VILNIAUS UNIVERSITETAS

Ernestas Filatovas

DAUGIAKRITERINIŲ OPTIMIZAVIMO UŽDAVINIŲ SPRENDIMAS INTERAKTYVIUOJU BŪDU

Daktaro disertacija

Technologijos mokslai, informatikos inžinerija (07T)

Vilnius, 2012

Disertacija rengta 2006–2011 metais Vilniaus universiteto Matematikos ir informatikos institute.

Mokslinė vadovė

dr. Olga Kurasova (Vilniaus universiteto Matematikos ir informatikos institutas, technologijos mokslai, informatikos inžinerija – 07T).

Padėka

Nuoširdžiai dėkoju darbo vadovei dr. O. Kurasovai už atkaklų ir nuoseklų vadovavimą, vertingas mokslines konsultacijas, visokeriopą pagalbą ir kantrybę studijuojant doktorantūroje bei rengiant disertaciją. Dėkoju habil. dr. prof. G. Dzemydai ir doc. dr. T. Petkui už sugeneruotas idėjas, panaudotas šiame darbe, už patarimus, pasiūlymus ir bendradarbiavimą kuriant ir skelbiant šiame darbe pateiktus rezultatus.

Esu dėkingas disertacijos recenzentams habil. dr. prof. J. Mockui ir prof. dr. D. Navakauskui atidžiai perskaičiusiems disertaciją ir pateikusiems vertingų pastabų bei patarimų, padėjusių pagerinti šio darbo kokybę.

Ačiū Vilniaus universiteto Matematikos ir informatikos instituto Sistemų analizės skyriaus kolegoms už kritiką ir draugišką pagalbą, rengiant disertaciją.

Nuoširdžiai dėkoju savo artimiesiems ir draugams už jų paramą, moralinį palaikymą, kantrybę ir supratingumą.

Dėkoju Lietuvos valstybiniam studijų fondui už suteiktą finansinę paramą disertacijos rengimo metu.

Dėkoju visiems, kurie tiesiogiai ar netiesiogiai prisidėjo prie šio darbo.

Ernestas Filatovas

Reziumė

Praktikoje dažnai tenka spręsti sudėtingus daugiakriterinius optimizavimo uždavinius, kai kriterijai būna prieštaringi, o galutinis apsisprendimas priklauso nuo sprendimų priėmėjo. Kai sprendimų priėmėjas dalyvauja sprendimo procese interaktyviai, tai jis gali koreguoti prioritetus ir siekiamus tikslus uždavinio sprendimo eigoje, kas įgalina spęsti uždavinius, turinčius daug kriterijų ir apribojimų. Be to, sprendimo priėmėjui svarbu gauti sprendinius iš visos Pareto aibės. Interaktyviam uždavinių sprendimui būtina sprendimų paramos sistema, kurios grafinė sąsaja yra pritaikyta sprendžiamam uždaviniui. Šio darbo tyrimų sritis yra interaktyvus daugiakriterinių optimizavimo uždavinių sprendimas bei sprendimų paramos sistemos.

Disertacijoje nagrinėjant daugiakriterinio optimizavimo metodus, didesnis dėmesys skirtas metodams, užtikrinantiems gaunamų sprendinių tolygų pasiskirstymą Pareto aibėje bei interaktyviems metodams. Pasiūlytas ir ištirtas daugiakriterinių optimizavimo uždavinių sprendimo būdas, leidžiantis spręsti daugiakriterinius optimizavimo uždavinius interaktyviai ir užtikrinantis gaunamų sprendinių tolygų pasiskirstymą Pareto aibėje. Sukurta ir ištirta interaktyvi daugiakriterinių optimizavimo uždavinių sprendimų paramos sistemą, apjungianti pasiūlytą optimizavimo uždavinių sprendimo būdą, sprendimo proceso vizualizavimą ir jo lygiagretinimą. Taip pat pasiūlyta sprendimo strategija, pagal kurią sprendžiant daugiakriterinį optimizavimo uždavinį pasitelkiamas kompiuterių klasteris. Ši strategija eksperimentiškai ištirta ir palyginta su kitų autorių pasiūlyta strategija. Pagrįstas sukurtos strategijos pranašumas. Palyginta, kaip efektyviai patyręs ir nepatyrę sprendimų priėmėjai išnaudoja kompiuterių klasterį sprendžiant praktinį uždavinį. Eksperimentiškai ištirtas darbo su sukurtos sprendimų paramos sistemos apsimokymo laikas, būtinas sprendimų priėmėjui perprasti sprendžiamo uždavinio specifiką, siekiant greičiau rasti tinkamą sprendinį.

Abstract

In practice, optimization problems are often multiple criteria. The criteria are usually contradictory, so the final decision depends on a decision maker. When the problem is solved interactively, the decision maker can change his/her preferences in decision process. Moreover, it is important to obtain solutions from the whole Pareto front. A decision support system adapted to the specific of the problem is essential for solving multiple criteria optimization problems interactively. The objects of research are multiple criteria optimization problems, interactive methods for solving these problems, interactive decision support systems, and application of parallel computing in decision support systems.

Multiple criteria optimization methods are analyzed in the dissertation. The focus of attention is the methods for a uniform distribution of solutions on the Pareto front as well as the interactive methods. An interactive way for solving multicriteria optimization problems, which finds alternative solutions uniformly distributed on the Pareto front is proposed and investigated in this dissertation. An interactive decision support system which integrates the created interactive solving way, the decision process visualization and parallelization for multiple criteria optimization is developed. The solving strategies, when a multiple criteria optimization problem is solved interactively, using a computer cluster are developed and compared experimentally. The time required for a decision maker to learn to solve a multiple criteria optimization problem by the interactive decision support is investigated experimentally.

Žymėjimai

Simboliai

$A_{ij}(X)$	netiesinė funkcija, nusakanti i-tojo ingrediento j-tosios maistinės charakteristikos reikšmę
b_{k}	baudos koeficientas
β_i	interpoliacijos i-tasis svorinis koeficientas
D	leistinoji sritis
$f_i(X)$	i-tasis kriterijus
$\hat{f}_i(X)$	i-tasis normuotas kriterijus
$f_2'(X)$	maistinių charakteristikų leistinų normų pažeidimų suma
$\mathbf{F}(X) = (f_1(X), \dots, f_m(X))$	kriterijų tikslo funkcijų vektorius
$\mathbf{F}^{\mathrm{id}}(X) = (f_1^{\mathrm{id}}(X), \dots, f_m^{\mathrm{id}}(X))$	idealus vektorius
$\mathbf{F}^{\mathrm{u}}(X) = (f_1^{\mathrm{u}}(X), \dots, f_m^{\mathrm{u}}(X))$	utopinis vektorius
$\mathbf{F}^{\mathrm{n}}(X) = (f_1^{\mathrm{n}}(X), \dots, f_m^{\mathrm{n}}(X))$	nadir vektorius
$\hat{\mathbf{F}}(X) = (\hat{f}_1(X), \dots, \hat{f}_m(X))$	normuotas kriterijų tikslo funkcijų vektorius

φ	pasiskirstymo įvertis
h	lopo viršūnių skaičius
k	nelygybinių apribojimų skaičius
ψ_i	i-tojo ingrediento svorio vieneto kaina
K_t	jungtinio kriterijaus reikšmės laiko momentu t
m	tikslo funkcijų (kriterijų) skaičius
n	kintamųjų skaičius
n'	surastų nedominuojančių sprendinių skaičius
N^i	i-tosios lopo viršūnės koordinačių vektorius
p	lygybinių apribojimų skaičius
P	kompiuterių-darbininkų skaičius
P^j	tikėtino j-tojo sprendinio vietos vektorius
$r = (r_1, r_2, \dots, r_m)$	atramos taškas
$R_j^{\min}$	rekomenduojamas leistinas minimalus j-tosios maistinės charakteristikos kiekis pašare
$R_j^{\max}$	rekomenduojamas leistinas maksimalus j-tosios maistinės charakteristikos kiekis pašare
R^n	n-matė Euklidinė erdvė
s	svorinių koeficientų w_i reikšmių parinkimo žingsnis
t	laiko momentas
t_ε	duomenų siuntimo ir kitos laiko sąnaudos
t_f	užduoties formavimo trukmė
t_s	užduoties sprendimo trukmė
t_v	bendra vienakriterinio uždavinio sprendimo trukmė
ν	jungtinio kriterijaus reikšmių procentų vidurkis
$w = (w_1, w_2, \dots, w_m)$	svorinių koeficientų rinkinys

w_i	i-tasis svorinis koeficientas
x_i	i-tasis kintamasis
$x_i^{\min}$	minimali i-tojo ingrediento dalis pašare
$x_i^{\max}$	maksimali i-tojo ingrediento dalis pašare
$X = (x_1, x_2, \ldots, x_n)$	kintamųjų vektorius
X_0	pradinių reikšmių kintamųjų vektorius
$\mathbf{Z}$	kriterijų reikšmių aibė

Santrumpos

ASKS	adaptyvus svertinės kriterijų sumos metodas (angl. *Adaptive Weighted Sum Method, AWS*)
NBI	angl. *Normal-Boundary Intersection Method*
NC	angl. *Normal Constraint Method*
PP	angl. *Physical Programming Method*
SKS	svertinės kriterijų sumos metodas (angl. *Weighted Sum Method*)
SP	sprendimų priėmėjas (angl. *Decision Maker, DM*)
SPS	sprendimų paramos sistema (angl. *Decision Support System, DSS*)

Turinys

1

Įvadas

1.1. Tyrimų sritis ir problemos aktualumas

Jau nuo senų laikų žmonės susiduria su optimizavimo problemomis ir ieško optimalių sprendimų. Žmonijai vystantis ir didėjant žmonių poreikiams, optimizavimo problemos vis sudėtingėja. Šiais laikais optimizavimo uždaviniai sprendžiami pasitelkiant optimizavimo teorines žinias bei kompiuterių resursus, kas žymiai palengvina uždavinių formalizavimą, sprendimo metodų parinkimą bei paspartina uždavinių sprendimą.

Praktikoje optimizavimo uždaviniai dažnai būna daugiakriteriniai. Tokie uždaviniai sprendžiami daugelyje žmonijos veiklos srityse: procesų valdyme, ekonomikoje, lėktuvų konstravime, tiltų statyme ir kitose. Su daugiakriteriniais optimizavimo uždaviniais susiduriama ir kasdieniniame gyvenime: perkant automobilį, renkantis poilsinę kelionę, sudarant maitinimosi racioną. Tokie uždaviniai sutinkami ten, kur reikia pasirinkti kompromisą tarp dviejų ar daugiau kriterijų, o kriterijai dažniausiai būna prieštaringi – mažinant vieno kriterijaus reikšmę, kito didėja. Pavyzdžiui, siekiant padidinti pelną, būtina sumažinti išlaidas; didinant automobilio galingumą, siekiama sumažinti kuro sąnaudas; mažinant tam tikros detalės svorį, būtina padidinti jos atsparumą. Dažnai kasdieniniame gyvenime daugiakriteriniai uždaviniai sprendžiami intuityviai. Tačiau sėkmingai tokio tipo uždaviniai sprendžiami tik matematiniais metodais. Daugiakriteriniai uždaviniai gali turėti kelis ar keliasdešimt kriterijų, ir priimant sprendimą reikia visus juos suderinti. Dažnas atvejis, kai tokio tipo uždaviniai neturi vieningo optimalaus sprendimo, tuomet ieškomi tik Pareto prasme optimalūs sprendiniai, dar vadinami nedominuojančiais.

Sprendžiant daugiakriterinį optimizavimo uždavinį, gaunama daug skirtingų Pareto optimalių sprendinių, o galutinis sprendimas priklauso nuo eksperto (sprendimo priėmėjo), kuris dalyvauja uždavinio sprendime. Vienas iš paprasčiausių daugiakriterinio optimizavimo būdų yra pagrįstas uždavinio suvedimu į vienakriterinį uždavinį naudojant skaliarizavimą. Dažnai tam taikomas gerai žinomas svertinės kriterijų sumos metodas (angl. *Weighted Sum Method*), kuriame sudaroma vienakriterinė tikslo funkcija, susumavus visų kriterijų tikslo funkcijas, padaugintas iš svertinių koeficientų. Galimi du svorinių koeficientų reikšmių parinkimo būdai. Vienu atveju koeficientų reikšmės generuojamos atsitiktinai, sudaromi keli koeficientų rinkiniai, sprendžiami vienakriteriniai uždaviniai, gaunami keli alternatyvūs sprendiniai, iš kurių sprendimų priėmėjas parenka jam tinkamiausią. Kitu atveju sprendimų priėmėjas pats interaktyviai parenka svorinių koeficientų reikšmes, atsižvelgdamas į sprendžiamo uždavinio specifiką, turimą patirtį bei kompetenciją. Tai įgalina sprendimų priėmėją dalyvauti ne tik sprendimo priėmime, bet ir sprendimo procese.

Svertinės kriterijų sumos metodo trūkumas be to, kad tik iškilomis tikslo funkcijoms randami visi Pareto sprendiniai, yra tas, kad gauti sprendiniai netolygiai pasiskirsto Pareto aibėje, todėl būtina ieškoti metodų, kurie minėto trūkumo neturi.

Jei pasirenkamas daugiakriterinio optimizavimo būdas, pagrįstas suvedimu į vienakriterinį uždavinį, tai uždavinys turi būti sprendžiamas daug kartų esant įvairiems svorinių koeficientų reikšmių rinkiniams. Kadangi šie vienakriteriniai uždaviniai gali būti sprendžiami nepriklausomai vienas nuo kito, sprendimo procesą galima paspartinti, pasitelkus daugiaprocesorinius kompiuterius ar kompiuterių klasterius. Būtina nustatyti sprendimo strategijas, siekiant efektyviai išnaudoti skaičiavimo resursus, įvertinant sprendimo priėmėjo galimybes formuoti užduotis.

Dar viena problema, su kuria susiduriama sprendžiant daugiakriterinį optimizavimo uždavinį interaktyviai, yra naudojamos sprendimų paramos sistemos kokybė. Sistema turi užtikrinti naudojimosi patogumą, greitį, tarpinių ir galutinių sprendinių vizualizavimą. Tas padeda sprendimų priėmėjui greičiau įsigilinti į sprendžiamą uždavinį ir perprasti jo specifiką, todėl būtina sukurti sistemą, kurioje būtų integruoti optimizavimo metodai, sprendimo proceso vizualizavimas bei jo lygiagretinimas.

Šioje disertacijoje sprendžiamos dvi problemos:

1. Interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo, užtikrinančio sprendinių tolygų pasiskirstymą Pareto aibėje, sukūrimas ir analizė.
2. Interaktyvios sprendimų paramos sistemos sukūrimas ir analizė.

1.2. Tyrimo objektas

Disertacijos tyrimo objektas yra daugiakriteriniai optimizavimo uždaviniai, interaktyvūs jų sprendimo metodai, interaktyvios sprendimų paramos sistemos bei lygiagrečių skaičiavimų taikymas sprendimų paramos sistemose.

1.3. Darbo tikslas ir uždaviniai

Darbo tikslas – pasiūlyti interaktyvų daugiakriterinių optimizavimo uždavinių sprendimo būdą, kurio pagalba sprendimų paramos sistemoje, veikiančioje kompiuterių klasteryje, sprendimų priėmėjas gauna alternatyvius sprendinius, tolygiai pasiskirsčiusius Pareto aibėje.

Siekiant tikslo buvo sprendžiami šie uždaviniai:

- ištirti esamus daugiakriterinių optimizavimo uždavinių sprendimo metodus bei interaktyvias sprendimų paramos sistemas, skirtas daugiakriteriniams optimizavimo uždaviniams spręsti;
- pasiūlyti interaktyvų daugiakriterinių optimizavimo uždavinių sprendimo būdą, kuriuo surandami alternatyvūs sprendiniai, tolygiai pasiskirstę Pareto aibėje;
- sukurti interaktyvią daugiakriterinių optimizavimo uždavinių sprendimų paramos sistemą, apjungiančią pasiūlytą optimizavimo uždavinių sprendimo būdą, sprendimo proceso vizualizavimą ir jo lygiagretinimą;
- sukurti ir palyginti sprendimo strategijas, interaktyviai sprendžiant daugiakriterinį optimizavimo uždavinį, pasitelkus kompiuterių klasterį;
- ištirti darbo su sukurtos sprendimų paramos sistemos apsimokymo laiką, būtiną sprendimų priėmėjui perprasti sprendžiamo uždavinio specifiką, siekiant greičiau rasti tinkamą sprendinį.

1.4. Tyrimo metodika

Analizuojant mokslinius pasiekimus daugiakriterinio optimizavimo bei sprendimų paramos sistemų kūrimo srityse buvo naudoti informacijos paieškos, sisteminimo, analizės, lyginamosios analizės ir apibendrinimo metodai. Remiantis eksperimentinio tyrimo metodu, atlikta statistinė tyrimų rezultatų analizė, kurios rezultatams įvertinti panaudotas apibendrinimo metodas.

1.5. Darbo mokslinis naujumas

1. Pasiūlytas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas, apjungiantis svertinės kriterijų sumos ir adaptyvų svertinės kriterijų sumos metodus, leidžiantis spręsti daugiakriterinius optimizavimo uždavinius interaktyviai ir užtikrinantis gaunamų sprendinių tolygų pasiskirstymą Pareto aibėje.
2. Eksperimentiškai palygintos sprendimo strategijos, pagal kurias sprendžiant daugiakriterinį optimizavimo uždavinį pasitelkiamas kompiuterių klasteris.
3. Eksperimentiškai ištirtas darbo su sprendimų paramos sistema apsimokymo laikas, būtinas sprendimų priėmėjui perprasti sprendžiamo daugiakriterinio optimizavimo uždavinio specifiką, siekiant greičiau rasti tinkamą sprendinį.

1.6. Darbo rezultatų praktinė reikšmė

Sukurta interaktyvi daugiakriterinių optimizavimo uždavinių sprendimų paramos sistema adaptuota pašarų sudėties sudarymo daugiakriteriniam uždaviniui spręsti. Ši sistema gali būti pritaikyta ir kitiems panašaus pobūdžio uždaviniams, pvz., dietos sudarymo, mokinių valgiaraščio sudarymo ir kt.

Dalis tyrimų atlikta pagal Lietuvos valstybinio mokslo ir studijų fondo remtą mokslininkų grupių projektą „Žmogiškojo faktoriaus tyrimas daugiakriteriniuose optimizavimo uždaviniuose skaičiuojant lygiagrečiai“, registracijos Nr. T-07134, vykdymo laikas 2007 metai.

1.7. Ginamieji teiginiai

1. Pasiūlytas daugiakriterinių optimizavimo uždavinių sprendimo būdas, kuriuo surandami alternatyvūs sprendiniai, tolygiai pasiskirstę Pareto aibėje, leidžiantys uždavinį spręsti interaktyviai.
2. Interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo, sprendimo proceso vizualizavimo ir jo lygiagretinimo integravimas į sukurtą sprendimų paramos sistemą užtikrina efektyvų sprendimo procesą, kas padeda sprendimų priėmėjui rasti jam tinkamą sprendinį.
3. Sukurta sprendimo strategija daugiakriteriniams optimizavimo uždaviniams spręsti, kurioje užduočių formavime dalyvauja kompiuteris, leidžia greičiau surasti sprendimų priėmėjui tinkamą sprendinį, naudojant didesnį kompiuterių skaičių.
4. Sprendimų priėmėjas greičiau apsimoko dirbti su sprendimų paramos sistema, kai uždavinio sprendimui naudojamas kompiuterių klasteris, tačiau kompiuterių skaičius turi būti toks, kad sprendimų priėmėjas spėtų įvertinti gaunamus sprendinius.

1.8. Darbo rezultatų aprobavimas

Tyrimų rezultatai publikuoti 5 moksliniuose leidiniuose: 4 periodiniuose recenzuojamuose mokslo žurnaluose, 1 konferencijos pranešimų medžiagoje. Tyrimų rezultatai buvo pristatyti ir aptarti šiose nacionalinėse ir tarptautinėse konferencijose Lietuvoje ir užsienyje:

1. The 65th Meeting of the European Working Group on Multiple Criteria Decision Aiding. April 12–14, 2007, Poznan, Poland.
2. Lietuvos jaunųjų mokslininkų konferencija „Operacijų tyrimas ir taikymai (LOTD – 2007)“. 2007 m. gegužės 18 d., Vilnius, Lietuva.

3. Informatikos doktorantų vasaros mokykla „Modernios duomenų gavybos ir analizės technologijos“. 2007 m. rugsėjo 9–15 d., Druskininkai, Lietuva.
4. International Networking for Young Scientists “High Performance Scientific Computing (INYS2008)”. February 5–8, 2008, Druskininkai, Lithuania.
5. The 20th International EURO Mini Conference „Continuous Optimization and Knowledge-Based Technologies (EurOPT-2008)“. May 20–23, 2008, Neringa, Lithuania.
6. The 5th International Vilnius Conference and EURO Mini Conference „Knowledge-Based Technologies and OR methodologies for Strategic Decisions of Sustainable Development (KORSD-2009)“. September 30 – October 3, 2009, Vilnius, Lithuania.
7. The 15th International Conference „Mathematical Modelling and Analysis (MMA2010)“, May 26–29, 2010, Druskininkai, Lithuania.
8. 15-oji tarptautinė mokslinė kompiuterininkų konferencija, „Kompiuterininkų dienos – 2011“, 2011 m. rugsėjo 22–24 d., Klaipėda, Lietuva.

1.9. Disertacijos struktūra

Disertaciją sudaro 5 skyriai ir literatūros sąrašas. Disertacijos skyriai: Įvadas, Daugiakriterinis optimizavimas ir sprendimų paramos sistemos, Interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas, Eksperimentiniai tyrimai, Bendrosios išvados. Papildomai disertacijoje pateikti naudotų žymėjimų ir santrumpų sąrašai. Disertacijos apimtis 139 puslapiai, kuriuose pateikti 29 paveikslai ir 5 lentelės. Disertacijoje remtasi 193 literatūros šaltiniais.

2

Daugiakriterinis optimizavimas ir sprendimų paramos sistemos

Šiame skyriuje pateikta daugiakriterinio optimizavimo apžvalga. Pateiktos pagrindinių sąvokų apibrėžtys. Trumpai aprašyti keli sprendinių vertinimo matai. Pateikta daugiakriterinio optimizavimo metodų klasifikacija. Didelis dėmesys skirtas interaktyviems metodams, o taip siekiantiems pagerinti sprendinių pasiskirstymą Pareto aibėje. Taip pat apžvelgiama sprendimų paramos sistemų (SPS) (angl. *Decision Support System*, *DSS*) vystymosi raida, SPS kūrimo principai bei jų klasifikacija. Nagrinėjamos SPS, kurios taikomos daugiakriteriniams optimizavimo uždaviniams spręsti, ir tos, kurių pagalba sprendžiami kiti uždaviniai.

Praktikoje optimizavimo uždaviniai dažnai būna daugiakriteriniai (Žilinskas 2000; Kalanta 2003; Bui and Alam 2008). Pagrindinės daugiakriterinio optimalumo (Edgeworth 1881) ir Pareto optimalumo (Pareto 1906) sąvokos buvo suformuotos jau daugiau nei prieš šimtą metų, tačiau didesnio susidomėjimo daugiakriteriniai optimizavimo uždaviniai sulaukė dvidešimto amžiaus šeštame dešimtmetyje (Koopmans 1951; Kuhn and Tucker

1951; Hurwicz 1958; Charnes and Cooper 1961). Pirmoji monografija šioje srityje buvo išleista 1972 metais (Lee 1972). Pirmieji interaktyvūs daugiakriterinių optimizavimo uždavinių sprendimo metodai atsirado aštuntame dešimtmetyje: STEM metodas (Benayoun *et al.* 1971) ir Geoffriono-Dyerio-Feinbergo metodas (Geoffrion *et al.* 1972). Vėliau ši optimizavimo kryptis populiarėjo ir sulaukdavo vis daugiau susidomėjimo. Galima išskirti šiuos svarbius darbus (Yu 1974; Zeleny 1974; Isermann 1974; Keeney and Raiffa 1976), kuriuose buvo pristatyti daugiakriterinio tiesinio programavimo metodai. Aštunto dešimtmečio pabaigoje ir devinto pradžioje išleista daug monografijų, apibendrinančių nuveiktus darbus daugiakriterinio optimizavimo srityje. Toliau išvardinama tik dalis jų (Hwang and Masud 1979; Chankong and Haimes 1983; Sawaragi *et al.* 1985; Steuer 1985; Tabucanon 1988). 1984 metais į daugiakriterinio optimizavimo metodų gretas įsiliejo ir genetiniai algoritmai, buvo pristatytas VEGA metodas (Schaffer 1984). Devintame dešimtmetyje, paplitus mikroprocesoriniams kompiuteriams bei ženkliai išaugus kompiuterių skaičiavimo pajėgumams, vizualizavimas bei interaktyvus daugiakriterinis optimizavimas sulaukė dar didesnio susidomėjimo (Korhonen and Laakso 1986; Angehrn and Luthi 1990; Belton and Vickers 1990; Pomerol and Barba-Romero 1993; Miettinen and Mäkelä 1993). Iki dabartinių laikų ši mokslinė sritis sparčiai plėtojama. Galima pažymėti tokius pastarųjų metų darbus (Deb *et al.* 2010; Miettinen *et al.* 2010; Luque *et al.* 2011; Zujevs and Eiduks 2011) bei svarbiausių darbų rinkinį *Encyclopedia of Optimization 2009*, kuriama nemažai dėmesio skirta daugiakriteriniams optimizavimo uždaviniams (Floudas and Pardalos 2009).

Daugiakriteriniai optimizavimo uždaviniai plačiai nagrinėjami bei sprendžiami ir Lietuvoje. Nemažai mokslininkų J. Mockus, A. Žilinskas, J. Žilinskas, G. Dzemyda, E. K. Zavadskas sprendžia įvairius optimizavimo uždavinius, taip pat ir daugiakriterinius. Paminėtini tokie svarbūs darbai (Mockus and Mockus 1991; Brauers and Zavadskas 2010; Radziukyniene and

Zilinskas 2008; Mockus 2011). Pastaruoju metu sėkmingai apginta nemažai disertacijų, kuriose sprendžiami daugiakriteriniai optimizavimo uždaviniai. Daug nuveikė A. Žilinsko auklėtinės. A. Mackutė-Varoneckienė savo disertacijoje (Mackutė-Varoneckienė 2006) daugiakriterinių optimizavimo uždavinių, kurių kriterijus ir ribojimus aprašančios funkcijos gali būti daugiaekstremalios ir netolydžios, sprendimui pasiūlė naują globaliojo optimizavimo metodą, kuris remiasi vizualizavimo ir statistinės duomenų analizės metodais. I. Steponavičė savo disertacijoje (Steponavičė 2010) analizavo tolygaus sprendinių išdėstymo Pareto aibėje algoritmus ir jų taikymus rizikos valdyme. J. Skūpienė savo darbe (Skūpienė 2010) sprendė daugiakriterinius uždavinius siekiant įvertinti mokinių sukurtas programas informatikos olimpiadose. J. Mockaus vadovaujamos L. Pupeikienės disertacijoje (Pupeikienė 2009) spręstas daugiakriterinis uždavinys profiliuotų mokyklų tvarkaraščiams sudaryti. J. Žilinskas kartu su doktorantu A. Lančinsku nagrinėja globalaus daugiakriterinio optimizavimo algoritmus (Lančinskas *et al.* 2011). Taipogi yra darbų, kuriuose daugiakriteriniai optimizavimo uždaviniai sprendžiami interaktyviai (Dzemyda and Petkus 2001; Petkus 2001).

Nuolat kuriami nauji daugiakriterinio optimizavimo metodai bei plėtojami esantys, ir šis procesas vyksta iki šiol.

2.1. Pagrindinės sąvokos

Daugiakriterinio optimizavimo uždavinio matematinė formuluotė išreiškiama šia formule (Zionts 1989):

$$\min_{X \in \mathbf{D}} \mathbf{F}(X) = [f_1(X), f_2(X), \dots, f_m(X)]^T. \qquad (2.1)$$

Leistinąją sritį $\mathbf{D} \subset R^n$ (apribojimus) nusako nelygybių ir lygybių sistemos:

$$g_j(X) \leq 0, \qquad j = 1, 2, \dots, k, \qquad (2.2)$$

$$h_l(X) = 0, \qquad l = 1, 2, \dots, p, \tag{2.3}$$

čia m – tikslo funkcijų (kriterijų) skaičius; k – nelygybinių apribojimų skaičius; p – lygybinių apribojimų skaičius; $X = (x_1, x_2, \dots, x_n)$ yra n nepriklausomų komponenčių sprendinio kintamųjų vektorius; $X \in \mathbf{D}$, $\mathbf{F}(X) \in R^m$ – kriterijų tikslo funkcijų vektorius; $f_i(X): R^n \to R^1$. Kriterijų reikšmių aibė $\mathbf{Z}$ apibrėžiama kaip rinkinis $\{\mathbf{F}(X) \mid X \in \mathbf{D}\}$.

Kriterijai tarpusavyje gali būti prieštaringi, todėl vienareikšmiškai negalima atsakyti, kuris sprendinys yra geriausias pagal visus kriterijus. Dažniausiai nėra vieno optimalaus sprendinio, todėl sprendžiant daugiakriterinius optimizavimo uždavinius randama visa aibė taškų, kurie tenkina optimumo apibrėžimą.

Toliau yra pateikiami keli svarbiausi apibrėžimai, kuriais dažniausiai remiamasi sprendžiant daugiakriterinius uždavinius (Marler and Arora 2004).

2.1. Apibrėžimas. Taškas $X^* \in \mathbf{D}$ yra *Pareto optimalus*, jeigu neegzistuoja kito taško $X \in \mathbf{D}$ tokio, kad $\mathbf{F}(X) \leq \mathbf{F}(X^*)$ ir bent vienai funkcijai $f_i(X) < f_i(X^*)$. Toks taškas X^* dar vadinamas *efektyviuoju* sprendiniu.

Sprendžiant daugiakriterinius optimizavimo uždavinius kartais būna svarbūs ne tik Pareto optimalūs sprendiniai, bet ir sprendiniai, tenkinantys kitus kriterijus, pvz., silpnas Pareto optimumas.

2.2. Apibrėžimas. Taškas $X^* \in \mathbf{D}$ yra *silpnai Pareto optimalus*, jeigu neegzistuoja kito taško $X \in \mathbf{D}$ tokio, kad $\mathbf{F}(X) < \mathbf{F}(X^*)$.

Taškas yra silpnai Pareto optimalus, jeigu nėra kito taško, kuris pagerina sprendinį pagal visus kriterijus, tuo tarpu taškas yra Pareto optimalus, jeigu nėra kito taško, kuris pagerina sprendinį nors pagal vieną iš kriterijų, nepabloginant pagal kitus.

2.3. Apibrėžimas. Aibė visų Pareto optimalių (efektyviųjų) taškų vadinama *Pareto aibe*.

2.4. Apibrėžimas. Tikslo funkcijų (kriterijų) vektorius $\mathbf{F}(X^*) \in \mathbf{Z}$ yra *nedominuojantis*, jeigu neegzistuoja kito vektoriaus $\mathbf{F}(X) \in \mathbf{Z}$ tokio, kad

$\mathbf{F}(X) \leq \mathbf{F}(X^*)$ bent vienai funkcijai $f_i(X) < f_i(X^*)$. Priešingu atveju $\mathbf{F}(X^*)$ yra *dominuojantis*.

2.5. Apibrėžimas. Tikslo funkcijų (kriterijų) vektorius $\mathbf{F}(X^*) \in \mathbf{Z}$ yra *silpnai nedominuojantis*, jeigu neegzistuoja kito vektoriaus $\mathbf{F}(X) \in \mathbf{Z}$ tokio, kad $\mathbf{F}(X) < \mathbf{F}(X^*)$. Priešingu atveju $\mathbf{F}(X^*)$ yra *silpnai dominuojantis*.

Praktiniams uždaviniams 2.1. ir 2.4. apibrėžimai yra ekvivalentūs. Tačiau efektyvumas apibrėžiamas sprendinio kintamųjų vektoriams leistinojoje srityje, o dominavimas – tikslo funkcijų vektoriams kriterijų reikšmių aibėje.

2.6. Apibrėžimas. Taškas $\mathbf{F}^{\mathrm{id}}(X) \in \mathbf{Z}^m$ vadinamas *idealiu vektoriumi*, jeigu kiekvienam $i = 1, 2, \dots, m$ $f_i^{\mathrm{id}}(X) = \min_X\{f_i(X) \mid X \in \mathbf{D}\}$.

Paprastai idealus vektorius neegzistuoja. Kai kuriems optimizavimo algoritmams reikalingas sprendinys, kuris yra griežtai geresnis už kiekvieną paieškos srities sprendinį. Tada naudojamas utopinis vektorius.

2.7. Apibrėžimas. Taškas $\mathbf{F}^{\mathrm{u}}(X) \in \mathbf{Z}^m$ vadinamas *utopiniu vektoriumi*, jeigu jis nežymiai mažesnis negu idealus taškas, t. y. kiekvienam $i = 1, 2, \dots, m$ $f_i^{\mathrm{u}}(X) = f_i^{\mathrm{id}}(X) - \varepsilon_i$, čia $\varepsilon_i > 0$.

