

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Kristina Bespalova

**Mokyklos tvarkaraščių optimizavimas interneto
aplinkoje įvertinant pedagoginius reikalavimus**

Magistro darbas

Darbo vadovas

prof. habil.dr. J. Mockus

Kaunas, 2009

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Kristina Bespalova

**Mokyklos tvarkaraščių optimizavimas interneto
aplinkoje įvertinant pedagoginius reikalavimus**

Magistro darbas

Recenzentas

dr. Lina Pupeikienė

2006-05

Vadovas

prof. habil.dr. J. Mockus

2009-05

Atliko

IFM-3/2 gr. stud.

Kristina Bespalova

2009-05-15

Kaunas, 2009

Turinys

1. ĮVADAS.....	7
2. MOKYKLOS TVARKARAŠČIŲ UŽDAVINIŲ ANALIZĖ	9
2.1. Įvadas	9
2.2. Tvarkaraščių sudarymo uždavinys.....	9
2.3. Tvarkaraščių uždavinio sprendimas.....	12
2.3.1. Euristiniai optimizavimo metodai	12
2.4. Tvarkaraščių sudarymo programinė įranga.....	15
2.5. Situacijos Lietuvoje įvertinimas.....	19
2.6. Išvados	20
3. MOKYKLOS TVARKARAŠČIŲ SUDARYMO IR OPTIMIZAVIMO PROGRAMINĖS SISTEMOS PROTOTIPAS	21
3.1. Mokyklos tvarkaraščių sudarymo ir optimizavimo programinės sistemos reikalavimai	21
3.1.1. Veiklos kontekstas	21
3.1.2. Programinės sistemos panaudojimas.....	22
3.1.3. Funkciniai reikalavimai.....	23
3.1.4. Nefunkciniai reikalavimai	23
3.1.5. Tvarkaraščio reikalavimų suskirstymas	24
3.2. Pirminių duomenų failas	25
3.3. Tvarkaraščio sudarymo principai.....	29
3.4. Tvarkaraščio optimizavimo procesas	31
3.5. Mokyklos tvarkaraščių sudarymo ir optimizavimo programinės sistemos architektūra	34
3.5.1. Sistemos statinis vaizdas	34
3.5.2. Paketų detalizavimas.....	34
3.5.3. Išdėstymo vaizdas	36
4. MOKYKLOS TVARKARAŠČIŲ SUDARYMO IR OPTIMIZAVIMO PROGRAMINĖS SISTEMOS KOKYBĖ.....	37
4.1. Kokybės vertinimas.....	37
4.2. Trūkumai ir tolesnio tobulinimo galimybės.....	38
5. MOKYKLOS TVARKARAŠČIŲ SUDARYMO IR OPTIMIZAVIMO PROGRAMINĖS SISTEMOS EKSPERIMENTINIS TYRIMAS	40
6. IŠVADOS	45
7. TERMINŲ IR SANTRAUKŲ ŽODYNAS.....	50
PRIEDAI.....	51

Paveikslai

1 pav. Mokyklos tvarkaraščių požymių diagrama	11
2 pav. Veiklos konteksto diagrama	21
3 pav. Panaudos atvejų diagrama	22
4 pav. Darbo lapo „Dalykai“ pavyzdys	25
5 pav. Darbo lapo „mokytojai“ pavyzdys	26
6 pav. Darbo lapo „3b“ pavyzdys	28
7 pav. Duomenų struktūros forma	29
8 pav. Tvarkaraščio struktūra	30
9 pav. Mokinio lango uždarymas	33
10 pav. Sistemos išskaidymas į paketus	34
11 pav. Parametrų ir metodo paėmimo klasių diagrama	34
12 pav. Tvarkaraščio sudarymo klasių diagrama	35
13 pav. Sistemos išdėstymo vaizdas	36
14 pav. Programinės sistemos darbo laiko priklausomybė nuo iteracijų skaičiaus	40
15 pav. Tvarkaraščio pagerėjimas įvedus mokinių langų uždarymą	41
16 pav. Tvarkaraščio pagerėjimo priklausomybė nuo paralelių klasių skaičiaus	42
17 pav. Mokyklos turimų kabinetų skaičiaus įtaka sudaromam tvarkaraščiui	43
18 pav. Specialiai įrengtų kabinetų skaičiaus įtaka sudaromam tvarkaraščiui	43
19 pav. Mokinių tvarkaraščių palyginimas, kur dalykų prioritetai nevertinami ir kur įvertinami	44

Lentelės

1 lentelė. Skirtingų pirminių duomenų optimizavimo suvestinės	42
---	----

Bespalova K. Web-based optimization of school timetables with didactic constraints; Master's paper in science of informatics, supervisor Prof. J. Mockus; Department of Software Engineering, Faculty of Informatics, Kaunas University of Technology, 2009.

SUMMARY

Scheduling of profiled schools is a relevant problem both in Lithuania and other countries. There are no methods of polynomial complexity created for solving this problem, therefore different heuristic methods are applied.

The main aim of this paper is to present the software system of the scheduling of profiled school with limited resources and analyze the impact of the pedagogical requirements of the optimal schedule search.

The literature analyzed in the paper is related to solving the tasks of school scheduling. Contemporary solutions are reviewed. The presented prototype of school scheduling and optimization software does not depend on the operating system. Its advantages and disadvantages are pointed out. The paper presents the results of the experimental research. They show that after introduction of new requirements into the system, the calculations increase and scheduling becomes extremely complex. However, only this way the solutions are achieved that satisfy the requirements of individual educational institution. The system evaluates the heuristic criteria of individual educational institution and the schedule achieved is optimal.

Santrauka

Profiliuotų mokyklų tvarkaraščių sudarymo uždavinys yra labai aktuali problema tiek Lietuvoje, tiek kitose šalyse. Šiems uždaviniams spręsti nėra sukurtų polinominio sudėtingumo metodų, todėl taikomi įvairūs euristiniai metodai.

Pagrindinis šio darbo tikslas yra pristatyti profiliuotos mokyklos tvarkaraščių su ribotais ištekliais sudarymo programinę sistemą ir ištirti pedagoginių reikalavimų įtaką optimalaus tvarkaraščio paieškai.

Darbe analizuojama literatūra susijusi su mokyklos tvarkaraščių uždavinių sprendimu, apžvelgiami šiuo metu egzistuojantys sprendimai. Pristatomas visiškai nuo operacinės sistemos nepriklausomas mokyklos tvarkaraščio sudarymo ir optimizavimo programinės sistemos prototipas. Pateikiami jo trūkumai ir tobulintinos vietos. Darbe pateikiami atlikto eksperimentinio tyrimo rezultatai, kuriuose buvo nustatyta, kad į sistemą įvedus naujus reikalavimus padidinami programos skaičiavimai, darosi sudėtingesnis tvarkaraščio sudarymas, tačiau tik taip gaunami sprendimai, kurie tenkina individualius mokymo įstaigos reikalavimus. Sistemoje įvertinami individualūs mokymo įstaigos euristiniai kriterijai, kurių dėka gaunamas tvarkaraštis artimas realiam.

1. ĮVADAS

Pamokų tvarkaraštis – tai pagrindinis mokyklos dokumentas, reglamentuojantis mokytojo ir moksleivių darbą.

Tvarkaraščių sudarymas yra labai aktuali problema įvairiose mokymo įstaigose. Tai sudėtingas planavimo uždavinys kurį bandoma išspręsti ne vieną dešimtmetį. Mokyklos tvarkaraščių charakteristikos labai priklauso nuo šalies, švietimo sistemos ir mokymo įstaigos kuriai tvarkaraštis reikalingas, todėl kiekvienas sukurtas problemos sprendimo modelis turi ribotą panaudojimą.

Lietuvoje nuo 2000 m. bendrojo lavinimo mokyklose įvestas profiliuotas mokymas. Bendrasis ugdymas orientuotas į vaiką, jo polinkius ir gebėjimus. Mokyklose sudaromos realios galimybės specializuotis. Kadangi profiliuotose klasėse mokosi daug mokinių, kurie renkasi skirtingus dalykus, iškyla lankstaus pamokų tvarkaraščio su mobiliomis moksleivių grupėmis sudarymo problema.

Vis dar nėra tokių mokyklos pamokų tvarkaraščių, kurie atitiktų visus jiems keliamus apribojimus ir asmeninius pageidavimus. Tačiau sudaromi tvarkaraščiai turi tenkinti švietimo ir mokslo ministerijos patvirtintus valstybinių ugdymo planų reikalavimus bei ugdymo organizavimo mokykloje ypatumus ir galimybes. Mokymo įstaigoms reikalingas optimalus, kokybiškas tvarkaraštis, nes mokinių ir mokytojų darbo patogumas priklauso nuo tvarkaraščio pagal kurį jie dirba. Pamokų tvarkaraštis pirmiausia turi būti palankus moksleiviui, nes pamokos yra svarbiausia jo veikla ir turi neabejotinos įtakos vaiko sveikatai ir protiniam darbingumui.

Mokyklose tvarkaraščius sudarinėjantys žmonės nori naudotis patogia ir kuo mažiau darbo reikalaujančia programine sistema. Tvarkaraščiai turi būti sudaromi automatiškai.

Optimalių tvarkaraščių sudarymo uždaviniai yra NP sudėtingumo. Tokius uždavinius galima spręsti pilnu binariniu perrinkimu ar šakų ribų metodais. Tačiau praktikoje sutinkamiems uždaviniams spręsti tokius metodus realizuoti labai sunku, o dažnai ir neįmanoma. Pastaruoju metu tvarkaraščių optimizavimo uždaviniams spręsti taikomi įvairūs euristiniai metodai.

Šiame darbe sprendžiamas mokyklų tvarkaraščių planavimo ir optimizavimo uždavinys. Kuriamą interneto aplinkoje dirbanti mokyklos tvarkaraščių sudarymo ir optimizavimo programinė sistema skirta Lietuvos bendrojo lavinimo profiliuotų mokyklų tvarkaraščių sudarymui. Siekiant patenkinti kuo daugiau tvarkaraščiams keliamų reikalavimų, ši sistema nuolatos tobulinama. Įvedami nauji subjektyvūs parametrai, kurių pagalba vartotojas gali įtakoti tvarkaraščio sudarymą, kad šis labiau atitiktų mokyklos poreikius.

Sukurta programinė sistema tinka ne tik mokyklų tvarkaraščiams sudaryti ir optimizuoti, bet yra naudojama aukštosiose studijose kaip geras diskretaus optimizavimo pavyzdys. Darbe pritaikytas modeliuojamojo atkaitinimo parametrų optimizuojamas naudojant Bayes metodą.

Darbas sudarytas iš šių dalių: analizės, projektinės, tyrimo ir eksperimento.

Analizės dalyje analizuojami tvarkaraščių sudarymo uždaviniai, jų sprendimo būdai, pateikiami metodai naudojami tvarkaraščiams rasti ir optimizuoti. Aptariamos euristinės paradigmos, kurios naudojamos tvarkaraščių sudarymo uždaviniams spręsti. Aptariama ir įvertinama tvarkaraščiams sudaryti naudojama programinė įranga bei apžvelgiama situacija Lietuvos bendrojo lavinimo mokyklose.

Projektinėje dalyje pateikiama sukurtos programinės įrangos techninės-projektinės dokumentacijos esminiai aspektai.

Tyrimo dalyje pateikiama sukurtos programinės įrangos kokybės analizė.

Eksperimentinėje dalyje pateikiami atlikto eksperimentinio tyrimo rezultatai. Įvertinama vartotojo nustatomų subjektyvių parametrų ir mokymo įstaigos užduodamų ribojimų įtaka optimizavimo procesui.

Darbo tikslas - pristatyti mokyklos tvarkaraščių su ribotais ištekliais sudarymo programinę sistemą ir ištirti pedagoginių reikalavimų įtaką optimalaus tvarkaraščio paieškai.

Šiame darbe pristatomas sprendimas sukurtas panaudojant paskirstytos sistemos principus, kliento serverio architektūrą ir realizuotas naudojant Java Servlet technologiją.

Darbo rezultatai buvo pristatyti ir aptarti šiose konferencijose:

- „Mokyklos profiliuotų klasių tvarkaraščių sudarymo ir optimizavimo sistema“. 12-oji Lietuvos jaunujų mokslininkų konferencija „Mokslas – Lietuvos ateitis“ sekcijoje „Informacinės technologijos“. Vilnius, 2009m. balandžio 9d. (1 Priedas).
- „Subjektyvių parametrų įvertinimas vidurinės mokyklos tvarkaraštyje“ 14-oji tarpuniversitetinė magistrantų ir doktorantų mokslinė konferencija „INFORMACINĖS TECHNOLOGIJOS“. Kaunas, 2009m. gegužės 8d. (2 Priedas).

Padėkos:

Dėkoju prof. habil. dr. Jonui Mockui už uždavinio formulavimą, pagalbą jį sprendžiant, už vertingą kritiką pastabas ir diskusijas.

Dėkoju Kauno technologijos universiteto programų inžinerijos katedros darbuotojams už suteiktas žinias ir visiems, kurie prisidėjo prie šio darbo vertingomis pastabomis ir pasiūlymais.

Dėkoju savo šeimai už dviejų metų kantrybę, supratimą, palaikymą ir skatinimą.

2. MOKYKLOS TVARKARAŠČIŲ UŽDAVINIŲ ANALIZĖ

2.1. Įvadas

Mokyklos tvarkaraščio sudarymo problemos nagrinėjamos ne vieną dešimtmetį. Tyrimai šioje srityje gana aktyvūs, nes vyksta nuolatinės reformos švietimo sistemoje visame pasaulyje, o tai kuria naujas problemas, kurioms reikia ieškoti sprendimų.

Apžvelgsime tvarkaraščių sudarymo uždavinį, šiam uždaviniui spręsti taikomus optimizavimo metodus ir egzistuojančius sprendimus.

2.2. Tvarkaraščių sudarymo uždavinys

Tvarkaraščių sudarymo uždavinys iškyla įvairiose gyvenimo veiklos srityse. Su šiuo sudėtingu planavimo uždaviniu susiduria ir mokymo įstaigos, nes reikia planuoti mokinių ir mokytojų užsiėmimų tvarkaraščius. Ypač sudėtingi profiliuotų ir aukštųjų mokyklų tvarkaraščių sudarymo uždaviniai.

Profiluotos mokyklos tvarkaraščio sudarymas tuo sudėtingesnis, kuo daugiau pasirinkimo laisvės duodama mokiniams. Tokiuose tvarkaraščiuose turi būti planuojamos pamokos su mobiliomis moksleivių grupėmis. Įtakos turi ir tai, kiek mokykloje dirba to paties dalyko mokytojų bei kiek mokykla turi įrengtų specializuotų kabinetų (pvz. informatikos, chemijos ir t. t.).

Mokyklos tvarkaraščių kokybės problemos atsiranda dėl:

- didelės mokinių pasirenkamų dalykų įvairovės;
- mokytojų stokos ar dirbančių nepagrindinėje darbovietėje paslaugų;
- sudėtingų dalykų pamokų nukėlimo į pamokų tvarkaraščių pabaigą;
- ugdymo proceso organizavimo nesrautiniu būdu (Žvirdauskas et al. 2006).

Mokyklos tvarkaraščius sudarinėjantis žmogus sudarinėdamas tvarkaraštį nori naudotis patogiai ir kuo mažiau darbo reikalaujančia programine sistema, kurios vartotojo sąsaja būtų jų gimtąja kalba. Tvarkaraščiai turi būti sudaromi automatiškai (Burke et al. 2002). Dažnai sprendžiama dviejų pakopų problema, t.y. pirmiausiai tvarkaraštis sudaromas, o po to ieškant geresnio sprendimo - optimizuojamas (Birbas et al. 2009).

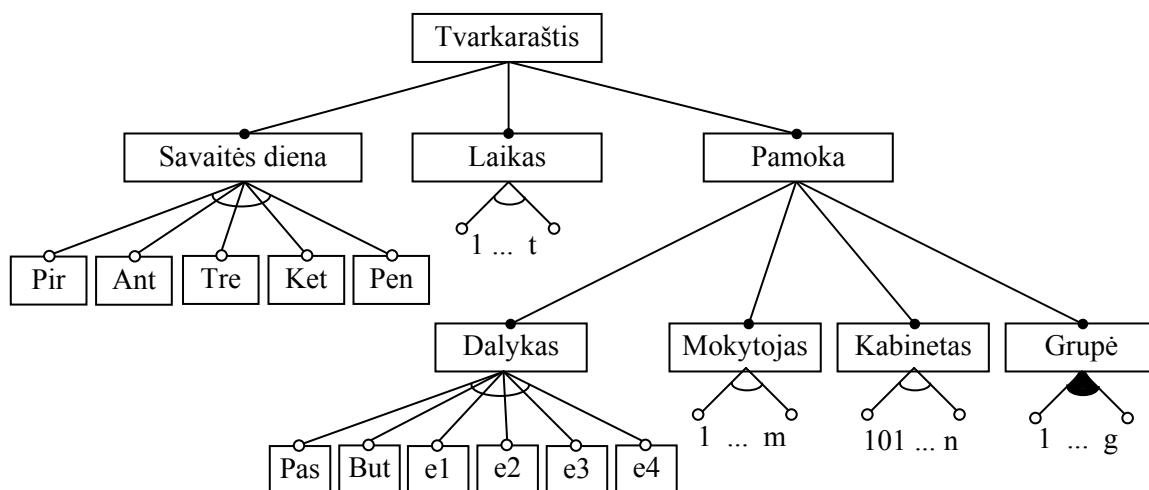
Mokymo įstaigoms reikalingas optimalus, kokybiškas tvarkaraštis, tenkinantis ne tik bendruosius ugdymo planų reikalavimus (Ugdymo planai), bet ir įstaigos individualius ugdymo organizavimo mokykloje ypatumus ir galimybes. Mokinių ir mokytojų darbo patogumas priklauso nuo tvarkaraščio pagal kurį jie dirba. Ypatingai pamokų tvarkaraštis

svarbus moksleiviui, nes pamokos yra svarbiausia jo veikla ir turi neabejotinos įtakos vaiko sveikatai ir protiniam darbingumui.

Mokyklos tvarkaraščiams keliami reikalavimai:

- Pamokų tvarkaraštyje neturi būti pamokų daugiau už nustatytą maksimumą, per dieną mokiniams negali vykti daugiau kaip septynios pamokos.
- Kiekvienai disciplinai skiriama tiek valandų, kiek nurodyta ugdymo plane.
- Kaitalioti sunkius ir lengvus dalykus.
- Kaitalioti pamokas atsižvelgiant į profilį: įtempto protinio darbo reikalaujančias pamokas kaitalioti su pamokomis, turinčiomis fizinio darbo elementų.
- Neplanuoti dviejų iš eilės vienos disciplinos ar panašaus profilio pamokų, nes tai greičiau vargina moksleivius.
- Sudarant tvarkaraštį, atsižvelgti į mokinių darbingumą. Darbingiausiu moksleivio dienos metu mokyti sunkius dalykus.
- Reikia įvertinti, kad kai kuriems dalykams reikalingos dvigubos pamokos.
- Tinkamai išdėstyti savaitės pamokas: jei savaitėje yra keletas tos pačios disciplinos pamokų, išdėstyti jas kas antrą ar trečią dieną.
- Reikia atsižvelgti, kad kūno kultūros, choreografijos, muzikos, piešimo, darbų pamokos nebūtų pirmos ar paskutinės, nes šios pamokos stabilizuoja protinį moksleivio darbą.
- Mokytojams ir mokiniams turi būti minimizuotas „tvarkaraščio langų“ skaičius.
- Tvarkaraštyje negali būti tokios situacijos kai mokinys ar mokytojas tuo pačiu metu dalyvauja dvejose pamokose.
- Tvarkaraštyje negali būti tokios situacijos kai vienu metu vykta ne viena pamoka tame pačiame kabinete.
- Pagal dalyko specifiką ar pakraipą paskaitoms tvarkaraštyje priskirti specialiai tam įrengtus kabinetus;
- Mokytojas gali mokyti tik savo mokomus dalykus, kuriuos mokyti turi teisę;
- Mokytojams dirbantiems keliose darbovietėse turi būti numatytos jų pakeičiamos „laisvos“ dienos, kurių metu negali vykti jų vedamos pamokos.
- Tvarkaraščio sudarymo programinė sistema turėtų reikalauti kuo mažiau vartotojo rankinio darbo.
- Tvarkaraščio sudarymo programinės sistemos sąsaja neperkrauta, vartotojo intuityviai suprantama, be apmokymo.

Mokyklos tvarkaraščių sudarymo programų šeimynai galima nubraižyti požymių diagramą (1 pav.), kurioje produktai apibūdinami požymiais. Požymių modeliavimas susitelkia ties labai matomų produkto charakteristikų apibūdinimu kalbant apie produkto bendrumą bei variantiškumą, nei sistemos detalių apibūdinimą.



1 pav. Mokyklos tvarkaraščių požymių diagrama

Mokymo įstaigose dažniausiai reikia suplanuoti savaitinius užsiėmimų tvarkaraščius. Mokyklos tvarkaraščio sudarymo uždavinys apima mokinių ir mokytojų susitikimų planavimą tam tikrose patalpose, fiksuotame laikotarpių skaičiuje taip, kad nė vienas iš dalyvių nebūtų keliose vietose tuo pačiu metu.

Pamokos tvarkaraštyje išdėliojamos penkiose darbo savaitinės dienose, tam tikrais laikais. Vienos pamokos privalomi komponentai yra mokomas dalykas, mokytojas, kabinetas ir mokinių grupė. Profiliuotose mokyklose mokomi dalykai skirstomi į dalykų grupes:

- pasirenkamų dalykų grupė iš kurios mokinys gali pasirinkti keletą dalykų;
- būtinieji dalykai, kurios mokinys privalo išklausti;
- 4 alternatyvių dalykų grupės iš kurių mokinys gali pasirinkti tik po vieną dalyką.

Mokytojas, mokinys ir kabinetas tuo pačiu metu gali dalyvauti tik vienoje pamokoje. Tas pats dalykas gali būti mokomas tuo pačiu metu kito mokytojo ir kitiems mokiniams.

Mokyklų tvarkaraščių sudarymo uždavinius bandoma išspręsti jau ne vieną dešimtmetį. Šie uždaviniai labai sudėtingi, o nuolatinės reformos švietimo sistemose keičia jų reikalavimus ir skatina ieškoti lankstesnių sprendimų.

2.3. Tvarkaraščių uždavinio sprendimas

Mokyklos tvarkaraščiams sudaryti naudojami įvairūs algoritmai, kurie apdoroja formalizuotus pradinius tvarkaraščio duomenis. Tvarkaraščio sudarymo uždavinys priskiriamas NP (angl. *Nondeterministic Polynomial*) sudėtingumo uždavinių klasei (Cooper et al. 1996). Norint rasti optimalų sprendinį reiktu atlikti pilna dvinarį perrinkimą, o tai sunku įgyvendinti, nes šis perrinkimas reikalauja didelių skaičiavimo ir laiko sąnaudų. Uždavinio sprendimas labai priklauso nuo kintamųjų skaičiaus. Formule (1) išreikštas uždavinio skaičiavimo laikas.

$$T > C2^{mn} \quad (1)$$

kur, m – tikslumas, o n – kintamųjų skaičius.

Tokiuose uždaviniuose dažniausiai randamas apytikslis sprendinys, siekiant, kad jis būtų kuo artimesnis optimaliam. Sprendiniai gaunami naudojant sveikaskaičio programavimo algoritmus. Po to gautą sprendinį galima perrinkti euristiniais algoritmais. Pagrindinė šių uždavinių problema, kad nėra leistino sprendinio optimalumo patikrinimo sąlygų.

Taikant euristinius metodus uždaviniai sprendžiami nepalyginamai greičiau, tačiau su tam tikra paklaida.

2.3.1. Euristiniai optimizavimo metodai

Optimizavimo uždaviniai aktualūs įvairiose žmonių veiklos srityse. Optimizavimo problemų gausu praktikoje. Matematiniais uždaviniais jos virsta tik sudarius tam tikrus nagrinėjamų procesų, reiškinių, ar situacijų matematinius modelius. Vienas svarbiausių optimizavimo etapų yra matematinio uždavinio formulavimas, nuo jo priklauso uždavinio išsprendžiamumas ir gautų rezultatų atitikimas praktiniam uždaviniui (Apynis 2005; Žilinskas 2000).

Sprendžiant optimizavimo uždavinius tenka naudoti įvairius euristinius paieškos algoritmus. Euristiniai algoritmai yra optimizavimo uždavinio sprendimo procedūros, kurių tikslas rasti aukštos kokybės, bet nebūtinai optimalius sprendinius per nustatytą skaičiavimo laiką (Misevičius et al. 2006).

Tvarkaraščiams optimizuoti taikomi apytiksliai euristiniai metodai. Jie remiasi prioritetinėmis taisyklėmis, evoliucijos idėjomis, vietine paieška ir kt. (Hoos et al. 2004). Naudojamos euristikos:

- Monte-Karlo (angl. *Monte Carlo*)

Monte-Carlo metodas, tai metodas kuriame uždavinio sprendimui naudojamas atsitiktinių skaičių generatorius. Optimizavimo metu, kiekviename žingsnyje iš galimų

sprendinių aibės parenkamas atsitiktinis sprendinys ir lyginamas su esamu. Palyginus priimamas geresnis surastas sprendinys. Naudojant šį metodą reikalingas didelis iteracijų skaičius pusiausvyrai rasti. Kai iteracija trunka trumpai, jis gana gerai tinka. Nagrinėjant sudėtingesnes situacijas ir siekiant didesnio tikslumo vienos iteracijos laikas labai prailgėja (Alexandrov et al. 2003; Mockus 2000; Bilgin et al. 2007). Monte-Karlo metodas labai gerai tinka funkcijoms, priklausančioms nuo daugelio kintamųjų ir turinčioms daugybę smailių. Šiam metodui visiškai nesvarbi integruojamos funkcijos forma, nei integruojamų duomenų skaičius.

