

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Jurgita Mekionytė

**Hibridizuotų genetinių algoritimų parametru
optimizavimo tyrimas**

Magistro darbas

Darbo vadovas
prof. habil. dr. Jonas Mockus

Kaunas, 2011

Summary

Cutting stock problem (CSP) is a relevant area of construction and stitching industries. Due to the combinatorial complexity of a problem, it is impossible to solve it exactly in every instance. The objective of the master thesis is to find an approximate solution of this problem.

The problem was solved by employing the meta-heuristic genetic method with additional heuristics *Bottom Left Fill (BLF)* and *Bottom Left (BL)*. To improve the efficiency of this method the *Bayesian approach (BA)* was applied for parameters optimization. This is a the novelty of the work.

The methodology of solving cutting stock problems and optimization of control parameters for the proposed version of genetic algorithms was formulated and described. A comparative analysis of parameters optimization was performed. The results of the comparative analysis and their application for genetic methods is the main part this master thesis.

Thesis also includes a discussion about the development of software of the optimal cutting stock problem. The creation of initial data is described. Recommendations for the further development of the program are considered as well.

Santrauka

Žaliavų taupymo problema (angl. *Cutting Stock Problem, CSP*) yra aktuali visada. Optimaliai planuojant ir pjaustant medžiagas mažėja ne tik sunaudojamų medžiagų kiekiai ir išlaidos pastarosioms įsigyti, gerėja įmonės rodikliai ir taupomi gamtos išteklių. Kadangi ši problema yra priskiriama *NP sunkių* problemų klasei, todėl buvo pasirinkti apytiksliai sprendimo algoritmai. Magistrinis darbas taip pat yra skirtas minėtai problemai spręsti.

Magistriniame darbe įgyvendinta metodų hibridizacija ir ištirtas metodų parametru optimizavimas su *Bayes' o metodu* (angl. *Bayes approach, BA*) šiems algoritmams:

- *genetinis* (alg. *genetic, GA*) algoritmas hibridizuotas su *žemiausio kairėn* (angl. *Bottom Left, BL*) metodu;
- *genetinis* algoritmas hibridizuotas su *žemiausio kairėn užpildymo* (angl. *Bottom Left Fill, BLF*) metodu.

Lyginant ištirtus parametru optimizavimus gauta, kad hibridizacijos su žemiausio kairėn algoritmu rezultatai prastesni už hibridizacijos su žemiausio kairėn užpildymo algoritmo rezultatus. Tačiau pastaroji hibridizacija veikia gerokai ilgiau nei hibridizacijos su žemiausio kairėn metodu.

Tyrimo metu gauta, kad funkcija $f(x)$ didėja, kai didėja Bayes' o algoritmo iteracijų kiekis. Tačiau didėjant Bayes' algoritmo iteracijų kiekiui taip pat didėja ir šių metodų veikimo trukmė.

Turinys

1. Įvadas.....	5
2. Analitinė dalis.....	6
2.1. Žaliavų taupymo problema.....	6
2.2. Žemiausio kairėn ir žemiausio kairėn užpildymo algoritmai.....	8
2.3. Genetinis algoritmas.....	11
2.4. Bayes'o metodas	14
2.5. Bayes'o metodas papildytas euristikomis	15
3. Žaliavų taupymo problemos uždavinio sprendimo schema. Tiriamų metodų pritaikymas šiam uždaviniui.....	17
3.1. Žaliavų taupymo problemos uždavinio sprendimo schema	17
3.2. Hibridizuotų genetinių algoritmų veikimo principas	18
3.3. Hibridizuoto genetinio algoritmo realizacija.....	19
3.4. Žemiausio kairėn algoritmas	22
3.5. Žemiausio kairėn užpildymo algoritmas	23
4. Pradinės realizacijos tyrimas ir siūlomi tobulinimai parametrų optimizavimui.....	26
4.1. Kitų metodų pritaikymas parametrų optimizavimui	26
4.2. Sprendinių aibės išplėtimas.....	26
4.3. Sistemos vartotojo sąsajos tobulinimas.....	26
5. Parametrų optimizavimo tyrimas.....	27
5.1. Genetinio algoritmo hibridizuoto su žemiausio kairėn algoritmu parametrų optimizavimo apžvalga	28
5.2. Genetinio algoritmo hibridizuoto su žemiausio kairėn užpildymo algoritmu parametrų optimizavimo apžvalga.....	30
5.3. Hibridizuotų genetinių algoritmų parametrų optimizavimo apžvalgos palyginimas.....	32
6. Išvados.....	33
7. Literatūra	34
8. Terminų ir santrumpų žodynas	37

A priedas.....	38
B priedas.....	42

1. Įvadas

Pjovimo arba pakavimo uždaviniai (angl. *Cutting and packing problems, Cutting Stock Problem*) yra plačiai paplitę įvairiose pramonės šakose: metalo, medienos, stiklo, popieriaus, drabužių (Ferreira, et al., 1990) (Haessler, et al., 1991) (Haessler, et al., 1979). Šių uždavinių sprendimo tikslas – rasti būdą, kaip iš tam tikros formos ploto išpjauti kuo daugiau reikiamų šablonų, paliekant minimalų nepanaudotą plotą ir tai atlikti per patį trumpiausią įmanomą laiką. Optimalus šio uždavinio sprendimas bet kurioje pramonės šakoje leidžia sutaupyti daug žaliavų, o tuo pačiu sutaupyti ir dalį pinigų už kuriuos būtų įsigijami audiniai. Šioje srityje plačiai dirbama ir dabar (Bağ, et al., 2011), (Wang, et al., 2010), (Macedoa, et al., 2010) ieškant naujų tikslesnių ir greitesnių optimizavimo metodų.

Šiame darbe nagrinėjama negiljotininio, dvimačio, stačiakampių pjaustymo uždavinio atliekų minimizavimo problemos sprendimo metodo parametrų optimizavimo problema. Buvo pasirinktos tokios H. Dyckhoff klasifikacijos (Dyckhoff, 1990), pagal kurias buvo parinkti metodai tyrimui (būdas – orientuotas į išpjovą):

- Medžiaga ir pjaunamas objektas yra dviejų dimensijų;
- Žaliavų yra pakankamai arba yra deficitas;
- Lakštų yra ribotas arba neribotas kiekis, tačiau visi lakštai yra vienodo dydžio;
- Išpjovų asortimentas neribojamas.

Dėl uždavinio kombinatorinio sudėtingumo (Garey, et al., 1990) neįmanoma tiksliai ir visais atvejais pateikti optimalų jo sprendinį, todėl buvo susiaurinta pjaustymo problemos klasė. Nagrinėjamas atvejis, kai pjaustomi objektai ir iš jų gaunamos detalės yra stačiakampiai, medžiagos prigimties nepaisoma, nes tai neturi esminio skirtumo nagrinėjamai problemai. Kai smulkios detalės (išpjovos) pjaustomos iš didelių objektų (lakštų), iškyla atliekų minimizavimo problema, t.y. neišku koku būdu pjaustyti pasirinktą objektą, kad atliekų kiekis būtų mažiausias. Pagal viršuje nurodytas klasifikacijas pasirinktas apytikslis sprendimo metodas. T.y. *hibridinis genetinis algoritmas* (angl. *Hybrid genetic algorithm, HGA*) – didelio efektyvumo metodas, kuris eksperimento būdu buvo išbandytas net su vienu sudėtingiausiu – kvadratinio paskirstymo uždaviniu. Tyrimo rezultatai patvirtino, kad šis algoritmas yra pranašesnis tiek už standartinius genetinius, tiek už kitus euristinius metodus (Misevičius, et al., 2005).

Pagrindinis šio darbo tikslas - realizuoti pasirinktus hibridinius metodus bei iširti jų parametrų optimizavimą. Kadangi tai yra apytiksliai sprendimo metodai, priklausantys nuo įvairių parametrų, tai darbo eigoje buvo atliktas parametrų įtakos tyrimas sprendinio kokybei, t.y. buvo palyginti hibridizuotieji metodai apjungti su Bayes'o metodu. Darbe, taip pat, pateikta pjaustymo uždavinių sprendimo metodika.

2. Analitinė dalis

2.1. Žaliavų taupymo problema

Žaliavų taupymo problema (angl. *Cutting Stock Problem, CSP*) buvo visada. Optimaliai planuojant ir pjaustant medžiagas, mažėja ne tik sunaudojamų medžiagų kiekiai ir išlaidos pastarosioms įsigyti, gerėja įmonės rodikliai, bet ir taupomi gamtos išteklių. Visais laikais buvusią aktualią problemą identifiko L. V. Kantorovich 1939 metais (Kantorovich, 1960). Nuo to laiko buvo pradėta kurti ir analizuoti šios problemos sprendimo būdus. *Žaliavų taupymo problema* kaip pagrindinė ar šalutinė užduotis yra sprendžiama daugelyje sričių:

- Stiklo, geležies, medienos, audinių, odos supjaustymas pramoniniuose procesuose (tikslas: rasti geriausią supjaustymo šabloną, kad būtų mažiausias atliekų kiekis);
- Objektų išdėstymo ir talpinimo uždavinys logistikos procesuose (tikslas: efektyviausiai išdėstyti prekes ant paletės transportavimui kiek galima didesniais kiekiais);
- Efektyvaus patalpų padalinimo uždavinys (tikslas: sukurti patalpų padalinimo schemą, efektyviausiam erdvės išnaudojimui);
- Kompiuterio atminties valdymo uždavinys (tikslas: optimaliai paskirstyti kompiuterių atmintį, kad būtų pagreitinami kompiuterių skaičiavimo procesai).

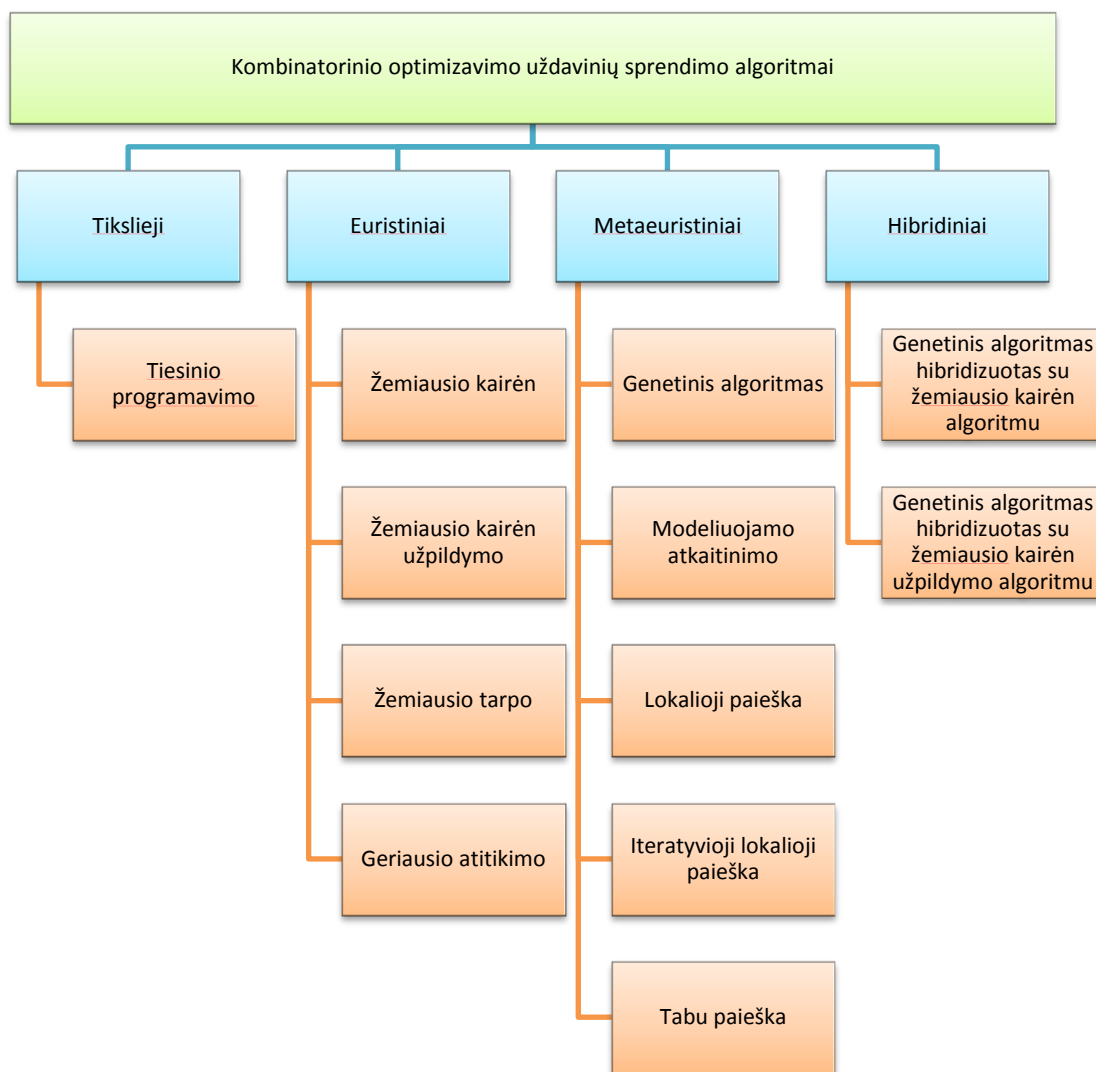
Sprendžiant tokias užduotis, yra naudojami optimizavimo metodai. Šiuose metoduose paprastai susiduriama su diskrečiais dydžiais ir optimizavimo procesas grindžiamas elementariomis kombinatorinėmis operacijomis, todėl pjaustymo ir pakavimo uždavinys priskiriamas *kombinatorinio optimizavimo* (angl. *Combinatorial optimization, CO*) užduočių klasei (Dyckhoff, 1990). Šiems pjaustymo uždaviniams yra būdinga, kad didėjant jų apimčiai, tiksliai ir greitai juos išspręsti pasidaro neįmanoma. Todėl algoritmų sudėtingumo teorijoje tokie uždaviniai priskiriami *NP sunkių* problemų klasei (angl. *Nondeterministic polynomial hard, NP hard*) (Garey, et al., 1990). Be to, kai kurių pjaustymo užduočių sprendimo variantuose reikalaujama operuoti sveikaisiais skaičiais, taigi susiduriama su *sveikaskaičio programavimo* uždaviniu (angl. *Integer programming*) (Gomory, et al., 1963) (Gomory, 1965). Dėl šių priežasčių tikslieji metodai yra panaudojami ribotai, t.y. šie metodai taikomi, tačiau nedidelės apimties užduotims spręsti (Burke, et al., 2004).

Sparčiai besivystant kompiuterinėms technologijoms, pradėtas simuliuoti pjaustymo procesas kompiuterio pagalba. Todėl atsivėrė plačios galimybės taikyti euristinius ir metaeuristinius sprendimo metodus, kurie sėkmingai panaudoti apytiksliam pjaustymo uždavinio sprendimui (Burke, et al., 2004).

Pagrindiniai šio uždavinio sprendimo metodai anksčiau literatūroje buvo skirstomi į tris abibendrinančias klases. Tačiau pastaruoju metu atsirado dar viena klasė – hibridizuotieji

algoritmai. 1 paveikslėlyje matome kaip yra skirstomi kombinatorinio optimizavimo uždavinių sprendimo algoritmai. Žemiau pateikiami visi klasių aprašymai:

- **Tikslieji algoritmai** – tokie metodai garantuoja tikslų ir tikrai patį optimaliausią sprendinį nepaisant kokio sudėtingumo optimizavimo problema ar koks duomenų, kuriuos siekiama optimizuoti, kiekis. Svarbiausias algoritmų minusas: esant dideliems kiekams duomenų ar itin sudėtingiems duomenims metodų veikimas gali būti labai ilgas. Pirmieji šiuos algoritmus nagrinėjo P. C. Gilmore ir R. E. Gomory (Gomory, 1965). Naudodami tiesinio programavimo ideologiją jie gavo optimalius sprendinius, tačiau tik nedidelės apimties uždaviniams spręsti.
- **Euristiniai metodai** – metodai orientuoti į konkrečią ir tiksliai apibrėžtą problemą. Jų veikimas sąlyginai (lyginant su tiksliaisiais algoritmais) trumpas, tačiau šie metodai dažniausiai neranda/nepateikia optimalaus sprendinio. Žaliavų taupymo problemai spręsti labiausiai žinomi yra *žemiausio kairėn užpildymo* algoritmas (angl. *Bottom Left Fill, BLF*) A. Albano ir G. Sapuppo pristatytas 1980 metais (Albano, et al., 1980), *žemiausio kairėn* algoritmas (angl. *Bottom Left, BF*) pristatytas S. Jakobs (Jakobs, 1996), tobulintas D. Liu ir H. Teng 1999 metais (Liu, et al., 1999) ir *geriausio atitikimo* algoritmas (angl. *Best Fit*), kuris gauna gerus rezultatus ir su didelės apimties uždaviniais, pristatytas 2004 metais E. K. Burke, G. Kendall ir G. Whitwell (Burke, et al., 2004).
- **Metaeuristiniai metodai** - orientuoti į apibrėžtą panašių problemų grupę, paprastai remiasi euristiniais metodais. Dažniausiai parenka pakankamai tikslus sprendinius per sąlyginai (lyginant su tiksliaisiais algoritmais) trumpą laiką. Kombinatoriniai optimizavimo uždaviniai yra pagrindinė sritis, kurioje naudojami metaeuristiniai metodai (Papadimitriou, et al., 1982). Geriausiai žinomi (žaliavų taupymo problemai spręsti) yra du algoritmai: 1990 metais pristatytas *modeliuojamo atkaitinimo* algoritmas (angl. *Simulated Annealing*) (Dagli, et al., 1990) ir 1995 metais B. Kröger pristatytas *genetinis algoritmas* (angl. *Genetic Approach, GA*) giljotininiams sukirpimo uždaviniams spręsti (Kröger, 1995).
- **Hibridiniai algoritmai** – metodai, kurie apjungia kelis skirtingus metodus (dažniausiai metodų junginyje vienas yra iš euristinių metodų, o kitas - iš metaeuristinių). Metodai apjungiami, kad kompensuotų vienas kito trūkumus. Žaliavų taupymo problemai spręsti naudojami *žemiausio kairėn* ir *genetinio* algoritmų junginys (Jakobs, 1996) arba *žemiausio kairėn užpildymo* ir *genetinio* algoritmų junginys (Hopper, et al., 1999).