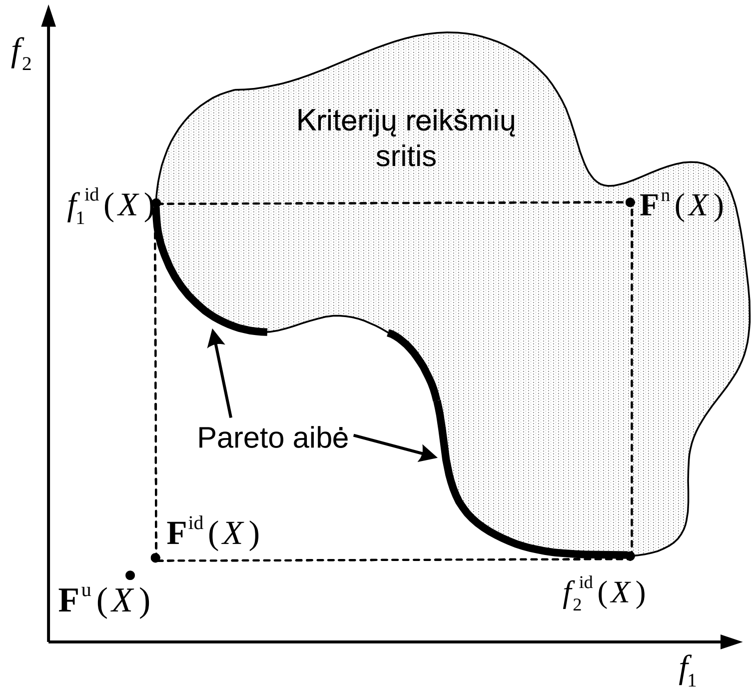
Reiktų pastebėti, kad dažnai literatūroje idealus ir utopinis vektoriai yra sutapatinami, t. y. utopinis vektorius yra toks pats kaip ir idealus.

2.8. Apibrėžimas. Taškas $\mathbf{F}^{\mathrm{n}}(X) \in \mathbf{Z}^m$ vadinamas *nadir vektoriumi*, jeigu kiekvienam $i = 1, 2, \dots, m$ $f_i^{\mathrm{n}}(X) = \max_X\{f_i(X) \mid X \in \mathbf{D}\}$.

Priklausomai nuo Pareto optimalių sprendinių aibės iškilumo ir tolydumo nadir vektorius gali atitikti egzistuojantį sprendinį arba ne.

Praktikoje nadir ir idealus vektoriai dažnai naudojami kiekvieno kriterijaus normavimui: $\hat{f}_i(X) = \left(f_i(X) - f_i^{\mathrm{id}}(X)\right) / \left(f_i^{\mathrm{n}}(X) - f_i^{\mathrm{id}}(X)\right)$.

2.1 pav. pavaizduota Pareto aibė, utopinis ir nadir vektoriai dviejų kriterijų atveju.



2.1 pav. *Pareto aibė, utopinis ir nadir vektoriai*

Pareto taškų radimas yra pagrindinis tikslas sprendžiant daugiakriterinius optimizavimo uždavinius. Kai tikslo funkcijos yra tolydžios, Pareto optimalių sprendinių skaičius dažnai yra begalinis. Sudėtingų uždavinių atveju Pareto sprendinių aibę analitiškai rasti labai sudėtinga arba neįmanoma. Todėl praktiniuose uždaviniuose ieškoma Pareto aibės taškų, kurie kiek galima tolygiau padengtų Pareto aibę.

2.2. Daugiakriterinio optimizavimo metodų klasifikacija

Šiuo metu yra sukurta daug daugiakriterinio optimizavimo metodų, kurie gali būti klasifikuojami įvairiai. Viena iš populiariausių klasifikacijų remiasi tuo, ar Pareto optimalūs sprendiniai yra generuojami ar ne (Cohon 1985). Išskiriamos dvi grupės: (1) sprendinius generuojantys metodai (angl. *Generating Methods*) ir (2) pirmenybe grįsti metodai (angl. *Preference Based Methods*). Sprendinius generuojančiais metodais surandama tiksli Pareto aibė arba jos aproksimacija, ir keletas nedominuojančių sprendinių yra pateikiami sprendimų priėmėjui

(SP) (angl. *Decision Maker, DM*), kuris iš jų pasirenka vieną tinkamiausią. Be to, optimizuojant nenaudojama jokios informacijos, susijusios su kriterijų svarbumu. Priešingai yra pirmenybe grįstuose metoduose – čia yra atsižvelgiama į kriterijų svarbą. SP turi formaliai išryškinti kriterijus, kuriems teikia pirmenybę, ir ta informacija naudojama optimizuojant pasirinktu šios grupės metodu.

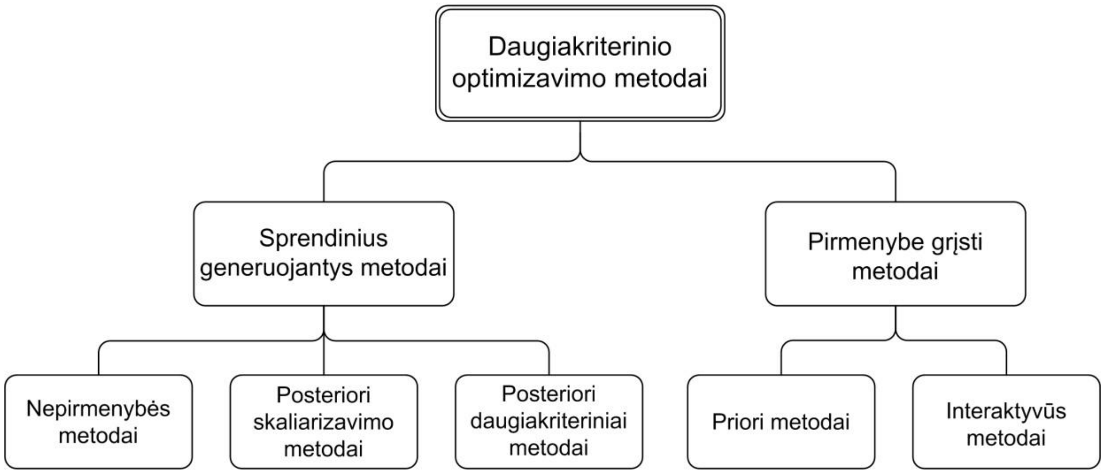
Sprendžiant daugiakriterinius optimizavimo uždavinius yra būtinas sprendimų priėmėjo dalyvavimas (Huang *et al.* 2005; Miettinen and Makela 2006; Klamroth and Miettinen 2008). Toliau aprašoma plačiai žinoma klasifikacija, kuri praplečia ankščiau pateiktą. Metodai klasifikuojami pagal tai, kaip SP dalyvauja uždavinio sprendimo procese (Hwang and Masud 1979; Lieberman 1991; Miettinen 1999; Diwekar 2008):

- Nepirmenybės metodai (angl. *No Preference Methods*).
- Posteriori metodai (angl. *Posteriori Methods*).
- Priori metodai (angl. *Priori Methods*).
- Interaktyvūs metodai (angl. *Interactive Methods*).

Metodų apibendrinta klasifikacija pateikta 2.2 pav.

Nepirmenybės metoduose nenaudojama jokios informacijos, gautos iš sprendimų priėmėjo apie kriterijų svarbą, ir sprendimo rezultatas būna vienas Pareto optimalus sprendinys. Kelius sprendinius galima gauti naudojant skirtingus nepirmenybės metodus arba skirtingas metrikas.

Posteriori metodai dar vadinami metodais, generuojančiais Pareto optimalius sprendinius (angl. *Methods for Generating Pareto Optimal Solutions*). Naudojant šios grupės metodus, surandama daug Pareto optimalių sprendinių, kurie yra pateikiami sprendimų priėmėjui. Iš pateiktų sprendinių SP pasirenka labiausiai jį tenkinantį. Šių metodų ypatumas tas, kad sprendinys nepriklauso nuo SP prioritetų ir siekiamų tikslų. Tačiau šių metodų trūkumas tas, kad jie reikalauja didelių skaičiavimo resursų. Be to, SP turi pasirinkti sprendinius iš daugybės alternatyvų. Taigi vienas iš šių metodų uždavinių yra rasti būdą, kaip patogiai pateikti gautus sprendinius sprendimų priėmėjui, kad SP galėtų juos palyginti.



2.2 pav. *Daugiakriterinio optimizavimo metodų klasifikacija*

Priori metoduose prieš sprendimo procesą SP turi nurodyti savo teikiamus prioritetus ir siekiamus tikslus. Jeigu gaunamas sprendinys atitinka SP reikalavimus, tai sprendimo procese SP gaišta nedaug laiko. Tačiau problema yra ta, kad SP prieš sprendimo procesą nebūtinai žino, kokių rezultatų jis gali tikėtis ir kiek įgyvendinami yra siekiami tikslai. Todėl tuo atveju, kai sprendinys SP netenkina, jam reikia keisti prioritetus bei koreguoti siekiamus tikslus.

Interaktyviuose metoduose SP aktyviai dalyvauja uždavinio sprendimo procese. Metodų specifika ta, kad dėl sprendžiamo uždavinio sudėtingumo SP negali tinkamai nustatyti prioritetų ir siekiamų tikslų prieš sprendimo procesą. Tačiau SP gali koreguoti prioritetus ir siekiamus tikslus sprendimo proceso eigoje. Sprendimas atliekamas iteratyviai. SP, nustatęs įvairius tikslus ir pakeitęs prioritetus, gauna skirtingus sprendinius, kuriuos įvertina ir tokiu būdu susipažįsta su sprendžiamo uždavinio specifika. Taigi kiekviena iteracija interaktyviuose metoduose susideda iš dviejų dalių: kompiuterio skaičiavimų ir sprendimo priėmėjo veiksmų. Interaktyvūs metodai reikalauja daugiausia SP laiko lyginant su kitais metodais, tačiau įgalina išspręsti sudėtingus uždavinius su daug kriterijų ir apribojimų. Čia nebūtina tinkamai nustatyti apribojimų uždavinio sprendimo pradžioje, kas yra būtina priori metoduose. Be to, taupomas kompiuterių skaičiavimo laikas, nes nereikia gauti daugybės sprendinių, kaip posteriori metoduose, o gaunami tik tie sprendiniai, kurie tenkina SP reikalavimus.

Yra sukurta daug įvairių metodų ir jų modifikacijų, priklausančių kiekvienai aprašytai grupei. Tolimesniuose skyreliuose bus paminėti ir aprašyti populiariausi ir dažniausiai naudojami metodai.

2.2.1. Nepirmenybės metodai

Nepirmenybės metodų grupės didžiąją dalį sudaro globalaus kriterijaus metodai (Yu 1973; Zeleny 1973) ir metodai, kurie jais remiasi. Globalaus kriterijaus metoduose yra minimizuojamas atstumas tarp vadinamojo atramos taško $r = (r_1, r_2, \dots, r_m) \in R^m$ (angl. *Reference Point*) ir numatomo sprendinio. SP turi nustatyti atramos tašką ir metriką atstumų skaičiavimui. Atramos tašku dažniausiai pasirenkamas utopinis vektorius. Tada minimizuojamas atstumas tarp sprendinio ir utopinio vektorių. Taigi optimizavimo uždavinys formuluojamas taip:

$$\min_{X \in \mathrm{D}} \left(\sum_{i=1}^{m} \left| f_i(X) - f_i^{\mathrm{u}}(X) \right|^{p} \right)^{\frac{1}{p}}, \tag{2.4}$$

čia $1 \le p < \infty$. p reikšmės pasirinkimas suteikia galimybę pasirinkti metriką ir skirtingais būdais skaičiuoti atstumus tarp sprendinio ir utopinio vektorių: miesto kvartalo (angl. *City Block, Manhattan*) metrika, kai $p = 1$; Euklido metrika, kai $p = 2$; Čebyševo metrika, kai $p = \infty$. Šios grupės metoduose visi kriterijai yra vienodai svarbūs.

Taip pat galima paminėti tokius nepirmenybės metodus, kaip *Nash Arbitration and Objective Product Method* (Straffin 1996) ir *Multiobjective Proximal Bundle Method* (Miettinen and Mäkelä 1995).

2.2.2. Posteriori metodai

Posteriori metodų grupės didžiausias privalumas yra tas, kad sprendinys nepriklauso nuo SP prioritetų. Posteriori metodų grupė gali būti suskirstyta į du pogrupius: metodai, transformuojantys uždavinį į vienakriterinį naudojant skaliarizavimą, bei daugiakriterinės paieškos metodai (Rangaiah 2008).

Pirmo pogrupio metoduose daugiakriterinis optimizavimo uždavinys yra transformuojamas į vieną arba keletą vienakriterinių optimizavimo uždavinių ir optimizuojama „skaliarizuota“ funkcija. Yra daug būdų sudaryti skaliarizuotą funkciją. Skaliarizavimo procedūra gali būti paprasta, tačiau pats sudarytų vienakriterinių optimizavimo uždavinių sprendimas nėra lengvas. Be to, reikia žinių ir patirties norint tam tikram uždaviniui parinkti tinkamą daugiakriterinio optimizavimo metodą. Išsprendus eilę vienakriterinių optimizavimo uždavinių, gaunami skirtingi Pareto sprendiniai ir SP turi pasirinkti tinkamiausią.

Svertinės kriterijų sumos (SKS) metodas

Dažnai sprendžiant daugiakriterinius optimizavimo uždavinius, naudojamas svertinės kriterijų sumos (SKS) metodas (angl. *Weighted Sum Method*) (Gass and Saaty 1955; Zadeh 1963). Kiekvienai tikslo funkcijai suteikiamas svorinis koeficientas ir minimizuojama kriterijų tikslo funkcijų suma. Tokiu būdu daugiakriterinis optimizavimo uždavinys transformuojamas į vienakriterinį. Uždavinys formuluojamas taip:

$$\min_{X \in \mathrm{D}} \sum_{i=1}^{m} w_i f_i(X), \tag{2.5}$$

čia w_i yra i-tojo kriterijaus svorinis koeficientas, $0 < w_i \leq 1$, $\sum_{i=1}^{m} w_i = 1$. Esant skirtingiems svorinių koeficientų reikšmių rinkiniams $w = (w_1, w_2, \dots, w_m)$, gaunami skirtingi sprendiniai. Šiuo metodu surasti sprendiniai yra Pareto optimalūs. Šio metodo privalumas tas, kad jo įgyvendinimas yra nesudėtingas, todėl šis metodas išlieka populiarus sprendžiant praktinius uždavinius. Be to, šiuo metodu rasti sprendiniai gali būti panaudoti kaip pradiniai sprendiniai kitiems metodams. Šio metodo trūkumas yra tas, kad nuosekliai keičiant svorinių koeficientų reikšmes nebūtinai gaunami sprendiniai bus tolygiai pasiskirstę Pareto aibėje (Das and Dennis 1997). Sprendimų priėmėjui kartais būna sudėtinga nustatyti reikiamas svorinių koeficientų reikšmes, kai jis mažai žino apie sprendžiamo uždavinio specifiką. Be to, jei funkcijos f_i(X) nėra iškilos, ne visi Pareto sprendiniai bus rasti nepriklausomai nuo to, kokie svorinių koeficientų reikšmių rinkiniai $w = (w_1, w_2, \dots, w_m)$ bebūtų parinkti (Ritzel *et al.* 1994).

Kitas dažnai naudojamas metodas yra ε-ribojimų metodas (angl. *ε-constraint Method*) (Haimes *et al.* 1971). Šis metodas remiasi tuo, kad minimizuojama vieno kriterijaus tikslo funkcija, o kitų kriterijų funkcijos yra suvedamos į apribojimus. Uždavinys formuluojamas taip:

$$\min_{X \in \mathrm{D}} f_l(X), \qquad l \in \{1, \dots, k\}, \tag{2.6}$$

esant apribojimams:

$$f_i(X) \le \varepsilon_i\,, \qquad i = 1, \dots, k, \;\; i \ne l, \tag{2.7}$$

čia ε_i yra kriterijų viršutiniai rėžiai. Sprendžiant praktinius uždavinius sudėtinga nustatyti kriterijų viršutinius rėžius taip, kad sprendinių aibė nebūtų tuščia, ypač esant dideliam kriterijų skaičiui.

Prie posteriori daugiakriterinės paieškos metodų pogrupio priskiriama didelė klasė metaeuristinių metodų, kurie iš karto generuoja Pareto aibės sprendinius, t. y. iš karto sprendžia daugiakriterinį optimizavimo uždavinį, nesuvedant jo į vienakriterinį (Abraham and Jain 2005). Euristiniais metodais siekiama rasti geros kokybės sprendinį (artimą optimaliam, bet nebūtinai optimalų) per priimtiną skaičiavimų laiką. Šie metodai remiasi tam tikrų taisyklių, kurios vadinamos euristikomis, naudojimu sprendžiant optimizavimo uždavinį. Šios taisyklės lemia sprendimą, kurį reikia priimti konkrečiose situacijose, kas gali padėti gauti geresnį sprendinį, sprendžiant konkretų uždavinį. Metaeuristiniai metodai susideda iš aukšto abstrakcijos lygio nurodymų rinkinių. Skirtingai nei euristinių metodų, tokių nurodymų paskirtis yra formaliai aprašyti ne vieno uždavinio, o tam tikros klasės uždavinių sprendimo principą. Sąvoka „metaeuristiniai metodai“ yra platesnė nei „euristiniai metodai“. Prie metaeuristinių metodų priskiriama: tabu paieška (angl. *Tabu Search*), atkaitinimo modeliavimas (angl. *Simulated Annealing*), skruzdžių kolonijos elgsenos imitavimo metodai (angl. *Ant Colony Optimization*), genetiniai algoritmai (angl. *Genetic Algorithms*) ir kt. Metaeuristiniai metodai pastaruoju metu tapo populiarūs. Šie metodai negarantuoja, kad artimas optimaliam sprendinys bus surastas vienodai greitai visais atvejais, tačiau šie metodai suranda sprendinius, artimus optimaliems sprendiniams, sprendžiant daugelį praktinių uždavinių. Plataus taikymo galimybės padidina šių metodų patrauklumą. Tačiau vienas iš pagrindinių šių metodų trūkumų tas, kad jie reikalauja daug skaičiavimo resursų, nes randa daugybę ne Pareto optimalių sprendinių (Shan and Wang 2005). Be to, šie metodai neužtikrina tolygaus Pareto aibės padengimo ar sprendinių radimo iš visos Pareto aibės.

Pastaruoju metu genetiniai algoritmai sulaukė išskirtinio susidomėjimo. Genetiniai algoritmai yra evoliucinių skaičiavimų metodų atšaka, naudojanti gamtoje egzistuojančius gyvybės evoliucinius mechanizmus: paveldėjimą, mutaciją, natūraliąją atranką ir rekombinaciją. Sprendinių paieška remiasi sprendinių (populiacijų) generavimu ir įvertinimu. Pirmasis genetinių algoritmų daugiakriterinio optimizavimo metodas VEGA buvo pristatytas 1984 metais. Nuo tada sukurta daug įvairių metodų, kurie gali būti suskirstyti į grupes, atsižvelgiant į pagrindinius metodų komponentus ir sprendinių paieškos būdus. Dažniausiai išskiriamos dvi grupės: algoritmai, naudojantys elito atranką (angl. *Elitist Algorithms*), ir nenaudojantys elito atrankos algoritmai (angl. *Non-Elitist Algorithms*). Elito atrankos algoritmuose konstruojant naują populiaciją, leidžiama keliems sėkmingiausiems sprendiniams pereiti į naują populiacijai visai nepakitus. Pirmos grupės metodai yra: VEGA, MOGA, NPGA, NSGA ir kt.; antros grupės: NSGA-II, SPEA, SPEA 2, PAES, PESA (Guliashki *et al.* 2009). Nors genetiniai algoritmai reikalauja didelių skaičiavimo resursų, tačiau pasižymi geromis lygiagretinimo savybėmis bei nesudėtingu įgyvendinimu. Todėl, didėjant skaičiavimo resursams, augant kompiuterių greičiams, kompiuterius jungiant į kompiuterių klasterius (angl. *Computer Clusters*) ir gridus (angl. *Grids*), genetiniai algoritmai sulaukia vis didesnio susidomėjimo. Atsiranda naujų metodų ir jų modifikacijų. Genetiniai algoritmai vis dažniau pritaikomi praktiniams uždaviniams, tačiau išlieka trūkumas, kad jais gaunami sprendiniai gali nebūti Pareto optimalūs.

2.2.3. Priori metodai

Prie priori metodų grupės gali būti priskirtas 2.2.2 skyrelyje aprašytas SKS metodas, kai SP pats parenka svorinių koeficientų $w = (w_1, w_2, \dots, w_m)$ reikšmes prieš sprendimo procesą. Analogiškai yra ir su ε-ribojimų metodu. Tuo atveju SP pats parenka to kriterijaus tikslo funkciją, kuri bus minimizuojama, ir nustato kitų kriterijų viršutinius rėžius.

Prie šios metodų grupės taipogi priskiriamas plačiai naudojamas naudingumo funkcijos metodas (angl. *Value Function Method*) (Keeney and Raiffa 1976). Šis metodas puikiai tinka tada, kai SP gali tiksliai matematiškai suformuluoti savo reikalavimus ir siekiamus tikslus per naudingumo funkciją $U: R^m \to R^1$. Uždavinys formuluojamas taip:

$$\min_{X \in \mathrm{D}} U(\mathbf{F}(X)), \tag{2.8}$$

Šis metodas atrodo paprastas, tačiau dažnai sudėtinga sudaryti funkcijos matematinę išraišką.

Kitas dažnai naudojamas metodas yra leksikografinis rikiavimas (angl. *Lexicographic Ordering*) (Fishburn 1974). Čia SP turi surikiuoti tikslo funkcijas pagal svarbą. Po rikiavimo svarbiausia funkcija yra minimizuojama esant pradiniams uždavinio apribojimams. Jeigu gaunamas tik vienas sprendinys, jis ir yra uždavinio galutinis sprendinys. Priešingu atveju yra minimizuojama antra pagal svarbą funkcija, o pradiniai apribojimai papildomi naujais, kurie užtikrina, kad pati svarbiausia funkcija išlaikys gautą optimalią reikšmę. Jeigu naujai suformuoto uždavinio sprendinys yra vienintelis, tai jis yra uždavinio galutinis sprendinys, priešingu atveju optimizuojama trečia pagal svarbą funkcija ir t. t. Metodas yra gana paprastas, tačiau dažnai sprendimų priėmėjui yra sudėtinga vienareikšmiškai nustatyti tikslo funkcijų svarbą. Be to, kuo mažesnė funkcijos svarba, tuo ji mažiau turi įtakos galutiniam sprendiniui.

Šiai metodų grupei priklauso ir pats pirmas specialiai daugiakriteriniam optimizavimui sukurtas metodas – tikslo programavimas (angl. *Goal Programming*) (Charnes *et al.* 1955; Charnes and Cooper 1961). Metodo idėja yra ta, kad SP tikslo funkcijoms $\mathbf{F}(X) = (f_1(X), f_2(X), \dots, f_m(X))$ nurodo siekiamą tikslo vektorių $\mathbf{F}^0 = (f_1^0, f_2^0, \dots, f_m^0)$ ir minimizuojami nukrypimai $d = (d_1, d_2, \dots, d_m)$ nuo šio vektoriaus. Nukrypimai atlieka apribojimų vaidmenį. Siekiant sumodeliuoti absoliučią reikšmę, nukrypimas d_i dalinamas į teigiamą ir neigiamą dalis: $d_i = d_i^+ - d_i^-$, čia $d_i^+ \geq 0$, $d_i^- \geq 0$ ir $d_i^+ d_i^- = 0$. Taigi $|d_i| = d_i^+ + d_i^-$. Optimizavimo uždavinys formuluojamas taip:

$$\min_{X \in \mathbf{D}, d^+, d^-} \sum_{i=1}^{m} (d_i^+ + d_i^-), \tag{2.9}$$

esant apribojimams:

$$f_i(X) - d_i^+ + d_i^- = f_i^0, \quad i = 1, 2, \dots, m, \tag{2.10}$$

$$d_i^+, d_i^- \geq 0, \quad i = 1, 2, \dots, m, \tag{2.11}$$

$$d_i^+ d_i^- = 0, \quad i = 1, 2, \dots, m. \tag{2.12}$$

Metodas turi keletą modifikacijų: svorinis tikslo programavimas (Charnes and Cooper 1977), kuriame minimizuojama svorinė nukrypimų suma; leksikografinis tikslo programavimas (Ignizio 1983), kuriame po leksikografinio rikiavimo minimizuojami nukrypimai $|d_i| = d_i^+ + d_i^-$;

minimakso tikslo programavimas (Flavell 1976), kuriame minimizuojamas nukrypimų maksimumas. Metodas dėl savo paprastumo yra dažnai taikomas praktiniuose uždaviniuose. Jis yra efektyvus, kai SP tinkamai pasirenka tikslo vektorių $\mathbf{F}^0$, ir jo reikšmės yra kriterijų reikšmių aibėje **Z**. Kai kriterijų reikšmių aibę sudėtinga nustatyti, metodas tampa labai neefektyviu, ir ne visi gauti sprendiniai yra Pareto optimalūs.

2.2.4. Interaktyvūs metodai

Šiuo metu yra sukurta daug įvairių interaktyvių metodų. Negalima išryškinti geriausio, nes vieni metodai labiau tinka vieniems praktiniams uždaviniams, o kiti kitiems. Be to, sprendimų priėmėjams skirtingai sekasi spręsti uždavinius skirtingais metodais. Metodai skiriasi vienas nuo kito tuo, kokia informacija pateikiama sprendimų priėmėjui, kokia yra informacijos pateikimo forma, ir kokiu būdu daugiakriterinis uždavinys transformuojamas į vienakriterinį (skaliarizuojamas). Interaktyvūs metodai gali būti klasifikuojami skirtingai, dažniausiai atsižvelgiant į juose įgyvendintus vienakriterinio uždavinio sprendimų metodus. Pateiktoje klasifikacijoje (Kaliszewski 2006) metodai suskirstomi į tris klases:

- Svertinių kriterijų metodų klasė (angl. *Weight Method Class*).
- Atramos taško metodų klasė (angl. *Reference Point Method Class*).
- Apribojimų metodų klasė (angl. *Constraint Method Class*).

Svertinių kriterijų metodais gautų Pareto optimalių sprendinių aibė yra susieta su svorių, su kuriais buvo išspręstas uždavinys, aibe. Tokiu būdu SP reikalavimai nustatomi svoriais. Tokiuose metoduose yra ieškoma SP tenkinančių svorių aibė ir stengiamasi šią aibę sumažinti, kad galutiniame variante SP galėtų pasirinkti labiausiai jį tenkinantį sprendinį. Populiariausi šios klasės metodai yra *Zionts-Wallenius method* (Zionts and Wallenius 1976), *Interactive Weighted Tchebycheff method* (Steuer and Choo 1983), *Interactive*

Weighted-Sum method (Steuer 1985), *Dell-Karwan method* (Dell and Karwan 1990).

Atramos taško metoduose SP prioritetai nurodomi pasirenkant atramos tašką, kuris gali būti bet kuris aibės R^m vektorius, bet dažniausiai tai yra trokštamas SP sprendinys. Minimizuojamos iš tikslo funkcijų ir atramos taškų sudarytos skaliarizavimo funkcijos, kurios projektuoja atramos tašką į Pareto aibę. Atramos taško metodų idėja detaliai aprašyta straipsnyje (Wierzbicki 1982). Pasirenkant skirtingus atramos taškus, gaunami skirtingi Pareto aibės sprendiniai, taigi SP gali sudaryti Pareto aibės poaibį ir iš šio poaibio pasirinkti tinkamiausią sprendinį. Dažniausiai naudojami tokie metodai: *Reference Point Method* (Wierzbicki 1980; Wierzbicki 1999), *STOM method* (Nakayama and Sawaragi 1984; Nakayama 1995), *GUESS method* (Buchanan 1997), *Light Beam Search method* (Jaszkiewicz and Slowinski 1994), *Reference Direction Approach method* (Korhonen and Laakso 1986; Korhonen 1988), *Reference Direction method* (Narula *et al.* 1994), *PROJECT method* (Luque and Jian-Bo Yang Wong 2009), *NAUTILUS method* (Miettinen *et al.* 2010).

Apribojimų metodų klasės metoduose SP nustato reikalavimus įvedant, nuimant, sugriežtinant arba sušvelninat esamus apribojimus, atsižvelgiant į sprendinį, gautą su prieš tai nustatytais apribojimais. Galima išskirti tokius metodus: *STEM method* (Benayoun *et al.* 1971), *ISWT method* (Chankong and Haimes 1978), *SPOT method* (Sakawa 1982), *NIMBUS method* (Miettinen and Mäkelä 1995), *GAMMA-L method* (Vassilev *et al.* 2003).

Pastaraisiais metais buvo sukurta interaktyvių metodų, kuriuose pritaikyti ir genetiniai algoritmai (Deb and Kumar 2007; Thiele *et al.* 2009; Deb *et al.* 2010; Sindhya *et al.* 2011).

Kiekvieno interaktyvaus metodo efektyvumas priklauso nuo kelių faktorių: informacijos tipo, kurią SP nurodo siekdamas pagerinti gautą sprendinį; laiko, reikalingo minimizuoti skaliarizuotą funkciją; galimybių sprendimų priėmėjui apsimokyti spręsti praktinį uždavinį; kiekvienoje iteracijoje gautų ir vertinamų sprendinių skaičiaus.

2.3. Pareto aibės tolygaus padengimo metodai

Sprendžiant daugiakriterinius optimizavimo uždavinius, sprendimų priėmėjui svarbu rasti sprendinių iš visos Pareto aibės, kad SP galėtų gauti kuo daugiau informacijos apie sprendžiamą uždavinį ir pasirinkti tinkamą sprendinį. Kai uždaviniai reikalauja didelių skaičiavimo resursų arba SP negali gauti daug Pareto optimalių sprendinių, būtina rasti sprendinius, kurie tolygiai padengtų Pareto aibę (Grosan and Abraham 2008).

Kaip minėta 2.2.2 skyrelyje, svertinės kriterijų sumos metodas neužtikrina sprendinių radimo iš visos Pareto aibės, kai funkcijos f_i(X) nėra iškilos, ir tolygaus sprendinių pasiskirstymo Pareto aibėje. Todėl buvo sukurta daug metodų, siekiant išvengti šių trūkumų. Tolesniuose skyreliuose bus aprašyti pagrindiniai Pareto aibės tolygaus padengimo metodai. Reiktų pastebėti, kad šie metodai 2.2 poskyryje pateiktoje daugiakriterinio optimizavimo metodų klasifikacijoje priskiriami prie posteriori metodų grupės.

2.3.1. NBI metodas

Vienas iš žinomiausių tolygaus sprendinių radimo Pareto aibėje metodų yra *Normal-Boundary Intersection* (NBI) metodas (Das and Dennis 1998). Jame keičiant svorinių koeficientų reikšmes gaunami sprendiniai, tolygiai pasiskirstę Pareto aibėje. Be to, gautų sprendinių pasiskirstymas nepriklauso nuo tikslo funkcijų normavimo (Eddy and Lewis 2001; Messac and Mattson 2002). Šis metodas remiasi idėja, kad Pareto optimalus sprendinys turi būti tarp utopinės tiesės ir utopinio taško $\mathbf{F}^{\mathrm{u}}(X)$. Utopinė tiesė (dviejų kriterijų atveju) – tai tiesė, jungianti taškus $f_1^{\mathrm{u}}(X)$ ir $f_2^{\mathrm{u}}(X)$. Pareto aibė randama įvedant papildomus lygybinius apribojimus, funkcijų reikšmių aibę suskirstant tiesėmis, statmenomis utopinei tiesei. Šių tiesių susikirtimo taškai su utopine tiese nustatomi svoriniais koeficientais. Maksimizuojamas tiesių atkarpų tarp utopinės tiesės ir kriterijų reikšmių ribos ilgis. Taigi sprendžiama eilė vienakriterinių uždavinių. Daugelio kriterijų atveju utopinė tiesė tampa hiperplokštuma.

Metodo privalumas yra tas, kad jo pagalba surandami Pareto optimalūs sprendiniai tose kriterijų reikšmių aibės srityse, kur funkcijos $f_i(\mathrm{X})$ nėra iškilos. Metodo trūkumai tokie: (1) sudėtinguose netiesiniuose uždaviniuose nelengva rasti optimalius sprendinius, naudojant lygybinius apribojimus; (2) ne visi gauti sprendiniai yra Pareto optimalūs, todėl juos reikia pašalinti, naudojant Pareto filtrą; (3) kai kriterijų daugiau negu du, padengiama ne visa Pareto aibė (Messac *et al.* 2003; Kim and de Weck 2006).

Vėliau buvo sukurta NBI metodo modifikacija *Modified Normal-Boundary Intersection* (mNBI) (Shukla 2007). Šis metodas išsiskyrė tuo, kad visi juo gaunami sprendiniai yra Pareto optimalūs ir šiuo požiūriu teoriškai prilygsta SKS metodui.

2.3.2. NC metodas

Normal Constraint (NC) metodas (Messac *et al.* 2003) buvo sukurtas siekiant pašalinti NBI metodo trūkumus. Šis metodas remiasi gerai žinomu faktu, kad Pareto aibė yra ant dalies kriterijų reikšmių aibės ribos (Miettinen 1999). Pirma, kriterijų reikšmių aibėje surandami taip vadinami „inkaro taškai“. Kiekvienas inkaro taškas atitinka $f_i^{\mathrm{u}}(X)$. Antra, brėžiama utopinė tiesė (dviejų kriterijų atveju), einanti per inkaro taškus (kaip ir NBI metode). Šio metodo pagrindinis skirtumas nuo NBI yra tas, kad vietoj lygybinių apribojimų naudojami nelygybiniai, todėl NC metodas yra stabilesnis ir lankstesnis. Šio metodo trūkumas tas, kad ne visi gauti sprendiniai yra Pareto optimalūs, kai yra daugiau negu du kriterijai. Tačiau vėlesnėje metodo modifikacijoje (Messac and Mattson 2004) šis trūkumas buvo pašalintas. Ieškodamas sprendinių metodas aprėpia visą kriterijų reikšmių sritį ir tolygiai juos išdėsto Pareto aibėje.

NC ir NBI trūkumas tas, kad būsimų Pareto aibės sprendinių skaičius atitinka utopinės tiesės suskirstymui parinktų svorinių koeficientų skaičių. Taigi priklausomai nuo kriterijų reikšmių aibės ribos pavidalo, atstumai tarp gautų Pareto aibės sprendinių gali šiek tiek skirtis, t. y. sprendiniai gali ne visai tolygiai padengti Pareto aibę.