– Modeliuojamojo atkaitinimo (angl. *Simulated Annealing*)

Modeliuojamojo atkaitinimo metodas labai paplitęs globaliojo optimizavimo metodas. Jis yra taikomas tolydiems ir diskretiems optimizavimo uždaviniams spręsti. Modeliuojamojo atkaitinimo metodas mėgstamas dėl jo paprastumo ir dėl gero teorinio pagrindimo. Šiame metode svarbus parametras yra temperatūra, mažėjanti su kiekvienu atkaitinimo proceso imitavimo žingsniu. Šis parametras reguliuojamas įvedant temperatūros atnaujinimo (aušinimo) funkciją. Algoritme yra specialus nuo temperatūros priklausantis tikimybinis patvirtinimo kriterijus, pagal kurį kiekviename žingsnyje priimamas geresnis už esamą surastas sprendinys bei su nedidele tikimybe gali būti pasirinktas blogesnis sprendinys. Ši savybė leidžia „ištrūkti“ iš lokaliųjų optimumo taškų, ieškant globaliojo optimumo (Mockus 2000; Abramson 1991). Algoritmo efektyvumui didelę įtaką turi optimizavimo parametrai (Felinskas et al. 2003).

Taikant atkaitinimo algoritmą tvarkaraščiams optimizuoti pradinis tvarkaraštis gerinamas tikrinant įvairius perstatinius. Po kiekvienos iteracijos geriausias surastas tvarkaraštis yra išsimenamas. Perėjimai į blogesnius tvarkaraščius yra atliekami su tam tikromis tikimybėmis, tikintis sekančiuose žingsniuose rasti dar geresnį sprendimą. Mažėjant iteracijų skaičiui, perėjimo į blogesnį variantą tikimybė darosi vis mažesnė.

Tvarkaraščius sudarant nedideliame užduočių kiekiui, galima efektyviai taikyti paprastus paieškos metodus – visų variantų perrinkimą ar godžiųjų algoritmų taikymą. Modeliuojamojo atkaitinimo metodas tokiais atvejais gali būti neveiksmingas laiko sąnaudų prasme, nes, norint gauti optimalų sprendinį iteracijų skaičių reikia didinti (Sakalauskas et al. 2006A).

– Bayes metodas (angl. *Bayes*)

Perspektyvus kelias euristicinių metodų efektyvumui padidinti yra parametru optimizavimas, panaudojant Bayes metodą. Šio metodo pagrindą sudaro matematinė naudingumo tikimybė ir racionalus variantas išrenkamas pagal vidutinės naudos maksimumą. Tam, kad galėtų taikyti šį metodą, projektuotojas turi turėti daugiau informacijos apie techninį sprendinį supančią aplinką ir sugebėti nustatyti tikimybinį atskirų aplinkos būklių

pasiskirstymo tankį. Bayes metode, iteracijų duomenys parenkami pagal prieš tai buvusių rezultatus (Laumanns et al. 2004; Mockus 2000). Bayes metodas dažnai naudojamas Modeliuojamojo atkaitinimo metodo praktiniam efektyvumui padidinti, optimizuojant jo parametrus.

– Paieška su draudimais (angl. *Tabu Search*)

Šis metodas remiasi tam tikru sprendinių uždraudimu, kurio tikslas „ištrūkti“ iš lokaliųjų optimumų. Metodo pagrindinis parametras – tabu sąrašas (angl. *tabu list*). Jis iš pradžių būna tuščias, o vėliau papildomas ir modifikuojamas. Į sąrašą optimizavimo metu įtraukiami jau nagrinėti sprendiniai ir jų aplinkos. Taip šiame metode draudžiama kurį laiką grįžti prie jau nagrinėtų sprendinių ir jų aplinkų, tol kol draudimas bus išstumtas iš tabu sąrašo, kuris yra fiksuoto ilgio ir nuolat kintantis (Machado et al. 2001; Costa 1994). Paieška su draudimais yra iteracinis geresnių sprendinių paieškos procesas, įgalinantis daug kartų „pereiti“ nuo vieno lokaliojo optimumo prie kito įsimenant geriausiąjį iš surastų sprendinių.

– Genetiniai algoritmai (angl. *genetic algorithms*)

Genetinių algoritmų veikimas yra pagrįstas evoliucijos, vykstančios gyvojoje gamtoje, t.y., natūraliosios atrankos proceso imitavimu. Genetiniai algoritmai yra metodas analizuoti duomenis, jų dėka galima rasti apytikslį užduoties sprendimą. Pagrindinės naudojamos sąvokos yra „individas“ ir „populiacija“. „Individas“ yra tam tikras elementarus, vienetas ar objektas kuris daugiau neskaidomas. Didesnė ar mažesnė „individų“ grupė sudaro „populiaciją“. Genetiniuose algoritmuose individai vaizduojami sprendiniais, kurie yra koduojami simbolių eilutėmis – chromosomomis. Optimizuojamos funkcijos nežinomi kintamieji užkoduojami simbolių rinkiniais, kurie yra apjungiami į chromosomas. Optimizavimo metu ieškoma galimų sprendinių aibės, t.y. sprendžiant uždavinį operuojama ne su viena chromosoma, o su jų populiacija (Glibovec et al.2003; Erben 1996).

Genetiniuose algoritmuose atliekamos mutacijos ir kryžminimo funkcijos, kurių metu perimama vertinga informacija iš jau esančio sprendinio. Užtikrinama galimybė gauti bet kokią sprendinį iš visų galimų sprendinių aibės.

– Išbarstytoji, išsklaidytoji paieška (angl. *Scatter search*)

Išbarstytoji paieška operuoja sprendinių rinkiniu, vadinamu atraminium. Nauji sprendiniai gaunami kombinuojant atraminio rinkinio sprendinius. Sprendiniai į atraminį rinkinį atrenkami procedūromis, panašiomis į taikomas genetiniuose algoritmuose (Glover et al. 2003). Išbarstytoji paieška pradedama pradinio sprendinių rinkinio generavimu. Po to generuojami nauji sprendiniai, jie imami iš pradinio sprendinių rinkinio įvairiais poaibiais. Gautieji sprendiniai modifikuojami ir iš jų atrenkami „geriausi“.

Optimizavimo srityje daug nuveikęs yra profesorius Jonas Mockus ir jo studentai. Jo tinklapyje (Mockus 2009) nagrinėjami pagrindiniai optimizavimo uždaviniai ir jų sprendimo metodai, čia galima rasti įvairių euristinių metodų pritaikymo pavyzdžių. Studijų ir tyrimų objektas yra ekonominių, socialinių ir techninių sistemų modeliavimas ir optimizavimas. Uždaviniai parinkti taip, kad jie galėtų iliustruoti optimizavimo klausimus charakteringus plačioms praktinių uždavinių šeimoms.

Dauguma praktinių tvarkaraščių uždavinių yra gana sudėtingi ir klasikiniiais metodais galima gauti tik leistiną sprendinį, o jis nebus optimalus. Dažnai uždavinių apimtys būna didelės, su labai daug ribojimų ir perrinkti visus galimus variantus pasidaro neįmanoma, todėl ieškoma metodų, besiremiančių įvairiomis euristinėmis paradigmomis (Hoos et al. 2004).

Galime konstruoti tvarkaraščių optimizavimo algoritmus, pritaikydami įvairias euristikas. Paprasčiausia pritaikyti Monte-Karlo metodą su atsitiktine paieška, šis metodas greitas ir nereikalauja didelių skaičiavimų. Įvedus temperatūros bei aušinimo funkcijas, tvarkaraščiams optimizuoti galime pritaikyti modeliuojamojo atkaitinimo algoritmą. Galima sukonstruoti genetinį tvarkaraščių optimizavimo algoritmą, kur kursime ir modifikuosime chromosomų populiacijas, o atskira chromosoma laikysime kiekvieną sąrašą. Siekiant didesnio efektyvumo realizuojami deriniai įvairių metodų.

Daugelyje tvarkaraščių uždavinių nėra žinomos sprendinio optimalumo sąlygos, todėl algoritmai dažniausiai stabdomi atlikus iš anksto nustatytą iteracijų skaičių. Tačiau šis kriterijus nepatogus, kai optimalus sprendinys randamas atlikus mažesnę nei nustatytas iteracijų skaičių. Priimant sprendimą dėl optimumo radimo, galima pasinaudoti informacija apie optimizavimo metu gautas geriausias tikslo funkcijos reikšmes (Sakalauskas et al.2006B).

Tvarkaraščiams sudaryti yra taikomi paprasčiausi optimizavimo algoritmai. Sudėtingi šiuolaikiniai euristiniai algoritmai, standartinėse programinėse sistemose beveik nėra diegiami.

2.4. Tvarkaraščių sudarymo programinė įranga

Mokyklos užsiėmimų tvarkaraščių sudarymas yra gana sudėtinga problema. Dar daug kur tvarkaraščiai sudaromi rankiniu būdu, tačiau vis dažniau mokymo įstaigos pasinaudoja šiam uždaviniui spręsti sukurtų programinių sistemų teikiamomis paslaugomis. Mokyklos tvarkaraščiams sudaryti naudoja specialią programinę įrangą. Naudojamos programinės sistemos ne visada tenkina vartotojų lūkesčius, bet padeda atlikti šį sudėtingą planavimo uždavinį.

Tvarkaraščiams sudaryti naudojama programinė įranga:

„Mimosa“

„Mimosa“ – universali tvarkaraščio sudarymo sistema skirta įvairaus tipo ir dydžio mokymo įstaigoms. Šios programos pagrindinė idėja yra labai paprasta: „praktiškai neįmanoma automatiškai sukurti gero tvarkaraščio“ (Mimosa). „Mimosa“ automatiškai sudėlioja tvarkaraštį, pagal rankiniu būdu suvestus nurodymus. Tvarkaraštis optimizuojamas Monte-Karlo metodu. Programa baigia darbą kai neranda geresnio tvarkaraščio varianto. Sukurtas tvarkaraštis gali turėti daug langų. Programoje tvarkaraščio keitimui yra patogi grafinė aplinka. Dirbant šia programa svarbu žinoti tvarkaraščio sudarymo taisykles ir subtilumus. Programa veikia Win98 / WinME / WinNT 3.x / WinNT 4.x / Windows2000 / WinXP / Windows Vista Starter / Windows Vista Home Basic / Win platformose. Programa anglų klaba. Programos licencija vienam vartotojui kainuoja 480.00 USD. Galima išbandymui parsisiųsti nemokamą demonstracinę versiją skirtą mažoms mokymo įstaigoms.

„Rector“

Programa „Rector“ specialiai sukurta mokyklos pamokų tvarkaraščių sudarymui 1997 metais. 2002 metai buvo parengta tarptautinė programos Rector 3 versija ir programa buvo išversta į lietuvių kalbą. Darbas su programa paprastas, visi darbo etapai programoje yra aiškūs ir suprantami. Mokyklos tvarkaraštis gali būti sudarytas automatinio arba „rankiniu“ būdu, yra galimybė tuos būdus derinti, fiksuojant pagrindinius krūvius, išdėstant juos rankiniu būdu, o likusius krūvius išdėstant automatiškai. Pamokos tvarkaraščio tinklelyje gali būti perkeliama tiesiog pelytės pagalba. Programoje leidžia sudaryti ir mokytojų tarifikacijos lentelę, galima vesti mokytojų kvalifikacijos apskaitą. „Rector“ programa sudarytas tvarkaraštis neoptimizuojamas. Licencijos kaina vienam mokyklos kompiuteriui yra 500 litų, perkant licenciją antram mokyklos kompiuteriui, jos kaina 150 Lt. (Rector).

„aSc Tvarkaraščiai“

„aSc Tvarkaraščiai“ yra Slovakijos IT kompanijos Applied Software Consultants s.r.o. produktas. Pasak šios programos kūrėjų, programa „aSc Tvarkaraščiai“ sugeba sukurti visos mokyklos tvarkaraštį, į jį sudėdama absoliučiai visas reikiamas pamokas, jei tik tai yra menkiausia galimybė, patikrindama tūkstančius įvairiausių galimų variantų (aSc). Šia programa galima sukurti mokyklos, gimnazijos, licejaus, aukštosios mokyklos tvarkaraščius, kurie atitiks užduotus pedagoginius ir organizacinius reikalavimus. Sudarytą tvarkaraštį galima redaguoti, keičiant sąlygas ar rankiniu būdu išdėstant pamokas. Tvarkaraštį galima lyginti su anksčiau sukurtais tvarkaraščiais. Programa turi testavimo įrankius kurie leidžia surasti tvarkaraščio nesudarymo priežastis. „aSc Tvarkaraščiai“ neturi optimizavimo algoritmų, įvertinami tik susidarę mokytojų langai. Programa „aSc Tvarkaraščiai“ veikia bet

kokiame kompiuteryje, kuriame yra įdiegta Windows 98 ar aukštesnė Windows operacinė sistema (Windows 2000/ 2003 / ME / NT / XP / Vista). Tvarkaraščių kūrimo programą realizuota ir lietuvių kalba. Produkto kaina 687 Lt (be PVM).

„School schedule optimization program“

„School schedule optimization program“ programinė sistema skirta profiliuotų mokyklų tvarkaraščio sudarymui. Ši sistema sukurta anglų kalba, tačiau turi labai išsamią pagalbą vartotojui lietuvių kalba. „School schedule optimization program“ programinė sistema sukurta naudojant JAVA kalbą. Programinės sistemos sąsaja aiški ir neperkrauta (Pupeikienė 2006). Vartotojas dirbdamas su šia sistema dalyvauja tik sudarant pirminį duomenų failą, nustatant programoje esančius parametrus ir pasirenkant optimizavimo metodą. Pirminiame duomenų faile suvedama mokyklos tvarkaraščio sudarymui reikalinga informacija t.y. informacija apie mokomus dalykus, mokymo įstaigoje dirbančius pedagogus ir mokinių savaitinį užimtumą.. Sistema atsižvelgdama į nustatytus apribojimus automatiškai sudaro pirminį tvarkaraštį, atlieka optimizavimą, kiekvienam tvarkaraščio variantui apskaičiuoja baudos taškus ir išrinkusi geriausią pateikia rezultatus vartotojui. Rezultatai išvedami į ekraną ir į „zip“ archyvą, kurį vartotojas gali parsisiųsti į savo kompiuterį. Programa neleidžia atlikti rankinio tvarkaraščio koregavimo. Iškilus tokių darbų poreikiui, tvarkaraštį reikia parsisiųsti ir pakeitimus atlikti savo kompiuteryje. „School schedule optimization program“ programine sistema galima naudotis laisvai, tik veiksmai susiję su kopijavimu turi būti suderinti su autoriais.

„Lantiv Timetable“

„Lantiv Timetable“ programa skirta mokymo įstaigos tvarkaraščio sudarymui (Lantiv). Programa gali dirbti automatinio ir rankinio režimu. Neturi tvarkaraščio optimizavimo galimybių. „Lantiv Timetable“ įvertina nurodytus apribojimus tokius kaip: pamokų skaičių per dieną, pamokų trukmę, jungtines pamokas ir kitus. Rankiniame režime galima daryti tvarkaraščio pakeitimus, tokius kaip atskirų elementų kopijavimas ar perkėlimas. Kiekviename programos lange yra instrukcijos kaip naudotis programa. Sekant šiomis instrukcijomis, galima sudaryti mokyklos pamokų tvarkaraštį. Spausdinti galima bendrą, savaitės, dienos ar asmeninį tvarkaraštį. Programa veikia Windows 2000, XP, Vista operacinėse sistemose. „Lantiv Timetable“ Licencija metamas kainuoja 137 EUR.

„Wise Timetable“

„Wise timetable“ programa skirta rankiniam ir automatiniam tvarkaraščio sudarymui. Yra dvi versijos: pirmoji skirta universitetų, o antroji pradinių ir vidurinių mokyklų tvarkaraščių sudarymui (Wise). Paskutinė versija išleista 2007 metais (3.2 versija). Ši

programa nedaro jokių apribojimų mokymo įstaigos dydžiui, gali išvesti metų ar savaitės tvarkaraščius. Programa suderinama su visomis Windows platformomis. „Wise timetable“ programos skirtos mokykloms licencija vienam asmeniui kainuoja 1950 EUR.

„ABC Timetable“

„ABC Timetable“ programa skirta mokyklų ir kitų veiklų tvarkaraščių sudarymui. Šioje programinėje sistemoje paprasčiausiai pasirenkamas dienų skaičius, nurodomi dalykai, pasirenkamas norimas projektas, spalvos, šriftas ir saugomas sudarytas tvarkaraštis. Tvarkaraščiai sudaromas pilnai rankiniu būdu. Ši programinė sistema platinama nemokamai. Veikia Win95 / Win98 / WinME / WinNT 4.x / Windows2000 / WinXP platformose (Abc).

„ACTion“

„ACTion“ programa gyvuoja nuo 1993 metų. Tai mokyklos tvarkaraščio sudarymo programa, kuri mažiau nei per dvidešimt sekundžių sudaro mokyklos tvarkaraštį. Tai universiteto studento Jaco van Niekerk sukurtas matematinis modelis. Programa vis dar aktyviai tobulinam ir kiekvienais metais pateikiamos naujos versijos. Didžioji dalis pakeitimų atsiranda įgyvendinant sistemos vartotojų pageidavimus. Programa galima naudotis nemokamai (Action). Veikia Windows platformoje.

„FET“

„FET“ Tai atviro kodo, nemokama programinė sistema skirta automatiniam mokymo įstaigų tvarkaraščių sudarymui. Programa tvarkaraščius leidžia sudaryti pusiau automatiškai bei rankiniu būdu. Ji naudoja paprastą, bet efektyvų tvarkaraščio sudarymo algoritmą. Tvarkaraštis sudaromas per 5 – 20 minučių. „FET“ turi galimybę importuoti ir eksportuoti duomenis XML formatu. Veikia GNU/Linux, Windows, Mac platformose (Fet).

„iMagic“

„iMagic“ programa skirta mokyklų, kolegijų ir universitetų tvarkaraščiams kurti. Automatiškai kuria tvarkaraštį klasei, kabinetui ir mokytojui. Turi galimybę dirbančius mokytojus pakeisti pavadojančiais. Programa veikia Windows 95/98/ME/NT/2000/XP platformose. Galima parsisiųsti programos versiją, kuria galima naudotis 30 dienų nemokamai. „iMagic“ kainuoja \$99.00 (iMagic).

“GIFT School”

“GIFT School” – tvarkaraščių sudarymo sistema. Ji tinka įvairaus tipo ir dydžio mokykloms. Vartotojo sąsaja paprasta. Automatiškai kuria ir optimizuoja tvarkaraštį mokykloms. Sudaromi mokyklos, klasių, mokytojų ir mokytojų laisvų dienų tvarkaraščiai. Turi galimybę dirbančius mokytojus pakeisti pavadojančiais. Programinės sistemos

išbandymui galima parsisiųsti nemokamą demonstracinę versiją. „GIFT School” programa kainuoja \$259.00 (Gift).

Mokyklos tvarkaraščiams keliami reikalavimai labai priklauso nuo šalies švietimo sistemos, mokymo įstaigos, todėl sukurti sprendimai turi ribotą panaudojimą. Lietuva taip pat nėra išimtis. Čia tvarkaraščiams keliami savi ribojimai ir reikalavimai, todėl nevisos siūlomos programinės sistemos yra tinkamos tvarkaraščiams sudaryti.

2.5. Situacijos Lietuvoje įvertinimas

Lietuvoje bendrojo lavinimo mokyklos yra profiliuotos. Profiliuotose mokyklose vienas iš sudėtingesnių uždavinių yra lankstaus pamokų tvarkaraščio sudarymas su mobiliomis moksleivių grupėmis. Sudaromi pamokų tvarkaraščiai turi tenkinti Lietuvos Respublikos švietimo ir mokslo ministerijos patvirtintus valstybinių ugdymo planų reikalavimus bei ugdymo organizavimo mokykloje ypatumus ir galimybes.

Lietuvoje 2007-2008m.m. bendrojo lavinimo mokyklų buvo 14732. Jose mokėsi 524333 moksleiviai (pradinis mokymas – 135752 moksleiviai, pagrindinis mokymas – 275848 moksleiviai, vidurinis mokymas – 112733 moksleiviai) ir dirbo 42846 pedagogai (Statistika).

Šiuo metu Lietuvoje plačiausiai naudojamos dvi tvarkaraščių sudarymo sistemos, tai „Mimosa“ ir „aSc Tvarkaraščiai (Gaidukevičienė et al. 2005).

„Mimosa“ programa sudarytas tvarkaraštis nevisiškai tenkina mokymo įstaigas. Nėra paprasta suvesti pradinis duomenis į pačią programą. Sudėtinga ir painu suprasti, kada ir ką reikia rašyti, o padarius bent menkiausią klaidą, programa neveikia. (Alaburdienė et al.). Programoje nenaudojami jokie sudėtingi optimizavimo metodai. Paleidus tvarkaraščio optimizavimo procesą programos darbas pasidaro neaiškus. Neaišku kada programa baigs darbą ir apskritai ar ji kažką daro. Labiausiai šis įrankis tinka tvarkaraščio koregavimui rankiniu būdu. Dirbant šia programa labai svarbu žinoti tvarkaraščio sudarymo taisykles ir subtilumus. Daugiausiai laiko užimantis darbas – duomenų suvedimas. Ši programa reikalauja daugiau nei 50% žmoniškųjų resursų.

„aSc Tvarkaraščiai“ sukuria gerą tvarkaraštį, tačiau programa neturi tvarkaraščio optimizavimo galimybių. Programa labiau tinkama tradicinių klasių tvarkaraščiams sudaryti, bet ne individualius mokinių tvarkaraščius. „aSc Tvarkaraščiai“ visiškai nevertina mokinių langų. Programa turi galimybę importuoti pradinis tvarkaraščio duomenis į programą iš XML bylos. Šioje XML byloje yra tik pagrindiniai duomenys apie klases, disciplinas, mokytojus ir kabinetus. Programa viską daro etapais ir kiekviename etape yra paaiškinimai ką vartotojas turi daryti. Ši programa reikalauja apie 30% žmoniškųjų resursų.

2.6. Išvados

- 1 Sudaromi tvarkaraščiai turi būti kokybiški, jie turi tenkinantis ne tik bendruosius ugdymo planų reikalavimus, bet ir įstaigos individualius ugdymo organizavimo ypatumus ir galimybes.
- 2 Mokyklos tvarkaraščių sudarymo uždaviniai yra labai sudėtingi. Jie turi labai daug ribojimų. Perrinkti visus galimus sprendinių variantus yra neįmanoma, todėl tenka naudoti įvairius euristinius paieškos algoritmus. Tokiuose uždaviniuose dažniausiai randamas apytikslis sprendinys, siekiant, kad jis būtų kuo artimesnis optimaliam.
- 3 Tvarkaraščių charakteristikos labai priklauso nuo šalies, švietimo sistemos ir mokymo įstaigos kuriai tvarkaraštis reikalingas, todėl kiekvienas sukurtas problemos sprendimo modelis turi ribotą panaudojimą.
- 4 Tvarkaraščiams sudaryti naudojama programinė įranga dažniausiai reikalauja daug rankinio darbo. Dauguma programinės įrangos neturi tvarkaraščio optimizavimo algoritmų, o kurios turi, tai turi paprasčiausius.

3. MOKYKLOS TVARKARAŠČIŲ SUDARYMO IR OPTIMIZAVIMO PROGRAMINĖS SISTEMOS PROTOTIPAS

Mokyklos tvarkaraščių sudarymo ir optimizavimo programinė sistema skirta vidurinių mokyklų profiliuotų klasių tvarkaraščių sudarymui. Mokyklos tvarkaraščio optimizacinis modelis palengvina eksperimentinius šio polinomiškai nesprenžiamo uždavinio eksperimentinius tyrimus. Sukurta programinė sistema tinka ne tik mokyklų tvarkaraščiams sudaryti ir optimizuoti, bet jau yra naudojama aukštosiose studijose kaip geras diskretaus optimizavimo pavyzdys.

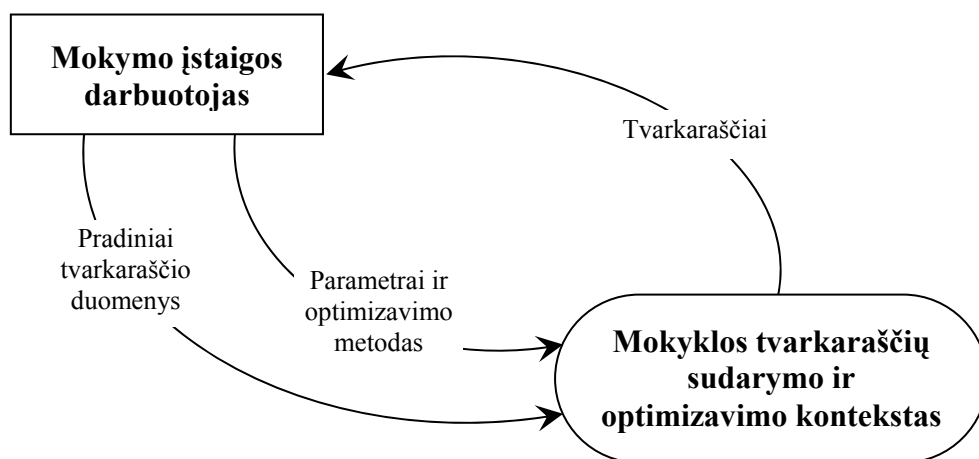
3.1. Mokyklos tvarkaraščių sudarymo ir optimizavimo programinės sistemos reikalavimai

Mokyklos tvarkaraščių sudarymo ir optimizavimo programinė sistema yra savarankiška, nėra jokios kitos sistemos dalis. Tai programa skirta darbui interneto aplinkoje.

Potencialūs kuriamos programinės sistemos vartotojai - asmenys, kurie vidurinėse mokyklose sudarinėja pamokų tvarkaraščius. Vartotojai turėtų mokėti dirbti Microsoft Office programa „Excel“ ir mokėti naudotis WWW tinklo naršykle.

