

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS INSTITUTAS

Lina PUPEIKIENĖ

OPTIMIZAVIMO METODŲ TYRIMAS  
IR TAIKYMAS PROFILIUOTŲ MOKYKLŲ  
TVARKARAŠČIŲ SUDARYMO  
UŽDAVINIUOSE

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,  
INFORMATIKOS INŽINERIJA (07T)



Vilnius LEIDYKLA TECHNICA 2009

Disertacija rengta 2004–2009 metais Matematikos ir informatikos institute.

**Darbo mokslinis vadovas**

prof. habil. dr. Jonas MOCKUS (Matematikos ir informatikos institutas,  
technologijos mokslai, informatikos inžinerija – 07T).

*<http://leidykla.vgtu.lt>*

VG TU leidyklos TECHNIKA 1603-M mokslo literatūros knyga

ISBN 978-9955-28-416-1

© Pupeikienė, L., 2009

© Vilniaus Gedimino technikos universitetas, 2009

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY  
INSTITUTE OF MATHEMATICS AND INFORMATICS

Lina PUPEIKIENĖ

INVESTIGATION AND APPLICATION  
OF PROFILED SCHOOL SCHEDULING  
TASKS OPTIMIZATION METHODS

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,  
INFORMATICS ENGINEERING (07T)



LEIDYKLA  
Vilnius TECHNIKA 2009

Doctoral dissertation was prepared at Institute of Mathematics and Informatics in 2004–2009.

**Scientific Supervisor**

Prof Dr Habil Jonas MOCKUS (Institute of Mathematics and Informatics, Technological Sciences, Informatics Engineering – 07T).

# Santrauka

Profiliuotos mokyklos tvarkaraščio kūrimas yra aktualus uždavinys tiek Lietuvoje, tiek kitose šalyse. Nėra žinoma polinominių būdų šiai problemai spręsti.

Pagrindinis šios daktaro disertacijos objektas yra ištirti euristinius metodus, skirtus profiliuotos mokyklos tvarkaraščio optimizavimui. Tvarkaraščio formavimo kriterijai, kurie yra reikalingi realiame gyvenime, nustatomi vektorinio optimizavimo metodais bei realizuojami nuo operacinės sistemos nepriklausoma programine įranga.

Eksperimentinių tyrimų uždavinys – surasti tokius euristinių metodų parametrus, kurie minimizuotų numatytą nuokrypį.

Disertacijoje aprašomi keturi optimizavimo metodai: lokalus determinuotas (LD), lokalus atsitiktinis (LA), atkaitinimo modeliavimo (AM) ir AM parametrų optimizavimas naudojant Bayes (BA) metodą. Kombinuotas AM ir Bayes metodas duoda geriausias rezultatus.

Pirmajame apžvelgiama su mokyklos tvarkaraščių formavimu susijusi literatūra. Analizuojami tinkamiausi optimizavimo metodų darbo aspektai. Analizuojamos populiariausios programavimo kalbos, tinkančios kurti mokyklų tvarkaraščių optimizavimo programą.

Antrajame skyriuje formuluojamas profiliuotų mokyklų tvarkaraščio kūrimo matematinis modelis. Analizuojami profiliuotose mokyklose naudojami euristiniai parametrai. Atlikti populiarių tvarkaraščių programų vertinimai ir analizės. Įvardijami šių programų trūkumai.

Trečiajame skyriuje, remiantis 2 skyriaus analize ir išvadomis, pateiktas profiliuotos mokyklos lanksčios tvarkaraščio optimizavimo programos projektavimas. Pateiktas pradinio duomenų failo projektavimas, kuris yra informatyvus ir įvertina daugelį mokyklos nuostatuose apibrėžtų euristinių parametrų. Aptariami suformuluoti siūlymai programos plėtojimui.

Disertaciją sudaro trys skyriai, bendrosios išvados ir rekomendacijos, literatūros sąrašas ir autoriaus publikacijų sąrašas. Disertaciją sudaro 124 puslapiai, 53 iliustracijos ir 10 lentelių. Disertacijos temai skaityti pranešimai 5 konferencijose. Paskelbti 2 straipsniai leidiniuose, įtrauktuose į Mokslinės informacijos instituto sąrašą.

# Abstract

The problem of profiled school scheduling is important for Lithuanian schools and for similar schools in many other countries. No polynomial time methods are known for this problem.

The objective of this PhD thesis is to investigate heuristic methods for optimization of profiled school schedules. The convenience of application in real-life situations is provided by the vector optimization approach using platform-independent software implementation.

The task of experimental investigation is to select such parameters of heuristic methods that minimize expected deviation from the optimum.

Four optimization methods were regarded: Local Deterministic (LD), Local Randomized (LR), Simulated Annealing (SA), and SA with parameters optimized using the Bayesian approach (BA). The composite method of AM and BA provided the best results.

In Chapter 1 of the dissertation, various aspects of work of optimization methods as well as popular program languages suitable for school schedule optimization are analyzed. Literature about school scheduling is analysed.

In Chapter 2, conclusions are drawn how the optimization of heuristic parameters influences the speed and accuracy of finding the optimal solution. A technical rating analysis of popular schedule programs is made and technical disadvantages are listed. Criteria for evaluating the quality of results are proposed that include heuristic parameters in search of optimal schedules. Recommendations are states how to assess the choice and optimization of heuristic parameters and methods of the optimization program used.

In Chapter 3, software meant for school schedule optimization in Lithuanian schools program is discussed. Creation of initial data file is described. Proposals for a further development of the program are considered as well.

There are three chapters, general conclusions and recommendations, the list of the references, and the list of publications. The thesis contains 124 pages, 53 figures and 10 tables. The results were presented in five conferences and were published in 2 papers in referred scientific journals.

# Žymėjimai

## Santrumpos

- BA – Bayes metodas (angl. Bayes), optimizuojantis AM parametrus.
- GA – Genetinis algoritmas (angl. Genetic algorithm).
- LA – Lokalus optimizavimo metodas, turintis atsitiktinę paiešką.
- LO – Lokalus optimizavimo metodas.
- AM – Atkaitinimo modeliavimo metodas (angl. Simulated Annealing).
- MK – Monte-Karlo metodas (angl. Monte-Carlo).

## Apibrėžimai

- Bauda – Nusizengimas optimalaus tvarkaraščio formavimo taisyklėms, kuris vertinamas skaičiais.
- Euristika – Teorinio tyrimo, tiesos ieškojimo loginių būdų ir metodinių taisyklių visuma.
- Grupė – Mokykloje nustatytas moksleivių skaičius, kurie mokosi identišką discipliną.
- JAVA – Aukšto lygio programavimo kalba, skirta rašyti dirbančioms tinkle ir palaikomoms bet kokios platformos kompiuterių programoms.
- Klasė – Tai mokykloje oficialiai dokumentuose pavadintas moksleivių sąrašas.
- Langas – Laisva akademinė valanda moksleiviui arba mokytojui.
- Metaeuristika – Sudėtingesnės strategijos, kurios gali valdyti kitas euristikas, tam, kad surastume ypač sudėtingų uždavinių sprendinius. Metaeuristikų pavyzdžiai – Tabu paieška, modeliuojamojo atkaitinimo metodas, genetiniai algoritmai ir kt.
- Patalpa – Vieta, esanti mokyklos viduje, kur mokytojas gali mokinti moksleivių grupę.
- Pedagoginė didaktika – Pamokų eiliškumą ar svarbumą nusakantis parametras.
- Programavimo kalba – Dirbtinė kalba, skirta aprašyti aparato (ypač kompiuterio) funkcionavimą.

- Servlet – Serveryje esantis komponentas, gaunantis iš kliento (pvz., naršyklės) užklausą bei generuojantis atsaką.
- Tvarkaraščio kaimynai – Naujas tvarkaraštis, kuris gaunamas eliminavus vieną langą.



---

# Turinys

<b>ĮVADAS</b> .....	<b>1</b>
Tiriamoji problema.....	1
Darbo aktualumas.....	1
Tyrimų objektas .....	2
Darbo tikslas.....	2
Darbo uždaviniai .....	3
Tyrimų metodika.....	3
Mokslinis darbo naujumas ir jo reikšmė .....	4
Darbo rezultatų praktinė reikšmė .....	4
Ginamieji teiginiai.....	4
Darbo rezultatų aprobavimas .....	5
Disertacijos struktūra .....	5
<b>1. OPTIMIZAVIMO METODŲ IR EURISTINIŲ PARAMETRŲ</b>	
<b>VERTINIMAS</b> .....	<b>6</b>
1.1. Literatūros apžvalga.....	6
1.2. Tyrime naudojamų optimizavimo metodų analizė.....	12
1.2.1. Lokalaus determinuoto (LD) metodo analizė .....	12
1.2.2. Lokalaus atsitiktinio (LA) metodo analizė .....	13
1.2.3. Monte Karlo (MK) metodo analizė.....	14
1.2.4. Atkaitinimo modeliavimo (AM) metodo analizė.....	15

1.2.5. AM metodo, pritaikyto mokyklos tvarkaraščiuose, analizė.....	17
1.2.6. Bayes (BA) metodo analizė .....	18
1.2.7. Tiesioginio Bayes (BA) metodo, papildyto euristikomis (BHA), analizė .....	19
1.2.8. Optimizavimo metodų vertinimas.....	21
1.3. Populiarių programavimo kalbų analizė.....	22
1.3.1. Programavimo kalbos priklausomybės nuo OS analizė.....	22
1.3.2. Programavimo kalbos darbo greičio analizė.....	23
1.3.3. Programavimo kalbos galimybių analizė.....	23
1.3.4. Programavimo kalbos daugiakalbiškumo galimybių analizė.....	24
1.3.5. Programavimo kalbos saugumo ir dinamiškumo analizė.....	25
1.4. Pirmojo skyriaus išvados.....	25
<b>2. OPTIMIZAVIMO METODŲ TYRIMAS IR VERTINIMAS .....</b>	<b>27</b>
2.1. Euristikų, naudojamų tvarkaraščiams optimizuoti, tyrimas .....	27
2.1.1. Profiliuotos mokyklos tvarkaraščio matematinis modelio formavimas .....	29
2.1.2. Baudos taškų skaičiavimo funkcijos formulavimas.....	30
2.1.3. Tvarkaraštį įtakojančių euristinių parametrų formulavimas .....	32
2.1.4. Pareto optimalių tvarkaraščių paieškos formulavimas.....	34
2.2. Euristinių tvarkaraščių formavimo parametrų parinkimas ir optimizavimas.....	36
2.2.1. Euristikų, atitinkančių įstaigos vidaus taisykles, tyrimas.....	36
2.2.2. Euristikų, orientuotų į sparčių tvarkaraščio paiešką, tyrimas.....	38
2.3. Mokyklos euristikų analizės vertinimo rezultatai.....	50
2.4. Rekomendacijos optimizavimo metodams vertinti .....	54
2.5. Lietuvos mokyklose naudojamų programų lyginamoji tyrimas .....	55
2.5.1. Programos „Rector“ tyrimas .....	55
2.5.2. Programos „aSc TimeTables 2008“ tyrimas .....	56
2.5.3. Programos „Mimosa“ tyrimas.....	58
2.5.4. Programos „School – Complete“ tyrimas .....	59
2.5.5. Programos „School Schedule optimization program by Auris“ tyrimas.....	60
2.5.6. Programos „School Schedule optimization program by Vidunas“ tyrimas .....	61
2.5.7. Nagrinėtų programų palyginimas tarpusavyje .....	61
2.6. Antrojo skyriaus išvados .....	63
<b>3. OPTIMIZAVIMO PROGRAMOS KŪRIMO ETAPAI.....</b>	<b>64</b>
3.1. Dabartinės būsenos tyrimas.....	64
3.2. „Optima“ projektavimas.....	66

3.3. Rekomendacijos „Optima“ programai plėtoti .....	70
3.4. Trečiojo skyriaus išvados .....	70
<b>BENDROSIOS IŠVADOS .....</b>	<b>72</b>
<b>LITERATŪROS SĄRAŠAS.....</b>	<b>74</b>
<b>AUTORIAUS MOKSLINIŲ PUBLIKACIJŲ DISERTACIJOS TEMA SĄRAŠAS.....</b>	<b>80</b>
<b>PRIEDAI.....</b>	<b>82</b>
A priedas. Metodų darbo tyrimo metu gauti rezultatai .....	82
B priedas. Naudojimosi instrukcija programa „Optima“ .....	93



---

# Ivadas

## Tiriamoji problema

*Optimizavimo metodų interneto aplinkoje tyrimas ir taikymas profiliuotų mokyklų tvarkaraščių sudarymo uždaviniuose.*

## Darbo aktualumas

Disertacijoje tiriama mokyklos tvarkaraščių optimizavimo klausimai nagrinėjant profiliuotų mokyklų tvarkaraščių formavimą. Esminis skirtumas tarp pagrindinės ir profiliuotos mokyklos yra tas, kad profiliuotoje mokykloje mokomųjų disciplinų yra žymiai daugiau nei pagrindinėje. Taip profiliuotos mokyklos moksleiviai gali pasirinkti žymiai daugiau disciplinų nei pagrindinės mokyklos moksleiviai. Iš tokio praplėsto galimų pasirinkti disciplinų sąrašo moksleivis formuoja savo būsimą mokymosi kursą. Toks praplėstas disciplinų sąrašas suteikia galimybę moksleiviui formuoti tolesnę veiklos kryptį. Tačiau tai labai apsunkina tvarkaraščio formavimą. Tvarkaraštis formuojamas kiekvienai moksleivių grupei pasirinkusiai mokykloje mokomą discipliną. Tvarkaraščio formavimo sudėtingumas eksponentiškai auga didėjant moksleivių kiekiui. Iš to seka, kad profiliuotų mokyklų tvarkaraščių formavimo uždaviniai priklauso NP

sudėtingumo klasei (Felinskas *et al.* 2005), (Mockus 2004). Todėl praktikoje šias problemas spręsti sunku (Garey *et al.* 1979). Šiems uždaviniams nėra sukurtų polinominio sudėtingumo metodų, todėl pagrindinis dėmesys skiriamas euristiniams metodams.

Optimizavimo programų darbas priklauso nuo tinkamo metodų parinkimo duotai uždavinių klasei. Preliminarūs skaičiavimai parodė, kad mokyklų tvarkaraščių optimizavimui tinka plačiai naudojamas atkaitinimo modeliavimo metodas (Felinskas *et al.* 2005). Tačiau atkaitinimo modeliavimo efektyvumas priklauso nuo tinkamo parametrų parinkimo (Mockus 2004). Naujas šio darbo elementas yra atkaitinimo modeliavimo parametrų optimizavimas naudojant pritaikytus Bayes metodus. Kitas naujumo elementas – tai vektorinio optimizavimo teorijos taikymas nustatant tokius Pareto optimalius tvarkaraščius, kuriuose būtų įvertinti subjektyvūs kriterijai pagal konkrečias mokykloje suformuotas sąlygas (Felinskas *et al.* 2003). Abu naujumo elementai yra skiriamoji duoto darbo savybė lyginant su visais anksčiau atliktais mokyklų tvarkaraščių optimizavimo darbais.

Praktinio panaudojamumo galimybės priklauso nuo programinės realizacijos, todėl šių galimybių tyrimui buvo sukurta originali programa skirta darbui interneto aplinkoje. Tam, kad programa būtų visiškai nepriklausoma nuo aplinkos buvo realizuotas JAVA kalba servlet režimas. Sprendžiant pagal publikacijas, tai pirmas servlet režimo panaudojimas mokyklų tvarkaraščių formavimo ir optimizavimo programoms realizuoti. Šiuo atveju suformuoto mokyklos tvarkaraščio optimizavimas atliekamas itin paprastai – naudojant naršyklę ir be jokių papildomų priemonių.

## **Tyrimų objektas**

Euristinių, polinomiškai nesprenžiamų uždavinių optimizavimo metodų tyrimas ir taikymas formuojant profiliuotų mokyklų tvarkaraščius.

## **Darbo tikslas**

Ištirti euristinius polinomiškai nesprenžiamų uždavinių optimizavimo metodus formuojant profiliuotų mokyklų tvarkaraščius.

## Darbo uždaviniai

Darbo tikslui pasiekti darbe reikia spręsti šiuos uždavinius:

1. Atlikti literatūros mokyklos tvarkaraščių optimizavimo klausimais analizę.
2. Atlikti programavimo kalbų, tinkančių mokyklų tvarkaraščiams optimizuoti analizę tam, kad sukurtą produktą vartotojai lengvai įsisavintų.
3. Atlikti mokyklos tvarkaraščiams optimizuoti tinkamiausių optimizavimo metodų analizę. Ištirti jų veikimo principus.
4. Ištirti mokyklose naudojamų euristinių parametrų įtaką sprendimo greičiui bei tikslumui parenkant tinkamą optimizavimo metodą šiems parametrams optimizuoti.
5. Įvertinti būtinų ir pageidaujamų parametrų įtaką skirtingiems optimizavimo metodams ieškant tikslesnio sprendinio.
6. Išanalizavus tvarkaraščių formavimo principus, parengti jų vertinimo ir taikymo rekomendacijas euristinių parametrų parinkimui ir optimizavimui.
7. Parengti Lietuvos Respublikoje naudojamų mokyklų tvarkaraščių optimizavimo programų testinius skaičiavimus bei šių programų lyginamąją analizę.
8. Sukurti ir išbandyti praktinį profiliuotos mokyklos tvarkaraščių optimizavimo funkcionavimo pavyzdį, t. y. Lietuvos Respublikos mokykloms skirtą profiliuotos mokyklos tvarkaraščių formavimo ir optimizavimo programą.
9. Realizuoti rezultatus interneto aplinkoje, sudarant patogias praktinio pritaikymo sąlygas. Taip pat parengti siūlymus tolimesniam optimizavimo programos tobulinimui bei plėtojimui.

## Tyrimų metodika

Optimizavimo priemonių kūrimo principų bendrajai analizei ir jos rekomendacijoms parengti naudojami mokslinės literatūros tyrinėjimo ir lyginamosios analizės metodai. Optimizavimo priemonės vertinimo analizei ir taikymui, euristinių parametrų parinkimui ir optimizavimui, programos panaudojamumui įvertinti naudojamas eksperimentinio tyrimo metodas.

Optimizavimo programos euristinių parametų ir optimizavimo metodų rekomendacijų analizei atlikti taikomas lyginamosios analizės metodas. Rezultatų apibendrinimas atliktas taikant analitinio tyrimo metodą.

## Mokslinis darbo naujumas ir jo reikšmė

Rengiant disertaciją buvo gauti šie informatikos inžinerijos mokslui nauji rezultatai:

1. *Efektyvesnis sprendinys* yra pasiekiamas kitaip nei žinomoje literatūroje, nes atkaitinimo modeliavimo parametrus optimizuojame Bayes metodu.
2. *Vektorinio optimizavimo teorijos naujoviškas taikymas* nustatant tokius Pareto optimalius tvarkaraščius, kurie įvertintų mokyklos suformuotas sąlygas atspindinčius kriterijus.
3. JAVA kalbos servlet režimo pritaikymas kuriant tokią naują tvarkaraščių optimizavimo programą, kuri užtikrintų, kad *praktinio panaudojimo sąlygos būtų patogios*, o operacinė sistema neįtakotų programos veikimo.

## Darbo rezultatų praktinė reikšmė

Gauti tyrimų rezultatai gali būti naudojami bendrojo lavinimo įstaigose (profiluotose mokyklose). Disertacijoje suformuoti euristiniai metodai su optimizuojamais euristikų parametrais buvo pritaikyti Marijampolės Rygiškių Jono gimnazijoje.

## Ginamieji teiginiai

1. Atkaitinimo modeliavimo parametų optimizavimas naudojant Bayes metodą yra naujas ir efektyvus būdas padidinti sprendimo tikslumą.
2. Specifinių, mokykloje nustatytų, sąlygų įvertinimas formuluojant mokyklos tvarkaraščių optimizavimo uždavinius pasiekiamas taikant vektorinio optimizavimo teorijos skaliarizacijos metodą. Jis užtikrina Pareto optimalumą.
3. JAVA kalbos servlet režimo pritaikymas sudaro patogias praktinio panaudojimo sąlygas bei užtikrina visišką nepriklausomumą nuo operacinės sistemos.



## Darbo rezultatų apibavimas

Darbo rezultatai buvo paskelbti 6 leidiniuose: 2 straipsniai, publikuoti recenzuojamuose periodiniuose leidiniuose, įtrauktuose į Lietuvos mokslo tarybos tvirtinamą sąrašą, 1 straipsnis leidiniuose, įtrauktuose į mokslinės informacijos konferencijos darbų (ISI Proceedings) sąrašą ir 4 straipsniai kituose konferencijų rinkiniuose.

Disertacijoje atliktų tyrimų rezultatai buvo paskelbti šiose tarptautinėse bei nacionalinėse konferencijose:

- 11-oji Lietuvos jaunųjų mokslininkų konferencija „*Mokslas – Lietuvos ateitis*“, Vilnius, Vilniaus Gedimino Technikos Universitetas, 2008 balandžio 10.
- 2nd International Conference „*Informatics in Secondary Schools: Evolution and Perspectives*“ (ISSEP 2006). Vilnius, Lithuania 7–11 November, 2006,
- KTU mokslinė-teminė konferencija „*Informacinės technologijos 2006*“, Kaunas, Kauno technologijos universitetas, 2006 sausio 25–26.
- Kompiuterininkų dienos 2005. „*12-osios mokslinės kompiuterininkų konferencijos pranešimų medžiaga*“, Klaipėda, 2005 m. rugsėjo 15–17.
- KTU mokslinė-teminė konferencija „*Informacinės technologijos 2005*“, Kaunas, Kauno technologijos universitetas, 2005 sausio 26–28.

## Disertacijos struktūra

Darbą sudaro 3 skyriai, tarp jų bendrosios išvados ir rekomendacijos, literatūros sąrašas ir publikacijų sąrašas. Taip pat yra du priedai.

Darbo apimtis yra 124 puslapiai, įskaitant priedus, tekste panaudotos 53 paveikslai ir 10 lentelių. Rašant disertaciją buvo panaudoti 89 literatūros šaltiniai.

## Optimizavimo metodų ir euristinių parametrų vertinimas

Skyriuje apžvelgiama su mokyklos tvarkaraščių formavimu susijusi literatūra. Analizuojami tinkamiausi optimizavimo metodų darbo aspektai. Analizuojamos populiariausios programavimo kalbos, tinkančios kurti mokyklų tvarkaraščių formavimo ir optimizavimo programą.

### 1.1. Literatūros apžvalga

Šioje disertacijoje analizuojami optimalių mokymo įstaigos tvarkaraščių formavimo uždaviniai, išskylantys bendrojo lavinimo įstaigose sudarant metinius tvarkaraščius. Tai ypač sudėtingi uždaviniai. Sprendžiant šiuos uždavinius reikia suplanuoti ir optimizuoti užsiėmimų tvarkaraščius moksleivių grupėse. Taip pat suformuoti moksleivių, besimokančių to paties lygio konkrečią discipliną, grupę, nurodyti šiai grupei disciplinos mokytoją, patalpą ir laiką. Tenka atsižvelgti į įvairius ribojimus vertinant šias sąlygas, (Gaidukevičienė *et al.* 2005).

Bendrojo lavinimo mokykla yra trijų pakopų – pradinė, pagrindinė ir vidurinė (t. y. profiliuota). Pradinės mokyklos pakopa apima I–IV bendrojo lavinimo mokyklose klases. Baigę pradinę mokyklą, moksleiviai tęsia mokymąsi

pagrindinėje bendrojo lavinimo mokyklos pakopoje, apimančioje V–X bendrojo lavinimo mokyklos klases. Trečioji bendrojo lavinimo mokyklos pakopa – profiliuota mokykla, kuri apima XI–XII klases. Profiliuotoje mokykloje daugėja mokomųjų disciplinų pasirinkimo variantų. Atsižvelgiant į moksleivių interesus, mokyklos galimybes ir konkrečią situaciją, mokykla gali siūlyti humanitarinį, gamtamokslinį, techninį, ekonominį-komercinį profilius ir kitus variantus. Nors diferenciacija visais aspektais išlieka, naujas ugdymo turinio integracijos lygis pasiekiamas orientuojantis į sisteminių mokymą, įskaitant abstraktų ir probleminių mąstymą. Atsižvelgiant į aukštesnę moksleivių motyvaciją, keliami didesni reikalavimai savarankiškam darbui. Platesnis disciplinų sąrašas suteikia galimybę moksleiviui formuoti norimą veiklos kryptį, bet labai apsunkina tvarkaraščio formavimą. Tvarkaraštis sudaromas kiekvienai mokomai disciplinai grupuojant moksleivius, pasirinkusius identišką disciplinos lygį ir darbo valandų kiekį. Tačiau tai padaryti paprasta jei turime nedaug moksleivių, bet daug mokytojų, patalpų ir disciplinų. Taigi tvarkaraščio formavimas tampa labiau sudėtingas didėjant moksleivių kiekiui mokykloje ir / arba didėjant pasirenkamų disciplinų sąrašui. Sugrupuoti moksleiviai tą pačią discipliną turi mokintis visada kartu. Negalima leisti, kad moksleivis konkrečios disciplinos pamokas lankytų kaskart su skirtingomis grupėmis. Tai įvertinus žymiai sudėtingiau formuoti ir optimizuoti tvarkaraštį.

Profiliuotojo lavinimo įstaigos optimalaus tvarkaraščio formavimo problema yra labai aktuali problema. Ji pradėta nagrinėti maždaug XX a. viduryje. Viskas prasidėjo dėl didelės problemų įvairovės sudarant mokymo įstaigos tvarkaraščius. Kita priežastis buvo faktas, kad mokymo metodika kinta. Taigi kartu kinta ir disciplinos (de Wera 1985), nors tada dar nebuvo jokių profiliavimų. Autoriai J.S.Appleby, D.V. Blake, E.A. Newman (Appleby *et al.* 1961) savo straipsnyje nagrinėja vidutinio dydžio mokymo įstaigos tvarkaraštį. Jų aprašoma vidutinė mokykla turi 30 moksleivių grupių ir 35 pamokas per savaitę. Buvo apskaičiuota, kad turint tik tiek duomenų galimas skirtingų suformuotų tvarkaraščių kiekis apytikriai yra 10 500. Pagrindinė autorių aprašoma problema – kaip paskirstyti moksleivių grupes mokytojams taip, kad per savaitę nei vienas mokytojas nemokintų dviejų grupių vienu metu. Tvarkaraščių formuotojas turėdavo dirbti šimtus valandų formuojant tokį, šių laikų akimis žiūrint, nedidelį tvarkaraštį. Autorių pristatoma programa turėjo sumažinti laiko sąnaudas ir perrinkti daug daugiau galimų variantų per pora valandų. Autoriai kurdami programą naudojami tik logiškais tvarkaraščių formuotojų nuostatais. Pastarieji šiuos nuostatus buvo suformavę iš patirtyje padarytų klaidų bei logiškų nuostatų pagreitinančių darbą formuojant mokyklos tvarkaraštį. Taigi pagrindiniai tvarkaraščių formavimo nuostatai buvo šie:

- Darbo dienų skaičiaus per savaitę ribojimas: paprastai visas tvarkaraštis sudaromas tik vienai savaitei ir kartojamas tam tikrą savaitį skaičių.

- Disciplinų skaičiaus per dieną ribojimas.
- Moksleivių klasių skaičius ir jų paskirstymas: galimos kelios mažesnės moksleivių klasės vienodo amžiaus moksleiviams.

Tvarkaraščiai buvo sudaromi:

- Kiekvienos moksleivių klasės pamokų tvarkaraštį.
- Kiekvienos disciplinos periodiškumo tvarkaraštį.
- Kiekvieno mokytojo tvarkaraštį, nurodant patalpą, kur jis dirba ir discipliną, kurią moko.

Vėliau buvo pradėta naudoti euristinius parametrus. De Werra (de Wera 1971) aprašė euristinį metodą, skirtą tvarkaraščiams formuoti. Šis metodas naudoja kitą metodą apjungiantį į visumą atskirų tvarkaraščių formavimo problemas. Šis metodas buvo pritaikytas mokymo įstaigoje, kur buvo maždaug 50 moksleivių grupių, 80 mokytojų bei 35 pamokos per savaitę. Metodas nesutalpino į tvarkaraštį mažiau negu 5% moksleivių grupių ir mokytojų.

Aust (Aust, 1976) aprašė metodą, skirtą pagreitinti labai sudėtingų tvarkaraščių formavimą. Šiame metode jau buvo pritaikyti euristiniai parametrai, kurie įtakojo:

- Fiksuotų bei uždraustų disciplinų skirstymą tvarkaraštyje.
- Moksleivių pogrupių formavimą.
- Specialios paskirties patalpų skirstymą tvarkaraštyje.
- Mokytojų darbo valandų skirstymą tvarkaraštyje.

Mokyklos tvarkaraščių formavimo metodiką, kurioje buvo naudojama Lagraneno relaksacija su subgradiiento optimizavimu, aprašė Triphaty (Triphaty 1984). Šioje metodikoje buvo naudojama „šakų ir ribų“ procedūra, kuri dirbo pagal specialiai suformuotą disciplinų seką (Ulph 1977). Ši metodika buvo panaudota formuojant tvarkaraštį, turintį 900 pamokų per metus.

Nuo XX a. paskutiniojo dešimtmečio pradžios mokymo įstaigos tvarkaraščiams formuoti buvo pradėtas naudoti atkaitinimo modeliavimo (angl. Simulated Annealing) metodas (Abramson 1991). Paplito euristinės paieškos sistemos. Tačiau iškilo daug sunkumų rašant programą, kuri apimtų visas reikalingas euristikas. Abramson aprašė Monte Karlo (angl. Monte-Carlo (MK)) metodo schemą kaip atkaitinimo modeliavimo metodą ir panaudojo optimizuojant mokymo įstaigos tvarkaraščius. Kadangi įprastas Monte Karlo (MK) technologijas buvo galima naudoti ribotai (Macon *et al.*, 1966), Abramson (Abramson 1991) patobulino šį metodą ir pavadino jį „atkaitinimo modeliavimo“ (AM) vardu. Pagrindinės problemos formuojant tvarkaraštį išliko tos patčios:

- Mokytojas ar moksleivis negali būti dvejose vietose vienu metu.
- Patalpa negali būti naudojama dviejų mokytojų ar dviejų moksleivių grupių vienu metu.

Kad įvertinti objektyvius parametrus buvo panaudota „įkainojimo

funkcija“ arba kitaip vadinama „baudų taškų sumavimo funkcija“ (angl. Cost function). „Įkainojimo funkcija“ skaičiuoja konfliktų kiekį suformuotame tvarkaraštyje. Taigi ši funkcija savotiškai skaičiuoja gauto sprendinio kokybę. Geriausias tvarkaraštis yra tas, kurio kaina / arba baudos taškų suma yra lygi nuliui. Taip optimizavimo problema pavirsta į tvarkaraščio kainos minimizavimą. Formuojant tvarkaraštį kainos buvo nustatomos esant tokiems pažeidimams:

- Moksleivių pogrupiai netelpa į mokykloje esantį patalpų kiekį.
- Nepakanka mokytojų.
- Nesuformuojamos moksleivių grupės ar pogrupiai.

Tai subjektyvūs parametrai, kuriuos pagal vidaus taisykles ir šiandien gali vertinti kiekviena bendrojo lavinimo įstaiga.

Norint suformuoti optimalų tvarkaraštį, reikia minimizuoti langų skaičių per dieną tiek moksleiviams, tiek mokytojams. Bardadym tyrinėjo universitetų, problematika artimų profiliuotom mokyklom, tvarkaraščius. (Bardadym, 1996). Sudarant universitetų tvarkaraščius autorius suklasifikavo 5 universitetų tvarkaraščių formavimo uždavinius:

- Fakultetų tvarkaraščių sudarymas: reikia pasirinkti tam tikros kvalifikacijos dėstytojus numatomoms disciplinoms dėstyti.
- Grupių-dėstytojų tvarkaraščių sudarymas: paskirstyti užsiėmimus, kai mažiausias vienetas tvarkaraštyje yra studentų grupė.
- Užsiėmimų tvarkaraščių sudarymas: paskirstyti užsiėmimus, kai mažiausias vienetas tvarkaraštyje yra atskiras studentas.
- Egzaminų (sesijų) tvarkaraščių sudarymas: paskirstyti egzaminus studentams taip, kad studentai neturėtų dviejų egzaminų tuo pačiu metu.
- Auditorijų skirstymas: susiejus grupes ir dėstytojus, šioms grupė-dėstytojas poroms nurodyti auditorijas.

Šie, tvarkaraščių formavimą įtakojantys uždaviniai, yra labai artimi profiliuotų mokyklų tvarkaraščių formavimą įtakojantiems uždaviniams. Apie tai plačiau rašoma antrajame skyriuje.

Be aukščiau išvardintų metodų populiarūs yra ir genetiniai metodai, vadinami genetiniais algoritmais. Jų formulavimas taip pat buvo pasiūlytas XX-jo amžiaus antroje pusėje. Pirmasis jį sugalvojo J.H. Holland 1960-aisiais metais (Genetic algorithms 2002). Genetiniai algoritmai tvarkaraščių uždaviniuose sėkmingai pradėti taikyti XX-jo amžiaus paskutiniajame dešimtmetyje (Abela 1992), (Adamidis *et al.* 1997), (Chorbev *et al.* 2007), (Lee *et al.* 2001). Jų veikimas yra pagrįstas evoliucija, kuri vyksta gyvojoje gamtoje. Taigi jie imituoja natūraliosios atrankos procesą. Pagrindinės sąvokos, kurios naudojamos modeliuojant biologinės evoliucijos procesus, yra „individas“ ir „populiacija“. „Individas“ yra atskiras sprendinys. „Populiacija“ – sprendinių aibė (grupė,

rinkinys). Turėti „individus“ nepakanka. Reikia įvertinti jų „tinkamumą“. „Individo tinkamumas“ – asocijuojamas su tikslo funkcijos reikšme duotajam sprendiniui. Šių algoritmų darbo rezultatai yra interpretuojami kaip ir gyvosios gamtos evoliucijoje išliekantys tik „vertingiausi individai“. Taigi šiuo konkrečiu atveju sprendžiant mokyklos tvarkaraščio formavimo ir optimizavimo uždavinį siekiama suformuoti kuo „geresnį“ sprendinį, t. y. sprendinį su kuo mažesne tikslo funkcijos reikšme.

Genetiniuose algoritmuose individai yra vaizduojami sprendiniais. Šie sprendiniai yra koduojami simbolių eilutėmis – chromosomomis. Kiekvienas optimizuojamos funkcijos nežinomas kintamasis yra užkoduojamas simbolių, vaizduojančių genus, rinkiniu. Šie rinkiniai yra apjungiami į chromosomas. Optimizavimo metu yra ieškoma ne vieno galimo sprendinio, o jų aibės. Taigi sprendžiant uždavinį yra operuojama ne su viena chromosoma, o su jų populiacija.

Genetiniai algoritmai kilo iš metaeuristinių metodų šeimos. Šiai šeimai būdingos tokios savybės (Felinskas *et al.*, 2006):

1. Operuojama su viena ar keletu leistinų sprendinių „populiacijų“.
2. Atskiri sprendiniai iš vienos ar kelių „populiacijų“ parenkami tolimesniam nagrinėjimui, teikiant pirmenybę tiems sprendiniams, kurie turi „geresnes“ tikslo funkcijos reikšmes.
3. Naudojamos euristinės procedūros, skirtos naujiems leistiniams sprendiniams formuoti, dalinant gautą populiaciją į sprendinių poras ir operuojant šiomis poromis.
4. Naudojamos taisyklės naujų sprendinių generavimui iš atskirų anksčiau gautų sprendinių.

Savybių (2)–(4) realizavimas leidžia populiaciją atnaujinti iteraciniu būdu, paliekant tik „tiksliausius“ sprendinius. Formalizuojant genetinių algoritmų aprašymą, aukščiau minėtos bendrosios savybės susiejamos su atitinkamomis euristinėmis procedūromis:

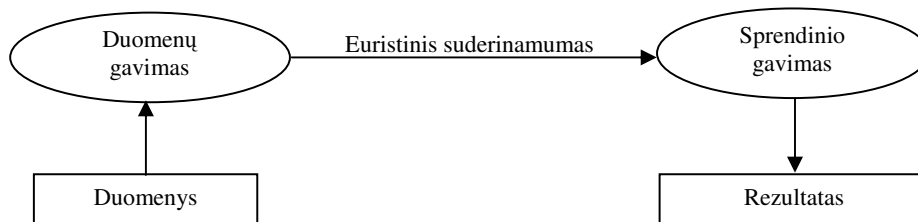
- (2) savybė vadinama atrankos procedūra.
- (3) savybė vadinama kryžminimu.
- (4) savybė vadinama mutavimu.

Vienas didžiausių genetinių algoritmų trūkumų yra tas, kad jie labai lėti (Abela, 1992). Todėl jie nepriėjo profiliuotos mokyklos tvarkaraščių formavimo programose. Norint pritaikyti šį algoritmą profiliuotos mokyklos tvarkaraščių formavime, reikia labai kruopščiai aprašyti „kryžminimo“ funkciją. Šioje funkcijoje turi būti numatyti tiek Lietuvos Respublikos Švietimo ir mokslo ministerijos nuostatai, tiek individualūs mokyklos nuostatai.