2.3.3. PP metodas

Physical programming (PP) metodo (Messac 1996) privalumas tas, kad jo pagalba surandami sprendiniai ir tose kriterijų reikšmių aibės srityse, kur funkcijos $f_i(X)$ nėra iškilos (Messac and Mattson 2002). Šiame metode, skirtingai nuo NBI ir NC metodų, nenaudojami svoriniai koeficientai. Čia SP nurodo savo teikiamus prioritetus, suskirstydamas kriterijus į kelias klases (funkcijų klases) (Eddy and Lewis 2001). Optimizavimas remiasi apibendrintos funkcijos, suformuotos iš SP nurodytų prioritetų (funkcijų klasių), minimizavimu. Vėliau metodas buvo modifikuotas taip, kad surasti sprendiniai tolygiai padengtų visą Pareto aibę (Utyuzhnikov *et al.* 2009).

PP metodo trūkumas tas, kad jame yra keli laisvai parenkami parametrai, kurių optimalių reikšmių parinkimas reikalauja žinių apie Pareto aibės formą ir padėtį (Erfani and Utyuzhnikov 2011).

2.3.4. Adaptyvus svertinės kriterijų sumos metodas

Skirtingai nuo 2.3.1–2.3.3 skyreliuose aprašytų metodų, adaptyvus svertinės kriterijų sumos metodas (ASKS) (angl. *Adaptive Weigted Sum Method, AWS*) (Kim and de Weck 2006) ieškodamas tolygiai pasiskirsčiusių sprendinių, automatiškai prisitaiko prie Pareto aibės pavidalo.

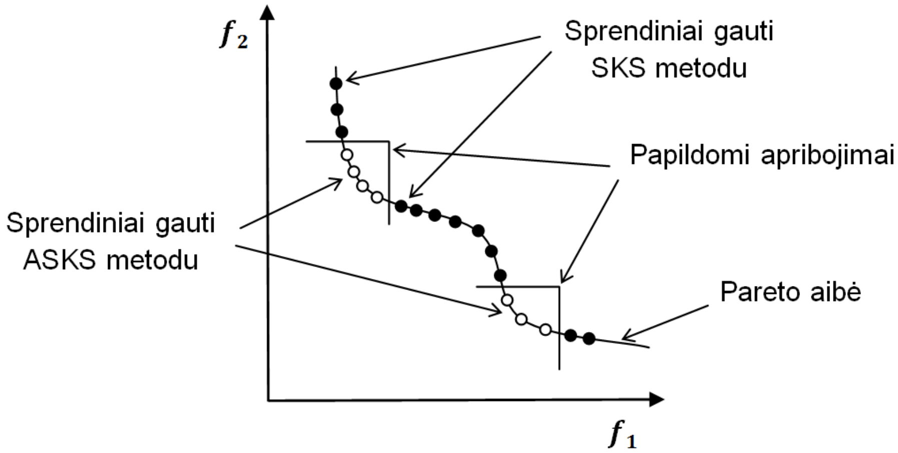
ASKS metodas remiasi SKS metodu, aprašytu 2.2.2 skyrelyje. Žinoma, kad dviejų kriterijų atveju Pareto aibė yra kreivė. Šio metodo idėja yra tokia:

naudojant SKS metodą, Pareto aibėje randami keli sprendiniai ir aibė suskirstoma į intervalus; kiekviename intervale įvedami papildomi apribojimai ir naudojant SKS metodą surandami nauji sprendiniai.

ASKS metodo žingsniai, sprendžiant dviejų kriterijų uždavinius, yra tokie:

1. Sunormuojamos tikslo funkcijos.
2. Naudojant SKS metodą, surandami Pareto aibės sprendiniai esant kelioms skirtingoms svorinių koeficientų reikšmėms.
3. Apskaičiuojami Euklidiniai atstumai tarp kaimyninių sprendinių. Pašalinami persidengiantys Pareto aibės sprendiniai. Šis žingsnis reikalingas, nes naudojant SKS metodą, su skirtingomis svorinių koeficientų reikšmėmis kartais gaunami tie patys Pareto sprendiniai.
4. Pareto aibė suskirstoma į intervalus, kurie sudaromi iš kaimyninių Pareto aibės sprendinių, ir apskaičiuojami intervalų ilgiai.
5. Nustatomi tie intervalai, kuriuose bus ieškoma Pareto sprendinių. Intervale ieškomi sprendiniai tuo atveju, jeigu jo ilgis didesnis už prieš sprendimo procesą nustatytą maksimalų intervalo ilgį ir šis intervalas nėra pašalintas iš paieškos. Jeigu tokių intervalų nėra, sprendimo procesas baigiamas.
6. Kiekviename intervale ieškoma Pareto aibės sprendinių, naudojant SKS metodą su įvestais papildomais nelygybiniais apribojimais, kurie apriboja kriterijų galimų reikšmių aibę.
7. Jeigu intervale nerandama nei vieno Pareto aibės sprendinio, tai šiame intervale vėliau nebus ieškoma sprendinių, t. y. šis intervalas pašalinamas iš paieškos. Einama į 3-ą žingsnį.

SKS ir ASKS metodais gautų sprendinių, sprendžiant dviejų kriterijų minimizavimo uždavinį, išsidėstymas Pareto aibėje pavaizduotas 2.3 pav.



2.3 pav. *SKS ir ASKS metodais gauti sprendiniai dviejų kriterijų atveju*

Toliau yra pateikta ASKS metodo sprendžiamo uždavinio tikslo funkcijos formuluotė.

Normuotos tikslo funkcijos apskaičiuojamos pagal formulę:

$$\hat{f}_i(X) = \frac{f_i(X) - f_i^{\mathrm{u}}(X)}{f_i^{\mathrm{n}}(X) - f_i^{\mathrm{u}}(X)}. \tag{2.13}$$

Dviejų kriterijų atveju SKS metodu sprendžiamas uždavinys formuluojamas taip:

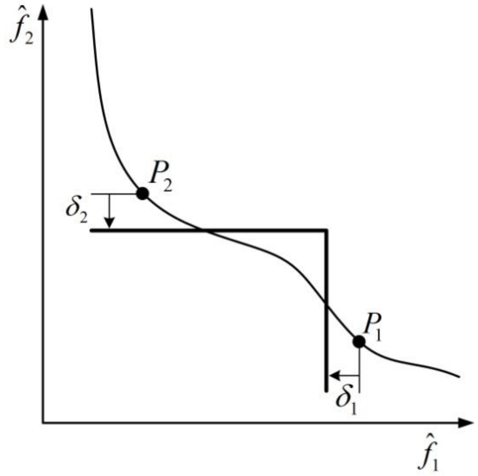
$$\min_{X \in \mathrm{D}} \left(w_1 \hat{f}_1(X) + (1 - w_1) \hat{f}_2(X) \right), \tag{2.14}$$

esant apribojimams:

$$\hat{g}_j(X) \leq 0, \qquad j = 1, 2, \dots, k, \tag{2.15}$$

$$\hat{h}_l(X) = 0, \qquad l = 1, 2, \dots, p, \tag{2.16}$$

čia $w_1 \in [0; 1]$, $\hat{g}_j(X)$ – normuoti uždavinio nelygybiniai apribojimai, $\hat{h}_l(X)$ – normuoti uždavinio lygybiniai apribojimai. Keičiant svorinio koeficiento w_1 reikšmes, gaunami skirtingi Pareto aibės sprendiniai. Tokiu būdu Pareto aibė suskirstoma į intervalus. Tolimesnei sprendinių paieškai pasirinktame intervale pridedami du papildomi nelygybiniai apribojimai. Jie yra lygiagretūs kiekvienai tikslo funkcijos ašiai. Apribojimų atstumai δ_1 ir δ_2 nuo pasirinktų sprendinių nukreipti į vidų link tikslo funkcijų $\hat{f}_1(X)$ ir $\hat{f}_2(X)$ (2.4 pav.).



2.4 pav. *Apribojimų nustatymas ASKS metode*

Taigi kiekviename intervale sprendžiamas toks minimizavimo uždavinys:

$$\min_{X\in \mathrm{D}} \left(w_1 \hat{f}_1(X) + (1 - w_1)\hat{f}_2(X) \right), \tag{2.17}$$

esant apribojimams:

$$\hat{f}_1(X) \le P_1^{\mathrm{x}} - \delta_1, \tag{2.18}$$

$$\hat{f}_2(X) \le P_2^{\mathrm{y}} - \delta_2, \tag{2.19}$$

$$\hat{g}_j(X) \le 0, \qquad j = 1, 2, \dots, k, \tag{2.20}$$

$$\hat{h}_l(X) = 0, \qquad l = 1, 2, \dots, p, \tag{2.21}$$

čia $w_1 \in [0; 1]$, δ_1 ir δ_2 – atstumai nustatomi sprendimų priėmėjo, P_1^{x} ir P_2^{y} – nagrinėjamo intervalo galų *x* ir *y* koordinatės.

Kai kriterijų daugiau negu du, metodo principas išlieka panašus. Trijų kriterijų atveju Pareto aibė tampa paviršiumi ir padengiama tinkleliu, sudarytu iš lopų (angl. *Patch*). Šie lopai gali turėti skirtingą viršūnių skaičių. Daugelio kriterijų atveju Pareto aibė tampa hiperpaviršiumi. Kai viršūnių skaičius yra didesnis už kriterijų skaičių, viršūnės arba jas jungiančios kraštinės nebūtinai bus toje pačioje hiperplokštumoje. Šiuo atveju yra sudėtinga pasirinkti tinkamus nelygybinius apribojimus, todėl metodo autoriai siūlo naudoti lygybinius apribojimus, kas palengvina ir lopų skaidymą ieškant Pareto sprendinių. Lygybiniai apribojimai yra tiesės tarp nadir vektoriaus ir tikėtinų sprendinių, o tikrieji sprendiniai bus ant šių tiesių.

Trijų kriterijų atveju uždavinio tikslo funkcija yra tokia:

$$\mathbf{F}(X) = w_2[w_1 f_1(X) + (1 - w_1) f_2(X)] + (1 - w_2) f_3(X). \qquad (2.22)$$

Daugelio kriterijų atveju, kai kriterijų skaičius m, tikslo funkcijos sudaromos rekursyviai pagal formulę:

$$\mathbf{F}^m(X) = w_{m-1}\mathbf{F}^{m-1}(X) + (1 - w_{m-1}) f_m(X), \text{kai } m \geq 2, \qquad (2.23)$$

čia $\mathbf{F}^1(X) = f_1(X)$.

Tikėtino j-tojo sprendinio vietos vektorius apskaičiuojamas pagal formulę:

$$P^j = \sum_{i=1}^{h} \beta_i N^i, \qquad (2.24)$$

čia β_i – interpoliacijos svorinis koeficientas, $\beta \in [0; 1]$, h – lopo viršūnių skaičius, N^i – i-tosios lopo viršūnės koordinačių vektorius. P^j vektorius normuojamas pagal formulę:

$$\hat{P}_i^j = \frac{P_i^j - f_i^{\mathrm{u}}(X)}{f_i^{\mathrm{n}}(X) - f_i^{\mathrm{u}}(X)}. \qquad (2.25)$$

Taigi ieškant Pareto sprendinių lopuose ASKS metodu sprendžiamas toks optimizavimo uždavinys:

$$\min_{X \in \mathbf{D}} w\hat{\mathbf{F}}(X), \qquad (2.26)$$

esant apribojimams:

$$\frac{(\hat{P}^j - \hat{\mathbf{F}}^{\mathrm{n}}(X))(\hat{\mathbf{F}}(X) - \hat{\mathbf{F}}^{\mathrm{n}}(X))}{\left|\hat{P}^j - \hat{\mathbf{F}}^{\mathrm{n}}(X)\right|\left|\hat{\mathbf{F}}(X) - \hat{\mathbf{F}}^{\mathrm{n}}(X)\right|} = 1, \qquad (2.27)$$

$$\hat{g}_j(X) \leq 0, \qquad j = 1, 2, \dots, k, \qquad (2.28)$$

$$\hat{h}_l(X) = 0, \qquad l = 1, 2, \dots, p, \qquad (2.29)$$

čia $w = -(\hat{P}^j - \hat{\mathbf{F}}^{\mathrm{n}}(X))$ – svorinių koeficientų vektorius, $\hat{g}_j(X)$ – normuoti uždavinio nelygybiniai apribojimai, $\hat{h}_l(X)$ – normuoti uždavinio lygybiniai apribojimai. Normuotas nadir vektorius lygus vienam, t. y. $\hat{\mathbf{F}}^{\mathrm{n}}(X) = (1, 1, \dots, 1)$, utopinis vektorius lygus nuliui, t. y. $\hat{\mathbf{F}}^{\mathrm{u}}(X) = (0, 0, \dots, 0)$ Reiktų pažymėti tai, kad tikrasis normuotas sprendinys $\hat{P}^{j*}$ dažniausiai skiriasi nuo tikėtino sprendinio P^j.

Metodo privalumas yra tas, kad jo pagalba surandami Pareto optimalūs sprendiniai ir tose kriterijų reikšmių aibės srityse, kur funkcijos $f_i(\mathrm{X})$ nėra iškilos. Be to, šis metodas prisitaiko prie Pareto aibės pavidalo: Pareto aibė padengiama sprendiniais su iš anksto nustatytu atstumu tarp jų. Metodo trūkumas yra tas, kad daugelio kriterijų atveju ne visi gauti sprendiniai yra Pareto optimalūs, todėl juos reikia pašalinti, naudojant Pareto filtrą. Be to, kai kriterijų daugiau nei trys, metodo įgyvendinimas yra sudėtingas: dažnai Pareto aibė yra nereguliarios formos, tuo atveju yra sudėtinga padalinti vienodos formos lopais; sudėtinga įvesti nelygybinius apribojimus, o lygybiniai apribojimai stipriai apriboja kriterijų reikšmių aibę, todėl sudėtinga rasti juos tenkinančius sprendinius. Be to, tokie skaičiavimai reikalauja daug skaičiavimo resursų.

Šio metodo modifikaciją, pavadintą *Algorithm of Adjustable Weights* (Steponavičė 2010), pasiūlė A. Žilinsko vadovaujama doktorantė I. Steponavičė. Čia ASKS metodo lopus atitinka simpleksai, o lopų viršūnes – simpleksų viršūnės. Pasiūlytame metode vietoje papildomų apribojimų įvedimo, kaip yra ASKS metode, surandama ilgiausia simplekso kraštinė ir ji dalinama pusiau, tokiu būdu vieną simpleksą suskaidant į du. Toks procesas vyksta iteratyviai, kol Pareto aibė nepadengiama norimu sprendinių skaičiumi arba norimu sprendinių tankumu. Šis metodas paprasčiau įgyvendinamas, tačiau nėra toks tikslus kaip ASKS.

2.4. Sprendinių vertinimo matai

Sprendžiant daugiakriterinius optimizavimo uždavinius, svarbu įvertinti optimizavimo metodus ir palyginti juos tarpusavyje. Tam būtini matai, nustatantys, ar metodo surasti sprendiniai yra Pareto aibės sprendiniai ir kaip tolygiai gauti sprendiniai padengia Pareto aibę.

2.4.1. Apibendrintas atstumas

Apibendrintas atstumas (angl. *Generational Distance*) (Van Veldhuizen and Lamont 1998) buvo pasiūlytas siekiant įvertinti, kaip surastų nedominuojančių vektorių aibės elementai yra nutolę nuo Pareto aibės sprendinių. Apibendrintas atstumas skaičiuojamas pagal formulę:

$$\delta = \frac{\sqrt{\sum_{i=1}^{n'} d_i^2}}{n'}, \tag{2.30}$$

čia n' – yra surastų nedominuojančių sprendinių skaičius, d_i – Euklidinis atstumas tarp kiekvieno surasto sprendinio ir artimiausio Pareto aibės sprendinio. Iš formulės (2.30) akivaizdu, kad jeigu $\delta = 0$, tai visi surasti sprendiniai priklauso Pareto aibei. Būtina pastebėti, kad šis matas gali būti taikomas tik tiems uždaviniams, kur iš anksto tiksliai žinoma Pareto aibė.

2.4.2. Hipertūris

Hipertūris (angl. *Hypervolume*) (Zitzler and Thiele 1998) – dar vienas matas įvertinti gautus sprendinius, pastaruoju metu įgijęs didelį populiarumą. Hipertūrio apibrėžimas, taip pat žinomas kaip S-metrika (angl. *S-metric*) arba Lebego matas (Fleischer 2003) remiasi politopų (Bringmann and Friedrich 2010) arba hiperkūbų (Friedrich *et al.* 2011) tūriais.

Vietoje to, kad ieškoti tikslo funkcijos sprendinių, formuojančių Pareto aibę, yra minimizuojamas hipertūris. Sprendimo procese gautų sprendinių aibės $A \subseteq \mathbf{D}$ hipertūris gali būti apibrėžtas srities, apribotos aibe A ir fiksuotu atramos tašku $r = (r_1, r_2, \dots, r_m) \in R^m$ hipertūriu. Hipertūris skaičiuojamas pagal formulę:

$$\theta(A) = \leftthreetimes\left(\bigcup_{a \in A} [f_1(a), r_1] \times [f_2(a), r_2] \times \cdots \times [f_m(a), r_m]\right), \tag{2.31}$$

čia $\leftthreetimes(S)$ – aibes S Lebego rodyklis, $([f_1(a), r_1] \times [f_2(a), r_2] \times \cdots \times [f_m(a), r_m])$ – m-atis hiperkūboidas sudarytas iš taškų, kurie yra silpnai dominuojantys tašku a, bet nėra silpnai dominuojantys tašku r (Bringmann and Friedrich 2010).

Šio mato tikslas yra nustatyti hipertūrį, apribotą atramos tašku ir gautais nedominuojančias sprendiniais, surastais uždavinio sprendimo eigoje.

2.4.3. Santykinė paklaida

Santykinė paklaida (angl. *Error Ratio*) (Van Veldhuizen 1999) naudojama siekiant nustatyti, kokia procentinė dalis surastų nedominuojančių sprendinių priklauso Pareto aibei. Santykinė paklaida skaičiuojama pagal formulę:

$$\epsilon = \frac{\sum_{i=1}^{n'} e^i}{n'}, \tag{2.32}$$

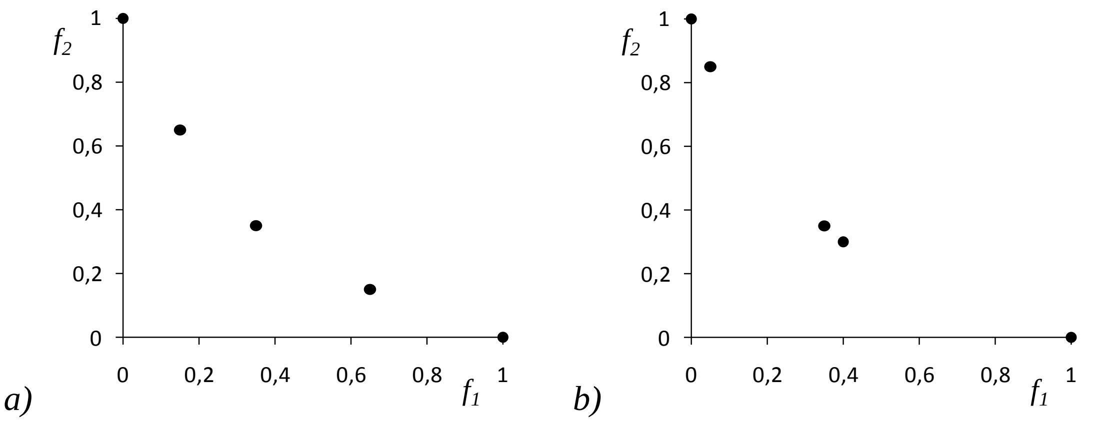
čia n' – surastų nodominuojančių sprendinių skaičius. Jei i-tasis sprendinys priklauso Pareto aibei, tai $e^i = 0$, priešingu atveju $e^i = 1$. Jei $\epsilon = 0$, tai reiškia, kad visi surasti sprendiniai priklauso Pareto aibei.

2.4.4. Pasiskirstymo įvertis

Sprendžiant daugiakriterinį uždavinį svarbus surastų nedominuojančių vektorių pasiskirstymas Pareto aibėje. Kai yra žinoma surastos Pareto aibės „pradžia“ ir „pabaiga“, galima įvertinti, kaip tolygiai pasiskirstę gautieji sprendiniai. Darbe (Schott 1995) buvo pasiūlytas *pasiskirstymo įvertis* (angl. *Spread*), kuris įvertina atstumų svyravimą tarp gretimų nedominuojančių vektorių. Pasiskirstymo įvertis esant dviem kriterijams $f_1(X)$ ir $f_2(X)$ yra skaičiuojamas pagal formulę:

$$\varphi = \sqrt{\frac{1}{n'-1}\sum_{i=1}^{n'}\left(\overline{d} - d_i\right)^2}, \tag{2.33}$$

čia $d_i = \min_j\left(\left|f_1^i(X) - f_1^j(X)\right| + \left|f_2^i(X) - f_2^j(X)\right|\right)$, $i, j = 1, 2, \dots, n'$; $\overline{d}$ – visų d_i vidurkis; n' – surastų nedominuojančių sprendinių skaičius; $f_1^i(X)$, $f_2^i(X)$ – i-tojo sprendinio pirmojo ir antrojo kriterijaus reikšmės. Kai pasiskirstymo įverčio reikšmė $\varphi = 0$, tai visi surasti sprendiniai yra tolygiai pasiskirstę, jei tikslo funkcija nėra trūki. 2.5 pav. yra pavaizduoti du atvejai, kai išsprendus dviejų kriterijų optimizavimo uždavinį, gauta po penkis sprendinius.



2.5 pav. *Sprendinių pasiskirstymas Pareto aibėje:*
a) kai $\varphi = 0$*; b) kai* $\varphi = 0{,}35$

Pirmu atveju, kuris pavaizduotas 2.5 a pav., gauti sprendiniai yra tolygiai pasiskirstę Pareto aibėje, o pasiskirstymo įverčio reikšmę $\varphi = 0$. Antrame atvejyje, kuris pavaizduotas 2.5 b pav., akivaizdu, kad gauti sprendiniai nėra tolygiai pasiskirstę Pareto aibėje, o pasiskirstymo įverčio reikšmė $\varphi = 0{,}35$.

Šis matas gali būti naudojamas sprendžiant dviejų kriterijų optimizavimo uždavinius. Analogiškas matas didesnio kriterijų skaičiaus uždaviniams buvo pasiūlytas darbe (Zhou *et al.* 2006).

2.5. Sprendimų paramos sistemų apžvalga

Sprendžiant daugiakriterinius optimizavimo uždavinius interaktyviai būtina sprendimų paramos sistema. Kalbant apie sprendimų paramos sistemas, reikia pateikti kelių pagrindinių sąvokų apibrėžtis.

Sprendimų priėmimas – tai racionalaus arba neracionalaus alternatyvų pasirinkimo procesas, kurio metu siekiama gauti sprendimų priėmėją tenkinantį sprendinį.

Sprendimų paramos sistema (SPS) – tai kompiuterizuota sistema, padedanti sprendimų priėmėjui suformuoti uždavinius, juos išspręsti ir priimti sprendimus, naudojant informacines technologijas, duomenis, modelius ir žinias.

Sprendimų priėmėjas (SP) – tai žmogus, priimantis sprendimus sprendimų paramos sistemos pagalba, vadovaujantis savo patirtimi ir žiniomis.

2.5.1. Sprendimų paramos sistemų vystymosi istorija

Mokslininkai ir inžinieriai kuria ir tyrinėja sprendimų paramos sistemas jau daugiau nei 50 metų. Didesnį praktinį pritaikymą sprendimų paramos sistemos įgavo atsiradus minikompiuteriams, paskirstytiems skaičiavimas ir operacinėms sistemoms, gebančiomis vienu metu aptarnauti daug vartotojų (angl. *Timesharing Systems*).

1952 metais G. Dantzig tapo tyrinėtoju-matematiku korporacijoje *Rand*, kurioje pradėjo kurti tiesinio programavimo programines realizacijas kompiuteriams. Šešto dešimtmečio viduryje D. Engelbart ir jo kolegos sukūrė sistemą, pavadintą *NLS* (*oNLine System*). Ši sistema palengvino elektroninių bibliotekų kūrimą ir saugojimą bei elektroninių dokumentų paiešką, naudojant hipertekstinę technologiją. NLS sistemoje taipogi buvo numatyta video telekonferencijos galimybė – tai buvo grupinių sprendimų paramos sistemų pirmtakas. J. Forrester buvo įtrauktas į *SAGE* (*Semi-Automatic Ground Environment*) oro gynybos sistemos Šiaurės Amerikai kūrimą, kuri pradėjo veikti 1962 metais. SAGE buvo pirmoji duomenimis grįsta sprendimų paramos sistema. Šių žmonių novatoriški darbai turėjo didelę įtaką kompiuterizuotų sprendimų paramos sistemų vystymuisi (Power 2004). 1960 metais J. C. R. Licklider paskelbė savo idėjas apie interaktyvaus uždavinių sprendimo kompiuteriu svarbą ateityje sprendžiant įvairiausius uždavinius (Licklider 1960). Jis manė, kad kompiuterio ir žmogaus sąveika, sprendžiant praktinius uždavinius, pagerins rezultatų kokybę bei efektyvumą. J. C. R. Licklider savo straipsnyje pateikė orientyrus kompiuterių tyrinėjimams keliems dešimtmečiams į priekį.

Dvidešimto amžiaus septintame dešimtmetyje mokslininkai pradėjo naudoti kompiuterinius kiekybinius modelius sprendimų priėmime ir planavime (Raymond 1966; Turban 1967; Urban 1967; Holt and Huber 1969). R. L. Ferguson ir C. H. Jones paskelbė pirmojo eksperimentinio tyrimo rezultatus, gautus naudojant kompiuterinę sprendimų paramos sistemą. Jie tyrė gamybos proceso planavimą naudojant IBM 7094 (Ferguson and Jones 1969).

Persilaužimas sprendimų paramos sistemų vystymesi įvyko paskelbus M. S. Scott Morton disertaciją (Scott Morton 1967). Jo tyrimai aprėpė interaktyvios modeliais grįstos paramos sistemos projektavimą, sukūrimą ir testavimą. M. S. Scott Morton tyrinėjo, kaip kompiuteriai ir analitiniai modeliai gali pagelbėti vadybininkams daryti svarbius verslo sprendimus planuojant gamybą.

1970 metais J. D. C. Little apibrėžė kriterijus modelių ir sistemų kūrimui, siekiant palengvinti sprendimų priėmimą valdyme. Jis pasiūlė šiuos kriterijus: stabilumas, valdymo patogumas, paprastumas, užbaigtumas (Little 1970). Šie kriterijai lieka svarbūs kuriant ir šiuolaikines SPS.

1975 metais S. Alter paskelbė savo disertaciją (Alter 1975), kuri turėjo didelę įtaką teorijos apie sprendimų paramos sistemas vystymuisi. S. Alter darbas praplėtė verslo ir vadybos sprendimų paramos sistemų taikymo sritį. Tęsiant savo tyrimus, išanalizavęs 56 SPS, jis pasiūlė sprendimų paramos sistemų klasifikaciją (Alter 1980) jas suskirstydamas į septynias grupes:

- **Dokumentų saugyklos sistemos.** Tokios sistemos skirtos pasiekti duomenų elementams.
- **Duomenų analizės sistemos.** Tokiose sistemose duomenys yra analizuojami kompiuteriniais įrankiais, pritaikytais tam tikriems uždaviniams.

- **Skaičiavimo modeliais grįstos sistemos.** Tokiose sistemose skaičiuojamos galimų įvykių pasekmės.
- **Analitinės informacinės sistemos.** Tokios sistemos leidžia prieiti prie kelių su sprendžiamu uždaviniu susietų duomenų bazių ir modelių.
- **Parodomaisiais modeliais grįstos sistemos**. Tokiose sistemose naudojant sukurtus modelius, įvertinamos įvykių pasekmės.
- **Optimizavimo modeliais grįstos sistemos.** Tokiose sistemose pateikiami nurodymai randant optimalius sprendinius.
- **Pasiūlymais grįstos sistemos.** Tokiose sistemose yra kelių žingsnių vedlys, padedantis priimti sprendimą gerai struktūrizuotuose ir gerai žinomuose uždaviniuose.

1978 metais P. G. W. Keen ir M. S. Scott Morton išleido pirmąjį vadovėlį apie SPS projektavimą, kūrimą, įvertinimą ir tobulinimą (Keen and Scott Morton 1978). Tai tapo svarbia mokymo priemone verslo mokyklose.

Aštuntame dešimtmetyje atsirado daugiau ir kitų svarbių publikacijų, skirtų valdymo sistemoms, strateginiam planavimui ir sprendimų paramos sistemoms (Sprague and Watson 1979).

Pirmoji tarptautinė konferencija, skirta sprendimų paramos sistemoms, įvyko 1981 Atlantoje, JAV.

R. D. Hackathorn ir P. G. W. Keen suskirstė sprendimų paramos sistemas į tris skirtingas klases: vieno vartotojo SPS, grupinio vartojimo SPS, organizacijos SPS (Hackathorn and Keen 1981).

R. H. Jr. Sprague ir E. D. Carlson išleista knyga *Building Effective Decision Support Systems* buvo šios srities vienas iš svarbiausių leidinių (Sprague and Carlson 1982). Knygoje aprašytos sprendimų paramos sistemų sudedamos dalys: duomenų bazė, modelių bazė, sąryšio su vartotoju modulis ir valdymo programinė įranga. Knygoje išdėstomi praktiniai lengvai suprantami patarimai, kaip organizacijoms kurti SPS.

Devintame dešimtmetyje susidomėjimas sprendimų paramos sistemomis labai išaugo, atsirado su SPS susijusių studijų dalykų universitetuose ir kitose organizacijose, kas dar labiau išplėtė šią sritį. To pasekmėje buvo sukurta įvairiausių SPS, kurios buvo pritaikytos karinėje pramonėje, versle, statyboje, transporto valdyme, žemės ūkyje ir kitose srityse, kas ženkliai padidino sprendimų bei darbo našumą, automatizavo kai kuriuos procesus bei palengvino žmonių darbą.

Per paskutiniuosius du dešimtmečius kuriamos SPS pasižymėjo tuo, kad naudodavo įvairiausius kiekybinius modelius, buvo sujungtos su didelėmis duomenų bazėmis bei palaikė grupinius uždavinių sprendimus (Power 2007).

Baigiantis praėjusiam tūkstantmečiui pradėtos sparčiai kurti saitynu grįstos SPS. 1998 metais D. J. Power apibrėžė saitynu grįstą SPS, kaip kompiuterizuotą sistemą, kurioje „bendravimas“ su sprendimų priėmėju vyksta naudojant interneto naršyklę. Serveris, kuriame yra patalpinta SPS, yra susietas su sprendimų priėmėjo tinklu naudojant TCP/IP protokolą (Power 1998).

Toliau yra pateikiama viena iš pagrindinių šiuolaikiškų SPS klasifikacijų (Power 2004). SPS yra suskirstytos į penkias grupes:

- **Modeliais grįstos SPS.** Tokiose sistemose pagrindą sudaro finansiniai, optimizavimo arba imitaciniai modeliai ir metodai. Paprasti kiekybiniai modeliai užtikrina pakankamą funkcionalumo lygį. Modeliais grįstos SPS naudoja sprendimų priėmėjo nurodomus duomenis ir modelio parametrus. Šiose SPS nėra reikalingos didelės duomenų bazės.
- **Duomenimis grįstos SPS.** Tokiose sistemose akcentuojamas priėjimas prie įmonėje sukauptų vidinių duomenų bei prie nuolat atsirandančių išorinių duomenų ir šių duomenų valdymas. Paprasta failų sistema, prieinama užklausų pagalba, užtikrina pakankamą funkcionalumo lygį. Didelės duomenų saugyklos ir analitinis jų apdorojimas realiu laiku (angl. *On-line Analytical Processing, OLAP*) užtikrina aukščiausią

funkcionalumo lygį.

- **Komunikacija grįstos SPS.** Tokiose sistemose naudojamas tinklas ir komunikacijų technologijos siekiant palengvinti bendradarbiavimą ir bendravimą priimant sprendimus. Jos užtikrina kelių sprendimų priėmėjų bendradarbiavimą, sprendžiant iškeltą uždavinį. Komunikacijos technologijos – dominuojantis SPS architektūros komponentas. Tarp sistemos įrankių yra grupinio darbo programinė įranga, video konferencijos ir kompiuterizuotos skelbimų lentos. Pastaraisiais metais padidėję interneto greičiai žymiai praplėtė sinchronine komunikacija grįstų SPS galimybes.
- **Dokumentais grįstos SPS.** Tokios sistemos naudoja kompiuterizuotas saugyklas ir apdorojimo technologijas siekiant užtikrinti dokumentų paiešką ir analizę. Didžiules duomenų bazes sudaro skenuoti dokumentai, hipertekstiniai dokumentai, vaizdai, garsai, video failai. Naudojant dokumentais grįstą SPS, galima valdyti nestruktūrizuotą informaciją, saugomą įvairiuose elektroniniuose formatuose. Paieškos sistemos pagalba SPS duomenys susiejami tarpusavyje. Pastaruoju metu, naudojant saityną, galima prieiti prie didelio dokumentų skaičiaus, kas palengvina dokumentais grįstų SPS kūrimą bei vystymą.
- **Žiniomis grįstos SPS.** Tokiose sistemose yra integruotas ekspertinis modulis, galintis pasiūlyti veiksmus arba rekomenduoti sprendimus sprendimų priėmėjams. Ekspertinis modulis – tai įrankis, susidedantis iš žinių apie konkrečią sritį, išmanantis šios srities uždavinius ir turintis patirtį juos sprendžiant. Pastaruoju metu ekspertinių sistemų ir reliacinių duomenų bazių apjungimas saityno pagalba praplėtė žiniomis grįstų SPS naudojimą.