3.1.1. Veiklos kontekstas

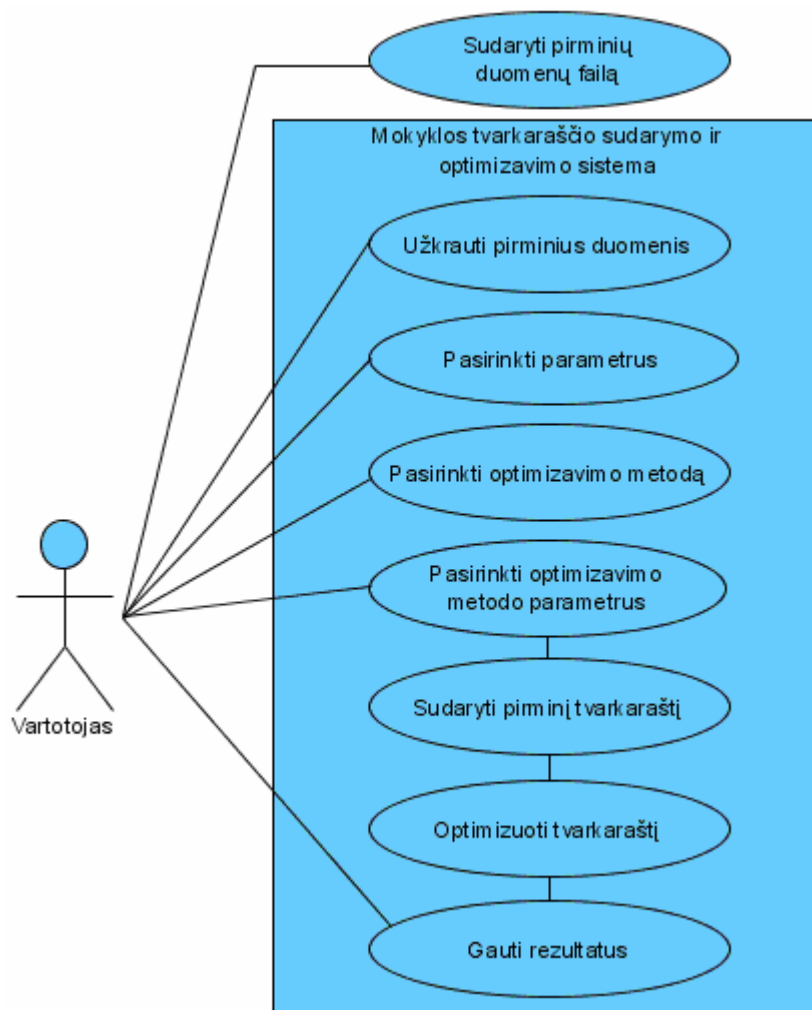
Programinės sistemos veiklos kontekstas (2 pav.).



2 pav. Veiklos konteksto diagrama

Mokymo įstaigos darbuotojas dirbdamas su programine sistema pateikia pirminio tvarkaraščio duomenis bei nurodo programoje esančius parametrus ir optimizavimo metodą. Sistema atsižvelgdama į nustatytus apribojimus automatiškai sudaro pirminį tvarkaraštį, atlieka optimizavimą ir pateikia rezultatus vartotojui.

3.1.2. Programinės sistemos panaudojimas



3 pav. Panaudos atvejų diagrama

Panaudos atvejų sąrašas:

- *Sudaryti pirminių duomenų failą* - apima procesą, kurio metu vartotojas suveda pirminius duomenis Microsoft Office programa „MS Excel“ ir išsaugo „XML“ formatu. Šis failas turi būti ruošiamas laikantis formos (3 Priedas).
- *Užkrauti pirminius duomenis* - apima procesą, kurio metu vartotojas sistemai nurodo pirminių duomenų failą ir informacija iš jo sukeliama į sistemą.
- *Pasirinkti parametrus* - apima procesą, kurio metu vartotojas parametrų įvedimo lange suveda darbui reikalingus parametrus ir šie patikrinus išsaugomi sistemoje.
- *Pasirinkti optimizavimo metodą* - apima procesą, kurio metu vartotojas pasirenka vieną iš sistemos siūlomų optimizavimo metodų.
- *Pasirinkti optimizavimo metodo parametrus* - apima procesą, kurio metu vartotojas parametrų įvedimo lange suveda darbui reikalingus parametrus, kurie patikrinami ir išsaugomi sistemoje.

- *Sudaryti pirminį tvarkaraštį* - apima procesą, kurio metu paimami pirminiai tvarkaraščio duomenys ir sudėliojami į tvarkaraščio matricą.
- *Optimizuoti tvarkaraštį* - apima procesą, kurio metu paimamas pirminis tvarkaraštis, vykdomas optimizavimo procesas pagal vartotojo pasirinktus parametrus ir metodą. Sukuriamas tvarkaraščių „zip“ archyvas.
- *Gauti rezultatus* - apima procesą, kurio metu paimamas optimizuotas tvarkaraštis ir formuojama mokyklos tvarkaraščio ataskaita ekrane. Vartotojui pageidaujant formuojamas pasirinkto mokytojo, mokinio ar kabineto tvarkaraštis. Vartotojas parsisiunčia į savo kompiuterį tvarkaraščių archyvą. Programos darbo rezultatų pavyzdžiai pateikiami priede (4 Priedas).

3.1.3. Funkciniai reikalavimai

- Sistema turi leisti vartotojui pasirinkti pirminį duomenų failą;
- Sistema turi įsikelti duomenis iš pirminio duomenų failo;
- Nepavykus įkelti pirminių duomenų, sistema turi apie tai informuoti ir problemos priežastis pranešti vartotojui;
- Sistema turi sudaryti pirminį tvarkaraštį;
- Nepavykus sudaryti pirminio tvarkaraščio, sistema turi apie tai informuoti ir problemos priežastis pranešti vartotojui;
- Sistema turi leisti koreguoti optimizavimo parametrus;
- Sistema turi leisti pasirinkti optimizavimo metodą;
- Sistema turi optimizuoti pirminį tvarkaraštį pagal nustatytus optimizavimo parametrus ir pasirinktą optimizavimo metodą;
- Sistema turi pranešti vartotojui kiek pavyko pagerinti pirminį tvarkaraštį;
- Sistema turi sudaryti mokytojų, mokinių, grupių ir kabinetų tvarkaraščius.
- Sistema turi išvesti į ekraną vartotojo pasirinktus tvarkaraščius;
- Sistema turi leisti parsisiųsti tvarkaraščių archyvą.

3.1.4. Nefunkciniai reikalavimai

- Sistemos vartotojai turi intuityviai naudotis programine sistema be apsimokymo;
- Korporatyvinis dizainas;
- Turi būti vartotoja sąsaja nacionaline (lietuvių) kalba;
- Tikimybė, kad sistema nebus galima pasinaudoti minimizuoti iki 2 %;

- Vartotojo sąsaja turi būti prieinama iš kiekvieno prie interneto prijungto kompiuterio, turinčio naršyklę;
- Sistemos duomenys turi vienareikšmiškai atitikti duomenis paimtus iš pirminio duomenų failo;
- Programinėje sistemoje negalima naudoti ką nors įžeidžiančių terminų ar iliustracijų;

3.1.5. Tvarkaraščio reikalavimų suskirstymas

Sudaromam tvarkaraščiui keliama eilė reikalavimų. Vieni iš jų yra privalomi, kuriuos būtina patenkinti, kiti priimami kaip pageidavimai, kuriuos sistema leidžia pažeisti. Mokyklos tvarkaraščių sudarymo ir optimizavimo programinėje sistemoje reikalavimai suskirstomi šia tvarka:

Privalomi reikalavimai

- Mokinys vienu metu gali dalyvauti tik vienoje pamokoje;
- Mokytojas vienu metu gali dalyvauti tik vienoje pamokoje;
- Kabinete vienu metu gali vykti tik viena pamoka;
- Mobilios moksleivių grupės formuojamos tik iš to paties lygio moksleivių, neviršijant pamokoje leistino mokinių skaičiaus.
- Mokytojui neviršyti maksimalaus valandų skaičiaus, kurį jis gali dirbti per savaitę;
- Tvarkaraštyje naudojami tik pirminiame duomenų faile nurodyti mokyklos resursai.
- Pagal dalyko specifiką ar pakraipą pamokoms tvarkaraštyje priskiriami specialiai tam įrengti kabinetai.

Pageidavimai

- Pamokas tvarkaraštyje sudėti taip, kad nebūtų viršijamas vartotojo nurodytas maksimalus dienos pamokų skaičius;
- Mokytojams minimizuoti „tvarkaraščio langų“ skaičių;
- Mokiniams minimizuoti „tvarkaraščio langų“ skaičių;
- Sudėtingiausius dalykus tvarkaraštyje dėti pirmoje dienos pusėje;
- Neplanuoti dviejų iš eilės vienos disciplinos ar panašaus profilio pamokų;
- Mokytojams dirbantiems keliose darbovietėse turi būti numatytos jų pageidaujamos „laisvos“ dienos, kurių metu turėtų nevykti jų vedamos pamokos.

3.2. Pirminių duomenų failas

Mokyklos tvarkaraščio optimizavimo programinės sistemos darbui reikalingas pirminių duomenų failas, kuriame suvedama informacija reikalinga tvarkaraščio sudarymui. Dokumentas ruošiamas Microsoft Office programa „MS Excel“ ir saugomas „XML“ formatu.

Pirminis duomenų failas kuriamas laikantis nurodytos formos. Ši forma buvo vis tobulinam, kad būtų paprasta ir patogi darbui. Užpildyto duomenų failo pavyzdys pateiktas prieduose (3 Priedas). Nesilaikant formos duomenys bus klaidingai suprasti ir nenuskaityti į programą darbui.

XML dokumento pirmasis darbo lapas turi būti pavadintas „Dalykai“ (4 pav.). Jame saugoma visa bendra informacija apie mokymo įstaigoje mokomus dalykus.

	A	B	C	D	E	F	G	H	I
1	Dalykai	grupes	prioritetai	max studentų skaičius		Maksimalus mokinių skaičius su kuriuo gali dirbti mokytojas.			
2	Ps	e1	1	30		Prioritai, kurie suteikiami dalykui pagal sudėtingumą. 5- sudėtingiausi; 1- lengviausi.			
3	T	e1	1	30					
4	E	e1	1	30					
5	Lk	but	5	30		Grupės, į kurias skirstomi mokykloje mokomi dalykai.			
6	A	e2	3	20					
7	V	e2	3	20					
8	R	e2	3	20		Dalykų, kurie yra mokomi mokykloje, pavadinimų trumpinimai.			
9	Pr	e2	3	20					
10	A	e3	3	20					
11	V	e3	3	20		"e" - dalykų grupė, iš kurios galima pasirinkti tik vieną dalyką.			
12	R	e3	3	20					
13	Pr	e3	3	20					
14	Ist	pas	4	30		"pas" - dalykų grupė, iš kurios galima pasirinkti keletą dalykų.			
15	Geo	pas	4	30					
16	Mat	but	5	30		"but" - dalykai, kuriuos būtina pasirinkti.			
17	Inf	pas	4	30					
18	Biol	pas	4	30					
19	Fiz	pas	5	30					
20	Chem	pas	5	30					
21	Mu	pas	1	30					
22	Sok	pas	1	30					
23	Tea	pas	1	30					
24	Da	pas	1	30					
25	k.k.	but	3	30					
26	lot	pas	2	20					
27	isp	pas	2	20					
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					
33									

4 pav. Darbo lapo „Dalykai“ pavyzdys

Stulpelyje „A“ *Dalykai* – tai visų mokykloje mokomų dalykų trumpiniai. Jie turi būti surašyti vienas po kitu. Taip pat šie trumpiniai, tokiu pat pavidalu, bus vartojami pildant mokytojų langą (Sheet2) ir mokinių pasirinkimų lenteles.

Stulpelyje „B“ dalykų *Grupės* – visi dalykai turi būti skirstomi į tokias grupes:

- „e“ – tai grupė, kurioje mokinys gali pasirinkti tik vieną dalyką. Pavyzdžiui: grupę „e1“ sudaro trys dalykai: Tikyba (T), Etika (E) ir Psichologija (Ps). Mokinys gali pasirinkti tik vieną iš šių dalykų. Jei yra sudaroma kita tokių dalykų grupė, ji pažymima raide „e“ su kitu numeriu. Pavyzdžiui, užsienio kalbų grupavimas.
- „but“ – tai būtinieji dalykai, kuriuos mokinys privalo pasirinkti.
- „pas“ – tai dalykai, kuriuos mokinys gali rinktis kartu. Pavyzdžiui, mokinys mokslo metais gali mokintis Chemiją (Chem), Biologiją (Biol), Geografiją (Geo) ir Istoriją (Ist).

Stulpelyje „C“ *Prioritetai* – tai visų mokykloje mokomų dalykų suskirstymas pagal jų sudėtingumą. Sudėtingiausi dalykai turi būti mokomi pirmoje dienos pusėje ir turi indeksą 5, o lengviausi indeksą 1. Sudarant tvarkaraštį, nuo dalyko prioriteto priklausys kurioje dienos pusėje bus mokoma pamoka. Vartotojas

Stulpelyje „D“ *Max mokinių skaičius* – tai maksimalus mokinių skaičius su kuriuo mokytojas gali dirbti konkretaus dalyko pamokoje. Sudarant tvarkaraštį formuojamos mobilios mokinių grupės, kuriose mokinių skaičius bus nedidesnis už vartotojo nurodytą maksimalų mokinių skaičių.

Dalykų prioritetus ir mokinių skaičių vartotojas gali nustatyti individualiai, pagal savo mokyklos tvarkaraščio reikalavimus.

Antrasis XML dokumento darbo lapas turi būti pavadintas „mokytojai“ (5 pav.). Čia saugoma informacija apie mokymo įstaigos dirbančius mokytojus.

	A	B	C	D	E	F	G	I
1								
2	grupe	Eil.	Dalykas	Pavarde_Vardas	Kabinetas	Val.sk	Trump	Pageidavimai / Laisvadieniai
3	1	1	Etika	Martinaitiene Laura	212; 213; 201;	22;	E	
4		2	Etika	Patamsiene Jolanta	212; 213; 201;	16;	E	1;2;
5		3	Tikyba	Maciuleviciene Natalija	213; 212; 201;	13;	T	1;2;
6		4	Tikyba	Vasiliauskaite Nijole	213; 212; 201;	27;	T	
47	2	5	Matematika	Bukauskiene Rimante	102;	26;	Mat	
48		6	Matematika	Jankauskiene zivile	101;	22;	Mat	
49		7	Matematika	Keniausiene Ada	115;	16;	Mat	2;4;5;
50		8	Matematika	Medisauskiene Regina	113;	22;	Mat	
51		9	Matematika; Informatika;	Osipaviciute Ruta	115; #214;	18;5;	Mat;Inf;	
52		10	Matematika	Senulyte Sigita	113;	22;	Mat	
53		11	Matematika	Sudvojiene Terese	113;	24;	Mat	
54		12	Matematika; Informatika;	Veikulyte Ramune	101; #214;	21;5;	Mat;Inf;	
55		13	Matematika; Informatika;	Velickiene Laima	101; #214;	4;5;	Mat;Inf;	1;2;3;
56		14	Matematika	ziuriene Daiva	102;	8;	Mat	2;3;5;
57		15	Informatika	Rafalovic Jaroslov	214;	21;	Inf	
58		16	Informatika	Rasimas Darius	214;	26;	Inf	

5 pav. Darbo lapo „mokytojai“ pavyzdys

Stulpelyje „A“ *Grupės* – stulpelis formuojamas vartotojo nuožiūra. Vartotojas gali mokytojus skirstyti taip kaip jam patinka: pagal dalykus, pagal pasirinkimus, pagal pavardes ir t.t. Šis stulpelis skirtas vizualiai paieškai pradinių duomenų faile ir programos darbui jokios įtakos neturi. Jis gali būti ir nepildomas. Svarbiausia išliktų stulpelyje „A“ pavadinimas „grupe“.

Stulpelyje „B“ *Eil.* – įrašomas mokytojo eilės numeris esamam sąrašė. Taip pat skirtas tik vizualiai mokytojo paieškai pradinių duomenų faile ir neturi įtakos programos darbui.

Stulpelyje „C“ *Dalykas* – nurodomas pilnas dalyko pavadinimas kurį moko mokytojas. Pavyzdžiui: „Informatika“, „Matematika“ ir t.t. Jei mokytojas moko du dalykus, šiame stulpelyje juos reikia abu paminėti atskiriant vieną nuo kito kabliataškiu. Pavyzdžiui: „Matematika; Informatika“, „Anglų; Vokiečių“, ir t.t. Tie patys dalykai turi būti rašomi vienodai. Negalima rašyti vienam mokytojui „Matematika, o kitam mokytojui „matematika“ .

Stulpelyje „D“ *Pavarde_Vardas* – šiame stulpelyje nurodoma mokytojo pavardė ir vardas. Pvz: Pavardenis Vardenis.

Stulpelyje „E“ *Kabinetas* – nurodomas kabineto, kuriame mokytojas dirba, numeris. Jei mokytojas neturi savo kabineto, šiame langelyje įrašomi kabinetų numeriai kuriuose jis gali dirbti. Jei mokytojas moko du dalykus, tai iš pradžių išvardijami kabinetų numeriai, kuriuose mokytojas moko pirmojo dalyko, po to dedamas „#“ ir išvardijami kabinetų numeriai, kuriuose moko antrojo dalyko. Jei mokytojas dirba ne viename kabinate, šiame stulpelyje juos reikia paminėti atskiriant vieną nuo kito kabliataškiu. Pavyzdžiui: „201; 202;“. Šiame stulpelyje, pagal mokytojo mokomo dalyko specifiką ar pakraipą, reikia priskirti specialiai tam įrengtus mokyklos kabinetus;

Stulpelyje „F“ *Val. sk.* – nurodomas maksimalus valandų skaičius, kurį mokytojas gali dirbti per savaitę. Nurodžius valandas, po skaičiaus, dedamas kabliataškis. Pavyzdžiui: „18;“. Jei mokytojas moko du dalykus, tada nurodomos dvejios, viena po kitos valandos (pirmosios pirmajam dalykui, antrosios – antrajam). Pavyzdžiui: „18;5;“.

Stulpelyje „G“ *Trump* – nurodomas mokomo dalyko trumpinys. Pavyzdžiui: „Mat“, „Inf“ ir t.t. Jei mokytojas moko du dalykus, jie išvardijami atskiriant vieną nuo kito kabliataškiu. Pavyzdžiui: „Mat;Inf;“

Stulpelyje „I“ *Pageidavimai/Laisvadieniai* – nurodomos savaitės dienos (skaičiais), kada mokytojas pageidauja turėti laisvas dienas. Pavyzdžiui: „3;4;5;“, „5;“, „1;5;“ ir t.t.

Trečiojo XML dokumento darbo lapo pavadinimas turi būti toks pat, kaip ir klasės pavadinimas (6 pav.). Pavyzdžiui, „5a“. Visi likusieji lapai turi būti skirti kitoms klasėms. Paskutinis lapas turi likti kaip šablonas. Jo pavadinimas turi būti „Šablonas“.

Šiame lape suvedama visa klasės mokinių savaitinė informacija. Į klasės pageidavimų langą galima nukopijuoti šabloną (iš lapo „Šablonas“). Taip sutaupoma laiko paruošiant klasės pageidavimų lentelę. Jei dalykų trumpiniai sutampa – tada belieka tik išvardinti mokinius ir jų pageidavimus. Primename, kad trumpiniai turi sutapti visuose dokumento lapuose.

Stulpelyje „Viso“ yra automatiškai skaičiuojamas mokinio valandų skaičius per savaitę. Šį langelį pildyti nėra būtina, jis programos darbui įtakos neturi.

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.	Dor. ugd.		Leit.k.	1 užs.kalba			2 užs.kalba			Ist	Geo	Mat	Inf	Biol	Fiz	Chem	Menai			K	Pasirenkami				Viso									
2		P	S	T	E	A	V	R	Pr	A	V	R	Pr							M	m	D	a	T	e	K	b	r	i	t	e	k	s	g	
3	1	3b	Pavarde	1	Vardas	1							2	5	1	4	1	2			1			2			2							31	
4	2	3b	Pavarde	2	Vardas	2						2		5	1	4	1	2			1			2	2									31	
5	3	3b	Pavarde	3	Vardas	3						2		5	1	4	1	2			1			2	2									31	
6	4	3b	Pavarde	4	Vardas	4						2		5	1	4	1	2			1			2				2						31	
7	5	3b	Pavarde	5	Vardas	4						2		5	1	4	1	2				1		2				2						31	
8	6	3b	Pavarde	6	Vardas	5						2		5	1	4	1	2				1		2	2									31	
9	7	3b	Pavarde	7	Vardas	6						2		5	1	4	1	2			1			2				2						31	
10	8	3b	Pavarde	8	Vardas	7						2		5	1	4	1	2				1		2	2									31	
11	9	3b	Pavarde	9	Vardas	8						2		5	1	4	1	2				1		2				2						31	
12	10	3b	Pavarde	10	Vardas	9						2		5	1	4	1	2				1		2	2									31	
13	11	3b	Pavarde	11	Vardas	10						2		5	1	4	1	2			1			2				2						31	
14	12	3b	Pavarde	12	Vardas	11						2		5	1	4	1	2				1		2	2									31	
15	13	3b	Pavarde	13	Vardas	12						2		5	1	4	1	2				1		2	2									31	
16	14	3b	Pavarde	14	Vardas	13						2		5	1	4	1	2				1		2				2						31	
17	15	3b	Pavarde	15	Vardas	14						2		5	1	4	1	2				1		2				2						31	
18	16	3b	Pavarde	16	Vardas	15						2		5	1	4	1	2				1		2				2						31	
19	17	3b	Pavarde	17	Vardas	16						2		5	1	4	1	2				1		2				2						31	
20	18	3b	Pavarde	18	Vardas	17						2		5	1	4	1	2				1		2	2									31	
21	19	3b	Pavarde	19	Vardas	18						2		5	1	4	1	2				1		2				2						31	
22	20	3b	Pavarde	20	Vardas	19						2		5	1	4	1	2			1			2	2									31	
23	21	3b	Pavarde	21	Vardas	20						2		5	1	4	1	2				1		2	2									31	
24	22	3b	Pavarde	22	Vardas	21						2		5	1	4	1	2				1		2				2						31	
25	23	3b	Pavarde	23	Vardas	22						2		5	1	4	1	2				1		2				2						31	
26	24	3b	Pavarde	24	Vardas	23						2		5	1	4	1	2				1		2	2									31	
27	25	3b	Pavarde	25	Vardas	24						2		5	1	4	1	2				1		2	2									31	
28	26	3b	Pavarde	26	Vardas	23						2		5	1	4	1	2				1		2				2						31	
29	27	3b	Pavarde	27	Vardas	24						2		5	1	4	1	2				1		2	2									31	
30	28	3b	Pavarde	28	Vardas	25						2		5	1	4	1	2				1		2				2						31	
31	29	3b	Pavarde	29	Vardas	25						2		5	1	4	1	2				1		2				2						31	
32																																			

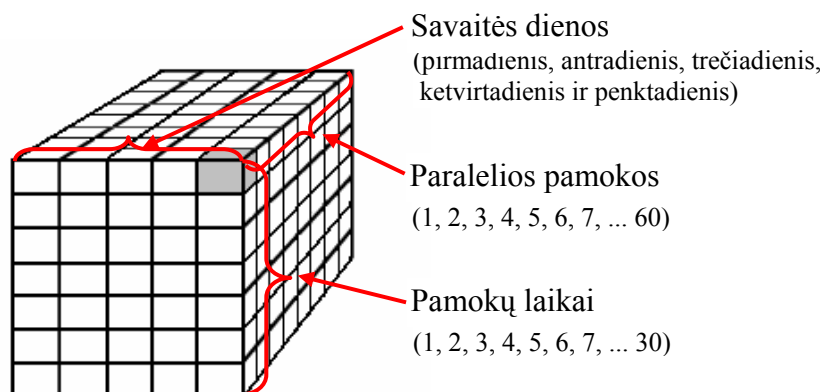
6 pav. Darbo lapo „3b“ pavyzdys

Šiame lape nurodomi mokinių pasirinkimai. Mokinio eilutės ir dalyko inicialu pažymėto stulpelio susikirtime nurodomas savaitinių valandų skaičius. Stulpelyje „Viso“ programa automatiškai suskaičiuoja kiek pamokų per savaitę turės mokinys. Šis skaičius negali būti mažesnis nei 30 ir didesnis nei 34.

Įvedus tvarkaraščio sudarymui reikalingą informaciją duomenų failas saugoma „XML“ formatu. Išsaugojus bet kokių kitu formatu sistema duomenų nesupras ir jie nebus nuskaityti į programą, o tai reiškia, kad nebus sudarytas tvarkaraštis.

3.3. Tvarkaraščio sudarymo principai

Tvarkaraščiui sudaryti buvo pasirinkta duomenų struktūros forma kubas (7 pav.). Vienoje kubo ašyje yra savaitės dienos, jos 5, nes pamokos išdėliojamos penkiose darbo savaitinės dienose. Kitoje kubo ašyje pamokų laikas. Programoje numatyta, kad šis laikas gali plėstis iki 30 pamokų per dieną, tačiau sudarant tvarkaraštį stengiamasi pamokas sudėti į kuo mažesnę pamokų laikotarpių skaičių. Trečiojoje kubo ašyje išdėstomos paraleliai vykstančios pamokos.

