TURINYS:

Pratarm 3
Santrauka 4
1 vadas 6
2 Termin odynas 7
3 Optimalus tvarkara i sudarymas. SA ir GMJ teorinis tyrimas 8
3.1 SA ir GMJ metodai 8
3.2 Minimizavimas naudojant sistemoje diegtus metodus 9
3.3 Tvarkara i sudarymas 11
3.3.1 Euristikos pritaikymas tvarkara i optimizavime 12
3.3.2 Mokyklos tvarkara i sudarymas 13
3.3.3 Tvarkara i sudarymo apra ymas 14
3.3.4 Tvarkara i sudarymo ypatumai 15
3.3.5 Pagrindiniai reikalavimai tvarkara i sudarymui 15
3.3.6 Duomen sudarymo s lygos 16
3.3.7 Baudos ta k skai iavimas 16
3.3.8 Optimizavimas kaitaliojant laukus 18
3.3.9 Pradinio tvarkara io sudarymas 19
3.3.10 Tvarkara io optimizavimo apra ymas 19
3.3.11 Kaitaliojimo ir vertinimo algoritmo apra ymas 20
3.4 Duomen strukt ra 22
3.5 Programin realizacija 24
3.5.1 Apleto versija 24
3.5.2 Sistemos programin realizacija 24
4 Modeli eksperimentinis tyrimas 26
4.1 Tyrimo metod apra ymas 26
4.2 SA gauti steb jim rezultatai 27
4.3 GMJ gauti steb jim rezultatai 28
5 Gaut rezultat palyginimas su kitais 38
5.1 Gauti rezultatai 38
5.2 Gaut rezultat palyginimas 39
5.3 „MIMOSA“ ir „aSc Timetables 2003“ 44
5.4 „RECTOR“ 44
6 Programin s rangos realizacija 45
6.1 Sistemos funkcinis apra ymas 45

6.2 Vartotojo atmintin 45
6.3 Detalioji sistemos atmintin 45
6.3.1 Serverio startavimas 46
6.4 Programos startavimas: 46
6.4.1 Pradinis programos langas: 47
6.4.2 Programos meniu “help” funkcija 48
6.4.3 Optimizavimo programos pasirinkimas 48
6.4.4 Duomen vedimas 49
6.4.5 Optimizavimo kriterij pasirinkimas 50
6.4.6 Optimizavimo parametr parinkimas 52
6.4.7 Pasirinkto mokytojo tvarkara iai 54
6.5 Pasirinkto mokinio tvarkara iai 55
6.5.1 Mokyklos tvarkara iai: 56
6.6 Sistemos instaliavimas 57
6.6.1 Programin s rangos instaliavimas 57
6.6.2 Serverio “Tomcat” paleidimas ir i jungimas 59
6.6.3 Operacin s sistemos reikalavimai 59
7 I vados 60
Literat ros s ra as 61

Pratarmë

Optimali tvarkara i sudarymo sistema sprend ia labai sud ting , ta iau teoriniu po i riu dom ir svarb bei prakti k u davin . Sistemoje yra diegta daugelis vairi optimizavimo metod , kuri kiekvienas skirtingai sprend ia tvarkara i optimizavimo u davin . Moksliniu po i riu i sistema yra labai naudinga tiriant kiekvien optimizavimo u davin , nes galima palyginti gautus rezultatus u duodant u daviniams tas pa ias s lygas, t. y. vedus vienodus duomenis, nusta ius vienodus parametrus ir kt. Taip pat i sistema leid ia ne tik atlikti mokslinius tyrimus, bet ir eiliniam tvarkara i sudarin tojui matyti pradinius ir suoptimizuotus rezultatus. Sistema suprantamoj formoj i veda ekran bendr bei individualius mokytoj tvarkara ius. Jei reikia, naujai persirinkus kriterijus ji greitai i naujo suoptimizuos pradinius duomenis ir parodys naujai gautus rezultatus. Net ir taikant eilin je mokykloje i programa mano nuomone yra prakti kesn nei programos „MIMOSA“, „aSc Timetables 2003“ ir „RECTOR“. Ji nereikalauja dideli vartotojo pastang , mok jimo dirbti kompiuteriu ir daug laiko susipa inti su programos darbu, o gautas tvarkara tis yra pakankamai geras. ia programa tvarkara tis sudaromas palyginti labai greitai ir jame patogiai matomi pilnas dalyko pavadinimas ir klas s numeris. Gauti rezultatai tvarkingai i d stomi vienas po kitu. Programa „RECTOR“ yra labai elementari, nes joje tvarkara ius reikia d lioti rankiniu b du. Programa „aSc Timetables 2003“ tvarkara ius sudarin ja ma daug kaip ir programa „MIMOSA“. Ji tvarkara t suoptimizuoja, bet ne visada sudaro piln tvarkara t . Jei klas s ne profiliuotos, tada ji sudaro piln tvarkara t . Jei klas s profiliuotos, tai ka kuriuos pogrupius ir j paskaitas reikia sud lioti tvarkara t rankiniu b du. Tai n ra patogu kaip ir programose „MIMOSA“. Visos pamin tos programos yra licencijuojamos Lietuvoje.

Santrauka

iame darbe realizuotas SA – Simulated Annealing („Imituojan io atau im “) metodo algoritmas ir GMJ – Global Minimizer for Java („Globalaus minimumo paie ka, realizuota Java kalba“), t. y. taip vadinami optimizavimo u davini sprendimo r mai.

SA – metodas prasmingumu traukia d mes kaip metodas, tinkantis spr sti daugel problem . Praktiniuose pritaikymuose SA yra efektyviai pritaikytas garsiame „keliaujan io pardav jo“ u davinyje. is metodas yra sukurtas remiantis analogi ku termodinamikoje naudojamu metodu, kuris yra skirtas konkretaus skys io arba i lydyto (kaitinto) metalo alimui ir kristalizavimuisi. Proceso esm yra l tas alimas, naudojant pakankamai laiko tam, kad perskirstyti judrumo netenkan ius elementus. Tai techninis „Annealing“ apibr imas. Taip apibr us metod yra garantuojama, kad net ir tuo atveju, kai atsiras blogiai i d stytas variantas, jis taip pat bus i nagrin tas.

GMJ – tai globalaus optimizavimo programin s rangos vadinamieji optimizavimo u davini sprendimo r mai. is metodas realizuoja:

- globalius optimizavimo algoritmus (metodai),
- funkcijas, kurios turi b ti optimizuotos (u duotys),
- objektus, kurie turi b ti i vedami ekran (analiz).

Pasinaudojus iais darbo metu realizuotais iais modeliais, buvo atlikti i sam s tyrimai: i analizuota mokytoj , moksleivi skai iaus, ir kit vairi ribojim taka optimali tvarkara i sudarymui. Tyrimui buvo naudoti Marijampol s kolegijos tvarkara iui sudaryti skirti duomenys.

Atlikta teorinis vairiais metodais sudaryt optimali tvarkara i palyginimas su kit firm programom ir sudarytais tvarkara iais.

In this work was created algorithm of SA – Simulated Annealing method and GMJ – Global Minimizer for Java. GMJ it's solution of so called frames of optimised tasks.

The method of SA is a technique that has attracted significant attention as suitable for optimization problems of large scale. For practical purposes, simulated annealing has effectively "solved" the famous travelling salesman. At the heart of the method of simulated annealing is an analogy with thermodynamics, specifically with the way that liquids freeze and crystallize, or metals cool and anneal. So the essence of the process is slow cooling, allowing ample time for redistribution of the atoms as they lose mobility. This is the technical definition of annealing, and it is essential for ensuring that a low energy state will be achieved.

GMJ – it's a software solution of so called frames of global optimisation tasks. This method realizes:

- algorithms of global optimisation (methods),
- functions, that's can be optimised (tasks),
- objects, that's can be showed on display (analysis).

After the program was created, exhaustive inquiry of in this work realized methods was done: analysed influence of teachers, students and others different restrains multipliers to school scheduler making. The research was done using the data of college of Marijampol .

There was accomplished theoretical and experimental comparison of SA and the school schedule programs of others firms.

1 vadas

io tiriamojo darbo tikslas yra suprojektuoti profiliuot mokykl pamok tvarkara i sudarymo ir optimizavimo sistem . Dirban io autenti ka optimizavimo sistema tvarkara i sudarin tojo tikslas yra tik bendr ir priverstini dalyk pradini duomen suvedimas. Abeji pamok tvarkara iai, tiek bendras, tiek individualus, yra automati kai sudaromi optimizavimo sistemos. ie tvarkara iai gali b ti koreguojami tvarkara i sudarin tojo atsi velgiant ka kuriuos papildomus faktorius, kuri n ra nei pagrindiniame bendr , nei priverstini dalyk pradiniuose duomenyse. Pamok tvarkara i sudarin tojas gali daryti tak optimizavimo sistemos gautiems rezultatams keisdamas pradinius duomenis.

iuo metu Lietuvoje did ioji dalis mokykl yra profiliuotos. Taip yra tod l, kad ank iau mokykl mokymo modeliai vert visus vaikus mokytis visko. Specializuotis i esm s vaikams tekdavo tik papildomo laiko ir papildomo kr vio s skaita. iuo metu yra atsi velgiama kiekvieno vaiko norus bei sugeb jimus. Taip yra suma intas privalom dalyk skai ius. Taigi vaikas gali pasirinkti daugiau sau patinkan i ir jo pom gius bei sugeb jimus tenkinan i dalyk . Kadangi mokykloje yra daug vaik , kurie renkasi skirtingus dalykus, i kyla tvarkara i sudarymo problema. i problema tuo sud tingesn , kuo daugiau pasirinkimo laisv s duodama mokiniams ir kuo daugiau yra tenkinami j poreikiai.

Daugelis moni stengiasi palengvinti darb ir sukurti prakti kas bei lengvai valdomas tvarkara i sudarymo ir optimizavimo programas. Per pastaruosius kelet met Lietuvoje teko susipa inti su keletu tvarkara i sudarymo program . Pla iausia paplit yra tokios programos:

1. MIMOSA – kuri i leido kompanija „Mimosa software OY:
2. RECTOR – kuri i leido „(и т мы- г ммы-(ви ;
3. aSc Timetables 2003 – kuri i leido „Applied Software Consultants s.r.o“;

Autoriai, kurie n ra tekste pamin ti prie citat :

8 psl. Cohn H, Fielding M., ir Abboud N, Sakawa M, Inuiguchi M.;

12 psl. Kaneko K, Yoshikawa M, Nakakuki Y.;

13 psl. Carter MW, Laporte G. ir Schaerf A.;

15 psl. SELIM SM.;

24-25 psl. Schaerf A. ir Meisels A, EllSana J, Gudes E.;

2 Termino žodynas

Pranešimas – konkreti informacijos išraiška.

Duomenys – tai konkreti koduota informacija.

Programa tai algoritmai, kuriuos supranta ir gali atlikti kompiuteris.

Servletas – programa parašyta Java kalba, kuri galima įdėti į WWW puslapį ir vykdyti išnutolusio kompiuterio.

Apletas – Tai Java programa, kuri yra išvykdoma per internetą, bet skaičiavimus vykdo vartotojo personaliniame kompiuteryje;

HTML – hiperteksto formatavimo kalba skirta WWW puslapių kūrimui.

Java – aukšto lygio programavimo kalba, skirta rašyti programoms dirbančioms tinkle ir palaikomoms bet kokios platformos kompiuterių.

WWW – pasaulio kompiuterinis tinklas.

JDK – JAVA Developmet Kit– Java programinės įrangos kūrimo bibliotekos;

Profiled school – profiliuota mokykla;

Stand-alone – Tai Java programa, kuri dirba nepriklausomai nuo kitų programų;

SA – Simulated Annealing;

GMJ – Global Minimizer for Java;

OS – Operacinė Sistema;

Longer-Job – tai ilgesnio darbo euristika: programa darbą pradeda nuo didžiausio užduočių variantoir vis pereina prie mažesnio;

3 Optimalus tvarkaraščio sudarymas. SA ir GMJ teorinis tyrimas

3.1 SA ir GMJ metodai

SA metodas patrauk d mes , kaip metodas, kuris tinka didel ms optimizavimo problemoms spr sti. Labiausia jis tinka ten, kur reikia rasti global maksimum i daugelio maksimum gaut optimizavimo intervaluose. Praktiniuose pritaikymuose SA yra efektyviai pritaikytas garsiame „keliaujan io pardav jo“ u davinyje. Taip pat nagrin jama galimyb , kaip panaudoti metod kosmose. is metodas tapo daug sud tingesniu nei kombinatorinis nuo tada, kai paveld tos problemos v l pasirei k . SA turi atsitiktinius bandym ingsnius. D l to yra b tinos kai kurios papildomos s lygos.

io metodo pagrindas yra analogi kas termodinamikoje naudojamam metodui, skirtam konkre iam skys iui arba i lydytam (kaitintam) metalui alti ir kristalizuotis. Termodinamikoje visi elementai, esantys skystu pavidalu, juda laisviau vienas kito at vilgiu. Jei skystis la l iau, terminis judrumas ma ja i l to. Atomai gali da niau susidurti vienas su kitu. Taip kristalo pavidalas tvarkingai i d stomas i daugelio toki susidurian i atom . Proceso esm yra l tas alimas, naudojant pakankamai laiko tam, kad perskirstyti judrumo netenkan ius elementus. is kristalas rei kia, kad sistema yra minimalios energijos b kl s. Stebinantis faktas yra tas, kad l tai lanti sistema, nat raliai priart ja prie minimali energij turin ios b kl s. Bet jei skystis arba skystas metalas la greitai arba „staiga atau ta“, tai atomai i sid stys chaoti kai, nesudarys tvarkingo kristalo, i b kl bus nepasiekiama, bet iek tiek priart s beform s, turin ios iek tiek daugiau energijos nei nat raliu b du lantis metalas, b kl s.

Taigi SA metodo esm yra l tas „ alimas“ atsi velgiant pakankam laiko kiek , kad perskirstyt tvarkara i variantus pagal j optimal i sid stym . Tai techninis SA apibr imas ir tai svarbus garantas, kad net ir tuo atveju, kai atsiras blogiausias variantas, jis taip pat bus i nagrin tas.

Nors analoginis termodinamikos metodas n ra tobulas, bet ia yra prasm , kad is minimizavimo algoritmas, skirtingai nei kiti, vykdo l t „ alim “. Visais atvejais neliko tro kimo greitai surasti sprendinius: nuo prad ios ta ko taip greitai per i r ti visus variantus, kiek tai manoma. Tai pirmauja skai iuojant lokal , bet neb tinai global , minimum . Nat raliai atsirad s SA minimizavimo algoritmas yra sudarytas i dviej visi kai skirting proced r . is algoritmas yra kil s Boltzmanno tikimyb s padalinimo, ir atrodo taip:

$$\text{Prob}(E) \sim \exp(-E/kT) \qquad (3.1)$$

Pritaikius id j , kad sistema i laiko ilumos pusiausvyr prie temperat ros T ir i laiko energij , tikimybi kai i skirt i skirting energijos b sen E. Taip pat yra labai ma as ansas, kad net ir esant

emai temperat rai sistema tur s pakankamai energijos. Tod l sistema yra sugalvota taip, kad i gauti i lokalaus energijos minimumo geresn ir daugiau globali reik m . Dydis k (Boltzmanno konstanta) yra konstanta, kuri susieja temperat r ir energij . Kitais od iais tariant, sistemos temperat ra kartais pakyla lygiai taip pat, kaip ir nusileid ia; bet temperat ros em jimas yra ma iau tik tinas nei svarbus jos did jimo nukrypimas.

1953 m. Metropolis pirmasis registravo ias pagal skai iavimus teori kai gimiringas id jas. S kmingai pasirinktas pasi lymas dirbtinai pritaik termodinamin sistem ir pakeit jos energijos E konfig racij nuo E_1 iki E_2 su tikimybe $p = \exp[-(E_2 - E_1) / kT]$. sp jimas: jei $E_2 < E_1$, tai i tikimyb yra didesn nei nauda; tokiais atvejais atsitiktiniai pasikeitimai nustatomi kaip tikimyb $p = 1$. Sistema visada ie kos pagal tokius nustatymus. i pagrindin schema, kai visada po ingsn ie koma geresnio varianto, kartais suras ir blogesn variant . O sistema blogesn variant suras tuo re iau, kuo tikimyb iam variantui surasti bus blogesn . Visa tai inoma ir pavadinta kaip „Metropolito algoritmas“.

Kad sudaryti Metropolito algoritm kitoms termodinamin ms sistemoms reikia vykdyti tokius reikalavimus:

1) turi b ti sistemos konfig racij galimybi apra ymas ;
2) prie konfig racijos turi b ti atsitiktini pasikeitim generatorius; ie pasikeitimai yra SA sistemos „pasirinkimai“;
3) tikslo funkcija E (energijos analogas) kurios tikslas – minimizavimas;
4) valdymo parametras T (temperat ros analogas) ir geriausio tvarkara io i rinkimas, kuris nurodo, kaip suma inamas vis variant skai ius; po to kiek atsitiktini konfig racijos pakeitim T kiekviename ingsnyje buvo atlikta ir kiek tvarkara io sudarymo u duo i reik jo fizi kai atlikti ir/arba kiek j nebuvo atlikta ar jos vertintos kaip klaidos.

3.2 Minimizavimas naudojant sistemoje ádiegtus metodus

ioje optimizavimo sistemoje yra u programuoti trys optimizavimo metodai:

1. Profiled school,
2. SA,
3. GMJ.

1. „Profiled school“ metodas yra pats papras iausias. Jis minimaliai optimizuoja probleminius u davinius ir ie ko funkcijos f(x) lokalaus minimumo. iuo metodu dirbdama sistema daro minimalius pertvarkymus, kuriuos pasirenka pagal tam tikrus atsitiktinumo elementus. Tai ne tik io metodo, bet taip pat pla iai naudojam komercini sistem „MIMOSA“ ir „RECTOR“ pagrindas.

2. SA metodas yra sud tingesnis. Norint iuo metodu gauti gerus rezultatus reikia tur ti daug laiko. Pagrindin SA optimizavimo id ja yra pritaikyti ir optimizuoti probleminius u davinius, kuriuos sudaro N kintam j t. y. rasti (idealiu atveju global) minimum ka kokiai funkcijai f(x), kuri sudaro daugelis lokali minimum ir kur x yra vektorius sudarytas i N-kintam j . Keturi pagrindiniai elementai, apra yti Metropolio algoritme, iuo atveju yra tokie: f funkcijos kintamasis yra funkcijos ie komas tikslas. Sistemos b sena priklauso nuo parametro x. Valdantis parametras T yra ka kas pana aus temperat r , skaitant ir geriausio tvarkara io i rinkim , k sistema ir orientuojasi. Ir ketvirtas turi b ti konfig racijose atsitiktini funkcij generatorius, kurio veikimo b das yra sudarytas i atsitiktini ingsni nuo x iki x + Δx.

Norint apra yti paskutin element i keturi auk iau pamin t element , kyla daugiausia problem . Efektyviausias ios problemos sprendimas yra toks: atsitiktini pasikeitim generatorius bus neu baigtas, jei sistema gali rasti dar blogesn variant , vis d lto daugiausia bus ie koma geresni variant . Geras generatorius neturi tapti neveiksmingas siaurame poky i intervale; taip pat turi tapti vis labiau neveiksmingu kai vyksta konvergencija minimum . Kad gyvendinti SA minimizavim nepertraukiamai kontroliuojant interval , reikia naudoti modifikuot geresnio varianto paie kos vienpus metod . Jis svarbus pakei iant vien x kintam j vienpus sistemos b senos apra ym N + 1 kintamajam. Metropolio metodo gyvendinimas yra iek tiek subtilus: reikia prid ti teigiam , logaritmi kai atsitiktinai proporcingai temperat rai T paskirstyt kintam j , kad atsarginis funkcijos kintamasis b t susij s su kiekvienu vienpusiu kra tutiniu kintamuoju ir atimti pana kintam j i kiekvieno naujo funkcijos kintamojo, kuris yra patikrintas kaip tinkantis pakeitimui kintamasis. Kaip paprastas Metropolito metodas, is metodas po ingsn visada ie ko geresnio sudaryto varianto, bet kartais randa ir blogesn . Kai T -> 0, is algoritmas kaip tik suma ina paie k blogesnio varianto vienpusiu metodu ir konvertuoja lokal minimum .

3. GMJ metodo parametr optimizavimo s lygos priklauso nuo u davinio ir nuo laiko, kuriuos reikia optimizuoti. GMJ gali b ti veikti kaip apletas arba kaip Java programa, kuri dirba nepriklausomai nuo kit program (stand-alone) pritaikymas. Pirmasis gali veikti per internet . O antrasis negali b ti taip paprastai pritaikytas. Jame yra tam tikri apribojimai, kurie reikalauja papildom program , kad palaikyti sistem . Tai n ra did iul kli tis pritaikyti sistemos variant kaip papras iausi aplet . Tik yra reikalinga papildoma programin ranga leid ianti skaityti/ra yti i / fail , bei prisijungti prie ios sistemos. ie reikalavimai turi b ti vykdyti tik dirbant su Java programa, kuri dirba nepriklausomai nuo kit program (stand-alone). Naudojant i program , daugiausia naujausi JAVA savybi yra palaikomos ir panaudojamos i JDK. Prie astis yra ta, kad interneto nar ykl s yra seniai pritaikytos prastai naujiems JDK. Jeigu JDK yra diegtas, tada, papildomai diegus reikaling programin rang , i program galima nesunkiai paleisti naudojant papras iausias

interneto nar ykles. Ta iau patogiau yra naudoti apleto versij , nes norint naudotis ia programa nereikia diegti jokios papildomos programin s rangos.