Reiktų pabrėžti, kad dažnai tą pačią SPS negalima griežtai priskirti tik vienai grupei. Ji gali turėti kelių grupių savybes.

Iš pateiktos klasifikacijos galima daryti išvadą, kad SPS yra sudėtinga struktūra, sudaryta iš kelių tarpusavyje susijusių komponentų. Kuriant SPS, autoriams reikia išmanyti ir tarpusavyje derinti duomenų bazes, dirbtinį intelektą, žmogaus ir kompiuterio sąveiką, imitacinius modelius, optimizavimo metodus, programinę įrangą, telekomunikaciją (Ravindranath 2009).

Kalbant apie SPS, reiktų nesumenkinti sprendimų priėmėjo vaidmens. Nuo jo patirties, atliekamų veiksmų, pasirenkamų sprendimo kelių priklauso galutinis sprendžiamos problemos rezultatas, todėl čia sprendimų priėmėjo vaidmuo yra labai didelis ir ypač svarbus.

Mokslinių publikacijų skaičius SPS srityje sparčiai didėja. Galima pažymėti tokius kelis svarbius naujus darbus (Mir and Quadri 2009; Liu *et al.* 2010; Niu *et al.* 2009; Power and Sharda 2009) bei darbų rinkinius (Burstein and Holsapple 2008; Jao 2010), kur yra sukoncentruota, tai kas pasiekta šioje srityje pastaraisiais metais.

Lietuvoje SPS sričiai irgi atkreipiamas nemenkas dėmesys. E. K. Zavadskas ir jo kolegos jau daugelį metų plėtoja sprendimų paramos sistemų teorinę bei praktinę dalį pastatų statybos ir jų gyvavimo proceso srityse bei su tuo susijusiose srityse. Jie yra paskelbę keletą monografijų šia tematika: „Sprendimų paramos sistemos statyboje“ (Zavadskas *et al.* 1999), „Internetinė sprendimų parama“ (Kaklauskas and Zavadskas 2002); taip pat mokomųjų knygų: „Sprendimų paramos sistema statyboje taikant daugiakriterinius sintezės metodus“ (Šarka 2008), „Intelektinė ir biometrinė sprendimų parama“ (Kaklauskas and Zavadskas 2010). Šie mokslininkai yra sukūrę kelias SPS, kurios detaliai nagrinėjamos straipsniuose (Zavadskas and Turskis 1997; Kaklauskas *et al.* 2005; Trinkūnienė 2006). Sukurtos sistemos yra plėtojamos, tobulinamos ir jų pagrindų kuriamos naujos.

G. Dzemyda ir V. Šaltenis yra sukūrę daugiakriterinę SPS (Dzemyda and Šaltenis 1994), kurioje įgyvendinti trys sprendimų priėmimo metodai.

D. Dzemydienė taipogi nagrinėja sprendimų priėmimo bei SPS mokslo sritį Lietuvoje. Savo paskelbtuose straipsniuose tyrinėja SPS kūrimo ir taikymo ypatumus skirtinguose socialinio gyvenimo ir darnaus vystymosi srityse (Dzemydienė 1996; Dzemydienė 2000; Dzemydienė and Maskeliūnas 2008). 2006 metais išleistoje monografijoje (Dzemydienė 2006) nemažai dėmesio skiriama SPS raidos ypatumams, projektavimo metodams ir priemonėms, SPS taikymo aspektams. D. Dzemydienė kartu su savo auklėtiniais yra sukūrusi keletą SPS (Dzemydienė 1998; Dzemydienė and Kazemikaitienė 2005; Dzemydienė and Naujikienė 2006; Dzemydienė and Dzindzalieta 2009).

Pastaraisiais metais susidomėjimas Lietuvoje SPS dar padidėjo. Galima išskirti kelis darbus, skirtus SPS taikymams. Mokslininkų kolektyvas, vadovaujamas E. K. Zavadsko ir G. Dzemydos, sukūrė saitynu grįstą biometrinę SPS, skirtą vartojo emocinės būklės ir darbo produktyvumo bei nuovargio analizei (Kaklauskas *et al.* 2008). Ši sistema naudoja informaciją apie pelytės judinimo greitį, mygtukų spragtelėjimo dažnį, žymeklio padėtį ekrane ir kt. K. Lapin savo tyrimuose prisideda prie SPS *SKY-Scanner*, skirtos lėktuvų pakilimo ir nutupimo laiko kontrolei, tobulinimo (Lapin 2010; Lapin *et al.* 2011). P. Danėnas ir G. Garšva sukūrė SPS, skirtą paskolų rizikos įvertinimui (Danėnas and Garšva 2011). V. Stasytytė savo disertacijoje (Stasytytė 2011) pristatė investicijų portfelio SPS, padedančią formuoti investicijų portfelio investavimo strategijas kapitalo rinkoje pasirinktose vertybinių popierių rinkose.

SPS įvairovė ir pritaikymas aprėpia įvairiausias verslo ir mokslo sritis. Toliau disertacijoje išsamiau išnagrinėtos daugiakriterinio optimizavimo sprendimų paramos sistemos.

2.5.2. Daugiakriterinio optimizavimo sprendimų paramos sistemos

Be ankstesniame skyrelyje minėtų SPS, šiuo metu kuriamos ir tiriamos sistemos, skirtos daugiakriteriniam sprendimų priėmimui. Daugiakriterinis sprendimų priėmimas – tai uždavinių, turinčių daugiau negu vieną kriterijų sprendimas. Daugiakriterinis sprendimų priėmimo procesas gali būti suskirstytas į dvi grupes (Zavadskas *et al.* 2006; Turskis 2008):

- Daugiakriterinis sprendimų priėmimas remiantis alternatyvom (angl. *Multiple Attribute Decision Making, MADM*).
- Daugiakriterinio optimizavimo sprendimų priėmimas (angl. *Multiple Objective Decision Making, MODM*).

Pirmu atveju sprendimo priėmėjo tikslas – pasirinkti vieną iš baigtinės aibės alternatyvų (sprendinių). Tam yra sukurti įvairūs metodai (Figueira *et al.* 2005), kurie surikiuoja arba suklasifikuoja alternatyvas, atsižvelgiant į kelis sprendimo priėmėjo prioritetus bei poreikius. Antru atveju sprendiniai nėra žinomi ir sprendimo priėmėjo tikslas yra surasti sprendinį arba sprendinius, kurie atitiktų SP reikalavimus. Tam tikslui naudojami daugiakriterinio optimizavimo metodai. Čia formuojamas tiesinių arba netiesinių funkcijų rinkinys, atsižvelgiant į uždavinyje siekiamus tikslus. Po to šios funkcijos yra optimizuojamos. Vėliau SP pasirenka jį tenkinantį sprendinį. Priklausomai nuo pasirinkto optimizavimo metodo sprendiniai ir gautų sprendinių skaičius gali būti skirtingi. Toliau darbe detaliau nagrinėjamos antros grupės SPS.

Pradedant nuo dvidešimto amžiaus septinto dešimtmečio yra sukurta daug daugiakriterinių SPS. Kuriant SPS, mokslininkai ir specialistai siekia ištirti ir išanalizuoti skirtingus uždavinius verslo, finansų, resursų paskirstymo, žemės ūkio, švietimo srityse (Eom *et al.* 1993; Weistroffer *et al.* 2005).

Pirmosios SPS, skirtos daugiakriteriniams matematinio programavimo uždaviniams spręsti, buvo sukurtos aštuntame dešimtmetyje (Dyer 1973; Wallenius and Zionts 1976). Tačiau dėl tuometinės kompiuterinių resursų ir technologijų stokos, sukurtos SPS neturėjo grafinės sąsajos. Jos buvo gana primityvios, lyginant su šiuolaikinių kompiuterių galimybėmis bei rėmėsi tiesinio programavimo (angl. *Linear Programming*) ir tikslo programavimo (angl. *Goal Programming*) metodais (Wallenius 1991). Vystantis technologijoms bei atsirandant naujiems optimizavimo metodams, buvo kuriamos naujos sistemos. Gerai žinomos yra šios:

- VIG (Korhonen 1987);
- TOMMIX (Antunes *et al.* 1992);
- DINAS (Ogryczak *et al.* 1992);
- MOLP-16 (Vassilev *et al.* 1993);
- MOIP (Vassilev *et al.* 1997);
- MOMILP (Alves and Climaco 2000);
- PROMOIN (Caballero *et al.* 2002);
- MultiGen (Mirrazavi *et al.* 2003);
- MultiDecision-1 (Vassilev *et al.* 2006);
- MultiDecision-2 (Vassilev *et al.* 2008).

Žinomiausios daugiakriterinio optimizavimo SPS, skirtos netiesiniams daugiakriterinio optimizavimo uždaviniams spręsti, yra šios:

- CAMOS (Osyczka 1988; Osyczka 1992);
- DIDAS ir kelios jos modifikacijos (Lewandowski *et al.* 1989);
- MONP-16 (Vassilev *et al.* 1993);
- LBS (Jaszkiewicz and Slowinski 1994);
- NIMBUS (Miettinen *et al.* 1996);
- WWW-NIMBUS (Miettinen and Mäkelä 2000) (pritaikyta saitynui);
- IND-NIMBUS (Ojalehto *et al.* 2007) (atnaujinta NIMBUS versija).

Dauguma išvardintų daugiakriterinio optimizavimo SPS remiasi interaktyviais metodais, detaliai aprašytais darbuose (Miettinen 1999; Branke *et al.* 2008). Sistemos DIDAS, VIG, CAMOS, DINAS, LBS remiasi atramos taško ar krypties interaktyviais metodais. O sistemos MOLP-16, MONP-16, NIMBUS, PROMOIN, MultiDecision-1 remiasi klasifikavimu pagrįstais metodais. MultiDecision-2 pasižymi tuo, kad joje įgyvendinti skirtingų tipų metodai.

2.5.3. Sprendimų paramos sistemos kūrimo principai

Akivaizdu, kad sprendžiant kelių kriterijų optimizavimo uždavinius daugiakriterinio optimizavimo metodais, randami keli Pareto optimalūs sprendiniai. Todėl sprendžiant tokius uždavinius, sprendimų priėmėjas turi pasirinkti vieną iš jam priimtiniausių. Kai sprendimas vyksta keliais etapais, SP nukreipia sprendimo eigą, pasinaudojant savo žiniomis apie sprendžiamą uždavinį ir atsižvelgiant į siekiamus tikslus. Kad būtų galima interaktyviai spręsti daugiakriterinį optimizavimo uždavinį ir dalyvauti sprendimo proceso eigoje, būtina sukurti kompiuterizuotą sprendimų paramos sistemą. Kuriant SPS, reikia atsižvelgti ne tik į prieinamus kompiuterių skaičiavimo resursus ar optimizavimo metodus, bet ir į tai, kad SPS grafinę vartotojo sąsają turi būti patogi, lanksti ir draugiška.

Gali būti išskirtos dvi SPS grupės: bendro pobūdžio sistemos ir konkretaus uždavinio sistemos (Vassilev *et al.* 2008). Pirmos grupės SPS skirtos įvairių daugiakriterinių optimizavimo uždavinių sprendimui. Dažniausiai juose yra įgyvendinti tik vienas ar keli tos pačios grupės metodai dėl šių priežasčių:

- Skirtingi metodai sukurti skirtingų tipų uždavinių sprendimui.
- Skirtinguose metoduose naudojami skirtingi duomenų tarp sprendimų priėmėjo ir SPS perdavimo būdai, kas sukelia didelių sunkumų, kuriant vartotojo grafinės sąsajos modulį.
- Skirtinguose metoduose naudojamos skirtingos uždavinių sprendimo strategijos.

- SPS kūrėjai yra suinteresuoti įgyvendinti savo sukurtą metodą.

Antros grupės SPS yra skirtos vieno ar kelių konkrečių uždavinių sprendimui. Tokia interaktyvi SPS, skirta daugiakriterinio optimizavimo pašarų sudėties sudarymo uždavinio sprendimui, pasiūlyta ir analizuota darbe (Dzemyda and Petkus 2001). Pagrindinis tokių sistemų privalumas tas, kad grafinė vartotojo sąsaja yra pritaikyta šių uždavinių sprendimui. Tokiose sistemose dažnai įgyvendinti keli skirtingų grupių optimizavimo metodai. Norint tokią SPS pritaikyti kito uždavinio sprendimui, tenka modifikuoti jos grafinę sąsają. Kartais tokios SPS yra bendro pobūdžio sistemų sudedamosios dalys.

Pagrindinės daugiakriterinio optimizavimo SPS sudedamosios yra šios (Miettinen 1999):

- sprendimo modelis;
- optimizavimo paketas;
- sąsają tarp modelio, optimizavimo paketo ir sprendimų priėmėjo.

Taip pat svarbi dalis yra uždavinio arba sprendimo proceso lygiagretinimo galimybė, kas gali žymiai paspartinti daug skaičiavimo resursų reikalaujančio uždavinio sprendimą.

2.6. Antrojo skyriaus apibendrinimas ir išvados

Šiame skyriuje yra atlikta daugiakriterinių optimizavimo uždavinių sprendimo metodų analizė. Pagal sprendimų priėmėjo dalyvavimą daugiakriterinių optimizavimo uždavinių sprendime išskirtos kelios metodų grupės: nepirmenybės, posteriori, priori ir interaktyvūs metodai. Kai sprendimų priėmėjas dalyvauja sprendimo procese interaktyviai, tai jis gali koreguoti prioritetus ir siekiamus tikslus uždavinio sprendimo eigoje. Tai įgalina išspręsti sudėtingus uždavinius su daug kriterijų ir apribojimų.

Vienas iš tokių metodų yra svertinės kriterijų sumos metodas. Dėl savo paprastumo ir sprendimų priėmėjo prioritetų nustatymo svorinių koeficientų pagalba patogumo, bei dėl nesudėtingo įgyvendinimo svertinės kriterijų sumos metodas išlieka dažnai naudojamas praktiniams uždaviniams spręsti. Be to, metodu galima rasti visus Pareto aibės sprendinius, kai kriterijų tikslo funkcijos yra iškilos. Šis metodas gali būti naudojamas interaktyviam uždavinių sprendimui. Tačiau gaunami sprendiniai nebūtinai bus tolygiai pasiskirstę Pareto aibėje. Be to, jei kriterijų tikslo funkcijos nėra iškilos, ne visi Pareto aibės sprendiniai bus rasti. Todėl nėra tikslinga apsiriboti vien šio metodo taikymu sprendžiant daugiakriterinius optimizavimo uždavinius.

Šiame skyriuje apžvelgti Pareto aibės tolygaus padengimo metodai. Jie yra svarbūs, nes sprendimų priėmėjas gali gauti sprendinius iš visos Pareto aibės, o ne tik iš jos dalies. Vienas iš tokių metodų, adaptyvus svertinės kriterijų sumos metodas remiasi svertinės kriterijų sumos metodu. Naudojant šį metodą randami pakankamai tolygiai išsidėstę sprendiniai iš visos Pareto aibės, tačiau metodas nėra interaktyvus, koks yra svertinės kriterijų sumos metodas. Būtina sukurti daugiakriterinių optimizavimo uždavinių sprendimo būdą, apjungiantį svertinės sumos ir adaptyvų svertinės sumos metodus, kuris būtų interaktyvus bei užtikrintų gaunamų sprendinių tolygų pasiskirstymą Pareto aibėje.

Sprendžiant daugiakriterinius optimizavimo uždavinius interaktyviai būtina sprendimų paramos sistema. Šiame skyriuje apžvelgtos dviejų tipų sprendimų paramos sistemos skirtos: daugiakriteriniam sprendimų priėmimui remiantis alternatyvom ir daugiakriterinio optimizavimo sprendimų priėmimui. Nesudėtingiems daugiakriteriniams optimizavimo uždaviniams spręsti gali būti naudojamos jau sukurtos bendro pobūdžio sistemos. Tačiau sudėtingiems praktiniams uždaviniams jų galimybių nepakanka, todėl tokiems uždaviniams būtina sukurti atskirą sprendimų paramos sistemą, kuri gali būti pritaikoma ir panašių uždavinių grupei.

3

Interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas

Šiame skyriuje aprašomas darbe siūlomas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimų būdas. Pateikiamas sprendimų paramos sistemos modelis, pagrindžiamas programinės įrangos pasirinkimas. Detaliai aprašoma sukurta sprendimų paramos sistema, analizuojamas sprendimo proceso lygiagretinimas, siūlomos sprendimo strategijos, kai daugiakriterinis optimizavimo uždavinys sprendžiamas, pasitelkus kompiuterių klasterį. Taip pat aprašomas sukurtos sistemos bei pasiūlyto interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo pritaikymas praktiniam pašarų sudėties sudarymo uždaviniui spręsti.

Skyriuje pateikti rezultatai publikuoti darbuose (A 1, A 3, A 4 ir B 1). Eksperimentinių tyrimų rezultatai pateikti 4 skyriuje.

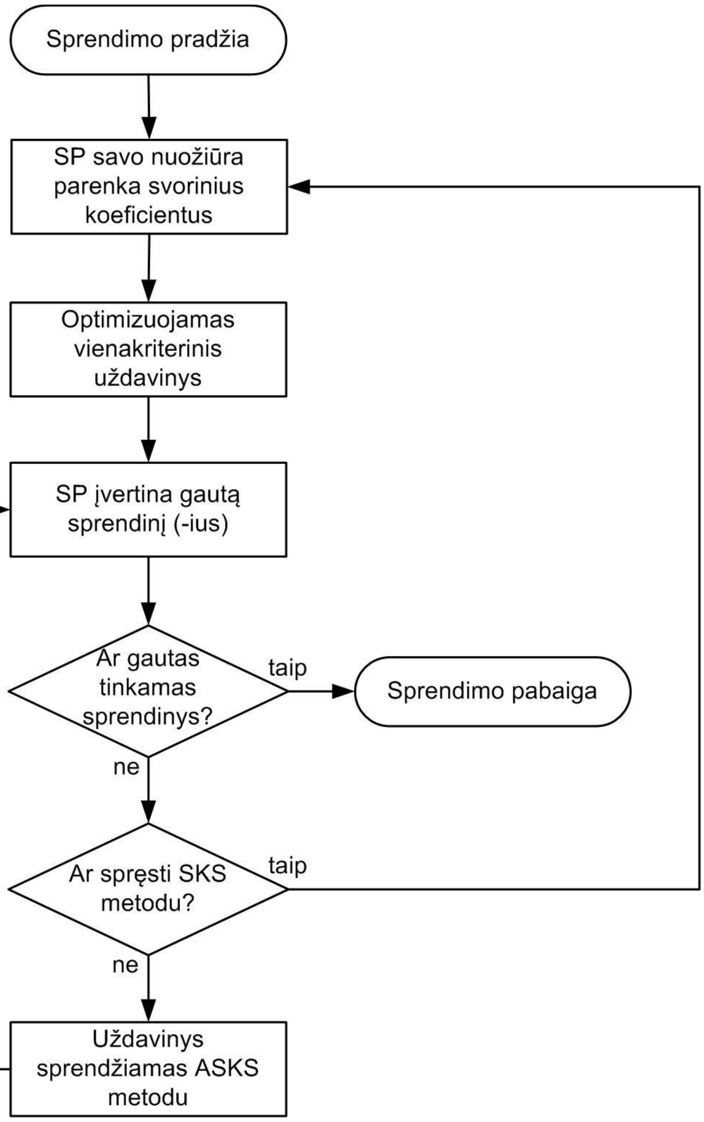
3.1. Siūlomas interaktyvus sprendimo būdas

Šiame poskyryje pateikiamas siūlomas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas, apjungiantis svertinės kriterijų sumos ir adaptyvų svertinės kriterijų sumos metodus. Siūlomo interaktyvaus sprendimo būdo žingsniai yra tokie:

1. Suformuotas optimizavimo uždavinys sprendžiamas svertinės kriterijų sumos metodu (SKS), aprašytu 2.2.2 skyrelyje:
   1.1. Sprendimų priėmėjas savo nuožiūra pasirenka svorinių koeficientų $w = (w_1, w_2, \ldots, w_m)$ reikšmes;
   1.2. Optimizuojamas suformuotas vienakriterinis uždavinys ir randamas sprendinys;
   1.3. SP gavęs sprendinį, jį įvertina. Jei SP gavo jį tenkinantį sprendinį, sprendimo procesas sustabdomas, priešingu atveju galimos dvi alternatyvos: (a) atsižvelgiant į gautus rezultatus, SP pakeičia svorinių koeficientų $w = (w_1, w_2, \ldots, w_m)$ reikšmių rinkinį ir einama į 1.2. punktą; (b) einama į 2-ą punktą.
2. Uždavinys sprendžiamas adaptyviu svertinės kriterijų sumos metodu (ASKS), aprašytu 2.3.4 skyrelyje.
3. Visi sprendiniai, gauti ASKS metodu, yra išsaugomi.
4. SP peržiūri ir įvertina gautus sprendinius. Jeigu tarp gautų sprendinių SP randa jį tenkinantį, sprendimo procesas sustabdomas. Priešingu atveju SP pasirenka vieną ar kelis gautus sprendinius ir pakeičia svorinių koeficientų $w = (w_1, w_2, \ldots, w_m)$ reikšmių rinkinius, einama į 1.2. punktą.

Siūlomo interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo blokinė schema pavaizduota 3.1 pav.

Kaip minėta 2.3.4 skyrelyje, ASKS remiasi SKS metodu. ASKS integravimas į pasiūlytą sprendimo būdą užtikrina gautų sprendinių tolygų pasiskirstymą Pareto aibėje. Taigi SP turės daugiau skirtingų alternatyvių sprendinių, kurių galėjo negauti pats formuodamas užduotis, t. y. parinkdamas svorinių koeficientų $w = (w_1, w_2, \dots, w_m)$ reikšmių rinkinius. Tačiau ASKS metodu gautų sprendinių skaičius neturi būti per didelis, kad SP sugebėtų per priimtiną laiką juos įvertinti ir pasirinkti tinkamiausius, nes kitaip sprendimo būdas prarastų savo interaktyvumą.



3.1 pav. *Siūlomo interaktyvaus sprendimo būdo blokinė schema*

Siūlomas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas grindžiamas tuo, kad SP gali dalyvauti ne tik sprendimų priėmime, kaip būtų tuo atveju, jeigu uždavinys būtų sprendžiamas tik ASKS metodu, bet ir sprendimo proceso eigoje, savo nuožiūra parinkdamas svorinių koeficientų $w = (w_1, w_2, \dots, w_m)$ reikšmes, taip išnaudojant savo ekspertinę patirtį, žinias apie uždavinio specifiką ir sprendimą nukreipiant norima linkme.

Siūlomas interaktyvus sprendimo būdas, apjungiantis SKS ir ASKS metodus, užtikrina gaunamų Pareto optimalių sprendinių tolygų pasiskirstymą, ko nėra kituose interaktyviuose metoduose.

3.2. Sprendimo proceso lygiagretinimas

Kelių procesorių panaudojimas suteikia daugiau galimybių, sprendžiant sudėtingus daugiakriterinius optimizavimo uždavinius interaktyviai. Galimi du optimizavimo proceso lygiagretinimo variantai:

1. Optimizavimo algoritmo lygiagretinimas.
2. Sprendimo proceso paskirstymas tarp procesorių.

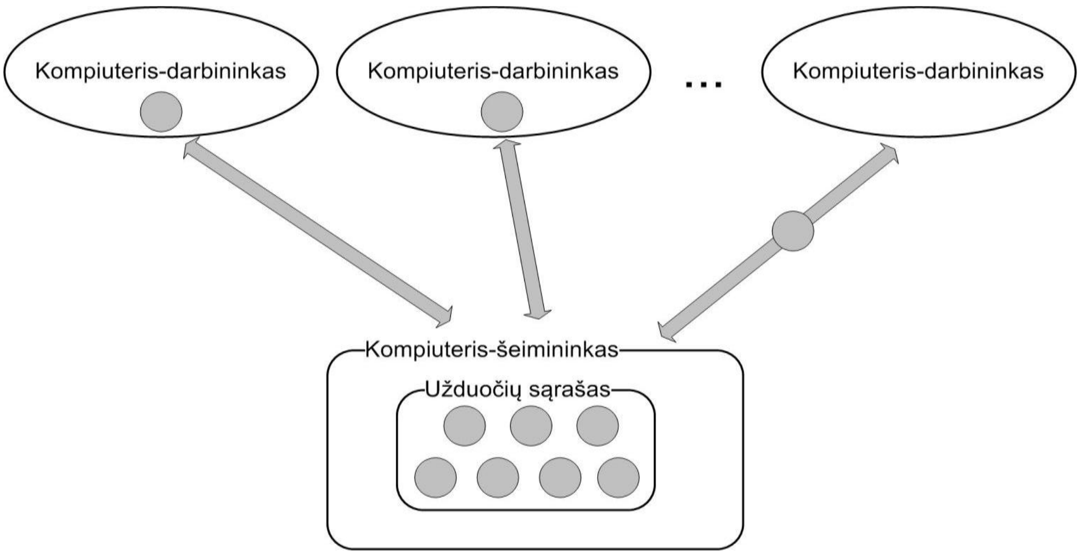
Pirmuoju atveju optimizavimo algoritmo pagrindu kuriama jo lygiagreti versija. Šio būdo privalumas tas, kad įmanoma išvengti procesorių prastovų – procesoriai tarpusavyje apsikeičia duomenimis optimizavimo proceso metu. Tačiau dėl kiekvieno metodo ypatumų ne visi lygiagretūs algoritmai gali būti efektyvūs. Taip pat svarbi algoritmo programinė realizacija. Nuo turimų kompiuterinių resursų, programuotojo kompetencijos ir architektūros supratimo priklauso lygiagretaus algoritmo efektyvumas.

Sprendžiant daugiakriterinius optimizavimo uždavinius interaktyviai, sprendimo procesas dažniausiai yra iteracinis: sprendžiama eilė vienakriterinių optimizavimo uždavinių, gaunama eilė sprendinių, iš kurių SP pasirenka jį tenkinantį. 3.1 poskyryje pasiūlytu interaktyviu daugiakriterinių optimizavimo uždavinių sprendimo būdu vienakriterinį optimizavimo uždavinį gali spręsti atskiras procesorius nepriklausomai nuo kitų. Todėl šiame darbe pasirinktas antrasis lygiagretinimo variantas. Sprendimo proceso paskirstymo tarp procesorių privalumai yra tokie:

- sprendžiant vienakriterinį optimizavimo uždavinį, procesoriai neturi keistis duomenimis tarpusavyje;
- vienakriterinis optimizavimo uždavinys sprendžiamas efektyviais, praktikoje patikrintais metodais;
- nereikia sinchronizuoti procesorių darbo (procesorių skaičiavimo laikas gali skirtis), todėl galima naudoti heterogeninį kompiuterių klasterį;
- paskirstymo procesas lengviau programiškai realizuojamas negu optimizavimo algoritmo lygiagreti versija.

Sprendimų priėmėjų patirtis bei jų užduočių formavimo (gautų sprendinių įvertinimo, naujų svorinių koeficientų reikšmių parinkimo) laikas gali skirtis, todėl konkretaus uždavinio sprendimui reiktų atsakingai rinktis kompiuterių (procesorių) skaičių.

Pasiūlytoje SPS sprendimo proceso lygiagretinimui buvo pasirinktas šeimininkas-darbininkai (angl. *master-slaves*) modelis (Čiegis 2005). Modelio schema pavaizduota 3.2 pav.



3.2 pav. *Sprendimo proceso lygiagretinimo principas pagal šeimininko-darbininkų modelį*

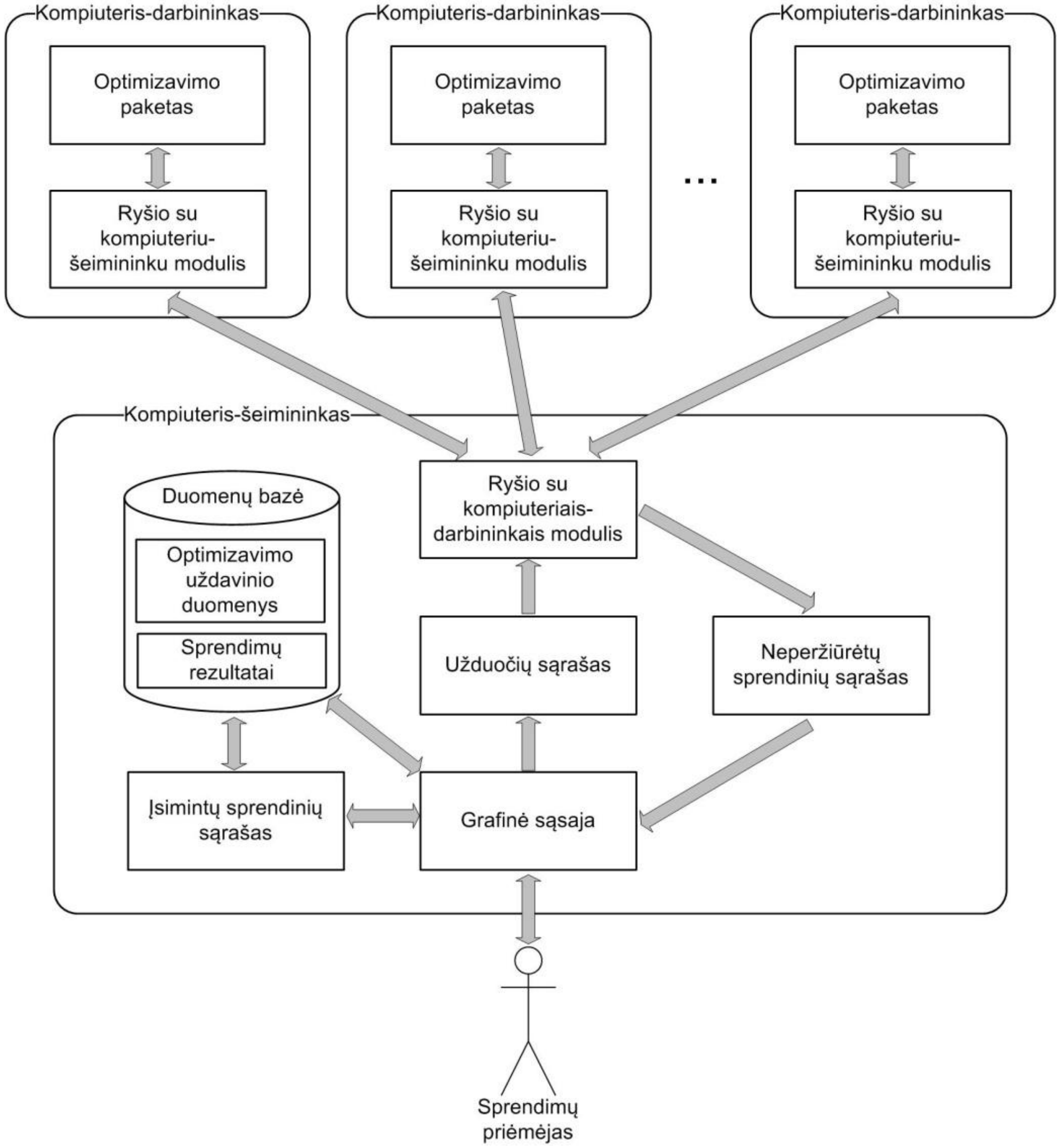
Modelio veikimo principas yra toks: vienas kompiuteris-šeimininkas kaupia užduotis ir jas dalija kompiuteriams-darbininkams, o kompiuteriai-darbininkai tas užduotis sprendžia. Šeimininko-darbininkų lygiagretinimo būdas yra „vienas su daugeliu" komunikacijos modelis, t. y. vienas kompiuteris, vadinamas *šeimininku,* „bendrauja" su daugeliu kitų kompiuterių, vadinamų *darbininkais*. Darbininkai tarpusavyje informacija nesikeičia. Kompiuteriuose su keliais procesoriais modelio esmė nesikeičia – šeimininku arba darbininkais tampa procesoriai.

3.3. Daugiakriterinio optimizavimo SPS modelis

Šiame poskyryje aprašomas interaktyvios sprendimų paramos sistemos modelis, skirtas daugiakriteriniams optimizavimo uždaviniams spręsti, kai uždavinys suvedamas į vienakriterinį skaliarizavimo pagalba. Modelis apjungia optimizavimą, sprendimo proceso vizualizavimą ir jo lygiagretinimą. Modelio schema pavaizduota 3.3 pav. 2.5.2 skyrelyje pateiktose daugiakriterinio optimizavimo SPS apjungiamas optimizavimas ir sprendimo proceso vizualizavimas, bet nenumatyta sprendimo proceso lygiagretinimo galimybė.

Grafinė sąsaja

Sprendimo priėmėjo dialogas su SPS vyksta per grafinę sąsają. Naudojant grafinės sąsajos įrankius, SP gali pasiruošti uždavinio sprendimui, autorizuotis arba registruotis sistemoje, pasirinkti reikiamą kompiuterių-darbininkų skaičių. SP uždavinio sprendimo eigoje gali pasirinkti optimizavimo metodą, formuoti vienakriterinius uždavinius, įvertinti gautus sprendinius, išsaugoti tinkamus ir pašalinti netinkamus. Grafinės sąsajos dėka SP gauna tokią informaciją: koks yra skaičius suformuotų, tačiau nepradėtų spręsti užduočių; kiek yra laisvų kompiuterių-darbininkų; koks yra gautų, bet dar neperžiūrėtų sprendinių skaičius. Žinodamas šią informaciją, SP gali planuoti savo darbą.



3.3 pav. *Sprendimų paramos sistemos modelis*

Užduočių sąrašas

Kai naudojant grafinę sąsają, SP suformuoja užduotį, ji įrašoma į užduočių sąrašą. Užduotys šiame sąraše saugomos, kol ryšio su kompiuteriais-darbininkais modulis nepersiunčia jas kompiuteriams-darbininkams.