Formuojant profiliuotos mokyklos tvarkaraštį neįmanoma apsieiti be euristikų. Tačiau euristikos negali būti bet kokios. Jos turi būti klasifikuotos ir atitikti Lietuvos Respublikos Švietimo ir mokslo ministerijos nuostatus bei

mokyklos vidaus taisykles. Clancey (1985) įrodinėja, kad svarbios klasifikavimo charakteristikos priklauso toms kategorijoms, kurias iš visos galimos sprendinių grupės pažymi specialistas. Panašių problemų pagrindinis bruožas yra nepakankamos klasifikacijos. Taigi atitikimas tarp duomenų ir kategorijos yra tiesioginis. Sudėtingesnių problemų pagrindiniai reikalavimai, suformuoti tam, kad atpažintų tinkamų šakų ir lygių hierarchijas, gali būti ir neįvykdyti. Šiuo atveju reikia panaudoti, kaip Clancey vadina, euristikos klasifikaciją: nehierarchinius ryšius tarp duomenų ir kategorijų susieti su vartotojo sąsaja, o leistinas sąvokas susieti su kitom sistemom. Tai geriausiai parodo 1.1 paveiksle esantis brėžinys, kuriame yra vaizduojami trys pagrindiniai euristinės klasifikacijos žingsniai:

1. Duomenų gavimas.
2. Euristinis jų tarpusavio suderinamumas.
3. Sprendinio gavimas.



**1.1 pav.** Euristinės klasifikacijos struktūros sąsajos

1. *Duomenų gavimas*. Tai naudojama gauti abstraktiems duomenims iš duotų tam tikros grupės duomenų.
2. *Euristinis suderinamumas*. Tarp pradinių duomenų ir galutinių rezultatų šį suderinamumą yra sunku pasiekti. Formuojami pradiniai duomenys, iš kurių pritaikomi tik tam tikri abstraktūs duomenys. Jie sudaro sprendinius, iš kurių gaunami tikėtini ir abstraktūs sprendiniai.
3. *Sprendinio gavimas*. Lyginant sprendinio radimą su siaura sprendinių grupe, reikia identifikuoti ir surikiuoti pradinius sprendinius šioje grupėje. Taigi reikia argumentų apie esamus duomenis arba reikia tikslesnio tolesnio duomenų sukomplektavimo.

Papildžius optimizavimo metodus euristikom yra formuojamas profiliuotos mokyklos tvarkaraštis. Apie tai plačiau yra aprašyta antrajame skyriuje.

## 1.2. Tyrime naudojamų optimizavimo metodų analizė

Nėra tokio universalaus optimizavimo metodo, kurio pagalba būtų galima efektyviai išspręsti bet kokią uždavinį. Tik labai specialiais atvejais optimizavimo uždavinį pasiseka išspręsti analitiškai, t. y. formule užrašyti sprendinį. Ne visada paprasta net patikrinti, ar duotasis taškas  $X \in A$  (čia  $A$  – sprendinių aibė) ir ar yra nagrinėjamojo uždavinio minimumo taškas. Surasti optimalų sprendinį baigtinėje potencialių sprendinių aibėje galima atlikus pilną aibės peržiūrą, t. y. baigtiniu žingsnių skaičiumi. Tačiau baigtinių žingsnių skaičius gali būti ir labai didelis (Žilinskas 2005).

Kiekvienas optimizavimo uždavinys gali būti maksimizavimo arba minimizavimo. Šioje disertacijoje bus nagrinėjamas tik minimizavimo uždavinys. Optimizavimo uždaviniui spręsti bus tiriami keturi skirtingo tipo optimizavimo metodai:

1. Lokalus determinuotas (LD).
2. Lokalus atsitiktinis (LA).
3. Atkaitinimo modeliavimo (AM).
4. Atkaitinimo modeliavimas su Bayes (BA).

Tačiau ne vien nuo optimizavimo metodo priklauso geriausio sprendinio radimas. Kiekvienas metodas turi ir savo optimizuojamus euristinius parametrus. Taigi optimalaus sprendinio radimą įtakoja optimizuojami skirtingi euristiniai parametrai.

### 1.2.1. Lokalaus determinuoto (LD) metodo analizė

Tai labiausia išstobulinta optimizavimo metodų dalis, skirta lokaliajam minimumui rasti. Paprastai šie metodai naudoja informaciją apie tikslo funkcijos savybes tik tuose taškuose, per kuriuos yra leidžiamasi iš atsitiktinai sugeneruoto pradinio taško. Tokį metodą pritaikę globaliojo optimizavimo uždaviniui, rastume tik vieną lokalųjį minimumą (Dzemyda *et al.* 2007).

Iš anksto yra gana sunku nustatyti kokį, lokalaus ar globalaus, optimizavimo uždavinį turime. Optimizuojamos funkcijos lokalaus sprendinio radimas dažnai yra paprastesnis nei globalaus. Lokaliajam ekstremumui nustatyti paprastai pakanka kompiuterio laiko bei kitokių resursų, kurie kinta polinomiškai priklausomai nuo uždavinio matavimų skaičiaus. Bendru atveju **tikslo funkcija** atrodys taip:

$$f(\tau) . \quad (1.1)$$

Praktikoje dažnai taikomas paprasčiausias būdas – ieškoma lokaliųjų minimumų pradendant leisti iš skirtingų pradinių taškų. **Lokoliojo minimumo**



**taškas** yra tas taškas, kuriame tikslo funkcijos reikšmė ne didesnė už reikšmes bet kuriame gretutiniame taške.

$$f(\tau_\varepsilon) \leq f(\tau), \tau \in \varepsilon, \quad (1.2)$$

čia  $\varepsilon$  – šalia  $\tau_\varepsilon$  esantys „kaimynai“ leistinų sprendinių srityje  $A$ .

Jei toks taškas yra tik vienas, tai uždavinys yra išspręstas. Bet jei randami keli lokalieji minimumai, turime globaliojo optimizavimo uždavinį (Dzemyda *et al.* 2007). Taigi lokalis minimumas  $\tau_\varepsilon$  bus tik tam tikroje leistinų sprendinių srityje  $A$  atžvilgiu greta esančių kaimynų  $\varepsilon$  ((1.2) formulė). Kaimynai  $\varepsilon$  šiuo atveju yra tvarkaraščiai suformuoti eliminuojant tik po vieną langą. Langų eliminavimo tvarka yra fiksuota.

Tačiau jei leidžiantis iš daugelio pradinių taškų randamas praktiškai tas pats lokalis minimumas, vis tiek negalima teigti, kad ekstremumas tik vienas. Daugiamatės leistinos srities ištirti patikimai neįmanoma ypač jei leistinoji sritis yra labai didelė.

Uždavinių su ribojimais lokalaus optimizavimo metodai sudaromi iš metodų be ribojimų juos modifikuojant taip, kad būtų įvertinama ne tik funkcijos mažėjimo kryptis, bet ir ribojimai. Jei pradiniu tašku bus pasirinktas atsitiktinis taškas  $x_1$ , iš pradžių jis bus lokalis minimumas. Po to bus ieškoma leistinosios srities viduje tam tikrą iteracijų skaičių. Tikslesnis rastas variantas bus išsaugotas. Taigi paskutinis išsaugotasis variantas taps lokaliuoju minimumu, kai baigsis užduotų iteracijų skaičius.

### 1.2.2. Lokalaus atsitiktinio (LA) metodo analizė

Optimizavimo metuose svarbi lokalojo optimumo, kitaip tariant, optimumo tam tikroje erdvėje, sąvoka. Tai pasyvus metodas, kai parenkant bandymų taškus visai neatsižvelgiama į anksčiau atliktų bandymų rezultatus. Tokia strategija gali būti pritaikyta tik pradinėse optimizavimo stadijose, kai apie tikslo funkciją dar nėra daug informacijos (Dzemyda *et al.* 2007).

Bendra paieškos schema atrodo taip:

- Leistinoje sprendinių srityje  $A$  parenkami bandymų taškai  $\tau_1, \tau_2, \dots, \tau_n$ .
- Juose apskaičiuojamos tikslo funkcijos, apibrėžtos formule (1.1), reikšmės.
- Taikant lokalų paieškos metodą uždavinio sprendiniu laikoma minimali gautoji tikslo funkcijos reikšmė

$$\min_i f(\tau_i), \quad i = 1, \dots, n. \quad (1.3)$$

Minimumo taškas skaičiuojamas taip:

$$\tau_A = \arg \min_A f(\tau), \quad (1.4)$$

čia  $A$  – leistinių sprendinių aibė.

**Lokalus atsitiktinis minimumas** yra išrenkamas tiksliausia suformuotas sprendinys iš lokalių minimumų, gautų lokalaus minimizavimo metodu iš atsitiktinai sugeneruotų pradinių taškų. Lokali minimizacija dažnai vykdoma ne nusileidimo, o paieškos metodais. Taigi lokalus atsitiktinis metodas skiriasi nuo lokalaus determinuoto metodo. Šiuo atveju lokalusis minimumas  $\tau_A$  bus tik tam tikroj leistinių sprendinių srity  $A$  atžvilgiu greta esančių kaimynų  $\varepsilon$ . Kaimynai šiuo atveju yra tos sekos, kurios gaunamos eliminuojant kelis langus atsitiktinai pasirenkant jų eliminavimo tvarką.

Lokali atsitiktinės paieškos realizacija labai paprasta: užtenka turėti lokalaus minimizavimo metodą, kuris būtų kombinuojamas su pradinių taškų generavimu. Nemažai sudėtingesnių metodų gali būti interpretuojami kaip jo patobulinimai.

### 1.2.3. Monte Karlo (MK) metodo analizė

**Monte Karlo metodas** – kompiuterinis skaičiavimo metodas, kuriame uždaviniui spręsti naudojamas atsitiktinių skaičių generatorius. Šis metodas atsirado 1949 metais, kartu su pirmaisiais elektroniniais kompiuteriais, kur buvo pritaikyta tikimybių teorija sudėtingiems procesams atominiuose reaktoriuose modeliuoti.

Monte Karlo metodas – skaičiavimo metodas, pagrįstas statistiniu modeliavimu ir gautų rezultatų apdorojimu statistiniais metodais. Šis metodas leidžia brangiai kainuojančius bandymus pakeisti modeliavimu kompiuteriais ir labai sumažina tyrimų trukmę. Monte Karlo metodai dažniausiai naudojami fizikinių ir matematinių sistemų modeliavimui, kai neįmanoma gauti tikslių rezultatų naudojant deterministinį algoritmą.

Norint atlikti išėjimo parametro reikšmių išsibarstymo tyrimą, reikia turėti ryšio funkciją  $y = f(\tau_1, \tau_2, \dots, \tau_n)$ . Tikslumo tyrimas susideda iš tokių etapų:

1. Modeliuojami elementų parametrų skirstiniai  $W(\tau_i)$ .
2. Skaičiuojamos išėjimo parametro  $y$  reikšmės, esant atsitiktinėms  $\tau_i$  reikšmių kombinacijoms, atitinkančioms  $W(\tau_i)$  dėsnius, t. y. modeliuojamas kūrybinis procesas.
3. Modeliavimo rezultatai apdorojami statistiniais metodais.

Šio apdorojimo tikslas yra įvertinti skaitines išėjimo parametro charakteristikas (vidutinę reikšmę ir dispersiją  $Dy$ ), nustatyti išėjimo parametro skirstinį  $w(y)$  arba surasti tikimybę, kad išėjimo parametro reikšmės bus

duotosiose ribose, kintant elementų parametrų reikšmėms pagal skirstinius (Onno 1981).

Tiriant išėjimo parametrų tikslumą statistinių bandymų metodu, reikalingos elementų parametrų atsitiktinės reikšmės. Šių reikšmių modeliavimui naudojami atsitiktinių skaičių generatoriai. Didžiausią praktinę reikšmę turi tolygiojo skirstinio atsitiktinės reikšmės intervale  $[0, 1]$ . Skaičiai su kitokiais norimais skirstiniais  $w(\tau)$  gaunami naudojant vienodos tikimybės skaičius ir sprendžiant lygtį parametro  $\tau$  atžvilgiu, esant įvairioms tikimybės  $P$  reikšmėms ( $0 \leq P \leq 1$ ).

Tokiu būdu gaunami pseudoatsitiktiniai skaičiai  $\tau_i$ , pasiskirstę pagal skirstinį  $w(\tau_i)$ . Naudojami ir kitokie atsitiktinių skaičių gavimo būdai. Nepriklausomai nuo atsitiktinių skaičių gavimo būdo, apie jų kokybę galima spręsti iš gauto statistinio skirstinio sutapimo su norimu teoriniu skirstiniu. Apie skirstinių sutapimo laipsnį sprendžiama iš sutapimo kriterijų. Praktikoje plačiausiai naudojami Pirsono ir Kolmagorovo sutapimo kriterijai.

Statistinių bandymų metodo pagrindiniai privalumai yra šie:

1. Galima tirti išėjimo parametrų tikslumą, esant bet kokiems elementų parametrų skirstiniams.
2. Galima gauti rezultatus su norimai maža paklaida. Kai bandymų skaičius  $N$  artėja prie begalybės, skaičiavimų paklaida artėja prie nulio.
3. Galima paskaičiuoti kiekybines išėjimo parametrų charakteristikas (vidutinę reikšmę, dispersiją), rasti skirstinį arba tikimybę, kad išėjimo parametras bus duotosiose ribose.

Palyginus su natūrinių bandymų metodu, atsitiktinių bandymų metodas reikalauja mažai lėšų, laiko ir išėjimo parametrų  $y$  tikslumo tyrimui atlikti. Statistinių bandymų metodo trūkumas yra tas, kad sunku generuoti tarpusavyje priklausomų atsitiktinių dydžių reikšmes, t. y. sunku tyrinėti tikslumą, kai elementų parametrai yra priklausomi atsitiktiniai dydžiai.

#### 1.2.4. Atkaitinimo modeliavimo (AM) metodo analizė

Atkaitinimo modeliavimo kilo iš Monte Karlo metodo. Jis gali būti pritaikytas ieškant optimizavimo problemų sprendimų. Plačiau aprašyta teorija ir praktika yra van Laarhoven knygoje (van Laarhoven *et al.* 1987) bei Mockaus knygoje (Mockus 2004). Šis metodas patraukė dėmesį kaip metodas, kuris tinka sudėtingoms optimizavimo problemoms spręsti. Labiausiai jis tinka ten, kur reikia rasti globalų minimumą iš daugelio minimumų gautų optimizavimo intervaluose. Praktiniuose pritaikymuose AM yra efektyviai pritaikytas garsiajame „keliaujančio pirklio“ uždavinyje. Taip pat nagrinėjama galimybė kaip panaudoti šį metodą kosmose. Kai paveldėtosios problemos vėl pasireiškė AM metodo sudėtingumas išaugo žymiai daugiau, nei kombinatorinio.

Atkaitinimo modeliavimo metodo pagrindas yra analogiškas metalurgijoje naudojamam metodui, skirtam konkrečiam išlydytam (įkaitintam) metalui šalti ir kristalizuotis. Metodas imituoja rinkinyje esančių karštų vibruojančių atomų šalimą. Kai atomai yra labai įkaitę, jie gali laisvai ir greitai judėti aplink ir turėti tendenciją atsitiktinai susidurti vienas su kitu. Lėtai šalant kristalo pavidalas tvarkingai išdėstomas iš daugelio jį susiduriančių atomų. Proceso esmė yra lėtas šalimas, naudojant pakankamai laiko tam, kad perskirstytume judrumo netenkančius elementus. Šis kristalas reiškia, kad sistema yra minimalios energijos būklės. Tai yra, kai masė yra šalta – nėra jokio atomų judėjimo ir masės sandara yra užšaldyta (Abramson 1991). Bet jei skystas metalas šąla greitai t. y. „staiga aušta“, tai atomai išsidėstys chaotiškai ir nesuformuos tvarkingo kristalo. Taisyklingo kristalo būklė bus nepasiekiamą. Ši būklė tik šiek tiek priartės prie beformės, turinčios šiek tiek daugiau energijos nei natūraliu būdu šalantis metalas, būklės. Taigi įkaitintas metalas gali būti atšaldyti dviem būdais:

- Atšaldomas staiga: nedaug šansų optimaliam sprendimui rasti.
- Atšaldomas pamažu: šansai surasti optimalų sprendimą išauga.

Metalo „šalimas“ modeliuojamas kaip perėjimo į „blogesni“ suformuotą tvarkaraštį tikimybės mažinimu. Tai techninis AM apibrėžimas. Kartu tai svarbus garantas, kad net ir tuo atveju, kai atsiras „blogiausias variantas“, jis taip pat bus išnagrinėtas. Nors analoginis metodas metalurgijoje nėra tobulas, bet šis minimizavimo metodas, skirtingai nei kiti, vykdo lėtą „šalimą“. Visais atvejais neliko poreikio greitai surasti sprendinius: nuo pradžios taško kiek įmanoma greičiau peržiūrint visus variantus. Tai pirmąją skaičiuojant lokalų, bet nebūtinai globalų minimumą. Natūraliai atsiradęs AM minimizavimo metodas susideda iš dviejų visiškai skirtingų procedūrų. Galimybė pereiti į „blogesni“ tvarkaraštį patobulina tikslesnio tvarkaraščio radimo galimybes.

AM metodo veikimo esmė yra ta, kad bet kokioje duotoje temperatūroje gaunama nauja atomų konfigūracija, kai tik sistemos energija šiek tiek sumažėja. Jeigu sistemos energija yra didesnė, tai nauja konfigūracija gaunama tada, kai tokio temperatūros padidėjimo tikimybė yra mažesnė nei tikėtasi prie duotos temperatūros. Ši tikimybė yra skaičiuojama pagal tokią formulę:

$$P(\Delta c) = e^{-\Delta E / KT} , \quad (1.5)$$

čia  $K$  – Boltzmano konstanta. Plačiau apie tai rašo van Laarhoven (van Laarhoven *et al.* 1987) bei Mockus (Mockus 2004).

Daugelis optimizavimo problemų gali būti įvertintos kaip objektų kiekis, kurio reikia norint suformuoti optimalų tvarkaraštį minimizuojant objektyviąją funkciją. Atomų vibravimas yra keičiamas pagal objektus. Objektyvios funkcijos reikšmė keičiasi su sistemos energija. Pradinis tvarkaraštis yra suformuojamas atsitiktinai skirstant tvarkaraštyje įvertinus objektus. Tam, kad būtų galima

įvertinti tvarkaraščio formavime naudojamus objektus, reikia suteikti jiems kažkokią vertę t. y. „kainą“. Pradinė „kaina“ ( $c_0$ ) yra apskaičiuojama sumuojant nustatytų objektų „kainą“, o pradinė temperatūra ( $T_0$ ) yra nustatoma iš anksto. Vėlesni perstatymai yra kuriami atsitiktinai išrenkant objektus, juos naujai paskirstant ir įvertinant tvarkaraščių „kainų“ skirtumą ( $\Delta c$ ). Jei  $\Delta c \leq 0$ , tai išimamas naujasis tvarkaraštis. Jei  $\Delta c > 0$ , tada sprendimas, kurį tvarkaraštį išiminti, priklauso nuo tikimybės. Jei tikimybė yra didesnė nei atsitiktinai pasirinktas skaičius intervale  $[0,1]$ , tai išimamas naujasis tvarkaraštis. Ši tikimybė esant temperatūrai  $T$ , yra skaičiuojama taip:

$$P(\Delta c) = e^{-\Delta c/T} \quad (1.6)$$

Paprasčiausia technologija išrenkant atsitiktinius objektus yra naudojimas pseudo-atsitiktinis kintamųjų skirstymas vienetų intervale. Radus sėkmingą perstatymą, temperatūra mažėja šalimo greičiu  $R$  tokiu, kad  $NT = T_{n-1} * R$ , kur  $0 \leq R < 1$ , o  $T$  yra realusis skaičius (Abramson 1991) bei Mockus (Mockus 2004).

Vienas iš AM privalumų lyginant su kitais, visada tikslesnio sprendimo ieškančiais („kopiantys į kalną“ metodai), metodais yra tas, kad yra maža tikimybė AM metodui užstrigti lokaliame minimume. Taip yra todėl, kad „kaina“ gali vienodai išaugti arba sumažėti (Simulated annealing 2008).

Tai – standartinis AM metodo aprašymas. Optimizuojant suformuotą mokyklos tvarkaraštį naudojamas šiam uždaviniui spręsti pritaikytasis AM metodas.

### 1.2.5. AM metodo, pritaikyto mokyklos tvarkaraščiuose, analizė

Profiliuotos mokyklos tvarkaraštis yra formuojamas kaitaliojant tvarkaraščiu sudaryti naudojamus duomenis. Tiksliausias suformuotas tvarkaraštis yra įrašomas po kiekvienos iteracijos. Pakeitimas į „blogesnį“ tvarkaraštį yra vykdomas su žymiai mažesne tikimybe, nei pakeitimas į tikslesnį. Taip yra todėl, kad pagerėtų paieškos būklė. Norint suformuoti optimalų tvarkaraštį, pirmas atliekamas žingsnis yra rezultatų gerinimas naudojant AM metodą. Kiekviena operacija pakeičiant mokyklos tvarkaraštį  $i$  į naujai suformuotą mokyklos tvarkaraštį  $i + 1$  yra vykdoma su tokia tikimybe:

$$r_{i+1} = \begin{cases} e^{\frac{-h_{i+1}}{x/\ln(1+(i+1))}}, & \text{kai } h_{i+1} > 0, \\ 1, & \text{kitu atveju} \end{cases} \quad (1.7)$$

čia  $x$  – pradinė „temperatūra“;  $h$  – skirtumas: tikslesnio tvarkaraščio  $i$  baudos taškai minus naujai suformuoto tvarkaraščio  $i + 1$  baudos taškai.

Šio metodo skirtumas nuo tradicinio AM yra tas, kad čia optimizuojamas

parametras  $x$ . Jei iteracijų skaičius yra fiksuotas, tai tvarkaraščio „šalimas“ taip pat turi būti įvertintas. Pats geriausias būdas tai įgyvendinti yra įterpti dar vieną parametą  $x_2$ . Taigi (1.7) formulė atrodys taip:

$$r_{i+1} = \begin{cases} e^{\frac{-h_{i+1}}{x_1 / \ln(1+x_2^{2(i+1)})}}, & \text{kai } h_{i+1} > 0, \\ 1, & \text{kitu atveju} \end{cases} \quad (1.8)$$

čia  $x_1 \geq 0$  – apibrėžia pradinę AM „temperatūrą“;  $x_2 \geq 0$  – apibrėžia AM „šalimo“ greitį.

### 1.2.6. Bayes (BA) metodo analizė

**Bayes teorema** – tikimybių teorijos teorema, kuri nustato įvykio tikimybę, kai stebint žinoma tik dalis informacijos apie įvykius. Kitaip tariant, Tom Bayes įrodė kaip galima daug tiksliau nustatyti tikimybę, imant iš anksto žinomą informaciją ir naujų stebėjimų duomenis (Bajeso teorema 2008).

Plačiai Bayes algoritmą nagrinėjo profesorius J. Mockus knygoje „A set of examples of global and discrete optimization 2“ (Mockus 2004). Tradicinė skaitmeninė analizė optimizavimo metodams yra tiksli visoms juose optimizuojamose funkcijose. Ji apima konkrečius metodus bei gerai tinka analizuojant sudėtingus atvejus. Norint išvengti didelio klaidų kiekio, reikia atlikti daug bandymų. Tačiau tokių bandymų kiekis didėja eksponentiškai didėjant uždavinio sudėtingumui. Tai ir yra pagrindinė kliūtis nagrinėjant sudėtingus atvejus. Tačiau kaip alternatyva yra naudojama vidurkių analizė. Čia klaidų vidurkis nėra ribotas, bet kiek įmanoma sumažinamas. Kaip rezultatas yra imamas optimizuojamų funkcijų aibės vidurkis. Vidurkių analizė yra vadinama Bayes priartėjimu (angl. Bayesian Approach (BA)) (Mockus 2002, 2004, Mockus *et al.* 2000).

Yra keletas būdų, kaip pritaikyti Bayes teoremą optimizacijoje. Pirmasis, tiesioginis BA, yra apibrėžiamas fiksuojant svarbesnius pasiskirstymus  $P$  funkcijų  $f(x)$  aibėje bei minimizuojant Bayes rizikos funkciją  $R(x)$  (Mockus 2002, 2004, Mockus *et al.* 2000). Rizikos funkcija  $R(x)$  yra tikėtinas nuokrypis nuo globalaus minimumo fiksuotame taške  $x$ . Pasiskirstymas  $P$  sudarytas kaip stochastinis modelis iš  $f(x)$ ,  $x \in R^m$ , čia  $f(x)$  gali būti determinuota arba stochastinė funkcija. Nagrinėjant Gauso atvejį tarkime, kad  $(n+1)$ -sis stebėjimas yra paskutinis, matome:

$$R(x) = \frac{1}{\sqrt{2\pi s_n(x)}} \times \int_{-\infty}^{+\infty} \min_{1 \leq i \leq n} (c_n, f(x)) e^{-\frac{1}{2} \left( \frac{y - m_n(x)}{s_b(x)} \right)^2} dz, \quad (1.9)$$

čia  $c_n = \min_i f(x_i) - \varepsilon$ ;  $m_n(x)$  – sąlyginė išimtis atsižvelgiant į stebimas reikšmes  $f(x_i)$ ,  $i=1, \dots, n$ ;  $s_n^2(x)$  – sąlyginė dispersija,  $\varepsilon > 0$  – koregavimo parametras.

Rizikos funkcijos  $R(x)$  minimumas yra apskaičiuojamas taške

$$x_{n+1} = \arg \max_x \frac{s_n(x)}{m_n(x) - c_n}. \quad (1.10)$$

Tiesioginio BA tikslas skirtas rasti, kiek įmanoma mažesnes vidurkių klaidas išlaikant gerėjimo sąlygas. Tai paprastai naudojama tolydžiuose atvejuose.

Kitas kelias – BA euristinis priartėjimas (angl. Bayesian Heuristic Approach – BHA). Jis fiksuoja svarbesnius pasiskirstymus  $P$  pagalbinių funkcijų  $f_k(x)$  aibėje. Šios funkcijos apibrėžia geriausias reikšmes, įgytas  $K$  kartų panaudojus euristiką  $h(x)$ . Tarkime, kad euristika  $h(x)$  priklauso nuo tolydžių parametrų  $x \in R^m$ . Ši euristika padeda optimizuoti pradinę funkciją  $C(y)$ , kuri yra sudaryta iš kintamųjų  $y \in R^n$ , čia  $m < n$ . Kaip paprastai komponentai  $y$  yra diskretūs kintamieji. Nustatanti prioritetus euristika yra pagrįsta žinovo nuomone. Abiejų metodų skirtumai bei panašumai (BA ir BHA) plačiau nagrinėjami 1.2.7. poskyryje.

### 1.2.7. Tiesioginio Bayes (BA) metodo, papildyto euristikomis (BHA), analizė

Prieš nagrinėjant Bayes metodus mokslo srityje buvo nagrinėjami Wiener procesai. Apie juos taip pat rašė J. Mockus (Mockus, 2004) bei jo cituojami autoriai. Wiener modelis numato beveik visų paprastų funkcijų  $f(x)$  tolydumus. Modelis numato, kad padidėjimas  $f(x_4) - f(x_3)$  ir  $f(x_2) - f(x_1)$ ,  $x_1 < x_2 < x_3 < x_4$  yra stochastiškai nepriklausomi. Čia  $f(x)$  priklauso Gauso  $(0; \sigma x)$  bet kokiam fiksuotam taškui  $x > 0$ .

Wiener modelis taip pat priklauso ir daugiamačiams atvejams (Mockus 2004). Paprastai nesudėtingi apytikriai stochastiniai modeliai vartotojams patinka labiau, ypač jei  $m > 1$ . Apytikriai modeliai yra sukonstruoti kaitaliojant tradicines suderintas Kolmogorovo sąlygas. Šios sąlygos reikalauja atvirkštinių  $n$ -tosios eilės matricių, kad apskaičiuotų sąlygines išimtis  $m_n(x)$  ir skirtumus  $s_n^2(x)$ .

Reguliarios suderintos sąlygos kaitaliojamos kai:

1. Funkcija  $R(x)$  yra tolydi.
2.  $x_n$  artėja į globalų minimumą.
3.  $m_n(x)$  ir  $s_n(x)$  išraiškos yra supaprastinamos.

Taigi šiame metode supaprastinta rizikos funkcija  $R(x)$  atrodo taip:

$$R(x) = \min_{1 \leq i \leq n} f(x_i) - \min_{1 \leq i \leq n} \frac{\|x - x_i\|^2}{f(x_i) - c_n}. \quad (1.11)$$

Pagrindinis tiesioginio Bayes tikslas yra sumažinti numatomą nuokrypį. Bet tiesioginis Bayes metodas taip pat yra papildytas tokiais asimptotiniais parametrais:

$$d^* / d_a = \left( \frac{f_a - f^* + e}{e} \right)^{1/2}, n \rightarrow \infty \quad (1.12)$$

čia  $d^*$  –  $x_i$  taškų atstumas nuo globalaus optimumo;  $d_a$  – yra  $x_i$  taškų, esančių nuo globalaus optimumo, atstumų vidurkis galimoje srityje;  $f_a$  – yra funkcijos  $f(x)$  vidurkis šioje srityje;  $f^*$  – yra funkcijos  $f(x)$  vidurkis prie globalaus minimumo;  $e$  – koregavimo parametras, kuris buvo aprašytas formulėje (1.6).

Tai reiškia, kad tiesioginis BA užtikrina kiekvienos funkcijos  $f(x)$  artėjimą į globalų minimumą. Jis taip pat užtikrina prie globalaus minimumo stebėjimų srityje esančių taškų  $x_i$  didesnę skaičių, ypač jei  $n$  yra didelis. Koregavimo parametras  $e$  turi panašią įtaką kaip temperatūra atkaitinimo modeliavimo metode. Tačiau metodai BA ir AM panašūs tik išoriškai.

BA metodo atveju paprastai asimptotiniai parametrai praktiškai neįtakoja sprendinio, nes BA yra pritaikytas mažos apimties sprendimuose. Pasirinkus tašką  $x_{n+1}$ , tikėtino nuokrypio  $R(x)$  minimizavimas tampa papildoma ir sudėtinga optimizavimo problema. Tai reiškia, kad tiesioginis BA metodas gali būti naudojamas kai kintamųjų  $m < 20$ . Tačiau dirbant su profiliuotos mokyklos uždaviniu  $m$  yra gerokai didesnis. Taigi buvo nutarta naudoti euristinį BA metodą. Jis naudojamas optimizuoti paprastoms objektinėms funkcijoms, turinčioms daug kintamųjų. Paprastai problemos sprendžiamos naudojant euristikas, kurios sudaromos vadovaujantis žinovo nuomone. Dažniausia euristikos siejasi su atsitiktinio maišymo procedūromis, kurios priklauso nuo tam tikrų empiriškai apibrėžtų parametrų. Tokiais parametrais gali būti:

1. Pradinė temperatūra AM metode.
2. Skirtingų atsitiktinio maišymo algoritmų tikimybė.

Apimdamas šias problemas, tiesioginis BA metodas tampa tinkamu įrankiu, apimančiu įvairių euristinių parametrų ir jų maišaties optimizavimą. Taigi toks BA metodas yra vadinamas euristiniu BA metodu (BHA). Šis BHA metodas buvo naudojamas formuojant optimalų mokyklos tvarkaraštį. BHA metodas, kaip ir AM metodas, buvo pritaikytas mokymo įstaigos tvarkaraščiams optimizuoti.



### 1.2.8. Optimizavimo metodų vertinimas

Kai optimizavimo uždavinys koku nors metodu išsprendžiamas baigtiniu žingsnių skaičiumi  $N$ , metodo efektyvumas gali būti vertinamas algoritmų sudėtingumo teorijos kriterijais. Tada įvertinama žingsnių skaičiaus  $N$  priklausomybė nuo uždavinio parametrų. Šios priklausomybės tipas apsprendžia metodo sudėtingumo klasę. Kuo greičiau auga  $N$ , tuo sudėtingesnis metodas, arba tuo menkesnis jo efektyvumas (Žilinskas 2005).

Bendru atveju, baigtiniu žingsnių skaičiumi neįmanoma garantuoti sprendinio suradimo, todėl apsiribojama apytiksliu uždavinio sprendimu. Čia apytiksliu sprendiniu vadinamas taškas, apytiksliai atitinkantis (pakankamas arba būtinas) minimumo sąlygas. Uždavinys sprendžiamas iteratyviai: pradedama kuriame nors taške  $X_0$  ir konstruojama seka  $X_i$ , kurios nariai su indeksais  $i \geq N$  apytiksliai atitinka minimumo sąlygas. Jei baigtiniu žingsnių skaičiumi uždavinys neišsprendžiamas, reikalaujama, kad metodas bent jau gerėtų, t. y., kad metodo generuojamos sekos ribiniai taškai sutaptų su uždavinio sprendiniais. Gerėjantys metodai lyginami jų gerėjimo greičio požiūriu (Žilinskas 2005).

Lokalus atsitiktinis metodas skirtingai nei lokalus determinuotas naudoja atsitiktinę paiešką. Tai pats paprasčiausias tvarkaraščio formavimo būdas. Šiame darbe jis naudojamas tam, kad palyginti jį su kitais tvarkaraščio optimizavimo metodais. Toliau nagrinėjami tik tokie metodai, kurie yra efektyvesni už atsitiktinę paiešką.

Atkaitinimo modeliavimo metode atsitiktiniu būdu yra generuojami sprendiniai. Šis metodas darbo metu, pagal tam tikras taisykles, vieną sprendinį keičia kitu. Nauji sprendiniai yra generuojami pasirinkto sprendinio aplinkoje, apskaičiuojant tikslo funkcijos reikšmes. Paprastai aplinkos gylis (spindulys) yra mažinamas optimizavimo metu, pradedant nuo tam tikro fiksuoto dydžio. Tam, kad pereiti prie naujojo sprendinio yra taikoma Metropolio taisyklė, leidžianti su tam tikra tikimybe pasirinkti sprendinį su „blogesne“ tikslo funkcijos reikšme. Atitinkamai parinkus aplinkos mažinimo bei Metropolio taisyklės parametrus, galima pasiekti metodo artėjimą į globalųjį optimumą (Žilinskas 2005).

Bayes metodas buvo pasirinktas AM parametrų optimizavimui. Kadangi AM efektyvus tik tada, kai tiksliai žinomi pradiniai parametrai, buvo nutarta automatizuoti AM metodą naudojant Bayes metodus. Norint rasti efektyvų sprendinį iš karto AM parametrus nustatyti sunku. Įvertinant, kad kiekvienas uždavinys turi specifinius reikalavimus, yra vis sunkiau nustatyti AM parametrus. Tam, kad rasti optimalų sprendinį buvo pasirinktas Bayes metodas, kuris efektyviai optimizuoja AM parametrus.

### 1.3. Populiarių programavimo kalbų analizė

Programavimo kalbos sukurtos tam, kad programuotojai galėtų žymiai paprasčiau rašyti programas. Programavimo kalbos aprašymas susideda iš sintaksinių bei semantinių taisyklių. Viena iš svarbiausių programavimo kalbos paskirčių – ne tik programinio kodo užrašymo bet ir jo perskaitymo ir perpratimo paprastumas.

Kai kurios kalbos dažniau sutinkamos tik tam tikrose joms būdingose nišose. Vienos kalbos buvo sukurtos specialiai tam tikros srities problemoms spręsti, o kitos išpopuliarėjo savo srityje dėl istorinių aplinkybių (Programavimo kalba, 2008). Trumpai apžvelgus programavimo kalbas, jų klasifikaciją, buvo pasirinkta programavimo kalba, kuri tiktų realizuoti tvarkaraščių formavimo ir optimizavimo programą. Kuriant šią programą siekiama realizuoti anksčiau aprašytuosius optimizavimo metodus bei juos palyginti tarpusavyje. Taigi programoje turi būti realizuoti „lokalus determinuotas“, „lokalus atsitiktinis“, „atkaitinimo modeliavimo“ ir „Bayes“ optimizavimo metodai. Profiliuotos mokyklos tvarkaraščių optimizavimo programa turi dirbti tiek Windows, tiek Linux operacinėje sistemoje (*angl.* Operating System (OS)). Taip pat ši programa turi būti orientuota į vartotoją ir turi būti intuityviai valdoma. Tinkamiausia, atitinkanti anksčiau išvardintus kriterijus, programavimo kalba tokiai programai kurti, buvo pasirinkta JAVA.

Programavimo kalba JAVA yra objektiškai orientuota kalba, 1991 metais sukurta Džeimso Goslingo ir kitų Sun Microsystems inžinierių. Kalba oficialiai paskelbta ir išleista 1995 metais. JAVA kalbos pirminis tikslas buvo pakeisti C++ kalbą (Java (kalba) 2008). Kūriniai, sukurti šia kalba gali būti naudojami tiek lokaliame kompiuteryje, tiek nuotoliniu būdu. Vartotojui ypač patogi yra nuotoliniu būdu naudojama programa, nes dirbant su tokia programa nereikia jokių specialių ar papildomų darbo su kompiuteriu įgūdžių.

#### 1.3.1. Programavimo kalbos priklausomybės nuo OS analizė

JAVA programavimo kalba buvo sukurta remiantis populiariausios objektiškai orientuotos kalbos C++ sintakse ir Smalltalk bibliotekomis. Prieš vykdant JAVA kalba parašytą programą, ji yra sukompiliuojama į tarpinį kodą. Šis kodas yra vykdomas JAVA Virtualios Mašinos (JVM) interpretatoriaus. Skirtingai nuo C++, teoriškai JAVA programos gali būti sukompilijuotos, pernešamos ir vykdomos tarp įvairių operacinės sistemos. Taip yra todėl, kad JVM programinė įranga yra realizuota visoms pagrindinėms operacinės sistemos. Nepaisant tokio puikaus sprendimo, praktikoje visgi atsiranda problemų dėl skirtingų JVM realizacijų, jų defektų arba abstrakčios vartotojo sąsajos, kuri yra nesuderinama su operacinės sistemos specifika. JAVA programavimo sistemoje

programuotojo parašytas kodas kompiliuojamas ne į procesoriui specifinę, o į tarpinę formą. Ši tarpinė forma nepriklauso nuo procesoriaus tipo ar operacinės sistemos, todėl iškart tinka vykdyti įvairiose aplinkose.