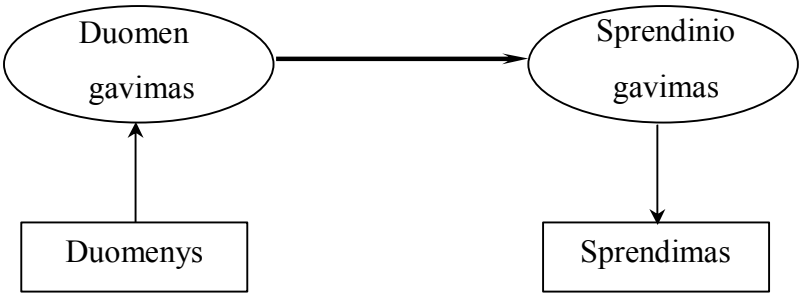
GMJ apleto versijoj yra galimi tokie optimizavimo metod pasirinkimai:

- Bayes – Bayeso pagal Mock ,
- Exkor – Bayeso koordina i paie kos pagal ilinsk ,
- Globt – grupavimas pagal Torn ,
- LBayes – atsitiktinis aproksimavimas pagal Mock ,
- Mig1 – Monte Karlo.

3.3 Tvarkarašèiø sudarymas

ios studijos siekia sukurti model , kuris sudarin t auk tos kokyb s mokyklos tvarkara ius panaudojant tvarkara io sudarymo inias. Tvarkara i ini esm yra ta, kad jos susideda i strukt rini duomen komponent , taisykli komponent ir euristikos. Tvarkara i sudarymo programa yra sukurta tam, kad b t galima apibr ti u duotis bet kuriame norimame nustatyme naudojant euristines funkcijas ir kad leist i rinkti geriausi laik i kit programos darbo metu gaut laik . Euristin s funkcijos yra skirtos pagerinti sistemos atliekamiems veiksmams. Kai kuri taisykli pirmumas gali palengvinti skirting tvarkara i sudarymus. [Kong et al., 12 (3): 81-93 JUN 1999, KNOWLEDGE-BASED SYSTEMS]

Clancey (1985) rodin ja, kad svarbios klasifikavimo charakteristikos yra tos kategorijos, kurias i visos galimos sprendini grup s pa ymi specialistas. Pana i problem pagrindinis objekto bruo as yra pakankamos klasifikacijos, taigi atitinkamumas tarp duomen ir kategorijos yra tiesioginis. Sud tingesn ms problemoms pagrindin s savyb s, kad atpa int tinkam ak ir lygi hierarchijas, gali b ti ir nepatenkintos. iuo atveju reikia panaudoti kaip Clancey vadina Euristikos klasifikavim¹ : nehierarchinius ry ius tarp duomen ir kategorij susieti su vartotojo s saja, o leistinas s vokas susieti su kitom sistemom. Tai geriausia parodo br inys 5.1, kuriame yra vaizduojami trys pagrindiniai euristin s klasifikacijos ingsniai: sprendinio gavimas i pateikt duomen ir sprendimo gerinimas.



1 pav. Euristinės klasifikacijos struktûros sˡsajos

- Duomenø gavimas. Visa tai da niausia naudojama gauti abstraktiems duomenims i duot tam tikros grup s duomen .
- Euristinis suderinamumas. suderinamum tarp tam tikros srities pradini duomen ir galutini rezultat yra sunku pasiekti. Stebimi duomenys, i kuri paprastai pritaikomi abstrakt s duomenys, duoda sprendinius, i kuri gaunami labiau tik tini ir abstrakt s sprendiniai.
- Sprendimo gerinimas. Lyginant sprendinio gavim su siaura sprendini grupe, reikalinga identifikuoti ir surikiuoti pradinius sprendinius ioje grup je. Tai tur t pareikalauti argument apie esamus duomenis arba tur t pareikalauti tolimesnio duomen sukomplektavimo.

3.3.1 Euristikos pritaikymas tvarkaraščiø optimizavime

Papras iausi prioritet rinkinys yra apibr tas pagal ilgesnio darbo euristik (Longer-Job) euristik :

$$h_i(j) = \frac{\tau_j - A_i}{A^i - A} + a, \qquad (3.2)$$

kur

$$A_i = \min_j \tau_j,\ A^i = \max_j \tau_j,\ a > 0. \qquad (3.3)$$

ia $u(\omega, q)$

kur τ_j yra j darbo apimtis/laikas.

3.3.2 Mokyklos tvarkaraščio sudarymas

Tradicinio mokyklos tvarkaraščio sudarymas

Optimali mokyklos tvarkara i sudarymas yra tik vienas i tvarkara io optimizavimo pavyzd i . iuo metu yra svarstoma galimyb , kaip pa alinti mokyklos tvarkara tyje susidariusius mokytoj „langus“. Tokios laisvos valandos vadinamos „mokytoj tarpas tarp pamok “, arba tiesiog „tarpais tarp pamok “. Dabar ie koma toki tvarkara i optimizavimo program , kurios suma int mokytoj „lang “ skai i 5-12 klasi pamok tvarkara tyje. Kiti mokykloje esantys faktoriai yra specifiniai ir turi b ti traukti sistem , kuri skirta specifin ms mokykloms.

Profiliuotos mokyklos tvarkaraščio sudarymas

Algoritmas apra o tradicini profiliuot mokykl pavyzd ius. iose mokyklose kiekvienai klasei yra fiksuotas pamok tvarkara tis. Ta iau io algoritmo apra ymas yra kur kas sud tingesnis, jei kiekvienas mokinys gali pasirinkti jam patinkan ius dalykus. Tarkim, jei keli mokiniai pasirinko religij , keletas kit – etik ir taip toliau. Taigi klasi kiekis nat raliai i auga, nes toki padalyt klasi tvarkara tyje vienu metu vykstan ios pamokos yra skirtingos. Bet klas s yra padalytos kelet grupi ir paprastai ios grup s prisijungia viena prie kitos.

Vadinamosiose „profiliuotose mokyklose“ 11 ir 12 klasi moksleiviams yra tradici kai skirta apie 14 – 16 dalyk . Tai rei kia, kad susidaro tiek skirting ir individuali pamok tvarkara i , kiek yra tose klas se moksleivi . Rei kia, kad daugiau n ra prast (stabili) klasi kaip anks iau. Jos pakeistos vadinam sias „besidomin ias grupes“. Jos kei iasi kiekvien pamok . Individual s 11 ir 12 klasi moksleivi pasirinkimai yra kitokie nei mokom dalyk „rinkinys“. i dalyk seka ir atitinkami kabinetai yra apibr ti sudarant pagrindin mokyklos tvarkara t .

iuo atveju reikalingas naujas po i ris apie optimal tvarkara t . Pavyzdys – komercin s sistemos „MIMOSA“, „RECTOR“ arba „aSc Timetables 2003“.

Programa „RECTOR“ yra labai elementari. Reikia suvesti duomenis ir rankiniu b du juos i d lioti. Pati programa nevykdo nei tvarkara io sudarymo, nei optimizavimo. iai programai 100 proc. reikalingas mogaus rank darbas.

„MIMOSA“ padeda sudaryti pus tin pamok tvarkara t vedant ka kuriuos pataisymus ranka pavyzd iui, kad u pildyti ka kurias laisvas pamokas ir informuoti apie pa eidimus bei nepatogumus sudarin jant tvarkara t . Sudarant tvarkara t programa „MIMOSA“ ma daug 50 proc. reikalingas mogaus rank darbas. Tod l palyginti j su kitomis sistemomis yra labai sud tinga. ia prasme „MIMOSA“, „RECTOR“ ir kitos pana ios sistemos gali sudominti daugiausia tik kaip „Palaikan ios sistemos“.

Programa „aSc Timetables 2003“ tvarkara ius sudarin ja pana iai kaip ir programa „MIMOSA“. Programa taip pat gali dirbti daug resurs neturin iame kompiuteryje. Duomenis ji skaito

ir transformuoja programos „MS Excel“ format . Programa vest tvarkara t iek tiek suoptimizuoja, bet ne visada sudaro piln tvarkara t . Jei klas s n ra profiliuotos, tada ji sudaro piln tvarkara t . Jei klas s profiliuotos, tai ka kuriuos pogrupius ir j paskaitas reikia sud lioti tvarkara t rankiniu b du. Tai n ra patogu kaip ir programoje „MIMOSA“. i programa taip pat reikalauja apie 50 proc. mogi k j resurs . Ji dirba ma daug kaip ir programa „MIMOSA“, tik jos valdymas yra daug paprastesnis nei programos „MIMOZA“. Programa „aSc Timetables 2003“ yra i leista ir lietuvi kalba.

3.3.3 Tvarkaraščio sudarymo aprašymas

Kad gauti ger profiliuotos mokyklos tvarkara t , pirmiausia reikia sudaryti 4 kintam j dvigubas masyv :

$$schedec[M][V][G][K] \tag{3.4}$$

kur

M – mokytoj kiekis;

V – dirbam valand skai ius per savait ;

G – skirting mokini grupi kiekis;

K – kabinet (auditorij) kiekis;

Jei $$schedec[i][j][k][l] = 1 \tag{3.5}$$

rei kia, kad mokytojas(i) valand (j) klas je(l) moko mokini grup (k);

Jei $$schedec[i][j][k][l] = 0 \tag{3.6}$$

rei kia pamok n ra;

Pagrindinis tvarkara tis yra vaizduojamas kaip trij kintam j dvigubas masyvas:

$$schedg[M][V][K] \tag{3.7}$$

Kad b t galima pagrindin tvarkara t atvaizduoti tik su dviem kintamaisiais, reikia sudaryti dviej kintam j masyv :

$$schedg\ 2[M][VK] \tag{3.8}$$

kur VK yra laiko ir kabineto apibr t kintam j seka.

Individual s tvarkara iai yra apibr ti kiekvienai grupei k kaip trij kintam j masyvai:

$$sched_k[M][V][K]. \tag{3.9}$$

Kad tvarkara t b t galima atvaizduoti su dviem kintamaisiais, reikia sudaryti dviej kintam j masyv :

$$schedg2_k[M][VK] \qquad (3.10)$$

Dvinaris atvaizdavimas yra paprastas, patogiai suprantamas ir turi apibr tus ribojimus, ta iau reikalauja daug atminties, bet yra patogus reali profiliuot mokykl tvarkara iams sudaryti.

3.3.4 Tvarkaraš èi ø sudarymo ypatumai

N ra toki mokykl tvarkara i , kurie atitikt visus apribojimus ir individualius reikalavimus, nes jie prie tarauja vienas kitam. Pavyzd iui, mokslo ir vietimo ministerija apibr ia tam tikr taisykli skai i , kad sudaryti mokyklos mokytoj ir mokini tvarkara ius, kuriuose neb t „lang “ ir pana iai. Tokiu atveju reikia rasti tok varianta, kuris kiek manoma labiau tenkint abi puses: tiek mokytojus, tiek mokinius.

Kompromisinis sprendinys surandamas apibr iant baudos ta kus pradiniuose duomenyse atsirandantiems pa eidimams ir nepaisomiems nepatogumams. Yra ie koma tokio tvarkara io, kuris suma int baudos tak skai i . Negali b ti tik toks pagrindinis duomen pa eidimas:

- Asmuo negali b ti dvejose vietose vienu metu;

Tarkim visi kiti apribojimai gali b ti ir pa eisti, t. y. mokytojas ar mokinys gali tur ti „langus“ ir t.t. Visus galimus pa eidimus reikia vertinti baudos ta k kiekiu.

3.3.5 Pagrindiniai reikalavimai tvarkaraš èi ø sudarymui

Idealaus tvarkara io ir mokiniui, ir mokytojui sudaryti prakti kai ne manoma, nes j tarpusavio interesai da niausia susikerta. Siekiamam kompromisui pagalb buvo pasitelktas „baudos ta k “ skai iavimas. Dabar stengiamasi sudaryti tok tvarkara t , kuriame baud skai ius b t kiek manoma ma esnis. Baud skai ius niekada nebus lygus nuliui, nes fizin s mokytojo ir mokinio galimyb s ne visada tai leid ia. Pavyzd iui, nei mokinys, nei mokytojas negali vienu metu b ti dvejose vietose.

Taigi pagrindiniai reikalavimai b t tokie:

- kiekvienai grupei k savaitinis valand skai ius turi b ti: $v_{sk} \le V_{sk} < 168$;
- kiekvienai grupei k per dien pamok turi b ti: $v_{dk} \le V_{dk} < 24$;

  11 ir 12 klasei per dien turi b ti: $V_{sk} < 5V_{sd}$;

  likusioms $V_{sk} < 5V_{sd}$;
- kiekvienam mokytojui i yra skirtas V_i savaitini pamok skai ius,

- tai pa iai grupei k savaitini tam tikro dalyko n pamok gali b ti V_{kn},
- tai pa iai grupei vienu metu gali vykti tik viena pamoka,
- jei yra poreikis, vieno dalyko pamokos privalo vykti dvi pamokas i eil s,
- jei yra draudimas, vieno dalyko pamokos negali vykti dvi pamokas i eil s,
- jei reikia, vieno dalyko pamokos gali vykti dvi pamokas i eil s,
- 1 – 10 klasi mokiniai negali tur ti „lang “.

Nepatogumai takojantys tvarkara i sudarym yra:

- mokytoj langai,
- mokini langai (11-12 klas ms),
- nepatogios valandos,
- nepatogios dienos,
- nepatogi pamok seka.

3.3.6 Duomenø sudarymo s¹lygos

Yra penkios pagrindin s ir svarbiausios duomen sudarymo s lygos:

- bet kuriuo duotu momentu mokytojas gali b ti tik vienoj paskaitoj;
- bet kuriuo duotu momentu mokinys gali dalyvauti tik vienoj paskaitoj;
- neleistini mokini „langai“;
- n ra leid iamos dvi to pa io dalyko, viena paskui kit einan ios, pamokos (nebent yra i im i);
- pamok skai ius per dien yra ribotas.

3.3.7 Baudos taškø skaièiavimas

Normatyvinës baudos

Pagrindiniai tvarkara i sudarin tojo nuostatai negali b ti pa eisti. Ta iau gali pasitaikyti keletas nedideli pa eidim , kurie pagerina kitus optimizavime naudojamus parametrus. Tokie pa eidimai yra leid iami, bet tokiu atveju bauda u pa eidimus c_r turi b ti didel .

$$C_n = \sum_r c_r N_r \qquad (3.11)$$

kur

c_r – bauda u norm r pa eidim ;

N_r – pa eidim kiekis;

r kinta $r = 1,..,9.$

Tikrasis c_r kiekis yra specialiai apskai iuojamas ir priklauso nuo dabartin s mokyklos situacijos. Tod l tai yra parametrai, u duodami vartotojo naudojant grafin vartotojo s saj .

Nepatogumø baudos

Nepatogum baudos vartotojo skiriamos kiekvienam nepatogumo atvejui:

1. c_i yra bauda kiekvieno mokytojo i langui;
2. c_k yra bauda kiekvienos klas s k langui;
3. c_{ji} yra bauda u pamok j, kuri vyksta mokytojui i nepatogiu laiku;
4. c_{li} yra bauda u dien l, kuri yra nepatogi mokytojui i;
5. c_{jk} yra bauda u pamok j, kuri vyksta klasei k nepatogiu laiku;
6. c_s yra bauda u nepatogi pamok s sek ;

Bendras nepatogum baud skai ius skai iuojamas taip:

$$C_c = \sum_i c_i L_i + \sum_k c_k L_k + \sum_i \sum_j c_{ji} L_i^j + \\ + \sum_i \sum_l c_{li} L_i^l + \sum_k \sum_j c_{jk} L_k^j + \sum_s c_s L_s \qquad (3.12)$$

kur

L_i yra mokytojo i lang skai ius;

L_k yra g klas s k lang skai ius;

L_i^j yra skai ius pamok j, vykstan i mokytojui i nepatogiu laiku;

L_i^l yra skai ius dien l, vykstan i mokytojui i nepatogiu laiku;

L_k^j yra skai ius pamok j, vykstan i klasei k nepatogiu laiku;

L_s yra skai ius nepatogiai i sid s iusi pamok sek s;

Galutinis baudos ta k skai iavimas atliekamas taip:

$$C = C_n + C_c \qquad (3.13)$$

Optimizavimo problema yra:

$$\min_{\tau \in \Theta} C(\tau) \qquad (3.14)$$

kur $C(\tau)$ yra visas tvarkara io ô baud skai ius, È yra tvarkara i seka sudaryta taip, kad atitikt fizinius reikalavimus. Apie baudas $C(\tau)$ yra pasikliaunama specialisto vertinimu, tod l j skai iavimas yra priimamas kaip euristika.

3.3.8 Optimizavimas kaitaliojant laukus

Pagrindinis tvarkara tis gaunamas kaitaliojant duomenis. Geriausias gautas tvarkara tis yra ra omas po kiekvienos iteracijos. Pakeitimas blogesn tvarkara t yra vykdomas su daug ma esne tikimybe nei pakeitimas geresn . Taip yra daroma tam, kad pagerinti suliejimo b kl .

Norint sudaryti optimal tvarkara t , pirmas atliekamas ingsnis yra bandymas pasir pinti rezultat suliejimu naudojant SA metod . Kiekviena operacija su mokyklos tvarkara iu i pakei iant j perstatyt mokyklos tvarkara t i + 1 yra vykdoma su tokia tikimybe:

$$r_{i+1} = \begin{cases} \dfrac{-h_{i+1}}{e^{x/\ln(1+N)}}, & \text{kai } h_{i+1} > 0 \\ 1, & \text{kitu atveju;} \end{cases} \qquad (3.15)$$

kur

N – iteracij kiekis;

x – „pradin temperat ra“;

Logaritminis „ alimo greitis“ ln(1+N) yra gautas i suliejimo s lyg . (Cohn ir Fielding, 1999)

io metodo skirtumas nuo tradicinio SA yra tas, kad ia optimizuojamas parametras x ir jei iteracij skai ius fiksuotas (N=K), tai tvarkara io „ alimas“ taip pat turi b ti vertintas. Pats geriausias b das tai gyvendinti yra terpti dar vien parametr x_2 .Taigi (3.16) formul atrodys taip:

$$r_{i+1} = \begin{cases} \dfrac{-h_{i+1}}{e^{x_1/\ln(1+x_2 N)}}, & \text{kai } h_{i+1} > 0 \\ 1, & \text{kitu atveju;} \end{cases} \qquad (3.16)$$

kur

$x_1 \geq 0$ –apibr ia pradin SA „temperat r “, o $x_2 \geq 0$ apibr ia „ alimo greit “;

SA ir kitos diskretaus optimizavimo technologijos yra apra ytos knygoje (Mockus, 2000) Knapsack pavyzdyje.

3.3.9 Pradinio tvarkaraščio sudarymas

Kai kuriais atvejais pradinis tvarkara tis yra sudaromas paties vartotojo. Kitais atvejais jis yra sudaromas pasinaudojant euristika;

Pradinis tvarkara tis, sudarytas naudojant euristik , techni kai yra naudojamas kaip startas. Rezultatai gaunami i sudarymo operacij sekos. is sudarymo kelias, kurio metu gaunamas mokyklos tvarkara tis, yra patogus gauti pirmo mokytojo tvarkara iui, bet labai sud tingas gauti paskutinio mokytojo tvarkara iui.

Kaip paprastai euristika apibr ia pasikeitimus. Tai rei kia, kad pirmas subjektas (klas ar asmuo) turi didesn prioritet , kad „gauti“ geresn tvarkara t . Tvarkara tis sudaromas pagal gryn euristik da niausia neatlieka kai kuri svarbi reikalavim . Tod l vedamas atsitiktinis parinkimas. Tai padeda sudaryti geresn pradin tvarkara t atsitiktinai parenkat tvarkara io ribas ir praple ia paie kos srit .

Nuskaitomas vartotojo duomen failas masyv . Atsi velgiant nustatytus apribojimus, jei nustatytos tam tikros besit sian ios pamokos, jos sudedamos tvarkara t pirmos. ios pamokos dedamos pirmadien , o likusios i metamos kitas dienas. Toliau likusi pamok sud jimas vykdomas tokia tvarka:

- pasirenkama diena ir pamoka,
- tikrinama ar pamoka gali vykti (vertinami fiziniai apribojimai),
- sudedamos visos pamokos, kurios gali vykti kartu.

Taip pat atliekamas patikrinimas ar mokytojas t dien neturi i eigin s (ar tai nustatyta apribojimuose). Viena pamoka vyksta vien kart per dien , jei kitaip nenurodyta apribojimuose. Reikia pasteb ti, jog formuojant pradin tvarkara t n ra kreipiamas d mesys mokini ar mokytoj lang skai i , o stengiamasi sutalpinti visus dalykus bendr tvarkara t . I prad i bandoma sutalpinti tvarkara i sudarin tojo pasirinkt maksimal pamok skai i , i galim variant punkto "Number of lessons". Jei tai padaryti nepavyksta, tada atliekamas masyvo i mai ymas (eilu i sukeitimas vietomis) ir v l formuojamas tvarkara tis. Jei atlikus i mai ym visos pamokos netilpo, tada jos d liojamos prie nurodyto maksimalaus pamok skai iaus prid jus vienet . Tokiu atveju u vien pamok pridedamas maksimalus baudos ta k kiekis 10. Sudarius pradin tvarkara t jis pertvarkomas, t. y. pirmas pamokas keliamos tos pamokos, kurias lanko daugiausiai mokini , tam, kad suma inti lang skai i pradiniame tvarkara tyje. Paskui atliekamas optimizavimas.