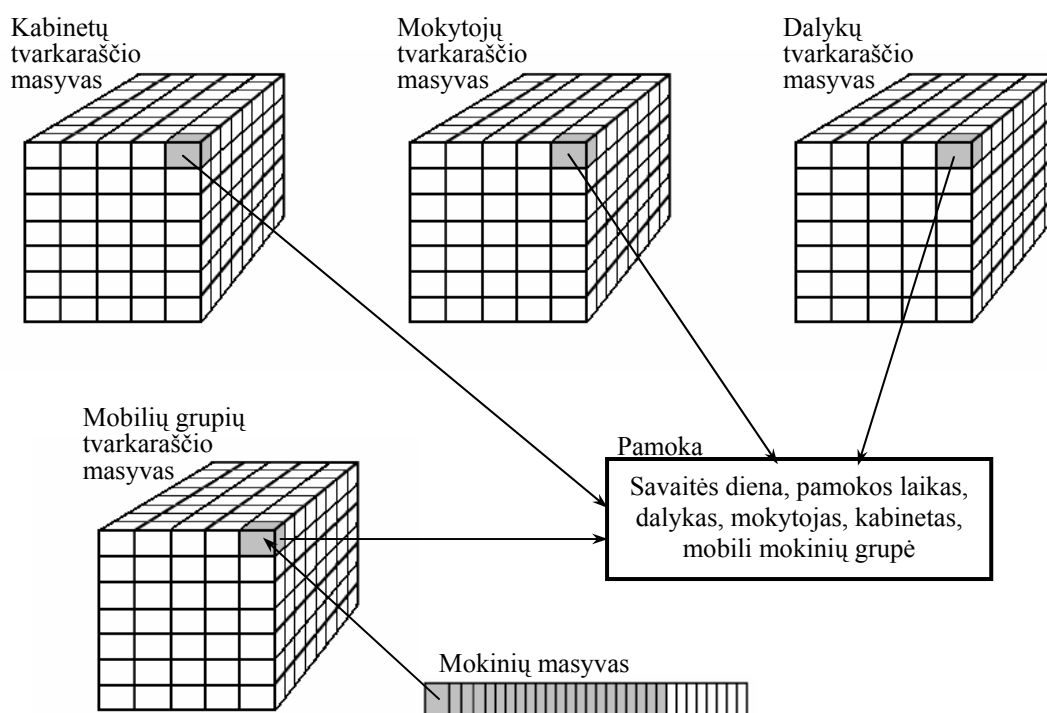


7 pav. Duomenų struktūros forma

Mokyklos tvarkaraščio sudarymo ir optimizavimo programinė sistema nuskaičiusi pradinis tvarkaraščio duomenis į masyvus sudaro pirminį tvarkaraštį. Šio etapo pagrindinis tikslas sutalpinti visus reikiamus dalykus į bendrą tvarkaraštį. Jau nuo pirmų žingsnių stengiamasi jį sudaryti kuo geresnį. Pamokos į tvarkaraštį sudedamos pagal dalykam suteiktus prioritetus. Pirmiausiai tvarkaraštyje sudedami aukščiausią prioritetą turintys dalykai, o po to eilės tvarka pagal sudėtingumą išdėstomi kiti. Sudėtingiausi dalykai sudedami pirmoje dienos pusėje, darbingiausiu mokinio dienos laiku. Pamokos į tvarkaraštį dedamos taip, kad dvi dienas iš eilės tuo pačiu metu nebūtų mokomas tas pats dalykas. Šio etapo metu įvertinami visi privalomi reikalavimai, t.y. fiziniai ribojimai kurie privalo būti patenkinti:

Profiliuotose mokyklose tos pačios klasės mokiniai turi galimybę pasirinkti mokytis skirtingus dalykus. Todėl sudarant pirminį tvarkaraštį formuojamos moksleivių mobilios grupės, pagal jų pasirinktus dalykus. Programinė sistema užtikrina, kad mobilias grupes galima formuoti tik iš to paties lygio mokinių. Šiame etape įvertinamas pirminiame duomenų faile prie dalyko nurodytas maksimalus mokinių skaičius su kuriuo mokytojas gali dirbti pamokoje. Šį mokinių skaičių vartotojas gali nusistatyti individualiai, pagal savo mokyklos reikalavimus.

Taip formuojant mobilias grupes konkreitiems dalykams, priskiriant jiems mokytojus ir kabinetus sukuriamos tvarkaraščio pamokos. Pamokos įvertinus ribojimus sudedamos į tvarkaraštį. Pamokų sudėjimas į tvarkaraštį tai tarsi atskirų duomenų kubų pildymas (8 pav.).



8 pav. Tvarkaraščio struktūra

Informacija apie vieną tvarkaraščio pamoką kaupiama duomenų kubuose tomis pačiomis koordinatėmis. Turėdami taip sukauptą informaciją galime formuoti įvairius tvarkaraščius. Tvarkaraščiai vienas nuo kito gali skirtis savo detalumu, priklausomai nuo to iš kurių duomenų kubų informacija pateikiama vartotojui.

Mokyklos tvarkaraščio sudarymo ir optimizavimo programinėje sistemoje sudaromi šie mokyklos tvarkaraščiai:

- Mokyklos tvarkaraštis;
- Individualūs mokinių tvarkaraščiai;
- Individualūs mokytojų tvarkaraščiai;
- Individualūs kabinetų tvarkaraščiai;
- Grupių tvarkaraščiai.

Sudarius pirminį tvarkaraštį atliekamas optimizavimo procesas. Optimizavimas orientuotas į mokytojų ir mokinių langų uždarymą bei pamokų perkėlimą iš mokytojo pageidaujamo laisvadienio.

3.4. Tvarkaraščio optimizavimo procesas

Mokyklos tvarkaraščio sudarymo ir optimizavimo programinėje sistemoje sprendžiamas daugiakriterinis optimizavimo uždavinys. Optimizavimą pagal daugelį kriterijų reiktų suprasti kaip alternatyvos pasirinkimą pagal daugelį ją apibūdinančių požymių. Sprendžiant daugiakriterinius optimizavimo uždavinius, tenka siekti kompromiso tarp įvairių kriterijų. Kompromisiniai sprendiniai gali būti įvairūs. Iš jų vertingais laikomi tik tie sprendiniai, kurie yra optimalūs Pareto prasme (Žilinskas 2000).

Tam, kad daugiakriterinį uždavinį pakeisti vienkriteriu buvo įvestas "baudos taškų" skaičiavimas. Kriterijams vartotojas nustato svorius atsižvelgdamas į jų svarbą, o tai priklauso nuo mokymo įstaigos keliamų reikalavimų tvarkaraščiui. Kuo daugiau baudos taškų skiriame kriterijui, tuo didesnis dėmesys kreipiamas į tą apribojimą. Ieškomas toks tvarkaraštis, kuris minimizuotų bendrą baudos taškų funkciją. Optimizavimo metu svarbu maksimizuoti bendrą rezultatą ir visai nesvarbu kurios komponentės sąskaita jis gaunamas.

Optimaliu tvarkaraščiu laikomas tas už kurį neradom geresnio, kurio baudos taškų skaičius yra mažiausias. Baudos taškai skaičiuojami pagal (2) formulę.

$$C = \sum_r c_r N_r, \quad (2)$$

kur c_r -bauda už normų r pažeidimą; N_r -pažeidimų kiekis; r kinta nuo 1 iki 9.

Vartotojas, dirbdamas su mokyklos tvarkaraščio sudarymo ir optimizavimo programine sistema, gali įtakoti tvarkaraščio sudarymą nustatydamas optimizavimo parametrus (kriterijus) pagal tam tikros mokyklos reikalavimus.

Baudos taškai apskaičiuojami įvertinant šiuos parametrus:

- langus mokytojams tarp pamokų;
- langus mokiniams tarp pamokų;
- mokytojų laisvadienių pažeidimą;
- išdirbtų valandų viršijimą;
- dalykų prioritetų pažeidimą;
- mokinių sumaišymą grupėse.

Baudų taškai mokytojų langui – nurodo kiek bus skiriama baudos taškų, jei mokytojas turės langą tarp dėstomų pamokų. Programoje vartotojas nurodo kiek yra skiriama baudos taškų už vieną langą. Jei langų bus daugiau – programa pati padaugins šį skaičių iš susidariusių langų kiekio. Paprastai mokytojų yra žymiai mažiau nei mokinių, todėl jų tvarkaraščio langams skiriamas didelis baudos taškų kiekis. Jei norima sudaryti tvarkaraštį mokinio naudai, šį skaičių reikia vesti nedidelį

Baudų taškai mokinių langui – nurodo kiek bus skiriama baudos taškų, jei mokinys turės langą tarp pamokų. Kaip ir mokytojams, nurodomi baudos taškai už vieną langą.

Baudų taškai mokytojų laisvadienių pažeidimui – nurodo kiek bus skiriama baudos taškų, jei mokytojas turės dirbti per pageidaujamus „laisvadienius“. Kuo daugiau skiriama baudos taškų, tuo didesnė tikimybė, kad mokytojas nedirbs per savo „laisvadienius“

Baudų taškai išdirbtų valandų viršijimui – nurodo, kiek baudos taškų bus skiriama, jei bus viršytas pamokų skaičius per dieną. Jei nustatėm, kad per dieną turi būtine daugiau kaip 7 pamokos, o programa sudėliojo 8, tada bus skiriami baudos taškai esamam tvarkaraščiui.

Baudų taškai dalykų prioritetų pažeidimui – nurodo, kiek baudos taškų bus skiriama, jei aukšto prioriteto dalykas dėstomas antroje dienos pusėje.

Baudos taškai mokinių sumaišymui grupėse – nurodo, kiek baudos taškų bus skiriama, jei uždarinėjant mokinių langus atsiras papildomas mokinių sumaišymas grupėse.

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

$$\min_{\tau \in \theta} C(\tau), \quad (3)$$

kur $C(\tau)$ yra tvarkaraščio τ bendras baudos taškų skaičius, θ yra eilė sudarytų tvarkaraščių tenkinančių fizinius apribojimus.

Vartotojui sudaryta galimybė įtakoti optimizavimo procesą. Vartotojas gali individualiai nustatyti skiriamus baudos taškus atsižvelgdamas į jų svarbą mokyklai. Tai leidžia įvertinti individualius mokymo įstaigos euristinius kriterijus.

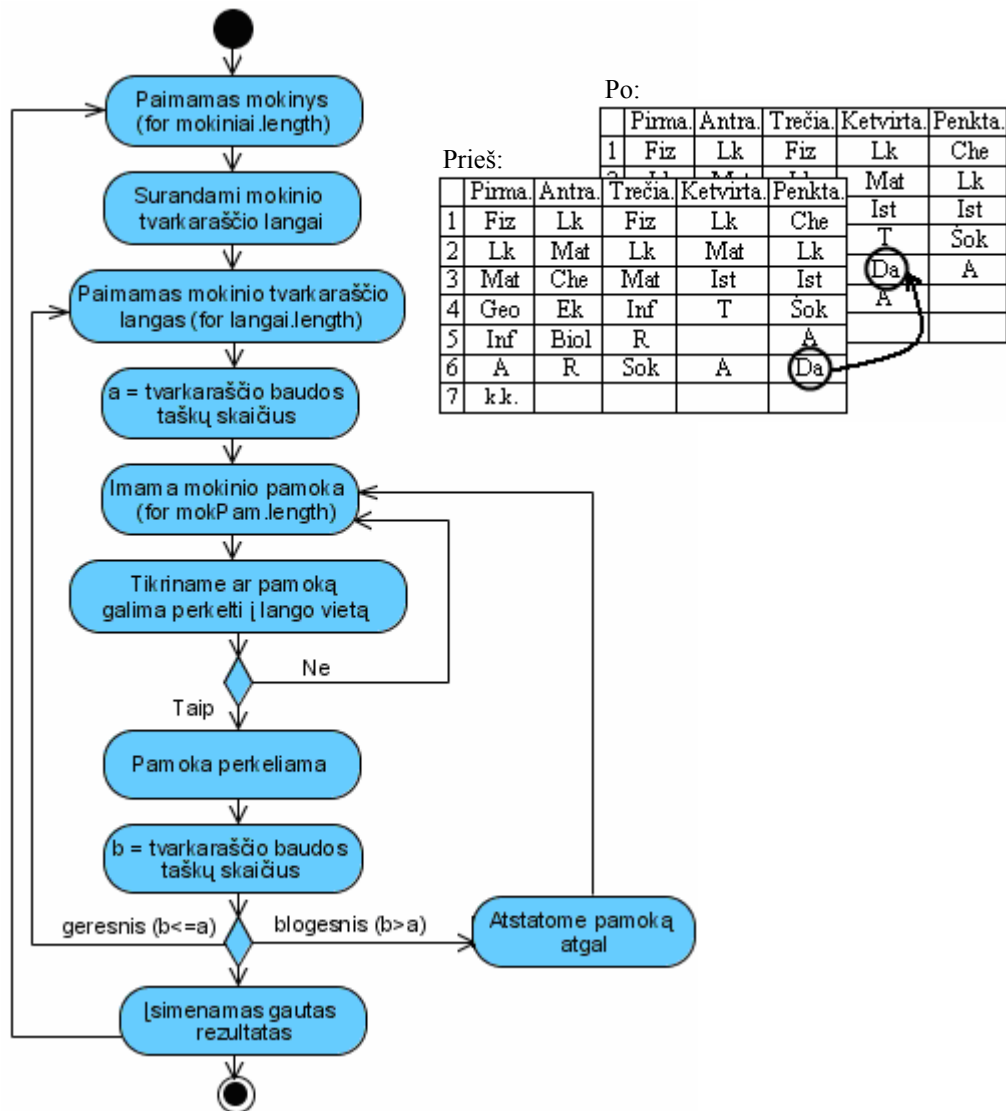
Mokyklos tvarkaraščio sudarymo ir optimizavimo programinėje sistemoje realizuoti 3 optimizavimo metodai:

- Monte-Karlo;
- Modeliuojamojo atkaitinimo;
- Bayes.

Optimizavimas orientuotas į mokytojų ir mokinių langų uždarymą bei pamokų perkėlimą iš mokytojo pageidaujamo laisvadienio. Naujoje programinės sistemos versijoje realizuotas mokinių langų uždarymo algoritmas.

Programinės sistemos darbo metu, pradėjus tvarkaraščio optimizavimo procesą pirmiausiai atliekamas mokinių langų uždarymas. Toliau tvarkaraštis optimizuojamas pagal vartotojo pasirinktą optimizavimo metodą, uždarinėjant mokytojų langus ar perkeltiant pamokas iš mokytojų pageidaujamų laisvadienių. Baigus su mokytojais vėl atliekamas mokinių langų uždarymas.

Mokinių langų uždarymo metu surandamas mokinio langas tvarkaraštyje. Imama mokinio paskutinė dienos pamoka ir tikrinama ar ją galima perkelti į mokinio surasto lango vietą (9 pav.). Jei šios pamokos perkelti negalima, imama kita ir taip ieškoma kol surandama ką galima perkelti. Jei galimos perkelti pamokos nerandame, tvarkaraštyje paliekamas langas. Taip bandoma uždaryti visus mokyklos tvarkaraštyje esančius mokinių langus.



9 pav. Mokinio lango uždarymas

Mokinio langą galima uždaryti tuo atveju, jei randama lango metu kitiems mokiniams vykstanti norima perkelti pamoka. Tikrinama ar pamokoje į kurią norime perkelti mokinį yra laisvų vietų, t.y. ar perkėlus nebus viršytas maksimalus mokinių skaičius su kuriuo mokytojas gali dirbti pamokoje. Programinė sistema užtikrina, kad mokinys bus perkeltas tik į to paties lygio mokinių grupę, prie kurios buvo galima prijungti mokinį formuojant mobilią grupę.

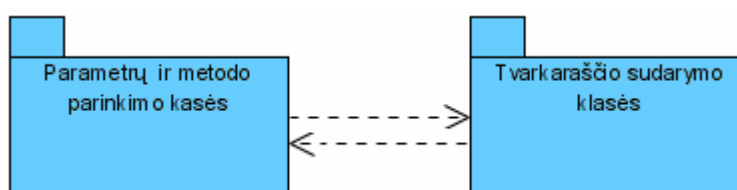
Mokytojų ir mokinių langų uždarymas labai priklauso nuo vartoto užduotų parametrų ir sudaryto pirminio tvarkaraščio kokybės.

3.5. Mokyklos tvarkaraščių sudarymo ir optimizavimo programinės sistemos architektūra

Šioje dalyje aprašyti mokyklos tvarkaraščių sudarymo ir optimizavimo programinės sistemos komponentai, jų ryšiai vienas su kitu ir su aplinka bei principai, kuriais pagrįstas projektas, kad projektavimo sprendimai būtų gerai suprantami ir įsisavinami.

Programinės sistemos architektūrinius apribojimus lemia pasirinktos kūrimo priemonės: XML technologijos, Java objektinio programavimo kalba, JSP dinaminiai puslapiai integruojantys Java, Tomcat konteineris.

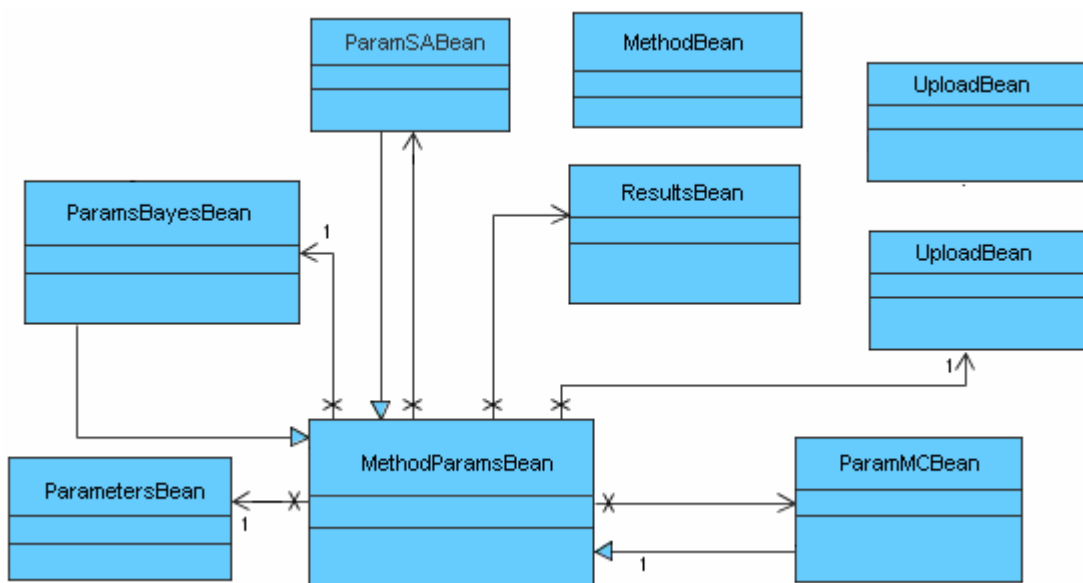
3.5.1. Sistemos statinis vaizdas



10 pav. Sistemos išskaidymas į paketus

3.5.2. Paketų detalizavimas

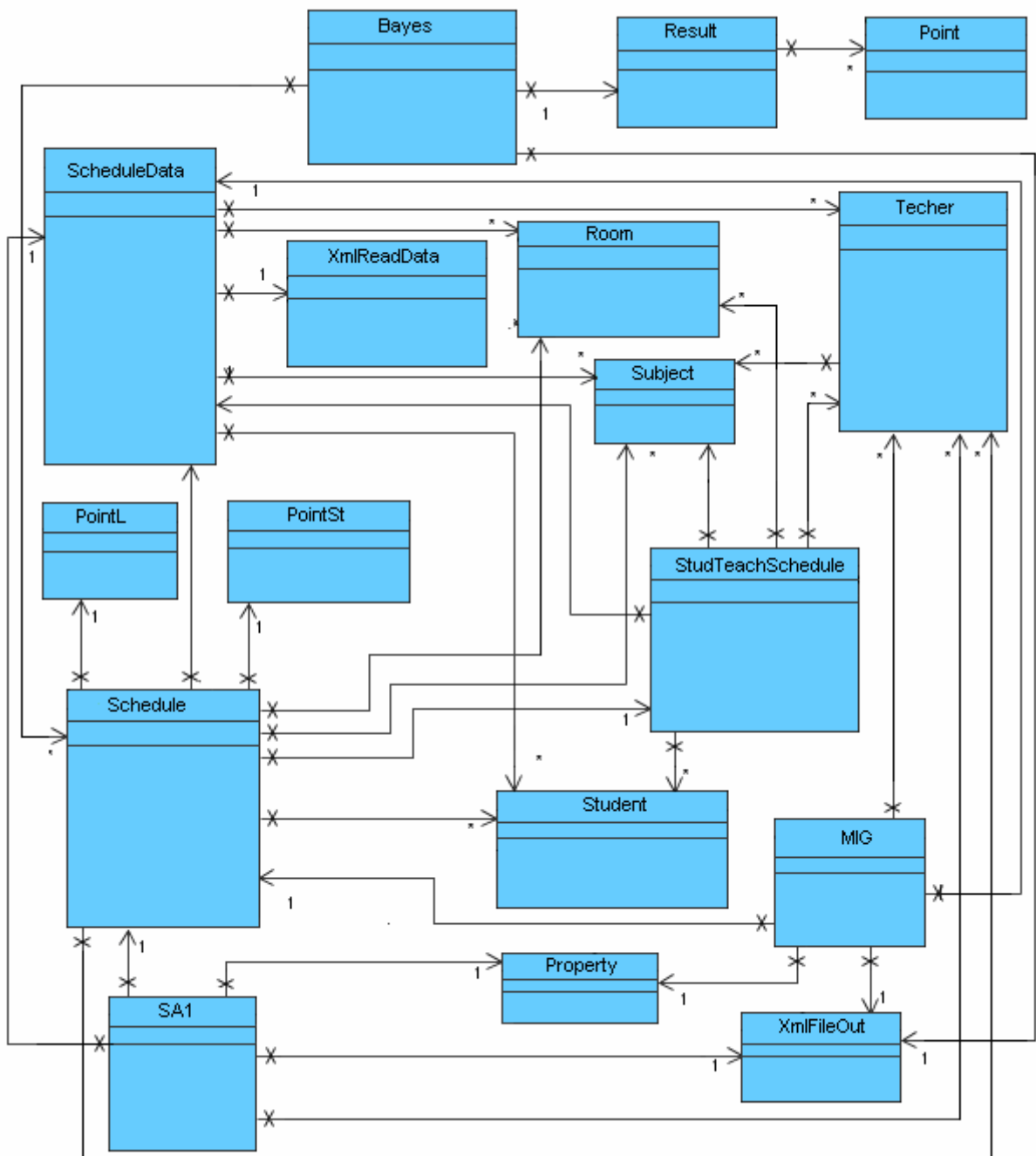
Parametų ir metodo paėmimo klasės yra skirtos optimizavimo parametrams nustatyti ir optimizavimo metodui pasirinkti.



11 pav. Parametų ir metodo paėmimo klasių diagrama

Parametų ir metodo paėmimo klasių paketas bendradarbiauja su JSP komponentais ir su tvarkaraščio sudarymo klasių paketu. Priimami iš JSP perduodami vartotojo duomenys ir perduodami į tvarkaraščio sudarymo klasių paketą.

Tvarkaraščio sudarymo klasės skirtos pirminių duomenų saugojimui, pirminio tvarkaraščio sudarymui, bendro optimalaus tvarkaraščio sudarymui ir sudarymui mokytojų ir mokinių tvarkaraščių.



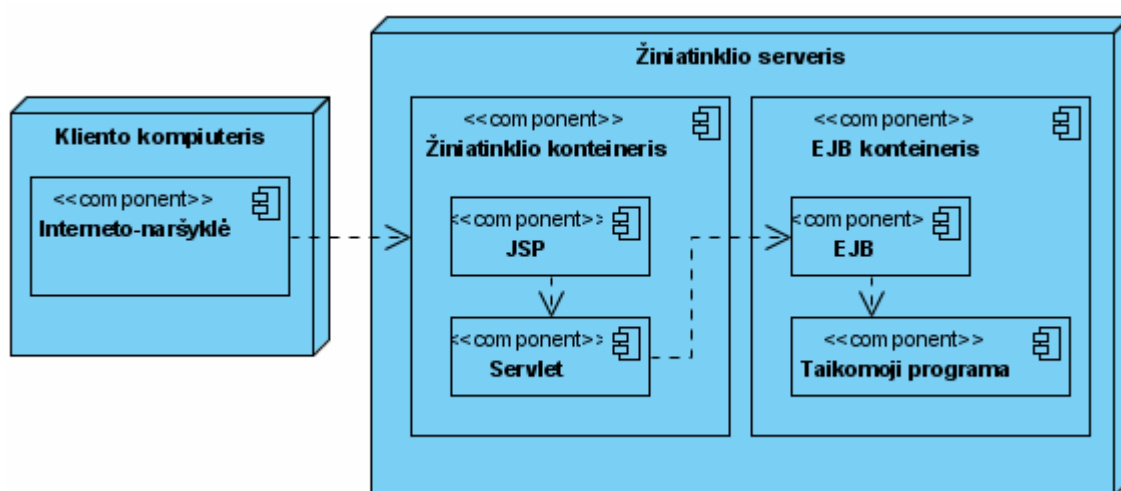
12 pav. Tvarkaraščio sudarymo klasių diagrama

Tvarkaraščio sudarymo klasių paketas skirtas tvarkaraščių sudarymui ir optimizavimui. Tvarkaraščio sudarymo klasių paketas bendradarbiauja su parametų ir metodo paėmimo klasių paketu. Darbui turi būti vartotojo nurodyti optimizavimo parametrai ir pasirinktas optimizavimo metodas.

3.5.3. Išdėstymo vaizdas

Servlet ir JSP komponentai, naudojami kartu, leidžia atskirti verslo logiką nuo vaizdinės dalies, taip supaprastinant interneto sistemos kūrimą ir palaikymą. JAVA kalbos servlet režimas užtikrina visišką nepriklausomumą nuo operacinės sistemos. Pašalinama būtinybė daryti sudėtingas diegimo ir atnaujinimo procedūras.

Vartotojas turi turėti galimybę pasinaudoti programine sistema iš bet kurios darbo vietos turinčios WWW tinklo naršyklę. Kliento naršyklei paprašius puslapio ir specialiam interneto serveriui nusprendus pagal adresą, kad reikia vykdyti būtent šios klasės metodus, klientui bus įrašytas atsakymas. Servlet atlieka loginius veiksmus ir galiausiai nurodo JSP serveriui klientui pateikti JSP puslapį.



13 pav. Sistemos išdėstymo vaizdas

Kai vartotojas internetinėje svetainėje paspaudžia ant programos paleidimo nuorodos, automatiškai paleidžiama programa darbui. Tai leidžia vartotojui naudotis programine sistema be jokių instaliavimo procedūrų. Servlet panaudojimas įgalina pradėti dirbti su taikomąja programa per bet kokią naršyklę, su bet kokia platforma, iš visur per tinklą, saugiu būdu. Naudojant servlet režimą programinės sistemos veikimo greitis priklauso nuo Interneto ryšio greičio serveryje ir nuo vartotojo ryšio spartos.

Programinė sistema realizuojama naudojant Java programavimo kalbą. Java kalboje yra įdiegtas saugumo mechanizmas, kuris patikrina ar programinis kodas nėra kenksmingas, kiekvienam veiksmui, kuris keičia programos išorę. Java kalba yra interpretuojama ir tai lemia jos nepriklausomumą nuo architektūros ir operacinės sistemos, saugumą bei paprastumą, kadangi interpretavimo aplinkoje gali būti atliekami įvairūs pagalbiniai servisai, Java priemonėmis sukurtos aplikacijos gali būti perkeltos į serverius ir pritaikytos naudojimui, nepriklausomai nuo ateityje pasirinktos techninės įrangos ar operacinės sistemos.

4. MOKYKLOS TVARKARAŠČIŲ SUDARYMO IR OPTIMIZAVIMO PROGRAMINĖS SISTEMOS KOKYBĖ

Produkto kokybė yra užsakovo norų ir realių sistemą kuriančio kolektyvo galimybių kompromiso rezultatas. Darbe buvo atlikta mokyklos tvarkaraščių sudarymo ir optimizavimo programinės sistemos kokybės analizė.

