

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Darius MAČIŪNAS

DAUGIAKRITERIS GLOBALUS SIJYNŲ
OPTIMIZAVIMAS GENETINIAIS
ALGORITMAIS

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,
MECHANIKOS INŽINERIJA (09T)



Vilnius LEIDYKLA TECHNICA 2013

Disertacija rengta 2009–2013 metais Vilniaus Gedimino technikos universitete.

Mokslinis vadovas

prof. habil. dr. Rimantas BELEVIČIUS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, mechanikos inžinerija – 09T).

VGTU leidyklos TECHNIKA 2121-M mokslo literatūros knyga
<http://leidykla.vgtu.lt>

ISBN 978-609-457-492-4

© VGTU leidykla TECHNIKA, 2013

© Darius Mačiūnas, 2013

darius.maciunas@vgtu.lt

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Darius MAČIŪNAS

**MULTI-OBJECTIVE GLOBAL
OPTIMIZATION OF GRILLAGES USING
GENETIC ALGORITHMS**

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,
MECHANICAL ENGINEERING (09T)



Vilnius LEIDYKLA TECHNICA 2013

Doctoral dissertation was prepared at Vilnius Gediminas Technical University in 2009–2013.

Scientific Supervisor

Prof Dr Habil Rimantas BELEVIČIUS (Vilnius Gediminas Technical University, Technological Sciences, Mechanical Engineering – 09T).

Reziumė

Disertacijoje nagrinėjamos sijynų globalaus daugiakriterio optimizavimo problemos, taikant genetinius algoritmus. Pagrindiniai tyrimo objektai – mechaninės sijynų konstrukcijos. Pagrindinis disertacijos tikslas – sukurti skaičiavimo technologijas, skirtas sijynų topologijos optimizavimui ir sinchroniniam topologijos ir matmenų optimizavimui.

Pagrindinis darbo uždavinys – optimalių sijynų schemų sudarymas, kai apibrėžtos kraštinės sąlygos ir veikia statinės apkrovos. Ši uždavinį galima išskaidyti į tokius uždavinius: optimalaus polių skaičiaus nustatymas, optimalių polių padėčių nustatymas ir optimalių jungiančiųjų sijų skerspjūvio matmenų nustatymas.

Disertaciją sudaro įvadas, trys skyriai, rezultatų apibendrinimas, naudotos literatūros ir autoriaus publikacijų disertacijos tema sąrašai ir trys priedai.

Įvade aptariama tiriamoji problema, darbo aktualumas, aprašomas tyrimų objektas, formuluojamas darbo tikslas bei uždaviniai, aprašoma tyrimų metodika, darbo mokslinis naujumas, darbo rezultatų praktinė reikšmė, ginamieji teiginiai, pristatomos disertacijos tema autoriaus paskelbtos publikacijos ir pranešimai konferencijose, autoriaus dalyvavimas užsienio stažuotėje bei disertacijos struktūra.

Pirmasis skyrius skirtas literatūros apžvalgai: sijynų pagrindiniams globalaus ir daugiakriterio optimizavimo metodams. Pagrindžiamas genetinių algoritmų pasirinkimas.

Antrajame skyriuje aprašoma skaičiavimo technologija sijynų topologijai optimizuoti panaudojus genetinius algoritmus. Formuluojama optimizavimo problema. Pateikiami sijynų topologijos optimizavimo rezultatai.

Trečiajame skyriuje aprašyta skaičiavimo technologija sijynų sinchroniniam topologijos ir matmenų optimizavimui panaudojus genetinius algoritmus. Apibrėžiami optimizavimo uždavinio sprendiniai, kai tikslo funkcijos kriterijų įtaka optimizavimo uždaviniui yra vienoda bei panaudojus adaptyvų genetinį algoritmą ir genetinį algoritmą su paskirstymo strategija.

Disertacijos bendrosios išvados apibendrina darbo rezultatus ir jų reikšmę.

Disertacijos tema paskelbti 7 straipsniai: vienas – straipsnių rinkinyje, (Thomson ISI Web of Science), vienas – recenzuojamame užsienio mokslo žurnale, trys – recenzuojamame Lietuvos mokslo žurnale, du – recenzuojamoje tarptautinės konferencijos medžiagoje. Disertacijos tema perskaityti 8 pranešimai Lietuvos bei kitų šalių konferencijose.

Abstract

The dissertation investigates the issues of global multi-objective optimization of grillages, applying genetic algorithms. The main objects of research are grillage-type mechanical structures. The primary purpose of this dissertation is development and employment of suitable calculation technologies for simultaneous topology and size optimization of grillages.

The paper approaches the following major task – generation of schemes of optimal grillages under a set of the static loads and boundary conditions. The task may be splitted as follows: identification of the optimal number of piles, optimal positions of piles and optimal cross-sectional dimensions of connecting beams.

The dissertation consists of introduction, 3 chapters, general conclusions, references and 3 annexes.

The introduction reveals the investigated problem, importance of the thesis and the object of research and describes the purpose and tasks of the paper, research methodology, scientific novelty, the practical significance of results and defended statements. The author's publications on the subject of the dissertation and the material of made presentations in conferences are presented, foreign internship of the author and the structure of the dissertation introduced.

Chapter 1 revises literature, including the analysis of the main global and multi-objective optimization methods of grillages. The reasoning is introduced for choosing genetic algorithms for optimization.

Chapter 2 describes the calculation technology, enabling to apply genetic algorithms for topology optimization of grillages. Formulation of optimization problem is presented. The solutions of topology optimization of grillages are given.

In chapter 3 the calculation technology for simultaneous topology and size optimization of grillages using genetic algorithms is described and results of optimization of grillages are presented: using classical genetic algorithm, adaptive genetic algorithm and genetic algorithm with distribution strategy.

General conclusions of the dissertation summarize the results and their significance.

7 articles focusing on the subject of the dissertation are published: one article – in the Thomson ISI Web of Science register, one article – in reviewed foreign scientific journal, three articles – in reviewed Lithuanian scientific journals, two articles – in material reviewed during international conferences. 8 presentations on the subject have been given in conferences at national and international level.

Žymėjimai

Simboliai

E – medžiagos tamprumo modulis;

f – tikslo funkcija;

R – realiųjų skaičių aibė;

X – problemos (galimų sprendinių aibė) erdvė, t. y. fenomas;

x – problemos galimas sprendinys, t.y. fenotipas;

w – svorinis koeficientas;

$g(x)$ – tikslo funkcijų suma;

S – problemos sprendinių erdvė;

G – problemos paieškos erdvė, t.y. genomai;

g – genotipas;

Y – tikslo erdvė;

$v(x)$ – tinkamumo vertė;

V – tinkamumo verčių aibė;

j – atsitiktinis skaičius (priklausantis sveikųjų skaičių intervalui) kryžminimo pozicijai;

k – individo (genotipo) ilgis (genų skaičius);

F – kriterijų rinkinys;

x^* – globalus optimumas;

X^* – optimalių sprendinių aibė;

x_l^* – lokalus optimumas;
 n – tikslo funkcijų skaičius;
 ε_l – kriterijų viršutiniai rėžiai;
 N – natūralusis skaičius;
 R^+ – teigiamų realiųjų skaičių poaibis;
 σ^2 – dispersija;
 $[K]$ – lenkiamo strypo elemento standumo matrica;
 A – sijos skerspjūvio plotas;
 $[K_{11}]$, $[K_{12}]$, $[K_{22}]$ – lenkiamo strypo elemento standumo matricos submatricos;
 $[K_{12}]^T$ – transponuota submatrica;
 J , I_z , I_y – sijos skerspjūvio inercijos momentai;
 a – baigtinių elementų ansamblis;
 u – baigtinių elementų mazgų poslinkis;
 F_a – aktyviosios jėgos;
 R_l – reakcijos jėga i -tajame poliuje;
 M – lenkimo momentas sijoje;
 κ – strypo kreivis;
 N_i – Hermito antrosios eilės interpoliacinės funkcijos;
 x_l – lokalioji strypo elemento koordinatė;
 $f_1(x)$ – didžiausia reakcijos jėga poliuje;
 $f_2(x)$ – didžiausias absoliutiniu dydžiu lenkimo momentas sijoje;
 N_a – polių skaičius sijoje;
 N_k – sijų skaičius sijoje;
 m – vienos sijos atkarpų skaičius, kur skaičiuojamas lenkimo momentas;
 L – ilgis;
 δ – mažiausias galimas atstumas tarp gretimų polių;
 $p_kryž$ – kryžminimo tikimybė;
 p_mut – mutacijos tikimybė;
 w_s – svorinių koeficientų santykis;
 p_j – atrankos tikimybė;

$u_i, u_j, v_i, v_j, w_i, w_j$ – lenkiamo strypo baigtinio elemento atitinkamai i -tojo mazgo ir j -tojo mazgo poslinkiai atitinkamai x, y, z koordinačių ašių kryptimis;

$\varphi_{xi}, \varphi_{xj}, \varphi_{yi}, \varphi_{yj}, \varphi_{zi}, \varphi_{zj}$ – lenkiamo strypo baigtinio elemento atitinkamai i -tojo mazgo ir j -tojo mazgo posūkiai apie atitinkamai x, y, z koordinačių ašis;

$F_{xi}, F_{xj}, F_{yi}, F_{yj}, F_{zi}, F_{zj}$ – lenkiamo strypo baigtinio elemento atitinkamai i -tajame mazge ir j -tajame mazge veikiančios jėgos atitinkamai x, y, z koordinačių ašių kryptimis;

$F_{Mxi}, F_{Mxj}, F_{Myi}, F_{Myj}, F_{Mzi}, F_{Mzj}$ – lenkiamo strypo baigtinio elemento atitinkamai i -tajame mazge ir j -tajame mazge veikiantys momentai apie atitinkamai x, y, z koordinačių ašis.

Santrumpos

GA – genetinis algoritmas;

BEM – baigtinių elementų metodas;

GO – globalusis optimizavimas;

VEGA – vektoriškai įvertintas genetinis algoritmas;

MOGA – daugiakriteris genetinis algoritmas;

NPGA – nišinis Pareto genetinis algoritmas;

NSGA – nedominuojantis atrinkimo genetinis algoritmas;

Pop – populiacija;

NSSGA – nedominuojantis atrinkimo ir dalinimosi genetinis algoritmas;

VG TU – Vilniaus Gedimino technikos universitetas;

FMF – Fundamentinių mokslų fakultetas;

AGA – adaptyvus genetinis algoritmas;

GAPS – genetinis algoritmas su paskirstymo strategija;

PP – programinis paketas;

KPP – komercinis programinis paketas;

APDO – adaptyvus populiacijos dydžio operatorius;

PD – populiacijos dydis;

AMO – adaptyvus mutacijos operatorius;

VNSB – vienas nepriklausomas skaitinis bandymas.

Turinys

ĮVADAS	1
Problemos formulavimas	1
Darbo aktualumas	3
Tyrimų objektas	3
Darbo tikslas	3
Darbo uždaviniai	4
Tyrimų metodika	4
Darbo mokslinis naujumas	5
Darbo rezultatų praktinė reikšmė	5
Ginamieji teiginiai	5
Darbo rezultatų aprobavimas	6
Disertacijos struktūra	7
1. GLOBALAUS DAUGIAKRITERIO SIJYNŲ OPTIMIZAVIMO METODŲ APŽVALGA	9
1.1. Globalus optimizavimas	10
1.1.1. Daugiakriteris optimizavimas	12
1.1.2. Pareto optimizavimas	13
1.1.3. Svertinės kriterijų sumos metodas	15
1.2. Globalaus optimizavimo metodai	18
1.2.1. Monte-Karlo metodas	20
1.2.2. Evoliuciniai skaičiavimai	21
1.2.3. Genetiniai algoritmai	21

1.3. Genetinių algoritmų taikymo sijynų optimizavimui apžvalga.....	33
1.4. Pirmojo skyriaus išvados ir disertacijos uždavinių formulavimas.....	40
2. GENETINIS ALGORITMAS SIJYNŲ TOPOLOGIJOS OPTIMIZAVIMUI.....	41
2.1. Sijynų optimizavimas	42
2.2. Sijynų charakteristikų nustatymas	43
2.3. Sijynų optimizavimo problemos formuluotė	48
2.4. Optimizavimo uždavinio apribojimai	51
2.5. Sijynų kodavimas realiųjų skaičių seka	52
2.6. Optimizavimo algoritmas	53
2.7. Optimizavimo uždavinio sprendimas genetiniu algoritmu	56
2.8. Tikslų funkcijos kriterijų įtaka optimizavimo uždaviniui.....	64
2.9. Antrojo skyriaus išvados	66
3. GENETINIS ALGORITMAS SIJYNŲ TOPOLOGIJOS IR MATMENŲ SINCHRONINIAM OPTIMIZAVIMUI.....	67
3.1. Pareto-optimalūs sprendiniai	68
3.2. Optimalūs genetiniai parametrai ir sijos skerspjūvio matmenys	71
3.3. Tikslų funkcijos kriterijų įtaka optimizavimo uždaviniui skirtingos topologijos sijynuose	73
3.4. Optimizavimo uždavinio sprendimas, kai tikslų funkcijos komponentų įtaka uždaviniui vienoda.....	75
3.5. Adaptyvus genetinis algoritmas.....	77
3.6. Genetinis algoritmas su paskirstymo strategija	81
3.7. Trečiojo skyriaus išvados	83
BENDROSIOS IŠVADOS	85
LITERATŪRA IR ŠALTINIAI	87
AUTORIAUS PUBLIKACIJŲ DISERTACIJOS TEMA SĄRAŠAS.....	101
PRIEDAI.....	103
A priedas. 10 polių ir 15 polių sijynai	103
B priedas. 18 polių sijynas.....	106
C priedas. 20 polių sijynas.....	108

Contents

INTRODUCTION	1
Problem of research.....	1
Topicality of the work	3
Object of reserch	3
Aim of the work	3
Tasks of the work	4
Methodology of research.....	4
Scientific novelty.....	5
Practical value	5
Defended propositions.....	5
Approval of the work results	6
The scope of the scientific work.....	7
1. REVIEW OF METHODS FOR GLOBAL MULTI-OBJECTIVE OPTIMIZATION OF GRILLAGES.....	9
1.1. Global optimization	10
1.1.1. Multi-objective optimization	12
1.1.2. Pareto optimization.....	13
1.1.3. Weighted Sum Method	15
1.2. Methods of global optimization.....	18
1.2.1. Method of Monte-Carlo	20
1.2.2. Evolutionary computation.....	21
1.2.3. Genetic algorithms.....	21

1.3. Review of genetic algorithms for grillage optimization	33
1.4. Conclusions of the first chapter and formulation of the dissertation tasks	40
2. GENETIC ALGORITHM FOR TOPOLOGY OPTIMIZATION OF GRILLAGES ...	41
2.1. Optimization of grillages	42
2.2. Identification of characteristics of grillages.....	43
2.3. Formulation of optimization problem of grillages.....	48
2.4. Constraints of optimization problem	51
2.5. Coding of grillages into string of real numbers	52
2.6. Optimization algorithm	53
2.7. Solution of optimization problem using genetic algorithm	56
2.8. Contribution of components of the objective function to optimization problem...	64
2.9. Conclusions of the second chapter	66
3. GENETIC ALGORITHM FOR SIMULTANEOUS TOPOLOGY AND SIZE OPTIMIZATION OF GRILLAGES	67
3.1. Pareto-optimal solutions.....	68
3.2. Optimal values of genetic parameters and dimensions of beam cross-section	71
3.3. Contribution of components of the objective function to optimization problem for grillages with different topology.....	73
3.4. Solution of optimization problem, when contribution of components of the objective function to optimization problem is equivalent.....	75
3.5. Adaptive genetic algorithm	77
3.6. Genetic algorithm with distribution strategy	81
3.7. Conclusions of the third chapter	83
GENERAL CONCLUSIONS	85
REFERENCES	87
AUTHOR'S LIST OF PUBLISHED WORKS ON THE TOPIC OF THE DISSERTATION	101
APPENDIX.....	103
Appendix A. 10 pile and 15 pile grillages	103
Appendix B. 18 pile grillage	106
Appendix C. 20 pile grillage	108

Ivadas

Problemos formulavimas

Darbe nagrinėjamas rostverkinių pamatų, kuriuos sudaro poliai ir juos jungiančios sijos, optimizavimas. Toliau darbe tokie pamatai vadinami sijynais. Sijynų globalusis optimizavimas yra sudėtinga kompleksinė problema: optimalus sijynas yra toks, kuris turi mažiausią įmanomą polių skaičių bei kurio jungiančiosios sijos yra mažiausio skerspjūvio. Matematiškai tokio sijyno projektavimo uždavinys yra globaliojo optimizavimo uždavinys, apimantis du skirtingus optimizavimo uždavinius: sijyno optimizavimą siekiant mažiausių absoliutiniu dydžiu lenkimo momentų jungiančiosiose sijose bei sijyno optimizavimą siekiant mažiausių atraminių reakcijų poliuose. Taigi, šį uždavinį galima priskirti prie daugiakriterių uždavinių, kadangi šiame uždavinyje tarpusavyje „varžosi“ keli kriterijai: atraminės reakcijos jėga poliuje ir lenkimo momentas jungiančiojoje sijoje. Keli skirtingi optimizavimo uždaviniai yra apjungiami į vieną, todėl siūloma kompromisinė tikslo funkcija, kuri bus minimizuojama. Paprastai tokių uždavinių sprendinys – tai ne unikali optimali tikslo funkcijos reikšmė, o kompromisinių sprendinių rinkinys, vadinamas Pareto-optimaliais sprendiniais. Pareto sprendinys yra optimalus ta prasme, kad nė vienas tikslo funkcijos kriterijus negali būti „pagerintas“ nepabloginus bent vieno iš likusiųjų tikslo funkcijos kriterijų.

Paprasčiausias daugiakriterio optimizavimo metodas yra pagrįstas svorinių koeficientų metodika. Tikslo funkcija (paprastai konstrukcijos kaina) kaip taisyklė priklauso nuo daugelio projektavimo kintamųjų skaičiaus, todėl atsiranda didelis kompiuterio skaičiavimo išteklių poreikis. Praktinių optimizavimo uždavinių sprendimas perrinkimo algoritmais nėra įmanomas dėl didžiulio nagrinėjamų variantų skaičiaus. Be to, tokiems optimizavimo uždaviniams nėra žinomų ir polinominių sprendimo algoritmų.

Optimizavimo kriterijai yra šie: atraminės reakcijos jėga poliuje ir lenkimo momentas jungiančiojoje sijoje. Optimizavimo (šiuo atveju minimizavimo) uždavinio projektavimo kintamieji yra polių padėties po sijynu bei jungiančiųjų sijų skerspjuvio matmenys. Sijyno jungiančiųjų sijų skerspjuvio matmenys nustatomos taip:

1. Nustatomas didžiausias lenkimo momentas sijyne, kylantis dėl sijyną veikiančių aktyviųjų jėgų.
2. Nustatomas sijos skerspjuvis, kuris atlaiko surastą didžiausią lenkimo momentą sijoje.
3. Nustatomas sijos skerspjuvio matmenys (aukštis ir plotis) pagal įvestą apribojimą didžiausiam sijos skerspjuvio matmenų santykiui.

Polių padėtimis sijyne neįvedami jokie suvaržymai (išskyrus kelis technologinius apribojimus).

Sijynų optimizavimo problemai yra taikomi tokie apribojimai:

- Statinės pusiausvyros apribojimas.
- Stiprumo apribojimai poliams: didžiausia reakcijos jėga poliuje negali viršyti poliaus laikančiosios galios.
- Stiprumo apribojimai jungiančiosioms sijoms: didžiausias lenkimo momentas sijoje negali viršyti leistinojo lenkimo momento.
- Kraštinės sąlygos polių padėtimis: mažiausias galimas atstumas tarp gretimų polių negali būti mažesnis už leistiną mažiausią atstumą (dėl polialalės darbo ypatumų).
- Kraštinės sąlygos polių padėtimis: optimizavimo metu poliai gali judėti tik išilgai jungiančiųjų sijyno sijų.
- Kraštinės sąlygos sijos skerspjuvio matmenims.

Disertacijoje numatoma atlikti skaitinius sijynų tyrimus ir spręsti sijynų optimizavimo technologijos kūrimo problemą pritaikant bei modifikuojant VGTU Teorinės mechanikos katedros kolektyvo sukurtą inžinerinių struktūrų optimizavimo technologiją. Remiantis šia technologija sijynai bus analizuojami baigtinių

elementų metodu, Sijynų optimizavimas bus atliekamas taikant genetinius algoritmus (GA). GA priklauso tikimybinių algoritmų klasei ir labai dažnai tyrėjų yra naudojami globalaus optimizavimo uždaviniuose.

Darbo aktualumas

Sijynai – sudaryti iš polių ir jungiančiųjų sijų – yra vienos populiariausių bei efektyviausių polinių pamatų inžinerinės konstrukcijos. Sijynų optimizavimo rezultatai turi didelę reikšmę ekonominiu požiūriu, nes ypatingai svarbu gebėti greitai suprojektuoti pigią ir tuo pačiu racionalią bei patvarią konstrukciją. Todėl inžineriniu požiūriu šios problemos sprendimo rezultatai turi didelę reikšmę kuriant efektyvią sijynų optimizavimo technologiją.

Apytikriai galima laikyti, kad mažiausia kaina bus tokių sijynų, kurie turi mažiausią įmanomą polių skaičių bei kurių jungiančiosios sijos yra mažiausio skerspjuvio. Matematinio požiūriu tokių sijynų projektavimo uždavinys yra globaliosios optimizacijos uždavinys, apjungiantis du skirtingus optimizavimo uždavinius: sijyno optimizavimą siekiant mažiausių absoliutiniu dydžiu lenkimo momentų jungiančiosiose sijose bei sijyno optimizavimą siekiant mažiausių atraminių reakcijų poliuose. Akivaizdu, kad galima vienu metu spręsti abu šiuos uždavinius taikant kompromisinę tikslo funkciją su parenkamais svoriniais koeficientais atraminėms reakcijoms bei lenkimo momentams minimizuoti.

Tyrimų objektas

Darbo tyrimų objektas – mechaninės sijynų konstrukcijos, apkrautos statine apkrova. Sprendžiamas tokių konstrukcijų daugiakriteris globalus optimizavimo uždavinys.

Darbo tikslas

Šio darbo pagrindinis tikslas – sukurti skaičiavimo technologiją, kuri leistų, panaudojus genetinį algoritmą (GA), atlikti nagrinėjamos inžinerinei praktikai aktualios apimties sijynų topologijos bei matmenų sinchroninį optimizavimą.

Darbo uždaviniai

Darbo tikslui pasiekti darbe reikia spręsti šiuos uždavinius:

1. Atlikti mokslinės literatūros apžvalgą apie sijynų pagrindinius globalaus ir daugiakriterio optimizavimo metodus.
2. Pritaikius GA koncepciją, sukurti ir išbandyti skaičiavimo technologiją, kuri leistų optimizuoti atskirai sijynų topologiją ir sijynų sijų skerspjuvio matmenis.
3. Pritaikius GA koncepciją, sukurti ir išbandyti skaičiavimo technologiją sinchroniniam sijynų topologijos bei matmenų optimizavimui.
4. Pritaikius adaptyvaus genetinio algoritmo (AGA) koncepciją, sukurti ir išbandyti skaičiavimo technologiją sinchroniniam sijynų topologijos bei matmenų optimizavimui.
5. Pritaikius GA su paskirstymo strategija (GAPS) koncepciją, sukurti ir išbandyti skaičiavimo technologiją sinchroniniam sijynų topologijos bei matmenų optimizavimui.

Tyrimų metodika

Darbe taikomi skaitiniai skaičiavimo ir sprendimo metodai. Tikslą funkciją sudaro didžiausia atraminė reakcija poliuje ir didžiausias absoliutiniu dydžiu lenkimo momentas jungiančiojoje sijoje su parenkamais svoriais. Šią tikslo funkciją siekiama minimizuoti, nepažeidžiant statinės pusiausvyros ir didžiausių įtempimų apribojimų. Deja, bendrajam momentų-reakcijų minimizavimo uždaviniui globaliojo sprendinio gauti iš anksto neįmanoma jokiai tikslo funkcijos svorių kombinacijai. Tikslą funkcijos reikšmės nustatymas, reakcijų ir momentų reikšmės nustatymas bei apribojimų tikrinimas realizuotas baigtinių elementų metodo (BEM) pagalba, originalia programa, sukurta VGTU Teorinės mechanikos katedros kolektyvo. Ši programa buvo pritaikyta, kad būtų tinkama nagrinėjamo uždavinio sprendimui. Optimizavimo uždavinys sprendžiamas originaliu genetiniu algoritmu. Programos ir algoritmai yra patikrinti baigtinių elementų programiniu paketu ANSYS.

Darbo mokslinis naujumas

Rengiant disertaciją buvo gauti šie mechanikos inžinerijos mokslui nauji rezultatai:

1. Pritaikius GA koncepciją, pasiūlyta ir įgyvendinta originali programinė sijynų sinchroninio topologijos ir matmenų optimizavimo technologija. Ši skaičiavimo technologija suteikia galimybę vienu metu siekti dviejų tikslų: nustatyti optimalią polių išdėstymo schemą bei nustatyti optimalius sijų skerspjūvio matmenis – tokiu būdu sprendžiamas mišrus topologijos optimizavimo-matmenų nustatymo uždavinys.
2. Pritaikius adaptyvaus genetinio algoritmo (AGA) koncepciją, pasiūlyta ir įgyvendinta originali programinė sijynų sinchroninio topologijos ir matmenų optimizavimo technologija. Ši skaičiavimo technologija suteikia galimybę gauti vidutiniškai 3% geresnius optimizavimo rezultatus (mažesnes tikslo funkcijos reikšmes) nei taikant GA koncepciją.
3. Pritaikius GA su paskirstymo strategija (GAPS) koncepciją, pasiūlyta ir įgyvendinta originali programinė sijynų sinchroninio topologijos ir matmenų optimizavimo technologija. Ši skaičiavimo technologija suteikia galimybę gauti vidutiniškai 5% geresnius optimizavimo rezultatus (mažesnes tikslo funkcijos reikšmes) nei taikant GA koncepciją.

Darbo rezultatų praktinė reikšmė

Pasiūlytos skaičiavimo technologijos ir gauti tyrimų rezultatai tinkami inžinerinei praktikai priimtinos apimties polinių pamatų sijynų optimizavimo uždavinių sprendimui. Pritaikius GA, paprastai randami keli racionalūs sprendiniai su artimomis tikslo funkcijos reikšmėmis, tačiau atitinkančias skirtingas polių išdėstymo schemas bei skirtingus sijų skerspjūvio matmenis. Tai naudinga inžinerinėje praktikoje, kadangi projektuotojas gali rinktis jam priimtinesnį variantą.

Ginamieji teiginiai

1. Pritaikius adaptyvaus genetinio algoritmo (AGA) koncepciją, pasiūlyta programinė sijynų sinchroninio topologijos ir matmenų optimi-

zavimo technologija, kuri suteikia galimybę gauti vidutiniškai 3% geresnius optimizavimo rezultatus (mažesnes tikslo funkcijos reikšmes) nei taikant GA koncepciją.

2. Pritaikius GA su paskirstymo strategija (GAPS) koncepciją, pasiūlyta programinė sijynų sinchroninio topologijos ir matmenų optimizavimo technologija, kuri suteikia galimybę gauti vidutiniškai 5% geresnius optimizavimo rezultatus (mažesnes tikslo funkcijos reikšmes) nei taikant GA koncepciją.

Darbo rezultatų aprobavimas

Disertacijos tema yra atspausdinti 7 moksliniai straipsniai: vienas – recenzuojamame mokslo žurnale, įtraukta į Thomson ISI Web of Science sąrašą (Moccus *et al.* 2012); vienas – recenzuojamame užsienio mokslo žurnale, cituojamame duomenų bazėje Scopus (Mačiūnas and Belevičius 2012), trys – recenzuojamame Lietuvos mokslo žurnale, cituojamame duomenų bazėje Index Copernicus (Belevičius *et al.* 2011; Mačiūnas *et al.* 2011; Mačiūnas *et al.* 2012a), du – recenzuojamoje tarptautinių konferencijų medžiagoje (Mačiūnas 2012; Mačiūnas *et al.* 2012b).

Disertacijoje atliktų tyrimų rezultatai buvo paskelbti aštuoniose mokslinėse konferencijose Lietuvoje ir užsienyje:

- Jaunųjų mokslininkų konferencijoje „*Mokslas – Lietuvos ateitis*“ 2011–2012 m. Vilniuje.
- Tarptautinėje konferencijoje „*Optimization and Analysis of Structures*“ 2011, Tartu, Estijoje.
- Tarptautinėje konferencijoje „*Mechanika*“ 2012 m. Kaune.
- Tarptautinėje virtualioje konferencijoje „*The 1st Virtual International Conference on Advanced Research in Scientific Areas*“, 2012 m., (organizatorius iš Slovakijos).
- VGTU mokslinėje konferencijoje „*Respublikinis Lietuvos skaičiuojamosios mechanikos seminaras*“ 2010, 2011, 2012 m., Vilniuje.

2012 m. autorius dalyvavo užsienio stažuotėje Tartu universitete (Tartu, Estija). Stažuotės metu (2012 10 01 – 2012 10 31) autorius susipažino su Tartu universitete taikomais struktūrų optimizavimo metodais, išlaikė numatytų dalykų egzaminus bei užmezgė mokslinius kontaktus su Tartu universiteto mokslininkais – vienas iš jų prof. Jaan Lellep.

Disertacijos struktūra

Disertaciją sudaro įvadas, trys skyriai, išvados, literatūros šaltinių ir autoriaus publikacijų sąrašai bei trys priedai.

Darbo apimtis yra 120 puslapių įskaitant priedus, tekste panaudotos 36 numeruotos formulės, 50 paveikslų ir 16 lentelių. Rašant disertaciją buvo panaudoti 186 literatūros šaltiniai.

1

Globalaus daugiakriterio sijynų optimizavimo metodų apžvalga

Nuo pat egzistencijos pradžios žmonija siekė tobulumo daugelyje sričių. Žmonija nori pasiekti didžiausią laimės laipsnį su kuo mažiausiom pastangom. Pavyzdžiui ekonomikoje pelnas ir pardavimai turi būti kuo didesni, o išlaidos – kuo mažesnės. Taigi, optimizavimas yra vienas seniausių mokslų, kuris netgi persikelia į kasdienį gyvenimą (Neumaier 2006). Žmonija vystosi, auga jos poreikiai, tad ir optimizavimo problemos vis sudėtingėja.

Skyriuje analizuojami globalaus sijynų optimizavimo metodai, kuriais remiantis inžinerinėje praktikoje yra optimizuojamos sijynų tipo konstrukcijos; aprašomi globalaus sijynų optimizavimo tipai. Modernūs globalaus optimizavimo metodai yra aprašyti galybėje literatūros šaltinių (Goldberg and Samtani 1986; Törn and Žilinskas 1989; Horst and Pardalos 1994; Kearfott 1996; Moccus *et al.* 1996; Floudas 1999; Žilinskas and Klause 2002; Tawarmalani and Sahinidis 2002; Žilinskas A. and Žilinskas J. 2002; Riolo and Worzel 2003; Neumaier 2004; Cheng *et al.* 2007; Belevičius and Šešok 2008; Šešok *et al.* 2010; Belevičius *et al.* 2011). Ypatingas dėmesys skiriamas genetiniam algoritmui, kuris gana dažnai taikomas (Holland 1967; Bagley 1967; Rosenberg 1967; Holland 1975; Michalewicz 1995; Hajela *et al.* 1998; Saka *et al.* 2000; Coello 2000; Kim *et al.* 2005; Belevičius and Šešok 2008; Šešok and

Belevičius 2009; Belevičius *et al.* 2011) sijynų tipo konstrukcijų optimizavimui. Globalaus optimizavimo metodai skirti surasti globalųjį sprendinį ar bent jau priartėti prie jo, todėl reikia masyvių skaičiavimų, kurio metu analizuojami ir apdorojami labai dideli informacijos kiekiai. Tai sukelia didžiulių kompiuterio resursų poreikį sprendžiant globalaus optimizavimo uždavinius. Stochastiniai globalaus optimizavimo metodai sprendžiant tokius uždavinius yra perspektyvūs bei efektyvūs (Weise 2009; Šešok *et al.* 2010). Iš stochastinių globalaus optimizavimo metodų didžiausias dėmesys bus skirtas genetiniams algoritmams, pagrįstiems mus supančios aplinkos procesų imitavimu ir konkrečiai – biologinės evoliucijos modeliavimu (Gen 1997; Langdon and Poli 2002; Puiša 2005; Reeves 2002; Weise 2009). Be to, darbe aptarta sijynų optimizavimo vystymosi istorija bei šiandieninė situacija.

1.1. Globalus optimizavimas

Globalus optimizavimas – tai skaitinės analizės ir taikomosios matematikos šaka, skirta optimizavimui. Globalaus optimizavimo (GO) tikslas yra surasti geriausius galimus elementus x^* iš aibės X remiantis rinkiniu kriterijų $F = \{f_1, f_2, \dots, f_n\}$ (Weise 2009). Šie kriterijai yra išreiškiami matematinėmis funkcijomis, dar vadinamomis tikslo funkcijomis.

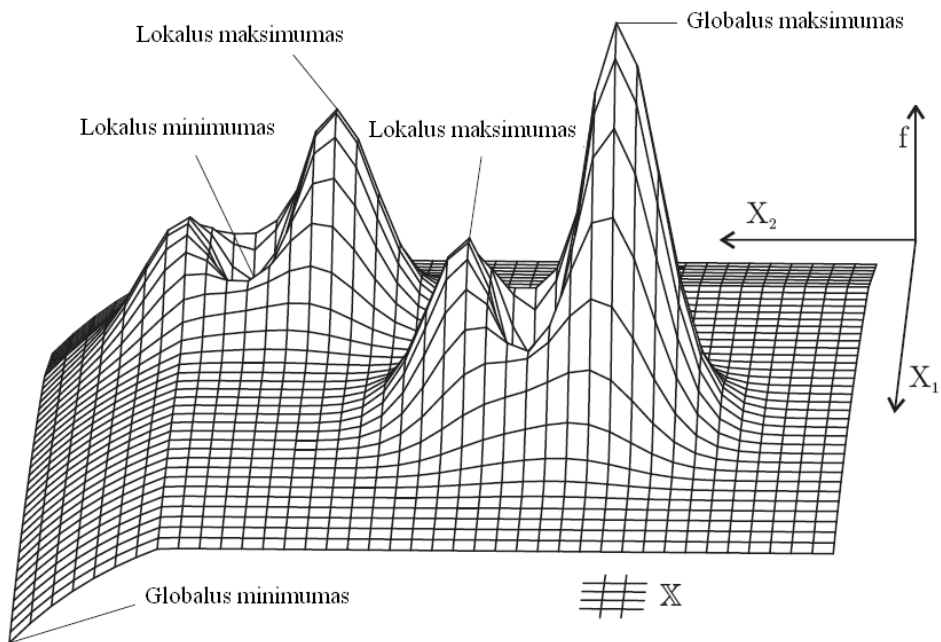
Tikslo funkcija $f : X \rightarrow Y$ (čia $Y \subseteq R$) – tai matematinė funkcija, kuriai optimizavimo metu siekiama surasti geriausią vertę. Tikslo funkcijos poaibis Y bei jo ribos turi priklausyti realių skaičių poaibiui ($Y \subseteq R$). Aibė X yra vadinama problemos galimų sprendinių erdve. GO apima visus metodus, kurie gali būti naudojami siekiant rasti geriausius elementus x^* iš aibės X kriterijų $f \in F$ atžvilgiu. Reikia pažymėti, kad adaptuojant globalaus optimizavimo metodą tam tikrai uždavinių klasei yra labai svarbu, jog algoritme būtų tinkamai įvertinta visa žinoma informacija apie uždavinį – tai leidžia gauti geresnius sprendinius (Šešok 2008a).

