KAUNO TECHNOLOGIJOS UNIVERSITETAS
MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS

Ieva Jankauskaitė

Užsiėmimų tvarkaraščio sudarymas taikant genetinį algoritmą

Vadovas
Doc. dr. Bronė Narkevičienė

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS

MATEMATIKOS IR GAMTOS MOKSLŲ. FAKULTETAS

Užsiėmimų tvarkaraščio sudarymas naudojant genetinį algoritmą

Baigiamasis magistro projektas

Taikomoji matematika, 612B92002

Vadovas

(parašas) Doc. dr. B. Narkevičienė

(data)

Recenzentas

(parašas) dr. B. Narijauskaitė

(data)

Projektą atliko

(parašas) Ieva Jankauskaitė

(data)

KAUNAS, 2015

Jankauskaitė Ieva Timetable scheduling using genetic algorithm: Master‘s work in applied mathematics / supervisor Doc. Dr Bronė Narkevičienė; Faculty of Mathematics and Natural Sciences, Kaunas University of Technology. – Kaunas, 2015. – 39psl.

SUMMARY

Schedule is an essential tool for the time management. It helps to lay out formed tasks in some particular order. Timetable formation is known problem in many fields. It is also acute problem in educational scope. It depends on the timetable whether educator’s and student’s natural resources will be used effectively.

This work is about University timetable formation and the methods used to generate it. We will analyze how genetic algorithm can find the best option of choosing allowed timetable. We create a program based on genetic algorithm in finding acceptable schedule. This is one of the most common timetable formation methods used in various fields. Also, we will try to get a proper schedule with the schedule creation software “aSc Tvarkaraščiai” and compare results.

Turinys

Įžanga 9

1. Literatūros apžvalga 11

1.1 Tvarkaraščių sudarymo problema 11

1.2 Tvarkaraščių sudarymo metodai 13

1.3 Genetinis algoritmas 15

1.4 Tvarkaraščių sudarymo programinė įranga 19

2. Tyrimo metodai 21

2.1 Pradiniai duomenys 21

2.2 Genetinio algoritmo praktinis taikymas 22

2.2.1 Chromosomos pateikimas programoje 22

2.2.2 Kryžminimo vaizdavimas programoje 23

2.2.3 Mutacijos apibrėžimas programoje 23

2.2.4 Algoritmo realizavimas programoje 24

2.3 Tinkamumo funkcijos apibrėžimas programoje 24

3. Tyrimų rezultatai ir jų aptarimas 26

3.1 Pradiniai parametrai ir duomenys 26

3.2 Tyrimo rezultatai 27

3.2.1 Genetinio algoritmo rezultatai 27

3.2.2 Programine įranga „aSc Tvarkaraščiai“ gautas rezultatas 32

4. Išvados 38

Literatūros sąrašas..... 39

Paveikslėlių sąrašas

Pav. 1 Tvarkaraščių sudarymo uždavinių klasifikacija 11

Pav. 2 Tvarkaraščio diagrama 13

Pav. 3 Genetinio algoritmo schema 16

Pav. 4 Kryžminimas per vieną tašką 17

Pav. 5 Kryžminimas per du taškus 17

Pav. 6 Kryžminimas ir Mutacija 18

Pav. 7 Binarinis chromosomos kodavimas 19

Pav. 8 Chromosomos pateikimas 22

Pav. 9 Kryžminimo vaizdavimas 23

Pav. 10 aSc Tvarkaraščiai programa gautas rezultatas 33

Lentelių sąrašas

Lentelė 1 Generacijų skaičius keičiant mutacijos sk. 29

Lentelė 2 Generacijų skaičius parenkant parametrus 30

Lentelė 3 Tvarkaraštis gautas taikant genetinį algoritmą 30

Lentelė 4 Tvarkaraštis gautas genetiniu algoritmu su papildomomis sąlygomis 31

Lentelė 5 Perkonstruotas tvarkaraštis 33

Lentelė 6 Tvarkaraštis sudėliotas taikant papildomas sąlygas..... 35

Grafikų sąrašas

Figure 1 Generacijų priklausomybė nuo darbo valandų sk. 27

Figure 2 Generacijų priklausomybė nuo darbo valandų sk. 27

Figure 3 Tinkamumo rodiklio augimas 5gr. 28

Figure 4 Tinkamumo rodiklio augimas 10gr. 28

Įžanga

Grafikas arba tvarkaraštis yra pagrindinis laiko valdymo įrankis, sudarytas iš viso laiko, kuris tinka (yra galimas) užduotims, renginiams ar planuojamiems veiksmams atlikti ir yra išdėliotas tam tikra tvarka. Akivaizdžius tvarkaraščių pavyzdžius dažniausiai pastebi žmonės, kurie nuolat naudojasi viešuoju transportu. Tačiau su įvairiomis darbų arba užduočių sudarymo bei kalendorinio planavimo problemomis susiduriama inžinerinio projektavimo, vadybos metu. Optimalių tvarkaraščių radimas yra labai aktuali problema įvairiose veiklos srityse: logistikoje, vadyboje, versle, projektavime, gamyboje, ekonomikoje ir kitose srityse.

Tvarkaraštis yra labai aktualus švietimo ir ugdymo įstaigose. Nuo jo priklauso, ar pedagogų, dėstytojų ir moksleivių, studentų žmogiškieji ištekliai bus panaudoti efektyviai. Daugelyje įstaigų už tvarkaraščių sudarymą yra atsakingas žmogus, nes programinė įranga yra mokama, o kai kurios programos yra labai sudėtingos ir reikalauja papildomų žinių ir gebėjimų. Leistiną tvarkaraščio sprendinį, kuris tenkina visus reikalavimus, rasti klasikiniais metodais nesunku, tačiau nebūtinai sudarytas tvarkaraštis bus optimalus. Dažniausiai tvarkaraščių optimizavimo problemoms spręsti taikomi euristiniai metodai.

Šiame darbe spręsime universiteto tvarkaraščio sudarymo ir optimizavimo problemą. Optimaliam tvarkaraščiui sudaryti naudosime genetinį algoritmą ir tinkamumo funkciją. Taip pat mėginsime sudaryti optimalų tvarkaraštį naudojant tam skirtą programą ”aSc Tvarkaraščiai“, kuri specialiai skirta ugdymo įstaigų tvarkaraščių sudarymui. Panagrinėsime šią programą detaliau.

Tikslas iš uždaviniai:

Darbo tikslas – sudaryti studijų tvarkaraštį taikant genetinį algoritmą ir programinę įrangą „aSc Tvarkaraščiai“. Išanalizuoti ir palyginti gautus rezultatus.

Darbo uždaviniai:

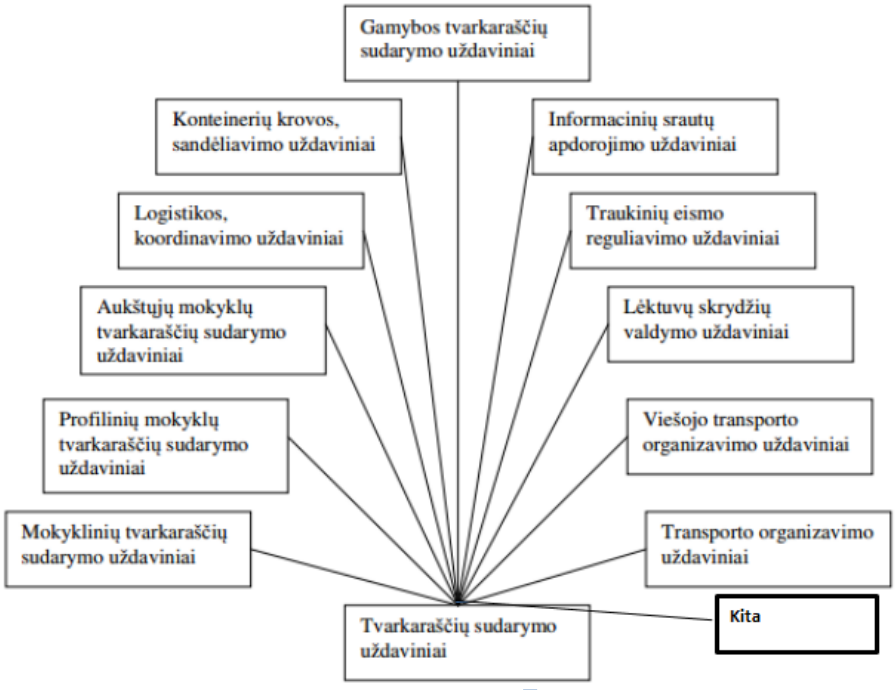
- Išnagrinėti tvarkaraščio sudarymui skirtus metodus;
- Išanalizuoti programinę įrangą skirtą tvarkaraščiams sudaryti;
- Išanalizuoti genetinio algoritmo veikimą;
- Sukurti programą tvarkaraščiams kurti, naudojant genetinį algoritmą;

- Gauti tinkamus tvarkaraščius, naudojant sukurtą programą ir programinę įrangą „aSc Tvarkaraščiai“;
- Išanalizuoti ir palyginti gautus rezultatus.

1. Literatūros apžvalga

1.1 Tvarkaraščių sudarymo problema

Tvarkaraščių sudarymo uždavinys svarbus daugelyje veiklos sričių. Mokymo įstaigos taip pat kiekvienais metais stengiasi sukurti kuo geresnius tvarkaraščius, kurie tenkintų tiek studentų (mokinių), tiek dėstytojų (pedagogų) poreikius. Ypač sudėtingi yra aukštųjų ir profilinių mokyklų tvarkaraščių sudarymo uždaviniai. Svarbią įtaką kokybiško tvarkaraščio sudarymui turi specializuotų kabinetų skaičius ir tos pačios disciplinos dėstytojų, studentų (mokinių) ir grupių (klasių) skaičius. Jei dėstytojų (pedagogų), kabinetų ir disciplinų turime daug, o studentų (mokinių) ne, tai tokiu atveju tvarkaraštį sudaryti gana paprasta.



Pav. 1 Tvarkaraščių sudarymo uždavinių klasifikacija

Tvarkaraščių formavimo problema pradėta nagrinėti visai neseniai. Viskas prasidėjo dėl didėjančių problemų norint sudaryti tinkamą tvarkaraštį. Taip pat kito mokymo metodai. Kaip teigia Appleby, J.S., Blake, D.V., Newman, E.A. (1960m.) straipsnyje „Techniques for producing school timetables on a computer and their application to other scheduling problems“ išspausdintame *The Computer Journal 3* žurnale, tipinėje to laikmečio mokykloje su 30 klasių ir 35-iomis pamokomis per savaitę, galimų kombinacijų skaičius gaunamas labai didelis – apie 10^{500}. Tačiau tinkamų tvarkaraščių, kurie tenkintų numatytas sąlygas (išvardintos žemiau), atsirastų nedidelė dalis.