4.1. Kokybės vertinimas

Patikimumas. Svarbiausias programinės sistemos kokybės aspektas yra jos patikimumas, t.y. geba veikti nustatytą laiko periodą nenukrypstant nuo specifikacijoje pateiktų reikalavimų. Programinės sistemos patikimumas buvo padidintas ištaisius testavimo fazės metu aptiktas klaidas. Testavime buvo naudojama „baltos dėžės“ ir „juodos dėžės“ testavimo strategija. Visi pastebėti sistemos veikimo sutrikimai buvo ištaisyti.

Korektiškumas. Sukurta programinė įranga sudaro mokyklos tvarkaraščius ir juos optimizuoja pagal vartotojo užduotus parametrus. Duomenys sistemoje vienareikšmiškai atitinka pirminių duomenų failo duomenis ir perduodant iš vieno komponento į kitą nesugadinami. Visi programinės sistemos sąsajos langai veikia korektiškai. Įvykus klaidai nuskaitant duomenis iš pirminių duomenų failo ar nurodžius netinkamus parametrus vartotojui sistema išveda paaiškinančius pranešimus. Iš anksto numatyti visas galimas situacijas, kuriant panašaus sudėtingumo sistemas, yra neįmanoma. Sistema dar kuriama ir ji turės būti tobulinama.

Našumas. Programinės sistemos veikimo greitis labia priklauso nuo Interneto ryšio Tomcat serveryje bei nuo vartotojo ryšio spartos. Sudėtingesni optimizavimo metodai yra imlūs laikui. Programos darbo metu atliekami skaičiavimai kurių kiekis priklauso nuo vartotojo užduotų parametrų. Pasirinkus optimizavimo veiksmą vartotojas turi galimybę pasirinkti optimizavimo iteracijų skaičių, tai garantuoja geresnį optimizavimo proceso veikimą, tačiau prailgina šio proceso trukmę.

Prieinamumas. Programinė sistema įdiegta 4 serveriuose. Taip užtikrinta tikimybė, kad sistema bus parengta darbui ir veiks visuomet, kai bus norima ją pasinaudoti.

Vartoseną. Programinės sistemos vartotojo sąsaja paprasta ir lengvai suprantama. Sistema turi vartotojo dokumentaciją ir labai išsamią pagalbą vartotojui lietuvių kalba. Realizuoti pagrindiniai funkciniai mygtukai, ji neapkrauta nereikalingais elementais, paaiškinimai ir pranešimai tikslūs ir aiškūs. Norint naudotis programine sistema jokių specialių mokymų ar įgūdžių nereikia, pakanka elementarių naudojimosi kompiuteriu žinių.

Saugumas. Visi duomenys apie mokytojus ir mokinius saugomi pradiniam duomenų faile. Ši failą turi tvarkaraščio sudarinėtojas. Programinės sistemos sugeneruotus rezultatus vartotojas išsisaugo savo kompiuteryje. Mokyklos tvarkaraščio sudarymo ir optimizavimo programinė sistema realizuota Java Servlet pagrindu, o ši technologija suteikia galimybę prireikus užtikrinti saugumo reikalavimus.

Kol kas programos naudojimas yra viešas, skirtas mokymo tikslams, todėl saugumo klausimai nebuvo sprendžiami.

Testuojamumas. Testavimas sistemos gana sudėtingas, nes vieno komponentų veikimas labai priklauso nuo kitų. Testuojant net menkiausią metodą reikalingi dideli testinių duomenų rinkiniai.

Reikalavimai programinei sistemai keičiasi jau sistemos kūrimo metu, nes keičiasi aplinka. Todėl net nauja pateikta sistema netenkina visų srities reikalavimų.

Mūsų kuriama programinė sistema nuolat tobulinama. Vis įvedami nauji parametrai kurių pagalba vartotojas gali įtakoti tvarkaraščio sudarymą, kad šis labiau atitiktų mokyklos keliamus reikalavimus tvarkaraščiui, t.y. ugdymo organizavimo mokykloje ypatumus ir galimybes. Vartotojo nurodomų parametrų dėka sistema tampa atsparesnė aplinkos pokyčiams.

4.2. Trūkumai ir tolesnio tobulinimo galimybės

Mokyklos tvarkaraščio sudarymo ir optimizavimo programinė sistema nuolat tobulinama ieškant naujų efektyvių sprendimų. Kad programinė sistema būtų naudojama realioms uždaviniais spręsti, ji privalo tenkinti vartotojo reikalavimus. Mūsų sukurta programinė sistema vis dar turi kur tobulėti.

Pasiūlymai tolimesniam programinės sistemos tobulinimui:

- Praplėsti vartotojo sąsają įvedant tvarkaraščio rankinį koregavimą. Šiuo metu vartotojas tvarkaraštį gali koreguoti tik pasisiuntęs į savo kompiuterį. Tai gana painus ir sudėtingas darbas. Atliekant rankinį tvarkaraščio koregavimą atsiranda didelė klaidų tikimybė, nes visų daromų pakeitimų korektiškumą vartotojas turi sutikrinti pats. Tvarkaraščių sudaryme rankinis koregavimas ankščiau ar vėliau tampa aktualus, kad ir kaip sėkmingai būtų sugeneruotas tvarkaraštis.
- Būtų tikslinga padaryti kiekvienam mokytojui atskirai baudos taškų nustatymą dėl tvarkaraščio langų, nes ne visiems mokytojams jie turi vienodą svarbą. Pvz. mokyklos administracijos darbuotojai mokykloje būna visą dieną ir tvarkaraščio langai jiems nėra svarbūs, o mokytojams kurių darbo krūvis mažas geriau turėti kompaktišką tvarkaraštį.

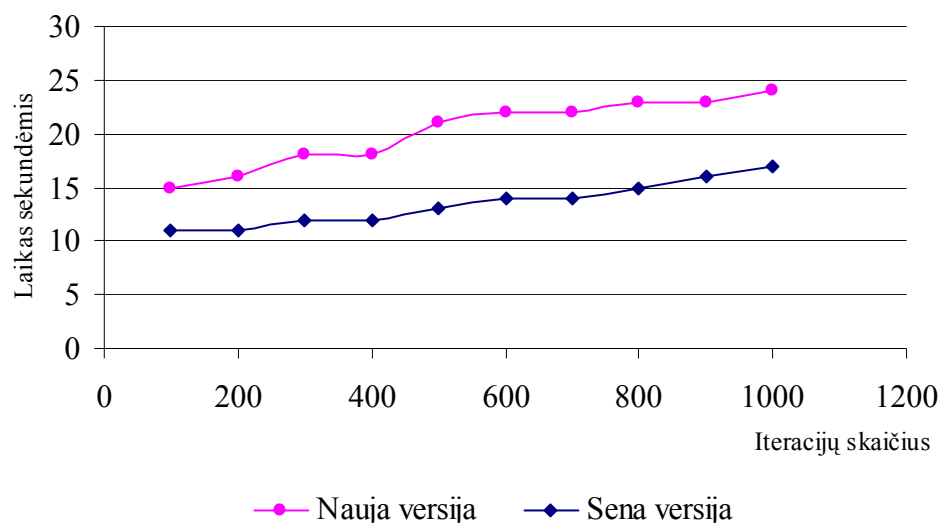
- Šiuo metu automatiškai sudarius tvarkaraštį nėra informuojamas vartotojas apie pamokas kuriose susidarė mažas mokinių skaičius. Realiose mokyklose reikalingas ribojimas pvz., kad dalyką pasirinkus mažiau kaip penkiems mokiniams, pamoka nebus skaitoma. Sistema turėtų išvesti tokių pamokų sąrašą, kad vartotojas apie jas žinotų ir galėtų ieškoti sprendimų.
- Mokinių ir mokytojų tvarkaraščių išvedime reikėtų, kad būtų išvedama pilna informacija apie pamoką. Mokinių tvarkaraščiuose prie dalyko trumpinio išvesti mokytojo pavardę ir kabineto numerį. O mokytojui - grupės pavadinimą ir kabineto numerį. Šiuo metu pilna informacija matoma užvedus pelytės žymeklį ant konkrečios pamokos.
- Šiuo metu sistemoje pamokai mokytojas priskiriamas automatiškai. Reikėtų sistemoje papildomo apribojimo kurioms klasėms mokytojas gali ar pageidauja vesti pamokas.
- Nutolusiame kompiuteryje veikianti programa patogi mokykloms turinčioms silpnus kompiuterius. Vis daugėja mokyklų kurios apsirūpinusios gera informacine technika, todėl joms būtų patogiau naudoti JavaWebStart technologija realizuotą sistemos versiją. Tokiu atveju visi programos skaičiavimai būtų atliekami vartotojo kompiuteryje ir programos darbo neįtakotų serverio greitis bei vartotojo turimo ryšio sparta. Šiuo metu sistemos funkcionalumas pasinaudojant JavaWebStart technologija yra dalinai realizuotas. Nėra pilnai realizuota vartotojo sąsaja.

5. MOKYKLOS TVARKARAŠČIŲ SUDARYMO IR OPTIMIZAVIMO PROGRAMINĖS SISTEMOS EKSPERIMENTINIS TYRIMAS

Optimalius tvarkaraščių sprendinius galime rasti tik mažos apimties uždaviniams. Realūs mokyklos tvarkaraščiai yra didelės apimties. Net ir neoptimalaus sprendinio radimas pareikalauja didelių skaičiavimo ir laiko sąnaudų.

Testuojant sukurtą programinę sistemą, naudojomes testiniais pirminių duomenų failais kuriuose suvesti duomenys atitinka vidutinio dydžio mokyklos duomenų kiekius. Gaunami rezultatai labai priklauso nuo vartotojo užduotų parametrų ir nuo mokinių pasirinkimų įvairovės. Norėdami gauti tikslesnius rezultatus, visus vieno tipo eksperimentus atlikome su tais pačiais nustatymais, keisdami tik tiriamą parametną ar resursą.

Programinė sistema darbo metu atlieka skaičiavimus kurių kiekis priklauso nuo vartotojo užduotų parametrų. Paveiksle (14 pav.) pavaizduota programos darbo laiko priklausomybė nuo iteracijų skaičiaus naujoje ir senoje programos versijoje naudojant Monte-Karlo optimizavimo metodą.



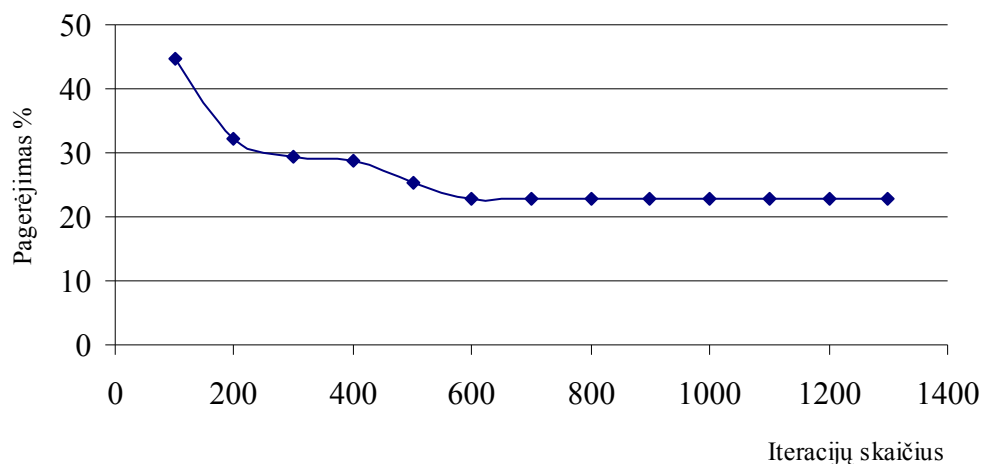
14 pav. Programinės sistemos darbo laiko priklausomybė nuo iteracijų skaičiaus

Didesnis iteracijų skaičius garantuoja geresnį optimizavimo rezultatą, tačiau prailgina šio proceso trukmę. Įvedus naujus apribojimus pailgėjo programinės sistemos atliekamų skaičiavimų laikas. Tačiau šis kokybės pablogėjimas mažiau svarbus lyginant su tuo, kad įvedus naujus parametrus ir ribojimus gaunamas tvarkaraštis geriau tenkinantis jam keliamus reikalavimus.

Ankstesnėje programinės sistemos versijoje optimizavimas buvo orientuotas į mokytojų langų uždarymą bei pamokų perkėlimą iš mokytojo pageidaujamo laisvadienio. Tvarkaraštyje susidarę mokinių langai buvo vertinami, bet neuždarinėjami. Įvedus mokinių langų uždarymą buvo gauti geresni optimizavimo rezultatai. Paveiksle (15 pav.) pateikiami programinės sistemos darbo rezultatai kai uždarinėjami mokytojų ir mokinių langai.

Darbui nustatyti parametrai:

- Baudos taškai mokytojų langams – 300;
- Baudos taškai mokinių langams – 5;
- Baudos taškai mokytojo laisvadienio pažeidimui – 400;
- Baudos taškai išdirbtų valandų viršijimui – 700;
- Baudos taškai dalykų prioritetų pažeidimui –100.

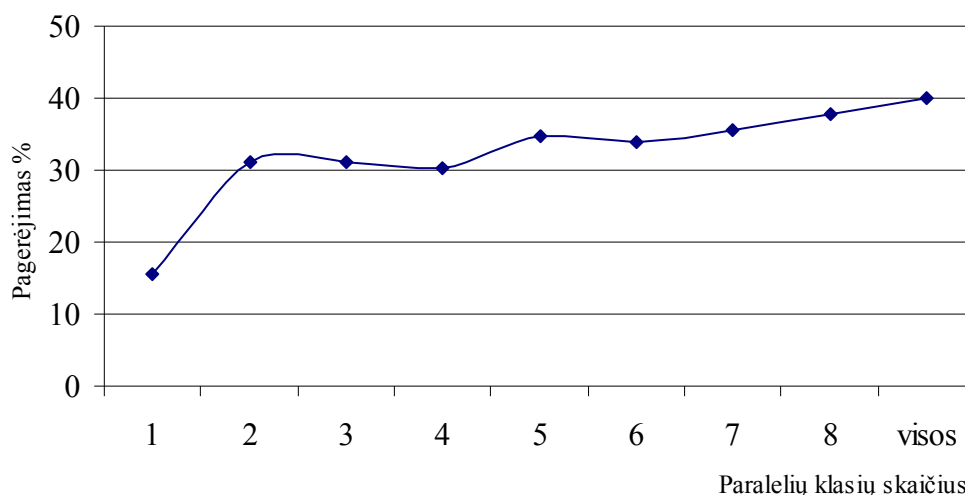


15 pav. Tvarkaraščio pagerėjimas įvedus mokinių langų uždarymą

Įvedus mokinių langų uždarymą gaunamas geresnis tvarkaraštis, net mokinių langams nustačius mažą baudos taškų skaičių. Skiriant baudos taškus mokytojų ir mokinių langams reikia nepamiršti, kad mokinių mokykloje yra daug daugiau nei mokytojų.

Paveiksle (15 pav.) matome, kad didinant iteracijų skaičių tvarkaraščio pagerėjimas nusistovi ties 20%.

Profiliuotos mokyklos tvarkaraščiui didelę įtaką turi paralelių klasių skaičius. Kuo mokykla daugiau turi paralelių klasių, kurių mokinius galima jungti į mobilies grupes, tuo gaunami geresni tvarkaraščių sudarymo rezultatai. Esant nors trims paralelių klasių komplektams, jau net pirminis tvarkaraštis gaunamas gana kompaktiškas. Paralelių klasių skaičius įtakoja ir optimizavimo procesą (16 pav.).



16 pav. Tvarkaraščio pagerėjimo priklausomybė nuo paralelių klasių skaičiaus

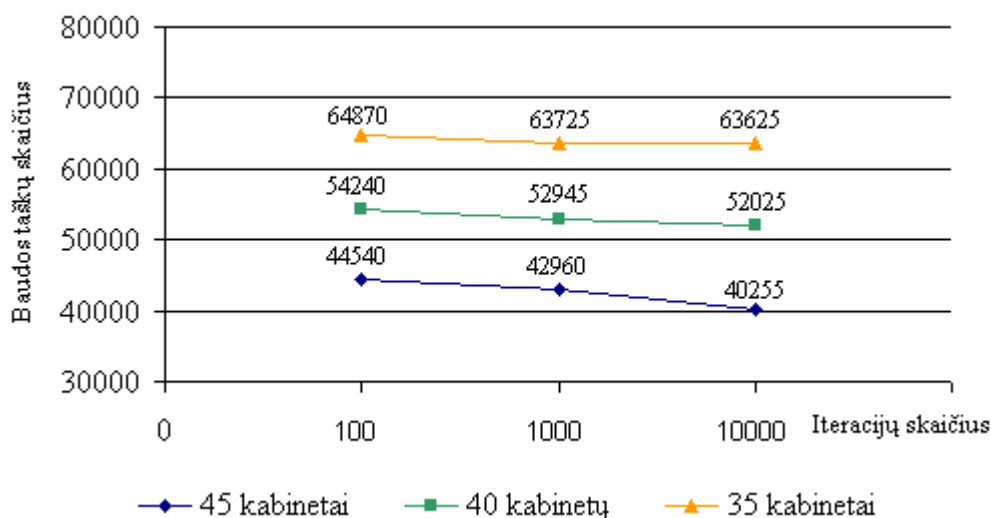
Blogiausi optimizavimo rezultatai gaunami kai turima po vieną paralelių klasių komplektą. Pasiekus tam tikrą paralelių klasių skaičių stebimas tik nežymus pagerėjimas.

Kai paralelių klasių daugiau nei trys ir mokinių pasirinkimas įvairus galima gauti tokią situaciją, kad mokytojų langai visiškai neuždarinėjami ar uždaromi minimaliai. Mokinių susimaišymas toks didelis, kad norint perkelti pamoką vis atsiranda vienas ar daugiau mokinių kurie riboja šį perkėlimą, nes tuo metu turi kitą pamoką. 1 lentelėje pateikiami tokių atvejų pavyzdžiai.

1 lentelė. Skirtingų pirminių duomenų optimizavimo suvestinės

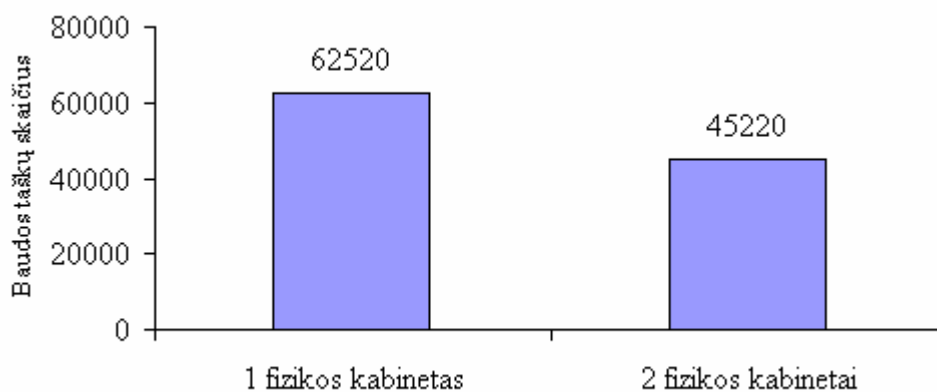
Klasių skaičius	16	24	24
Mokinių skaičius	434	644	644
Metodas	MIG	MIG	SA
Baudos taškai mokytojų langams	300	300	300
Baudos taškai mokinių langams	5	5	5
Baudos taškai mokytojo laisvadienio pažeidimui	400	400	400
Baudos taškai išdirbtų valandų viršijimui	700	700	700
Baudos taškai dalykų prioritetų pažeidimui	100	100	100
Iteracijos	1911	1729	100
Mokytojų langų pirminiame tvarkaraštyje	12	11	47
Mokytojų langų galutiniame tvarkaraštyje	11	11	46
Mokinių langų pirminiame tvarkaraštyje	92	137	305
Mokinių langų galutiniame tvarkaraštyje	15	65	271
Mokytojų laisvadienių pažeidimų pirminiame tvarkaraštyje	38	35	0
Mokytojų laisvadienių pažeidimų galutiniame tvarkaraštyje	32	31	0
Valandų skaičiaus viršijimas per dieną pirminiame tvarkaraštyje	0	0	0
Valandų skaičiaus viršijimas per dieną galutiniame tvarkaraštyje	0	0	0
Baudos taškai pirminio tvarkaraščio	4250	4160	47625
Baudos taškai po mokinių langų uždarymo	3950	3845	47625
Baudos taškai po mokytojų perkėlimų	3685	3830	47230
Galutinio tvarkaraščio baudos taškai po mokinių langų uždarymo	3535	3780	47155

Programos darbo rezultatai priklauso ir nuo mokyklos turimų resursų. Mokyklos turimų kabinetų įtaka sudarytam tvarkaraščiui pavaizduota paveiksle (17 pav.). Tvarkaraštis sudaromas 24 mokinių klasėms, optimizavimas atliekamas Monte-Karlo metodu.



17 pav. Mokyklos turimų kabinetų skaičiaus įtaka sudaromam tvarkaraščiui

Kai mokykla turi nepakankamai kabinetų, tenka ieškoti sprendimų kaip efektyviau išnaudoti turimus resursus. Darbo metu nustatėme, kad geresni tvarkaraščio sudarymo rezultatai gaunami, kai du skirtingi mokytojai dirbantys tame pačiame kabinete, moko skirtingo prioriteto dalykus. Ypatingai didelę įtaką tvarkaraščio kokybei daro mokyklos specialiai įrengtų kabinetų skaičius (18 pav.).



18 pav. Specialiai įrengtų kabinetų skaičiaus įtaka sudaromam tvarkaraščiui

Jeigu mokykloje yra trūkumas specialiai įrengtų kabinetų (pagal dalyko specifiką ar pakraipą), tvarkaraštyje tie kabinetai gaunami perkrauti ir dažnai dėl to viršijamas dienos pamokų lestinas skaičius. Tvarkaraštyje atsiranda sudėtingų dalykų pamokos antroje dienos pusėje. Mokyklai turint pakankamai kabinetų tvarkaraštis sudaromas geresnis.

Dalykams suteikus prioritetus, ir pagal juos dėlioiant pamokas, tvarkaraštis gavosi daug patogesnis darbui. Visi sunkūs dalykai tvarkaraštyje sudedami pirmoje dienos pusėje, mokinio darbingiausiu dienos metu.

Senoje versijoje:

	Monday	Tuesday	Wednesday	Thursday	Friday
1	Lk	Lk	Lk	Lk	Lk
2	Mat	Mat	Mat	Mat	k.k.
3	k.k.	T	A	A	A
4	A	R	R	Ist	Ist
5	Biol	Biol	Geo	Geo	Fiz
6	Da	Mm	Te	Te	
7					

Naujoje versijoje:

	Pirmadienis	Antradienis	Trečiadienis	Ketvirtadienis	Penktadienis
1	Lk	Mat	Lk	Mat	Lk
2	Mat	Lk	Fiz	Chem	Fiz
3	Chem	Ist	Ist	Lk	Geo
4	Geo	Inf	Biol	Biol	A
5	Da	A	k.k.	A	Sok
6	Sok	pb	Da	k.k.	
7		T			

19 pav. Mokinių tvarkaraščių palyginimas, kur dalykų prioritetai nevertinami ir kur įvertinami

Paveiksle (19 pav.) matome, kad senojoje versijoje dalykai į tvarkaraštį buvo dedami iš eilės einant per savaitės dienas. Pirmą sudedami visi būtini dalykai, o po to visi kiti. Naujoje versijoje dalykam suteikus prioritetus, pamokos pasiskirsto įvairiau. Pirmoje dienos pusėje sudedami dalykai turintys aukštesnį prioritetą ir dedama taip, kad dvi dienas iš eilės, tuo pačiu metu, nebūtų vedama to paties dalyko pamoka.

6. IŠVADOS

1. Mokyklos tvarkaraščių sudarymo uždaviniai yra labai sudėtingi. Jie turi labai daug ribojimų. Perrinkti visus galimus sprendinių variantus yra neįmanoma, todėl tenka naudoti įvairius euristinius paieškos algoritmus. Tokiuose uždaviniuose dažniausiai randamas apytikslis sprendinys, siekiant, kad jis būtų kuo artimesnis optimaliam.
2. Remiantis atlikta analize nustatyta, kad Lietuvos mokymo įstaigose naudojamos komercinės programos labiau pritaikytos tradicinėms mokykloms. Jos reikalauja iš vartotojo nemažai rankinio darbo, bet turi patogią ir gražią vartotojo sąsają.
3. Mūsų sukurta programinė sistema visiškai nepriklauso nuo operacinės sistemos, ji realizuota JAVA kalba Servlet režimu. Programinė sistema nereikalauja papildomos programinės įrangos, nereikia daryti sudėtingų diegimo ir atnaujinimo procedūrų. Atliekant srities analizę nepavyko rasti kitos mokyklos tvarkaraščio sudarymo programinės sistemos realizuotos Servlet režimu.
4. Sukurta programinė sistema kuri gana efektyviai optimizuoja profiliuotos mokyklos tvarkaraščius įvertindama privalomus ir pageidaujamus reikalavimus.
5. Sukurta sistema nereikalauja didelių vartotojo pastangų ar mokėjimo dirbti kompiuteriu. Gana patogi naudotis, tereikia perprasti pirminio duomenų failo sudarymą. Tvarkaraštis sudaromas ir optimizuojamas automatiškai. Vartotojui nereikia galvoti kur kokia pamoka galėtų vykti, sistema pati kontroliuoja, kad pamokos nesidubliuotų.
6. Sukurta programinė sistema tinka ne tik mokyklų tvarkaraščiams sudaryti ir optimizuoti, bet jau yra naudojama aukštosiose studijose kaip geras diskretaus optimizavimo pavyzdys.
7. Nustatyta, kad į sistemą įvedus naujus subjektyvius parametrus padidinami programos skaičiavimai, darosi sudėtingesnis tvarkaraščio sudarymas, tačiau tik jų dėka gaunami sprendimai, kurie tenkina individualius mokymo įstaigos reikalavimus.
8. Nustatyta, kad optimizavimo metu naudojant mokinių ir mokytojų „langu“ uždarymą buvo gauti apie 20 % geresni optimizavimo rezultatai.