Euristika (Pearl 1984; Rayward-Smith *et al.* 1996; Michalewicz and Fogel 2004) – tai optimizavimo algoritmo dalis, kuri naudoja algoritmo surinktą informaciją, kad padėtų nuspręsti, kuris kitas galimas sprendinys turėtų būti bandomas skaičiavimams arba kaip gali būti gautas kitas individas (Weise 2009). GO naudojamos euristikos padeda nuspręsti, kurią galimų sprendinių aibę reikia nagrinėti toliau, kad būtų galima gauti geresnį sprendinį. Viena vertus, deterministiniai algoritmai paprastai naudoja euristikas tam, kad nustatytų galimų sprendinių eiliškumo tvarką. Kita vertus, tikimybiniai metodai tolesniems skai-

čiavimams gali naudoti tik tokius paieškos erdvės elementus, kurie buvo atrinkti euristicų pagalba.

Metaeuristika – tai metodas, skirtas spręsti labai bendro pobūdžio problemas. Šis metodas abstrakčiu būdu apjungia tikslo funkcijas arba euristikas, paprastai neanalizuodamas jų struktūros ar vidinės sandaros, t.y. traktuodamas jas kaip juodosios dėžės procedūras (Glover and Kochenberger 2003; Blum and Roli 2003; Gonzalez 2007; Weise 2009). Ši kombinacija dažnai yra atliekama stochastiškai naudojant statistikos duomenis, gautus iš paieškos erdvės arba remiantis natūralios gamtos reiškinių ar fizinių procesų modeliais.

Optimizavimo algoritmai gali būti skirstomi į tokius, kurie siekia surasti vienakriterių tikslo funkcijų f geriausias reikšmes bei į tokius, kurie vienu metu optimizuoja tikslo funkcijų rinkinius F . Optimizuojant vienakriteres tikslo funkcijas f , priklausomai nuo to, ko ieškome, optimumas gali būti arba maksimumas arba minimumas. 1.1 paveiksle yra parodyta tokia funkcija f , kurią apibrėžia dviejų matmenų erdvė $X = (X_1, X_2)$.



1.1 pav. Dviejų matmenų funkcijos globalus ir lokalus optimumai (Weise 2009)

Fig. 1.1. Global and local optima of a two-dimensional function (Weise 2009)

Čia išskiriamas lokalus optimumas ir globalus optimumas. Globalus optimumas yra optimumas visoje aibėje X , tuo tarpu lokalus optimumas – tai X poaibio optimumas. Vienos tikslo funkcijos $f: X \rightarrow R$ lokalus optimumas $x_i^* \in X$ yra arba lokalus minimumas, arba lokalus maksimumas; o analogiškai globalus optimumas $x^* \in X$ – tai arba globalus minimumas, arba globalus maksimumas. GO algoritmas – tai optimizavimo algoritmas, kuris pasitelkia tokias priemones, kurios užkerta kelią konvergavimui į lokalų optimumą ir padidina tikimybę rasti globalų optimumą (Weise 2009).

1.1.1. Daugiakriteris optimizavimas

Realiame pasaulyje optimizavimo uždaviniai dažnai būna daugiakriteriai (Bui and Alam 2008; Kalanta 2003), todėl inžinerinių problemų sprendimas paprastai realizuojamas sinchroniškai (vienu ir tuo pačiu metu) optimizuojant daug įvairių kriterijų. Nors esminės daugiakriterio optimalumo (Edgeworth 1881) sąvokos suformuluotos XIX a. pabaigoje, tačiau ypatingo dėmesio daugiakriterio optimizavimo problemos sulaukė XX a. viduryje (Koopmans 1951; Charnes and Cooper 1961; Lee 1972). Buvo analizuojami daugiakriterio tiesinio programavimo metodai (Zeleny 1974; Keeney and Raiffa 1976). Toliau vystantis ir populiarėjant šiai optimizavimo kryptčiai (Hwang and Masud 1979; Chankong and Haimes 2008), daugiakriterio optimizavimo metodų grupę papildė ir genetiniai algoritmai:

1. VEGA (*angl.* Vector Evaluated Genetic Algorithm) metodas (Schaffer 1984).
2. MOGA (*angl.* Multi Objective Genetic Algorithm) metodas (Fonseca and Fleming 1993).
3. NPGA (*angl.* Niche Pareto Genetic Algorithm) metodas (Horn *et al.* 1994).
4. NSGA (*angl.* Nondominated Sorting Genetic Algorithm) metodas (Srinivas and Deb 1994).

Ši mokslinė sritis intensyviai plėtojama ir pastaraisiais metais (Luque *et al.* 2011; Lačinskas *et al.* 2011; Erfani and Utyuzhnikov 20011).

Lietuvoje taip pat plačiai analizuojamos ir sprendžiamos daugiakriterio optimizavimo problemos. Mokslininkai J. Mockus, R. Belevičius, A. Žilinskas, J. Žilinskas, G. Dzemyda, D. Šešok nagrinėja ir sprendžia įvairias optimizavimo problemas, taip pat ir daugiakriterio optimizavimo problemas. Galima paminėti tokius reikšmingus mokslinius darbus (Mockus and Mockus 1991; Dzemyda and

Petkus 2001; Belevičius *et al.* 2002; Žilinskas and Žilinskas 2002; Šešok *et al.* 2010; Mockus 2011; Belevičius *et al.* 2011).

Taigi, sprendžiant daugiakriterio optimizavimo problemas, optimizavimo metodai yra taikomi rinkiniams F , sudarytiems iš n tikslo funkcijų f_i , kurių kiekviena charakterizuoja vieną optimizuojamą kriterijų (Weise 2009):

$$F = \{f_i : X \rightarrow Y_i : 0 < i \leq n, Y_i \in R\}. \quad (1.1)$$

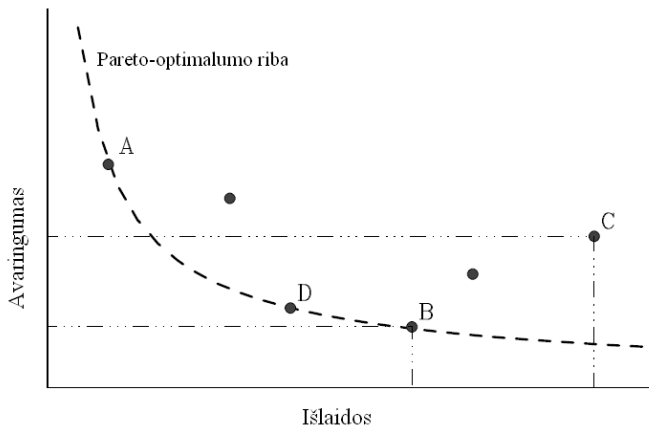
Daugiakriterio optimizavimo principas skiriasi nuo vienakriterio optimizavimo. Vienakriterio optimizavimo tikslas yra surasti geriausią sprendinį, kuris atitinka mažiausią arba didžiausią tikslo funkcijos reikšmę (Reklaitis *et al.* 2006). Daugiakriteriame optimizavime su konfliktuojančiais (konkuruojančiais) kriterijais vieno optimalaus sprendinio neegzistuoja, t.y. dėl skirtingų kriterijų sąveikos atsiranda kompromisinių sprendinių aibė, taip vadinami Pareto-optimalūs sprendiniai (Steuer 1986). Todėl daugiakriterio optimizavimo tikslas yra surasti kiek įmanoma daugiau Pareto-optimalių sprendinių. Šie sprendiniai yra pateikiami sprendimų priėmėjui, kuris pasirenka vieną tinkamiausią sprendinį. Todėl sprendimo priėmėjo dalyvavimas yra būtinas sprendžiant daugiakriterius optimizavimo uždavinius (Klamroth and Miettinen 2008; Miettinen and Makela 2006; Huang *et al.* 2005).

1.1.2. Pareto optimizavimas

Vilfredas Paretas (Vilfredo Pareto) padėjo matematinius pagrindus daugiakriteriam optimizavimui (McLure 2001; Pareto 1986), kuris atsižvelgia į tarpusavyje konfliktuojančius kriterijus bei įvertina jų tarpusavio ryšius bei sąveiką. Pareto optimalumas tapo svarbia sąvoka žaidimų teorijoje, ekonomikos, socialiniuose ir inžineriniuose moksluose (Chinchuluun *et al.* 2008; Osborne and Rubinstein 1994). Jis apibrėžia sprendinių, kurie gali būti gauti optimaliai suderinant (pasiekiant kompromiso) konfliktuojančius kriterijus, ribas. Remiantis šiomis ribomis sprendimų priėmėjas (ar tai būtų žmogus, ar algoritmas) gali galutinai pasirinkti tokią konfigūraciją, kuri jo nuomone tinka geriausiai (Fonseca and Fleming 1995; Galperin 1997; Fonseca and Fleming 1998). Optimalią aibę X^* sudaro visi optimalūs elementai. Parastai egzistuoja be galo daug optimalių sprendinių. Kadangi kompiuterių atmintis yra ribota, todėl įmanoma rasti tik baigtinę optimalių sprendinių aibę (poaibį).

Elementas $x^* \in X$ yra Pareto-optimalus (tuo pačiu priklauso optimalios aibės X^* daliai), jei jis nedominuoja kito elemento atžvilgiu problemos erdvėje X . Elementas x_1 dominuoja (jam yra teikiama pirmenybė) elemento x_2 atžvilgiu, jei x_1 yra geresnis nei x_2 bent vienoje tikslo funkcijoje ir neblogesnis visų

kitų elementų atžvilgiu. Pareto optimizavimo prasme, X^* yra vadinamas Pareto aibe arba Pareto ribomis (*angl.* Pareto Frontier). Panagrinėkime šį aspektą tokiu hipotetiniu pavyzdžiu – viena iš avaringumo keliuose priežasčių yra techniškai netvarkingų automobilių eksploatavimas (Šmigelskas ir Vitkūnas 2007), todėl būtina didinti išlaidas techninei automobilių priežiūrai. 1.2 paveiksle pavaizduoti Pareto-optimalūs sprendiniai esant dviems optimizuojamiems kriterijams – išlaidoms ir avaringumui – kurie abu turi būti minimizuojami (Deb 1999). Taškas A žymi sprendinį, kuris gautas esant arti mažiausios išlaidų reikšmės, bet tuo pačiu esant beveik didžiausiai avaringumo reikšmei. Taškas B tuo tarpu žymi sprendinį, kuris atspindi beveik mažiausią avaringumo reikšmę, bet tuo pačiu atspindi ir beveik didžiausią išlaidų reikšmę. Jei abu kriterijai (išlaidos ir avaringumas) yra svarbūs, tai neįmanoma nustatyti, ar sprendinys A yra geresnis už sprendinį B , ar priešingai. Vienas sprendinys yra geresnis už kitą remiantis vienu kriterijumi, bet blogesnis remiantis kitu kriterijumi. Be to, egzistuoja daug tokių sprendinių (pvz. sprendinys D), kurie taip pat priklauso Pareto-optimaliajai ribai (aibe) ir neįmanoma nustatyti absoliučios hierarchijos tarp sprendinių A , B ir D . Visi šie sprendiniai (1.2 pav. pažymėti išgaubta punktyrine kreive) yra vadinami Pareto-optimaliais sprendiniais. Pareto-optimalūs sprendiniai yra optimalūs ta prasme, kad nė vieno sprendinio neįmanoma pagerinti, nepabloginus kito sprendinio. Taip pat egzistuoja ir ne Pareto-optimalūs sprendiniai (pvz. sprendinys C), kadangi jie yra blogesni už Pareto-optimalius sprendinius (pvz. sprendinys D) pagal abu kriterijus.



1.2 pav. Pareto-optimalių sprendinių koncepcija (Deb 1999)
Fig. 1.2. The concept of Pareto-optimal solutions (Deb 1999)

Todėl pagrindiniai daugiakriterio GO uždaviniai yra šie:

- surasti kiek įmanoma daugiau Pareto-optimalių sprendinių;
- užtikrinti, kad šie sprendiniai tarpusavyje būtų kuo labiau skirtingi (Deb 1999).

1.1.3. Svertinės kriterijų sumos metodas

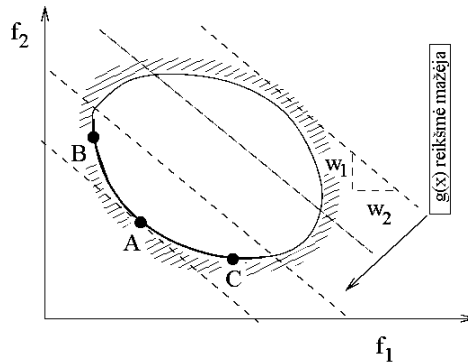
Sprendžiant daugiakriterius optimizavimo uždavinius vienas paprasčiausių būdų nustatyti, kas yra optimalu – tai svertinės kriterijų sumos metodas (*angl.* Weighted Sum Method/Linear Aggregation) (Weise 2009; Deb 1999; Zadeh 1963). Kiekviena tikslo funkcija $f_i(x) \in F$ yra dauginama iš svorinio koeficiento w_i , kuris charakterizuoja funkcijos f_i svarbą ir minimizuojama arba maksimizuojama kriterijų tikslo funkcijų suma $g(x)$. Tokiu būdu daugiakriteris optimizavimo uždavinys transformuojamas į vienakriterį. Jei $g(x)$ yra minimizuojama, tai uždavinys formuluojamas taip:

$$\min_{x \in X} g(x) = \min_{x \in X} \sum_{i=1}^n w_i f_i(x), \quad (1.2)$$

čia w_i – i -tojo kriterijaus svorinis koeficientas (dažniausiai trupmeninis), tenkinantis sąlygas: $0 < w_i < 1$, $\sum_{i=1}^n w_i = 1$. Visi Pareto-optimalūs sprendiniai turi būti išsidėstę galimų sprendinių aibėje X .

Šio metodo realizavimas yra nesudėtingas. Parenkamas atsitiktinis svorinių koeficientų vektorius ir optimizuojant vienakriterį uždavinį $g(x)$ gaunamas optimalus sprendinys. Norint gauti skirtingus Pareto-optimalius sprendinius, reikia parinkti skirtingus atsitiktinius svorinių koeficientų vektorius ir kiekvieną kartą optimizuoti $g(x)$. Metodo veikimas iliustruotas (1.3 pav.) sprendžiant hipotetinę problemą (Deb 1999). 1.3 paveiksle pavaizduota galimos paieškos erdvė funkcijos erdvėje, turinčioje du kriterijus f_1 ir f_2 , kurie turi būti minimizuojami. Kiekvienas taškas, esantis galimų sprendinių aibėje, apibūdina f_1 ir f_2 reikšmes. Optimizuojant (1.2), kai yra fiksuotas svorinių koeficientų vektorius, gaunama hiper-plokštuma (linija dviems kriterijams f_1 ir f_2) su tam tikra fiksuota orientacija funkcijos erdvėje. Optimalus sprendinys yra taške, kuriame fiksuotos orientacijos hiper-plokštuma turi bendrą liestinę su galimos paieškos erdve. Šis sprendinys pažymėtas taške A , kur linija – koordinatinių ašių f_1 ir f_2 sistemoje proporcingai atidėta atitinkamai w_2 ir w_1 reikšmėms – yra liestinė galimų

sprendinių aibei. Pasirinkus skirtingą atsitiktinį svorinių koeficientų vektorių bei suradus atitinkamą bendros liestinės tašką galima gauti kitokius sprendinius kaip B arba C . Esant skirtingiems svorinių koeficientų vektorių rinkiniams $w = (w_1, w_2, w_3, \dots, w_n)$, gaunamas skirtingų sprendinių (A , B , C ir kitų) rinkinys, kuris sudaro Pareto-optimalių sprendinių aibę (1.3 pav. parodyta pastorinta linija).



1.3 pav. Išgaubta Pareto-optimalumo riba (Deb 1999)
Fig. 1.3. Convex Pareto-optimal front (Deb 1999)

Sprendiniai, gauti šiuo metodu, gali būti naudojami kaip kitų metodų pradiniai sprendiniai (Filatovas 2012). Šio metodo vienas iš trūkumų yra tas, kad nuosekliai kintant svorinių koeficientų reikšmėms rasti sprendiniai nebūtinai bus tolygiai pasiskirstę Pareto aibėje (Das and Dennis 1997). Be to, sprendimų priėmėjui dažnai tampa sudėtinga nustatyti tinkamas svorinių koeficientų reikšmes, ypač kai jis nedaug žino apie nagrinėjamos problemos specifiką.

Svertinės kriterijų sumos metodas yra tinkamas ieškant Pareto-optimalių sprendinių daugiakriteriams optimizavimo uždaviniams, turintiems išgaubtą Pareto-optimalumo ribą (1.3 pav.).

Kai Pareto-optimalumo riba nėra išgaubta (1.4 pav.) kiekvienam svorinių koeficientų vektoriui galima rasti tik A arba B sprendinius, o visų kitų Pareto-optimalių sprendinių neįmanoma rasti. Tokiu atveju dažnai naudojamas ε -ribojimų metodas (*angl.* ε -constraint Method) (Haimes *et al.* 1971).

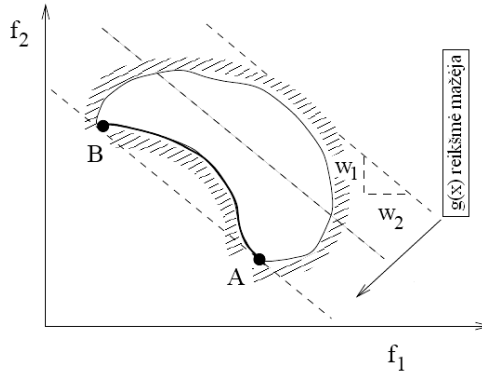
ε -ribojimų metodas pagrįstas tuo, kad minimizuojama vieno kriterijaus tikslo funkcija, o kitų kriterijų funkcijos yra suvedamos į apribojimus (Deb 1999). Uždavinys formuluojamas taip:

$$\min_{x \in X} f_j(x), \quad j \in \{1, \dots, N\}, \quad (1.3)$$

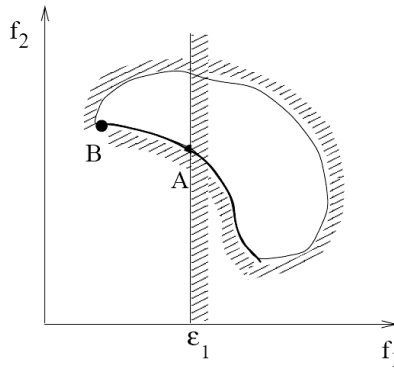
kai egzistuoja tokie apribojimai:

$$f_l(x) \leq \varepsilon_l, \quad l=1, \dots, N, \quad l \neq j \quad (1.4)$$

čia ε_l – kriterijų viršutiniai rēžiai, N – natūralusis skaičius. Pareto-optimalūs sprendiniai gali būtī rasti, parinkus tinkamas ε_l reikšmes l -tajai tikslo funkcijai. Šis metodas gali surasti neišgaubtą Pareto-optimalumo ribą (1.5 pav.).



1.4 pav. Neišgaubta Pareto-optimalumo riba (Deb 1999)
 Fig. 1.4. Non-convex Pareto-optimal front (Deb 1999)



1.5 pav. ε -ribojimų metodas (Deb 1999)
 Fig. 1.5. ε -constraint method (Deb 1999)

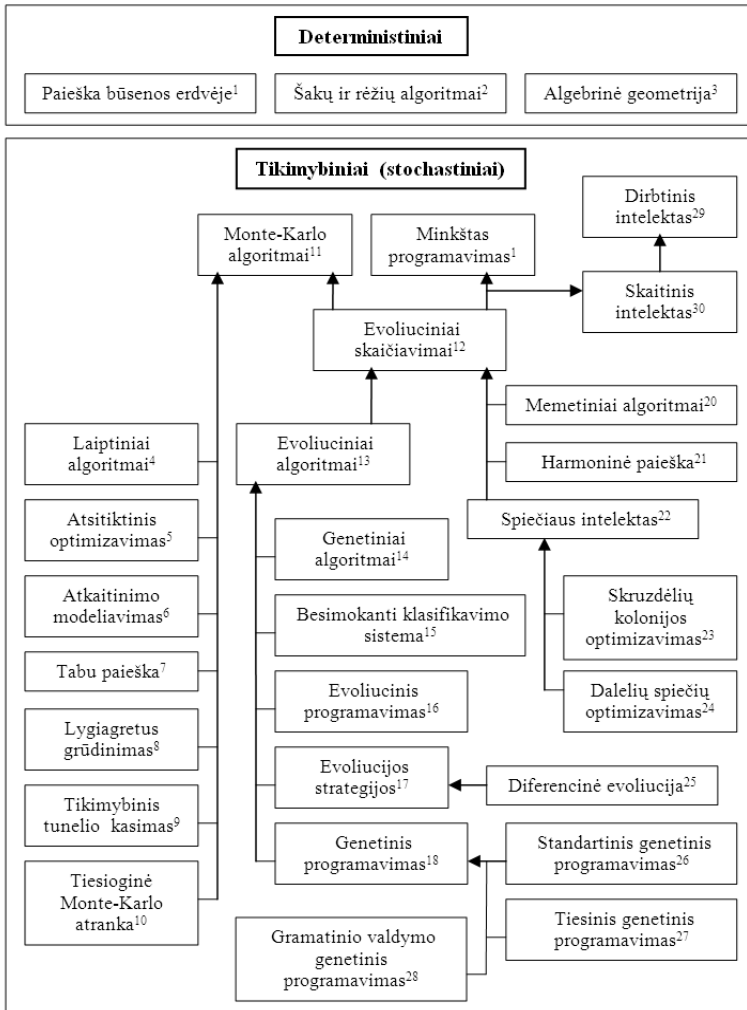
Pirmoji tikslo funkcija yra panaudota kaip apribojimas $f_1(x) \leq \varepsilon_1$. Tačiau metodo trūkumas yra tas, kad iš anksto turi būti žinomos tinkamų ε_i reikšmių ribos kiekvienai tikslo funkcijai.

Be to, egzistuoja keletas kitų metodų (Steuer 1986), kaip Tikslinis programavimas (angl. Goal Programming), minimumo-maksimumo metodas (angl. Min-Max method). Visi metodai reikalauja tam tikrų žinių apie nagrinėjamą problemą. Pats didžiausias visų šių metodų trūkumas yra tas, jog visi jie turi būti taikomi daug kartų tam, kad tikėtinai kiekvieną kartą būtų rastas vienas skirtingas Pareto-optimalus sprendinys (Deb 1999). Todėl šie metodai nėra patrauklūs ir tai yra viena iš priežasčių, kodėl daugiakriterio optimizavimo problemų yra vengiama spręsti praktikoje. Pagrindiniai klasikinių optimizavimo metodų (Svertinės kriterijų sumos metodas, ε – ribojimų metodas ir kt.), naudojamų daugiakriteriam optimizavimui, trūkumai (Deb 1999):

1. Tam, kad būtų surasta daug Pareto-optimalių sprendinių, algoritmą būtina paleisti daug kartų.
2. Didžioji dauguma algoritmų reikalauja tam tikrų žinių apie nagrinėjamą problemą.
3. Kai kurie algoritmai yra jautrūs Pareto-optimalumo ribos formai.
4. Pareto-optimalių sprendinių išsibarstymas priklauso nuo tikslo funkcijos formos.
5. Šie klasikiniai metodai nėra patikimi sprendžiant problemas, susijusias su neapibrėžtumu ar atsitiktinumu.
6. Kadangi klasikinės vienakriterės tikslo funkcijos nėra efektyvios sprendžiant paieškos diskrečioje erdvėje problemas (Deb and Goyal 1998; Deb 1995), tai jos taip pat bus neefektyvios sprendžiant daugiakriterio optimizavimo problemas su paieška diskrečioje erdvėje.

1.2. Globalaus optimizavimo metodai

Remiantis GO metodų taksonomija, optimizavimo algoritmai gali būti skirstomi į dvi pagrindines kategorijas: deterministinius ir tikimybinus (stochastinius) algoritmus (1.6 pav.). Deterministiniai algoritmai visada teikia tą patį apibrėžtą optimizavimo uždavinio sprendinį. Dažniausiai tokie algoritmai remiasi jautrumo analizės (t. y. tikslo funkcijos gradientų projektavimo kintamųjų atžvilgiu) informacija ir reikalauja itin didelių skaičiavimo resursų.



1.6 pav. Optimizavimo algoritmų klasifikacija (Weise 2009)
 Fig. 1.6. Classification of optimization algorithms (Weise 2009)

¹State Space Search, ²Branch and Bound, ³Algebraic Geometry, ⁴Hill Climbing, ⁵Random Optimization, ⁶Simulated Annealing (SA), ⁷Tabu Search (TS), ⁸Parallel Tempering, ⁹Stochastic Tunneling, ¹⁰Direct Monte Carlo Sampling, ¹¹Monte Carlo Algorithms, ¹²Evolutionary Computation (EC), ¹³Evolutionary Algorithms (EA), ¹⁴Genetic Algorithms (GA), ¹⁵Learning Classifier System (LCS), ¹⁶Evolutionary Programming, ¹⁷Evolution Strategy (ES), ¹⁸Genetic Programming (GP), ¹⁹Soft Computing, ²⁰Memetic Algorithms, ²¹Harmonic Search (HS), ²²Swarm Intelligence (SI), ²³Ant Colony Optimization (ACO), ²⁴Particle Swarm Optimization (PSO), ²⁵Differential Evolution (DE), ²⁶Standard Genetic Programming, ²⁷Linear Genetic Programming, ²⁸Grammar Guided Genetic Programming, ²⁹Artificial Intelligence (AI), ³⁰Computational Intelligence (CI).

Deterministiniais optimizavimo algoritmais įmanoma išspręsti tik palyginti nedidelės apimties uždavinius. Didesnės apimties uždaviniams efektyviau naudoti tikimybinius algoritmus. Pirmieji darbai šioje srityje pradėti jau apie 1950 metus (Bremermann 1962; Friedberg 1958; Robbins and Monro 1951) bei tapo vienu iš svarbiausių optimizavimo tyrimų sričių. Reikia pažymėti, kad stochastiniai GO metodai negarantuoja, jog tikrai bus rastas globalus sprendinys, bet dažniausiai galima prognozuoti, jog aukštos kokybės sprendinys bus surastas per priimtina laiką.

1.2.1. Monte-Karlo metodas

Ypatingai svarbi tikimybinių algoritmų šeima yra Monte-Karlo (*angl.* Monte-Carlo) algoritmai. Monte-Karlo algoritmas – tai atsitiktinės atrankos algoritmas, kurio rezultatai gali būti neteisingi su tam tikra (paprastai maža) tikimybe. Tai nereiškia, kad naudojant šiuos algoritmus gaunami neteisingi rezultatai – tiesiog jie nebūtinai yra globalus optimumas (Weise 2009). Kita vertus, šiek tiek prastesnis nei geriausias įmanomas sprendinys yra geriau, nei tas (geriausias), kuriam rasti reikia daugybės metų.

Monte-Karlo metodas – Monte-Karlo algoritmų klasė, kuri imituoja įvairių tipų sistemų elgseną ir yra laikomas paprasčiausiu nedeterministiniu GO metodu (Mackeown 2001; Liu 2002; Robert and Casella 2004). Šį metodą 1946 metais Los Alamos mokslinėje laboratorijoje sukūrė Stanislavas Ulamas (*Stanislaw Ulam*) ir Džonas fon Neimanas (*John von Neumann*). Monte-Karlo metodas remiasi atsitiktinių dydžių generavimu ir yra priskiriamas prie skaitinių matematiinių uždavinių sprendimo metodų. Svarbiausi metodo bruožai:

- labai nesudėtingas metodo realizavimas: sudaroma programa, atliekanti vieną nepriklausomą atsitiktinį bandymą, o po to bandymas kartojamas N kartų, t.y. atliekami N vienas nuo kito nepriklausomų atsitiktinių bandymų;
- gauto sprendinio tikslumas priklauso nuo atliktų bandymų skaičiaus N . Tikimybė surasti globalųjį sprendinį artėja prie 100%, kai N artėja prie begalybės.

Nors Monte-Karlo metodas ir nėra efektyvus GO metodas, tačiau dėl itin nesudėtingo jo realizavimo, kai kurių tyrėjų jis naudojamas naujų metodų kokybės įvertinimui (Huang and Du 2006; Mockus 2000). Jei naujuoju metodu gaunami rezultatai yra prastesni nei naudojant Monte-Karlo metodą – tai toks metodas laikomas netinkamu.

Inžinieriai gana dažnai modifikuoja klasikinį Monte-Karlo metodą įtraukdami įvairias euristicas, tokiu būdu siekdami gauti geresnius sprendinius (Fonseca *et al.* 2007).

1.2.2. Evoliuciniai skaičiavimai

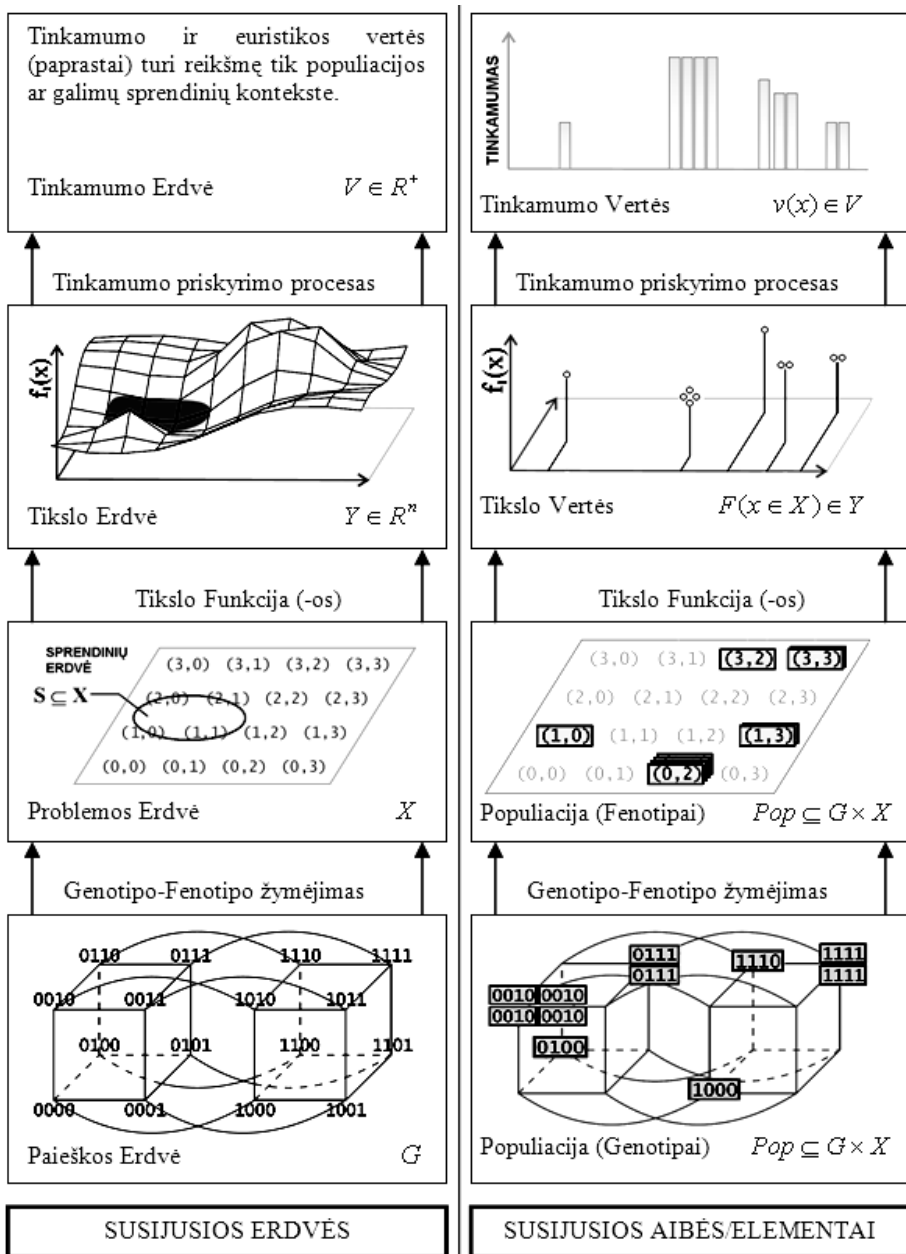
Svarbi tikimybinių Monte-Karlo metaeuristikų klasė yra Evoliuciniai skaičiavimai (Heitkotter and Beasley 1998). Šie metodai apima visus algoritmus, kurie yra pagrįsti daugybinių sprendinių kandidatų aibe (vadinamą populiacija), kuri yra gerinama su kiekviena iteracija. Sudėtinėmis šios optimizavimo srities dalimis taip pat yra Minkšto programavimo (*angl.* Soft Computing) bei Dirbtinio intelekto (*angl.* Artificial Intelligence) skyriai. Vieni iš svarbiausių šiuos skyrius sudarančių narių yra Evoliuciniai algoritmai (*angl.* Evolutionary Algorithms) (Alba and Chicano 2006; Abraham *et al.* 2005; Coello *et al.* 2002; Schwefel 1993) ir Spiečiaus intelektas (*angl.* Swarm Intelligence) (Bonabeau *et al.* 1999; Kennedy *et al.* 2001; Engelbrecht 2006; Chan and Tiwari 2007).

1.2.3. Genetiniai algoritmai

Viena populiariausių Evoliucinių algoritmų atšaka – Genetiniai algoritmai (GA) (*angl.* Genetic Algorithms). Pastaraisiais metais sulaukta ypatingo susidomėjimo šiais algoritmais. GA yra pagrįsti gamtoje egzistuojančiais gyvybės evoliuciniais mechanizmais: natūralia atranka, paveldėjimu, mutacija, rekombinacija (kryžminimu). GA yra priemonė, leidžianti kompiuterio pagalba atkartoti tam tikros populiacijos individų evoliuciją, artėjant prie geriausio užduoties sprendinio. Sprendinių paieška remiasi sprendinių (populiacijų) generavimu ir įvertinimu.

Genetinių algoritmų šaknys siekia 1950 metų vidurį, kai biologas (Barricelli 1954; Barricelli 1957; Barricelli 1962; Barricelli 1963) ir kompiuterinių skaičiavimų mokslų atstovas (Fraser 1957) pradėjo taikyti kompiuterines simuliacijas, siekdami išigilinti į genetinius procesus bei natūralią evoliuciją ir atranką. (Bremermann 1962) ir (Bledsoe 1962a; Bledsoe 1962b; Bledsoe 1961) pritaikė evoliucijos dėsnius sprendžiant nelygybes ir optimizuojant funkcijas. Genetinių algoritmų koncepciją pirmasis pasiūlė Džonas H. Hollandas (John H. Holland) 1975 metais (Holland 1975). Remdamiesi jo darbais (Holland 1967; Holland 1969), svarbius tyrimus, susijusius su paieškos erdve, atliko (Bagley 1967), (Rosenberg 1967), (Frantz 1972). Nuo to laiko genetiniai algoritmai sparčiai vystėsi bei populiarėjo, tad buvo sukurta daug įvairių jo formų bei modifikacijų (Coello 2000).