Ryšio su kompiuteriais-darbininkais modulis

Sudarant SPS modelį buvo atsižvelgta ir į disertacijoje (Petkus 2001) pasiūlytą kompiuterio tinklo panaudojimo modelį interaktyviam daugiakriterinio optimizavimo uždavinio sprendimui. Ryšio su kompiuteriais-darbininkais modulis yra užduočių sąrašo ir kompiuterių-darbininkų tarpininkas, t. y. tam tikras užduočių vykdymo dispečeris, siunčiantis suformuotas užduotis laisviems kompiuteriams-darbininkams. Jei laisvų kompiuterių-darbininkų nėra, ryšio su kompiuteriais-darbininkais modulis lauks, kol kuris nors atsilaisvins ir išsiųs suformuotą užduotį išlaikant suformuotų užduočių eiliškumą – ankščiausiai suformuotos užduotys išsiunčiamos ankščiausiai. Ryšio su kompiuteriais-darbininkais modulis nuolat laukia sprendinių iš kompiuterių-darbininkų bei naujų sprendimo priėmėjo suformuotų užduočių.

Optimizavimo paketas

Kiekviename kompiuteryje-darbininke yra optimizavimo paketas, skirtas vienakriterinio uždavinio sprendimui. Šis paketas išsprendžia gautą iš ryšio su kompiuteriu-šeimininku modulio vienakriterinį uždavinį ir rezultatą perduoda ryšio su kompiuteriu-šeimininku moduliui.

Ryšio su kompiuteriu-šeimininku modulis

Ryšio su kompiuteriu-šeimininku modulis persiunčia išspręstą užduotį ryšio su kompiuteriais-darbininkais moduliui, kuris yra kompiuteryje-šeimininke. Kai tik kompiuteris-šeimininkas gauna sprendinį, jei tuo metu užduočių sąraše yra nors viena suformuota užduotis, ją ryšio su kompiuteriais-darbininkais modulis išsiunčia atsilaisvinusiam kompiuteriui-darbininkui. Užduotį priima ryšio su kompiuteriu-šeimininku modulis ir ją perduoda optimizavimo paketui.

Neperžiūrėtų sprendinių sąrašas

Ryšio su kompiuteriais-darbininkais modulis iš kompiuterių-darbininkų gautus užduočių sprendinius įrašo į neperžiūrėtų sprendinių sąrašą. Grafinėje sąsajoje atnaujinama informacija apie išspręstų, bet dar neperžiūrėtų sprendinių skaičių ir laisvų kompiuterių-darbininkų skaičių. Sprendimų priėmėjas bet kuriuo momentu gali peržiūrėti gautus sprendinius neperžiūrėtų sprendinių sąraše.

Įsimintų sprendinių sąrašas

SP, įvertinęs gautą sprendinį, gali jį išsaugoti į įsimintų sprendinių sąrašą. Vėliau įsimintus sprendinius SP gali peržiūrėti, pasirinktus ištrinti ar redaguoti. Pasibaigus sprendimo procesui, įsimintų sprendinių sąrašas išsaugomas duomenų bazėje.

Duomenų bazė

Duomenų bazėje saugomi uždavinio sprendimui reikalingi duomenys, informacija apie registruotus sistemoje sprendimų priėmėjus. Duomenų bazėje yra kaupiama ir saugoma informacija apie uždavinių sprendimo proceso eigą: kiekvieno SP visos suformuotos užduotys ir gauti sprendiniai. Taip pat duomenų bazėje saugomas kiekvieno SP įsimintų sprendinių sąrašas, kurį SP formuoja uždavinio sprendimo metu.

Pagal šį modelį suprojektuota ir sukurta SPS leidžia interaktyviai spręsti daugiakriterinius optimizavimo uždavinius pasitelkiant kompiuterių klasterį.

3.4. Programinės įrangos pasirinkimo pagrindimas

3.3 poskyryje aprašyto modelio įgyvendinimui pasirinkta kompiuterinė matematinė sistema Matlab (versija R2009a). Matlab – tai aukšto lygio programavimo kalba, skirta algoritmų įgyvendinimui, turinti interaktyvią aplinką, duomenų analizės ir vizualizavimo galimybes.

Matlab yra pritaikyta matriciniams skaičiavimams, todėl skaičiavimo veiksmai su masyvais (vektoriais ir matricomis) yra atliekami greitai. Kuriamoje SPS sprendžiami optimizavimo uždaviniai, didžioji dalis skaičiavimų ir yra būtent veiksmai su masyvais.

Į Matlab integruota daugybė specializuotų paketų (angl. *Toolboxes*), sudarytų iš specialių tarpusavyje susietų funkcijų, kurie praplečia Matlab sistemą ir įgalina spręsti atitinkamų klasių uždavinius. Tarp tokių paketų yra ir optimizavimo paketas (angl. *Optimization Toolbox*), kuriame yra įgyvendinti plačiai naudojami įvairių klasių klasikiniai bei šiuolaikiniai optimizavimo metodai. Tarp jų yra metodai, skirti tolydžių ir diskrečių uždavinių su apribojimais ir be jų sprendimui. Optimizavimo pakete yra funkcijos, skirtos tiesiniam programavimui, kvadratiniam programavimui, netiesiniam optimizavimui, daugiakriteriniam optimizavimui.

Taipogi Matlab sistemoje yra lygiagrečių ir paskirstytų skaičiavimų paketas (angl. *Parallel Computing Toolbox*). Jo pagalba galima spręsti didelių skaičiavimų reikalaujančius uždavinius, naudojant kelių procesorių kompiuterius, vaizdo plokštės procesorius ir kompiuterių klasterius. Lygiagrečią programos dalį galima sukurti naudojant MPI (angl. *Message Passing Interface*) standartą, kurio funkcijos yra integruotos į Matlab. Be to, Matlab yra specialios konstrukcijos, kurios įgalina lygiagretinti uždavinį be programavimo naudojant MPI, pvz., lygiagretus „*for*“ ciklas, specialūs paskirstytų masyvų tipai ir lygiagretūs skaitiniai algoritmai. Naudojant Matlab paskirstytų skaičiavimų serverį (angl. *Distributed Computing Server*), galima spręsti uždavinius kompiuterių klasteriuose, griduose arba naudojant debesų kompiuterijos technologijas (angl. *Cloud Computing*).

Matlab vaizdinio programavimo galimybės leidžia sukurti grafinę vartotojo sąsają (angl. *Graphical User Interface, GUI*), kas yra ypač aktualu kuriant interaktyvią SPS. Darbui su grafinę vartotojo sąsaja Matlab sistemoje yra speciali vaizdinė projektavimo terpė – GUIDE (angl. *GUI Development*

Environment). Jos galimybės nenusileidžia kitoms programavimo kalboms ir terpėms.

Matlab kompiliatoriaus (angl. *Compiler*) pagalba galima sukurti nepriklausomą programinę įrangą, kuri gali veikti kompiuteriuose be Matlab sistemos įdiegimo. Į programinę įrangą galima įtraukti uždavinio sprendimui reikalingus Matlab paketus.

3.5. Sukurtos SPS aprašymas

Šiame poskyryje aprašoma sprendimų paramos sistemą, sukurta pagal modelį, aprašytą 3.3 poskyryje. Sistemoje įgyvendintas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas, pasiūlytas 3.1 poskyryje. Sprendžiant daugiakriterinį optimizavimo uždavinį, SP vaidmuo yra labai svarbus. Jo patirtis ir turima formali bei neformali informacija apie sprendžiamą uždavinį įtakoja visą sprendimo procesą (Dzemyda and Petkus 2001). Todėl turi būti sukurta SPS, turinti patogią grafinę sąsają, kuri palengvintų uždavinio sprendimą, leistų vizualiai peržiūrėti ir įvertinti gautus rezultatus bei planuoti tolesnį sprendimo procesą. Sprendimų priėmime grafinis pateikimas yra labai svarbus, sprendžiant daugiakriterinius optimizavimo uždavinius (Vassilev *et al.* 2006; Blasco *et al.* 2008; Ginevičius and Podvezko 2008; Dowhań *et al.* 2009).

SPS, skirta daugiakriteriniams uždaviniams spręsti, sukurta naudojant matematinį paketą Matlab. Sukurtoje SPS yra apjungta:

- optimizavimo metodai;
- sprendimo proceso lygiagretinimas;
- grafinė vartotojo sąsaja.

Sistemoje, įgyvendinant pasiūlytą interaktyvų daugiakriterinių optimizavimo uždavinių sprendimo būdą, vienakriterinio uždavinio sprendimui buvo naudojama Matlab funkcija *fmincon*, skirta netiesinio optimizavimo uždaviniams su apribojimais spręsti. Detaliai ši funkcija buvo ištirta sprendžiant polių padėčių pamatuose optimizavimo uždavinį (A 3). Be to, ji buvo palyginta su T. Petkaus pasiūlytoje SPS naudojamu optimizavimo paketu (B 1). Šioje funkcijoje yra įgyvendinti keli optimizavimo algoritmai: *Trust-Region-Reflective Optimization* (Coleman and Li 1994; Coleman and Li 1996), *Active-Set Optimization* (Han 1977; Powell 1978; Gill *et al.* 1981), *Interior-Point Optimization* (Byrd *et al.* 1999; Waltz *et al.* 2006), *SQP Optimization* (Nocedal and Wright 2000). Funkcija *fmincon* pati parenka tinkamiausią algoritmą arba reikiamą algoritmų kombinaciją optimalaus sprendinio paieškai.

Sukurtoje SPS sprendimo proceso lygiagretinimui buvo panaudotas Matlab lygiagrečių ir paskirstytų skaičiavimų paketas ir paskirstytų skaičiavimų serveris. Uždavinio sprendimo procesas lygiagretinamas sukurtoje sistemoje, naudojant MPI standarto funkcijas bei specialias Matlab konstrukcijas – paskirstytus masyvus. Sprendžiant daugiakriterinį optimizavimo uždavinį sukurtos SPS pagalba, SP gali pasirinkti procesorių skaičių, atsižvelgiant į uždavinio sudėtingumą ir prieinamų kompiuterių skaičių.

Grafinė vartotojo sąsaja sukurta, naudojant vaizdinę projektavimo terpę Matlab GUIDE. Siekiant, kad sukurta SPS veiktų be Matlab įdiegimo į kompiuterius, buvo panaudotas Matlab kompiliatorius.

Sukurtoje SPS galima atlikti tokius veiksmus:

- pasirinkti vartotoją (sprendimų priėmėją), pridėti naują, arba pašalinti esamą;
- pasirinkti uždavinio sprendimui naudojamų kompiuterių (procesorių) skaičių;
- išsaugoti visų SP gautus rezultatus bei sprendimo proceso eigoje gautus tarpinius sprendinius;
- pašalinti netinkamus sprendinius;

- peržiūrėti gautus rezultatus, juos lyginti tarpusavyje;
- pasirinkti sprendinį redagavimui;
- patogiai ir greitai keisti svorinių koeficientų reikšmes;
- stebėti sprendimo proceso eigą;
- spręsti uždavinį pasiūlytu interaktyviu daugiakriterinių optimizavimo uždavinių sprendimo būdu.

3.6. Sprendimo strategijos

Šiame poskyryje aprašomos kelios sprendimo strategijos daugiakriteriniams optimizavimo uždaviniams spręsti, naudojant 3.1 poskyryje pasiūlytą interaktyvų sprendimo būdą.

Pasiūlytame interaktyviame sprendimo būde sprendimų priėmėjas parenka skirtingus svorinių koeficientų $w = (w_1, w_2, \ldots, w_m)$ reikšmių rinkinius, formuojami vienakriteriniai optimizavimo uždaviniai, jie yra išsprendžiami optimizavimo paketu ir SP, gavęs optimizavimo rezultatą (sprendinį), jį įvertina. Toliau darbe suformuotą vienakriterinį optimizavimo uždavinį, esant vienam svorinių koeficientų $w = (w_1, w_2, \ldots, w_m)$ reikšmių rinkiniui, vadinsime *užduotimi*.

Suformuotas daugiakriterinis optimizavimo uždavinys sprendžiamas naudojant kelis kompiuterius (procesorius), juos suskirsčius į „šeimininką“ ir „darbininkus“. Kompiuteris-šeimininkas kontroliuoja visų kompiuterių-darbininkų darbo eigą, siunčia kompiuteriams-darbininkams suformuotas užduotis, jas surenka iš kompiuterių-darbininkų, o kompiuteriai-darbininkai gauna ir vykdo kompiuterio-šeimininko paskirtas užduotis, jas išsprendžia ir siunčia gautą sprendinį kompiuteriui-šeimininkui. Taigi daugiakriterinis optimizavimo uždavinys sprendžiamas lygiagrečiai su skirtingomis svorinių koeficientų $w = (w_1, w_2, \ldots, w_m)$ reikšmėmis: skirtingi kompiuteriai-darbininkai sprendžia tą patį uždavinį tik su skirtingais svorinių koeficientų reikšmių rinkiniais, o SP įvertina gautus rezultatus.

Sprendimo procesui paspartinti galimos kelios daugiakriterinių optimizavimo uždavinių sprendimo strategijos.

3.6.1. Pirma sprendimo strategija

Šią sprendimo strategiją pasiūlė G. Dzemyda ir T. Petkus (Dzemyda and Petkus 2001). Joje SP formuoja visas užduotis ir įvertina gautus sprendinius.

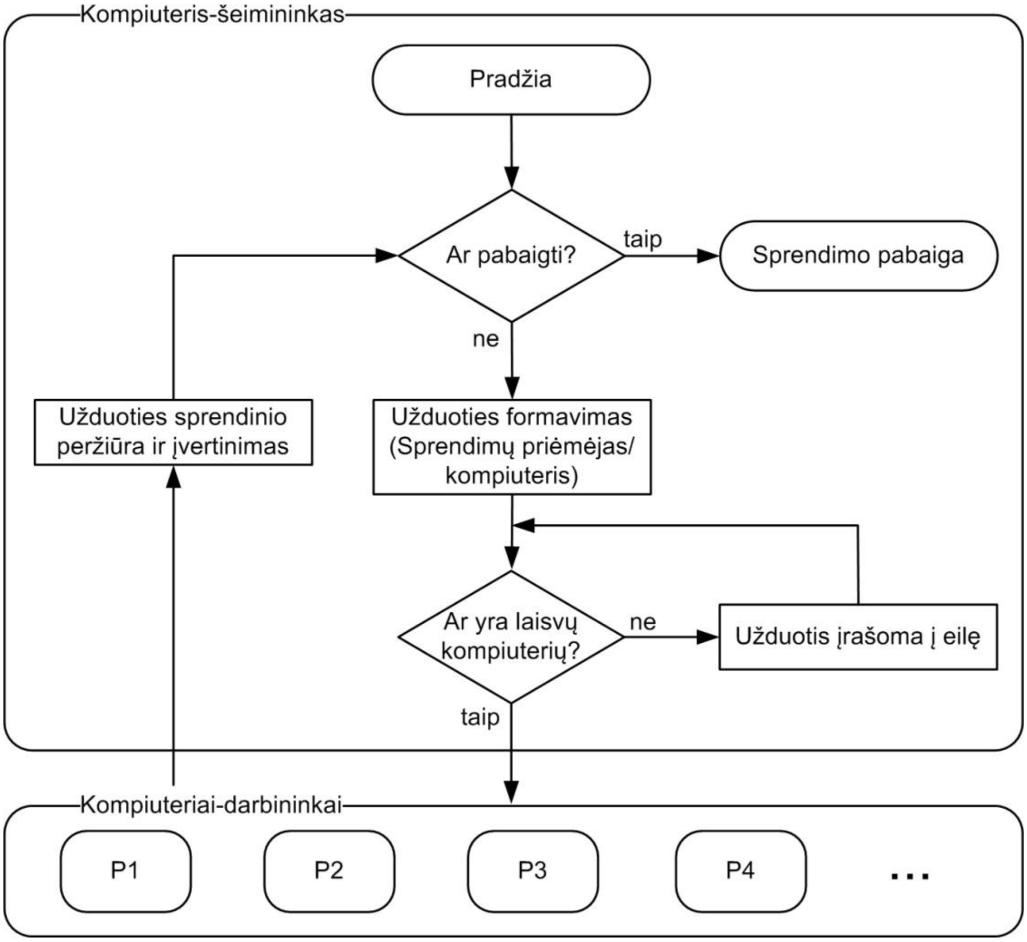
Sprendžiant daugiakriterinius optimizavimo uždavinius lygiagrečiai ir naudojant keletą kompiuterių-darbininkų, vienam SP pakankamai sunku suformuoti užduotis taip, kad visi kompiuteriai-darbininkai turėtų darbo, t. y. visą laiką būtų užimti. Svorinių koeficientų $w = (w_1, w_2, \dots, w_m)$ reikšmių parinkimas sprendimų priėmėjui užtrunka pakankamai ilgai. Be to, SP ypač daug laiko gaišta formuodamas užduotis uždavinio sprendimo pradžioje. Kuo daugiau yra naudojama kompiuterių-darbininkų, tuo kompiuterių užimtumas yra mažesnis. Todėl didelio kompiuterių-darbininkų skaičiaus naudojimas, sprendžiant uždavinį pirma strategija, nėra efektyvus.

3.6.2. Antra sprendimo strategija

Šiame darbe siūloma nauja daugiakriterinių optimizavimo uždavinių sprendimo strategija, kurios pagalba siekiama išvengti kompiuterių-darbininkų prastovų ir paspartinti uždavinio sprendimą (A 1). Siūloma svorinių koeficientų reikšmes generuoti atsitiktinai, bet atsižvelgiant į prieš tai gautą geriausią sprendinį. Šioje strategijoje užduotis formuoja SP, o kompiuteris-šeimininkas generuoja užduotis tuomet, kai SP nespėja, t. y. atsiranda laisvų kompiuterių-darbininkų. Kompiuteris, formuodamas užduotis, generuoja svorinių koeficientų reikšmes artimas svorinių koeficientų reikšmėms, esant kurioms, gautas SP nuomone geriausias sprendinys, t. y. svorinių koeficientų reikšmės generuojamos atsitiktinai, pridedant arba atimant atsitiktinį skaičių prie svorinių koeficientų reikšmių, su kuriomis gautas iki tol geriausias sprendinys. Tačiau atsitiktinė skaičiaus vertė parinkta taip, kad svorinių koeficientų reikšmės pasikeistų nekardinaliai, nes būtina atsižvelgti į SP nustatomus prioritetus. Todėl siūlomoje strategijoje atsitiktinis skaičius neviršija 10 % nuo svorinių koeficientų reikšmių, su kuriomis gautas iki tol geriausias sprendinys.

Naudojant antrą strategiją, yra išsprendžiama kompiuterių-darbininkų prastovų problema, kai kompiuterių-darbininkų yra pakankamai daug, ir SP tam tikru momentu nespėja formuoti naujų užduočių. Generuojant naujas užduotis šia strategija, kompiuteris tampa pagalbininku sprendimų priėmėjui.

Pirmos ir antros strategijų apibendrinta veikimo kompiuterių klasteryje schema pateikiama 3.4 pav.



3.4 pav. *Strategijų veikimo schema*

Palyginkime, kuo skiriasi uždavinio sprendimo strategija, kai užduotis formuoja tik SP (pirma strategija), nuo strategijos, kai užduotis padeda generuoti kompiuteris-šeimininkas (antra strategija).

Bendra vienakriterinio uždavinio sprendimo trukmė yra $t_{\mathrm{v}} = t_{\mathrm{f}} + t_{\mathrm{s}} + t_{\varepsilon}$, čia t_{f} – užduoties formavimo trukmė, t_{s} – užduoties sprendimo trukmė, t_{ε} – duomenų siuntimo ir kitos laiko sąnaudos.

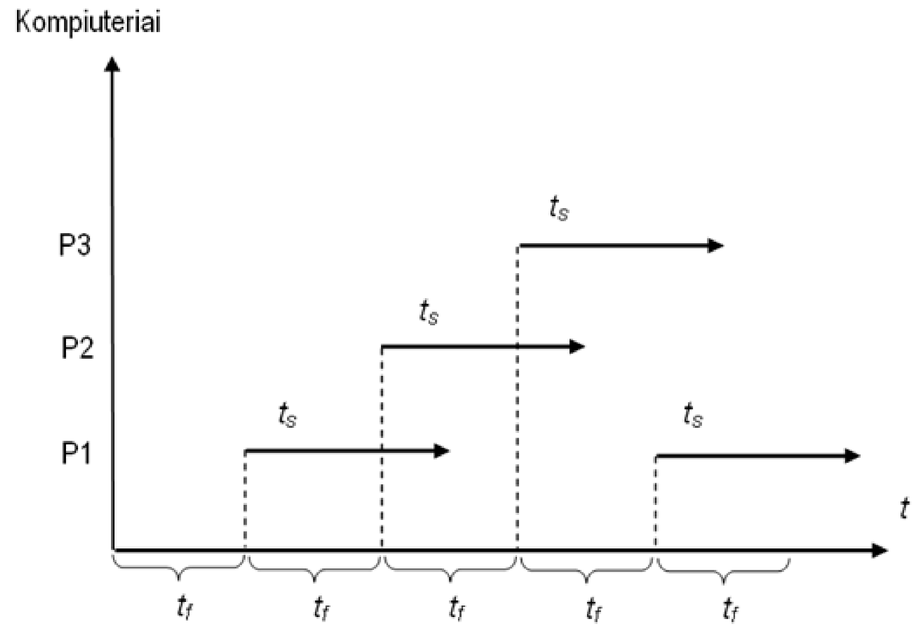
Pirmoje strategijoje užduoties formavimo trukmė t_{s} yra kintanti. Darbo pradžioje pirmųjų užduočių formavimas atima daugiau SP laiko. Tolimesnės užduotys yra formuojamos beveik pastoviu greičiu. Taigi nusistovėjusi užduoties formavimo trukmė t_{f} yra priklausoma nuo uždavinio sprendimui naudojamų kompiuterių skaičiaus. Šis dydis t_{f} taip pat priklauso nuo SP patirties bei užduoties sprendimo trukmės t_{s} (didelis iš tinklo „grįžusių“ rezultatų skaičius verčia SP skubėti).

Išanalizuokime nusistovėjusį darbo režimą su tam tikru pakankamu kompiuterių-darbininkų skaičiumi. Šiuo atveju darome prielaidą, kad t_{f} ir t_{s} yra pastovūs, o P yra pakankamas kompiuterių-darbininkų skaičius. Šis skaičius yra lygus:

$$P = \frac{t_{\mathrm{s}}}{t_{\mathrm{f}}}. \tag{3.1}$$

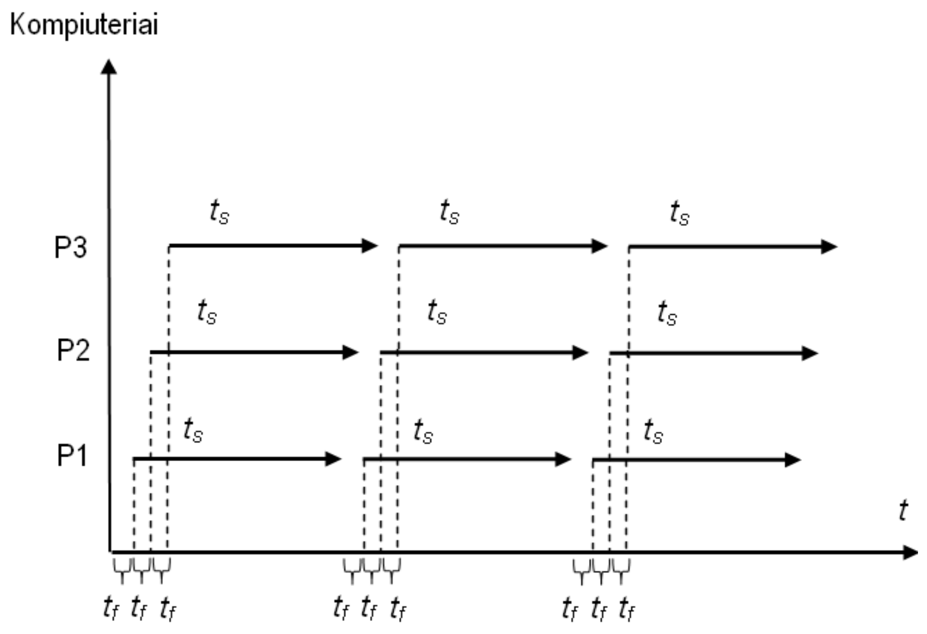
Aptarkime pavyzdį, pateiktą 3.5 pav. Nusistovėjusiu atveju pakankamas kompiuterių skaičius uždaviniui spręsti yra trys arba keturi kompiuteriai (šis skaičius priklauso nuo SP patirties): tik tuomet nei kompiuteriai, nei SP neturės reikšmingų prastovų. Prastovų trukmė yra lygi $Pt_{\mathrm{f}} - t_{\mathrm{s}}$:

- jei $Pt_{\mathrm{f}} - t_{\mathrm{s}} = 0$, tai prastovų nebus (tiek kompiuterių, tiek ir SP);
- jei $Pt_{\mathrm{f}} - t_{\mathrm{s}} > 0$, tai kompiuteris patirs prastovų (šis atvejis yra iliustruojamas 3.5 pav.);
- jei $Pt_{\mathrm{f}} - t_{\mathrm{s}} < 0$, tai prastovų patirs SP (P nebus pakankamas) (Petkus 2001).



3.5 pav. *Pirmos strategijos užduočių formavimo ir sprendimo laikas*

Jei užduotis yra formuojama ne SP, o kompiuterio-šeimininko (antra strategija), tai užduoties formavimo laikas t_f yra žymiai mažesnis, todėl sumažėja ir kompiuterių prastovų. Šis atvejis yra iliustruojamas 3.6 pav., kuriame matome, kad kompiuteris-šeimininkas gali spėti suformuoti užduočių dideliam kompiuterių-darbininkų skaičiui, išvengiant prastovų.



3.6 pav. *Antros strategijos užduočių formavimo ir sprendimo laikas*

3.7. Praktinio uždavinio formuluotė

Pasiūlyto daugiakriterinių optimizavimo uždavinių sprendimo būdo tyrimui ir sukurtos sprendimų paramos sistemos testavimui buvo pasirinktas kombinuotųjų pašarų sudarymo gyvulininkystėje bei paukštininkystėje uždavinys. Gyvulininkystėje ir paukštininkystėje geras ekonominis kiaušinių ir mėsos gavybos efektyvumas pasiekiamas tinkamiausiai išnaudojant genetines gyvulių bei paukščių galimybes bei suteikiant jiems visavertę mitybą. Žinoma, kad paukštininkystėje pašarams skiriama 65-70 % lėšų. Esant tokioms sąlygoms ir atsižvelgiant į pastaruoju metu augančias pašarų sudedamųjų dalių kainas, didžioji dalis paukštininkų priversti peržiūrėti paukščių maitinimo programas ir ieškoti būdų pašarų kainai sumažinti (Petkus 2001).

Pašarų sudėties optimizavimo uždaviniams spręsti praktikoje naudojama gana daug skirtingų ingredientų (žaliavų): kukurūzai, kviečiai, miežiai, avižos, rugiai, valgomoji druska, premiksas, fermentinis preparatas ir t. t. Kiekvienas ingredientas pasižymi jam būdingomis savybėmis, maistinėmis charakteristikomis: energija, proteinai, riebalai, ląsteliena, kalcis, fosforas, kalis, natris, chloras, valinas ir t. t. Šio uždavinio sprendimui yra panaudota T. Petkaus sudaryta 50 ingredientų su skirtingomis 14 maistinėmis savybėmis (charakteristikomis) duomenų bazė (Petkus 2001). Sprendžiant pašarų sudėties optimizavimo uždavinį praktikoje vertinamos minėtos pašarų maistinės charakteristikos.

Turint reikiamus duomenis apie galimus ingredientus ir jų maistines charakteristikas, norint spręsti pašarų sudėties optimizavimo uždavinį, reikia žinoti minimalių ir maksimalių normų lygmenis. Kiekvienai atskirai gyvulių ir paukščių rūšiai reikia skirtingų duomenų rinkinių minimalioms ir maksimalioms maistinių charakteristikų normoms, be to, gali skirtis tos pačios rūšies gyvuliams ar paukščiams kiekviename jų amžiaus tarpsnyje.

Nagrinėjamame uždavinyje minimalių ir maksimalių maistinių charakteristikų normų lygmenys buvo pasirinkti tam atvejui, kai kombinuotieji pašarai skiriami viščiukams broileriams (mėsos gamybai) nuo gimimo iki septynių dienų amžiaus.

Sprendžiant kombinuotųjų pašarų sudarymo uždavinį įprastai minimizuojama pašarų savikaina:

$$f(X) = \sum_{i=1}^{n} x_i \psi_i, \tag{3.2}$$

esant apribojimams

$$R_j^{\min} \le R_j \le R_j^{\max}, \tag{3.3}$$

$$\sum_{i=1}^{n} x_i = 1, \tag{3.4}$$

$$x_i^{\min} \le x_i \le x_i^{\max}, \tag{3.5}$$

čia $R_j = \sum_{i=1}^{n} x_i A_{ij}(X)$, $j = 1, \dots, k$; x_i – i-tojo ingrediento dalis pašare; n – ingredientų skaičius pašare; k – maistinių charakteristikų skaičius pašare; $R_j^{\min}$ – rekomenduojamas leistinas minimalus j-tosios maistinės charakteristikos kiekis pašare; $R_j^{\max}$ – rekomenduojamas leistinas maksimalus j-tosios maistinės charakteristikos kiekis pašare; $A_{ij}(X)$ – netiesinė funkcija, nusakanti i-tojo ingrediento j-tosios maistinės charakteristikos reikšmę; ψ_i – i-tojo ingrediento svorio vieneto kaina; $x_i^{\min}$ ir $x_i^{\max}$ yra minimali ir maksimali i-ojo ingrediento dalis pašare; ingredientai $X = (x_1, x_2, \dots, x_n)$ turi įtakos kitų ingredientų maistinėms charakteristikoms.

Taigi sprendžiamas vienakriterinis optimizavimo uždavinys (3.2) su apribojimais. Būtina atkreipti dėmesį, kad $R_j^{\min}$ ($R_j^{\max}$) yra tik rekomenduojami leistini minimalūs (maksimalūs) maistinių charakteristikų kiekiai pašare. Sprendžiant uždavinį su tokias apribojimais gaunama didelė pašarų savikaina. Paukštininkystės versle stengiamasi ją sumažinti. Iš praktikos yra žinoma, kad tam tikrų apribojimų pažeidimai neturės esminės įtakos gaunamai produkcijos kokybei. Paukštininkystės ekspertui yra sudėtinga formalizuoti galimus pažeidimus, todėl siūloma apribojimus (3.3) perkelti į kriterijus ir uždavinį spręsti kaip daugiakriterinį. Tuomet ekspertas dalyvaujant uždavinio sprendime, remiasi savo patirtimi, formalia ir neformalia informacija (žiniomis) apie sprendžiamą uždavinį, ir gavęs sprendinį su jį tenkinančia savikaina, gali įvertinti pažeidimų visumos riziką produkcijos kokybei.

Sprendžiant daugiakriterinį pašarų sudarymo uždavinį, išskiriamos dvi prieštaraujančių kriterijų grupės:

1. Pašarų savikaina

$$f_1(X) = \sum_{i=1}^{n} x_i \psi_i. \tag{3.6}$$

2. Maistinių charakteristikų rekomenduojamų leistinų minimalių ir maksimalių normų pažeidimai

$$f_j(X) = \begin{cases} 0, & \text{jei } \ R_j^{\min} \le R_j \le R_j^{\max} \\ R_j - R_j^{\max}, & \text{jei } \ R_j - R_j^{\max} > 0 \\ R_j^{\min} - R_j, & \text{jei } \ R_j^{\min} - R_j > 0 \end{cases} \tag{3.7}$$

čia $R_j = \sum_{i=1}^{n} x_i A_{ij}(X)$, $j = 2, \dots, m$; $(m-1)$ – maistinių charakteristikų skaičius pašare.

Pirmos ir antros grupių kriterijai yra prieštaringi – daugiau pažeidus leistinas maistinių charakteristikų normas, galima gauti mažesnę savikainą.