JAVA programavimo kalba taip pat yra nepriklausoma nuo architektūros. Šioje kalboje visi kintamųjų tipai yra griežtai apibrėžti, todėl nekyla tokių problemų, kaip kinta sveikųjų skaičių reikšmių diapazonas. Priklausomai nuo architektūros – architektūrų skirtumus išsprendžia JVM – JAVA kalba yra interpretuojama. Tai lemia jos nepriklausomumą nuo architektūros ir operacinės sistemos, saugumą bei paprastumą. Todėl interpretavimo aplinkoje gali būti atliekami įvairūs pagalbiniai servisai, pvz., automatinis atminties laisvinimas ir t. t. (Šilingas 2005).

Pagrindinis ir vienas stipriausių JAVA konkurentų yra C#. Tai – liaudiškai vadinama „JAVA“ sukurta firmos „Microsoft“. Ji puikiai dirba Windows aplinkoje, tačiau kitose aplinkose reikia naudoti papildomą emuliatorių „mono“. Šis emuliatorius sumažina C# stabilumą ir efektyvumą.

### 1.3.2. Programavimo kalbos darbo greičio analizė

JAVA palaiko visas pagrindines objektiškai orientuotos programavimo kalbos konstrukcijas – klases ir objektus, paveldėjimą, sąsajas ir jų realizavimą, informacijos slėpimą, pagrįstą matomumo specifikavimu, metodų perdengimą ir perkrovimą, polimorfizmą, skirstymą į paketus ir t. t. Tačiau JAVA turi ir standartines bibliotekas tinklinių programų kūrimui bei paprastam paskirstytų objektų realizavimui, taip pat specialias programas – įskiepius – vykdymui Internet naršyklėse.

Kadangi JAVA yra interpretuojama, JAVA programų veikimas daugeliu atveju buvo lėtesnis už sukompilijuotas programas parašytas C / C++, Fortran, Pascal arba kitomis kompiliuojamomis programavimo kalbomis. Tačiau tai buvo tik pirmosiose JAVA versijose, kur tarpinis kodas buvo interpretuojamas. Tada, būtent dėl tokių priežasčių, JAVA pelnė lėtai dirbančios platformos reputaciją. Tačiau dabar ši problema dažniausiai išskirta tik kuriant atskiras programas su grafine vartotojo sąsaja. Testai rodo, kad kliento / serverio programų veikimas yra dažnai net greitesnis už analogiškas C++ realizacijas. Dabartinės sistemos paprastai prieš vykdydamos tarpinį kodą, jį kompiliuoja. Todėl vykdyimo greitis panašus ar tik nežymiai lėtesnis.

### 1.3.3. Programavimo kalbos galimybių analizė

Daugumos programavimo kalbų galutinis produktas yra vykdomasis modulis, susidedantis iš konkrečių instrukcijų asmeninio kompiuterio procesoriuje. Be to, kiekviena sistema (Windows, Unix, Linux) naudoja savas

papildomas bibliotekas. Todėl vienos sistemos produktas dažnai neveikia kitoje sistemoje. Šio trūkumo neturi programa, parašyta JAVA kalba. Esmė ta, kad čia negeneruojamos instrukcijos procesoriui, bet programos tekstas (kodas) kompiliuojamas į specialų objektinį kodą, vadinamą bait-kodu.

JVM – tai speciali Sun Microsystems firmos sukurta programa, kuri yra pritaikyta kiekvienai operacinei sistemai atskirai. Ji įtraukta į populiariausių naršyklių sudėtį. Be to, ją galima įdiegti ir atskirai.

JAVA leidžia tiesiogiai atstatyti išeitį kodą iš sukompiliuoto. Tai dažnai nėra pageidautina, ypač jei sistema nėra kuriama remiantis „atviro kodo“ principais. Šiai problemai spręsti pasiūlyti obfuskatoriai, kurie iškraipo sukompilijuotos programos kodą taip, kad jis būtų nesuprantamas, bet nepakeičia šio kodo funkcionalumo.

Dėl paprastumo, saugumo, išimčių valdymo bei neseniai įdiegto sąlygų patikrinimo mechanizmo, JAVA yra atspari klaidoms ir leidžia kurti aukštos kokybės programinę įrangą.

#### **1.3.4. Programavimo kalbos daugiakalbiškumo galimybių analizė**

JAVA turi taip vadinamą „natyvią sąsają“, kurios dėka nesunku ją jungti su esančiomis C, C++, C# ar FORTRAN bibliotekomis. Dažniausiai to prireikia jei būtina naudoti šiomis kalbomis parašytas matematinės ar kitokias bibliotekas.

JAVA nuo pat pradžių buvo kuriama kaip nemokama sistema. Visos jos standartinės bibliotekos ir pagrindiniai kūrimo įrankiai – kompiliatorius, interpretatorius, dokumentacijos generatorius ir kt. – yra nemokami ir laisvai platinami. Oficialiose JAVA interneto svetainėse, galima pasklaidyti standartinių bibliotekų išeitį kodą, viešai registruoti surastus defektus, dalyvauti kuriant naujas JAVA technologijas. Dėka pasiteisinusio sistemos atvirumo, buvo sukurta ir daug įvairių nemokamų įrankių skirtų JAVA programų kūrimui, tokių kaip integruotos kūrimo aplinkos (pvz. Eclipse, J2EE), technologijų serveriai (pvz. JBoss, Tomcat), bibliotekos ir karkasai (pvz. JUnit, Struts). Vystantis JAVA kalbai, vis didėjantį jos populiarumą lėmė sparčiai auganti bibliotekų ir technologijų įvairovė. Tai leidžianti lengviau ir sparčiau realizuoti įvairaus specializuoto funkcionalumo programinę įrangą. Šiuo metu JAVA turi bene labiausiai išvystytą technologinę platformą paskirstytų kliento / serverio sistemų kūrimui.

Pačioje JAVA kalboje yra įdiegtos konstrukcijos – gijos, jų valdymas, sinchronizavimas remiantis objektų monitoriais – reikalingos konkurentiniam programavimui. Tai labai palengvina programavimą konkurentinio programavimo reikalaujančiuose uždaviniuose, tokiuose kaip su daug klientų dirbantis serveris arba didelius skaičiavimus atliekanti programa, pagrįsta grafine

vartotojo sąsaja.

### 1.3.5. Programavimo kalbos saugumo ir dinamiškumo analizė

JAVA kalboje yra įdiegtas saugumo mechanizmas, kuris patikrina, ar programinis kodas nėra kenksmingas kiekvienam veiksmui, kuris keičia programos išorę, pvz. sukuria, šalina arba modifikuoja failus, reikalinga tiesiogiai suteikti programai atitinkamus leidimus. Be to, JAVA kalba neturi konstrukcijų, leidžiančių tiesiogiai valdyti atmintį, atlikti žemo lygio operacinės sistemos funkcijas. Visos JAVA programos bendrauja su operacinėmis sistemomis.

JAVA programavimo kalbos saugumas užtikrinamas keliais lygiais:

1. Programinis lygis:
  - a. Kadangi nėra adresų aritmetikos, tai neįmanoma programiškai sugadinti atminties.
  - b. Neįmanoma perpildyti masyvo.
  - c. Negalima ne tik rašyti, bet ir skaityti informacijos už masyvo ir eilutės ribų.
2. Bait-kodo tikrintojas patikrina ar:
  - a. Neperpildytas stekas.
  - b. Su objektais neatliekamos draudžiamos operacijos.
  - c. Tinkamai naudojami registrai.
  - d. Teisingai keičiami kintamųjų tipai.
3. „Smėlio dėžės“ principas „apletui“. Šiuo atveju „įtartina“ (neturinti skaitmeninio parašo) programa kaip vaikas įkeliama į smėlio dėžę ir negali:
  - a. Skaityti / rašyti failo iš / į kliento asmeninio kompiuterio.
  - b. Išmesti, pervardinti, kurti naujų failų ir t. t. kliento asmeniniame kompiuteryje.

JAVA programavimo kalba labai sparčiai vystosi, todėl labai dažnai atsiranda įvairių naujų savybių, bibliotekų, technologijų, leidžiančių efektyviau kurti programinę įrangą.

## 1.4. Pirmojo skyriaus išvados

1. Nustatyta, kad profiliuotų mokyklų tvarkaraščio uždavinys polinomiškai nesprenžiamas, todėl jo optimizavimui reikalingi euristiniai optimizavimo metodai, kurie suteikia galimybę suformuoti optimalų tvarkaraštį pagal griežtus reikalavimus, tačiau neįvertinant euristinio mokytojo svarbumo mokykloje. Šio tvarkaraščio pagrindu profiliuota mokykla gali kurti realų tvarkaraštį.

2. Atlikus literatūros analizę nustatyta, kad JAVA kalbos servlet režimas patogus mokykloms, nes užtikrina visišką nepriklausomumą nuo operacinės sistemos bei nereikalauja papildomos programinės įrangos instaliavimo.

---

## Optimizavimo metodų tyrimas ir vertinimas

Skyriuje formuluojamas profiliuotų mokyklų tvarkaraščio formavimo matematinis modelis. Analizuojamas euristinių parametrų naudojimas, profiliuotų mokyklų tvarkaraščių formavime. Formuluojamos išvados apie euristinių parametrų optimizavimo poveikį, uždavinio sprendimo rezultatui. Formuojamos rekomendacijos, kaip vertinti euristinių parametrų pasirinkimą ir optimizavimą, bei pirmajame skyriuje aprašytus metodus. Atlikti populiarių tvarkaraščių formavimo programų vertinimai ir analizės. Įvardijami šių populiarių tvarkaraščių formavimo programų trūkumai.

Skyriaus tematika paskelbti trys autorės straipsniai (Pupeikienė *et al.* 2005), (Pupeikienė *et al.* 2006), (Pupeikienė *et al.* 2009).

### 2.1. Euristikų, naudojamų tvarkaraščiams optimizuoti, tyrimas

Norint suformuoti bet kokią profiliuotos mokyklos tvarkaraštį, iš pradžių reikia turėti pradinių duomenų aibę (plačiau apie pradinių duomenų aibės sudarymą rašoma trečiajame skyriuje). Šio konkretaus tvarkaraščių formavimo ir

optimizavimo uždavinio sprendimui didelę įtaką turi pradinų duomenų aibėje aprašyti tokie euristiniai parametrai:

- **Disciplinų grupių skirstymas:** mokykloje dėstomos disciplinos, kurios gali būti būtinosios, pasirenkamosios arba pasirenkamosios iš tam tikro alternatyvių disciplinų sąrašo.
- **Mokytojų skirstymas:** mokykloje dirbančių mokytojų duomenys: vardas ir pavardė, dėstomos disciplinos valandų kiekis bei nurodytos moksleivių klasių sąrašas, kurių moksleivius jie moko. Taip pat nurodoma pasirinktas / pasirinkti laisvadienis / laisvadieniai (nurodoma, kokios darbo dienos pageidaujamos laisvos).
- **Moksleivių skirstymas:** mokykloje besimokantis moksleivis, nepriklausomai nuo klasės numerio, gali pasirinkti pasirenkamąsias disciplinas pagal savo norus ar sugebėjimus. Taigi duomenų faile nurodoma moksleivio pavardė ir jo pageidavimai. Šių pageidavimų / pasirinkimų sąrašas negali nusižengti mokyklos vidaus taisyklėms.

Įvertinant šiuos euristinius parametrus pradinės duomenų aibės sudarymo metu, galima suformuoti ir optimizuoti profiliuotos mokyklos tvarkaraštį. Norint suformuoti mokyklos optimalų tvarkaraštį, atsižvelgiant į mokymo įstaigos vidaus taisykles ir tinkantį konkrečiai profiliuotai mokyklai, reikia įvertinti mokykloje naudojamus anksčiau aprašytus euristinius parametrus. Geriausias euristinių parametru vertinimo būdas yra baudos taškai (apie baudos taškų sumavimo funkcijos sudarymą rašoma 2.1.2 skyriuje). Euristinius parametrus įvertinus baudos taškais, tvarkaraščių optimizavimas tampa lankstus, bei lengvai pritaikomas individualiai mokyklai.

Norint suformuoti optimalų tvarkaraštį, reikia minimizuoti suskaičiuotąją baudų sumą per dieną tiek moksleiviams, tiek mokytojams. Truputį modifikavus Bardadym (Bardadym 1996) suformuluotus universitetų tvarkaraščių formavimo nuostatus, gaunami nuostatai, tinkantys profiliuotų mokyklų tvarkaraščio formavimui. Jie skirstomi taip:

- **Moksleivių grupių tvarkaraščiai:** reikia suformuoti moksleivių, kurie mokosi tą pačią discipliną vienodu lygiu, grupę.
- **Moksleivių grupių ir mokytojų tvarkaraščiai:** paskirstyti tam tikros kvalifikacijos mokytojus pamokoms vesti, kai mažiausias vienetas tvarkaraštyje yra jau suformuota moksleivių grupė.
- **Užsiėmimų tvarkaraščiai:** paskirstyti užsiėmimus, kai mažiausias vienetas tvarkaraštyje yra atskirai suformuota moksleivių grupė, turinti ją mokantį mokytoją.
- **Patalpų tvarkaraščiai:** Mokytoją ir jo mokomą moksleivių grupę, paskirti į neužimtą, disciplinai dėstyti tinkančią, patalpą.



### 2.1.1. Profiliuotos mokyklos tvarkaraščio matematinis modelio formavimas

Formuojant profiliuotos mokyklos tvarkaraštį galima keisti vietomis pamokų darbo laiką moksleivių grupėms ir mokytojams. Mokytojų tvarkaraštyje esantys langai čia nėra labai svarbūs. Taip yra, todėl, kad tie patys mokytojai dėsto pamokas ir pagrindinėje mokykloje. Taigi mokytojų tvarkaraštyje esančius langus, esančius profiliuotoje mokykloje, galima eliminuoti į lango vietą tvarkaraštyje įterpiant pagrindinėje mokykloje mokomą pamoką. Tačiau moksleivių tvarkaraštyje langų turi būti kiek įmanoma mažiau. Tai ypač sudėtinga, nes kiekvienas moksleivis mokosi pagal individualų tvarkaraštį. Taigi reikia suformuoti optimalų pamokų tvarkaraštį tiek moksleiviams, tiek mokytojams. Tam reikia įvertinti mokymo įstaigoje naudojamus tvarkaraščių formavimo reikalavimus – vadinamąsias euristicas – bei juos įvertinti baudos taškais. Suformavus profiliuotos mokyklos tvarkaraštį, prie jo vėliau bus jungiamas pagrindinės mokyklos tvarkaraštis.

Jei būtų sudaromas tik pagrindinės mokyklos tvarkaraštis, moksleivių pasirinkimai būtų interpretuojami kaip pamokų eilės. Pamokų eiliškumą taip pat įtakotų ir moksleivių grupių, kuriose tuo metu gali mokintis moksleiviai, eiliškumas. Tam pakaktų suformuoti keturių matmenų matricą (Mockus 2002)

$$\text{schedule } [M][V][G][K], \quad (2.1)$$

čia  $M$  – mokytojų skaičius;  $V$  – galimų dirbti valandų skaičius per savaitę;  $G$  – mokomų moksleivių grupių skaičius;  $K$  – patalpų skaičius, kuriose vykti pamokos.

Sudarinėjant profiliuotos mokyklos tvarkaraštį, matrica (2.1) buvo suformuota taip:

$$\text{schedule } [D[M]][V][G[S]][K], \quad (2.2)$$

čia  $D[M]$  – visų disciplinų matrica;  $M$  – mokytojų skaičius;  $V$  – galimų dirbti valandų skaičius per savaitę;  $G[S]$  – visų moksleivių grupių matrica;  $S$  – moksleivių skaičius grupėje;  $K$  – patalpų skaičius, kuriose gali vykti pamokos.

Kad būtų lengviau interpretuoti mokykloje esantį užimtumą, aprašysime šią (2.2) matricą kaip dvejetainę. Taigi, jeigu šios matricos elementai būtų vienetai arba nuliai, matytume, ar pamoka yra vykdoma, ar ne. Pavyzdžiui,

$$\text{schedule } [d[m]][l][g[s]] = 1 \quad (2.3)$$

reiškia, kad  $d[m]$  – disciplina, kurią gali dėstyti mokytojas  $m$ , vyksta laiku  $l$ , kurio metu yra dėstoma nurodyta disciplina, moksleivių grupės  $g[s]$  moksleiviui, kuris įrašytas  $s$  numeriu. Jei

$$\text{schedule}[d[m]][l][g[s]] = 0 \quad (2.4)$$

Reiškia, kad pamoka nevyksta. Kadangi tik keletas pamokų vyksta specialiai paskirtose patalpose, tai šią (2.2) aprašytą matricą galima aprašyti taip:

$$\text{scheduleX}[D[M]][G[S]][K]. \quad (2.5)$$

Individualūs tvarkaraščiai yra saugomi taip pat trimatėse matricose

$$\text{schedule}_i[D[M]][G[S]][K]. \quad (2.6)$$

### 2.1.2. Baudos taškų skaičiavimo funkcijos formulavimas

Jei mokykla yra didelė, sukurti optimalų, visus būtinus ir pageidaujamus mokyklos nustatytus ribojimus atitinkantį tvarkaraštį, fiziškai yra neįmanoma. Atsiranda reikalavimai, kurie prieštarauja vieni kitiems. Lietuvos Respublikos Švietimo ir mokslo ministerijos nustatyti ribojimai ir taisyklės yra papildomos kiekvienos mokyklos vidaus taisyklėmis ir ribojimais. Tačiau pagrindinis tikslas vertinant šiuos ribojimus yra suformuoti tvarkaraštį, kuriame būtų įvertinti Lietuvos Respublikos Švietimo ir mokslo ministerijos reikalavimai. Taip pat šis tvarkaraštis turi būti priimtinas tiek moksleiviams, tiek mokytojams.

Pagrindiniai ir svarbiausi tvarkaraščio **būtinai ribojimai**:

1. Darbo dienų  $d$  skaičius per savaitę turi būti  $d \leq 5$ .
2. Mokytojas negali dirbti vienu metu keliose skirtingose vietose.
3. Mokytojas negali turėti daugiau nei 36 darbo valandas per savaitę.
4. Moksleivis vienu metu negali dalyvauti keliose pamokose.
5. Moksleivis  $i$  gali turėti  $28 \leq i \leq 32$  pamokas per savaitę.
6. Pamokų  $p$  skaičius per dieną turi būti  $p \leq 7$ .
7. Moksleivių  $i$  skaičius pamokos  $p$  metu turi būti  $15 \leq i \leq 30$ .
8. Vienoje patalpoje negali vykti kelios skirtingo pobūdžio disciplinos vienu metu (pavyzdžiui, matematika ir fizika).
9. Disciplinos, reikalaujančios specialių priemonių ar patalpų, turi vykti specialiai tam skirtose patalpose (pavyzdžiui, kūno kultūra, informatika).

Pagrindinius **reikalingus ribojimus** tvarkaraštyje sudaro:

1. Mokytojų tvarkaraštyje esančių langų eliminavimas.
2. Moksleivių tvarkaraštyje esančių langų eliminavimas.
3. Nepriimtinos darbo valandos.
4. Nepriimtinos darbo dienos.
5. Nepriimtina pamokų eilės tvarka.
6. Moksleivių grupės sudėties kitimas mokantis identiškos disciplinos.

Formaliai jokie būtinų ribojimų pažeidimai nėra galimi. Galimi tik kai

kurie labai nežymūs būtinų ribojimų pažeidimai, jei tai žymiai pagerina tvarkaraščio kokybę. Todėl baudos taškai  $c_r$  įvertinantys būtinus ribojimus, turi būti nustatomi labai griežtai. Taigi pagrindinė būtinų ribojimų baudos taškų vertinimo funkcija atrodo taip:

$$F_f = \sum_r c_r N_r, \quad (2.7)$$

čia  $c_r$  – bauda už būtino ribojimo  $r$  pažeidimą;  $N_r$  – būtino ribojimo pažeidimo numeris;  $r$  – šiuo atveju  $r = 1, \dots, 9$ .

Kai kurias būtinų ribojimų  $c_r$  baudas kiekviena mokykla įvertina pagal suformuotas individualias mokyklos vidaus taisykles. Kiekvienoje mokykloje baudų įvertinimai bus skirtingi. Paprastai tvarkaraščių sudarinėtojo baudos yra skiriamos už tokius nepatogumus:

- $c_m$  – bauda, skiriama mokytojo  $m$  tvarkaraštyje esančiam langui.
- $c_s$  – bauda, skiriama moksleivių grupės  $s$  tvarkaraštyje esančiam langui.
- $c_{mv}$  – bauda, skiriama mokytojui  $m$  už nepriimtą darbo valandą  $v$ .
- $c_{md}$  – bauda, skiriama mokytojui  $m$  už nepriimtą darbo dieną  $d$ .
- $c_{sv}$  – bauda, skiriama moksleiviui  $s$  už nepriimtą darbo valandą  $v$ .
- $c_{pd}$  – bauda, skiriama už pedagoginės didaktikos  $pd$  pažeidimą.
- $c_{mg}$  – bauda, skiriama už mokytojo  $m$  mokomos moksleivių grupės  $g$  sąrašo kaitą.

Taigi pageidaujamų ribojimų baudos taškų sumos skaičiavimo funkcija atrodo taip:

$$F_n = \sum_m c_m L_m + \sum_s c_s L_s + \sum_m \sum_v c_{mv} L_m^v + \sum_m \sum_d c_{md} L_m^d + \sum_s \sum_v c_{sv} L_s^v + \sum_{pd} c_{pd} L_{pd} + \sum_n c_{mg} L_n, \quad (2.8)$$

čia  $L_m$  – mokytojo  $m$  tvarkaraštyje esantis langų skaičius;  $L_s$  – moksleivių grupės  $s$  tvarkaraštyje esantis langų skaičius;  $L_m^v$  – mokytojui  $m$  nepriimtų darbo valandų  $v$  skaičius;  $L_m^d$  – mokytojo  $m$  nepriimtų darbo dienų  $d$  skaičius;  $L_s^v$  – moksleivių grupės  $s$  nepriimtų darbo valandų  $v$  skaičius;  $L_{pd}$  – pedagoginės didaktikos  $pd$  pažeidimų skaičius;  $L_n$  – moksleivių grupės sąrašo kaitos pažeidimų skaičius  $n$ .

Visų baudos taškų sumos skaičiuojamo tokiu būdu:

$$F = F_f + F_n. \quad (2.9)$$

Optimalus bus tas tvarkaraštis, kuris turi mažiausią baudos taškų sumą. Taigi norint rasti optimalų tvarkaraštį, reikia optimizuoti (šiuo atveju

minimizuoti) baudos taškų sumos funkciją

$$\min_{\tau \in A} F(\tau), \quad (2.10)$$

čia  $F(\tau)$  – baudos taškų suma suformavus tvarkaraštį  $\tau$ ;  $A$  – tvarkaraščių, atitinkančių būtinus ribojimus, aibė.

Parametras  $F(\tau)$  priklauso nuo baudos taškų nustatymo pagal individualius mokykloje suformuotus reikalavimus. Todėl yra vertinamas kaip euristika.

### 2.1.3. Tvarkaraštį įtakojančių euristinių parametru formulavimas

Sudarinėjant profiliuotos mokyklos tvarkaraščius reikia įvertinti ne tik Lietuvos Respublikos Švietimo ir mokslo ministerijos nuostatus, bet ir mokyklos vidaus taisykles. Tvarkaraštis turi atitikti pagrindinius tvarkaraščio formavimo kriterijus. Šie kriterijai yra euristiniai ir universaliai tinka visoms profiliuotoms mokykloms. Visi šiuos kriterijus apimantys ribojimai pavaizduoti 2.1 paveiksle. Svarbiausi tvarkaraščio formavimo kriterijai yra tokie:

1. Būtinai ribojimai.
2. Pageidaujami ribojimai.

Norint suformuoti kokybišką tvarkaraštį, kuris tiktų bet kuriai profiliuotai mokyklai, reikia tinkamai parinkti šiuos kriterijus. Bet kurioje mokykloje didžiausią įtaką turi **būtinai ribojimai**, kurie negali būti pažeisti. Jiems įvertinti yra nustatomos didžiausios baudos. Tai tokie ribojimai, kai:

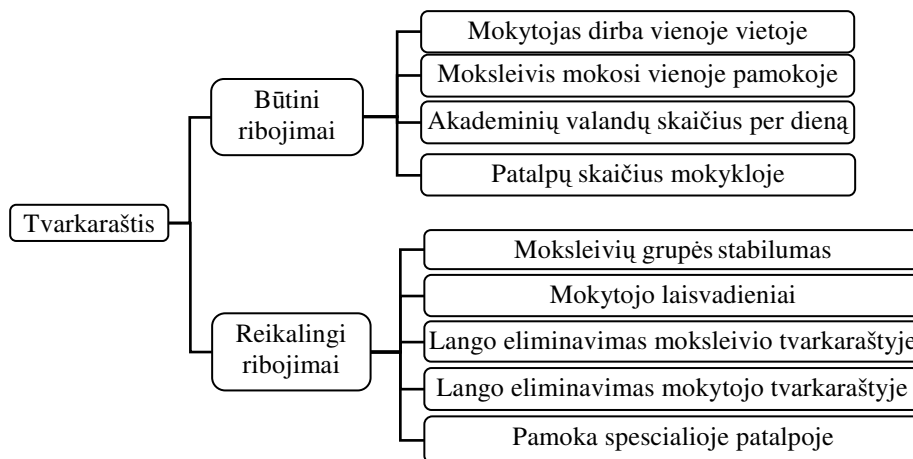
- a. Mokytojas vienu metu gali būti tik vienoje vietoje.
- b. Moksleivis vienu metu gali dalyvauti tik vienoje pamokoje.
- c. Pamokų skaičius per dieną negali būti didesnis nei mokyklos nuostatuose nustatytasis skaičius.
- d. Patalpų skaičius, kuriuose gali vykti pamokos, kiekvienoje mokykloje yra individualus ir fiksuotas.

**Reikalingi ribojimai** vertinami kiekvienoje mokykloje skirtingai. Tai nustato mokykloje suformuotos vidaus taisyklės bei moksleivių ir mokytojų individualūs pasirinkimai. Šie nepatogumų ribojimai skirtingai gali įtakoti mokykloje sudaromą tvarkaraštį.

Nepatogumų ribojimai yra šie:

1. Mokytojas vienu metu gali mokinti tik vieną moksleivių grupę vienos pamokos metu.
2. Mokytojas neturi dirbti per nepriimtina savaitės dieną.
3. Mokytojo tvarkaraštyje esantis langas turi būti eliminuotas.
4. Moksleivio tvarkaraštyje esančio lango eliminavimas.
5. Disciplina privalo vykti tik tam skirtoje vietoje.

Plačiau kalbant apie 1 reikalingą ribojimą, reikia paminėti, kad pamokų eilės tvarka nėra chaotiška. Kiekvienoje mokykloje yra nustatyta kokios disciplinos turi vykti ryte, kokios dienos vidury ir kokios turi būti dienos pabaigoje. Tai nurodoma *pedagoginėse didaktikose*. Šių didaktikų įverčiai nurodomi pradinių duomenų faile. Taip pat moksleivis turi dalyvauti pasirinktose disciplinos pamokoje tik pas tą discipliną mokantį mokytoją ir tik su tai disciplinai mokintis suformuota moksleivių grupe.



**2.1 pav.** Profiliuotos mokyklos tvarkaraštį įtakojantys euristiniai kriterijai

2 reikalingas ribojimas yra naudojamas mažai pamokų mokykloje turintiems mokytojams. Pavyzdžiui, jei mokytojas dirba tik 5 valandas per savaitę, jis nori, kad jo darbo valandos tilptų tik į vieną dieną. Kitokiu atveju šiam mokytojui yra nepatogu dirbti visas penkias darbo dienas po vieną pamoką per dieną.

3 ir 4 reikalingi ribojimai yra labai panašūs. 3 ribojimas yra labai griežtai vertinamas kiekvienoje mokykloje. Jei neeliminuoju mokytojo tvarkaraštyje atsiradusio lango (ribojimas 4) gaunamas tikslesnis tvarkaraštis moksleiviams, tai tvarkaraštyje esantis langas yra paliekamas. Mokytojo tvarkaraštyje esantis langas eliminuojamas, jei tai pagerina (ar bent jau nepablogėja) moksleivių tvarkaraštį (ribojimas 3).

5 reikalingas ribojimas yra skirtas tam tikroms disciplinoms, kurios turi vykti tik specialios paskirties patalpose. Tokios disciplinos yra „Kūno kultūra“, „Informatika“, „Biologija“, „Chemija“ ir t. t.

Kiekvienoje profiliuotoje mokykloje pagrindiniai reikalingi ribojimai, kuriuos reikia eliminuoti mokytojui yra mokytojo tvarkaraštyje atsiradę langai,

laisvadienių ignoravimas ir moksleivių grupės, lankančios tą pačią discipliną, nestabilumas. O pagrindiniai reikalingi ribojimai, kuriuos reikia eliminuoti moksleiviui – tik moksleivio tvarkaraštyje atsiradę langai. Taigi sudarant tvarkaraštį yra labai sunku suderinti visus būtinus ir reikalingus ribojimus. Šiuos ribojimus įvertinus skirtingai, gaunami iš esmės skirtingi tvarkaraščiai. Taigi šios būtinų ir reikalingų ribojimų euristicos lengvai pritaikomos prie individualių mokymo įstaigos poreikių. Plačiau apie euristicų poveikį kalbama atliekant šių euristicų techninę analizę 2.2.2 skyriuje.

#### 2.1.4. Pareto optimalių tvarkaraščių paieškos formulavimas

Išnagrinėję profiliuotos mokyklos tvarkaraščio baudų taškų sumavimo funkcijos minimizavimo uždavinį matome, kad jis turi daug kriterijų. Labai sunku išnagrinėti visus galimus variantus, jei jų yra šimtai ar net tūkstančiai. Šiuo atveju išnagrinėti visus galimus suformuoti tvarkaraščius yra sudėtinga, nes kiekvienas tvarkaraštis apibūdinamas dešimtimis kriterijų. Taigi matome, kad optimalių tvarkaraščių formavimo uždavinys yra vektorinio optimizavimo uždavinys. Euristiniais metodais vargu ar pavyktų iširti didelę variantų aibę ir, atmetus aiškiai nepriimtinius tvarkaraščius. Iš gautos likusių suformuotų tvarkaraščių aibės išrinkti tik tam tikrą, tiksliausiai mokykloje suformuotas taisykles įvertinusių tvarkaraščių skaičių.

Panagrinėkime profiliuotos mokyklos tvarkaraščio pavyzdį, pateiktą 2.1 lentelėje. Čia pavaizduoti skaliariniai kriterijai, naudojami profiliuotos mokyklos tvarkaraščiui formuoti.

Pateiktoje sekoje t. y. vektoriuje

$$f(x) = (f_i(x), i = 1, \dots, m) \quad (2.11)$$

reikia surasti tokį variantą, kuriame nė vienas kriterijus nėra pagerinamas. Reikia sudaryti Pareto optimumų aibę. Pareto optimumas nusakomas aibe  $X^*$ . Šiai aibei priklauso nariai, kurie pagal visus kriterijus ne blogesni už kitus ir bent pagal vieną yra geresni. Taigi Pareto optimali reikšmių aibė  $X^*$  šiuo atveju (pagal 2.1 lentelėje pateiktus duomenis) bus tokia:

$$X^* = \{1, 2, 4, 5, 6, 7\}. \quad (2.12)$$

Pareto optimali reikšmių aibė  $X^*$  (pagal 2.2 lentelėje pateiktus duomenis) bus tokia:

$$X^* = \{8\}. \quad (2.13)$$

Taškas  $x^*$  bus efektyvus sprendinys (optimalus Pareto prasme), kur  $x^* \in X^*$  tada, jei neatsiras tokio  $x$ , kad

$$\begin{aligned} f_i(x) &\geq f_i(x^*), \quad \forall i, \\ f_j(x) &> f_j(x^*), \quad \exists j. \end{aligned} \quad (2.14)$$

Jei Pareto optimali aibė yra maža – surasti optimalų variantą nėra sunku. Tačiau jei Pareto aibė yra labai didelė – reikalinga skaliarizacija. Kadangi daugiakriterinį uždavinį labai sunku optimizuoti, tai natūralu yra pakeisti daugiakriterinį uždavinį vienkriteriniu. Taigi kiekvienam skaliariniam kriterijui suteikiamas svoris. Svorijų dydžiai nustatomi pagal skaliarinio kriterijaus svarbumą. Po to sumuojami visi skaliariniai uždavinio kriterijai su svoriais. Gaunamas vienas kriterijus. Taigi iš formulių (2.7) ir (2.8) matom:

$$\begin{aligned} x(c) = \arg \min_{\tau} ((\sum_r c_r N_r) + (\sum_m c_m L_m + \sum_s c_s L_s + \sum_m \sum_v c_{mv} L_m^v + \\ + \sum_m \sum_d c_{md} L_m^d + \sum_s \sum_v c_{sv} L_s^v + \sum_{pd} c_{pd} L_{pd} + \sum_n c_{ng} L_n)). \end{aligned} \quad (2.15)$$

2.1 lentelė. Tvarkaraščių skaliarinių kriterijų sąrašas

Nr.	Langai mokytojų tvarkaraštyje	Langai moksleivių tvarkaraštyje	Ignoruoti mokytojų laisvadieniai	Pamokos, viršijusios nustatytą skaičių per dieną	Baudos taškų suma
1	14	843	82	12	44255
2	16	860	87	11	42545
3	16	1090	85	18	58915
4	36	656	77	10	37030
5	17	491	81	16	42185
6	38	581	76	10	38585
7	24	525	76	13	38085

2.2 lentelė. Papildytas tvarkaraščių skaliarinių kriterijų sąrašas

Nr.	Langai mokytojų tvarkaraštyje	Langai moksleivių tvarkaraštyje	Ignoruoti mokytojų laisvadieniai	Pamokos, viršijusios nustatytą skaičių per dieną	Baudos taškų suma
1	14	843	82	12	44255
2	16	860	87	11	42545
3	16	1090	85	18	58915
4	36	656	77	10	37030

2.2 lentelės pabaiga

Nr.	Langai mokytojų tvarkaraštyje	Langai moksleivių tvarkaraštyje	Ignoruoti mokytojų laisvadieniai	Pamokos, viršijusios nustatytą skaičių per dieną	Baudos taškų suma
5	17	491	81	16	42185
6	38	581	76	10	38585
7	24	525	76	13	38085
8	10	490	73	6	31605

Pagal formulę (2.15) būtų skaičiuojama visų baudų suma optimaliame tvarkaraštyje. Taigi optimalus sprendinys pagal Pareto optimumą ir pagal 2.1 lentelėje pateiktus duomenis bus  $x^* = 4$ , o pagal 2.2 lentelėje pateiktus duomenis bus  $x^* = 8$ .

## 2.2. Euristicinių tvarkaraščių formavimo parametrų parinkimas ir optimizavimas

Tvarkaraščių formavimo uždavinio esmė yra ta, kad formavimas susideda iš struktūrinių duomenų komponentų, taisyklių komponentų ir euristikos. Euristicinės funkcijos yra skirtos tiksliai įvertinti tvarkaraščių formavimo veiksmus. Rekomendacijų, euristiniams parametrų parinkti ir optimizuoti, formulavimas gali palengvinti skirtingų dydžių tvarkaraščių formavimą. Euristicinius parametrus reikia parinkti optimaliai. Nuo jų priklauso optimizavimo rezultatai ir optimalaus tvarkaraščio radimo greitis. Buvo atlikta euristinių parametrų analizė ir nuspręsta, kaip formali dalis turi būti pateikta tiek kompiuteriui, tiek operatoriui. Taigi pagrindinės sąlygos, pagal kurias grupuojamos ir optimizuojamos euristikos, yra šios:

1. Euristicos turi įvertinti skirtingas profiliuotose mokyklose suformuotas vidaus taisykles.
2. Euristicos turi būti paprastai redaguojamos ir orientuotos į sparčią optimalaus tvarkaraščio paiešką.

### 2.2.1. Euristicų, atitinkančių įstaigos vidaus taisykles, tyrimas

Profiluotoje mokykloje pagal jos vidaus taisykles suformuoti euristiciniai parametrai buvo aprašyti 2.1.3 skyrelyje. Šie parametrai yra grupuojami tokiu būdu:

1. Mokomų disciplinų grupavimas.



2. Mokytojų pasirinkimų vertinimas.
3. Moksleivių pasirinkimų vertinimas.

Norint, kad programa efektyviai sugrupuotų moksleivius, esančius klasės sąrašuose, į moksleivių grupes bei šioms grupėms mokinti paskirstytų mokytojus, pirmiausia reikia apjungti mokymo įstaigoje dėstomas disciplinas. Dėstomos disciplinos čia gali būti trijų rūšių:

- **Būtinosis:** šias disciplinas privalo mokytis visi moksleiviai. Darbo valandų skaičių per savaitę, skirtas dėstyti disciplinoms, nurodomas mokyklos vidaus taisyklėse.
- **Pasirenkamosios:** moksleiviai šias disciplinas gali mokintis arba ne. Šias disciplinas jie gali rinktis nurodydami disciplinos dėstymo lygį ir darbo valandų skaičių per savaitę kiekvienai disciplinai atskirai.
- **Pasirenkamosios (viena iš sąrašė esančių disciplinų):** šias disciplinas paprastai mokykla grupuoja į atskirus alternatyvių disciplinų sąrašus. Sąrašai formuojami pagal kategorijas, t. y. kalbos, menai ir t. t. Moksleiviai gali pasirinkti tik vieną iš kiekviename sąrašė esančių disciplinų.