3.3.10 Tvarkaraščio optimizavimo aprašymas

Optimizavimas orientuotas mokini lang tarp pamok ma inim . Optimizavimo algoritmas atrodo taip:

1. Iteracijos numeris nustatomas vienet .

2. Keliaujama per mokini s ra .
3. Parenkamas atsitiktinis skai ius x.
4. Jei x didesnis u m s pasirikt tikimyb ("Probability"), tai atlikin jami pakeitimai mokinio tvarkara tyje, o jei x ma esnis u tikimyb – einama prie kito mokinio.
5. Ie komi pasirinkto mokinio langai. Kai langas surandamas, tada ie komos kiekvienos dienos pirmos ir paskutin s pamokos, kurios gali b ti terptos vietoje io lango. Jei tos pa ios dienos tinka ir pirma, ir paskutin , atsitiktinio skai iaus pagalba parenkama viena i j .
6. Vietoje lango terpiama rasta pamoka, o vietoje rastos pamokos padaroma laisva vieta.
7. Kai per i rimi visi mokinio langai keliaujama punkt numeris 3.
8. Kai per i rimi visi mokiniai, padidinamas iteracijos numeris ir einama punkt numeris 2.
9. Kai atliktos visos iteracijos, pateikiamas geriausias tvarkara io variantas.

vykd ius visus punktus, geriausias variantas, laikomas tas, kuris turi ma iausi baudos ta k skai i , o ne tas, kuris atitinka visus apribojimus. Tai daroma tod l, kad pavyzd iui kokio nors apribojimo ne vykdymas gali labai enkliai suma inti mokini lang skai i .

3.3.11 Kaitaliojimo ir ávertinimo algoritmo aprašymas

Pradin tvarkara t , kur vartotojas veda, sistema priima kaip teising . is tvarkara tis yra optimizuojamas pagal ank iau apra yt kaitaliojimo algoritm . is algoritmas yra sudarytas naudojantis pagrindiniu kaitaliojimo ablonu, kuris susij s su „Bayeso Eutistiniu Priart jimu" [Mockus, 2000], kuris sudaro tvarkara t palank mokytojams.

Algoritmo veikimas:

1. dabartinio tvarkara io ra as Mokytojai0 nuskaitomas i duomen failo masyv Mokytojai[15][30];
2. nustatomas mokytoj lang skai iaus ra as;
3. nustatoma pradin iteracij skai iaus seka it = 0;
4. dabartini iteracij skai iaus seka skai iuojama taip: it := it +1, kur it < K;
5. jeigu it = K, procesas yra stabdomas ir atspausdinamas tuo metu sudarytas tvarkara tis;
6. nustatoma pradin mokytoj skai iaus seka i = – 1;
7. dabartini mokytoj skai iau seka yra skai iuojama taip: i := i + 1;
8. jei i > 14 reikia gr ti ingsn numeris 4;
9. sukuriamas pastoviai paskirstomas tvarkara tis pagal atsitiktin skai i $\xi \in [0,1]$;
10. jei $\xi < \chi$ reikia gr ti ingsn numeris 7;

11. pa ymimi i – tojo mokytojo „langai“ ir gr tama ingsn numeris 7 (ia n ra lang);
12. pa ymimos i – tojo mokytojo „atviros“ pamokos ir gr tama ingsn numeris 7 (ia n ra laisv pamok);
13. vykdomas keitimas sukei iant „langus“ su „atvirom“ pamokom;
14. tikrinamas esamos b kl s vykdomumas (tinkamumas) ir jei keitimai yra vykdomi, tada galima pereiti ingsn numeris 16;
15. tikrinamas esamos b kl s vykdomumas ir jei keitimai n ra vykdomi, tada programa „gr ina“ mokytoj langus ir gr ta ingsn numeris 7;
16. tvarkara tyje naujai perstatomas mokytoj lang skai ius;
17. palyginamas prie tai gautame tvarkara tyje esantis mokytoj lang skai ius su dabar perstatyto tvarkara io mokytoj lang skai iumi;
18. jei dabar perstatyto mokytojo tvarkara io lang skai ius yra ma esnis nei prie tai buvusio, tada ra omas naujai po perstatymo gautas tvarkara tis ir gr tama ingsn numeris 7;
19. viskas kartojama tol, kol nelieka nauj variant . Baigusis ciklui i spausdinamas geriausias variantas;

3.4 Duomenø struktûra

Duomen failas apibr ia d stom dalyk s ra ir mokini pasirinkimus. Failo formatas yra tekstinis (gal n yra txt). Viena eilut yra skirta tik vienam dalykui.

Duomen failo strukt ra yra tokia:

1. klas s, kuriai skirtas dalykas, numeris (gali ir neb ti);
2. dalyko pavadinimas;
3. mokytojo pavard ar kodas;
4. klas s, kurioje bus d stomas dalykas, numeris;
5. kiek pamok vyks per savait (skai ius);
6. dalyk pasirinkusi mokini s ra as (arba pogrupio pavadinimas);

D stomas dalykas | mokytojo pavard | -(kabinetas)- | pamok kiekis per savait | mokinys1 | mokinys2 | mokinys3 | … | mokinys N |^

Duomenys turi b ti atskirti simboliais „|“; Eilut s pabaiga ymima simboliais „|^“. Kabinetai tur t b ti ra yti tarp simboli „-(***)-“, nes jie bus matomi tvarkara tyje ir vartotojui patogiau bus jei jie skirsis nuo kit duomen ;

Pavyzd iui:

11Angl K | Kar iauskiene | -(203)- | 5 | Kalinauskas | Kazakevi ius | Justutyt | Zakeviciute |^

12Angl K | Kar iauskiene | -(203)- | 6 | Vainorius | Klimavicius | Sviderskaite | Jonusis |^

.

.

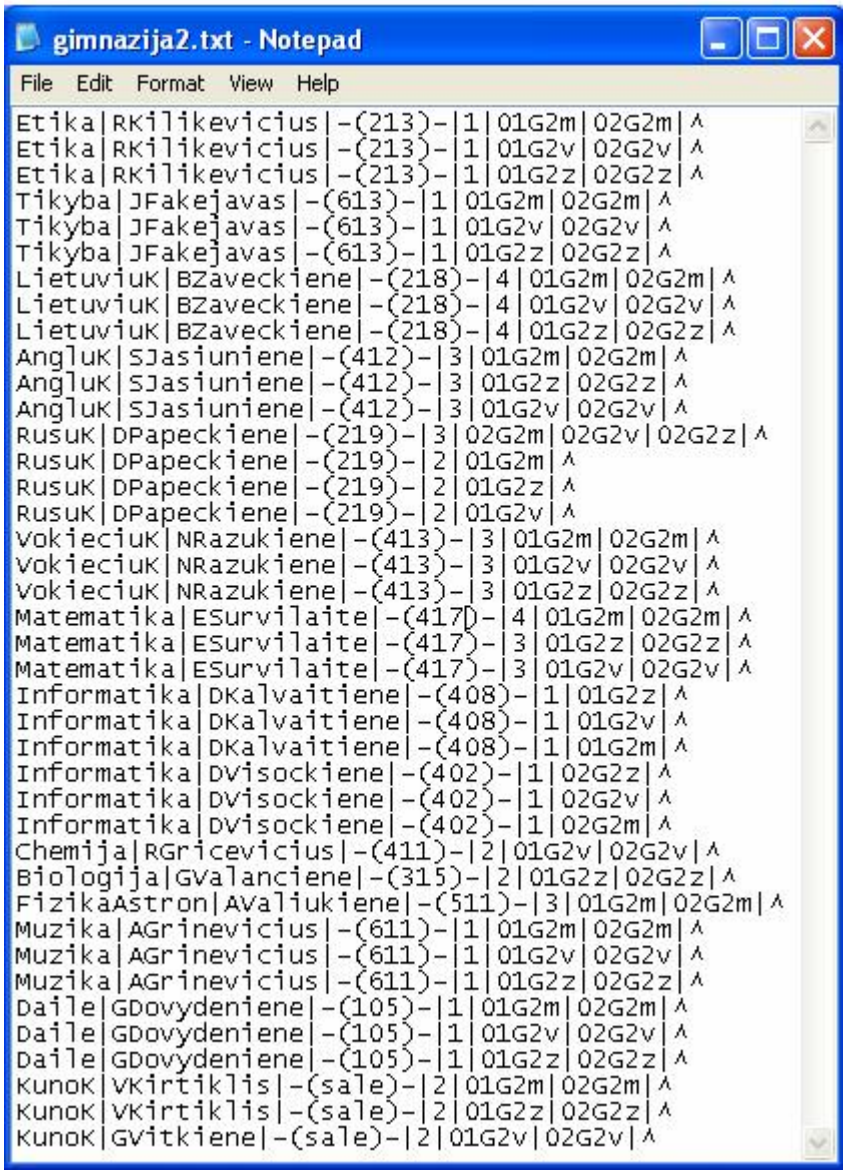
.

12Fizinis | Navikas | -(sale)- | 3 | Vainorius | Klimavicius | Jonusis |^

Duomenø sudarymo analizë:

- tvarkara tis yra sudarytas kaip ra masyvas Mokytojai[15][30],
- ra as Mokytojai[i][1] rei kia i-tojo mokytojo pavard , pvz.: D. Visockien , D. Kalvaitien ir t.t.
- ra as Mokytojai[i][2] rei kia i-tojo mokytojo d stoma dalyk , pvz.: Matematika, Informatika ir t.t.
- ra as Mokytojai[i][3] rei kia i-tojo kurioje klas je turi vykti pamoka, pvz.: 315, 408 ir t.t.
- ra as Mokytojai[i][4] rei kia i-tojo mokytojo pamok skai i per savait , pvz.: 2,3,5 ir t.t.
- ra as Mokytojai[i][j], j=3,...,30 rei kia, kad i-tojo mokytojo d stom dalyk pasirinko j mokini ;

Paveiksle Nr.: 2 yra pavaizduoti Marijampol s kolegijos duomenys, skirti sudaryti savait s tvarkara iui.



2 pav. Pradiniai duomenys

3.5 Programinė realizacija

Siekiant kuo efektyviau vizualizuoti u davin ir i naudoti interneto privalumus buvo pasirinkta Java programavimo kalba. Daugel princip Java programavimo kalba yra paveld jusi i C++. Java kalba yra sukurta C++ kalbos pagrindu. Kaip ir daugelis objekti kai orientuot kalb , Java taip pat traukia bendruosius interneto protokolus ir daugel kit dalyk , kuri d ka i populiar jo i kalba.

Nepriklausomumas nuo platformos, t.y. galimyb lengvai program perkelti i vienos kompiuterio sistemos kit , yra vienas did iausi Java privalum kuriant ir realizuojant tokius bei pana ius projektus. Java kalbos klasi strukt ra sudaro galimyb lengvai sukurti kod , perne am nuo vienos platformos prie kitos.

„School schedule optimization program“ programa yra para yta Java programine kalba, tod l ji gali b ti vykdoma kaip Java “apletas” arba kaip Java vykdomoji programa (Java Application). Privalumas vykdant program kaip Java “aplet ” yra akivaizdus, ji gali b ti pasiekiama per internet ir paleid iama vykdyti ir Html formato failo. ios programos u davinys yra sprend iamas prijungus kitus optimizavimo modelius panaudojus pa ias naujausias Java programin s kalbos savybes.

3.5.1 Apleto versija

i versija gyvendina tik pirm j stadij , kur galutinis baud skai ius yra minimizuojamas i karto naudojant paprastesn euristin sukeitimo metod . Tai pagerina darb su pradiniu tvarkara iu. Patobulinti SA ir BHA metodai yra realizuoti servleto versijoje, nes servleto veikimo b das yra patogesnis naudojant silpnuose mokykl kompiuteriuose ir daug lengviau apdoroja besikei ian ius ar pasikeitusius duomenis.

3.5.2 Sistemos programinė realizacija

Programin ranga, skirta sudaryti profiliuot mokykl tvarkara iams, yra pagalbin priemon pana i tradicini mokyklos tvarkara i sudarin jimo programas (Mockus, 2000). Jos labai pana ios. Svarbiausios naujos id jos programoje „School schedule optimization program“ yra gyvendintos atsi velgiant profiliuot mokykl tvarkara i sudarym .

Pirmiausia tvarkara tis vertintas pagal abu – objektyv ir subjektyv – faktorius iame kelyje. Teorin ios minties strukt ra yra vadinamoji Pareto optimizacija. Trumpiausias kelias, kad pasiekti Pareto optimal mokyklos tvarkara t , reikia paskirti kelet baudos ta k pageidaujamiems ir nepageidaujamiems nukrypimams skai iuoti. Galutinis baud skai ius turi b ti kiek manoma ma esnis. iuo atveju baudos ta kai parodo tvarkara ius sudarin jan i moni subjektyvias i vadas. B kl atspindi objektyvius juridin ir fizin faktorius. Remiantis vektorinio optimizavimo teorija papras iausias ta iau pilnai korekti kas b das balansuoti vairius faktorius yra baudos ta k vedimas

minimizuojant nukrypimus nuo pageidaujam rezultat . Tvarkara tis bus tuo priimtinesnis konkre iose s lygose, kuo baudos ta k jame bus ma iau.

Antra nauja id ja – euristinis optimizavimas i renkant geriausius parametrus. Ji apima tris optimizavimo stadijas. Pirmoje – naudojami tie patys ne ym s euristiniai pakeitimai kaip ir tradicini mokykl tvarkara i sudaryme. Tai veikia jei vienas prasideda i beveik optimalaus mokyklos tvarkara io, kuris paprastai yra realiai naudojamas tradicin se mokyklose. Profiliuotose mokyklose kai kuri euristini pakeitim „globalizacija“ yra reikalinga. Elementarus lokalus optimizavimas, primenantis „MIMOSA“ naudojam , perstatin jant mokytoj „langus“ ir vedus papildom parametr – mokytojo „praleidimo“ tikimyb , „globalizuoja“ tvarkara t , ta iau ne ymiai.

Antrasis ingsnis buvo u baigtas naudojant SA tvarkara i sudarym , kuris taiko tuos pa ius fiksuotus parametrus kaip pradin „temperat ra“ bei „ alimo“ greitis. Tre ias fiksuotas perimetras fiksuoja galimyb „praleisti“ laukiant eil je mokytoj . iuo atveju pirmuoju tampa tas mokytojas, kuris turi ma tikimyb pakli ti tvarkara t , nes turi surink s labai didel baud skai i . N ra sunku prane ti, kad gaunant rezultat , kai rezultat gavimui naudojamas SA metodas, yra pasikliaunama visais trimis fiksuotais parametrais.

Tre ioje stadijoje visi ie parametrai yra optimizuojami naudojant metod atsitiktiniai globaliai optimizacijai, kurios pagrindas yra pritaikytas Bayeso euristiniame priart jime (BHA) (Bayesian Heuristic Approach)(Mockus, 2000). Kiekviena stadija gali b ti vairiai susijus, kas leid ia daug lengviau palyginti rezultatus.

Preliminar s bandymai parod , kad tinkamai realizavus ias naujas mokslines mintis b t galima sukurti pilnai tinkamus tvarkara i optimizavimo metodus esamoms profiliuotoms mokyklos bei juos lengvai pritaikyti naujiems mokymo metodams.

Dabar yra sukurtos ir naudojamos ios sistemos dvi versijos: apletas ir servletas. Apleto versijoj yra naudojama gra esn ir lankstesn grafika. Programa vykdoma grei iau, nes ji yra vykdoma vartotojo personaliniame kompiuteryje. Servleto versija yra vykdoma naudojant serverio resursus. i programa turi prana um jei vartotojo personalinio kompiuterio resursai yra riboti.

4 Modelio eksperimentinis tyrimas

4.1 Tyrimo metodo aprašymas

1 lentelė. Tyrimui naudojami metodai.

Metodo pavadinimas	Metodo kūrėjas	Matomos charakteristikos
Exkor	Bayeso koordinatės paieškos pagal Zilinsk	Tai Vynerio proceso realizacija; Vynerio procesas yra padarytas į Brauno judesį. Šis metodas prognozuoja, kad su kiekvienu bandymu bus įgyjama vis daugiau. Tai vienmatis procesas. Jis pasižymi Markovo kumo savybe: spendžiamas pagalbinis uždavinys ir šiek tiek geriausias rezultatas. Pagal šį rezultatą atliekami skaičiavimai.
Bayes	Bayeso pagal Mock	Tai daugiamečio metodo „Exkor“ realizacija. Be to šis metodas yra statistinis.
LBayes	Atsitiktinis aproksimavimas pagal Mock	Tai statistinė aproksimacija. Funkcija yra įkilusi, bet matuojamos tik reikšmės su paklaida. Metodas yra tik lokalus.
Globt	Grupavimas pagal Torn	Šis metodas veikia taip: iš pradžių atliekama atsitiktinė klasterizacija. Po to metodas šiek tiek pasistumia gręžimo kryptimi. Tada gauti kasteriukai yra optimizuojami pagal euristines savybes ir įšį renkami tik „geri“ klasteriukai. Ir vėl procesas kartojamas iš naujo, bet tik su gerai klasteriukais, kol gaunasi tik vienas klasteriukas. Su šiuo vienu klasteriuku atliekami visi skaičiavimai.
Mig1	Monte Karlo	Tai atvirkštinis „Exkor“ metodas.

4.2 SA gauti stebėjimø rezultatai

Pradiniai duomenys imami visi vienodi. Baudos ta kai nustatyti taip:

- Mokinio tvarkara tyje esantis langas: 10
- Mokytojo tvarkara tyje esantis langas: 1
- Dvi, viena paskui kit , i d stytos negalimos pamokos: 5
- Mokytojo d stomos pamokos, i d s per jo „laisvadienius“: 10
- Pamok skai ius per dien : 10

Iteracijø skaièius: 10

2 lentelë. Gauti rezultatai kai iteracijø skaièius yra 10.

Metodo pavadinimas	Pradinis baudø skaièius	Baudø skaièius po optimizavimo	Pradinë tikimybë	Tikimybë po optimizavimo
Bayes	1251	86	0,5	0,625
Exkor	1251	86	0,5	0,25
Globt	1251	83	0,5	0,63219
LBayes	1251	80	0,5	0,5
Mig1	1251	94	0,5	0,3897

Pradiniai duomenys tokie pat kaip ir prie tai atliktame bandyme.

Iteracijø skaièius: 5000

3 lentelë. Gauti rezultatai kai iteracijø skaièius yra 5000.

Metodo pavadinimas	Pradinis baudø skaièius	Baudø skaièius po optimizavimo	Pradinë tikimybë	Tikimybë po optimizavimo
Bayes	1251	65	0,5	0,68458
Exkor	1251	73	0,5	0,375
Globt	1251	72	0,5	0,13271
LBayes	1251	79	0,5	0,91168
Mig1	1251	69	0,5	0,50927

4.3 GMJ gauti stebėjimø rezultatai

Pirmame stulpelyje yra metodo vardas ir kiek iteracij jam buvo u duota, antrame – u davinio s lygos, tre iame po optimizavimo gautas rezultatas.

Gautos iteracijos yra iteracij skai ius nusakantis per kiek kart buvo gautas geriausias rezultatas. Funkcija F(x) charakterizuoja geriausi tam tikro metodo (Bayes, LBayes, Mig1, Globt, Exkor) skai i ir parodo jo optimali dal . Tikimyb parodo prie kokios tikimyb s buvo gautas F(x) rezultatas.

4 lentelė. Gauti rezultatai atliekant bandymus su metodu "Bayes".