Kiekviena grupė lanko numatytą užsiėmimų skaičių. Kiekvienas kabinetas turi skirtingą vietų skaičių. Įmanomas grafikas yra toks, kuriame visos paskaitos buvo priskirtos laiko tarpsniui ir kabinetams taip, kad būtų tenkinamos numatytos sąlygos:

- Tvarkaraštyje užsiėmimų neturi būti daugiau nei nustatytas užsiėmimų skaičius;
- Kiekvienai disciplinai skirtas valandų kiekis pagal nustatytą ugdymo planą;
- Neplanuoti dviejų iš eilės tos pačios disciplinos užsiėmimų;
- Kai kurie užsiėmimai turi eiti iš eilės;
- Tvarkaraštyje turime vengti laisvų pamokų;
- Kabinete vienu metu vyksta tik vienas užsiėmimas;
- Dėstytojas (pedagogas) ar ugdytinis negali būti keliuose užsiėmimuose vienu metu;
- Dėstytojas (pedagogas) dėsto tik įgytos disciplinos užsiėmimus;
- Kabinetas yra pakankamai didelis, kad tilptų visi ugdytiniai ir sudarytos tinkamos darbo sąlygos;

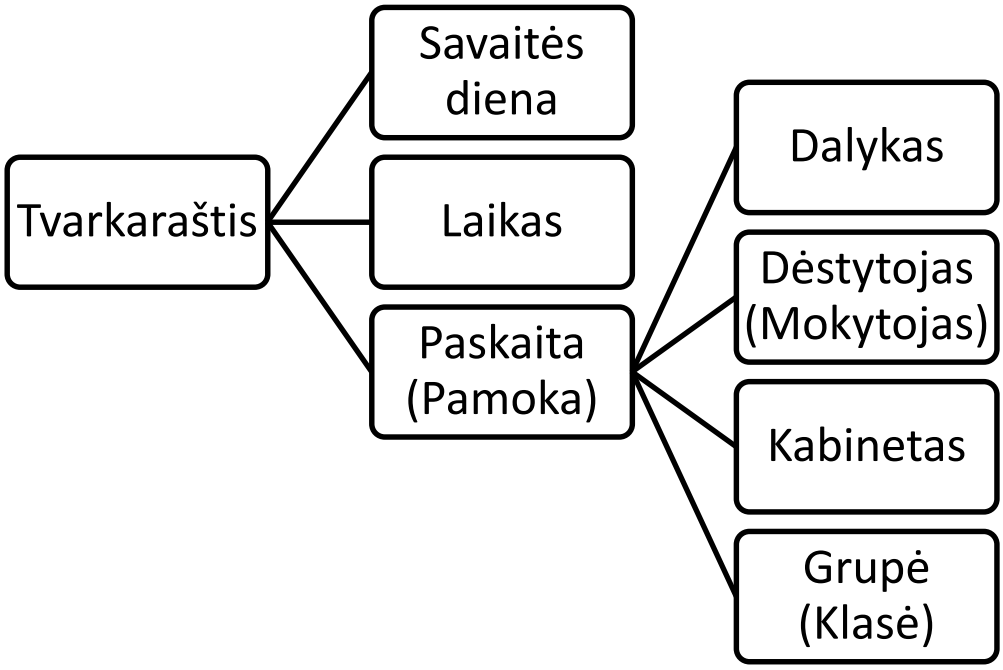
Šie reikalavimai priklauso konkrečiai nuo kiekvienos ugdymo įstaigos. Vienur šių reikalavimų gali būti daugiau, kitur - mažiau.

Mokymo įstaigų tvarkaraščių sudarymo uždavinius galima suklasifikuoti į 5 grupes (Felinskas, 2007):

- Mokymo įstaigų tvarkaraščiai: reikia paskirsti tam tikros kvalifikacijos dėstytojus (mokytojus) numatomoms disciplinoms;
- Dėstytojų (pedagogų) tvarkaraščiai: paskirstyti užsiėmimus, kai mažiausias vienetas tvarkaraštyje yra ugdytinių grupė;
- Užsiėmimų tvarkaraščiai: paskirstyti užsiėmimus, kai mažiausias vienetas tvarkaraštyje yra atskiras ugdytinis;
- Egzaminų (sesijų) tvarkaraščiai: paskirstyti egzaminus ugdytiniams taip, kad jie neturėtų dviejų egzaminų tuo pačiu metu;
- Kabinetų skirstymas: susiejus grupės ir dėstytojo (pedagogo) poras, priskirti kabinetams.

Tvarkaraščiai ugdymo įstaigose yra kuriami savaitei. Uždavinys yra suplanuoti studentų bei dėstytojų bendrą paskaitų tam tikrame kabinete fiksuotą momentų skaičių taip, kad nė vienas iš

dalyvaujančiųjų nebūtų keliose vietose tuo pačiu metu. Tvarkaraštyje nuo pirmadienio iki penktadienio (darbo dienomis) užsiėmimai yra paskirstomi valandomis, laikantis nuostatų. Diagrama atrodytų taip, kaip pavaizduota žemiau:



Pav. 2 Tvarkaraščio diagrama

1.2 Tvarkaraščių sudarymo metodai

Kai ieškome problemos sprendimo, mes dažniausiai siekiame rasti geriausią variantą tarp kitų. Visi galimi variantai sudaro paieškos erdvę. Vienas taškas erdvėje atitinka vieną galimą sprendimą, kuris gali būti įvertintas tinkamu problemai išspręsti, t.y. mes gauname funkciją, kurios argumentas yra sprendimas, o reikšmė nusako tinkamumą. Kartais tokia erdvė yra apibrėžiama gana lengvai, tačiau dažniausiai mes žinome tik kelis jos taškus. Genetinio algoritmo proceso metu yra generuojami kiti taškai ir šitaip ieškoma sprendimo.

Problema yra ta, kad tokia erdvė gali būti labai sudėtinga ir galima nežinoti, kaip ieškoti sprendimo ir nuo ko pradėti. Yra daug metodų, kaip surasti tinkamą reikšmę, bet ne visi jie veda prie geriausio sprendimo. Kai kurie iš šių metodų yra:

- Modeliuojamojo atkaitinimo (ang. Simulated Annealing) algoritmas ;
- Genetiniai algoritmai (ang. Genetic Algorithms);

- Paieška su draudimais (ang. Tabu Search);
- Skruzdžių kolonijos (ang. Ant colony systems);
- Memetiniai algoritmai (ang. Memetic algorithms);
- Išbarstytoji, išsklaidytoji paieška (ang. Scatter search);
- Spiečių algoritmai (ang. Swarm algorithms);
- Dirbtinių neuroninių tinklų metodai (ang. Neural networks);
  Ir kiti.

Kartais, norint gauti geriausią sprendinį, reikia derinti kelis metodus tarpusavyje.

Atkaitinimo metodas. Šis metodas yra vienas iš labiausiai paplitusių globaliojo optimizavimo metodų. Taikomas spręsti tiek tolydiems, tiek diskretiems optimizavimo uždaviniams. Šis būdas sukurtas remiantis analogijomis iš statistinės mechanikos. Temperatūra yra svarbiausias parametras. Ji mažėja su kiekvienu atkaitinimo proceso imitavimo žingsniu. Parametras reguliuojamas įvedant temperatūros atnaujinimo (aušinimo) funkciją. Algoritme įvedamas nuo temperatūros priklausantis specialus tikimybinis patvirtinimo kriterijus. Juo remiantis kiekviename žingsnyje yra priimamas surastas geresnis sprendinys nei esantis, bet gali būti priimtas ir blogesnis sprendinys, tačiau su nedidele tikimybe. Tokia savybė leidžia išeiti iš lokaliųjų optimumo taškų ieškant globaliojo optimumo (Felinskas, 2007).

Bajeso metodas (ang. Bayes). Naudojantis šiuo metodu galime optimizuoti parametrus, kad būtų didesnis metodų efektyvumas. Šio metodo pagrindas – matematinė naudingumo tikimybė ir racionalus variantas, išrenktas pagal vidutinės naudos maksimumą. Bajeso metodas dažniausiai naudojamas atkaitinimo metodo efektyvumui didinti, optimizuojant jo parametrus (Bespalova, 2009).

Išbarstytoji paieška. Tai evoliucinis metodas, naudojamas kombinatorinio ir netiesinio optimizavimo uždaviniams spręsti. Išbarstytoji paieška operuoja sprendinių rinkiniu, vadinamu atraminiu. Gautus rinkinio sprendinius kombinuojant, gaunami nauji sprendiniai. Kaip ir genetiniuose algoritmuose taip ir čia yra naudojamos panašios procedūros atrenkant sprendinius į atraminį rinkinį. Tačiau šiuo metodu pakanka operuoti su nedideliu sprendinių rinkiniu. Išbarstytosios paieškos yra realizuojamos keliais etapais. Paieška pradedama generuojant pradinio sprendinio rinkinį. Naudojant esamą atraminį sprendinių rinkinio poaibį yra generuojami nauji sprendiniai. Iš jų atrenkami geriausi, kurie yra priskiriami rinkiniui. Išbarstytoji paieška stabdoma atlikus numatytų iteracijų skaičių (Bespalova, 2009).

Monte – Karlo metodas. Šiuo metodu problemos sprendimui naudojamas atsitiktinių skaičių generatorius. Optimizuojant kiekviename žingsnyje iš visų galimų sprendinių aibės yra parenkamas atsitiktinis sprendinys ir lyginamas su esamu. Naudojantis šiuo metodu pusiausvyrai rasti naudojamas didelis iteracijų skaičius. Šis metodas tinkamiausias, kai iteracijų vykdymo laikas nėra ilgas. Kai iteracijų daug ir jos užtrunka pakankamai ilgai, ypač sudėtingėjant problemai - laikas labai prailgėja (Bespalova, 2009).

„Įkainojimo" funkcija. Norint įvertinti objektyvius parametrus buvo panaudota įkainojimo funkcija, kitaip dar vadinama baudos taškų sumavimo funkcija (angl. Cost function). Ši funkcija skaičiuoja neatitikimų skaičių gautame tvarkaraštyje. Funkcija nusakoma gauto sprendinio kokybė. Geriausias tvarkaraštis laikomas tas, kurio baudos taškų suma yra lygi nuliui arba yra jam artima (L. Pupeikienė, 2009).

1.3 Genetinis algoritmas

Šiuo būdu sudarydami tvarkaraštį naudosime genetinį algoritmą, kadangi jis yra labiausiai tinkamas ir dažniausiai naudojamas praktikoje. Daugybei sričių, kurioms reikalingas tvarkaraščių ir grafikų sudarymas, programinės įrangos paketai yra paremti genetiniais algoritmais. Genetinis algoritmas ypač naudojamas, kai turime didelę populiaciją.

