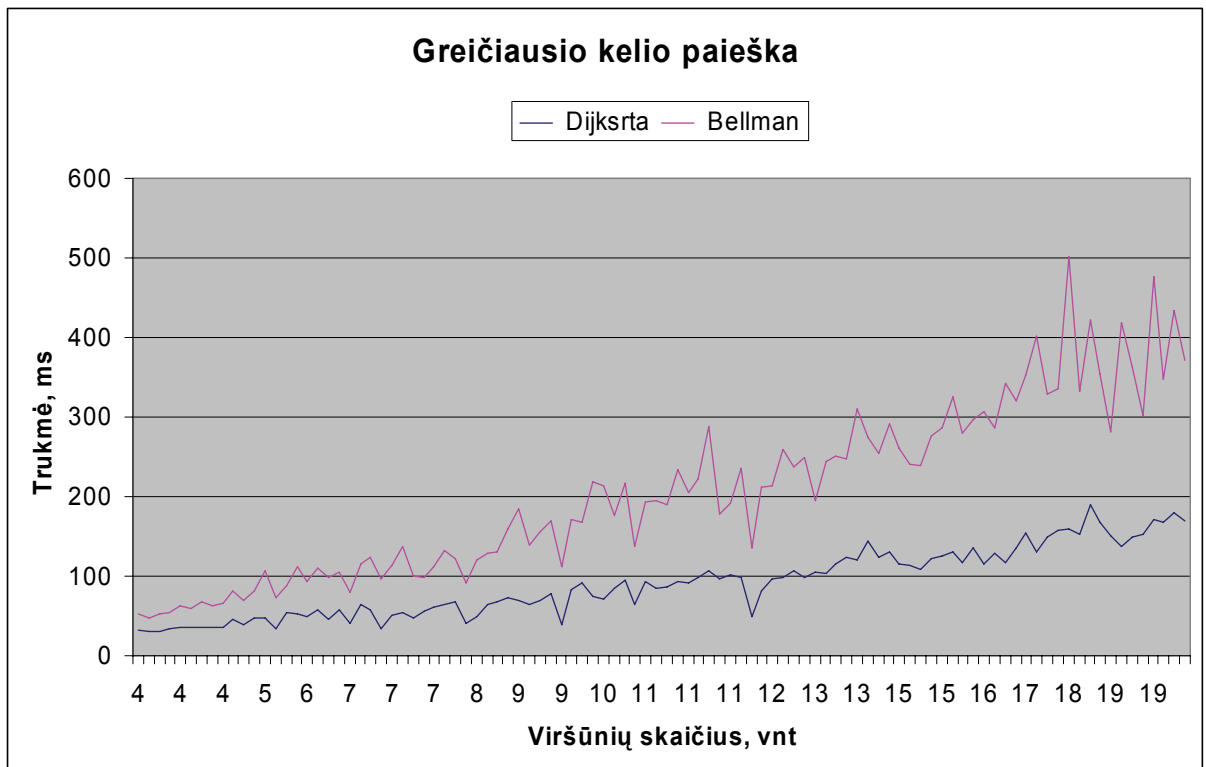
Atlikti šiems palyginimams buvo sukurta atskira programa. Programa sugeneruoja tam tikrą kiekį grafų, su skirtingu viršūnių ir kraštinių skaičiumi. Abu algoritmai skaičiuoja nuo tos pačios viršūnės trumpiausius kelius į visas kitas grafo viršūnes.



### 3.2.1. Algoritmų palyginimas mažo dydžio grafuose

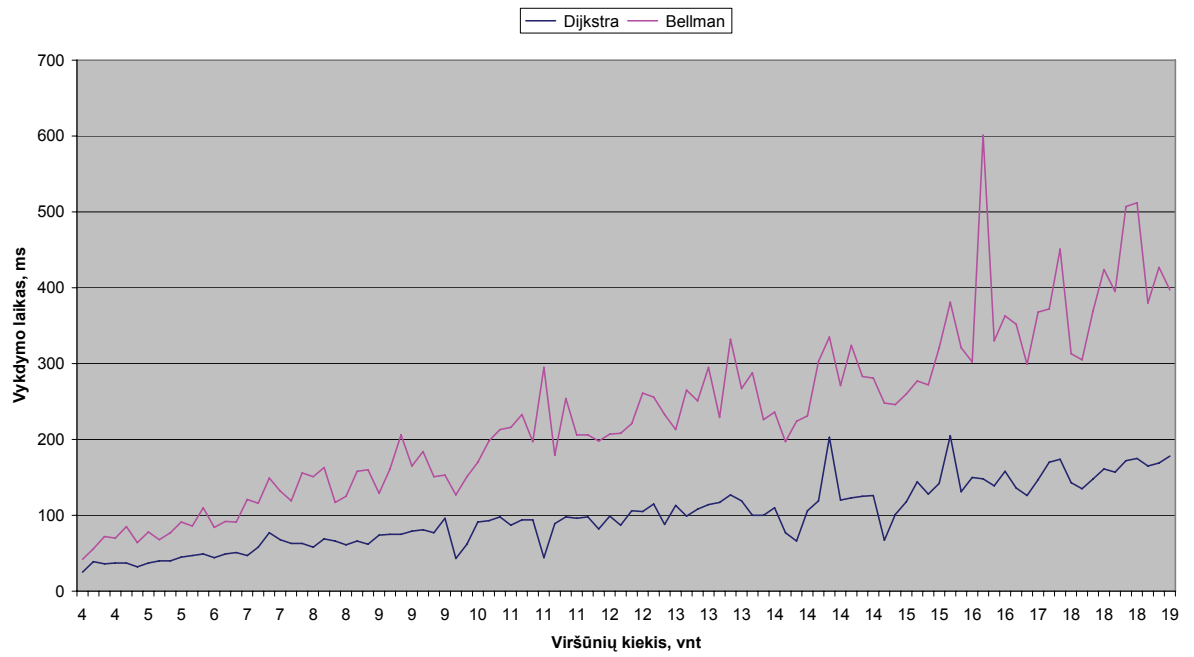
- Algoritmai vykdomi po 100 kartų.
- Generuojama 100 skirtingų grafų
- Viršūnių skaičius svyruoja nuo 4 iki 20.

Eksperimento duomenys atvaizduoti sekančiuose dvejuose grafikuose.



Paveikslas 4. Pirmasis eksperimento bandymas (maži grafai)

### Greičiausio kelio paieška (2)



Paveikslas 5. Antrasis eksperimento bandymas (maži grafai)

### 3.2.2. Algoritmų palyginimas didelio dydžio grafuose

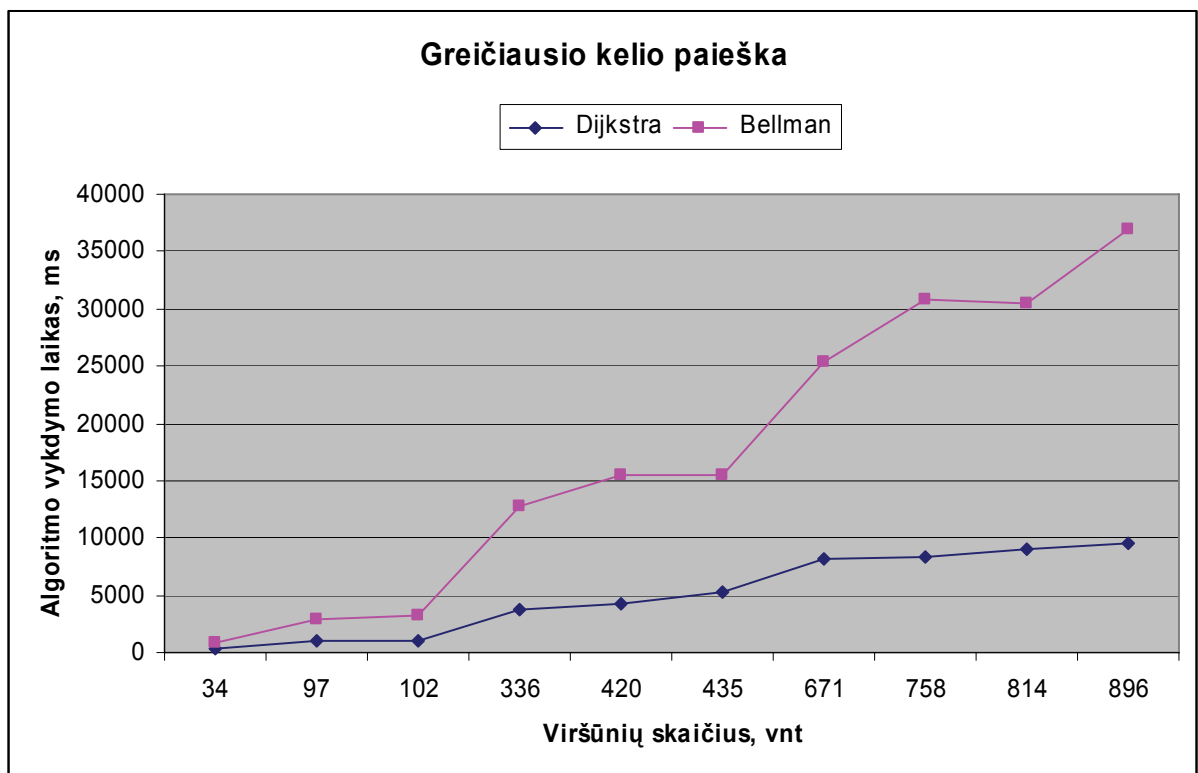
- Generuojama 10 skirtingų grafų
- Viršūnių skaičius svyruoja nuo 4 iki 10000.

Eksperimento duomenys atvaizduoti sekančiuose dvejuose grafikuose bei lentelėse.

Algoritmai vykdomi 10 kartų.

Viršūnių skaičius	Kraštinių skaičius	Dijkstra algoritmo vykdymo trukmė (ms)	Bellman-Ford algoritmo vykdymo trukmė (ms)
34	102	304	837
97	301	951	2941
102	312	1030	3315
336	995	3719	12736
420	1248	4190	15514
435	1339	5312	15491
671	2015	8095	25445
758	2248	8326	30744
814	2380	9006	30463
896	2668	9590	36921

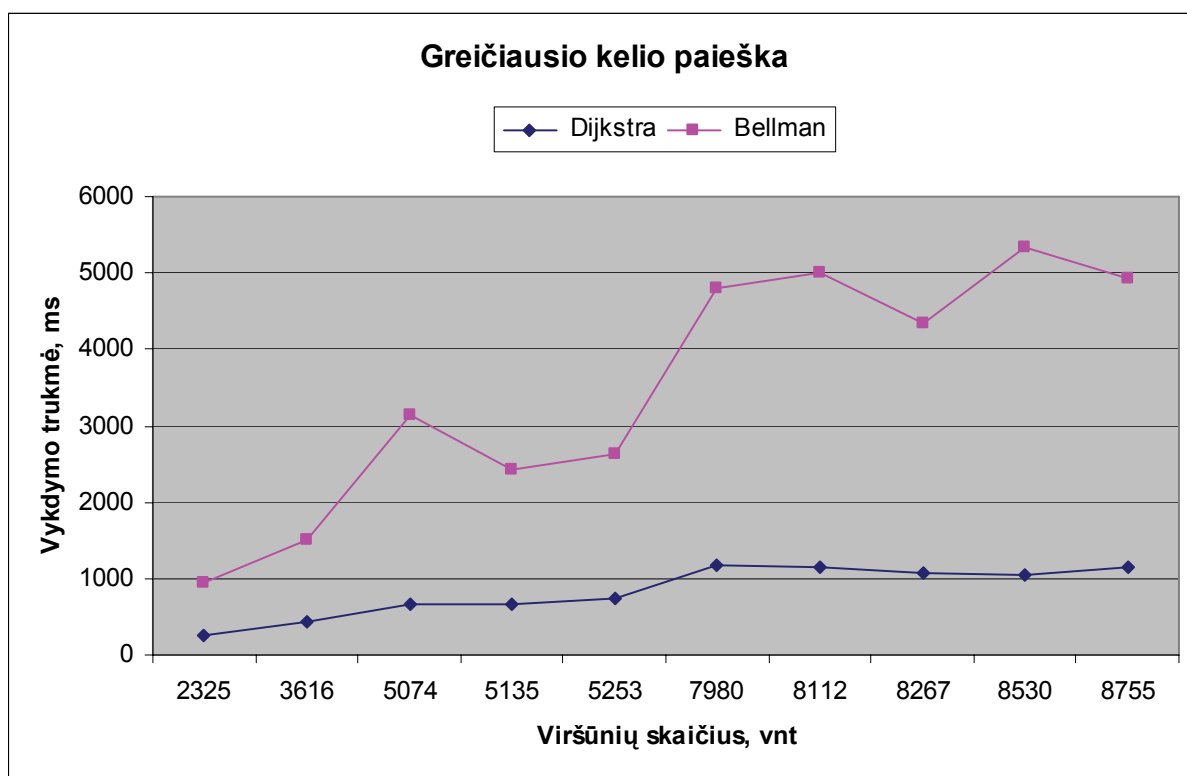
Lentelė 3. Algoritmų palyginimas ieškant grafuose su dideliu viršūnių kiekiu (pirmasis eksperimentas)



Paveikslas 6. Paieška dideliuose grafuose (pirmasis eksperimentas)

Viršūnių skaičius	Kraštinių skaičius	Dijkstra algoritmo vykdymo trukmė (ms)	Bellman-Ford algoritmo vykdymo trukmė (ms)
2325	6997	267	954
3616	10976	445	1512
5074	15358	674	3152
5135	15553	656	2421
5253	15857	737	2642
7980	23970	1173	4801
8112	24397	1143	5015
8267	24714	1065	4342
8530	25724	1055	5338
8755	26138	1151	4933

Lentelė 4. Algoritmų palyginimas ieškant grafuose su dideliu viršūnių kiekiu (antrasis eksperimentas)



Paveikslas 7. Paieška dideliuose grafuose (antrasis eksperimentas)

### 3.2.3. Algoritmų palyginimo rezultatai

Visais atvejais Dijkstra algoritmas vykdymo trukme lenkia Bellman-Ford algoritmą. Bellman-Ford algoritmo vykdymo trukmė didesne 3-6 kartus.

Bellman-Ford algoritmas netinkamas naudoti su dideliais grafais, kadangi toleruojamas algoritmo vykdymo laikas gali būti iki 1.5 sekundės.

#### 4. Greičiausio kelio radimo sistemos prototipas

Sistemos architektūrai pavaizduoti yra naudojami penki vaizdai: panaudojimo atvejų, loginis, procesų, išdėstymo ir duomenų. Visi jie ir juos sudarantys modeliavimo elementai yra pateikti 5 lentelėje.