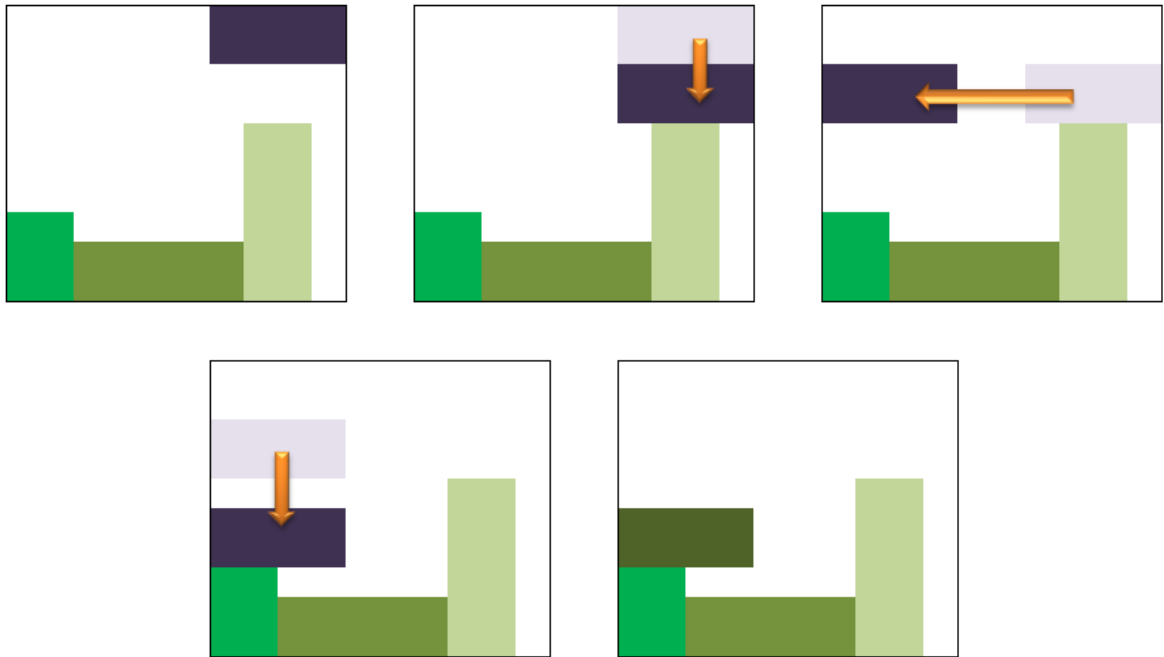


1 paveikslėlis. Kombinatorinio optimizavimo uždavinių sprendimo algoritmų klasifikacija

2.2. Žemiausio kairėn ir žemiausio kairėn užpildymo algoritmai

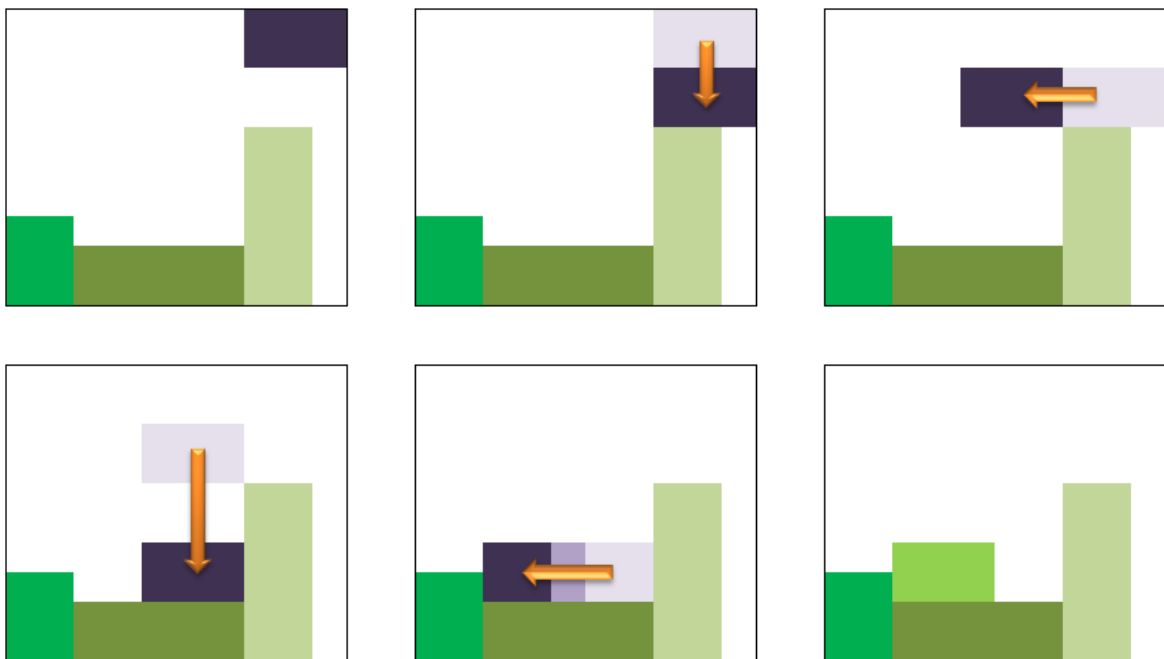
Euristiniai metodai pateikia pakankamai gerus per trumpą laiko tarpą sprendimus. Labiausiai dokumentuoti yra *žemiausio kairėn* (angl. *Bottom Left, BL*) ir *žemiausio kairėn užpildymo* (angl. *Bottom Left Fill, BLF*) algoritmai. 1996 metais Jakobs panaudojo *BL* metodą ortogonaliamai dvimatei negiljotininei pjaustymo problemai spręsti (Jakobs, 1996).

Algoritmui paduodamas stačiakampių, kuriuos reikia išpjaustyti, sąrašas. Atitinkamai detalės yra dėliojamos ant lakšto (pjaustomo objekto) tokiu būdu: pirmiausia stačiakampis yra orientuojamas į viršutinį dešinį lakšto kampą, tada atliekami perkėlimo veiksmai iš pradžių į apačią, paskui į kairę. Jakobs metodo veikimo principas pavaizduotas 2 paveikslėlyje. Violetine spalva pavaizduotas pjaunamas stačiakampis. Žalia spalva – jau išpjauti stačiakampiai.



2 paveikslėlis. Jakobs žemiausio kairėn euristinio algoritmo veikimo principas

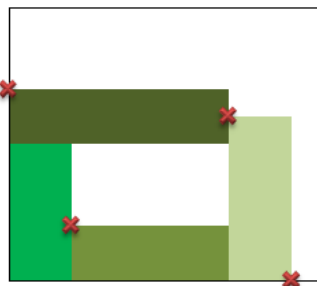
D. Lui ir H. Teng 1999 metais (Liu, et al., 1999) sukūrė pagerintą žemiausio kairėn algoritmo euristiką, suteikiančią judesiui žemyn prioritetą tokį, kad figūros perstumiamos į kairę tik tada, kai negalimas slydimas žemyn. Buvo parodyta (ko negalima pasakyti apie Jakobs metodą), kad visą laiką egzistuoja mažiausiai viena stačiakampių seka, galinti duoti optimalų sprendimą. Šis veikimo principas pavaizduotas 3 paveikslėlyje. Violetine spalva pavaizduotas pjaunamas stačiakampis. Žalia spalva – jau išpjauti stačiakampiai.



3 paveikslėlis. D. Liu Ir H. Teng pagerinto žemiausio kairėn euristinio algoritmo veikimo principas

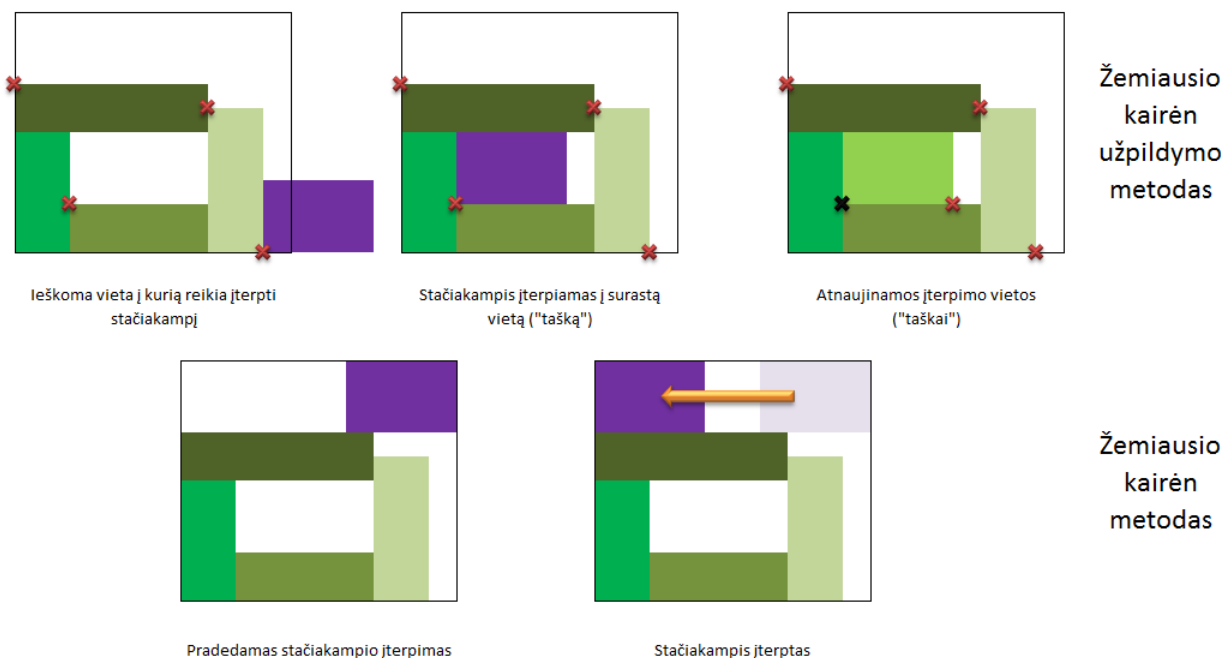
Žemiausio kairėn užpildymo metodas yra modifikuota žemiausio kairėn dėliojimo euristikos versija. Pradedant nuo apatinio kairiojo kampo papildomai sudaromas taškų sąrašas,

kuris parodo, kur būtų galima įdėti figūrą. Pjaustymo metu algoritmas pradeda nuo žemiausio ir kairiausio taško, kuriame yra talpinama figūra. Tikrinama, ar talpinamas stačiakampis nesikerta su kokia nors kita anksčiau įdėta figūra arba lakšto kraštu. Jeigu nesikerta, figūra paliekama ir taškų sąrašas yra atnaujinamas, kitu atveju nagrinėjamas sekantis taškų sąrašo elementas (taškas), procesas tęsiasi tol, kol įmanoma įdėti figūrą taip, kad ši nesikirstų su kitomis.



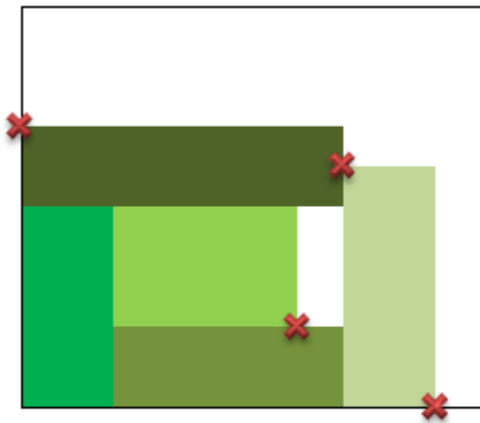
4 paveikslėlis. Galimos figūrų įdėjimo vietos žemiausio kairėn užpildymo metode

4 paveikslėlyje parodyti taškai, kuriose įmanoma talpinti sekančią figūrą. Raudonai pažymėti taškai į kuriuos galima įterpti naujai pjaunamus stačiakampius. Kaip jau minėta, paieškos procesas pradeda nuo žemiausio kairiausio taško (situacija, kai taškai yra viename lygyje, imamas kairiausias taškas). Taigi akivaizdu, kad žemiausio kairėn užpildymo metodas gali užpildyti susidariusias vakansijas.

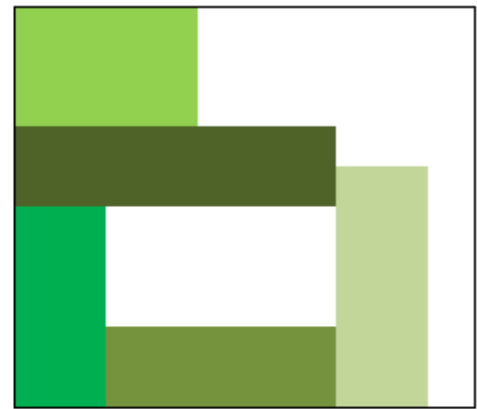


5 paveikslėlis. Pagerinto žemiausio kairėn ir žemiausio kairėn užpildymo metodų veikimo principų palyginimas

5 paveikslėlyje matome skirtumus tarp žemiausio kairėn ir žemiausio kairėn užpildymo metodų veikimų. Violetine spalva pažymėtas įterpiamas stačiakampis. Šių įterpimų rezultatus matome 6 paveikslėlyje.



Žemiausio kairėn užpildymo metodas



Žemiausio kairėn metodas

6 paveikslėlis. Žemiausio kairėn užpildymo ir žemiausio kairėn metodų vykdymo rezultatai

Metodų vykdymo rezultatuose matome, kad susidariusi vakansija niekada nebus užpildyta problemą sprendžiant žemiausio kairėn metodu, todėl šiuo atžvilgiu žemiausio kairėn užpildymo metodas yra pranašesnis. Verta paminėti, kad sprendinio kokybė priklauso nuo stačiakampių pateikimo eiliškumo. Nustatyta, kad žemiausio kairėn algoritmo sudėtingumas yra $O(n^2)$, kai N pjaustomų detalių skaičius, atitinkamai žemiausio kairėn užpildymo - $O(n^3)$. žemiausio kairėn užpildymo sudėtingumas didesnis, todėl laiko sąnaudų prasme jis dirba ilgiau, tačiau gautas sprendinys yra tikslesnis.

2.3. Genetinis algoritmas

Pirmasis genetinių algoritmų sąvoką pasiūlė A.S. Fraser (Fraser, 1957). Didesnį šių algoritmų naudojimą inspiravo 1970 metais parašyta J. H. Holland knyga *Natūralių ir dirbtinių sistemų prisitaikymas* (angl. *Adaptation in Natural and Artificial Systems*) (Holland, 1992). Jau pradėdant šiuo darbu genetiniai algoritmai buvo sėkmingai išbandyti sprendžiant įvairius optimizavimo uždavinius (pvz.: tvarkaraščių sudarymo problema, optimalios santuokos datos problema, optimalios dietos problema).

Šie algoritmai simuliuoja gamtinę genetinę sistemą (t.y. imitavo natūralų atrankos procesą). Pradžioje pagrindinės sąvokos, kurios buvo naudojamos aprašant genetinius algoritmus, buvo skoliniai iš biologijos evoliucijų procesų – *individas* (elementarus, paprastai neskaidomas vienetas) ir *populiacija* (individų grupė). Dar vienas ne mažiau svarbus dalykas genetiniuose algoritmuose yra *individo tinkamumas* (laipsnis, kaip individas gali prisitaikyti prie aplinkos, išgyventi ir reprodukuotis). Tačiau po J.H. Hollan knygos išleidimo buvo pradėti naudoti daugiau matematikoje/informatikoje priimtini terminai – vietoj individo dabar naudojama sprendinys (dažniausiai atvaizduojamas simbolių eilutėmis – chromosomomis,

kuriose kiekvienas funkcijos nežinomas užkoduojamas simboliu, vaizduojančiu geną), vietoje populiacijos naudojama sprendinių aibė, individo tinkamumas asocijuojamas su tikslo funkcijos reikšme duotajam sprendiniui.

Ieškant optimalių sprendimų operuojama su vienu ar daugiau sprendinių, o tolimesniam nagrinėjimui visada pasirenkamas tas, kuris turi tinkamesnes rezultatus gauti funkcijos reikšmes (atrankos procedūra). Viduje metodo naudojamos euristinės procedūros, skirtos naujiems sprendiniams formuoti, dalinant sprendinių aibes į sprendinių poras ir operuojant su šiomis poromis (kryžminimas). Nauji sprendiniai yra generuojami iš anksčiau gautų sprendinių (mutacija). Dėl šių savybių realizuojant metodo veikimą, sprendinių aibę galima atnaujinti iteraciniu būdu.