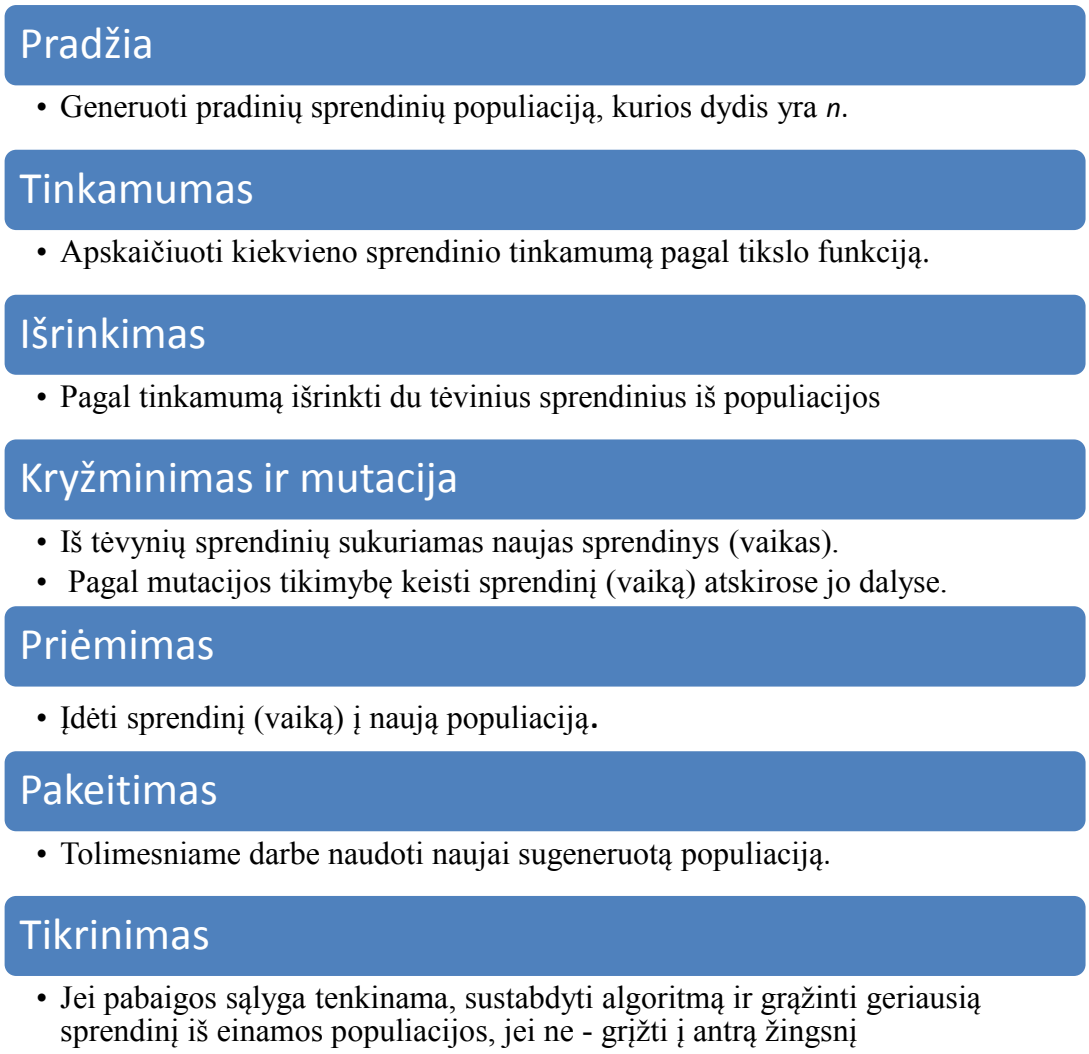
1.3.1 Biologinis pagrindas

Gyvi organizmai yra sudaryti iš ląstelių. Kiekviena ląstelė turi chromosomų rinkinį. Chromosomos yra DNR eilutės, viso organizmo modelis. Genus sudaro DNR blokai, kurie turi savo vietas chromosomoje (lokusus) ir užkoduoja tam tikrus proteinus. Kiekvienas genas nusako tam tikrą požymį, pvz. plaukų spalvą. Visi galimi požymiai (pvz. ruda, kaštoninė) yra vadinami alelėmis (ang. allele). Visa genetinė medžiaga yra vadinama genomu, o genotipu vadinamas konkretus rinkinys. Iš genotipo išsivystęs individas yra vadinamas fenotipu – fizinės ir vidinės charakteristikos [8].

Reprodukcijos metu įvyksta rekombinacija ir iš abiejų tėvų genų yra gaunama visiškai nauja chromosoma. Jos metu yra galimybė, kad DNR elementai pakis, t.y. mutuos. Tai gali atsitikti, dėl netiksliai nukopijuotų genų ar prarandama dalis informacijos, įvyksta pasislinkimai ir panašūs dalykai. Mutacija ir rekombinacija yra atsakingos už genų kitimą ir evoliuciją. Evoliucijos metu palikuonys paveldi ar įgauna tam tikras savybes [8].

1.3.2 Genetinio algoritmo schema

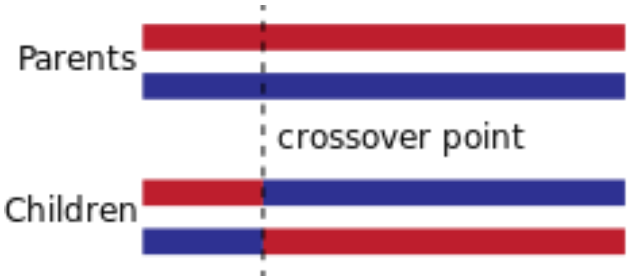
Pirmame pateiktos schemos žingsnyje (žr. Pav.3) sukuriama pradinė sprendinių aibė (populiacija). Paprasčiausias pradinių sprendinių parinkimo būdas yra jų generavimas atsitiktine tvarka. Populiacijos dydis nesikeičia visu algoritmo vykdymo metu, todėl jis yra vienas iš genetinio algoritmo parametrų. Tada algoritmas vykdo ciklą, kuris gali būti apribotas maksimaliu iteracijų skaičiumi arba vykdymo laiku. Kiekvienoje ciklo iteracijoje mėginama prieiti prie vis geresnės populiacijos. Pirmiausia yra įvertinama esama populiacija. Toliau kuriama nauja populiacija. Naują populiaciją kuriame iš senosios, pritaikius genetinius operatorius: kryžminimą ir mutaciją [8].



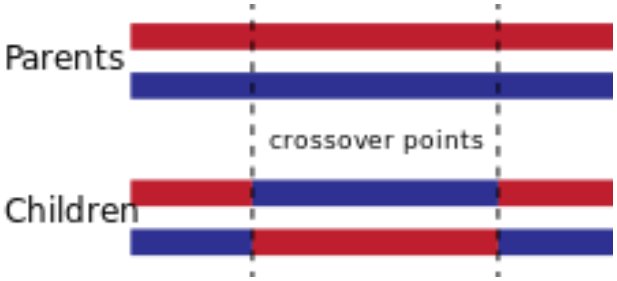
Pav. 3 Genetinio algoritmo schema

1.3.3 Kryžminimas

Kryžminimui parenkame mažiausiai du tėvus. Rekombinacija yra būdas genetiniuose algoritmuose perteikti vienos chromosomos ar kelių chromosomų sandarą į kitą kartą. Atsitiktiniu būdų parenkamas vienas arba daugiau pjūvio taškų. Tada naują sprendinį gauname paėmus tėvinių sprendinių dalis, gautas perkirtus juos per pjūvio taškus. Ji yra biologinės rekombinacijos analogas, kadangi būtent biologiniu procesu ir paremta. Kryžminimas su vienu rekombinacijos tašku: parenkamas rekombinacijos taškas tėvų duomenyse. Vėliau visi „tėvų“ (pav. parents) duomenys, buvę už taško, apkeičiami vietomis. Rezultatas – „vaikai“ (pav. Children), kurie turi dalį vieno ir kito tėvo (genetinių) duomenų (žr. Pav 4.). Kryžminimas su dviem rekombinacijos taškais: pasirenkami du rekombinacijos taškai tėvų duomenyse. Vėliau visi „tėvų“ duomenys, esantys tarp šių taškų sukeičiami vietomis. Rezultatas – „vaikai“, kurie turi dalį vieno ir kito tėvo duomenų (žr. Pav. 5) [8].



Pav. 4 Kryžminimas per vieną tašką

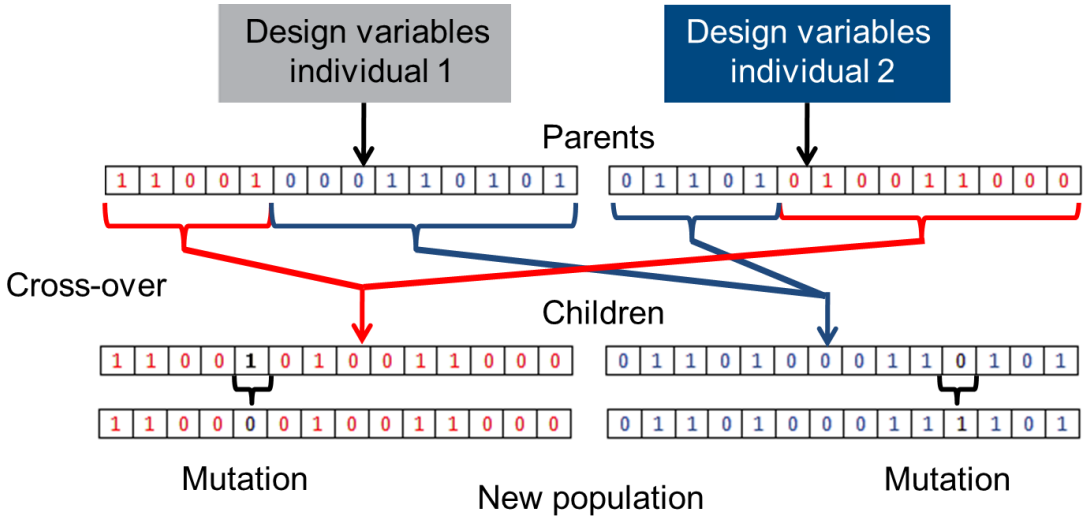


Pav. 5 Kryžminimas per du taškus

1.3.4 Mutacija

Gautiems naujiems sprendiniams pritaikoma mutacija. Jos metu tam tikros naujų sprendinių dalys yra pakeičiamos (žr. 6 Pav.). Taip daroma norint išvengti populiacijos užstrigimo lokaliame

optimume. Svarbu atkreipti dėmesį į tai, kad mutacijos tikimybė neturėtų būti labai didelė. Kuo didesnė ši tikimybė, tuo labiau genetinis algoritmas panašėja į paprasčiausia atsitiktinį algoritmą. Klasikinėje schemoje mutacijos tikimybė nesikeičia algoritmo vykdymo metu, todėl galima tarti, kad mutacijos tikimybė yra vienas iš genetinio algoritmo parametrų [8].

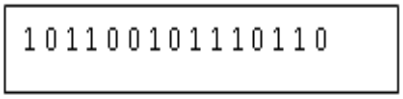


Pav. 6 Kryžminimas ir Mutacija

Nors visas algoritmas atrodo paprastai, tačiau yra daugybė parametrų ir būdų jam realizuoti. Pagrindiniai klausimai: kaip reprezentuoti sprendimus chromosomomis, ir kokiu būdu jas užkoduoti, kaip apibrėžti pagrindinius genetinio algoritmo operatorius: kryžminimą ir mutaciją, kaip parinkti tėvus, kad tikimybė gauti geresnį sprendimą būtų kuo didesnė.