9. Nustatyta, kad profilio pakraipa savo reikšmę įgauna tik didelėse mokyklose, turinčiose daugiau nei dvi paraleles klases. Esant nors trimis paralelių klasių komplektams tvarkaraštis gaunamas gana kompaktiškas.
10. Išanalizuota mokyklos turimų resursų ir užduodamų parametrų įtaka sudaromo tvarkaraščio kokybei.
11. Programinė sistema vis dar kuriama ieškant geresnių sprendimų tvarkaraščių sudaryme. Švietimo sistemoje vykstančios nuolatinės reformos skatina ieškoti sprendimų, kad programa būtų kuo lankstesnė pokyčiams.
12. Tikimės ateityje mokyklos tvarkaraščių sudarymo ir optimizavimo programinė sistema bus pritaikyta realių mokyklų tvarkaraščių sudaryme ir padės mokykloms išspręsti šį sudėtingą planavimo uždavinį.

LITERATŪRA

1. Abc. ABC Timetable, [žiūrėta 2007-11-22]. Prieiga per internetą:
<http://www.dirfile.com/abc_timetable.htm >
2. Abramson, D. *Constructing school timetables using simulated annealing: sequential and parallel algorithms*, Management Science, 1991, vol. 37, nr.1, p. 98-113.
3. Action. ACTion Scheduling Software, [žiūrėta 2007-11-22]. Prieiga per internetą:
<<http://www.timetable.co.za/>>
4. Alaburdienė, R.; Dovidauskaitė, S.; Nekiūnienė, V.; Rekerta, G.; Alaburda, M. Pamokų tvarkaraščiai, [žiūrėta 2008-10-25]. Prieiga per internetą:
<<http://www.soften.ktu.lt/~mockus/mimosa/index.htm>>
5. Alexandrov, V.; Dimov, I.; Karaivanova, A.; K.Tan, C. J. *Parallel Monte Carlo algorithms for information retrieval. Mathematics and Computers in Simulation*, 2003, vol 62, Issues 3–6.
6. Apynis, A. *Optimizavimo metodai*. Vilniaus universiteto leidykla, Vilnius, 2005, ISBN 9986-19-828-3.
7. aSc. aSc Tvardaraščiai, [žiūrėta 2008-10-25]. Prieiga per internetą:
<http://help.asctimetables.com/index.php?lang_id=10>
8. Bilgin, B.; Ozcan, E.; Korkmaz, E. E. *An Experimental Study on Hyper-Heuristics and Exam Scheduling*, PATAT2006, Springer-Verlag, selected papers, LNCS 3867, 2007, p. 394–412.
9. Birbas, T.; Daskalaki, S.; Housos, E. *School timetabling for quality student and teacher schedules*, Journal of Scheduling, 2009, vol. 12, Issue 2, p.177-197.
10. Burke, E.K.; Petrovic, S. *Recent research directions in automated timetabling*, European Journal of Operational Research 140, 2002, p.266–280.
11. Cooper, T.B.; Kingston, J.H. *The complexity of timetable construction problems*. In: Proceedings of the 1st International Conference on Practice and Theory of Automated Timetabling (PATAT 1995), LNCS 1153, Springer-Verlag, 1996, p. 283-295.
12. Costa, D. *A tabu search algorithm for computing an operational timetable*, European Journal of Operational Research, 1994, vol. 76, p. 98-110.

13. Erben, W.; Keppler, J. *A genetic algorithm solving a weekly coursetimetabling problem*, in: E. Burke, P. Ross (Eds.), *The Practice and Theory of Automated Timetabling: Selected Papers from the First International Conference*, Lecture Notes in Computer Science, Springer, Berlin, 1996, vol. 1153, p. 198–211.
14. Felinskas, G.; Sakalauskas, L. *Pareto tipo modeliai modeliujamojo atkaitinimo algoritmuose*. Lietuvos Matematikos rinkinys, T.43, 2003, spec. nr., p.573-578, ISSN 0132-2818.
15. Fet. FET Scheduling Software, [žiūrėta 2007-11-22]. Prieiga per internetą: <<http://lalescu.ro/liviu/fet/>>
16. Gaidukevičienė, R.; Kurilovas, E. *Comparative study of profiled school scheduling programs in Lithuania*, *Informatics in education*, 2005, vol. 4, nr.1, ISSN 1648-5831.
17. Gift. GIFT School, [žiūrėta 2007-11-22]. Prieiga per internetą: <http://www.schoolsoftwares.com/timetable_Software_overview.html>
18. Glibovec, N.N.; Medvidj, S.A. *Генетические алгоритмы и их использование для решения задачи составления расписания*, *Кибернетика и системный анализ*, 2003, nr.1.
19. Glover, F.; Laguna, M.; Marti, R. *Scatter Search*, *Advances in Evolutionary Computation: Theory and Applications*, A. Ghosh and S. Tsutsui (Eds.), Springer-Verlag, New York, 2003.
20. Hoos, H.H.; Stutzle, Th. *Stochastic Local Search. Foundations and Applications*. Morgan Kaufmann/Elsevier, 2004.
21. iMagic. iMagic Scheduling Software, [žiūrėta 2007-11-23]. Prieiga per internetą: <<http://rbytes.net/software/imagic-timetable-master-review/>>
22. Lantiv. Lantiv Timetabler [žiūrėta 2007-11-22] Prieiga per internetą: <<http://www.lantiv.com/en/default.html>>
23. Laumanns, M.; Bayesian, J.O. *Optimization Algorithms for Multi-objective Optimization*, 2004, ISSN 0302-9743 (Print), 1611-3349 (Online)
24. Machado, J.M.; Shiyou, Y.; Ho, S.L.; Peihong, N. *A common Tabu search algorithm for the global optimization of engineering problems*, *Computer methods in applied mechanics and engineering*, 2001, vol.190.
25. Mimoso. Mimoso Scheduling Software, [žiūrėta 2008-10-25]. Prieiga per internetą: <<http://www.mimosasoftware.com/>>

26. Misevičius, A.; Bukšnaitis, V.; Blonskis, J. *Euristiniai algoritmai: tikslai, iššūkiai, metodologija, perspektyvos*, Informacijos mokslai, 2006, ISSN 1392–0561.
27. Mockus, J. *A Set of Examples of Global and Discrete Optimization: Application of Bayesian Heuristic Approach*, Kluwer Academic Publishers, 2000, ISBN 0-7923-6359-0.
28. Mockus, J. *A Set of Examples of Global and Discrete Optimization by Jonas Mockus*, [žiūrėta 2009-01-22]. Prieiga per internetą: <<http://www.soften.ktu.lt/~mockus/>>
29. Pupeikienė, L. *Mokyklos profiliuotų klasių tvarkaraščių optimizavimo sistema*. Informacinės technologijos 2006: konferencijos pranešimų medžiaga; Kauno technologijos universitetas. T.1. Kaunas, 2006, ISBN 9955-09-993-3.
30. Rector. Pamokų tvarkaraštis "Rector", [žiūrėta 2008-10-25]. Prieiga per internetą: <<http://www.rector.spb.ru/lt/index.html>>
31. Sakalauskas, L.; Bartkutė, V.; Felinskas, G. *Optimality testing in stochastic and heuristic algorithms, Technological and economic development of economy*, t. XII, 2006, nr.1, p. 4-10, ISSN 1392-8619.
32. Sakalauskas, L.; Felinskas, G. *Tvarkaraščių su ribotais ištekliais sudarymo euristiniai algoritmai, jų tyrimas bei taikymai*, Informacijos mokslai, 2006, 38 tomas, p. 90-103. ISSN 1392-1487.
33. Statistika. Statistikos departamentas, [žiūrėta 2009-04-21]. Prieiga per internetą: <<http://www.stat.gov.lt> >
34. Ugdymo planai. 2007-2008 m. m. bendrieji ugdymo planai, [žiūrėta 2007-10-27]. Prieiga per internetą: <<http://www.smm.lt/ugdymas/bendrasis/index.htm#12> >
35. Wise.Wise timetable, [žiūrėta 2007-11-22]. Prieiga per internetą: <<http://www.wisetimetable.com>>
36. Žilinskas, A. *Matematinis programavimas*, VDU, Kaunas, 2000.
37. Žvirdauskas, D.; Adaškevičienė, V.; Tarnauskas, K.; Žvirdauskienė, R. *Priemonių skirtų vyresniųjų klasių mokinių mokymosi krūvio mažinimui, veiksmingumas*. Lietuvos Respublikos švietimo ir mokslo ministerijos tyrimas, 2006, [žiūrėta 2008 m. spalio 15 d.]. Prieiga per internetą: <http://www.smm.lt/svietimo_bukle/docs/tyrimai/Kruviu_ataskaita_0702.pdf >

7. TERMINŲ IR SANTRAUKŲ ŽODYNAS

Euristika – kokia nors taisyklė, principas ar metodika, nusakanti, kaip ir kokį veiksmȧ reikia atlikti, kad surasti kokį nors sprendinį ar priimti kokį nors sprendimȧ.

NP – tai tokia sudėtingumo klasė, apjungianti problemas, išsprendžiamas nedeterministinė Tiuringo mašina per polinomini̇ laikȧ – $m(n)$.

Optimizavimo parametrai – tai kabinetų skaičius, maksimalus valandų skaičius dienoje, baudos taškai dėl mokinių langų, baudos taškai dėl mokytojų langų, baudos taškai dėl laisvadienio pažeidimo, išdirbtų valandų viršijimo, ar pedagoginių reikalavimų pažeidimo.

Tvarkaraščio langas - vienos pamokos trukmės laisvo laiko tarpas tarp pamokų, neskaitant pietų pertraukai skirto laiko.

Objektyvūs parametrai, tai parametrai, kuriuos nustato pati mokykla savo nuožiūra. Tai reiškia, kad mokykla pati nurodo kiek kabinetų ji turi, ir kiek pamokų per dieną nori turėti.

Baudos taškai, tai taškai, pagal kuriuos bus vertinamas tvarkaraštis. Baudos taškai yra skiriami už nustatytų sąlygų netenkinimą. Pavyzdžiui: Jei mokinys turi langus – jo tvarkaraštis nėra idealus ir yra skiriami baudos taškai už kiekvieną mokinio langą.

XML - yra W3C rekomenduojama bendros paskirties duomenų struktūrų bei jų turinio aprašomoji kalba. Pagrindinė XML kalbos paskirtis yra užtikrinti lengvesnį duomenų keitimą tarp skirtingo tipo sistemų. XML – speciali duomenų vaizdavimo Internet'e kalba.

Java – objektinė programavimo kalba.

Java Servlet – Sun Microsystems sukurta technologija dinaminių puslapių generavimui. Servlet tai serveryje esantis komponentas, susietas su užklaustos adresu, generuojantis atsakymus.

Serveris - nutolęs kompiuteris, kuriame yra įdiegta speciali valdymo programa, tarpininkaujanti tarp vartotojo ir tame kompiuteryje esančios skaitmeninės informacijos.

Java Server Pages (JSP) – technologija, leidžianti dinamiškai generuoti HTML, XML, ar kito tipo puslapius. Ši technologija suteikia galimybę į statinį puslapį įterpti Java kodą ir kitą dinaminį turinį.

Enterprise Java Beans (EJB) – pagrindinė J2EE platformos technologija, realizuojanti komponentinę serverio pusės architektūrą.

SA - Modeliuojamojo atkaitinimo (Simulated Annealing) algoritmas.

MIG - Monte Karlo optimizavimo metodas.

Bayes -Bayes optimizavimo metodas.

PRIEDAI

PIRMINIO DUOMENŲ FAILO UŽPILDYMO PAVYZDYS

Darbo langas “Dalykai”:

Dalykai	grupes	priority	MaxStud
Ps	e1	1	30
T	e1	1	30
E	e1	1	30
Lk	but	5	30
A	e2	3	20
V	e2	3	20
R	e2	3	20
Pr	e2	3	20
A	e3	3	20
V	e3	3	20
R	e3	3	20
Pr	e3	3	20
Ist	pas	4	30
Geo	pas	4	30
Mat	but	5	30
Inf	pas	4	30
Biol	pas	4	30
Fiz	pas	5	30
Chem	pas	5	30
Mu	pas	1	30
Sok	pas	1	30
Tea	pas	1	30
Da	pas	1	30
k.k.	but	3	30
lot	pas	2	30
isp	pas	2	30
gp	pas	2	30
br	pas	4	30
pb	pas	2	30
ek	pas	4	30
pc	pas	2	30

Darbo langas "Mokytojai":

grupe	Eil.	Dalykas	Pavarde_Vardas	Kabinetas	Val.sk	Trump	Pageidavimai / Laisvadieniai	
1	1	Etika	Martinaitiene Laura	212; 213; 201;	22;	E		
	2	Etika	Patamsiene Jolanta	212; 213; 201;	16;	E	1;2;	
	3	Tikyba	Maciuleviciene Natalija	213; 212; 201;	13;	T	5;	
	4	Tikyba	Vasiliauskaite Nijole	213; 212; 201;	27;	T		
	5	Psichologija	Lasauskiene Lilija	201;	1;	Ps		
2	6	Leit.k.	Bundoniene Orinta	202;	21;	Lk		
	7	Leit.k.	Kalvelienu Laima	203;	21;	Lk		
	8	Leit.k.	Kucinskiene Anzelika	216; 219;	20;	Lk		
	9	Leit.k.	Miliseviciute Sigita	212; 208;	23;	Lk	5;	
	10	Leit.k.	Paradauskiene Asta	201; 207;	22;	Lk		
	11	Leit.k.	Poskeviciute Dalia	212; 206;	20;	Lk		
	12	Leit.k.	Ramoskiene Aukse	213; 218;	25;	Lk		
	3	13	Anglu	Kruopienu Ausra	204;	25;	A	1;
		14	Anglu	Bosulajeva Tatjana	205;	20;	A	
		15	Anglu	Gontiene Ausra	206;	24;	A	
		16	Anglu	Januskeviciute Irena	218;	25;	A	
		17	Anglu	Januskienu Birute	219;	24;	A	
18		Vokieciu	Bieliauskienu Aldona	207; 219;	16;	V	4;5;	
19		Vokieciu	Kanapienu Zita	207; 202;	17;	V	1;5;	
20		Rusu	Mazintiene Birute	208;	27;	R		
21		Rusu	Smitaite Ona	221;	14;	R	1;5;	
22		Rusu	Zuravliova Nina	221; 208	26;	R		
4		23	Prancuzu	Bielunienu Palma	209; 221;	16;	Pr	3;4;
	24	Prancuzu	Rudminienu Migle	213; 209;	18;	Pr	1;	
	25	Istorija	Abraitenu Danguole	210;	10;	Ist		
	26	Istorija	Jonykienu Odeta	211;	11;	Ist		
	27	Istorija	Guobienu Jurgita	221; 211;	23;	Ist		
	28	Istorija	Kavaliauskaite Regina	221; 210;	14;	Ist	1;5;	
	29	Geografija	Butkute Karina	109;	30;	Geo		
	6	30	Matematika	Bukauskienu Rimante	102;	26;	Mat	
31		Matematika	Jankauskienu Zivile	101;	22;	Mat		
8	32	Matematika	Keniausiene Ada	115;	16;	Mat	4;5;	
	33	Matematika	Medisaukienu Regina	113;	22;	Mat		
	34	Matematika; Informatika;	Osipaviciute Ruta	115; #214;	18;5;	Mat;Inf;		
	35	Matematika	Senulyte Sigita	113;	22;	Mat		
	36	Matematika; Informatika;	Veikutyte Ramune	101; #214;	21;5;	Mat;Inf;		
	37	Matematika; Informatika;	Velickienu Laima	101; #214;	4;5;	Mat;Inf;	1;	
	38	Biologija	Balsevicienu Irma	106;	18;	Biol		
	39	Biologija	Kanciauskas Vidmantas	107;	24;	Biol		
	40	Biologija	Karosevicienu Ina	107;	9;	Biol	1;5;	
	41	Geografija	Tarasevicienu Rima	108;	30;	Geo		
	9	42	Fizika	Adomenas Algimantas	103;	21;	Fiz	
43		Fizika	Kavaliauskas Ovidijus	103;	24;	Fiz		
10	44	Chemija	Atstopaite Nomeda	104;	28;	Chem		
	45	Chemija	Vickackienu Lina	104;	24;	Chem		
11	46	Muzika	Blinstubienu Dalia	105;	20;	Mu		
	47	Muzika	Vaitkienu Vilija	105;	20;	Mu	5;	
	48	Sokis	Aliavienu Daiva	110;	30;	Sok		
	49	Sokis	Dilienu Vida	203;	20;	Sok		
	50	Teatras	Alkienu Vaida	222;	20;	Tea	4;5;	
	51	Teatras	Juskevicienu Valda	222;	30;	Tea		
	52	Daile	Dalienu Sigita	113;	30;	Da		
	53	Daile	Kerienu Zita	115;	30;	Da		
12	54	Daile	Stanionis Vytas	113; 209;	30;	Da		
	55	Kuno kultura	Aliubavicius Rimgaudas	112;	20;	k.k.		
	56	Kuno kultura	Bundonis Arunas	112;	10;	k.k.		
	57	Kuno kultura	Gaina Giedrius	114;	26;	k.k.		
	58	Kuno kultura	Kasiulynaite Aldona	112; 114;	12;	k.k.	3;	
13	59	Lotynu	Leka Nika	203;	8;	lot	1;2;	
	60	Ispanu	Sinkevicienu Ausra	202;203;	10;	isp		
	61	Geopolitika	Martinkevicius Algis	203;	10;	gp		
	62	Braizyba	Ramonienu Salvinija	116; 115;	20;	br	1;	
	63	PB	Dainauskas Dainius	203;	6;	pb	1;2;	
	64	Ekonomika	Sinkieniu Laura	203;	27;	ek		
	65	PComputer	Ciuksaite Giedre	208; 211;	14;	pc		

Darbo langas "3a":

Eil.	Pavarde Vardas	Dor. ugd.			Leit.k.	1 užs.kalba				2 užs.kalba				Ist	Geo	Mat	Inf	Biol	Fiz	Chem	Menai				KK	Pasirenkami						Viso			
		Ps	T	E		A	V	R	Pr	A	V	R	Pr								Mu	Sok	Tea	Da		lot	isp	gp	br	pb	ek		pc		
1	3a_Pavarde_1 Vardas_1		1		6	4				2			4	2	3	1	2	1	1		1			1				2						31	
2	3a_Pavarde_2 Vardas_2		1		6	4				2			4	2	3	1	2	1	1		1			1							2			31	
3	3a_Pavarde_3 Vardas_3		1		6	4					2		4	2	3	1	2	1	1		1			1							2			31	
4	3a_Pavarde_4 Vardas_4		1		6	4						2	4	2	3	1	2	1	1		1			1				2						31	
5	3a_Pavarde_5 Vardas_4		1		6	4					2		4	2	3	1	2	1	1		1			1				2						31	
6	3a_Pavarde_6 Vardas_5		1		6	4						2	4	2	3	1	2	1	1		1			1							2			31	
7	3a_Pavarde_7 Vardas_6			1	6	4					2		4	2	3	1	2	1	1		1			1							2			31	
8	3a_Pavarde_8 Vardas_7			1	6	4						2	4	2	3	1	2	1	1		1			1							2			31	
9	3a_Pavarde_9 Vardas_8		1		6	4							2	4	2	3	1	2	1	1		1			1				2					31	
10	3a_Pavarde_10 Vardas_9			1	6	4						2		4	2	3	1	2	1	1		1			1						2			31	
11	3a_Pavarde_11 Vardas_10			1	6	4						2		4	2	3	1	2	1	1		1			1						2			31	
12	3a_Pavarde_12 Vardas_11			1	6	4						2		4	2	3	1	2	1	1		1			1						2			31	
13	3a_Pavarde_13 Vardas_12		1		6	4							2	4	2	3	1	2	1	1		1			1						2			31	
14	3a_Pavarde_14 Vardas_13		1		6	4						2		4	2	3	1	2	1	1		1			1						2			31	
15	3a_Pavarde_15 Vardas_14			1	6	4						2		4	2	3	1	2	1	1		1			1				2					31	
16	3a_Pavarde_16 Vardas_15			1	6	4							2	4	2	3	1	2	1	1		1			1						2			31	
17	3a_Pavarde_17 Vardas_16			1	6	4						2		4	2	3	1	2	1	1		1			1				2					31	
18	3a_Pavarde_18 Vardas_17			1	6	4						2		4	2	3	1	2	1	1		1			1				2					31	
19	3a_Pavarde_19 Vardas_18		1		6	4							2	4	2	3	1	2	1	1		1			1						2			31	
20	3a_Pavarde_20 Vardas_19		1		6	4						2		4	2	3	1	2	1	1		1			1						2			31	
21	3a_Pavarde_21 Vardas_20			1	6	4						2		4	2	3	1	2	1	1		1			1					2				31	
22	3a_Pavarde_22 Vardas_21		1		6	4							2	4	2	3	1	2	1	1		1			1						2			31	
23	3a_Pavarde_23 Vardas_22			1	6	4							2	4	2	3	1	2	1	1		1			1							2			31
24	3a_Pavarde_24 Vardas_23		1		6	4							2	4	2	3	1	2	1	1		1			1							2			31
25	3a_Pavarde_25 Vardas_24			1	6	4						2		4	2	3	1	2	1	1		1			1						2				31

Tokia lentelė pildoma visoms mokinių klasėms.

MOKYKLOS PROFILIUOTŲ KLASIŲ TVARKARAŠČIŲ SUDARYMO IR OPTIMIZAVIMO SISTEMA

Kristina Bespalova

Magistrantas,

Kauno technologijos universitetas,

kristina.bespalova@stud.ktu.lt

Anotacija. Straipsnyje pristatomi "School schedule optimization" programinė sistema ir jos patobulinimai. Šiuo metu mokinių mokymosi krūvis labai didelis. Mokiniais ir mokytojams labai svarbu turėti patogų ir mokyklos tikslus atitinkantį tvarkaraštį. Tvarkaraštis turi tenkinti pedagoginius reikalavimus. Sudėtingiems mokymo dalykams turi būti skirtas darbingiausias mokinių dienos laikas. „School schedule optimization“ programinės sistemos panaudojimas galėtų padėti mokykloms geriau paskirstyti mokinių ir mokytojų darbo krūvį.

Reikšminiai žodžiai: profiliuotos mokyklos tvarkaraštis, euristiniai parametrai, euristiniai optimizavimo metodai.

Įvadas

Mokyklos tvarkaraščio sudarymo problemos nagrinėjamos ne vieną dešimtmetį. Šių problemų charakteristikos labai priklauso nuo šalies, švietimo sistemos ir mokymo įstaigos kuriai tvarkaraštis reikalingas, todėl kiekvienas sukurtas problemos sprendimo modelis turi ribotą panaudojimą. Tyrimai šioje srityje gana aktyvūs, nes vyksta nuolatinės reformos švietimo sistemose visame pasaulyje, o tai kuria naujas problemas, kurioms reikia ieškoti sprendimų.

Tvarkaraščio sudarymo uždavinys priskiriamas NP sudėtingumo uždavinių klasei (Cooper et al. 1996). Norint rasti optimalų sprendinį reiktu atlikti pilna dvinarį perrinkimą, o tai sunku įgyvendinti, nes šis perrinkimas reikalauja didelių skaičiavimo ir laiko sąnaudų. Tvarkaraščiams sudaryti taikomi apytiksliai euristiniai metodai (Sakalauskas et al. 2006). Naudojant euristikas peržiūrima tik tam tikra leistinų sprendinių dalis. Iš jų išrenkamas geriausias sprendinys. Naudojamos euristikos: "MonteKarlo" (Bilgin et al. 2007), modeliujamojo atkaitinimo (angl. *Simulated Annealing*) (Abramson 1991), Bajeso (Mockus 2000), paieškos su draudimais (Costa 1994), genetiniai (Erben 1996), deriniai įvairių metodų ir kt.

Lietuvoje bendrojo lavinimo mokyklos yra profiliuotos. Bendrasis ugdymas orientuotas į vaiką, jo poreikius ir gebėjimus. Profiliuotose mokyklose sudaromos realios galimybės specializuotis, skirti daugiau laiko gilintis į dalykines sritis, atitinkančias poreikius ir interesus. Profiliuotose mokyklose vienas iš

sudėtingesnių uždavinių yra lankstaus pamokų tvarkaraščio sudarymas su mobiliomis moksleivių grupėmis.

Nėra tokių tvarkaraščių, kurie atitiktų visus apribojimus ir asmeninius pageidavimus. Tvarkaraštis turi tenkinti švietimo ir mokslo ministerijos patvirtintus valstybinių ugdymo planų reikalavimus bei ugdymo organizavimo mokykloje ypatumus ir galimybes.

Mokyklos tvarkaraščių kokybės problemos atsiranda dėl:

- didelės mokinių pasirenkamų dalykų įvairovės;
- mokytojų stokos ar dirbančių nepagrindinėje darbovietėje paslaugų;
- ugdymo proceso organizavimo nesrautiniu būdu;
- sudėtingų dalykų pamokų nukėlimo į pamokų tvarkaraščių pabaigą (Žvirdauskas et al. 2006).

Mokyklos tvarkaraščius sudarinėjantis žmogus sudarinėdamas tvarkaraštį nori naudotis patogia ir kuo mažiau darbo reikalaujančia programine sistema. Tvarkaraščiai turi būti sudaromi automatiškai (Burke et al. 2002). Mokymo įstaigoms reikalingas optimalus, kokybiškas tvarkaraštis. Mokinių ir mokytojų darbo patogumas priklauso nuo tvarkaraščio pagal kurį jie dirba.

"School schedule optimization" sistema skirta Lietuvos bendrojo lavinimo profiliuotų mokyklų tvarkaraščių sudarymui. Siekiant patenkinti kuo daugiau vartotojo poreikių, ši sistema nuolat tobulinama. Vis įvedama naujų subjektyvių parametrų, kurių pagalba vartotojas gali įtakoti tvarkaraščio sudarymą, kad šis labiau atitiktų mokyklos keliamus reikalavimus. Šiame

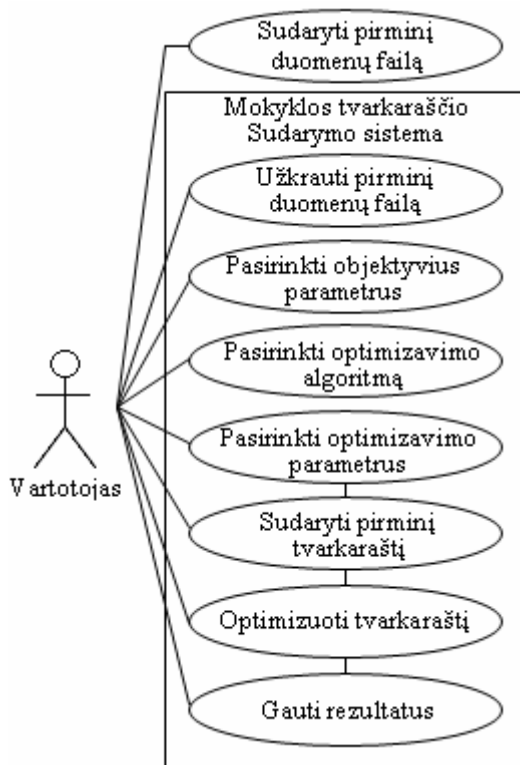
straipsnyje apžvelgiama "School schedule optimization" sistema ir jos patobulinimai.