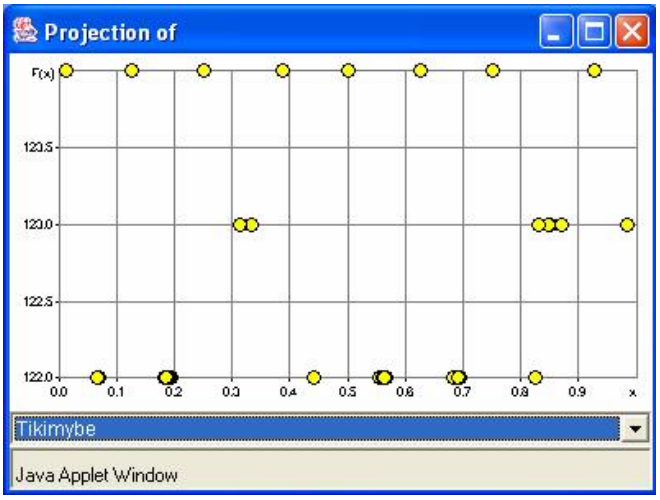
Metodas		Uždavinys	Rezultatas		
Vardas	Iteracijos	Kiek kartø optimizuoja uždavinį	Iteracijos	F(x)	Tikimybė
Bayes	5000	2	999	16.0	0.871
		6	999	10.0	0.625
		13	999	10.0	0.750
		25	999	8.0	0.500
		50	999	8.0	0.500
		100	999	8.0	0.500
	10	2	99	14.0	0.531
		6	99	13.0	0.250
		13	99	9.0	0.750
		25	99	14.0	0.438
		50	99	8.0	0.500
		100	99	8.0	0.500

Paveiksl lis 3 parodo kaip metodo Bayes reik m s priklauso nuo atsitiktini kintam j ;

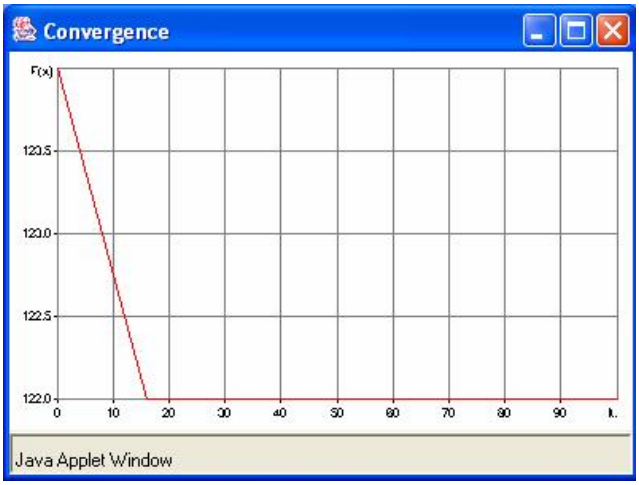
Paveiksl lis 4 parodo kaip metodo Bayes reik m kinta did jant iteracij skai ius;

Paveiksl lis 5 vaizduoja metodo Bayes funkcijos F(x) gautus rezultatus;

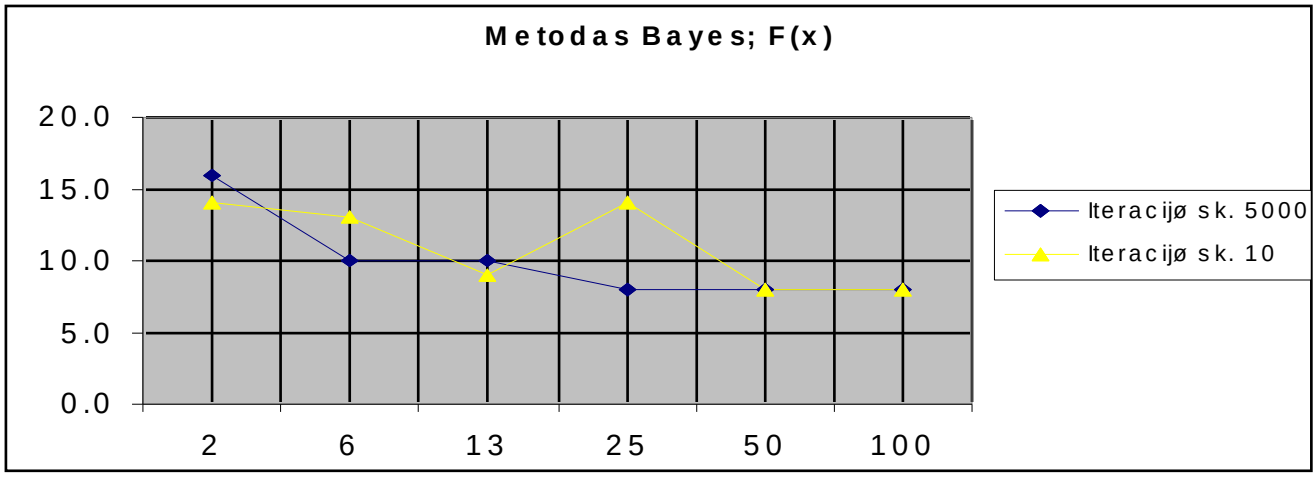
Paveiksl lis 6 vaizduoja prie kokios metodo Bayes tikimyb s buvo gauti F(x) rezultatai.



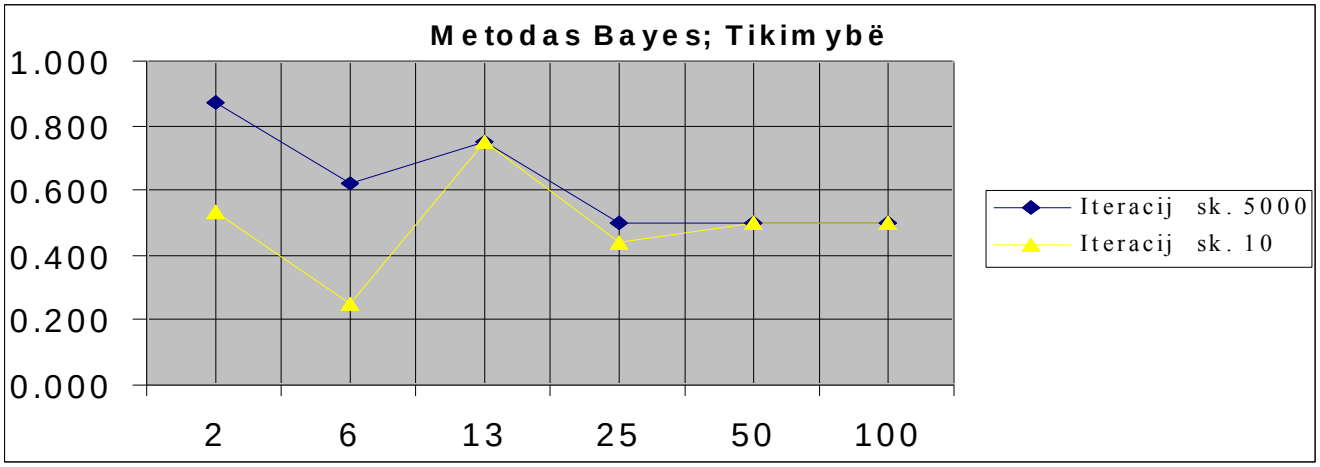
3 pav. Metodo Bayes reikšmės priklausomybė nuo atsitiktinio kintamojo;



4 pav. Metodo Bayes reikšmės kitimas didėjant iteracijos skaičiui;



5 pav. Vaizduoja metodo Bayes funkcijos F(x) gautus rezultatus;



6 pav. Vaizduoja prie kokios metodo Bayes tikimybės buvo gauti F(x) rezultatai;

5 lentelė. Gauti rezultatai atliekant bandymus su metodu “LBayes”.

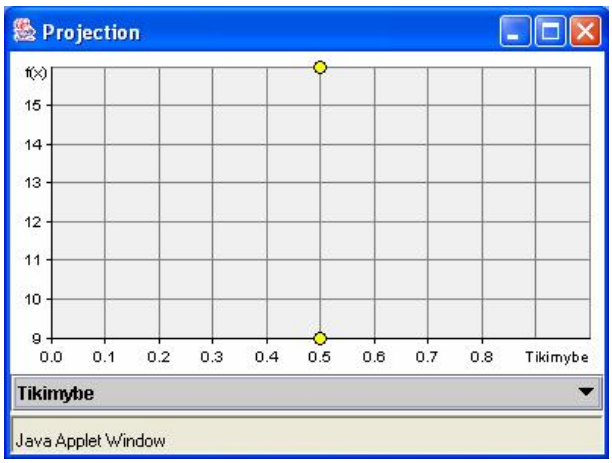
Metodas		Uždavinys	Rezultatas		
Vardas	Iteracijos	Kiek kartø optimizuoja uždavinį	Iteracijos	F(x)	Tikimybė
LBayes	1000	2	999	15.0	0.500
		6	999	13.0	0.500
		13	999	11.0	0.500
		25	999	8.0	0.500
		50	999	8.0	0.500
		100	999	8.0	0.500
	100	2	99	18.0	0.500
		6	99	18.0	0.500
		13	99	11.0	0.500
		25	99	8.0	0.500
		50	99	9.0	0.500
		100	99	8.0	0.500

Paveiksl lis 7 parodo kaip metodo LBayes reik m s priklauso nuo atsitiktini kintam j ;

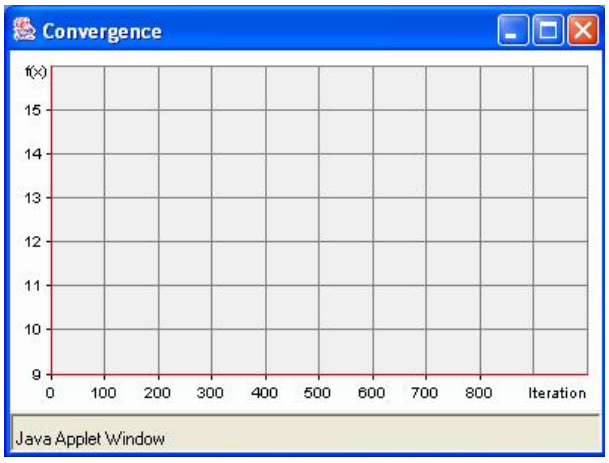
Paveiksl lis 8 parodo kaip metodo LBayes reik m kinta did jant iteracij skai ius;

Paveiksl lis 9 vaizduoja metodo LBayes funkcijos F(x) gautus rezultatus;

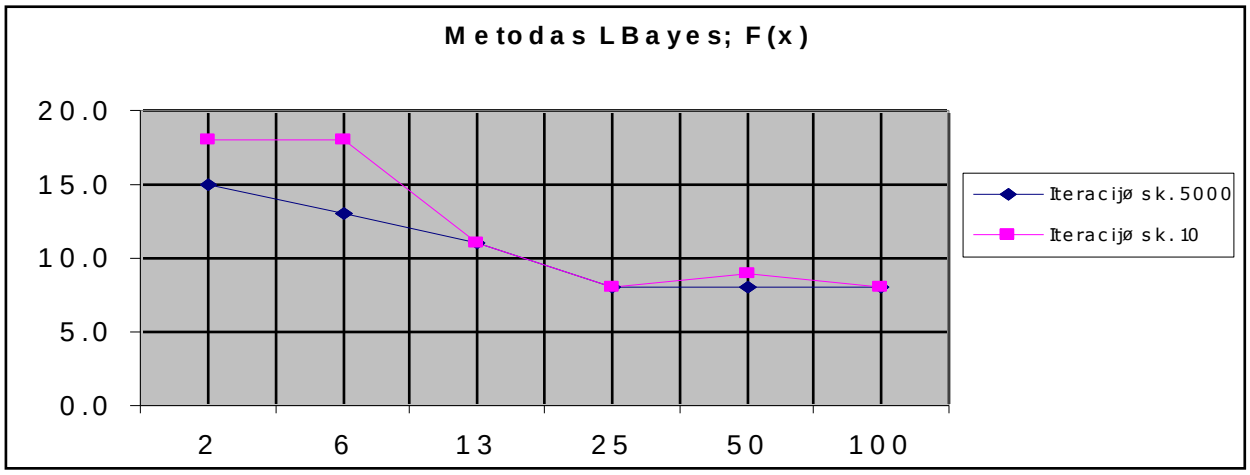
Paveiksl lis 10 vaizduoja prie kokios metodo LBayes tikimyb s buvo gauti F(x) rezultatai;



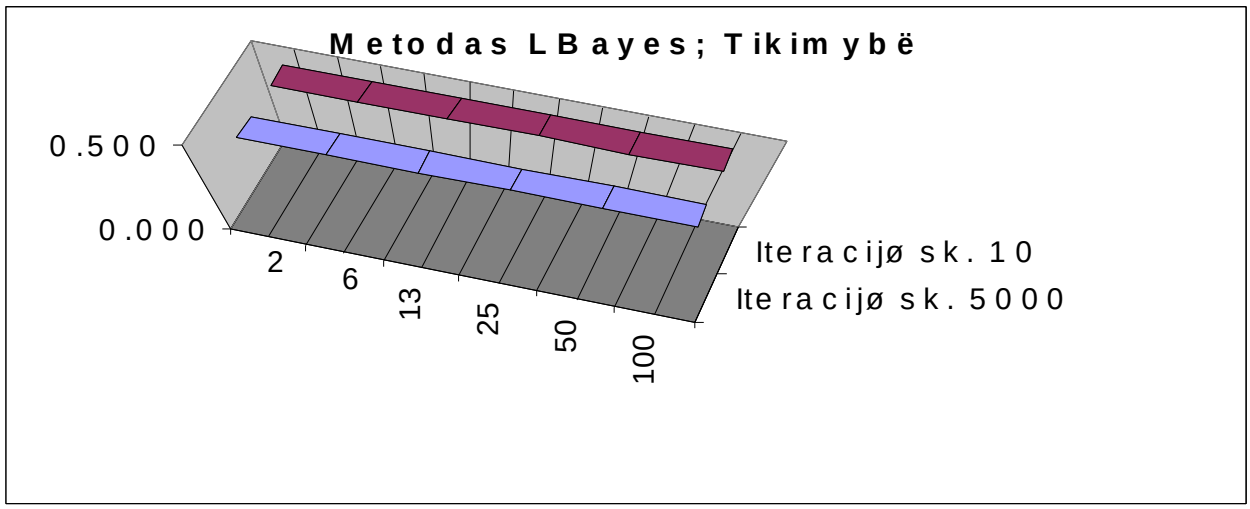
7 pav. Metodo LBayes reikšmės priklausomybė nuo atsitiktiniø kintamøjø;



8 pav. Metodo LBayes reikšmės kitimas didėjant iteracijø skaièiui;



9 pav. Vaizduoja metodo LBayes funkcijos F(x) gautus rezultatus;



10 pav. Vaizduoja prie kokios metodo LBayes tikimybės buvo gauti F(x) rezultatai;

6 lentelė. Gauti rezultatai atliekant bandymus su metodu “Exkor”.

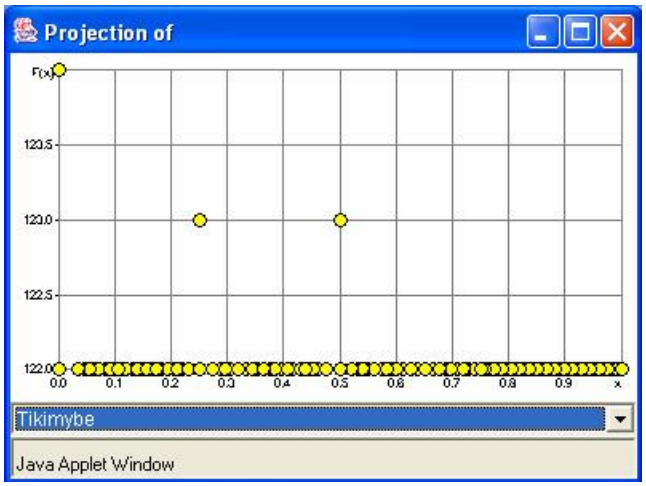
Metodas		Uždavinys	Rezultatas		
Vardas	Iteracijos	Kiek kartų optimizuoja uždavinį	Iteracijos	F(x)	Tikimybė
Exkor	5000	2	100	17.0	0.653
		6	100	20.0	0.327
		13	100	14.0	0.306
		25	100	18.0	0.265
		50	100	15.0	0.204
		100	100	11.0	0.122
	10	2	99	19.0	0.750
		6	99	17.0	0.345
		13	99	13.0	0.750
		25	99	10.0	0.500
		50	99	8.0	0.250
		100	99	8.0	0.250

Paveiksl lis 11 parodo kaip metodo Exkor reik m s priklauso nuo atsitiktini kintam j ;

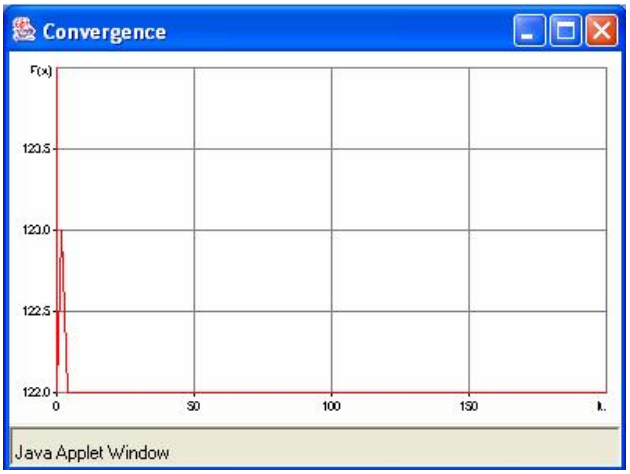
Paveiksl lis 12 parodo kaip metodo Exkor reik m kinta did jant iteracij skai ius;

Paveiksl lis 13 vaizduoja metodo Exkor funkcijos F(x) gautus rezultatus;

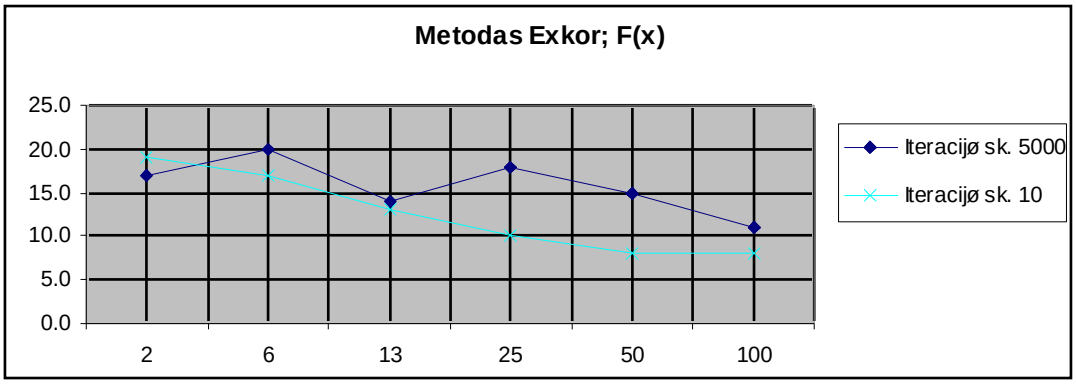
Paveiksl lis 14 vaizduoja prie kokios metodo Exkor tikimyb s buvo gauti F(x) rezultatai;



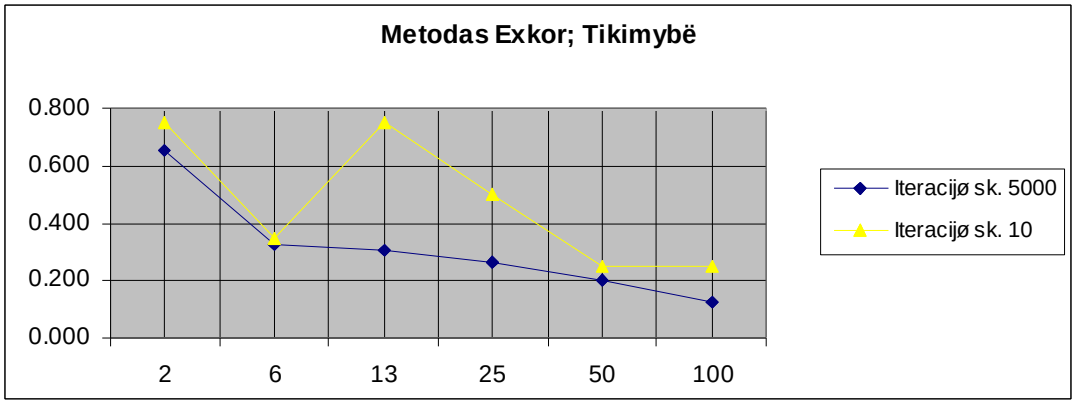
11 pav. Metodo Exkor reikšmės priklausomybė nuo atsitiktiniø kintamøjø;



12 pav. Metodo Exkor reikšmės kitimas didėjant iteracijø skaièiui;



13 pav. Vaizduoja metodo Exkor funkcijos F(x) gautus rezultatus;



14 pav. Vaizduoja prie kokios metodo Exkor tikimybės buvo gauti F(x) rezultatai;

7 lentelė. Gauti rezultatai atliekant bandymus su metodu “Globt”.