Vaizdas	Sudarantys elementai
Panaudojimo atvejų	Panaudojimo atvejų diagramos.
Loginis	Klasių diagramos; Sistemos išskaidymas į paketus.
Procesų	Būsenų diagramos; Sekų diagramos; Bendradarbiavimo diagramos; Veiklos diagramos.
Išdėstymo	Išdėstymo diagramos.
Duomenų	Duomenų bazės modelis

Lentelė 5. Architektūros modeliavimo elementai

## 4.1. Panaudojimo atvejų vaizdas



Paveikslas 8. Panaudojimo atvejų diagrama

#### 4.1.1.1. Panaudojimo atvejų sąrašas

Numeris	1
Tikslas	Prisijungti vartotojui prie sistemos administratoriaus teisėmis
Aktoriai	Administratorius
Nefunkciniai reikalavimai	Nėra
Prieš-sąlygos	Nėra
Sužadinimo sąlyga	Sužadinama vartotojui pasirinkus atitinkamą meniu punktą
Po-sąlyga	Vartotojas prisijungęs administratoriaus teisėmis
Pagrindinis scenarijus	Vartotojas startuoja programą Įvedamas vartotojo vardas, slaptažodis Sistema tikrina prisijungimo duomenis Sistema pateikia administratoriaus langą
Alternatyvūs scenarijai	Neteisintas vartotojo vardas arba slaptažodis. Sistema pateikia klaidos pranešimą

Lentelė 6. Administratoriaus prisijungimas

Numeris	2
Tikslas	Nustatyti, pakeisti sistemos parametrus
Aktoriai	Administratorius
Nefunkciniai reikalavimai	nėra
Prieš-sąlygos	Turi būti prisijungęs administratoriaus teisėmis
Sužadinimo sąlyga	Iškviečiamas vartotojo pasirinkimu iš pagrindinio meniu
Po-sąlyga	Sistemos parametrai tampa pakeisti,



	apie tai pranešama vartotojui
Pagrindinis scenarijus	<ol style="list-style-type: none"> <li>1) Tikrinama, ar prisijungęs vartotojas administratoriaus teisėmis</li> <li>2) Administratorius pasirenka naujas parametrų reikšmes</li> <li>3) Sistemoje pakeičiami parametrai. Pateikiamas pranešimas apie tai</li> </ol>
Alternatyvūs scenarijai	3a) Jei duomenys neišsaugoti, gražinamas pranešimas apie klaidą

Lentelė 7. Parametrų nustatymas

Numeris	3
Tikslas	Nustatyti, pakeisti sistemos duomenis
Aktoriai	Administratorius
Nefunkciniai reikalavimai	nėra
Prieš-sąlygos	Turi būti prisijungęs administratoriaus teisėmis
Sužadinimo sąlyga	Iškviečiamas vartotojo pasirinkimu iš pagrindinio meniu
Po-sąlyga	Sistemos duomenys tampa pakeisti, apie tai pranešama vartotojui
Pagrindinis scenarijus	<ol style="list-style-type: none"> <li>1) Tikrinama, ar prisijungęs vartotojas administratoriaus teisėmis</li> <li>2) Administratorius parenka norimą peržiūrą</li> <li>3) Administratorius koreguoja reikiamus duomenis (pvz. avarija kelio atkarpoje)</li> <li>4) Sistemoje pakeičiami duomenys. Pateikiamas pranešimas apie tai</li> </ol>
Alternatyvūs scenarijai	4a) Jei duomenys neišsaugoti, gražinamas pranešimas apie klaidą

Lentelė 8. Stebėjimas koregavimas

Numeris	4
Tikslas	Gauti statistines ataskaitas
Aktoriai	Administratorius
Nefunkciniai reikalavimai	nėra
Prieš-sąlygos	Turi būti prisijungęs administratoriaus teisėmis
Sužadinimo sąlyga	Iškviečiamas vartotojo pasirinkimu iš pagrindinio meniu
Po-sąlyga	Vartotojui pateikta ataskaita
Pagrindinis scenarijus	<ol style="list-style-type: none"> <li>1) Tikrinama, ar prisijungęs vartotojas administratoriaus teisėmis</li> <li>2) Administratorius pasirenka ataskaitos tipą, jei reikia įveda ataskaitos parametrus</li> <li>3) Sistema pateikia suformuotą ataskaitą</li> </ol>
Alternatyvūs scenarijai	3a) Klaidos atveju gražinamas pranešimas

Lentelė 9. Ataskaitų formavimas

Numeris	5
Tikslas	Prisijungti vartotojui prie sistemos
Aktoriai	Vairuotojas
Nefunkciniai reikalavimai	Vairuotojas turi turėti specialų įrenginį su programine įranga
Prieš-sąlygos	Nėra
Sužadinimo sąlyga	Vartotojas startuoja įrenginį ir programą
Po-sąlyga	Vartotojas prisijungęs prie sistemos
Pagrindinis scenarijus	<ol style="list-style-type: none"> <li>1) Vartotojas startuoja įrenginį ir programą</li> </ol>

	2) Programa identifikuojasi 3) Sistema pateikia vairuotojo langą
Alternatyvūs scenarijai	3a) Nepavykus identifikacijai pateikiamas klaidos pranešimas

Lentelė 10. Vairuotojo prisijungimas

Numeris	6
Tikslas	Pagal pateiktus parametrus nustatyti maršrutą, kuriuo keliaudamas vairuotojas pasiektų tikslą
Aktoriai	Vairuotojas
Ryšiai su kitais panaudojimo atvejais	Apibendrina panaudojimo atvejus: 7 Maršruto suradimas įvertinant trukmę, 8 Maršruto suradimas įvertinant atstumą
Nefunkciniai reikalavimai	nėra
Prieš-sąlygos	Vartotojas prisijungęs prie sistemos
Sužadinimo sąlyga	Iškviečiamas vartotojo pasirinkimu iš pagrindinio meniu, po to periodiškai tai atlieka sistema
Po-sąlyga	Vairuotojui pateiktas maršrutas kaip nuo jo buvimo vietos galima pasiekti nurodytą tikslą
Pagrindinis scenarijus	1) Tikrinama, ar identifikuotas vartotojas 2) Vairuotojas pasirenka pradinę ir galutinę vietas 3) Kreipiamasi į duomenų bazę ir gaunami užklausti duomenys 4) Atliekama paieška ir gražinama informacija apie rastą maršrutą

	<p>5) Pagal gautą rezultatą pateikiamas maršrutas vartotojui</p> <p>5a) pateikiamas jau buvęs maršrutas</p> <p>5b) pateikiamas naujas maršrutas (avarijos atveju apie tai sistema praneša vartotojui)</p>
Alternatyvūs scenarijai	3a) Klaidos atveju, gražinamas pranešimas

Lentelė 11. Maršruto radimas

Numeris	7
Tikslas	Pagal pateiktus parametrus nustatyti maršrutą, kuriuo keliaudamas vairuotojas greičiausiai pasiektų tikslą
Aktoriai	Vairuotojas
Ryšiai su kitais panaudojimo atvejais	Specializuoja panaudojimo atvejį: 6 Maršruto suradimas
Nefunkciniai reikalavimai	nėra
Prieš-sąlygos	Vartotojas prisijungęs prie sistemos
Sužadinimo sąlyga	Iškviečiamas vartotojo pasirinkimu iš pagrindinio meniu, po to periodiškai tai atlieka sistema
Po-sąlyga	Vairuotojui pateikiamas maršrutas kaip greičiausiai nuo jo buvimo vietos galima pasiekti nurodytą tikslą
Pagrindinis scenarijus	<ol style="list-style-type: none"> <li>1) Tikrinama, ar identifiкуotas vartotojas</li> <li>2) Vairuotojas pasirenka pradinę ir galutinę vietas</li> <li>3) Kreipiamasi į duomenų bazę ir gaunami užklausti duomenys</li> <li>4) Atliekama paieška (paieškos algoritme naudojamas trukmės parametras) ir</li> </ol>

	<p>gražinama informacija apie rastą maršrutą</p> <p>5) Pagal gautą rezultatą pateikiamas maršrutas vartotojui</p> <p>5a) pateikiamas jau buvęs maršrutas</p> <p>5b) pateikiamas naujas maršrutas (avarijos atveju apie tai sistema praneša vartotojui)</p>
Alternatyvūs scenarijai	3a) Klaidos atveju, gražinamas pranešimas

Lentelė 12. Maršruto suradimas įvertinant trukmę

Numeris	8
Tikslas	Pagal pateiktus parametrus nustatyti maršrutą, kuriuo keliaudamas vairuotojas pasiektų tikslą nuvažiuodamas mažiausią kelią
Aktoriai	Vairuotojas
Ryšiai su kitais panaudojimo atvejais	Specializuoja panaudojimo atvejį: 6 Maršruto suradimas
Nefunkciniai reikalavimai	nėra
Prieš-sąlygos	Vartotojas prisijungęs prie sistemos
Sužadinimo sąlyga	Iškviečiamas vartotojo pasirinkimu iš pagrindinio meniu, po to periodiškai tai atlieka sistema
Po-sąlyga	Vairuotojui pateikiamas maršrutas kaip greičiausiai nuo jo buvimo vietos galima pasiekti nurodytą tikslą
Pagrindinis scenarijus	<ol style="list-style-type: none"> <li>1) Tikrinama, ar identifiukuotas vartotojas</li> <li>2) Vairuotojas pasirenka pradinę ir galutinę vietas</li> <li>3) Kreipiamasi į duomenų bazę ir gaunami</li> </ol>

	<p>užklausti duomenys</p> <p>4) Atliekama paieška (paieškos algoritme naudojamas atstumo parametras) ir gražinama informacija apie rastą maršrutą</p> <p>5) Pagal gautą rezultatą pateikiamas maršrutas vartotojui</p> <p style="padding-left: 40px;">5a) pateikiamas jau buvęs maršrutas</p> <p style="padding-left: 40px;">5b) pateikiamas naujas maršrutas (avarijos atveju apie tai sistema praneša vartotojui)</p>
Alternatyvūs scenarijai	3a) Klaidos atveju, gražinamas pranešimas

Lentelė 13. Maršruto suradimas įvertinant atstumą

Numeris	9
Tikslas	Gauti informaciją apie esamas avarijas
Aktoriai	Vairuotojas
Nefunkciniai reikalavimai	nėra
Prieš-sąlygos	Vartotojas prisijungęs prie sistemos
Sužadinimo sąlyga	Iškviečiamas vartotojo pasirinkimu iš pagrindinio meniu
Po-sąlyga	Vartotojui pateikta informacija apie avarijas
Pagrindinis scenarijus	<p>1) Tikrinama, ar identifikuotas vartotojas</p> <p>2) Kreipiamasi į duomenų bazę ir gaunami užklausti duomenys</p> <p>3) Pateikiama informacija apie avarijas</p>
Alternatyvūs scenarijai	3a) Klaidos atveju gražinamas pranešimas

Lentelė 14. Avarijų peržiūra

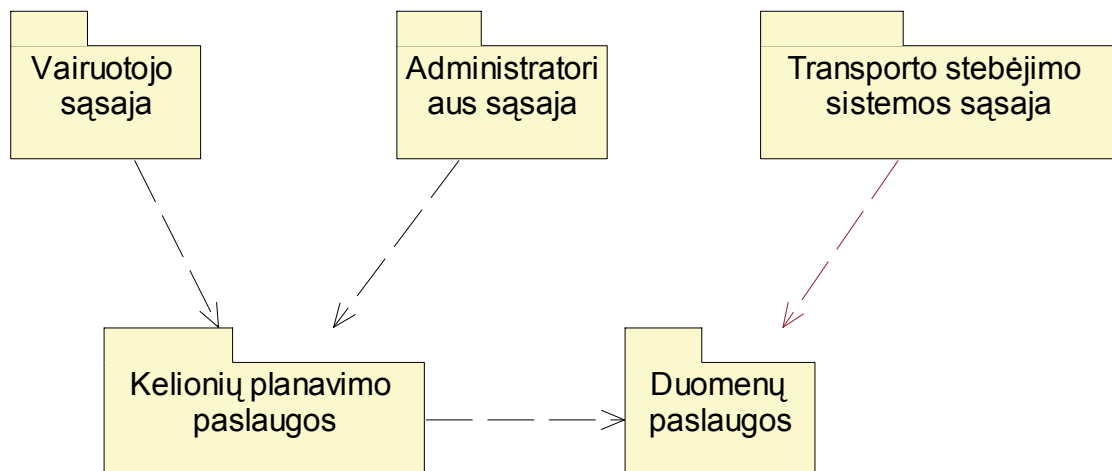
Numeris	10
Tikslas	Papildyti duomenų bazę naujais duomenimis
Aktoriai	Transporto stebėjimo sistema
Nefunkciniai reikalavimai	Nėra
Prieš-sąlygos	Turi būti prijungta prie sistemos
Sužadinimo sąlyga	Sužadinamas automatiškai atitinkamu periodu
Po-sąlyga	Duomenų bazė tampa papildyta naujais duomenimis
Pagrindinis scenarijus	<ol style="list-style-type: none"> <li>1) Tikrinama, ar nesibaigė duomenų atnaujinimo laikas (tam tikru periodu)</li> <li>2) Jei nesibaigė, vykdomas pirmasis punktas</li> <li>3) Jei baigėsi, tikrinami ar duomenys teisingi ir įrašomi nauji duomenys apie transporto srautą</li> <li>4) Vykdomas pirmasis punktas</li> </ol>
Alternatyvūs scenarijai	3a) Jei baigėsi, tikrinami ar duomenys teisingi ir jei ne tai vykdomas pirmasis punktas

Lentelė 15. Transporto srautų duomenų atnaujinimas

## 4.2. Sistemos statinis vaizdas

### 4.2.1. Apžvalga

Sistema suskaidyta į paketus, pavaizduotus paveikslėlyje žemiau.

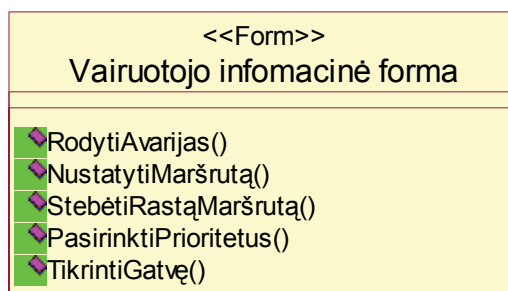


Paveikslas 9. Paketų diagrama

### 4.2.2. Paketų detalizavimas

#### 4.2.2.1. Paketas "Vairuotojo sąsaja"

Paketas "Vairuotojo sąsaja" apima labai paprastą vairuotojo vartotojo sąsają. Ji skirta maršrutų paieškai, avarijų peržiūrai, patikrinti ar įvesta jam reikiama gatvė į duomenų bazę.



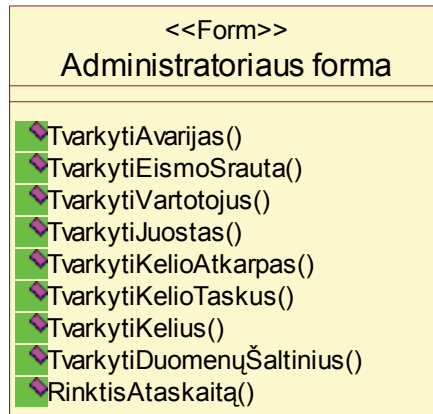
Paveikslas 10. Vairuotojo paketo detalizacija

Šios klasės sąryšį su „Kelionių planavimo paslauga“ paketu pamatysime jo detalizacijoje.



#### 4.2.2.2. Paketas “Administratoriaus sąsaja”

Paketas “Administratoriaus sąsaja“ apima sudėtingesnę ir žymiai atsakingesnę vartotojo sąsają. Ji skirta visų duomenų bazės duomenų tvarkymui (įterpimui, šalinimui, redagavimui), ataskaitų traukimui, vartotojų valdymui, avarijų stebėjimui, tvarkymui, duomenų šaltinių konfigūravimui.