Genetinio algoritmo įgyvendinimui yra žinoma labai daug būdų, kaip turėtų būti atliekami pradinų sprendinių parinkimo, atrankos, kryžminimo ir mutacijos procedūros (pvz. su viena ar keliomis populiacijomis, su visomis kartu ar tam tikromis grupėmis, galima atrinkti pradinius sprendinius pagal jų *individo tinkamumą*). Šie metodai gali būti vykdomi kartu arba kiekvienas atskirai. *Genetinio* algoritmo etapai:

1. Sudaroma pradinė populiacija: dažniausiai pasirenkama atsitiktinai, tačiau jeigu sprendžiamas labai sudėtingas uždavinys, rekomenduojama naudoti iš anksto apibrėžtą euristinį metodą (pvz. lokalsios paieškos). Pradinė populiacija saugoma chromosomomis, kurios dažniausiai realizuojamos dvejetainėmis arba sveikųjų skaičių koduotėmis;
2. Sukryžminamos chromosomos: naudojant pasirinktą kryžminimo tipą, dažniausiai naudojama dviejų „tėvų“ vieno taško kryžminimo schema (pateikta žemiau esančiame paveikslėlyje), tačiau galimas ir kelis kartus vykdomas kryžminimas su skirtingais „tėvais“ ar skirtingu kiekiu taškų;



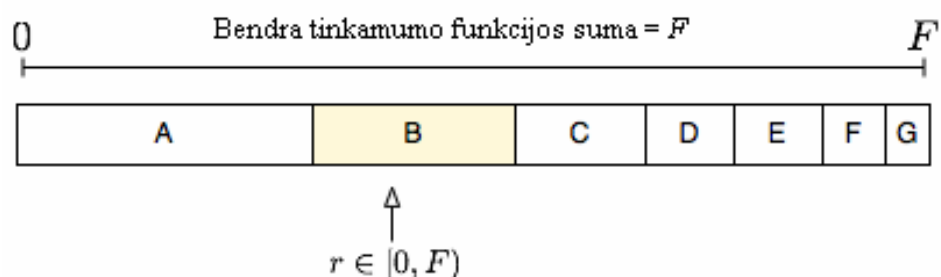
7 PAVEIKSLĖLIS. VIENATAŠKIS IR DVITAŠKIS KRYŽMINIMAS

3. Chromosomoje vykdoma mutacija: vienu metu vyksta tik vienoje chromosomoje, dažniausiai pakeičiamas vienas chromosomos genas arba jo pozicija. Pritaikoma visoms chromosomoms su tam tikra mutacijos tikimybe. Pagrindinis mutavimo uždavinys – perrinkti įvairius sprendinius arba sukurti tokių, kurių neįmanoma gauti kryžminimo

būdu. Tačiau įvedant mutacijos tikimybę yra rizika, kad mutacija sukels chaosą ir/ar išsigimimą populiacijoje;

4. Vykdoma atranka: sudaroma tarpinė sprendinių aibė, iš kurios išrenkami sprendiniai, kurie toliau dalyvaus procese. Tolesnį chromosomos dalyvavimą procese apsprendžia chromosomą atitinkanti tikslo funkcijos reikšmė, t.y. labiau tikėtina, kad sprendiniai su geresnėmis tikslo funkcijų reikšmėmis pateks į tolesnę sprendinių aibę. Pagrindiniai naudojami atrankų būdai:

- Turnyrinė atranka: atsitiktinai iš visų chromosomų pasirenkama t chromosomų, iš kurių išrenkama geriausia, jas lyginant. Tokios atrankos metu populiacijos nereikia rūšiuoti, todėl šis algoritmas yra $O(n)$ sudėtingumo ir yra lengvai realizuojamas (n – individų skaičius populiacijoje).
- Proporcinga atranka: paremta principu, kad galimybė toliau dalyvauti procese kiekvienai chromosomai yra tiesiogiai (arba pasirinktinai atvirkščiai) proporcinga jį atitinkančios tikslo funkcijos reikšmei. Šioje atrankoje yra tikėtina galimybė atrinkti tolimesniam procesui ir pačius blogiausius sprendinius.



8 paveikslėlis. Proporcinga atranka *genetiniame algoritme*

- Elito atranka: dažniausiai naudojamas atrankos būdas, ignoruojantis įprastą eigą, tačiau gaunantis efektyviausią variantą, kai į tolimesnę eigą pereiti leidžiama kelioms pačioms geriausioms pagal tikslo funkcijos reikšmę chromosomoms visiškai nepaveiktoms nei kryžminimo, nei mutacijos.
5. Vykdomi 2-4 etapai, kol gaunamas tenkinantis sprendimas, siekiant išvengti klonavimo problemos (dėl kryžminimo ir mutacijos yra tikimybė, kad gali atsirasti vienodų chromosomų). Tinkamai parinkus metodo parametrus (tėvų ir taškų kiekis kryžminime bei mutacijos tikimybė) šios problemos galima išvengti. Svarbiausias šio etapo tikslas – tinkamai nustatyti nutraukimo sąlygas. Dažniausios sąlygos yra šios:
- Pasiektas maksimalus iteracijų skaičius;
 - Išnaudoti tam skirti pinigai ar laikas;
 - Pasiakta stabilizacijos būseną, kai nebegaunama geresnių sprendinių;
 - Dėl aukščiau minėtų priežasčių kombinacijos.

2.4. Bayes'o metodas

Bayes'o teorema – tikimybių teorijos teorema, kuri nustato įvykio tikimybę, kai žinoma tik dalis informacijos apie įvykius (stebėjimo metu). Kitaip tariant, T. Bayes įrodė, kaip galima tiksliau nustatyti tikimybę, imant iš anksto žinomą informaciją ir naujų stebėjimų duomenis. (Augutis, et al., 2004) (Augutis, et al., 2004). Plačiai žinoma Bayes'o formulė:

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}, \text{ kur:}$$

$\Pr(A)$ – apriorinė (žinoma prieš eksperimentą) A hipotezės tikimybė;

$\Pr(A/B)$ – A hipotezės tikimybė, atsirandant B įvykiui (aposteriorinė tikimybė);

$\Pr(B/A)$ - įvykio B atsiradimo tikimybė esant teisingai A hipotezei;

$\Pr(B)$ – įvykio B tikimybė.

Bayes algoritmą išsamiai nagrinėjo prof. J. Mockus knygoje „A set of examples of global and discrete optimization 2“ (Mockus, 2004). Tradicinė skaitmeninė analizė optimizavimo metodams yra tiksli visoms juose optimizuojams funkcijoms. Ji apima konkrečius metodus bei gerai tinka analizuojant sudėtingus atvejus. Norint išvengti didelio klaidų kiekio, reikia atlikti daug bandymų. Tačiau tokių bandymų kiekis didėja eksponentiškai didėjant uždavinio sudėtingumui. Tai ir yra pagrindinė kliūtis nagrinėjant sudėtingus atvejus. Tačiau, kaip alternatyva, yra naudojama vidurkių analizė. Čia klaidų vidurkis nėra ribotas, bet, kiek įmanoma, sumažinamas. Kaip rezultatas yra imamas optimizuojamų funkcijų aibės vidurkis. Vidurkių analizė yra vadinama *Bayes'o priartėjimu* (angl. *Bayesian Approach, BA*) (Mockus, et al., 2000), (Mockus, 2004), (Pupeikienė, 2009).

Yra du būdai, kaip pritaikyti Bayes'o teoremą optimizacijoje:

- *Tiesioginis Bayes'o metodas* (angl. *Bayes approach, BA*) - apibrėžiamas fiksuojant svarbesnius pasiskirstymus P funkcijų $f(x)$ aibėje bei minimizuojant Bayes rizikos funkcija $R(x)$ (Mockus, 2004) (Mockus, et al., 2000). Rizikos funkcija $R(x)$ yra tikėtinas nuokrypis nuo globalaus minimumo fiksuotame taške x . Čia pasiskirstymas P sudarytas kaip stochastinis modelis iš $f(x)$, $x \in R^m$, o $f(x)$ gali būti determinuota arba stochastinė funkcija. *Tiesioginio Bayes'o algoritmo* tikslas, skirtas rasti kiek įmanoma mažesnes vidurkių klaidas, išlaikant gerėjimo sąlygas. Naudojamas tolydžiuose atvejuose.
- *Bayes euristinis priartėjimas* (angl. *Bayesian Heuristic Approach, BHA*). Jis fiksuoja svarbesnius pasiskirstymus P pagalbinių funkcijų $f_K(x)$ aibėje. Šios funkcijos apibrėžia geriausias reikšmes, įgytas K kartu panaudojus euristika $h(x)$. Kur euristika $h(x)$ priklauso nuo tolydžių parametrų $x \in R^m$. Ši euristika taip pat padeda optimizuoti pradinę funkciją $C(y)$, kuri yra sudaryta iš kintamųjų $y \in R^n$,

čia $m < n$. Kaip paprastai komponentai y yra diskretūs kintamieji, todėl naudojama diskrečiuose atvejuose. (Mockus, 2004), (Pupeikienė, 2009).

Tolesniame skyriuje nagrinėjama Bayes'o metodo ir Bayes'o metodo papildyto euristikomis metodų skirtumai bei panašumai.

2.5. Bayes'o metodas papildytas euristikomis

Prieš nagrinėjant Bayes metodus, mokslo srityje buvo nagrinėjami Wiener procesai. Apie juos rašė prof. J. Mockus (Mockus, 2004) bei L. Pupeikienė (Pupeikienė, 2009). Wiener modelis numato beveik visų paprastų funkcijų $f(x)$ tolydumus. Taip pat šis modelis numato, kad padidėjimas $f(x_4) - f(x_3)$ ir $f(x_2) - f(x_1)$, $x_1 < x_2 < x_3 < x_4$ yra stochastiškai nepriklausomi. Čia $f(x)$ priklauso Gauso $(0, \sigma x)$ bet kokiam fiksuotam taškui $x > 0$.

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

Reguliarios suderintos sąlygos kaitaliojamos, kai:

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

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

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

Pagrindinis tiesioginio Bayes'o metodo tikslas yra sumažinti numatomą nuokrypį. Tiesioginis Bayes metodas taip pat yra papildytas tokiais asimptotiniais parametrais:

$$d^*/d_a = \left(\frac{f_a - f^* + \epsilon}{\epsilon} \right)^{1/2}, n \rightarrow \infty, \text{ kur:}$$

d^* – x_i taškų atstumas nuo globalaus optimumo;

d_a – yra x_i taškų, esančių nuo globalaus optimumo, atstumų vidurkis galimoje srityje;

f_a – yra funkcijos $f(x)$ vidurkis šioje srityje;

f^* – funkcijos $f(x)$ vidurkis prie globalaus minimumo;

ϵ – koregavimo parametras.

Tai reiškia, kad tiesioginis Bayes'o metodas užtikrina kiekvienos funkcijos $f(x)$ artėjimą į globalų minimumą. Jis taip pat užtikrina prie globalaus minimumo stebėjimų srityje esančių

taškų x_i didesnį skaičių, ypač jei n yra didelis. Koregavimo parametras ϵ turi panašią įtaką kaip temperatūra atkaitinimo modeliavimo metode. Tačiau *Bayes'o* ir *atkaitinimo modeliavimo* metodai panašūs tik išoriškai.

Bayes'o metodo atveju paprastai asimptotiniai parametrai praktiškai neįtakoja sprendinio, nes *Bayes'o algoritmas* yra pritaikytas mažos apimties sprendimuose. Pasirinkus tašką x_{n+1} , tikėtino nuokrypio $R(x)$ minimizavimas tampa papildoma ir sudėtinga optimizavimo problema. Tai reiškia, kad tiesioginis *Bayes'o* metodas gali būti naudojamas, kai kintamųjų $m < 20$ (Pupeikienė, 2009).

Tačiau dirbant su žaliavų taupymo problemos uždaviniu, m yra gerokai didesnis. Todėl buvo nutarta naudoti *euristinį Bayes'o metodą*. Jis naudojamas optimizuoti paprastoms objektinėms funkcijoms, turinčioms daug kintamųjų. Paprastai problemos sprendžiamos naudojant euristicas, kurios sudaromos vadovaujantis tos srities specialisto nuomone. Dažniausia euristicos siejasi su atsitiktinio maišymo procedūromis, kurios priklauso nuo tam tikro empiriškai apibrėžto parametro. (Mockus, 2004) Tokiais parametrais gali būti:

1. Pradinė temperatūra *modeliuojamo atkaitinimo metode*;
2. Skirtingų atsitiktinio maišymo algoritmų tikimybė.

Apimdamas šias problemas, tiesioginis *Bayes'o metodas* tampa tinkamu įrankiu, apimančiu įvairių euristicinių parametrų ir jų maišymo optimizavimą. (Mockus, 2004)

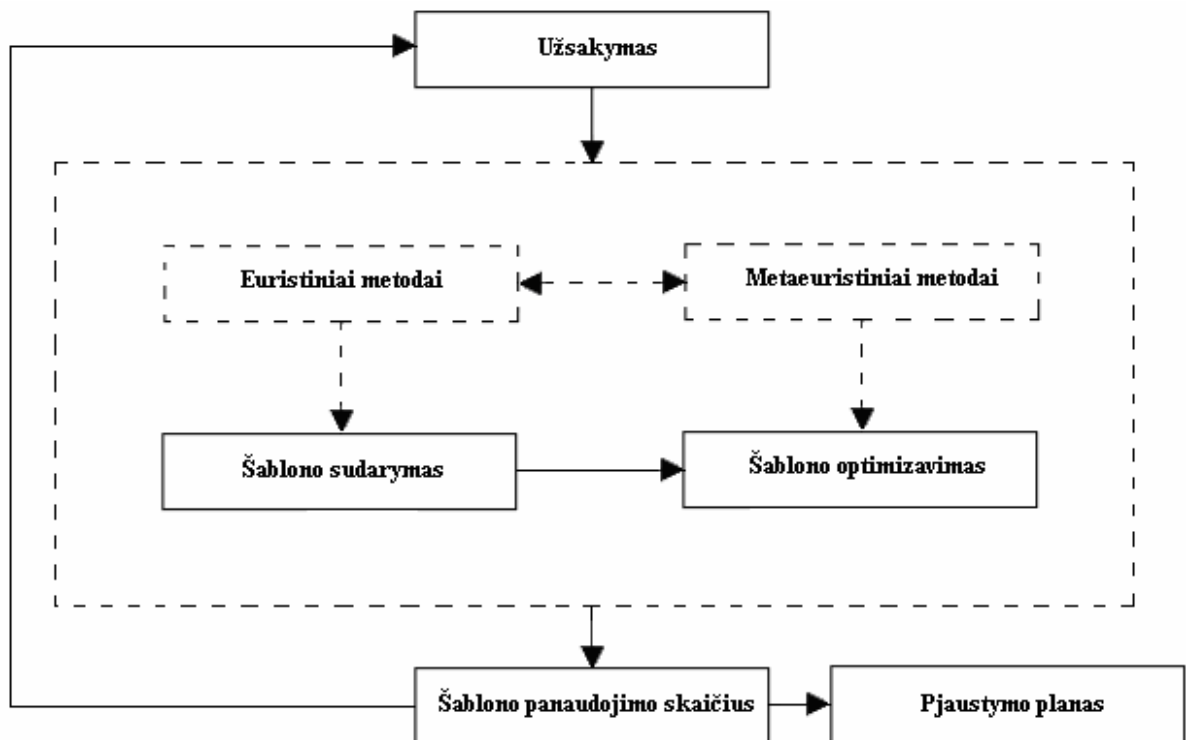
Taigi toks *Bayes'o metodas* yra vadinamas *euristiniu Bayes'o metodu* (angl. *Bayesian Heuristic Approach, BHA*). Šis *euristinis Bayes'o metodas* buvo naudojamas optimizuojant *genetinio algoritmo hibridizuoto su žemiausio kairėn* ir *genetinio algoritmo hibridizuoto su žemiausio kairėn užpildymo algoritmu* parametrus.

3. Žaliavų taupymo problemos uždavinio sprendimo schema. Tiriamų metodų pritaikymas šiam uždaviniui

3.1. Žaliavų taupymo problemos uždavinio sprendimo schema

Žaliavų taupymo problema priklauso *NP sunkių uždavinių* klasei, o tai reiškia, kad nėra greito algoritmo, kuris visais atvejais pateiktų globaliai optimalų sprendinį, todėl uždavinio sprendimas bus vykdomas apytikslių – euristinių ir metaeuristinių metodų pagalba.

Naudosime J. Pokšto sukurtą uždavinio sprendimo schemą, pateiktą žemiau esančiame paveikslėlyje (Pokštas, 2007).



9 paveikslėlis. Bendra pjaustymo uždavinio sprendimo schema

Duotam figūrų užsakymui išpjauti, sudarysime *pjaustymo šablonus*. *Pjaustymo planas* yra visų sudarytų šablonų visuma, kuri reprezentuoja visą užduoties sprendimą duotajam užsakymui. Pjaustymo plane nurodoma kiek, kaip ir kokius stačiakampius išpjauti. Jame visada galioja sąlyga, kad atliekų kiekis yra minimalus ir jame išpjaunama dalis ar visi stačiakampiai. Tai priklausys nuo to, koks bus pasirinktas pradinis lakštų ir pjaunamų stačiakampių kiekis, ar medžiagų pakankamai ar yra jų trūkumas ir kaip sėkmingai pavyks išspręsti uždavinį.

Šablono sudarymo užduotį vykdys du pasirinkti euristiniai metodai – *žemiausio kairėn* ir *žemiausio kairėn užpildymo*. Todėl dažnu atveju pats optimaliausias sprendinys gali būti nerastas. Dauguma euristinių pjaustymo metodų nedaro jokio sprendinių perrinkimo, nenagrinėja jų paieškos erdvės ir nelygina jų tarpusavyje, todėl šių metodų negalime vadinti optimizavimo

algoritmais. Tačiau optimaliausio sprendimo radime jie turi didelę reikšmę - jų pagalba konstruojamas pats sprendinys ir pateikiamas jo kokybės įvertinimas.