Kiekvienoje mokykloje disciplinų rūšys gali būti skirtingos. Tai priklauso nuo mokyklos profilio, dydžio, darbuotojų skaičiaus, darbuotojų kvalifikacijos ir daugelio kitų veiksnių. Disciplinų rūšių parinkimas turi atitikti tiek mokyklos, tiek Lietuvos Respublikos Švietimo ir mokslo ministerijos taisykles.

Kiekvienoje mokykloje turi būti optimaliai parinkta mokytojo dėstoma disciplina, paskirstytos jo darbo valandos, optimaliai nurodytos moksleivių grupės ir atsižvelgta į mokytojui nepriimtinas darbo dienas. Moksleivių grupės, juos mokančiam mokytojui, turi būti sugrupuotos pagal moksleivių pageidaujamą disciplinos lygį ir valandų kiekį. Kiekvieno mokytojo duomenis nusako tokie parametrai:

- Mokomos disciplinos pavadinimas.
- Patalpa(-os) kurioje(-iose) mokytojas gali dirbti.
- Darbo valandų skaičius per savaitę.
- Pavadinimai klasių, kurių moksleivius gali mokinti.
- Nepriimtinos darbo dienos.

Sudarius disciplinų ir jas dėstančių mokytojų sąrašą, reikia suformuoti moksleivių grupes iš moksleivių klasių sąrašų. Šios grupės formuojamos pagal moksleivių pasirinkimus. Moksleiviams galima nurodyti disciplinas iš jau suformuoto disciplinų sąrašo, valandų per savaitę, skirtų disciplinai mokintis bei disciplinos mokymo lygį. Kiekvienas moksleivis, pagal mokykloje suformuotus nuostatus, gali nurodyti disciplinas, jų dėstymo lygį ir jų kiekį per savaitę.

Turint disciplinų sąrašus, mokytojų ir moksleivių pasirinkimus, galima pildyti matricą, pagal 2.1.1 skyriuje pateiktą (2.2) formulę.

### 2.2.2. Euristicų, orientuotų į sparčią tvarkaraščio paiešką, tyrimas

Suformavus pradinį tvarkaraštį yra skaičiuojami jo baudos taškai. Šie baudos taškai yra perskaičiuojami kiekvienoje iteracijoje optimizavimo metu. Kiekviename optimizavimo metode jie skaičiuojami skirtingai. Šioje disertacijoje yra nagrinėjami tokie optimizavimo metodai:

1. Lokalusis determinuotas (LD).
2. Lokalus atsitiktinis (LA).
3. Atkaitinimo modeliavimo (MA).
4. Bayes (BA).

Kiekvieno metodo darbo metu buvo nustatyti vienodi mokyklos euristiciniai parametrai. Atliekant tyrimus tvarkaraštis buvo formuojamas pagrindinius mokyklos parametrus nustačius kaip vidutinės profiliuotos mokyklos parametrus. Pagrindiniai profiliuotos mokyklos euristicinių parametrų įverčiai yra tokie:

- Patalpų kiekis mokykloje – 60 vienetų.
- Akademinų valandų per dieną – 7 valandos.
- Bauda už mokytojo tvarkaraštyje esantį langą – 300 taškų.
- Bauda už moksleivio tvarkaraštyje esantį langą – 5 taškų.
- Bauda už mokytojo darbą nepriimtiniu laiku – 10 taškų.
- Bauda už akademinų valandų per dieną skaičiaus viršijimą – 2000 taškų.
- Bauda už pedagoginės didaktikos pažeidimą – 5 taškų.

Toliau vaizduojami rezultatai, atlikus po 100 bandymų su skirtingais optimizavimo metodų parametrais ir skirtingais optimizavimo metodais. Norint sužinoti, kuris suformuotas tvarkaraštis bus geriausias, tikrinti baudos taškų sumos skirtumą. Baudos taškų sumos skirtumas gaunamas iš baudos taškų sumos, gautos suformavus pradinį tvarkaraštį, atėmus baudos taškų sumą, gautą optimizavimo metodui baigus darbą. Geriausias tvarkaraštis bus tas, kurio baudos taškų sumos skirtumas gaunamas didžiausias.

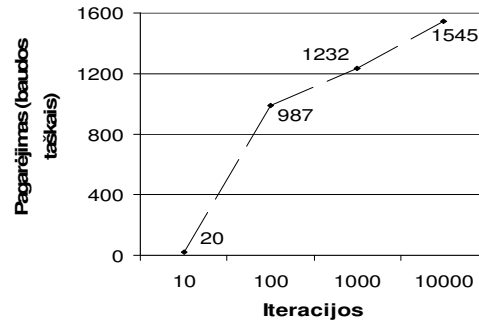
**Lokaliajame determinuotame metode (LD)** nustatomas tik iteracijų skaičiaus. Atlikus bandymus gauti rezultatai aprašyti „A priedo“ 1A lentelėje, bei pavaizduoti 2.2 ir 2.3 paveiksluose.

#### Apibendrinimas:

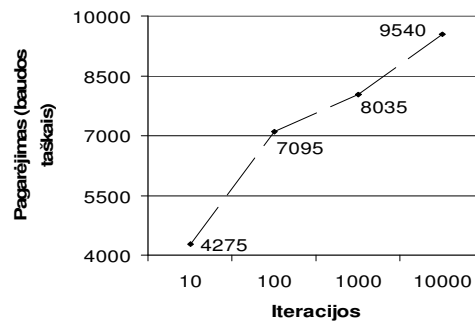
- Kuo daugiau iteracijų yra nustatoma, tuo tikslesnis rezultatas yra randamas.
- Atlikus po 100 bandymų skirtingam iteracijų kiekiui, gautas pagerėjimų vidurkis nėra didelis (2.2 paveikslas). Iš to galime daryti išvadą, kad tvarkaraštyje nėra atsižvelgiama į visus mokykloje suformuotų

reikalavimų euristinius įvertinimus. Galime daryti išvadą, kad metodas dažnai „užstringa“ lokaliame minimume.

- Tik atsitiktiniais atvejais suformuotuose tvarkaraščiuose yra atsižvelgiama į mokykloje suformuotų reikalavimų euristinius įvertinimus (2.3 paveikslas).
- Metodo darbo laikas 7–16 sekundžių.
- Didinant iteracijų kiekį matomas rezultatų gerėjimas.



2.2 pav. Lokaliu determinuotu metodu suformuoti pagerėjimų vidurkiai



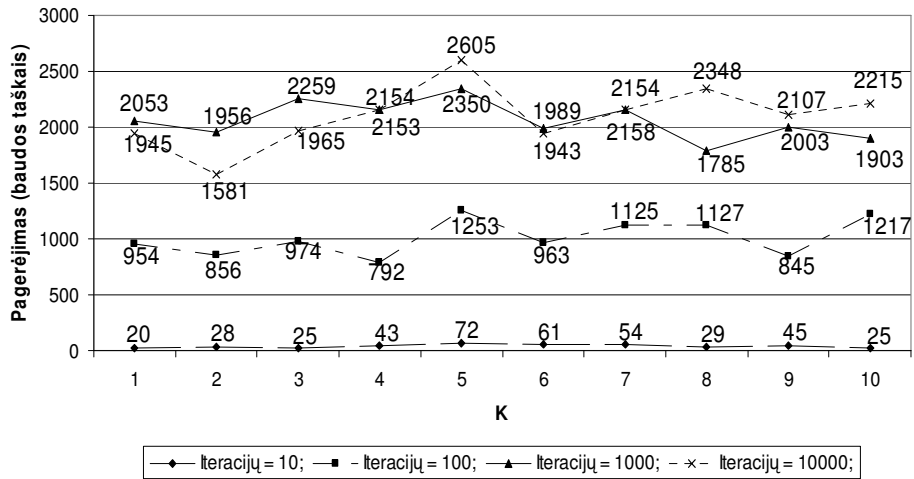
2.3 pav. Lokaliu determinuotu metodu suformuoti geriausi rezultatai

### Išvada:

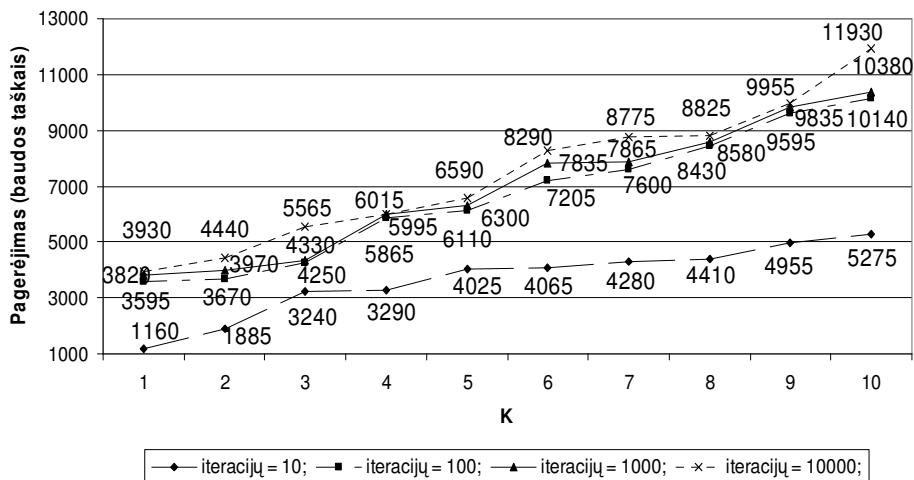
Metodas dažnai „užstringa“ lokaliame minimume (2.2 paveikslas) todėl, buvo nutarta profiliuotų mokyklų tvarkaraščiams optimizuoti LD metodą papildyti atsitiktinumo funkcija. Papildytas metodas buvo pavadintas lokaliu atsitiktiniu (LA) metodu.

**Lokaliajame atsitiktiniame metode (LA)** nustatomas iteracijų skaičius ir skaičius  $K$ . Kaičius  $K$  nusako, kiek kartų bus analizuojamas atsitiktinai

suformuotas mokytojų duomenimis užpildytas sąrašas, eliminuojant tvarkaraštyje atsiradusius langus, ieškant tikslesnio tvarkaraščio varianto. Atlikus bandymus gauti rezultatai aprašyti „A priedo“ 2A lentelėje, bei pavaizduoti 2.4 ir 2.5 paveiksluose.



2.4 pav. Lokaliu atsitiktiniu metodu suformuoti pagerėjimų vidurkiai



2.5 pav. Lokaliu atsitiktiniu metodu suformuoti geriausi rezultatai

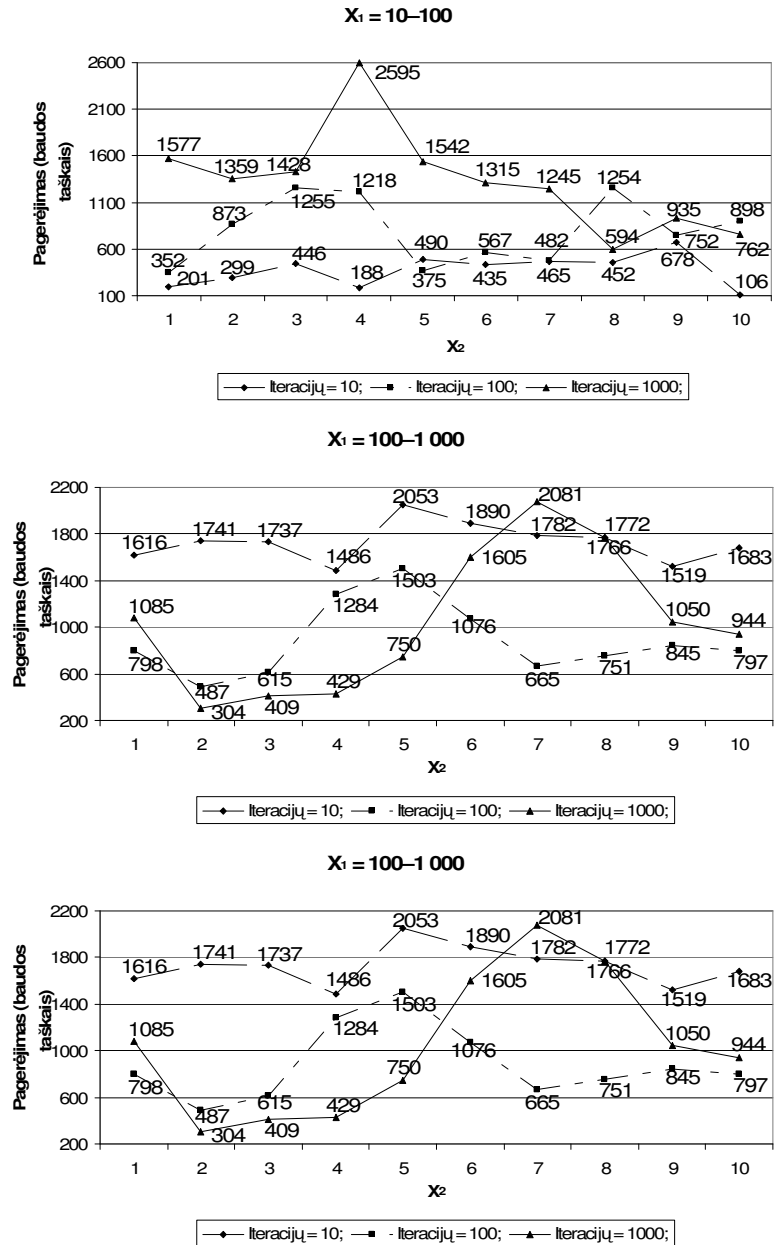
**Apibendrinimas:**

- Kuo daugiau iteracijų yra pasirenkama, tuo tikslesnis rezultatas yra randamas, kadangi nuolat naujai formuojamas mokytojų duomenimis užpildytas sąrašas.
- Lokalų determinuotą metodą papildžius parametru  $K$  ir atliekant po 100 bandymų skirtingam nustatant  $K$  ir iteracijų skaičių (2.4 paveikslas) rezultatų gerėjimas tampa mažai pastebimas. Tai parodo grafike esanti kreivė, kuri tai kyla, tai leidžiasi.
- Atlikus po 100 bandymų skirtingam iteracijų kiekiui, gautas pagerėjimų vidurkis (skaičiai) nėra didelis (2.2 paveikslas). Iš to galime daryti išvadą, kad tvarkaraštyje nėra atsižvelgiama į visus mokykloje suformuotų reikalavimų euristinius įvertinimus.
- Rezultatų pagerėjimas, atliekant po 100 bandymų skirtingam  $K$  ir iteracijų kiekiui, mažai priklauso nuo  $K$  (2.4 paveikslas). Grafike matyti, kad rezultatai, gauti kai  $K = 1$  yra panašūs į rezultatus kai  $K = 10$ .
- Atlikus po 100 bandymų su skirtingais  $K$  ir iteracijų kiekiais (2.4 paveikslas), paprastai tiksliausi rezultatai visais atvejais suformuojami kai  $K=5$ . Grafike matome, kad šis rezultatas visada yra didesnis nei kiti, gauti atlikus tiek pat iteracijų.
- Analizuojant tik geriausius rezultatus gautus atlikus po 100 bandymų (2.5 paveikslas), pastebimas metodo rezultatų gerėjimas. Tai matoma grafike, kur kreivės nuolat kyla.
- Geriausi rezultatai (2.5 paveikslas) atlikus po 100 bandymų suformuojami kai  $K=10$ .
- Nurodžius daugiau nei 5 000 iteracijų, metodas jų visų nepatikrina. Iš to galime daryti išvadą, kad LA metodas taip pat „užstringa“ lokaliame minimume.
- Metodo darbo laikas 10 – 42 sekundės.

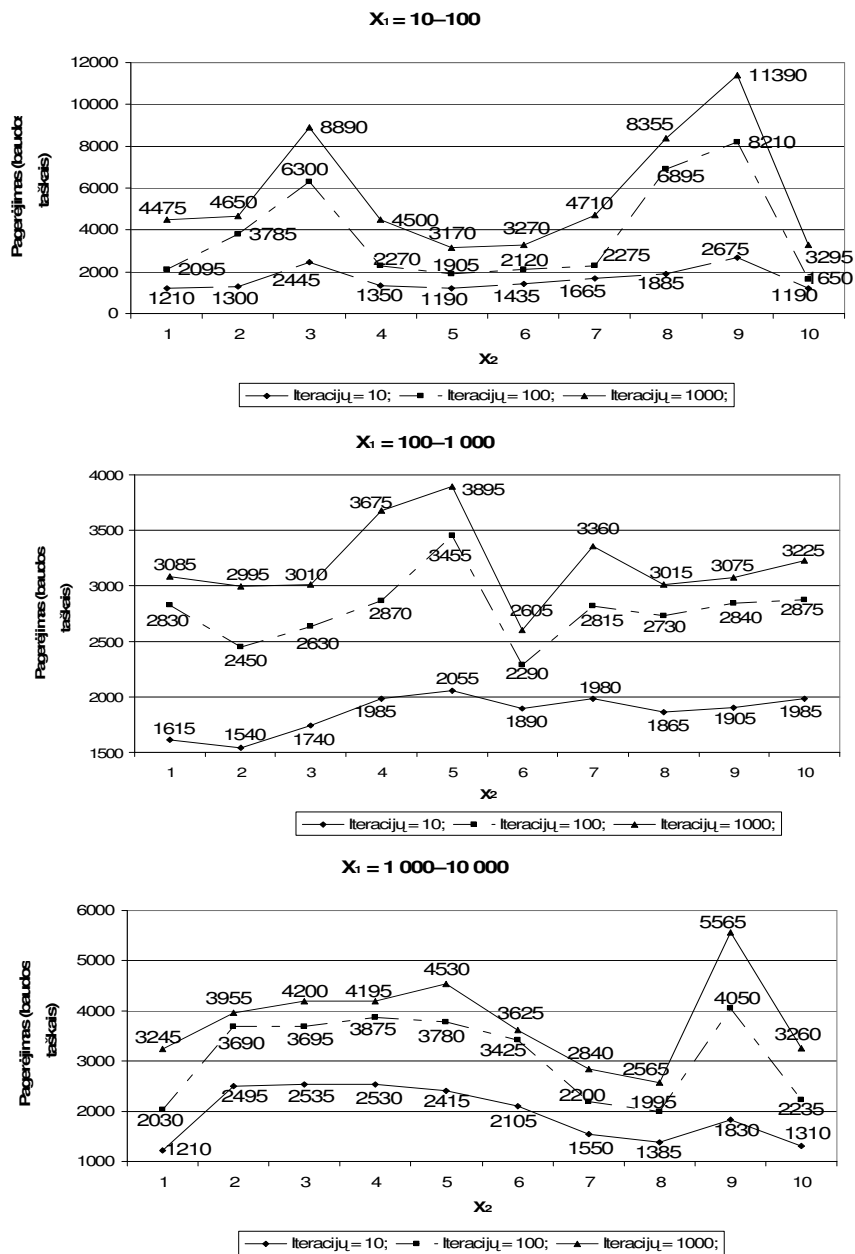
**Išvada:**

LA metodas taip pat dažnai „užstringa“ lokaliame minimume (2.4 paveikslas) todėl, buvo nutarta LA metodą papildyti papildomomis atsitiktinumų funkcijomis. Papildytas metodas buvo pavadintas atkaitinimo modeliavimo (AM) metodas.

**Atkaitinimo modeliavimo metodas (AM)** nustatomas iteracijų skaičius, pradinė „temperatūra“ ( $X_1$ ) ir „šalimo“ greitis ( $X_2$ ). Atlikus bandymus gauti rezultatai aprašyti „A priedo“ 3A ir 4A lentelėse, bei pavaizduoti 2.6, 2.7, 2.8 ir 2.9 paveiksluose. Pradžioje buvo tiriama priklausomybė nuo  $X_2$  gautus rezultatus grupuojant pagal parametą  $X_1$ .



2.6 pav. Atkaitinimo modeliavimo metodu suformuoti pagerėjimų vidurkiai (pagal  $X_2$ )



2.7 pav. Atkaitinimo modeliavimo metodu suformuoti geriausi rezultatai (pagal  $X_2$ )

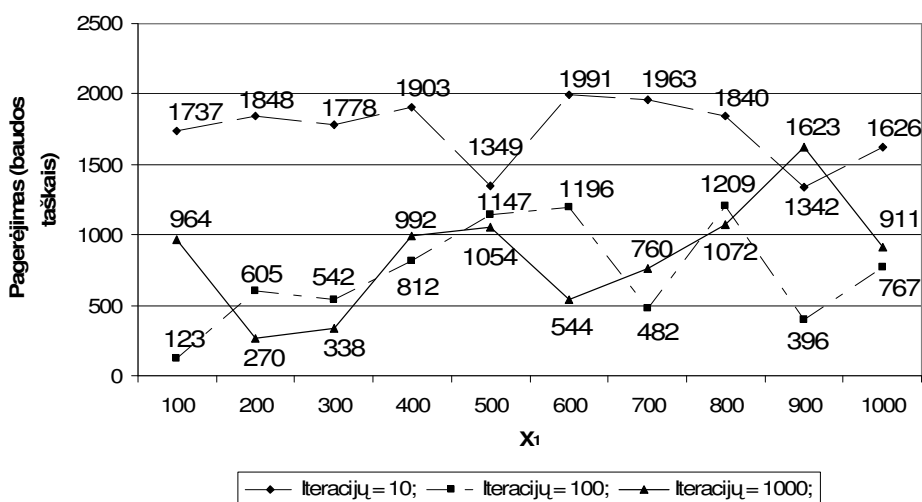
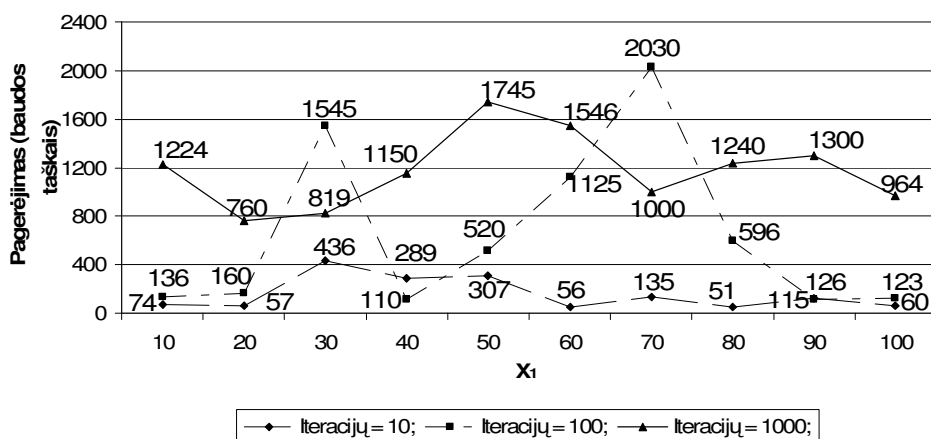
**Apibendrinimas:**

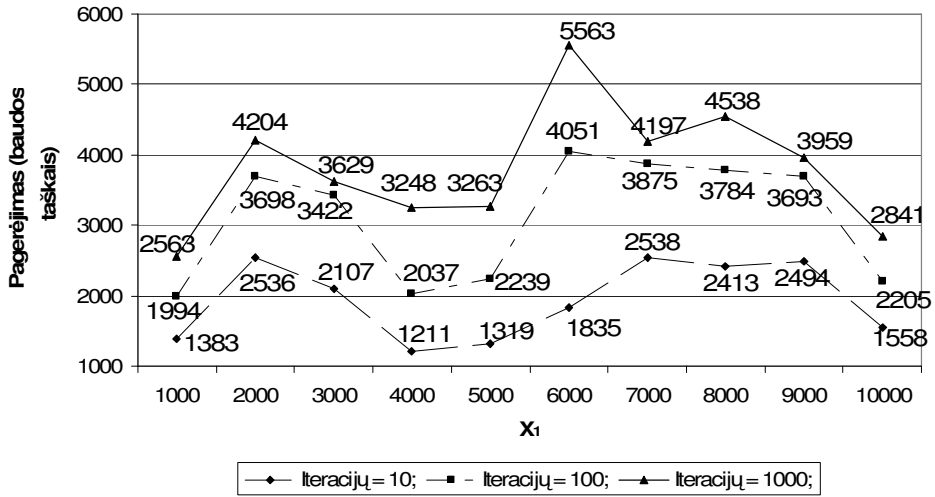
- Atliekant daug bandymų ir juos grupuojant pagal parametą  $X_1$  (2.6 paveikslas), rezultatų gerėjimas yra mažai pastebimas. Tai parodo 2.6 grafikuose esančios kreivės, kurios tai kyla, tai leidžiasi.
- Atliekant po 100 bandymų kiekvienu atveju, geriausi rezultatų vidurkiai gaunami uždavus po 1 000 iteracijų kai  $X_1=10-100$  ir kai  $X_1=1\ 000-10\ 000$  (2.6 paveikslas). Pirmojo ir trečiojo grafikų kreivėse esantys rezultatai paprastai yra didesni už kitus.
- Atliekant po 100 bandymų, kai  $X_1=100-1\ 000$ , geriausi vidutiniai rezultatai yra gaunami nustačius 100 iteracijų (2.6 paveikslas). Antrojo grafiko kreivėje esantys rezultatai paprastai yra didesni už kitus.
- Analizuojant 2.6 paveiksle esančius grafikus matome, kad geriausi vidutiniai rezultatai buvo suformuoti kai:
  - $X_1=10-100$ :
    - Nurodžius 10 iteracijų ir  $X_2=9$ .
    - Nurodžius 100 iteracijų ir  $X_2=4$  bei  $X_2=8$ .
    - Nurodžius 1000 iteracijų ir  $X_2=4$ .
  - $X_1=100-1\ 000$ :
    - Nurodžius 10 ir 100 iteracijų ir  $X_2=5$ .
    - Nurodžius 1000 iteracijų ir  $X_2=7$ .
  - $X_1=1\ 000-10\ 000$ :
    - Nurodžius 10 iteracijų ir  $X_2=5$ .
    - Nurodžius 100 iteracijų ir  $X_2=6$ .
    - Nurodžius 1 000 iteracijų ir  $X_2=3$  bei  $X_2=6$ .
- Geriausi rezultatai, atlikus po 100 bandymų, visais atvejais suformuoti nustačius 1 000 iteracijų (2.7 paveikslas). Visų 2.7 paveiksle esančių grafikų kreivėse esantys rezultatai visada yra didesni už kitus.
- Analizuojant 2.7 paveiksle esančius grafikus matome, kad visada geriausi rezultatai buvo suformuoti kai:
  - $X_1=10-100$ :
    - Nurodžius 10 iteracijų ir  $X_2=9$ .
    - Nurodžius 100 iteracijų ir  $X_2=4$  bei  $X_2=8$ .
    - Nurodžius 1000 iteracijų ir  $X_2=4$ .
  - $X_1=100-1\ 000$ :
    - Nurodžius 10 ir 100 iteracijų ir  $X_2=5$ .
    - Nurodžius 1000 iteracijų ir  $X_2=7$ .
  - $X_1=1\ 000-10\ 000$ :
    - Nurodžius 10 iteracijų ir  $X_2=5$ .
    - Nurodžius 100 iteracijų ir  $X_2=6$ .
    - Nurodžius 1 000 iteracijų ir  $X_2=3$  bei  $X_2=6$ .



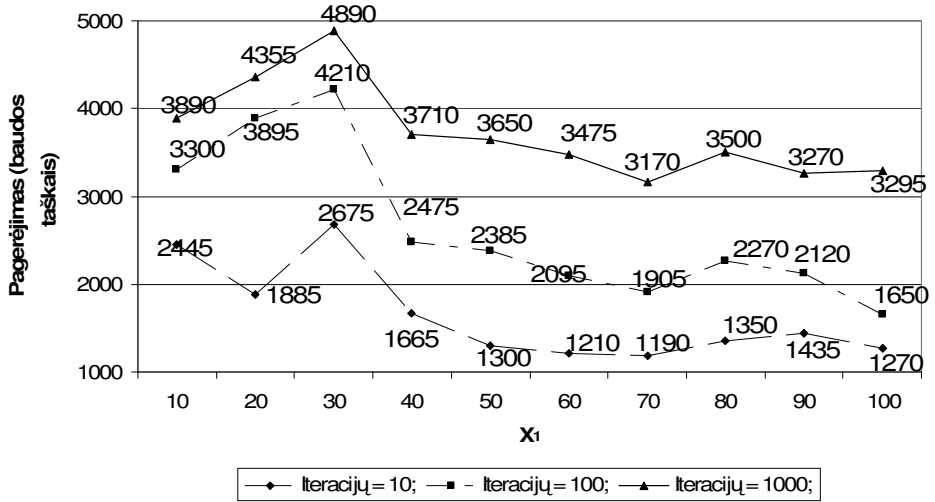
- Visus bandymus grupuojant pagal  $X_1$ , suformuoti rezultatai kito nevienodai. Tai matome visuose 2.8 ir 2.9 grafikuose.
- Metodo darbo laikas 21–345 sekundės.

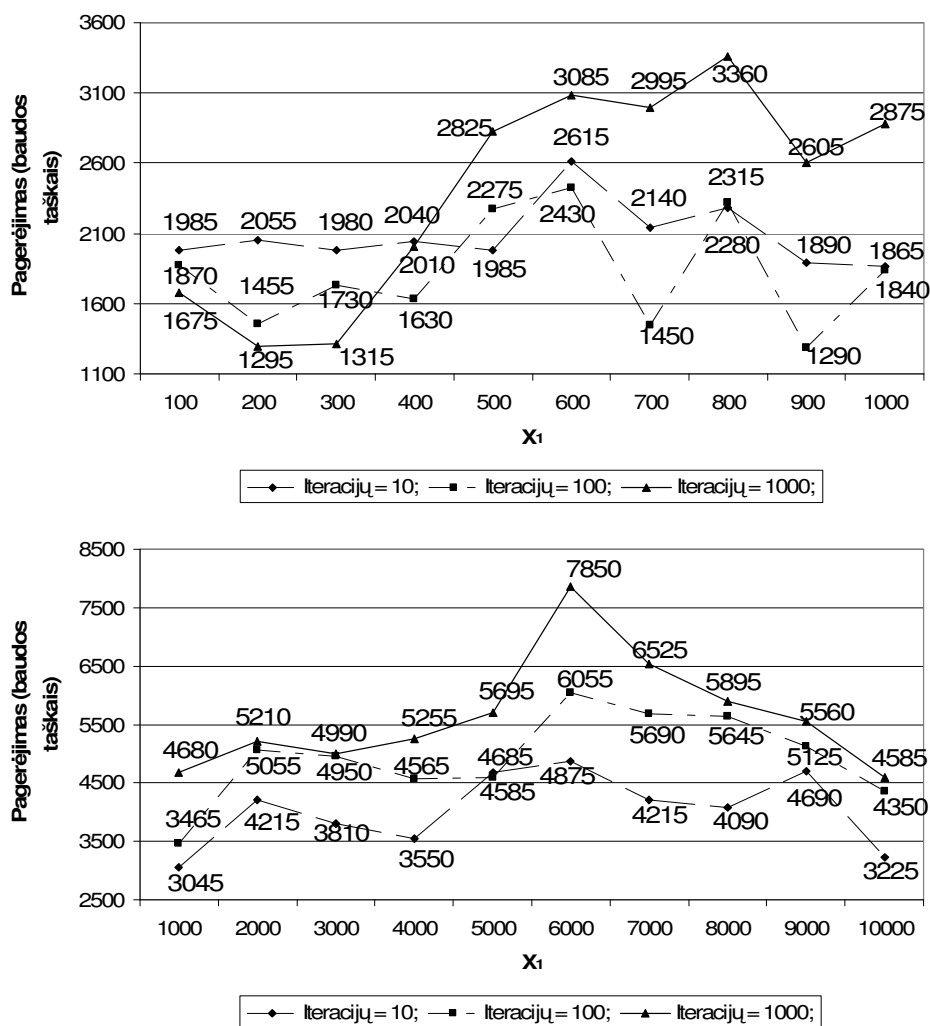
Kadangi rezultatus grupuojant pagal analizuojamą parametą  $X_2$  nepavyko nustatyti optimalių euristinių parametų, buvo nuspręsta euristinius parametrus analizuoti grupuojant pagal parametą  $X_1$ . Analizė atlikta vykdant po 100 bandymų juos grupuojant pagal  $X_1$ . T. y. buvo tiriama priklausomybė nuo  $X_1$  (čia  $1 < X_2 < 10$ ).





2.8 pav. Atkaitinimo modeliavimo metodu suformuoti pagerėjimų vidurkliai (pagal X<sub>1</sub>)





2.9 pav. Atkaitinimo modeliavimo metodu suformuoti geriausi rezultatai (pagal  $X_1$ )

#### Apibendrinimas:

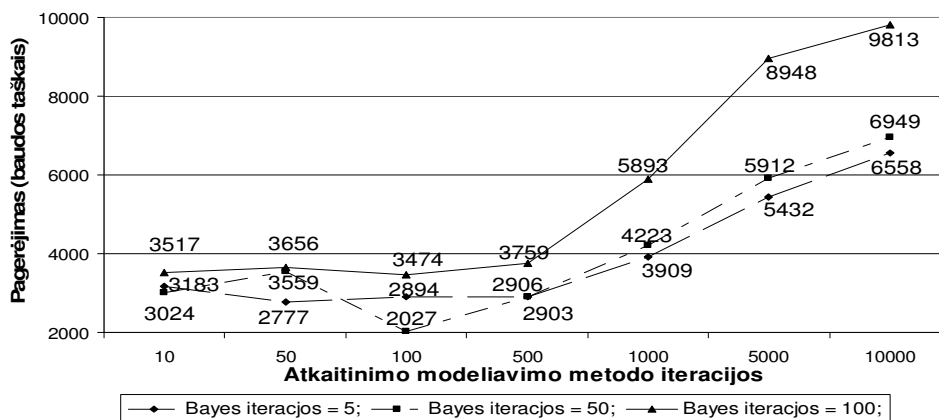
- Atliekant daug bandymų ir juos grupuojant pagal parametą  $X_2$  (2.8 paveikslas), rezultatų gerėjimas yra mažai pastebimas. Tai parodo 2.8 grafikuose esančios kreivės, kurios tai kyla, tai leidžiasi.
- Atliekant po 100 bandymų kiekvienu atveju (2.8 paveikslas), geriausi rezultatų vidurkiai gaunami:

- $X_1=10-100$ , nustačius 100 iteracijų.
- $X_1=100-1\ 000$ , nustačius 10 iteracijų.
- $X_1=1\ 000-10\ 000$ , nustačius 1 000 iteracijų.
- Geriausi vidutiniai rezultatai (2.8 paveikslas) buvo gauti kai  $X_2=1-10$ :
  - Nurodžius 10 iteracijų ir:
    - $X_1=30$ .
    - $X_1=600$  bei  $X_1=700$ .
    - $X_1=2\ 000$  ir  $X_1=7000$ .
  - Nurodžius 100 iteracijų ir:
    - $X_1=30$  bei  $X_1=70$ .
    - $X_1=600$  bei  $X_1=700$ .
    - $X_1=2\ 000$  bei  $X_1=6\ 000$ .
  - Nurodžius 1 000 iteracijų ir:
    - $X_1=50$ .
    - $X_1=900$ .
    - $X_1=6\ 000$ .
- Geriausi rezultatai, atlikus po 100 bandymų, visais atvejais suformuoti nustačius 1 000 iteracijų (2.9 paveikslas). Visų 2.7 paveiksle esančių grafikų kreivėse esantys rezultatai paprastai yra didesni už kitus.
- Analizuojant 2.9 paveiksle esančius grafikus, kai  $X_2=1-10$ , matome, kad visada geriausi rezultatai buvo suformuoti kai:
  - Nurodžius 10 iteracijų ir:
    - $X_1=30$ .
    - $X_1=600$ .
    - $X_1=6\ 000$ .
  - Nurodžius 100 iteracijų ir:
    - $X_1=30$ .
    - $X_1=600$  bei  $X_1=800$ .
    - $X_1=6\ 000$ .
  - Nurodžius 1 000 iteracijų ir:
    - $X_1=30$ .
    - $X_1=800$ .
    - $X_1=6\ 000$ .
- Visus bandymus grupuojant pagal  $X_2$ , suformuoti rezultatai kito nevienodai. Tai matome visuose 2.8 ir 2.9 grafikuose.
- Metodo darbo laikas 21–345 sekundės.

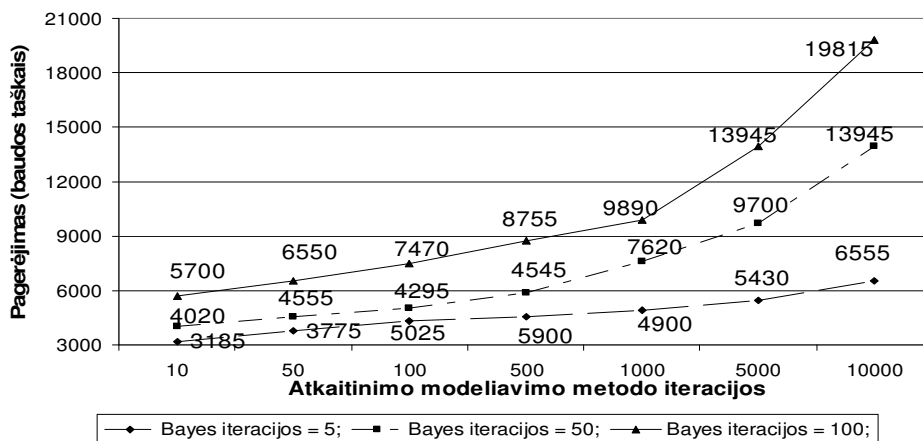
**Išvada:**

Nustatyti prie kokių AM  $X_1$  ir  $X_2$  parametrų rezultatų gerėjimas būtų pastovus – nepavyko. Kadangi AM metodą labai įtakoja šie parametrai, buvo nutarta AM metodą automatizuoti naudojant Bayes (BA) metodą.