Atsižvelgiant į tai, kad ingredientų dalių x_i suma pašare turi būti lygi 1, pasiūlyta tokia optimalios pašarų sudėties sudarymo uždavinio formuluotė:

$$\min_{X \in \mathrm{D}} \left(w_1 f_1(X) + \sum_{j=2}^{m} w_j f_j^2(X) \right), \tag{3.8}$$

esant apribojimams:

$$\sum_{i=1}^{n} x_i = 1, \tag{3.9}$$

$$x_i^{\min} \le x_i \le x_i^{\max}, \tag{3.10}$$

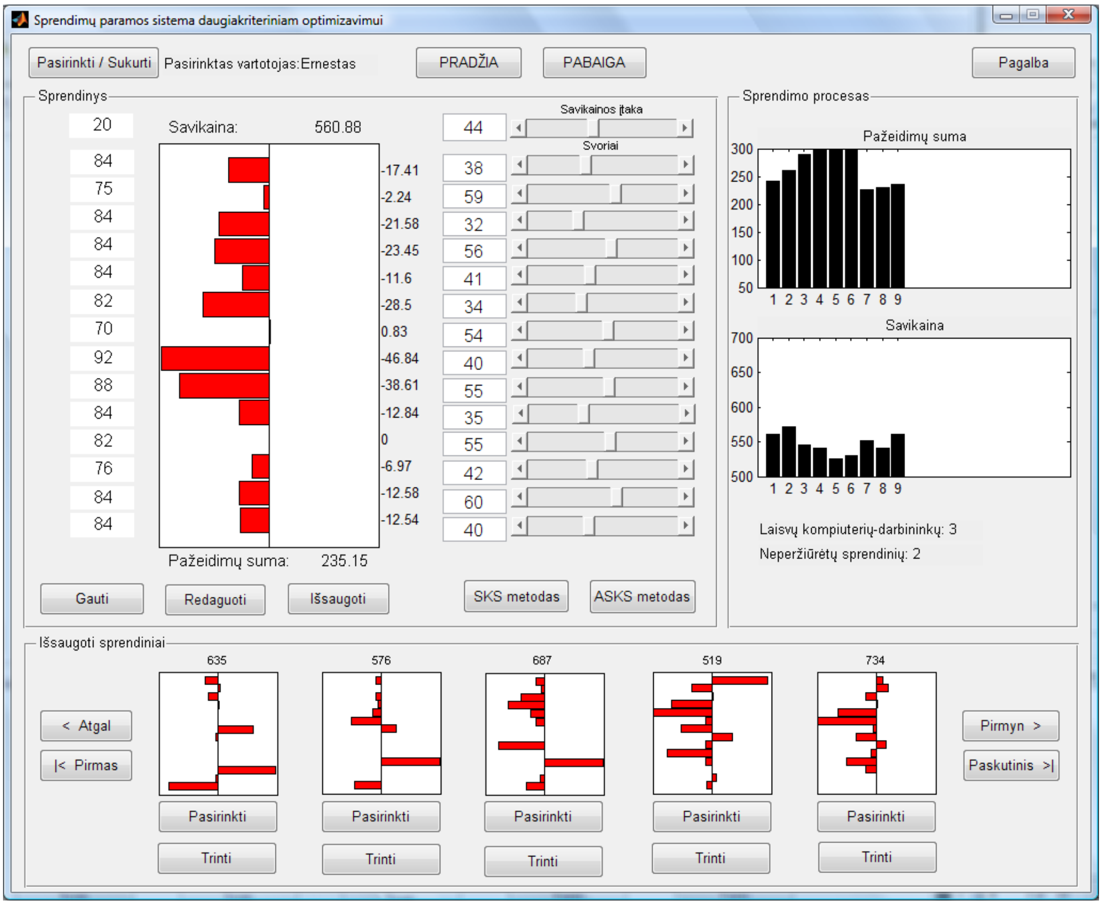
čia $\sum_{j=1}^{m} w_j = 1$, n – kintamųjų skaičius, m – kriterijų skaičius. Analizuojamu atveju $n = 50, m = 15$.

Uždavinio specifika yra ir ta, kad tikslo funkcijoje naudojami kriterijų $f_j(X),\ j = 2, \dots, m$ kvadratai. Tai, kaip ir apibrėžimo srities vertimas stačiakampe, išplečia potencialių lokalaus optimizavimo algoritmų spektrą šiam uždaviniui spręsti. Be to, tikslo funkcija (3.8) yra iškila.

3.8. Sukurtos SPS pritaikymas praktiniam uždaviniui

Dalis sukurtos SPS grafinės sąsajos elementų buvo specialiai pritaikyta pasirinktam uždaviniui siekiant maksimaliai užtikrinti sprendimų priėmėjo ir SPS sąveiką. Optimizavimo uždavinio tikslas – surasti sprendinį su mažiausiais rekomenduojamų leistinų maistinių charakteristikų pašare pažeidimais, tenkinančiais sprendimo priėmėjo reikalavimus, ir su kiek galima mažiausia savikaina. Pagrindinis SPS langas yra pateiktas 3.7 pav.

Prieš sprendžiant šį daugiakriterinį optimizavimo uždavinį, reikia pasirinkti duomenų bazėje registruotą sprendimų priėmėją arba registruoti naują SP. Tam veiksmui atlikti lango viršuje, kairiame kampe yra mygtukas *Pasirinkti / Sukurti*. Jį paspaudus, atsiranda papildomas langas, kuriame galima pasirinkti vieną iš registruotų sprendimų priėmėjų. Jeigu sprendimų priėmėjas su SPS dirba pirmą kartą, įvedus sprendimo priėmėjo vardą, yra sukuriamas naujas sprendimų paramos sistemos vartotojas ir jis registruojamas duomenų bazėje. Tokiu būdu suteikiama galimybė kaupti reikiamą informaciją apie kiekvieno SP sprendimo proceso eigą ir pasiektus rezultatus, kuriuos vėliau galima bus analizuoti. Taip pat bus galima palyginti skirtingų sprendimų priėmėjų gautus rezultatus tarpusavyje.



3.7 pav. *Sprendimų paramos sistemos grafinė sąsaja*

Norint pradėti spręsti daugiakriterinį optimizavimo uždavinį, reikia paspausti mygtuką *PRADŽIA*. Atlikus šį veiksmą, iš duomenų bazės yra paimami optimizavimo uždavinio sprendimui reikalingi duomenys: rekomenduojamas leistinas minimalus ir maksimalus maistinių charakteristikų kiekis pašare, ingredientų svorių vienetų kaina, minimalios ir maksimalios ingredientų dalys pašare, pradinės kintamųjų reikšmės. Sprendimų priėmėjui pateikiamas duomenų bazėje išsaugotų sprendinių sąrašas, kuriame sukaupti sprendimo priėmėjo išsaugoti sprendiniai, gauti ankstesnių eksperimentų eigoje.

Viršutiniame kairiame lango kampe, stačiakampyje, vaizduojamas paskutinis gautas sprendinys (toliau *tarpinis sprendinys*) arba sprendinys, pasirinktas redagavimui. Diagrama, sudaryta iš keturiolikos horizontalių juostelių, parodo pažeidimus (nukrypimus nuo maistinių charakteristikų rekomenduojamų leistinų minimalių ir maksimalių normų), kurių reikšmės gali turėti įtakos sprendimų priėmėjo tolesniam sprendimo procesui. Dešinėje diagramos pusėje, šalia kiekvienos horizontalios juostelės rodomos pažeidimų reikšmės, o kairėje pusėje keturiolikos kriterijų svorinių koeficientų $w = (w_2, w_3, \dots, w_m)$, kai $m = 15$, reikšmės, su kuriomis buvo gautas šis sprendinys. Virš diagramos yra pateikiamas apskaičiuotas vienas iš uždavinio kriterijų $-$ savikaina, kurios svorinio koeficiento reikšmę (pirmas svorinis koeficientas w_1) sprendimų priėmėjas irgi gali keisti. Savikaina kartu su diagrama, vaizduojančia rezultatus, sprendimų priėmėjui leidžia spręsti apie gauto sprendinio tinkamumą. Po diagrama rodoma apskaičiuota pažeidimų suma. Kai tik SP gauna visą vizualią informaciją, jis intuityviai susieja pasirinktas svorinių koeficientų reikšmes su gautomis pažeidimų ir savikainos reikšmėmis.

SP, paspaudus mygtuką *Gauti*, gali peržiūrėti naujai gautą tarpinį sprendinį. Tada diagramoje bus atvaizduotas šis sprendinys. Paspaudžiant mygtuką *Redaguoti*, galima atvaizduotą tarpinį sprendinį parinkti redagavimui. Jei naujai gautas sprendinys, SP nuomone, yra pakankamai geras, jį galima įsiminti ir vėliau prireikus palyginti su kitais, ar toliau jį redaguoti, t.y. keisti svorinius koeficientų $w = (w_1, w_2, \dots, w_m)$, kai $m = 15$, reikšmes. Paspaudus mygtuką *Išsaugoti*, atvaizduotas tarpinis sprendinys bus išsaugotas į SP sprendinių sąrašą ir patalpintas į šio sprendimų priėmėjo išsaugotų sprendinių duomenų bazę.

Lango viduryje yra blokas, skirtas svorinių koeficientų $w = (w_1, w_2, \dots, w_m)$, kai $m = 15$, reikšmių pakeitimui. Formuojant naujas užduotis, SP gali parinkti naujas arba keisti pasirinkto tarpinio sprendinio svorinių koeficientų reikšmes, jas įvedant į teksto laukelius arba naudojant slinkties juostas. Paspaudus mygtuką *SKS metodas*, optimizuojamas vienakriterinis uždavinys, esant pasirinktoms svorinių koeficientų reikšmėms.

Lango dešinėje pusėje vaizduojamos dvi stulpelinės diagramos. Vienoje jų atvaizduojamos gautų sprendinių savikainų reikšmės, o kitoje pažeidimų sumos. Sprendimo eigoje duomenys diagramose yra nuolat papildomi, tokiu būdu SP gali sekti sprendimo proceso eigą ir stebėti, kaip jam sekasi uždavinio sprendimas.

Po šiomis diagramomis pateikiamas ir nuolat atnaujinamas tuo laiko momentu esančių laisvų kompiuterių skaičius arba suformuotų, bet dėl kompiuterių-darbininkų užimtumo nesprendžiamų užduočių skaičius bei gautų, bet dar neperžiūrėtų sprendinių skaičius.

Lango apačioje stačiakampiuose matome uždavinio sprendimo eigoje išsaugotus tarpinius sprendinius, pateiktus diagramose. Virš kiekvienos diagramos yra pateikiama savikaina. Lange vaizduojami tik penkių gautų ir išsaugotų sprendinių diagramos, tačiau, pasinaudojus mygtukais *Pirmyn, Paskutinis, Atgal* ir *Pirmas* galima peržiūrėti kitus išsaugotus sprendinius. Paspaudus mygtuką *Pasirinkti*, galima pasirinkti vieną iš išsaugotų sprendinių tolimesniam redagavimui, t.y. svorinių koeficientų $w = (w_1, w_2, \dots, w_m)$, kai $m = 15$, reikšmių pakeitimui. Taip pat numatyta galimybė iš išsaugotų sprendinių grupės panaikinti sprendimų priėmėjo manymu prastus ir tolesnei analizei bei sprendimų eigai nebetinkamus sprendinius. Tai atliekama mygtuko *Trinti* pagalba.

Paspaudus mygtuką *ASKS metodas*, uždavinys bus sprendžiamas *ASKS* metodu, po to SP galės peržiūrėti ir įvertinti gautus rezultatus, juos išsaugoti ir pasirinkti juos tolimesniam redagavimui. Paspaudus mygtuką *SKS metodas*, uždavinys vėl bus sprendžiamas SKS metodu.

Paspaudus mygtuką *PABAIGA* uždavinio sprendimo procesas sustabdomas. Norint susipažinti su SPS naudojimo instrukcija ir ypatumais reikia paspausti mygtuką *Pagalba*.

3.9. Siūlomo interaktyvaus sprendimo būdo pritaikymas praktiniam uždaviniui

Sukurtos SPS pagalba yra minimizuojamas penkiolikos kriterijų pašarų sudėties sudarymo uždavinys (savikaina ir maistinių charakteristikų leistinų normų pažeidimai). ASKS metodo įgyvendinimas, kai kriterijų daugiau nei trys, sudėtingas ir reikalauja didelių skaičiavimų. Be to, atveju, kai kriterijų skaičius yra trys ir daugiau, ne visi ASKS metodu gauti sprendiniai yra Pareto optimalūs, todėl juos reikia pašalinti, naudojant Pareto filtrą. Dėl šių priežasčių siūloma daugiakriterinio optimizavimo uždavinio kriterijus suskirstyti į tarpusavyje prieštaraujančių kriterijų grupes. Tos pačios grupės kriterijai suvedami į vieną kriterijų juos susumuojant, o kriterijų grupei suteikiant vieną svorinį koeficientą.

Pašarų sudėties optimizavimo uždavinyje (3.8) yra dvi tarpusavyje prieštaraujančios kriterijų grupės: savikaina ir maistinių charakteristikų leistinų normų pažeidimai. Todėl ASKS metodu sprendžiamas dviejų kriterijų uždavinys:

$$\min_{X \in \mathrm{D}} \left(w_1' f_1(X) + w_1' f_2'(X) \right), \tag{3.11}$$

čia $f_1(X)$ – savikaina, $f_2'(X) = \sum_{j=2}^{15} f_j(X)$ – maistinių charakteristikų leistinų normų pažeidimų suma, $w_1' + w_2' = 1$.

Optimizavimo uždavinys (3.8) sprendžiamas 3.1 poskyryje pasiūlytu interaktyviu sprendimo būdu tokia veiksmų seka:

1. Penkiolikos kriterijų optimizavimo uždavinys (3.8) sprendžiamas SKS metodu. Sprendimų priėmėjas interaktyviai parenka svorinių koeficientų rinkinių $w = (w_1, w_2, \dots, w_m)$, kai $m = 15$, reikšmes. Su kiekvienu rinkiniu sprendžiamas vienakriterinis optimizavimo uždavinys. SP įvertina gautus sprendinius ir išsaugo tinkamiausius.
2. Jei SP negauna jį tenkinantį sprendinį, dviejų kriterijų uždavinys (3.11)

išsprendžiamas ASKS metodu. Apskaičiuojama savikaina ir visų keturiolikos reikalavimų pažeidimai, kurie atvaizduojami grafiškai.

3. SP įvertina ASKS metodu gautus rezultatus, iš jų pasirenka tinkamiausius. Negavęs tenkinančių sprendinių, SP gali bandyti pagerinti gautus sprendinius keičiant svorinių koeficientų $w = (w_1, w_2, \dots, w_m)$, kai $m = 15$, reikšmes ir sprendžiant uždavinį (3.8) SKS metodu.
4. Kai SP gauna jį tenkinantį sprendinį, sprendimo procesas sustabdomas.

3.10. Trečiojo skyriaus apibendrinimas ir išvados

Šiame skyriuje pasiūlytas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas, apjungiantis svertinės kriterijų sumos metodą su adaptyviu svertinės kriterijų sumos metodu. Jis užtikrina gaunamų sprendinių tolygų Pareto aibės padengimą bei sprendimo proceso interaktyvumą, kadangi sprendimų priėmėjui suteikiama galimybė dalyvauti uždavinio sprendimo eigoje nurodant norimas svorių koeficientų reikšmes.

Taip pat aprašytas sprendimų paramos sistemos modelis, apjungiantis optimizavimo metodus, sprendimo proceso vizualizavimą bei jo lygiagretinimą. Pasiūlytas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas buvo įgyvendintas pagal šį modelį sukurtoje sprendimų priėmimo sistemoje. Sistemos privalumas yra tas, kad sprendimų priėmėjas gali stebėti sprendimų proceso eigą, matyti tarpinius rezultatus, kas palengvina sprendimų priėmimą. Be to, naudojant kompiuterių klasterį, galima spręsti daug skaičiavimo resursų reikalaujančius optimizavimo uždavinius, kompiuterius-darbininkus naudojant atskiriems vienakriteriniams optimizavimo uždaviniams spręsti nepriklausomai vienas nuo kito.

Pasiūlyta interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo strategija, kurioje svorinių koeficientų reikšmes parenka net tik sprendimų priėmėjas, bet kai jis nespėja, svorinių koeficientų reikšmes generuoja kompiuteris atsižvelgiant į prieš tai gautą geriausią sprendinį. Strategijos pagalba išvengiama kompiuterių-darbininkų prastovų ir paspartinamas uždavinio sprendimo procesas.

Sukurta sistema ir pasiūlytas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas yra pritaikyti praktiniam pašarų sudėties sudarymo uždaviniui spręsti, bet jie gali būti lengvai pritaikomi ir kitiems panašaus pobūdžio uždaviniams. Be to, sukurta sistema gali būti naudojama tolesnei lyginamajai analizei, kadangi joje saugomi visų sprendimų priėmėjų tam tikru laiku gauti tarpiniai ir galutiniai sprendiniai, kurių pagrindu galima analizuoti, kuriam sprendimų priėmėjui greičiau pavyko įsigilinti į sprendžiamą uždavinį ir perprasti jo specifiką. Sprendimų priėmėjų gauti sprendiniai bus naudingi kitiems sprendimų priėmėjams pradėjus spręsti tą patį uždavinį.

4

Eksperimentiniai tyrimai

Šiame skyriuje pateikiami gauti eksperimentinių tyrimų rezultatai, publikuoti autoriaus darbuose (A 1, A 2, A 4 ir B 1). Siekiant ištirti sukurtą interaktyvų daugiakriterinių optimizavimo uždavinių sprendimo būdą bei sprendimų paramos sistemą, kurioje įgyvendintas šis būdas, buvo atlikti eksperimentiniai tyrimai naudojant vieną kompiuterį bei kompiuterių klasterį.

4.1. Praktinio uždavinio formuluotės pasirinkimas

Sprendžiant pašarų sudėties vienakriterinį optimizavimo uždavinį (3.2) su apribojimais (3.3)–(3.5), gaunama pašarų savikaina lygi 652. Kaip minėta 3.7 poskyryje leistini minimalūs ir maksimalūs maistinių charakteristikų kiekiai pašare yra tik rekomenduojami. Jei apribojimai (3.3) perkeliami į kriterijus, ir sprendžiamas daugiakriterinis uždavinys (3.8), ekspertui suteikiama galimybė savo nuožiūra pažeisti apribojimus, siekiant sumažinti pašarų savikainą. Iš praktikos yra žinoma, kad tam tikrų apribojimų pažeidimai neturės esminės įtakos gaunamai produkcijos kokybei.

Daugiakriterinis pašarų sudėties optimizavimo uždavinys buvo spręstas darbe (Petkus 2001). Jame tikslo funkcija skiriasi nuo pateiktos formulės (3.8) tuo, kad apribojimas (3.9) įtrauktas į tikslo funkciją su baudos koeficientu b_{k}. Spręsto uždavinio formuluotė buvo tokia:

$$\min_{X\in \mathrm{D}}\left(w_1 f_1(X)+\sum_{j=2}^{m} w_j f_j^2(X)+b_{\mathrm{k}}\left(\sum_{i=1}^{n} x_i-1\right)^2\right), \tag{4.1}$$

esant apribojimams:

$$x_i^{\min}\le x_i\le x_i^{\max}. \tag{4.2}$$

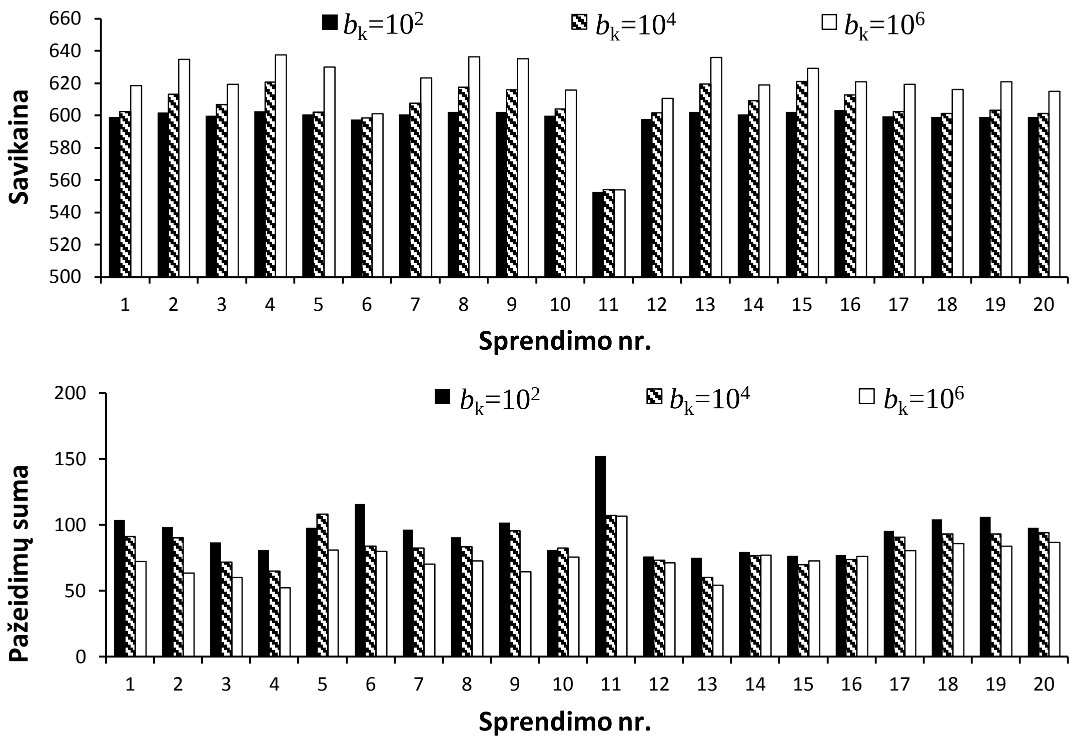
Šiame poskyryje pateikiami rezultatai, kai buvo spręstas minimizavimo uždavinys esant tikslo funkcijai (4.1). Buvo atlikta eilė eksperimentų su skirtingais svorinių koeficientų $w=(w_1,w_2,\dots,w_m)$, kai $m=15$, reikšmių rinkiniais ir skirtingomis baudos koeficiento b_{k} reikšmėmis. Gauti rezultatai palyginti tarpusavyje.

Kiekvienam svorinių koeficientų reikšmių rinkiniui buvo gautas sprendinys su skirtingomis koeficiento b_{k} reikšmėmis. Iš gautų 20 sprendinių rezultatų buvo apskaičiuoti vidurkiai, kurie pateikti 4.1 lentelėje.

4.1 lentelė. *Savikainos ir pažeidimų sumos vidurkiai esant skirtingoms* b_{k} *reikšmėms*

b_{k} reikšmė	Savikainos vidurkis	Pažeidimų sumos vidurkis
10^2	598,15	94,54
10^4	605,58	84,03
10^6	619,60	74,17

4.1 pav. pateikiami dvidešimties sprendinių rezultatai, kai buvo parinktos baudos koeficiento reikšmės $b_{\mathrm{k}}=10^2,10^4,10^6$. Gautiems sprendiniams apskaičiuota savikaina ir pažeidimų suma. Matome, kad kuo mažesnė koeficiento b_{k} reikšmė, tuo gautų sprendinių savikaina mažesnė, o pažeidimų suma didesnė.



4.1 pav. *Savikaina ir pažeidimų suma esant skirtingoms b_k reikšmėms*

Taip pat eksperimentiškai nustatyta, kad kai baudos koeficiento b_k reikšmė yra didelė ($b_k \geq 10^6$), gauti rezultatai beveik nesiskiria lyginat su rezultatais, gautais sprendžiant uždavinį, kai tikslo funkcija (3.8) (B 1). Tačiau esant mažesnei b_k reikšmei ji įtakoja gaunamus rezultatus, todėl tolimesniuose tyrimuose bus sprendžiamas uždavinys (3.8) su papildomu apribojimu (3.9).

4.2. Sprendimo strategijų lyginamoji analizė

Šiame poskyryje pateikiami tyrimų rezultatai, gauti sprendžiant pašarų sudėties sudarymo uždavinį (3.8), naudojant skirtingas sprendimo strategijas. Šio tyrimo rezultatai paskelbti darbe (A 1). Tyrime dalyvavo vienas SP, kuris jau turėjo įgūdžių spręsti analizuojamą uždavinį naudojant sukurtą SPS.

Eksperimentai buvo atliekami dvejomis strategijomis, pateiktomis 3.6 poskyryje, sprendžiant daugiakriterinį optimizavimo uždavinį (3.8) keliais kompiuteriais, kai vienakriterinio uždavinio sprendimo trukmė yra lygi vienai minutei. Vienakriterinio uždavinio sprendimo laikas apribotas, kad jis nedarytų įtakos strategijų tyrimui. Fiksuojamas sprendinys, kurį per vieną minutę pavyko gauti naudojamu optimizavimo metodu.

Buvo naudojamas 25 kompiuterių klasteris, sudarytas iš kompiuterių (Pentium 4, 3.2 GHz), sujungtų į vietinį tinklą (1 Gbps). Kompiuteriuose įdiegta operacinė sistema Windows XP.

Tiriant strategijas, daugiakriterinis uždavinys sprendžiamas svertinės kriterijų sumos metodu. Uždaviniams spręsti pirma strategija naudojami vienas ir šeši kompiuteriai-darbininkai, o sprendžiant optimizavimo uždavinius antra strategija – vienas, šeši, 12 ir 24 kompiuteriai-darbininkai.

Eksperimentais siekiama nustatyti:

- sukurtų strategijų efektyvumą;
- naudojamų kompiuterių-darbininkų skaičiaus įtaką rezultatams.

Siekiant palyginti sprendimo strategijas, būtina įvesti uždavinio sprendinio „gerumo“ matą, o tai daugiakriterinio uždavinio atveju yra gana problematiška, kadangi SP turi įvertinti visų kriterijų reikšmių tinkamumą. Atsižvelgiant į tai, kad savikaina yra iš vienos kriterijų grupės, o maistinių charakteristikų rekomenduojamų leistinų minimalių ir maksimalių normų pažeidimai iš kitos ir šių grupių kriterijai yra prieštaringi, tai siūloma naudoti jungtinį kriterijų (4.3), pasiūlytą darbe (Petkus 2001). Sprendžiant daugiakriterinį optimizavimo uždavinį mažiausias jungtinis kriterijus nebūtinai atitinka geriausią pasiektą sprendinį, todėl jis naudojamas tik strategijų, o ne sprendinių palyginimui. Jungtinio kriterijaus reikšmės laiko momentu t skaičiuojamos pagal formulę:

$$K_t = \sqrt{(\hat{f}_{1t})^2 + (\hat{f}'_{2t})^2}, \tag{4.3}$$

čia t – laiko momentas; $\hat{f}_{1t}$ – normuota savikaina, gauta laiko momentu t; $\hat{f}'_{2t}$ – normuota reikalavimų pažeidimų suma, gauta laiko momentu t; $f'_{2t} = \sum_{j=2}^{m} f_{jt}(x_1, \ldots, x_n)$. Reikšmės $\hat{f}_{1t}$ suvedamos į intervalą $[0; 1]$

naudojantis formule: $\hat{f}_{1t} = (f_{1t} - a)/(b - a)$, čia a – mažiausia visų eksperimentų metu gauta savikaina, b – didžiausia visų eksperimentų metu gauta savikaina, f_{1t} – savikaina, gauta laiko momentu t. $\hat{f}'_{2t}$ reikšmės gaunamos analogiškai kaip ir $\hat{f}_{1t}$. Grafiškai pateikiant jungtinio kriterijaus K_t reikšmes, vaizduojamos mažiausios iki to momento gautos reikšmės.

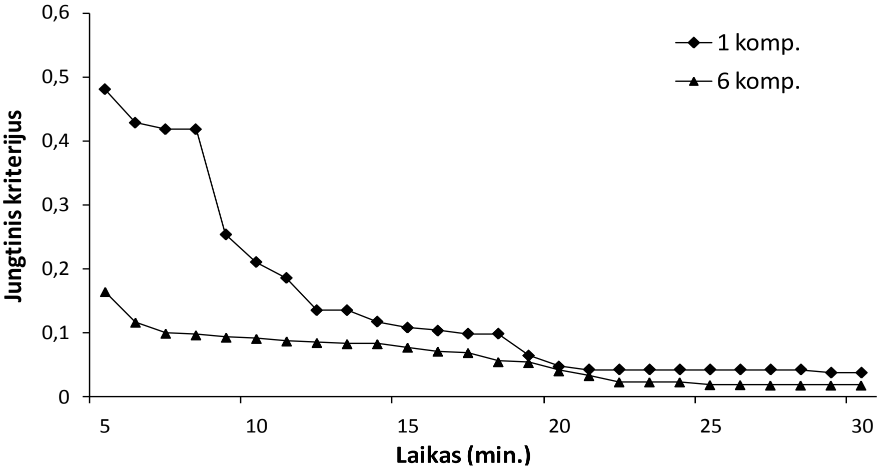
Toliau eksperimentu vadinsime sprendimo proceso dalį, kai daugiakriterinis uždavinys (3.8) spręstas ne mažiau kaip 30 minučių naudojant kompiuterių tinklą. Tyrimo metu kiekvienas eksperimentas kartojamas po dešimt kartų. Kiekvieno eksperimento metu sudarytas uždavinio sprendimo protokolas: pradedant nuliniu laiko momentu, gavus sprendinį, fiksuojami reikalavimų pažeidimai, savikaina ir jungtinis kriterijus, apimantis reikalavimų pažeidimus ir savikainą. Kiekvienai uždavinio sprendimo minutei skaičiuojama iki to laiko momento pasiekta jungtinio kriterijaus minimali reikšmė. Po to apskaičiuojamas šio kriterijaus reikšmių dešimties eksperimentų vidurkis.

Tegu pirmuoju atveju gauto jungtinio kriterijaus reikšmė tam tikru laiko momentu sudaro 100 %. Apskaičiuojama, kiek procentų nuo pirmojo atvejo sudaro antruoju atveju gauta jungtinio kriterijaus reikšmė. Tai apskaičiuojama kiekvienai lyginamų atvejų sprendimo minutei. Po to apskaičiuojamas gautų rezultatų vidurkis ν pagal formulę:

$$\nu = \sum_{t=1}^{T} \frac{K_{t2}}{K_{t1}} \times 100\ \%, \qquad (4.4)$$

čia t – laiko momentas ($t = 1, \dots, T$), T – uždavinio sprendimo laikas, K_{t1} ir K_{t2} – pirmojo ir antrojo lyginamų atvejų jungtinio kriterijaus reikšmės laiko momentu t. Šis vidurkis bus naudojamas norint skaitiškai įvertinti analizuojamus atvejus. Reiktų pažymėti, kad kuo šio vidurkio reikšmė mažesnė, tuo skirtumas tarp lyginamų atvejų yra didesnis.

4.2 pav. pavaizduoti rezultatai, gauti sprendžiant daugiakriterinį optimizavimo uždavinį pirma strategija naudojant vieną ir šešis kompiuterius-darbininkus. Kaip reikėjo tikėtis, sprendžiant uždavinį šešiais kompiuteriais-darbininkais gaunami geresni rezultatai (mažesnė jungtinio kriterijaus reikšmė) ir jie pasiekiami per trumpesnį laiką. Lyginant du atvejus, kai uždavinys sprendžiamas vienu ir šešiais kompiuteriais-darbininkais, jungtinio kriterijaus reikšmių procentų vidurkis $\nu = 51\ \%$. Tas reiškia, kad šešių kompiuterių naudojimas leidžia pasiekti beveik du kartus geresnius rezultatus jungtinio kriterijaus prasme.

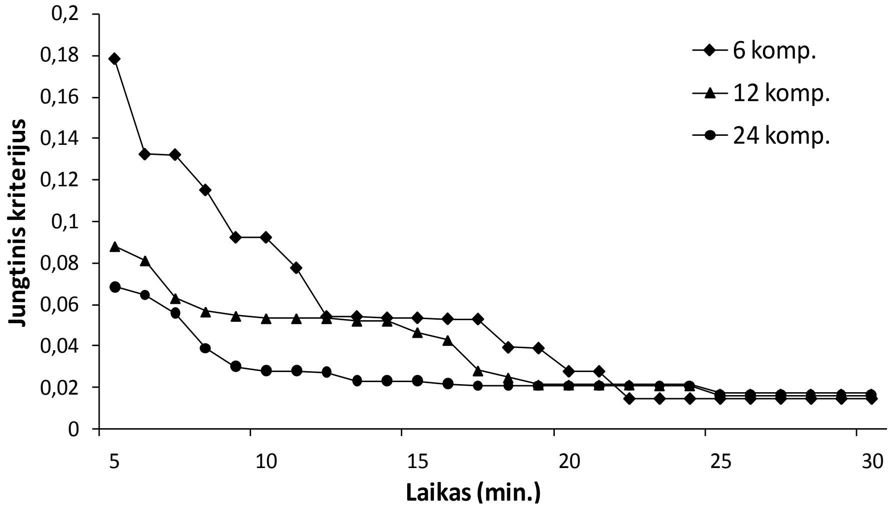


4.2 pav. *Pirmos strategijos efektyvumo priklausomybė nuo kompiuterių-darbininkų skaičiaus*

Atlikus daug eksperimentų pastebėta, kad sprendžiant daugiakriterinį optimizavimo uždavinį (3.8) pirma strategija, naudoti daugiau nei šešis kompiuterius-darbininkus nėra prasminga, nes atsiranda kompiuterių-darbininkų prastovos – SP nespėja įvertinti gautų sprendinių ir formuoti naujas užduotis.

Sprendžiant daugiakriterinį optimizavimo uždavinį antra strategija, pasiūlyta šiame darbe, tolesniame užduočių formavime dalyvauja ir SP, ir kompiuteris. Šios strategijos tyrimui yra atlikti eksperimentai šešiais, 12 ir 24 kompiuteriais-darbininkais. Šios strategijos ypatumas tas, kad naudojant daug kompiuterių-darbininkų, kai SP nespėja suformuoti užduočių, tai padaro kompiuteris-šeimininkas, tokiu būdu išvengiama kompiuterių-darbininkų prastovų.

Analizuojant rezultatus, pateiktus 4.3 pav., nesudėtinga pastebėti, kad, uždavinį sprendžiant didesniu kompiuterių-darbininkų skaičiumi, greičiau pasiekiami geresni rezultatai jungtinio kriterijaus prasme. Mažiausia jungtinio kriterijaus reikšmė $K_t = 0{,}014$ pasiekta sprendžiant uždavinį šešiais kompiuteriais-darbininkais. Kai uždavinys sprendžiamas dvylika kompiuterių-darbininkų, mažiausia reikšmė $K_t = 0{,}017$, o dvidešimt keturiais $K_t = 0{,}016$. Uždavinį sprendžiant kompiuterių klasteriu, sudarytu iš 12 kompiuterių-darbininkų, pakankamai „geras" rezultatas pasiekiamas 17-ą sprendimo minutę, t. y. jungtinio kriterijaus reikšmės $K_t = 0{,}028$ skirtumas nuo mažiausios yra 0,01. Sprendžiant 24 kompiuteriais-darbininkais toks rezultatas pasiekiamas jau 10-ą minutę. Šešias kompiuteriais-darbininkais, panašus rezultatas yra pasiekiamas tik 20-ą minutę, tačiau tokio kompiuterių-darbininkų skaičiaus naudojimas leidžia sprendimo pabaigoje gauti geriausius rezultatus. Tai galima paaiškinti tuo, kad SP turi daugiau laiko gautų rezultatų įvertinimui ir naujų svorinių koeficientų reikšmių parinkimui. Be to, didesnę dalį užduočių formuoja SP, o ne kompiuteris.