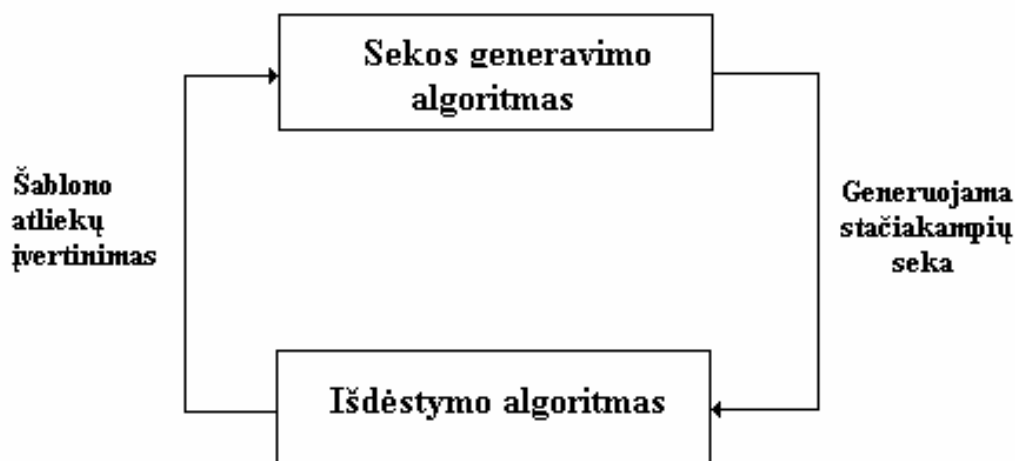
Dėl minėtų euristinių pjaustymo algoritmų ypatumų, šablono optimizavimui pasirinktas metaeuristinis metodas, kuris vykdo sprendinių perrinkimą ir palyginimą, taip pat, yra universalus ir tinka ne būtinai pjaustymo užduotims spręsti. Kadangi euristiniai metodai labai specifiški, o pasirinktas metaeuristinis metodas pakankamai universalus optimaliam sprendiniui gauti, juos apjungėme. Ypač ir dėl to, kad šiame darbe pasirinktas metaeuristinis genetinis algoritmas pats savarankiškai užduoties išspręsti negali. Tuo tarpu, pasirinktieji euristiniai tai padaro laisvai.

3.2. Hibridizuotų genetinių algoritmų veikimo principas

Hibridizuoto genetinio algoritmo veikimo principas grįstas dviem žingsniais:

1. Naudojantis genetiniu algoritmu generuojama stačiakampių seka;
2. Naudojantis euristiniu metodu (šiuo atveju – žemiausio kairėn ir žemiausio tarpo užpildymo) stačiakampiai išdėstomi ant medžiagos lakšto.

Kaip matome, šiuos du žingsnius vykdys skirtingi algoritmai, tačiau finale gaunamas pjaustymo šablonas yra abiejų algoritmų rezultatas, todėl šiuos metodus būtina hibridizuoti. Šio metodo veikimo schema nubrėžta 10 paveikslėlyje. Kur sekos generavimo algoritmas – genetinis, o išdėstymo algoritmas - žemiausio tarpo arba žemiausio tarpo užpildymo (priklausomai nuo to su koku algoritmu genetinis algoritmas buvo hibridizuotas).



10 paveikslėlis. Hibridinio genetinio algoritmo principas

Šie algoritmai taip pat pateiks kiekvienos sekos išdėstymo kokybinį įvertinimą, kuris apibrėžiamas formule:

$$\left(\sum_{i=0}^{n_0} w_i \cdot l_i \right) / (W \cdot L)$$

kur:

n_0 – stačiakampių išdėstytų ant lakšto kiekis;

w_i – i-tojo stačiakampo plotis;

l_i – i-tojo stačiakampo ilgis;

W – lakšto plotis;

L – lakšto ilgis.

Taigi uždavinys bus sprendžiamas kompleksiskai apjungiant (hibridizuojant) tam tikrus aukščiau išvardintus algoritmus. Šiame darbe nagrinėjami du hibridiniai algoritmai – genetinis algoritmas, apjungtas kartu su žemiausio kairėn algoritmu ir genetinis algoritmas, apjungtas su žemiausio kairėn užpildymo algoritmu.

3.3. Hibridizuoto genetinio algoritmo realizacija

Kaip minėta anksčiau, genetinio algoritmo įgyvendinimo schemos gali būti labai įvairios, todėl tinka daugeliui uždavinių spręsti. Algoritmo realizacijai buvo naudojamas J.Pokšto pritaikymas prieš tai apibrėžtam pjaustymo uždaviniui (Pokštas, 2007).

Chromosomas sudaro genai, todėl šiuo atveju genų atitinkantys stačiakampiai yra indeksuojami ir kiekvienas stačiakampo indeksas atitinka tam tikrus genus chromosomoje. Kiekviena genų seka chromosomoje reiškia ne ką kitą, bet stačiakampių išplovimo iš medžiagos tvarką – eiliškumą. Todėl norint suformuoti tam tikrą sprendinį, keičiama genų seka, t.y. genai sukeičiami vietomis arba perkeliami per tam tikrą genų skaičių į vieną arba kitą pusę. Taip suformuojama chromosoma, kurią paskui interpretuoja euristinis algoritmas, t.y. išdėsto ir gražina atliekų kiekį, kuris yra kokybinis sprendimo įvertinimo kriterijus.

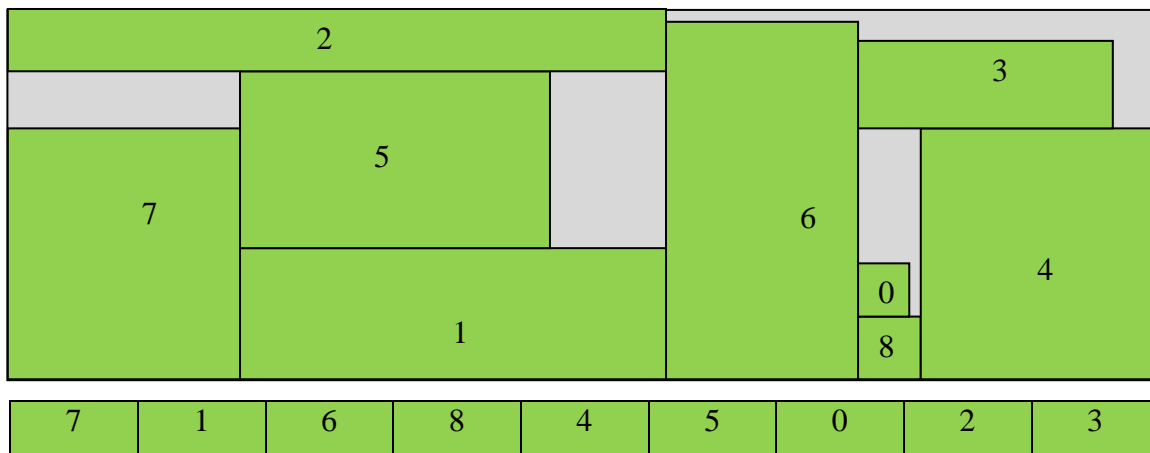
Pavyzdys: turime vieną lakštą ir 9 skirtingo dydžio stačiakampius. Genetinis algoritmas, panaudojęs kryžminimo operatorių sukūrė tokią genų seką (chromosomą), kuri pavaizduota 11 paveikslėlyje.

7	1	6	8	4	5	0	2	3
---	---	---	---	---	---	---	---	---

11 paveikslėlis. Chromosoma, kai lakštas talpina visus stačiakampius

Kaip ir minėta genų eiliškumas 11 paveikslėlyje reiškia figūrų išplovimo eiliškumą. Šia indeksų seką toliau nagrinės žemiausio kairėn arba žemiausio kairėn užpildymo algoritmai. Kadangi chromosomą mes apibrėžiame kaip tam tikrą sprendinį - pjaustymo šabloną, tai

gaunamas rezultatas yra išdėstyti stačiakampiai ant medžiagos lakšto arba, kitaip tariant, išpjauti iš to lakšto (žr. 12 paveikslėlį, kuriame žaliai pažymėti išpjauti stačiakampiai, o pilkai pažymėta nepanaudota lakšto dalis t.y. atliekos).

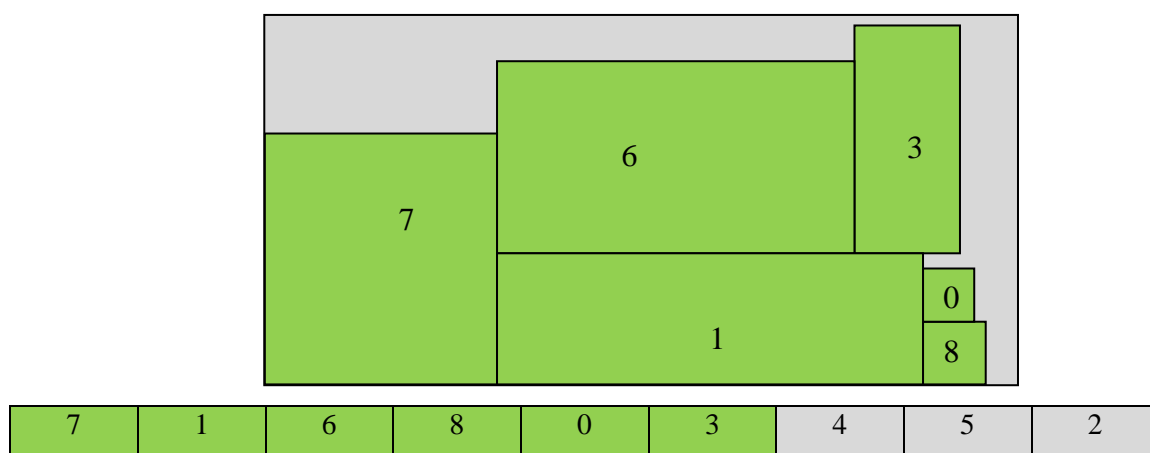


12 paveikslėlis. Chromosoma, kai lakštas talpina visus stačiakampius

Tačiau ne visada lakštai talpina visus stačiakampius, kuriuos norima išpjauti. Tokiu atveju naudojamos kintamo ilgio chromosomos, kurios turi dvi zonas:

1. Išpjautų stačiakampių indeksai;
2. Neišpjautų stačiakampių indeksai.

Tokį zonavimą naudosime kiekvienam medžiagos lakštui (žr. 13 paveikslėlį, kuriame žaliai pažymėti išpjauti stačiakampiai, o pilkai pažymėta nepanaudota lakšto dalis t.y. atliekos).



13 paveikslėlis. Chromosoma, kai lakštas netalpina visų stačiakampių

Toks chromosomos apibrėžimas naudojamas dėl kryžminimo ir mutacijos operatorių logikos supaprastinimo.

Hibridizuoto genetinio algoritmo vykdymo žingsniai:

Generuojama pradinė populiacija. Pradinę populiaciją generuojame atsitiktiniu būdu. Tai dažniausiai pasitaikantis genetinių algoritmų panaudojimo praktikoje būdas - pradinės populiacijos individai paprasčiausiai konstruojami mutacijos operatoriaus pagalba.

Vykdoma elitinė atranka. Populiacija išrikiuojama pagal individų tinkamumą (užimamo ploto mažėjimo kryptimi) pagal tai, koks yra elitinės atrankos buferio dydis, atitinkamai perkeliama skaičius individų, kurie yra kaip kandidatai į naują populiaciją be jokių pakeitimų (t.y. be mutacijos ir kryžminimo).

Vykdomas kryžminimas. Parenkamos dvi chromosomos kryžminimui, paeiliui einama per jų genus ir vykdomas antrojo tėvo geno įterpimas su iš Bayes metodo gauta tikimybe. Paprastai generuojamas skaičius $x \sim T[0,1]$, kai x mažiau už tikimybę, tai įterpimas vykdomas. Vaikas, kuris bus kandidatas patekti į naują populiaciją, formuojamas pagal šias taisykles:

1. Įvyksta įterpimo įvykis, nustatoma įterpimo pozicija;
2. Tam, kad išvengti dublikatų, įterpiamo indekso reikšmė pašalinama iš pirmojo tėvo;
3. Antrojo tėvo reikšmė įterpiama pirmajam pagal nustatytą poziciją;
4. Taip modifikuotas pirmasis tėvas tampa vaiku.

Ši veiksmų seka pavaizduota 14 paveikslėlyje. Pirmojo tėvo chromosomoje raudona spalva pažymėtas genas bus perkeliamas į antrojo tėvo chromosomoje mėlynai pažymėto geno vietą. Žaliai pažymėti genai, kurie būtų išpjauti, naudojant tokį pjaustymo planą.

Pirmas tėvas:	10	7	5	6	8	0	3	4	1	2	9
Antras tėvas:	5	0	8	3	4	2	1	9	6	10	7
Vaikas:	10	7	8	5	6	0	3	4	1	2	9

14 paveikslėlis. Kryžminimo schema

Vykdoma mutacija. Mutacijos pagrindas – dviejų atsitiktinai pasirinktų genų sukeitimas vietomis. Genus galima pasirinkti iš abiejų chromosomos zonų, t.y. ir iš išpjautų ir neišpjautų stačiakampių zonų. Genų mutacijos schema pavaizduota 15 paveikslėlyje. Mėlynai pažymėti genai, kuriems vykdoma mutacija t.y. genų sukeitimas vietomis chromosomoje.

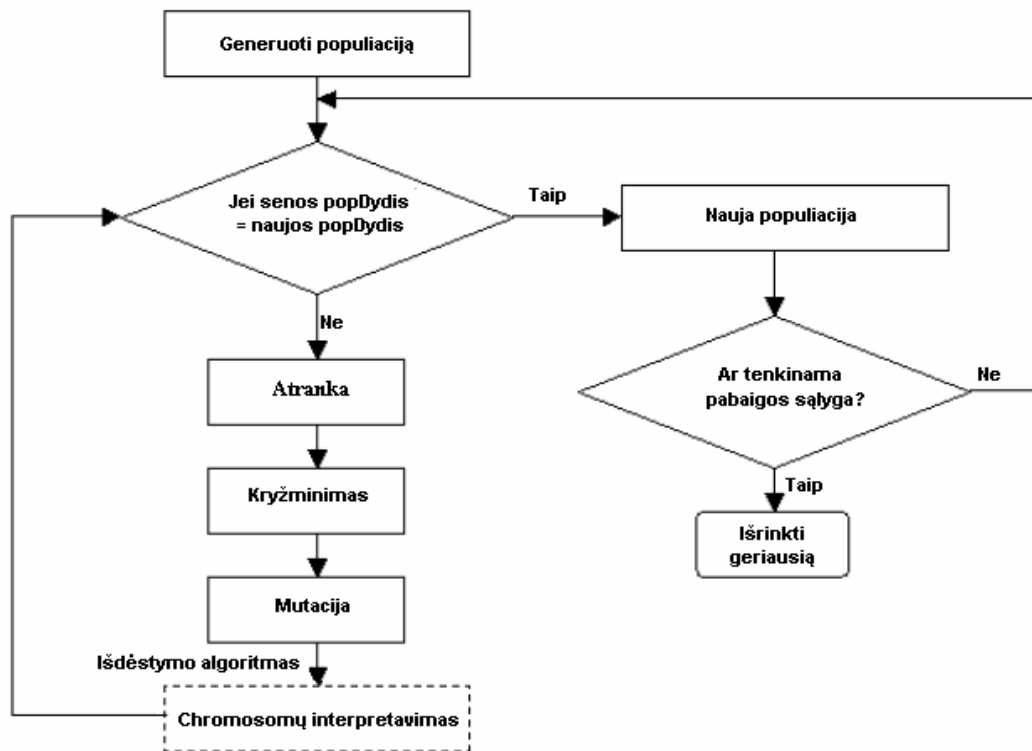
10	7	5	6	8	0	3	4	1	2	9
10	4	5	6	8	0	3	7	1	2	9

15 paveikslėlis. Mutacijos schema

Jeigu nekinta populiacijos dydis, tuomet visi „tėvinės“ populiacijos genai yra vėl išrikiuojami pagal jų užimamą plotą. Tuomet pirmieji *popDyd* nariai tampa naujos populiacijos nariais. Čia *popDydis* – populiacijos dydis, su kuria bus vykdomi skaičiavimai euristiniame algoritme.

Algoritmo vykdymas baigiamas po tam tikro iš anksto nustatyto epochų skaičiaus arba, kai surandamas šablonas, kurio atliekų kiekis lygus 0.

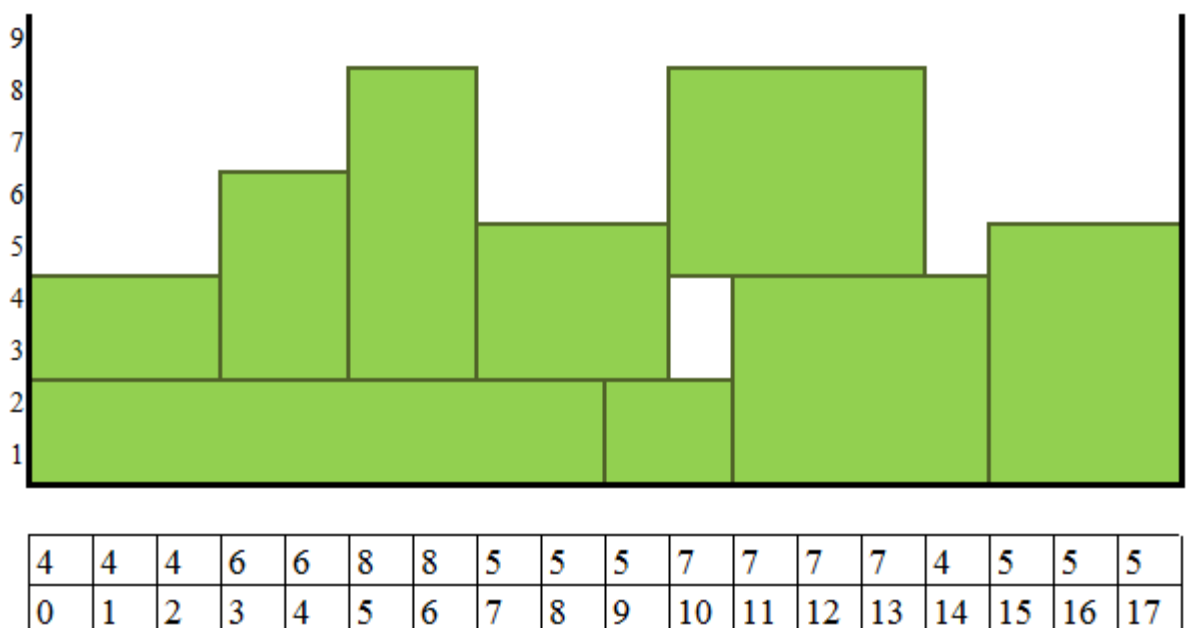
Bendra šio algoritmo veikimo schema pavaizduota 16 paveikslėlyje.