Optimizavimo struktūra naudojant GA (1.7 pav.) yra sudaryta iš erdvių ir aibių bei perėjimų (*angl.* mapping) tarp jų. Problemos erdvė (problemos galimų



1.7 pav. Optimizavimo procesų struktūra genetiniu algoritmu (Weise 2009)

Fig. 1.7. Structure of optimization processes using genetic algorithm (Weise 2009)

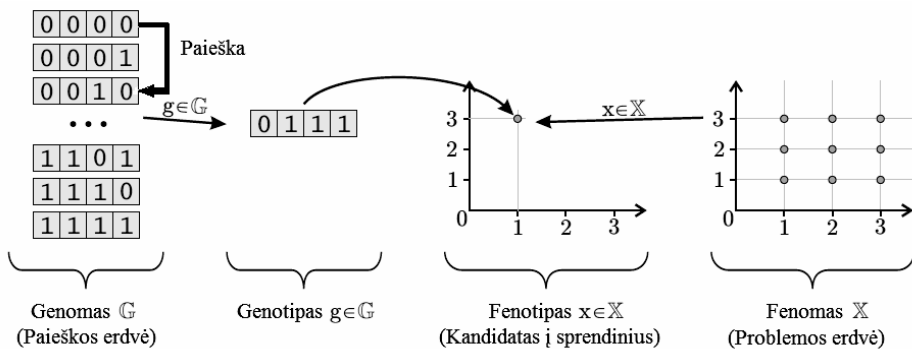
sprendinių erdvė) X (fenomas) – tai visų elementų x , galinčių būti optimizavimo problemos sprendiniais, aibė. Kiekviena optimizavimo problema gali būti apibrėžiama keliomis problemos erdvėmis.

Problemos erdvei dažnai taikomi šie apribojimai:

- loginiai apribojimai. Jie atmeta elementus, kurie negali būti sprendiniais;
- praktiniai apribojimai. Jie neleidžia, pavyzdžiui, minimizuojant realiąją funkciją, nagrinėti visų be išimties realių skaičių.

Tam tikros optimizavimo problemos galimas sprendinys arba kandidatas į sprendinius (*angl.* Solution Candidate) x – tai problemos erdvės X elementas. Šie elementai x dar vadinami fenotipais. Visų optimizavimo problemos sprendinių aibė yra vadinama sprendinių erdve S . Sprendinių erdvę sudaro (bei gali būti jai lygi) globali optimali aibė X^* : $X^* \subseteq S \subseteq X$. Be to, gali egzistuoti sprendinių $x \in S$, kurie nepriklauso X^* aibės elementams.

Optimizavimo problemos paieškos erdvė G – tai visų elementų g , kurie gali būti apdoroti paieškos operacijomis, aibė. Genetiniuose algoritmuose paieškos erdvė vadinama genomu, kurio terminą apibrėžė vokiečių biologas Winkler: genomas – tai visa organizmo paveldima informacija, kuri sudaryta iš genų ir chromosomų (Winkler 1920). Optimizavimo problemos paieškos erdvės G elementai $g \in G$ yra vadinami genotipais. Tam tikra genotipo informacija, kuri užkoduoja fenotipines savybes, vadinama genais.



1.8 pav. Ryšys tarp geno ir fenomo (Weise 2009)

Fig. 1.8. The relation of genome and phenotype (Weise 2009)

Kai kurioms problemoms paieškos erdvė G gali būti identiška problemos erdvei X . Jei paieškos erdvė skiriasi nuo problemos erdvės, tai tuomet reikalingi perėjimai nuo vienos erdvės prie kitos. 1.8 paveiksle yra parodyta kokiais ryšiais yra susiję paieškos erdvė (genomas) ir problemos erdvė (fenomas).

Tikslo erdvė Y – tai erdvė apie tikslo funkcijų poaibius:

$$F = \{f_i : X \mapsto Y_i : 0 < i \leq n, Y_i \subseteq R\} \Rightarrow Y = Y_1 \times Y_2 \times \dots \times Y_n. \quad (1.5)$$

Aibė F tikslo erdvėje Y sužymi problemos erdvės X elementus x ir tokiu būdu suteikia informaciją optimizatoriui apie problemos sprendinių kokybę.

Problemos erdvės X elemento x tinkamumo (*angl.* Fitness) vertė $v(x) \in V$ atitinka jo, kaip sprendinio, naudingumą arba jo pirmenybę tolesniuose optimizavimo proceso žingsniuose. Erdvė apie visas galimas tinkamumo vertes V paprastai yra teigiamų realiųjų skaičių poaibis $V \subseteq R^+$.

Tinkamumo terminą evoliucinių algoritmų bendruomenė pasiskolino iš biologijos (Paul 1992; Sober 2001). Kai buvo sukurti pirmieji genetinių algoritmų pritaikymai, pagrindinis dėmesys buvo skirtas vienakriteriam optimizavimui. Todėl vienakriterė funkcija buvo vadinama tinkamumo funkcija (*angl.* Fitness Function). Nors šis požiūris iš principo yra pasenęs, tačiau sprendžiant paprastas problemas (apibrėžiamas vienakriterė tikslo funkcija) galima naudoti šį traktavimą sutapatinant tikslo funkcijos reikšmę su tinkamumo reikšme, t.y. $v(x) = f(x)$, $x \in X$. Tačiau nagrinėjant daugiakriterio optimizavimo problemas yra susiduriama su tinkamumo priskyrimo procesu (*angl.* Fitness Assignment Process). Šio proceso metu yra sukuriama funkcija $v: X \mapsto R^+$, kuri priskiria tinkamumo skaliarinę reikšmę kiekvienam populiacijos individui. Tokiu būdu yra įvertinama ne tik individo (sprendinio) kokybė, bet taip pat ir visos populiacijos įvairovė. Todėl individo (sprendinio) tinkamumas priklauso ne tik nuo pačio sprendinio, bet ir nuo visos populiacijos. Vienas iš labiausiai paplitusių tinkamumo priskyrimo procesų yra jau aukščiau (1.1.3 poskyryje) minėta svertinė kriterijų suma.

Paieškos operacijos yra naudojamos optimizavimo algoritmuose, kad būtų galima ištirti bei išanalizuoti paieškos erdvę G . Genetiniuose algoritmuose yra naudojamos tokios paieškos operacijos kaip mutacija ir kryžminimas.

Populiacija Pop – tai sąrašas individų, panaudotų optimizavimo proceso metu:

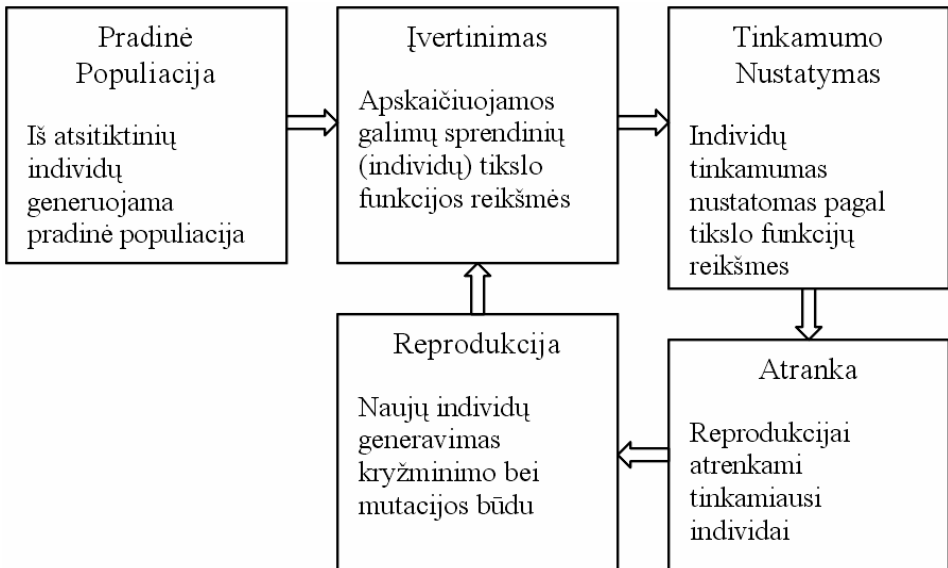
$$Pop \subseteq G \times X, \quad (1.6)$$

Individų kodavimo būdas labai priklauso nuo sprendžiamos optimizavimo problemos. Dažniausiai individai yra koduojami bitų (vienetų ir nulių) arba realiųjų skaičių sekomis (Weise 2009).

Norint genetiniu algoritmu surasti problemos sprendimą, būtini du pagrindiniai elementai: tinkama pradinė duomenų struktūra ir atrankos kriterijus. Paprastai genetiniam algoritmui reikalinga, jog problemos sprendimas galėtų būti pateikiamas kaip duomenų struktūra, kad būtų galima lengvai kurti šios struktūros pakeistas kopijas. Antrasis elementas – tai tikslo funkcija, aprašanti atrankos kriterijų, kurios dėka galima taikyti kiekybinius atrankos kriterijus, atrenkant „geresnius“ ir „blogesnius“ sprendinius.

Pagrindinis GA ciklas (1.9 pav.) sudarytas iš tokių stadijų:

- Generuojama pradinė populiacija.
- Ciklo pradžia:
 1. Individų įvertinimas, tinkamumo nustatymas ir atranka.
 2. Atliekama reprodukcija (vyksta kryžminimas ir mutacija) ir gaunama nauja populiacija.
- Ciklas kartojamas, jeigu netenkinama nutraukimo sąlyga.



1.9 pav. Pagrindinis genetinio algoritmo ciklas (Weise 2009)

Fig. 1.9. The basic cycle of genetic algorithm (Weise 2009)

Genetiniuose algoritmuose individai (chromosomos) dažniausiai yra koduojami tokiais būdais: bitų seka, natūrinių skaičių seka, realiųjų skaičių seka, raidžių/ženklų seka, loginių reikšmių seka. Kodavimo būdo pasirinkimas priklauso nuo sprendžiamos problemos pobūdžio.

1.9 paveiksle parodytos GA ciklo stadijos. Disertacijoje pademonstruotas GA veikimas pasitelkus pavyzdį su realiųjų skaičių sekomis. GA ciklo stadijos pasitelkus bitų sekas yra detaliai pateiktos (Šešok 2008).

Pradinės populiacijos generavimas – optimizavimas paprastai inicijuojamas sugeneruojant atsitiktinius genotipus g , t.y. sugeneruojami optimizavimo problemos paieškos erdvės G (genomo) elementai $g \in G$. Tai atliekama generuojant realiųjų skaičių sekas (individus). Iš anksto apibrėžiamas sekų skaičius (individų skaičius populiacijoje), kuris turi būti sugeneruotas. Populiacijos dydis pasirenkamas pagal problemos sudėtingumą. Tradiciškai pradinė populiacija generuojama atsitiktiniu būdu, apimant didelę sprendinių paieškos erdvę. Tarkime, kad buvo sugeneruota 4 individų populiacija (1.1 lentelė).

1.1 lentelė. Pradinė individų populiacija
Table 1.1. Initial population of individuals

Individo (sekos) nr.	Sugeneruotas individas (seka)	Tikslo funkcijos reikšmė, f	Tikslo funkcijos reikšmių suma $\sum f$
1	10,3 2,9 37,1 0,5 8,4	90	300
2	22,0 0,2 65,8 4,7 0,1	120	
3	78,5 6,0 14,3 2,6 9,2	60	
4	91,4 1,6 73,5 8,4 3,0	30	

Populiacijos individų įvertinimas (*angl.* evaluation) – apskaičiuojama kiekvieną populiacijos individą (galimą sprendinį) atitinkanti tikslo funkcijos reikšmė f_i , $f \in F$, $F = \{f_1, f_2, \dots, f_n\}$. Nuo optimizavimo problemos ir pasirinkto individų kodavimo būdo priklauso kaip kiekvienu atveju bus apskaičiuojama tikslo funkcijos reikšmė. Nesigilindami į tai kaip apskaičiuojama kiekvieno individo tikslo funkcijos reikšmė tarkime, kad mokame tai apskaičiuoti.

Populiacijos individų tinkamumo nustatymas (*angl.* fitness assignment) – pagal tikslo funkcijos reikšmes yra nustatomas vieno individo pranašumas lyginant su kitu individu. Pranašesni individai yra tinkamesni būti atrinktiems reprodukcijai. Taigi, nustatomos kiekvieno individo tinkamumo vertės $v(x)$. Jei ieškomas globalus minimumas, tai tinkamesni yra tie individai, kurių tikslo funkcijos reikšmė yra mažesnė, o globalaus maksimumo atveju – kurių tikslo funkcijos reikšmė yra didesnė.

Atranka (*angl.* selection) – reprodukcijai (naujai populiacijai formuoti) atrenkami tinkamiausi individai. Egzistuoja gana daug atrankos metodų bei jų modifikacijų, iš kurių populiariausi yra šie:

- Ruletės rato atranka (*angl.* Roulette Wheel Selection) (De Jong 1975; Goldberg 1989).
- Turnyrinė atranka (*angl.* Tournament Selection) (Brindle 1981; Wetzel 1983; Blickle and Thiele 1995; Miller and Goldberg 1996; Lee *et al.* 2008).
- Klasifikavimo (reitingų) atranka (*angl.* Ranking Selection) (Baker 1985; Whitley 1989; Goldberg and Deb 1990).
- VEGA atranka (*angl.* VEGA Selection) (Schaffer 1984; Richardson *et al.* 1989; Fonseca and Fleming 1993).

Ruletės rato atranka yra vienas žinomiausių atrankos metodų. Tarkime, kad optimizuojant yra ieškomas globalus maksimumas. Tuomet individas, atitinkantis didesnę tikslo funkcijos reikšmę, turi didesnę galimybę (tikimybę) būti atrinktas reprodukcijai. Ant ruletės rato individai išdėstomi taip (1.10 pav.), kad ruletės segmentų plotas būtų tiesiogiai proporcingas individų tikslo funkcijos reikšmėms (Weise 2009). Sukant ruletės ratą, tinkamesni (užimantys didesnę ruletės rato plotą) individai turi didesnę tikimybę būti atrinktais reprodukcijai, ir atvirkščiai. Pateikto pavyzdžio atveju gauname, kad visų populiacijoje esančių individų tikslo funkcijų reikšmių suma yra lygi 300 ($90 + 120 + 60 + 30 = 300$). Tuomet yra gaunama informacija apie tai, kokią santykinę dalį (procentais) nuo šios sumos sudaro populiacijoje esančio atskiro individo tikslo funkcijos reikšmė bei kokia yra tikimybė, kad bet kuris individas būtų atrinktas reprodukcijai (1.2 lentelė).

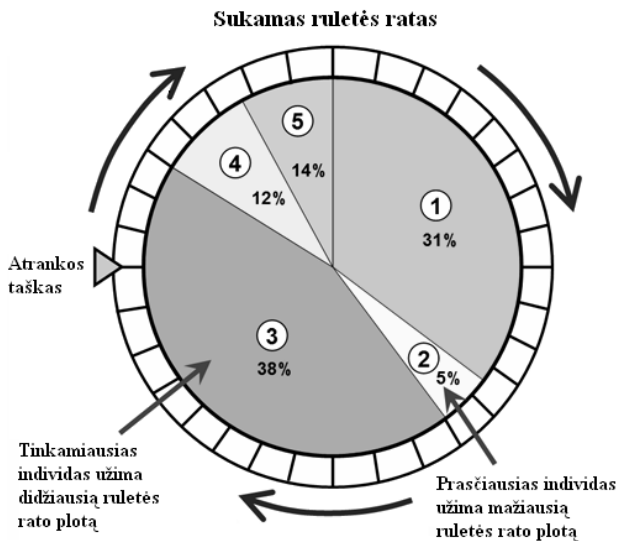
1.2 lentelė. Pradinės individų populiacijos charakteristikos
Table 1.2. Characteristics of initial population of individuals

Individo nr.	Sugeneruotas individas (seka)	Tikslo funkcijos reikšmė, f	Santykinė dalis nuo $\sum f$	Atrankos tikimybė %
1	10,3 2,9 37,1 0,5 8,4	90	0,3	30
2	22,0 0,2 65,8 4,7 0,1	120	0,4	40
3	78,5 6,0 14,3 2,6 9,2	60	0,2	20
4	91,4 1,6 73,5 8,4 3,0	30	0,1	10

Todėl šiuo atveju (ieškant globalaus maksimumo), didžiausią tikimybę (40%) būti atrinktam į kitą populiaciją turi individas (nr. 2) su didžiausia tikslo funkcijos reikšme (120). Tuo pačiu, mažiausią tikimybę (10%) būti atrinktam į kitą populiaciją turi individas (nr. 4) su mažiausia tikslo funkcijos reikšme (30).

Ruletės rato atranka (kaip ir dauguma atrankos funkcijų) yra sukurta taip, kad leistų ir nedidelę dalį sprendinių, kurie yra mažiau tinkami. Tokiu būdu užtikrinama, kad išliktų individų (sprendinių) įvairovė ir būtų išvengta pirmalaikio konvergavimo į lokalų optimumą.

Antroji ir po jos sekančios populiacijos yra kuriamos pritaikant tokį genetinį mechanizmą kaip reprodukcija. Reprodukcija – tai kryžminimo ir mutacijos kombinacija.



1.10 pav. Ruletės rato atranka

(<http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g02.php/>)

Fig. 1.10. Roulette wheel selection

(<http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g02.php/>)

Kryžminimas (*angl.* crossover) – tai reprodukcijai atrinktų individų (genotipų) apsikeitimas genais pagal tam tikrą dėsnį. Tarp visų evoliucinių algoritmų, genetiniai algoritmai turi kryžminimo operaciją, kuri galbūt labiausiai artima natūraliai evoliucijai.

Priklausomai nuo kryžminimo pozicijų skaičiaus, galima išskirti tokius kryžminimo būdus:

- vienos pozicijos kryžminimas;
- dviejų pozicijų kryžminimas;
- daugiapozicinis (n - pozicijų) kryžminimas.

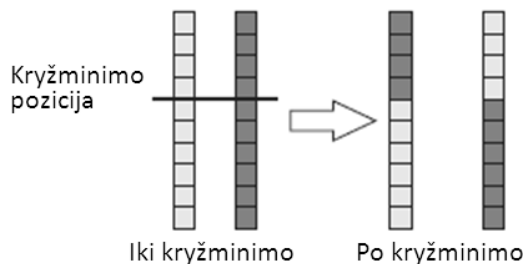
Kryžminimo proceso metu populiacijos individai yra sugrupuojami po du ir atliekant vienos pozicijos kryžminimą, abu individai yra padalinami atsitiktinai nustatytoje pozicijoje j :

$$j \in [0; k - 1], \quad (1.7)$$

čia j – atsitiktinis skaičius, priklausantis sveikųjų skaičių intervalui, k – individo genų skaičius. Jei $j = 0$, tai kryžminimo operacija nebus vykdoma. Tokiu atveju individų pora išlieka nepakitusi.

Taigi, du nauji individai (genotipai) yra sukuriami taip (1.11 pav.), kad:

- pirmasis naujas individas susideda iš pirmojo individo pirmosios dalies ir antrojo individo antrosios dalies;
- antrasis naujas individas susideda iš pirmojo individo antrosios dalies ir antrojo individo pirmosios dalies;



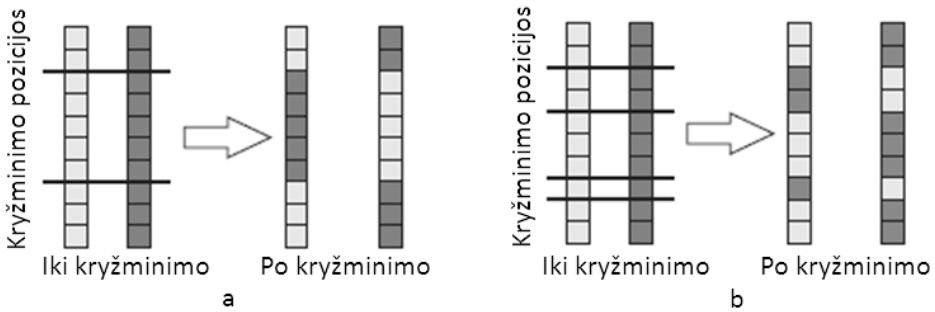
1.11 pav. Vienos pozicijos kryžminimas (Weise 2009)

Fig. 1.11. Single-point crossover (Weise 2009)

Atliekant dviejų pozicijų kryžminimą, abu individai yra padalinami dviejose pozicijose ir du nauji individai (genotipai) yra sukuriami taip (1.12 a pav.), kad:

- pirmasis naujas individas susideda iš pirmojo individo pirmosios bei trečiosios dalies ir antrojo individo antrosios dalies;
- antrasis naujas individas susideda iš pirmojo individo antrosios dalies ir antrojo individo pirmosios ir trečiosios dalies.

Analogiškai, atliekamas ir daugiapozicinis kryžminimas (1.12 b pav.).



1.12 pav. Dviejų pozicijų (a) ir daugiapozicinis (b) kryžminimas (Weise 2009)

Fig. 1.12. Two-point (a) and multi-point (b) crossover (Weise 2009)

Jei joks kitas dėsnis (ar taisyklė) nėra apibrėžtas, galima teigti, kad kryžminimo tikimybė $p_{kryž}$ remiantis formule (1.7) gali būti apskaičiuojama:

$$p_{kryž} = \left(1 - \frac{1}{k}\right) \cdot 100\% , \tag{1.8}$$

čia k – individo (genotipo) ilgis (genų skaičius). Nagrinėjant šį pavyzdį (1.3 lentelė) kryžminimo tikimybė yra 80% ($p_{kryž} = \left(1 - \frac{1}{5}\right) \cdot 100\%$). Kryžminimo procese dalyvavusių individų porų apsikeitę genai 1.3 lentelėje yra pažymėti paryškintu šriftu. Taigi, didėjant individų ilgiui, kryžminimo tikimybė didėja, ir atvirkščiai.

1.3 lentelė. Populiacijos individų kryžminimas
Table 1.3. Crossover of individuals in population

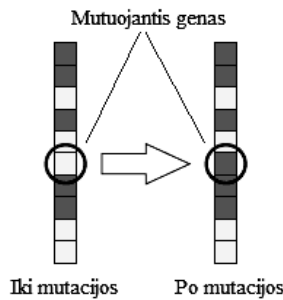
Individų iki kryžminimo		Individų kryžminimas		Individų po kryžminimo	
Nr.	Genai	Poros nr.	Pozicija	Nr.	Genai
1	10,3 2,9 37,1 0,5 8,4	1	2	1	10,3 2,9 14,3 2,6 9,2
3	78,5 6,0 14,3 2,6 9,2			2	78,5 6,0 37,1 0,5 8,4
2	22,0 0,2 65,8 4,7 0,1	2	3	3	22,0 0,2 65,8 2,6 9,2
3	78,5 6,0 14,3 2,6 9,2			4	78,5 6,0 14,3 4,7 0,1

Jei kryžminimo tikimybė yra pastovi (pvz. $p_{kryž} = 65\% = const$), tai tikimybė, kad kryžminimo pozicija generuojama intervale $[i; k - 1]$, lygi 65%, o 35% tikimybė, kad bus generuojama nulinė kryžminimo pozicija.

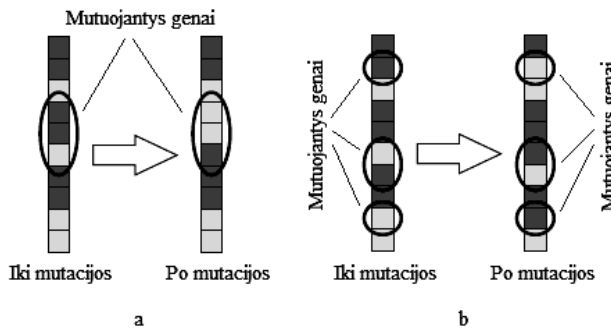
Mutacija (*angl.* mutation) – tai reprodukcijai atrinkto individo (genotipo) geno reikšmės atsitiktinis pasikeitimas pagal tam tikrą dėsnį. Taigi, mutaciją galima apibūdinti kaip metodą, padedantį išsaugoti individų įvairovę. Galima išskirti du mutacijos tipus:

1. vieno geno mutacija (1.13 pav.);
2. multigeninė mutacija (1.14 pav.).

Bitų seka užkoduoto individo genas su tam tikra tikimybe pakeičia savo reikšmę į priešingą.



1.13 pav. Vieno geno mutacija (Weise 2009)
 Fig. 1.13. Single-gene mutation (Weise 2009)



1.14 pav. Multigeninė mutacija (Weise 2009)
 Fig. 1.14. Multi-gene mutation (Weise 2009)

Genotipo (individo) g , užkoduoto realiųjų skaičių seka, elemento (geno) g_i modifikavimas gali būti atliktas, pakeičiant jį skaičiumi iš normaliojo skirstinio, kurio tikėtinas vidurkis g_i :

$$g_i^{\text{naujas}} \sim N(g_i, \sigma^2), \quad (1.9)$$

čia σ^2 – dispersija.

1.4 lentelėje yra pateiktas pavyzdys, kaip mutacija gali modifikuoti populiacijos individus. Mutacijos metu modifikuoti genai 1.4 lentelėje yra pažymėti paryškintu šriftu.

1.4 lentelė. Populiacijos individų mutacija
Table 1.4. Mutation of individuals in population

Individų iki mutacijos		Mutuojančio geno nr.	Individų po mutacijos	
Nr.	Genai		Nr.	Genai
1	10,3 2,9 14,3 2,6 9,2	2	1	10,3 5,7 14,3 2,6 9,2
2	78,5 6,0 37,1 0,5 8,4	3	2	78,5 6,0 71,4 0,5 8,4
3	22,0 0,2 65,8 2,6 9,2	4	3	22,0 0,2 65,8 8,3 9,2
4	78,5 6,0 14,3 4,7 0,1	5	4	78,5 6,0 14,3 4,7 9,7

Taigi, pritaikius GA operatorius – atranką, kryžminimą ir mutaciją – iš pradinės individų populiacijos gaunama nauja individų populiacija (1.5 lentelė). Gavus naują individų populiaciją, pakinta tuos individus atitinkančių tikslo funkcijų reikšmės, kurių vienos galėjo priartėti prie globalaus sprendinio, o kitos – nutolti. Kiekvienai naujai populiacijai cikliškai taikant aprašytus GA operatorius, galima įvertinti gaunamų sprendinių progresą.

1.5 lentelė. Pradinė ir nauja populiacijos
Table 1.5. Initial and new populations

Individo nr.	Pradinės populiacijos individai	Naujos populiacijos individai
1	10,3 2,9 37,1 0,5 8,4	10,3 5,7 14,3 2,6 9,2
2	22,0 0,2 65,8 4,7 0,1	78,5 6,0 71,4 0,5 8,4
3	78,5 6,0 14,3 2,6 9,2	22,0 0,2 65,8 8,3 9,2
4	91,4 1,6 73,5 8,4 3,0	78,5 6,0 14,3 4,7 9,7

Taigi, galima išskirti esminius GA bruožus:

1. Optimizuojant gali būti atrenkami ne tik tinkamesni sprendiniai, bet ir nedidelė dalis mažiau tinkamų sprendinių. Tokiu būdu užtikrinama, kad išliktų sprendinių (individų) įvairovė ir būtų išvengta pirma laikio sprendinio konvergavimo į lokalų optimumą. Todėl galima teigti, kad GA yra tinkamas algoritmas globalaus optimizavimo uždavinių sprendimui, t.y. GA gali surasti globalųjį optimumą.
2. Ieškodamas tinkamesnio sprendinio GA dirba su visa individų populiacija, o ne su vienu atskiru individu. Taigi, prie globalaus sprendinio artėja visa individų populiacija.
3. Algoritmą su tais pačiais genetiniais parametrais būtina vykdyti kelis kartus, nes GA naudoja atsitiktinius dydžius.
4. Algoritmo nutraukimui būtina apibrėžti kriterijų (pvz. iteracijų skaičių), nes neįmanoma nustatyti, kada GA suras geriausią uždavinio sprendinį.

1.3. Genetinių algoritmų taikymo sijynų optimizavimui apžvalga

Optimalios būsenos (struktūros) paieška – tai vienas pagrindinių principų pasaulyje. Natūralios gamtos principas – Išlieka tik stipriausi (*angl.* Survival of the Fittest) – (Spencer 1864) bei biologinės evoliucijos dėsniai (Darwin 1859) lemia geresnį visų gyvybės rūšių prisitaikymą prie supančios aplinkos. Čia lokalus optimumas – tai gerai prisitaikiusios gyvybės rūšys, kurios supančioje aplinkoje vyrauja kitų gyvybės formų atžvilgiu.

Mokslinis susidomėjimas inžinerinių struktūrų optimizavimu prasidėjo maždaug XX a. pradžioje (Michell 1904). Apytiksliai iki XX a. vidurio buvo vykdomi tik analitiniai skaičiavimai, vėliau vis plačiau pradėti naudoti kompiuteriai, o globaliuoju optimizavimu ypatingai susidomėta nuo XX a. 8-to dešimtmečio. Šiais laikais optimizavimo problemos sprendžiamos pasitelkiant teorines žinias bei kompiuterinius išteklius, kas ypač palengvina sprendimo metodų parinkimą ir uždavinių formalizavimą bei žymiai paspartina uždavinių sprendimą.

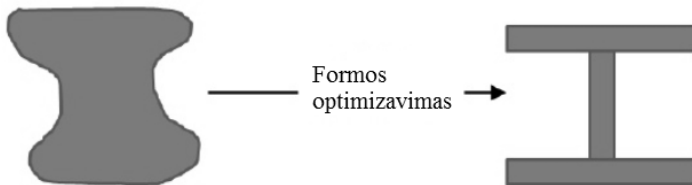
Struktūrų optimizavimas pateikia tokius rezultatus, kurie atskleidžia optimalias duotas projektavimo erdvės struktūrines charakteristikas veikiant apibrėžtomis apkrovoms bei esant apibrėžtomis kraštinėms sąlygoms. Kadangi daugelis inžinerinių ar taikomojo mokslo problemų yra aprašomos diferencinėmis arba integralinėmis lygtimis, todėl jų sprendimas pateikia tikslius sprendinius (rezul-

tatus). Tačiau dėl daugelio realaus pasaulio problemų sudėtingumo tikslių sprendinių gauti neįmanoma arba neįmanoma jų gauti per priimtina laiką. Todėl yra siekiama per priimtina laiką gauti aproksimuotus sprendinius su priimtinu tikslumu. Šiuo tikslu gali būti naudojamas vienas labiausiai paplitusių metodų – baigtinių elementų metodas (BEM). BEM – skaitinis funkcionalinio uždavinio sprendimo metodas, kai uždavinys suskaidomas į mažas dalis (uždavinius), kuriuos išsprendus gaunamas viso uždavinio sprendinys. Kontinualios struktūros naudojant BEM diskretizuojamos (suskaidomos) į elementus, t. y. sudaromas elementų tinklėlis. Elementų savybės bei ryšiai tarp jų yra aprašomi matematiškai kaip nežinomos reikšmės specifiniuose elementų taškuose, vadinamuose mazgais (Oate 2009). Taip yra pereinama nuo kontinualios struktūros su begaliniu laisvės laipsnių skaičiumi prie diskretaus modelio su baigtiniu laisvės laipsnių skaičiumi. Elementai baigtinių elementų tinklėlyje paprastai kuriami naudojant tris realiųjų skaičių rinkinius (Leiva *et al.* 2007):

1. tinklelio koordinates;
2. geometrines charakteristikas, aprašančias fizinius matmenis;
3. medžiagos charakteristikas, apibūdinančias fizinę medžiagą, iš kurios bus pagaminta struktūra.

Taigi, remiantis BEM egzistuoja trys pagrindiniai struktūrų optimizavimo tipai (Leiva *et al.* 2007; Christensen and Klarbring 2008), kai elementai yra optimizuojami visi kartu arba grupėmis:

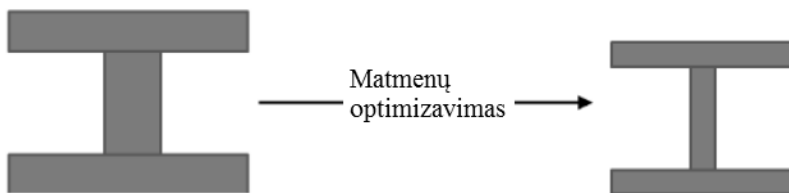
1. formos optimizavimas (t. y. optimizuojamas mazgų tinklėlyje išsidėstymas) (1.15 pav.);
2. matmenų optimizavimas (1.16 pav.);
3. medžiagos charakteristikų optimizavimas.



1.15 pav. Formos optimizavimas

(<http://www.scribd.com/doc/28808604/An-Introduction-to-Structural-Optimization>)

Fig. 1.15. Shape optimization (<http://www.scribd.com/doc/28808604/An-Introduction-to-Structural-Optimization>)



1.16 pav. Matmenų optimizavimas

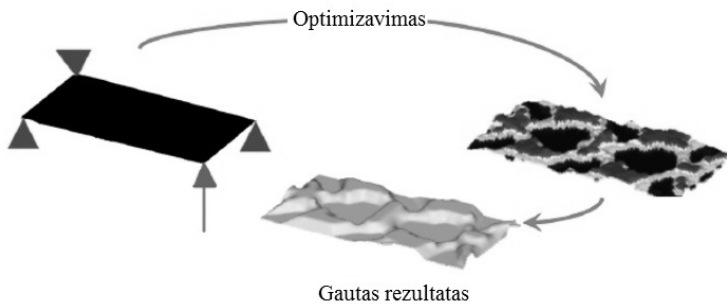
(<http://www.scribd.com/doc/28808604/An-Introduction-to-Structural-Optimization>)

Fig. 1.16. Size optimization (<http://www.scribd.com/doc/28808604/An-Introduction-to-Structural-Optimization>)

Specializuoti šių trijų optimizavimo tipų modeliai skirstomi taip:

- topografijos optimizavimas – tai specializuotas formos optimizavimo modelis, kai tinklėlis yra projektuojamas duotąja arba normaline kryptimi 2-D kevalų elementuose (kevalų elementuose plokštumoje). Pvz., iškilumų ir standumo briaunų projektavimas plastikinėms detalėms (1.17 pav.);
- topologijos optimizavimas – tai struktūrinio optimizavimo atšaka, kuri dažnai siejama su formos optimizavimu (Leiva *et al.* 2007). Topologijos optimizavimo tikslas yra gauti skirtingą struktūros formą palyginus su pradine struktūros forma. Tačiau formos ir topologijos optimizavimas yra labai skirtingas, jeigu įvertinsime kaip jie yra realizuojami. Formos optimizavimas yra susijęs su mazgų tinklėlyje išsidėstymo optimizavimu, o topologijos optimizavimas – su medžiagos savybių optimizavimu, t. y. specializuotas medžiagos optimizavimo modelis, kai kiekvieno elemento medžiagos charakteristikos projektuojamos atskirai. Optimizavimo pabaigoje elementų medžiagos charakteristikų reikmė lygi nuliui arba jų nominaliai vertei. Elementai su nulinėmis medžiagos charakteristikomis pašalinami iš konstrukcijos, o su nominaliomis vertėmis – paliekami (1.18 pav.). Kairėje pavaizduota pradinė konstrukcija, kuri bus optimizuojama, o dešinėje tamsiau pažymėti konstrukcijos elementai, kurie yra svarbiausi elementai (paliekami struktūroje po optimizavimo), tuo tarpu šviesiau pažymėti tie elementai, kurie yra pašalinami iš konstrukcijos po optimizavimo.