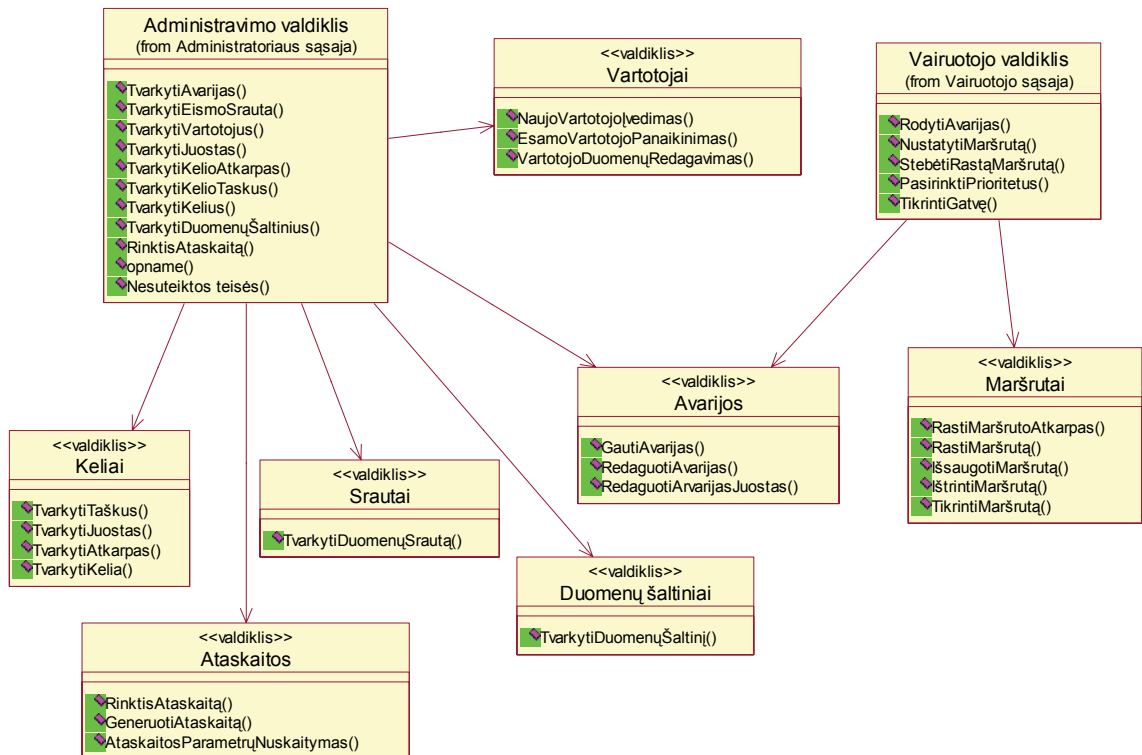


Paveikslas 11. „Vairuotojo sąsajos“ paketo detalizacija

Šios klasės sąryšį su „Kelionių planavimo paslaugų“ paketu pamatysime jo detalizacijoje.

### 4.2.2.3. Paketas „Kelionių planavimo paslaugos”

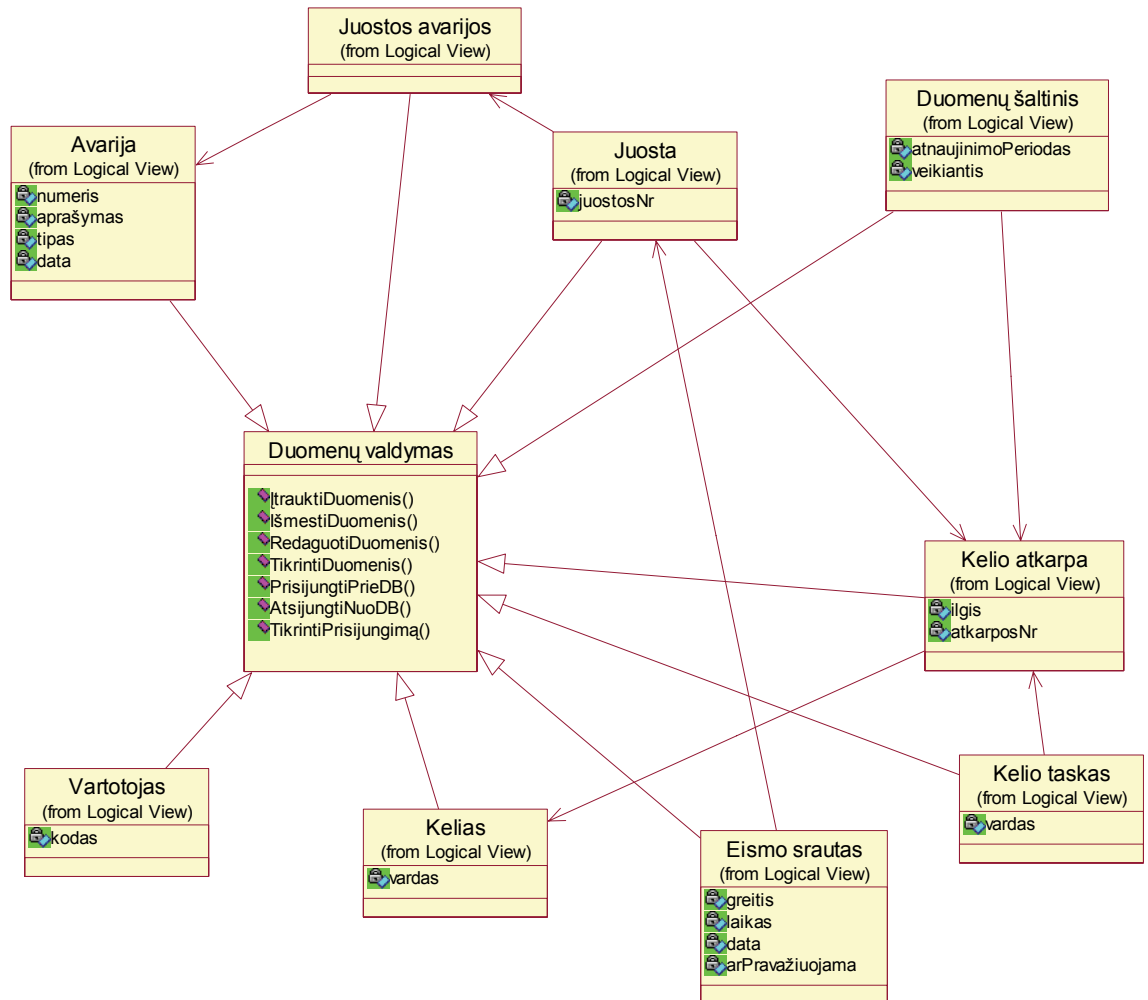
Šis paketas apima vairuotojo bei administratoriaus sąsajų funkcijų realizaciją. Tai yra tarpinis paketas tarp vartotojo sąsajos ir duomenų bazės.



Paveikslas 12. „Kelionių planavimo paslaugų“ paketo detalizacija

#### 4.2.2.4. Paketas „Duomenų paslaugos“

„Duomenų paslaugų“ paketas apima klases, kurios sąveikauja su duomenų baze. Paketas skirtas duomenų šrautų valdymui.



Paveikslas 13. „Duomenų paslaugų“ paketo detalizacija

#### 4.2.2.5. Paketas „Transporto stebėjimo sistemos sąsaja”

„Transporto stebėjimo sistemos sąsaja“ paketas yra pats paprasčiausias, tačiau vienas iš svarbiausių. Nuo šios paketo priklauso kitų paketų veikla.

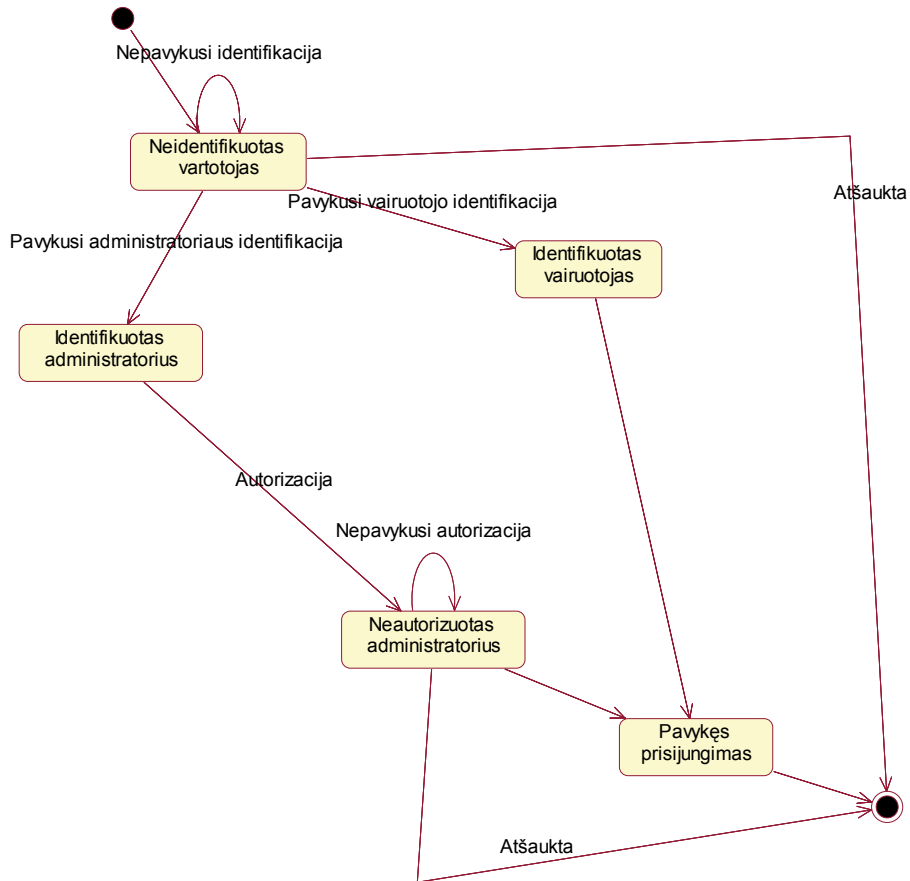
TransportoSrautoDuomenys
■ IšsaugotiDuomenis()
■ SiųstiSrautoDuomenis()
■ TrintiDuomenis()
■ PrisijungtiPrieDB()
■ AtsijungtiNuoDB()

Paveikslas 14. „Transporto stebėjimo sistemos sąsajos“ paketo detalizacija

### 4.3. Sistemos dinaminis vaizdas

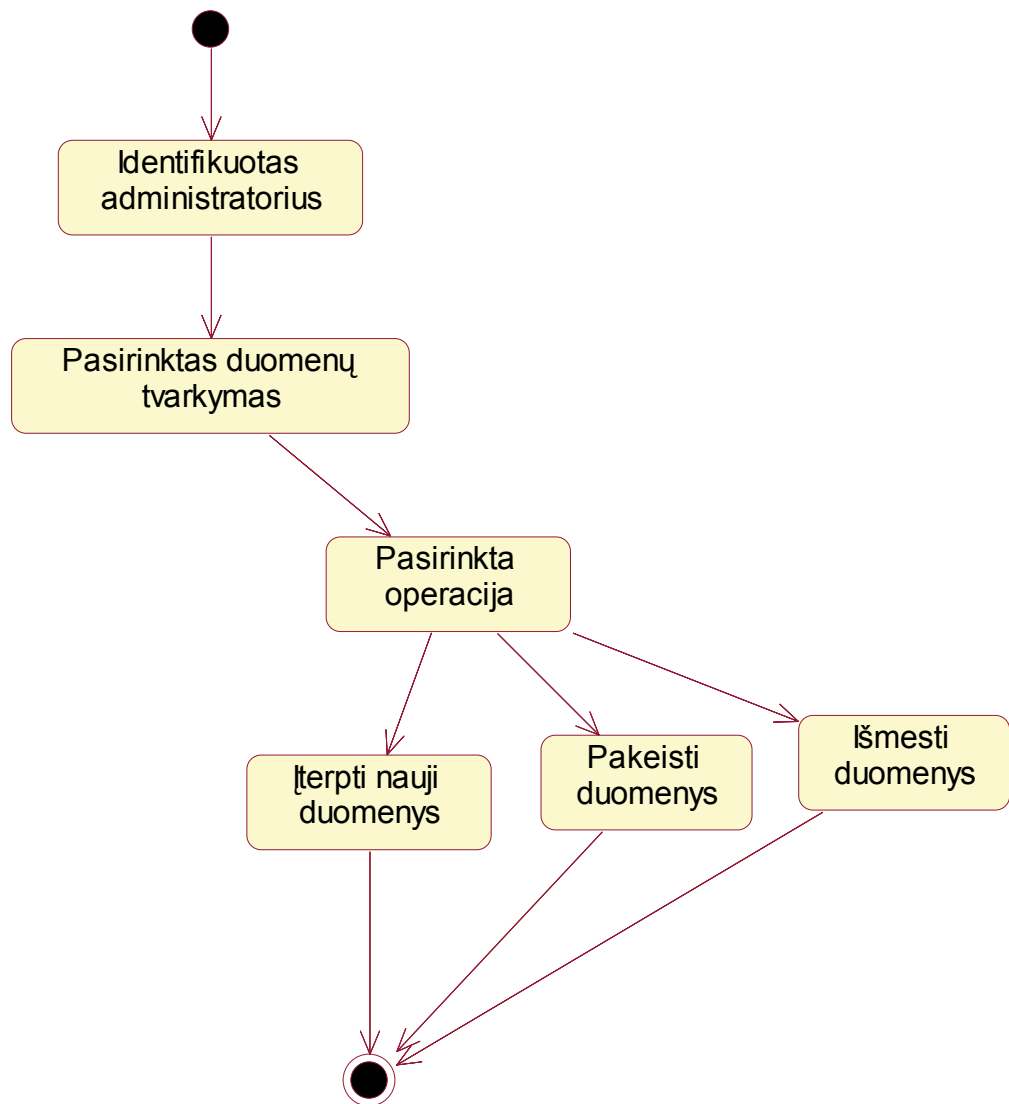
#### 4.3.1. Būsenų diagramos

##### 4.3.1.1. Prisijungimas



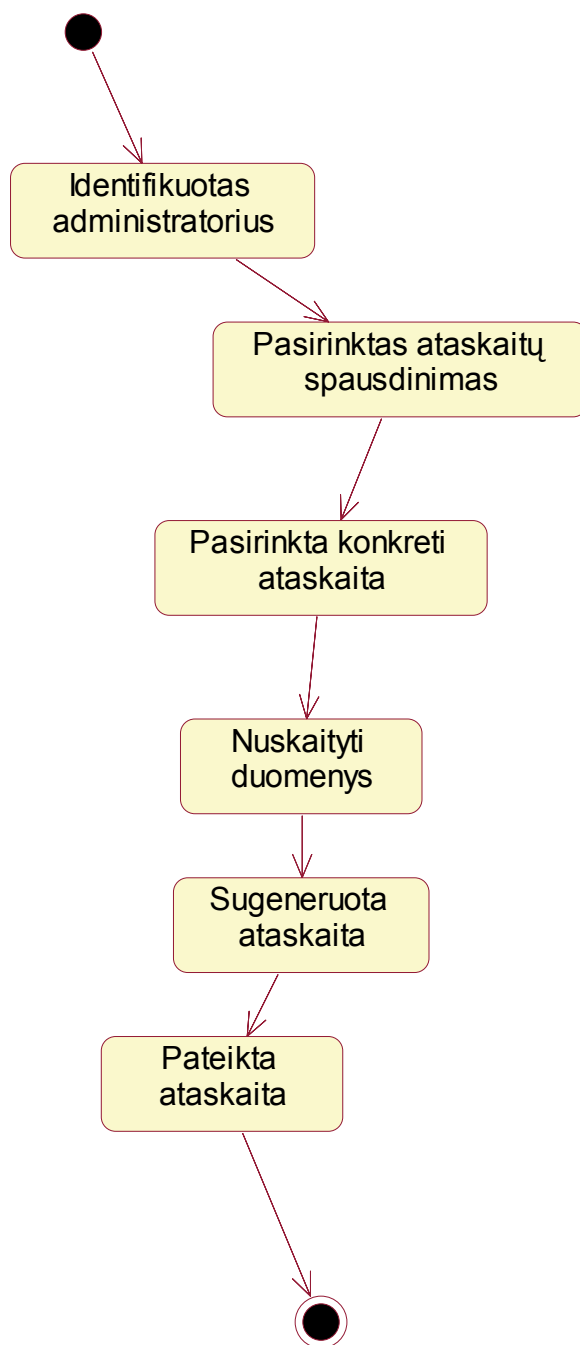
Paveikslas 15. Prisijungimo būsenų diagrama