16 paveikslėlis. Bendra genetinio algoritmo schema

3.4. Žemiausio kairėn algoritmas

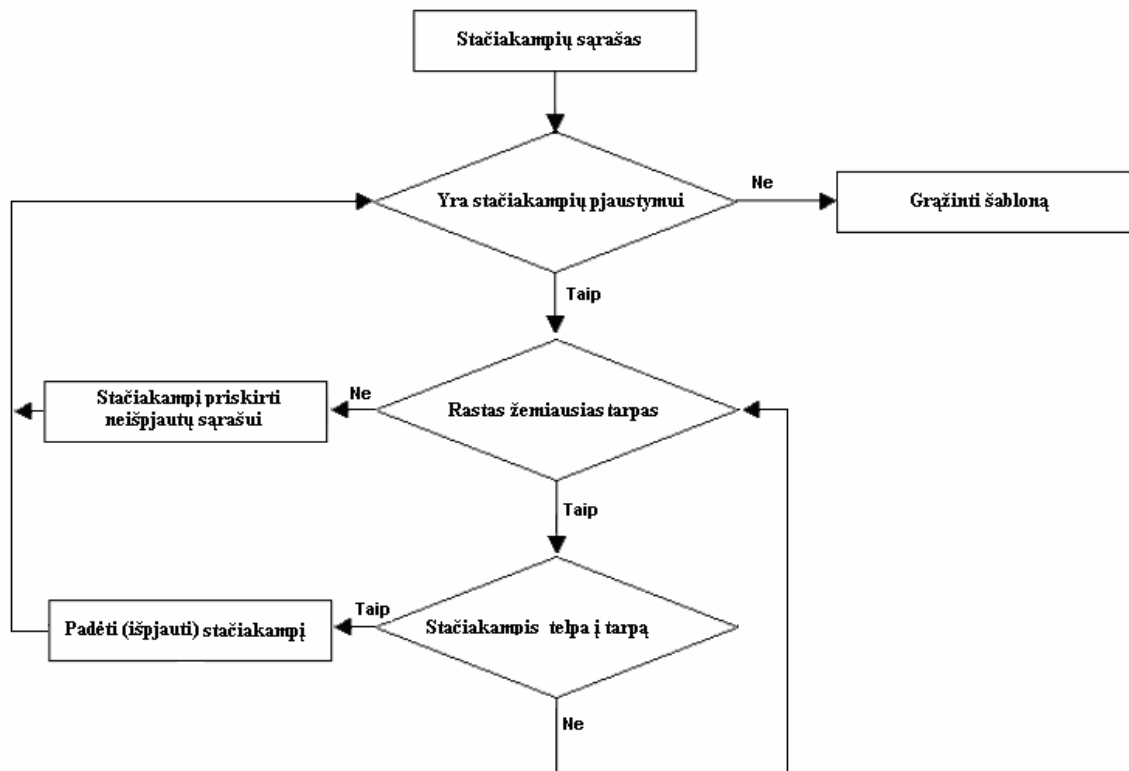
Žemiausio kairėn algoritme medžiagos reprezentavimui panaudotas vienmatis masyvas, kurį toliau vadinsime lygių masyvu. Kiekvieno lygių masyvo elemento indeksas lygus koordinatei x , o pačio elemento reikšmė yra lygi jau išdėstytų detalių aukščiui koordinatei x atžvilgiu (koordinatėi y). Lygių masyvas pavaizduotas 17 paveikslėlyje. Žalia spalva pažymėti išpjauti iš lakšto stačiakampiai.



17 paveikslėlis. Lygių masyvo samprata

Taigi žemiausio tarpo koordinatę galime rasti, nustačius mažiausio lygių masyvo elemento reikšmės indeksą. Tarpo ilgis randamas suskaičiavus, kiek paeiliui yra viena kitai lygių masyvo elementų reikšmių. Remiantis 17 paveikslėliu, matome, kad žemiausias kairiausias kampas įterpimui yra kurio $x = 0$, o tarpo ilgis lygus 3. Šis metodas nemato dešimtame stulpelyje esančio tarpo. Šis tarpas jau priskirtas į atliekas.

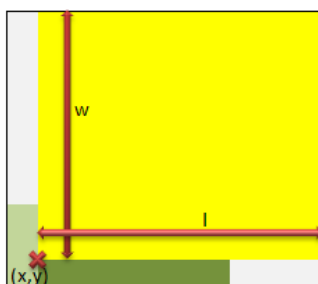
Bendra šio algoritmo veikimo schema pavaizduota 18 paveikslėlyje.



18 paveikslėlis. Žemiausio kairėn algoritmo schema

3.5. Žemiausio kairėn užpildymo algoritmas

Algoritmo vykdymo metu nagrinėjamas taškas, kur galima būtų idėti stačiakampį sąrašas, pradedant nuo žemiausio ir kairiausio „taško“. Kiekvienas „taškas“ aprašomas savo kairiausio žemiausio taško koordinatėmis (x,y) , ilgiu l ir pločiu w , kurie apibrėžia laisvą erdvę, į kurią galima idėti stačiakampį. Taško samprata pavaizduota 19 paveikslėlyje.



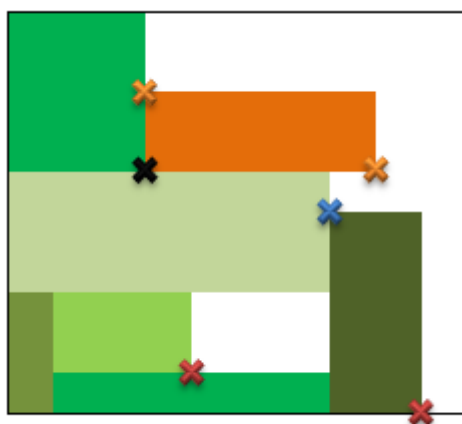
19 paveikslėlis. „Taško“ samprata

19 paveikslėlyje raudonai pažymėta kairiausias žemiausias „taško“ taškas, kurio koordinatės ir yra nurodomos kaip „taško“ koordinatės. Geltonai – visas „taškas“. Žaliai pažymėti iki šiol jau išpjauti stačiakampiai. Suradus kairiausią žemiausią „tašką“, tikrinama ar stačiakampis telpa į tą tarpą, jeigu ne, ieškomas toks tarpas, į kurį šis stačiakampis tilptų. Algoritmo vykdymo pabaigoje apskaičiuojamos atliekos pagal formulę:

$$\left(\sum_{i=0}^{n_0} w_i \cdot l_i \right) / (W \cdot L)$$

kuri jau aptarta 3.1 skyrelyje.

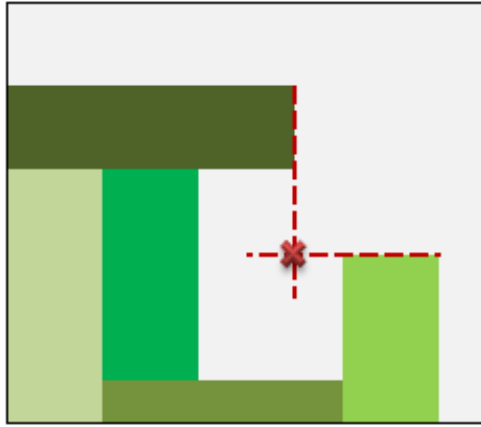
Po kiekvieno stačiakampio įdėjimo (išpjovimo) vykdomas taškų sarašo atnaujinimas (žr. 20 paveikslėlį). Pirmiausia randamas stačiakampio kairys viršutinis ir dešinys apatinis, tuomet jiems atitinkamai randami apatinis ir viršutinis taškai. Šie taškai pažymėti oranžine spalva, bus įtraukti į taškų sarašą, jei prieš tai jame dar nebuvo. Žaliu fonu nuspalvintas stačiakampiai, kuris bus išpjautas einamuoju momentu. Oranžiniu fonu nuspalvintas stačiakampis, kuris pjaunamas einamuoju momentu. Stačiakampio atsiradimas įtakoja aplinkinius kaimyninius taškus, todėl iš sąrašo eliminuojamas juoda spalva pažymėtas taškas bei pakeičiamas mėlyna spalva pažymėto taško ilgis.



20 paveikslėlis. Taškų atnaujinimo samprata

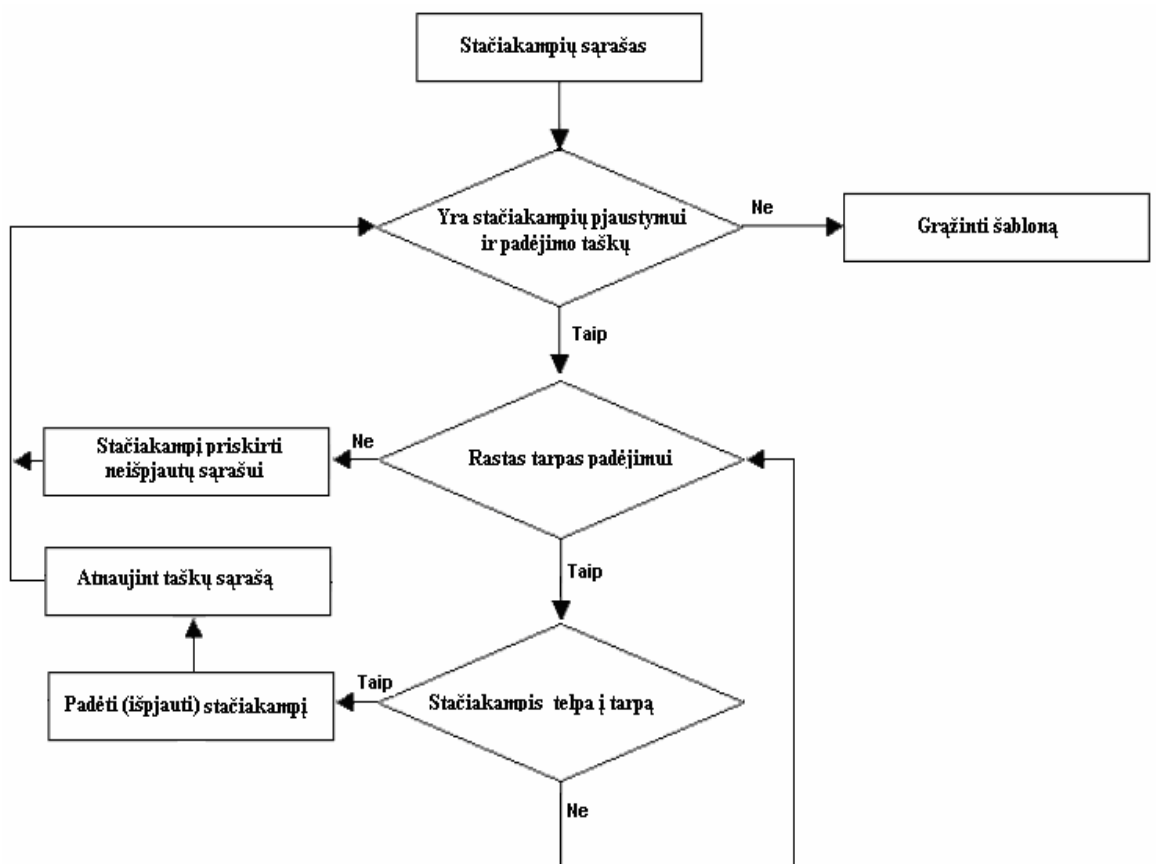
Taigi, kaip matome iš paveikslėlio algoritme naudojami keturių tipų taškai su kuriais atliekami skirtingi veiksmai. Tam, kad palengvinti taškų paiešką, po kiekvienos idėtos figūros, taškų sąrašas yra perrikiuojamas x ir y koordinatžių didėjimo kryptimi.

Tačiau egzistuoja dar vieno tipo taškas, kuris algoritme nenagrinėjamas – t.y. kabantis taškas, kuris yra pavaizduotas 21 paveikslėlyje.



21 paveikslėlis. Kabantis taškas

Bendra šio algoritmo veikimo schema pavaizduota 22 paveikslėlyje.



22 paveikslėlis. Žemiausio kairėn užpildymo veikimo schema

4. Pradinės realizacijos tyrimas ir siūlomi tobulinimai parametru optimizavimui

Relizacijos metu buvo realizuoti du hibridizuoti genetiniai algoritmai. Taip pat panaudotas Bayes'o algoritmas šių metodų parametru optimizavimui.

Sistema realizuota naudojant Java programavimo kalbą. Java priemonėmis sukurtos aplikacijos gali būti perkeltos į serverius ir pritaikytos naudojimui, nepriklausomai nuo ateityje pasirinktos techninės įrangos ar operacinės sistemos.

Tačiau, kaip ir visada, besinaudojant sistema (šiuo atveju atliekant parametru optimizavimo tyrimą) kyla naujų minčių sistemos tobulinimui. Žemiau pateikiami siūlomi sistemos patobulinimai.

4.1. Kitų metodų pritaikymas parametru optimizavimui

Šiuo metu naudojantis sistema galima algoritmų optimizavimo parametrus galima ištirti tik naudojant Bayes'o algoritmą. Todėl rekomenduojama papildyti parametru optimizavimą kitokiais metodais.

4.2. Sprendinių aibės išplėtimas

Kadangi žaliavų pjaustymo uždavinys aktualus ne vien stačiakampių išpjovimui iš lakštų, tačiau ir kitokių formų objektams, todėl rekomenduojama išplėsti sprendinių aibę. T.y. ją papildyti naujais galimais sprendiniais – apskritimais, trikampaiais ar kitokiomis figūromis.

4.3. Sistemos vartotojo sąsajos tobulinimas

Ne visi žmonės yra linkę naudotis informacinių technologijų teikiamomis naujovėmis (ypač tie, kurie iki šiol nesinaudojo kompiuterinėmis sistemomis savo darbe), todėl sistemų kūrėjai turėtų orientotis ne tik į sistemos funkcionalumą, bet ir sistemos sąsajos patrauklumą, sudėtingumą, patogumą vartotojui, vartotojų apmokymo galimybes, atnaujinimo paprastumą. Šiuo metu sistemos sąsaja ribota – siūloma:

- įvesti grafinį sistemos rezultatų atvaizdavimą vartotojui lengviau suprantama forma;
- įvesti grafinį stačiakampių įvedimą (jeigu sistemoje bus išplėsta sprendinių aibė siūloma įvesti ir papildytų figūrų įvedimą per vartotojo sąsają).

5. Parametrų optimizavimo tyrimas

Suformavus pradinį pjaustymo planą, skaičiuojamos atliekos. Šios atliekos yra perskaičiuojamos kiekvienoje iteracijoje optimizavimo metu. Pagrindiniai žaliavų taupymo problemos euristinių parametrų įverčiai yra nurodyti lakštų, stačiakampių matmenys.

Toliau vaizduojami rezultatai, atlikus po 10 bandymų su skirtingais optimizavimo metodų parametrais (*Bayes'o algoritmo* iteracijų kiekis) ir skirtingais optimizavimo metodais (*genetinis algoritmas hibridizuotas su žemiausio kairėn užpildymo metodu* ir *genetinis algoritmas hibridizuotas su žemiausio kairėn metodu*). Norint sužinoti, kuris suformuotas pjaustymo planas yra geriausias, vertinamas sunaudoto ploto procentinė dalis (toliau tyrime apibrėžiama kaip funkcija $f(x)$, kur x – iteracijų kiekis):

$$f(x) = S_{proc} = \frac{S}{W \cdot L} \cdot 100\%, \text{ kur:}$$

S – išpjovus medžiagos lakštą gautas atliekų plotas (m^2);

W – lakšto plotis;

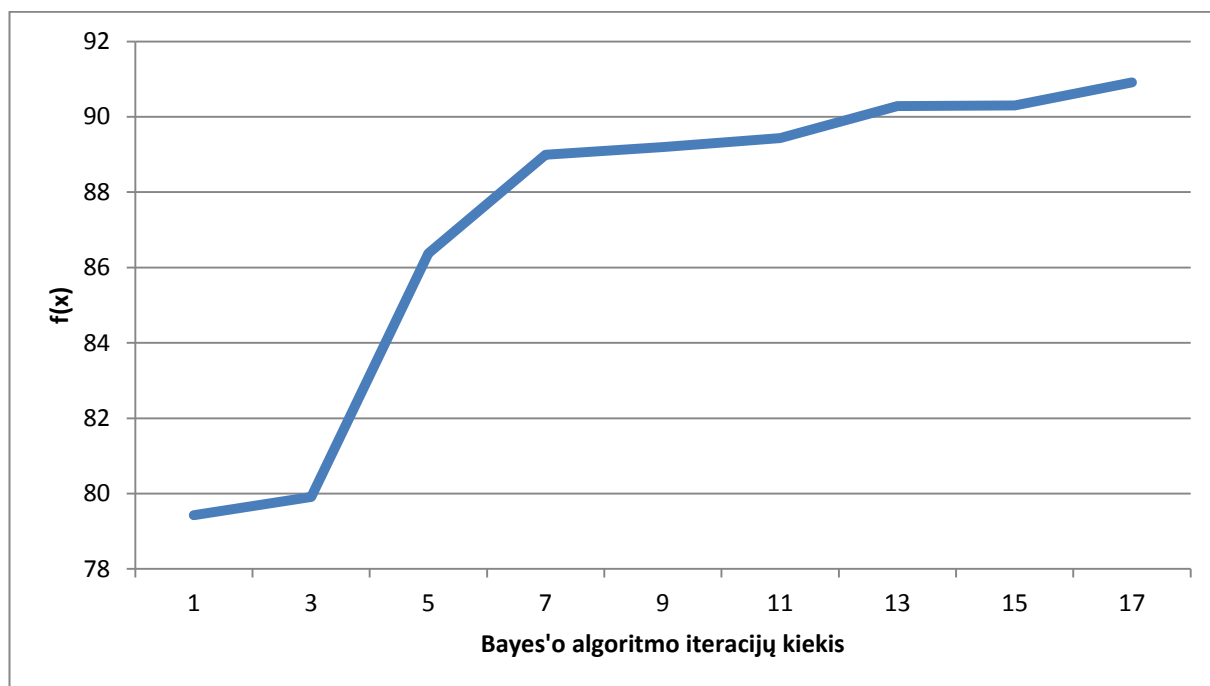
L – lakšto ilgis.