- topometrijos optimizavimas – tai specializuotas matmenų optimizavimo modelis, kai kiekvieno elemento matmenys projektuojami atskirai. Topometrijos optimizavimas skiriasi nuo matmenų optimizavimo tuo, kad esama papildomų reikalavimų: simetrijos sąlygos, mažiausias elemento dydis ir kt. (Leiva *et al.* 2007). Kita savybė, kurios neturi matmenų optimizavimas, bet kurią turi topometrijos optimizavimas – tai elementų išsidėstymas šaškių lentos langelių principu. 1.19 paveikslo kairėje pavaizduotas pradinis automobilio konstrukcinių elementų storio pasiskirstymas, o dešinėje šviesiau pažymėti tie elementai, kurių storis turi būti padidintas, kad būtų pasiektas norimas savųjų virpesių dažnis (Leiva *et al.* 2007);



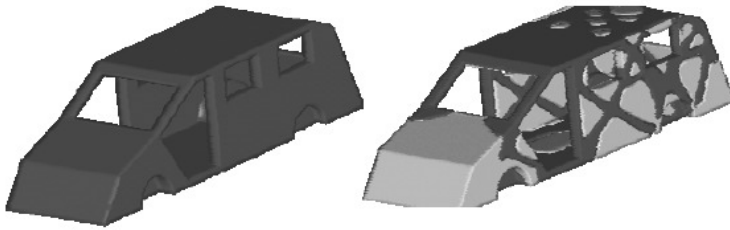
1.17 pav. Topografijos optimizavimas

(<http://www.scribd.com/doc/28808604/An-Introduction-to-Structural-Optimization>)

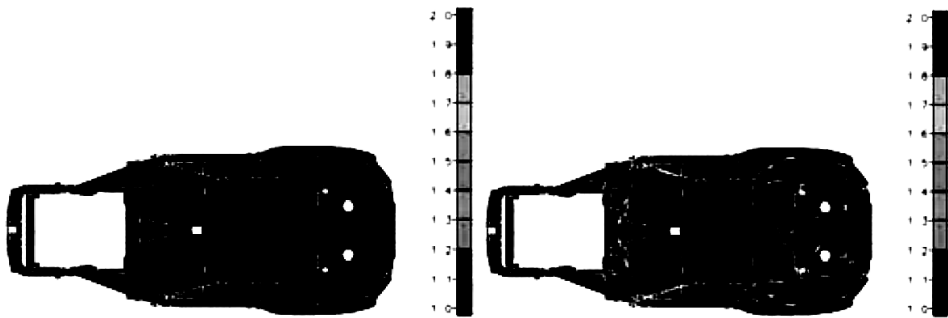
Fig. 1.17. Topography optimization

(<http://www.scribd.com/doc/28808604/An-Introduction-to-Structural-Optimization>)

Inžinerinių struktūrų globalaus optimizavimo srityje ypač didelio dėmesio sulaukė genetiniai algoritmai (Goldberg and Samtani 1986; Hajela 1992; Chapman *et al.* 1993; Chapman and Jakiela 1996; Lemonge and Barbosa 2000; Pezeshk and Camp 2002; Min and Yi-min 2007), nes juos ypatingai paranku naudoti sprendžiant diskrečiojo optimizavimo problemas, tarp jų – ir sijynų optimizavimo problemas (Hajela *et al.* 1998; Saka 1998; Saka *et al.* 2000; Kim *et al.* 2005; Belevičius and Šešok 2008; Šešok and Belevičius 2009; Belevičius *et al.* 2011).



1.18 pav. Topologijos optimizavimas (Leiva *et al.* 2007)
 Fig. 1.18. Topology optimization (Leiva *et al.* 2007)



1.19 pav. Topometrijos optimizavimas (Leiva *et al.* 2007)
 Fig. 1.19. Topometry Optimization (Leiva *et al.* 2007)

(Kim *et al.* 2001) minimizavo polių nuosėdžius po pamatų plokštėmis: tai leido tam tikrais metodais sumažinti uždavinio projektavimo kintamųjų kiekį. (Chan *et al.* 2009) apjungė sijyno polių išdėstymo topologijos optimizavimo uždavinį su pačių polių matmenų nustatymo uždaviniu, tačiau drastiškai mažinamas projektavimo kintamųjų skaičius, polių apjungiant į kelias vienodų charakteristikų grupes.

(Belevičius and Valentinavičius 2000; Belevičius and Valentinavičius 2001; Belevičius *et al.* 2002) sprendė sijynų optimizavimo uždavinį nagrinėjant kiekvieną sijyno siją atskirai ir siekiant vienu atveju mažiausių lenkimo momentų kiekvienoje sijoje atskirai, kitu atveju – mažiausių atraminių reakcijų poliuose, trečiu atveju – iš pradžių mažiausių atraminių reakcijų, po to nuosekliai pereinant prie mažiausių lenkimo momentų paieškos.

XX a. pabaigoje – XXI a. pradžioje genetiniai algoritmai (GA) tampa vis populiariesni (Coello 1996; Dinger 1998; Yeo and Agyei 1998; Zheng *et al.*

1999; Jakiela *et al.* 2000; Lemonge *et al.* 2003), be to jų populiarumas auga ir toliau (Zuo *et al.* 2009; Al-Shihri 2010; Guo *et al.* 2011), kadangi lyginant su kitais metodais GA leidžia gana lengvai išvengti tam tikrų apribojimų, pvz., nėra būtina žinoti, ar tikslo funkcija yra tolydi, nėra būtina skaičiuoti tikslo funkcijos išvestines ir kt. Optimizuojamą tikslo funkciją galima traktuoti kaip abstrakčią „juodąją dėžę“. GA nukreipia paiešką į perspektyvias paieškos erdves G (Srinivas and Patnaik 1994), todėl šie algoritmai yra pranašesni už atsitiktinės paieškos metodus.

Egzistuoja ir tam tikrų GA trūkumų. (Sanchez-Caballero *et al.* 2012) išskyrė tokius du trūkumus: komplikuotas optimizavimo parametrų parinkimas ir stipri nagrinėjamos problemos priklausomybė nuo optimizavimo parametrų bei jų parinkimo. (Tazawa *et al.* 1996) akcentuoja tokius GA trūkumus: per ankstyvą konvergavimą bei nepakankamą pajėgumą įvykdyti lokalią paiešką. Taip pat negalima visada garantuoti, kad problemos sprendimas naudojant GA bus greitas, o gautas sprendinys bus globalus. Kartais GA gali naudoti didelius kompiuterio resursus bei veikti lėčiau negu kiti metodai (Kennedy 1993). Todėl natūralu, kad mokslininkai bei inžinieriai bando sukurti tokius algoritmus, kurie būtų pagrįsti genetinių algoritmų koncepcija, tačiau eliminuotų minėtus trūkumus (Adeli and Nai-Tsang 1993; Adeli and Nai-Tsang 1994). Vienas iš naudojamų būdų – tai GA apjungimas su kitais optimizavimo algoritmais, kai gaunamas hibridinis GA, kuris pasižymi tuo, kad per žymiai trumpesnį laiką galima gauti geresnių rezultatų. (Kim *et al.* 2009) sukūrė tokį hibridinį GA, kuris apjungia realiųjų skaičių GA su NSSGA (*angl.* Non-dominated Sorting and Sharing Genetic Algorithm), tokiu būdu žymiai sutrumpino sijynų daugiakriterio optimizavimo laiką. Sijynų optimizavimo laikas buvo sutrumpintas, kai (Šešok and Belevičius 2009) pasiūlė hibridinį GA, kuris sukonstruotas iš GA ir lokaliosios stochastinės paieškos strategijos.

Atliekant sijynų optimizavimą, galima pabandyti atlikti optimizavimą vienu metu taikant keletą optimizavimo krypčių: matmenų ir topologijos, formos ir topologijos arba formos, matmenų ir topologijos. Tokiu būdu padidėja tiek paieškos erdvė, tiek galimybės surasti geresnį sprendinį, t.y. geresnę sijyno konstrukciją. Todėl vertinant globalaus optimizavimo aspektu – tai labai perspektyvi strategija ieškant optimalaus konstrukcijos varianto. Akivaizdu, kad toks sinchroniškas keleto optimizavimo tipų įgyvendinimas inžineriniu aspektu yra gana sudėtingas, kadangi žymiai išauga projektavimo kintamųjų skaičius. Sinchroninio optimizavimo problemos vis dar mažai analizuotos. (Kim *et al.* 2009) sinchroniškai optimizuoja tokius sijynų parametrus kaip tūris ir kaina. (Zhou and Li 2012) analizuoja sijynus trikampaiais baigtiniais elementais ir optimizuoja sijynų svorį įvertindamas vienu metu tris faktorius: lenkimo momentus, poslinkius ir medžiagos tūrį.

Atliktas įvairių optimizavimo algoritmų palyginimas įrodo, kad šiuo metu GA yra vienas efektyviausių ir labiausiai tinkamų metodų įvairių inžinerinių struktūrų optimizavimui (Belevičius *et al.* 2011; Sanchez-Caballero *et al.* 2012).

Genetinis algoritmas pasižymi tuo, kad jis gana dažnai per anksti konverguoja į lokalų sprendinį, t. y. po tam tikro populiacijų skaičiaus konvergavimo greitis tampa nuliniu, todėl yra pasiūlytos kelios išeitys: 1) genetinio algoritmo adaptyvumo pritaikymas, kai konverguojama į lokalų sprendinį. Tokiu atveju iš lokalaus sprendinio aplinkos ištrūkti galima šuoliškai keičiant genetinius GA parametrus. Tokia strategija darbe vadinama adaptyviu genetiniu algoritmu (AGA); 2) genetinio algoritmo su paskirstymo strategija (GAPS) pritaikymas. Šia strategija siekiama padidinti populiacijos (sprendinių) įvairovę ir taip pagerinti optimizavimo rezultatus.

Genetinio algoritmo adaptyvumo galimybės nagrinėtos (De Jong 1975; De Jong 1980; Schaffer and Morishima 1987; Shaefer 1987; Davis 1989; Saravanan *et al.* 1995; Deb and Goyal 1996; Hinterding *et al.* 1996; Deb and Beyer 2001; Luchian and Gheorghies 2003). De Jong pasiūlė GA su populiacijų adaptyvumu, o (Srinivas and Patnaik 1994) – GA su adaptyviomis kryžminimo ir mutacijos tikimybėmis, siekiant išlaikyti populiacijos įvairovę bei GA konvergavimo greitį. (Lobo and Lima 2007) apžvelgė populiacijos dydžio adaptyvumo schemas genetiniuose algoritmuose bei pasiūlė rekomendacijas kuriant populiacijos adaptyvumo mechanizmus genetiniams ir evoliuciniams algoritmams.

Genetinio algoritmo su paskirstymo strategija pritaikomumas analizuotas daugelio mokslininkų, iš kurių paminėtini šie: (Hacioglu and Ozkol 2005) pasiūlė paskirstymo strategijas, pagrįstas genetinių operacijų išskaidymu bei individų kodavimu realiųjų skaičių eilute; (Power *et al.* 2005) pasiūlė genetinį algoritmą su paskirstymo strategija, kai taikomas specialus atrankos mechanizmas, užtikrinantis individų įvairovę populiacijose; (Yao *et al.* 2007) analizuoja kaip pritaikyti migracijos strategiją paskirstymo genetiniam algoritmui; (Li and Huang 2012) sukūrė paskirstymo strategiją sujungdamas lygiagretųjį genetinį algoritmą su kryptingai adaptyvia migracijos strategija.

Lietuvoje taip pat yra atlikta daug mokslinių darbų optimizavimo srityje, pvz., (Čyras 1983) pasiūlė matematinius modelius tampriai-plastinių plokštelių analizei ir optimizavimui; (Atkočiūnas and Čižas 2002) išanalizavo A. Čyro darbus, kuriuose taikomi matematinio programavimo metodai įvairių struktūrų optimizavimui, formuluojami mechaninių struktūrų matematiniai modeliai ir pasiūlyti optimizavimo algoritmai bei gauti rezultatai; (Belevičius *et al.* 2002) analizavo sijynų optimizavimą ir pasiūlė naudoti genetinius algoritmus tokių konstrukcijų optimizavimui; (Rusakevičius and Belevičius 2002) pasiūlė algoritmus laminuotų lenkiamų plokštelių optimizavimui; (Atkočiūnas *et al.* 2007) pasiūlė kaip pritaikyti netiesinį programavimą optimalių rėmų konstrukcijų projektavimui; (Šešok and Belevičius 2009) pasiūlė hibridinį genetinį algoritmą

sijynų optimizavimui; (Belevičius *et al.* 2011) atliko optimalaus polių išdėstymo sijynuose tyrimus, palygindami įvairių algoritmų rezultatus.

1.4. Pirmojo skyriaus išvados ir disertacijos uždavinių formulavimas

Mokslinėje literatūroje nėra skirta pakankamai dėmesio sinchroniniam sijynų topologijos ir matmenų optimizavimui. Svarbiausios išvados:

1. Globalaus daugiakriterio sijynų topologijos ir matmenų optimizavimo inžinerinei praktikai aktualios apimties uždaviniai iki šiol dar nėra galutinai išspręsti.
2. Sijynų optimizavimui labiau tinka stochastiniai (tikimybiniai) globalaus daugiakriterio optimizavimo algoritmai, kadangi minėtų uždavinių sprendimui reikalingi dideli kompiuteriniai ištekliai, viršijantys asmeninių kompiuterių pajėgumą.
3. Daugelio tyrėjų patirtis rodo, kad GA gali būti naudojami atliekant sijynų topologijos bei matmenų optimizavimą.

Remiantis aukščiau pateiktomis išvadomis formuluojami pagrindiniai darbo uždaviniai:

1. Pritaikius GA koncepciją, sukurti ir išbandyti skaičiavimo technologiją, kuri leistų optimizuoti atskirai sijynų topologiją ir sijynų sijų skerspjūvio matmenis.
2. Pritaikius GA koncepciją, sukurti ir išbandyti skaičiavimo technologiją sinchroniniam sijynų topologijos bei matmenų optimizavimui.
3. Pritaikius adaptyvaus genetinio algoritmo (AGA) koncepciją, sukurti ir išbandyti skaičiavimo technologiją sinchroniniam sijynų topologijos bei matmenų optimizavimui.
4. Pritaikius GA su paskirstymo strategija (GAPS) koncepciją, sukurti ir išbandyti skaičiavimo technologiją sinchroniniam sijynų topologijos bei matmenų optimizavimui.

2

Genetinis algoritmas sijynų topologijos optimizavimui

Šiame skyriuje yra aprašyta, kaip galima pritaikyti genetinių algoritmų koncepciją sijynų topologijai optimizuoti. Optimizavimas yra neatsiejama inžinerinės praktikos dalis. Šioje disertacijoje nagrinėjama viena specifinė statybos inžinerijos struktūrų optimizavimo uždavinių klasė – polių padėčių optimizavimas sijyno tipo pamatuose. Pirmoje skyriaus dalyje yra pristatoma skaičiavimo technologija ir programinė įranga, kurių pagalba baigtinių elementų metodu (BEM) galima gauti visas optimizavimui būtinas sijynų tipo konstrukcijos charakteristikas: sijų įrašas, mazgų poslinkius, lenkimo momentus sijose. Antroje skyriaus dalyje yra pademonstruota, kaip užkoduoti sijyną realiųjų skaičių seka bei kaip iš realiųjų skaičių sekos gauti ją atitinkantį sijyną. Trečioje skyriaus dalyje pateikiami uždavinio sprendimo genetiniu algoritmu (GA) rezultatai. Šio skyriaus medžiaga paskelbta šiose autoriaus publikacijose (disertacijos tema): Mačiūnas *et al.* 2011, Belevičius *et al.* 2011.

2.1. Sijynų optimizavimas

Sijynai, kuriuos sudaro poliai ir jungiančiosios sijos, yra populiariausios ir efektyviausios pamatų schemos, ypač kai statybos vyksta ant silpnų gruntų. Poliai yra pamatiniai statinio elementai, kurie per jungiančiąsias sijyno sijas perima viso statinio apkrovas. Jungiančiosios sijos gaminamos statybvietėje, o poliai paprastai yra pagaminti gamykloje, todėl jų matmenys yra apibrėžti projektavimo etape. Duotajai sijyno geometrinei schemai, apkrovai ir polių charakteristikoms tinkamiausias yra toks sijynas, kuris yra paremtas mažiausiu įmanomu polių skaičiumi ir kurio jungiančiosios sijos yra mažiausio skerspjūvio. Apytikriai galima laikyti, kad pigiausios sijos bus tokios, kuriose kyla mažiausi (absoliutiniu dydžiu) lenkimo momentai. Matematinio požiūriu tokio sijyno projektavimo uždavinys yra globalaus optimizavimo uždavinys, apjungiantis du skirtingus optimizavimo uždavinius: mažiausio polių skaičiaus ir mažiausio sijų tūrio paieška. Kadangi kiekvieno poliaus laikančioji galia yra žinoma, pirmąjį optimizavimo uždavinį galima apibrėžti tokiu būdu – didžiausios reakcijos jėgos poliuose minimizavimas duotajam polių komplektui. Analogiškai, antroji problema atitinka didžiausio lenkimo momento (absoliutiniu dydžiu) jungiančiosiose sijose minimizavimą. Kadangi lenkimo momentai priklauso taip pat nuo sijų standumo, todėl sijų skerspjūvio matmenys turi būti nustatomi vienu metu. Abu uždaviniai gali būti sujungiami į vieną uždavinį taikant kompromisinę tikslo funkciją.

Kaip ir visi globalaus optimizavimo uždaviniai, šiai problemai spręsti reikia didžiulių skaičiavimo išteklių (Čiegis *et al.* 2006), o deterministiniais globalaus optimizavimo algoritmais skaičiuojant net ir lygiagrečiaisiais kompiuteriais, idealaus sprendinio nepavyksta gauti ir nesudėtingam 10 polių sijynui. Inžinerinei praktikai parankesni stochastiniai optimizavimo algoritmai, negarantuojantys globalaus sprendinio, tačiau surandantys racionalų sprendinį taip pat ir gerokai didesnės apimties uždaviniams (Belevičius *et al.* 2011). Kelių stochastinių algoritmų (Atsitiktinė paieška, Modifikuota atsitiktinė paieška, Atkaitinimo modeliavimas, Genetinis algoritmas, Simplex metodas, Kvazi-Niutono metodas, NEWUOA metodas) optimizavimo uždavinio sprendimo rezultatų palyginimas rodo, kad pranašiausi yra Genetinis algoritmas (Goldberd 1989) ir Atkaitinimo modeliavimo algoritmas (Groenwold and Hindley 2002). Uždavinio sprendimą ypač paspartina toks tipinių optimizavimo algoritmų modifikavimas, kai uždavinys formuluojamas įtraukiant visą iš anksto žinomą informaciją apie uždavinį.

2.2. Sijynų charakteristikų nustatymas

Prieš atliekant mechaninės struktūros optimizavimą, būtina gebėti nustatyti tos struktūros charakteristikas, apskaičiuoti tikslo funkcijos reikšmę bei patikrinti optimizavimo uždavinio kraštines sąlygas (apribojimus). Darbe yra naudojamas vienas populiariausių ir žinomiausių metodų – tai baigtinių elementų metodas (Spyrakos and Raftoyiannis 1997; Barauskas *et al.* 2004; Zienkiewicz and Taylor 2005; Rao 2010), kurio pagalba yra nustatomi sijynų konstrukcijos sijų įrašos, įtempimai, mazgų poslinkiai, lenkimo momentai sijose ir kt. Egzistuoja nemažai universalių komercinių programinių paketų (KPP) – ABAQUS, ALGOR, ANSYS, COSMOS, GENESIS ir kt. – kuriuose yra realizuotas BEM. Vienas pagrindinių šių programų privalumų – tai galimybė inžinieriui nustatyti konstrukcijos charakteristikas su mažiausiais programavimo įgūdžiais. Pakanka įsisavinti naudojamo programinio paketo programavimo kalbos sintaksę bei pritaikyti pagrindinius pakete naudojamus uždavinio apibrėžimo ir rezultatų gavimo principus. Gautų rezultatų patikimumas – tai dar vienas KPP privalumas, nes tyrėjui nebereikia verifikuoti gautų rezultatų, kas akivaizdžiai neišvengiama naudojant pačių tyrėjų sukurtas programas. Nepaisant to, vienas pagrindinių universalių KPP trūkumų – optimizavimą atliekančios programos apjungimas su baigtinių elementų metodą realizuojančia programa, nes paprastai jos būna parašytos skirtingose programinėse aplinkose. Todėl jas susiejant viena su kita būtini papildomi programavimo ir suderinimo veiksmai, be to optimizavimo proceso laikas pailgėja (Šešok 2007; Šešok 2008). Vieno KPP programos iškvietimas iš kito KPP valdančiosios programos gali užtrukti iki 0,5 sek. (Šešok 2008), o tai yra nepriimtina, ypač sprendžiant didelės apimties optimizavimo uždavinius, kadangi optimizuojant daug kartų tenka atlikti šias iškvietimo operacijas. Dėl to optimizavimo uždavinio sprendimas, priklausomai nuo uždavinio apimties, gali užtrukti iki keliasdešimt valandų ar net daugiau (Šešok ir Belevičius 2009).

Kadangi inžinieriams (projektuotojams) optimizavimo laikas yra ypač aktualus, tad autorius išsikėlė tikslą sukurti metodiką, kuria naudojantis būtų galima spręsti didelės apimties struktūros optimizavimo uždavinius per priimtina inžinieriams laiką. Todėl KPP naudojimo buvo atsisakyta ir FORTRAN programavimo kalbos pagalba buvo sukurta programa, realizuojanti baigtinių elementų metodą sijynams. Tam buvo panaudota VGTU Teorinės mechanikos katedros darbuotojų sukurta programa (skirta sijynų analizei), kuri autoriaus buvo modifikuota ir pritaikyta nagrinėjamam uždaviniui spręsti. Ši programa gali nustatyti mechanines sijynų charakteristikas: įrašas (reakcijas poliuose, lenkimo momentus sijose) ir kt. Ši programa buvo patikrinta ANSYS programiniu paketu, t.y. buvo sprendžiami testiniai uždaviniai, kurių rezultatai – gauti tiek naudojant ANSYS programinį paketą, tiek autoriaus sukurta programą – buvo palyginti.

Programos patikrinimui buvo panaudota programų sistemų inžinerijoje žinoma metodika, skirta sukurti programinio paketo kokybės tyrimams (Čaplinskas 1996). Buvo atliekami tokie testai:

1. Programinio paketo (PP) našumo įvertinimo testai.
2. PP normalaus veikimo patikrinimo testai.
3. Ribinių situacijų patikrinimo testai.
4. Klaidingų duomenų apdorojimo patikrinimo testai.
5. Specialių paskirčių testai.

Testų rezultatai parodė, kad ANSYS programiniu paketu gauti sprendiniai sutapo su autoriaus sukurtu programinio paketo pagalba gautais sprendimais.

Programos veikimo algoritmas sudarytas iš tokių dalių:

1. Sijyno modelį aprašančių duomenų perdavimas (sijyno duomenys yra užkoduoti realiųjų skaičių eilute).
2. Duomenų apie sijyno modelį dekodavimas pagal iš anksto nustatytą algoritmą (t.y. sijyno, atitinkančio realiųjų skaičių eilutę, gavimas).
3. Uždavinio sprendimas baigtinių elementų metodu.
4. Sprendimo rezultatai: reikalingas polių skaičius, atraminės reakcijos poliuose ir lenkimo momentai jungiančiose sijose.

Toliau išsamiai yra aptartas kiekvienas algoritmo žingsnis:

Sijyno modelį aprašančių duomenų perdavimas: sijyno duomenys, kurie užkoduoti realiųjų skaičių seka, yra perduodami programai. Optimizavimo genetiniu algoritmu programa taip pat naudos realiųjų skaičių seką, todėl šios abi programos bus tarpusavyje lengvai suderinamos.

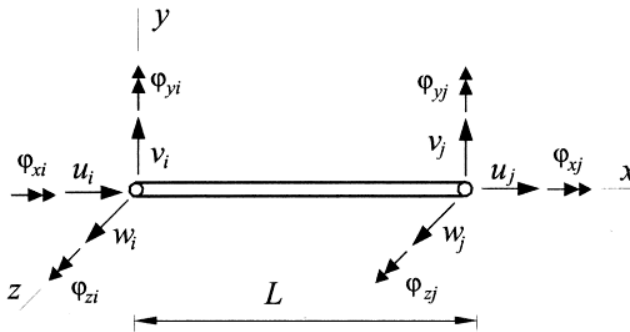
Duomenų apie sijyno modelį dekodavimas pagal iš anksto nustatytą modelį: programa pagal atitinkamą suprogramuotą sijyno modelį atkoduoja realiųjų skaičių seką, t.y. nustato programai pateikto sijyno sijų išsidėstymo schemą. Modelyje taip pat yra saugomi duomenys apie išorines apkrovas, įtvirtinimus ir mazgų koordinatas. Kiekvienam atskiram optimizavimo uždaviniui sijyno modelis sukuriamas vieną kartą. Norėdami išspręsti kitą sijyno optimizavimo uždavinį, būtina atitinkamai pakeisti turimą arba suprogramuoti naują modelį. Šis procesas gali užtrukti iki 20–30 min. 2.5 poskyryje pateiktas vienas iš sijynų kodavimo realiųjų skaičių eilute metodų.

Uždavinio sprendimas baigtinių elementų metodu: tiesioginis sijyno analizės uždavinys yra sprendžiamas baigtinių elementų metodu: poliai idealizuojami spyruoklinėmis atramomis (t. y. baigtinio elemento tinklelio mazgais su duotomis tampriosiomis kraštinėmis sąlygomis), o polius jungiančios sijos – lenkiamo

strypo baigtiniais elementais. Tarp dviejų atramų naudojamas vienas strypo elementas. Vienu elementu idealizuojama ir gembinė (jei tokių yra) sijyno sija. *A priori* laikoma, kad polių laikančioji galia yra žinoma prieš optimizavimą. Kadangi optimizavimo trukmė labiausiai priklauso nuo tiesioginio sijyno analizės uždavinio sprendimo laiko, todėl buvo sukurta (naudojant FORTRAN programavimo kalbą) greita uždaviniui pritaikyta originali programa.

Lenkiamųjų strypų elementai turi du mazgus su šešiais laisvės laipsniais (Spyrakos and Raftoyianis 1997) kiekviename mazge (2.1 pav.). Šio elemento standumo matrica yra tokia (Spyrakos and Raftoyianis 1997):

$$[K] = \begin{bmatrix} [K_{11}] & [K_{12}] \\ [K_{12}]^T & [K_{22}] \end{bmatrix}, \quad (2.1)$$



2.1 pav. Lenkiamo strypo baigtinio elemento laisvės laipsniai
Fig. 2.1. Degrees of freedom of bending beam finite element

2.1 paveiksle pažymėti tokie dydžiai:

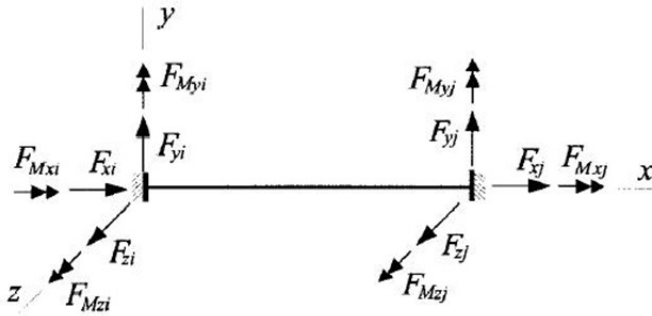
- $u_i, u_j, v_i, v_j, w_i, w_j$ – atitinkamai i -tojo mazgo ir j -tojo mazgo poslinkiai atitinkamai x, y, z koordinačių ašių kryptimis;
- $\varphi_{xi}, \varphi_{xj}, \varphi_{yi}, \varphi_{yj}, \varphi_{zi}, \varphi_{zj}$ – atitinkamai i -tojo mazgo ir j -tojo mazgo posūkiiai apie atitinkamai x, y, z koordinačių ašis;
- L – ilgis.

$$[K_{11}] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{12EI_z}{L^3} & & & & \frac{6EI_z}{L^2} \\ 0 & & \frac{12EI_y}{L^3} & & -\frac{6EI_y}{L^2} & \\ 0 & & & \frac{GJ}{L} & & \\ 0 & & -\frac{6EI_y}{L^2} & & \frac{4EI_y}{L} & \\ 0 & \frac{6EI_z}{L^2} & & & & \frac{4EI_z}{L} \end{bmatrix}, \quad (2.2)$$

$$[K_{12}] = \begin{bmatrix} -\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12EI_z}{L^3} & & & & \frac{6EI_z}{L^2} \\ 0 & & -\frac{12EI_y}{L^3} & & -\frac{6EI_y}{L^2} & \\ 0 & & & -\frac{GJ}{L} & & \\ 0 & & \frac{6EI_y}{L^2} & & \frac{2EI_y}{L} & \\ 0 & -\frac{6EI_z}{L^2} & & & & \frac{2EI_z}{L} \end{bmatrix}, \quad (2.3)$$

čia E – sijos medžiagos tamprumo modulis; A – sijos (t. y. lenkiamo strypo baigtinio elemento) skerspjūvio plotas; J, I_z, I_y – sijos skerspjūvio inercijos momentai; submatrica $[K_{22}]$ sutampa su $[K_{11}]$, bet nediagonalieji nariai yra priešingu ženklų; $[K_{12}]^T$ – transponuota submatrica.

Visos aktyviosios jėgos veikiančios sijyne yra sutelkiamos į lenkiamo strypo baigtinio elemento mazgines jėgas (Spyrakos and Raftoyianis 1997) (2.2 pav.): $F_{xi}, F_{xj}, F_{yi}, F_{yj}, F_{zi}, F_{zj}$ – atitinkamai i -tajame mazge ir j -tajame mazge veikiančios jėgos atitinkamai x, y, z koordinačių ašių kryptimis; $F_{Mxi}, F_{Msj}, F_{Myi}, F_{Myj}, F_{Mzi}, F_{Mzj}$ – atitinkamai i -tajame mazge ir j -tajame mazge veikiančios momentai apie atitinkamai x, y, z koordinačių ašis.



2.2 pav. Lenkiamo strypo baigtinio elemento mazginės jėgos
Fig. 2.2. Nodal forces of bending beam finite element

Pagrindinė statikos lygtis yra ši:

$$[K]^a \{u\}^a = \{F_a\}^a, \quad (2.4)$$

čia indeksas a žymi baigtinių elementų ansamblį (formulėse toliau neberodomas), $\{u\}$ – mazgų poslinkiai, $\{F_a\}$ – aktyviosios jėgos.

Sprendimo rezultatai: išsprendus lygtį (2.4) nustatomi nagrinėjamo sijyno mazgų poslinkiai. Tuomet galima apskaičiuoti reakcijos jėgas poliuose (R_i) bei lenkimo momentus sijose (M) atitinkamai pagal (2.5) ir (2.6) formules.

$$R_i = \sum_j [K_{ij}] \cdot u_j, \quad (2.5)$$

$$M = EI_z \kappa, \quad (2.6)$$

$$\kappa = -\frac{d^2}{dx_i^2} \left(\sum N_i u_i \right), \quad (2.7)$$

čia κ – strypo kreivis; N_i – Ermito antrosios eilės interpoliacinės funkcijos (Barauskas *et al.* 2004) (2.8), (2.9), (2.10), (2.11); u_i, u_j – mazgo i -tojo ir j -tojo laisvės laipsnio poslinkis. Kadangi lygtyje (2.7) figūruoja lokaliaji strypo elemento koordinatė, todėl lenkimo momentus galima suskaičiuoti bet kuriame norimeame strypo taške.

$$N_1 = 1 - \frac{3x_i^2}{L^2} + \frac{2x_i^3}{L^3}, \quad (2.8)$$

$$N_2 = x_l - \frac{2x_l^2}{L} + \frac{x_l^3}{L^2}, \quad (2.9)$$

$$N_3 = \frac{3x_l^2}{L^2} - \frac{2x_l^3}{L^3}, \quad (2.10)$$

$$N_4 = -\frac{x_l^2}{L} + \frac{x_l^3}{L^2}, \quad (2.11)$$

čia x_l – lokalioji koordinatė išilgai strypo ašies.

BEM programos sprendinys nustato sistemos globaliąją pusiausvyrą. Taip pat yra tikrinama, ar sistema nėra mechanizmas, t. y. yra tikrinama, ar sistemos globaliosios standumo matricos $[K]$ determinantas nėra lygus nuliui. Jeigu determinantas lygus nuliui, tai sistema yra neišsprendžiama.

2.3. Sijynų optimizavimo problemos formuluotė

Nagrinėjamos problemos formuluotė – polių po jungiančiosiomis sijomis optimalios išdėstymo schemas radimas, siekiant mažiausių atraminių reakcijų poliuose (duotam polių skaičiui) ir mažiausių absoliutinių dydžiu lenkimo momentų jungiančiosiose sijyno sijose. Idealus sijynas yra toks, kuriame reakcijos jėgos poliuose yra tolygiai pasiskirsčiusios bei kuriame lenkimo momentai sijose taip pat yra tolygiai pasiskirstę. Mokslininkų patirtis atskirai minimizuojant reakcijas bei lenkimo momentus sijynuose (Belevičius and Valentinavičius 2000; Belevičius *et al.* 2002) rodo, kad praktiniuose sijynų optimizavimo uždaviniuose tikslo funkcijos pasižymi tuo, jog turi daug lokalių minimumo taškų. Kitas komplikuotas tokių uždavinių bruožas yra tai, kad paprastai tikslo funkcijos yra labai jautrios polių padėčių pokyčiams (Belevičius *et al.* 2011): kartais netgi nedidelis vieno poliaus padėties pasikeitimas lemia didelius tikslo funkcijos reikšmės pokyčius. Dėl visų minėtų aplinkybių polių išdėstymo užduotis praktiniuose sijynuose tampa ypatingai sudėtingu globalaus optimizavimo uždaviniu. Kadangi polių išdėstymo uždavinyje iš anksto žinomas globalus sprendinys – tereikia visą sijyną veikiančią apkrovą padalinti iš poliaus laikančiosios galios, tad toks uždavinys itin parankus įvertinti, ar siūloma technologija leidžia gauti gerą sprendinį. Dėl techninių priežasčių sijyno kampuose paprastai visuomet įrengiami poliai, kurie niekada nekeičia savo padėties ir todėl negali dalyvauti optimizavimo procese. Šiame darbe tokių polių (nejudamų atramų) nėra, kadangi jų buvimas matematiškai, matyt, neleistų rasti globalaus sprendinio. Šiame darbe jokie suvar-

žymai polių padėtims nėra nustatomi, išskyrus kelis technologinius apribojimus (žr. 2.4 poskyrį).

Inžinerijoje globalaus optimizavimo uždaviniuose paprastai susiduriama su keliais ir dažnai konkuruojančiais tarpusavyje tikslais (Marcelin 2011; Mohammadzadeh and Etemadee 2011). Paprastai tokių uždavinių sprendinys yra ne unikali optimali tikslo funkcijos reikšmė, o kompromisinių sprendinių – vadinamų Pareto-optimaliais sprendiniais – rinkinys. Kiekvienas Pareto sprendinys yra optimalus ta prasme, kad tikslo funkcijos komponento reikšmė negali būti „pagerinta“, „nepabloginus“ bent vieno iš likusių tikslo funkcijos komponentų reikšmės. Todėl svarbiausias daugiakriterio optimizavimo tikslas yra keletas Pareto-optimalių sprendinių radimas, kad būtų parodyta konkuruojančių tikslų kompromiso suderinamumo informacija.