#### 4.3.1.2. Parametrų nustatymas bei kitų duomenų tvarkymas (taip pat ir duomenų atnaujinimas)



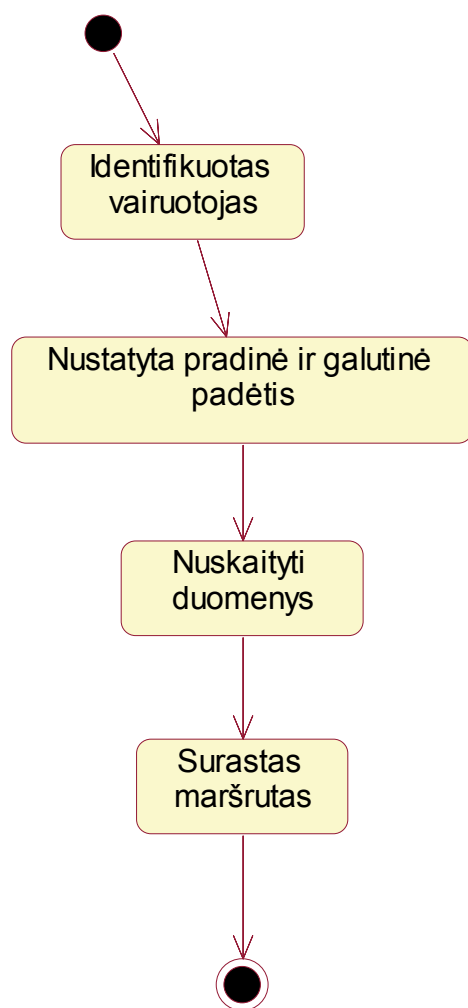
Paveikslas 16. Parametrų nustatymo bei kitų duomenų tvarkymo būsenų diagrama

### 4.3.1.3. Ataskaitų formavimas



Paveikslas 17. Ataskaitos formavimo būsenų diagrama

#### 4.3.1.4. Maršruto radimas

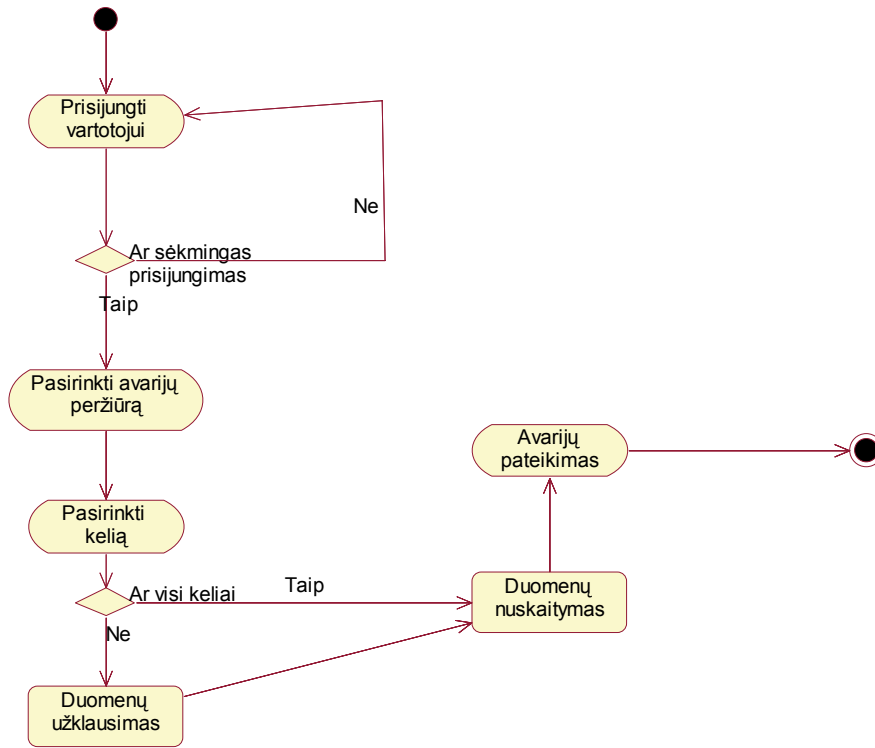


Paveikslas 18. Maršruto radimo būsenų diagrama



## 4.3.2. Veiklos diagramos

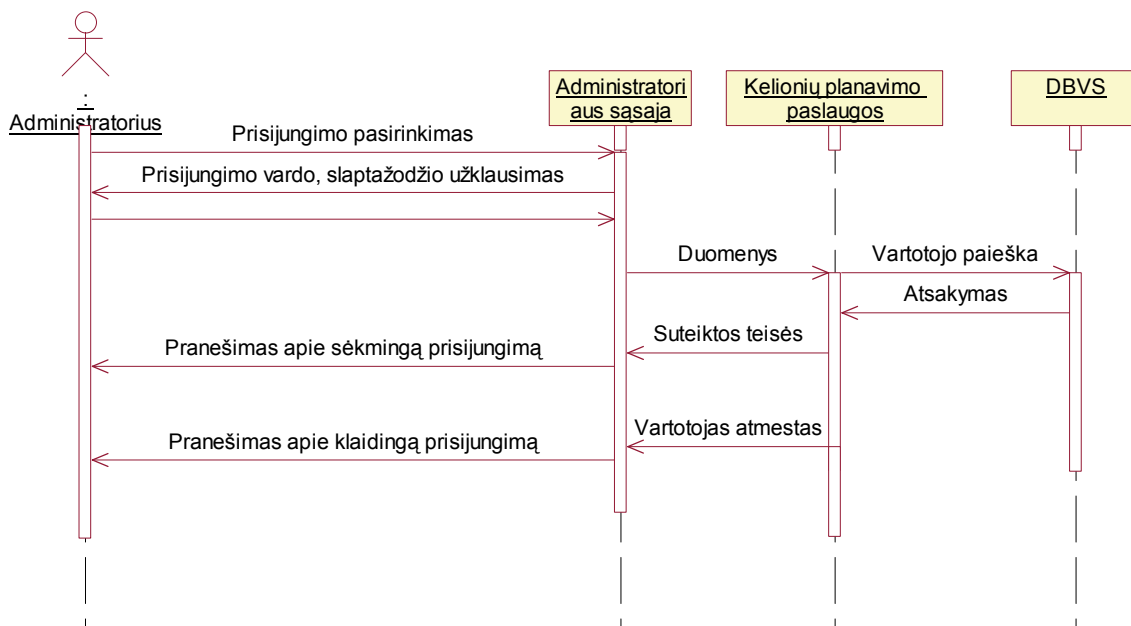
### 4.3.2.1. Avarijų peržiūra



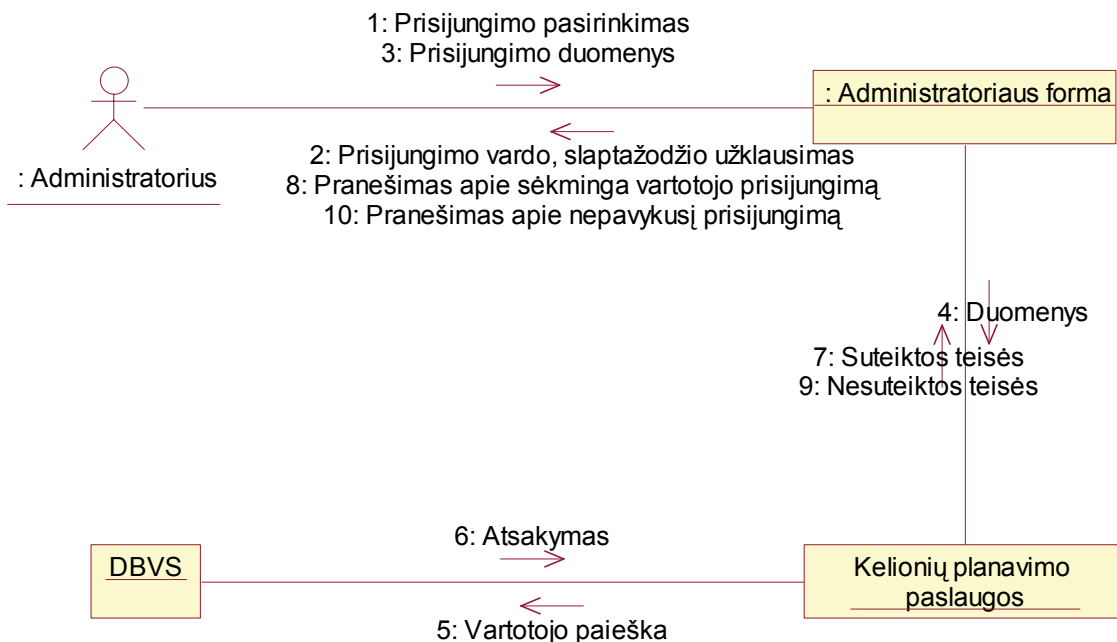
Paveikslas 19. Avarijų peržiūros veiklos diagrama