**Bayes metodas (BA)** optimizuoja AM metodo parametrus  $X_1$  ir  $X_2$ . Šiame metode yra nustatomos  $X_1$  ir  $X_2$  parametrų ribos, AM iteracijos ir Bayes iteracijos. Visų šių parametrų įtaka rezultatui aprašoma „A priedo“ 5A lentelėje bei vaizduojama 2.10 ir 2.11 paveiksluose.



2.10 pav. Bayes metodu suformuoti pagerėjimų vidurkiai



2.11 pav. Bayes metodu suformuoti geriausi rezultatai

#### Apibendrinimas:

- Kuo daugiau iteracijų yra nustatoma, tuo tikslesnis rezultatas yra randamas. Tiksliausi rezultatai suformuojami kai AM iteracijų kiekis yra 10 000 ir BA iteracijų yra 100 (2.10 ir 2.11 paveikslai). Matome, kad abiejuose grafikuose rezultatai yra didžiausi.

- Nestabilus rezultatu gėrėjimas yra kai AM užduota 10–500, o BA 50 iteracijų. Grafike 2.10 šiame intervale matome ir pagėrėjimus, ir pablogėjimus.
- Optimalūs vidutiniai rezultatai visada tik gerėja kai AM iteracijų yra daugiau nei 500 (2.10 paveikslas). Grafike 2.10 kreivės nuolat auga.
- Geriausi rezultatai, suformuoti atlikus po 100 bandymų, visada gerėja (2.11 paveikslas). Jų kreivės, esančios 2.11 grafike, visada auga augant AM ir BA iteracijų skaičiui.
- Kuo daugiau pasirenkama AM ir BA iteracijų – tuo ilgiau reikia laukti rezultatų.
- Metodo darbo laikas nuo 2 minučių iki 10 valandų (darbo laikas labai priklauso nuo kompiuterio resursų arba interneto greičio).

**Išvada:**

Šiuo metodu suformuoti rezultatai yra tiksliausi lyginant su rezultatais, gautais naudojant anksčiau išvardintus metodus. Šio metodo rezultatai nuolat gerėja kai AM iteracijų yra daugiau nei 500. Automatizavus AM metodą optimalus rezultatas bus suformuojamas parinkus 10 000 AM ir 100 BA iteracijų.

### 2.3. Mokyklos euristikų analizės vertinimo rezultatai

Kiekviena profiliuota mokykla, formuodama pradinių duomenų failą gali nustatyti dalinius euristinius ribojimus:

- Disciplinų grupavimą.
- Disciplinų prioritetus.
- Moksleivių grupės dydį.
- Mokytojo mokomų klasių sąrašą.
- Mokytojui nepriimtinas darbo dienas.

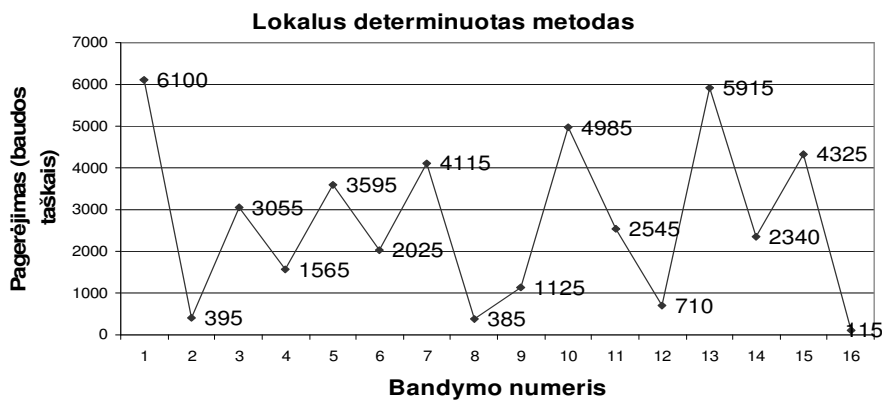
Šie ribojimai įtakoja tvarkaraščio formavimą. Kadangi jie suformuoti mokyklos nuostatuose ir taisyklėse, tai jie nustatomi pradinių duomenų faile. Norint juos pakeisti, reikia keisti vidaus nuostatus, derinti su individualiais žmonėmis. Tačiau mokykloje naudojami taip vadinami dinamiškieji euristiniai parametrai, kurie gali kisti. Dinamiški euristiniai mokymo įstaigos parametrai yra šie:

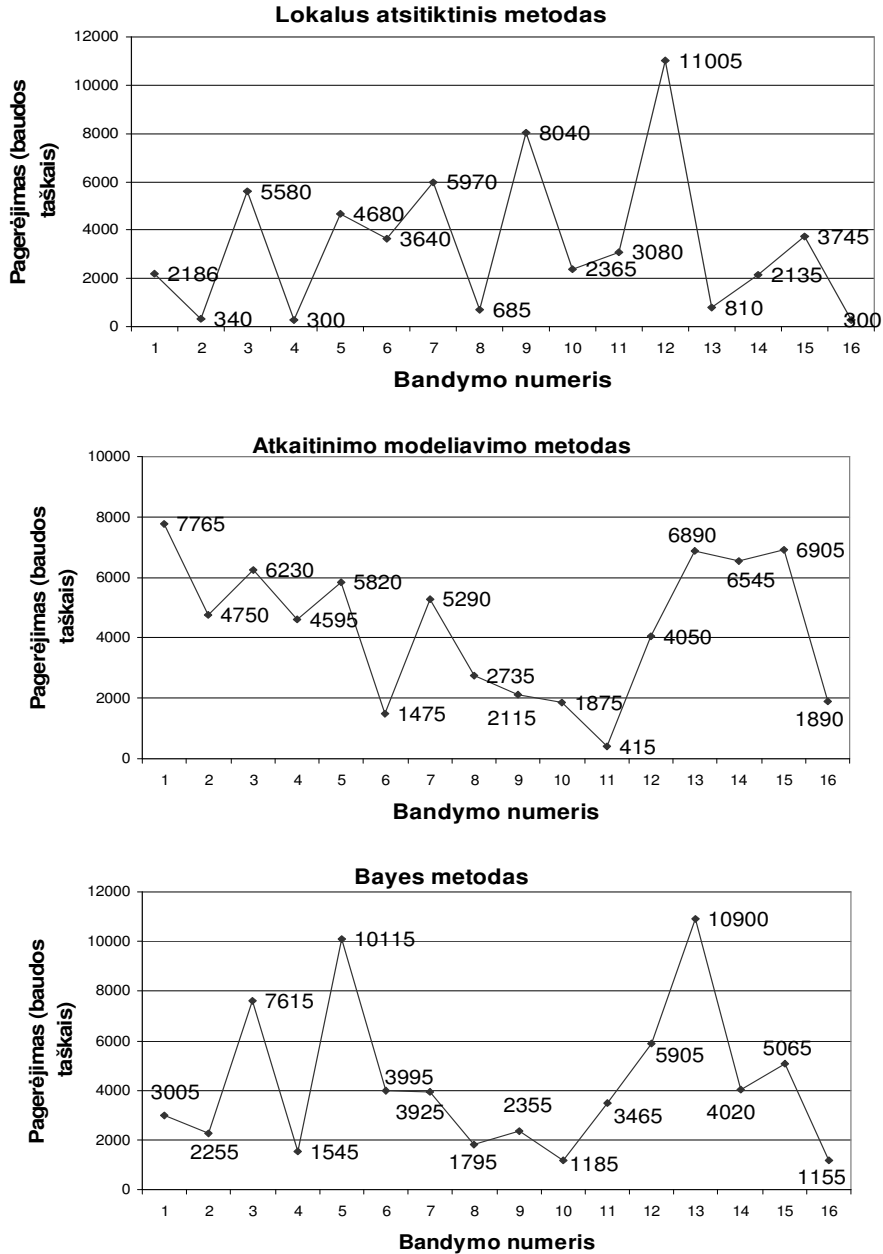
1. **Patalpų kiekis mokykloje** – laisvų patalpų skaičius, kuriose gali vykti pamokos moksleivių grupėms. Jis priklauso nuo mokyklos dydžio ir neužimtų patalpų skaičiaus.
2. **Akademinių valandų skaičius per dieną** – kiek maksimaliai akademinių valandų per dieną mokykloje gali vykti pamokos.

Kiekvienoje mokymo įstaigoje gali būti nustatytas skirtingas akademių valandų skaičius per dieną.

3. **Mokytojo tvarkaraštyje esančio lango įvertinimas** – nustatoma kiek bausdos taškų skirti jei mokytojo tvarkaraštyje yra langas. Kadangi mokytojų yra žymiai mažiau nei moksleivių, bausdos už jų tvarkaraštyje esančius langus yra žymiai didesnės.
4. **Moksleivio tvarkaraštyje esančio lango įvertinimas** – nustatoma, kiek bausdos taškų skirti jei moksleivio tvarkaraštyje yra langas.
5. **Mokytojo darbas per jam nepriimtinas darbo dienas** – jei mokytojas turi mažai akademių valandų jis nenori dirbti kažkurią darbo dieną / dienas. Tokiu atveju yra nustatomi bausdos taškai, kad mokytojui tą savaitės dieną / dienas paliktų laisvas.
6. **Bauda, jei viršijamas akademių valandų skaičius per dieną** – jei viršijamas šis skaičius, tai būna pats didžiausias pažeidimas sudarant tvarkaraštį. Todėl šiam pažeidimui įvertinti skiriama daugiausia bausdos taškų. Šis pažeidimas įvykdomas tik tada, kai neįmanoma išskirstyti pamokų į nurodytą akademių valandų skaičių per dieną.
7. **Pedagoginių didaktikų pažeidimas** – bauda skiriama už neteisingą pamokų skirstymą. Pedagoginės didaktikos nurodo, kurios disciplinos yra svarbiausios ir turi būti savaitės dienos sąrašo pradžioje ir kurios turi būti vėliau. Sąrašo pabaigoje įterpiamos mažiausiai svarbios ar nesudėtingos disciplinos.

Duomenys pateikti „A priedo“ 6A lentelėje ir pavaizduoti 2.12 paveiksle. Labiausiai susumuotus bausdos taškus įtakoja parametrai, kurių numeriai yra 1, 2, 3 ir 7. Todėl buvo nuspręsta atlikti tyrimą nustatant skirtingus šiuos euristicinius parametrus. Atlikus tyrimą buvo gauti rezultatai, aprašyti „A priedo“ 6A lentelėje bei pavaizduoti 2.12 paveiksle.





2.12 pav. Metodų priklausomybė nuo euristinių mokyklos parametru



Atliekant tyrimą optimizavimo metodų parametrai buvo nekeičiami. Buvo keičiami tik dinamiški, tvarkaraštį įtakojantys, mokymo įstaigos parametrai.

**Gautų rezultatų apibendrinimas:**

- Pedagoginėms didaktikoms baudos taškų nustatyta daugiau nei vidutiniškai. Tvarkaraštį suformuoti ir optimizuoti sunkiau. Kuo baudos pedagoginėms didaktikoms bus didesnės, tuo tiksliau vidinius mokyklos nuostatus atitiks suformuotos pamokų sekos, tuo mažiau galimybių pakeisti pamokos vietą tvarkaraštyje.
- Moksleivių tvarkaraštyje esančių langų įvertinimas artimas mokytojų tvarkaraštyje esančių langų įvertinimui. Kiekvieno moksleivio tvarkaraštyje bus žymiai mažiau langų nei mokytojo tvarkaraštyje. Toks tvarkaraštis yra artimas Lietuvos Respublikos Švietimo ir mokslo ministerijos nuostatuose suformuotam profiliuotos mokyklos tvarkaraščiui. Tačiau tokio suformuoto tvarkaraščio trūkumas yra tas, kad moksleivių grupės yra nuolat besikeičiančios kiekvienos disciplinos pamokų metu. Pagrindinis skirtumas nuo tikrojo mokykloje naudojamo tvarkaraščio yra tas, kad moksleivių grupės sudėtis nepastovi. Moksleivių grupės sudėtis gali keistis tik keičiantis disciplinai. Šiuo atveju moksleivių grupės sudėtis kinta nuolat.
- Mokytojų tvarkaraštyje esančių langų įvertinimas proporcingas moksleivių tvarkaraštyje esančių langų įvertinimui. Suformuotame tvarkaraštyje langų mokytojams yra žymiai mažiau nei moksleiviams. Tvarkaraštyje esantys langai eliminuojami mokytojo pamoką perkeltiant į galimą tvarkaraštyje esantį langą. Taip eliminuojant mokytojo tvarkaraštyje esantį langą, moksleivių tvarkaraštyje gali susidaryti langai. Tačiau taip tvarkaraštyje eliminuojant langus moksleivių grupės sudėtis išlieka stabili.
- Patalpų, kuriose gali vykti pamokos, kiekis yra labai mažas, o akademinė valandų skaičius per dieną yra lygus 7. Tvarkaraštis sudaromas žymiai dažniau keičiant pamokų vietą tvarkaraštyje ir moksleivių grupės sudėtį. Šiuo atveju mokytojo tvarkaraštyje yra mažiau langų nepaisant, ar mokytojo tvarkaraštyje esančio lango įvertis proporcingas ar artimas moksleivio tvarkaraštyje esančio lango įverčiui. Moksleivių tvarkaraštyje bus daug langų, jei mokytojo tvarkaraštyje esantis langas įvertinamas proporcingai moksleivio tvarkaraštyje esančiam langui. Moksleivių grupės sudėtis nuolatos kinta. Suformuotas moksleivių tvarkaraštis labiau komplikuoatas ir labiau neatitinkantis Lietuvos Respublikos Švietimo ir mokslo ministerijos nuostatų.

- Patalpų, kuriose gali vykti pamokos, kiekis yra labai didelis, o akademinė valandų skaičius per dieną yra lygus 4. Suformuotas tvarkaraštis labai artimas profiliuotos mokyklos tvarkaraščiui. Paprastai mokykloje tvarkaraštis yra kuriamas pradžioje formuojant profiliuotos mokyklos tvarkaraštį. Taigi visos mokykloje esančios patalpos, kuriose gali vykti pamokos, yra neužimtos. formuojant tvarkaraštį pamokos skirtingiems mokytojams gali vykti vienu metu (vienoje paralelėje). Taigi moksleivių grupė išlieka stabili.

## 2.4. Rekomendacijos optimizavimo metodams vertinti

Ištyrus disertacijoje minimus optimizavimo metodus galima daryti šių metodų optimalių parametrų parinkimo vertinimus (2.3 lentelė):

2.3 lentelė. Metodų analizė tarpusavyje palyginant jų gautus rezultatus

	Rezultatų efektyvumas	Geriausi rezultatai	Tikslumas	Darbo greitis
LD	Dažnai „užstringa“ lokaliai minimume.	Nurodžius 1 000–10 000 Iteracijų.	Netikslus. Langų eliminavimo eilės tvarka yra fiksuota. Dažnai „užstringa“ lokaliai minimume	Darbo laikas nuo 7 iki 45 sekundžių.
LA	Suformuoti rezultatai tikslesni nei LD.	Nurodžius 1 000–10 000 iteracijų ir $K = 5$ .	Tikslesnis už LD. LD papildytas langų eliminavimo eilės tvarkos atsitiktiniu generavimu.	
AM	Rezultatai priklauso nuo parametrų parinkimo.	Parametrai nenusatatyti. Rezultatai labai kinta keičiantis parametrų dydžiams.	Tikslesnis už LA [vertinama pablogėjimo tikimybė. Langų eliminavimo eilės tvarka generuojama atsitiktinai.	Darbo laikas nuo 21 sekundės iki 10 valandų
BA	Suformuotas tvarkaraštis artimas realiam mokyklos tvarkaraščiui.	Nurodžius daug BA ir AM iteracijų. Rezultatai yra tiksliausi kai AM iteracijų yra daugiau nei 500.	Tikslesnis už AM. Tai AM paieškos automatizavimas. (Globalaus minimumo paieška atsitiktinai generuojant pradinius duomenis).	

Tyrimas buvo atliktas naudojant tokį pradinį duomenų failą, kuriame yra 340 moksleivių, 66 mokytojai. Mokymo įstaigoje mokoma 68 skirtingos disciplinos.

**Apibendrinimas:**

- Greičiausiai tvarkaraštį suformuoja ir optimizuoja lokalieji metodai.
- Lokalus atsitiktinis metodas yra tikslesnis už lokalų determinuotą.
- Atkaitinimo modeliavimo metodas naudoja tikimybę, bei atsitiktinumus, todėl šie parametrai labai įtakoja jo rezultatus. Norint gauti tikslesnius rezultatus reikia atlikti daugybę bandymų.
- Atkaitinimo modeliavimo metodo automatizavimas naudojant Bayes metodą leidžia suformuoti tikslesnius rezultatus nei paprastas atkaitinimo modeliavimo metodas.
- Kuo daugiau iteracijų nurodoma Bayes metode, tuo tikslesnis tvarkaraštis bus suformuotas.

## 2.5. Lietuvos mokyklose naudojamų programų lyginamoji tyrimas

Profiliuotų mokyklų tvarkaraščiams formuoti specialiai pritaikytos kompiuterinės programos nėra. Lietuvos Respublikos mokyklose tvarkaraščių sformuotojai dirba rankiniu būdu arba pasitelkdami komercines, tačiau tokių tvarkaraščių formavimui visiškai nepritaikytas programas. Komercinės programos puikiai formuoja pagrindinės mokyklos tvarkaraštį, tačiau, kaip parodė tyrimai, nesuformuoja profiliuotos mokyklos tvarkaraščio. Lietuvos Respublikos mokyklose dažniausia naudojamos tokios komercinės programos:

- „Rector“.
- „aSc Timetables 2008“.
- „Mimosa“.

Taip pat išanalizuotos profiliuotos mokyklos tvarkaraščius kuriančios nekomercinės, bei Lietuvos Respublikos mokyklose dar nenaudojamos, programos:

- „School – Complete“.
- „School Schedule optimization program by Auris“.
- „School Schedule optimization program by Vidunas“.

### 2.5.1. Programos „Rector“ tyrimas

Kompiuterinę programą „Rector“ sukūrė Rusijos IT kompanija „P. Yu. Smykalov“ (Rector 2008). Tai pati paprasčiausia ir elementariausia tvarkaraščių

formavimo programa. Autoriai reklaminiame buklete rašo, kad programa dirba automatiškai, tačiau šios funkcijos aptikti nepavyko. Viskas „Rector“ programoje paremta žmogiškaisiais resursais: tiek duomenų įvedimas, tiek kriterijų nustatymai, tiek optimalaus tvarkaraščio formavimas. Programa darbo pobūdžiu panaši į MS Excel. Ji turi mažai programinių funkcijų. Skiriasi nuo naujai atverto Excel lango tuo, kad formuojant tvarkaraštį rodo perspėjimus, kur duomenis įterpti galima ir kur ne. Tai šiek tiek geriau nei Excel programoj. Tačiau pasitelkus „Macro“ komandas Excel programoje, tai būtų nesunkiai išsprendžiama problema. Optimizavimą vykdo pats vartotojas rankiniu būdu.

Programa „Rector“ skirta tik pagrindinės mokyklos tvarkaraščiui formuoti. Norint šią programą taikyti profiliuotoje mokykloje iškyla daug sunkumų. Pagrindiniai sunkumai būtų tokie:

- Sąryšių tarp moksleivių ir mokytojų nurodymas.
- Moksleivių individualių reikalavimų nustatymas.
- Mokytojų individualių reikalavimų nustatymas.

Darbą su šia programa galima prilyginti darbui, rankiniu būdu ant stalo dėliojant lapelius su duomenimis. Apskaičiuoti baudos taškus ar kitaip įvertinti programos darbą nepavyko.

### 2.5.2. Programos „aSc TimeTables 2008“ tyrimas

Šią kompiuterinę programą sukūrė Slovakijos IT kompanija „Applied Software Consultants s.r.o.“ (aSc TimeTables 2008). Programa yra naudojama kai kuriose Lietuvos Respublikos mokyklose. Ši programa taip pat skirta suformuoti tvarkaraštį pagrindinei mokyklai. Pasak kūrėjų, tai vienintelė programa, išversta į lietuvių kalbą ir sugebanti sukurti visos mokyklos tvarkaraštį. Taip pat, pasak kūrėjų, programa įterpia absoliučiai visas reikiamas pamokas į tvarkaraštį, jei tik tai tam yra galimybė. Patikrina tūkstančius įvairiausių galimų variantų. Tvarkaraščiai gali būti kiek vienasavaitiniai, tiek daugiasavaitiniai.

Reklaminiame buklete apie programą „aSc TimeTables 2008“ rašoma, kad jos pagalba galima:

- Įrašyti ir išsaugoti pagrindinius duomenis.
- Aprašyti įvairias sąlygas.
- Greitai sukurti tvarkaraštį.
- Atnaujinti tvarkaraštį bet kuriuo mokslo metų metu.
- Atspausdinti kiekvienos klasės, patalpos, mokytojo ar bendrą mokyklos tvarkaraštį.
- Taip pat tvarkaraščius galima eksportuoti į Excel lenteles ar „html“ formatą, patalpinti internete.

Visi šie privalumai labai efektyvūs pagrindinės mokyklos tvarkaraštyje. Tačiau ši programa yra visiškai netinkama bandant sukurti profiliuotos mokyklos tvarkaraštį. Analizuojant programos veikimo principus susiduriama su tokiomis esminėmis tvarkaraščio kūrimo problemomis:

- Nėra galimybės suskirstyti klasę į daugiau nei 4 pogrupius.
- Sunku kaskart nurodyti suformuotai grupei visus resursus (mokytojus, disciplinas, pamokų per savaitę skaičių ir patalpas).
- Labai sunku grupuoti moksleivius pagal disciplinos lygį ir pamokų per savaitę skaičių.
- Įvedus 96 moksleivius programos darbo laikas labai pailgėja.
- Neranda nė vieno optimalaus sprendimo vidutinio dydžio profiliuotai mokyklai.
- Likusias disciplinas, kurių programa nesugeba išskirstyti tvarkaraštyje, reikia įterpti rankiniu būdu.

2.4 lentelė. Darbo su programą „aSc TimeTables 2008“ analizė

	Darbo laikas	Peržiūrėti variantai	Neįvykdytos sąlygos	Likusios disciplinos	Kūrinio sudėtingumas	Apribojimai
2 klasės 47 moksleiviai	00:07:40	141200	0	27	Mažas	„Švelnus“
	00:10:58	182146	20	26	Vidutinis	„Švelnus“
	01:54:05	1624528	33	24	Didelis	„Švelnus“
	00:35:29	1353174	0	29	Mažas	„Griežtas“
	05:19:07	4895895	16	26	Vidutinis	„Griežtas“
	63:52:36	68675510	26	27	Didelis	„Griežtas“
4 klasės 106 moksleiviai	28:53:21	955869	38	45	Mažas	„Švelnus“
	56:46:33	4525965	42	39	Vidutinis	„Švelnus“
	nedirbo	---	---	---	Didelis	„Švelnus“
	85:37:40	352951459	63	55	Mažas	„Griežtas“
	nedirbo	---	---	---	Vidutinis	„Griežtas“
	nedirbo	---	---	---	Didelis	„Griežtas“
13 klasių 315 moksleivių	nedirbo	---	---	---	Mažas	„Švelnus“
	nedirbo	---	---	---	Vidutinis	„Švelnus“
	nedirbo	---	---	---	Didelis	„Švelnus“
	nedirbo	---	---	---	Mažas	„Griežtas“
	nedirbo	---	---	---	Vidutinis	„Griežtas“
	nedirbo	---	---	---	Didelis	„Griežtas“

Kadangi profiliuotos mokyklos moksleivio tvarkaraštis yra individualus, tai į šią programą „klasių“ vietoje kiekvieno moksleivio duomenys buvo įrašyti atskirai. Mokytojai, mokomos disciplinos, jų lygiai bei valandos ir patalpos taip pat buvo nurodyti kiekvienam moksleiviui individualiai. Kiekvieną moksleivių grupę teko formuoti rankiniu būdu. Sudarant tokias moksleivių grupes, skaičiuojant rankomis moksleivių kiekį grupėje, buvo nurodoma, kokie moksleiviai sudaro grupę. Taip pat buvo atsižvelgta, kad moksleivių skaičius grupėje turi būti skirtingas, priklausomai kokiai disciplinai formuojama grupė. Taigi įrašius duomenis į programą suformuoti rezultatai pateikti 2.4 lentelėje. Vidutinėje profiliuotoje mokykloje apytiksliai yra 13 klasių, kur kiekvienoje klasėje apytiksliai mokosi 20 – 30 moksleivių.

#### **Išvados:**

- Programa nesuformuoja profiliuotos mokyklos pamokų tvarkaraščio. Taigi galime daryti išvadą, kad programa visiškai netinka profiliuotos mokyklos tvarkaraščiui formuoti.
- Nors moksleivių duomenų programoje nėra labai daug (96 moksleiviai), programos darbo laikas yra labai ilgas (daugiau nei 85 valandos).
- Įrašius į programą visų profiliuotos mokyklos moksleivių duomenis ir juos suskirsčius rankiniu būdu, programos darbas tampa neįmanomas. Ji nesuformuoja jokio pamokų tvarkaraščio. Parodomas pranešimas apie nesėkmingus bandymus formuoti tvarkaraštį ir siūloma kreiptis į gamintoją.
- Dirbant su šia programa mokyklos pamokų tvarkaraščio formavimą reikėjo užbaigti rankiniu būdu eliminuojant tvarkaraštyje esančius langus arba formuoti visą tvarkaraštį rankiniu būdu.

### **2.5.3. Programos „Mimosa“ tyrimas**

Šią kompiuterinę programą sukūrė Suomijos IT kompanija „Mimosa Software Ltd.“ (Mimosa 2008). Programa yra naudojama kai kuriose Lietuvos Respublikos mokyklose. Ji panaši į anksčiau nagrinėtąją „aSc TimeTables 2008“ programą. Kaip ir „aSc TimeTables 2008“ programa, „Mimosa“ tvarkaraštį formuoja tik pagrindinei mokyklai. Šis produktas tik palengvina realių tvarkaraščių formavimą. Leidžia juos koreguoti bei tobulinti. Padeda eliminuoti tvarkaraštyje esančius langus. Praneša apie ribojimų neatitikimus. Tačiau daugiau nei 50% sukurto tvarkaraščio kokybės priklauso nuo paties kūrėjo. Dėl programoje esančių lentelių, kurias reikia pildyti, pavadinimų vienodumo, šia programa yra sudėtinga naudotis. Ji yra kaip pagalbiniis produktas formuojant mokyklos tvarkaraštį.

Pasak programos kūrėjų šios programos galimybės didelės. Čia yra leidžiama:

- Moksleivių klases skirstyti į pageidaujamą pograpių skaičių.
- Apjungti kelių klasių ar jų grupių moksleivius per atskiras pamokas.
- Tos pačios klasės moksleiviams nurodyti kelis mokytojus dėstančius tą pačią discipliną (pavyzdžiui, kai discipliną gali mokintis tik 20 moksleivių, o klasėje ją nori mokintis 30 moksleivių).
- Suformuoti atskirus pamokų tvarkaraščius kiekvienam moksleiviui.
- Nustatyti tvarkaraščio formavimo ribojimus mokytojams ir moksleiviams.
- Nurodyti užsiėmimų svarbą tvarkaraštyje.
- Nurodyti ar automatiškai suskirstyti visas disciplinas, ar tik dalį disciplinų, likusias įterpiančias po vieną rankiniu būdu.
- Atspausdinti individualius disciplinų tvarkaraščius mokytojams, moksleiviams ar klasėms.
- Atspausdinti patalpų užimtumo tvarkaraščius ir / ar padaryti internetinę tvarkaraščių versiją.

Visi šie privalumai naudingi tiek pagrindinėms, tiek ir profiliuotos mokyklos tvarkaraštyje. Analizuojant programos veikimo principus buvo iškilę tokios esminės tvarkaraščio kūrimo problemos:

- Pradinių duomenų įrašymas į programą yra sudėtingas, nes programoje kelių pildomų lentelių pavadinimai vienodi.
- Įrašant pradinius duomenis nėra realizuoto intuityvaus duomenų įrašymo į programą būdo. (nėra aišku kas ir kur turi būti įrašyta).
- Sunku kaskart nurodyti tvarkaraščio formavimo ribojimus, nes nėra aiškiai nurodyta, kurioje lentelėje koks ribojimas turi būti nustatytas.
- Painiai formuojamas tvarkaraštis bandant optimizuoti pradinius duomenis.
- Programa darbo metu nesuformavo nei vieno optimalaus sprendimo profiliuotai mokyklai.
- Likusias disciplinas, kurių programa neįterpė į tvarkaraštį, reikėjo įterpti rankiniu būdu.

#### **2.5.4. Programos „School – Complete“ tyrimas**

Tai – pati pirmoji lietuviška tvarkaraščių formavimo kompiuterinė programa, kurią sukūrė Lietuvos Respublikos aukštosiose mokyklose studijuojantys studentai, (School-COMLETE 2003). Ji sukurta pagrindinėms mokykloms. Išanalizavus šią, studentų (magistrantų) sukurtą mokyklos tvarkaraščio programą, buvo pastebėta, kad stengiamasi tik techniškai įterpti

visas pamokas į tvarkaraštį, siekiant eliminuoti tvarkaraštyje esančius mokytojų ir moksleivių langus. Dėl prasto rezultatų pateikimo ir sudėtingo pradinių duomenų failo sudarymo ši programa nevertinama Lietuvos Respublikos mokyklose. Ji nebuvo taikoma mokyklose, nes reikia kruopštaus darbo formuojant pradinių duomenų failą. Jei tvarkaraštis netinka arba moksleivis pakeičia savo reikalavimus, tuo pakeitimus sunku įrašyti į pradinių duomenų failą. Čia nėra vertinami individualūs mokyklos taisyklėmis nustatyti euristiniai ribojimai. Ši programa turi tokius trūkumus:

- Programoje nėra įvertintos pedagoginės didaktikos.
- Duomenų failo pavadinimas turi būti konkretus. Tai yra pradinių duomenų failo pavadinimas turi būti „mokytojai.txt“. Kitokiu pavadinimu failo programa nenuskaito.
- Pradinis duomenų failas turi būti kuriamas tik „txt“ formatu.
- Pradinių duomenų failo formavimas yra nepatogus ir komplikuoatas.
- Nėra vertinami mokytojų reikalavimai.
- Dirbant nutolusiame kompiuteryje piko valandomis programa dirba lėtai, o tai sukelia daug nepatogumų.

### **2.5.5. Programos „School Schedule optimization program by Auris“ tyrimas**

Patobulintoji „School – Complete“ kompiuterinė programa pavadinta „School Schedule optimization program by Auris“ (School Schedule optimization by Auris 2005) vardu. Tai – paskutinė programos versija naudojanti „csv“ formato pradinių duomenų failus. Programa turi keletą optimizavimo metodų. Tvarkaraščio gerumą įvertina pagal nustatytuosius baudos taškus (apie tvarkaraščio vertinimą baudos taškais plačiau kalbama 2.2.2. skyriuje). Baudos taškai yra nustatomi įvertinant disciplinų svarbą, įvertinant moksleivių bei mokytojų tvarkaraštyje esančius langus, bei įvertinant kitus tvarkaraščio kokybę įtakojančius parametrus. Programa turi keletą trūkumų:

- Programoje nėra įvertintos pedagoginės didaktikos.
- Nėra nustatytas moksleivių grupės atskirai disciplinai dydis.
- Pradinis duomenų failas turi būti kuriamas „csv“ formatu.
- Pradinių duomenų failo formavimas yra komplikuoatas.
- Programa sukurta tik pagrindinių mokyklų tvarkaraščiams formuoti.
- Neįmanoma nustatyti akademinų valandų skaičiaus per dieną (ji pakeičiant dingsta vėlesnių valandų metu vykstančios pamokos).
- Dirbant nutolusiame kompiuteryje piko valandomis programa dirba lėtai, o tai sukelia daug nepatogumų.



- Rezultatai rodomi tik interneto lange. Norint išsaugoti visus rezultatus reikia saugoti kiekvieno mokytojo ar moksleivio tvarkaraštį atskirai.

### 2.5.6. Programos „School Schedule optimization program by Vidunas“ tyrimas

Tai „School Schedule optimization program by Auris“ modifikuota kompiuterinė programa, kurioje yra pritaikytas genetinis algoritmas (School Schedule optimization by Vidunas, 2005). Darbas su šia programa žymiai sudėtingesnis, nei su prieš tai nagrinėta. Pradinių duomenų failo formatas yra toks pats kaip ir prieš tai nagrinėtosios programos. Įskaitant 2.5.5 skyriuje nagrinėtos programos trūkumus, ši programa turi papildomų trūkumų:

- Programa skirta ne mokyklos tvarkaraščiams formuoti, o genetinių algoritmų tyrimams atlikti.
- Pakeitus pradinį programos nustatymus, programa dažnai rodo klaidas, kurių prasmė nėra aiški.
- Pradinių duomenų failas sudaromas painiai. Paprastam vartotojui sunku susigaudyti.
- Parametrai, kuriuos reikia nustatyti optimizavimo metodo darbo metu, nepaiškinami ir sunkiai interpretuojami. Šių parametru yra labai daug.
- Programoje nėra intuityvios vartotojo sąsajos.

### 2.5.7. Nagrinėtų programų palyginimas tarpusavyje

2.5.1–2.5.6 skyriuose išanalizuotos populiariosios komercinės ir nekomercinės programos, apibendrintos 2.5 lentelėje.

2.5 lentelė. Analizuojamų programų palyginimo apibendrinimas

Pavadinimas	Programos veikimas	Programos tikslas	Pradiniai duomenys	OS kurioje programa dirba
„School – Complete“	Skirta pagrindinės mokyklos tvarkaraščiui formuoti. Įdiegtas lokalus optimizavimo metodas.	Tik įvykdyti visus duomenų faile nurodytus reikalavimus	Iš anksto įrašomi į duomenų failą. Windows 2000 / XP / Vista.	Linux.

2.5 lentelės pabaiga

Pavadinimas	Programos veikimas	Programos tikslas	Pradiniai duomenys	OS kurioje programa dirba
„School schedule optimization program“	Skirta profiliuotos mokyklos tvarkaraščiui formuoti. Įdiegti keli optimizavimo metodai.	Formuoti optimalų tvarkaraštį atsižvelgiant į visus vartotojo pradiniam faile		
„aSc TimeTables 2008“	Skirta pagrindinės mokyklos tvarkaraščiui formuoti. Įdiegtas lokalus optimizavimo metodas.	ir programos darbo metu nustatytus reikalavimus.	Įrašomi programos vykdymo metu arba importuojami iš Excel failo.	Windows 2000 / XP / Vista.
„MIMOSA“		Tik vykdyti visus vartotojo pradiniam faile nustatytus reikalavimus.		
„RECTOR“	Palengvina disciplinų skirstymą rankiniu būdu. Neautomatizuotas.			

**Išvados:**

- Komerčinės programos nėra pritaikytos profiliuotų mokyklų tvarkaraščiams formuoti bei optimizuoti. Nei viena nesuformavo profiliuotos mokyklos tvarkaraščio.
- Tiek komercinėse, tiek studentų sukurtose programose yra komplikotas pradinių duomenų įrašymas.
- Komerčinės programos priklausomos nuo operacinės sistemos. Linux operacinėje sistemoje jos nedirba.
- Komerčinėse programose nėra efektyvių optimizavimo metodų. Šie metodai greitai „užstringa“ lokaliai minimume arba išvis nesuformuoja tvarkaraščio.
- Dirbant su komercinėmis programomis dažniausia darbą reikia užbaigti vartotojui rankiniu būdu eliminuojant tvarkaraštyje esančius langus.
- Studentų sukurtos programos nėra pritaikytos profiliuotai mokyklai.

## 2.6. Antrojo skyriaus išvados

1. Vektorinio optimizavimo teorijos skaliarizacijos metodo taikymas, formuluojant profiliuotos mokyklos tvarkaraščių optimizavimo uždavinius, leidžia įvertinti individualius mokymo įstaigos euristicinius kriterijus užtikrinant Pareto optimalumą. Sukurtas tvarkaraštis yra artimas realiam mokymo įstaigos tvarkaraščiui.
2. Atkaitinimo modeliavimo parametrų optimizavimas naudojant Bayes metodus yra efektyvus būdas padidinti sprendimo tikslumą, nes:
  - a. Patikrinama žymiai daugiau variantų.
  - b. Per tą patį laiką randa tikslesnį sprendinį nei paprastas atkaitinimo modeliavimo metodas.
3. Nurodytos euristikos parametrai skirtingai įtakoja optimizavimo metodo sprendimo greitį ir tikslumą. Tiksliausias sprendinys gaunamas kombinuotu atkaitinimo modeliavimo ir Bayes metodu.
4. Lietuvos mokyklose naudojamos kompiuterinės programos nėra pritaikytos profiliuotų mokyklų tvarkaraščiams optimizuoti. Todėl darbas yra labai aktualus.