4.3 pav. *Antros strategijos efektyvumo priklausomybė nuo kompiuterių-darbininkų skaičiaus*

Lyginant du atvejus, kai uždavinys sprendžiamas šešiais ir 12 kompiuterių-darbininkų, jungtinio kriterijaus reikšmių procentų vidurkis

$\nu = 84\ \%$. Tačiau atsižvelgiant į rezultatus, gautus iki 20-os minutės $\nu = 65\ \%$, o iki 10-os minutės $\nu = 52\ \%$.

Lyginant atvejus, kai uždavinys sprendžiamas šešiais ir 24 kompiuteriais-darbininkais, jungtinio kriterijaus reikšmių procentų vidurkis $\nu = 67\ \%$. Atsižvelgiant į rezultatus, gautus iki 20-os minutės $\nu = 42\ \%$, o iki 10-os minutės $\nu = 36\ \%$.

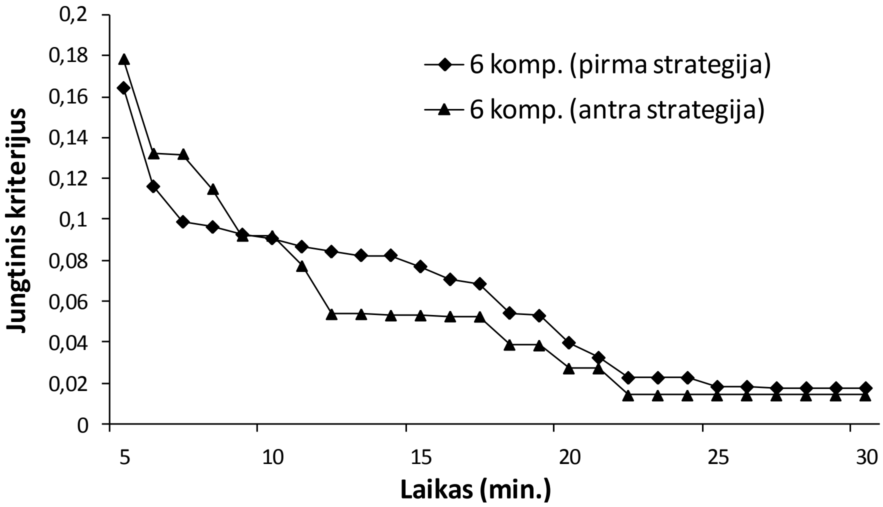
Lyginant atvejus, kai uždavinys sprendžiamas 12 ir 24 kompiuteriais-darbininkais, jungtinio kriterijaus reikšmių procentų vidurkis $\nu = 77\ \%$. Atsižvelgiant į rezultatus, gautus iki 20-os minutės $\nu = 67\ \%$, o iki 10-os minutės $\nu = 71\ \%$.

Gauti rezultatai parodė, kad didesnis kompiuterių-darbininkų skaičius leidžia gauti geresnius rezultatus greičiau, kas sutaupo sprendimo priėmėjo laiką. Daugiau didinti kompiuterių-darbininkų skaičių nėra tikslinga, nes skirtumas tarp rezultatų, gautų kai kompiuterių-darbininkų skaičius yra 12 ir 24 žymiai mažesnis, nei tarp rezultatų, gautų šešiais ir dvylika kompiuterių-darbininkų. Be to, naudojant didesnį kompiuterių-darbininkų skaičių, SP nespėja vertinti gaunamų sprendinių.

4.4 pav. yra vaizduojami abiejų strategijų gauti rezultatai, sprendžiant uždavinį šešiais kompiuteriais-darbininkais. Matome, kad uždavinio sprendimo pradžioje mažesnė jungtinio kriterijaus reikšmė pasiekiama sprendžiant pirma strategija, tačiau pradedant nuo 10-os minutės antra strategija tampa pranašesne. Be to, naudojant antrą strategiją, pasiekta mažiausia jungtinio kriterijaus reikšmė $K_t = 0{,}014$.

Lyginant atvejus, kai uždavinys sprendžiamas šešiais kompiuteriais-darbininkais pirma ir antra strategija, jungtinio kriterijaus reikšmių procentų vidurkis $\nu = 85\ \%$.

Uždavinį sprendžiant antra strategija gaunami geresni rezultatai nei pirma. Tokius rezultatus didžiąja dalimi nulemia SP dalyvavimas parenkant uždavinio sprendimo kelią. Eksperimento pradžioje antros strategijos rezultatai yra blogesni, nes SP tik ieško sprendimo kelio, o kai SP „pažįsta“ uždavinį, pasirenka teisingesnį sprendimo kelią ir gaunami geresni rezultatai.



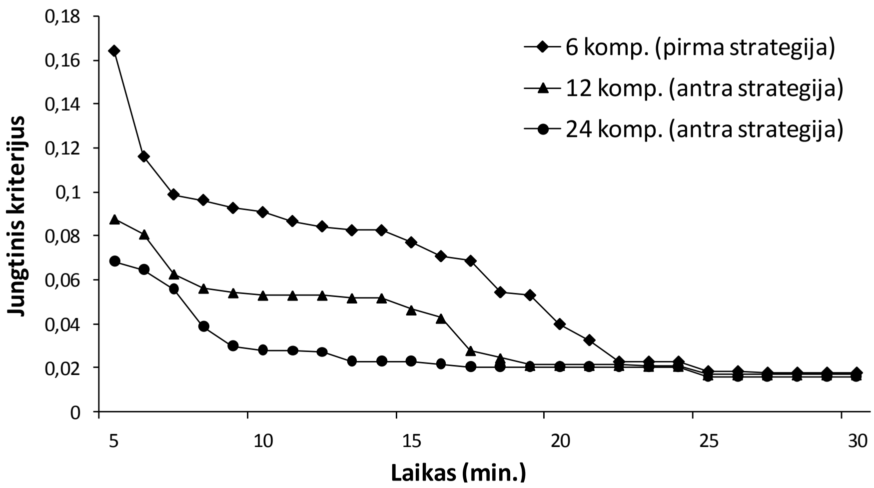
4.4 pav. *Strategijų palyginimas sprendžiant uždavinį šešiais kompiuteriais-darbininkais*

4.5 pav. yra pateikti gauti rezultatai, kai uždavinys sprendžiamas pirma strategija šešiais kompiuteriais-darbininkais ir antra strategija 12 ir 24 kompiuteriais-darbininkais. Analizuojant rezultatus nesudėtinga pastebėti, kad uždavinį sprendžiant antra strategija greičiau pasiekiami geresni rezultatai jungtinio kriterijaus prasme.

Lyginant atvejus, kai uždavinys sprendžiamas šešiais kompiuteriai-darbininkais pirma strategija ir 12 antra strategija, jungtinio kriterijaus reikšmių procentų vidurkis $\nu = 67$ %. Atsižvelgiant į rezultatus, gautus iki 20-os ir iki 10-os minutės $\nu = 56$ %.

Lyginant atvejus, kai uždavinys sprendžiamas šešiais kompiuteriai-darbininkais pirma ir 24 antra strategija, jungtinio kriterijaus reikšmių procentų vidurkis $\nu = 53$ %. Atsižvelgiant į rezultatus, gautus iki 20-os minutės $\nu = 37$ %, o iki 10-os minutės $\nu = 40$ %.

Matome, kad antra strategija yra pranašesnė už pirma ypač uždavinio sprendimo pradžioje.



4.5 pav. *Strategijų palyginimas sprendžiant uždavinį pirma strategija šešiais kompiuteriais-darbininkais ir antra strategija 12 ir 24 kompiuteriais-darbininkais*

4.3. Darbo su sukurtos SPS apsimokymo tyrimas

Atkreipsime dėmesį į tai, kad 4.1 poskyryje pateiktuose tyrimuose sprendimo priėmime dalyvavo vienas patyręs sprendimų priėmėjas, kuris jau turėjo įgūdžių spręsti uždavinį (3.8). Šiame poskyryje pateikti darbo su sukurtos SPS apsimokymo tyrimo rezultatai. Tyrimuose dalyvavo grupė nepatyrusių sprendimų priėmėjų, kurie neturėjo jokių įgūdžių sprendžiant uždavinį (3.8). Šių tyrimų rezultatai paskelbti straipsnyje (A 2).

Tyrimuose siekiama įvertinti sukurtos SPS naudojimo patogumą ir ištirti, kaip greitai SP išmoksta interaktyviai spręsti daugiakriterinius optimizavimo uždavinius, t. y. įsigilina į sprendžiamą uždavinį ir perpranta jo specifiką. SP savo nuožiūra pasirenka tinkamiausią sprendinį, vadovaujantis turima patirtimi, žiniomis apie sprendžiamo uždavinio specifiką ir siekiamais prioritetais bei intuicija. Tiriant sprendimo priėmėjų apsimokymą, reikia skaitinio apsimokymo įvertinimo. Čia kaip ir strategijų palyginime naudotas jungtinis kriterijus (4.3).

Analizuojant SP patirties įtaką bei darbo su SPS apsimokymą, interaktyviai sprendžiant nagrinėjamą uždavinį 3.6 poskyryje aprašytomis strategijomis, tirti penki atvejai:

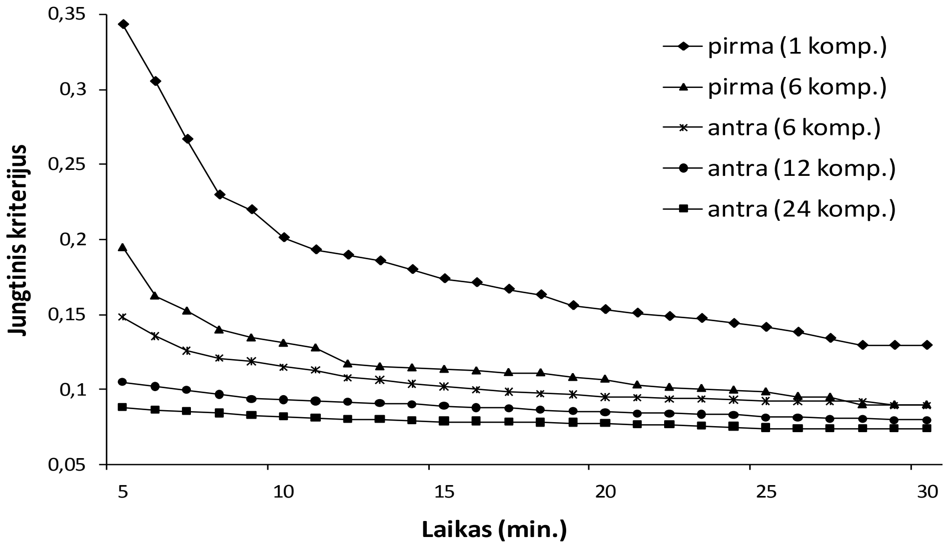
- uždavinys sprendžiamas pirma strategija vienu kompiuteriu-darbininku (toliau vadinsime *pirma (1 komp.)*);
- uždavinys sprendžiamas pirma strategija šešiais kompiuteriais-darbininkais (toliau vadinsime *pirma (6 komp.)*);
- uždavinys sprendžiamas antra strategija šešiais kompiuteriais-darbininkais (toliau vadinsime *antra (6 komp.)*);
- uždavinys sprendžiamas antra strategija 12 kompiuterių-darbininkų (toliau vadinsime *antra (12 komp.)*);
- uždavinys sprendžiamas antra strategija 24 kompiuteriais-darbininkais (toliau vadinsime *antra (24 komp.)*).

Gautų rezultatų vizualizavimui (ir analizei) buvo sukurta speciali programinė įranga, kuri uždavinio sprendimo protokolų duomenis pertvarkė į tinkamus vaizdavimui.

Darbo su SPS apsimokymo eksperimentuose dalyvavo 50 sprendimo priėmėjų. Kiekvienoje strategijoje su pasirinktu kompiuterių skaičiumi dalyvavo po 10 sprendimo priėmėjų. Kiekvienas SP atliko po 10 eksperimentų, o vieno eksperimento trukmė buvo 30 minučių. Taigi vienas SP eksperimente dalyvavo ne mažiau kaip penkias valandas. Vien tik eksperimentams atlikti buvo reikalingos didžiulės žmogaus darbo laiko sąnaudos.

Kiekvieną uždavinio sprendimo minutę skaičiuojama pasiekta jungtinio kriterijaus minimali reikšmė. Apskaičiuojamos vidutinės jungtinio kriterijaus reikšmės vidurkinant 10 sprendimų priėmėjų 10 eksperimentų rezultatus, kai uždavinys sprendžiamas ankščiau aprašytais atvejais.

Gauti rezultatai pateikti 4.6 pav. Matome, kad mažiausia jungtinio kriterijaus reikšmė $K_t = 0{,}074$ gauta, sprendžiant uždavinį 24 kompiuteriais-darbininkais antra strategija. Be to, tuo atveju pakankamai maža jungtinio kriterijaus reikšmė $K_t = 0{,}087$ gaunama jau penktą sprendimo minutę. Uždavinio sprendimo pabaigoje didžiausia jungtinio kriterijaus reikšmė $K_t = 0{,}140$ gaunama sprendžiant vienu kompiuteriu-darbininku pirma strategija. Gauti rezultatai dar kartą patvirtina, kad antra strategija yra pranašesnė už pirmą.

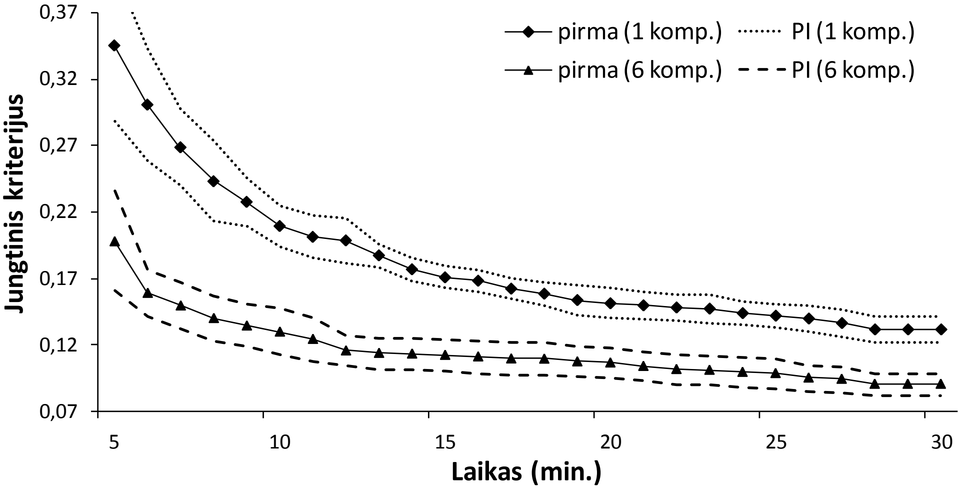


4.6 pav. *Vidutinės jungtinio kriterijaus reikšmės*

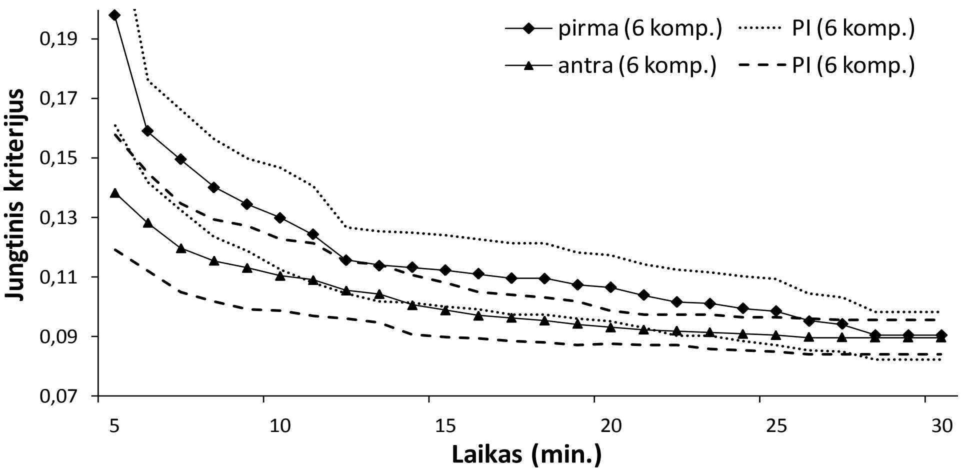
4.7–4.10 pav. pateiktos vidutinės jungtinio kriterijaus reikšmės kartu su pasikliautinaisiais intervalais PI (pasikliovimo lygmuo 0,95). Kad neperkrauti grafiką ir suprantamiau pateikti gautus rezultatus, atvejai pateikiami po du.

4.7 pav. matome, kad jungtinio kriterijaus vidurkių pasikliautinieji intervalai nepersipina, tas patvirtina, kad sprendžiant uždavinį pirma strategija šešių kompiuterių-darbininkų naudojimas leidžia gauti geresnius rezultatus, lyginant su rezultatais, gautais vienu kompiuteriu-darbininku. Be to, jungtinio kriterijaus reikšmių procentų vidurkis $\nu = 64$ %, jei laikysime, kad atveju *pirma (1 komp.)* jungtinio kriterijaus reikšmių, pasiektų kiekvieną uždavinio sprendimo minutę vidurkis yra 100 %.

Lyginant rezultatus, gautus sprendžiant uždavinį šešiais kompiuteriais-darbininkais pirma ir antra strategija, matome, kad jungtinio kriterijaus vidurkiai gauti antra strategija yra mažesni viso sprendimo metu (4.8 pav.). Tačiau vidurkių pasikliautinieji intervalai persipina, todėl šiuo atveju negalima griežtai teigti, kad antra strategija pranašesnė.

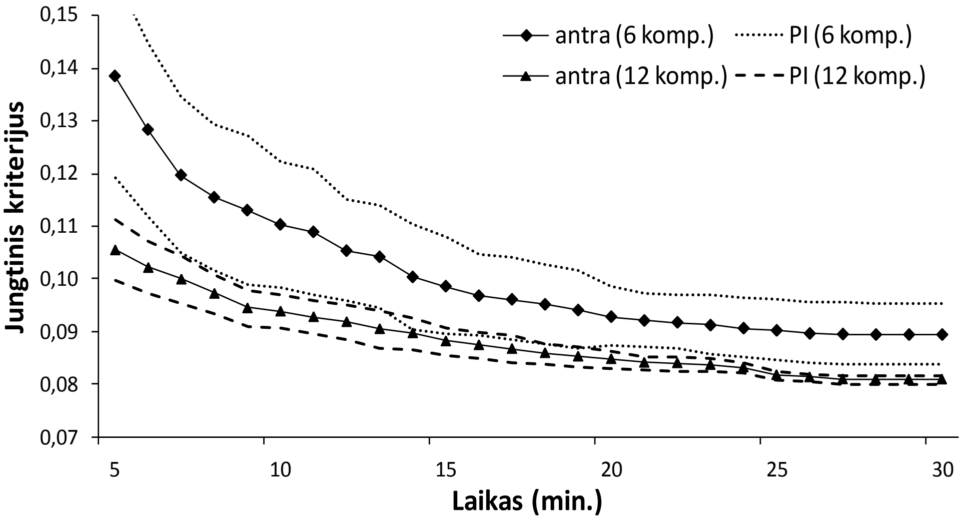


4.7 pav. *Vidutinės jungtinio kriterijaus reikšmės su pasikliautinaisiais intervalais atvejams pirma (1 komp.) ir pirma (6 komp.)*



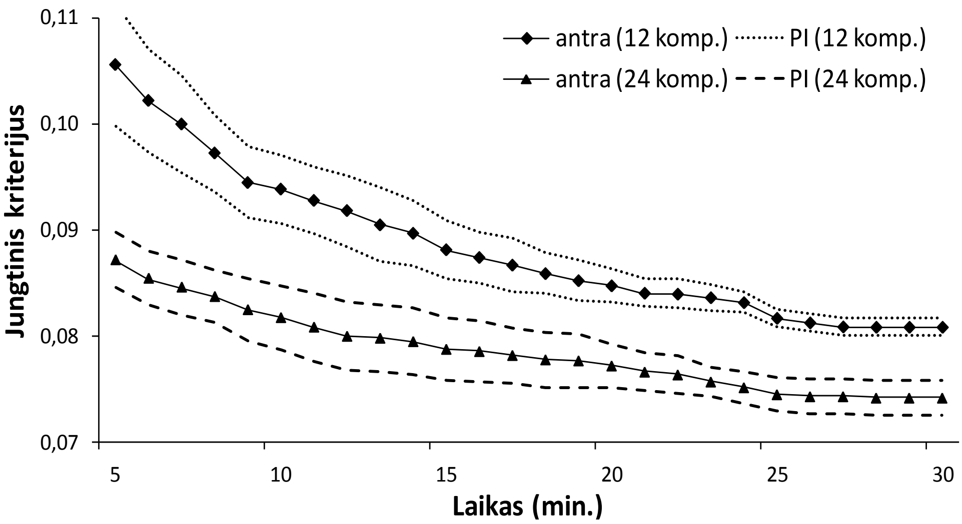
4.8 pav. *Vidutinės jungtinio kriterijaus reikšmės su pasikliautinaisiais intervalais atvejams pirma (6 komp.) ir antra (6 komp.)*

Kai uždavinys sprendžiamas šešiais ir 12 kompiuterių-darbininkų antra strategija, jungtinio kriterijaus vidurkiai, gauti naudojant 12 kompiuterių yra mažesni viso sprendimo metu (4.9 pav.). Vidurkių pasikliautinieji intervalai kai kuriais laiko momentais (uždavinio sprendimo viduryje nuo 7 iki 20 minutės) susiliečia, tačiau sprendimo pabaigoje to nėra. Todėl galima teigti, kad šešias kompiuteriais-darbininkais gaunami blogesni rezultatai jungtinio kriterijaus prasme. Jungtinio kriterijaus, gauto atveju *antra (12 komp.)*, reikšmių procentų vidurkis $\nu = 86$ % (nuo atvejo *antra (6 komp.)*).



4.9 pav. *Vidutinės jungtinio kriterijaus reikšmės su pasikliautinaisiais intervalais atvejams antra (6 komp.) ir antra (12 komp.)*

4.10 pav. matome, kad jungtinio kriterijaus vidurkių pasikliautinieji intervalai nepersipina, tas patvirtina, kad sprendžiant uždavinį antra strategija, 24 kompiuterių-darbininkų naudojimas leidžia gauti geresnius rezultatus negu 12 kompiuterių-darbininkų atveju. Jungtinio kriterijaus reikšmių procentų vidurkis $\nu = 89$ %, jei laikysime, kad atveju *antra (12 komp.)* jungtinio kriterijaus reikšmių, pasiektų kiekvieną uždavinio sprendimo minutę vidurkis yra 100 %.



4.10 pav. *Vidutinės jungtinio kriterijaus reikšmės su pasikliautinaisiais intervalais atvejams antra (12 komp.) ir antra (24 komp.)*

Svarbu palyginti, kaip išnaudoja kompiuterius-darbininkus patyręs ir nepatyrę sprendimų priėmėjai. 4.2 lentelėje pateikti jungtinių kriterijų reikšmių procentų vidurkiai ν, apskaičiuoti lyginant skirtingą kompiuterių-darbininkų skaičių, sprendžiant uždavinį pirma ir antra strategijomis. Matome, beveik visais lygintais atvejais patyrusio SP gautų jungtinių kriterijų reikšmių procentų vidurkiai ν yra mažesni, negu vidurkiai, kai uždavinį sprendžia nepatyrę sprendimų priėmėjai. Taigi patyręs SP sugeba geriau išnaudoti didesnį kompiuterių-darbininkų skaičių. Be to, ir patyręs, ir nepatyrę SP geriau išnaudoja kompiuterius-darbininkus uždavinio sprendimo pradžioje. Kai lyginami atvejai *pirma (6 komp.)* ir *antra (12 komp.)* sprendimo pradžioje (10 min.) nepatyrusių SP procentų vidurkis yra mažesnis negu patyrusio SP. Tai galima paaiškinti tuo, kad patyręs SP žymiai geriau išnaudoja šešis kompiuterius-darbininkus (atvejai *pirma (1 komp.)* ir *pirma (6 komp.)*), kai uždavinys sprendžiamas pirma strategija ir užduotis formuoja tik SP. Uždavinį sprendžiant 12 kompiuterių-darbininkų antra strategija, rezultatai patyrusiam SP pagerėja mažiau negu nepatyrusiems SP.

4.2 lentelė. *Jungtinio kriterijaus reikšmių procentų vidurkis* ν *lyginant patyrusio (P) ir nepatyrusių (N) sprendimų priėmėjų gautus rezultatus*

Laiko momentas (minutės)	*pirma (1 komp.)* ir *pirma (6 komp.)*		*pirma (6 komp.)* ir *antra (12 komp.)*		*pirma (6 komp.)* ir *antra (24 komp.)*		*antra (6 komp.)* ir *antra (12 komp.)*		*antra (12 komp.)* ir *antra (24 komp.)*	
	P	N	P	N	P	N	P	N	P	N
10 min.	34 %	58 %	56 %	53 %	40 %	47 %	52 %	78 %	71 %	86 %
20 min.	51 %	62 %	56 %	66 %	37 %	59 %	65 %	83 %	67 %	88 %
30 min.	51 %	64 %	67 %	73 %	53 %	65 %	84 %	86 %	77 %	89 %

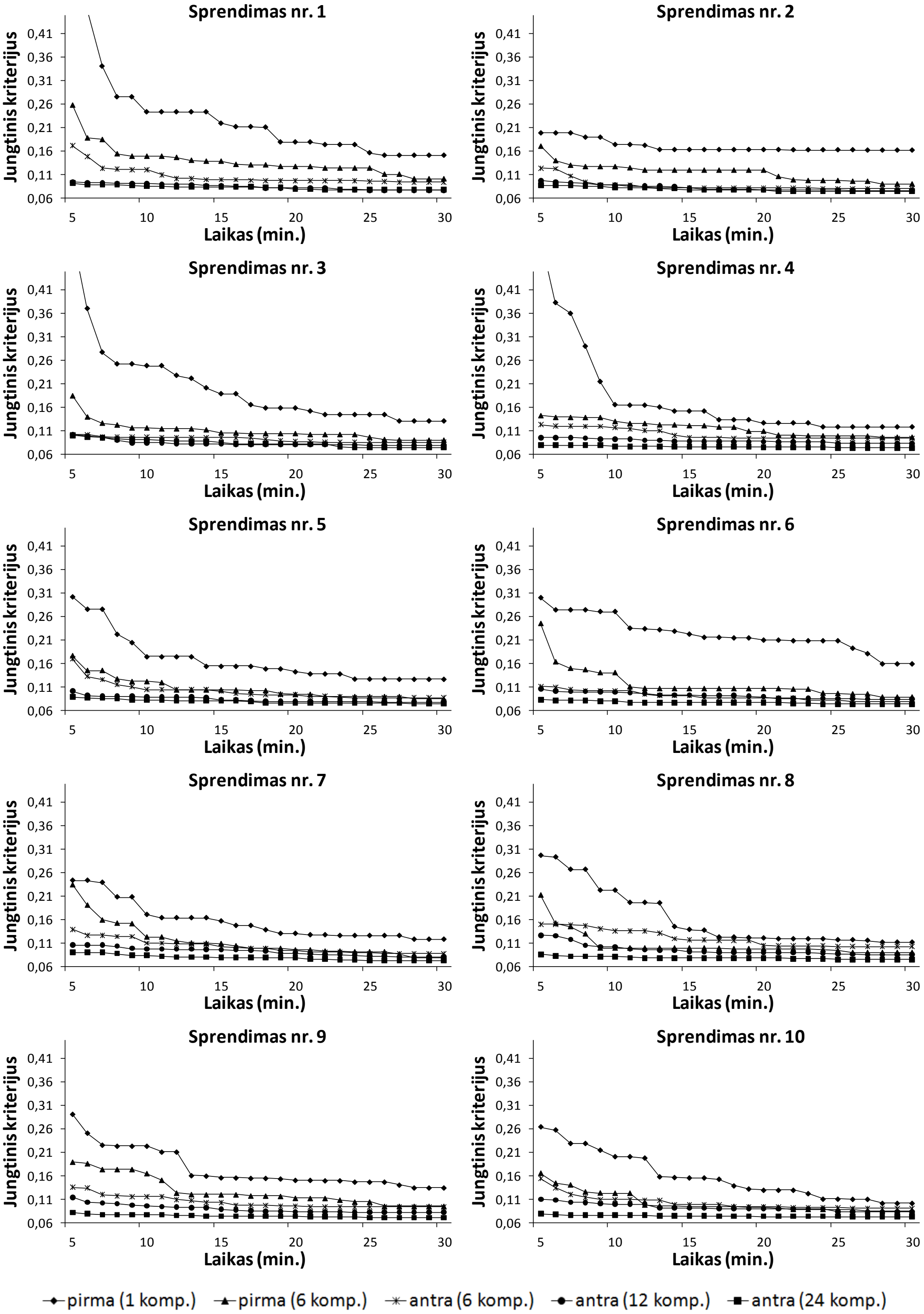
Toliau pateikiama gautų rezultatų priklausomybė nuo sprendimų priėmėjo įgytos patirties eksperimentų metu, t. y. kokie gaunami rezultatai, kai SP interaktyviai sprendžia uždavinį pirmą kartą, antrą kartą ir t. t., ar jam pavyksta apsimokyti, perprasti sprendžiamo uždavinio specifiką ir uždavinį kiekvieną kartą spręsti geriau. Čia gauti rezultatai lyginami jungtinio kriterijaus prasme.

4.11 pav. pateiktos vidutinės jungtinio kriterijaus reikšmės, gautos uždavinį sprendžiant pirmą kartą (Sprendimas nr. 1), antrą kartą (Sprendimas nr. 2) ir t. t. Analizuojant rezultatus pagal eksperimentų atlikimo tvarką, galime stebėti sprendimų priėmėjų apsimokymą ir apsimokymo įtaką tolesnių eksperimentų rezultatams, galime vertinti SP patirtį, kai daugiakriterinis optimizavimo uždavinys sprendžiamas skirtingomis strategijomis ir naudojant skirtingą kompiuterių skaičių.

Kai SP sprendžia uždavinį pirmą kartą (4.11 pav. Sprendimas nr. 1), esminio skirtumo tarp strategijų ir didesnio kompiuterių-darbininkų skaičiaus naudojimo nėra, nes SP dar neturi pakankamai įgūdžių ir jis neišnaudoja didesnio kompiuterių-darbininkų skaičiaus privalumo. Kai uždavinys sprendžiamas antrą kartą matome, kad visais atvejais gaunami geresni rezultatai nei pirmą kartą. Ženkliai geresni rezultatai gaunami uždavinį sprendžiant antra strategija – labiausiai tai pastebima lyginant pirmos ir antros strategijos rezultatus, kai uždavinys sprendžiamas šešiais kompiuteriais-darbininkais.

Svarbu pastebėti ir tai, kad uždavinį sprendžiant antrą kartą, kai naudojami 12 ir 24 kompiuteriai-darbininkai, gauti rezultatai mažai skiriasi lyginant su rezultatais, gautais naudojant tik šešis kompiuterius-darbininkus. Galima daryti išvadą, kad SP antro sprendimo metu dar nepakako laiko apsimokyti spręsti uždavinį, naudojant daug kompiuterių-darbininkų.

Tolesniuose sprendimuose (4.11 pav. Sprendimas nr. 3–10) matome, kad rezultatai, uždavinį sprendžiant šešiais kompiuteriais-darbininkais, mažai skiriasi lyginant su gautais antro sprendimo metu. Galima teigti, kad SP jau apsimokė spręsti uždavinį naudojant tokį kompiuterių-darbininkų skaičių pirmo sprendimo metu.



4.11 pav. *Vidutinės jungtinio kriterijaus reikšmės priklausomai nuo eksperimentinio sprendimo numerio*

Analizuojant kreives, pateiktas 4.11 pav., kai uždavinys sprendžiamas 12 ir 24 kompiuteriais-darbininkais, pastebime, kad iki penkto sprendimo gauti rezultatai labai panašūs, o vėliau skiriasi ir geresni rezultatai gaunami naudojant 24 kompiuterius-darbininkus bei gerėja iki dešimto (paskutinio) sprendimo. Galima daryti išvadą, kad sprendimų priėmėjai interaktyviai spręsdami daugiakriterinį optimizavimo uždavinį greičiau apsimoko, kai sprendimui naudojamas didesnis kompiuterių-darbininkų skaičius, išskyrus atvejį, kai naudojami 24 kompiuteriai-darbininkai, tačiau tuomet galiausiai pasiekiami geriausi rezultatai. Apsimokymas užtrunka ilgiau naudojant mažesnį kompiuterių-darbininkų skaičių.

Galima teigti, kad uždavinio sprendimui 24 kompiuteriai-darbininkai yra ribinis skaičius ir neverta jo didinti, nes sprendimų priėmėjui bus sudėtinga interaktyviai spręsti uždavinį, jis nespės tinkamai įvertinti iš kompiuterių-darbininkų gautų tarpinių sprendinių ir formuoti naujas užduotis.

Toliau pateikiama eksperimentuose gautų duomenų analizė, kai stebimos jungtinio kriterijaus reikšmės sprendimo pabaigoje. Vieno eksperimentinio sprendimo trukmė buvo 30 minučių, todėl analizuojame iki to laiko momento gautus geriausius rezultatus. Kiekvieno eksperimento sprendimo pabaigoje fiksuojamos geriausios gautos reikšmės, tuomet apskaičiuojamos vidutinės jungtinio kriterijaus reikšmės kiekvienai strategijai naudojant pasirinktą kompiuterių-darbininkų skaičių priklausomai nuo eksperimentinio sprendimo numerio. Rezultatai pateikti 4.3 lentelėje. Paryškintu šriftu pažymėtos trys mažiausios jungtinio kriterijaus reikšmės kiekvienam tirtam atvejui.