### 4.3.3. Sekų ir bendradarbiavimo diagramos

#### 4.3.3.1. Panaudojimo atvejis „Administratoriaus prisijungimas“

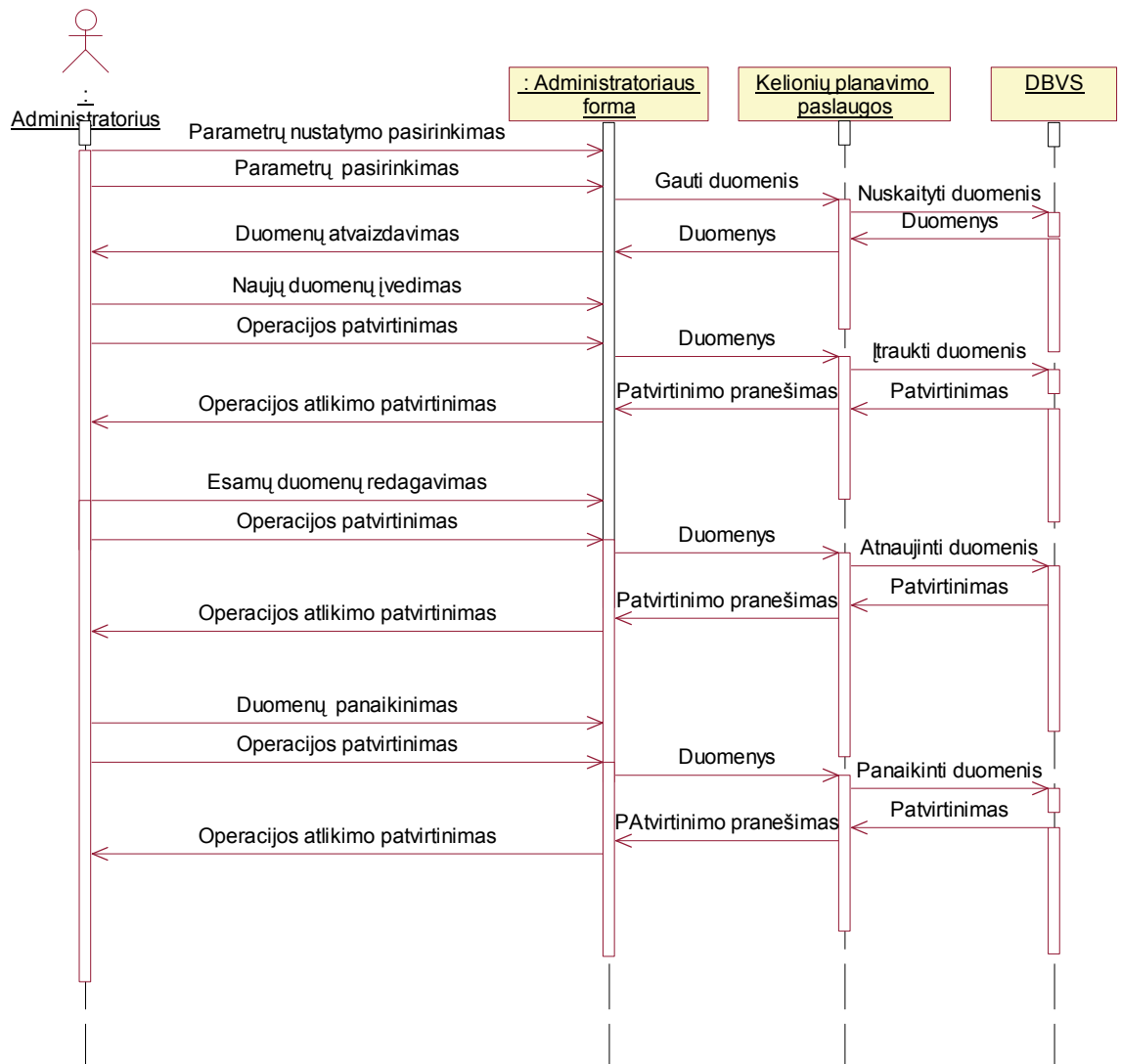


Paveikslas 20. PA „Administratoriaus prisijungimas“ sekų diagrama



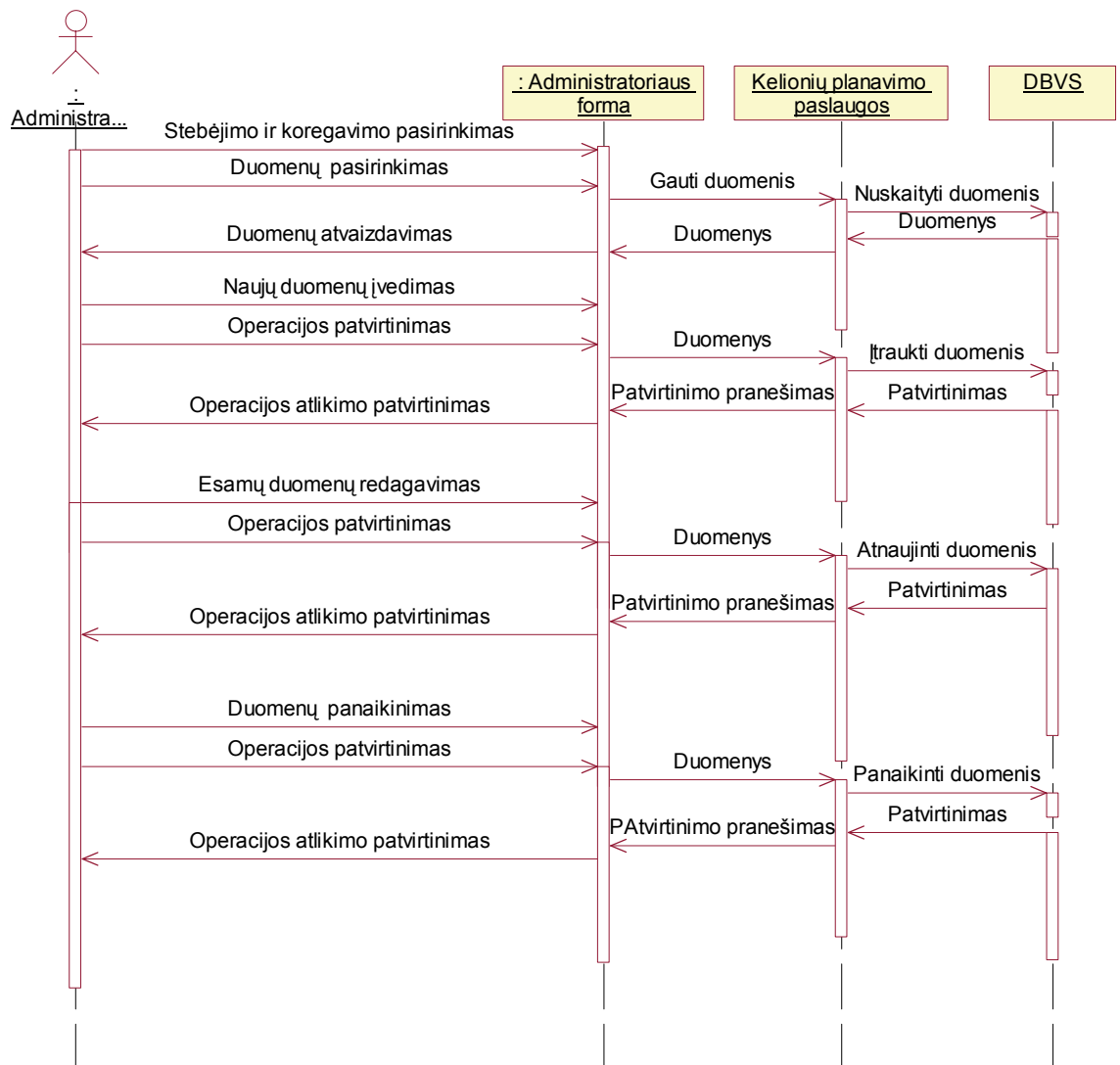
Paveikslas 21. PA „Administratoriaus prisijungimas“ bendradarbiavimo diagrama

### 4.3.3.2. Panaudojimo atvejis „Parametru nustatymas“



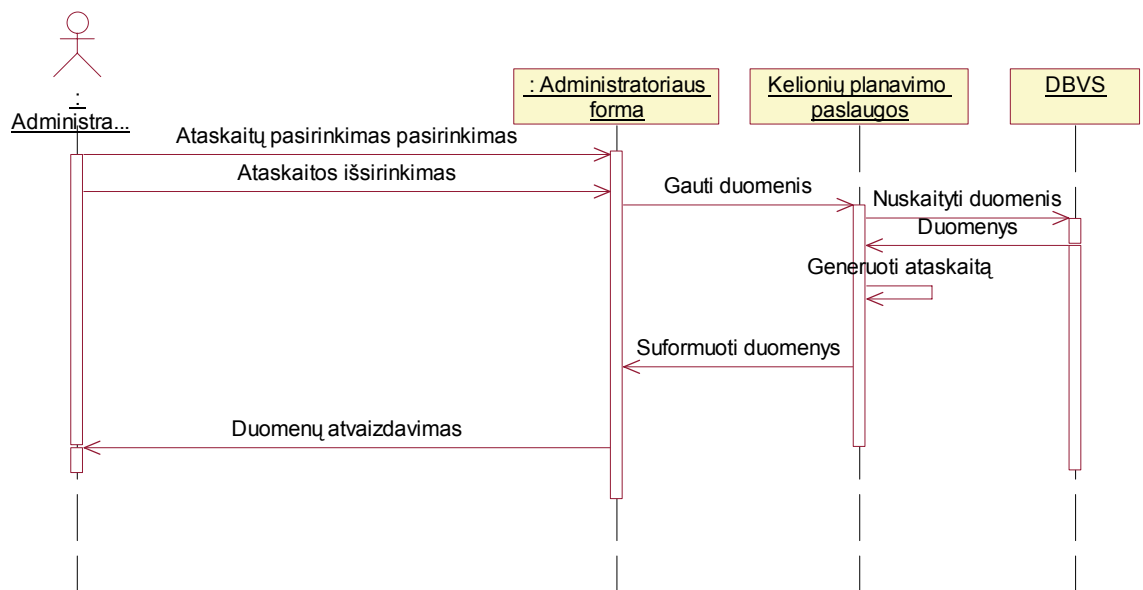
Paveikslas 22. PA „Parametru nustatymo“ sekų diagrama

### 4.3.3.3. Panaudojimo atvejis „Stebėjimas ir koregavimas“



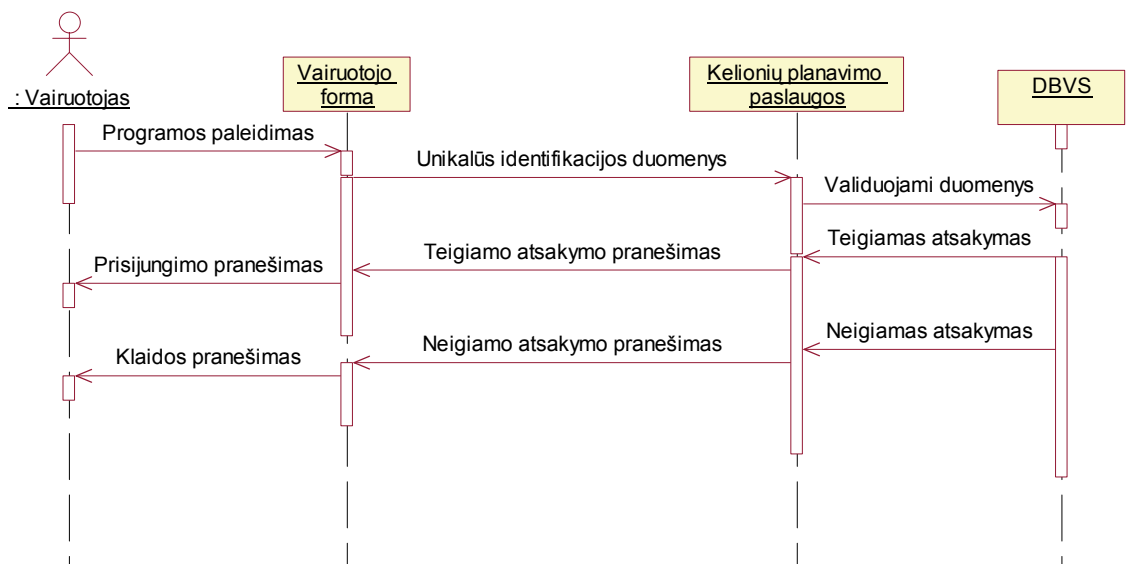
Paveikslas 23. PA „Stebėjimo ir koregavimo“ sekų diagrama

#### 4.3.3.4. Panaudojimo atvejis „Ataskaitų formavimas“



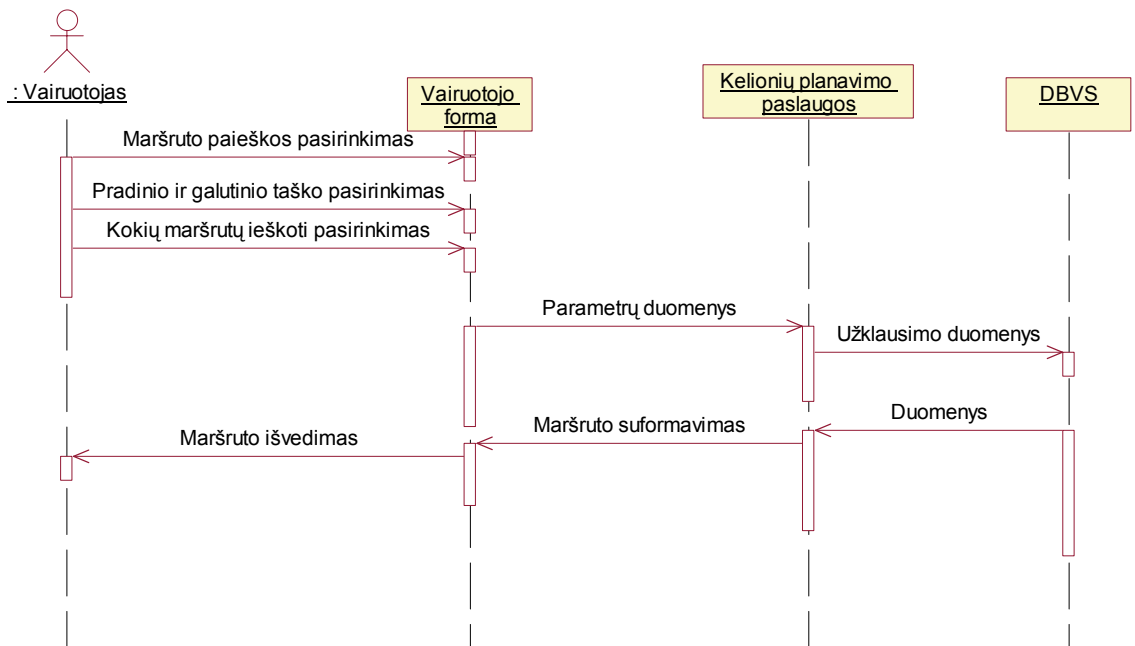
Paveikslas 24. PA „Ataskaitų formavimas“ sekų diagrama

#### 4.3.3.5. Panaudojimo atvejis „Vairuotojo prisijungimas“



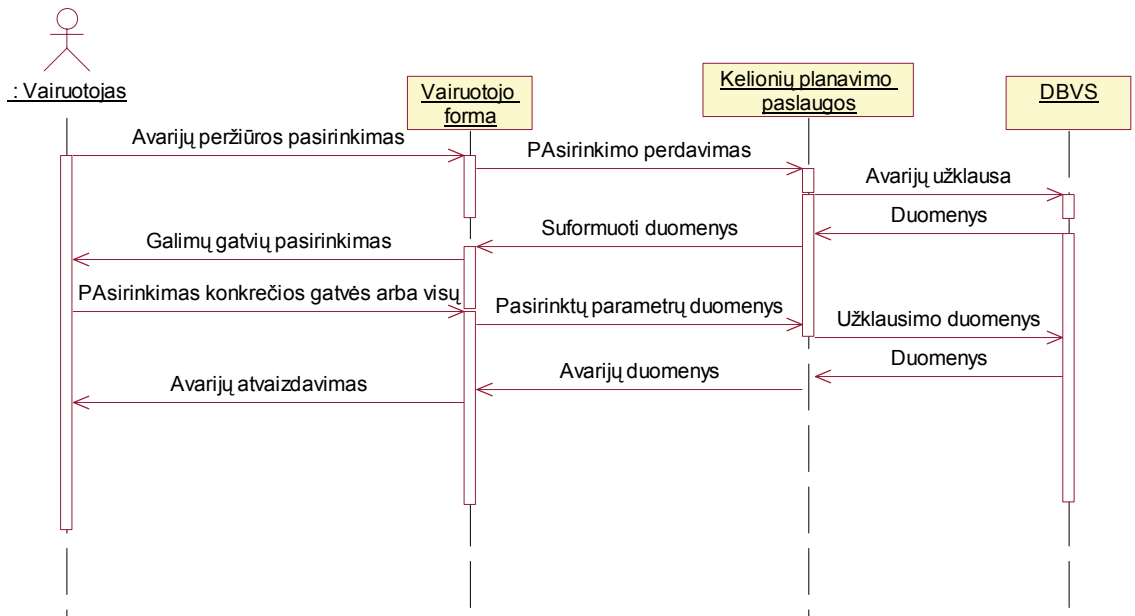
Paveikslas 25. PA „Vairuotojo prisijungimas“ sekų diagrama

#### 4.3.3.6. Panaudojimo atvejis „Maršruto radimas“



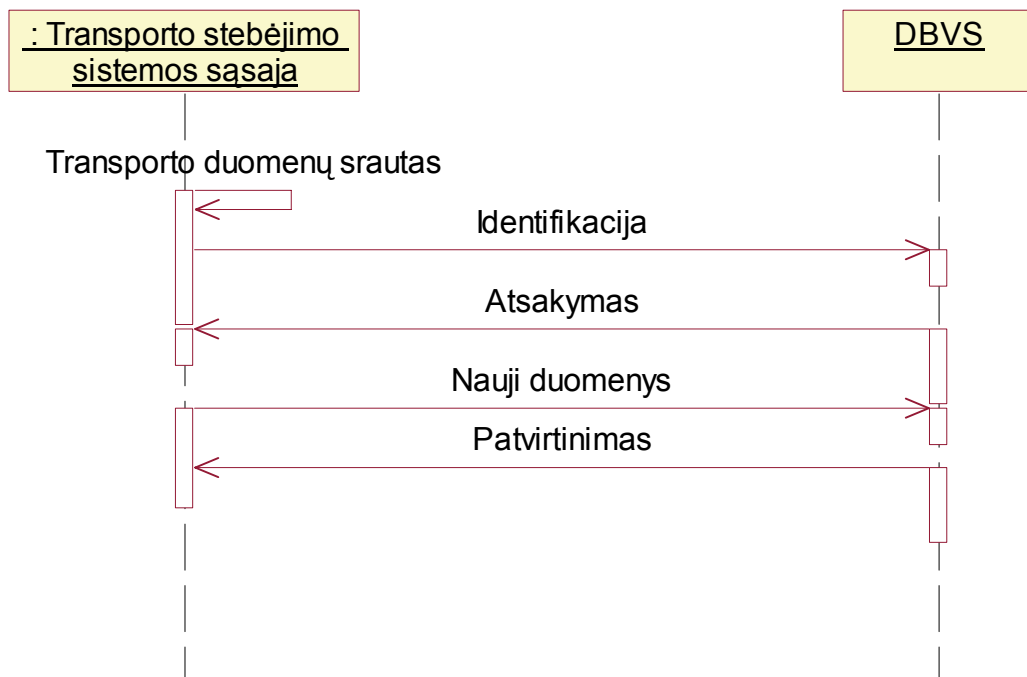
Paveikslas 26. PA „Maršruto radimo“ sekų diagrama