1.3.5 Chromosomos kodavimas

Yra daug chromosomų kodavimo būdų, ir jie labai susiję su sprendžiama problema. Dažniausiai yra naudojamas fiksuoto ilgio, fiksuotos tvarkos bitų kodavimas, kuris yra labiausiai ištirtas. Jis naudojamas dėl savo paprastumo ir galimybės užkoduoti praktiškai bet ką. Be to, jis buvo naudotas ir pačiuose pirmuose genetinių algoritmų bandymuose. Šio kodavimo chromosomą sudaro dviejų bitų, 0 ir 1, eilutė (žr. Pav 7). Toks metodas nėra natūralus sprendžiamoms problemoms, ir dažnai po kryžminimo arba mutacijų reikalingi pataisymai. Todėl yra naudojami ir kiti kodavimai, kaip kad simbolių arba pačių reikšmių perkėlimas į chromosomas [8].



Pav. 7 Binarinis chromosomos kodavimas

1.4 Tvarkaraščių sudarymo programinė įranga

Tikrai tik nedaugelis ugdymo įstaigų Lietuvoje naudojasi esamomis programomis, padedančiomis kurti pamokų tvarkaraščius, todėl dažniausiai sudarinėja juos rankiniu būdu. Trumpai apžvelgsime esamą ir dažniausiai naudojamą programinę įrangą tvarkaraščių kūrimui Lietuvoje.

„Mimosa“

Internetinis puslapis: http://www.mimosasoftware.com/

Mimosa yra Suomių IT kompanijos „Mimosa Software Ltd“, įsikūrusios 1986 metais, sukurta programinė įranga. Programa yra universaliausia planavimo programinė įranga pasaulyje. Skirtingai nei daugelis programų, Mimosa nebuvo orientuota į tam tikrą organizacijos planavimo procesą. Ji orientuota į pagrindinių iššūkių planavimą ir suteikia vartotojui orientuotus sprendimus. Mimosa gali būti naudojama bet kokioje organizacijoje ir išspręsti bet kokią tvarkaraščių užduotį. Todėl ji naudojama ne tik mokyklose, universitetuose bet ir plačiai naudojama kituose verslo kontekstuose, nuo susitikimo organizavimo iki didelių tarptautinių konferencijų ar muzikinių festivalių organizavimo. Ši programa automatiškai sudėlioja tvarkaraštį pagal suvestus duomenis ir jį suoptimizuoja. Kai programa neranda geresnio varianto, baigia darbą. Programa anglų kalba, todėl gali kilti sunkumų ja naudojantis.

„aSc Tvarkaraščiai“

Internetinis puslapis: http://help.asctimetables.com/index.php?lang_id=10

„aSc Tvarkaraščiai“ yra Slovakijos IT kompanijos „Applied Software Consultants, s.r.o. (Company with limited liability)“ produktas. Šia programa naudojasi daugiau nei 7500 ugdymo įstaigų visame pasaulyje. Tvarkaraščių programoje labai lengva įvesti visus reikalingus duomenis: disciplinas, grupes, darbuotojų darbo laiką, užsiėmimų tarpusavio ryšius, kabinetus. Galima suskaidyti arba jungti į grupes studentus (mokinius) bendram užsiėmimui. Galima pasirinkti tikslų

laiką arba laiko intervalą, kada disciplina turi vykti, kokiame kabinete ir kas turi vesti užsiėmimą. Per keliolika minučių programa patikrins tūkstančius variantų ir pati sudarys tvarkaraštį, atitinkantį užduotas sąlygas ir apribojimus:

- maksimalų langų skaičių, dienas, kada dėstytojai (mokytojai) gali vesti užsiėmimus;
- tolygų disciplinų išdėstymą per savaitę;
- patikrins užsiėmimų nuoseklumą pilnoms ir padalintoms grupėms;
- užsiėmimams paskirs kabinetus;
- patikrins visas kitas užduotas sąlygas.

Programa patikrins duomenis ir padės jums surasti ir pašalinti galimas klaidas. Ji taip pat patikrins, ar tvarkaraštis atitinka visas užduotas sąlygas. Programa pateiks tvarkaraščio statistinę informaciją.

Jūs galėsite atsispausdinti savo tvarkaraštį. Programa automatiškai sukurs tvarkaraščius atskiroms grupėms, pedagogams bei kabinetams. Kadangi programą galima rasti ir lietuvių kalba, tai labai palengvina darbą.

„Rector“

Internetinis puslapis: http://rector.spb.ru/lt/index.php

Rector kompiuterinė programa sukurta Rusijos IT kompanijos „P. Yu. Smykalov“ 1997 metais. Ji specialiai skirta ugdymo įstaigų tvarkaraščių sudarymui. Darbas su programa paprastas, nes visi etapai programoje aiškūs ir suprantami. Su šia programa galima paskirstyti krūvius, padalinti į grupes ir iš jų sudaryti srautus, numatyti ar užsiėmimai vyks vienu ar skirtingu metu. Yra galimybė nurodyti ar pamoka bus paskutinė, ar pirma, ar pamokos turi eiti nuosekliai. Taip pat yra galimybė tvarkaraštį sudaryti lyginėmis ir nelyginėmis savaitėmis. Programa pati suskaičiuoja valandas, skirtas darbų tikrinimui, užsiėmimų pasiruošimui.

2. Tyrimo metodai

Šiame darbe nagrinėsime paskaitų tvarkaraščių sudarymo uždavinį. Sukursime programą, kuri ieško leistinų tvarkaraščių naudodama genetinį algoritmą ir tinkamumo funkciją. Pritaikydami papildomas sąlygas mėginsime rasti optimalesnį tvarkaraštį. Taip pat panagrinėsime tvarkaraščiams sudarinėti skirtą programą „aSc Tvarkaraščiai“ ir mėginsime ja gauti optimalų tvarkaraštį. Gautus rezultatus išanalizuosime ir palyginsime.

2.1 Pradiniai duomenys

Pradinių duomenų pavyzdys, kuris naudojamas tvarkaraščio sudarymui yra vaizduojamas žemiau.

Objektai:

1. Prof – pažymi profesorių.
2. Course – pažymi paskaitą.
3. Room – pažymi kabinetą.
4. Group – pažymi studentų grupę.
5. Course class – pažymi klasę (laboratorija ar ne) ir priskiria profesorių, studentų grupę ir paskaitą.

- #prof
  - id - ID dėstytojo.
  - name – dėstytojo vardas.
- #course
  - id - ID paskaitos.
  - name– paskaitos pavadinimas.
- #room
  - name - kabinetas.
  - size – vietų skaičius.
  - lab – pažymėti true, jei kabinetas turi kompiuterinę įrangą ir false priešingu atveju.
- #group
  - id - ID grupės.
  - name – grupės pavadinimas.
  - size – studentų skaičius grupėje.
- #class
  - professor - ID dėstytojo; klasei priskirtas dėstytojas.
  - course - ID paskaitos; klasei priskirta paskaita
  - group - ID grupės; klasei priskiria grupę; kiekvienai klasei gali būti priskirtos kelios grupės.
  - duration – paskaitos laikas valandomis (vertė 1).

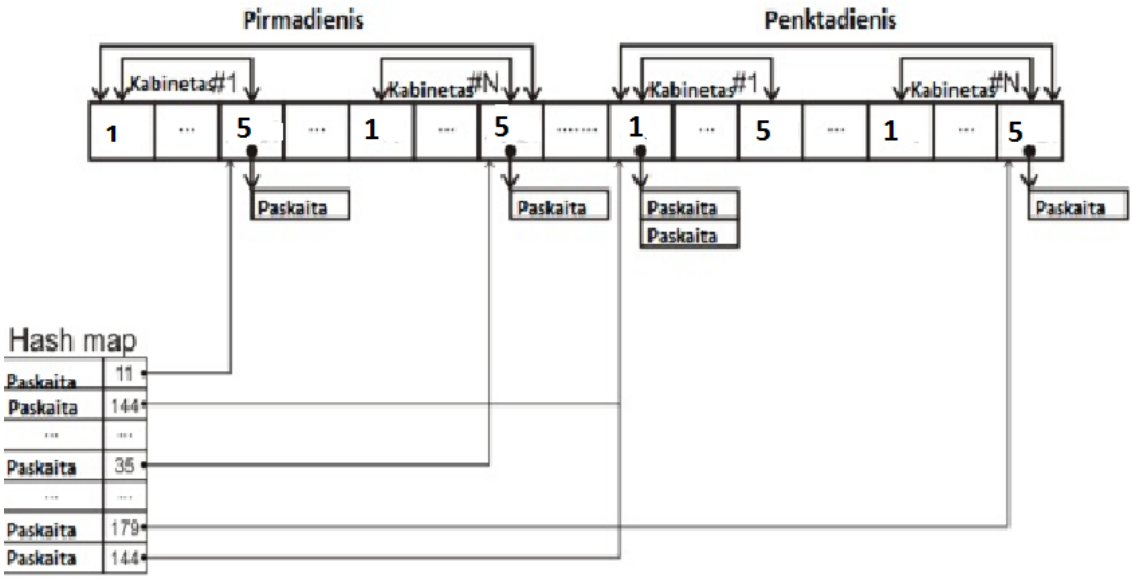
- lab - įrašoma, jei paskaitai reikalingas kabinetas su kompiuterine įranga.

2.2 Genetinio algoritmo praktinis taikymas

Pirmiausia, dirbdami su genetiniu algoritmu, turime apsvarstyti, kokiu būdu pristatyti sprendimą, kad tenkintų genetinį algoritmą, naudojant kryžminimą ir mutaciją. Be to, mes turim žinoti, kaip nustatyti, kad mūsų sprendinys yra tinkamas. Kitais žodžiais tariant, mes turime suskaičiuoti gauto sprendinio tinkamumo vertę.