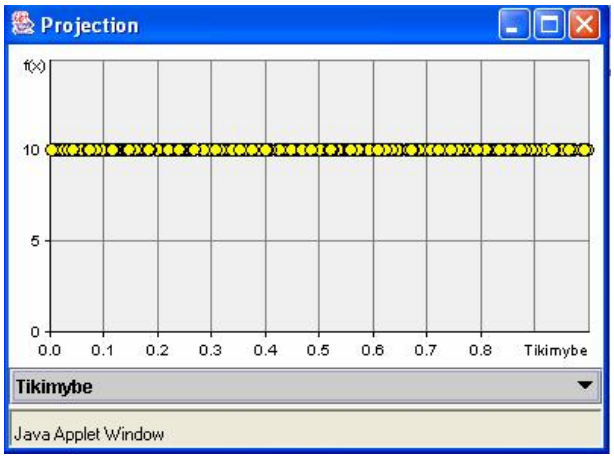
Metodas		Uždavinys	Rezultatas		
Vardas	Iteracijos	Kiek kartų optimizuoja uždavinį	Iteracijos	F(x)	Tikimybė
Globt	5000	2	999	16.0	0.744
		6	999	15.0	0.868
		13	999	8.0	0.721
		25	999	8.0	0.957
		50	999	8.0	0.351
		100	999	8.0	0.151
	10	2	99	13.0	0.016
		6	99	11.0	0.561
		13	99	13.0	0.668
		25	99	9.0	0.911
		50	99	9.0	0.191
		100	99	8.0	0.349

Paveiksl lis 15 parodo kaip metodo Globt reik m s priklauso nuo atsitiktini kintam j ;

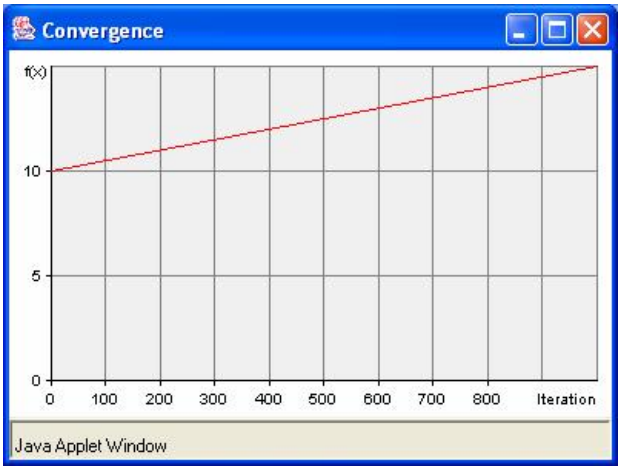
Paveiksl lis 16 parodo kaip metodo Globt reik m kinta did jant iteracij skai ius;

Paveiksl lis 17 vaizduoja metodo Globt funkcijos F(x) gautus rezultatus;

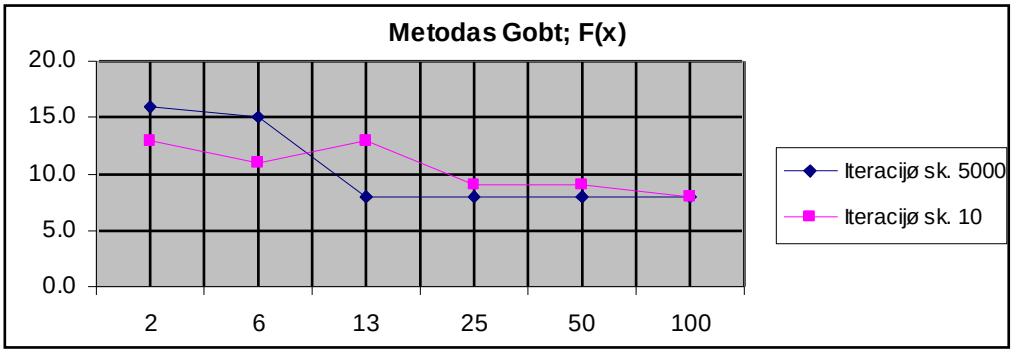
Paveiksl lis 18 vaizduoja prie kokios metodo Globt tikimyb s buvo gauti F(x) rezultatai;



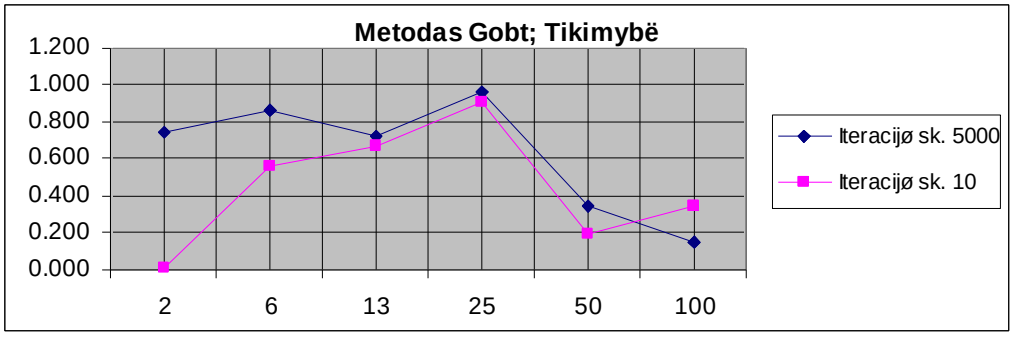
15 pav. Metodo Globt reikšmės priklausomybė nuo atsitiktiniø kintamøjø;



16 pav. Metodo Globt reikšmės kitimas didëjant iteracijø skaièiui;



17 pav. Vaizduoja metodo Globt funkcijos F(x) gautus rezultatus;



18 pav. Vaizduoja prie kokios metodo Globt tikimybës buvo gauti F(x) rezultatai;

8 lentelė. Gauti rezultatai atliekant bandymus su metodu “Mig1”.

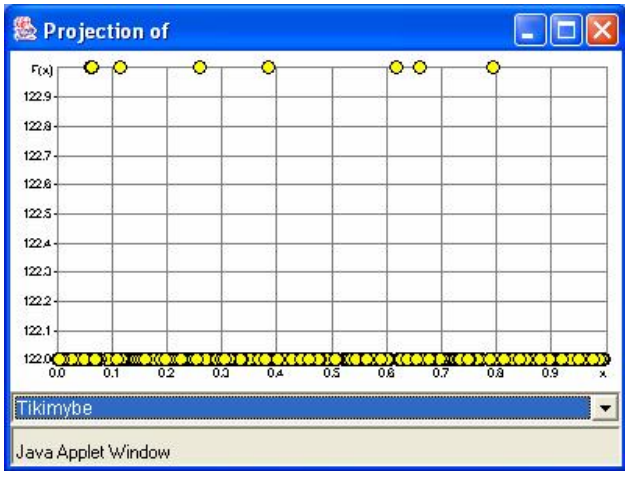
Metodas		Uždavinys	Rezultatas		
Vardas	Iteracijos	Kiek kartų optimizuoja uždavinį	Iteracijos	F(x)	Tikimybė
Mig1	1000	2	999	17.0	0.393
		6	999	13.0	0.239
		13	999	15.0	0.833
		25	999	8.0	0.634
		50	999	9.0	0.656
		100	999	8.0	0.235
	100	2	99	17.0	0.779
		6	99	14.0	0.456
		13	99	13.0	0.692
		25	99	11.0	0.913
		50	99	8.0	0.326
		100	99	9.0	0.979

Paveiksl lis 19 parodo kaip metodo Mig1 reik m s priklauso nuo atsitiktini kintam j ;

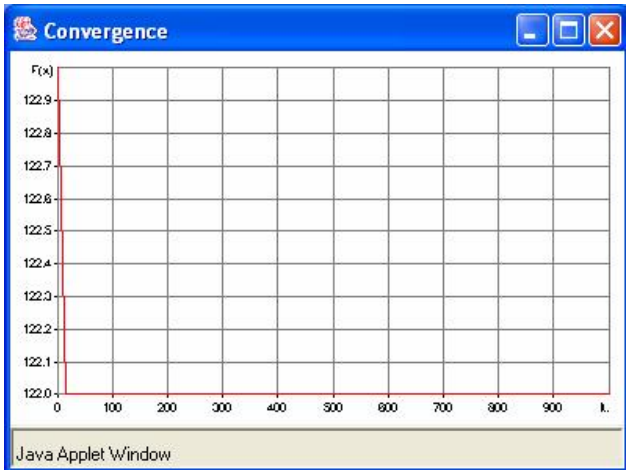
Paveiksl lis 20 parodo kaip metodo Mig1 reik m kinta did jant iteracij skai ius;

Paveiksl lis 21 vaizduoja metodo Mig1 funkcijos F(x) gautus rezultatus;

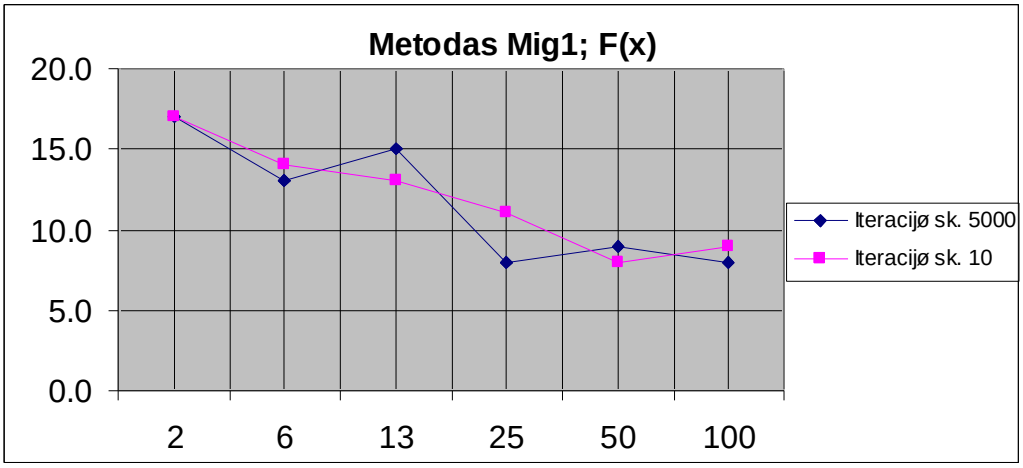
Paveiksl lis 22 vaizduoja prie kokios metodo Mig1 tikimyb s buvo gauti F(x) rezultatai;



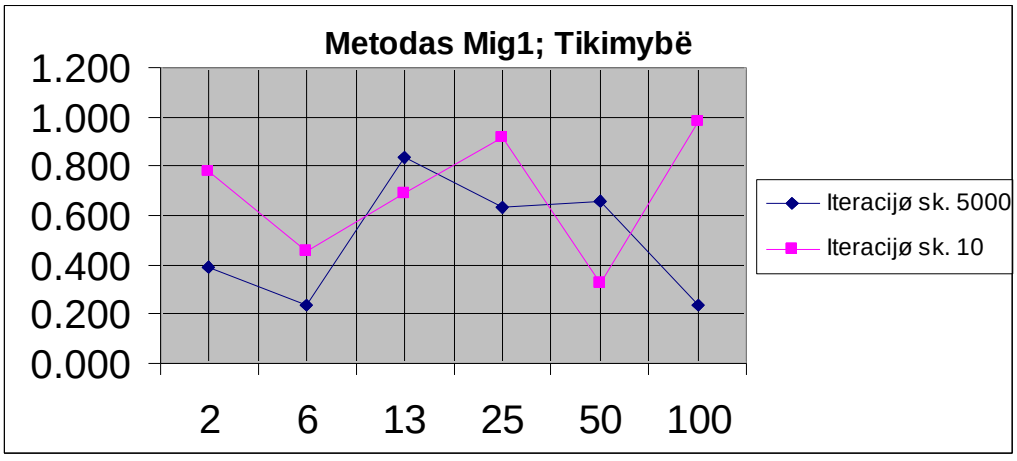
19 pav. Metodo Mig1 reikšmės priklausomybė nuo atsitiktinio kintamojo;



20 pav. Metodo Mig1 reikšmės kitimas didėjant iteracijos skaičiui;



21 pav. Vaizduoja metodo Mig1 funkcijos F(x) gautus rezultatus;



22 pav. Vaizduoja prie kokios metodo Mig1 tikimybės buvo gauti F(x) rezultatai;

5 Gautø rezultatø palyginimas su kitais

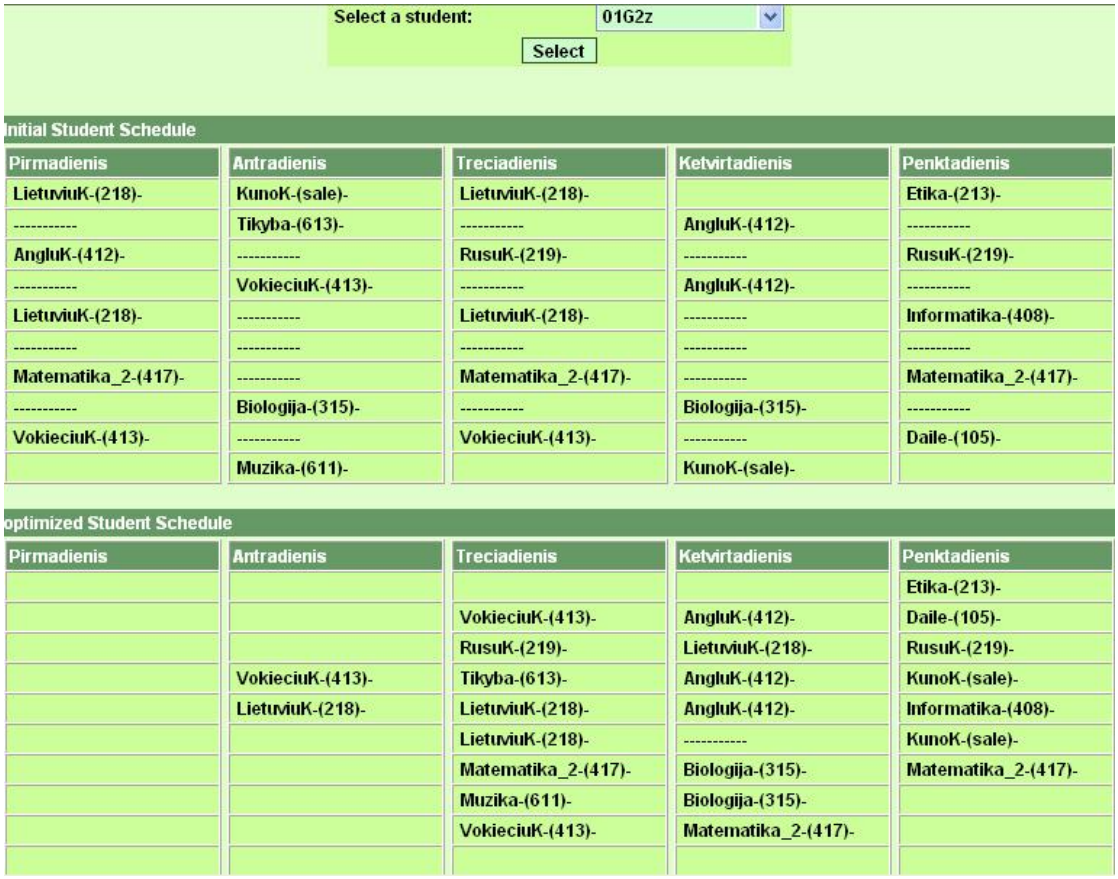
5.1 Gauti rezultatai

9 lentelë. Tiriamos programos palyginimas su kitomis programomis.

Pavadinimas	Programos tikslas	Matomos charakteristikos	Pradiniai duomenys	OS kurioje programa dirba
„School schedule optimization program“	Sudaryti optimal tvarkara t ;	Turi daug optimizavimo algoritm ;	I anksto suvedami duomen fail ;	Windows 2000/XP; Linux
„aSc Timetables 2003“		Nedaug optini- zuoja (da niausia darb reikia pabaigt vartotojui);	Suvedami pa ios programos vykdymo metu;	Windows 9x/2000/XP;
„MIMOSA“	vykdyti visus vartotojo reikalavimus;			
„RECTOR“				

5.2 Gautø rezultatø palyginimas

Apdorojus duomenis programa „School schedule optimization program“ gavau tokius duomenis:



Select a student: 01G2z

Select

Initial Student Schedule

Pirmadienis	Antradienis	Treciadienis	Ketvirtadienis	Penktadienis
LietuviuK-(218)-	KunoK-(sale)-	LietuviuK-(218)-		Etika-(213)-
----------	Tikyba-(613)-	----------	AngluK-(412)-	----------
AngluK-(412)-	----------	RusuK-(219)-	----------	RusuK-(219)-
----------	VokieciuK-(413)-	----------	AngluK-(412)-	----------
LietuviuK-(218)-	----------	LietuviuK-(218)-	----------	Informatika-(408)-
----------	----------	----------	----------	----------
Matematika_2-(417)-	----------	Matematika_2-(417)-	----------	Matematika_2-(417)-
----------	Biologija-(315)-	----------	Biologija-(315)-	----------
VokieciuK-(413)-	----------	VokieciuK-(413)-	----------	Daile-(105)-
	Muzika-(611)-		KunoK-(sale)-	

Optimized Student Schedule

Pirmadienis	Antradienis	Treciadienis	Ketvirtadienis	Penktadienis
				Etika-(213)-
		VokieciuK-(413)-	AngluK-(412)-	Daile-(105)-
		RusuK-(219)-	LietuviuK-(218)-	RusuK-(219)-
	VokieciuK-(413)-	Tikyba-(613)-	AngluK-(412)-	KunoK-(sale)-
	LietuviuK-(218)-	LietuviuK-(218)-	AngluK-(412)-	Informatika-(408)-
		LietuviuK-(218)-	----------	KunoK-(sale)-
		Matematika_2-(417)-	Biologija-(315)-	Matematika_2-(417)-
		Muzika-(611)-	Biologija-(315)-	
		VokieciuK-(413)-	Matematika_2-(417)-	

23 pav. Grupės 2Gž, pirmo pogrupio tvarkaraščio sudarymo rezultatai, gauti programa „School schedule optimization program“;

Select a student: 02G2z

Select

Initial Student Schedule

Pirmadienis	Antradienis	Treciadienis	Ketvirtadienis	Penktadienis
LietuviuK-(218)-	KunoK-(sale)-	LietuviuK-(218)-		Etika-(213)-
----------	Tikyba-(613)-	----------	AngluK-(412)-	----------
AngluK-(412)-	----------	Informatika-(402)-	----------	----------
----------	VokieciuK-(413)-	----------	AngluK-(412)-	----------
LietuviuK-(218)-	----------	LietuviuK-(218)-	----------	RusuK-(219)-
----------	RusuK-(219)-	----------	RusuK-(219)-	----------
Matematika_2-(417)-	----------	Matematika_2-(417)-	----------	Matematika_2-(417)-
----------	Biologija-(315)-	----------	Biologija-(315)-	----------
VokieciuK-(413)-	----------	VokieciuK-(413)-	----------	Daile-(105)-
	Muzika-(611)-		KunoK-(sale)-	

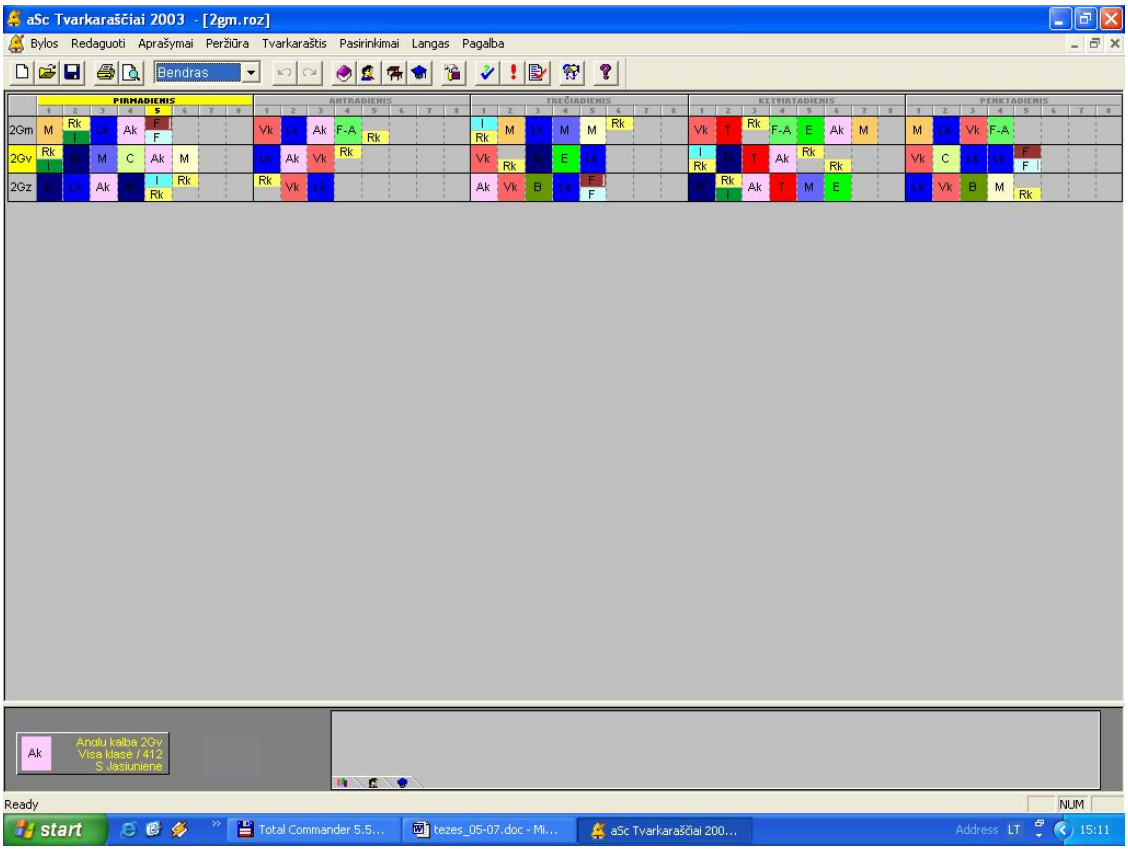
optimized Student Schedule

Pirmadienis	Antradienis	Treciadienis	Ketvirtadienis	Penktadienis
				Etika-(213)-
		VokieciuK-(413)-	AngluK-(412)-	Daile-(105)-
		Informatika-(402)-	LietuviuK-(218)-	----------
	VokieciuK-(413)-	Tikyba-(613)-	AngluK-(412)-	KunoK-(sale)-
	LietuviuK-(218)-	LietuviuK-(218)-	AngluK-(412)-	RusuK-(219)-
	RusuK-(219)-	LietuviuK-(218)-	RusuK-(219)-	KunoK-(sale)-
		Matematika_2-(417)-	Biologija-(315)-	Matematika_2-(417)-
		Muzika-(611)-	Biologija-(315)-	
		VokieciuK-(413)-	Matematika_2-(417)-	

24 pav. Grupės 2Gž, antro pogrupio tvarkaraščio sudarymo rezultatai, gauti programa „School schedule optimization program“;

Kadangi pradiniuose duomenyse grup s buvo suskirstytos pogrupius, tai programa rezultatus taip pat i ved pogrupiais. Tai atspindi ir 23 bei 24 paveikslai. Abiejuose paveiksluose auk iau matomas tvarkara tis parodo kokie buvo pradiniai duomenys (prie optimizavim). emiau matomas tvarkara tis jau yra suoptimizuotas. Taip ioje programoje galima pamatyti suoptimizuotus ir pradinius kiekvieno mokytojo individualius tvarkara ius, bei pilnus suoptimizuot ir pradin mokyklos tvarkara t .