"School schedule optimization" programinė sistema

"School schedule optimization" programinė sistema skirta profiliuotų mokyklų tvarkaraščių sudarymui. Programinė sistema sudaro:

- Mokyklos tvarkaraštį;
- Individualius mokinių tvarkaraščius;
- Individualius mokytojų tvarkaraščius;
- Individualius kabinetų tvarkaraščius;
- Grupių tvarkaraščius.
- Visų tvarkaraščių archyvą, kurį vartotojas gali parsisiųsti į savo kompiuterį.

Sistemos funkcionalumą patogiai aprašėme naudojantis UML (angl. *Unified Modeling Language*) panaudos atvejų diagramą (1 pav.).



1 Pav. Panaudos atvejų diagrama

"School schedule optimization" programinėje sistemoje atskiriami objektyvūs ir subjektyvūs parametrai. Tvarkaraštis optimizuojamas atsižvelgiant į abiejų rūšių parametrus. Taip vartotojas gali įtakoti tvarkaraščio sudarymą, kad šis atitiktų mokyklos

keliamus reikalavimus. Sistemoje realizuoti optimizavimo metodai:

- "MonteKarlo" (Bilgin et al. 2007);
- Modeliuojamojo atkaitinimo (Abramson 1991);
- Bajeso (Mockus 2000).

Darbo metu vartotojas gali pasirinkti norimą metodą, ar ieškant geriausio rezultato bandyti visus.

"School schedule optimization" programinės sistemos vartotojo sąsaja realizuota nacionaline lietuvių kalba. Kilus klausimams yra išsami pagalba vartotojui lietuvių kalba. Vartotojas dirbdamas su sistema dalyvauja tik sudarant pirminį duomenų failą, nustatant programoje esančius parametrus (ribojimus) ir pasirenkant optimizavimo metodą. Sistema atsižvelgdama į nustatytus apribojimus automatiškai sudaro pirminį tvarkaraštį, atlieka optimizavimą, kiekvienam tvarkaraščio variantui apskaičiuoja baudos taškus ir išrinkusi geriausią pateikia rezultatus vartotojui.

Rezultatai išvedami į ekraną ir į „zip“ archyvą, kurį vartotojas gali parsisiųsti į savo kompiuterį. Jame atskiruose kataloguose yra saugomi mokinių, mokytojų, kabinetų tvarkaraščiai (mokytojų – „Initial teachers“ bei „Optimized teachers“, mokinių – „Initial students“ bei „Optimized students“, kabinetų – „Initial rooms“ bei „Optimized rooms“) ir papildomos naudingos suvestinės.

Programinė sistema įdiegta *Apache-TomCat-6.0.18* servletų konteineryje, kuris palaiko Java servletus (angl. *Java Servlet*) ir JSP (angl. *JavaServer Pages*). Kai vartotojas internetinėje svetainėje paspaudžia ant programos paleidimo nuorodos, automatiškai paleidžiama programa darbui. Tai leidžia vartotojui naudotis programine sistema be jokių instaliavimo procedūrų.

"School schedule optimization" programinės sistemos darbui reikalingas pirminis duomenų failas, kuriame saugoma mokyklos tvarkaraščio sudarymui reikalinga informacija.

Pradinių duomenų įvedimas

Pirminis duomenų failas paruošiamas Microsoft Office programa „MS Excel“ ir išsaugomas „XML“ formatu. Šiame faile laikantis nurodytos formos suvedama mokyklos tvarkaraščio sudarymui reikalinga informacija. Nesilaikant formos duomenys gali būti klaidingai suprasti ar nenuskaityti į sistemą.

Naujoje programos versijoje padaryti pakeitimai pirminių duomenų failo sudaryme. Pakeistas pirmas "MS Excel" knygos lapas skirtas mokykloje mokomiems dalykams aprašyti. Dabar prie kiekvieno dalyko

papildomai nurodomas prioritetas ir maksimalus mokinių skaičius. Prioritetas – tai visų mokykloje mokomų dalykų suskirstymas pagal jų sudėtingumą. Sudėtingiausi dalykai turi būti mokomi pirmoje dienos pusėje ir turi indeksą 5, o lengviausi indeksą 1. Maksimalus mokinių skaičius – tai maksimalus mokinių skaičius su kuriuo mokytojas gali dirbti pamokoje. Dalykų prioritetus ir mokinių skaičių vartotojas gali nusistatyti individualiai, pagal savo mokyklos keliamus reikalavimus.

Pakeitimų atsirado ir antrame programos "MS Excel" knygos lape kuris skirtas mokykloje dirbantiems mokytojams aprašyti. Šiame lape be vardo, pavardės, dėstomo dalyko ir per savaitę dėstomų valandų skaičiaus atsirado kabinetų numeriai kuriuose mokytojas gali dirbti. Taip tvarkaraštyje galime specifiniams dalykams priskirti specialiai tam įrengtus kabinetus.

Pirminio tvarkaraščio sudarymas

"School schedule optimization" programinė sistema nuskaičiusi pradinius duomenis į masyvus sudaro pirminį tvarkaraštį. Šio etapo pagrindinis tikslas sutalpinti visus reikiamus dalykus į bendrą tvarkaraštį. Jau nuo pirmų žingsnių stengiamasi jį sudaryti kuo geresnį. Pamokos į tvarkaraštį sudedamos pagal dalykam suteiktus prioritetus. Pirmiausiai tvarkaraštyje sudedami aukščiausią prioritetą turintys dalykai, o po to eilės tvarka pagal sudėtingumą išdėstomi kiti. Sudėtingiausi dalykai sudedami pirmoje dienos pusėje, darbingiausiu mokinio dienos laiku. Pamokos į tvarkaraštį dedamos taip, kad dvi dienas iš eilės tuo pačiu metu nebūtų mokomas tas pats dalykas. Jau šio etapo metu įvertinami fiziniai ribojimai kurie privalo būti patenkinti:

- Mokinsys ar mokytojas vienu metu gali dalyvauti tik vienoje pamokoje.
- Kabinete vienu metu gali vykti tik viena pamoka.

Pirminio tvarkaraščio sudarymo metu pasikeitė mobilių grupių sudarymas. Ankstesnėje versijoje buvo formuojama grupė neviršijant 30-ies moksleivių skaičių. Šioje versijoje formuojant grupes neviršijamas pirminiame duomenų faile prie dalyko nurodytas maksimalus mokinių skaičius su kuriuo mokytojas gali dirbti pamokoje. Šį mokinių skaičių vartotojas gali nusistatyti individualiai, pagal savo mokyklos reikalavimus.

Tvarkaraščio optimizavimas

Optimizavimas orientuotas į mokytojų ir mokinių langų uždarymą, bei pamokų perkėlimą iš mokytojo

pageidaujamo laisvadienio. Tvarkaraščio langu vadiname vienos pamokos trukmės laisvo laiko tarpas tarp pamokų, neskaitant pietų pertraukai skirto laiko. Ankstesnėje "School schedule optimization" programinės sistemos versijoje optimizavimo metu buvo uždarinėjami rasti langai mokytojams ir perkeliamos pamokos iš mokytojo pageidaujamo "laisvadienio". Naujoje versijoje realizuotas ir mokinių langų uždarymo algoritmas.

Programinės sistemos darbo metu, pradėjus tvarkaraščio optimizavimo procesą pirmiausiai atliekamas mokinių langų uždarymas. Toliau tvarkaraštis optimizuojamas pagal vartotojo pasirinktą optimizavimo metodą, uždarinėjant mokytojų langus ar perkeliant pamokas iš mokytojų pageidaujamų laisvadienių. Baigus su mokytojais vėl atliekamas mokinių langų uždarymas.

Mokinių langų uždarymo metu surandamas mokinio langas tvarkaraštyje. Imama mokinio paskutinė dienos pamoka ir tikrinama ar ją galima perkelti į mokinio surasto lango vietą (2 pav.). Jei šios pamokos perkelti negalima, imama kita ir taip ieškoma kol surandama ką galima perkelti. Jei galimos perkelti pamokos nerandame, tvarkaraštyje paliekamas langas. Taip bandoma uždaryti visus mokyklos tvarkaraštyje esančius mokinių langus.

	Pirma.	Antra.	Trečia.	Ketvirta.	Penkta.	
1	Fiz	Lk	Fiz	Lk	Che	
2				Mat	Lk	
1	Fiz	Lk	Fiz	Lk	Che	Ist
2	Lk	Mat	Lk	Mat	Lk	T
3	Mat	Che	Mat	Ist	Ist	Da
4	Geo	Ek	Inf	T	Sok	A
5	Inf	Biol	R		A	
6	A	R	Sok	A	Da	
7	k.k.					

2 Pav. Mokinių langų uždarymas

Mokinio langą galima uždaryti tuo atveju, jei randama lango metu kitiems mokiniams vykstanti norima perkelti pamoka. Tikrinama ar pamokoje į kuria norime perkelti mokinių yra laisvų vietų, t.y. ar perkėlus nebus viršytas maksimalus mokinių skaičius su kuriuo mokytojas gali dirbti pamokoje.

Baudos taškų skaičiavimas

"School schedule optimization" programinėje sistemoje sprendžiamas daugiakriterinis optimizavimo uždavinys (Žilinskas 2000). Tam, kad daugiakriterinį uždavinį pakeisti vienkriteriu buvo įvestas "baudos taškų" skaičiavimas. Kriterijams vartotojas nustato svorius atsižvelgdamas į jų svarbą, o tai priklauso nuo mokymo

įstaigos keliamų reikalavimų tvarkaraščiui. Kuo daugiau baudos taškų skiriame kriterijui, tuo didesnis dėmesys kreipiamas į tą apribojimą. Ieškomas toks tvarkaraštis, kuris minimizuotų bendrą baudos taškų funkciją. Optimaliu tvarkaraščiu laikomas tas už kurį neradom geresnio, kurio baudos taškų skaičius yra mažiausias.

$$C = \sum_r c_r N_r, \quad (1)$$

kur C_r -bauda už normų r pažeidimą; N_r -pažeidimų kiekis; r kinta nuo 1 iki 9.

Baudos taškai apskaičiuojami įvertinant 6 parametrus:

- per dieną vykstančių pamokų skaičiaus padidinimas;
- langus mokiniams tarp pamokų;
- langus mokytojams tarp pamokų;
- mokytojų laisvadienių pažeidimą;
- mokytojo išdirbtų valandų viršijimą;
- dalykų prioritetų pažeidimą.

Naujoje programinės sistemos versijoje įvertinamas dalykų prioritetų pažeidimas. Baudų taškai dalykų prioritetų pažeidimui – nurodo, kiek baudos taškų bus skiriama, jei aukšto prioriteto dalykas dėstomas antroje dienos pusėje. Taip vartotojui palikta galimybė įtakoti optimizavimo procesą, leidžiant ar apribojant sudėtingų dalykų pamokų nukėlimą į tvarkaraščio pabaigą.

Papildomų parametrų įvedimas padidina programos skaičiavimus, tačiau juos įvertinus gaunamas tvarkaraštis kuris geriau tenkina vartotojo reikalavimus.

Programos darbo rezultatai

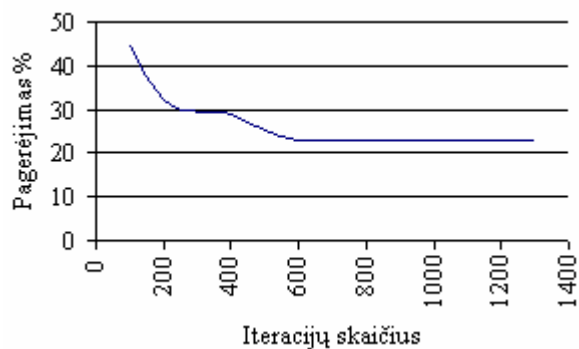
Optimalius tvarkaraščių sprendinius galime rasti tik mažos apimties uždaviniams. Realūs mokyklos tvarkaraščiai yra didelės apimties. Net ir neoptimalaus sprendinio radimas pareikalauja didelių skaičiavimo ir laiko sąnaudų.

Testuojant sukurtą programinę sistemą, naudojant testinius pirminių duomenų failais kuriuose suvesti duomenys atitinka vidutinio dydžio mokyklos duomenų kiekius. Gaunami tyrimo rezultatai labai priklauso nuo vartotojo užduotų parametrų ir nuo mokinių pasirinkimų įvairovės.

Programinė sistema darbo metu atlieka skaičiavimus kurių kiekis priklauso nuo vartotojo užduotų parametrų. Didesnis optimizavimo iteracijų skaičius garantuoja geresnį optimizavimo rezultatą, tačiau prailgina šio proceso trukmę.

Papildomų ribojimų įvedimas pailgino programos darbą, tačiau gautas tvarkaraštis geriau tenkina mokyklos reikalavimus.

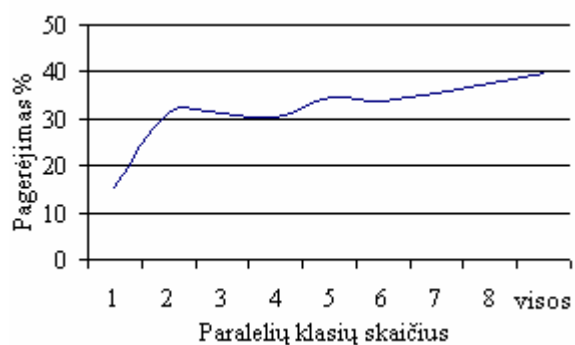
Įvedus mokinių langų uždarymo algoritmą gauti geresni programos darbo rezultatai. Grafike (3 pav.) pateikiami programinės sistemos darbo rezultatai prie tų pačių parametrų, keičiant tik iteracijų skaičių.



3 Pav. Tvarkaraščio pagerėjimas įvedus mokinių langų uždarymą

Didinant iteracijų skaičių tvarkaraščio pagerėjimas nusistovi ties 20%.

Profiliuotos mokyklos tvarkaraščiui didelę įtaką turi paralelių klasių skaičius. Kuo mokykla daugiau turi paralelių klasių, kurių mokinius galima jungti į mobilias grupes, tuo gaunami geresni tvarkaraščių sudarymo rezultatai. Esant nors trimis paralelių klasių komplektams, jau net pirminis tvarkaraštis gaunamas gana kompaktiškas. Paralelių klasių skaičius įtakoja ir optimizavimo procesą (4 Pav.).



4 Pav. Tvarkaraščio pagerėjimo priklausomybė nuo paralelių klasių skaičiaus

Blogiausi optimizavimo rezultatai gaunami kai turima po vieną paralelių klasių komplektą. Pasiekus tam tikrą paralelių klasių skaičių stebimas tik nežymus pagerėjimas.

Išvados

1. "School schedule optimization" programinė sistema gali padėti Lietuvos profiliuotoms bendrojo lavinimo mokykloms sudaryti pamokų tvarkaraštį.

2. Ši sistema nereikalauja didelių vartotojo pastangų ar mokėjimo dirbti kompiuteriu. Gana lengva naudotis, tereikia perprasti pirminio duomenų failo sudarymą.

3. Tvarkaraštis sudaromas automatiškai. Vartotojui nereikia galvoti kur kokia pamoka galėtų vykti, sistema pati kontroliuoja, kad pamokos nesidubliuotų.

4. Į sistemą įvedus naujus subjektyvius parametrus padidinami programos skaičiavimai, tačiau tik jų dėka sudaromi tvarkaraščiai, kurie tenkina konkrečios mokyklos keliamus reikalavimus.

5. Programinė sistema vis dar kuriama ieškant geresnių sprendimų tvarkaraščių sudaryme. Nuolatinės reformos švietimo sistemoje, skatina ieškoti sprendimų, kad programa būtų kuo lankstesnė pokyčiams.

6. Tikimės ateityje "School schedule optimization" programinė sistema bus pritaikyta realių mokyklų tvarkaraščių sudaryme ir padės mokykloms išspręsti šį sudėtingą planavimo uždavinį.

Padėkos

Dėkoju prof. habil. dr. Jonui Mockui už uždavinio formulavimą ir pagalbą jį sprendžiant.

Literatūra

- Abramson, D. 1991. *Constructing school timetables using simulated annealing: sequential and parallel algorithms*, Management Science, vol. 37, no. 1, 98-113.
- Bilgin, B.; Ozcan, E.; Korkmaz, E. E. 2007. *An Experimental Study on Hyper-Heuristics and Exam Scheduling*, PATAT2006, Springer-Verlag, selected papers, LNCS 3867, 394-412.
- Burke, E.K.; Petrovic, S. 2002. *Recent research directions in automated timetabling*, European Journal of Operational Research 140, 266-280.
- Cooper, T.B.; Kingston, J.H. 1996. *The complexity of timetable construction problems*. In: Proceedings of the 1st International Conference on Practice and Theory of Automated Timetabling (PATAT 1995), LNCS 1153, Springer-Verlag, 283-295.
- Costa, D. 1994. *A tabu search algorithm for computing an operational timetable*, European Journal of Operational Research, vol. 76, 98-110.
- Erben, W.; Keppler, J. 1996. *A genetic algorithm solving a weekly coursetimetabling problem*, in: E. Burke, P. Ross (Eds.), The Practice and Theory of Automated Timetabling: Selected Papers from the First International Conference, Lecture Notes in Computer Science, Springer, Berlin, vol. 1153, 198-211.

Mockus, J. 2000. *A Set of Examples of Global and Discrete Optimization: Application of Bayesian Heuristic Approach*. Kluwer Academic Publishers. ISBN 0-7923-6359-0.

Sakalauskas, L.; Felinskas, G. 2006. *Tvarkaraščių su ribotais ištekliais sudarymo euristiniai algoritmai, jų tyrimas bei taikymai*, Informacijos mokslai, 38 tomas, 90-103. ISSN 1392-1487.

Žilinskas, A. 2000. *Matematinis programavimas*. VDU. Kaunas.

Žvirdauskas, D.; Adaškevičienė, V.; Tarnauskas, K.; Žvirdauskienė, R. [interatyvus]. 2006. *Priemonių, skirtų vyresniųjų klasių mokinių mokymosi krūvio mažinimui, veiksmingumas*. Lietuvos Respublikos švietimo ir mokslo ministerijos tyrimas, [žiūrėta 2008 m. spalio 15 d.] Prieiga per internetą: <http://www.smm.lt/svietimo_bukle/docs/tyrimai/Kruviu_ataskaite_0702.pdf>

SCHOOL PROFILED CLASS SCHEDULING AND OPTIMIZATION SYSTEM

K. Bespalova

Summary

The article presents the "School schedule optimization" software system and its improvements. Currently, pupils academically load is very high. It is very important for a pupil to have convenient and appropriate school timetable that reflects schools aims and goals. It is a very difficult task to make such schedule without gaps. Schedule has to meet pedagogical requirements. Most productive time of the day has to be filled with more complex lessons. With the help of "School schedule optimization" software system the workload of teachers and pupils can be distribute

SUBJEKTYVIŲ PARAMETRŲ ĮVERTINIMAS VIDURINĖS MOKYKLOS TVARKARAŠTYJE

Kristina Bespalova

Kauno technologijos universitetas, K. Donelaičio g. 73, Kaunas, rastine@ktu.lt

Apžvalga. Straipsnyje pristatomi "School schedule optimization" programinės sistemos patobulinimai. Šiuo metu mokinių mokymosi krūvis labai didelis. Mokiniais ir mokytojams labai svarbu turėti patogų ir mokyklos tikslus atitinkantį tvarkaraštį. Tvarkaraštis turi tenkinti pedagoginius reikalavimus. Sudėtingiems mokymo dalykams turi būti skirtas darbingiausias mokinių dienos laikas. „School schedule optimization“ programinės sistemos panaudojimas galėtų padėti mokykloms geriau paskirstyti mokinių ir mokytojų darbo krūvį.

Annotation. The article presents the "School schedule optimization" software system improvements. Currently, pupils academically load is very high. It is very important for a pupil to have convenient and appropriate school timetable that reflects schools aims and goals. It is a very difficult task to make such schedule without gaps. Schedule has to meet pedagogical requirements. Most productive time of the day has to be filled with more complex lessons. With the help of "School schedule optimization" software system the workload of teachers and pupils can be distributed evenly through the day.

Raktiniai žodžiai: profiliuotos mokyklos tvarkaraštis, euristiniai parametrai, euristiniai optimizavimo metodai.

1 Įvadas

Mokyklos tvarkaraščio sudarymo problemos nagrinėjamos ne vieną dešimtmetį. Šių problemų charakteristikos labai priklauso nuo šalies, švietimo sistemos ir mokyklos kuriai tvarkaraštis reikalingas. Tyrimai šioje srityje gana aktyvūs, nes vyksta nuolatinės reformos švietimo sistemose visame pasaulyje, o tai kuria naujas problemas, kurioms reikia ieškoti sprendimų.

Tvarkaraščio sudarymo uždavinys priskiriamas NP sudėtingumo uždavinių klasei [1]. Norint rasti optimalų sprendinį reiktu atlikti pilna dvinarį perrinkimą, o tai sunku įgyvendinti, nes šis perrinkimas reikalauja didelių skaičiavimo ir laiko sąnaudų. Tvarkaraščiams sudaryti taikomi apytiksliai euristiniai metodai [2]. Naudojant euristikas peržiūrima tik tam tikra leistinų sprendinių dalis. Iš jų išrenkamas geriausias sprendinys. Naudojamos euristikos: "MonteKarlo" [3], modeliuojamojo atkaitinimo (Simulatet Annealing) [4], Bajeso [5], paieškos su draudimais [6], genetiniai [7], deriniai įvairių metodų ir kt.

Lietuvoje bendrojo lavinimo mokyklos yra profiliuotos. Jose sudaromos realios galimybės specializuotis, skirti daugiau laiko gilintis į dalykines sritis, atitinkančias poreikius ir interesus. Profiliuotose mokyklose vienas iš sudėtingesnių uždavinių yra lankstaus pamokų tvarkaraščio sudarymas su mobiliomis moksleivių grupėmis. Tiek mokinių, tiek mokytojų darbo patogumas priklauso nuo tvarkaraščio pagal kurį jie dirba. Tvarkaraštis turi tenkinti švietimo ir mokslo ministerijos patvirtintus valstybinių ugdymo planų reikalavimus bei ugdymo organizavimo mokykloje ypatumus ir galimybes.

Mokyklos tvarkaraščių kokybės problemos atsiranda dėl: didelės mokinių pasirenkamų dalykų įvairovės; mokytojų stokos ar dirbančių nepagrindinėje darbovietėje paslaugų; ugdymo proceso organizavimo nesrautiniu būdu; sudėtingų dalykų pamokų nukėlimo į pamokų tvarkaraščių pabaigą [8].

Mokyklos tvarkaraščius sudarinėjantis žmogus sudarinėdamas tvarkaraštį nori naudotis patogia ir kuo mažiau darbo reikalaujančia programine sistema. Tvarkaraščiai turi būti sudaromi automatiškai [9]. Mokymo įstaigoms reikalingas optimalus, kokybiškas tvarkaraštis.

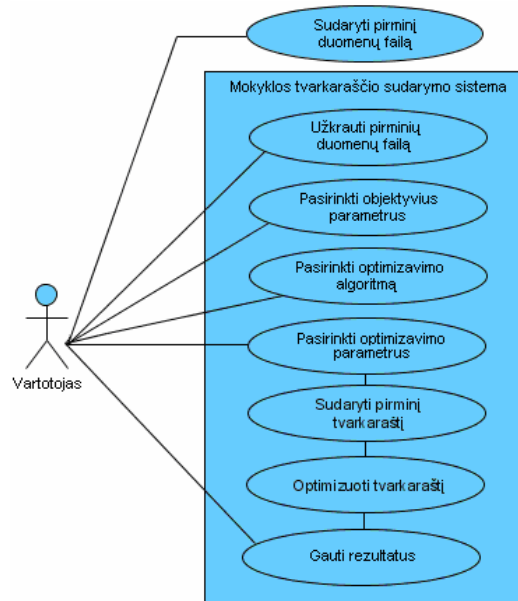
"School schedule optimization" sistema skirta profiliuotų mokyklų tvarkaraščių sudarymui. Siekiant patenkinti kuo daugiau vartotojo poreikių, ši sistema nuolatos tobulinama. Vis įvedama naujų subjektyvių parametrų, kurių pagalba vartotojas gali įtakoti tvarkaraščio sudarymą, kad šis labiau atitiktų mokyklos keliamus reikalavimus. Šiame straipsnyje apžvelgiami sistemos patobulinimai, kurie susiję su naujų parametrų įvedimu.

2 "School schedule optimization" programinė sistema

"School schedule optimization" programinė sistema skirta profiliuotų mokyklų tvarkaraščių sudarymui. Programinė sistema sudaro:

- Mokyklos tvarkaraštį;
- Individualius mokinių tvarkaraščius;
- Individualius mokytojų tvarkaraščius;
- Individualius kabinetų tvarkaraščius;
- Grupių tvarkaraščius.

- Visų tvarkaraščių archyvą, kurį vartotojas gali parsisiųsti į savo kompiuterį.



Paveikslas 1. Panaudojimo atvejų diagrama

"School schedule optimization" programinėje sistemoje yra atskiriami objektyvūs ir subjektyvūs parametrai. Tvarkaraštis optimizuojamas atsižvelgiant į abiejų rūšių parametrus. Taip vartotojas gali įtakoti tvarkaraščio sudarymą, kad šis atitiktų mokyklos keliamus reikalavimus. Šioje sistemoje naudojami "MonteKarlo" [3], modeliuojamojo atkaitinimo (Simulated Annealing)[4] ir Bajeso [5] optimizavimo metodai. Darbo metu vartotojas gali pasirinkti norimą metodą, ar ieškant geriausio rezultato išbandyti visus.