#### 4.3.3.7. Panaudojimo atvejis „Avarių peržiūra“



Paveikslas 27. PA „Avarių peržiūros“ sekų diagrama

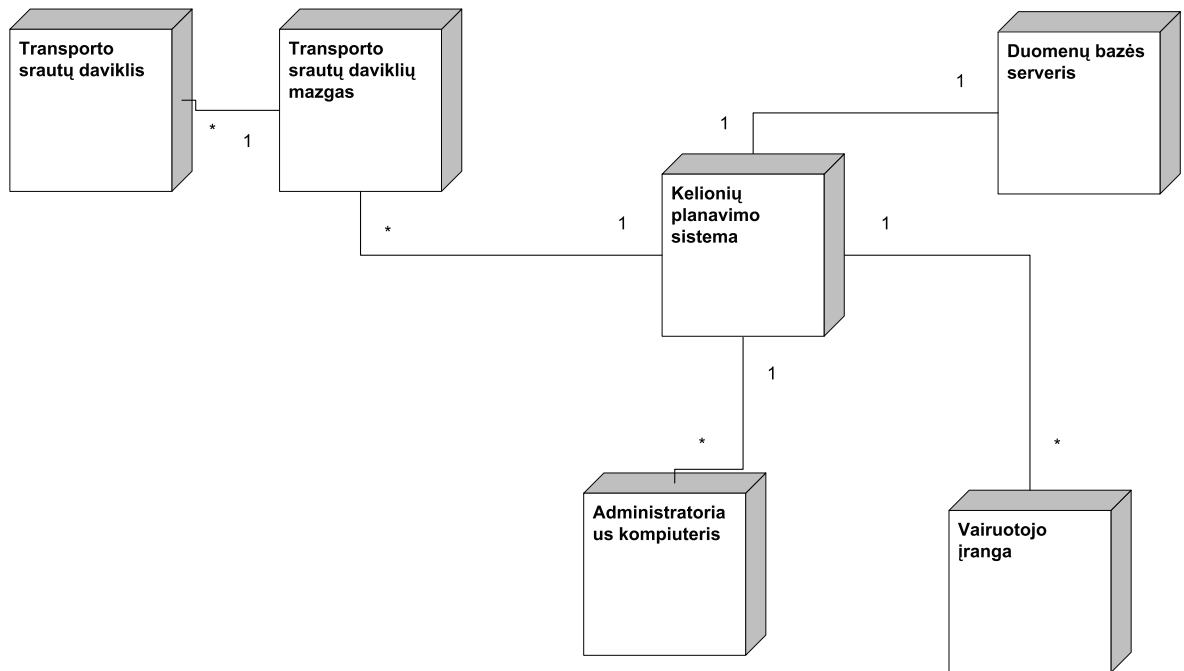
#### 4.3.3.8. Panaudojimo atvejis „Transporto srautų duomenų atnaujinimas“



Paveikslas 28. PA „Transporto srautų duomenų atnaujinimas“ sekų diagrama

#### 4.4. Išdėstymo vaizdas

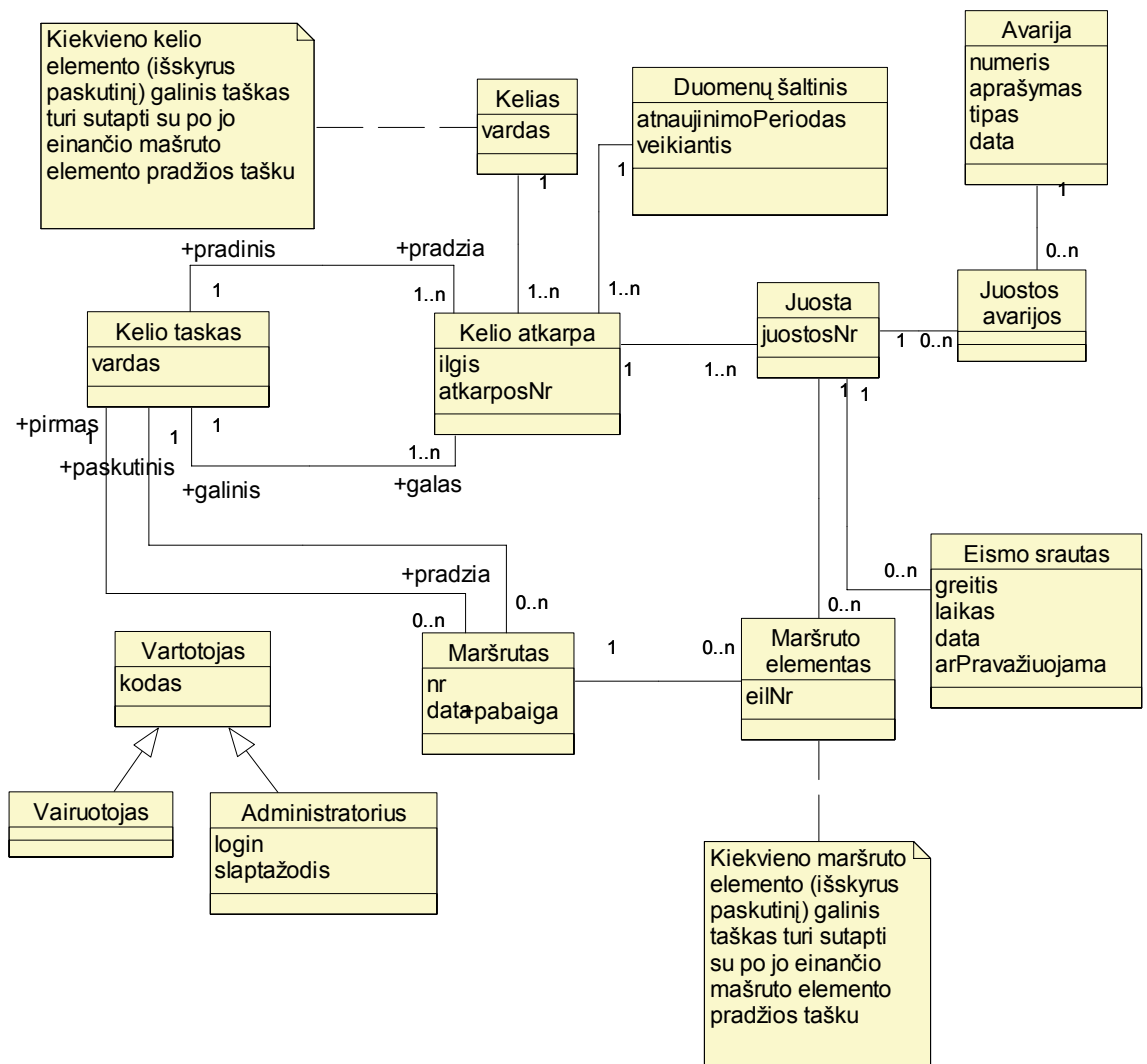
Žemiau pateikiama sistemos išdėstymo schema. Sistemą sudaro daugybė transporto srautų daviklių. Tai gali būti tiek kameros, tiek kitokie davikliai (priklauso nuo sistemos diegėjų). Jie yra sujungti į mazgus (dažniausiai kiekvienai sankryžai atskiras mazgas). Iš mazgo pastoviu intervalu keliauja duomenys į duomenų bazę. Iš jos duomenis naudoja įstaigos ar įmonės prižiūrinčios sistema (administratoriaus kompiuteris) bei sistemos naudotojams, t.y. vairuotojams.



Paveikslas 29. Sistemos prototipo išdėstymo vaizdas



## 4.5. Duomenų vaizdas



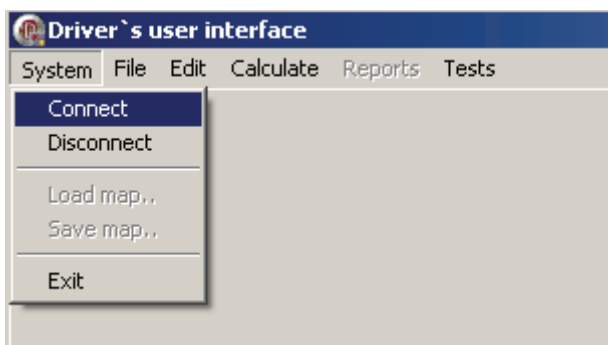
Paveikslas 30. Sistemos prototipo duomenų vaizdas

## 4.6. Sukurtos sistemos prototipo aprašymas

Šiame skyriuje bus aptariami tik esminiai langai bei funkcijos.

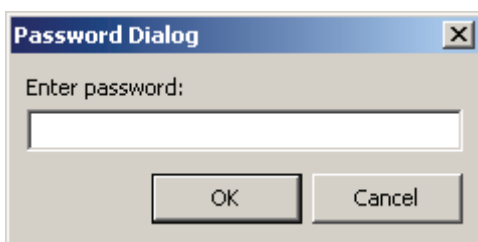
### 4.6.1. Prisijungimas prie sistemos

Administratorius, norintis dirbti su sistema, privalo prie jos prisijungti autentifikuodamasis. Autentifikacijos langas pasiekiamas per meniu System->Connect (pav. 31).



Paveikslas 31. Kaip pasiekti autentifikacijos langą

Autentifikacijai, administratorius turi būti įregistruotas vartotojų duomenų bazės lentelėje (tai padaroma instaliacijos ir sistemos konfigūracijos metu) ir jam turi būti žinomas slaptažodis (pav. 32).

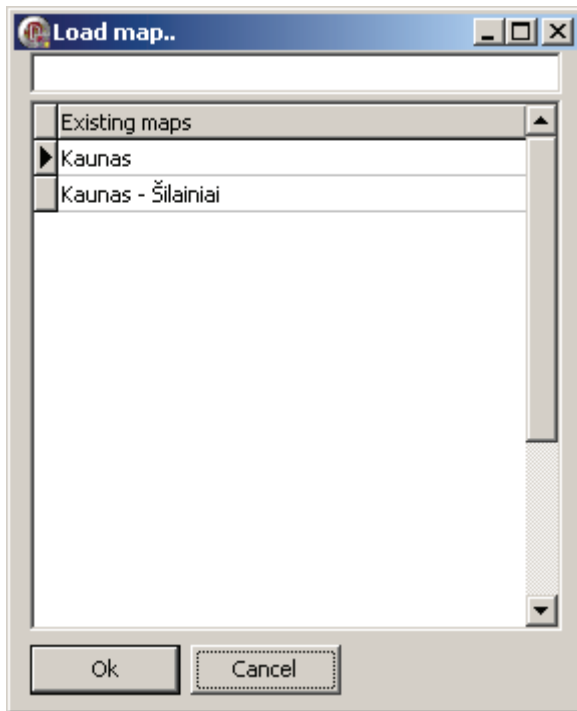


Paveikslas 32. Autentifikacijos langas

Prisijungus prie sistemos tampa aktyvuoti visi meniu punktai. Norint atsijungti nuosistemos reikia pasirinkti Disconnect meniu punktą, matomą pav. 31.

## 4.6.2. Žemėlapių užkrovimas, išsaugojimas

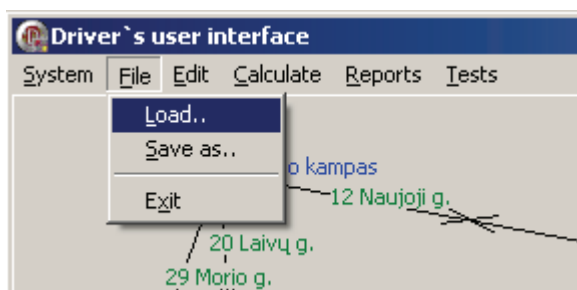
Žemėlapis užkraunamas pasirinkus „Load map..“, o išsaugojamas – „Save map..“ meniu punktą (31 pav.). Pasirinkus vieną iš šių punktų matomas žemėlapių sąrašas sekančiame lange (pav. 33).



Paveikslas 33. Žemėlapio pasirinkimas

Pasirinkus žemėlapiį spaudžiamas OK mygtukas. Saugant žemėlapiį būkite atsargūs, sutapus pavadinimams arba pasirinkus iš sąrašo, duomenys yra perrašomi negražinamai.

Žemėlapiai taip pat gali būti išsaugoti arba užkrauti per specialaus objekcinio formato failus (pav 34).



Paveikslas 34. Žemėlapių užkrovimas bei saugojimas į failą

Ši galimybė yra padaryta norint patikrinti duomenų bazės veikimą arba perduoti žemėlapius iš vienos sistemos į kitą, kurių duomenų bazės yra skirtingos. Tačiau užkrovus

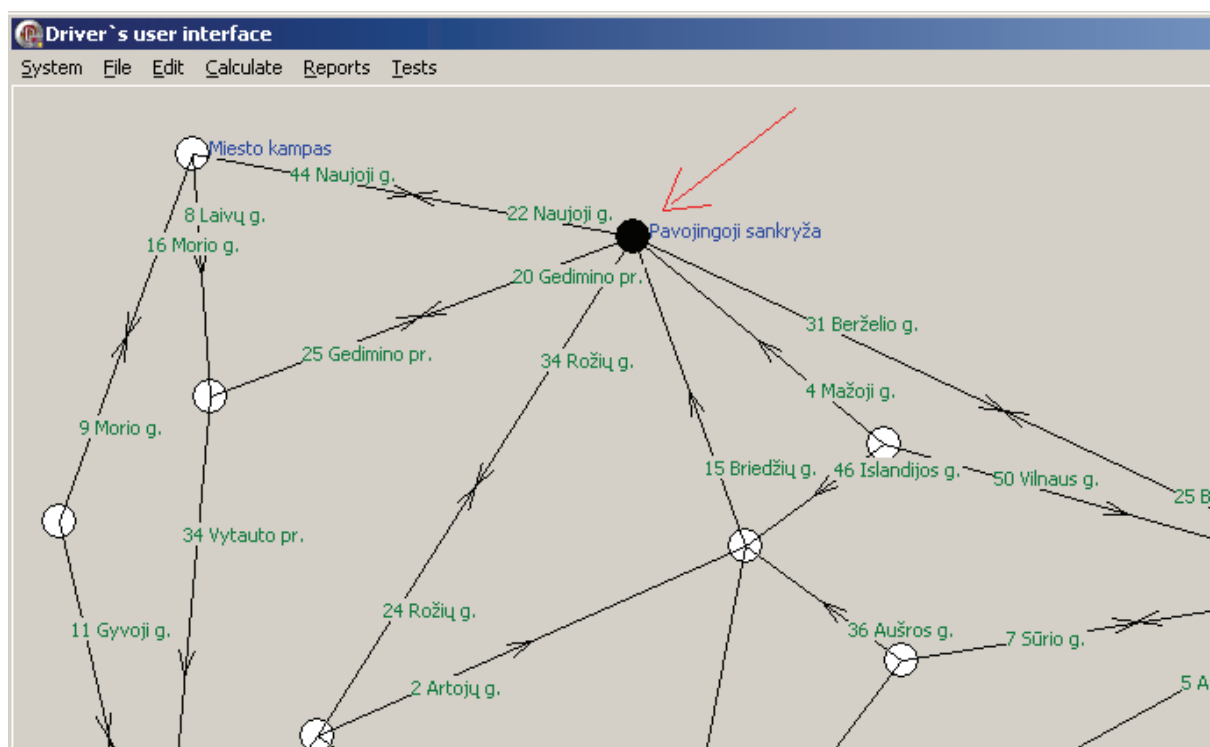


Dešinėje pusėje apačioje matyti pagalbinė panelė. Joje atvaizduotos kelių aiškinamosios spalvos.