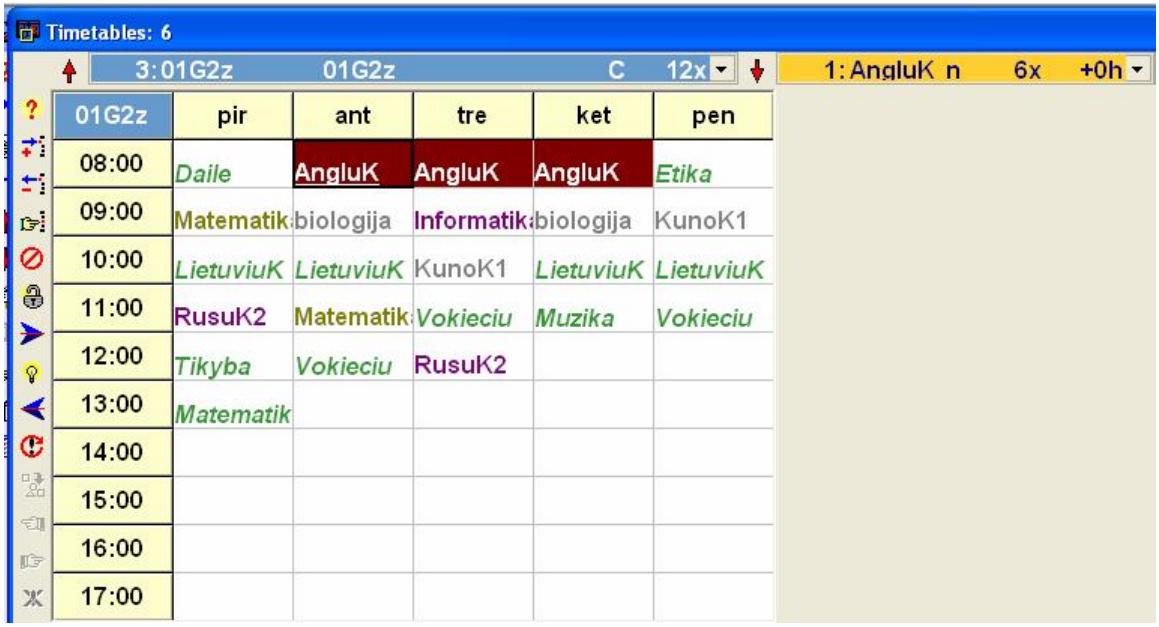
Apdorojus duomenis programa „aSc Timetables 2003“ gavau tokius duomenis:



25 pav. visu trijø grupiø tvarkaraš èio sudarymo rezultatai, gauti programa „aSc Timetables 2003“;

Kadangi pradiniuose duomenyse grup s buvo suskirstytos pogrupius, tai programa rezultatus taip pat i ved pogrupiais. Tai atspindi ir 25 paveikslas. Paveiksle matomas vis vest grupi tvarkara tis. Taip pat ia programa galima pa i r ti bendr (besimokan i moksleivi grupi), mokytoj (vis mokytoj darbo grafik viename lange) bei kabinet u imtumo (vis kabinet u imtumo grafik viename lange) tvarkara ius. Ta iau i programa nei veda atskir grupi ir pogrupi tvarkara i .

Apdorojus duomenis programa „MIMOSA“ gavau tokius duomenis:



01G2z	pir	ant	tre	ket	pen
08:00	Daile	AngluK	AngluK	AngluK	Etika
09:00	Matematik	biologija	Informatik	biologija	KunoK1
10:00	LietuviuK	LietuviuK	KunoK1	LietuviuK	LietuviuK
11:00	RusuK2	Matematik	Vokieciu	Muzika	Vokieciu
12:00	Tikyba	Vokieciu	RusuK2		
13:00	Matematik				
14:00					
15:00					
16:00					
17:00					

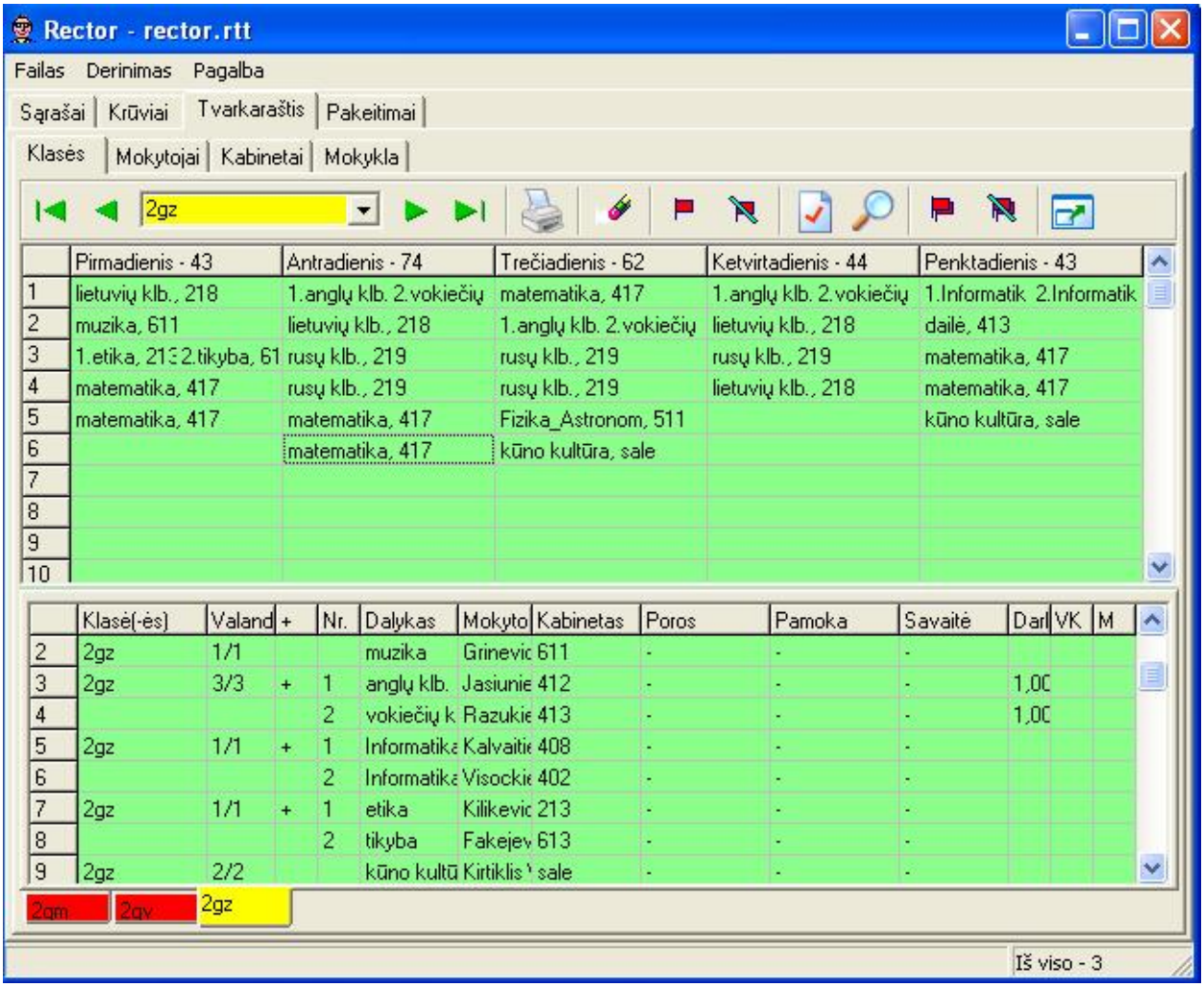
26 pav. Grupės 2Gž, pirmo pogrupio tvarkaraščio sudarymo rezultatai, gauti programa „MIMOSA“;

02G2z	pir	ant	tre	ket	pen
08:00	Daile	AngluK	AngluK	AngluK	Etika
09:00	Matematik	biologija	Informatik	biologija	KunoK1
10:00	LietuviuK	LietuviuK	KunoK1	LietuviuK	LietuviuK
11:00	Rusuk	Matematik	Vokieciu	Muzika	Vokieciu
12:00	Tikyba	Vokieciu	Rusuk	Rusuk	
13:00	Matematik				
14:00					
15:00					
16:00					
17:00					

27 pav. Grupės 2Gž, antro pogrupio tvarkaraščio sudarymo rezultatai, gauti programa „MIMOSA“;

Kadangi pradiniuose duomenyse grup s buvo suskirstytos pogrupius, tai programa rezultatus taip pat i ved pogrupiais. Abiejuose 26 ir 27 paveiksluose matomi tik vienos grup s tvarkara iai. Programa parodo tik atskirus kiekvienos grup s (jei tvarkara tis yra vestas grup mis) ar pogrupio (jei tvarkara ti yra vestas pogrupiais) tvarkara ius.

Apdorojus duomenis programa „RECTOR“ gavau tokius duomenis:



28 pav. Grupės 2Gž tvarkaraščio sudarymo rezultatai, gauti programa „RECTOR“;

Kadangi pradiniuose duomenyse grup s nebuvo suskirstytos pogrupius, tai programa rezultatus taip pat i ved grup mis. Tai atspindi ir 28 paveikslas. Paveiksle matomas vienos grup s tvarkara tis, kur vartotojas tur jo sud lioti savo rankomis. Programa d lioti tvarkara io net nepadeda. Paveikslo apa ioj raudonam fone matomos kitos vestos grup s.

5.3 „MIMOSA“ ir „aSc Timetables 2003“

Abi programos yra labai pana ios. J pagrindin id ja yra labai paprasta. Prakti kai ne manoma automati kai sukurti gero tvarkara io. Tod l programa „MIMOSA“ to padaryti ir nesistengia - ji tik sukuria aplink maksimaliai patogiai sud lioti ir derinti tvarkara t . Ji pateikia efektyvi ir reikaling informacij , ta iau vartotojas tvarkara t turi derinti rankiniu b du. Tai labai sud tinga kai duomen yra daug. Ir jei reikt k nors pakeisti pradiniuose duomenyse – prakti kai reikt perdaryti vis tvarkara t i naujo. Abi programos reikalauja apie 50% mogi k j resurs . Programos „aSc Timetables 2003“ privalumas prie program „MIMOSA” yra tas, kad ji yra i leista lietuvi kalba. Programa „aSc Timetables 2003“ stengiasi sudaryti optimalesn pamok tvarkara t , nei programa „MIMOSA“, bet tai irgi n ra pilnai gyvendinta. i programa turi vien prana um , prie program „MIMOSA“: duomenis ji skaito ir transformuoja programos „MS Excel“ format . Programa vest tvarkara t suoptimizuoja, bet ne visada sudaro piln tvarkara t . Abi programos nereikalauja nei ypating kompiuteri . Programa „MIMOSA“, prie ingai nei programa „aSc Timetables2003“, reikalauja ypatingo vartotojo pasiruo imo. Abi programos tur t veikti bet kuriame kompiuteryje, kuriame veikia Windows95 operacin sistema.

5.4 „RECTOR“

i programa nieko nesiskiria nuo programos „MS Excel“. Ji pati nei i d lioja duomenis, nei juos optimizuoja. Ji tik suteikia vartotojui patogi aplink , kad sud lioti pradinius duomenis. Tai s kmingai gali padaryti ir korporacijos „Microsoft“ „MS Office“ paketo programos „MS Word“ arba „MS Excel“.

6 Programinės árangos realizacija

6.1 Sistemos funkcinis aprašymas

i sistema yra skirta optimizuoti profiliuotos mokyklos tvarkara iui. Ji skirta kiek manoma suma inti mokini ir mokytoj lang skai i ir sudaryti kiek manoma optimalesn mokymo staigos tvarkara t .

6.2 Vartotojo atmintinė

Kadangi i tvarkara i optimizavimo programa (toliau programa) yra padaryta servlet (“Java” programinis kodas) re imu, tai tam kad ji veikt , pirmiausia turi b ti diegta serverio valdymo programa “Tomcat”. Tai yra lokalus personalinio kompiuterio serveris, kuris padeda vartotojui dirbti su ia programa nesinaudojant papildomo serverio pagalba. Taip pat galima ia programa talpinti server ir dirbti per kompiuterin tinkl .

Dirbant su personaliniu kompiuteriu vartotojas prie paleisdamas program , pirmiausia turi paleisti serverio program “Tomcat”. Po to paleid s “Microsoft Intermet Explorer” program , jis gali prad ti dirbti su programa.

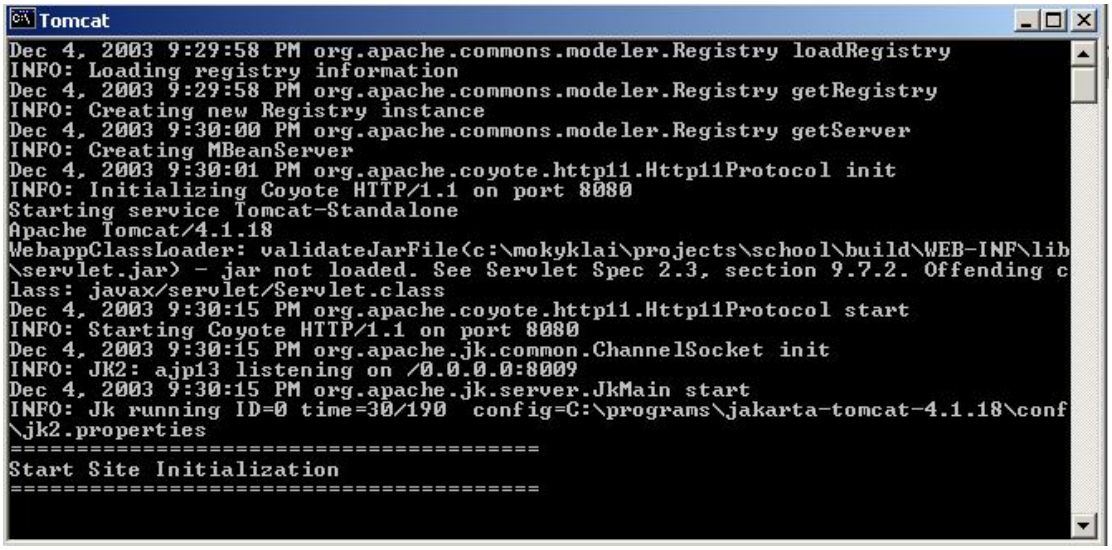
Dirbant su ia programa, kiekviename ingsnyje yra “Help” nuoroda su pagalba lietuvi ir angl kalbomis. Tuose failiukuose yra apra yta, k konkre ioje vietoje esantis u ra as rei kia.

6.3 Detalioji sistemos atmintinė

Duomen failo sudarymas yra apra ytas 3.4 sk.

6.3.1 Serverio startavimas

Suvedus mokymo staigos pradin tvarkara t “NOTEPAD” failiuk ir i saugojus j norimu pavadinimu su pl tiniu “txt”, vartotojas pirmiausia turi paleisti serverio program “Tomcat”:



29 pav. Serverio programos “Tomcat” darbo langas;

ia yra matoma kada ir kokia operacij atliko vartotojas. Taip pat testuojant program lang yra i vedama norima patikrinti testuotojo informacija.

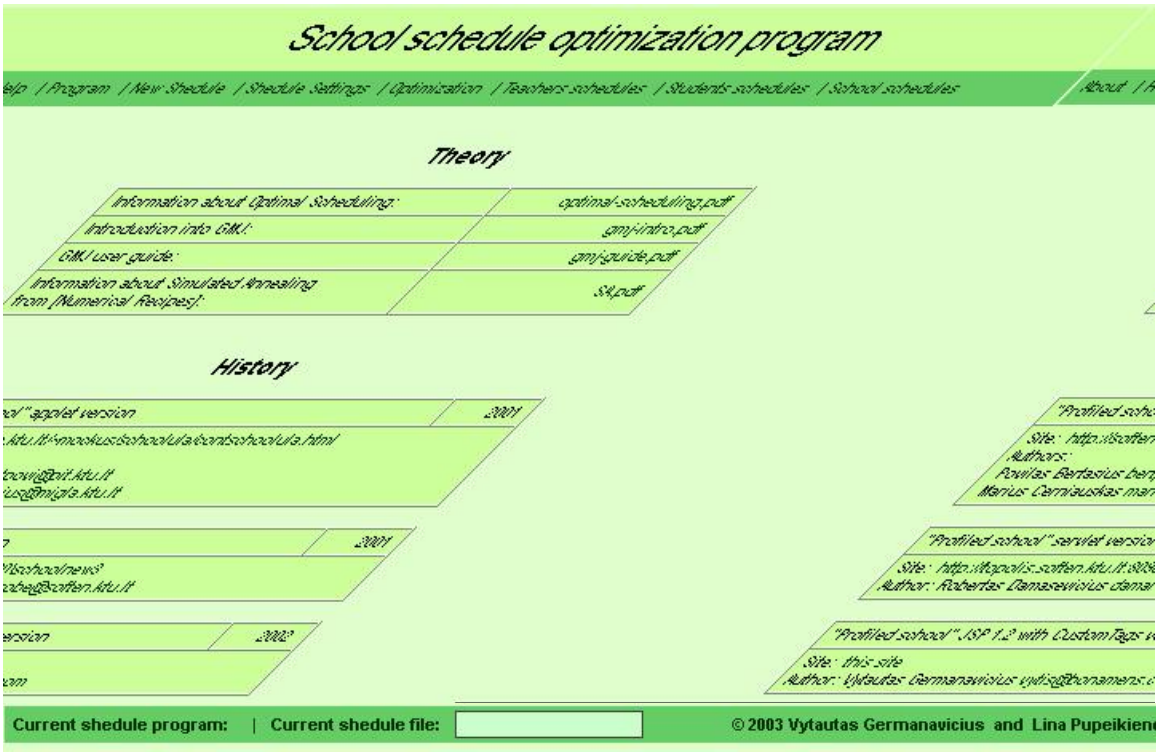
6.4 Programos startavimas:

Kad vartotojas gal t prad ti dirbti su programa, jis “Microsoft Internet Explorer” lango adreso vietoje turi surinkti lokalaus serverio adres . iai programai bus naudojamas adresas:

http://localhost:8080/NBMS

6.4.1 Pradinis programos langas:

iuo adresu i kvie iama programa atrodo taip:



30 pav. Programos pradinis langas arba meniu funkcija “About”;

iame lange yra:

- programos pavadinimas (kuris i lieka visuose tolimesniuose puslapiuose);
- programos valdymo meniu (kuris i lieka visuose tolimesniuose puslapiuose);
- pateikta informacija apie programos k rimo etapus (“History”) ir naudot literat r (“Theory”).
- Apa ioje esan ioje juostoje yra parodoma kokia optimizavimo programa yra naudojama (“Current shedule program:”), koks duomen failas (duomen failo pavadinimas) yra nuskaitytas (“Current shedule file:”) ir kas yra programos autoriai.

lang (jei vartotojas pageidauja) galima i kviesti paspaudus meniu punkt “About”.

6.4.2 Programos meniu “help” funkcija

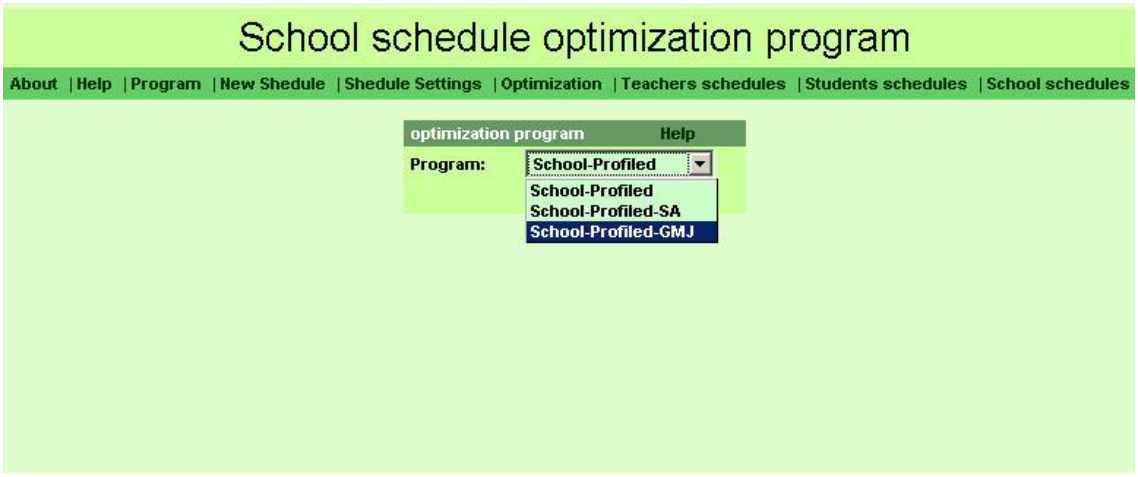
Pasirink s programos meniu punkt “Help” vartotojas i vys:

31 pav. Pasirinkta meniu funkcija “Help”;

ia yra apra yta kaip naudotis programa ingsnis po ingsnio.