2.2.1 Chromosomos pateikimas programoje

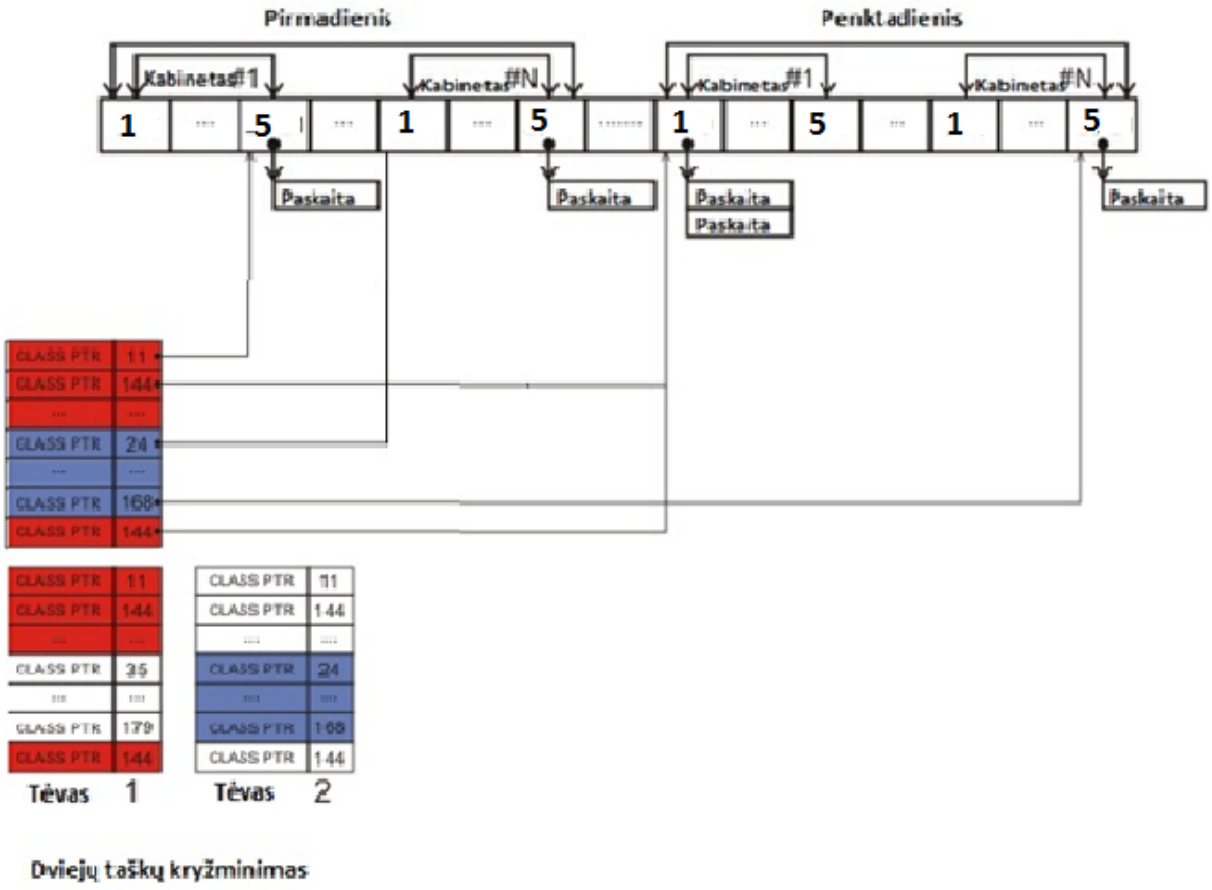
Užsiėmimo tvarkaraštį norit pateikti chromosoma mums reikės nustatyti vietą (laiko erdvę) kiekvienos paskaitos metu (sakome, kad laikas sugraduotas kas paskaitą) kiekvienam kabinetui, kiekvieną dieną. Taip pat imame, kad per dieną negali būti daugiau nei penkių užsiėmimų. Darbo dienų skaičius yra nuo Pirmadienio iki Penktadienio (viso 5 darbo dienos). Naudojame std::vector, kurio dydis apskaičiuojamas 5 x 5 x kabinetų skaičius. Vieta žymimas std::list, nes algoritmo vykdymo metu mes leidžiame, kad vyktų daugiau nei vienas užsiėmimas toje pačioje laiko erdvėje. Taip pat yra papildomas „smulkesnis“ žemėlapis (ang. hash map). Kiekviena užsiėmimo valanda turi atskirą įrašą vektoriuje, bet yra tik vienas užsiėmimo įėjimas matomas smulkesniame žemėlapyje. Pavyzdžiui, jei užsiėmimas yra pirmas ir tęsiasi tris paskaitas, ji turi turėti „įėjimus“ pirmą, antrą ir trečią paskaitomis.



Pav. 8 Chromosomos pateikimas

2.2.2 Kryžminimo vaizdavimas programoje

Kryžminimo operacija sujungia dviejų „tėvų“ duomenis smulkesniame žemėlapyje. Tada jis sukuria vietos vektorių pagal naujo smulkiojo žemėlapio turinį. Kryžminimo metu abiejų „tėvų“ smulkusis žemėlapis padalijamas į atsitiktinį dalių skaičių. Dalių skaičius chromosomos parametruose yra apibrėžtas kryžminimo taškų (plius vienas) skaičiumi. Tada paskaitomis kopijuojamos dalys iš „tėvų“ į naujas chromosomas ir formuojamas naujas vietos vektorius.



Pav. 9 Kryžminimo vaizdavimas

2.2.3 Mutacijos apibrėžimas programoje

Mutacijos operatorius yra gana paprastas. Jis atsitiktinai parenka užsiėmimą ir perkelia ją į kitą atsitiktinai parinktą vietą. Paskaitų skaičius gali būti perkeltas vienos operacijos metu ir apibrėžiamas mutacijos dydžiu chromosomos parametruose.

2.2.4 Algoritmo realizavimas programoje

Genetinis algoritmas yra gana paprastas. Kiekvienai generacijai atliekamos dvi pagrindinės operacijos:

1. Atsitiktinai parenkamas N tėvų porų iš esamos populiacijos ir sudaroma N naujų chromosomų naudojant kryžminimo operaciją tėvų poroms.
2. Atsitiktinai parenkame N chromosomų ir esamos populiacijos ir pakeičiame jas naujomis. Algoritmas nesirenka chromosomų "perstumdymui" jei jos yra tarp geriausių chromosomų visoje populiacijoje.

Šios dvi operacijos yra kartojamos iki tol, kol chromosomos pasiekia tinkamumo (ang. fitness) vertę, kuri lygi 1, tokiu atveju visi reikalavimai yra įgyvendinti. Genetinis algoritmas stebi M geriausių populiacijos chromosomų ir garantuoja, kad jos nebus pakeistos, kol jos yra tarp geriausių chromosomų.

2.3 Tinkamumo funkcijos apibrėžimas programoje

Tam, kad daugiakriterinį uždavinį pakoreguoti į vienkriterinį yra įvedamas vertės (ang. cost) skaičiavimas. Yra nustatomi svoriai pagal kriterijų svarbą. Jie priklauso nuo pačios ugdymo įstaigos. Kuo daugiau skiriame tam tikram kriterijui taškų, tuo didesnis dėmesys jam skiriamas. Dažniausiai taškų dydį įvertina tam tikri ekspertai. Įverčiai kinta priklausomai nuo ugdymo įstaigos, ugdytinių, dėstytojų (mokytojų) ir kitų parametrų.

Apibendrinant galime teigti, kad apribojimai negali būti pažeisti, tačiau praktiškai, galimi nedideli nukrypimai, jei tai pagerina sprendinio kokybę.

Tinkamumas (ang. Fitness)

Pažymėsime chromosomų tinkamumo vertes. Šio tvarkaraščio tinkamumo vertei paskaičiuoti naudojamos tik griežtos sąlygos. Tai atliekame:

- Kiekviena paskaita gali turėti 0 arba 5 taškus.
- Jei paskaitai naudojamas atskiras kabinetas, didiname taškus.
- Jei paskaitai reikalinga kompiuterinė klasė ir ji "įkurdinama" kabinete su kompiuteriais arba tokio reikalavimo nėra, mes didiname taškus.

- Jei paskaita yra “įkurdinama” kabinete, kuriame yra pakankamai vietos, taškai vėl pridedami.
- Jei dėstytojas neturi kitos paskaitos tuo pačiu metu – didiname taškus.
- Paskutinis dalykas kuri tikriname , tai ar studentų grupė, kuri lanko paskaitą turi tuo laiku kitų paskaitų, jei neturi, mes pridedame taškus.
- Jei paskaita netenkina bent vienos taisyklės bet kokioje laiko erdvėje, kurioje ji yra, tos sąlygos taškai nėra pridedami.
- Bendras tvarkaraščio įvertinimas yra visų paskaitų taškų suma.
- Tinkamumo vertė apskaičiuojama:

$$\frac{tvarkaraščio\ taškų\ suma}{maksimalus\ taškų\ skaičius}$$

kur

$$maksimalus\ taškų\ skaičius = paskaitų\ skaičius * 5$$

Tinkamumo vertė gali būti intervale nuo 0 iki 1.

3. Tyrimų rezultatai ir jų aptarimas

3.1 Pradiniai parametrai ir duomenys

Pradiniai duomenys:

Atliekant tyrimą genetiniu algoritmu daugelis parametrų, tokių kaip mutacijos skaičius, kryžminimo tikimybė, kryžminimo dydis, parenkami atsitiktinai, mutacijos tikimybė buvo parinkta 0,0015.

Tvarkaraščio duomenys:

- Studentų grupių skaičius – 10;
- Kabinetų skaičius – 9;
- Dėstytojų skaičius – 29;
- Disciplinų skaičius – 19;
- Paskaitų skaičius per dieną – 5;

Leistinam tvarkaraščiui sudaryti naudojamos šios pagrindinės sąlygos (ang. hard constraints) (Vakili M.T, 2007):

- Kabinete tam tikru laiko momentu vyksta tik viena paskaita;
- Kabinete yra pakankamai vietų studentams;
- Studentų grupė tuo pačiu metu neturi vykstančių kelių paskaitų;
- Dėstytojas neturi tuo pačiu metu dėstomų kelių paskaitų;
- Jei paskaitos metu reikalinga įranga, tai paskaita vyksta spec. kabinetuose.

Tvarkaraščiui optimizuoti naudosime kelias papildomas sąlygas: penktadieniais paskaitos vyksta tik iki pietų, ketvirtadieniais tvarkaraščiuose „vengti“ ketvirtos ir penktos paskaitos. Šios papildomos sąlygos dažniausiai naudojamos atsižvelgiant į studentų poreikius: ketvirtadienio vakarais vyksta renginiai studentams, o penktadienis yra nedarbinga diena, taip pat svetimuose miestuose studijuojantys studentai planuoja grįžti namo.

Darbo tikslas rasti tinkamą paskaitų tvarkaraštį. Tyrimui naudojame genetinį algoritmą. Taip pat panagrinėsime kaip tvarkaraščio suradimo iteracijų skaičius priklauso nuo parenkamų parametrų.

3.2 Tyrimo rezultatai

3.2.1 Genetinio algoritmo rezultatai

Mutacijos tikimybę parenkame 0,0015, o kiti genetinio algoritmo parametrai parenkami atsitiktinai. Kai tvarkaraštį generuojame 5 – ioms studentų grupėms ir parenkame, kad paskaitų skaičius per dieną gali būti maksimalus (6, 7, 8). Akivaizdu, kad kuo mažiau erdvės, į kurią reikia sutalpinti tvarkaraštį, atitinkantį visus reikalavimus, tada reikalingas didesnis generacijų skaičius (žr. Figure 1).

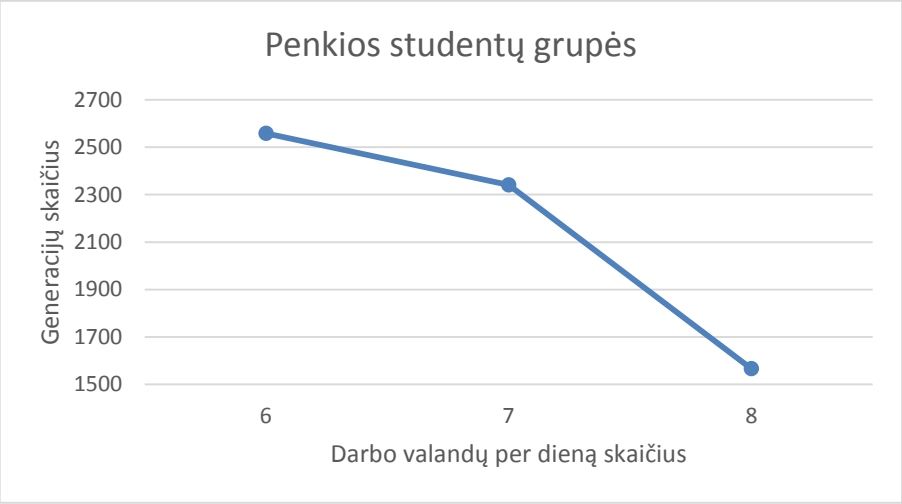


Figure 1 Generacijų priklausomybė nuo darbo valandų sk.