Visi tolimesni aprašyti bandymai atliekami prie tokių sąlygų:

- Genetinio algoritmo neoptimizuojami parametrai:
 - elitinės atrankos tikimybė - 0,1;
 - iteracijų (epochų) kiekis - 100;
- Bayes'o algoritmo neoptimizuojami parametrai, neturintys įtakos tyrimui:
 - vienas pirminis taškas;
- Lakšto matmenys:
 - ilgis (L) yra lygus 1000 m;
 - plotis (W) yra lygus 1000 m;
- Optimizavimui visada pateikiami 408 įvairaus dydžio stačiakampiai. Tarp kurių yra:
 - 183 stačiakampiai, kurių abi kraštinės iki 100 m;
 - 192 stačiakampiai, kurių bent viena kraštinė tarp 100 m ir 700 m;
 - 25 stačiakampiai, kurių abi kraštinės virš 700 m;
 - 8 stačiakampiai, kurių bent viena kraštinė netelpa, t.y. ilgis arba plotis didesnis nei 1000 m.

5.1. Genetinio algoritmo hibridizuoto su žemiausio kairėn algoritmu parametrų optimizavimo apžvalga

Atlikus po 10 bandymų skirtingam Bayes'o iteracijų kiekiui *genetinio algoritmo hibridizuoto su žemiausio kairėn metodu* gauti rezultatai surašyti *A priede*. Bandymų metu gauti kiekvienų iteracijų $f(x)$ bei metodo veikimo trukmės vidurkiai, pagal kuriuos nustatoma atitinkamai $f(x)$ funkcijos ir metodo veikimo trukmės priklausomybė nuo *Bayes'o algoritmo* iteracijų kiekio. Šios priklausomybės pavaizduotos grafiškai 23 ir 24 paveikslėliuose.



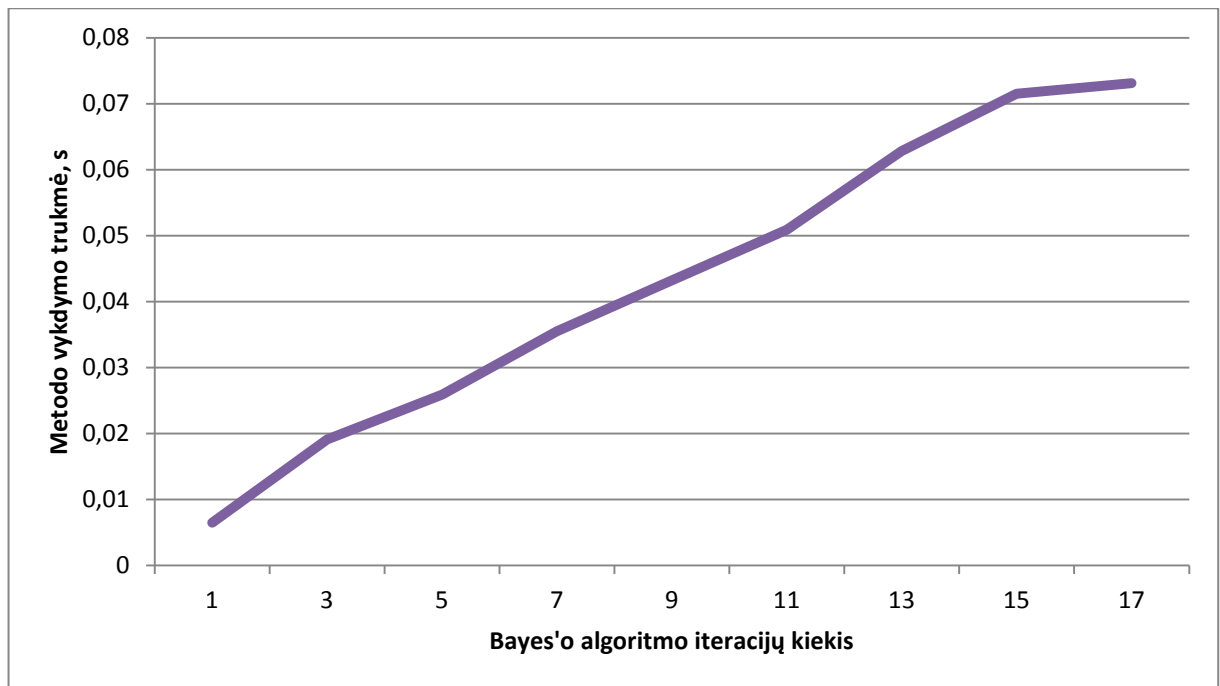
23 paveikslėlis. Genetinio algoritmo, hibridizuoto su žemiausio kairėn algoritmu, priklausomybė nuo Bayes'o algoritmo iteracijų kiekio.

Iš viršuje esančio paveikslėlio matome:

- Kuo daugiau iteracijų *Bayes'o algoritmui* yra nustatoma, tuo tikslesnis rezultatas yra randamas;
- Atlikus bandymus skirtingam iteracijų kiekiui, gautas funkcijos kitimas nėra didelis (vidutiniškas kilimas ~1,44 % iteracijų kiekiui padidėjus dviem iteracijomis žr. į 1 lentelę).

1 lentelė. Genetinio algoritmo, hibridizuoto su žemiausio kairėn algoritmu, funkcijos $f(x)$ pokyčiai kintant Bayes'o algoritmo iteracijų kiekiui

f(x) funkcijos pokytis	0,4871	6,4625	2,6155	0,2098	0,2372	0,8478	0,0187	0,6093
Bayes'o algoritmo iteracijų pokytis	1-3	3-5	5-7	7-9	9-11	11-13	13-15	15-17



24 paveikslėlis. Genetinio algoritmo, hibridizuoto su žemiausio kairėn algoritmu, veikimo trukmės priklausomybė nuo Bayes'o algoritmo iteracijų kiekio

Iš viršuje esančio paveikslėlio matome:

- Kuo daugiau iteracijų *Bayes'o algoritmui* yra nustatoma, tuo ilgiau ieškomas rezultatas;
- Algoritmo veikimo greitis kinta beveik tiesiškai;
- Metodo vykdymo trukmė, kai *Bayes'o algoritmo* iteracijų kiekis nuo 1 iki 17, yra nuo 3 iki 104 milisekundžių.

Genetinio algoritmo hibridizuoto su žemiausio kairėn algoritmu parametrų optimizavimo tyrimo rezultatai:

Šiuo metodu suformuoti rezultatai yra nevisiškai tikslūs, nes pateikiant šiuos duomenis, galima buvo gauti ir 100% užpildymą (ką įrodo tolesniame darbe tirtas *genetinis algoritmas, hibridizuotas su žemiausio kairėn užpildymo metodu*). Šio metodo rezultatai nuolat gerėja, kai *Bayes'o algoritmo* iteracijų kiekis didėja. Eksperimento metu nei vieną kartą nebuvo gautas 100% užpildymas.

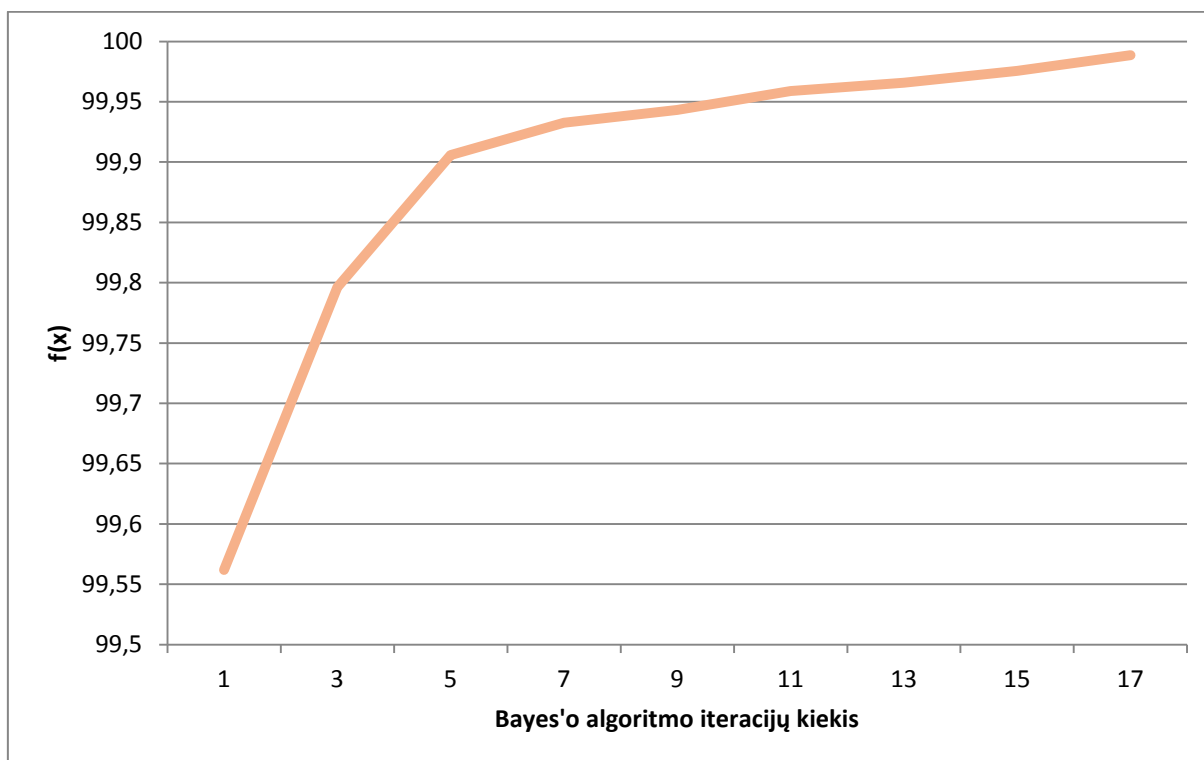
Prie optimalaus išdėstymo (t.y. šimtaprocentinio lakštų užpildymo) priartėjama tik pasirinkus didesnę nei 10 000 iteracijų kiekį.

Kuo daugiau pasirenkama *genetinio algoritmo* ir *Bayes'o algoritmo* iteracijų – tuo ilgiau reikia laukti rezultatų.

Metodas vykdomas labai greitai. Visų bandymų metu metodo vykdymo greitis neviršijo 104 milisekundžių.

5.2. Genetinio algoritmo hibridizuoto su žemiausio kairėn užpildymo algoritmu parametrų optimizavimo apžvalga

Atlikus po 10 bandymų skirtingam Bayes'o iteracijų kiekiui, *genetinio algoritmo hibridizuoto su žemiausio kairėn užpildymo metodu* gauti rezultatai surašyti B priede. Bandymų metu gauti kiekvienų iteracijų $f(x)$ bei metodo veikimo trukmės vidurkiai, pagal kuriuos nustatoma atitinkamai $f(x)$ funkcijos ir metodo veikimo trukmės priklausomybė nuo Bayes'o algoritmo iteracijų kiekio. Šios priklausomybės pavaizduotos grafiškai 25 ir 26 paveikslėliuose.



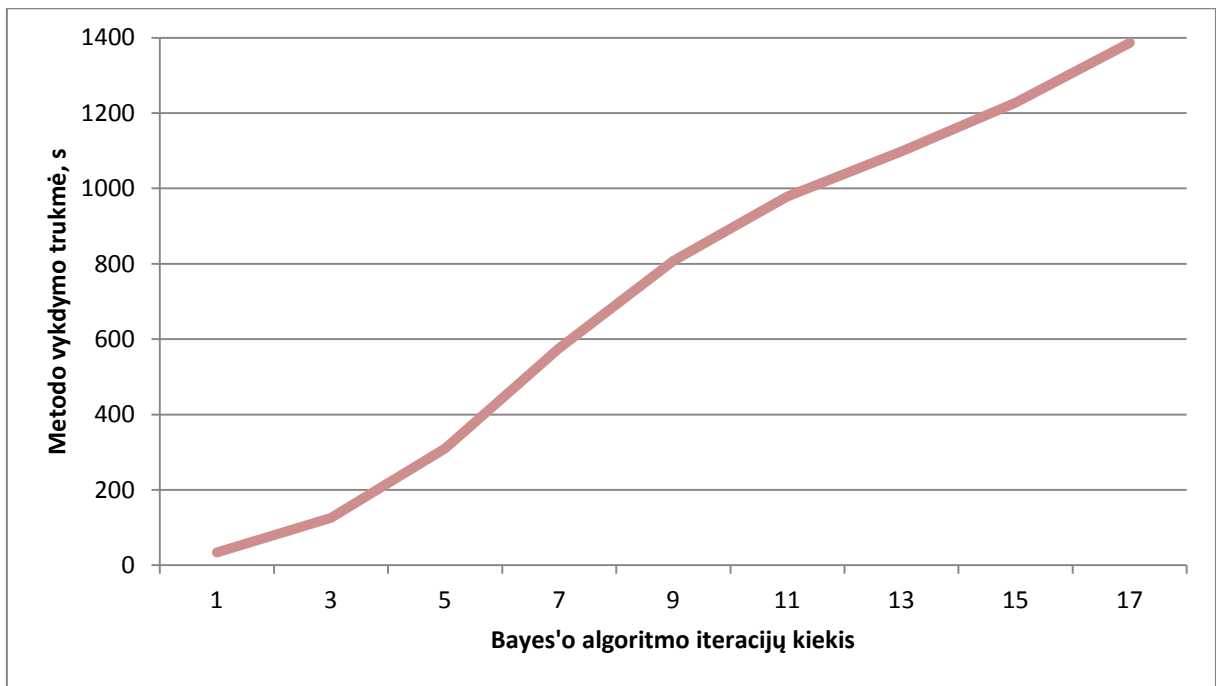
25 Paveikslėlis. Genetinio algoritmo, hibridizuoto su žemiausio kairėn užpildymo algoritmu, priklausomybė nuo Bayes'o algoritmo iteracijų kiekio.

Iš viršuje esančio paveikslėlio matome:

- Kuo daugiau iteracijų *Bayes'o algoritmui* yra nustatoma, tuo tikslesnis rezultatas yra randamas;
- Atlikus bandymus skirtingam iteracijų kiekiui, gautas funkcijos kitimas nėra didelis (vidutiniškas kilimas $\sim 0,053$ % iteracijų kiekiui padidėjus dviem iteracijomis žr. į 2 lentelę).
- Bandymų metu gauti labai tikslūs rezultatai net ir su viena *Bayes'o algoritmo* iteracija.

2 lentelė. Genetinio algoritmo, hibridizuoto su žemiausio kairėn užpildymo algoritmu, funkcijos $f(x)$ pokyčiai kintant Bayes'o algoritmo iteracijų kiekiui

f(x) funkcijos pokytis	0,2343	0,1098	0,0266	0,0106	0,0158	0,0070	0,0097	0,0129
Bayes'o algoritmo iteracijų pokytis	1-3	3-5	5-7	7-9	9-11	11-13	13-15	15-17



26 paveikslėlis. Genetinio algoritmo, hibridizuoto su žemiausio kairėn užpildymo algoritmu, veikimo trukmės priklausomybė nuo Bayes'o algoritmo iteracijų kiekio

Iš viršuje esančio paveikslėlio matome:

- Kuo daugiau iteracijų *Bayes'o algoritmui* yra nustatoma, tuo ilgiau ieškomas rezultatas;
- Algoritmo veikimo greitis kinta beveik tiesiškai;
- Metodo vykdymo trukmė, kai *Bayes'o algoritmo* iteracijų kiekis nuo 1 iki 17, yra nuo 7 sekundžių iki pusės valandos.

Genetinio algoritmo hibridizuoto su žemiausio kairėn užpildymo algoritmu parametrų optimizavimo tyrimo rezultatai:

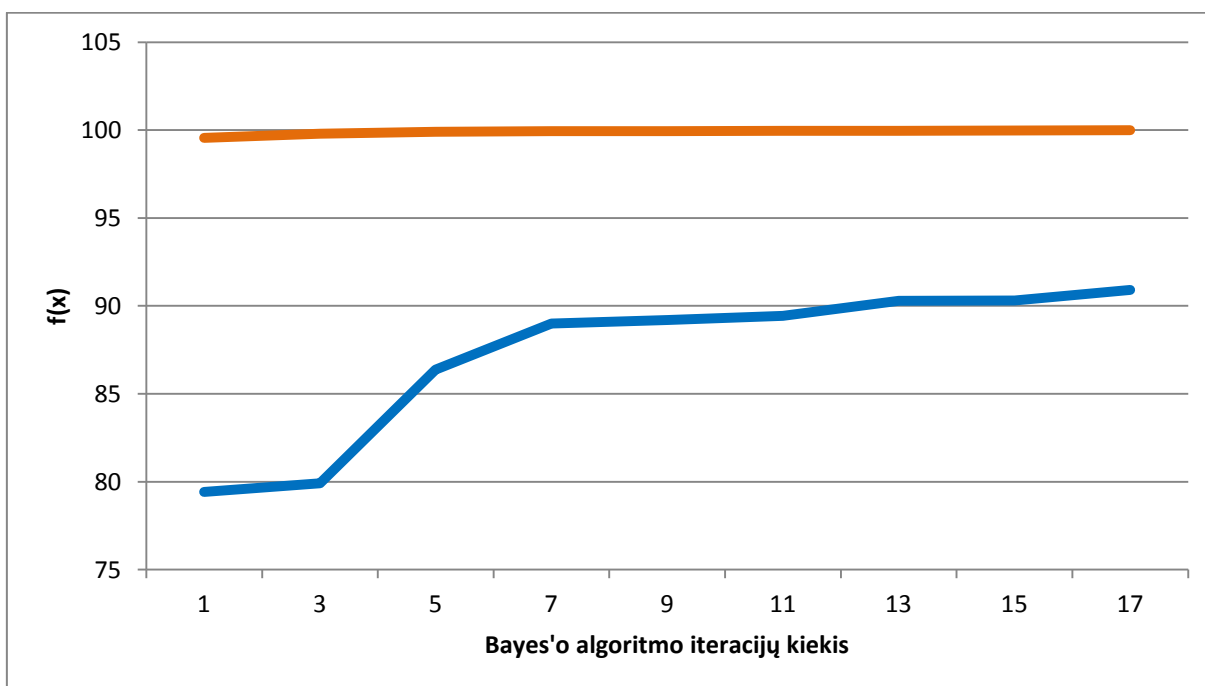
Šiuo metodu suformuoti rezultatai yra tikslūs, nes ne vieną kartą eksperimento metu buvo gautas ir 100% užpildymas. Šio metodo rezultatai nuolat gerėja, kai *Bayes'o algoritmo* iteracijų daugėja.