6.4.3 Optimizavimo programos pasirinkimas

Pasirink s programos meniu punkt “Program” vartotojas i vys:



32 pav. Pasirinkta meniu funkcija “Programs”;

Pirmiausia vartotojas turi pasirinkti vien i trij optimizavimo program :

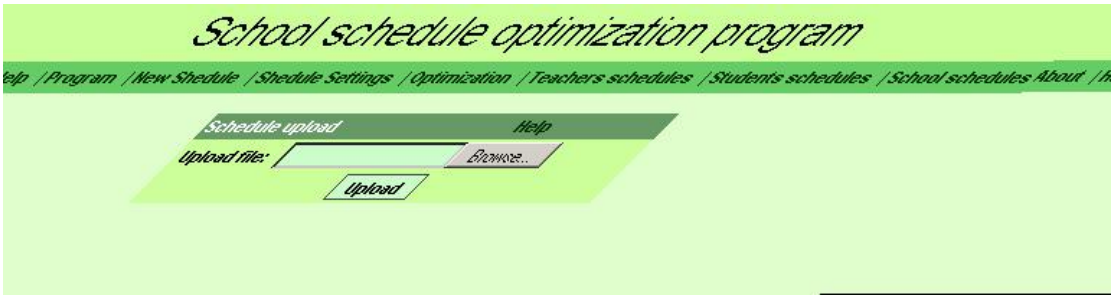
1. “Profiled school” - yra pati papras iausia optimizavimo programa. Ji dirba grei iausia ir tvarkara t optimizuoja ma iau nei likusios optimizavimo programos.
2. “School-profiled-SA” - yra vidutinio sud tingumo optimizavimo programa. Ji dirba ilgiau ir tvarkara t optimizuoja daug geriau nei pirmoji optimizavimo programa.

3. “School-profiled-GMJ” - yra sud tingiausia optimizavimo programa. Ji dirba ilgiausia ir tvarkara t optimizuoja daugiau nei prie tai buv optimizavimo programos.

Pasirinkus bet kuri i i optimizavimo program , reikia paspausti mygtuk “Select”, esant po optimizavimo program meniu.

6.4.4 Duomenø ávedimas

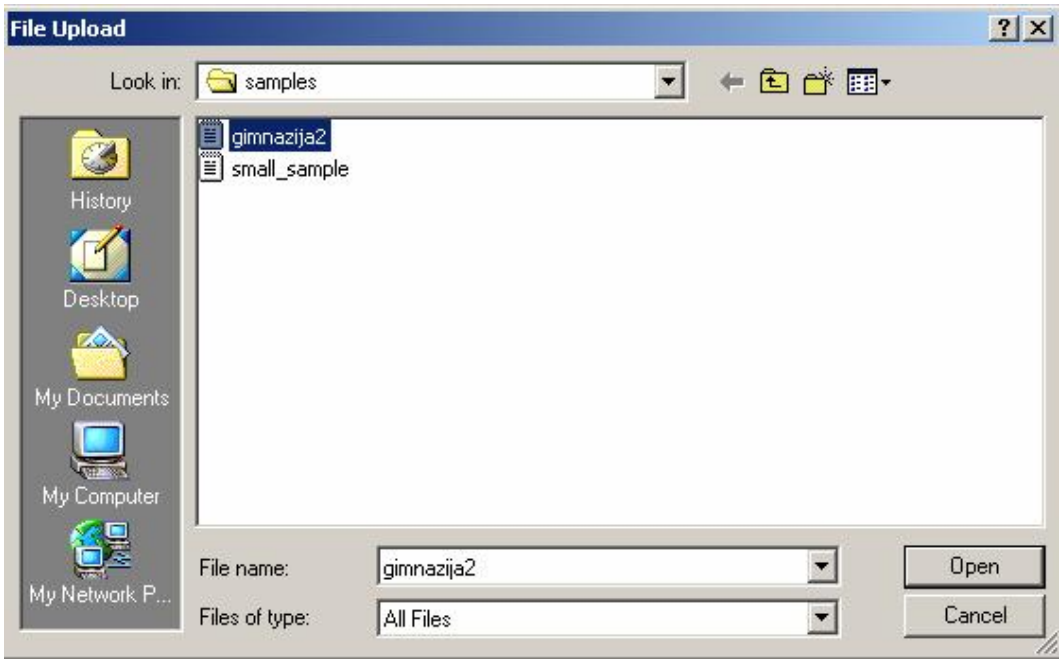
Pasirinkus bet kuria optimizavimo program ir paspaudus mygtuk “Select” atsiveria naujas langas, kuriame turime nurodyti fail su optimizuojamais duomenimis:



33 pav. Duomenø failo nuskaitymo langas (meniu punktas “New Shedule”);

Duomenø failo suradimas kompiuteryje

Paspaudus mygtuk “Browse…” atsiranda naujas langas:

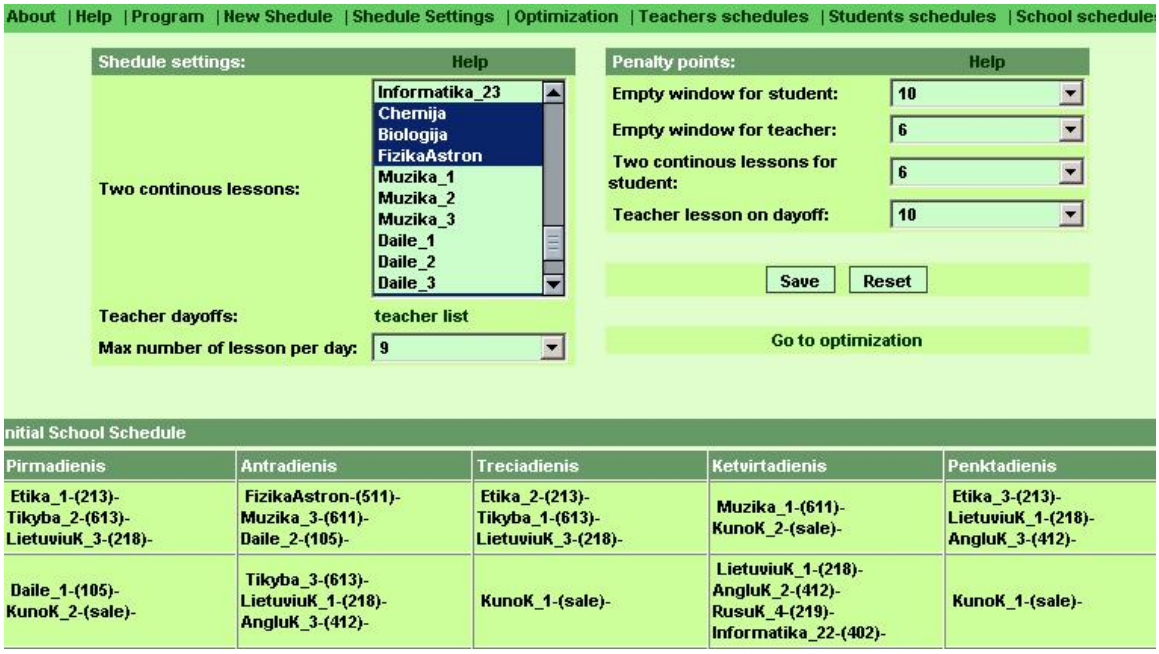


34 pav. Duomenø failo išrinkimas

Suradus kompiuteryje ir pasirinkus duomen fail , paspaud iame mygtuk “Open” ir automati kai gr tame duomen failo nuskaitymo lang . Tada paspaud iame mygtuk “Upload”. Taip programa nuskaito vartotojo pasirinktus duomenis.

6.4.5 Optimizavimo kriterijø pasirinkimas

Nuskai iusi duomenis programa automati kai pereina meniu punkto “Schedule settings” lang , kur reikia nurodyti duomen optimizavimo kriterijus:



35 pav. Duomenø optimizavimo kriterijø pasirinkimas;

Viduje yra trys ma esni langeliai:

1. “Schedule settings:”

- TWO CONTINIOUS LESSONS (dvi, viena paskui kit , i d stytos negalimos pamokos) - Nurodo kiek svarbu, kad negalimos pamokos neb t i d stytos viena paskui kit . Kuo didesnis baudos ta k skai ius, tuo ma esn tikimyb , kad dvi negalimos paskaitos bus i d stytos viena paskui kit .

- TECHER DAYOFFS (mokytoj laisvadieniai) - Mokytojas gali pasirinkti, kurias dvi darbo dienas per savaite nor t nedirbti.

- MAX NUMBER OF LESSON PER DAY (maksimalus pamok skai ius per dien) - Maksimalus pamok skai ius per dien

2. “Penalty points:”

- PENALTY POINTS (baudos ta□kai) - Baudos ta□kais galima nustatyti tvarkara□□io sudarym□: Kuo didesnis baudos ta□k□ skai□ius prie konkretaus po□ymio, tuo ma□esn□ tikimyb□, kad tas po□ymis pasitaikys tvarkara□tyje.

- EMPTY WINDOW FOR STUDENT (mokinio tvarkara□tyje esantis langas) - Parodo kaip svarbu, kad mokinys tur□t□ kiek □manoma ma□iau lang□.

- EMPTY WINDOW FOR TEACHER (mokytojo tvarkara□tyje esantis langas) - Parodo kaip svarbu, kad mokytojas tur□t□ kiek □manoma ma□iau lang□.

- TWO CONTINUES LESSONS (dvi, viena paskui kit□, i□d□stytos negalimos pamokos) - Nurodo kiek svarbu, kad negalimos pamokos neb□t□ i□d□stytos viena paskui kit□.

- TEACHER LESSON ON DAYOFF (pamokos per mokytoj□ “laisvadienius”) - Parodo kaip svarbu, kad mokytojas tur□t□ laisvas dienas, kurias jis pasirinko.

- Mygtukas “Save” - rei□kia norim□ (pasirinkt□) duomen□ i□saugojim□.

- Mygtukas “Reset” - rei□kia pradini□ duomen□ atstatym□.

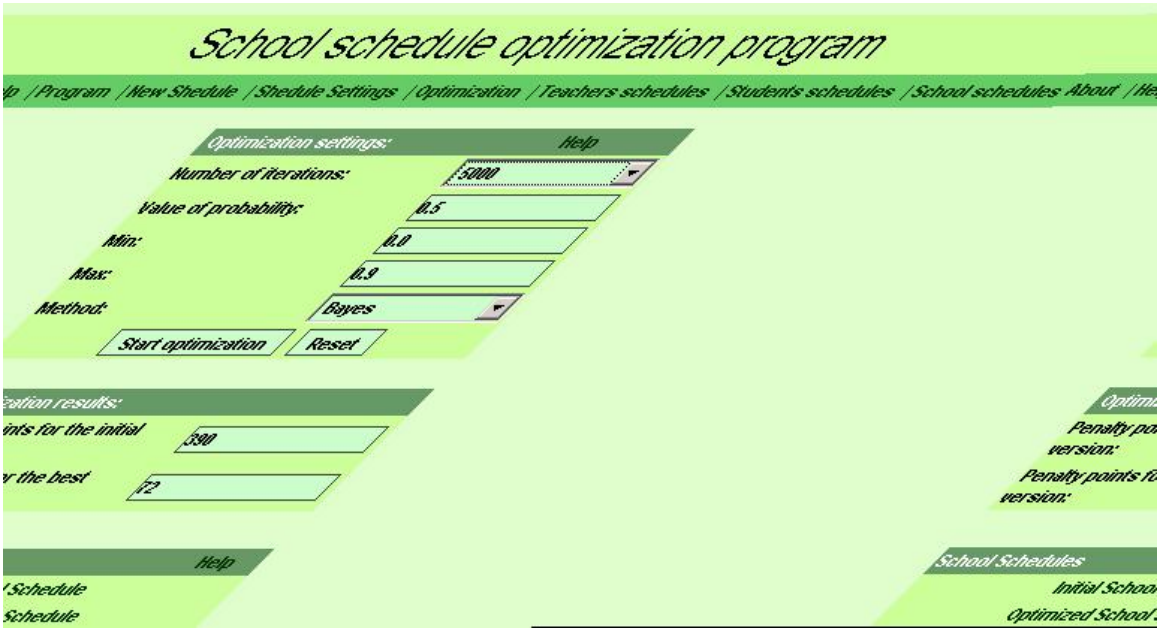
- “Go to optimization” – atsiveria optimizavimo parametr□ nustatymo langas;

3. Initial School Schedule

- Parodo pradin□ mokyklos tvarkara□t□.

6.4.6 Optimizavimo parametrø parinkimas

Optimizavimo parametr□ nustatymo langas:



36 pav. Optimizavimo parametrø nustatymas;

Kaip ir “optimizavimo kriterij□ pasirinkimo” lange, taip ir □iame lange yra trys ma□esni langeliai:

1. “Optimization settings:”

   - NUMBER OF ITERATION (iteracij□ kiekis) - Nurodo kaip tiksliai programa sudarys tvarkara□t□. Kuo didesnis iteracij□ kiekis, tuo programa parinks tikslesn□ tvarkara□□io variant□.
   - VALUE OF PROBABILITY (tikimyb□ pakeisti mokytoj□) - Kuo didesn□ tikimyb□, tuo grei□iau programa pakeis mokytojo tvarkara□t□ □ geresn□. Tikimyb□
     kinta nuo 0.0 iki 0.9.
   - X1 OR MIN (X1 arba MIN) - Tai globalizacijos laipsnis (kar□tumas). Jis parodo kiek energingai programa ie□ko naujo tvarkara□□io varianto.
   - X2 OR MAX (X2 arba MAX) - □alimo greitis. Jis parodo kiek greitai programa baigia paie□kas ("at□□la").
   - METHOD (metodas) – Tvarkara□□io optimizavimo metodas.

2. “Optimization results:”

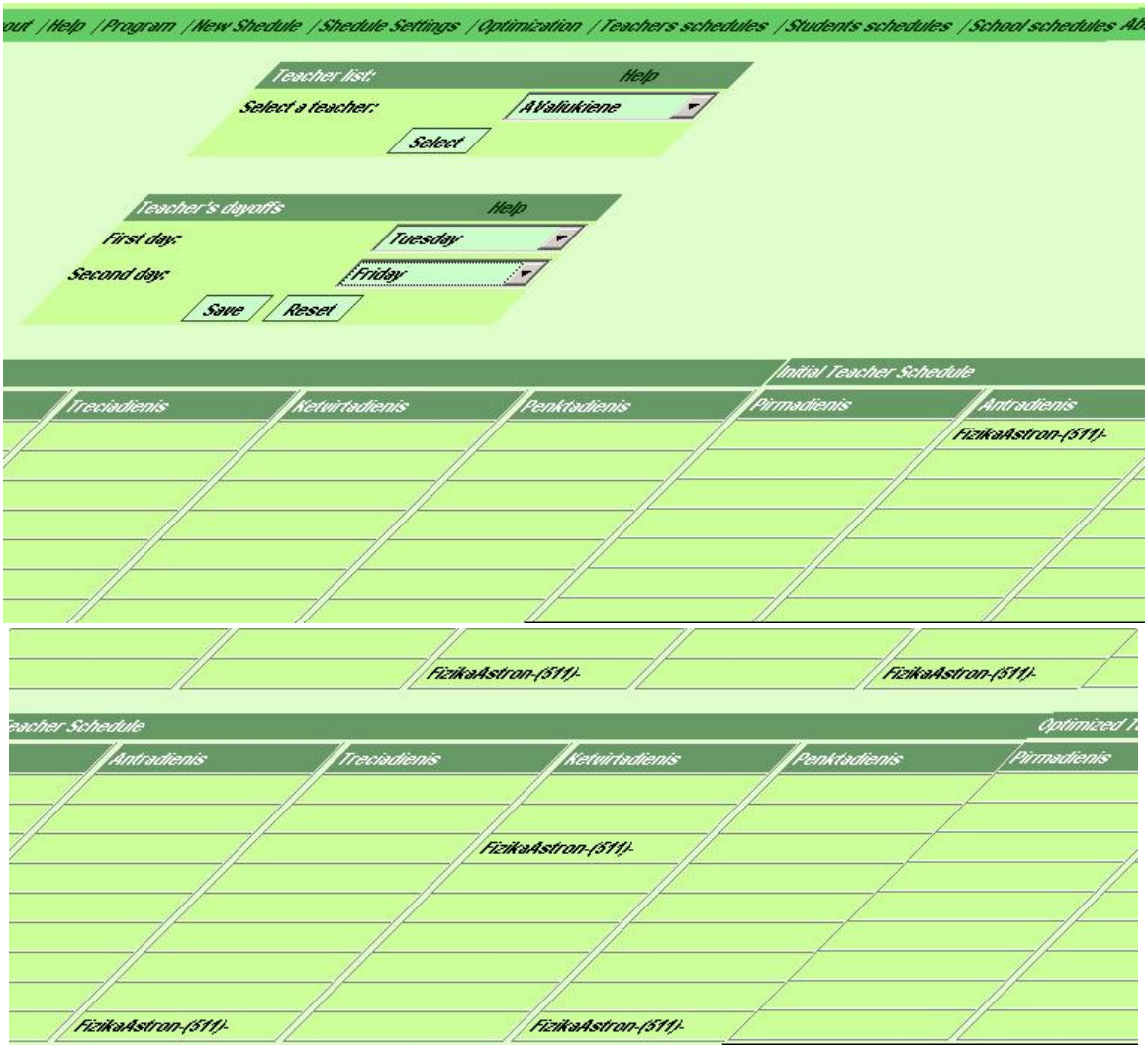
   - Penalty points for the initial version (baudos ta□k□ skai□iavimas pradin□je tvarkara□□io versijoje) – Suskai□iuoja kiek i□ viso baudos ta□k□ yra pradiniame tvarkara□tyje.
   - Penalty points for the best version version (baudos ta□k□ skai□iavimas optimizuotoje tvarkara□□io versijoje) – Suskai□iuoja kiek i□ viso baudos ta□k□ yra optimizuotame tvarkara□tyje.

3. “School Schedules:”

   - INITIAL SCHOOL SCHEDULE (pradinis mokyklos tvarkara□tis) - Pradinis mokyklos tvarkara□tis (prie□ optimizavim□).
   - OPTIMIZED SCHOOL SCHEDULE (optimizuotas mokyklos tvarkara□tis) - Suoptimizuotas mokyklos tvarkara□tis.

6.4.7 Pasirinkto mokytojo tvarkaraščiai

Pasirinkus programos meniu punktą “Teachers schedules” vartotojas išvys tokį langą:

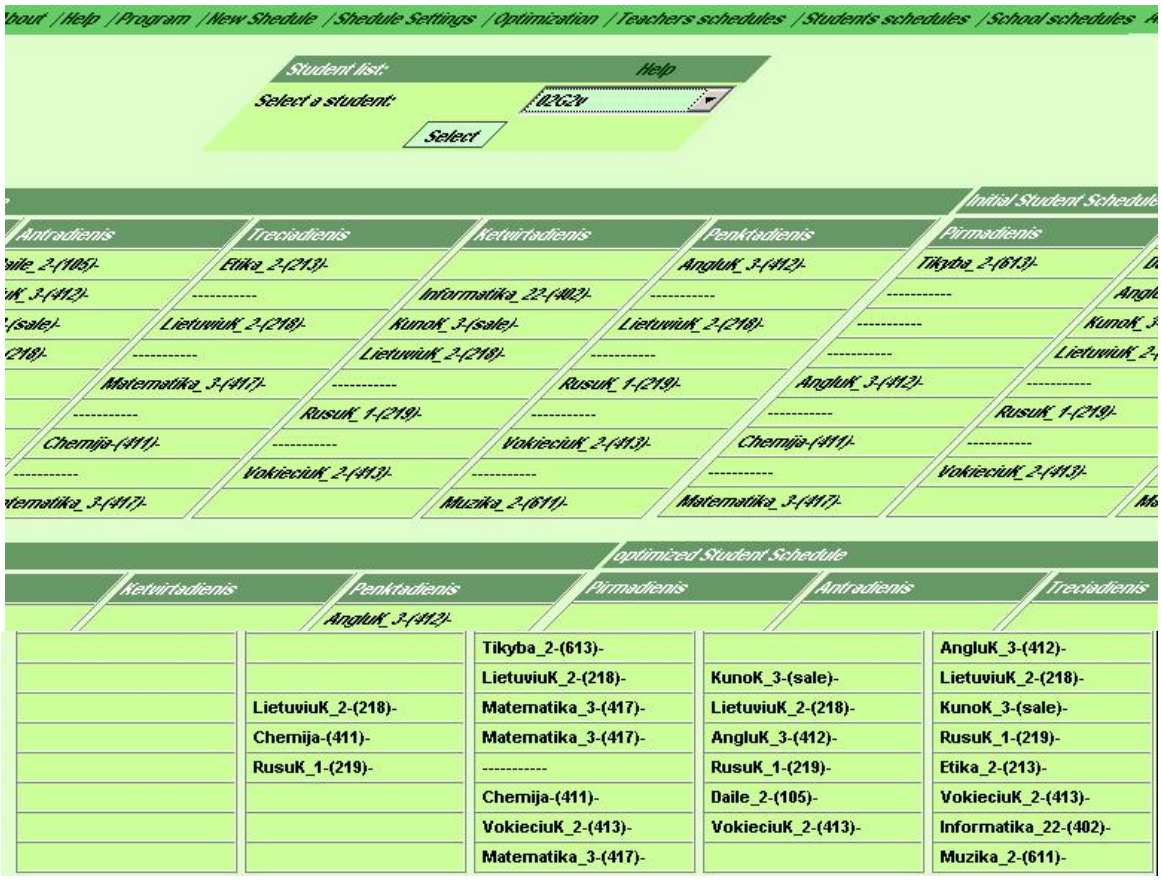


37 pav. Pasirinkto mokytojo tvarkaraščiai;

TEACHER LIST (mokytojo tvarkaraštis) - Pasirinkus mokytoją bus išvedamas jo laisvadienių pasirinkimo lentelė (Teacher's dayoffs), neoptimizuotas (Initial Teacher Schedule) ir optimizuotas (Optimized Teacher Schedule) tvarkaraščiai.