# 3

---

## Optimizavimo programos kūrimo etapai

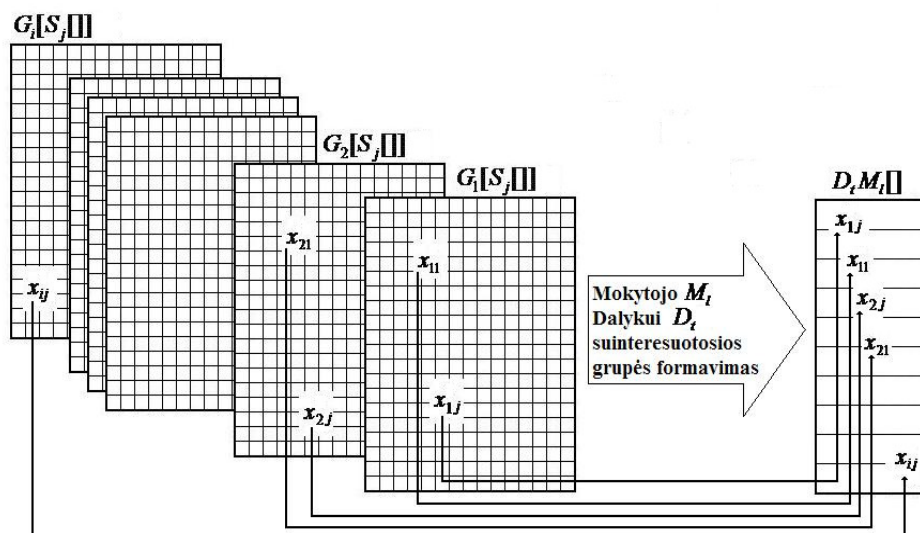
Šiame skyriuje, remiantis 2 skyriaus analize ir išvadomis, pateiktas lanksčios profiliuotos mokyklos tvarkaraščio optimizavimo programos projektavimas. Pateiktas pradinio duomenų failo projektavimas, kuris yra informatyvus ir įvertina daugelį mokyklos nuostatuose apibrėžtų euristinių parametrų. Suprojektuotas lankstus rezultatų išsaugojimas. Suformuluoti siūlymai programos plėtojimui.

Skyriaus tematika paskelbti trys autorės straipsniai (Pupeikienė *et al.* 2005), (Pupeikienė *et al.* 2006), (Pupeikienė *et al.* 2009).

### 3.1. Dabartinės būsenos tyrimas

Daugelyje profiliuotų mokyklų pradinis tvarkaraštis yra sudaromas rankomis, mechaniškai skirstant moksleivių į moksleivių grupes (2.2 pav.). Šiuo atveju visa tai atlieka kompiuteris. Jis užpildo 2.1.1 skyriuje aprašytą matricą (2.2), pradžioje užpildydamas ją, pradiniame faile esančia informacija, tokiu būdu:

- Iš grupės  $G$  įrašoma moksleivio  $S$ , nurodžiusio discipliną  $D$ , informacija.
- Į moksleivių grupę jungiami moksleiviai, nurodę identišką discipliną (tiek pavadinimu, tiek valandų skaičiumi, tiek lygiu).
- Suformavus moksleivių grupę vienai disciplinai, pradeda formuoti naują moksleivių grupę kitai disciplinai.
- Jei nėra maksimalaus disciplinai dėstyti leidžiamo moksleivių skaičiaus, moksleivių grupė suformuojama iš esamo moksleivių skaičiaus.
- Suformuota moksleivių grupė susiejama ją mokančiu mokytoju  $M$ .

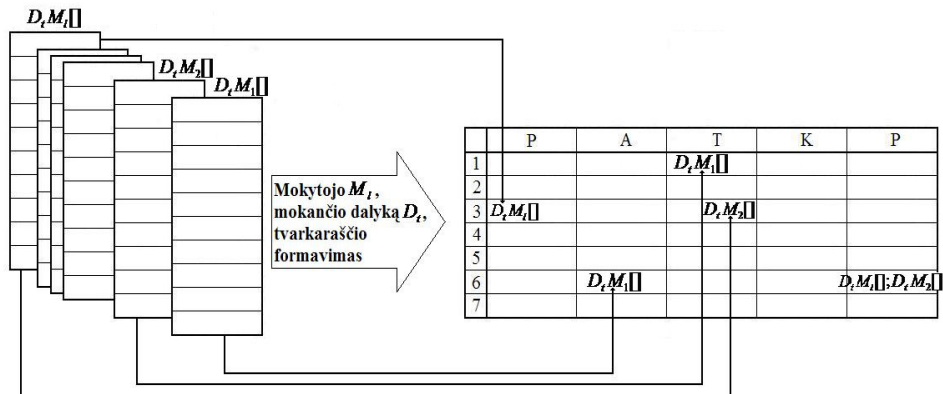


3.1 pav. Moksleivių grupės formavimas

Moksleivių grupės darbo vieta ir laikas yra parenkamas iš duotojo sąrašo. Šie duomenys gali kisti tvarkaraštyje kaskart sumuojant baudos taškus suformuotam tvarkaraščiui.

Sumuojant baudos taškus mokyklos tvarkaraštis skirstomas po savaitės dienas ir valandas (3.1 pav.), formuojant tvarkaraščius jau yra įvertinamos būtinų ribojimų euristicos. Šį darbą gali atlikti integruoti vadinamieji „godieji“ algoritmai. Pagrindinė „godžiųjų“ algoritmų problema yra ta, kad jie pradeda formuoti tvarkaraštį pirmam sąraše esančiam mokytojui ir baigia paskutiniam. Taigi pirmajam mokytojui yra suformuojamas puikus tvarkaraštis, o paskutiniajam – labai blogas. Norint išvengti tokio tvarkaraščių formavimo reikia nustatyti papildomus baudos taškus. Optimalus tvarkaraštis formuojamas

nustatant būtinų ir pakankamų ribojimų baudos taškus. Tačiau norint suformuoti specifinį tvarkaraštį, kuris būtų optimalus visiems, reikia pakoreguoti disciplinų prioritetus. Kuo didesni prioritetai nustatomi disciplinai, tuo tvarkaraštis bus formuojamas ne tik pagal būtinus bei pakankamus ribojimus, bet ir pagal nustatytų prioritetų dydį. Prioritetas čia suprantamas kaip pirmenybė formuoti tvarkaraštį pagal disciplinų svarbą tvarkaraštyje. Tai yra pagal tą discipliną, kuri yra svarbiausia. Taip sumuojamas mažesnę baudos taškų sumą turintis tvarkaraštis, atitinkantis mokyklos vidaus taisykles. Šiuo atveju disciplinų seka neprieštarauja Lietuvos Respublikos Švietimo ir mokslo ministerijos nuostatom.



3.2 pav. Profiliuotos mokyklos tvarkaraščio formavimas

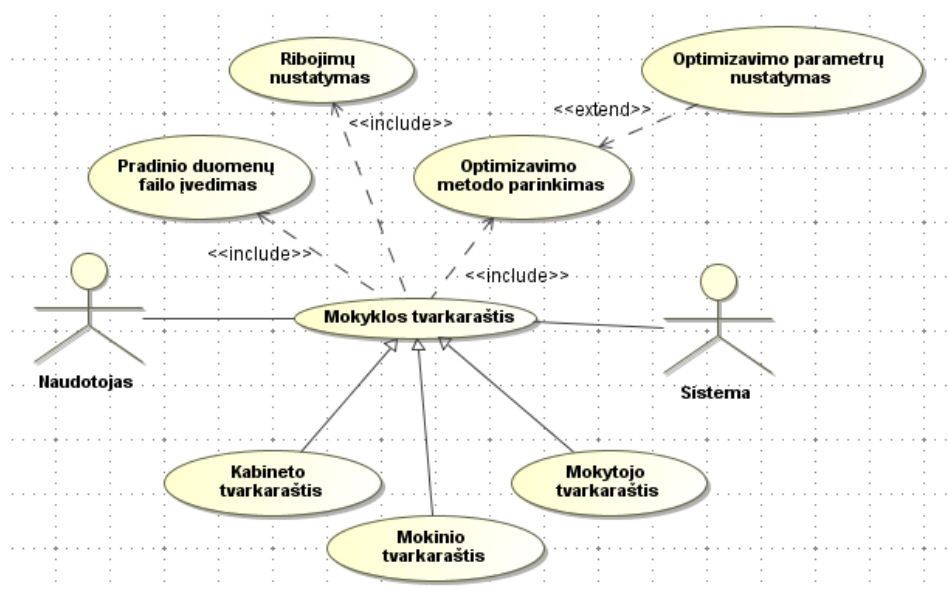
Šiuo metu Matematikos ir informatikos institute yra sukurta profiliuotos mokyklos tvarkaraščių optimizavimo programa „Optima“. Ši programa sukurta įvertinus euristinius būtinus ir pakankamus mokyklos parametrus. Parametrai, kurie būna fiksuoti kiekvienoje mokykloje pagal individualias taisykles, yra įvertinami pradinių duomenų faile. Parametrai, kurie gali kisti, yra įvertinti programos darbo metu. Taip pat jie gali būti keičiami programos darbo metu. Sukurtas pradinių duomenų failas, kurį formuoti yra nesunku net mažai patirties darbe su kompiuteriais turinčiam vartotojui. Įdiegta lanksti pagalbos sistema.

### 3.2. „Optima“ projektavimas

Projektuojant profiliuotų mokyklų tvarkaraščio kūrimo ir optimizavimo programą „Optima“, pagrindinis tikslas buvo – sukurti intuityvų programos valdymą. Taip pat buvo svarbu, kad programa optimizuotų suformuotą tvarkaraštį. Pagrindiniai reikalavimai programai buvo tokie:

- Suformuoti būtinus ir pakankamus mokyklos nuostatus įvertinanti pradinį duomenų failą.
- Sukurti intuityviai valdomą programą, kuri nereikalautų specialaus pasiruošimo ar specialių žinių.
- Suformuotus rezultatus pateikti tiek programos darbo aplinkoje, tiek archyve, kurį galima parsisiųsti į asmeninį kompiuterį.
- Sukurti programą, veikiančią bet kokioje operacinėje sistemoje.
- Įdiegti į programą kelis optimizavimo metodus.
- Įdiegti Bayes metodą, kuris optimizuotų atkaitinimo modeliavimo parametrus.
- Sukurti programą, kurioje būtų galima darbo metu įvertinti mokyklos tvarkaraščio formavimo ribojimus.

„Optima“ programai kurti buvo naudojama laisvai platinama programa „Eclipse“. Kaip minėta 1 skyriuje, programa „Optima“ realizuota JAVA programavimo kalba ir veikia interneto aplinkoje. Taigi „Optima“ veikia visose populiariose operacinėse sistemose. Taigi pagal šiuos reikalavimus 3.1 paveiksle pateikiama programos funkcionalumą parodanti diagrama.

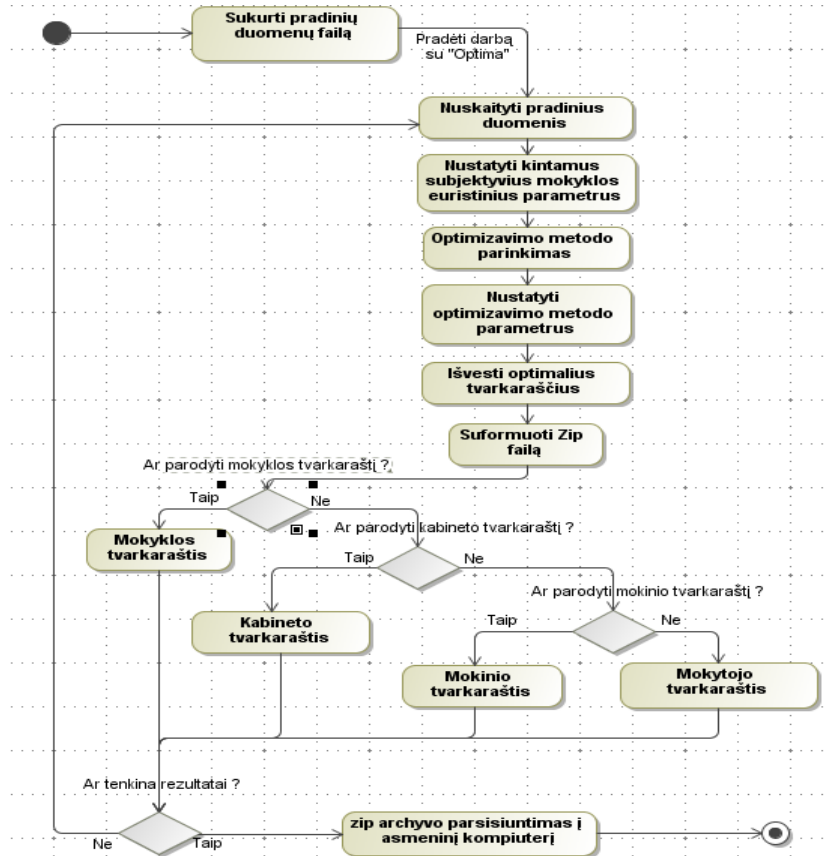


3.3 pav. Programos „Optima“ funkcionalumo diagrama

Pateiktoje programos funkcionalumo diagramoje pavaizduotos įrankio savybės. Šios savybės yra reikalingos optimalaus tvarkaraščio formavimui pagal

pradinių duomenų faile ir programoje nurodytus reikalavimus. Programos darbo metu atliekama daugiau funkcijų, bet čia parodytos tik svarbiausios.

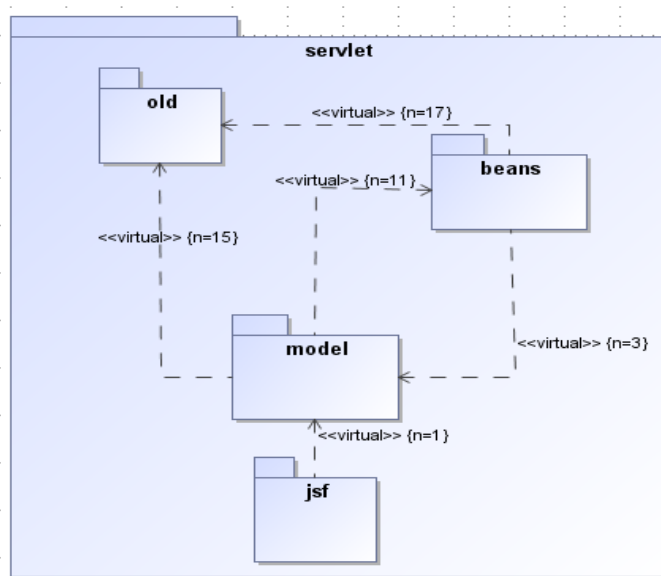
Pagal pateiktą programos funkcionalumo diagramą (3.3 pav.), suformuotas profiliuotos mokyklos tvarkaraščio kūrimo ir optimizavimo konceptualusis modelis. Jis vaizduojamas 3.4 paveiksle.



3.4 pav. Optimalaus tvarkaraščių kūrimo proceso konceptualaus modelio diagrama

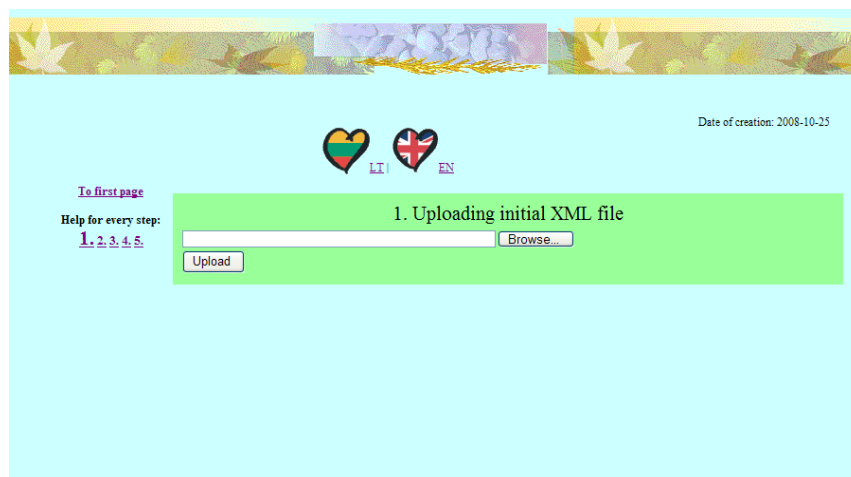
Pateiktame programos veikimo modelyje nėra įvertintas pradinių duomenų failo kūrimas. Tačiau šis veiksmas turėtų būti suvokiamas kaip atliekamas automatiškai norint pradėti darbą su programa. Kadangi šiai programai sukurtos 36 klasės, tai klasių modelis yra didelis ir painus. Todėl šioje disertacijoje pateikiami tik klases apjungiantys paketai (3.5 pav.).





3.5 pav. Programų paketų diagrama

Sukurta programa yra valdoma intuityviai. Nereikia specialaus pasiruošimo ar specialių darbo su kompiuteriu žinių. Užtenka pagrindinių darbo su kompiuteriu žinių ir interneto adresu, kurio pagalba yra pasiekiamą programa „Optima“. Sukurtos programos pradinis langas pavaizduotas 3.6 paveiksle.



3.6 pav. „Optima“ pradinis darbo langas

Programos „Optima“ darbo langas gali būti tiek angliškas, tiek lietuviškas. Yra įdiegta pagalba kiekviename žingsnyje. Plačiau apie programos veikimą ir rezultatų valdymą rašoma „B priede“. Darbas su programa vyksta 5 etapais:

1. Pradinių duomenų failo įrašymas.
2. dažnai kintančių mokyklos euristicinių parametrų nustatymas.
3. Optimizavimo metodo parinkimas.
4. Optimizavimo metodo parametrų nustatymas.
5. Rezultatų peržiūra.

### 3.3. Rekomendacijos „Optima“ programai plėtoti

Programa atitinka profiliuotos mokyklos reikalavimus. Tačiau ji nėra tobula. Gavus papildomų duomenų apie moksleivių pageidavimų fiksavimą, siūloma ateityje atlikti tokius patobulinius euristiciniams mokyklos parametrams vertinti:

- Suformuotosios moksleivių grupės stabilumo vertinimą.
  - Moksleivio pasirinktos disciplinos lygių fiksavimą.
  - Disciplinų, kurios turi būti alternatyvių disciplinų sąrašė, apjungimo fiksavimą.
  - Dėstomų disciplinų prioritetų nustatymą pradiniam duomenų faile.
  - Įdiegti „Multi-start“ metodą, naudojančią keletą procesorių.
- Taip pat siūlomi patobulinimai programos vizualiam valdymui pagerinti:
- Įdiegti disciplinų tvarkaraščio modifikavimą rankiniu būdu įterpiančias disciplinas iš vienos vietos į kitą.
  - Patobulinti rezultatų pateikimą.
  - Papildyti rezultatų pateikimą formuojant mokyklos tvarkaraščio „tinklelį“.

### 3.4. Trečiojo skyriaus išvados

1. Sukurtoji optimizavimo programa yra nauja ir turi didelę praktinę vertę tiriant optimizavimo metodų veikimą sudarant profiliuotų mokyklų tvarkaraščius.
2. Lyginant su esamomis laisvai platinamomis ir komercinėmis programomis, programa „Optima“ suteikia platesnes galimybes formuojant optimalų tvarkaraštį, kuris yra artimas realiam. Vartotojas gali intuityviai įvertinti arba suskaičiuoti pagal savo mokymo įstaigos resursus visus individualius poreikius, bei juos koreguoti.

3. Siūlomas kompleksinis optimizavimo metodas užtikrina pakankamą efektyvumą optimizuojant profiliuotos mokyklos tvarkaraštį įvertinant pagrindinius, būtinus ir pakankamus, reikalavimus.

---

## Bendrosios išvados

### **Bendrosios išvados:**

1. Nustatyta, kad profiliuotų mokyklų tvarkaraščio uždavinys polinomiškai nesprendžiamas, todėl jo optimizavimui reikalingi euristiniai optimizavimo metodai, kurie suteikia galimybę suformuoti optimalų tvarkaraštį pagal griežtus reikalavimus, tačiau neįvertinant euristinio mokytojo svarbumo mokykloje. Šio tvarkaraščio pagrindu profiliuota mokykla gali kurti realų tvarkaraštį.
2. Atlikus literatūros analizę nustatyta, kad JAVA kalbos servlet režimas patogus mokykloms, nes užtikrina visišką nepriklausomumą nuo operacinės sistemos bei nereikalauja papildomos programinės įrangos.
3. Nustatyta, kad pritaikytas vektorinio optimizavimo teorijos skaliarizacijos metodas, formuluojant profiliuotos mokyklos tvarkaraščių optimizavimo uždavinius, leidžia įvertinti individualius mokymo įstaigos euristinius kriterijus bei užtikrina Pareto optimumą.
4. Apskaičiuota, kad pritaikytas atkaitinimo modeliavimo parametru optimizavimas naudojant Bayes metodus yra efektyvus būdas padidinti sprendimo tikslumą, lyginant su paprastu atkaitinimo modeliavimo metodu, nes:
  - a. Patikrinama žymiai daugiau variantų.
  - b. Per tą patį laiką randa tikslesnį sprendinį.

5. Sukurtas ir prie konkrečių uždavinio sąlygų pritaikytas kompleksinis optimizavimo metodas efektyviai optimizuoja profiliuotos mokyklos tvarkaraštį įvertinant pagrindinius, būtinus ir pageidaujamus, reikalavimus.
6. Išanalizuota, kaip skirtingai nurodytos euristikos parametrai įtakoja optimizavimo metodo sprendimo greitį ir tikslumą. Tiksliausias sprendinys gaunamas kombinuotu atkaitimo modeliavimo ir Bayes metodu.
7. Remiantis eksperimentų metu gautais rezultatais nustatyta, kad:
  - a. Lietuvos mokyklose naudojamos komercinės kompiuterinės programos nėra pritaikytos profiliuotų mokyklų tvarkaraščiams optimizuoti.
  - b. Sukurtoji optimizavimo programa turi didelę praktinę vertę tiriant optimizavimo metodų veikimą sudarant profiliuotų mokyklų tvarkaraščius.
8. Sukurtoji programa suteikia platesnes galimybes suformuoti optimalų tvarkaraštį artimą realiam, lyginant su esamomis laisvai platinamomis ir komercinėmis programomis. Pritaikytos euristikos, kurių dėka vartotojas gali intuityviai įvertinti arba paskaičiuoti savo mokymo įstaigos resursus ir individualius poreikius, bei juos koreguoti.

---

## Literatūros sąrašas

Abboud, N.; Sakawa, M.; Inuiguchi, M. 1998. School scheduling using threshold accepting, *Cybernetics and Systems*. 29(6): 593–611., ISSN 0196-9722.

Abela, J. A. 1992. *Parallel Genetic Algorithm for Solving the School Timetabling Problem*. Division of Information Technology, C.S.I.R.O.

Abramson, D. A. 1991. Constructing school timetables using simulated annealing: sequential and parallel algorithms, in *Management Science*. 37(1): 98–113.

Abramson, D. A.; Dang, H. 1993. School timetables: a case study in simulated annealing, in *Lecture Notes in Economics and Mathematical Systems: Applied Simulated Annealing*, 103–124.

Abramson, D. A.; Krishnamoorthy, M.; Dag, H. 1996. Simulated annealing cooling schedules for the school timetabling problem, in *Asia-Pacific Journal of Operational Research*. 16: 1–22.

Adamidis, P.; Arakapis, P. 1997. Weekly lecture timetabling with genetic algorithms. in *Burke and Carter*, Proc. of PATAT-97, Toronto, Canada, 115–122, ISBN 0-7727-6703-3.

Alaburdienė, R. 2008. *Tvarkaraštis be „Mimosa“ – tvarkaraštis be ateities???* Prieiga per internetą: <<http://www.mockus.org/optimum/mimosa/alaburd.htm>>.

*Algoritmų sudėtingumas*. 2008. Prieiga per internetą: <[http://lt.wikipedia.org/wiki/Algoritm%C5%B3\\_sud%C4%97tingumas](http://lt.wikipedia.org/wiki/Algoritm%C5%B3_sud%C4%97tingumas)>.

- Appleby, J. S.; Blake, D. V.; Newman, E. A. 1961. Techniques for producing school timetables on a computer and their application to other scheduling problems, in *The Computer Journal*. 3(4): 237–245.
- Apte, A.; Jayasuriya, A.; Kennington, J.; Krass, I.; Mohamed, R.; Sorensen, S.; Whitler, J. 1998. Class scheduling algorithms for Navy training schools, in *Naval Research Logistics*. 45(6): 533–551, ISSN 0894-069X.
- aSc TimeTables. School scheduling software for primary/secondary schools*. 2008. Prieiga per internetą: <<http://www.asctimetables.com/>>.
- Aust, R. J. 1976. An improvement algorithm for school timetabling, in *Computer Journal*. 8: 135–146.
- Badri, M. A.; Davis, D. L.; Davis, D. F.; Hollingsworth, J. 1998. A multi-objective course scheduling model: combining faculty preferences for courses and times, in *Computers & Operations Research*. 25(4): 303–316.
- Bajeso teorema*. 2008. Prieiga per internetą: <[http://lt.pandapedia.com/wiki/Bajeso\\_teorema](http://lt.pandapedia.com/wiki/Bajeso_teorema)>.
- Bardadym, V. A. 1996. Computer-aided school and university timetabling: the new wave, in *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. 1153: 22–45, ISSN 0302-9743.
- Bayes' theorem*. 2008. Prieiga per internetą: <[http://en.pandapedia.com/wiki/Bayes%27\\_theorem](http://en.pandapedia.com/wiki/Bayes%27_theorem)>.
- Caldeira, J. P.; Rosa, A. C. 1997. School timetabling using genetic search, *Proceedings of the 2nd International Conference on the Practice and Theory of Automated Timetabling*. University of Toronto, Canada.
- Chorbev, I.; Dimitrovski, I.; Mihajlov, D.; Loskovska, S. 2007. Hybrid heuristics for solving the constrained modeled high school scheduling problem, in *EUROCON, 2007. The International Conference on „Computer as a Tool“*. 2007 September 9–12. Warsaw, 2242–2249, ISBN 978-1-4244-0813-9.
- Colomi, A.; Dorigo, M.; Maniezzo, V. 1998. Metaheuristics for high-school timetabling. computational optimization and applications, in *Springer*. Netherlands. 9(3): 275–298, ISSN 0926-6003.
- Complexity of Algorithms, Wolfram research*. 2008. Prieiga per internetą: <<http://mathworld.wolfram.com/topics/ComplexityofAlgorithms.html>>.
- Constraint programming*. 2008. Prieiga per internetą: <<http://www.aiai.ed.ac.uk/links/constr.html>>.
- Cooper, T. B.; Kingston, J. H. 1993. The solution of real instances of the timetabling problem, in *The Computer Journal*. 36(7): 645–653, ISSN 0010-4620.
- Cooper, T. B.; Kingston, J. H. 1995. The complexity of timetable construction problems, in *Selected papers from the First International Conference on Practice and Theory of Automated Timetabling, Lecture Notes In Computer Science*. 1995 August 29 – September 01, 1153: 283–295, ISBN 3-540-61794-9.

- Cooper, T. B.; Kingston, J. H. 1995. A Program for constructing high school timetables in The University of Sidney. *Technical Report*. 496, ISBN-0-86758-960-4.
- de Werra, D. 1985. An introduction to timetabling, in *European Journal of Operational Research*. 19: 151–162.
- de Werra, D. 1971. Construction of school timetables by flow methods, in *INFOR – Canadian Journal of Operations Research and Information Processing*. 9: 12–22.
- Di Stefano, C.; Tettamanzi, A. G. B. 2001. An evolutionary algorithm for solving the school time-tabling problem, in *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. 2037, ISBN 978-3-540-41920-4.
- Dignum, F.P.M.; Nuijten, W.P.M.; Janssen, L.M.A. 1995. Solving a time tabling problem by constraint satisfaction, *Technical Report*. Eindhoven University of Technology.
- Duong, T. A.; Lam, K. H. 2004. Combining constraint programming and simulated annealing on university exam timetabling, in *RIVF'04*. 205–210.
- Dzemyda, D.; Šaltenis, V.; Tiešis, V. 2007. *Optimizavimo metodai*, Matematikos ir informatikos institutas. ISBN 978-9986-680-41-3.
- Felinskas, G. 2006. Euristicinių metodų tyrimas ir taikymas tvarkaraščiams optimizuoti, *Daktaro tezės*. Matematikos ir Informatikos Institutas.
- Felinskas, G.; Sakalauskas, L. 2006. Optimization of resource constraint project schedules, in *Technological and Economic Development of Economy*. XII(4): 307–313, ISSN 1392-8619.
- Felinskas, G.; Sakalauskas, L. 2003. Pareto tipo modeliai modeliuojamojo atkaitinimo algoritmuose, in *Lietuvos Matematikos Rinkinys*. T.43: 573–578, ISSN 0132-2818.
- Felinskas, G.; Sakalauskas, L. 2005. Tvarkaraščių su ribotais ištekliais sudarymo algoritmai ir jų tyrimas, in *Matematika ir Matematinis Modeliavimas*. 1: 105–110, ISSN 1822-2757.
- Fernandes, C.; Caldeira, J. P.; Melicio, F.; Rosa, A. 1999. High school weekly timetabling by evolutionary algorithms, in *Proceedings of the 1999 ACM symposium on Applied computing*. San Antonio, Texas, United States. 344–350, ISBN 1-58113-086-4.
- Filho, G. R.; Lorena, L. A. N. A. 2001. Constructive evolutionary approach to school timetabling, in *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. 2037: 130–139, ISBN 978-3-540-41920-4.
- Gaidukevičienė, R.; Kurilovas, E. 2005. Comparative study of profiled school scheduling programs in lithuania, in *Informatics in Education*. 4(1): 19–42, ISSN 1648-5831.
- Garey, M. R.; Johnson, D. S. 1979. Computers and intractability: a guide to the theory of NP-completeness, *Freeman*. New York.
- Genetiniai algoritmai*. 2008. Prieiga per internetą: <[http://lt.wikipedia.org/wiki/Genetiniai\\_algoritmai](http://lt.wikipedia.org/wiki/Genetiniai_algoritmai)>.
- Genetic algorithms, (2002)*. 2008. Prieiga per internetą: <[http://www.pcai.com/web/ai\\_info/genetic\\_algorithms.html](http://www.pcai.com/web/ai_info/genetic_algorithms.html)>.



- General algorithms*, wolfram research. 2008. Prieiga per internetą: <<http://mathworld.wolfram.com/topics/GeneralAlgorithms.html>>.
- Glazebrook, K. D. 1979. Scheduling stochastic tasks with exponential service times on parallel processors, in *Journal of Applied Probability*. 16: 685–689.
- Jacobson, S. H.; Yücesan, E. 2004. Global optimization performance measures for generalized hill climbing algorithms, in *Journal of Global Optimization*. 29: 173–190.
- Java (kalba)*. 2008. Prieiga per internetą: <[http://lt.wikipedia.org/wiki/Java\\_%28kalba%29](http://lt.wikipedia.org/wiki/Java_%28kalba%29)>.
- Java technologija*. 2008. Prieiga per internetą: <[http://www.nokia.lt/link?cid=EDITORIAL\\_430689](http://www.nokia.lt/link?cid=EDITORIAL_430689)>.
- Kingston, J. H. 2007. The KTS high school timetabling system in *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. 3867: 308–323. ISBN 978-3-540-77344-3.
- Kong, SC.; Kowk, LF. 1999. A conceptual model of knowledge-based time-tabling system, in *Knowledge-Based Systems*. 12(3): 81–93. ISSN 0950-7051.
- Lee, H. S. C.; Hermosilla, A. Y. 2008. *Timetabling constrained system via genetic algorithm. (2001)*. Prieiga per internetą: <<http://courseweb.xu.edu.ph/courses/msit134/materials/PSCS2001/PS3.pdf>>.
- Lassaigne, R.; de Rougemont, M. 1999. Logika ir algoritmų sudėtingumas. ISBN 9986-34-029-2.
- Loukil, T.; Teghem, J.; Tuytens, D. 2003. Solving multi-objective production scheduling problems using metaheuristics, in *European Journal of Operational Research*. 161: 42–61.
- Macon, N.; Walker, E.E. 1966. A Monte-Carlo algorithm for assigning students to classes, in *Comm, ACM*. 9(6): 339–340.
- Melicio, F.; Caldeira, P.; Rosa, A. 2005. Solving real school timetabling problems with meta-heuristics, in *Proceedings of the 4th WSEAS International Conference on Applied Mathematics and Computer Science*. Rio de Janeiro, Brazil. ISBN 960-8457-17-3.
- MIMOSA*. 2008. Prieiga per internetą: <<http://www.mimosasoftware.com>>.
- Mockus, J. 2004. *A set of Examples of Global and Discrete Optimization 2 – Home Work for Graduate Students*, Kluwer Academic Publishers.
- Mockus, J. 2002. Bayesian heuristic approach to global optimization and examples, in *Journal of Global Optimization*. 22(1–4): 191–203. ISSN 0925-5001.
- Mockus, J.; Eddy, W.; Mockus, A.; Mockus, L.; Reklaitis, R. 2000. *Bayesian heuristic approach to discrete and global optimization – algorithm, visualisation, software, and applications*, Kluwer Academic Publishers.
- Monte Karlo metodos*. 2008. Prieiga per internetą: <[http://lt.wikipedia.org/wiki/Monte\\_Karlo\\_metodas](http://lt.wikipedia.org/wiki/Monte_Karlo_metodas)>.

- Nekiūnienė, V. 2008. *Mokyklos pamokų tvarkaraščio optimizavimo programos „School – COMPLETE: school scheduling problem“ analizė*. Prieiga per internetą: <<http://www.mockus.org/optimum/mimosa/nekiun.htm>>.
- Onno B. de Gans. 1981. A computer timetabling system for secondary schools in the Netherlands, in *European Journal of Operational Research*. 7(2): 175–182.
- Optimizavimas (matematika)*. 2008. Prieiga per internetą: <[http://lt.wikipedia.org/wiki/Optimizavimas\\_%28matematika%29](http://lt.wikipedia.org/wiki/Optimizavimas_%28matematika%29)>.
- Ozcan, E.; Alkan, A. 2002. Timetabling using a steady state genetic algorithm, in *Proceedings of the 4th PAPT*. 104–107.
- Petrovic, S.; Burke, E. 2004. University timetabling, handbook of scheduling: algorithms, models, and performance analysis, in *CRC Press*. 45: 1–23.
- Programavimo kalba*. 2008. Prieiga per internetą: <[http://lt.wikipedia.org/wiki/Programavimo\\_kalba](http://lt.wikipedia.org/wiki/Programavimo_kalba)>.
- RECTOR*. 2008. Prieiga per internetą: <<http://www.rector.ru>>.
- Ross, P.; Hart, E.; Corne, D. 2003. Genetic algorithms and timetabling, in *Advances in Evolutionary Computing: Theory and Applications*. 755–771. ISBN 3-540-43330-9.
- Saleh Elmohamed, M. A.; Coddington, P.; Fox, G. 1998. A comparison of annealing techniques for academic course scheduling., in *Lecture Notes in Computer Science*. Springer Berlin/Heidelberg. 1480: 92. ISBN 978-3-540-64979-3.
- Santos, H. G.; Ochi, L. S.; Uchoa, E. B. 2006. *Combining metaheuristics and integer programming on school timetabling problem*, Proc. of the 1st Workshop on Mathematical Contributions to Metaheuristics (MATHEURISTICS 2006), Bologna, Italy.
- Santos, H. G.; Ochi, L. S.; Souza, M. J. F. 2004. An efficient Tabu-Search heuristic for the school timetabling problems, in *Lecture Notes in Computer Science*. Springer. 3059: 468–482.
- Schaerf, A.; Schaerf, M. 1995. Local search techniques for high school timetabling, in *ICPTAT-95, Conference ScSc95*. 313–323.
- School scheduling, discussions about applications of MIMOSA and GMJI systems*. 2008. Prieiga per internetą: <<http://www.soften.ktu.lt/~mockus/newalg.pdf>>.
- Schroth, G. 1997. *Fundamentals of School Scheduling*, Technomic Publishing Co.
- School-Complete*. 2008. Prieiga per internetą: <<http://soften.ktu.lt/~mockus/>>.
- School Schedule optimization program by Auris*. 2008. Prieiga per internetą: <<http://soften.ktu.lt/~mockus/>>.
- School Schedule optimization program by Vydunas*. 2008. Prieiga per internetą: <<http://soften.ktu.lt/~mockus/>>.
- Souza, M. J. F.; Maculan, N.; Ochi, L. S. 2001. A GRASP-tabu search algorithm to solve a school timetabling problem, in *IV Metaheuristic International Conference, 2001*. Porto-Portugal. Proceedings of IV Metaheuristic International Conference (IV MIC). Porto: MIC/INESC-PORTO. 1: 53–59.

Silva, J. D.; Bruke, E. K.; Petrovic, S. 2003. *An introduction to multiobjective metaheuristic for scheduling and timetabling*, Automated Scheduling, Optimisation and Planning Research Group School of Computer Science and IT, University of Nottingham, UK.

*Simulated annealing*. 2008. Prieiga per internetą: <[http://en.wikipedia.org/wiki/Simulated\\_annealing](http://en.wikipedia.org/wiki/Simulated_annealing)>.

Šilingas, D. 2005. JAVA technologijų apžvalga, in *Konferencijos „Informacinės technologijos 2005“ pranešimų medžiaga, II sekcija*, Kaunas: Technologija, 50–58. ISBN 9955-09-788-4.

Thrampoulidis, KX.; Goumopoulos, C.; Housos, E. 1997. Rule handling in the day-to-day recourse management problem: an object-oriented approach, in *Information and Software Technology*. 39(3): 185–193. ISSN 0950-5849.

*Tikimybė*. 2008. Prieiga per internetą: <<http://lt.wikipedia.org/wiki/Tikimyb%C4%97>>.

Tripathy, A. 1984. School timetabling – a case in large binary integer linear programming, in *Management Science*. 30(12): 1473–1489.

Tripathy, A. 1984. School Timetabling – A case in Large Binary Integer Linear Programming. *Discrete Applied Mathematics*, 35(3): 313–323.

Ulph, A. 1977. Notes – an extension of a result in school timetabling, in *INFOR*. 15: 255–257.

Willemen, R. J. 2002. School timetable construction: algorithms and complexity, in *Technische Universiteit Eindhoven*. 2002 m. April 16d. ISBN 90-386-1011-4.

Valouxis, Ch.; Housos, E. 2003. Constraint programming approach for school timetabling, in *Computers & Operations Research*. 30(10): 1555–1572.

van Laarhoven, P. J. M.; Aarts, E. H. L. 1987. *Simulated annealing: theory and applications*, D. Reidel Publishing Company. Kluwer Academic Publishers Group. ISBN-10: 9027725136, ISBN-13: 978-9027725134.