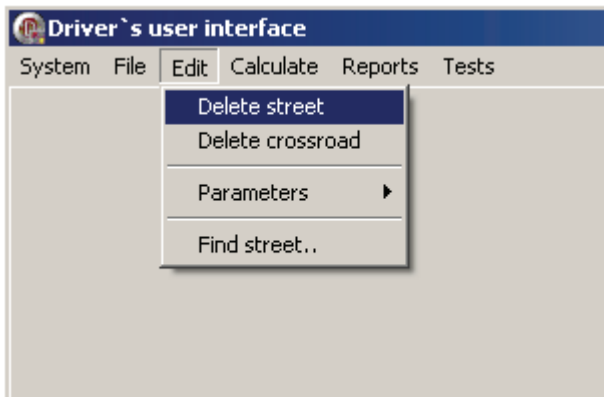
#### 4.6.4. Žemėlapių redagavimas

Sankryža pasirenkama kairiuoju pelės klavišu. Pažymėta sankryža atvaizduojama juodu apskritimu (pav. 36). Pažymėtos sankryžos duomenys atvaizduojami dešiniojoje viršutinėje panelėje. Ten galima juos redaguoti. Redaguojant informacija iškart atvaizduojama pagrindinėje panelėje, tačiau nereikia pamiršti, jog ši informacija išsaugota bus tik pasirinkus „Save map..“ meniu punktą.

Svarbu tai, kad norint ištrinti sankryžą, reikia prieš tai ją pažymėti. Ištrinti sankryžą (pav. 37) galima tik tuo atveju, jeigu ji neturi jokių kelių, o tai reiškia, jog prieš tai reikia ištrinti visus kelius, kurie ateina ar išeina iš šios sankryžos.

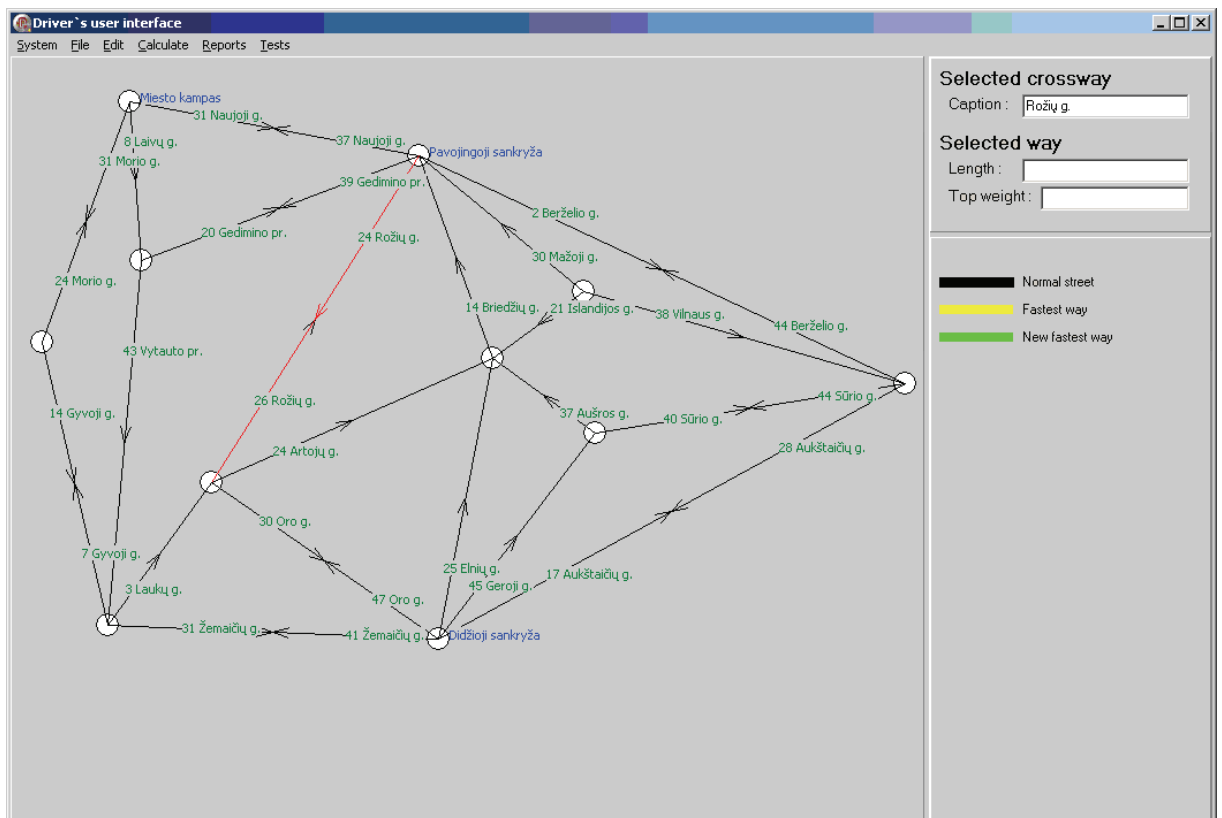


Paveikslas 36. Pasirinkta sankryža



Paveikslas 37. Žemėlapiu objektų redagavimas

Norint pasirinkti kelią, pirmiausia reikia pasirinkti sankryžą, kurioje jis prasideda. Pasirinktas kelias piešiamas raudona linija (pav. 38). “Delete street” meniu punkta gatvės ištrynimui galima rasti septintame paveiksle.

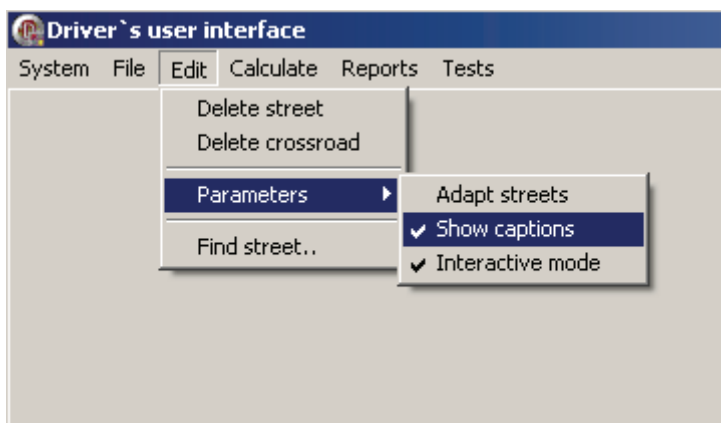


Paveikslas 38. Pažymėtas kelias.

#### 4.6.5. Peržiūros režimai

Administratorius gali parinkti keletą peržiūros režimų (pav. 39). Pagal nutylėjimą yra įjungti pavadinimų bei interaktyvus peržiūros režimai.

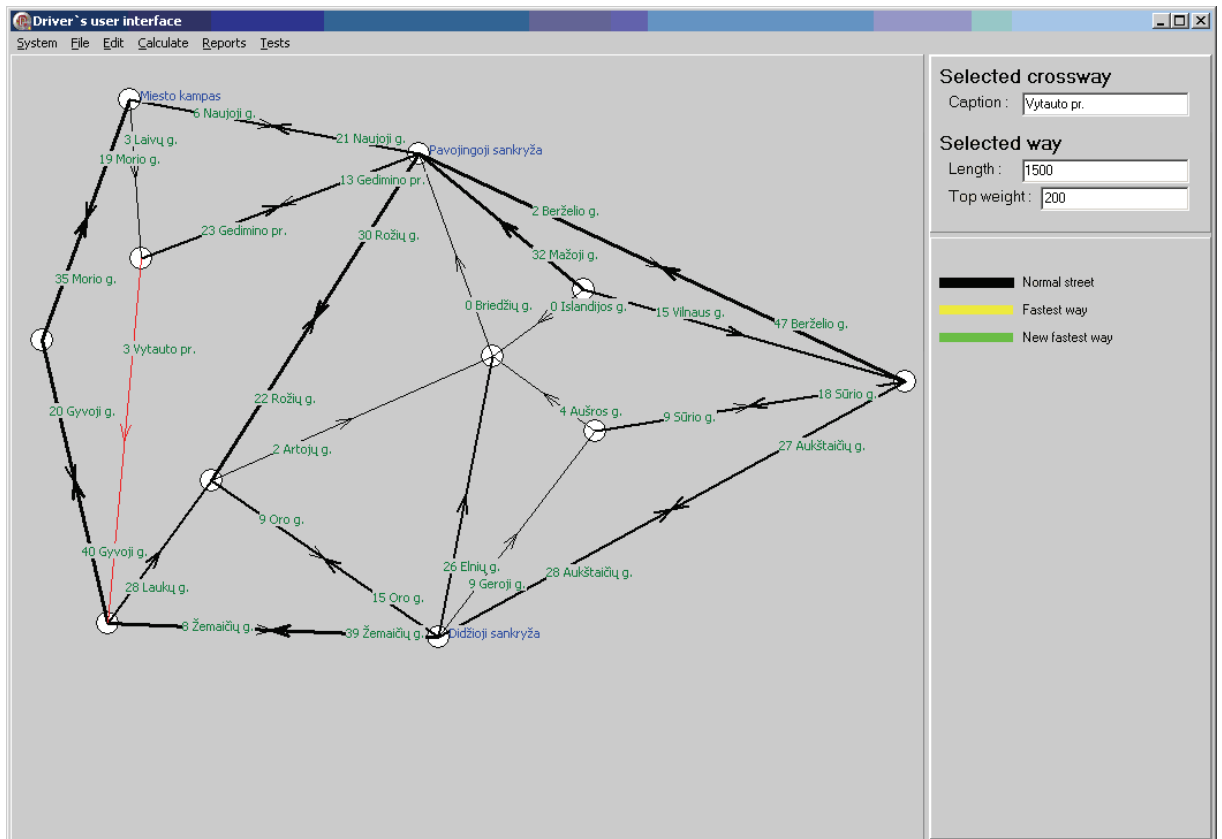
Peržiūros režimu „Adapt streets“ gatvių plotis piešiamas atsižvelgiant į jos apkrautumą (pav. 40). Šis režimas automatiškai įjungiamas vairuotojo programoje. Tai labai patogu, kai vairuotojas žino kelius ir vienu žvilgtelėjimu jis gali pasirinkti maršrutą ir be kompiuterio paieškos pagalbos.



Paveikslas 39. Peržiūros režimai

„Show captions“ – vartotojas mato gatvių bei sankryžų pavadinimus bei apkrautumą. Jį išjungti patogu kuriant, transformuojant žemėlapi.

„Interactive mode“ – realaus laiko režimas. Realio laiku atnaujinami duomenys apie autotransporto kelių apkrovimą. Tvarkant žemėlapi, rekomenduojama šį režimą išjungti.

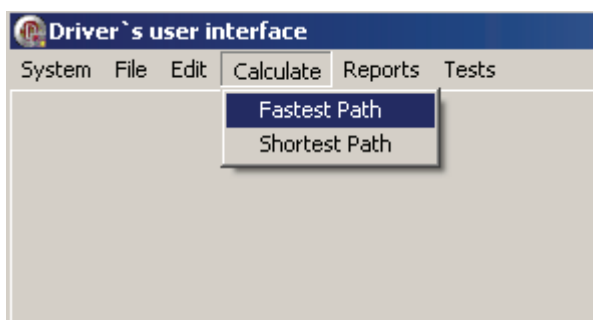


Paveikslas 40. Peržiūros režimas „Adapt streets“

#### 4.6.6. Greičiausio kelio paieška

Ši funkcija yra viena iš dažniausiai naudojamų vairuotojų. Norint ja pasinaudoti, reikia atlikti sekančius veiksmus:

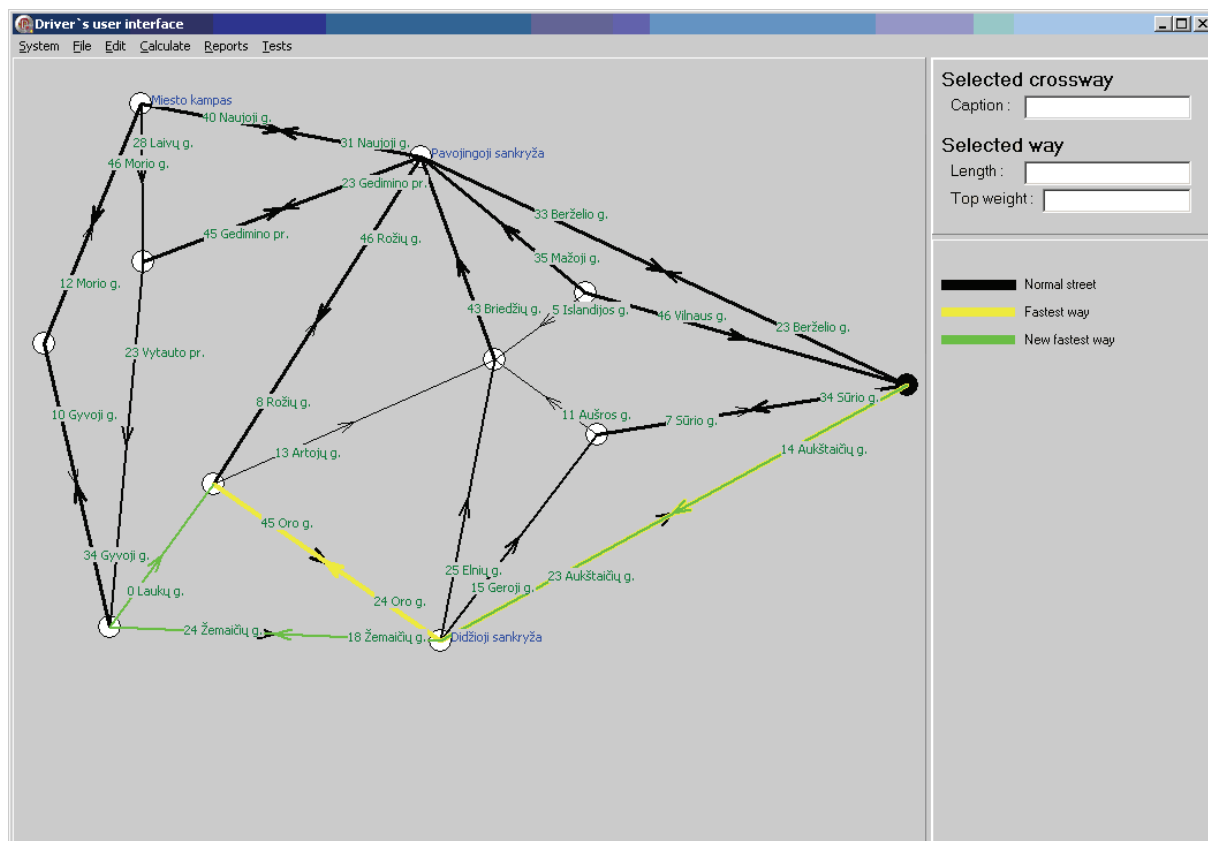
- pažymėti sankryžą iš kurios bus keliaujama
- pasirinkti „Fastest Path“ meniu punktą (pav. 41)
- pasirinkti galutinį tikslą-sankryžą, į kurią bus vykstama