Prie optimalaus išdėstymo (t.y. šimtaprocentinio lakštų užpildymo) priartėjama pasirinkus 15 – 17 iteracijų kiekį.

Kuo daugiau pasirenkama *genetinio algoritmo* ir *Bayes'o algoritmo* iteracijų – tuo ilgiau reikia laukti rezultatų.

Metodas vykdomas labai ilgai (lyginant su prieš tai tirtu *genetiniu algoritmu hibridizuotu su žemiausio kairėn metodu*). Bandymų metu kai kurių rezultatų reikėjo laukti net virš pusės valandos.

5.3. Hibridizuotų genetinių algoritmų parametrų optimizavimo apžvalgos palyginimas



27 paveikslėlis. hibridizuotų genetinių algoritmų funkcijos $f(x)$ priklausomybės nuo Bayes'o algoritmo iteracijų kiekio. Oranžine spalva pažymėtas hibridizuotas algoritmas su žemiausio kairėn užpildymo metodu.

Mėlyna spalva pažymėtas hibridizuotas algoritmas su žemiausio kairėn užpildymo metodu.

Išvada:

Iš 27 paveikslėlyje esančios diagramos matyti, kad *genetinis algoritmas hibridizuotas su žemiausio kairėn užpildymo algoritmu* gauna tikslesnius rezultatus su mažesniu kiekiu iteracijų. Tačiau iš anksčiau matytų diagramų matome, kad jo veikimo laikas yra gerokai ilgesnis.

6. Išvados

1. Darbo metu buvo realizuoti du optimizavimo metodai:
 - genetinis algoritmas hibridizuotas su žemiausio kairėn metodu;
 - genetinis algoritmas hibridizuotas su žemiausio kairėn užpildymo metodu;
2. Sukurta programinė įranga, kuria galima optimizuoti parametrus genetiniam algoritmui hibridizuotam su žemiausio kairėn metodu ir genetiniam algoritmui hibridizuotam su žemiausio kairėn užpildymo metodu;
3. Atlikus pradinės algoritmo realizacijos versijos analizę, pastebėti trūkūmai ir iškelti algoritmo realizacijos tobulinimo tikslai:
 - parametrų optimizaciją papildyti kitokiais metodais (t.y. parametrų optimizaciją vykdyti ne tik su Bayes'o algoritmu);
 - optimizacijos metodus pritaikyti spręsti ne vien stačiakampių pjovimui, bet ir kitokioms figūroms;
 - tobulinti sistemos sąsają su vartotoju, pateikti informaciją vartotojui lengviau suprantama forma.
4. Eksperimento metu išanalizuota, kad skirtingai nurodytos euristicos parametrai įtakoja optimizavimo metodo sprendimo greitį ir tikslumą;
5. Genetinis algoritmas hibridizuotas su žemiausio kairėn užpildymo algoritmu gauna geresnius rezultatus su mažesniu iteracijų kiekiu, tačiau genetinis algoritmas hibridizuotas su žemiausio kairėn metodu gauna apytikslius rezultatus per daug trumpesnę laiką nei prieš tai minėtasis.
6. Pritaikytas genetinio algoritmo parametrų optimizavimas, naudojant Bayes metoda, yra efektyvus būdas padidinti sprendimo tikslumą, lyginant su paprastu genetinio algoritmo metodu, nes:
 - patikrinama žymiai daugiau variantų;
 - per tą patį laiką randamas tikslesnis sprendinys.

7. Literatūra

Albano A. ir Sapuppo G. Optimal allocation of two-dimensional irregular shapes using heuristic search methods [Žurnalas] // IEEE Transactions on Systems, Man and Cybernetics. - 1980 m.. - T. 10. - psl. 242–248.

Augutis J. [et al.] Iterative estimation of reliability parameters using the Bayesian approach [Konferencija] // Risk analysis IV: Fourth international conference on computer. - Rhodes, Greece : [s.n.], 2004. - psl. 497-502.

Augutis J., Žutautaitė I. ir Augutienė E. Estimation of parameter in ageing model using Bayesian approach [Žurnalas] // Lietuvos matematikos rinkinys . - Vilnius : [s.n.], 2004 m..

Bak S. [et al.] A Parallel Branch-and-Bound Approach to the Rectangular Guillotine Strip Cutting Problem [Žurnalas] // INFORMS Journal on Computing. - Linthicum, Maryland, USA : Institute for Operations Research and the Management Sciences, 2011 m.. - 1 : T. 23.

Bertsimas Dimitris ir Tsitsiklis John Simulated Annealing [Žurnalas] // Statistic Science. - 1993 m.. - psl. 10-15.

Burke E. K., Kendall G. ir Whitwell G. A New Placement Heuristic for the Orthogonal Stock-Cutting Problem [Žurnalas] // Operations research. - Nottingham : School of Computer Science and Information Technology, University of Nottingham, 2004 m.. - T. 52. - psl. 655–671.

Dagli C. H. ir Hajakbari A. Simulated annealing approach for solving stock cutting [Konferencija] // IEEE International Conference. - Los Angeles, CA : Institute of Electrical and Electronics Engineers, 1990.

Dyckhoff Harald A typology of cutting and packing problems [Žurnalas] // European Journal of Operational Research. - 1990 m.. - psl. 145-159.

Ferreira J.S., Neves M.A. ir Fonseca P. A two-phase roll cutting problem [Žurnalas] // European Journal of Operational Research. - 1990 m.. - T. 44. - psl. 185 - 196.

Fraser Alex S. Simulation of genetic systems by automatic digital computers. II. Effects of linkage on rates under selection. [Žurnalas] // Australian Journal of Biological Sciences. - 1957 m.. - psl. 492 - 499.

Garey Michael R. ir Johnson David S. Computers and Intractability: A Guide to the Theory of NPCompleteness [Knyga]. - New York, NY : W. H. Freeman & Co, 1990.

Gomory Ralph E. ir Gilmore P. C. A Linear Programming Approach to the Cutting-Stock Problem [Žurnalas] // Operations research. - New York : [s.n.], 1963 m..

Gomory Ralph E. On the relations between integer and noninteger solutions to linear programs [Žurnalas] // Proceedings of The National Academy of Sciences. - New York : [s.n.], 1965 m..

Haessler R.W ir Sweeney P.E. Cutting stock problems and solution procedures [Žurnalas] // European Journal of Operational Research. - 1991 m.. - T. 54. - psl. 141 - 150.

Haessler R.W. ir Vonderembse M.A. A procedure for solving the master slab cutting stock problem in the stell industry stock problem in the stell industry [Žurnalas] // AIIE Transactions. - 1979 m.. - T. 11. - psl. 160 - 165.

Holland John H. Adaptation in Natural and Artificial Systems [Knyga]. - Ann Arbor : University of Michigan Press, 1992.

Hopper A. ir Turton B. A Genetic Algorithm for a 2D Industrial Packing Problem [Žurnalas] // Computers and Industrial Engineering. - Cardiff : [s.n.], 1999 m.. - psl. 375-378.

Jakobs S. On genetic algorithms for the packing of polygons [Žurnalas] // Operations resears. - New York : [s.n.], 1996 m.. - psl. 165-181.

Kantorovich L. V. Mathematical Methods of Organizing and Planning Production [Žurnalas] // Management Science. - 1960 m.. - psl. 366-422 (vertimas į anglų kalbą).

Kröger B. Guillotisable bin packing: A genetic approach [Žurnalas] // European Journal of Operational Research. - Osnabrück : [s.n.], 1995 m.. - psl. 645–661.

Liu D. ir Teng H. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles [Žurnalas] // European Journal of Operational Research. - Dalian : [s.n.], 1999 m.. - psl. 413–419.

Macedoa R., Alves C. ir Valério de Carvalho J.M. Arc-flow model for the two-dimensional guillotine cutting stock problem [Žurnalas] // Computers & Operations Research. - 2010 m.. - 6 : T. 37.

Misevičius Alfonsas, Bukšnaitis Vytautas ir Blonskis Jonas Kombinatorinio optimizavimo ir genetinių algoritmų aspektai [Žurnalas] // Informacijos mokslai. - 2005 m.. - psl. 307-314.

Mockus J. [et al.] Bayesian Heuristic Approach to Discrete and Global Optimization: Algorithms, Visualization, Software, and Applications [Knyga]. - [s.l.] : Kluwer Academic Publisher, 2000.

Mockus J. A set of examples of global and discrete optimization 2.Home Work for Graduate Students [Knyga]. - [s.l.] : Kluwer Academic Publishser, 2004.

Papadimitriou Christos H. ir Steiglitz Kenneth Combinatorial optimization-algorithms and complexity [Knyga]. - Engwood Cliffs, New Jersey : Prentice-Hall, 1982. - psl. 383 - 431.

Pokštas J. Pjaustymo uždavinio algoritmų realizacija ir tyrimas. Magistro darbas [Ataskaita]. - Kaunas : [s.n.], 2007.

Pupeikienė L. Optimizavimo metodų tyrimas ir taikymas profiliuotų mokyklų tvarkaraščių sudarymo uždaviniuose. Daktaro disertacija [Ataskaita]. - Vilnius : Technika, 2009.

Wang H. [et al.] Population Migration Algorithm for Integer Programming and its Application in Cutting Stock Problem [Žurnalas] // Advanced Materials Research. - 2010 m.. - Tom. 143-144. - psl. 899 - 904.

8. Terminų ir santrumpų žodynas

Santrumpos:

BA – Bayes metodas (angl. *Bayes approach, BA*), optimizuojantis *genetinio algoritmo* parametrus.

BL – žemiausio kairėn algoritmas (angl. *Bottom Left*)

BLF – žemiausio kairėn užpildymo algoritmas (angl. *Bottom Left Fill*)

CSP – žaliavų taupymo problema (angl. *Cutting Stock Problem*)

GA – genetinis algoritmas (angl. *Genetic algorithm*)

Apibrėžimai:

Atliekos – medžiagos likučiai, kai iš jos išpjaunamos detalės gaminiams.

Lakštas – medžiagos dalis iš kurios išpjaunamos detalės.

Medžiaga – audinys, plokštė ar kita materialinė medžiaga kuri yra naudojama kaip pagrindas pjautymui.

Metaeuristika – sudėtingesnės strategijos, kurios gali valdyti kitas euristikas tam, kad surastume ypač sudėtingų uždavinių sprendinius. Metaeuristikų pavyzdžiai – Tabu paieška, modeliuojamojo atkaitinimo metodas, genetinis algoritmas ir kt.

Pjaustymas – tai fizinio objekto arba jo dalies padalijimas į dvi ar keletą naujų dalių, panaudojant pjautymo įrankį.

Pjaustymo planas - visų sudarytų šablonų visuma, kuri reprezentuoja visą užduoties sprendimą duotajam užsakymui.

Pjaustymo šablonas - medžiagos lakšto supjovimo schema.

A priedas

Genetinio algoritmo, hibridizuoto su žemiausio kairėn algoritmu, parametrų optimizavimo tyrimo rezultatai

1A lentelė. Genetinio algoritmo, hibridizuoto su žemiausio kairėn algoritmu,
parametrų optimizavimo tyrimo rezultatai

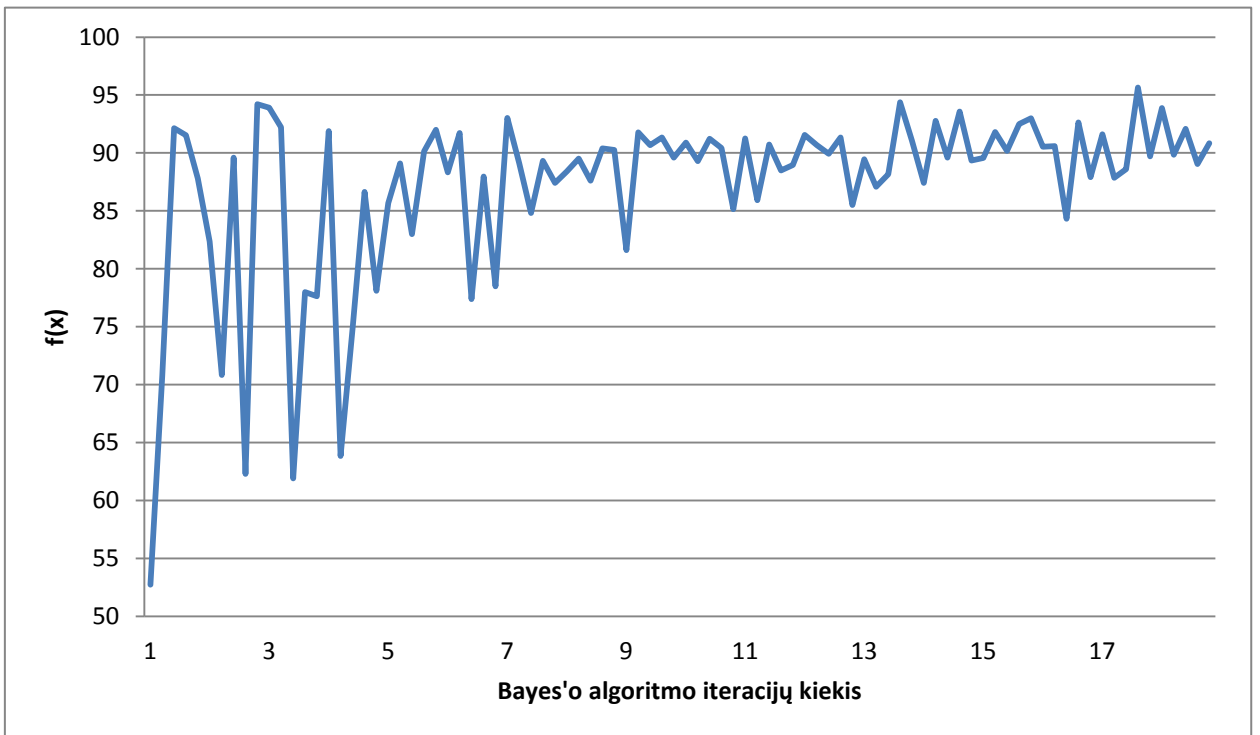
Iteracijos	Populiacija	Kryžminimo tikimybė	Mutacijos tikimybė	F(x)	Vykdymo trukmė (sekundėmis)
1	51	0,5	0,5	52,726	0,005
	51	0,5	0,5	70,783	0,003
	51	0,5	0,5	92,136	0,009
	51	0,5	0,5	91,539	0,01
	51	0,5	0,5	87,792	0,006
	51	0,5	0,5	82,36	0,006
	51	0,5	0,5	70,826	0,005
	51	0,5	0,5	89,591	0,003
	51	0,5	0,5	62,295	0,009
	51	0,5	0,5	94,204	0,009
3	51	0,5	0,5	93,905	0,026
	99,933	0,807	0,634	92,236	0,017
	97,64	0,942	0,058	61,915	0,016
	98,202	0,206	0,842	77,994	0,014
	95,47	0,322	0,672	77,634	0,023
	96,699	0,21	0,802	91,888	0,018
	51	0,5	0,5	63,832	0,019
	99,809	0,385	0,994	74,983	0,025
	98,044	0,729	0,631	86,632	0,014
	99,217	0,211	0,271	78,104	0,019
5	44,276	0,577	0,674	85,667	0,022
	98,732	0,389	0,495	89,09	0,032
	48,496	0,19	0,642	82,982	0,022
	51	0,5	0,5	90,144	0,019
	99,404	0,399	0,329	91,994	0,025
	69,928	0,745	0,518	88,32	0,026
	98,207	0,99	0,722	91,724	0,028
	99,878	0,396	0,999	77,374	0,031
	51	0,5	0,5	87,967	0,026
	73,181	0,935	0,202	78,486	0,028

1A lentelės tęsinys. Genetinio algoritmo, hibridizuoto su žemiausio kairėn algoritmu, parametrų optimizavimo tyrimo rezultatai