Kaip minėta (1.1.3 poskyryje) paprasčiausias daugiakriterio optimizavimo metodas yra svartinė kriterijų suma. Šis metodas yra tinkamas tokiems optimizavimo uždaviniams spręsti, kai galimų sprendinių erdvės projekcija į tikslo erdvę yra išgaubtos formos.

Atsižvelgiant į optimizavimo uždavinį su dviem kriterijais (tai yra ypač dažnas daugelio inžinerinių optimizavimo uždavinių atvejis) $f_1(x)$ ir $f_2(x)$ su duotais svoriniais koeficientais atitinkamai w_1 bei w_2 , daugiakriteris uždavinys gali būti konvertuotas į tokį vienakriterį optimizavimo uždavinį:

$$f = \min_{x \in D} f(x) = w_1 \cdot f_1(x) + w_2 \cdot f_2(x), \quad (2.12)$$

čia $f(x)$ – tolydžių kintamųjų $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$ netiesinė tikslo funkcija; n – projektavimo kintamųjų x (nusakančių polių padėtis ir jungiamųjų sijų skerspjūvio matmenis) skaičius; $D \subset \mathfrak{R}^n$ – projektavimo kintamųjų galima erdvė. Kaip rodo praktika, uždavinys – daugiaekstremis ir labai jautrus projektavimo kintamųjų pokyčiams, be to gali egzistuoti daug lokalių optimumų. Tariant, kad D yra išgaubtos formos, lygties (2.12) sprendinys yra optimalus duotoms svorių w_1 ir w_2 reikšmėms. Prie skirtingų w_1 ir w_2 reikšmių gali būti sugeneruota daug skirtingų optimalių sprendinių. Geriausias kompromisinis sprendinys gali būti atrinktas atsižvelgiant į kompromisinius sumetimus.

Šiame darbe pirmasis tikslo funkcijos (2.12) narys – tai didžiausia reakcijos jėga kylanti poliuje:

$$f_1(x) = \max_{i=1, \dots, N_a} R_i(x), \quad (2.13)$$

čia N_a – polių skaičius; $R_i(x)$ – reakcijos jėga i -tajame poliuje. Antrasis tikslo funkcijos (2.12) narys – tai didžiausias lenkimo momentas jungiančiojoje sijoje:

$$f_2(x) = \max_{i=1, \dots, m \cdot N_b} |M_i(x)|, \quad (2.14)$$

čia N_b – sijų skaičius; m – vienos sijos atkarpų skaičius, kuriose yra skaičiuojami lenkimo momentai $M_i(x)$. Paprastai pakanka trijų taškų (Mačiūnas *et al.* 2011), kad išilgai sijos būtų surastas lenkimo momentas, kurio vertė artima didžiausiai reikšmei.

Optimizavimo uždavinio ribojimų sistema:

$$R_{\max} \leq R_{leist}, \quad (2.15)$$

$$M_{\max} \leq M_{leist}, \quad (2.16)$$

$$(h_{sk} / b_{sk})_{\max} \leq (h_{sk} / b_{sk})_{leist}, \quad (2.17)$$

čia R_{\max}, R_{leist} – atitinkamai didžiausia ir leistina reakcijos jėga poliuje; M_{\max}, M_{leist} – atitinkamai didžiausias ir leistinas lenkimo momentas sijoje; $(h_{sk} / b_{sk})_{\max}, (h_{sk} / b_{sk})_{leist}$ – atitinkamai didžiausias ir leistinas sijos skerspjuvio matmenų santykis; h_{sk}, b_{sk} – atitinkamai sijos skerspjuvio aukštis ir plotis, čia $h_{sk} \in [h_{\min}; h_{\max}]$, $b_{sk} \in [b_{\min}; b_{\max}]$.

Projektavimo kintamųjų rinkinį sudaro polių padėtys x_i , $i = 1, \dots, N_a$ bei sijų skerspjuvio aukštis h_{sk} ir plotis b_{sk} , kurie visoms sijyno sijoms yra laikomi vienodais.

Pradiniai duomenys sijynų optimizavimo uždaviniui yra šie:

1. Apkrovų duomenys. Aktyviosios jėgos gali būti pateiktos sutelktųjų apkrovų ir lenkimo momentų bet kuriame sijos taške pavidalu arba paskirstytųjų trapecinių apkrovų bet kurioje sijos dalyje pavidalu.
2. Didžiausia leistina reakcijos jėga, galinti kilti bet kuriame iš polių.
3. Geometrinė jungiančiųjų sijų schema.
4. Mažiausias galimas atstumas tarp gretimų polių.
5. Nejudamų atramų (jei tokių yra) pozicijos.
6. Sijų medžiagos duomenys (vienos sijos medžiaga laikoma izotropine).
7. Sijų skerspjuvių duomenys (plotas, inercijos momentai).
8. Polių skaičius.
9. Vertikalus poliaus standis ir du lenkimo standžiai: išilgai sijos (po kuria yra polius) ir statmena kryptimi.

Sprendžiant sijynų topologijos ir matmenų optimizavimo uždavinį nustatomas toks duotų polių išdėstymas sijyne, kad reakcijos jėgos poliuose neviršytų laikančiosios polių galios, o lenkimo momentai sijose absoliutiniu dydžiu būtų galimai mažiausios reikšmės. Jei tokia polių išdėstymo schema negalima, tuomet polių skaičius turi būti padidintas.

2.4. Optimizavimo uždavinio apribojimai

Atraminiai poliai gali būti išsidėstę tik po jungiančiosiomis sijomis, todėl aki-vaizdus apribojimas polių padėtimis: optimizavimo metu poliai gali judėti tikslai išilgai jungiančiųjų sijų.

Visas optimizavimo uždavinio sprendimas yra sudarytas iš dviejų etapų. Pirmajame etape dvimatis sijynas, su konkrečiomis polių padėtimis bei sijynų skerspjuvio matmenimis, yra matematiškai transformuojamas į vienmatę struktūrą (2.3 pav.), kurioje poliai gali laisvai judėti ir keisti savo padėtį, o sijų skerspjuvio matmenys gali kisti. Antrajame etape atgalinė transformacija iš vienmatės struktūros atstato naujas polių padėtis bei naujus sijų skerspjuvio matmenis į dvimatį sijyną. Kraštinės sąlygos polių padėtimis:

$$0 \leq x_i \leq L, \quad i = 1, \dots, N_a, \quad (2.18)$$

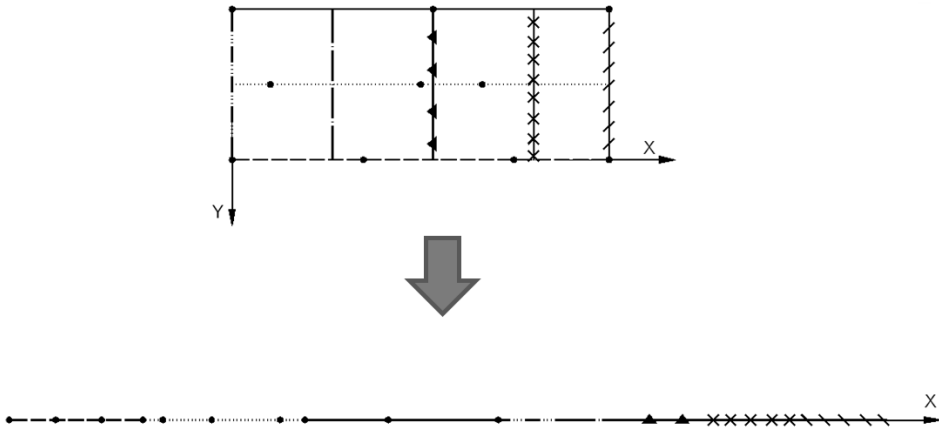
čia x_i – projektavimo kintamasis apibūdinantis i -tojo poliaus padėtį; L – bendras visų sijų ilgis sijyne.

Jei yra apibrėžtas mažiausias galimas atstumas δ tarp gretimų polių, tuomet papildomi apribojimai:

$$\|x_i - x_j\| \geq \delta, \quad i \neq j, \quad (2.19)$$

čia x – vienmatės polių koordinatės; $\|x_i - x_j\|$ apibrėžia atstumą tarp polių. Jei kuris nors sijynas netenkina šio ribojimo (2.19), tikslo funkcijai (2.12) taikoma iš anksto nurodyta bauda.

Stiprumo ribojimai (2.15, 2.16) yra patikrinami visiems poliams ir visoms sijoms. M_{leist} (2.16) lygtyje priklauso nuo leistinųjų įtempimų betone ir armatūroje, armatūros skerspjuvio ploto bei sijos skerspjuvio matmenų. Minėtas (2.16) apribojimas automatiškai mažina sijos skerspjuvio plotį bei didina sijos skerspjuvio aukštį, todėl papildomas apribojimas (2.17) sijos skerspjuvio matmenims turi būti įtrauktas į bendrą kraštinių sąlygų sąrašą.



2.3 pav. Dvimatis sijynas transformuojamas į vienmatę struktūrą
Fig. 2.3. Two-dimensional grillage is transformed to a one-dimensional structure

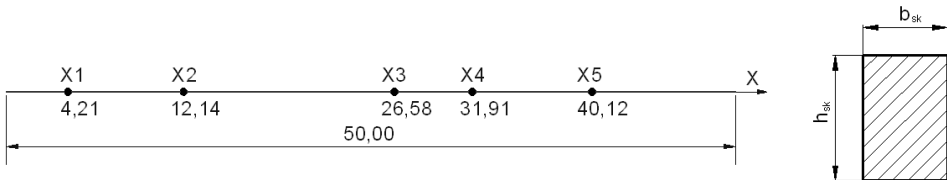
Dėl technologinių priežasčių sijyno kampuose paprastai visuomet įrengiami poliai. Sukurtose programose numatyta galimybė įrengti vadinamąsias nejudamas atramas, negalinčias keisti savo padėties ir todėl nedalyvaujančias optimizavimo procese. Šio darbo skaitiniuose bandymuose tokių atramų nėra, kadangi jų buvimas matematiškai, matyt, neleistų gauti globalaus sprendinio. Kai kurie kiti technologiniai ribojimai, pavyzdžiui mažiausias galimas atstumas tarp gretutinių polių (dėl poliakalės darbo ypatumų), taip pat gali kliudyti surasti idealią polių išdėstymo schemą.

2.5. Sijynų kodavimas realiųjų skaičių seka

Paprastai genetiniuose algoritmuose chromosomos (individai) koduojami bitų seka, natūrinių skaičių seka arba realiųjų skaičių seka (Weise 2009). Kodavimo būdo pasirinkimas priklauso nuo sprendžiamos problemos pobūdžio. Atlikti tyrimai (Collins and Eaton 1997) rodo, kad optimizuojant funkciją kodavimas realiųjų skaičių seka yra pranašesnis už kodavimą bitų seka. Šiame darbe sijynai yra koduojami realiųjų skaičių seka dar ir todėl, kad bus optimizuojamos polių koordinatės plokštumoje (2D koordinatinių sistemoje) bei sijynų skerspjūvio matmenys. Kadangi tiek koordinatinių, tiek matmenų vertės priklauso realiųjų skaičių aibei, todėl atrodo logiškas pasirinkimas koduojant sijynus naudoti realiuosius skaičius.

Individą galima apibrėžti kaip sijyno baigtinių elementų modelį, kurį sudaro tam tikru būdu atsitiktinai parinktos polių koordinatės vienmatėje sijyno struktūroje (2.3 pav.) bei tam tikru būdu atsitiktinai parinkti sijų skerspjūvio matmenys.

Tarkime, kad turime 5 (penkių) polių vienmatį sijyną (2.4 pav.), kurio bendras visų sijų ilgis sijyne yra 50 metrų, o sijų skerspjūvio matmenys (aukštis h_{sk} ir plotis b_{sk}) gali būti parenkami iš intervalo: $h_{sk} \in [0,1;0,9]$, $b_{sk} \in [0,1;0,9]$.



2.4 pav. Vienmatis sijynas ir sijos skerspjūvis

Fig. 2.4. One-dimensional grillage structure and cross-section of a beam

Tuomet individas (sijynas) gali būti užkoduotas realiųjų skaičių seka taip:

4,21 12,14 26,58 31,91 40,12 0,35 0,20

Pirmieji penki skaičiai nusako polių koordinatės vienmatėje sijyno struktūroje, o du paskutiniai – sijyno skerspjūvio aukštį ir plotį.

2.6. Optimizavimo algoritmas

Genetinis algoritmas – tai vienas efektyviausių ir populiariausių algoritmų sprendžiant struktūros optimizavimo problemas (Belevičius *et al.* 2011; Sanchez-Caballero *et al.* 2012). GA imituoja gamtos evoliucijos dėsnius (atranką, kryžminimą, mutaciją), kuriais remiantis sprendinių (individų) populiacija evoliucionuoja gerindama tikslo funkcijos reikšmes. Šiame darbe optimizavimo uždavinys sprendžiamas taikant GA schemą, kuri sudaryta iš tokių etapų:

- Pradinės individų populiacijos generavimas;
- Atranka ir elitizmas;
- Kryžminimas;
- Mutacija.

Pradinė individų populiacija yra generuojama taikant pakeistą atsitiktinės paieškos metodą. Projektavimo kintamieji (polių koordinatės bei sijų skerspjūvio aukštis h_{sk} ir plotis b_{sk}) yra generuojami atsitiktinai su tolygiu pasiskirstymu,

tačiau papildomi apribojimai priskiriami projektavimo kintamiesiems apibrėžiantiems polių koordinatėms: skirtumas tarp dviejų polių koordinatėms turi būti didesnis nei S , kuris empiriškai gaunamas pusę sijyno sijų ilgio padalijus iš polių skaičiaus:

$$|x_i - x_j| \geq S, \quad i \neq j, \quad S = \frac{L}{2N_a}. \quad (2.20)$$

Jei šie apribojimai (2.20) yra pažeidžiami, toks individas laikomas netinkamu, todėl yra brokuojamas ir į pradinę populiaciją neįtraukiamas. Toks euristikos pakeitimas yra pagrįstas tuo, kad sijyno apkrova statiniuose paprastai yra paskirstyta visoms sijyno sijoms, todėl poliai turi būti išsidėstę po visu sijynu. Tokios informacijos įvertinimas žymiai pagerina pradinės populiacijos kokybę, paspartina GA konvergavimą bei užtikrina daug geresnius uždavinio sprendimo rezultatus (Belevičius *et al.* 2011). Tik tokiems sugeneruotiems individams, kurie yra tinkami, yra skaičiuojami lenkimo momentai sijose bei reakcijos jėgos poliuose. Individai su mažiausiomis tikslo funkcijos reikšmėmis yra laikomi geriausiais rastais sprendiniais.

Po to, kai suformuojama pradinė populiacija, N kartų generuojama nauja individų populiacija, t. y. atliekamos atrankos taikant elitizmą, kryžminimo ir mutacijos procedūras.

Algoritme atranka atliekama ruletės principu (Goldberg 1989): individai su geresne tikslo funkcijos reikšme turi didesnę tikimybę būti atrinktais į kitą populiaciją, ir atvirkščiai, individai su blogesne tikslo funkcijos reikšme turi mažesnę tikimybę patekti į kitą populiaciją. „Blogų“ individų (genų) išsaugojimas leidžia išvengti per ankstyvo konvergavimo į lokalų optimumą. Tikimybė p_j , kad j -tasis individas su tikslo funkcijos reikšme f_j bus atrinktas į kitą populiaciją, apskaičiuojama:

$$p_j = \frac{1}{f_j \sum_i \frac{1}{f_i}}, \quad (2.21)$$

čia sumos operatoriumi yra apimama visa populiacija.

Atrinkti į kitą populiaciją individai yra suporuojami ir kryžminimas gali būti atliekamas tarp šių dviejų individų su tam tikra tikimybe $p_{\text{kryž}}$. Šiame darbe naudojamas vienos pozicijos kryžminimo metodas (28–31 psl.): abu individai yra padalinami atsitiktinai sugeneruotoje pozicijoje, kuri yra sveikasis skaičius iš intervalo $[1, N_a + 1]$. Tarkime turime du atrinktus kryžminimui individus (pvz., du 5-polių sijynus):

4,21 12,14 26,58 31,91 40,12 0,35 0,20
7,56 15,45 40,82 41,65 45,03 0,47 0,31

Pavyzdžiui, sugeneruojama kryžminimo pozicija 3:

4,21 12,14 26,58 | **31,91 40,12 0,35 0,20**
7,56 15,45 40,82 | 41,65 45,03 0,47 0,31

Tuomet, po kryžminimo operacijos bus gauti tokie individai:

4,21 12,14 26,58 41,65 45,03 0,47 0,31
7,56 15,45 40,82 **31,91 40,12 0,35 0,20**

Jeigu remiantis kryžminimo tikimybe $p_kryž$ gaunama, jog kryžminimo procedūra neturi būti atliekama, tuomet individai be pakeitimų perduodami į mutacijos stadiją.

Mutacija yra vykdoma su tikimybe p_mut ir taikoma kiekvienam projektavimo kintamajam atskirai. Projektavimo kintamojo, kuris apibrėžia polių koordinatės, reikšmė yra padidinama atsitiktiniu skaičiumi tolygiai pasiskirsčiusiu intervale $[-\tau, \tau]$, čia $\tau \approx (\sum L)/100$. Tarkime, iki mutacijos etapo to paties (2.4 pav.) 50 m ilgio sijyno vieno iš poliaus koordinatė yra 12,14 ir atsitiktinai sugeneruotas atsitiktinis skaičius $\tau = 0,5$; tuomet nauja poliaus koordinatės reikšmė po mutacijos bus 12,64 (arba 11,64, jei τ su „-“ ženklu). Projektavimo kintamojo, kuris apibrėžia sijyno sijų skerspjūvio matmenis, reikšmė yra padidinama atsitiktiniu skaičiumi tolygiai pasiskirsčiusiu intervale $[-\psi, \psi]$, čia $\psi \approx h_{sk_{min}}/10$ sijos skerspjūvio aukščiui h_{sk} , arba $\psi \approx b_{sk_{min}}/10$ sijos skerspjūvio pločiui b_{sk} . Tarkime, iki mutacijos etapo sijyno sijų skerspjūvio matmenys yra tokie: $h_{sk} = 0,50$, $b_{sk} = 0,20$; be to, tarkime $h_{sk_{min}} = 0,20$, $b_{sk_{min}} = 0,10$; tuomet po mutacijos nauji sijų skerspjūvio matmenys bus: $h_{sk} = 0,52$ (arba $h_{sk} = 0,48$, jei ψ su „-“ ženklu) ir $b_{sk} = 0,21$ (arba $b_{sk} = 0,19$, jei ψ su „-“ ženklu). Visos šios (mutacijos procedūrai) naudojamos reikšmės yra parinktos remiantis skaitiniais bandymais.

Kryžminimo ir mutacijos etapuose gali būti sugeneruojami individai, kurie pažeidžia apribojimus (2.16) ir (2.19). Tokiems netinkamiems individams yra taikomos baudos, t. y. gautos tikslo funkcijos reikšmės yra dirbtinai padidinamos. Taigi, individas iš gautos populiacijos nepašalinamas, tačiau yra tik labai nedidelė tikimybė, kad jis gali būti atrinktas į kitą populiaciją.

GA darbas yra vykdomas tol, kol yra sugeneruojamas iš anksto nustatytas populiacijų skaičius. Genetiniam algoritmui paprastai yra būdingas greitas kon-

vergavimas pradiniam sprendimo etape bei lėtas nemonotoninis konvergavimas į lokalųjį (tikėtina – ir į globalųjį) sprendinį po tam tikro generacijų skaičiaus. Sprendžiant uždavinį reikia suderinti tokius GA parametrus kaip mutacijos ir kryžminimo tikimybės, generacijų skaičius. Taip pat būtina apibrėžti genetinius operatorius. GA yra stochastiniai algoritmai, todėl būtina uždavinį skaičiuoti keletą kartų. Paprastai tokie atsitiktiniai skaičiavimai, kai parenkami optimalūs genetiniai parametrai, pateikia artimus sprendinius, kurie atitinka absoliučiai skirtingas polių išdėstymo topologijas bei skirtingus sijyno sijų skerspjūvio matmenis. Inžinerinėje praktikoje tai gali būti naudinga, nes projektuotojas gali pasirinkti tinkamą sprendinį (t. y. konkrečius sijų skerspjūvio matmenis bei konkrečią polių išdėstymo schemą) iš aibės sprendinių, turinčių skirtingus sijų skerspjūvio matmenis bei skirtingas topologijas.

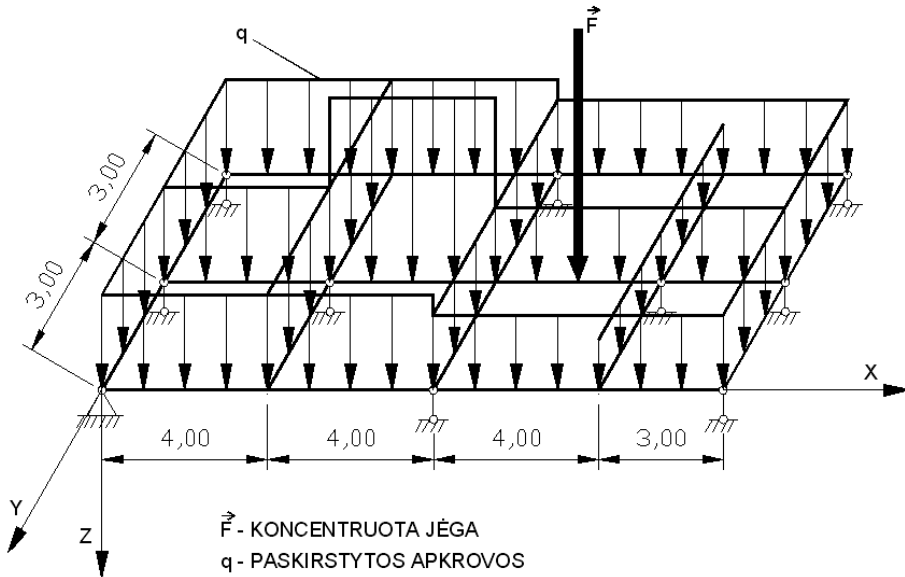
2.7. Optimizavimo uždavinio sprendimas genetiniu algoritmu

Išanalizavus GA pritaikymo galimybes aukščiau aprašytiems optimizavimo uždaviniams spręsti, buvo atlikti skaitiniai bandymai, parodantys kaip konkretūs GA parametrai lemia gaunamus rezultatus.

Pritaikius GA išspręstas optimizavimo uždavinys, kurio atraminių reakcijos jėgų poliuose minimizavimo globalūs sprendiniai jau yra gauti (Belevičius *et al.* 2011). Iš pradžių buvo išnagrinėtas nesudėtingas 10-ies polių sijynas (2.5 pav.), kurio visų sijų bendras ilgis yra 75,0 (čia ir toliau bus laikoma, kad optimizavimo uždavinio duomenų matavimo vienetai yra abstraktūs).

Sijyną veikia tam tikros išorinės apkrovos, kurios yra pateiktos koncentruotų jėgų (skaitinė vertė lygi 45,00) ir paskirstytų apkrovų (duotos tokios skaitinės vertės: 6,552; 12,960; 24,552; 30,960; 75,960) pavidalu. Globalųjį sprendinį reakcijų jėgoms poliuose galima gauti iš anksto, tereikia surasti santykį tarp sijyną veikiančių aktyviųjų jėgų sumos ir programos nustatyto polių skaičiaus. 10-ies polių (2.5 pav.) optimizavimo uždaviniui globalusis sprendinys reakcijų jėgoms poliuose yra lygus 183,7656 (Belevičius *et al.* 2011). Todėl pirmiausia buvo išspręstas 10-ies polių optimizavimo uždavinys – atraminių reakcijų jėgų (toliau – reakcijų) poliuose minimizavimo uždavinys. Po to buvo išspręsti lenkimo momentų minimizavimo uždavinys ir bendrasis sinchroninis lenkimo momentų ir reakcijų minimizavimo uždavinys. Šitame skyriuje visi šie trys uždaviniai buvo spręsti laikant, kad sijyno sijų skerspjūvio matmenys yra pastovūs (t. y. nebuvo sprendžiamas optimalių sijos skerspjūvio matmenų nustatymo uždavinys): $h_{sk} = const$ ir $b_{sk} = const$. Tokia sąlyga yra priimta tam, kad būtų įvertinta pasiūlyto algoritmo (kompromisinės tikslo funkcijos (2.12)) kokybė, t. y. būtų gali-

mybė palyginti atliktų bandymų rezultatus su (Šešok *et al.* 2010; Belevičius *et al.* 2011) gautais rezultatais.



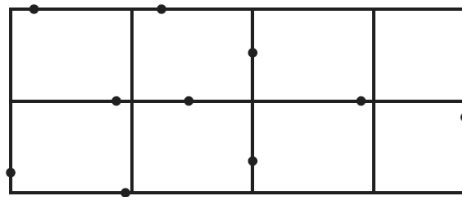
2.5 pav. 10 polių sijynas
Fig. 2.5. 10 pile grillage

Reakcijų minimizavimo uždavinys. Sprendžiant šį uždavinį tikslo funkcijoje (2.12) lenkimo momentui sijoje svarinis koeficientas $w_2 = 0$. Eksperimentiškai yra parinktos tokios GA genetinių parametų reikšmės: populiacijos dydis – 20 individų, kryžminimo tikimybė – 80%, mutacijos tikimybė – 20%, generacijų skaičius – 100. Šis uždavinys paprastai konverguoja per ~ 70 generacijų, todėl 100 populiacijų generavimas leidžia pagrįstai teigti, jog šiuo konkrečiu atveju geresnio sprendinio nebebus įmanoma gauti. Buvo atlikta 20 nepriklausomų skaitinių bandymų, t. y. uždavinys yra spręstas 2000 kartų. Akivaizdu, kad toks nepriklausomų skaitinių bandymų skaičius yra pakankamas, kad būtų gautas racionalus sprendinys. 2.1 lentelėje yra pateikti visi surūšiuoti sprendiniai pradedant geriausiu (190,35) ir baigiant blogiausiu (205,67). Geriausią sprendinį (190,35) atitinkanti polių išdėstymo schema yra pateikta 2.6 pav. Didžiausias absoliutiniu dydžiu lenkimo momentas esant tokiam polių išsidėstymui – 119,24, o didžiausia reakcija – 190,35. Vienas nepriklausomas atsitiktinis bandymas (100 generacijų) trunka apie 356 sek. (PC Intel® Xeon® 2,49 GHz 8,00 GB RAM).

2.1 lentelė. Reakcijų minimizavimo uždavinio rezultatai 10 polių sijynui

Table 2.1. Results of reactive forces minimization task for 10 pile grillage

Bandymas	Tikslo funkcija	Bandymas	Tikslo funkcija
1	190,35	16	197,34
2	190,57	17	197,71
3	191,16	18	197,82
4	192,49	19	198,06
5	192,56	20	198,66
6	192,70	21	198,76
7	193,51	22	199,13
8	194,11	23	200,00
9	194,49	24	200,48
10	194,91	25	200,50
11	195,34	26	201,58
12	196,54	27	201,78
13	196,69	28	202,68
14	197,27	29	204,79
15	197,33	30	205,67



2.6 pav. Polių išdėstymo schema reakcijų minimizavimo uždaviniui
Fig. 2.6. Pile distribution scheme for reactive forces minimization task

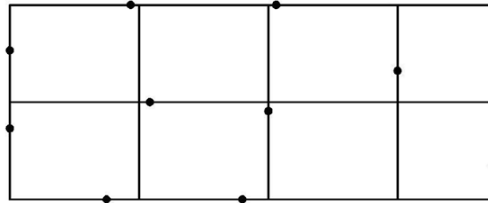
Palyginimui, geriausias to paties uždavinio (2.5 pav.) optimizavimo sprendinys (Šešok *et al.* 2010) yra 184,89, kuris yra tik apie 2,87% geresnis (t. y. mažesnis) nei šiame darbe gautas geriausias sprendinys (190,35). Be to, šiame darbe gautas geriausias sprendinys (190,35) yra tik apie 3,46% nutolęs (t. y. blogesnis) nuo globalaus sprendinio (183,7656).

2.2 lentelė. Lenkimo momentų minimizavimo uždavinio rezultatai 10 polių sijynui
Table 2.2. Results of bending moments minimization task for 10 pile grillage

Bandymas	Tikslo funkcija	Bandymas	Tikslo funkcija
1	63,70	16	79,32
2	64,02	17	79,51
3	66,92	18	79,90
4	69,45	19	80,17
5	70,33	20	80,32
6	75,41	21	82,16
7	75,85	22	83,26
8	75,90	23	84,03
9	76,64	24	85,68
10	77,42	25	88,30
11	78,37	26	89,15
12	78,42	27	89,20
13	79,03	28	91,17
14	79,07	29	92,58
15	79,23	30	101,69

Lenkimo momentų minimizavimo uždavinys. Sprendžiant šį uždavinį tikslo funkcijoje (2.12) lenkimo momentui sijoje svorinis koeficientas $w_1 = 0$. Eksperimentiškai yra parinktos tokios GA genetinių parametrų reikšmės: populiacijos dydis – 30 individų, kryžminimo tikimybė – 60%, mutacijos tikimybė – 20%, generacijų skaičius – 100. Šis uždavinys paprastai konverguoja per ~60 generacijų, todėl 100 populiacijų generavimas leidžia pagrįstai teigti, jog šiuo konkrečiu atveju geresnio sprendinio nebebus įmanoma gauti. Buvo atlikta 30 nepriklausomų skaitinių bandymų, t. y. uždavinys yra spęstas 3000 kartų. Akivaizdu, kad toks nepriklausomų skaitinių bandymų skaičius yra pakankamas, kad būtų gautas racionalus sprendinys. 2.2 lentelėje yra pateikti visi surūšiuoti sprendiniai pradedant geriausiu (63,70) ir baigiant blogiausiu (82,16). Geriausią sprendinį (63,70) atitinkanti polių išdėstymo schema yra pateikta 2.7 pav. Palyginus su reakcijų minimizavimo uždaviniu, didžiausias absoliutiniu dydžiu lenkimo momentas sumažėja apie 87,19% (nuo 119,24 iki 63,70), tačiau didžiausia reakcija padidėja apie 58,40% (nuo 190,35 iki 301,51). Vienas nepriklausomas atsitikti-

nis bandymas (100 generacijų) trunka apie 504 sek. (PC Intel® Xeon® 2,49 GHz 8,00 GB RAM).



2.7 pav. Polių išdėstymo schema lenkimo momentų minimizavimo uždaviniui

Fig. 2.7. Pile distribution scheme for bending moments minimization task

2.3 lentelė. Lenkimo momentų minimizavimo uždavinio rezultatai 10 polių sijynai
Table 2.3. Results of bending moments minimization task for 10 pile grillage

BE taškų skaičius	Bandymo trukmė, sek.	Geriausia tikslo funkcijos reikšmė
3	504	63,70
4	529	67,01
5	557	65,73
6	580	69,93
7	601	67,79
8	622	69,82
9	639	64,07

Viename baigtiniame elemente lenkimo momentus galima suskaičiuoti bet kuriuose pasirinktuose strypo taškuose (2.6). Todėl norint nustatyti optimalų baigtinio elemento (BE) taškų, kuriuose skaičiuojami lenkimo momentai sijoje, skaičių, būtina atlikti skaitinius bandymus. Mažiausias tokių taškų skaičius yra 3, t. y. lenkimo momentai skaičiuojami strypo pradiniam, viduriniame ir galiniame taškuose. Todėl lenkimo momentų minimizavimo uždavinys buvo spęstas, kai baigtinio elemento taškų, kuriuose skaičiuojami lenkimo momentai, skaičius buvo didinamas nuo 3-jų iki 9-ių imtinai (2.3 lentelė). Šis taškų skaičiaus intervalas buvo pasirinktas eksperimentiškai.

2.4 lentelė. Bendrojo sinchroninio lenkimo momentų ir reakcijų minimizavimo uždavinio rezultatai 10 polių sijynui

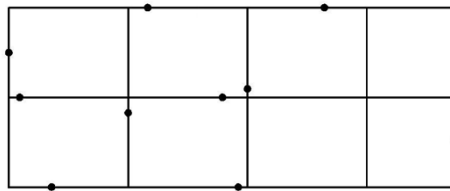
Table 2.4. Results of bending moments and reactive forces simultaneous minimization task for 10 pile grillage

Bandymas	Tikslo funkcija	Bandymas	Tikslo funkcija
1	141,32	16	152,37
2	143,03	17	152,59
3	143,86	18	154,42
4	144,68	19	154,46
5	145,81	20	155,60
6	146,09	21	155,98
7	147,10	22	156,75
8	147,19	23	156,79
9	147,42	24	156,86
10	147,84	25	158,34
11	148,65	26	159,26
12	148,69	27	159,32
13	149,34	28	159,63
14	149,81	29	160,30
15	151,26	30	161,59

Akivaizdu (2.3 lentelė), kad geriausia tikslo funkcijos (2.12) reikšmė (63,70) gauta, kai BE taškų, kuriuose skaičiuojami lenkimo momentai, skaičius (toliau *taškų skaičius*) yra lygus 3. Šį sprendinį (63,70) atitinkanti polių išdėstymo schema pateikta 2.7 pav. Kaip minėta aukščiau, šiuo atveju didžiausia reakcija lygi 301,51, o didžiausias absoliutiniu dydžiu lenkimo momentas – 63,70. Taškų skaičių didinant nuo 3-jų iki 9-ių, vieno nepriklausomo atsitiktinio bandymo (100 generacijų) trukmė didėja. Mažiausia bandymo trukmė (apie 504 sek.) gauta, kai taškų skaičius lygus 3 (PC Intel® Xeon® 2,49 GHz 8,00 GB RAM). Akivaizdu, kad lenkimo momentų minimizavimo uždaviniui optimalus taškų skaičius yra 3.

Bendrasis sinchroninis lenkimo momentų ir reakcijų minimizavimo uždavinys. Sprendžiant šį uždavinį tikslo funkcijoje (2.12) abu svoriniai koeficientai lygūs, t. y. $w_1 = w_2 = 0,5$. Deja, sprendžiant sinchroninį momentų ir reakcijų mi-

nimizavimo uždavinį (2.12) globaliojo sprendinio gauti iš anksto neįmanoma, kad ir kokia būtų tikslo funkcijos (2.12) svorinių koeficientų (w_1 ir w_2) kombinacija. Eksperimentiškai yra parinktos tokios GA genetinių parametrų vertės: populiacijos dydis – 30 individų, kryžminimo tikimybė – 50%, mutacijos tikimybė – 20%, generacijų skaičius – 100. Šis uždavinys paprastai konverguoja per ~70 generacijų, todėl 100 populiacijų generavimas leidžia pagrįstai teigti, jog šiuo konkrečiu atveju geresnio sprendinio nebebus įmanoma gauti. Buvo atlikta 30 nepriklausomų skaitinių bandymų, t. y. uždavinys buvo spręstas 3000 kartų. Akivaizdu, kad toks nepriklausomų skaitinių bandymų skaičius yra pakankamas, kad būtų gautas racionalus sprendinys. 2.4 lentelėje yra pateikti visi surūšiuoti sprendiniai pradedant geriausiu (141,32) ir baigiant blogiausiu (161,59). Geriausią sprendinį (141,32) atitinkanti polių išdėstymo schema yra pateikta 2.8 pav. Tokiam kompromisniam uždaviniui didžiausias absoliutiniu dydžiu lenkimo momentas – 84,98 (gauta jo geriausia reikšmė yra 63,70), o didžiausia reakcija – 197,65 (190,35). Vienas nepriklausomas atsitiktinis bandymas (100 generacijų) trunka apie 504 sek. (PC Intel® Xeon® 2,49 GHz 8,00 GB RAM).