Tokią pačią priklausomybę matome ir parinkus 10 studentų grupių, kurioms reikia sudaryti bendrą tvarkaraštį (žr. Figure 2).

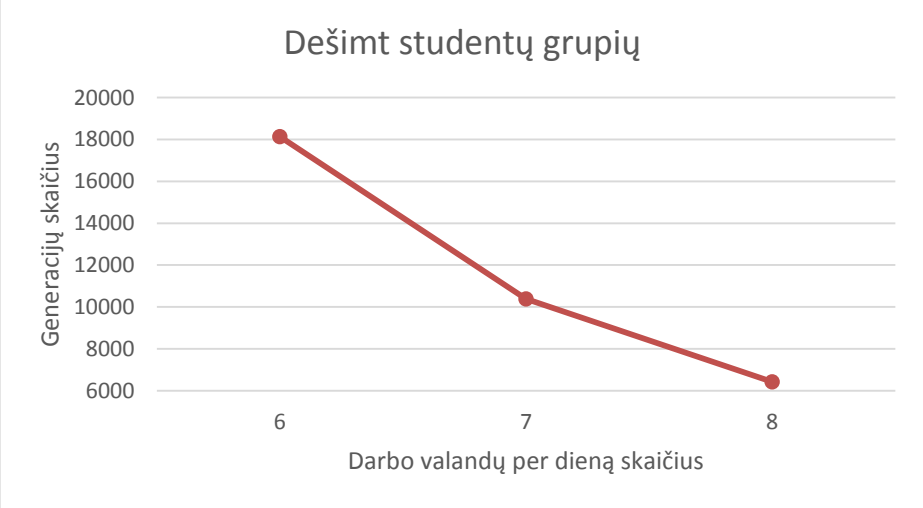


Figure 2 Generacijų priklausomybė nuo darbo valandų sk.

Žemiau esančiuose grafikuose matome, kad atliekant kuo daugiau generacijų, tuo tinkamumo rodiklis yra artimesnis 1. Lyginant šiuos du grafikus taip pat matome, kad kai ieškomas tvarkaraštis 5 grupėms, po 2000 generacijų, tinkamumo rodiklis yra didesnis 0,1, nei 10 grupių tvarkaraščio tinkamumo rodiklis (žr. Figure 3 ir Figure 4). Kai studentų grupių yra 5 – kios, matome, kad generacijų skaičius, pasiekus tinkamumo rodiklį lygų 1, yra 1543, kai darbo valandų skaičius yra 7val.

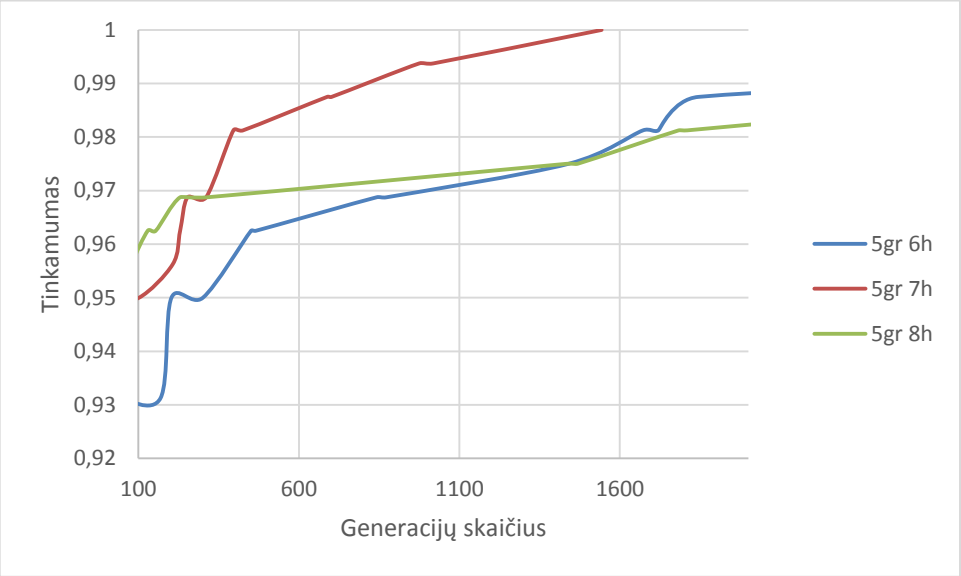


Figure 3 Tinkamumo rodiklio augimas 5gr.

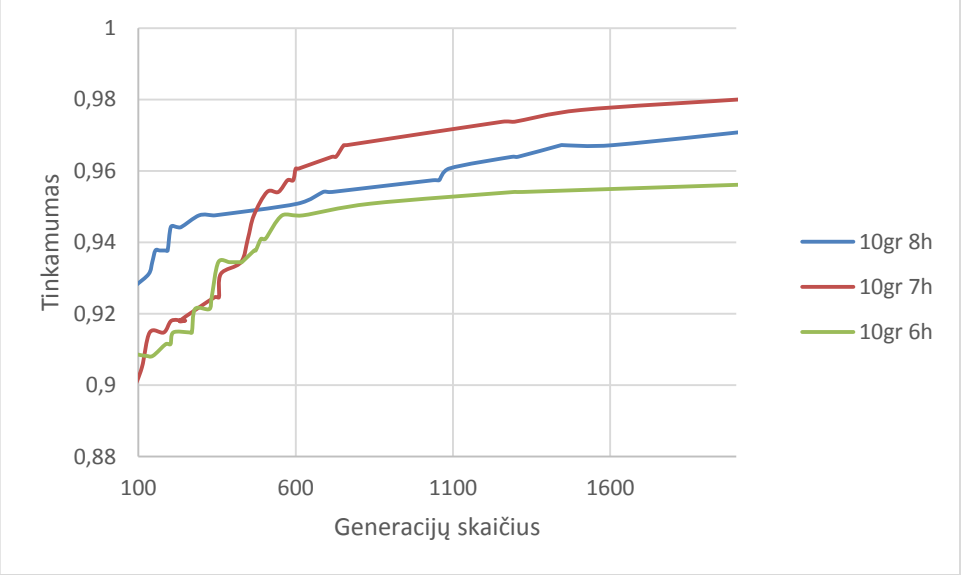


Figure 4 Tinkamumo rodiklio augimas 10gr.

Sugeneravome tvarkaraščius su skirtingais mutacijų dydžiais. Matome, kad lyginant paskaitų skaičių, kai parenkame 7 , generacijų skaičius būna arba didesnis nei 6 ir 8 paskaitos, arba

mažesnis. Analizuojant 5 studentų grupių generuojamus tvarkaraščius skirtingais mutacijos dydžiais, matome, kad parinkus mutacijos dydį 5 tvarkaraštį, su 6 paskaitomis per dieną, sugeneruoja daug greičiau, nei parinkus 15. O kai paskaitų skaičius 8 – atvirkščiai, prie 5 generuoja ilgiau nei prie 15.

Lentelė 1 Generacijų skaičius keičiant mutacijos sk.

Mutacijos skaičius	Studentų gr. Skaičius	Darbo valandų sk. per dieną	Generacijų sk.
5	5	6	828
		7	2644
		8	2020
	10	6	4430
		7	6887
		8	4836
10	5	6	1326
		7	1567
		8	2773
	10	6	3925
		7	5249
		8	4258
15	5	6	1712
		7	1561
		8	428
	10	6	8701
		7	6211
		8	6382

Žiūrėdami į žemiau esančią lentelę matome, kad parinkus mutacijos tikimybę 0,0015, 5 studentų grupėms su tam tikru mutacijos dydžiu generuoja ilgiau, nei parinkus papildomai dar ir kryžminimo taškų skaičių = 20. Generacijos dydis ženkliai sumažėja.

Lentelė 2 Generacijų skaičius parenkant parametrus

Mutuojančių paskaitų skaičius	Generacijų skaičius pasiekus 1	Generacijų skaičius pasiekus 1, kai kryžminimo taškų sk=20
10	1090	621

11	1265	525
12	3282	1650

Naudodama atsitiktinius genetinio algoritmo parametrus, gavau tvarkaraštį:

Lentelė 3 Tvarkaraštis gautas taikant genetinį algoritmą

	PIRMADIENIS					ANTRADIENIS				
	1	2	3	4	5	1	2	3	4	5
4/1			Ties.alg; P2; 109				Ties.alg.; P2; 101	Geometrija; P1; 109		Ob.prog; P4; 108
4/2							Ties.alg.; P2; 101			
3/1	Mat. Anal.; P28; 102		Duom.baz; P24; 102				Fizika2; P23; 106			Mat.anal; P28; 109
3/2	Mat. Anal.; P28; 102		Duom.baz; P24; 102		Mat.log; P27, 101				Fizika2; P23; 109	
2/1		Skait.met; P11; 109	Skait.met; P11; 103					Disk.trans; P16; 107	Mat.stat; P12; 103	
2/2	Lošimų t.; P15; 101k		Skait.met; P11; 103			Mat.stat; P12; 108			Mat.stat; P12; 103	
2/3			Skait.met; P11; 103						Mat.stat; P12; 103	
1/1				Riz.val; P18; 102	Alg.struk; P16; 102		Duom. A; P17; 107			
1/2		Duom.a.; P18; 107		Riz.val; P18; 102	Alg.struk; P16; 102					
1/3			Alg. Struk; P16; 104	Riz.val; P18; 102	Alg.struk; P16; 102	Duom.a.; P18; 109				

	TREČIADIENIS					KETVIRTADIENIS				
	1	2	3	4	5	1	2	3	4	5
4/1	Asm.sveik. ug; P10; 105	Mat.a.1; P7; 101			Geometrija; P1; 108					
4/2	Mat.a.1; P7; 108	Mat.a.1; P7; 101			Geometrija; P1; 108		Geometrija; P1; 105			
3/1				Mat.anal; P28; 108				Duom.baz; P25; 108		Fizika2; P21; 109
3/2				Mat.anal; P28; 108				Duom.baz; P25; 108		Fizika2; P21; 109
2/1	Mat.log; P26; 106	Inf.sis.pag; P14; 102	Mat.stat; P12; 108			Inf.sis.pag; P13; 103			Lošimų t; P15; 101	
2/2		Inf.sis.pag; P14; 102			Mat.stat; P12; 107	Inf.sis.pag; P13; 103				
2/3		Inf.sis.pag; P14; 102				Inf.sis.pag; P13; 103		Lošimų t; P15; 101		Skait.met; P11; 101
1/1	Mat.sis.mod; P19; 103			Alg. Struk; P16; 104			Duom.a; P17; 103			
1/2					Mat.sis.mod; P19; 106		Duom.a; P17; 103	Alg.struk; P16; 103		
1/3							Duom.a; P17; 103			