Yang, R. L. 2000. Convergence of the simulated annealing algorithm for continuous global optimization, in *Journal of optimization theory and applications*. 104(3): 691–716.

Yoshikawa, M.; Kaneko, K.; Yamanouchi, T.; Watanabe, M. 1996. A constraint-based high school scheduling system, in *IEEE Expert: Intelligent Systems and Their Applications*. 11(1): 63–72.

Žilinskas, A. 2005. *Matematinis programavimas*. ISBN 9986-501-51-2.

---

## Autoriaus mokslinių publikacijų disertacijos tema sąrašas

### **Straipsniai recenzuojamuose mokslo žurnaluose**

Pupeikienė, L.; Strukov, D.; Bivainis, V.; Mockus, J. 2009. Optimization algorithms in school scheduling programs: study, analysis and results, *Informatics in Education*. 8(1): 69–84. ISSN 0868-4952.

Pupeikienė, L.; Mockus, J. 2005. School scheduling optimization program, *Information Technology and Control*. 34(2): 161–170. ISSN 1392-124X.

### **Straipsniai leidiniuose recenzuojamuose duomenų bazėje (Thomson ISI Proceedings)**

Pupeikienė, L. 2006. News in the school schedule optimization program, in *Proceedings of the 2nd International Conference „Informatics in Secondary Schools: Evolution and Perspectives“ (ISSEP 2006)*. Vilnius, Lithuania, 7–11 November, 2006, 269–278. ISBN 9955-680-47-4 [ISI Proceedings].

**Straipsniai kitose leidiniuose**

Pupeikienė, L.; Strukov. D.; Bivainis, V.; Mockus, J. 2008. Mokyklos tvarkaraščių optimizavimo uždavinių tyrimas, in *11-osios Lietuvos Jaunųjų Mokslininkų konferencijos „Mokslas – Lietuvos ateitis“ straipsnių rinkinys, Informatika*. 2008 m. balandžio 10 d. Vilnius: Technika, 132–153. ISBN 978-9955-28-302-7.

Pupeikienė, L. 2006. Mokyklos profiliuotų klasių tvarkaraščių optimizavimo programos patobulinimai, in *Informacinės technologijos 2006. Mokslinės techninės konferencijos pranešimų medžiaga*. Kauno technologijos universitetas, Kaunas, Lietuva, 2006 m. sausio 25–26 d. Kaunas: Technologija, 146–152. ISBN 9955-09-993-3.

Gaidukevičienė, R.; Kurilovas, E.; Nijazova, A.; Pupeikienė, L. 2005. Profiliuotos mokyklos tvarkaraščio optimizavimo tyrimas, in *Kompiuterininkų dienos 2005. 12-osios mokslinės kompiuterininkų konferencijos pranešimų medžiaga*. Klaipėda, 2005 m. rugsėjo 15–17 d. Klaipėda, 231–239, ISBN 9986-34-147-7.

Pupeikienė, L. 2005. Mokyklos tvarkaraščių optimizavimo sistema, in *Informacinės technologijos 2005. Mokslinės techninės konferencijos pranešimų medžiaga*. Kauno technologijos universitetas, Kaunas, Lietuva, 2005 m. sausio 26–28 d. Kaunas: Technologija, 24–33. ISBN 9955-09-788-4.

---

## Priedai

### A priedas. Metodų darbo tyrimo metu gauti rezultatai

Lentelėse aprašomi rezultatų vidurkiai gauti atlikus po 100 bandymų (**REZULTATŲ VIDURKIS ATLIKUS 100 BANDYMŲ**) ir geriausi rezultatai, kurie gauti atliekant šiuos 100 bandymų (**GERIAUSI REZULTATAI GAUTI ATLIKUS 100 BANDYMŲ**). Kiekviena lentelė skirta skirtingiems optimizavimo metodams. Šių lentelių grafinė analizė atlikta disertacijos antrajame skyriuje.

**1A lentelė.** Lokalaus determinuoto metodo tyrimo rezultatai

<b>REZULTATŲ VIDURKIS ATLIKUS 100 BANDYMŲ</b>		
<b>Iteracijų kiekis</b>	<b>Baudos taškų skirtumas</b>	<b>Darbo laikas</b>
10	20	00:00:07
100	987	00:00:15
1000	1232	00:00:28
10000	1545	00:00:32
<b>GERIAUSI REZULTATAI, GAUTI ATLIKUS 100 BANDYMŲ</b>		
10	4275	00:00:07

1A lentelės pabaiga

<b>Iteracijų kiekis</b>	<b>Baudos taškų skirtumas</b>	<b>Darbo laikas</b>
100	7095	00:00:15
1000	8035	00:00:28
10000	9540	00:00:32

2A lentelė. Lokalaus atsitiktinio metodo tyrimo rezultatai

<b>REZULTATŲ VIDURKIS ATLIKUS 100 BANDYMŲ</b>				
<b>K</b>	Iteracijų = 10; Darbo laikas (00:00:10)	Iteracijų = 100; Darbo laikas (00:00:18)	Iteracijų = 1000; Darbo laikas (00:00:33)	Iteracijų = 10000; Darbo laikas (00:00:42)
	<b>Pagerėjimas</b>	<b>Pagerėjimas</b>	<b>Pagerėjimas</b>	<b>Pagerėjimas</b>
1	20	954	2053	1945
2	28	856	1956	1581
3	25	974	2259	1965
4	43	792	2154	2153
5	72	1253	2350	2605
6	61	963	1989	1943
7	54	1125	2158	2154
8	29	1127	1785	2348
9	45	845	2003	2107
10	25	1217	1903	2215
<b>GERIAUSI REZULTATAI, GAUTI ATLIKUS 100 BANDYMŲ</b>				
<b>K</b>	Iteracijų = 10; Darbo laikas (00:00:10)	Iteracijų = 100; Darbo laikas (00:00:18)	Iteracijų = 1000; Darbo laikas (00:00:33)	Iteracijų = 10000; Darbo laikas (00:00:42)
	<b>Pagerėjimas</b>	<b>Pagerėjimas</b>	<b>Pagerėjimas</b>	<b>Pagerėjimas</b>
1	1160	3595	3820	3930
2	1885	3670	3970	4440
3	3240	4250	4330	5565
4	3290	5865	5995	6015
5	4025	6110	6300	6590
6	4065	7205	7835	8290
7	4280	7600	7865	8775
8	4410	8430	8580	8825
9	4955	9595	9835	9955
10	5275	10140	10380	11930

3A lentelē. Atkaitinimo modeliavimo metodo tyrimo rezultatai (pagal  $X_2$ )

REZULTATŲ VIDURKIS ATLIKUS 100 BANDYMŲ						
10 ≤ $X_1$ ≤ 100						
$X_2$	Iteracijų = 10		Iteracijų = 100		Iteracijų = 1000	
	Pagerėjimas	laikas	Pagerėjimas	laikas	Pagerėjimas	laikas
1	201	00:00:13	352	00:00:20	1577	00:00:45
2	299	00:00:13	873	00:00:20	1359	00:00:45
3	446	00:00:13	1255	00:00:20	1428	00:00:45
4	188	00:00:13	1218	00:00:20	2595	00:00:45
5	490	00:00:13	375	00:00:20	1542	00:00:45
6	435	00:00:13	567	00:00:20	1315	00:00:45
7	465	00:00:13	482	00:00:20	1245	00:00:45
8	452	00:00:13	1254	00:00:20	594	00:00:45
9	678	00:00:13	752	00:00:20	935	00:00:45
10	106	00:00:13	898	00:00:20	762	00:00:45

100 ≤ $X_1$ ≤ 1000						
$X_2$	Iteracijų = 10		Iteracijų = 100		Iteracijų = 1000	
	Pagerėjimas	laikas	Pagerėjimas	laikas	Pagerėjimas	laikas
1	1616	00:00:13	798	00:00:20	1085	00:00:45
2	1741	00:00:13	487	00:00:20	304	00:00:45
3	1737	00:00:13	615	00:00:20	409	00:00:45
4	1486	00:00:13	1284	00:00:20	429	00:00:45
5	2053	00:00:13	1503	00:00:20	750	00:00:45
6	1890	00:00:13	1076	00:00:20	1605	00:00:45
7	1782	00:00:13	665	00:00:20	2081	00:00:45
8	1766	00:00:13	751	00:00:20	1772	00:00:45
9	1519	00:00:13	845	00:00:20	1050	00:00:45
10	1683	00:00:13	797	00:00:20	944	00:00:45

1000 ≤ $X_1$ ≤ 10000						
$X_2$	Iteracijų = 10		Iteracijų = 100		Iteracijų = 1000	
	Pagerėjimas	laikas	Pagerėjimas	laikas	Pagerėjimas	laikas
1	494	00:00:13	268	00:00:20	1171	00:00:45
2	210	00:00:13	289	00:00:20	891	00:00:45
3	433	00:00:13	499	00:00:20	2405	00:00:45



3A lentelės tęsinys

$X_2$	Iteracijų = 10		Iteracijų = 100		Iteracijų = 1000	
	Pagerėjimas	laikas	Pagerėjimas	laikas	Pagerėjimas	laikas
4	227	00:00:13	750	00:00:20	938	00:00:45
5	813	00:00:13	952	00:00:20	657	00:00:45
6	499	00:00:13	1076	00:00:20	1543	00:00:45
7	248	00:00:13	127	00:00:20	759	00:00:45
8	185	00:00:13	860	00:00:20	1074	00:00:45
9	530	00:00:13	697	00:00:20	1490	00:00:45
10	302	00:00:13	529	00:00:20	1205	00:00:45

## GERIAUSI REZULTATAI, GAUTI ATLIKUS 100 BANDYMŲ

 $10 \leq X_1 \leq 100$ 

$X_2$	Iteracijų = 10		Iteracijų = 100		Iteracijų = 1000	
	Pagerėjimas	laikas	Pagerėjimas	laikas	Pagerėjimas	laikas
1	1210	00:00:13	2095	00:00:20	4475	00:00:45
2	1300	00:00:13	3785	00:00:20	4650	00:00:45
3	2445	00:00:13	6300	00:00:20	8890	00:00:45
4	1350	00:00:13	2270	00:00:20	4500	00:00:45
5	1190	00:00:13	1905	00:00:20	3170	00:00:45
6	1435	00:00:13	2120	00:00:20	3270	00:00:45
7	1665	00:00:13	2275	00:00:20	4710	00:00:45
8	1885	00:00:13	6895	00:00:20	8355	00:00:45
9	2675	00:00:13	8210	00:00:20	11390	00:00:45
10	1190	00:00:13	1650	00:00:20	3295	00:00:45

 $100 \leq X_1 \leq 1000$ 

$X_2$	Iteracijų = 10		Iteracijų = 100		Iteracijų = 1000	
	Pagerėjimas	laikas	Pagerėjimas	laikas	Pagerėjimas	laikas
1	1616	00:00:13	2830	00:00:20	3085	00:00:45
2	1741	00:00:13	2450	00:00:20	3085	00:00:45
3	1737	00:00:13	2630	00:00:20	3010	00:00:45
4	1486	00:00:13	2870	00:00:20	3675	00:00:45
5	2053	00:00:13	3455	00:00:20	3895	00:00:45
6	1890	00:00:13	2290	00:00:20	2605	00:00:45
7	1782	00:00:13	2815	00:00:20	3360	00:00:45

3A lentelės pabaiga

$X_2$	Iteracijų = 10		Iteracijų = 100		Iteracijų = 1000	
	Pagerėjimas	laikas	Pagerėjimas	laikas	Pagerėjimas	laikas
8	1766	00:00:13	2840	00:00:20	3015	00:00:45
9	1519	00:00:13	2830	00:00:20	3075	00:00:45
10	1683	00:00:13	2875	00:00:20	3225	00:00:45

$1000 \leq X_1 \leq 10000$						
$X_2$	Iteracijų = 10		Iteracijų = 100		Iteracijų = 1000	
	Pagerėjimas	laikas	Pagerėjimas	laikas	Pagerėjimas	laikas
1	1210	00:00:13	2030	00:00:20	3245	00:00:45
2	2495	00:00:13	3690	00:00:20	3955	00:00:45
3	2535	00:00:13	3685	00:00:20	4200	00:00:45
4	2530	00:00:13	3875	00:00:20	4195	00:00:45
5	2415	00:00:13	3780	00:00:20	4530	00:00:45
6	2105	00:00:13	3425	00:00:20	3625	00:00:45
7	1550	00:00:13	2200	00:00:20	2840	00:00:45
8	1385	00:00:13	1995	00:00:20	2565	00:00:45
9	1830	00:00:13	4050	00:00:20	5565	00:00:45
10	1310	00:00:13	2235	00:00:20	3260	00:00:45

4A lentelė. Atkaitinimo modeliavimo metodo tyrimo rezultatai (pagal  $X_1$ )

REZULTATŲ VIDURKIS ATLIKUS 100 BANDYMŲ						
$1 \leq X_2 \leq 10$						
$X_1$	Iteracijų = 10		Iteracijų = 100		Iteracijų = 1000	
	Pagerėjimas	laikas	Pagerėjimas	laikas	Pagerėjimas	laikas
10	74	00:00:13	136	00:00:20	1224	00:00:45
20	57	00:00:13	160	00:00:20	760	00:00:45
30	436	00:00:13	1545	00:00:20	819	00:00:45
40	289	00:00:13	110	00:00:20	1150	00:00:45
50	307	00:00:13	520	00:00:20	1745	00:00:45
60	56	00:00:13	1125	00:00:20	1546	00:00:45
70	135	00:00:13	2030	00:00:20	1000	00:00:45
80	51	00:00:13	596	00:00:20	1240	00:00:45
90	126	00:00:13	115	00:00:20	1300	00:00:45
100	60	00:00:13	123	00:00:20	964	00:00:45

4A lentelės tęsinys

<b><math>1 \leq X_2 \leq 10</math></b>						
<b><math>X_1</math></b>	<b>Iteracijų = 10</b>		<b>Iteracijų = 100</b>		<b>Iteracijų = 1000</b>	
	<b>Pagerėjimas</b>	<b>laikas</b>	<b>Pagerėjimas</b>	<b>laikas</b>	<b>Pagerėjimas</b>	<b>laikas</b>
100	1737	00:00:13	123	00:00:20	964	00:00:45
200	1848	00:00:13	605	00:00:20	270	00:00:45
300	1778	00:00:13	542	00:00:20	338	00:00:45
400	1903	00:00:13	812	00:00:20	992	00:00:45
500	1349	00:00:13	1147	00:00:20	1054	00:00:45
600	1991	00:00:13	1196	00:00:20	544	00:00:45
700	1963	00:00:13	482	00:00:20	760	00:00:45
800	1840	00:00:13	1209	00:00:20	1072	00:00:45
900	1342	00:00:13	396	00:00:20	1623	00:00:45
1000	1626	00:00:13	767	00:00:20	911	00:00:45

<b><math>1 \leq X_2 \leq 10</math></b>						
<b><math>X_1</math></b>	<b>Iteracijų = 10</b>		<b>Iteracijų = 100</b>		<b>Iteracijų = 1000</b>	
	<b>Pagerėjimas</b>	<b>laikas</b>	<b>Pagerėjimas</b>	<b>laikas</b>	<b>Pagerėjimas</b>	<b>laikas</b>
1000	1383	00:00:13	1994	00:00:20	2563	00:00:45
2000	2536	00:00:13	3698	00:00:20	4204	00:00:45
3000	2107	00:00:13	3422	00:00:20	3629	00:00:45
4000	1211	00:00:13	2037	00:00:20	3248	00:00:45
5000	1319	00:00:13	2239	00:00:20	3263	00:00:45
6000	1835	00:00:13	4051	00:00:20	5563	00:00:45
7000	2538	00:00:13	3875	00:00:20	4197	00:00:45
8000	2413	00:00:13	3784	00:00:20	4538	00:00:45
9000	2494	00:00:13	3693	00:00:20	3959	00:00:45
10000	1558	00:00:13	2205	00:00:20	2841	00:00:45

<b>GERIAUSI REZULTATAI, GAUTI ATLIKUS 100 BANDYMŲ</b>						
<b><math>1 \leq X_2 \leq 10</math></b>						
<b><math>X_1</math></b>	<b>Iteracijų = 10</b>		<b>Iteracijų = 100</b>		<b>Iteracijų = 1000</b>	
	<b>Pagerėjimas</b>	<b>laikas</b>	<b>Pagerėjimas</b>	<b>laikas</b>	<b>Pagerėjimas</b>	<b>laikas</b>
10	2445	00:00:13	3300	00:00:20	3890	00:00:45
20	1885	00:00:13	3895	00:00:20	4355	00:00:45

4A lentelēs pabaiga

$X_1$	Iteraciju = 10		Iteraciju = 100		Iteraciju = 1000	
	Pagerējimas	laikas	Pagerējimas	laikas	Pagerējimas	laikas
30	2675	00:00:13	4210	00:00:20	4890	00:00:45
40	1665	00:00:13	2475	00:00:20	3710	00:00:45
50	1300	00:00:13	2385	00:00:20	3650	00:00:45
60	1210	00:00:13	2095	00:00:20	3475	00:00:45
70	1190	00:00:13	1905	00:00:20	3170	00:00:45
80	1350	00:00:13	2270	00:00:20	3500	00:00:45
90	1435	00:00:13	2120	00:00:20	3270	00:00:45
100	1270	00:00:13	1650	00:00:20	3295	00:00:45

$1 \leq X_2 \leq 10$						
$X_1$	Iteraciju = 10		Iteraciju = 100		Iteraciju = 1000	
	Pagerējimas	laikas	Pagerējimas	laikas	Pagerējimas	laikas
100	1985	00:00:13	1870	00:00:20	1675	00:00:45
200	2055	00:00:13	1455	00:00:20	1295	00:00:45
300	1980	00:00:13	1730	00:00:20	1315	00:00:45
400	2040	00:00:13	1630	00:00:20	2010	00:00:45
500	1985	00:00:13	2275	00:00:20	2825	00:00:45
600	2615	00:00:13	2430	00:00:20	3085	00:00:45
700	2140	00:00:13	1450	00:00:20	2995	00:00:45
800	2280	00:00:13	2315	00:00:20	3360	00:00:45
900	1890	00:00:13	1290	00:00:20	2605	00:00:45
1000	1865	00:00:13	1840	00:00:20	2875	00:00:45

$1 \leq X_2 \leq 10$						
$X_1$	Iteraciju = 10		Iteraciju = 100		Iteraciju = 1000	
	Pagerējimas	laikas	Pagerējimas	laikas	Pagerējimas	laikas
1000	3045	00:00:13	3465	00:00:20	4680	00:00:45
2000	4215	00:00:13	5055	00:00:20	5210	00:00:45
3000	3810	00:00:13	4950	00:00:20	4990	00:00:45
4000	3550	00:00:13	4565	00:00:20	5255	00:00:45
5000	4685	00:00:13	4585	00:00:20	5695	00:00:45
6000	4875	00:00:13	6055	00:00:20	7850	00:00:45
7000	4215	00:00:13	5690	00:00:20	6525	00:00:45
8000	4090	00:00:13	5645	00:00:20	5895	00:00:45
9000	4690	00:00:13	5125	00:00:20	5560	00:00:45
10000	3225	00:00:13	4350	00:00:20	4585	00:00:45

5A lentelė. Bayes metodo tyrimo rezultatai

REZULTATŲ VIDURKIS ATLIKUS 100 BANDYMŲ						
Bayes iteracijos = 5						
Iteracijos	$X_{1(\min)}$	$X_{1(\max)}$	$X_{2(\min)}$	$X_{2(\max)}$	Taškai	Laikas
10	10	10000	1	10	3183	00:00:20
50	10	10000	1	10	2777	00:00:45
100	10	10000	1	10	2894	00:01:05
500	10	10000	1	10	2903	00:03:28
1000	10	10000	1	10	3909	00:05:49
5000	10	10000	1	10	5432	00:08:55
10000	10	10000	1	10	6558	00:17:15

Bayes iteracijos = 50						
Iteracijos	$X_{1(\min)}$	$X_{1(\max)}$	$X_{2(\min)}$	$X_{2(\max)}$	Taškai	Laikas
10	10	10000	1	10	3024	00:02:15
50	10	10000	1	10	3559	00:04:53
100	10	10000	1	10	2027	00:12:15
500	10	10000	1	10	2906	00:21:27
1000	10	10000	1	10	4223	00:32:42
5000	10	10000	1	10	5912	00:48:36
10000	10	10000	1	10	6949	01:25:19

Bayes iteracijos = 500						
Iteracijos	$X_{1(\min)}$	$X_{1(\max)}$	$X_{2(\min)}$	$X_{2(\max)}$	Taškai	Laikas
10	10	10000	1	10	3517	00:32:22
50	10	10000	1	10	3656	00:48:12
100	10	10000	1	10	3474	00:57:34
500	10	10000	1	10	3759	01:35:18
1000	10	10000	1	10	5893	02:32:38
5000	10	10000	1	10	8948	03:55:23
10000	10	10000	1	10	9813	05:48:55

GERIAUSI REZULTATAI, GAUTI ATLIKUS 100 BANDYMŲ						
Bayes iteracijos = 5						
Iteracijos	$X_{1(\min)}$	$X_{1(\max)}$	$X_{2(\min)}$	$X_{2(\max)}$	Taškai	Laikas
10	10	10000	1	10	3185	00:00:20
50	10	10000	1	10	3775	00:00:45
100	10	10000	1	10	4295	00:01:05
500	10	10000	1	10	4545	00:03:28
1000	10	10000	1	10	4900	00:05:49

5A lentelės pabaiga

Iteracijos	$X_{1(\min)}$	$X_{1(\max)}$	$X_{2(\min)}$	$X_{2(\max)}$	Taškai	Laikas
5000	10	10000	1	10	5430	00:08:55
10000	10	10000	1	10	6555	00:17:15

Bayes iteracijos = 50						
Iteracijos	$X_{1(\min)}$	$X_{1(\max)}$	$X_{2(\min)}$	$X_{2(\max)}$	Taškai	Laikas
10	10	10000	1	10	4020	00:02:15
50	10	10000	1	10	4555	00:04:53
100	10	10000	1	10	5025	00:12:15
500	10	10000	1	10	5900	00:21:27
1000	10	10000	1	10	7620	00:32:42
5000	10	10000	1	10	9700	00:48:36
10000	10	10000	1	10	13945	01:25:19

Bayes iteracijos = 500						
Iteracijos	$X_{1(\min)}$	$X_{1(\max)}$	$X_{2(\min)}$	$X_{2(\max)}$	Taškai	Laikas
10	10	10000	1	10	5700	00:32:22
50	10	10000	1	10	6550	00:48:12
100	10	10000	1	10	7470	00:57:34
500	10	10000	1	10	8755	01:35:18
1000	10	10000	1	10	9890	02:32:38
5000	10	10000	1	10	13945	03:55:23
10000	10	10000	1	10	19815	05:48:55

**6A lentelė.** Lokalaus determinuoto, lokalaus atsitiktinio, atkaitinimo modeliavimo ir Bayes metodų priklausomybių tyrimo rezultatai kai mokyklos euristiniai parametrai yra skirtingi

Bandymo Nr.	Skirtumas tarp pradinųjų ir optimalių baudos taškų	Kabinetų kiekis	Pamokų skaičius per dieną	Mokytojo lango įvertis	Pedagoginių didaktikų pažeidimas
<b>Lokalus determinuotas metodas (100 iteracijų)</b>					
1	6100	20	6	30	10
2	395	20	6	30	100
3	3055	20	6	300	10
4	1565	20	6	300	100
5	3595	20	7	30	10
6	2025	20	7	30	100
7	4115	20	7	300	10

6A lentelės tęsinys

Bandy- mo Nr.	Skirtumas tarp pra- dinių ir optimalių baudos taškų	Kabi- netų kiekis	Pamokų skaičius per dieną	Mokytojo lango įvertis	Pedagoginių didaktikų pažeidimas
8	385	20	7	300	100
9	1125	60	6	30	10
10	4985	60	6	30	100
11	2545	60	6	300	10
12	710	60	6	300	100
13	5915	60	7	30	10
14	2340	60	7	30	100
15	4325	60	7	300	10
16	115	60	7	300	100
<b>Lokalus atsitiktinis metodas (100 iteracijų; K=2)</b>					
1	2186	20	6	30	10
2	340	20	6	30	100
3	5580	20	6	300	10
4	300	20	6	300	100
5	4680	20	7	30	10
6	3640	20	7	30	100
7	5970	20	7	300	10
8	685	20	7	300	100
9	8040	60	6	30	10
10	2365	60	6	30	100
11	3080	60	6	300	10
12	11005	60	6	300	100
13	810	60	7	30	10
14	2135	60	7	30	100
15	3745	60	7	300	10
16	300	60	7	300	100
<b>Atkaitinimo modeliavimo metodas (100 iteracijų; X<sub>1</sub>=1000; X<sub>2</sub>=1)</b>					
1	7765	20	6	30	10
2	4750	20	6	30	100
3	6230	20	6	300	10
4	4595	20	6	300	100
5	5820	20	7	30	10
6	1475	20	7	30	100
7	5290	20	7	300	10
8	2735	20	7	300	100

6A lentelės pabaiga

<b>Bandy- mo Nr.</b>	<b>Skirtumas tarp pra- dinių ir optimalių baudos taškų</b>	<b>Kabi- netų kiekis</b>	<b>Pamokų skaičius per dieną</b>	<b>Mokytojo lango įvertis</b>	<b>Pedagoginių didaktikų pažeidimas</b>
9	2115	60	6	30	10
10	1875	60	6	30	100
11	415	60	6	300	10
12	4050	60	6	300	100
13	6890	60	7	30	10
14	6545	60	7	30	100
15	6905	60	7	300	10
16	1890	60	7	300	100
<b>Bayes metodas (100 AM iteracijų; 5 BA iteracijų; <math>X_1=10...10000</math>; <math>X_2=1...10</math>)</b>					
1	3005	20	6	30	10
2	2255	20	6	30	100
3	7615	20	6	300	10
4	1545	20	6	300	100
5	10115	20	7	30	10
6	3995	20	7	30	100
7	3925	20	7	300	10
8	1795	20	7	300	100
9	2355	60	6	30	10
10	1185	60	6	30	100
11	3465	60	6	300	10
12	5905	60	6	300	100
13	10900	60	7	30	10
14	4020	60	7	30	100
15	5065	60	7	300	10
16	1155	60	7	300	100



## B priedas. Naudojimosi instrukcija programa „Optima“

Programa galima naudotis dviem būdais: nuotoliniu ir instaliuojant programą savo kompiuteryje. Naudojantis nuotoliniu, naršyklės adreso laukelyje tereikia surinkti vieną adresą iš šio sąrašo:

- <http://pilis.if.ktu.lt:8090/TvarkaMano/>
- <http://kopustas.elen.ktu.lt:8080/TvarkaMano/>
- <http://optimum2.mii.lt:8080/TvarkaMano/>
- <http://soften.ktu.lt:8080/TvarkaMano/>

Turint vidutinio greičio neribotą internetą galima dirbti nuotoliniu būdu. Gautus rezultatus „zip“ archyvu visada galima parsisiųsti į savo kompiuterį programai baigus optimizavimo procesą. Kaip dirbti su pačia programa ir paruošti pradinių duomenų failą yra aprašyta „B priedo“ skyriuje „Darbas su programa“. Jei internetas yra labai lėtas, arba yra ribotas duomenų siuntimo kiekis, galima programą instaliuoti. Instaliavimo procesas aprašytas skyriuose „Java platformos instaliavimas“ – „Programos paruošimas darbui“ imtinai. Kaip įdiegti programą į kompiuterį geriausia žino mokyklos informacinių technologijų darbuotojas.

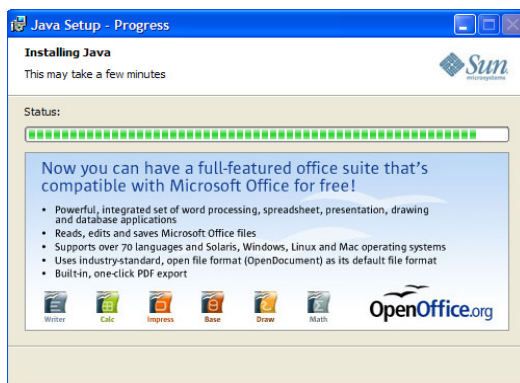
### Java platformos instaliavimas

Norint, kad programa veiktų stabiliai, reikia įdiegti naują JAVA „JRE“ platformą. Ją galite parsisiųsti iš puslapio [www.sun.com](http://www.sun.com) „Java Runtime Environment (JRE) 6“. Arba tiesiog paspausti nuorodą: <http://download.java.net/jdk7/binaries/>. Paleidus instaliacinį failą bus matomas java instaliavimo vedlys (1B pav.). Čia reikia paspausti mygtuką „Accept >“.



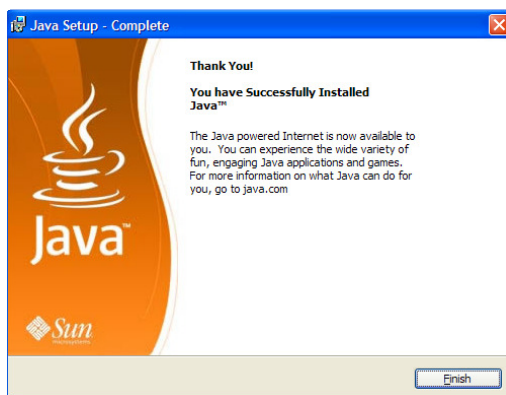
1B pav. Pirmasis žingsnis įdiegiant JAVA JRE

Reikia paspausti mygtuką „Accept >“.



2B pav. Veiksmai paspaudus „Accept“ mygtuką JAVA JRE įdiegimo metu

Programai baigus darbą reikia paspausti „Finish“ mygtuką.



3B pav. JAVA JRE diegimo pabaiga

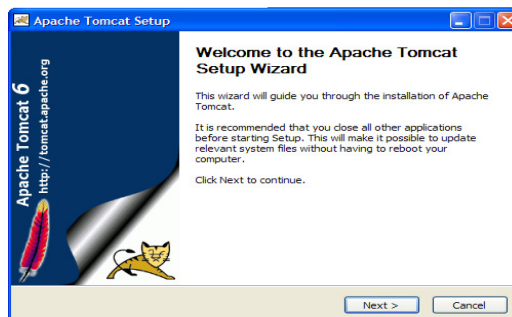
## Serverio paruošimas darbui su programa

### Serverio programos parsisiuntimas

Programai valdyti reikalingas serveris „Apache Tomcat“. Jį galima rasti puslapyje <http://tomcat.apache.org/>. Geriausia tinkama versija yra „Tomcat 6.x“. Parsisiųsti reikia versiją pavadinimu „Windows Service Installer“. Tai bus failas su plėtiniu „\*.exe“.

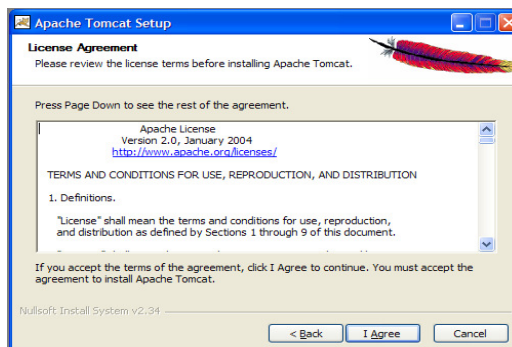
## Serverio programos įdiegimas

Įdiegti serverį reikia automatiškai. T. y. pagal jo prašomus parametrus (4B pav.):



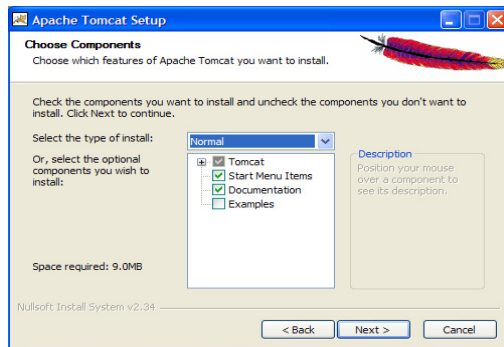
4B pav. Pirmas žingsnis įdiegiant „Tomcat 6.x“

Paleidus serverio diegimą, jūsų kompiuterio darbalaukyje bus matomas įdiegimo langas parodytas 4B pav. Čia reikia paspausti mygtuką „Next >“.



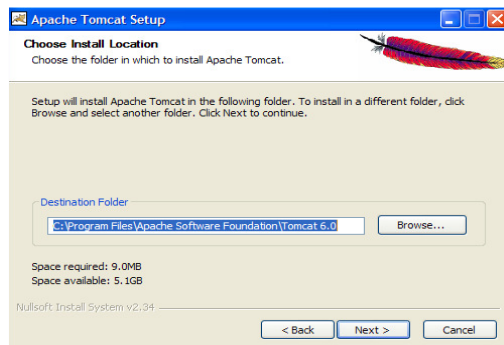
5B pav. Antras žingsnis įdiegiant „Tomcat 6.x“

Antrame žingsnyje reikia paspausti mygtuką „I Agree“.



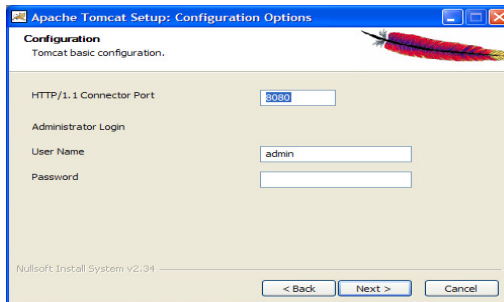
**6B pav.** Trečias žingsnis įdiegiant „Tomcat 6.x“

Trečiame žingsnyje reikia palikti bazinius nustatymus kaip parodyta 6B pav. ir paspausti „Next>“.



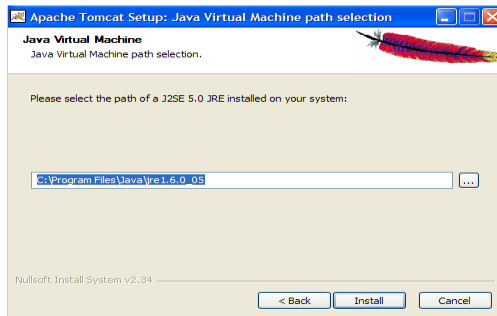
**7B Pav.** Ketvirtas žingsnis įdiegiant „Tomcat 6.x“

Ketvirtame žingsnyje galima nurodyti kur norite įdiegti savo serverį. Patartina palikti tokį pat adresą kaip nurodyta 7B pav. Jis būna nurodytas automatiškai. Taigi šiame žingsnyje geriausia paspausti mygtuką „Next>“.



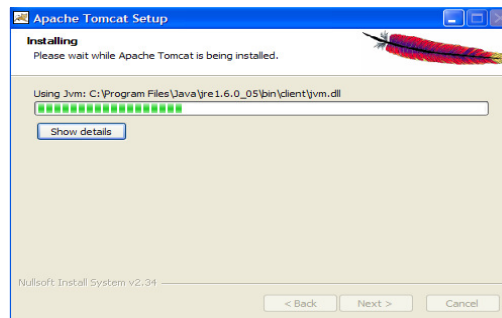
**8B pav.** Penktas žingsnis įdiegiant „Tomcat 6.x“

Penktame žingsnyje galima nurodyti serverio nustatymus. Geriausia palikti juo kaip parodyta 8B pav. ir paspausti mygtuką „Next >“

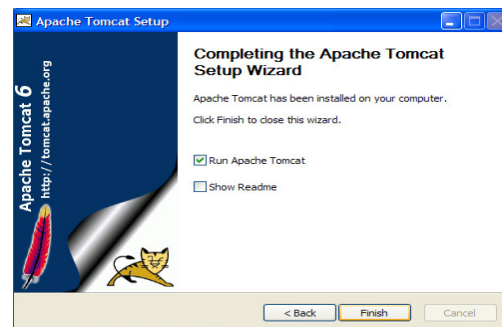


9B pav. Šeštas žingsnis įdiegiant „Tomcat 6.x“

Šiame žingsnyje reikia nurodyti kur yra JAVA platforma. Paprastai ji būna nustatoma automatiškai kaip parodyta 9B pav. Jei čia nėra nurodyta jokia direktorija, reikia ją nurodyti paspaudus ženkluką „...“ ir suradus tą direktoriją rankiniu būdu. Šiame žingsnyje reikia paspausti mygtuką „Install“.



10B pav. Serverio instaliavimas

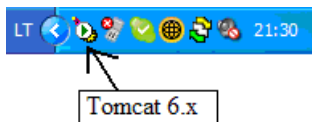


11B pav. Šeštas žingsnis įdiegiant „Tomcat 6.x“

Baigus įdiegti serverį reikia paspausti mygtuką „Finish“. Serveris įdiegtas. Dabar reikia sustabdyti veikiančią serverį, kad būtų galima įkelti tvarkaraščių optimizavimo programą.

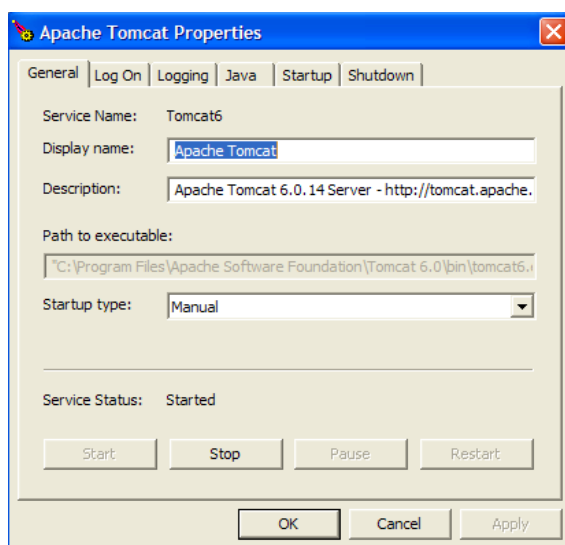
### Serverio programos paleidimas / stabdymas / perkrovimas

Baigus instaliaciją serveris automatiškai startuojamas. Jo valdymas dabar bus netoli laikrodžio (dešiniajame apatiniame darbalaukio kampe) ir atrodys taip kaip parodyta 12B pav.