2.8 pav. Polių išdėstymo schema bendrajam sinchroniam lenkimo momentų ir reakcijų minimizavimo uždaviniui

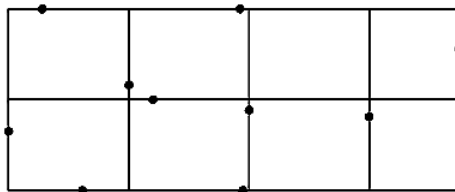
Fig. 2.8. Pile distribution scheme for bending moments and reactive forces simultaneous minimization task

Bendrasis sinchroninis lenkimo momentų ir reakcijų minimizavimo uždavinys, analogiškai kaip ir lenkimo momentų minimizavimo uždavinys, buvo išspręstas atlikus skaitinius bandymus. Buvo nustatytas optimalus BE taškų, kuriuose skaičiuojami lenkimo momentai sijoje, skaičius. Gauti rezultatai (2.5 lentelė) rodo, jog taškų skaičių didinant nuo 3-jų iki 9-ių, geriausia tikslo funkcijos (2.12) reikšmė (137,94) gauta, kai yra 5 taškai. Šį sprendinį (137,94) atitinkanti polių išdėstymo schema pateikta 2.9 pav. Aukščiau minėtam kompromisniam uždaviniui didžiausias absoliutiniu dydžiu lenkimo momentas – 77,79 (gauta jo geriausia reikšmė yra 63,70), o didžiausia reakcija – 198,10 (190,35). Vienas nepriklausomas atsitiktinis bandymas (100 generacijų) trunka apie 552 sek. (PC Intel® Xeon® 2,49 GHz 8,00 GB RAM).

2.5 lentelė. Bendrojo sinchroninio lenkimo momentų ir reakcijų minimizavimo uždavinio rezultatai 10 polių sijynui

Table 2.5. Results of bending moments and reactive forces simultaneous minimization task for 10 pile grillage

BE taškų skaičius	Bandymo trukmė, sek.	Geriausia tikslo funkcijos reikšmė
3	504	141,32
4	533	140,24
5	552	137,94
6	574	144,14
7	598	144,27
8	625	139,60
9	646	143,24



2.9 pav. Polių išdėstymo schema bendrajam sinchroniniam lenkimo momentų ir reakcijų minimizavimo uždaviniui

Fig. 2.9. Pile distribution scheme for bending moments and reactive forces simultaneous minimization task

Rezultatai (2.5 lentelė) rodo, kad didinant taškų skaičių nuo 3-jų iki 9-ių, skirtumas tarp geriausių tikslo funkcijos reikšmių yra labai nedidelis (didžiausias skirtumas apie 4,59%), tačiau bandymo trukmė pastebimai išauga: bandymo trukmė, kai taškų skaičius yra 5, padidėja apie 9,52% lyginant su bandymo trukme, kai taškų skaičius yra 3. Kadangi skaičiavimo laikas (bandymo trukmė) yra ribojantysis faktorius visiems globaliojo optimizavimo uždaviniams, tuomet akivaizdu, jog optimalus taškų skaičius yra 3, nes tada vieno nepriklausomo bandymo trukmė mažiausia – 504 sek. (PC Intel® Xeon® 2,49 GHz 8,00 GB RAM).

2.8. Tikslų funkcijos kriterijų įtaka optimizavimo uždaviniui

Bendrasis sinchroninis lenkimo momentų ir reakcijų minimizavimo uždavinys. Tikslų funkcijos (2.12) komponentų $f_1(x)$ ir $f_2(x)$ matavimo vienetai yra skirtingos prigimties. Todėl būtina atlikti tyrimus ir nustatyti tokią svorinių koeficientų w_1 ir w_2 santykio reikšmę (Venanzi and Materazzi 2007), kuriai esant komponentai $f_1(x)$ ir $f_2(x)$ turėtų vienodą įtaką tikslo funkcijai.

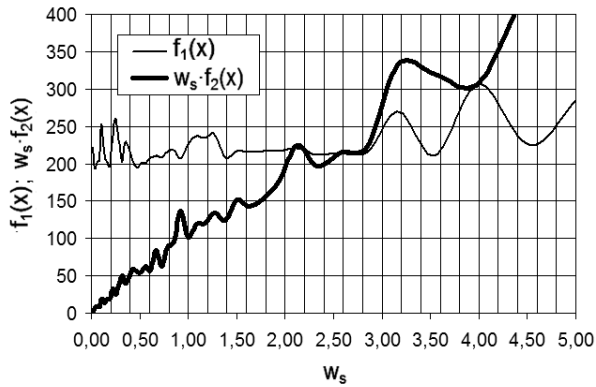
Padalijus tikslo funkciją (2.12) iš svorinio koeficiento $w_1 > 0$ ir įvedus svorinių koeficientų santykį $w_s = w_2 / w_1$ gauname atitinkamą tikslo funkcijos pavidalą:

$$f^* = \min_{x \in D} f(x) = f_1(x) + w_s \cdot f_2(x). \quad (2.22)$$

Kad būtų įvertinta, kokią įtaką tikslo funkcijai (2.22) turi komponentai $f_1(x)$ ir $w_s \cdot f_2(x)$, buvo atlikti skaitiniai bandymai, kai w_s reikšmės kinta intervale $[0,01;50,00]$ (čia $w_1 + w_2 = 1,0$). Eksperimentiškai yra parinktos tokios GA genetiinių parametrų vertės: populiacijos dydis – 28 individų, kryžminimo tikimybė – 50%, mutacijos tikimybė – 10%, generacijų skaičius – 100. Šis uždavinys paprastai konverguoja įvykus ~70 generacijų, todėl 100 populiacijų generavimas leidžia pagrįstai teigti, jog šiuo konkrečiu atveju geresnio sprendinio nebebus įmanoma gauti. Buvo atlikta 28 nepriklausomi skaitiniai bandymai, t. y. uždavinys buvo spręstas 2800 kartų. Akivaizdu, kad toks nepriklausomų skaitinių bandymų skaičius yra pakankamas, kad būtų gautas racionalus sprendinys.

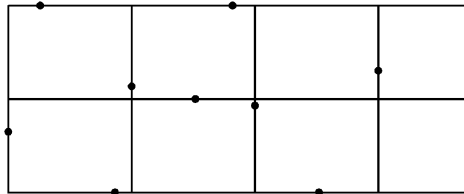
Iš bandymų rezultatų matyti, jog komponentai $f_1(x)$ ir $w_s \cdot f_2(x)$ turi vienodą įtaką tikslo funkcijai (2.22), kai $w_s \approx 2,57$ (2.10 pav.). Šiam uždaviniui gauta geriausia tikslo funkcijos (2.12) reikšmė yra 120,34 (ši sprendinį atitinkantis polių išdėstymas parodytas 2.11 pav.), didžiausias absoliutiniu dydžiu lenkimo momentas – 83,60 (gauta jo geriausia reikšmė yra 63,70), o didžiausia reakcija – 214,82 (190,35). Vienas nepriklausomas atsitiktinis bandymas (100 generacijų) trunka apie 402 sek. (PC Intel® Xeon® 2,49 GHz 8,00 GB RAM).

Kai $w_s > 2,57$ (2.10 pav.), tikslo funkcijai (2.22) didesnę įtaką turi komponentas $w_s \cdot f_2(x)$ (t. y. lenkimo momentas). Ir atvirkščiai, kai $w_s < 2,57$, didesnę įtaką tikslo funkcijai (2.22) turi komponentas $f_1(x)$ (t. y. reakcija).



2.10 pav. Komponentų $f_1(x)$ ir $f_2(x)$ įtaka tikslo funkcijai (10 polių sijynas)

Fig. 2.10. Contribution of components $f_1(x)$ and $f_2(x)$ to the objective function (10 pile grillage)



2.11 pav. Polių išdėstymo schema bendrajam sinchroniam lenkimo momentų ir reakcijų minimizavimo uždaviniui

Fig. 2.11. Pile distribution scheme for bending moments and reactive forces simultaneous minimization task

Iš grafiko (2.10 pav.) matyti, kad egzistuoja dar kelios w_s reikšmės (t. y. kreivių susikirtimo ar susilietimo taškai grafike), prie kurių $f_1(x)$ ir $w_s \cdot f_2(x)$ įtaka kompromisinei tikslo funkcijai (2.22) yra vienoda. Galima argumentuoti, kad taip atsitiko dėl skaičiavimo tikslumo (apvalinimo) įtakos, tad šias w_s reikšmes galima atmesti ir neįvertinti.

2.9. Antrojo skyriaus išvados

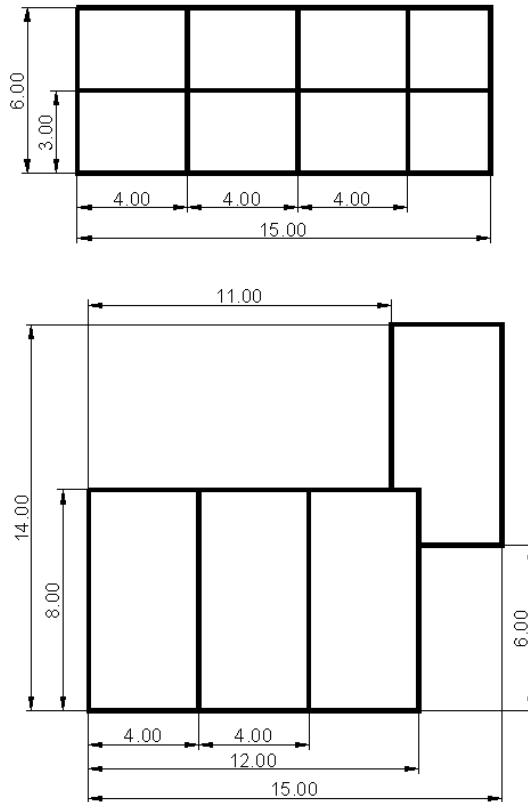
1. Kadangi disertacijoje nagrinėjamų uždavinių sprendimas reikalauja labai didelių kompiuterinių išteklių, todėl sijynų daugiakriteriam globaliajam optimizavimui labiau tinkami stochastiniai globalaus optimizavimo algoritmai, iš kurių genetiniai algoritmai yra vieni pranašiausių.
2. Pasiūlyta skaičiavimo technologija, kurioje naudojami genetiniai algoritmai, gali būti sėkmingai taikoma optimalioms polių padėtims sijynuose nustatyti.
3. Remiantis nustatyta priklausomybe tarp dviejų tikslo funkcijos kriterijų reikšmių ir svorinių koeficientų santykio reikšmių, galima pasirinkti įvairias sijynų schemas, priklausomai nuo to, kuris iš kriterijų (atraminė reakcija poliuje ar lenkimo momentas sijoje) turės didesnę, mažesnę ar vienodą įtaką optimizavimo uždaviniui. Tai gali būti naudinga inžinerinėje praktikoje, nes projektuotojas gali rinktis jam priimtinesnę sijyno schemą.

3

Genetinis algoritmas sijynų topologijos ir matmenų sinchroniniam optimizavimui

Šiame skyriuje yra aprašyta, kaip galima pritaikyti genetinių algoritmų koncepciją sijynų topologijai ir matmenims sinchroniškai optimizuoti. Informacija sijynų optimizavimo uždaviniams, turintiems nuo 10 iki 55 projektavimo kintamųjų, buvo gauta iš projektavimo biurų (Consultancy W. F. O. B. V., Paauw B. V. Aannemingsbedikf ir kt.) Olandijoje, kurie naudoja profesionalų skaičiavimo programinį paketą *MatrixFrame*, skirtą statybos inžinerijai. Šiuo programiniu paketu atliekama plieno ir gelžbetonio statinių analizė. Į šį programinį paketą yra integruota Teorinės mechanikos katedros (FMF, VGTU) darbuotojų sukurta programa, naudojanti lokalsios paieškos metodus vienakriteriam polių išdėstymo schemų optimizavimui. Šiame skyriuje pademonstruotas dviejų skirtingų topologijų sijynų (3.1 pav.) topologijos ir matmenų sinchroninis optimizavimas. Priklausomai nuo apkrovų konfigūracijos (3.2 pav.) mažiausias polių skaičius šiems sijynams yra atitinkamai 10 ir 15 polių. Pirmiausia yra nustatyti Pareto-optimalūs sprendiniai. Po to pateikiami kompromisinio optimizavimo uždavinio sprendimo genetiniu algoritmu (GA) rezultatai. Šio skyriaus medžiaga paskelbta šiose autoriaus publikacijose (disertacijos tema): Mačiūnas and Belevičius 2012,

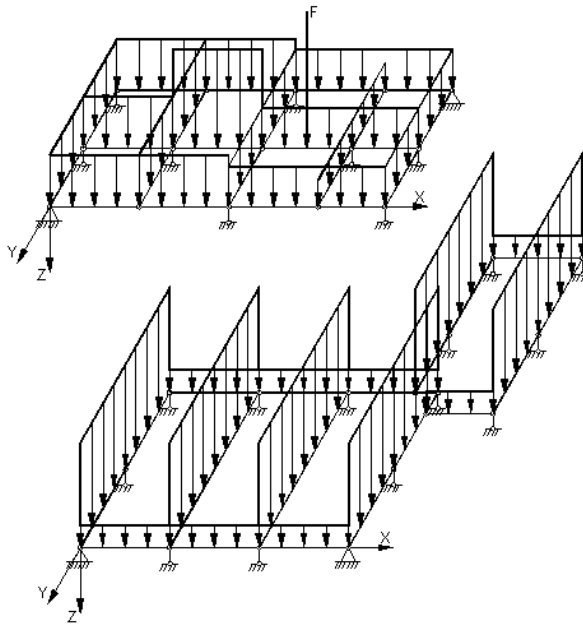
Mačiūnas *et al.* 2012a, Mockus *et al.* 2012, Mačiūnas *et al.* 2012b, Mačiūnas 2012.



3.1 pav. Dviejų skirtingų topologijų sijynai
Fig. 3.1. Two grillages with different topologies

3.1. Pareto-optimalūs sprendiniai

Svertinės kriterijų sumos metodas (1.1.3 poskyris), kuriuo pagrįsta tikslo funkcijos (2.12) formuluotė, yra tinkamas ieškant Pareto-optimalių sprendinių daugiakriteriams optimizavimo uždaviniams, turintiems išgaubtą Pareto-optimalumo ribą (1.3 pav.). Todėl pirmiausia būtina išnagrinėti tikslo funkcijos erdvės formą, t. y. atlikti tyrimus, kuriais būtų nustatyta, ar galimų sprendinių erdvė yra išgaubtos formos.



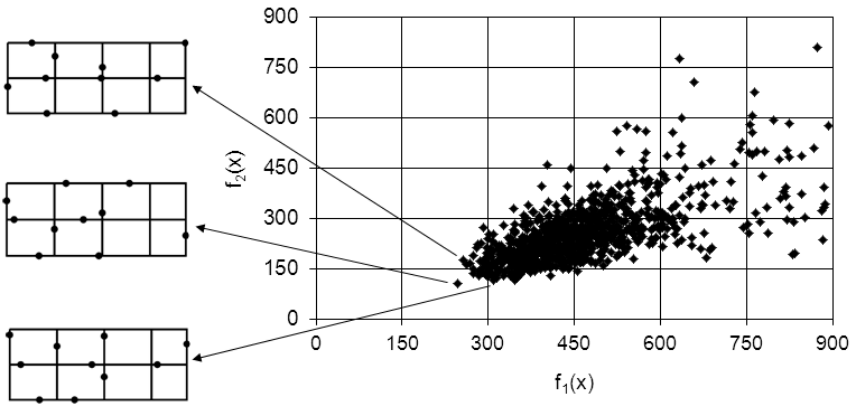
3.2 pav. Apkrovų konfigūracija: 10 polių ir 15 polių sijynai
Fig. 3.2. Load cases: 10 pile and 15 pile grilles

Tuo tikslu daug kartų buvo apskaičiuojami abu tikslo funkcijos (2.12) komponentai (2.13) ir (2.14) esant atsitiktinai parinktiems projektavimų kintamųjų rinkiniams, t. y. daug kartų buvo apskaičiuojami reakcijos poliuose bei lenkimo momentai sijose esant atsitiktinai parinktomis sijyno polių padėtimis bei sijų skerspjūvio matmenims.

Tyrimai atskleidė, kad visos uždavinio kraštinės sąlygos yra tenkinamos, todėl tikslo funkcijos (2.12) komponentų (2.13) ir (2.14) gautos reikšmės (3.3 pav. ir 3.4 pav.) yra patikimos.

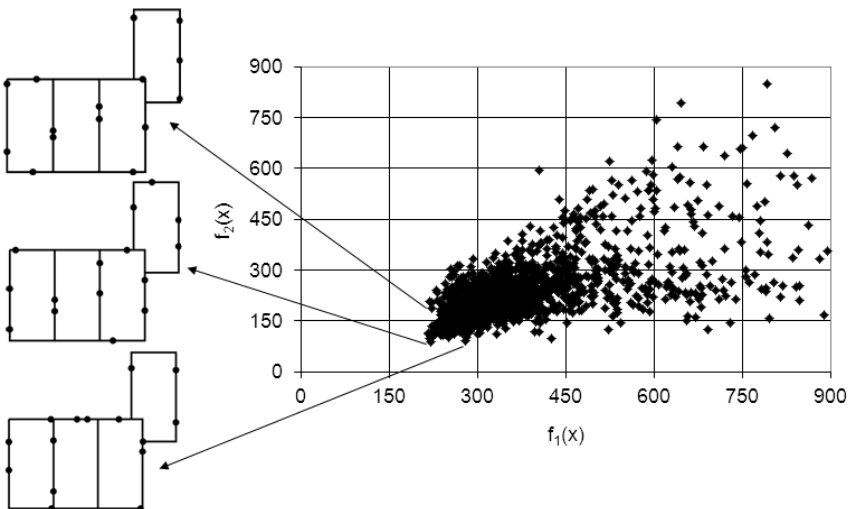
Iš gautų rezultatų (3.3 pav. ir 3.4 pav.) matyti, kad $f_1(x) - f_2(x)$ plokštumoje galimų sprendinių erdvė yra išgautos formos toje pusėje, kurioje yra laukiami optimalūs sprendiniai. Galima teigti, jog pasiūlyta kompromisinė tikslo funkcija (2.12) yra tinkama spręsti sijynų topologijos ir matmenų sinchroninio optimizavimo uždavinį ir visi įmanomi Pareto sprendiniai gali būti nustatyti.

3.3 ir 3.4 paveiksluose taip pat yra pademonstruotos polių išdėstymo schemas, kurios atitinka skirtingus Pareto-optimalius sprendinius. Tai gali būti naudinga inžinerinėje praktikoje, kadangi projektuotojas gali pasirinkti jam priimtinesnę schemą.



3.3 pav. Tikslų funkcijos galimi sprendiniai $f_1(x) - f_2(x)$ erdvėje (10 polių sijynas)

Fig. 3.3. Possible solutions of the objective function in $f_1(x) - f_2(x)$ space (10 pile grillage)



3.4 pav. Tikslų funkcijos galimi sprendiniai $f_1(x) - f_2(x)$ erdvėje (15 polių sijynas)

Fig. 3.4. Possible solutions of the objective function in $f_1(x) - f_2(x)$ space (15 pile grillage)

3.2. Optimalūs genetiniai parametrai ir sijos skerspjūvio matmenys

Genetinio algoritmo parametrai 10 polių ir 15 polių sijynams (3.1 pav. ir 3.2 pav.) buvo parinkti remiantis skaitiniais bandymais. Skaitinių bandymų metu populiacijos dydis kito nuo 10 iki 80 individų, mutacijos ir kryžminimo tikimybės – nuo 10% iki 90%. Individo (sprendinio) ilgis:

1. 12 genų 10 polių sijynui (10 polių pozicijų, sijos skerspjūvio aukštis ir plotis).
2. 17 genų 15 polių sijynui (15 polių pozicijų, sijos skerspjūvio aukštis ir plotis).

Tikslo funkcijos (2.12) svoriniai koeficientai $w_1 = w_2 = 0,5$. Prieš atliekant skaitinius tyrimus būtina nustatyti ribas, kuriose gali kisti sijos skerspjūvio matmenys. Sijos skerspjūvio matmenų nustatymui buvo parinkti praktiniu požiūriu priimtini intervalai:

1. Sijos skerspjūvio aukščiui: $h_{sk} \in [0,1;0,9]$.
2. Sijos skerspjūvio pločiui: $b_{sk} \in [0,1;0,9]$.

Visais atvejais buvo sugeneruota 100 populiacijų ir buvo gautos optimizavimo uždavinio optimalios genetinių parametrų vertės (3.1 lentelė) 10 polių ir 15 polių sijynams.

3.1 lentelė. Optimalios genetinių algoritmų parametrų vertės

Table 3.1. Optimal values of parameters genetic algorithms

Pavadinimas	Reikšmė	Reikšmė
Sijynas	10 polių	15 polių
Populiacijos dydis	50 individų	46 individų
Kryžminimo tikimybė	70%	50%
Mutacijos tikimybė	10%	10%

Kadangi GA yra stochastiniai algoritmai, todėl abu sijynai (10 polių ir 15 polių) buvo optimizuoti 30 kartų naudojant gautus genetinius parametrus tam, kad rezultatų pasiskirstymas (išsibarstymas) būtų priimtinas (3.2 lentelė).

3.2 lentelė. Optimizavimo rezultatai genetiniu algoritmu

Table 3.2. Optimization results using genetic algorithm

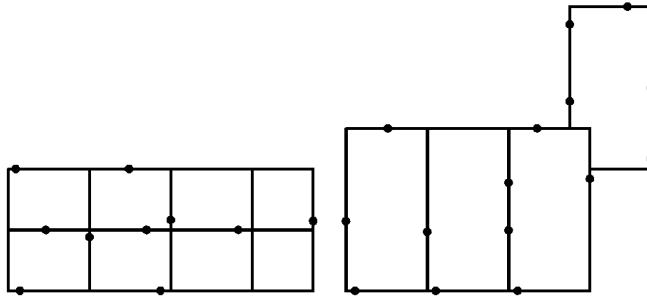
Pavadinimas	Reikšmė	Reikšmė
Sijynas	10 polių	15 polių
Algoritmo paleidimo skaičius	30	30
Generacijų skaičius	100	100
Geriausia tikslo funkcijos reikšmė	122,46	85,84
Rezultatų variacijos koeficientas	2,55%	2,93%
Sijos skerspjūvio matmenys $h_{sk} \times b_{sk}$	0,9 x 0,1	0,9 x 0,1
Sijos skerspjūvio matmenų santykis h_{sk} / b_{sk}	9,0	9,0

Polių išdėstymo schemas, atitinkančios geriausią tikslo funkcijos (2.12) sprendinį (122,46 – 10 polių sijynui ir 85,84 – 15 polių sijynui), yra pateiktos 3.5 pav.

Iš rezultatų matyti, kad sprendžiant optimizavimo uždavinį dėl esančių apribojimų (2.16) geriausią tikslo funkcijos (2.12) sprendinį (3.2 lentelė) gauname, kai sijos skerspjūvio aukščio matmuo įgyja didžiausią reikšmę iš intervalo $h_{sk} \in [0,1;0,9]$, o sijos skerspjūvio pločio matmuo įgyja mažiausią reikšmę iš intervalo $b_{sk} \in [0,1;0,9]$.

Atlikus skaitinius bandymus, kai skerspjūvio matmenys įgyja reikšmes iš intervalų $h_{sk} \in [0,2;0,8]$ ir $b_{sk} \in [0,2;0,8]$, gauname, kad geriausią tikslo funkcijos (2.12) sprendinį (133,63 – 10 polių sijynui ir 94,32 – 15 polių sijynui) gauname, kai sijos skerspjūvio aukščio matmuo įgyja didžiausią reikšmę ($h_{sk} = 0,8$), o sijos skerspjūvio pločio matmuo įgyja mažiausią reikšmę ($b_{sk} = 0,2$) (A priedas). Galima teigti, jog neegzistuoja tokia skerspjūvio matmenų kombinacija, prie kurios būtų gaunama geresnė tikslo funkcijos (2.12) reikšmė nei pateikta 3.2 lentelėje (A priedas).

Galima daryti išvadą, kad pasirinkus intervalus sijos skerspjūvio aukščiui ir pločiui, geriausią tikslo funkcijos (2.12) sprendinį gausime, kai sijos skerspjūvio aukščio matmuo įgis didžiausią reikšmę iš pasirinkto intervalo, o sijos skerspjūvio pločio matmuo įgis mažiausią reikšmę iš pasirinkto intervalo, t. y. optimizavimo algoritmas „stengiasi“, kad sijos skerspjūvis būtų kuo aukštesnis bei siauresnis. Dėl tokių sijos skerspjūvio geometrinų charakteristikų lenkimo momentai ($f_2(x)$) sijos įgyja mažiausias absoliutiniu dydžiu reikšmes bei tikslo funkcija (2.12) įgyja geresnes reikšmes.



3.5 pav. Polių išdėstymo schemas, atitinkančios geriausius tikslo funkcijos sprendinius (10 polių ir 15 polių sijynai)

Fig. 3.5. Pile placement schemes corresponding to the best solution of objective function (10 pile and 15 pile grillages)

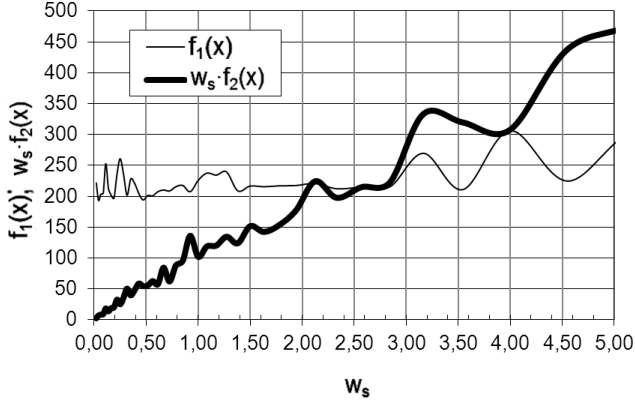
Tam, kad sijos skerspjūvio aukščio h_{sk} ir pločio b_{sk} didžiausias santykis $(h_{sk} / b_{sk})_{max}$ neįgytų tokių reikšmių, prie kurių galėtų kilti grėsmė stabilumui sijyno plokštumoje, būtina nustatyti leistiną sijos skerspjūvio aukščio h_{sk} ir pločio b_{sk} santykį $(h_{sk} / b_{sk})_{leist}$. Praktiniu požiūriu būtų priimtina, jei $(h_{sk} / b_{sk})_{leist} = 3,0$.

3.3. Tikslo funkcijos kriterijų įtaka optimizavimo uždaviniui skirtingos topologijos sijynuose

Remiantis 2.8 poskyryje pateikta metodika, įvertinta ir palyginta, kokią įtaką optimizavimo uždaviniui turi tikslo funkcijos (2.12) komponentai $f_1(x)$ ir $f_2(x)$ 10 polių ir 15 polių sijynuose (3.1 pav. ir 3.2 pav.). Pasitelkus kompromisinę tikslo funkciją (2.12) atitinkančią funkciją (2.22) bus palyginta, prie kokių svorinių koeficientų santykio $w_s = w_2 / w_1$ reikšmių, komponentų $f_1(x)$ ir $f_2(x)$ įtaka optimizavimo uždaviniui yra vienoda.

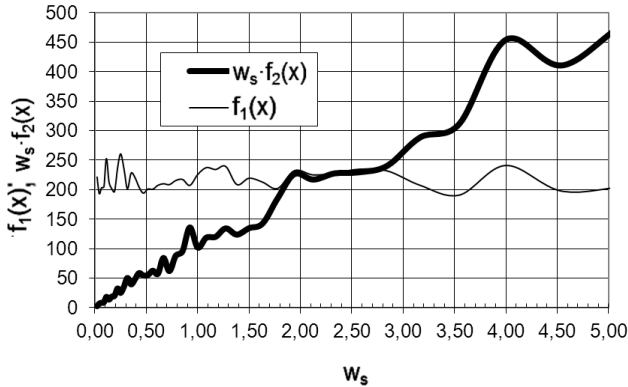
Atliktų tyrimų rezultatai atskleidė, kad abiejuose sijynuose (3.1 pav. ir 3.2 pav.) w_s reikšmės, apibrėžiančios komponentų $f_1(x)$ ir $f_2(x)$ vienodą įtaką optimizavimo uždaviniui, yra labai artimos: 10 polių sijynui $w_s \approx 2,57$ (kai $w_1 = 0,28$, $w_2 = 0,72$) (3.6 pav.) ir 15 polių sijynui $w_s \approx 2,33$ (kai $w_1 = 0,30$, $w_2 = 0,70$) (3.7 pav.). Taigi, kai $w_s > 2,57$ (3.6 pav.) ir $w_s > 2,33$ (3.7 pav.), tikslo funkcijai (2.22) didesnę įtaką turi komponentas $w_s \cdot f_2(x)$ (t. y. lenkimo momen-

tas). Ir atvirkščiai, kai $w_s < 2,57$ (3.6 pav.) ir $w_s < 2,33$ (3.7 pav.), didesnę įtaką tikslo funkcijai (2.22) turi komponentas $f_1(x)$ (t. y. reakcija).



3.6 pav. Komponentų $f_1(x)$ ir $f_2(x)$ įtaka tikslo funkcijai (10 polių sijynas)

Fig. 3.6. Contribution of components $f_1(x)$ and $f_2(x)$ to the objective function (10 pile grillage)



3.7 pav. Komponentų $f_1(x)$ ir $f_2(x)$ įtaka tikslo funkcijai (15 polių sijynas)

Fig. 3.7. Contribution of components $f_1(x)$ and $f_2(x)$ to the objective function (15 pile grillage)

Analogiški skaitiniai bandymai buvo atlikti su 18 polių bei 20 polių sijynais (B priedas ir C priedas) ir buvo gauta, kad 18 polių sijynui $w_s \approx 2,33$, o 20 polių sijynui $w_s \approx 2,57$ (B priedas ir C priedas). Galima teigti, jog tikėtinai bet kokiams apkrovoms veikiant bet kokios topologijos sijynus, komponentų $f_1(x)$ ir $f_2(x)$ įtaka optimizavimo uždaviniui bus vienoda, kai w_s reikšmė bus labai artima jau nustatytoms w_s reikšmėms 10 polių ($w_s \approx 2,57$) bei 15 polių ($w_s \approx 2,33$) sijynams. Remiantis aukščiau gautais rezultatais, toliau skaitiniuose bandymuose tikslo funkcijoje (2.12) bus naudojami tokie svoriniai koeficientai, kad komponentų $f_1(x)$ ir $f_2(x)$ įtaka optimizavimo uždaviniui būtų vienoda.

3.4. Optimizavimo uždavinio sprendimas, kai tikslo funkcijos komponentų įtaka uždaviniui vienoda

Optimizavimo uždavinio sprendimas pademonstruotas optimizuojant 10 polių ir 15 polių sijynus (3.1 pav. ir 3.2 pav.). Genetinio algoritmo parametrai šiems sijynams buvo parinkti remiantis skaitiniais bandymais (3.1 lentelė).

Tikslo funkcijos (2.12) svoriniai koeficientai (remiantis 3.3 poskyryje gautais rezultatais) yra tokie:

1. 10 polių sijynui: $w_1 = 0,28$, $w_2 = 0,72$ (t. y. $w_s \approx 2,57$).
2. 15 polių sijynui: $w_1 = 0,30$, $w_2 = 0,70$ (t. y. $w_s \approx 2,33$).

Prieš atliekant skaitinius tyrimus būtina nustatyti ribas, kuriose gali kisti šijos skerspjūvio matmenys. Šijos skerspjūvio matmenų nustatymui buvo parinkti praktiniu požiūriu priimtini intervalai:

1. Šijos skerspjūvio aukščiui: $h_{sk} \in [0,1;0,9]$.
2. Šijos skerspjūvio pločiui: $b_{sk} \in [0,1;0,9]$.

Be to, įvesta papildoma praktiniu požiūriu būtina sąlyga šijos skerspjūvio matmenims:

3. $(h_{sk} / b_{sk})_{max} \leq 3,0$.

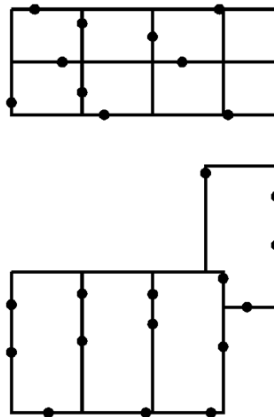
Visais atvejais buvo sugeneruota 100 populiacijų. Kadangi GA yra stochastiniai algoritmai, todėl abu sijynai (10 polių ir 15 polių) buvo optimizuoti 30 kartų, kad rezultatų pasiskirstymas (išsibarstymas) būtų priimtinas (3.3 lentelė).

Polių išdėstymo schemas, atitinkančios geriausią tikslo funkcijos (2.12) sprendinį (96,07 – 10 polių sijynui ir 78,68 – 15 polių sijynui), yra pateiktos 3.8 paveiksle.

3.3 lentelė. Optimizavimo genetiniu algoritmu rezultatai

Table 3.3. Optimization results using genetic algorithm

Pavadinimas	Reikšmė	Reikšmė
Sijynas	10 polių	15 polių
Algoritmo paleidimo skaičius	30	30
Generacijų skaičius	100	100
Geriausia tikslo funkcijos reikšmė	96,07	78,68
Rezultatų variacijos koeficientas	1,07%	1,88%
Sijos skerspjūvio matmenys $h_{sk} \times b_{sk}$	0,44 x 0,15	0,87 x 0,33
Sijos skerspjūvio matmenų santykis h_{sk} / b_{sk}	2,93	2,64

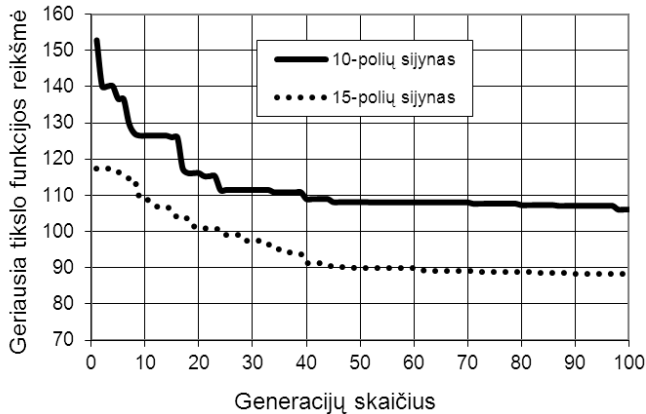


3.8 pav. Polių išdėstymo schemas, atitinkančios geriausius tikslo funkcijos sprendinius (10 polių ir 15 polių sijynai)

Fig. 3.8. Pile placement schemes corresponding to the best solution of objective function (10 pile and 15 pile grillages)

GA sprendinio tipinio konvergavimo istorijos (3.9 pav.) analizė atskleidė, kad geriausias sprendinys yra gaunamas konvergavimo istorijos pabaigoje, tačiau konvergavimo greičio kitimo pobūdis demonstruoja tam tikrus GA trūkumus. Pradinėje fazėje (vidutiniškai iki 30-os generacijos) GA konvergavimo greitis yra didelis palyginus su konvergavimo greičiu, sumažėjančiu daugiau nei 5% vėlesnėse fazėse (paprastai nuo 30-os generacijos), t. y. sprendinys konverguoja į lokalų optimalų sprendinį. Baigiamojoje fazėje (paprastai nuo 70-os generacijos) GA konvergavimo greitis praktiškai sumažėja iki nulio. Toks GA

konvergavimo greičio sumažėjimo pobūdis iš esmės susijęs su populiacijos individų asimiliacija, t. y. sprendiniai (individai) populiacijoje labai suvienodėja (supanašėja).