Paveikslas 41. Greičiausio kelio meniu pasirinkimas



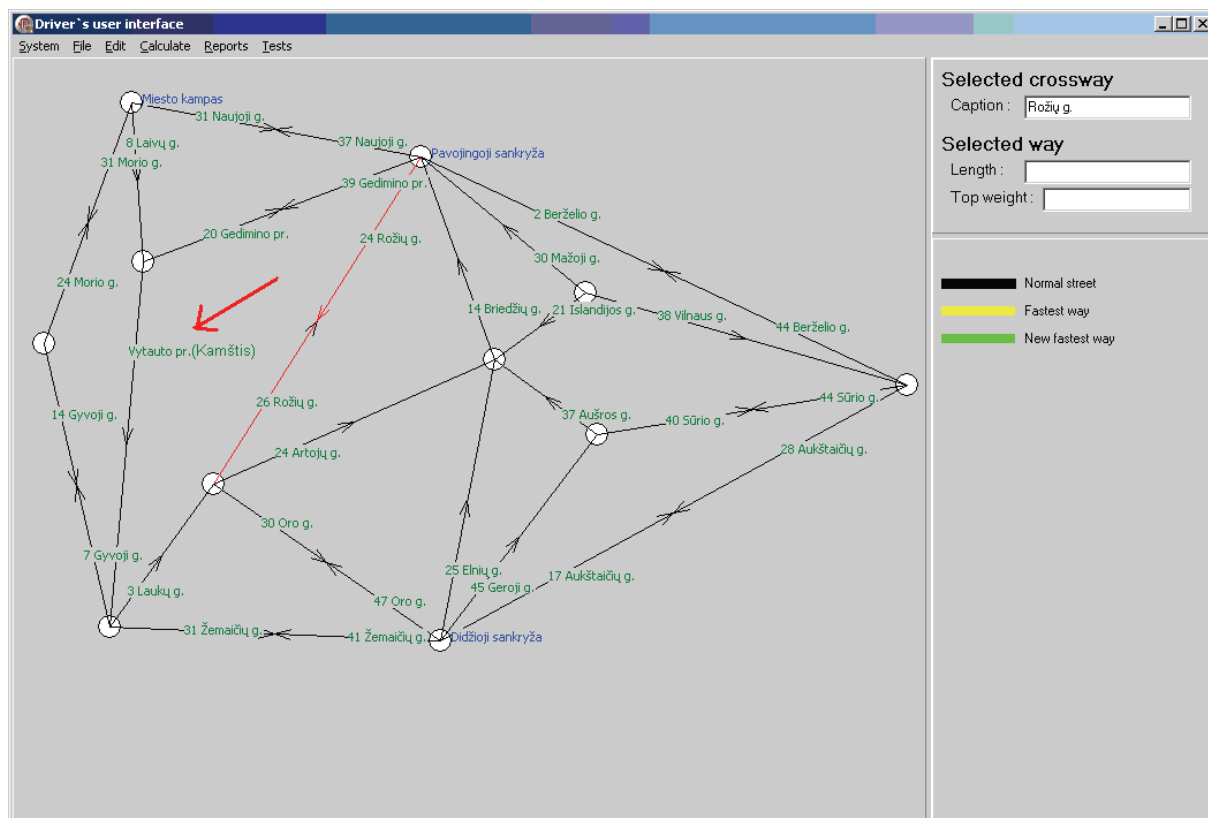
Rezultatą galima pamatyti 42-ame paveikslėlyje. Surastas greičiausias kelias yra atvaizduojamas geltona spalva. Jei įjungtas realaus laiko režimas, atsinaujinus duomenims perskaičiuojamas greičiausias kelias. Šis kelias piešiamas šviesiai žalia spalva. Persidengus pirmam rastam maršrutui ir –pačiam naujausiam, piešiamas šviesiai žalios spalvos kelias su apibrėžta geltona linija.



Paveikslas 42. Trumpiausio kelio atvaizdavimas

Analogiškai vyksta trumpiausio keli paieška, todėl ji aprašinėjama nebus.

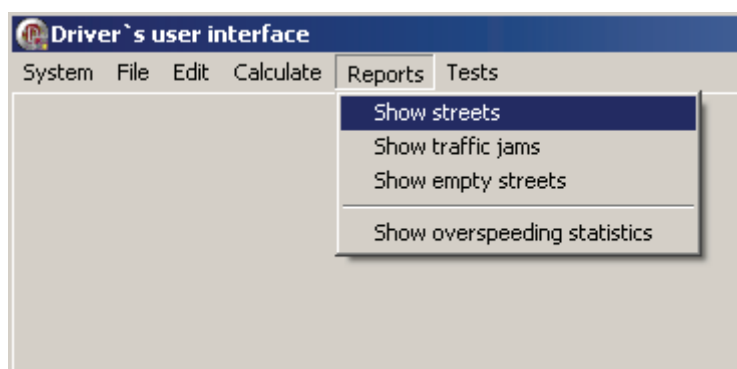
## Kamščio pavyzdys:



Paveikslas 43. Kamštis

### 4.6.7. Ataskaitos

Ataskaitos reikalingos sistemos administratoriui, bendradarbiaujant su kitmos institucijomis. Ataskaitos vėliau gali būti papildytos pagal poreikį. Kadangi ataskaitos nėra ypatingai susijusios su darbo uždaviniu, apžvelgsime tik pirmąją ataskaitą. Pasirinkus šia ataskaitą parodomi visi keliai, bei tuo momentu mašinų skaičius kelyje. Ataskaita naudinga norint sutikrinti ar visi keliai yra įtraukti į žemėlapi ir pan. (pav. 45).



Paveikslas 44. Ataskaitos

Ataskaitos aplinka aprašoma nebus.

Print Preview

Thumbnails Search result

### Street names report

Street	car amount at the moment
Briedžių g. ....	0
Geroji g. ....	0
Vilnaus g. ....	5
Vytauto pr. ....	5
Artojų g. ....	13
Aušros g. ....	20
Islandijos g. ....	20
Elnių g. ....	22
Aukštaičių g. ....	24
Oro g. ....	27
Mažoji g. ....	28
Gyvoji g. ....	30
Naujoji g. ....	41
Sūrio g. ....	41
Laukų g. ....	44
Laių g. ....	47
Gedimino pr. ....	49
Žemaičių a. ....	62

0% Page 1 of 1

Paveikslas 45. Gatvių pavadinimų ataskaita



## 5. Išvados

Kylant Lietuvos ekonomikai, daugėjant mašinų, intensyvėjant eismui reikia kuo greičiau imtis atitinkamų išvadų ir kurti Nacionalinę ITS infrastruktūrą.

Sprendžiant transporto kamščių problemą buvo atlikta egzistuojančių sprendimų (KAREN/FRAME, TelemArk, Actif, ARTIST) analizė.

Transporto kamščius galima sumažinti, optimizuojant atskirų transporto priemonių maršrutus. Kuo daugiau vartotojų naudosis maršruto optimizavimo sistema, tuo geriau bus išbalansuotas transporto apkrovimas.

Atskirų vairuotojų maršrutų optimizavimui buvo nagrinėti 8 algoritmai. Analizė parodė, kad tikslinga detaliau tirti 2 tikslius ir 2 euristinius algoritmus.

Atliktas Bellman-Fordo ir Dijkstra algoritmų tyrimas parodė, kad Dijkstra algoritmas duoda geresnius rezultatus.

Atlikta šių algoritmų skaičiavimo laiko priklausomybė nuo transporto sistemos dydžio parodė, kad didelėms sistemoms tikslinga naudoti euristinius algoritmus (A\* bei įvairias skruzdėlių algoritmo modifikacijas)

Suprojektuotas ir realizuotas transporto maršrutų optimizavimo sistemos prototipas

Prototipą ir algoritmų tyrimo rezultatus numatyta panaudoti planuojamoje Vilniaus transporto valdymo sistemoje

Sistemos prototipo duomenys yra saugojami kaip galima labiau detalizuotai ir diferencijuotai, todėl neprisirišama prie jokios konkrečios sistemos ir bet kuri sistema yra lengvai pritaikoma prie esamos architektūros valdiklių pagalba.

Speciali vairuotojų įranga (kartu su kelionių planavimo sistema) yra didelis privalumas vartotojui, nes nereikia diegti papildomos programinės įrangos, nereikia kompiuterinio raštingumo, vartotojo sąsaja yra paprasta, intuityvi ir suprantama.

## LITERATŪRA

- [1] Nigro, A.; De Palma, M.; Manfredi, S. *ARTIST – Italian experience in the deployment of the European ITS Framework architecture*. [Interaktyvus] [Žiūrėta 2004-11-04] Prieiga per internetą: <<http://www.frame-online.net/Articles/ArtistExp.PDF>>
- [2] *The Boost Graph Library*. [Interaktyvus] [Žiūrėta 2006-05-15] Prieiga per internetą: <[http://www.boost.org/libs/graph/doc/table\\_of\\_contents.html](http://www.boost.org/libs/graph/doc/table_of_contents.html)>
- [3] Department for Transport. *ITS Assist Home Page. ITS Information*. [Interaktyvus] [Žiūrėta 2005-01-02] Prieiga per internetą: <<http://www.its-assist.org.uk/information.htm>>
- [4] *ERTICO – ITS Europe is a multi-sector, public/private partnership pursuing the development and deployment of Intelligent Transport Systems and Services (ITS)*. [Interaktyvus] [Žiūrėta 2004-12-10] Prieiga per internetą: <[http://www.ertico.com/en/what\\_is\\_its/its\\_basics/](http://www.ertico.com/en/what_is_its/its_basics/)>
- [5] *European ITS Framework Architecture*, [Interaktyvus] [Žiūrėta 2004-11-07] Prieiga per internetą: <<http://www.frame-online.net/NationalArchs.htm>>
- [6] *European ITS Framework Architecture*. [Interaktyvus] [Žiūrėta 2004-11-11] Prieiga per internetą: <<http://www.frame-online.net/Brochures/Madrid%20folder/Insert%20Nov03%20TelemArk%20Finland.pdf>>
- [7] *European ITS Framework Architecture*. [Interaktyvus] [Žiūrėta 2004-11-21] Prieiga per internetą: <<http://www.frame-online.net/Brochures/Madrid%20folder/Insert%20Nov03%20ACTIF%20France.pdf>>
- [8] *European ITS Framework Architecture: INTELLIGENT TRANSPORT SYSTEM What they are and why they are important for transport today*. [Interaktyvus] [Žiūrėta 2004-11-02] Prieiga per internetą: <<http://www.frame-online.net/Brochures/ITS%20Planning%20Guide%20Issue%202%20.pdf>>

- [9] *European ITS Framework Architecture: INTELLIGENT TRANSPORT SYSTEM What they are and why they are important for transport today.* [Interaktyvus] [Žiūrėta 2004-11-02] Prieiga per internetą: <<http://www.frame-online.net/Brochures/About%20ITS.PDF>>
- [10] *European ITS Framework Architecture: What is ITS.* [Interaktyvus] [Žiūrėta 2004-11-02] Prieiga per internetą: <<http://www.frame-online.net/Brochures/What-is-ITS.PDF>>
- [11] Booch, G.; Rumbaugh, J.; Jacobson, I. *The Unified Modeling Language User Guide.*- Addison Wesley, 2000. 512p.
- [12] Jacobson, I.,; Booch, G.; Rumbaugh, J. *The unified software development process.* Addison-Wesley, 1999.
- [13] National Institute of Standards and Technology. *Dictionary of Algorithms and Data Structures.* [Interaktyvus] [Žiūrėta 2006-05-15] Prieiga per internetą: <<http://www.nist.gov/dads>>
- [14] Jesty, P. H.; Bossom, R. A. P. *Why do you need ITS architecture, European and national perspectives.* [Interaktyvus] [Žiūrėta 2004-11-03] Prieiga per internetą: <<http://www.frame-online.net/Articles/TS20%20Jesty%20Why%20you%20need%20%20architecture.pdf>>
- [15] Jesty, P. H.; Bossom, R. A. P. *How can ITS architecture be created? A European view.* [Interaktyvus] [Žiūrėta 2004-11-04] Prieiga per internetą: <<http://www.frame-online.net/Articles/TS23%20Bossom.pdf>>
- [16] *SCOOT - An Urban Traffic Control System developed by the Transport Research Laboratory and provided by Siemens Traffic Control Ltd and Peek Traffic Ltd.* [Interaktyvus] [Žiūrėta 2004-12-19] Prieiga per internetą: <<http://www.scoot-utc.com/whatis.htm>>
- [17] SIEMENS: *SITRAFFIC® C800V Precision and reliability at its best.* [Interaktyvus] [Žiūrėta 2005-01-12] Prieiga per internetą: <[http://www.industry.siemens.de/traffic/pdf/inhalt/support/downloads/stadtverkehr/detail\\_stadtverkehr\\_sitraffic\\_c800v\\_en.pdf](http://www.industry.siemens.de/traffic/pdf/inhalt/support/downloads/stadtverkehr/detail_stadtverkehr_sitraffic_c800v_en.pdf)>

- [18] SIEMENS: *SITRAFFIC C840V Traffic Controler*. [Interaktyvus] [Žiūrėta 2005-01-16] Prieiga per internetą: <[http://www.industry.siemens.de/traffic/pdf/inhalt/support/downloads/stadtverkehr/SITRAFFIC\\_C840V\\_eng.pdf](http://www.industry.siemens.de/traffic/pdf/inhalt/support/downloads/stadtverkehr/SITRAFFIC_C840V_eng.pdf)>
- [19] SIEMENS: *SITRAFFIC MOTION MX*. [Interaktyvus] [Žiūrėta 2005-01-18] Prieiga per internetą: <[http://www.industry.siemens.de/traffic/pdf/inhalt/support/downloads/stadtverkehr/detail\\_stadtverkehr\\_motion\\_en.pdf](http://www.industry.siemens.de/traffic/pdf/inhalt/support/downloads/stadtverkehr/detail_stadtverkehr_motion_en.pdf)>
- [20] SIEMENS: *Solutions for Urban Traffic*. [Interaktyvus] [Žiūrėta 2005-01-07] Prieiga per internetą: <[http://www.industry.siemens.de/traffic/pdf/inhalt/support/downloads/stadtverkehr/detail\\_stadtverkehr\\_en.pdf](http://www.industry.siemens.de/traffic/pdf/inhalt/support/downloads/stadtverkehr/detail_stadtverkehr_en.pdf)>
- [21] *Telvent, the Global RealTime IT Company*. [Interaktyvus] [Žiūrėta 2005-01-02] Prieiga per internetą: <<http://www.telvent.com/solutions/traffic.asp>>
- [22] *Urban Traffic Management & Control: Research Activities*. [Interaktyvus] [Žiūrėta 2004-12-17] Prieiga per internetą: <<http://www.utmc.gov.uk/research/index.htm>>
- [23] *WIKIPEDIA: Floyd-Warshall algorithm*. [Interaktyvus] [Žiūrėta 2006-05-15] Prieiga per internetą: <[http://en.wikipedia.org/wiki/Bellman-Ford\\_algorithm](http://en.wikipedia.org/wiki/Bellman-Ford_algorithm)>
- [24] *WIKIPEDIA: Shortest path problem*. [Interaktyvus] [Žiūrėta 2006-05-12] Prieiga per internetą: <[http://en.wikipedia.org/wiki/Shortest\\_path\\_problem](http://en.wikipedia.org/wiki/Shortest_path_problem)>



## TERMINŲ IR SANTRUMPŲ ŽODYNAS

ITS – Intelektualiosios transporto sistemos

TCP – *Transmission Control Protocol*

FRAME – *Framework Architecture Made for Europe*

AK – asmeninis kompiuteris

CPU – procesorius (*Computing Process Unit*)

DFS – *depth first search*

BFS – *breadth first search*