"School schedule optimization" programinės sistemos vartotojo sąsaja realizuota nacionaline lietuvių kalba. Kilus klausimams yra išsami pagalba vartotojui lietuvių kalba. Vartotojas dirbdamas su sistema dalyvauja tik sudarant pirminį duomenų failą ir nustatant programoje esančius parametrus (ribojimus) ir metodą. Sistema atsižvelgdama į nustatytus apribojimus automatiškai sudaro pirminį tvarkaraštį, atlieka optimizavimą, kiekvienam tvarkaraščio variantui apskaičiuoja baudos taškus ir išrinkusi geriausią pateikia rezultatus vartotojui.

3 Pakeitimai programoje

3.1. Pirminių duomenų įvedimas

Naujoje programos versijoje padaryti pakeitimai pirminių duomenų failo sudaryme. Pakeistas pirmas programos "MS Excel" knygos lapas skirtas mokykloje mokomiems dalykams aprašyti. Prie kiekvieno dalyko papildomai nurodomas prioritetas ir maksimalus mokinių skaičius. Prioritetai – tai visų mokykloje mokomų dalykų suskirstymas pagal jų sudėtingumą. Sudėtingiausi dalykai turi būti mokomi pirmoje dienos pusėje ir turi indeksą 5, o lengviausi indeksą 1. Maksimalus mokinių skaičius – tai maksimalus mokinių skaičius su kuriuo mokytojas gali dirbti pamokoje. Dalykų prioritetus ir mokinių skaičių vartotojas gali nusistatyti individualiai, pagal savo mokyklos keliamus reikalavimus.

Pakeitimų atsirado ir antrame programos "MS Excel" knygos lape kuris skirtas mokykloje dirbantiems mokytojams aprašyti. Šiame lape be vardo, pavardės, dėstomo dalyko ir per savaitę dėstomų valandų skaičiaus atsirado kabinetų numeriai kuriuose mokytojas gali dirbti. Taip tvarkaraštyje galime specifiniams dalykams priskirti specialiai tam įrengtus kabinetus.

3.2. Pirminio tvarkaraščio sudarymas

"School schedule optimization" naujoje versijoje patobulintas pirminio tvarkaraščio sudarymas. Jau nuo pirmų žingsnių stengiamasi jį sudaryti kuo geresnį. Pamokos į tvarkaraštį sudedamos pagal dalykam suteiktus prioritetus. Pirmiausiai tvarkaraštyje sudedami aukščiausią prioritetą turintys dalykai, o po to eilės tvarka pagal sudėtingumą išdėstomi kiti. Sudėtingiausi dalykai sudedami pirmoje dienos pusėje, darbingiausiu mokinio dienos laiku. Pamokos į tvarkaraštį dedamos taip, kad dvi dienas iš eilės tuo pačiu metu nebūtų mokomas tas pats dalykas.

Pirminio tvarkaraščio sudarymo metu pasikeitė mobilių grupių sudarymas. Ankstesnėje versijoje buvo formuojama grupė neviršijant 30-ies moksleivių skaičių. Šioje versijoje formuojant grupes neviršijamas pirminiame duomenų faile prie dalyko nurodytas maksimalus mokinių skaičius su kuriuo mokytojas gali dirbti pamokoje. Šį mokinių skaičių vartotojas gali nusistatyti individualiai, pagal savo mokyklos reikalavimus.

3.3. Mokinių langų uždarinėjimas

Tvarkaraščio langų vadiname vienos pamokos trukmės laisvo laiko tarpas tarp pamokų, neskaitant pietų pertraukai skirto laiko. Ankstesnėje "School schedule optimization" programinės sistemos versijoje optimizavimo metu buvo uždarinėjami rasti langai mokytojams ir perkeliamos pamokos iš mokytojo pageidaujamo "laisvadienio". Naujoje versijoje realizuotas ir mokinių langų uždarymo algoritmas.

Programinės sistemos darbo metu, pradėjus tvarkaraščio optimizavimo procesą pirmiausiai atliekamas mokinių langų uždarymas. Toliau tvarkaraštis optimizuojamas pagal vartotojo pasirinktą optimizavimo metodą, uždarinėjant mokytojų langus ar perkeltiant pamokas iš mokytojų pageidaujimų laisvadienių. Baigus su mokytojais vėl atliekamas mokinių langų uždarymas.

	Pirmadienis	Antradienis	Trečiadienis	Ketvirtadienis	Penktadienis
1	Fiz	Lk	Fiz	Lk	Chem
2	Lk	Mat	Lk	Mat	Lk
3	Mat	Chem	Mat	Ist	Ist
	Pirmadienis	Antradienis	Trečiadienis	Ketvirtadienis	Penktadienis
1	Fiz	Lk	Fiz	Lk	Chem
2	Lk	Mat	Lk	Mat	Lk
3	Mat	Chem	Mat	Ist	Ist
4	Geo	ek	Inf	T	Sok
5	Inf	Biol	R	A	A
6	A	R	Sok	A	Da
7	k.k.				

Paveikslas 2. Mokinių langų uždarymas

Mokinių langų uždarymo metu surandamas mokinio langas tvarkaraštyje. Imama mokinio paskutinė dienos pamoka ir tikrinama ar ją galima perkelti į mokinio surasto lango vietą. Jei šios pamokos perkelti negalima, imama kita ir taip ieškoma kol surandama ką galima perkelti. Jei galimos perkelti pamokos nerandame, tvarkaraštyje paliekamas langas. Taip bandoma uždaryti visus mokyklos tvarkaraštyje esančius mokinių langus.

Mokinio langą galima uždaryti tuo atveju, jei randama lango metu kitiems mokiniams vykstanti norima perkelti pamoka. Tikrinama ar pamokoje į kuria norime perkelti mokinį yra laisvų vietų, t.y. ar perkėlus nebūs viršytas maksimalus mokinių skaičius su kuriuo mokytojas gali dirbti pamokoje.

Uždarinėjant mokinių langus atsiranda papildomas mokinių išmaišymas grupėse. Jei mokyklai grupių išmaišymas nepriimtinas, vartotojas turi galimybę didinti baudos taškų skaičių skiriamą už grupių sumaišymą. Po šio nustatymo uždarinėjami tik tie mokinių langai kuriuos uždarius neatsiranda papildomo mokinių sumaišymo grupėse.

3.4. Baudos taškų skaičiavimas

"School schedule optimization" programinėje sistemoje sprendžiamas daugiakriterinis optimizavimo uždavinys [10]. Tam, kad daugiakriterinį uždavinį pakeisti vienkriteriu buvo įvestas "baudos taškų" skaičiavimas. Kriterijams vartotojas nustato svorius atsižvelgdamas į jų svarbą, o tai priklauso nuo mokymo įstaigos keliamų reikalavimų tvarkaraščiui. Optimaliu tvarkaraščiu laikomas tas už kurį neradom geresnio, kuriame baudos taškų skaičius yra mažiausias.

$$C = \sum_{\tau} c_{\tau} N_{\tau} \quad (1)$$

kur C_{τ} -bauda už normų r pažeidimą; N_r -pažeidimų kiekis; r kinta nuo 1 iki 9.

Baudos taškai apskaičiuojami įvertinant 7 parametrus (kriterijus):

- per dieną vykstančių pamokų skaičiaus padidinimas;
- langus mokiniams tarp pamokų;
- langus mokytojams tarp pamokų;
- mokytojų laisvadienių pažeidimą;
- mokytojo išdirbtų valandų viršijimą;
- pedagoginių didaktikų pažeidimą;
- grupių sumaišymą.

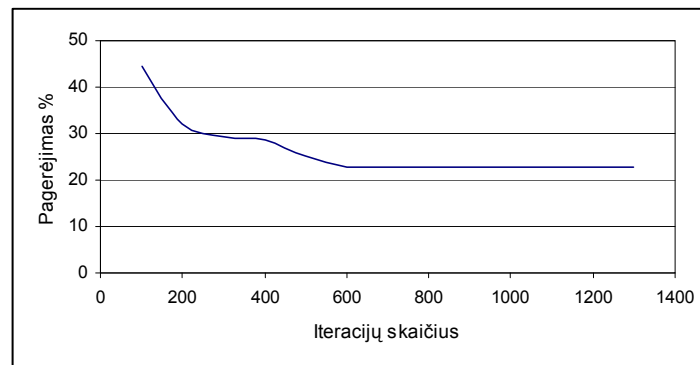
Naujoje versijoje įvertinami papildomi parametrai tokie kaip pedagoginių didaktikų pažeidimas ir grupių sumaišymas. Baudų taškai pedagoginių didaktikų pažeidimui – nurodo, kiek baudos taškų bus skiriama, jei aukšto prioriteto dalykas dėstomas antroje dienos pusėje. Baudų taškai grupių sumaišymui – nurodo, kiek baudos taškų bus skiriama jei mokinys mokosi dalyko ne su savo grupe.

Papildomų parametru įvedimas padidina programos skaičiavimus, tačiau juos įvertinus gaunamas geriau vartotojo reikalavimus tenkinantis tvarkaraštis.

4 Programos darbo rezultatai

Optimalius tvarkaraščių sprendinius galime rasti tik mažos apimties uždaviniams. Realūs mokyklos tvarkaraščiai yra didelės apimties. Net ir neoptimalaus sprendinio radimas pareikalauja didelių skaičiavimo ir laiko sąnaudų. Testuojant sukurtą programinę sistemą, buvo pasinaudota testiniais pirminių duomenų failais kuriuose suvesti duomenys atitinka vidutinio dydžio mokyklos duomenų kiekius.

Programinės sistemos darbo metu atliekami skaičiavimai kurių kiekis priklauso nuo vartotojo užduotų parametrų. Didėnis optimizavimo iteracijų skaičius garantuoja geresnį optimizavimo rezultatą, tačiau prailgina šio proceso trukmę. Papildomų ribojimų įvedimas ilgina programos darbą, tačiau gautas tvarkaraštis geriau tenkina mokyklos reikalavimus. Įvedus mokinių langų uždarymo algoritmą gauti geresni programos darbo rezultatai. Grafike pateikiami rezultatai prie tų pačių parametrų, keičiant tik iteracijų skaičių.



Paveikslas 3. Tvarkaraščio pagerėjimas įvedus mokinių langų uždarymą

Didinant iteracijų skaičių tvarkaraščio pagerėjimas nusistovi ties 20%.

5 Išvados

"School schedule optimization" programinė sistema padeda profiliuotoms mokykloms sudaryti pamokų tvarkaraštį. Ši sistema nereikalauja didelių vartotojo pastangų ar mokėjimo dirbti kompiuteriu.

Programinė sistema vis dar kuriama ieškant geresnių sprendimų tvarkaraščio sudaryme. Nuolatinės reformos švietimo sistemoje, skatina ieškoti sprendimų, kad programa būtų kuo lankstesnė pokyčiams. Vartotojas turėdamas galimybę sistemoje keisti įvairius parametrus galėtų pasiekti norimą rezultatą. Įvedamas naujų subjektyvių parametrų padidina programos skaičiavimus, bet tik per juos sudarome tvarkaraštį, kuris tenkina konkrečios mokyklos keliamus reikalavimus.

Tikimės ateityje "School schedule optimization" programinė sistema bus pritaikyta realių mokyklų tvarkaraščių sudaryme ir padės mokykloms išspręsti šį sudėtingą uždavinį.

Literatūros sąrašas

- [1] **Cooper T.B., Kingston J.H.** The complexity of timetable construction problems. In: Proceedings of the 1st International Conference on Practice and Theory of Automated Timetabling (PATAT 1995), LNCS 1153, Springer-Verlag. 1996, pp. 283-295.
- [2] **Sakalauskas L., Felinskas G.** Tvarkaraščių su ribotais ištekliais sudarymo euristiniai algoritmai, jų tyrimas bei taikymai, Informacijos mokslai. 2006, 38 tomas, pp. 90-103.
- [3] **Bilgin B., Ozcan E., Korkmaz E. E.** An Experimental Study on Hyper-Heuristics and Exam Scheduling, PATAT2006, Springer-Verlag, selected papers, LNCS 3867. 2007, pp. 394-412.
- [4] **Abramson D.** Constructing school timetables using simulated annealing: sequential and parallel algorithms, Management Science. 1991, vol. 37, no. 1, pp. 98-113.
- [5] **Mockus J.** A Set of Examples of Global and Discrete Optimization: Application of Bayesian Heuristic Approach. Kluwer Academic Publishers. 2000.
- [6] **Costa D.** A tabu search algorithm for computing an operational timetable", European Journal of Operational Research. 1994, vol. 76, pp. 98-110.
- [7] **Erben W., Keppler J.** A genetic algorithm solving a weekly coursetimetabling problem, in: E. Burke, P. Ross (Eds.), The Practice and Theory of Automated Timetabling: Selected Papers from the First International Conference, Lecture Notes in Computer Science, Springer, Berlin, 1996, vol. 1153, pp. 198-211.
- [8] **Žvirdauskas D., Adaškevičienė V., Tarnauskas K., Žvirdauskienė R.** Priemonių, skirtų vyresniųjų klasių mokinių mokymosi krūvio mažinimui, veiksmingumas. Lietuvos Respublikos švietimo ir mokslo ministerijos tyrimas. 2006, prieiga internete <http://www.smm.lt/svietimo_bukle/docs/tyrimai/Kruviu_ataskaita_0702.pdf>
- [9] **Burke E.K., Petrovic S.** Recent research directions in automated timetabling, European Journal of Operational Research 140. 2002, pp. 266-280.
- [10] **Žilinskas A.** Matematinis programavimas. VDU. Kaunas. 2000.

Mokyklos tvarkaraščio pavyzdys:

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, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Geo, Geo, Inf, Biol	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Geo, Geo, Inf, Biol	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Geo, Geo, Inf, Biol	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Geo, Geo, Inf, Biol	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Ist, Ist, Ist, Geo, Inf, Biol, Biol, br, A, A, A
2	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Ist, Ist, Ist, Inf, Biol, Biol, br, A, A	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Ist, Ist, Ist, Inf, Biol, Biol, br, A, A, A, A, V, R	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Geo, Inf, Biol, A, A	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Ist, Ist, Ist, Geo, Geo, Biol, A, A, A, A, V, R	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Biol, Biol, A, A, A, V, R
3	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Geo, Geo, Inf, A, V, R, k.k.	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Geo, Geo, Biol, Biol, A, A, A, A	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Geo, Geo, Inf, Biol, Biol, A, A	Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Geo, Geo, Inf, Biol, Biol, A, A, A, A, A, A	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Mat, Mat, Mat, Mat, Mat, Mat, Fiz, Fiz, Chem, Chem, Ist, Ist, Ist, Geo, Geo, Inf, Biol, Biol, A, A, A, A, A, A, Pr
4	Lk, Lk, Lk, Lk, Lk, Lk, Lk, Chem, Chem, Ist, Ist, Geo, Geo, Inf, Biol, Biol, A, A, A, A, A, A, A, V, R, R, Pr, k.k., k.k.	Mat, Mat, Mat, Mat, Mat, Mat, Chem, Ist, Ist, Ist, Geo, Geo, Inf, Biol, Biol, A, A, A, A, A, A, V, V, R, R, Pr, Pr, k.k., k.k., k.k.	Lk, Lk, Lk, Lk, Mat, Mat, Ist, Ist, Ist, Ist, Geo, Geo, Inf, Biol, Biol, A, A, A, A, A, A, V, R, R, Pr, Pr, k.k., k.k., k.k.	Mat, Mat, Mat, Mat, Ist, Geo, Geo, Inf, Biol, Biol, ek, A, A, A, A, A, A, A, A, V, V, R, R, Pr, Pr, k.k., k.k., k.k., k.k., pc, Ps, T, Mu, Tea	Lk, Lk, Mat, Geo, Geo, Inf, Biol, Biol, br, A, A, A, A, A, A, A, A, A, V, R, R, Pr, Pr, k.k., k.k., T, k.k., pc, Ps, Mu, Tea, Da, Da
5	Mat, Geo, Inf, Biol, Biol, br, ek, A, A, A, A, A, A, A, V, V, R, R, R, Pr, k.k., k.k., k.k., pc, Ps, T, Mu, Sok, Tea	Ist, Biol, Biol, br, ek, A, A, A, A, A, A, A, A, A, V, V, R, R, R, Pr, Pr, k.k., k.k., k.k., pc, Ps, T, Mu, Mu, Sok, Tea, Da, Da	Ist, Biol, Biol, ek, A, A, A, A, A, A, A, A, A, V, V, R, R, R, Pr, Pr, k.k., k.k., k.k., T, T, Mu, Mu, Sok, Tea, Da, Da	Ist, Biol, Biol, A, A, A, A, A, A, A, A, A, V, V, R, R, R, Pr, k.k., k.k., k.k., pb, T, T, Mu, Mu, Sok, Tea, Da, Da, Da	Ist, ek, A, A, A, A, A, A, A, A, V, V, R, R, R, Pr, Pr, k.k., k.k., T, k.k., pc, Ps, Mu, Sok, Tea, Da, Da, Da
6	Ist, Pr, A, A, A, A, A, A, V, V, R, R, R, Pr, k.k., k.k., k.k., pc, T, T, Mu, Mu, Sok, Tea, Da, Da, Da	ek, A, A, A, A, A, A, A, A, A, V, V, R, R, R, Pr, k.k., k.k., k.k., T, T, E, Mu, Mu, Sok, Tea, Da, Da, Da	ek, A, A, A, A, A, A, A, A, A, V, V, R, R, R, Pr, k.k., k.k., k.k., T, T, E, Mu, Mu, Sok, Tea, Da, Da, Da, k.k.	A, A, A, A, A, A, A, A, A, V, V, R, k.k., k.k., k.k., pb, pc, T, T, E, Mu, Mu, Sok, Tea, Da, Da, Da, Da	A, A, A, A, A, A, A, A, V, k.k., k.k., k.k., pb, T, T, E, Mu, Tea, Da, Da, Da, Da
7	A, A, A, A, A, V, R, k.k., k.k., T, T, E, Mu, Mu, Sok, Tea, Tea, Da, Da, Da, Da	A, A, V, k.k., k.k., Mu, T, T, E, Mu, Tea, Tea, Da, Da, Da	V, k.k., T, T, E, E, Tea, Tea, Da, Da, pb, k.k.	T, T, E, E, Tea, Da, pb, A, k.k., Mu	Tea, Pr, E, k.k.

Bauros taškai sudarius pradinį tvarkaraštį: 34 460, baudos taškai po tvarkaraščio optimizavimo: 27 810

[Parsisiųsti ZIP archyve supakuotus rezultatus](#)

Mokinio tvarkaraščio pavyzdys:

Mokinio 9a pavarde 14 Vardas 14 tvarkaraštis

Optimalus mokyklos tvarkaraštis

	Pirmadienis	Antradienis	Trečiadienis	Ketvitadienis	Penktadienis
1	Mat	Lk	Mat	Lk	Mat
2	Lk	Mat	A	Ist	Lk
3	Inf	R	Lk	A	Ist
4	A	A	R	k.k.	Biol
5	Biol	pc	T	Mu	k.k.
6	Da	k.k.		Da	

Mokytojo tvarkaraščio pavyzdys:

Mokytojo Balseviciene Irma tvarkaraštis

Optimalus mokyklos tvarkaraštis

	Pirmadienis	Antradienis	Trečiadienis	Ketvitadienis	Penktadienis
1	Biol	Biol	Biol	Biol	Biol
2	Biol	Biol	Biol	Biol	Biol
3		Biol	Biol	Biol	Biol
4	Biol	Biol	Biol	Biol	

Kabineto tvarkaraščio pavyzdys:

Kabineto 115 tvarkaraštis

Optimalus mokyklos tvarkaraštis

	Pirmadienis	Antradienis	Trečiadienis	Ketvitadienis	Penktadienis
1	Mat	Mat	Mat	Mat	Mat
2	Mat	Mat	Mat	Mat	Mat
3	Mat	Mat	Mat	Mat	
4		Mat		Mat	Da
5		Da	Da	Da	Da
6	Da	Da	Da	Da	Da
7	Da		Da		

Fragmentas iš tvarkaraščių archyvo failo - tvarkaraštis Dalykas-Klasės-Mokinių skaičius:

1-dienis / (Monday)					
(Nr.: 1) ->	Lk - 5-a - 25	Lk - 5-b - 25	Lk - 5-c - 25	Lk - 6-a - 25	Lk - 6-b - 26
(Nr.: 2) ->	Lk - 6-b - 26	Lk - 6-c - 25	Lk - 7-b - 25	Lk - 8-a - 30	Lk - 8-b - 30
(Nr.: 3) ->	Lk - 9-b - 25	Lk - 9-c - 25	Lk - 10-a - 25	Lk - 10-b - 29	Lk - 10-c - 25
(Nr.: 4) ->	Lk - 10-c - 25	Lk - 11-a - 29	Lk - 11-b - 30	Lk - 11-c - 29	Lk - 12-a - 29
(Nr.: 5) ->	Mat - 12-c - 29	Geo - 8-c;8-a;8-b - 30	Inf - 6-b - 26	Biol - 9-a - 25	Biol - 9-b - 25
(Nr.: 6) ->	Ist - 12-c;12-b - 30	Pr - 12-a;12-b - 20	A - 6-b - 20	A - 6-b - 6	A - 8-a;8-b - 20
(Nr.: 7) ->	A - 11-b - 19	A - 11-c - 20	A - 11-c - 9	A - 12-b - 20	A - 12-b - 5
2-dienis / (Tuesday)					
(Nr.: 1) ->	Lk - 7-b - 25	Lk - 8-a - 30	Lk - 8-b - 30	Lk - 8-c - 25	Lk - 9-a - 25
(Nr.: 2) ->	Lk - 5-a - 25	Lk - 5-b - 25	Lk - 5-c - 25	Lk - 6-a - 25	Lk - 6-b - 26
(Nr.: 3) ->	Lk - 11-a - 29	Lk - 11-b - 30	Lk - 11-c - 29	Lk - 12-a - 29	Lk - 12-b - 25
(Nr.: 4) ->	Mat - 11-a - 29	Mat - 11-b - 30	Mat - 11-c - 29	Mat - 12-a - 29	Mat - 12-b - 25
(Nr.: 5) ->	Ist - 12-c - 29	Biol - 6-b - 16	Biol - 9-c - 25	br - 6-b - 10	ek - 11-c;11-a - 30
(Nr.: 6) ->	ek - 12-c;12-a - 25	A - 6-b - 16	A - 9-c - 10	A - 10-a - 20	A - 10-a;10-b - 18
(Nr.: 7) ->	A - 11-c - 9	A - 12-b - 10	V - 11-c - 6	k.k. - 11-b - 25	k.k. - 11-c - 14
3-dienis / (Wednesday)					
(Nr.: 1) ->	Lk - 5-a - 25	Lk - 6-b - 26	Lk - 6-c - 25	Lk - 7-a - 29	Lk - 7-b - 25
(Nr.: 2) ->	Lk - 8-b - 30	Lk - 8-c - 25	Lk - 9-a - 25	Lk - 9-b - 25	Lk - 9-c - 25
(Nr.: 3) ->	Lk - 10-a - 25	Lk - 10-b - 29	Lk - 10-c - 25	Lk - 11-a - 29	Lk - 11-b - 30
(Nr.: 4) ->	Lk - 11-c - 29	Lk - 12-a - 29	Lk - 12-b - 25	Lk - 12-c - 29	Mat - 11-a - 29
(Nr.: 5) ->	Ist - 12-c - 29	Biol - 6-b - 16	Biol - 10-b;10-c - 30	ek - 12-a;12-b - 30	A - 6-b;6-c - 20
(Nr.: 6) ->	ek - 12-c - 12	A - 10-a - 20	A - 10-a;10-b - 14	A - 10-c - 20	A - 10-c - 5
(Nr.: 7) ->	V - 12-a - 16	k.k. - 11-c - 10	T - 11-b - 25	T - 11-c - 10	E - 11-c - 9
4-dienis / (Thursday)					
(Nr.: 1) ->	Lk - 8-b - 30	Lk - 8-c - 25	Lk - 9-a - 25	Lk - 9-b - 25	Lk - 9-c - 25
(Nr.: 2) ->	Lk - 5-a - 25	Lk - 6-b - 26	Lk - 6-c - 25	Lk - 7-a - 29	Lk - 7-b - 25
(Nr.: 3) ->	Lk - 12-b - 25	Lk - 12-c - 29	Mat - 10-a - 25	Mat - 10-b - 29	Mat - 10-c - 25
(Nr.: 4) ->	Mat - 11-b - 30	Mat - 11-c - 29	Mat - 12-a - 29	Mat - 12-c - 29	Ist - 12-b - 23
(Nr.: 5) ->	Ist - 12-c - 29	Biol - 10-c - 24	Biol - 12-b - 25	A - 6-b - 20	A - 6-b - 6
(Nr.: 6) ->	A - 10-a;10-c - 20	A - 10-c;10-b - 17	A - 11-b - 20	A - 11-b;11-c - 20	A - 11-c - 19
(Nr.: 7) ->	T - 11-c - 7	T - 12-b - 9	E - 11-c - 3	E - 12-a - 1	Tea - 11-a - 21
5-dienis / (Friday)					
(Nr.: 1) ->	Lk - 5-a - 25	Lk - 6-b - 26	Lk - 6-c - 25	Lk - 7-a - 29	Lk - 7-b - 25
(Nr.: 2) ->	Lk - 8-c - 25	Lk - 9-a - 25	Lk - 9-b - 25	Lk - 9-c - 25	Lk - 10-a - 25
(Nr.: 3) ->	Lk - 10-c - 25	Lk - 11-a - 29	Lk - 11-b - 30	Lk - 11-c - 29	Lk - 12-a - 29
(Nr.: 4) ->	Lk - 12-b - 25	Lk - 12-c - 29	Mat - 12-a - 29	Geo - 6-b - 26	Geo - 8-b - 28
(Nr.: 5) ->	Ist - 12-c;12-b - 30	ek - 12-b;12-a - 16	A - 6-b - 20	A - 6-b - 6	A - 7-b - 14
(Nr.: 6) ->	A - 10-c - 20	A - 10-c - 5	A - 11-b - 20	A - 11-b;11-c - 20	A - 11-c - 19
(Nr.: 7) ->	Tea - 11-c - 29	Pr - 8-a;8-b - 20	E - 11-a - 8	k.k. - 12-b - 15	