3.9 pav. Tipinis genetinio algoritmo konvergavimas
Fig. 3.9. Typical convergence of genetic algorithm

Todėl sprendimo stadijose, kur konvergavimo greitis sumažėja daugiau nei 5%, esami GA genetiniai parametrai nebetinkami ir yra reikalingas genetinių parametrų adaptavimas. Tikėtina, jog didesnė mutacijos tikimybė gali duoti impulsą visos populiacijos kokybiškam šuoliui iš lokalaus sprendinio taško bei suteikti galimybę genetiniam algoritmui atlikti paiešką didesniame galimų sprendinių erdvės plote. Be to, mažesnė populiacija turi daug lankstesnes adaptavimosi savybes. Tokiu būdu padidėja tikimybė surasti globalų optimalų sprendinį.

3.5. Adaptyvus genetinis algoritmas

Siūlomas euristinis adaptyvus genetinis algoritmas (AGA) sudarytas iš trijų pagrindinių dalių:

1. Nustatymas momento, kada reikalingas genetinio algoritmo adaptavimas.
2. Adaptyvus populiacijos dydžio operatorius, priklausantis nuo konvergavimo greičio.

3. Adaptyvus mutacijos operatorius, priklausantis nuo konvergavimo greičio.

Optimizavimo uždavinio sprendimas su AGA pademonstruotas optimizuojant 10 polių ir 15 polių sijynus (3.1 pav. ir 3.2 pav.). Genetinio algoritmo parametrai šiems sijynams buvo parinkti remiantis skaitiniais bandymais (3.1 lentelė). Taikomi parametrai (tikslo funkcijos (2.12) svoriniai koeficientai bei sijų skerspjūvio matmenys), kurie pateikti 3.4 poskyryje.

GA adaptavimo paleidimo momentas nustatomas įvertinat geriausio sprendinio populiacijoje konvergavimo greitį. Siūloma, kad adaptavimą reikia naudoti tuomet, kai skirtumas tarp dešimties iš eilės sugeneruotų tikslo funkcijos reikšmių yra mažesnis kaip 5,0%:

$$f_{jj} - f_{jj+10} < 5,0\%, \quad (3.1)$$

čia $f_{jj} - jj$ -toji tikslo funkcijos reikšmė.

Kai ši sąlyga (3.1) yra tenkinama, tuomet vykdomas GA adaptavimas taikant adaptyvų populiacijos dydžio operatorių bei adaptyvų mutacijos operatorių. Adaptyvus populiacijos dydžio operatorius (APDO) mažina populiacijos dydį (PD):

$$APDO = PD \cdot 0,75. \quad (3.2)$$

Kai pirmą kartą tenkinama adaptyvumo paleidimo sąlyga (3.1), pradinis populiacijos dydis yra sumažinamas 25% (3.2) ir gaunamas naujas populiacijos dydis. Kai vėl tenkinama (3.1) sąlyga, paskiausiai gautas populiacijos dydis yra vėl sumažinamas 25%. Tokia procedūra populiacijos dydžiui yra taikoma tol, kol tenkinamas apribojimas:

$$APDO > 10. \quad (3.3)$$

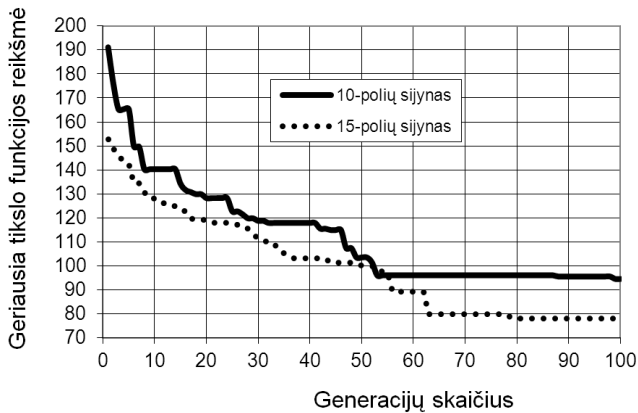
Adaptyvus mutacijos operatorius (AMO) didina mutacijos tikimybę p_mut :

$$AMO = p_mut \cdot 1,25. \quad (3.4)$$

Kai pirmą kartą yra tenkinama adaptyvumo paleidimo sąlyga (3.1), pradinė mutacijos tikimybė p_mut yra padidinama 25% (3.4) ir gaunama nauja mutacijos tikimybės reikšmė. Kai vėl tenkinama (3.1) sąlyga, paskiausiai gauta p_mut reikšmė yra vėl padidinama 25%. Tokia procedūra mutacijos tikimybei yra taikoma tol, kol tenkinamas apribojimas:

$$AMO \leq 100. \quad (3.5)$$

Atliktų skaitinių bandymų (sprendžiant optimizavimo uždavinį) su AGA rezultatai (3.10 pav.) rodo, jog greito konvergavimo fazė trunka beveik dvigubai ilgiau nei naudojant klasikinį GA. Adaptyviam GA paprastai pakanka 50–60 generacijų, kad pasiektų labai artimus ar net geresnius rezultatus lyginant su klasikinio GA.



3.10 pav. Tipinis adaptyvaus genetinio algoritmo konvergavimas
Fig. 3.10. Typical convergence of adaptive genetic algorithm

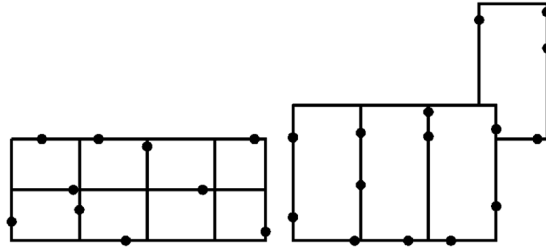
3.4 lentelė. Vieno nepriklausomo skaitinio bandymo (VNSB) trukmė
Table 3.4. Duration of one independent numerical experiment

Pavadinimas	Reikšmė	Reikšmė
Sijynas	10 polių	15 polių
VNSB trukmė su GA (100 generacijų)	504 sek.	698 sek.
VNSB trukmė su AGA (50-60 generacijų)	277 sek.	437 sek.
VNSB trukmės skirtumas tarp GA ir AGA	227 sek. (45,04%)	261 sek. (37,39%)

Vienas nepriklausomas skaitinis bandymas (VNSB) (100 generacijų) 10 polių sijynui su GA trunka apie 504 sek. (PC Intel® Xeon® 2,49 GHz 8,00 GB RAM). AGA dėl didesnio konvergavimo greičio paprastai suranda labai artimus ar net geresnius sprendinius (palyginus su GA) vidutiniškai per 277 sek. (50-60 generacijų). Šis skaičiavimų laikas 10 polių sijynui net 45,04% (arba 227 sek.)

trumpesnis palyginus su klasikiniu GA, o 15 polių sijynui – 37,39% (arba 261 sek.) trumpesnis (3.4 lentelė).

Kadangi skaičiavimo laikas yra lemiamas faktorius visiems globalaus optimizavimo uždaviniams, tai pasiūlytu adaptyviu genetiniu algoritmu per trumpesnę laiką gaunami geresni sijynų optimizavimo rezultatai (t. y. mažesnės tikslo funkcijos reikšmės) lyginant su klasikiniu genetiniu algoritmu.



3.11 pav. Polių išdėstymo schemas, atitinkančios geriausius tikslo funkcijos sprendinius (10 polių ir 15 polių sijynai)

Fig. 3.11. Pile placement schemes corresponding to the best solution of objective function (10 pile and 15 pile grillages)

3.5 lentelė. Optimizavimo rezultatai adaptyviu genetiniu algoritmu

Table 3.5. Optimization results of adaptive genetic algorithm

Pavadinimas	Reikšmė	Reikšmė
Sijynas	10 polių	15 polių
Algoritmo paleidimo skaičius	30	30
Generacijų skaičius	100	100
Geriausia tikslo funkcijos reikšmė	93,18	76,45
Rezultatų variacijos koeficientas	1,86%	2,06%
Sijos skerspjūvio matmenys $h_{sk} \times b_{sk}$	0,58 x 0,20	0,69 x 0,24
Sijos skerspjūvio matmenų santykis h_{sk} / b_{sk}	2,90	2,88

Polių išdėstymo schemas, atitinkančios geriausią tikslo funkcijos (2.12) sprendinį (93,18 – 10 polių sijynui ir 76,45 – 15 polių sijynui) su AGA, yra pateiktos 3.11 paveiksle. Skaitinių bandymų (sprendžiant optimizavimo uždavinį) su AGA rezultatai (3.5 lentelė) rodo, jog geriausios tikslo funkcijos (2.12) reikšmės 10 polių ir 15 polių sijynams buvo gautos atitinkamai 3,01% ir 2,83% geresnės nei su klasikiniu GA.

3.6. Genetinis algoritmas su paskirstymo strategija

Šiame skyriuje pademonstruota genetinio algoritmo su paskirstymo strategija (GAPS) pritaikymas sprendžiant sijynų optimizavimo uždavinį. Sijynų optimizavimo uždavinio sprendimas taikant GAPS yra sudarytas iš dviejų etapų:

1. Vienu metu yra paleidžiamas iš anksto nustatytas skaičius atskirų (savarankiškai veikiančių/tarpusavyje nepriklausomų) optimizavimo algoritmų (aprašytų 3.4 poskyryje), kurie yra vykdomi iki tokio momento, kai GA konvergavimo greitis sumažėja daugiau nei 5%. Pasiekus šį momentą, visi paleisti algoritmai yra sustabdomi ir iš kiekvienos atskiro algoritmo populiacijos atsitiktinai atrenkamas tam tikras iš anksto numatytas skaičius individų (sprendinių), iš kurių yra suformuojama nauja individų populiacija.
2. Paleidžiamas optimizavimo algoritmas (aprašytas 3.4 poskyryje), kuris geriausio sprendinio paiešką atlieka pradėdamas nuo naujai suformuotos individų populiacijos (paminėtos ankstesniame punkte).

Optimizavimo uždavinio sprendimas su GAPS pademonstruotas optimizuojant 10 polių ir 15 polių sijynus (3.1 pav. ir 3.2 pav.). Genetinio algoritmo parametrai šiems sijynams buvo parinkti remiantis skaitiniais bandymais (3.1 lentelė). Optimizavimo metu taikomi tokie parametrai (tikslo funkcijos (2.12) svoriniai koeficientai bei sijų skerspjūvio matmenys), kurie pateikti 3.4 poskyryje.

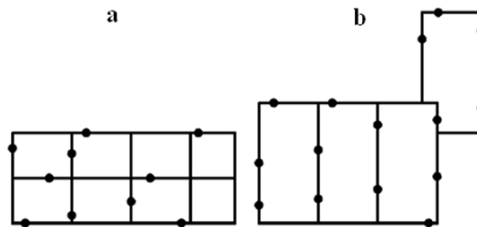
Skaitiniai bandymai taikant GAPS buvo atliekami, esant tokiems nustatymams šiai strategijai:

1. Vienu metu yra paleidžiami trys atskiri (tarpusavyje nepriklausomi) optimizavimo algoritmai (aprašytų 3.4 poskyryje), kurie yra vykdomi iki tokio momento, kai GA konvergavimo greitis sumažėja daugiau nei 5%. Pasiekus šį momentą visi paleisti algoritmai yra sustabdomi ir iš kiekvienos atskiro algoritmo populiacijos atsitiktinai atrenkami 16 individų (sprendinių) (toliau tekste ši operacija vadinama *atsitiktiniu atrankos būdu*), iš kurių yra suformuojama nauja individų populiacija, turinti 48 individus.
2. Paleidžiamas optimizavimo algoritmas (aprašytas 3.4 poskyryje), kuris geriausio sprendinio paiešką atlieka pradėdamas nuo naujai suformuotos 48 individų populiacijos.

GAPS paleidimo momentas bus nustatomas įvertinat geriausio sprendinio populiacijoje konvergavimo greitį. Siūloma, kad GAPS reikia paleisti tuomet,

kai skirtumas tarp dešimties iš eilės sugeneruotų tikslo funkcijos reikšmių yra mažesnis kaip 5,0% (3.1).

Pirmajame GAPS etape (t. y. tam momentui, kai tenkinama (3.1) sąlyga) kiekviename iš trijų atskirų optimizavimo algoritmų buvo gauti apčiuopiamai geresni (tačiau tarpusavyje labai supanašėję) sprendiniai (individai), iš kurių (aukščiau minėtu atsitiktiniu atrankos būdu) buvo suformuota nauja 48 individų (sprendinių) populiacija.



3.12 pav. Polių išdėstymo schemas, atitinkančios geriausius tikslo funkcijos sprendinius: a – 10 polių sijynas, b – 15 polių sijynas

Fig. 3.12. Pile placement schemes corresponding to the best solution of objective function: a – 10 pile grillage, b – 15 pile grillage

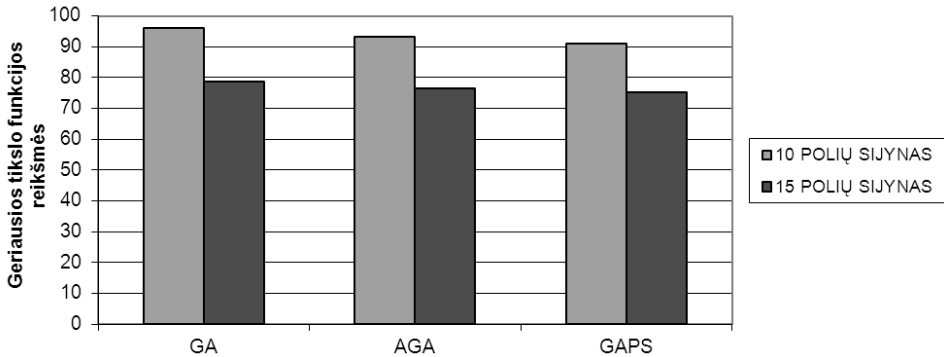
3.6 lentelė. Optimizavimo rezultatai taikant genetinį algoritmą su paskirstymo strategija
Table 3.6. Optimization results applying genetic algorithm with distribution strategy

Pavadinimas	Reikšmė	Reikšmė
Sijynas	10 polių	15 polių
Algoritmo paleidimo skaičius	30	30
Generacijų skaičius	100	100
Geriausia tikslo funkcijos reikšmė	91,15	75,34
Sijos skerspjūvio matmenys $h_{sk} \times b_{sk}$	0,63 x 0,22	0,38 x 0,13
Sijos skerspjūvio matmenų santykis h_{sk} / b_{sk}	2,86	2,92

Šioje naujoje 48 individų populiacijoje individai tarpusavyje skiriasi daug labiau, todėl antrajame GAPS etape optimizavimo algoritmas turi galimybę atlikti paiešką didesniame galimų sprendinių erdvės plote.

Skaitinių bandymų (sprendžiant optimizavimo uždavinį) su GAPS rezultatai (3.6 lentelė) rodo, jog geriausios tikslo funkcijos (2.12) reikšmės 10 polių ir 15 polių sijynams buvo gautos atitinkamai 5,12% ir 4,25% geresnės nei su klasiki-

niu GA. Geriausias tikslo funkcijos (2.12) reikšmės (taikant GAPS) atitinkančios polių išdėstymo schemos pateiktos 3.12 paveiksle.



3.13 pav. Algoritmų palyginimas
Fig. 3.13. Comparison of algorithms

Lyginant skaitinių bandymų rezultatus (3.5 lentelė ir 3.6 lentelė) matyti, jog geriausios tikslo funkcijos (2.12) reikšmės 10 polių ir 15 polių sijynams naudojant GAPS buvo gautos atitinkamai 2,18% ir 1,45% geresnės nei su AGA. Galima teigti, kad pritaikius AGA ir GAPS yra gaunami vidutiniškai nuo 3% iki 5% geresni rezultatai (t. y. mažesnės tikslo funkcijos reikšmės) nei naudojant klasikinį GA (3.13 pav.).

3.7. Trečiojo skyriaus išvados

1. Pasiūlyta kompromisinė tikslo funkcija yra tinkama spręsti sijynų topologijos ir matmenų sinchroninio optimizavimo uždavinį ir visi įmanomi Pareto optimalūs sprendiniai gali būti nustatyti.
2. Optimizavimo algoritmas geriausius sprendinius suranda, kai sijos skerspjūvio aukštis yra kiek įmanoma didesnis, o plotis kiek įmanoma mažesnis, nes tokiu atveju lenkimo momentai sijose įgyja mažiausias absoliutiniu dydžiu reikšmes, tuo pačiu ir tikslo funkcija įgyja optimalias reikšmes.
3. Remiantis nustatyta priklausomybe tarp dviejų tikslo funkcijos kriterijų reikšmių ir svorinių koeficientų santykio reikšmių, galima pasirinkti įvairias sijynų schemas, priklausomai nuo to, kuris iš kriterijų

(atraminė reakcija poliuje ar lenkimo momentas sijoje) turės didesnę, mažesnę ar vienodą įtaką optimizavimo uždaviniui. Tai gali būti naudinga inžinerinėje praktikoje, nes projektuotojas gali rinktis jam priimtinesnę sijyno schemą.

4. Pritaikius AGA, pasiūlyta ir sukurta skaičiavimo technologija, kuria gaunami 3,01% (10 polių sijynui) ir 2,83% (15 polių sijynui) geresni rezultatai (t. y. mažesnės tikslo funkcijos reikšmės) nei naudojant klasikinį GA.
5. Pritaikius GAPS, pasiūlyta ir sukurta skaičiavimo technologija, kuria gaunami 5,12% (10 polių sijynui) ir 4,25% (15 polių sijynui) geresni rezultatai (t. y. mažesnės tikslo funkcijos reikšmės) nei naudojant klasikinį GA.

Bendrosios išvados

1. Sijynų daugiakriteriam globaliajam optimizavimui labiau tinkami yra stochastiniai globalaus optimizavimo algoritmai, iš kurių genetiniai algoritmai yra vieni pranašiausių.

2. Remiantis nustatyta priklausomybe tarp dviejų tikslo funkcijos kriterijų reikšmių ir svorinių koeficientų santykio reikšmių, galima pasirinkti įvairias sijynų schemas, priklausomai nuo to, kuris iš kriterijų (atraminė reakcija poliuje ar lenkimo momentas sijoje) turės didesnę, mažesnę ar vienodą įtaką optimizavimo uždaviniui. Tai gali būti naudinga inžinerinėje praktikoje, nes projektuotojas gali rinktis jam priimtinesnę sijyno schemą.

3. Panaudojus GA, sukurta skaičiavimo technologija sijynų sinchroniam topologijos ir matmenų optimizavimui. Pritaikius šią technologiją, gali būti nustatyti visi galimi Pareto optimalūs sprendiniai.

4. Panaudojus AGA, sukurta skaičiavimo technologija sijynų sinchroniam topologijos ir matmenų optimizavimui. Pritaikius šią technologiją, gauti vidutiniškai 3% geresni rezultatai (t. y. mažesnės tikslo funkcijos reikšmės) nei naudojant klasikinį GA. Dėl padidėjusio konvergavimo greičio AGA akivaizdžiai sutaupto skaičiavimo laiko, kuris yra lemiamas veiksnys visiems globaliojo optimizavimo uždaviniams.

5. Panaudojus GAPS, sukurta skaičiavimo technologija sijynų sinchroniam topologijos ir matmenų optimizavimui. Pritaikius šią technologiją, gauti

vidutiniškai 5% geresni rezultatai (t. y. mažesnės tikslo funkcijos reikšmės) nei naudojant klasikinį GA.

Literatūra ir šaltiniai

Abraham, A.; Jain, L. C.; Goldberg, R. 2005. *Evolutionary Multiobjective Optimization – Theoretical Advances and Applications. (Advanced Information and Knowledge Processing)*. Springer, Berlin. 320 p.

Adeli, H.; Nai-Tsang, C. 1993. Integrated Genetic Algorithm for Optimization of Space Structures, *Journal of Aerospace Engineering (ASCE)* 6(4): 315–327.

Adeli, H.; Nai-Tsang, C. 1994. Concurrent Genetic Algorithm for Optimization of Large Structures, *Journal of Aerospace Engineering (ASCE)* 7(3): 276–296.

Alba, E.; Chicano, J. F. 2006. Evolutionary Algorithms in Telecommunications, in *IEEE Mediterranean Electrotechnical Conference, MELECON 2006*, 795–798.

Atkočiūnas, J.; Čižas, A. 2002. Aleksandras Čyras and Optimization in Structural Mechanics, *Journal of Civil Engineering and Management* 8(1): 4–33.

Atkočiūnas, J.; Merkevičiūtė, D.; Venskus, A.; Skaržauskas, V. 2007. Nonlinear Programming and Optimal Shakedown Design of Frames, *Mechanika* 2(64): 27–33.

Bagley, J. D. 1967. *The Behavior of Adaptive Systems which employ Genetic and Correlation Algorithms*. PhD thesis, The University of Michigan, College of Literature, Science, and the Arts, Computer and Communication Sciences Department, Ann Arbor, MI, USA.

Baker, J. E. 1985. Adaptive Selection Methods for Genetic Algorithms, in *Proceedings of the 1st International Conference on Genetic Algorithms*, 101–111.

Barauskas, R.; Belevičius, R.; Kačianauskas, R. 2004. *Baigtinių elementų metodo pagrindai [Finite Element Method Framework]*. Technika. Vilnius. 612 p.

Barricelli, N. A. 1954. Esempi numerici di processi di evoluzione, *Methodos* 6(21–22): 45–68.

Barricelli, N. A. 1957. Symbiogenetic Evolution Processes Realized by Artificial Methods, *Methodos* 9(35–36): 143–182.

Barricelli, N. A. 1962. Numerical Testing of Evolution Theories. Part I. Theroetical Introduction and Basic Tests, *Acta Biotheoretica* 16(1/2): 69–98.

Barricelli, N. A. 1963. Numerical Testing of Evolution Theories. Part II. Preliminary Tests of Performance. Symbiogenesis and Terrestrial Life, *Acta Biotheoretica* 16(3/4): 99–126.

Belevičius, R.; Valentinavičius, S. 2000. Optimization of Grillage-Type Foundations, *Civil Engineering* 6(4): 255–261.

Belevičius, R.; Valentinavičius, S. 2001. Optimization of Grillage-Type Foundations, in *Proceedings of 2nd European ECCOMAS and IACM Conference “Solids, Structures and Coupled Problems in Engineering”*. Cracow, Poland 26–29 June. CD-ROM 2001.

Belevičius, R.; Valentinavičius, S.; Michnevič, E. 2002. Multilevel Optimization of Grilages, *Journal of Civil Engineering and Management* 8(2): 98–103.

Belevičius, R.; Šešok, D. 2008. Global Optimization of Grillages Using Genetic Algorithms, *Mechanika* 6(74): 38–44.

Belevičius, R.; Ivanikovas, S.; Šešok, D.; Valentinavičius, S.; Žilinskas, J. 2011. Optimal Placement of Piles in Real Grillages: Experimental Comparison of Optimization Algorithms, *Information Technology and Control* 40(2): 123–132.

Bledsoe, W. W. 1961. *Lethally Dependent Genes Using Instant Selection*. Technical Report PRI 1, Panoramic Research Inc., Palo Alto, California, USA. Bledsoe, W. W. 1962a. *An Analysis of Genetic Populations*. Technical Report, Panoramic Research Inc., Palo Alto, California, USA.

Bledsoe, W. W. 1962b. *The Evolutionary Method in Hill Climbing: Convergence rates*. Technical Report, Panoramic Research Inc., Palo Alto, California, USA.

Blickle, T.; Thiele, L. 1995. A Mathematical Analysis of Tournament Selection, in *Proceedings of the Sixth International Conference on Genetic Algorithms*, 9–16.

Blum, C.; Roli, A. 2003. Metaheuristics in Combinatorial Optimization: Overview and conceptual comparison, *ACM Computing Surveys* 35(3): 268–308.

Bonabeau, E.; Dorigo, M.; Theraulaz, G. 1999. *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press USA, 1 ed. 320 p.

Brauers, W. K.; Zavadskas, E. K. 2010. *Is Robustness Really Robust? Robustness from the Point of View of Statistics and Econometrics with an Application for Multi-Objective*

Optimization. In W. K. Brauers; E. K. Zavadskas, *Multiple Criteria Decision Aiding* (17–42). New York: Nova Science Publishers.

Bremermann, H. J. 1962. Optimization Through Evolution and Recombination, in *Self-Organizing Systems*, Washington 93–106.

Brindle, A. 1981. *Genetic Algorithms for Function Optimization*. PhD thesis, University of Alberta, Department of Computer Science, Edmonton.

Bui, L. T.; Alam, S. 2008. *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*. IGI Global, 1 ed. 496 p.

Chan, F. T. S.; Tiwari, M. K. 2007. *Swarm Intelligence – Focus on Ant and Particle Swarm Optimization*. I-Tech Education and Publishing. 532 p.

Chan, C.M.; Zhang, L.M.; Ng, J.T.M. 2009. Optimization of Pile Groups Using Hybrid Genetic Algorithms, *Journal of Geotechnical and Geoenvironmental Engineering* 135 (4): 497-505.

Chapman, C.; Jakiela, M. 1996. Genetic Algorithm-Based Structural Topology Design With Compliance and Topology Simplification Considerations, *ASME Journal of Mechanical Design* 118(1): 89-98.

Chapman, C.; Saitou, K.; Jakiela, M. 1993. Genetic Algorithms as an Approach to Configuration and Topology Design, *Advances in Design Automation*. ASME 65: 485–498.

Charnes, A.; Cooper, W. W. 1961. *Management Models and Industrial Applications of Linear Programming*. New York: John Wiley & Sons. 859 p.

Chankong, V.; Haimes, Y. Y. 1983. *Multiobjective Decision Making: Theory and Methodology*. Dover Publications. 406 p.

Cheng, Y. M.; Li, L.; Chi, S. C. 2007. Performance Studies on Six Heuristic Global Optimization Methods in the Location of Critical Slip Surface, *Computers and Geotechnics* 34(6): 462–484.

Chinchuluun, A.; Pardalos, P. M.; Migdalas, A.; Pitsoulis, L. (editors). 2008. *Pareto Optimality, Game Theory and Equilibria*. Springer Optimization and Its Applications. Vol. 17. Springer. 866 p.

Christensen, P. W.; Klarbring, A. 2008. *An Introduction to Structural Optimization (Solid Mechanics and Its Applications)*. Springer, 1st edition. 224 p.

Coello Coello, C. A. 1996. *Multi-Objective Engineering Design Optimization Using Genetic Algorithms*. Phd thesis. Department of Computer Science, Tulane University.

Coello Coello, C. A. 2000. An Updated Survey of GA-Based Multiobjective Optimization Techniques, *ACM Computing Surveys* 32(2): 109–143.

Coello Coello, C. A.; Van Veldhuizen, D. A.; Lamont G. B. 2002. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers. 800 p.

Collins, J. J.; Eaton, M. 1997. Genocodes for Genetic Algorithms, in *Proceedings of the 2nd International Mendel Conference on Genetic Algorithms, MENDEL 1997*. Brno University of Technology, Brno, Czech Republic. 23–30.

Čaplinskas, A. 1996. *Programų sistemų inžinerijos pagrindai*. I dalis. Matematikos ir informatikos institutas. Vilnius. 294 p.

Čiegis, R.; Baravykaitė, M.; Belevičius, R. 2006. Parallel Global Optimization of Foundation Schemes in Civil Engineering, in *Applied Parallel Computing: 7th International Conference, PARA 2004*. Lyngby, Denmark 20–23 June, 2004: revised selected papers. *Lecture Notes in Computer Science*. Vol. 3732 (2006): 305–313.

Čyras, A. 1983. *Mathematical Models for the Analysis and Optimization of Elastoplastic Structures*. New York, John Wiley. 121 p.

Darwin, C. 1859. *On the Origin of Species*. John Murray, sixth edition. 432 p.

Das, I.; Dennis, J. E. 1997. A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems, *Structural and Multidisciplinary Optimization* 14(1): 63–69.

Davis, L. 1989. Adapting Operator Probabilities in Genetic Algorithms, in *Proceedings of the Third International Conference on Genetic Algorithms*, 61–69.

Deb, K. 1995. *Optimization for Engineering Design: Algorithms and Examples*. New Delhi: Prentice-Hall. 396 p.

Deb, K.; Goyal, M. 1996. A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and Informatics* 26(4): 30–45.

Deb, K.; Goyal, M. 1998. A Robust Optimization Procedure for Mechanical Component Design Based on Genetic Adaptive Search, *Transactions of the ASME: Journal of Mechanical Design* 120(2): 162–164.

Deb, K. 1999. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design, in *Proc. of Evolutionary Algorithms in Engineering and Computer Science (EUROGEN-99)*, 135–161.

Deb, K. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. Interscience Series in Systems and Optimization. New York: John Wiley & Sons, Ltd. 518 p.

Deb, K.; Beyer, H.-G. 2001. Self-Adaptive Genetic Algorithm with Simulated Binary Crossover, *Evolutionary Computation* 9(2): 197–221.

De Jong, K. A. 1975. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, Computer and Communication Sciences, University of Michigan.

De Jong, K. A. 1980. Adaptive System Design: A Genetic Approach, *IEEE Transactions on Systems, Man and Cybernetics*, 10(9): 566–574.

Dinger, R. H. 1998. Engineering Design Optimization With Genetic Algorithms, in *Proceedings of Conference Northcon/98*, 114–119.

Dzemyda, G.; Petkus, T. 2001. Application of Computer Network to Solve the Complex Applied Multiple Criteria Optimization Problems, *Infomatica* 12(1): 45–60.

Edgeworth, F. 1881. *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*. London: P. Keagan.

Engelbrecht, A. P. 2006. *Fundamentals of Computational Swarm Intelligence*. Wiley and Sons, Ltd., England. 672 p.

Erfani, T.; Utyuzhnikov, S. V. 2011. Directed Search Domain: a Method for Even Generation of Pareto Frontier in Multiobjective Optimization, *Engineering Optimization* 43(5): 467–484.

Filatovas, E. 2012. *Daugiakriterinių optimizavimo uždavinių sprendimas interaktyviuoju būdu*. Daktaro disertacija. (VU) Technika. ISBN 978-9955-28-273-0

Floudas, C. A. 1999. *Deterministic Global Optimization: Theory, Algorithms and Applications*. Kluwer, Dordrecht, 760 p.

Fonseca, C. M.; Fleming, P. J. 1993. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion, and Generalization, in *Proceedings of the Fifth International Conference on Genetic Algorithms*, 416–423.

Fonseca, C. M.; Fleming, P. J. 1995. An Overview of Evolutionary Algorithms in Multiobjective Optimization, *Evolutionary Computation* 3(1): 1–16.

Fonseca, C. M.; Fleming, P. J. 1998. Multiobjective Optimization and Multiple Constraint Handling With Evolutionary Algorithms – Part I: A unified formulation, *IEEE Transactions on Systems, Man and Cybernetics*, Part A: Systems and Humans, 28(1): 26–37.

Fonseca, J. R.; Friswell, M. I.; Lees, A. W. 2007. Efficient Robust Design via Monte Carlo Sample Reweighting, *International Journal for Numerical Methods in Engineering* 69(11): 2279–2301.

Frantz, D. R. 1972. *Nonlinearities in Genetic Adaptive Search*. PhD thesis, The University of Michigan, Ann Arbor, MI, USA.

Fraser, A. S. 1957. Simulation of Genetic Systems by Automatic Digital Computers, *Australian Journal of Biological Science* 10: 484–491.

Friedberg, R. M. 1958. A Learning Machine: Part I, *IBM Journal of Research and Development* 2: 2–13.

Galperin, E. A. 1997. Pareto Analysis vis-a-vis Balance Space Approach in Multiobjective Global Optimization, *Journal of Optimization Theory and Applications* 93(3): 533–545.

Gen, M. 1997. *Genetic Algorithms and Engineering Design (Engineering Design and Automation)*. Wiley-Interscience. 432 p.

Glover, F.; Kochenberger, G. A. (editors). 2003. *Handbook of Metaheuristics*. Springer, New York, USA. 556 p.

- Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley. New-York. 432 p.
- Goldberg, D. E.; Deb, K. 1990. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms, in *Proceedings of Foundations of Genetic Algorithms*, 69–93.
- Goldberg, D.; Samtani, M. 1986. Engineering Optimization via Genetic Algorithms, in *9th Conference of Electronic Computation*, New York. 471–482.
- Gonzalez, T. F. (editor). 2007. *Handbook of Approximation Algorithms and Metaheuristics*. Tailor & Francis Group. 1432 p.
- Groenwold, A. A.; Hindley, M. P. 2002. Competing Parallel Algorithms in Structural Optimization, *Structural and Multidisciplinary Optimization*, 24(5): 343–350.
- Guo, P.; Wang, X.; Han, Y. 2011. A Hybrid Genetic Algorithm for Structural Optimization with Discrete Variables, in *International Conference on Internet Computing & Information Services (ICICIS-2011)*, 223–226.
- Hacioglu, A.; Ozkol, I. 2005. Inverse Airfoil Design by Using an Accelerated Genetic Algorithm via Distribution Strategies, *Inverse Problems in Science and Engineering*, 13 (6): 563–579.
- Haimes, Y. Y.; Lasdon, L. S.; Wismer, D. A. 1971. On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization, *IEEE Transactions on Systems, Man and Cybernetics*, 296–297.
- Hajela, P. 1992. Stochastic Search in Structural Optimization: Genetic Algorithms and Simulated Annealing, in *Structural Optimization: Status and Promise*, 150: 611–637.
- Hajela, P.; Lee, E.; Cho, H. 1998. Genetic Algorithms in Topologic Design of Grillage Structures, *Computer-Aided Civil and Infrastructure Engineering* 13: 13–22.
- Hinterding, R.; Michalewicz, Z.; Peachey, T. C. 1996. Self-Adaptive Genetic Algorithm for Numeric Functions, in *Parallel Problem Solving from Nature, PPSN IV*, 420–429.
- Holland, J. H. 1967. *Nonlinear Environments Permitting Efficient Adaptation*. Volume II of Computer and Information Sciences. Academic Press, New York, USA.
- Holland, J. H. 1969. Adaptive Plans Optimal for Payoff-only Environments, in *Proceedings of the Second Hawaii International Conference on System Sciences*, Periodicals, 917–920.
- Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 211 p.
- Horn, J.; Nafploitis, N.; Goldberg, D. E. 1994. A Niche Pareto Genetic Algorithm for Multi-objective Optimization, in *Proceedings of the First IEEE Conference on Evolutionary Computation*, 82–87.
- Horst, R.; Pardalos, P. M. (editors). 1994. *Handbook of Global Optimization*. Springer, 880 p.