	PENKTADIENIS				
	1	2	3	4	5
4/1	Mat.a.1; P8; 101			Ob.prog; P3; 101	Geometrija; P1; 101
4/2		Ob.prog; P3; 109	Ties.alg; P2; 106	Ob.prog; P3; 101	Geometrija; P1; 101
3/1	Fizika2; P22; 109		Finan.pag; P29; 103		
3/2		Fizika2; P22; 108	Finan.pag; P29; 103		
2/1		Lošimū t; P15; 102			
2/2		Lošimū t; P15; 102	Disk.trans; P16; 102	Skait.met; P11; 105	
2/3	Disk.trans; P16; 107	Lošimū t; P15; 102			
1/1			Riz.val; P20; 101		Mat.sis.mod; P19; 103
1/2			Riz.val; P20; 101		Mat.sis.mod; P19; 103
1/3		Mat.sis.mod; P19; 107	Riz.val; P20; 101		Mat.sis.mod; P19; 103

Šis tvarkaraštis tinkamas naudojimui, nes tenkina visas pagrindines sąlygas. Tinkamumas pasiektas po 22278 atliktų generacijų. Tačiau šis tvarkaraštis nėra optimalus studentų atžvilgiu. Pritaikę papildomas sąlygas gauname tokį tvarkaraštį:

Lentelė 4 Tvarkaraštis gautas genetiniu algoritmu su papildomomis sąlygomis

	PIRMADIENIS					ANTRADIENIS				
	1	2	3	4	5	1	2	3	4	5
4/1		Ob.prog; P3; 101	Ties.alg; P2; 109			Mat.a.1; P8; 101		Ob.prog; P4; 108		Geometrija; P1; 101
4/2			Geometrija; P1; 101	Mat.a.1; P7; 108			Ob.prog; P3; 109	Ties.alg; P2; 106	Ob.prog; P3; 101	
3/1	Mat.anal; P28; 102		Duom.baz; P24; 102			Fizika2; P22; 109		Finan.pag; P29; 103		
3/2	Mat.anal; P28; 102		Duom.baz; P24; 102	Fizika2; P23; 109	Mat.log; P27, 101		Fizika2; P22; 108	Finan.pag; P29; 103		
2/1		Skait.met; P11; 109	Skait.met; P11; 103	Lošimų t; P15; 101		Mat.stat; P12; 108	Lošimū t; P15; 102			
2/2	Lošimų t.; P15; 101k		Skait.met; P11; 103				Lošimū t; P15; 102	Disk.trans; P16; 102	Skait.met; P11; 105	
2/3			Skait.met; P11; 103			Disk.trans; P16; 107	Lošimū t; P15; 102			
1/1				Riz.val; P18; 102	Alg.struk; P16; 102			Riz.val; P20; 101		Mat.sis.mod; P19;
1/2		Duom.a.; P18; 107		Riz.val; P18; 102	Alg.struk; P16; 102			Riz.val; P20; 101		Mat.sis.mod; P19;
1/3			Alg. Struk; P16; 104	Riz.val; P18; 102	Alg.struk; P16; 102		Mat.sis.mod; P19;	Riz.val; P20; 101		Mat.sis.mod; P19;

	TREČIADIENIS					KETVIRTADIENIS				
	1	2	3	4	5	1	2	3	4	5
4/1	Asm.sveik .ug; P10;	Mat.a.1; P7; 101			Geometrij a; P1; 108		Ties.alg.; P2; 101	Geometrij a; P1; 109		
4/2		Mat.a.1; P7; 101			Geometrij a; P1; 108		Ties.alg.; P2; 101			
3/1		Fizika2; P21; 109	Mat.anal; P28; 109	Mat.anal; P28; 108			Fizika2; P23; 106	Duom.baz ; P25; 108		
3/2		Fizika2; P21; 109		Mat.anal; P28; 108				Duom.baz ; P25; 108		
2/1	Mat.log; P26; 106	Inf.sis.pag ; P14; 102		Mat.stat; P12; 103				Disk.trans ; P16; 107		
2/2		Inf.sis.pag ; P14; 102		Mat.stat; P12; 103	Mat.stat; P12; 107	Mat.stat; P12; 108				
2/3		Inf.sis.pag ; P14; 102		Mat.stat; P12; 103		Lošimų t; P15; 101		Skait.met; P11; 101		
1/1	Mat.sis.m od; P19;			Alg. Struk; P16; 104			Duom. A; P17; 107			
1/2					Mat.sis.m od; P19;	Alg.struk; P16; 103				
1/3						Duom.a.; P18; 109				

	PENKTADIENIS				
	1	2	3	4	5
4/1					
4/2	Geometrij a; P1; 105				
3/1					
3/2					
2/1	Inf.sis.pag ; P13; 103				
2/2	Inf.sis.pag ; P13; 103				
2/3	Inf.sis.pag ; P13; 103				
1/1	Duom.a; P17; 103				
1/2	Duom.a; P17; 103				
1/3	Duom.a; P17; 103				

Kaip matome šis tvarkaraštis jau yra optimalesnis ir tenkina studentų poreikius. Taip pat matome, kad gautame tvarkaraštyje studentų grupei netenka daugiau nei keturių paskaitų per dieną. O penktadienį studentai turi tik pirmą paskaitą.

3.2.2 Programine įranga „aSc Tvarkaraščiai“ gautas rezultatas

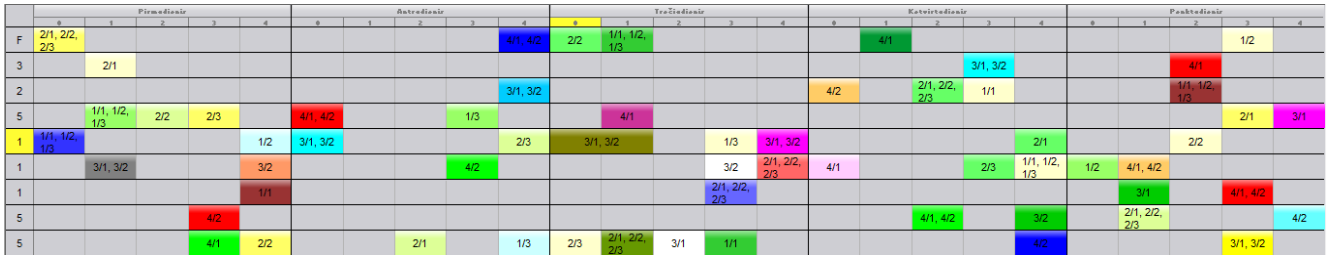
Ši programa labai paprasta ir suprantama, todėl ja lengva naudotis. Tereikia suvesti pagrindinę informaciją (pradinius duomenis):

- Kabinetus;
- Dėstytojus;

- Disciplinas;
- Studentų grupes.

Susiejame dėstytoją su paskaita ir studentų grupe, kuriai bus dėstomas vienas ar kitas dalykas. Tuomet programa patikrina ar pagal suvestus duomenis tvarkaraščio sudarymas yra įmanomas ir per kelias sekundes sugeneruoja vieną iš tinkamų tvarkaraščių.

Suvedę tik pagrindinius duomenis gavome tokį tvarkaraštį.



Pav. 10 aSc Tvarkaraščiai programa gautas rezultatas

Kad būtų galima lengviau palyginti gautus tvarkaraščius vienu ir kitu būdu, gautą tvarkaraštį su „aSc Tvrakaraščiai“ programa perkonstruojame (žr. Lentelė 5).

Lentelė 5 Perkonstruotas tvarkaraštis

	PIRMADIENIS					ANTRADIENIS				
	1	2	3	4	5	1	2	3	4	5
4/1				Ties.alg; P2; 107		Geometrija; P1; 104			Ob. Prog; P3; 103	
4/2		Geometrija; P1; 106				Geometrija; P1; 104			Ties.alg; P2; 108	
3/1		Duom.baz; P24; 108				Mat.anal; P28; 101			Fizika2; P25; 109	Fizika2; P21; 105
3/2		Duom.baz; P24; 108			Mat.log; P27; 108	Mat.anal; P28; 101			Fizika2; P25; 109	
2/1	Mat. Stat; P12; 103	Disk.trans; P16; 102						Skait.met; P11; 107		
2/2	Mat. Stat; P12; 103		Skait.met; P11; 104		Mat.stat; P12; 107					
2/3	Mat. Stat; P12; 103			Mat.stat; P12; 104						Skait.met; P11; 101
1/1	Duom.anal; P20; 101	Riz.val; P18; 104			Mat.sis.mod; P19; 109					
1/2	Duom.anal; P20; 101	Riz.val; P18; 104		Mat.sis.mod; P19; 101						
1/3	Duom.anal; P20; 101	Riz.val; P18; 104							Duom.anal; P18; 104	Mat.sis.mod; P19; 107

	TREČIADIENIS					KETVIRTADIENIS				
	1	2	3	4	5	1	2	3	4	5
4/1		Mat.s.1; P8; 104				Ob.prog; P4; 108	Asm.sveik.ug; P10; 103	Ties.alg; P2; 109		Geometrija; P1; 102
4/2						Mat.a.1; P7; 105		Ties.alg; P2; 109		Ob.prog; P3; 107
3/1	Finan.pag; P29; 101		Fizika2; P22; 107		Mat.log; P26; 101				Mat.anal; P28; 102	
3/2	Finan.pag; P29; 101			Fizika2; P22; 108	Mat.log; P26; 101				Mat.anal; P28; 102	Fizika2; P23; 106
2/1		Inf.sis.pag; P14; 107			Inf.sis.pag; P13; 108			Lošimų t; P15; 105		Lošimų t; P15; 101
2/2	Lošimų t; P15; 103	Inf.sis.pag; P14; 107			Inf.sis.pag; P13; 108			Lošimų t; P15; 105		
2/3	Disk.trans; P16; 107	Inf.sis.pag; P14; 107			Inf.sis.pag; P13; 108			Lošimų t; P15; 105	Lošimų t; P15; 108	
1/1		Duom.a; P3; 103		Duom.a; P17; 107					Alg.struk; P16; 105	Alg.struk; P16; 108
1/2		Duom.a; P3; 103								Alg.struk; P16; 108
1/3		Duom.a; P3; 103		Alg.struk; P16; 101						Alg.struk; P16; 108

	PENKTADIENIS				
	1	2	3	4	5
4/1		Mat.a.1; P7; 108		Geometrija; P1; 109	
4/2		Mat.a.1; P7; 108		Geometrija; P1; 109	Asm.sveik.ug; P9; 106
3/1		Fizika2; P23; 109		Duom.baz; P25; 107	Mat.log; P26; 104
3/2				Duom.baz; P25; 107	
2/1		Skait.met; P11; 106		Mat.stat; P12; 104	
2/2		Skait.met; P11; 106	Disk.trans; P16; 101		
2/3		Skait.met; P11; 106			
1/1			Mat.sis.mod; P19; 105		
1/2	Duom.anal; P18; 108		Mat.sis.mod; P19; 105	Alg.struk; P16; 103	
1/3			Mat.sis.mod; P19; 105		

Vienu ir kitu būdu gauti tvarkaraščiai tenkina pagrindines leistino tvarkaraščio sąlygas:

- Kabinete tam tikru laiko momentu vyksta tik viena paskaita;
- Kabinete yra pakankamai vietos studentams sėdėti;
- Studentų grupė tuo pačiu metu neturi vykstančių kelių paskaitų;
- Dėstytojas neturi tuo pačiu metu dėstomų kelių paskaitų;
- Jei paskaitos metu reikalinga įranga, tai paskaita vyksta spec. kabinetuose.