6.5 Pasirinkto mokinio tvarkaraščiai

Pasirinkus programos meniu punktą “Students schedules” vartotojas išvys tokį langą:

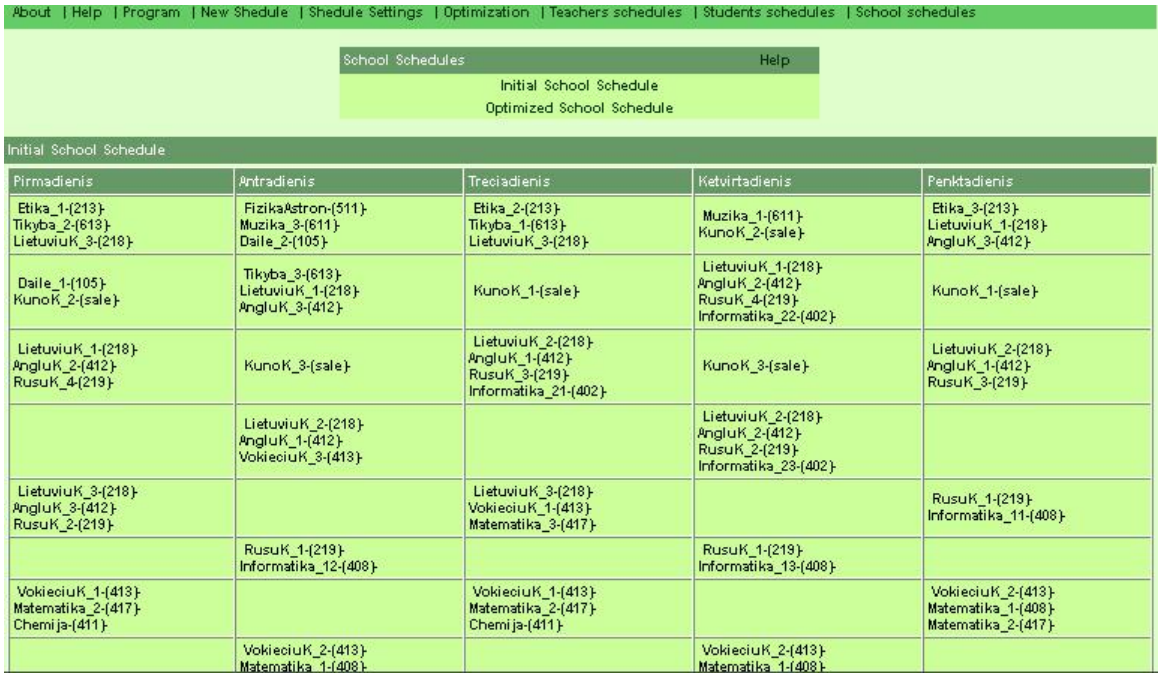


38 pav. Pasirinkto mokinio tvarkaraščiai;

STUDENT SCHEDULE (mokinio tvarkaraštis) - Pasirinkus mokinį bus išvedamas jo neoptimizuotas (Initial Student Schedule) ir optimizuotas (Optimized Student Schedule) tvarkaraščiai.

6.5.1 Mokyklos tvarkaraš è iai:

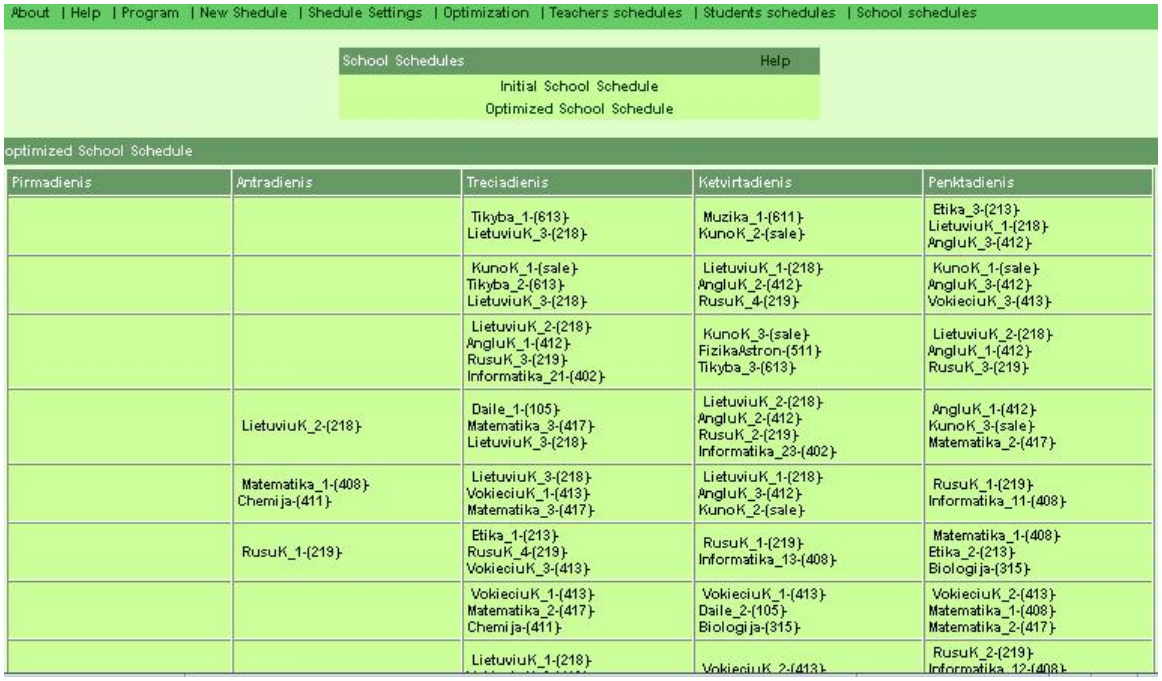
Pasirink□s programos meniu punkt□ “School schedules” vartotojas i□vys tok□ lang□:



Pirmadienis	Antradienis	Treciadienis	Ketvirtadienis	Penktadienis
Etika_1-(213) Tikyba_2-(613) LietuviuK_3-(218)	FizikaAstron-(511) Muzika_3-(611) Daile_2-(105)	Etika_2-(213) Tikyba_1-(613) LietuviuK_3-(218)	Muzika_1-(611) KunoK_2-(sale)	Etika_3-(213) LietuviuK_1-(218) AngluK_3-(412)
Daile_1-(105) KunoK_2-(sale)	Tikyba_3-(613) LietuviuK_1-(218) AngluK_3-(412)	KunoK_1-(sale)	LietuviuK_1-(218) AngluK_2-(412) RusuK_4-(219) Informatika_22-(402)	KunoK_1-(sale)
LietuviuK_1-(218) AngluK_2-(412) RusuK_4-(219)	KunoK_3-(sale)	LietuviuK_2-(218) AngluK_1-(412) RusuK_3-(219) Informatika_21-(402)	KunoK_3-(sale)	LietuviuK_2-(218) AngluK_1-(412) RusuK_3-(219)
	LietuviuK_2-(218) AngluK_1-(412) VokieciuK_3-(413)		LietuviuK_2-(218) AngluK_2-(412) RusuK_2-(219) Informatika_23-(402)	
LietuviuK_3-(218) AngluK_3-(412) RusuK_2-(219)		LietuviuK_3-(218) VokieciuK_1-(413) Matematika_3-(417)		RusuK_1-(219) Informatika_11-(408)
	RusuK_1-(219) Informatika_12-(408)		RusuK_1-(219) Informatika_13-(408)	
VokieciuK_1-(413) Matematika_2-(417) Chemija-(411)		VokieciuK_1-(413) Matematika_2-(417) Chemija-(411)		VokieciuK_2-(413) Matematika_1-(408) Matematika_2-(417)
	VokieciuK_2-(413) Matematika_1-(408)		VokieciuK_2-(413) Matematika_1-(408)	

39 pav. Pradinis mokyklos tvarkaraš tis;

Pasirinkus □iame lange “Optimized School Schedule” vartotojas i□vys:



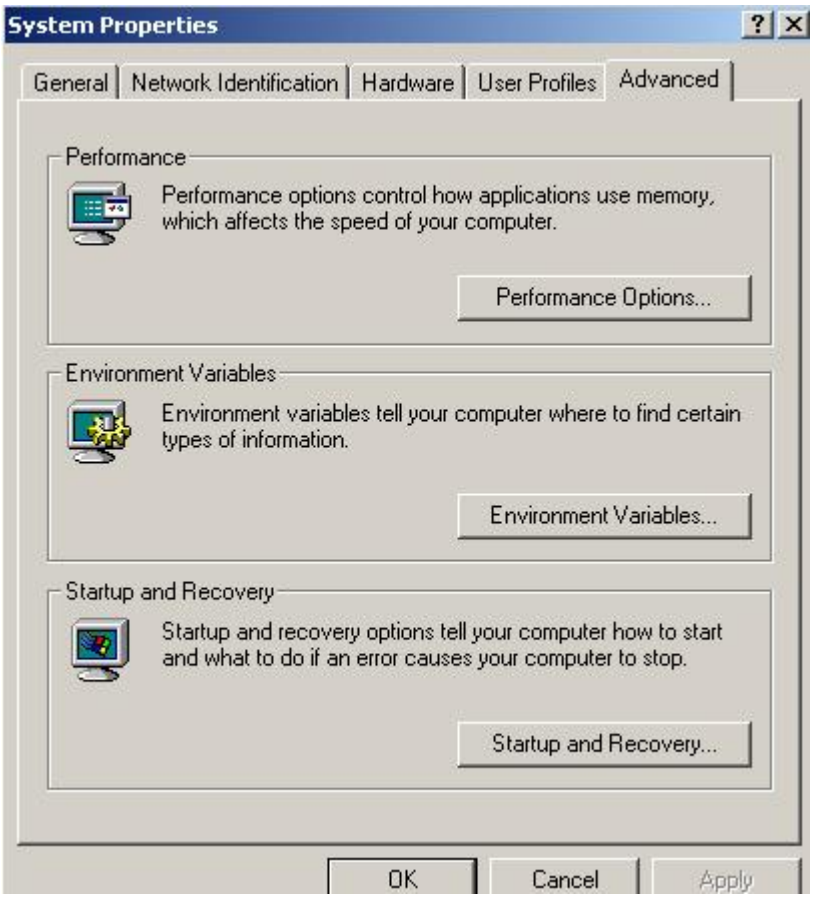
Pirmadienis	Antradienis	Treciadienis	Ketvirtadienis	Penktadienis
		Tikyba_1-(613) LietuviuK_3-(218)	Muzika_1-(611) KunoK_2-(sale)	Etika_3-(213) LietuviuK_1-(218) AngluK_3-(412)
		KunoK_1-(sale) Tikyba_2-(613) LietuviuK_3-(218)	LietuviuK_1-(218) AngluK_2-(412) RusuK_4-(219)	KunoK_1-(sale) AngluK_3-(412) VokieciuK_3-(413)
		LietuviuK_2-(218) AngluK_1-(412) RusuK_3-(219) Informatika_21-(402)	KunoK_3-(sale) FizikaAstron-(511) Tikyba_3-(613)	LietuviuK_2-(218) AngluK_1-(412) RusuK_3-(219)
	LietuviuK_2-(218)	Daile_1-(105) Matematika_3-(417) LietuviuK_3-(218)	LietuviuK_2-(218) AngluK_2-(412) RusuK_2-(219) Informatika_23-(402)	AngluK_1-(412) KunoK_3-(sale) Matematika_2-(417)
	Matematika_1-(408) Chemija-(411)	LietuviuK_3-(218) VokieciuK_1-(413) Matematika_3-(417)	LietuviuK_1-(218) AngluK_3-(412) KunoK_2-(sale)	RusuK_1-(219) Informatika_11-(408)
	RusuK_1-(219)	Etika_1-(213) RusuK_4-(219) VokieciuK_3-(413)	RusuK_1-(219) Informatika_13-(408)	Matematika_1-(408) Etika_2-(213) Biologija-(315)
		VokieciuK_1-(413) Matematika_2-(417) Chemija-(411)	VokieciuK_1-(413) Daile_2-(105) Biologija-(315)	VokieciuK_2-(413) Matematika_1-(408) Matematika_2-(417)
		LietuviuK_1-(218)	VokieciuK_2-(413)	RusuK_2-(219) Informatika_12-(408)

40 pav. Optimizuotas mokyklos tvarkaraš tis;

6.6 Sistemos instaliavimas

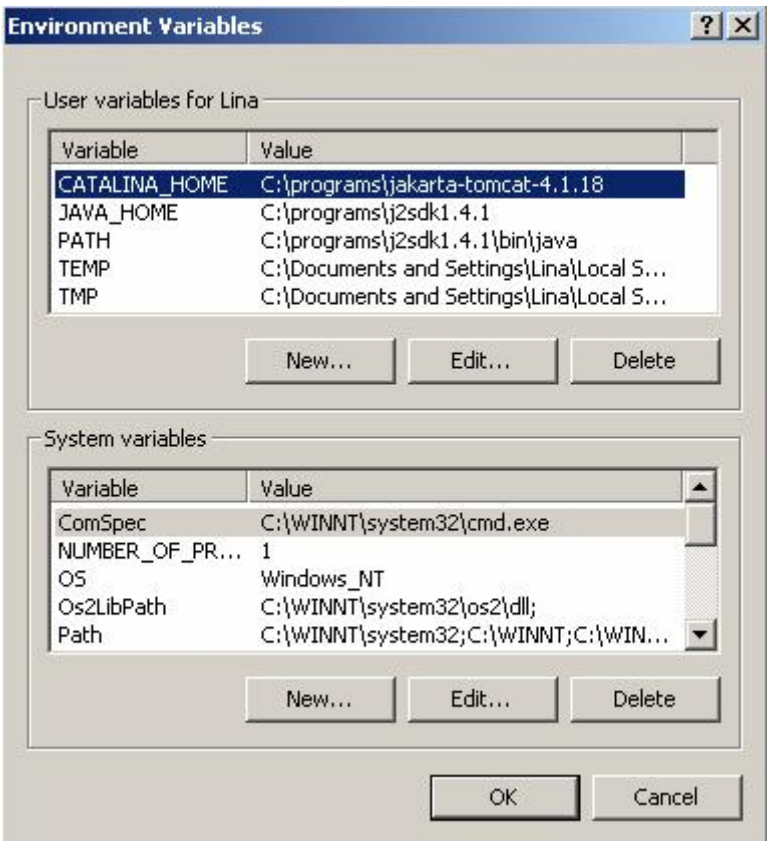
6.6.1 Programinė s á rangos instaliavimas

Sistemai instaliuoti bus reikalinga tik perkopijuoti du katalogus □ “C:/” disk□. Tada reik□s paspausti ant ikonos (esan□ios darbalaukyje “Desktop”) “My Computer” de□in□j□ pel□s klavi□□ ir pasirinkti meniu punkt□ “Properties”. Atsiradusiame lange vartotojas turi pasirinkti “puslap□” “Advanced”:



41 pav. “My Computer” ikonos “Properties”;

Jame reikia paspausti “Enviroment Variables” mygtuk□:



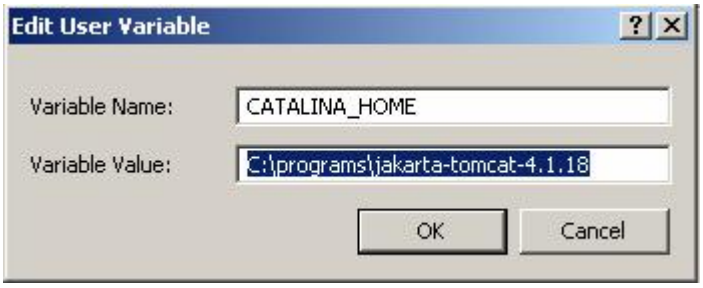
42 pav. “Enviroment Variables”;

□iame langelyje po u□ra□u “User variables” reikia kad b□t□ tokie nustatymai:

10 lentelė . Serverio parametrø nustatymai personaliniame kompiuteryje.

Variable	Value
CATALINA_HOME	C:\programs\jakarta-tomcat-4.1.18
JAVA_HOME	C:\programs\j2sdk1.4.1
PATH	C:\programs\j2sdk1.4.1\bin\java

Jeigu j□ n□ra, tada reikia spausti klavi□□ “New” ir □ra□yti reikiamus nustatymus:



43 pav. Serverio nustatymai;

6.6.2 Serverio “Tomcat” paleidimas ir išjungimas

Serverio paleidimo fail vartotojas ras tokiu adresu:

C:/programs/jakarta-tomcat-4.1.18/bin/startup.bat

Serverio ijungimo fail vartotojas ras tokiu adresu:

C:/programs/jakarta-tomcat-4.1.18/bin/shutdown.bat

Taip pat server galima ijungti paspaudus ant kryiuko, esanio deiniajame virutiniame kampe.

6.6.3 Operacinės sistemos reikalavimai

Minimals personalinio kompiuterio parametrai turi bti:

- 1000MHz CPU;
- 256 MB RAM;
- 32MB Video Card;

Operacin sistema Windows turi bti ne maesn nei Windows2000.

7 Išvados

Lygindama tiriam□ program□ su kitomis padariau tokias i□vadas:

1. Profiliuot□ mokykl□ tvarkara□□iui sudaryti yra b□tinas specialios programin□s □rangos panaudojimas;
2. Esamos programin□s □rangos pagrindinis tr□kumas – visos programos komercin□s ir orientuotos tik □ korporacijos Microsoft aplink□;
3. Dauguma komercini□ program□ reikalauja specialaus pasiruo□imo;
4. Mano nagrin□tose programose optimizavimas atliekamas labai primityviai ta□iau ir jis ne visada veikia;

Pasinaudojus daugeliu ank□□iau kurtos programos □school schedule optimization program“ element□ buvo sukurta nauja programos versija, kuri:

a. Nepriklauso nuo aplinkos;
b. Vykdo efektyvi□ optimizacij□;
c. Nereikalauja specialios programin□s □rangos;
d. Nereikalauja specialaus pasiruo□imo;
e. Duoda galimyb□ pasirinkti tiek objektyvius, tiek subjektyvius reikalavimus;
f. Programos praktinis pritaikymas Marijampol□s kolegijoje parod□ tolimesnes kryptis;

Literatûros sąrašas

[1] Mockus J., Bayesian heuristic approach to Scheduling, INFORMATIKA, INST MATHEMATICS & INFORMATICS, AKADEMIJOS 4, VILNIUS 2600, LITHUANIA,

2002 13 (3): 311-332

[2] Cohn H, Fielding M., Simulated annealing: Searching for an optimal temperature schedule, SIAM JOURNAL ON OPTIMIZATION, SIAM PUBLICATIONS, 3600 UNIV CITY SCIENCE CENTER, PHILADELPHIA, PA 19104-2688 USA,

19 1999 9 (3): 779-802 OCT

[3] Mockus J., Bayesian heuristic approach to global optimization and examples, JOURNAL OF GLOBAL OPTIMIZATION, KLUWER ACADEMIC PUBL, VAN GODEWIJCKSTRAAT 30, 3311 GZ DORDRECHT, NETHERLANDS,

2002 22 (1-4): 191-203 JAN

[4] Kong SC, Kwok LF., A conceptual model of knowledge-based time-tabling system, KNOWLEDGE-BASED SYSTEMS, ELSEVIER SCIENCE BV, PO BOX 211, 1000 AE AMSTERDAM, NETHERLANDS,

1999 12 (3): 81-93 JUN

[5] SELIM SM., Computer algorithm for constructing the departmental timetable, COMPUTERS & EDUCATION, PERGAMON-ELSEVIER SCIENCE LTD, THE BOULEVARD, LANGFORD LANE, KIDLINGTON, OXFORD, ENGLAND OX5 1GB,

1992 18 (4): 293-299 MAY

[6] Abboud N, Sakawa M, Inuiguchi M., School scheduling using threshold accepting, CYBERNETICS AND SYSTEMS, TAYLOR & FRANCIS LTD, ONE GUNPOWDER SQUARE, LONDON EC4A 3DE, ENGLAND,

1998 29 (6): 593-611 SEP

[7] Kaneko K, Yoshikawa M, Nakakuki Y., Improving a heuristic repair method for large-scale school timetabling problems, PRINCIPLES AND PRACTICE OF CONSTRAINT PROGRAMMING-CP'99 LECTURE NOTES IN COMPUTER SCIENCE, ELSEVIER SCIENCE BV, SPRINGER-VERLAG BERLIN, HEIDELBERGER PLATZ 3, D-14197 BERLIN, GERMANY,

1999 1713: 275-288

[8] Carter MW, Laporte G., Recent developments in practical course timetabling, PRACTICE AND THEORY OF AUTOMATED TIMETABLING II LECTURE NOTES IN COMPUTER SCIENCE , SPRINGER-VERLAG BERLIN, HEIDELBERGER PLATZ 3, D-14197 BERLIN, GERMANY,

1998 1408: 3-19

[9] Schaerf A., Local search techniques for large high school timetabling problems, IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS PART A-SYSTEMS AND HUMANS, IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS INC, 345 E 47TH ST, NEW YORK, NY 10017-2394 USA,

1999 29 (4): 368-377 JUL

[10] Schaerf A., A survey of automated timetabling, ARTIFICIAL INTELLIGENCE REVIEW, KLUWER ACADEMIC PUBL, SPUIBOULEVARD 50, PO BOX 17, 3300 AA DORDRECHT, NETHERLANDS,

1999 13 (2): 87-127 APR

[11] Meisels A, EllSana J, Gudes E., Decomposing and solving timetabling constraint networks, COMPUTATIONAL INTELLIGENCE, BLACKWELL PUBLISHERS, 350 MAIN STREET, STE 6, CAMBRIDGE, MA 02148-5023,\

1997 13 (4): 486-505 NOV