Huang, B.; Du, X. 2006. A Robust Design Method Using Variable Transformation and Gauss-Hermite Integration, *International Journal for Numerical Methods in Engineering* 66(12): 1841–1858.

Huang, H. Z.; Tian, Z. G.; Zuo, M. J. 2005. Intelligent Interactive Multiobjective Optimization Method and Its Application to Reliability Optimization, *IIE Transactions*, 37(11): 983–993.

Hwang, C. L.; Masud, A. S. 1979. *Multiple Objective Decision Making – Methods and Applications*. Berlin: Springer Verlag.

Jakiela, M. J.; Chapman, C.; Duda, J.; Adewuya, A.; Saitou, K. 2000. Continuum Structural Topology Design with Genetic Algorithms, *Computer Methods in Applied Mechanics and Engineering* 186(2-9): 339–356.

Kalanta, S. 2003. *Taikomosios optimizacijos pagrindai*. Vilnius: Technika. 336 p.

Keeney, R. L.; Raiffa, H. D. 1976. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Chichester: John Wiley & Sons Inc. 569 p.

Kearfott, R. B. 1996. *Rigorous Global Search: Continuous Problems*. Kluwer, Dordrecht, 284 p.

Kennedy, J.; Eberhart, R. C.; Shi, Y. 2001. *Swarm Intelligence: Collective, Adaptive*. Morgan Kaufmann Publishers. ISBN 1-55860-595-9.

Kennedy, S.A. 1993. Five Ways to a Smarter Genetic Algorithm, *AI Expert* 8(12): 35–38.

Kim, K.N.; Lee, S.-H.; Kim, K.-S.; Chung, C.-K.; Kim, M.M.; Lee, H.S. 2001. Optimal Pile Arrangement for Minimizing Differential Settlements in Piled Raft Foundations, *Computers and Geotechnics* 28(4): 235–253.

Kim, Y.; Gotoh, K.; Kim, K.; Toyosada, M. 2005. Optimum Grillage Structure Design Under a Worst Point Load Using Real-Coded Micro-Genetic Algorithm, in *Proceedings of the 15th International Offshore and Polar Engineering Conference*, Seoul, Korea, 730–735.

Kim, Y.; Kim, B.-L.; Park, J.-S. 2009. Pareto Optimisation of Grillage System with Multi-Objectives, *International Journal of Modelling, Identification and Control* 8(3): 213–221.

Klamroth, K.; Miettinen, K. 2008. Integrating Approximation and Interactive Decision Making in Multicriteria Optimization, *Operations Research*, 56(1): 222–234.

Koopmans, T. C. 1951. Analysis of Production as an Efficient Combination of Activities, *Activity Analysis of Production and Allocation* 33–97.

Lančinskas, A.; Ūilinskas, J.; Ortigosa, P. M. 2011. Local Optimization in Global Multi-Objective Optimization Algorithms, in *Proceedings of the 2011 Third World Congress on Nature and Biologically Inspired Computing (NaBIC)*, Spain, Salamanca: IEEE. 323–328.

- Langdon, W. B.; Poli, R. 2002. *Foundations of Genetic Programming*. Springer, 1st edition, 260 p.
- Lee, S. 1972. *Goal Programming for Decision Analysis*. Philadelphia: Auerbach Publications. 387 p.
- Lee, S.; Soak, S.; Kim, K.; Park, H.; Jeon, M. 2008. Statistical Properties Analysis of Real World Tournament Selection in Genetic Algorithms, *Applied Intelligence* 28(2): 195–205.
- Leiva, J. P.; Watson, B. C.; Kosaka, I. 2007. A Comparative Study of Topology and Topometry Structural Optimization Methods Within the Genesis Software, in *6th German LS-Dyna Forum*, Frankenthal, Germany, J-I: 23–32.
- Lemonge, A.; Barbosa, H. 2000. A Genetic Algorithm for Shape, Sizing and Configuration Optimization of Framed Structures, in *Proceedings of 21st Iberian Latin American Congress on Computational Methods in Engineering in CD-ROM Proc. of the 9th Conference in Electronic Computation*, Rio de Janeiro.
- Lemonge, A.; Barbosa, H.; Fonseca, L. 2003. A Genetic Algorithm for the Design of Space Framed Structures, in *XXIV CILAMCE – Iberian Latin-American Congress on Computational Methods in Engineering*, Ouro Preto, Brasil.
- Li, W.; Huang, Y. 2012. A Distributed Parallel Genetic Algorithm Oriented Adaptive Migration Strategy, in *Proceedings of 80th International Conference on Natural Computation (ICNC)*, Chongqing, 592–595.
- Liu, J. S. 2002. *Monte Carlo Strategies in Scientific Computing*. Springer. ISBN: 0-3879-5230-6. 346 p.
- Lobo, F. G.; Lima, C. F. 2007. Adaptive Population Sizing Schemes in Genetic Algorithms, *Studies in Computational Intelligence* 54: 185-204.
- Luchian, H.; Gheorghies, O. 2003. Integrated-Adaptive Genetic Algorithms, in *Proceeding of 7th European Conference on Artificial Life (ECAL 2003)*, Dortmund, Germany, September 14-17, 2003.
- Luque, M.; Ruiz, F.; Miettinen, K. 2011. Global Formulation for Interactive Multiobjective Optimization, *OR Spectrum*, 33(1): 27–48.
- Mackeown, P. K. 2001. *Stochastic Simulation in Physics*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. ISBN: 9-8130-8326-3.
- Marcelin, L. J. 2011. Cognitive Optimization of Mechanical Structures, *International Review of Mechanical Engineering (IREME)* 5(1): 88–91.
- McClure, M. 2001. *Pareto, Economics and Society: The Mechanical Analogy*. London: Routledge, 222 p.
- Michalewicz, Z. 1995. Genetic Algorithms, Numerical Optimization, and Constraints, in *Proceedings of the Sixth International Conference on Genetic Algorithms*, 151–158.
- Michalewicz, Z.; Fogel, D. B. 2004. *How to Solve It: Modern Heuristics*. Springer, second, revised and extended edition. ISBN: 978-3-54022-494-5. 554 p.

- Michell, A. G. M. 1904. The Limits of Economy of Material in Frame-Structures, *Philosophical Magazine* 8(47): 589–597.
- Miettinen, K.; Makela, M. M. 2006. Synchronous Approach in Interactive Multiobjective Optimization, *European Journal Of Operational Research* 170(3): 909–922.
- Miller, B. L.; Goldberg, D. E. 1996. Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise, *Evolutionary Computation*, 4(2): 113–131.
- Min, J.; Yi-min, C. 2007. GA Application Strategy of Multi-Objective Optimization, *Journal of Lanzhou University of Technology* 33(3): 92–94.
- Mockus, J. B.; Mockus, L. J. 1991. Bayesian Approach to Global Optimization and Application to Multiobjective and Constrained Problems, *Journal of Optimization Theory and Applications* 70(1): 157–172.
- Mockus, J.; Eddy, W.; Reklaitis, G. 1996. *Bayesian Heuristic Approach to Discrete and Global Optimization: Algorithms, Visualization, Software, and Applications (Nonconvex Optimization and Its Applications)*. Springer, 416 p.
- Mockus, J. 2000. *A Set of Examples of Global and Discrete Optimization: Applications of Bayesian Heuristic Approach (Applied Optimization)*. Springer. 340 p.
- Mockus, J. 2011. On the Pareto Optimality in the Context of Lipschitzian Optimization, *Informatica* 22(4): 521–536.
- Mohammadzadeh, A.; Etemadde, N. 2011. Optimized Positioning of Structure Supports with PSO for Minimizing the Bending Moment, *International Review of Mechanical Engineering (IREME)* 5(3): 422–425.
- Neumaier, A. 2004. Complete Search in Continuous Global Optimization and Constraint Satisfaction, *Acta Numerica (A. Iserles, ed.)*, Cambridge University Press, 271–369.
- Neumaier, A. 2006. Global Optimization and Constraint Satisfaction, in *Proceedings of GICOLAG Workshop (of the Research Project Global Optimization, Integrating Convexity, Optimization, Logic Programming and Computational Algebraic Geometry)*.
- Osborne, M. J.; Rubinstein, A. 1994. *A Course in Game Theory*. The MIT Press. ISBN: 0262650401. 352 p.
- Onate, E. 2009. *Structural Analysis with the Finite Element Method. Linear Statics. Vol.1: Basis and Solids*. Artes Graficas Torres S. L. ISBN: 978-1-4020-8732-5 (print), ISBN: 978-1-4020-8733-2 (el. version). 446 p.
- Pareto, V. F. 1896. *Cours d'Economie Politique*. F. Rouge, Lausanne/Paris, France. Vol. 2.
- Paul, D. 1992. *Fitness: Historical Perspectives*. In Evelyn Fox Keller and Elisabeth A. Lloyd, editors, *Keywords in Evolutionary Biology*, 112–114. Harvard University Press, Cambridge, MA, USA.

- Pearl, J. 1984. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. The Addison-Wesley Series in Artificial Intelligence. Addison-Wesley Pub (Sd). ISBN: 978-0-20105-594-8. 382 p.
- Pezeshk, S.; Camp, C. V. 2002. State of the Art on the Use of Genetic Algorithms in Design of Steel Structures, in *Recent Advances in Optimal Structural Design*, S. A. Burns, ed., American Society of Civil Engineers.
- Power, D.; Ryan, C.; Azad, R. M A. 2005. Promoting Diversity Using Migration Strategies in Distributed Genetic Algorithms, in *Proceedings of International Conference on Evolutionary Computation, In IEEE Congress, 2*: 1831–1838.
- Puiša, R. 2005. *The Adaptive Stochastic Algorithms for CAD-based Structure Optimization of Mechanical Parts*. Doctoral dissertation, Vilnius, p. 129–144.
- Rao. S. S. 2010. *The Finite Element Method in Engineering*. Butterworth-Heinemann, 5th edition, 726 p.
- Rayward-Smith, V. J.; Osman, I. H.; Reeves, C. R.; Smith, G. D. (editors). 1996. *Modern Heuristic Search Methods*. Wiley. ISBN: 978-0-47196-280-9. 294 p.
- Reeves. C. R. 2002. *Genetic Algorithms – Principles and Perspectives: A Guide to GA Theory (Operations Research/Computer Science Interfaces Series)*. Springer, 1st edition, 344 p.
- Reklaitis, G. V.; Ravindran, A.; Ragsdell, K. M. 2006. *Engineering Optimization Methods and Applications*. Ed.2. New York, John Wiley & Sons, p. 681.
- Richardson, J. T.; Palmer, M. R.; Liepins, G. E.; Hilliard, M. R. 1989. Some Guidelines for Genetic Algorithms with Penalty Functions, in *Proceedings of the Third International Conference on Genetic Algorithms ICGA*, 191–197.
- Riolo, R.; Worzel, B. 2003. *Genetic Programming Theory and Practice (Genetic Programming)*. Springer, 1st edition, 336 p.
- Robbins, H.; Monro, S. 1951. A Stochastic Approximation Method, *Annals of Mathematical Statistics*, 22(3): 400–407.
- Robert, C. P.; Casella, G. 2004. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag. Second ed. 645 p.
- Rosenberg, R. S. 1967. *Simulation of Genetic Populations With Biochemical Properties*. PhD thesis, The University of Michigan, College of Literature, Science, and the Arts, Computer and Communication Sciences Department, Ann Arbor, MI, USA.
- Rusakevičius, D.; Belevičius, R. 2002. Optimization of Laminated Bending Plates, *Journal Civil Engineering and Management* 8(3): 143–152.
- Saka, M. P. 1998. Optimum Design of Steel Grillage Systems Using Genetic Algorithms, *Computer-Aided Civil and Infrastructure Engineering* 13: 233–238.
- Saka, M. P.; Daloglu, A.; Malhas, F. 2000. Optimum Spacing Design of Grillage Systems Using a Genetic Algorithm, *Advances in Engineering Software* 31(11): 863–873.

Sanchez-Caballero, S.; Selles, M. A.; Pla-Ferrando, R.; Martinez Sanz A. V.; Peydro M. A. 2012. Recent Advances in Structural Optimization, *Annals of the Oradea University. Fascicle of Management and Technological Engineering* 11(1): 118–127.

Saravanan, N.; Fogel, D. B.; Nelson, K. M. 1995. A Comparison of Methods for Self-Adaptation in Evolutionary Algorithms, *BioSystems* 36: 157–166.

Schaffer, J. D. 1984. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. Unpublished doctoral thesis. Nashville: Vanderbilt University.

Schaffer, J. D.; Morishima, A. 1987. An Adaptive Crossover Distribution Mechanism for Genetic Algorithms, in *Proceedings of Second International Conference on Genetic Algorithms*, 36–40.

Schwefel, H. P. 1993. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc. 444 p.

Shaefer, C. G. 1987. The ARGOT strategy: Adaptive Representation Genetic Optimizer Technique, in *Proceedings of the Second International Conference on Genetic Algorithms*, 50–58.

Al-Shihri, M. A. 2010. Structural Optimization Using a Novel Genetic Algorithm for Rapid Convergence, *International Journal of Civil and Structural Engineering* 1(2): 123–138.

Sober, E. 2001. *The Two Faces of Fitness*. In Rama Shankar Singh, Costas B. Krimbas, Diane B. Paul, and John Beatty, editors, *Thinking about Evolution: Historical, Philosophical, and Political Perspectives*, chapter 15, pages 309–321. Cambridge University Press.

Spencer, H. 1864. *The Principles of Biology*. London & Edinburgh: Williams and Norgate, first edition, volume 1. 492 p.

Spyrakos, C.; Raftoyiannis, J. 1997. *Linear and Nonlinear Finite Element Analysis in Engineering Practice*. Algor Publishing Division. Pittsburgh. 497 p.

Srinivas, N.; Deb, K. 1994. Multi-Objective Function Optimization Using Non-Dominated Sorting Genetic Algorithms, *Evolutionary Computation* 2(3): 221–248.

Srinivas, M.; Patnaik, L. M. 1994. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms, *IEEE Transactions on Systems, Man and Cybernetics*. 24(4): 656–667.

Steuer, R. E. 1986. *Multiple Criteria Optimization: Theory, Computation, and Application*. New York, John Wiley & Sons, 546 p.

Šešok, D. 2007. Genetiniai algoritmai mechaninėms struktūroms optimizuoti, iš *10-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“*, įvykusios Vilniuje 2007 m. balandžio 12 d., pranešimų medžiaga. Vilnius: Technika, p. 222–227.

Šešok, D. 2008. *Santvarų topologijos optimizavimas genetiniais algoritmais*. Daktaro disertacija. Vilnius: Technika. ISBN 978-9955-28-273-0.

Šešok, D.; Belevičius, R. 2009. Global Optimization of Grillages with a Hybrid Genetic Algorithm, *Engineering Structures and Technologies* 1(2): 80–88. (in Lithuanian)

Šešok, D.; Mockus, J.; Belevičius, R.; Kačeniauskas, A. 2010. Global Optimization of Grillages Using Simulated Annealing and High Performance Computing, *Journal of Civil Engineering and Management* 16(1): 95–101.

Šmigelskas, M.; Vitkūnas R. 2007. Lietuvoje eksploatuojamų lengvųjų automobilių būklė ir avaringumas, iš *10-osios Lietuvos jaunųjų mokslininkų konferencijos „Mokslas – Lietuvos ateitis“*, įvykusios Vilniuje 2007 m. gegužės 3 d., pranešimų medžiaga. Vilnius: Technika, p. 362–366.

Tai, K.; Wang, S. Y.; Akhtar, S.; Prasad, J. 2003. Structural Topology Optimization Using a Genetic Algorithm and a Morphological Representation of Geometry, in *Proceedings of the Singapore-MIT Alliance (SMA) Annual Symposium*, 17-18 January 2002, Singapore.

Tawarmalani, M.; Sahinidis, N. V. 2002. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, Dordrecht, Vol. 65 in "Nonconvex Optimization And Its Applications" series, 504 p.

Tazawa, I.; Koakutsu, S.; Hirata, H. 1996. An Immunity Based Genetic Algorithm and Its Application to the VLSI Floorplan Design Problem, in *Proceedings of IEEE International Conference on Evolutionary Computations*, 417–421.

Törn, A.; Žilinskas, A. 1989. *Global Optimization*. New York: Springer-Verlag, 255 p.

Venanzi, I.; Materazzi, A. L. 2007. Multi-Objective Optimization of Wind-Excited Structures, *Engineering Structures* 29: 983–990.

Wang, Y. J.; Zhang, J. S. 2007. An Efficient Algorithm for Large Scale Global Optimization of Continuous Functions, *Journal of Computational and Applied Mathematics*, 206(2): 1015–1026.

Weise, T. 2009. *Global Optimization Algorithms – Theory and Application*. University of Kassel, Distributed Systems Group. 820 p.

Wetzel, A. 1983. *Evaluation of the Effectiveness of Genetic Algorithms in Combinatorial Optimization*. PhD thesis, University of Pittsburgh, Pittsburgh, PA.

Whitley, L. D. 1989. The GENITOR Algorithm and Selective Pressure: Why Rankbased Allocation of Reproductive Trials Is Best, in *Proceedings of the 3rd International Conference on Genetic Algorithms*, 116–121.

Winkler, H. 1920. *Verbreitung und Ursache der Parthenogenesis im Pflanzen- und Tierreiche*. Verlag Gustav Fischer, Jena.

Yao, C.; Chen, H.; Yu, J.; Li, J. 2007. Application of Distributed Genetic Algorithm Based on Migration Strategy in Image Segmentation, in *Proceedings of the 3rd International Conference on Natural Computation*, 218–222.

- Yeo, M. F.; Agyei, E. O. 1998. Optimising Engineering Problems Using Genetic Algorithms, *Engineering Computations* 15(2): 268–280.
- Zadeh, L. 1963. Optimality and Non-Scalar-Valued Performance Criteria, *IEEE Transactions on Automatic Control* 8(1): 59–60.
- Zeleny, M. 1974. *Linear Multiobjective Programming*. Berlin: Springer-Verlag. 220 p.
- Zheng, D.; Gen, M.; Cheng, R. 1999. Multiobjective Optimization Using Genetic Algorithms. *Journal of Engineering Valuation and Cost Analysis* 2(4): 303–310.
- Zhou, K.; Li, X. 2012. Optimization of Grillage-like Continuum by Triangle Plate Element, *The Open Civil Engineering Journal* 6: 8–14.
- Zienkiewicz, O. C.; Taylor, R. C. 2005. *The Finite Element Method for Solid and Structural Mechanics*. 6th Edition, Elsevier Butterworth-Heinemann, 736 p.
- Zuo, Z.; Xie, Y.; Huang, X. 2009. Combining Genetic Algorithms with BESO for Topology Optimization, *Structural and Multidisciplinary Optimization* 38(5): 511–523.
- Žilinskas, A.; Žilinskas, J. 2002. Global Optimization Based on a Statistical Model and Simplicial Partitioning, *International Journal of Computers and Mathematics with Applications* 44(7): 957–967.
- Žilinskas, A.; Klause, J. 2002. Subdivision, Sampling, and Initialization Strategies for Simplicial Branch and Bound in Global Optimization, *International Journal of Computers and Mathematics with Applications* 44(7): 943–955.

Autoriaus publikacijų disertacijos tema sąrašas

Straipsniai recenzuojamuose mokslo žurnaluose

Mockus, J.; Belevičius, R.; Šešok, D.; Kaunas, J.; Mačiūnas, D. 2012. On Bayesian Approach to Grillage Optimization, *Information Technology and Control* 41(4): 332–339. ISSN 1392-124X. (Thomson ISI Web of Science).

Mačiūnas, D.; Belevičius, R. 2012. Multi-Objective Optimization of Grillages Using Adaptive Genetic Algorithm, *International Review of Mechanical Engineering* 6(3): 432–439. ISSN 1970-8734. (Scopus).

Mačiūnas, D.; Kaunas, J.; Šešok, D. 2012a. Optimization of Grillages With Genetic Algorithms Integrating MatLab and Fortran Environments, *Mokslas – Lietuvos ateitis = Science – Future of Lithuania: Mechanika, medžiagų inžinerija, pramonės inžinerija ir vadyba* 4(6): 564–568. ISSN 2029-2341 (print), ISSN 2029-2252 (online). (Index Copernicus).

Belevičius, R.; Mačiūnas, D.; Šešok, D. 2011. The Minimization of Moments and Reactive Forces in Grillages With Genetic Algorithm, *Engineering Structures and Technologies / Statybinės konstrukcijos ir technologijos* 3(2): 56–63 (in Lithuanian). ISSN 2029-2317 (print), ISSN 2029-2325 (online). (Index Copernicus).

Mačiūnas, D.; Belevičius, R.; Kaunas, J. 2011. Multi-Objective Optimization of Grillages Applying the Genetic Algorithm, *Science – Future of Lithuania / Mokslas – Lietuvos*

ateitis: mechanika, medžiagų inžinerija, pramonės inžinerija ir vadyba 3(6): 47–52 (in Lithuanian). ISSN 2029-2341 (print), ISSN 2029-2252 (online). (Index Copernicus).

Straipsniai kituose leidiniuose

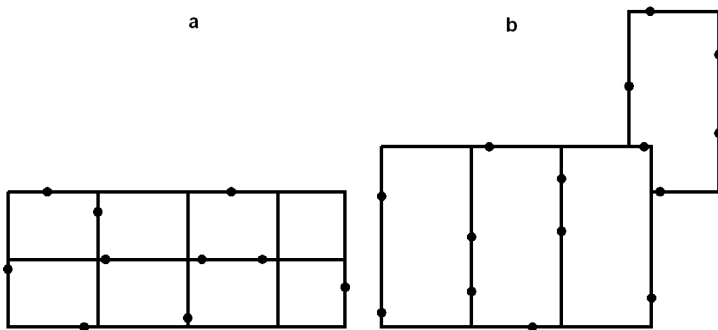
Mačiūnas, D.; Belevičius, R.; Šešok, D. 2012b. Multi-Objective Optimization of Grillages Using Genetic Algorithm With Parallel Strategy, in *Proceedings of the 1st Virtual International Conference on Advanced Research in Scientific Area (ARSA 2012)*, held in Zilina, Slovakia, on 3–7 December, 2012. EDIS-Publishing Institution of the University of Zilina, 1842–1848. ISSN 1338-9831. ISBN 978-80-554-0606-0. (Scopus).

Mačiūnas, D. 2012. Multi-Objective Approach to Grillage Optimization with Genetic Algorithm, in *Proceedings of the 17th International Conference „Mechanika-2012“*, held in Kaunas, Kaunas University of Technology, on 12–13 April, 2012. Kaunas: Technologija. 176–181. ISSN 1822-2951. (Academic Search Complete).

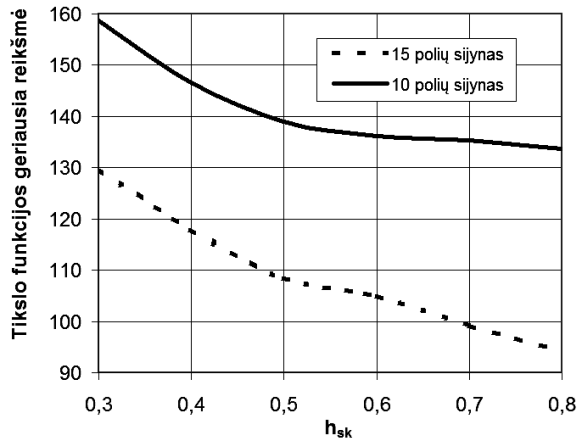
Priedai

A priedas. 10 polių ir 15 polių sijynai

Atlikus skaitinius bandymus, kai skerspjūvio matmenys įgyja reikšmes iš intervalų $h_{sk} \in [0,2;0,8]$ ir $b_{sk} \in [0,2;0,8]$, gauname, kad geriausią tikslo funkcijos (2.12) sprendinį (133,63 – 10 polių sijynui ir 94,32 – 15 polių sijynui).

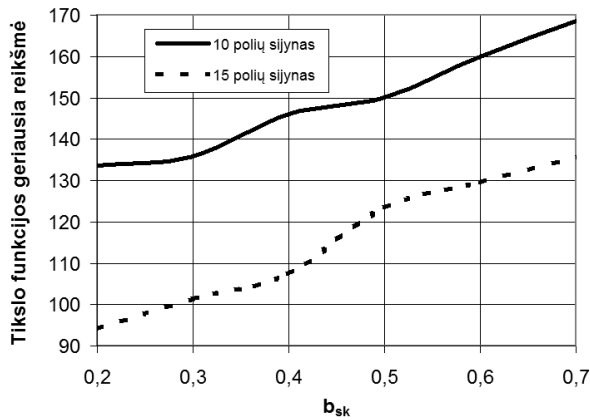


A.1 pav. Polių išdėstymo schemas: a) 10 polių sijynas, b) 15 polių sijynas
Fig. A.1. Pile placement schemes: a) 10 pile grillage, b) 15 pile grillage



A.2 pav. Tikslo funkcijos reikšmės, kai sijos skerspjūvio plotis pastovus ($b_{sk} = 0,2$), o aukštis kinta

Fig. A.2. Values of objective function, when beam cross-sectional width is constant ($b_{sk} = 0,2$) and height is variable



A.3 pav. Tikslo funkcijos reikšmės, kai sijos skerspjūvio aukštis pastovus ($h_{sk} = 0,8$), o plotis kinta

Fig. A.3. Values of objective function, when beam cross-sectional height is constant ($h_{sk} = 0,8$) and width is variable

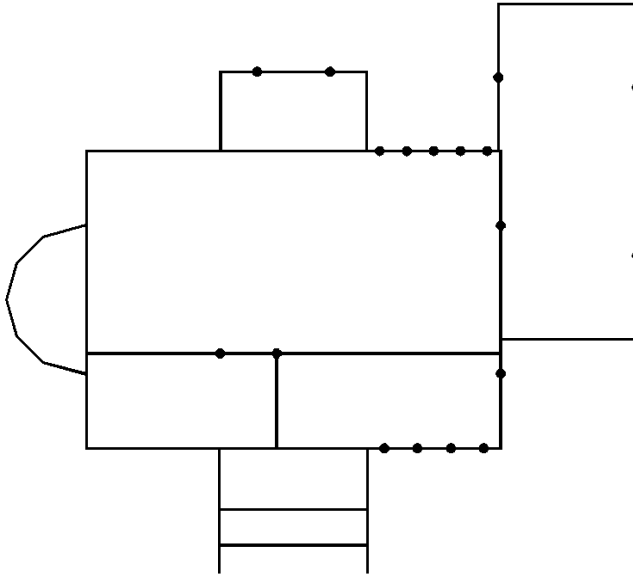
Šie sprendiniai gauti, kai sijos skerspjūvio aukščio matmuo įgyja didžiausią reikšmę ($h_{sk} = 0,8$), o sijos skerspjūvio pločio matmuo įgyja mažiausią reikšmę ($b_{sk} = 0,2$).

Polių išdėstymo schemas, atitinkančios geriausią tikslo funkcijos (2.12) sprendinį (133,63 – 10 polių sijynui ir 94,32 – 15 polių sijynui), yra pateiktos A.1 paveiksle.

Kai sijos skerspjūvio plotis pastovus ($b_{sk} = 0,2$), o aukštis h_{sk} didėja (A.2 pav.) – tuomet tikslo funkcijos reikšmė gerėja (šiuo atveju mažėja). Tai galima paaiškinti tuo, kad didėjant sijos skerspjūvio aukščiui h_{sk} mažėja didžiausia lenkimo momento sijoje reikšmė. Dėl tos pačios priežasties tikslo funkcijos reikšmė gerėja (šiuo atveju mažėja), kai sijos skerspjūvio plotis b_{sk} mažėja, o sijos skerspjūvio aukštis pastovus ($h_{sk} = 0,8$) (A.3 pav.).

B priedas. 18 polių sijynas

Atliktų tyrimų rezultatai atskleidė, kad 18 polių sijynui (B.1 pav.) tikslo funkcijos komponentų $f_1(x)$ ir $f_2(x)$ įtaka optimizavimo uždaviniui yra vienoda, kai svorinių koeficientų santykio reikšmė lygi 2,33 ($w_s \approx 2,33$). Kai svorinių koeficientų santykio reikšmė didesnė nei 2,33 ($w_s > 2,33$) (B.2 pav.), tai tikslo funkcijai (2.22) didesnę įtaką turi lenkimo momentas (t. y. komponentas $w_s \cdot f_2(x)$). Ir atvirkščiai, kai svorinių koeficientų santykio reikšmė mažesnė nei 2,33 ($w_s < 2,33$) – didesnę įtaką tikslo funkcijai (2.22) turi reakcija (t. y. komponentas $f_1(x)$).

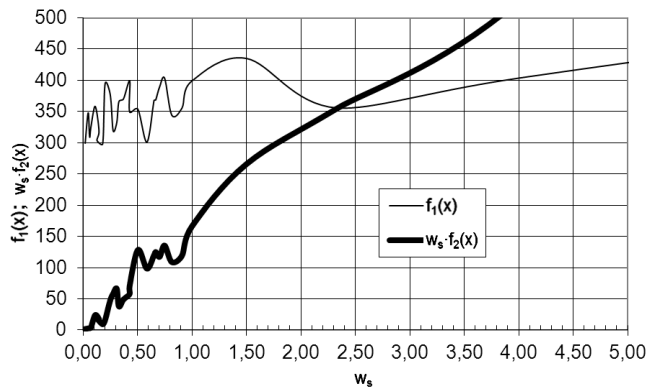


B.1 pav. 18 polių sijynas
Fig. B.1. 18 pile grillage

Tyrimų metu gauta priklausomybė (B.2 pav.) tarp dviejų tikslo funkcijos kriterijų (reakcijos ir lenkimo momento) reikšmių ir svorinių koeficientų santykio reikšmių. Skirtingos svorinių koeficientų santykio (w_s) reikšmės, atitinka skirtingas reakcijų ir lenkimo momentų reikšmes tikslo funkcijoje, t. y. skirtin-

gas sijynų schemas. Todėl projektuotojas gali pasirinkti tokias sijynų schemas, kuriose:

1. Reakcija ir lenkimo momentas turi vienodą įtaką optimizavimo uždaviniui ($w_s \approx 2,33$).
2. Reakcija turi didesnę įtaką optimizavimo uždaviniui ($w_s < 2,33$).
3. Lenkimo momentas turi didesnę įtaką optimizavimo uždaviniui ($w_s > 2,33$).

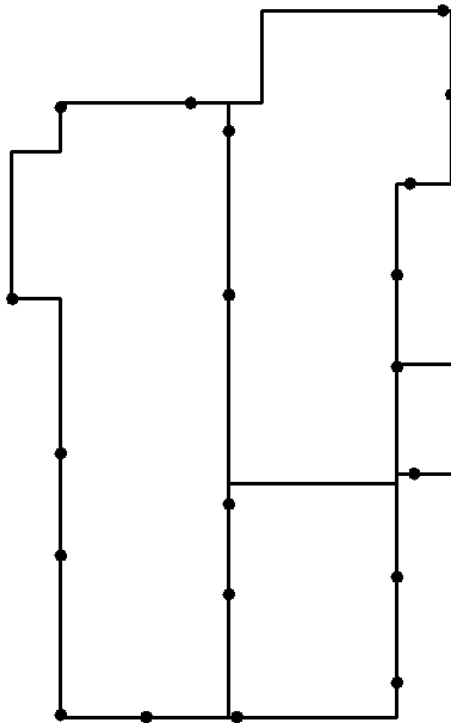


B.2 pav. Komponentų $f_1(x)$ ir $f_2(x)$ įtaka tikslo funkcijai (18 polių sijynas)

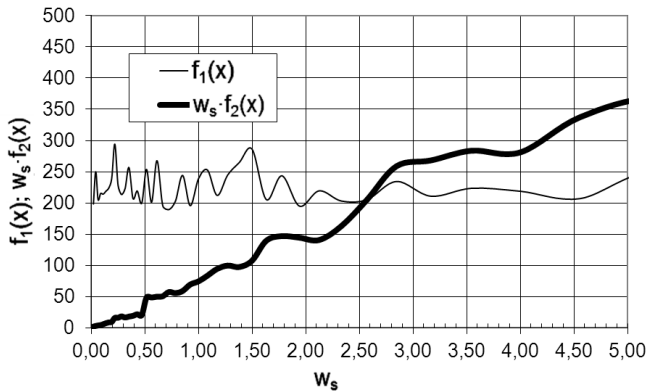
Fig. B.2. Contribution of components $f_1(x)$ and $f_2(x)$ to the objective function (18 pile grillage)

C priedas. 20 polių sijynas

Atliktų tyrimų rezultatai atskleidė, kad 20 polių sijynui (C.1 pav.) tikslo funkcijos komponentų $f_1(x)$ ir $f_2(x)$ įtaka optimizavimo uždaviniui yra vienoda, kai svorinių koeficientų santykio reikšmė lygi 2,57 ($w_s \approx 2,57$). Kai svorinių koeficientų santykio reikšmė didesnė nei 2,57 ($w_s > 2,57$) (C.2 pav.), tai tikslo funkcijai (2.22) didesnę įtaką turi lenkimo momentas (t. y. komponentas $w_s \cdot f_2(x)$). Ir atvirkščiai, kai svorinių koeficientų santykio reikšmė mažesnė nei 2,57 ($w_s < 2,57$) – didesnę įtaką tikslo funkcijai (2.22) turi reakcija (t. y. komponentas $f_1(x)$).



C.1 pav. 20 polių sijynas
Fig. C.1. 20 pile grillage



C.2 pav. Komponentų $f_1(x)$ ir $f_2(x)$ įtaka tikslo funkcijai (20 polių sijynas)
Fig. C.2. Contribution of components $f_1(x)$ and $f_2(x)$ to the objective function (20 pile grillage)

Tyrimų metu gauta priklausomybė (C.2 pav.) tarp dviejų tikslo funkcijos kriterijų (reakcijos ir lenkimo momento) reikšmių ir svorinių koeficientų santykio reikšmių. Skirtingos svorinių koeficientų santykio (w_s) reikšmės, atitinka skirtingas reakcijų ir lenkimo momentų reikšmes tikslo funkcijoje, t. y. skirtingas sijynų schemas. Todėl projektuotojas gali pasirinkti tokias sijynų schemas, kuriose:

1. Reakcija ir lenkimo momentas turi vienodą įtaką optimizavimo uždaviniui ($w_s \approx 2,57$).
2. Reakcija turi didesnę įtaką optimizavimo uždaviniui ($w_s < 2,57$).
3. Lenkimo momentas turi didesnę įtaką optimizavimo uždaviniui ($w_s > 2,57$).

Darius MAČIŪNAS

DAUGIAKRITERIS GLOBALUS SIJYNŲ OPTIMIZAVIMAS
GENETINIAIS ALGORITMAIS

Daktaro disertacija

Technologijos mokslai,
mechanikos inžinerija (09T)

MULTI-OBJECTIVE GLOBAL OPTIMIZATION OF
GRILLAGES USING GENETIC ALGORITHMS

Doctoral Dissertation

Technological Sciences,
Mechanical Engineering (09T)

2013 05 02. 10,25 sp. l. Tiražas 20 egz.
Vilniaus Gedimino technikos universiteto
leidykla „Technika“,
Saulėtekio al. 11, 10223 Vilnius,
<http://leidykla.vgtu.lt>
Spausdino UAB „Ciklonas“
J. Jasinskio g. 15, 01111 Vilnius