Tačiau lygindami du atsitiktinius tvarkaraščius gautus naudojant genetini algoritmą ir programą „aSc Tvarkaraščiai“ (tik pritaikius pagrindines sąlygas) matome, kad pirmu atveju tik vienai grupei ir tik kartą savaitėje yra keturios paskaitos, o pasitelkus programinę įrangą gautas

tvarkaraštis turi keletą tokių grupių, kurioms tam tikrą savaitės dieną yra keturios paskaitos. Tačiau tokiais atvejais negavome nė vienos dienos, kad studentai turėtų penkias paskaitas iš eilės.

Pritaikius ne tik pagrindines, bet ir papildomas sąlygas, gavome optimalesnius tvarkaraščius, kurie tenkina studentų poreikius.

Lentelė 6 Tvarkaraštis sudėliotas taikant papildomas sąlygas

	PIRMADIENIS					ANTRADIENIS				
	1	2	3	4	5	1	2	3	4	5
4/1	Asm.sveik. ug; P10; 107	Geometrija ; P1; 105	Mat.a.1; P7; 109			Ob.prog; P3; 101				
4/2	Ties.alg; P2; 101	Geometrija ; P1; 105	Mat.a.1; P7; 109			Ob.prog; P3; 101		Ob.prog; P3; 108		
3/1	Fizika2; P21; 103						Duom.baz; P24; 103	Duom.baz; P25; 107	Mat.anal; P28; 103	
3/2		Fizika2; P22; 104				Mat.log; P27; 107		Duom.baz; P25; 107	Mat.anal; P28; 104	
2/1			Mat.sis.P12 ; 104			Skait.met; P11; 109	Lošimų t; P15; 101	Skait.met; P11; 103		
2/2	Disk.trans; P16; 102	Skait.met; P11; 106				Skait.met; P11; 109	Lošimų t; P15; 101			
2/3		Mat.sis; P 12; 109	Lošimų t; P15; 106			Skait.met; P11; 109	Lošimų t; P15; 101			
1/1		Alg.struk; P16; 107	Duom.an; P17; 108	Alg.struk; P16; 105		Mat.sis.mo d; P19; 104		Duom.a; P20; 109	Duom.al; P17; 105	
1/2				Alg.struk; P16; 105		Mat.sis.mo d; P19; 104		Duom.a; P20; 109	Duom.al; P17; 105	
1/3			Mat.sis.mo d; P19; 105	Alg.struk; P16; 105		Mat.sis.mo d; P19; 104		Duom.a; P20; 109	Duom.al; P17; 105	

	TREČIADIENIS					KETVIRTADIENIS				
	1	2	3	4	5	1	2	3	4	5
4/1	Ties.alg; P2; 103;	Ob.prog; P4; 109	Geometrija; P1; 108	Mat.a.1; P8; 103			Geometrija; P1; 105	Ties.alg; P2; 105		
4/2						Mat.a.1; P7; 101	Geometrija; P1; 105	Ties.alg; P2; 105		
3/1	Fizika2; P22; 108	Mat.log; P26; 106	Mat.log; P26; 101	Fizika2; P21; 102			Gfinan.pag; P28; 101			
3/2	Fizika2; P23; 106		Mat.log; P26; 101	Fizika2; P21; 102			Gfinan.pag; P28; 101			
2/1	Lošimų t; P15; 105			Inf.sis.psg; P13; 104		Mat.sis;P12; 105	Fiziks 2; P25; 107	Disk.trans; P16; 107		
2/2		Lošimų t; P15; 108		Inf.sis.psg; P13; 104			Fiziks 2; P25; 107			
2/3	Disk; trans; P16; 109			Inf.sis.psg; P13; 104			Fiziks 2; P25; 107	Skait.met; P11; 104		
1/1	Mat.sis.mod; P19; 107			Riz.val; P18; 105						
1/2			Mat.sis.mos; P5; 104	Riz.val; P18; 105			Alg.struk; P16; 108			
1/3	Duom.a; P18; 104			Riz.val; P18; 105						

	PENKTADIENIS				
	1	2	3	4	5
4/1					
4/2	Asm.svik.ug; P9; 105	Geometrija; P1; 107			
3/1		Mat.anal; P28; 105			
3/2		Mat.anal; P28; 105			
2/1	Inf.sis.p; P14; 108				
2/2	Inf.sis.p; P14; 108	Mat.stat' P12; 109			
2/3	Inf.sis.p; P14; 108				
1/1					
1/2		Duom.a; P18; 108			
1/3		Alg,struk; P16; 101			

Gavome jau optimalesnį tvarkaraštį. Matome, kad studentams nėra paskutinių paskaitų, o penktadienį paskaitos baigiasi iki pietų.

Jei norime konkretesnio tvarkaraščio, galime pritaikyti daugiau papildomų sąlygų, ne tik tokias, kokias taikėme šiame darbe. Papildomos sąlygos parenkamos atsižvelgiant į universiteto, studentų ir dėstytojų poreikius.

Programa, naudojanti genetinį algoritmą tvarkaraščiams sudėlioti, parašytą C++ kalba. Pradiniai duomenys suvedami „rankiniu“ būdu. Papildomos sąlygos turi būti aprašomos programoje atskirai. Genetinio algoritmo programa, skirta tvarkaraščiams sudaryti, netinkama naudoti, jei studentų grupės yra skaidomos, t.y. jei studentai renkasi tam tikrus modulius patys. Naudojant programą paremtą genetiniu algoritmu sugeneruoti tinkamą tvarkaraštį užima daug laiko.

Programinė įranga „aSc Tvarkaraščiai“ yra visiškai paruošta vartotojui. Labai aiški veiksmų eiga. Suvedus pradinius duomenis, programa greitai randa tinkamą tvarkaraštį. Pritaikyti papildomus apribojimus taip pat nesudėtinga. Jei reikia nustatyti laiką, kada paskaitos negali vykti, tereikia pažymėti dienas ir laikus. Įmanoma ir studentų grupes išskaidyti į pogrupius, pagal pasirenkamus modulius. Bandomoji programos versija yra nemokama, tačiau norint naudotis pilna programos versija, už licenciją reikia mokėti.

4. Išvados

- Dažniausiai optimaliems tvarkaraščiams rasti yra naudojamas atkaitinimo metodas ir genetinis algoritmas, prie šių metodu kartu naudojama įkainojimo arba tinkamumo funkcija, kuri padeda optimizuoti tvarkaraštį pagal parenkamas sąlygas;
- Sukurta programa generuoja tinkamą tvarkaraštį su pagrindinėmis ir papildomomis sąlygomis;
- Programa leidžia optimizuoti tvarkaraštį, atsižvelgiant į papildomas sąlygas, pvz., studentų poreikius;
- Sukurtą programą reikia tobulinti, nes negalima buvo sudaryti tvarkaraščių studentams, pasirinkusiems gretutines studijas;
- .Monte – Karlo metodas tinkamas tik tuo atveju, kai iteracija trunka ne ilgai. Kai iteracijų daug ir jos užtrunka pakankamai ilgai, reikia daugiau laiko.
- Programa „Mimosa“ yra universaliausia planavimo programinė įranga pasaulyje. Ji nebuvo orientuota į tam tikrą organizacijos planavimo procesą. Kadangi ji nėra specialiai sukurta kuriai nors sričiai, ja naudotis yra sudėtinga.
- Programa „aSc Tvarkaraščiai“ sukurta ugdymo įstaigų tvarkaraščiams kurti. Programą galima rasti lietuvių kalba, todėl ja nesudėtinga naudotis. Taip pat labai patogiai ir suprantamai sukurti valdikliai.
- Naudojant programinę įrangą „aSc Tvarkaraščiai“ greitai galima rasti optimalų tvarkaraštį tenkinantį ne tik pagrindines sąlygas, bet ir papildomus apribojimus.
- Programinė įranga, skirta sudaryti įvairių sričių tvarkaraščiams, reikalauja nemažai „rankinio darbo“;

Literatūros sąrašas

1. Gražvydas Felinskas. (2007). Euristinių metodų tyrimas ir taikymas ribotų išteklių tvarkaraščiams sudaryti. Vilnius, Vytauto Didžiojo universitetas. [žiūrėta 2015.04.23]. Prieiga per internetą: http://www.mii.lt/files/mii_dis_07_felinskas.pdf.
2. Mohammad Taghi Vakili. (2007). Using a genetic algorithm optimizer tool to solve university timetable scheduling problem. Iran, University of Tabria. [Žiūrėta 2015.04.23]. Prieiga per internetą: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4555397&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D4555397
3. Pupeikienė Lina. Optimizavimo metodų tyrimas ir taikymas profiliuotų mokyklų tvarkaraščių sudarymo uždaviniuose. Vilnius: VGTU leidyklos technika, 2009, 124 p.
4. Zubavičius P. Užsiėmimų tvarkaraščių sudarymas taikant euristinius algoritmus. *Straipsnis iš 11-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis*“, 2008.
5. Omar Ibrahim Obaid, Mohd Sharifuddin Ahmad, Salam a. Mostafa, Mazin Abed Mohammed. Comparing performance of genetic algorithm with varying crossover in solving examination timetabling problem. *Emerging trends in computing and information sciences*, 2012, vol. 3, p. 1427-1434.
6. D. Abramson, J. Abela. (1992) A parallel genetic algorithm for solving the school timetabling problem.. [Žiūrėta 2015.04.23]. Prieiga prie interneto: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.2679&rep=rep1&type=pdf
7. Appleby, J. S.; Blake, D. V.; Newman, E. A. 1961. Techniques for Producing School Timetables on a Computer and their Application to other Scheduling Problems, *The Computer Journal* , 237–245.
8. [Žiūrėta 2015.05.03] http://webcache.googleusercontent.com/search?q=cache:8RjKRqevBHMJ:ik.su.lt/~grazvis/ea/Genetiniai%2520algoritmai/Genetiniai%2520algoritmai,%2520perstatymu%2520kodavimas.doc+&cd=1&hl=lt&ct=clnk&gl=lt
9. Bespalova K. (2009) Mokyklos tvarkaraščių optimizavimas interneto aplinkoje įvertinant pedagoginius reikalavimus. [Žiūrėta 2015.04.20]. Prieiga prie interneto: http://vddb.library.lt/fedora/get/LT-eLABa-001:E02~2009~D_20090824_150423-06683/DS.005.0.02.ETD