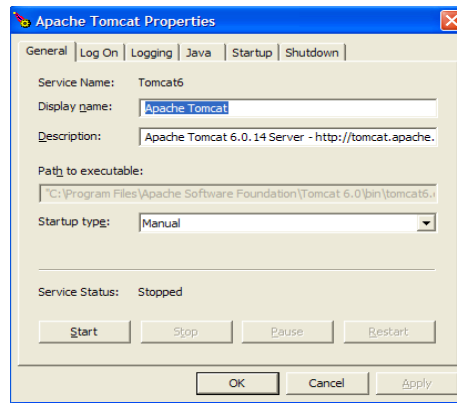
12B pav. Serverio valdymo ženklas

Dukart paspaudus pele ant šio ženklo pasirodo serverio valdymo langas (13B pav.).






13B pav. Serverio „Tomcat 6.x“ valdymo langas prieš sustabdant (darbo metu).

Norint sustabdyti serverį reikia paspausti mygtuką „Stop“. Tada serverio ženklas (esantis apatiniame dešiniajame darbalaukio kampe) tampa tokiu: . Serverio valdymo lange mygtukas „Start“ tampa aktyvus, o „Stop“ – neaktyvus (14B pav.).

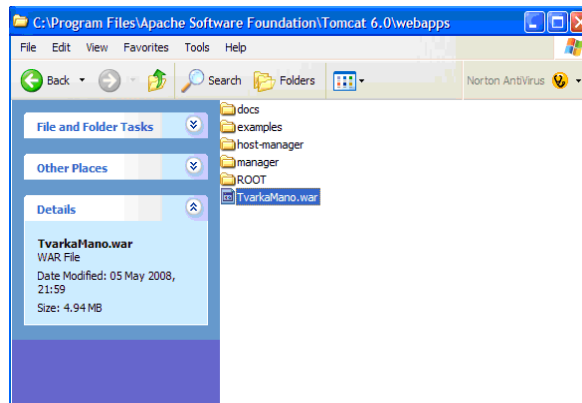


**14B. pav.** Serverio „Tomcat 6.x“ valdymo langas prieš paleidžiant (laukimo metu)

Norint perkrauti serverį užtenka jį sustabdyti („Stop“), palaukti kol valdymo ženklas pavirs iš „“ į „“ ir po kelių sekundžių vėl paleisti („Start“). Kai serverio valdymo ženklas pasikeis į „“ galima pradėti darbą su optimizavimo programa.

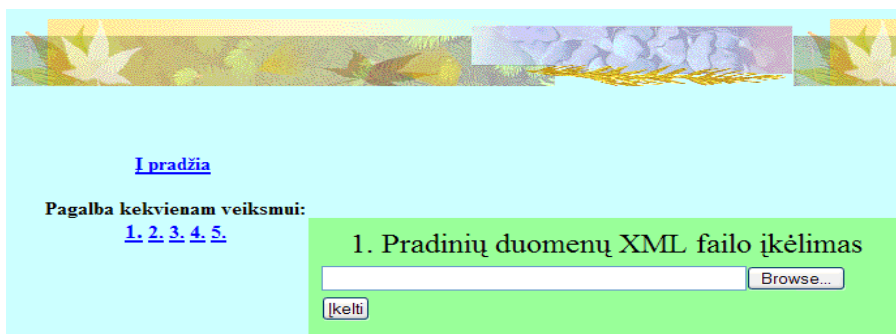
## Programos paruošimas darbui

Turimą tvarkaraščių optimizavimo programos failą „TvarkaMano.war“ reikia įkelti i valdantįjį serverio katalogą „webapps“. Šį katalogą rasite atlikę tokią veiksmų seką: C:/Program Files/Apache Software foundation/Tomcat 6.0/webapps. Į šį katalogą įklijuokite (perkeliate) jau turima programą „TvarkaMano.war“ (15B pav.).



**15B pav.** Programos paruošimas darbui

Nukopijavus programą reikia paleisti serverį (žiūrėti skyrių „Serverio programos paleidimas / perkrovimas / stabdymas“). Paleidus serverį, interneto naršyklės lange surinkite adresą: „http://localhost8080/ TvarkaMano“. Paspaudę „Enter“ išvysite langą atrodantį kaip pavaizduota 16B pav.



16B pav. Tvarkaraščių optimizavimo programos pradinis langas

- Paspaudus nuorodą „[I pradžia](#)“ jūs pateksite į dabar matomą puslapį. Tai naudinga kai programa baigia darbą, tačiau rezultatai jūsų netenkina ir jūs norite pakartoti eksperimentą.
- „Pagalba kiekviename žingsnyje“ yra pagalbos failai, kurie detaliai aprašo, ką reikia daryti, nurodyti atsivėrusiame lange.

## Darbas su programa

### Pirmasis žingsnis (pradinių duomenų failo paruošimas ir įkėlimas)

Pradinis duomenų failas sudaromas su MS Office programa MS Excel. Tačiau išsaugotas šis failas turi būti kaip XML failas. Išsaugodami nepamirškite nustatyti XML formato, arba naudokitės paruoštu šablonu. Toliau aprašysiu kaip paruošti pradinių duomenų failą (Excel knygą).

**1 lapas (Sheet 1)** – Šio darbo lapo pavadinimas turi būti „Dalykai“.

Šiame lape stulpelyje „A“ nurodomi **Dalykai**, o stulpelyje „B“ – alternatyvių disciplinų sąrašai (**Grupės**). **Dalykai** – tai visų mokykloje mokomų disciplinų trumpiniai. Jie turi būti surašyti vienas po kitu. Taip pat šie trumpiniai, tokiu pat pavidalu, bus vartojami pildant mokytojų lapą (Sheet2), bei pildant moksleivių pasirinkimų lenteles. **Gupės** – visos disciplinos turi būti skirstomi į tokias:



- „e“ – tai alternatyvių disciplinų sąrašas, iš kurio moksleivis gali nurodyti tik vieną discipliną. Pavyzdžiui: „e1“ sudaro trys disciplinos: Tikyba (T), Etika (E) ir Psichologija (Ps). Moksleivis gali nurodyti tik vieną iš šių disciplinų. Jei yra sudaromas kitas alternatyvių disciplinų sąrašas, jis pažymima raide „e“ su kitu numeriu.
- „but“ – tai būtinosios disciplinos, kurias moksleivis privalo nurodyti.
- „pas“ – tai disciplinos, kurias moksleivis gali rinktis kartu arba ne. Pavyzdžiui, moksleivis mokslo metais gali mokintis Chemiją (Chem.), Biologiją (Biol.), Geografiją (Geo.) ir Istoriją (Ist.) kartu arba tik Chemiją (Chem.) ir Biologiją (Biol.) ir t. t.

**Pavyzdys:**

	A	B	C	D
	Dalykai	grupės	prioritetai	Max mokinių sk.
1				
2	Ps	e1	1	30
3	T	e1	1	30
4	E	e1	1	30
5	Lk	but	5	25
6	A	e2	3	20
7	V	e2	3	20
8	R	e2	3	20
9	Pr	e2	3	20
10	A	e3	3	20
11	V	e3	3	20
12	R	e3	3	20
13	Pr	e3	3	20
14	Ist	pas	4	30
15	Geo	pas	4	30
16	Mat	but	5	30
17	Inf	pas	4	30
18	Biol	pas	4	30
19	Fiz	pas	5	30
20	Chem	pas	5	30
21	D	pas	1	30
22	M	pas	1	30
23	T	pas	1	30
24	S	pas	1	30
25	KK	but	3	30
26	lot	pas	2	30
27	isp	pas	2	30
28	gp	pas	2	30
29	br	pas	4	30
30	ek	pas	4	30
31	pb	pas	2	30
32	PC	pas	2	30

**17B pav.** Pradinių duomenų failo pirmojo lapo pavyzdys

**Vaizdus paaiškinimai:**

	A	B	C	D	E	F	G	H	I
	Dalykai	Grupės	Prioritetai	Max mokinių sk.					
1									
2	Ps	e1	1	30	Didžiausias mokinių skaičius grupėje, su kuriuo mokytojas gali dirbti pamokos metu				
3	T	e1	1	30		Prioritetai, kurie suteikiami daykui pagal sudėtingumą			
4	E	e1	1	30		Grupės, į kurias skirstomi mokykloje mokomi dalykai			
5	Lk	but	5	25		Dalykų, kurie yra mokomi mokykloje, pavadinimų trumpiniai			
6	A	e2	3	20					
7	V	e2	3	20					
8	R	e2	3	20					
9	Pr	e2	3	20					
10	A	e3	3	20	"e" - dalykų grupė, iš kurios galima pasirinkti tik vieną				
11	V	e3	3	20					
12	R	e3	3	20	"but" - dalykai, kuriuos mokinius privalo mokintis				
13	Pr	e3	3	20					
14	Ist	pas	4	30					
15	Geo	pas	4	30	"pas" - dalykų grupė iš kurios galima pasirinkti keletą dalykų				
16	Mat	but	5	30					
17	Inf	pas	4	30					
18	Biol	pas	4	30					
19	Fiz	pas	5	30					
20	Chem	pas	5	30					
21	D	pas	1	30					
22	M	pas	1	30					
23	T	pas	1	30					
24	S	pas	1	30					
25	KK	but	3	30					
26	lot	pas	2	30					
27	isp	pas	2	30					
28	gp	pas	2	30					
29	br	pas	4	30					
30	ek	pas	4	30					
31	pb	pas	2	30					
32	PC	pas	2	30					

**18B pav.** Pradinių duomenų failo pirmojo lapo struktūra

**2 lapas** Šio darbo lapo pavadinimas turi būti „Mokytojai“.  
**(Sheet2) Vaizdūs paaiškinimai:**

	A	B	C	D	E	F	G	H	I	J
1										
2	grupe	Eil.	Dalykas	Pavarde_Vardas	Kabinetas	Val.sk	Trump	Klases kuriom desto mokytojas	Pageidavimai / Laisvadieniai	
3			Pinas dalyko pavadinimas.	Mokytojo vardas ir pavardė.				Mokytojo pageidaujamų nedarbo dienų sąrašas.		
4				Kabineto, kuris priklauso mokytojui, numeris.				Klasė sąrašas, kurias moko mokytojas.		
5			Mokytojo eilės numeris					Dalyko pavadinimo trumpinys, kuris naudojamas knygos lape "Sablona", bei lape "Dayka".		
6										
7										
8			Vienodi arba tos pačios kategorijos dalykai, kurie gali būti apjungti į vieną grupę.							
9				Skaičius valandų, kiek mokytojas gali dirbti per savaitę.						
10										
11										
12										
13										

**19B pav.** Pradinių duomenų failo antrojo lapo struktūra

Šiame lape stulpelyje „A“ nurodomas **Grupės** (mokytojo, mokančio identišką ar alternatyvią discipliną, sąrašo numeris) numeris, stulpelyje „B“ – mokytojo sąrašo eilės numeris „Eil.“, stulpelyje „C“ – nurodomas pilnas disciplinos pavadinimas „Dalykas“, stulpelyje „D“ – nurodoma mokytojo **Pavardė Vardas**, stulpelyje „E“ – nurodoma mokytojui priklausančios patalpos – **Kabineto** (vienos ar kelių) numeris, stulpelyje „F“ – nurodomas mokytojo darbo valandų per savaitę skaičius „Val.sk.“, stulpelyje „G“ – nurodomas disciplinos trumpinys „Trump“, stulpelyje „H“ – visų klasių, kurias mokytojas moko, sąrašas „klasių, kuriom dėsto mokytojas“, stulpelyje „I“ – nurodomos dienos, kada mokytojas pageidauja turėti laisvą laiką „Pageidavimai / Laisvadieniai“.

Smulkesni nurodymai, kaip pildyti lentelės grafą:

- **Grupės** – stulpelis formuojamas vartotojo nuožiūra. Čia gali vartotojas mokytojus skirstyti taip kaip jam patinka: pagal disciplinas, pagal pageidavimus, pagal pavardes ir t. t. Šis stulpelis skirtas vizualiai paieškai pradinių duomenų faile. Programoje neturi jokios įtakos. Jis gali būti nepildomas. Svarbiausia išliktų pavadinimas stulpelyje „A“.
- **Eil.** – įrašomas mokytojo eilės numeris esamam sąrašė. Taip pat skirtas tik vizualiai mokytojo paieškai pradinių duomenų faile. Programoje įtakos neturi.
- **Dalykas** – nurodomas pilnas disciplinos pavadinimas kurią moko mokytojas. Pavyzdžiui: „Informatika“, „Matematika“ ir t. t. Jei mokytojas moko dvi disciplinas, šiame stulpelyje jas reikia abi paminėti atskiriant vieną nuo kitos kabliataškio pagalba. Pavyzdžiui: „Matematika; Informatika“, „Anglų; Vokiečių“, ir t. t. Visos disciplinos turi būti rašomi vienodai. Tai yra, jei keli mokytojai moko discipliną „Matematika“, tai visos disciplinos turi būti vienodai pavadintos (raidės turi būti vienodos visuose žodžiuose). Negalima rašyti vienam mokytojui „Matematika“, o kitam mokytojui „matematika“.
- **Pavarde Vardas** – šiame stulpelyje nurodoma mokytojo Pavardė ir Vardas. Pvz.: Pavardenis Vardenis.

- **Kabinetas** – nurodomas patalpos, kuri priklauso mokytojui, numeris. Jei mokytojas neturi mokykloje jam paskirtos patalpos, šiame Excel programos langelyje įrašomas nulis.
- **Val. sk.** – nurodomas maksimalus valandų skaičius, kurį mokytojas gali dirbti per savaitę. Parašius valandas, po skaičiaus, rašomas kabliataškis. Pavyzdžiui: „18;“. Jei mokytojas moko dvi disciplinas, tada nurodomos dvejų, viena po kitos sekančios valandos (pirmosios skirtos pirmai disciplinai, antrosios – antrai). Pavyzdžiui: „18;5;“.
- **Trump** – nurodomas mokomos disciplinos trumpinys. Pavyzdžiui: „Mat“, „Inf“ ir t. t. Šis trumpinys yra nurodomas lape „Dalykai“ ir visuose moksleivių klasių pasirinkimuose. Jei mokytojas dėsto dvi disciplinas jos išvardinamos atskiriant vieną nuo kitos kabliataškiu. Po paskutiniosios disciplinos taip pat turi būti rašomas kabliataškis. Pavyzdžiui: „Mat;Inf;“
- **Klasės, kuriom dėsto mokytojas** – išvardinamos klasės, kurias mokytojas moko. Kiekvienos klasės pavadinimas nurodomas ir kaip Excel knygos lapo pavadinimas. Išvardinant klases jos atskiriamos kabliataškiais. Pavyzdžiui: „7d; 2b; 2d;“ Jei mokytojas moko dvi disciplinas, tai iš pradžių išvardinamos pirmąją discipliną besimokančios klasės, po to rašomas „#“ ir išvardinamos klasės, kurios mokosi antrosios disciplinos. Pavyzdžiui: „12c; 11a; # 12a; 12b; 12e;“.
- **Pageidavimai / Laisvadieniai** – nurodomos savaitės dienos (skaičiais), kada mokytojas pageidauja turėti laisvas dienas. Pavyzdžiui „3;4;5;“, „5;“, „1;5;“ ir t. t.

## Pavyzdys:

	A	B	C	D	E	F	G	H	I
1									
2	grupe	Eil.	Dalykas	Pavarde_Vardas	Kabinetas	Val.sk	Tru	Klases kuriom dėsto mokytojas	Pageidavimai / Laisvadieniai
3	1	1	Etika; Tikyba	Martinaitienė Laura	101; #214;	30;30	E;T	11a; 11b; 11c; 11d; # 11a; 11b; 11c;	
4		2	Etika; Tikyba	Patamsienė Jolanta	101; #214;	30;30	E;T	11a; 11b; 11c; 11d; # 11a; 11b; 11c;	
5		3	Psichologija	Maculevičienė Nata	212; 213; 2	30;	Ps	11a; 11b; 11c; 11d; 11e; 11f; 11g;	1;2;3;
6	2	4	Leit.k.	Vasiliauskaitė Nijolė	212; 213; 2	35;	Lk	11a; 11b; 11c; 12a; 12b; 12c; 12d; 12e;	
7		5	Leit.k.	Lasauskienė Lilija	213; 212; 2	35;	Lk	11a; 11b; 11c; 12a; 12b; 12c; 12d; 12e;	
8		6	Leit.k.	Bundonienė Orianta	213; 212; 2	35;	Lk	12a; 12b; 12c; 12d; 12e; 12f;	
9		7	Leit.k.	Kalvelienė Laima	201;	35;	Lk	12a; 12b; 12c; 12d; 12e; 12f;	
10		8	Leit.k.	Kucinskiene Anzelis	202;	35;	Lk	11a; 11b; 11c; 12a; 12b; 12c; 12d; 12e;	
11		9	Leit.k.	Miliseviciute Sigita	203;	35;	Lk	11a; 11b; 11c; 12a; 12b; 12c; 12d; 12e;	
12		10	Leit.k.	Paradauskiene Asta	216; 219;	35;	Lk	12a; 12b; 12c; 12d; 12e; 12f;	
13		11	Leit.k.	Poskeviciute Dalia	212; 208;	35;	Lk	11a; 11b; 11c; 12a; 12b; 12c; 12d; 12e;	
14		12	Leit.k.	Ramoskiene Aukse	201; 207;	35;	Lk	11a; 11b; 11c; 12a; 12b; 12c; 12d; 12e;	
15		13	Leit.k.	Jaugiene Dona	212; 206;	35;	Lk	11a; 11b; 11c; 11d; 11e; 11f; 11g;	
16	3	14	Anglų	Zarauskiene Renatė	213; 218;	28;	A	11a; 11b; 12a; 12c; 12d; 12f;	1;4;5;
17		15	Anglų	Kruopiene Austra	202; 203;	28;	A	11a; 11c; 11d; 11g; 12a; 12c; 12f;	2;3;4;
18		16	Anglų	Bosulajeva Tajana	202; 203;	28;	A	11b; 11c; 11d; 11g; 12a; 12c; 12d; 12f;	
19		17	Anglų	Gontiene Austra	204;	28;	A	11e; 11f; 12a; 12c; 12d; 12f;	1;5;
20		18	Anglų	Januskeviciute Irena	205;	30;	A	11e; 12a; 12c; 12d; 12f; 12e;	
21		19	Anglų	Januskiene Birute	206;	28;	A	11c; 11d; 11f; 11g; 12b;	1;2;3;
22		20	Anglų	Kavalniene Elze	218;	23;	A	11c; 11d; 11g; 12a; 12c; 12d; 12f;	
23		21	Anglų	Liutkute Jurgita	219;	30;	A	12a; 12c; 12d; 12f;	1;2;5;
24		22	Vokiečių	Micinas Algis	221;	24;	V	11a; 11b; 11c; 11d; 11e; 11f; 11g;	

20B pav. Pradinių duomenų failo antrojo lapo pavyzdys

**3 lapas** Šio darbo lapo pavadinimas turi būti toks pat, kaip ir klasės.  
**(Sheet3)** Pavyzdžiui „11a“. Visi likusieji lapai turi būti skirti kitoms klasėms ir jų pageidavimams. Paskutinis lapas turi likti kaip šablonas. Jo pavadinimas turi būti „Šablonas“. Į klasės pageidavimų lapą galima nukopijuoti šabloną (iš lapo „Šablonas“). Taip sutaupysite laiko paruošdami klasės pageidavimų lentelę. Jei disciplinų trumpiniai sutampa – tada belieka tik išvardinti moksleivius ir jų pasirinkimus. Jei nesutampa – tada pakeisti trumpinius į kitokius. Primename, kad trumpiniai turi sutapti visuose knygos lapuose. Disciplinos, kurios priklauso kategorijai „e“, yra išvardinamos antroje eilutėje. Stulpelyje „Viso“ yra automatiškai skaičiuojamos moksleivio valandų kiekis per savaitę. Šio Excel programoje esančio langelio pildyti nėra būtina.

### Vaizdūs paaiškinimai:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1	Eil.	Pavarde	Dor.ugd	Leit.k	1 užs.kalba	2 užs.kalba	Ist	Geo	Mat	Inf	Biol	Fiz	Chem	Menai	K	Pasirenkami	Viso												
2			PsT	E	A	V	R	Pr	A	V	R	Pr									Mm	Da	Te	K	br	it	lek	sg	
3																													
4																													
5																													
6																													
7																													
8																													
9																													

21B pav. Pradinių duomenų failo trečiojo lapo struktūra

### Pavyzdys:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH
1	Eil.	Pavarde	Dor.ugd	Leit.k	1 užs.kalba	2 užs.kalba	Ist	Geo	Mat	Inf	Biol	Fiz	Chem	Menai	K	Pasirenkami	Viso																
2			PsT	E	A	V	R	Pr	A	V	R	Pr									D	M	T	S	K	lot	isp	gp	br	lek	pb	PC	
3	1	11e_Pavarde_1 Vardas_1	2		5	4									2	5	2		3	2	3			2								30	
4	2	11e_Pavarde_2 Vardas_2	2		5	4				3		5	1		4	2			4	2		4		2								32	
5	3	11e_Pavarde_3 Vardas_3	2		5	4				3		5	2		3				3		3			2								30	
6	4	11e_Pavarde_4 Vardas_4	2		5	4				2		5	2		3	2			3	2	3			2								31	
7	5	11e_Pavarde_5 Vardas_4		2	5	4				3	2	3			3				3		3			2								28	
8	6	11e_Pavarde_6 Vardas_5		2	5	4				2		5	2		3	2			3	2		3		2								31	
9	7	11e_Pavarde_7 Vardas_6	2		5	4				2		5	2		4	2			4	2	3		2									32	
10	8	11e_Pavarde_8 Vardas_7	2		4	4				2	2	5	1		3				2	3			2									30	
11	9	11e_Pavarde_9 Vardas_8	2		5	4				2	5	1			4	2			4	2		4	2									32	
12	10	11e_Pavarde_10 Vardas_9		2	5	4				2		5	2		4	2			4	2		3	2									31	
13	11	11e_Pavarde_11 Vardas_10	2		5	4				2		5	1		4	2			4	2		3	2									31	
14	12	11e_Pavarde_12 Vardas_11	2		5	4				2		5	2		3	2			3	2	3		2									31	
15	13	11e_Pavarde_13 Vardas_12	2		5	4			2		3		5	2	3				2	2		3	2									32	
16	14	11e_Pavarde_14 Vardas_13	2		5	4				2		5	2		3	2			3	2			4	2								32	
17	15	11e_Pavarde_15 Vardas_14		2	5	4				2		5	1		3	2			3	2		3	2									30	
18	16	11e_Pavarde_16 Vardas_15	2		5	4				2	2	5	1		3				2	3		2	3									31	
19	17	11e_Pavarde_17 Vardas_16		2	5	4				2		5	2		3				2	3		2										30	
20	18	11e_Pavarde_18 Vardas_17	2		5	4			2	3	2	3			3	2			3	2		3	2				1					32	
21	19	11e_Pavarde_19 Vardas_18	2		5	4			2		3		5	1		3	2		3	2		4	2									33	
22	20	11e_Pavarde_20 Vardas_19		2	5	4				3		5	1		4	2			4	2		3	2									32	
23	21	11e_Pavarde_21 Vardas_20	2		5	4				2	5	2			4	2	3		4	2	3		2									32	

22B pav. Pradinių duomenų failo trečiojo lapo pavyzdys

### Antrasis žingsnis (Objektyvių ir subjektyvių parametru nustatymas)

Nurodžius / pakeitus pradinius nustatymus pagal poreikius, reikia paspausti mygtuką „Nustatyti“ (23B pav.).

[I pradžia](#)

Pagalba kiekvienam veiksmui:  
[1.](#) [2.](#) [3.](#) [4.](#) [5.](#)

### 2. Pradinių parametų nustatymai

Kabinetų kiekis (mokykloje)	20
Didžiausias valandų skaičius per dieną	7
Baudų taškai: mokytojo langui	300
mokinio langui	5
mokytojo laisvadinių pažeidimui	400
išdirbtų valandų viršijimui	1700
pedagoginių didaktikų pažeidimui	100
<input type="button" value="Nustatyti"/>	

23B pav. Optimizavimo parametų nustatymas

Kiekviena mokykla objektyvius ir subjektyvius parametrus turi skirtingus. Jie skiriami atsižvelgiant į mokyklos dydį ir poreikius. Šiame programos lange reikia nustatyti šiuos parametrus:

#### Būtinai ribojimai:

- **Kabinetų kiekis (mokykloje)** – nurodo kiek maksimaliai patalpų gali būti užimta vienu metu.
- **Didžiausias valandų skaičius per dieną** – nurodo kiek norima turėti disciplinų per dieną.

#### Subjektyvūs parametrai:

- **Baudų taškai mokytojų tvarkaraštyje esančiam langui** – nurodo kiek bus skiriama baudos taškų, jei mokytojo tvarkaraštyje bus langas tarp dėstomų pamokų. Čia reikia nurodyti kiek yra skiriama baudos taškų už vieną mokytojo tvarkaraštyje esantį langą. Jei langų bus daugiau – programa pati padaugins šį skaičių iš susidariusių langų kiekio. Paprastai mokytojų yra žymiai mažiau nei moksleivių, todėl mokytojams skiriama daugiau baudos taškų kiekis už kiekvieną jų tvarkaraštyje esantį langą. Jei norima suformuoti tvarkaraštį moksleiviui, tai šis skaičius turi būti mažas (pvz. 10).
- **Baudų taškai moksleivių tvarkaraštyje esančiam langui** – nurodo kiek bus skiriama baudos taškų, jei moksleivio tvarkaraštyje bus langas. Kaip ir mokytojo taip ir moksleivio tvarkaraštyje esančiam langui įvertinti reikia nurodyti kiek yra skiriama baudos taškų už vieną moksleivio tvarkaraštyje esantį langą.
- **Baudų taškai mokytojų laisvadinių pažeidimui** – nurodo kiek bus skiriama baudos taškų, jei mokytojas turės dirbti per nepriimtinas darbo

dienas. Nepriimtinos darbo dienos yra nurodomos pradinių duomenų faile, lape „Mokytojai“. Kuo daugiau bus skiriama baudos taškų už šį pažeidimą tuo didesnė tikimybė, kad mokytojas nedirbs per jam nepriimtinas darbo dienas.

- **Baudų taškai išdirbtų valandų viršijimui** – nurodo, kiek baudos taškų bus skiriama, jei bus viršytas disciplinų skaičius per dieną. Pvz., jei nustatėm, kad per dieną turi būti tik 8 disciplinos, o programa suskirstė į 9, tada bus skiriami šioje grafoje nurodyti papildomi baudos taškai šiam tvarkaraščiui.

**Geriausias tvarkaraštis** bus išrenkamas iš daugybės suformuotų tvarkaraščių. Jis bus išrenkamas pagal baudos taškus. Geriausias tvarkaraštis – mažiausią baudos taškų skaičių turintis tvarkaraštis.

#### Parametrų nustatymo langas programos darbo metu.

24B pav. Optimizavimo parametrai nustatyti automatiškai

### Trečiasis žingsnis (Optimizavimo algoritmų nurodymas)

Nurodžius optimizavimo metodą reikia paspausti mygtuką „Pasirinkti“ (25B pav.).

25B pav. Optimizavimo metodo nurodymas

Šiame žingsnyje yra pasirenkamas vienas iš trijų optimizavimo algoritmų:

- **Lokalus** – tai pats paprasčiausias optimizavimo metodas, kuris visada keliauja tik į tikslesnį tvarkaraštį. Kitaip vadinamas lokaliu determinuotu metodu.
- **Lokalus atsitiktinis** – tai pagerintas lokalus metodas, kuris naudoja atsitiktinumus.
- **Simulated Annealing** – keliauja į tikslesnį, bet kartais nuklysdamas į blogesnį rezultatą. Tačiau visada parodo tik patį geriausią rezultatą. Kitaip yra vadinamas atkaitinimo modeliavimo metodu.
- **Bayes** – tai modeliuojamojo atkaitinimo metodo automatizavimas. Jis išrenka patį geriausią variantą, kuriuos surado modeliuojamasis atkaitinimas.

Apie kiekvieną algoritmą daugiau aprašyta ketvirtame žingsnyje.

### Ketvirtasis žingsnis (optimizavimo parametrų nustatymas)

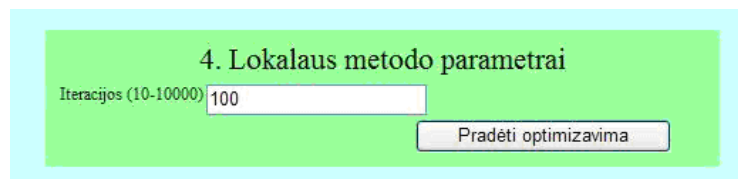
Nurodžius optimizavimo metodą (algoritmą), reikia nustatyti metodo parametrų reikšmes. Kiekvieno metodo parametrų reikšmės aprašomos tolesniuose skyreliuose atskirai.

#### Lokalus (Lokalus determinuotas) metodas:

Šiame metode reikia nurodyti tokius parametrus:

- **Iteracijos** – kiek kartų bus atliekama tikslesnio tvarkaraščio paieška.

Skiausteliuose yra nurodomos kokios gali būti minimalios ir maksimalios kiekvieno parametro reikšmės.



26B pav. Lokalaus determinuoto optimizavimo metodo parametrai

#### Lokalus atsitiktinis metodas:

Šiame metode reikia nurodyti tokius parametrus:

- **Iteracijos** – kiek kartų bus atliekamas tikslesnio tvarkaraščio paieška.
- **Jei nepagerėjo K kartų** – kiek kartų programa sukurs atsitiktinę duomenų seką, kurią po to optimizuos

Skiausteliuose yra nurodomos kokios gali būti minimalios ir maksimalios kiekvieno parametro reikšmės.

27B pav. Lokalaus atsitiktinio optimizavimo metodo parametrai

### Simulated Annealing (Atkaitinimo modeliavimo) metodas:

Šiame metode reikia nurodyti tokius parametrus:

- **Iteracijos** – kiek kartų bus vykdomas metodas.
- **Pradinė temperatūra  $X_1$**  – šis skaičius yra naudojamas formulėje, tikimybei pereiti į blogesnę variantą, skaičiuoti.
- **Šalimo greitis  $X_2$**  – nurodo kaip stipriai ieškoti tikslesnio varianto.

Skiausteliuose yra nurodomos kokios gali būti minimalios ir maksimalios kiekvieno parametro reikšmės

28B pav. Atkaitinimo modeliavimo optimizavimo metodo parametrai

### Bayes metodas:

Šis metodas automatizuoja atkaitinimo modeliavimo (AM) metodą. Čia reikia nurodyti kiek iteracijų turės pats AM metodas „**Iteracijos metodui SA**“, ir kiek kartų jis bus kartojamas „**Iteracijos metodui Bayes**“. SA kartojamas kiekvieną kartą vis su naujais parametrais, kuriuos programa parenka pagal užduotus intervalus. Intervalų duomenys atitinka AM metode aprašytus duomenis „ $X_1$ “ ir „ $X_2$ “. Tačiau čia yra nurodomos jų minimalios ir maksimalios reikšmės. Intervalų parametrai:





Po tvarkaraščių yra nurodoma baudos taškai prieš optimizavimą ir naujo (geriausio) tvarkaraščio baudos taškai.

Paspaudus nuorodą „**Parsisiųsti ZIP archyve supakuotus rezultatus**“ bus siūloma kompiuteryje išsaugoti visa pradinių duomenų ir gautų rezultatų archyvą. Jame atskiruose kataloguose yra saugomi moksleivių bei mokytojų tvarkaraščiai (mokytojų – „Initial teachers“ bei „Optimized teachers“, moksleivių – „Initial students“ bei „Optimized students“) ir papildomos naudingos suvestinės (Mokyklos tvarkaraštis, patalpų tvarkaraštis ir kiti).

[I pradžia](#)

Pagalba kiekvienam žingsniui:  
[1.](#) [2.](#) [3.](#) [4.](#) [5.](#)

### 5. Optimizavimo rezultatai

#### Mokyklos tvarkaraštis

Optimalus mokyklos tvarkaraštis

	Pirmadienis	Antradienis	Trečiadienis	Ketvirtadienis	Penktadienis
1	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, k.k., Mat, k.k., k.k., k.k., k.k., k.k., k.k., Ps, T, E, A, A, A, A, A, A, A, A, A, A	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, T, k.k., k.k., k.k., k.k., k.k., k.k., T, E, A, A, A, A, A, A, A, A, A, A	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, k.k., k.k., k.k., k.k., k.k., k.k., T, T, E, E, A, A, A, A, A, A, A, A, A, A	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, k.k., k.k., k.k., k.k., k.k., k.k., k.k., k.k., T, T, E, E, A, A, A, A, A, A, A, A, A, A	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, k.k., k.k., k.k., k.k., k.k., k.k., T, T, E, E, A, A, A, A, A, A, A, A, R, Pr
2	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat,	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat,	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat,	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat,	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat,

**31B pav.** Rezultatai, rodomi programai baigus darbą

Yra dar keletas kitų galimybių. Norit peržiūrėti atskirus moksleivių, mokytojų, patalpų, grupių tvarkaraščius nėra būtina siųsti visa archyvą. Galima peržiūrėti gautus rezultatus ir tik tada nuspręsti reikia parsisiųsti archyvą ar ne. Norint peržiūrėti kitokius tvarkaraščius, reikia paspausti su pelyte ant nuorodos „**Mokyklos tvarkaraštis**“. Atsivers štai toks langas (32B pav.):

### 5. Optimizavimo rezultatai

[Parsisiųsti ZIP archyve supakuotus rezultatus](#)

Tvarkaraščiai

Savininkas

**32B pav.** Rezultatų rodymo nurodymai

Laukelyje „**Tvarkaraščiai**“ galima pasirinkti kokį tvarkaraštį norite matyti:

- „**Pradinis**“ – parodys nurodytą tvarkaraštį suformuotą prieš pradendant optimizuoti duomenis.
- „**Optimalus**“ – parodys nurodytą tvarkaraštį, suformuotą po optimizavimo.
- „**Pradinis ir optimalus**“ – parodys nurodytą tvarkaraštį suformuotą prieš pradendant optimizuoti duomenis, bei tą patį tvarkaraštį po optimizavimo.

Laukelyje „**Savininkas**“ – galima nurodyti kieno tvarkaraštį norite matyti:

- „**Mokykla**“ – rodo visos mokyklos tvarkaraštį.
- „**Grupė**“ – rodo nurodytos klasės tvarkaraštį.
- „**Moksleivis**“ – rodo nurodyto moksleivio tvarkaraštį.
- „**Mokytojas**“ – rodo nurodyto mokytojo tvarkaraštį.
- „**Kabinetas**“ – rodo konkrečios patalpos užimtumą.

Paskutinis laukelis atsiranda tik „**savininkui**“ nurodžius:

- **Grupę** – visų galimų klasių sąrašas. Jis nuskaitomas iš pradinių duomenų failo pagal Excel knygos lapų pavadinimus.
- **Moksleivį** – visų galimų moksleivių sąrašas.
- **Mokytoją** – visų galimų mokytojų sąrašas.
- **Kabineta** – visų galimų patalpų sąrašas.

Pavyzdžiui, nurodžius laukelyje „**Tvarkaraščiai**“ – „**Pradinis ir optimalus**“, laukelyje „**Savininkas**“ – „**Mokytojas**“, o atsiradusiame mokytojų sąrašo nurodžius konkretų mokytoją ir paspaudus klavišą „**Peržiūrėti**“ buvo gauti rezultatai parodyti 33B pav.

5. Optimizavimo rezultatai  
Mokytojo Maciulevičienė Natalija tvarkaraštis

Pradinis tvarkaraštis					
	Pirmadienis	Antradienis	Trečiadienis	Ketvirtadienis	Penktadienis
2				T	T
3	T	T	T	T	

Optimalus mokyklos tvarkaraštis					
	Pirmadienis	Antradienis	Trečiadienis	Ketvirtadienis	Penktadienis
2				T	T
3	T	T	T	T	

Baos taškai sudarius pradinį tvarkaraštį: 14,140, baudos taškai po tvarkaraščio optimizavimo: 13,860  
[Parsisiųsti ZIP archyvą supakuotus rezultatus](#)

**33B pav.** Konkretaus mokytojo pradinis ir optimalus tvarkaraščiai

Atminkite, Jūs gausite individualius rezultatus pagal Jūsų mokyklą ir joje dirbančius mokytojus bei moksleivius.

Lina PUPEIKIENĖ

OPTIMIZAVIMO METODŲ TYRIMAS IR TAIKYMAS PROFILIUOTŲ MOKYKLŲ  
TVARKARAŠČIŲ SUDARYMO UŽDAVINIUOSE

Daktaro disertacija

Technologijos mokslai,  
informatikos inžinerija (07T)

INVESTIGATION AND APPLICATION OF PROFILED SCHOOL SCHEDULING  
TASKS OPTIMIZATION METHODS

Doctoral Dissertation

Technological Sciences,  
Informatics Engineering (07T)

lina.pupeikiene@gmail.com

2009 04 06. 11,75 sp. I. Tiražas 20 egz.  
Vilniaus Gedimino technikos universiteto leidykla „Technika“,  
Saulėtekio al. 11, 10223 Vilnius,  
<http://leidykla.vgtu.lt>  
Spausdino UAB „Biznio mašinų kompanija“,  
J. Jasinskio g. 16A, 01112 Vilnius