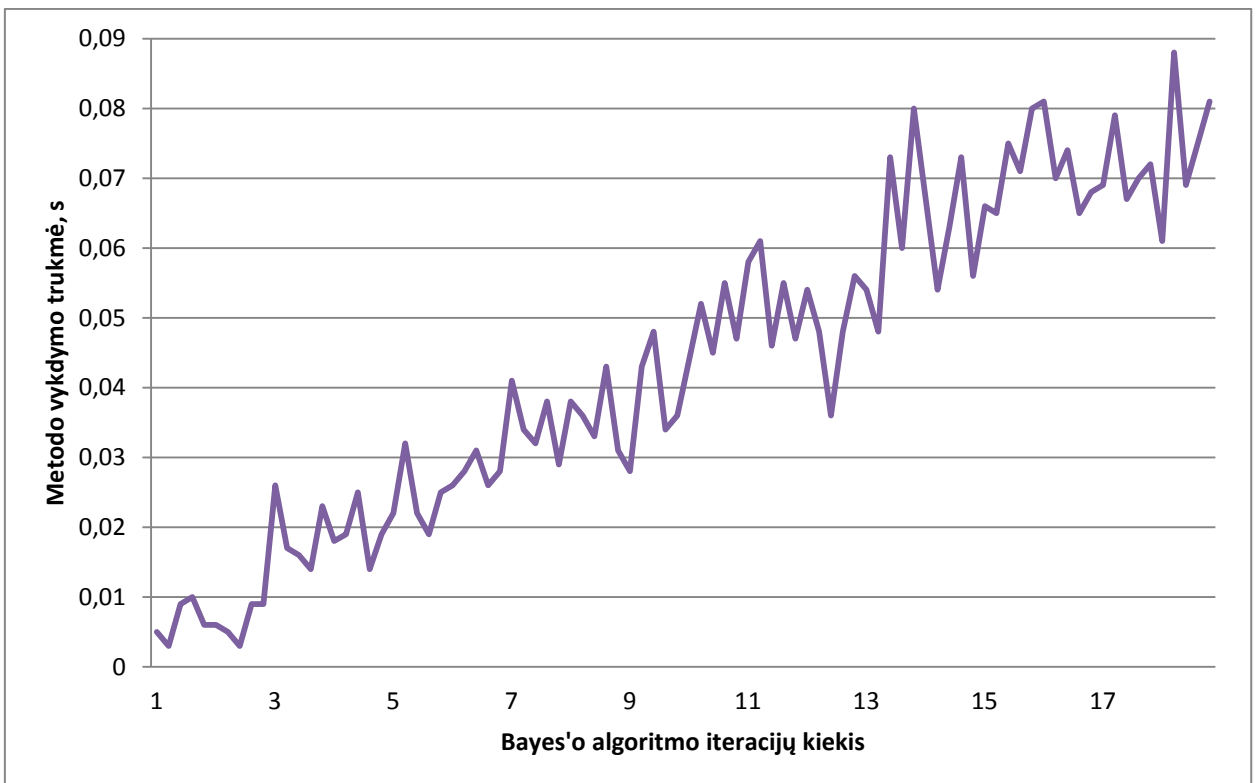
Iteracijos	Populiacija	Kryžminimo tikimybė	Mutacijos tikimybė	F(x)	Vykdymo trukmė (sekundėmis)
7	99,447	0,157	0,884	93,017	0,041
	99,317	0,384	0,197	89,132	0,034
	51	0,5	0,5	84,819	0,032
	98,496	0,478	0,613	89,334	0,038
	97,025	0,549	0,207	87,402	0,029
	71,6	0,917	0,64	88,416	0,038
	96,13	0,065	0,843	89,521	0,036
	97,33	0,931	0,143	87,615	0,033
	98,571	0,419	0,829	90,388	0,043
	97,222	0,295	0,147	90,259	0,031
9	51	0,5	0,5	81,624	0,028
	99,726	0,738	0,978	91,773	0,043
	99,62	0,59	0,758	90,666	0,048
	69,995	0,198	0,736	91,342	0,034
	80,995	0,933	0,431	89,604	0,036
	95,23	0,507	0,689	90,901	0,044
	90,32	0,91	0,975	89,287	0,052
	99,289	0,543	0,246	91,236	0,045
	98,211	0,793	0,97	90,426	0,055
	97,425	0,828	0,624	85,142	0,047
11	99,63	0,138	0,465	91,262	0,058
	99,321	0,263	0,843	85,932	0,061
	21,114	0,962	0,341	90,723	0,046
	86,637	0,529	0,381	88,481	0,055
	90,933	0,85	0,385	88,96	0,047
	32,359	0,078	0,873	91,57	0,054
	98,455	0,343	0,582	90,666	0,048
	19,472	0,868	0,193	89,923	0,036
	93,656	0,336	0,535	91,34	0,048
	78,049	0,781	0,093	85,516	0,056
13	46,207	0,534	0,47	89,452	0,054
	64,551	0,381	0,377	87,091	0,048
	68,744	0,634	0,789	88,158	0,073
	30,315	0,974	0,743	94,366	0,06
	93,978	0,407	0,748	91,058	0,08
	97,414	0,518	0,566	87,416	0,067
	94,407	0,597	0,597	92,765	0,054
	98,777	0,264	0,333	89,611	0,063
	91,807	0,081	0,351	93,581	0,073
	92,019	0,605	0,206	89,353	0,056

1A lentelės tęsinys. Genetinio algoritmo, hibridizuoto su žemiausio kairėn algoritmu, parametrų optimizavimo tyrimo rezultatai

Iteracijos	Populiacija	Kryžminimo tikimybė	Mutacijos tikimybė	F(x)	Vykdymo trukmė (sekundėmis)
15	70,492	0,783	0,162	89,567	0,066
	51	0,5	0,5	91,805	0,065
	95,017	0,077	0,883	90,175	0,075
	99,232	0,814	0,435	92,486	0,071
	75,1	0,635	0,832	93,002	0,08
	98,385	0,407	0,345	90,539	0,081
	97,299	0,936	0,467	90,593	0,07
	99,091	0,807	0,015	84,315	0,074
	57,385	0,702	0,458	92,637	0,065
	79,908	0,699	0,014	87,919	0,068
17	85,231	0,668	0,647	91,609	0,069
	90,121	0,976	0,669	87,856	0,079
	83,644	0,644	0,157	88,589	0,067
	64,181	0,404	0,121	95,658	0,07
	99,984	0,387	0,345	89,712	0,072
	87,899	0,679	0,813	93,882	0,061
	62,225	0,86	0,018	89,839	0,088
	77,918	0,495	0,603	92,09	0,069
	77,893	0,392	0,618	89,05	0,075
	62,986	0,091	0,4	90,846	0,081



28A paveikslėlis. Genetinio algoritmo, hibridizuoto su žemiausio kairėn algoritmu, funkcijos $f(x)$ priklausomybė nuo Bayes'o algoritmo iteracijų kiekio



29A paveikslėlis. Genetinio algoritmo, hibridizuoto su žemiausio kairėn algoritmu, veikimo trukmės priklausomybė nuo Bayes'o algoritmo iteracijų kiekio

B priedas

Genetinio algoritmo, hibridizuoto su žemiausio kairėn užpildymo algoritmu, parametrų optimizavimo tyrimo rezultatai

1B lentelė. Genetinio algoritmo, hibridizuoto su žemiausio kairėn užpildymo algoritmu, parametrų optimizavimo tyrimo rezultatai

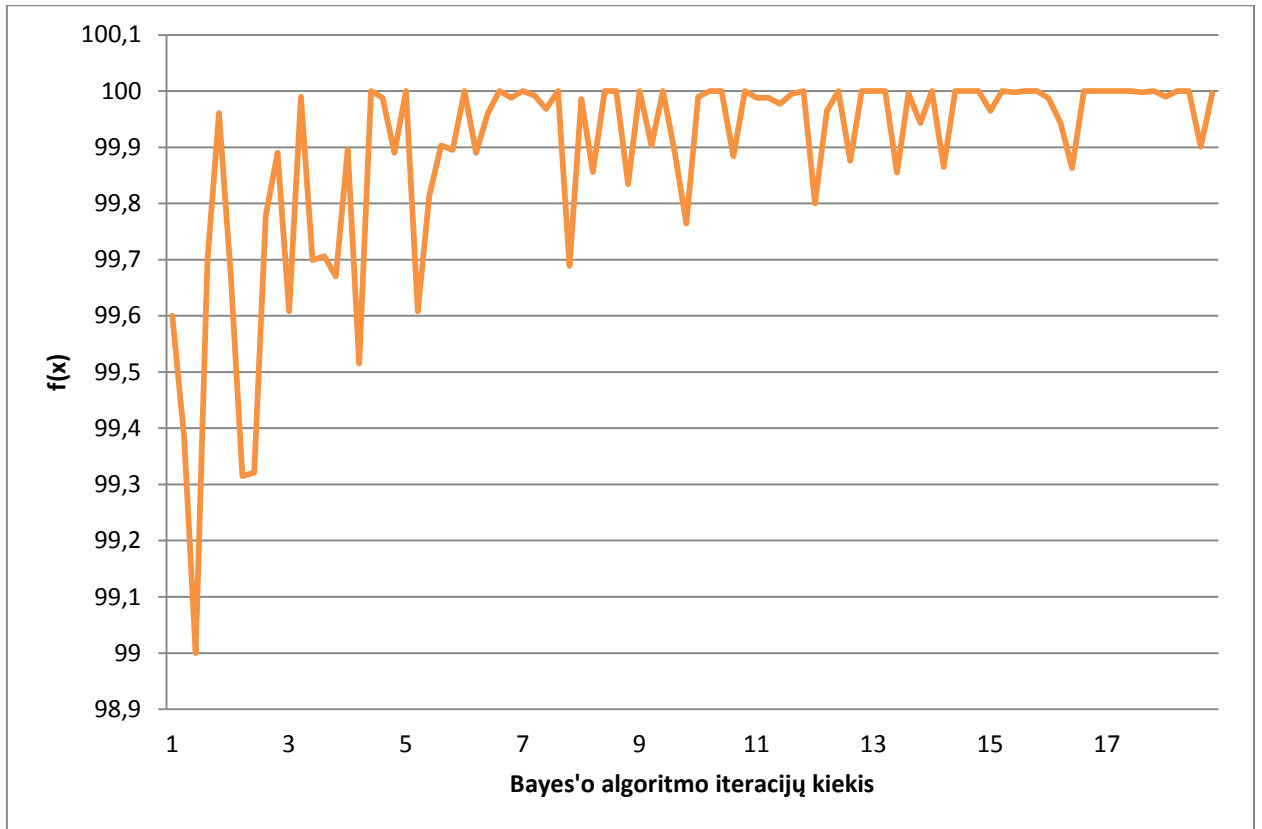
Iteracijos	Populiacija	Kryžminimo tikimybė	Mutacijos tikimybė	F(x)	Vykdymo trukmė (sekundėmis)
1	51	0,5	0,5	99,6	65
	51	0,5	0,5	99,384	23
	51	0,5	0,5	99	54
	51	0,5	0,5	99,699	9
	51	0,5	0,5	99,961	32
	51	0,5	0,5	99,67	45
	51	0,5	0,5	99,315	23
	51	0,5	0,5	99,321	7
	51	0,5	0,5	99,778	21
	51	0,5	0,5	99,89	63
3	91,377	0,121	0,486	99,608	54
	99,486	0,993	0,574	99,99	65
	51	0,5	0,5	99,699	187
	76	0,837	0,254	99,706	345
	51	0,5	0,5	99,67	143
	26,5	0,75	0,75	99,895	93
	94,483	0,519	0,546	99,515	95
	26,5	0,75	0,75	100	87
	97,188	0,522	0,169	99,988	136
	81	0,192	0,379	99,89	54
5	88	0,391	0,638	100	123
	26,5	0,75	0,75	99,608	342
	87,75	0,875	0,625	99,814	60
	80,484	0,849	0,616	99,903	268
	71,555	0,592	0,611	99,895	254
	99,446	0,995	0,288	100	683
	88,801	0,721	0,352	99,89	273
	98,808	0,541	0,879	99,961	268
	78,974	0,924	0,307	100	562
	51	0,5	0,5	99,988	264

1B lentelės tęsinys. Genetinio algoritmo, hibridizuoto su žemiausio kairėn užpildymo algoritmu, parametrų optimizavimo tyrimo rezultatai

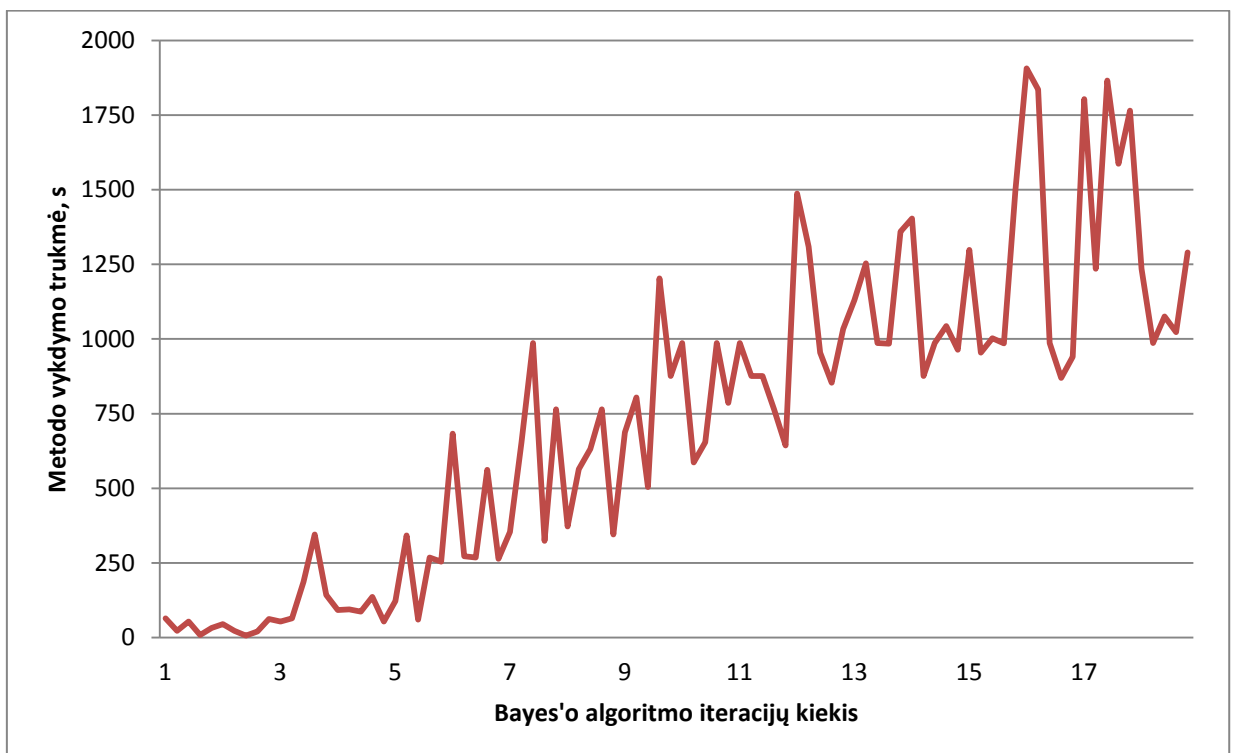
Iteracijos	Populiacija	Kryžminimo tikimybė	Mutacijos tikimybė	F(x)	Vykdymo trukmė (sekundėmis)
7	98,178	0,783	0,183	100	354
	6,674	0,125	0,132	99,992	654
	61,225	0,539	0,096	99,968	987
	99,079	0,992	0,392	100	324
	90,154	0,703	0,833	99,689	765
	76,226	0,876	0,129	99,986	372
	98,047	0,256	0,162	99,856	564
	67,105	0,028	0,409	100	632
	76,34	0,532	0,782	100	765
	98,305	0,105	0,76	99,834	345
9	76,537	0,311	0,324	100	687
	95,074	0,531	0,119	99,903	804
	99,721	0,48	0,976	100	504
	99,666	0,821	0,107	99,89	1203
	46,453	0,14	0,866	99,764	876
	71,983	0,945	0,808	99,99	987
	62,065	0,817	0,243	100	587
	4,627	0,115	0,573	100	654
	99,507	0,537	0,931	99,884	987
	5,49	0,316	0,201	100	786
11	99,414	0,99	0,242	99,988	987
	37,798	0,96	0,846	99,988	876
	93,631	0,419	0,994	99,977	876
	73,679	0,389	0,836	99,995	765
	74,878	0,538	0,108	100	643
	41,798	0,974	0,55	99,8	1487
	70,948	0,506	0,435	99,965	1309
	79,145	0,469	0,882	100	954
	86,602	0,959	0,753	99,876	854
	98,543	0,671	0,766	100	1034
13	96,908	0,159	0,395	100	1130
	64,614	0,275	0,217	100	1253
	65,155	0,176	0,673	99,855	987
	89,424	0,553	0,634	99,996	984
	98,155	0,74	0,508	99,943	1359
	73,844	0,084	0,046	100	1404
	46,843	0,698	0,733	99,865	876
	56,201	0,662	0,394	100	986
	63,637	0,491	0,471	100	1043
	81,945	0,683	0,899	100	964

1B lentelės tęsinys. Genetinio algoritmo, hibridizuoto su žemiausio kairėn užpildymo algoritmu, parametrų optimizavimo tyrimo rezultatai

Iteracijos	Populiacija	Kryžminimo tikimybė	Mutacijos tikimybė	F(x)	Vykdomo trukmė (sekundėmis)
15	94,584	0,312	0,678	99,965	1298
	77,391	0,025	0,693	100	954
	70,382	0,092	0,841	99,998	1003
	74,954	0,247	0,425	100	985
	89,756	0,015	0,614	100	1498
	58,237	0,163	0,137	99,987	1906
	70,536	0,834	0,957	99,943	1835
	81,107	0,872	0,068	99,863	987
	99,821	0,463	0,654	100	870
	65,268	0,536	0,612	100	940
17	93,196	0,562	0,971	100	1803
	95	0,765	0,657	100	1235
	74,061	0,747	0,9	100	1865
	80,727	0,786	0,345	99,998	1587
	84,774	0,13	0,681	100	1765
	69,453	0,881	0,761	99,99	1235
	96,065	0,517	0,187	100	987
	99,767	0,857	0,572	100	1076
	95,108	0,465	0,679	99,901	1023
	97,983	0,152	0,23	99,996	1290



30B paveikslėlis. Genetinio algoritmo, hibridizuoto su žemiausio kairėn užpildymo algoritmu, funkcijos $f(x)$ priklausomybė nuo Bayes'o algoritmo iteracijų kiekio



31B paveikslėlis. Genetinio algoritmo, hibridizuoto su žemiausio kairėn užpildymo algoritmu, veikimo trukmės priklausomybė nuo Bayes'o algoritmo iteracijų kiekio