Matome, kad sprendžiant uždavinį pirma strategija naudojant tik vieną kompiuterį-darbininką, geriausi rezultatai gauti paskutiniųjų sprendimų metu, patirtis įgyjama pakankamai lėtai. To priežastimi yra santykinai nedidelis tarpinių rezultatų skaičius, todėl sprendimų priėmėjui apsimokyti reikia daugiau laiko. SP, sprendžiant uždavinį pirma strategija šešiais kompiuteriais-darbininkais, apsimoko greičiau (5-7 sprendimas), nes gauna ir gali įvertinti žymiai didesnį skaičių tarpinių sprendinių.

Kai sprendimų priėmėjui užduotis padeda formuoti ir kompiuteris-šeimininkas (*antra (6 komp.)*), pasiekiami dar geresni rezultatai. SP dar greičiau apsimoko uždavinį sprendžiant antra strategija naudojant 12 kompiuterių. To priežastis yra ta, kad SP turi galimybę analizuoti daug tarpinių sprendinių. Tačiau, kai SP sprendžia uždavinį 24 kompiuteriais, apsimokymas vyksta lėčiau ir rezultatai iki penkto sprendimo panašūs į rezultatus, gautus naudojant 12 kompiuterių, o pradedant šeštu sprendimu rezultatai „aplenkia“ kitus atvejus. Galima daryti išvadą, kad naudojant daug kompiuterių-darbininkų (24), SP mokymasis priimti sprendimus vyksta lėtai, bet galiausiai gaunami geresni rezultatai.

4.3 lentelė. *Jungtinio kriterijaus K_t vidutinės reikšmės priklausomai nuo sprendimo numerio*

Sprendimo numeris	Atvejis					Bendras vidurkis
	pirma (1 komp.)	*pirma (6 komp.)*	*antra (6 komp.)*	*antra (12 komp.)*	*antra (24 komp.)*	
1	0,1547	0,0920	0,0943	**0,0770**	0,0778	0,0991
2	0,1441	0,0923	**0,0801**	**0,0758**	0,0742	0,0933
3	0,1436	0,0935	**0,0857**	0,0793	0,0747	0,0954
4	0,1246	0,0988	0,0939	0,0832	0,0740	0,0949
5	**0,1144**	**0,0822**	0,0872	**0,0776**	0,0742	**0,0871**
6	0,1779	**0,0842**	**0,0790**	0,0835	0,0736	0,0996
7	0,1209	**0,0762**	0,0876	0,0806	**0,0732**	**0,0877**
8	**0,1045**	0,0974	0,1022	0,0853	0,0752	0,0929
9	0,1289	0,0915	0,0940	0,0828	**0,0717**	0,0938
10	**0,0987**	0,0880	0,0912	0,0835	**0,0730**	**0,0869**

4.4. Pasiūlyto interaktyvaus sprendimo būdo tyrimas

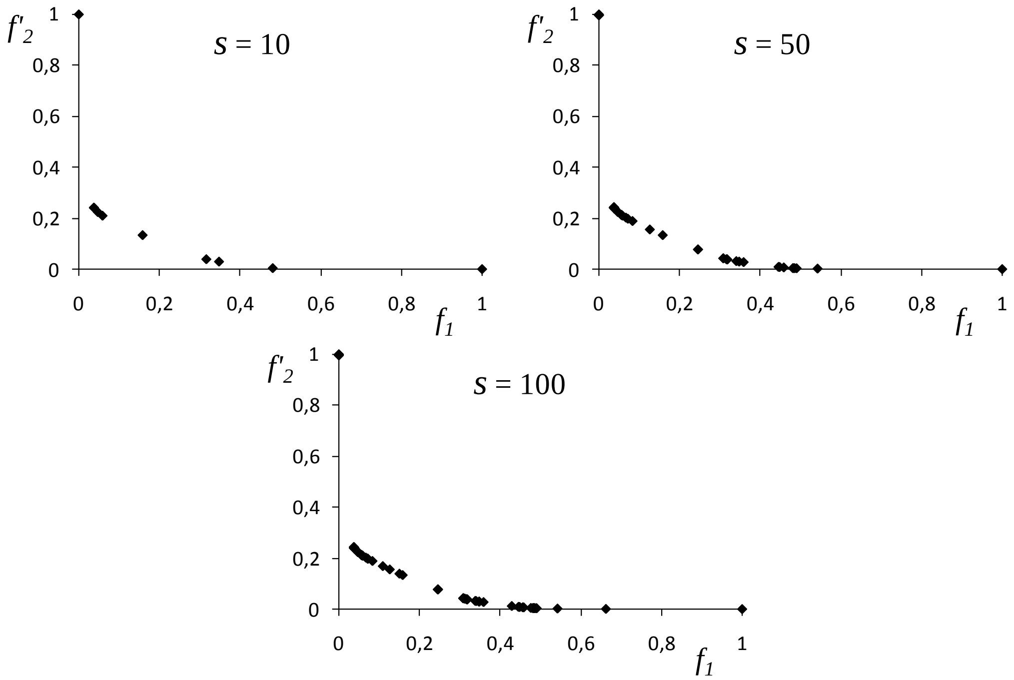
Šiame poskyryje pateikti tyrimų rezultatai, gauti pasiūlytu interaktyviu daugiakriterinių optimizavimo uždavinių sprendimo būdu, apjungiančiu SKS ir ASKS metodus (3.1 poskyris), sprendžiant pašarų sudėties sudarymo uždavinį (3.8). Šių tyrimų rezultatai paskelbti straipsnyje (A 4).

4.4.1. ASKS metodo tyrimas

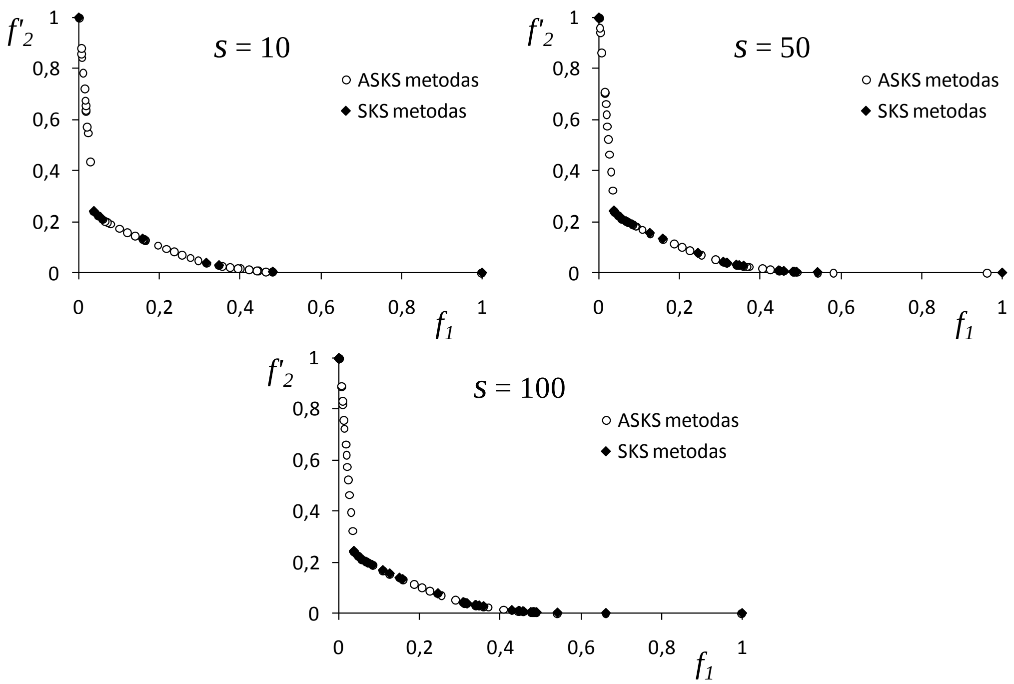
Kaip jau buvo minėta 2.3.4 skyrelyje, ASKS metodas padeda surasti Pareto optimalius sprendinius, kurie būtų tolygiai pasiskirstę Pareto aibėje. Tačiau dėl metodo reikalaujamų didelių skaičiavimo resursų ir įgyvendinimo sudėtingumo, kai kriterijų skaičius yra daugiau nei trys, čia sprendžiamas dviejų kriterijų uždavinys (3.11). Pirmas kriterijus – pašarų savikaina, o kitas – maistinių charakteristikų leistinų normų pažeidimų suma. Šių kriterijų reikšmės yra normuojamos intervale $[0; 1]$.

Pradžioje tirtas sprendinių pasiskirstymas Pareto aibėje, kai uždavinys (3.11) sprendžiamas SKS metodu ir kintamųjų pradinės reikšmės, sudarančios vektorių X_0, yra fiksuotos. Svorinio koeficiento w_1' reikšmės yra parenkamos nuo 0 iki 1 skirtingu žingsniu s, o svorinio koeficiento w_2' reikšmės yra apskaičiuojamos pagal formulę: $w_2' = 1 - w_1'$. Buvo atlikti eksperimentai su trimis skirtingomis žingsnio s reikšmėmis ($s = 10; 50; 100$). Gauti rezultatai pateikti 4.12 pav. Akivaizdu, kad sprendiniai nėra tolygiai pasiskirstę Pareto aibėje. Nors didinant žingsnio s reikšmę sprendinių skaičius didėja, tačiau dalis sprendinių sutampa, arba jie yra labai arti vienas kito. Matome, kad kai kuriose Pareto aibės dalyse sprendinių nebuvo rasta.

Gavus SKS metodo rezultatus, toliau uždavinys yra sprendžiamas ASKS metodu, kuriuo randami sprendiniai tose Pareto aibės srityse, kuriose nebuvo rasta SKS metodu. Rezultatai pateikti 4.13 pav. Matome, kad buvo gauta papildomų sprendinių, kurie tolygiau užpildo Pareto aibę. Tačiau dalyje Pareto aibės nėra gauta sprendinių, kadangi ten dėl kriterijų ypatumų sprendinių neegzistuoja.



4.12 pav. *Sprendiniai, gauti SKS metodu, sprendžiant uždavinį su skirtingomis žingsnio s reikšmėmis* ($s = 10; 50; 100$)



4.13 pav. *Sprendiniai, gauti ASKS metodu, sprendžiant uždavinį su skirtingomis žingsnio s reikšmėmis* ($s = 10; 50; 100$)

Gautiems sprendiniams yra paskaičiuotos pasiskirstymo įverčio φ (2.33) reikšmės, siekiant parodyti sprendinių pasiskirstymo tolygumą Pareto aibėje. Kaip minėta 2.4.4 skyrelyje, kuo mažesnė šio įverčio reikšmė, tuo sprendiniai tolygiau padengia Pareto aibę. 4.4 lentelėje matome, kad įverčio reikšmės gaunamos žymiai mažesnės taikant ASKS metodą, esant tai pačiai s reikšmei.

4.4 lentelė. *Pasiskirstymo įverčio φ reikšmės taikant SKS ir ASKS metodus*

s reikšmė	SKS metodas		ASKS metodas	
	Pareto aibės taškų skaičius n'	φ reikšmė	Pareto aibės taškų skaičius n'	φ reikšmė
10	11	0,261	45	0,084
50	37	0,132	52	0,056
100	66	0,110	52	0,053

Pareto aibės sprendinių skaičius n' yra mažesnis už s reikšmę, nes dalis gautų sprendinių sutampa. SKS metodu gautų sprendinių φ reikšmė yra didesnė net esant didelei n' reikšmei ($n' = 66$), lyginant su ASKS metodu gauta reikšme, kai n' reikšmė yra mažesnė ($n' = 45$).

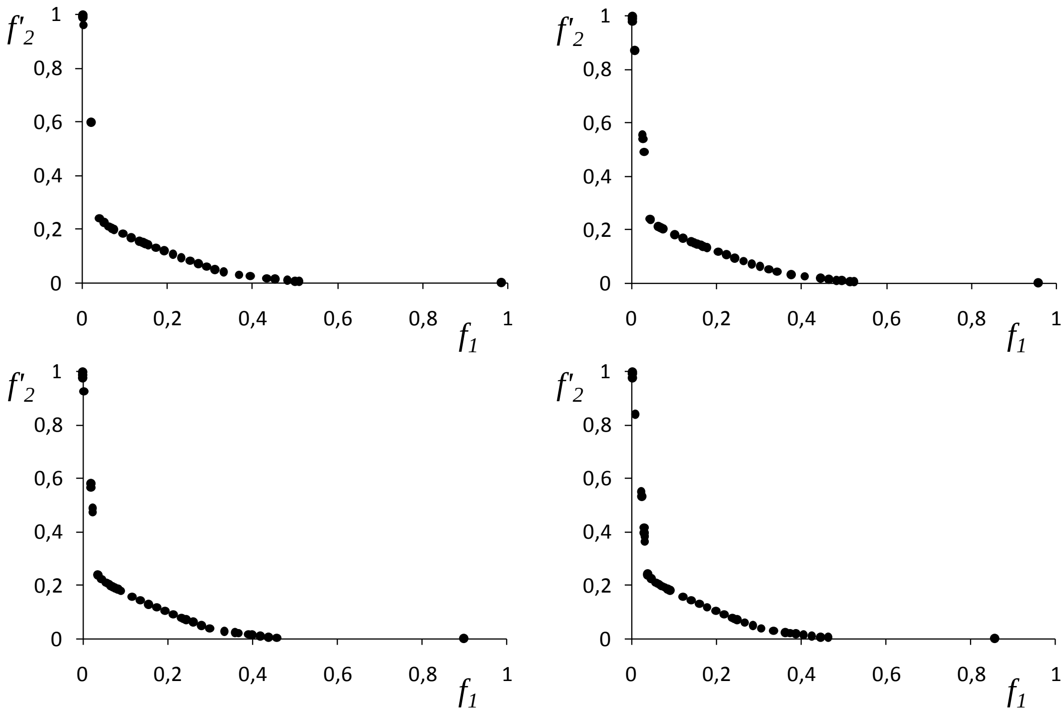
Be to, šis tyrimas parodė, jog pakanka nedidelės s reikšmės, kad taikant ASKS metodą, gauti pakankamą skaičių skirtingų tolygiai pasiskirsčiusių sprendinių: kai žingsnis $s = 10$, Pareto aibės taškų skaičius $n' = 45$, o kai žingsnis $s = 100$, Pareto aibės taškų skaičius $n' = 52$. Tuo atveju įverčio φ reikšmė skiriasi nežymiai: $\varphi = 0{,}084$, kai žingsnis $s = 10$, ir $\varphi = 0{,}053$, kai žingsnis $s = 100$. Taigi kitame tyrime, taupant skaičiavimų laiką, buvo parinktas žingsnis $s = 10$.

Toliau, naudojant ASKS metodą, buvo ištirtas gautų sprendinių pasiskirstymas priklausomai nuo kintamųjų pradinių reikšmių, sudarančių vektorių X_0. 4.5 lentelėje pateikiami dešimties eksperimentų labiausiai išsiskiriantys rezultatai. Matome, kad gautų sprendinių skaičius ir sprendinių pasiskirstymas Pareto aibėje šiek tiek skiriasi.

4.5 lentelė. *Pasiskirstymo įverčio reikšmės ir Pareto aibės taškų skaičius, kai žingsnis* $s = 10$

Pareto aibės taškų skaičius n'	φ reikšmė
42	0,096
42	0,101
41	0,077
39	0,109
39	0,092
39	0,091
38	0,096
37	0,098
35	0,112
33	0,097

4.14 pav. pateikiami keturi vizualiai labiausiai išsiskiriantys rezultatai sprendžiant (3.11) uždavinį ASKS metodu, kai parinktos skirtingos pradinės reikšmės, sudarančios vektorių X_0, o žingsnis $s = 10$.



4.14 pav. *Sprendiniai, gauti ASKS metodu, sprendžiant uždavinį su skirtingais* X_0*, kai žingsnis* $s = 10$

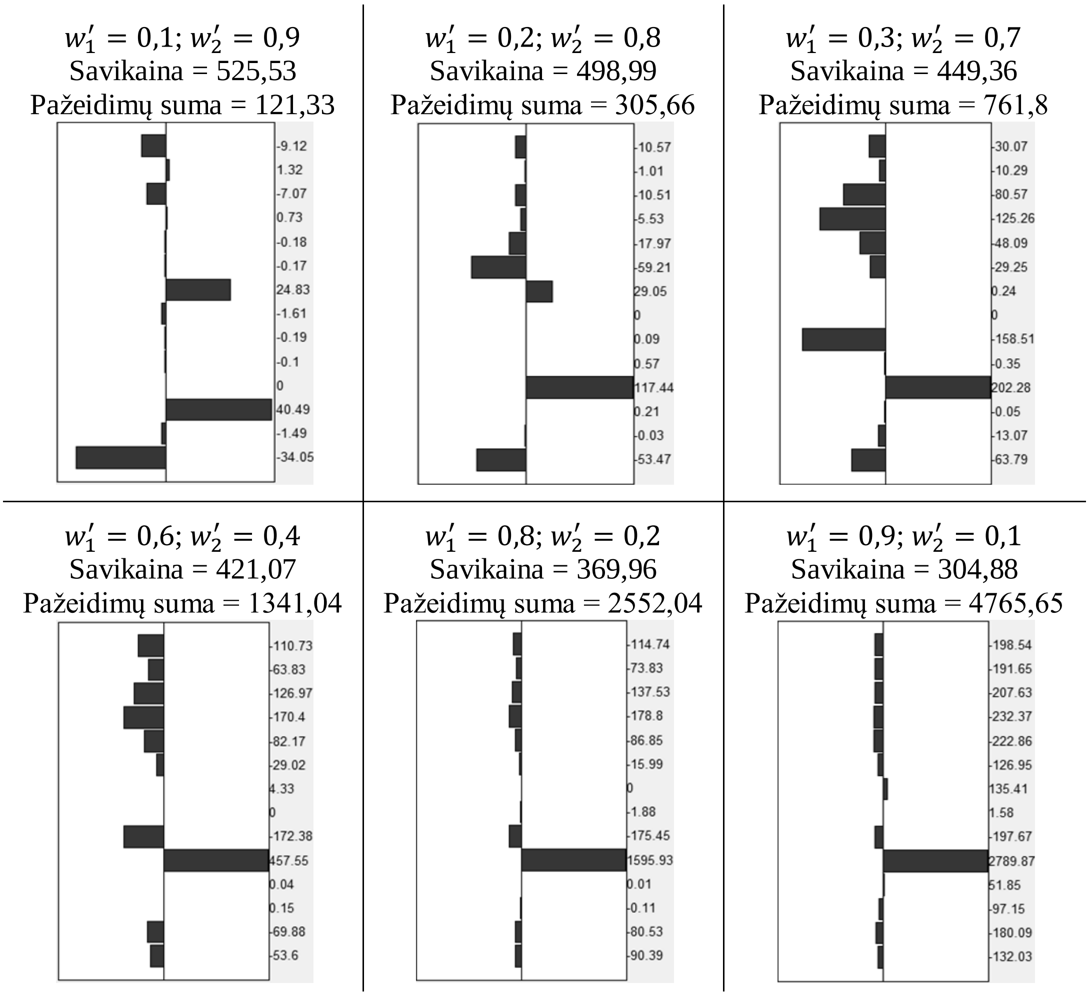
Taigi atsižvelgiant į eksperimentų rezultatus, pateiktus 4.5 lentelėje ir 4.14 pav. galima teigti, kad pradinių reikšmių, sudarančių vektorių X_0 parinkimas turi įtakos uždavinio (3.11) gaunamiems sprendiniams. Taip pat, atsižvelgiant į gautus rezultatus, daugiausiai sprendinių gaunama vienoje Pareto aibės dalyje, kai $f_1 \in (0{,}05;\ 0{,}55)$ ir $f_2' \in (0{,}004;\ 0{,}23)$. Kriterijaus $f_1(X)$ reikšmės kinta didesniame intervale negu kriterijaus $f_2'(X)$ reikšmės. Taigi nedaug sumažėjus maistinių charakteristikų leistinų normų pažeidimų sumai (f_2'), savikaina (f_1) ženkliai padidėja.

4.4.2. Interaktyviu sprendimo būdu gaunamų rezultatų analizė

Pasiūlyto interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo esmė yra suteikti sprendimų priėmėjui galimybę gauti sprendinius iš visos Pareto aibės. SP gauna Pareto aibėje tolygiai išsidėsčiusius alternatyvius sprendinius, ko nepavyksta pasiekti taikant SKS metodą, kai SP pats formuoja užduotis. Be to, pasirinkus labiausiai tenkinančius sprendinius, gautus ASKS metodu, SP gali siekti juos pagerinti, nurodant savo prioritetus kriterijams, koreguojant svorinių koeficientų reikšmes.

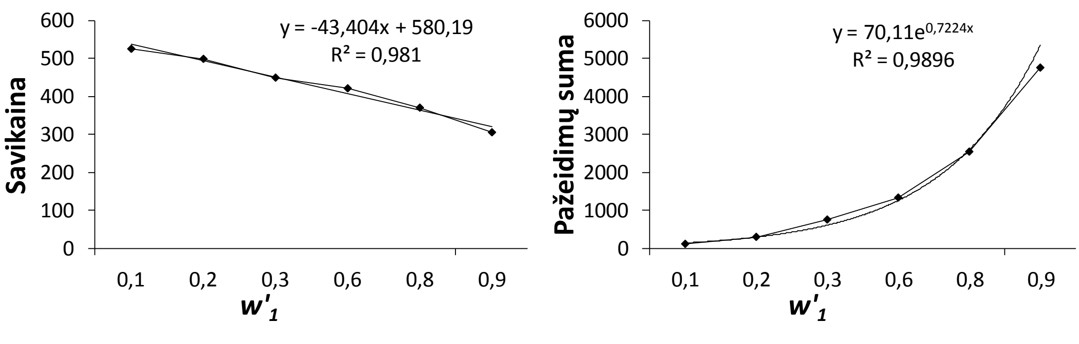
ASKS metodu išsprendžiamas dviejų kriterijų uždavinys (3.11). Apskaičiuojamos savikainos ir 14-os maistinių charakteristikų leistinų normų pažeidimai ir gauti rezultatai (4.15 pav.) pateikiami sprendimų priėmėjui.

Kadangi sprendžiamas dviejų kriterijų uždavinys, tai nebuvo atsižvelgiama į kiekvienos maistinės charakteristikos leistinos normos pažeidimą, o tik į pažeidimų sumą. Tačiau SP turės informacijos apie galimas savikainos ir pažeidimų sumos reikšmes, kurių gali siekti keičiant prioritetus (svorinių koeficientų reikšmes) atskiroms maistinėms charakteristikoms ir savikainai, t. y. sprendžiant 15-os kriterijų uždavinį SKS metodu. Kaip matome 4.15 pav., didinant svorinio koeficiento w_1' reikšmę, savikaina mažėja, o pažeidimų suma didėja, be to pažeidimų suma didėja daug sparčiau, nei mažėja savikaina.



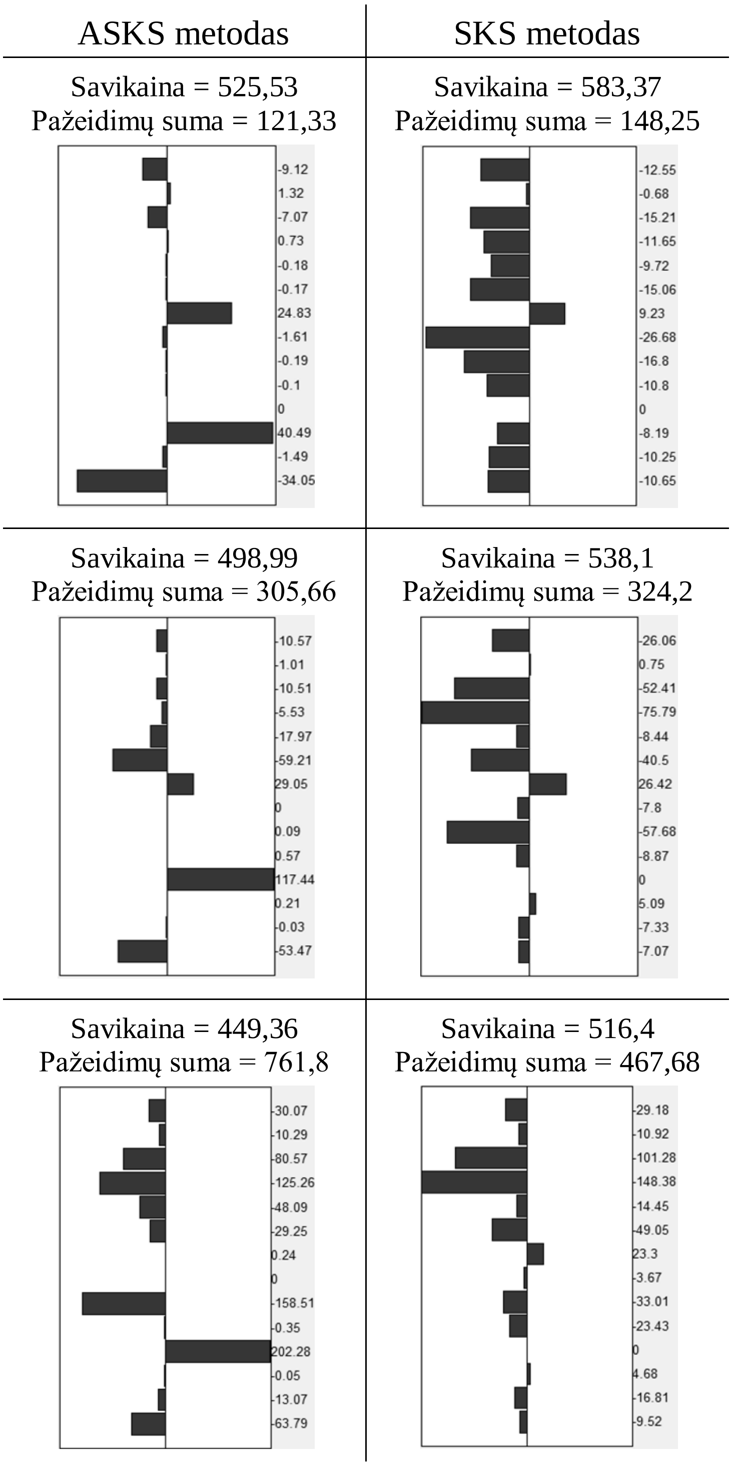
4.15 pav. *Sprendinių pavyzdžiai, gauti ASKS metodu ir pateikti sprendimų priėmėjo vertinimui*

4.16 pav. pateikta savikainos ir pažeidimų sumos priklausomybė nuo svorinių koeficientų reikšmių. Matome, kad tiesiškai mažėjant savikainai, pažeidimų suma auga eksponentiškai.



4.16 pav. *Savikainos ir pažeidimų sumos priklausomybė nuo svorinių koeficientų reikšmių*

4.17 pav. pateikti šiame darbe pasiūlyto interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo rezultatai.



4.17 pav. *Sprendinių pavyzdžiai, gauti pasiūlytu interaktyviu sprendimo būdu*

Akivaizdu, kad ASKS metodu gauti rezultatai skirsis nuo rezultatų, kuriuos SP gavo pats keisdamas svorinių koeficientų reikšmes, uždavinį sprendžiant SKS metodu. SP gavęs sprendinį ASKS metodu, keisdamas svorinių koeficientų reikšmes, uždavinį sprendė SKS metodu, siekdamas gauti sprendinį, tenkinantį jo prioritetus. Kairėje pusėje pateikti ASKS metodu gauti sprendiniai, dešinėje SKS metodu gauti sprendiniai po eilės spendimo priėmėjo svorinių koeficientų reikšmių koregavimo. Matome, kad pažeidimų suma padidėja, tačiau tam tikri pažeidimai sumažėjo, kas yra svarbu sprendimų priėmėjui.

4.5. Ketvirtojo skyriaus rezultatai ir išvados

Šiame skyriuje pateikti eksperimentinių tyrimų rezultatai. Nagrinėti rezultatai, kai į tikslo funkciją įtraukiamas vienas iš apribojimų. Atlikta dviejų sprendimų strategijų lyginamoji analizė. Pirmoje strategijoje užduotis formuoja (parenka svorinių koeficientų reikšmes) tik sprendimų priėmėjas, o antroje, šiame darbe pasiūlytoje strategijoje, užduotis formuoja ir kompiuteris. Tirta, kaip greitai SP išmoksta interaktyviai spręsti daugiakriterinį optimizavimo uždavinį naudojant sukurtą SPS, įsigilina į uždavinį ir pažįsta jo specifiką. Palyginti rezultatai, gauti, kai uždavinį sprendžia ir patyręs sprendimų priėmėjas, ir nepatyrę. Taip pat pateikti sukurto interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo, apjungiančio SKS ir ASKS metodus, eksperimentinių tyrimų rezultatai.

Eksperimentiškai ištyrus praktinio uždavinio tikslo funkcijos formuluotę, kai vienas iš apribojimų yra įtraukiamas į tikslo funkciją su baudos koeficientu, nustatyta, kad kuo mažesnė baudos koeficiento b_{k} reikšmė, tuo gautų sprendinių savikaina mažesnė, o pažeidimų suma didesnė. Kai baudos koeficiento reikšmė yra didelė ($b_{\mathrm{k}} \geq 10^6$), gauti rezultatai beveik nesiskiria nuo rezultatų, gautų sprendžiant uždavinį, kai minėtas apribojimas nėra įtraukiamas į tikslo funkciją. Kadangi baudos koeficiento b_k reikšmė įtakoja gaunamus rezultatus, todėl siūloma spręsti uždavinį su papildomu apribojimu.

Ištyrus sprendimo strategijas, galima daryti šias išvadas:

- Kai daugiakriterinis optimizavimo uždavinys sprendžiamas pirma strategija šešiais kompiuteriais-darbininkais, jungtinio kriterijaus reikšmių vidurkis ν sudaro 51 % jungtinio kriterijaus reikšmių vidurkio, gauto naudojant vieną kompiuterį-darbininką. Tas reiškia, kad šešių kompiuterių-darbininkų naudojimas leidžia pasiekti beveik du kartus geresnius rezultatus jungtinio kriterijaus prasme. Sprendžiant uždavinį pirma strategija, naudoti daugiau nei šešis kompiuterius nėra tikslinga, kadangi sprendimų priėmėjas nespėja formuoti užduotis ir atsiranda kompiuterių-darbininkų prastovos.

- Daugiakriterinį optimizavimo uždavinį sprendžiant antra strategija, naudojant šešis kompiuterius-darbininkus, jungtinio kriterijaus reikšmė, kad jos skirtumas nuo mažiausios būtų 0,01, pasiekiama tik 20-ą minutę; naudojant 12 kompiuterių-darbininkų 17-ą minutę; o naudojant 24 kompiuterius-darbininkus 10-ą minutę. Vadinasi, naudojant 24 kompiuterius-darbininkus yra greičiau pasiekiami geresni rezultatai jungtinio kriterijaus prasme. Daugiau didinti kompiuterių-darbininkų skaičių nėra tikslinga, nes skirtumas tarp rezultatų, gautų, kai kompiuterių-darbininkų skaičius yra 12 ir 24, yra žymiai mažesnis, nei tarp rezultatų, gautų šešiais ir 12 kompiuterių-darbininkų. Atsižvelgiant į rezultatus, gautus iki 10-os sprendimo minutės, jungtinio kriterijaus reikšmių, gautų naudojant 12 kompiuterių-darbininkų, vidurkis ν sudaro 52 % jungtinio kriterijaus reikšmių, gautų naudojant šešis kompiuterius-darbininkus, vidurkio. Jungtinio kriterijaus reikšmių, gautų naudojant 24 kompiuterius-darbininkus, vidurkis ν sudaro 71 % nuo jungtinio kriterijaus reikšmių, gautų naudojant 12 kompiuterių-darbininkų, vidurkio. Be to, naudojant didesnį kompiuterių-darbininkų skaičių, sprendimų priėmėjas nespės vertinti gaunamų sprendinių ir kompiuterių prastovos neišvengiamos.

- Lyginant tarpusavyje abiejų strategijų rezultatus, nustatyta, kad, naudojant šešis kompiuterius-darbininkus, uždavinio sprendimo pradžioje mažesnė jungtinio kriterijaus reikšmė pasiekiama sprendžiant pirma strategija, tačiau pradedant nuo 10-os minutės antra strategija tampa pranašesne, kadangi gaunama mažesnė jungtinio kriterijaus reikšmė. Atsižvelgiant į rezultatus, gautus iki 10-os sprendimo minutės, jungtinio kriterijaus reikšmių, gautų naudojant 12 kompiuterių-darbininkų antra strategija, vidurkis ν sudaro 56 % jungtinio kriterijaus reikšmių, gautų naudojant šešis kompiuterius-darbininkus pirma strategija, vidurkio; jungtinio kriterijaus reikšmių, gautų naudojant 24 kompiuterius-darbininkus antra strategija, vidurkis sudaro 40 % jungtinio kriterijaus reikšmių, gautų naudojant šešis kompiuterius-darbininkus pirma strategija, vidurkio. Taigi antra strategija yra pranašesnė už pirma ypač uždavinio sprendimo pradžioje.

Ištyrus sprendimų priėmėjo apsimokymą spręsti daugiakriterinį uždavinį naudojant sukurtą interaktyvią sprendimų paramos sistemą, galima daryti šias išvadas:

- Kai SP sprendžia uždavinį pirmą kartą, esminio skirtumo tarp strategijų ir didesnio kompiuterių-darbininkų skaičiaus naudojimo nėra, nes SP dar neturi pakankamai įgūdžių ir jis neišnaudoja didesnio kompiuterių-darbininkų skaičiaus privalumo.
- Kai uždavinys sprendžiamas antrą kartą, visais analizuotais atvejais gaunami geresni rezultatai jungtinio kriterijaus prasme nei gauti, sprendžiant pirmą kartą.

- Uždavinį sprendžiant trečią ir vėlesnius kartus, rezultatai, gauti naudojant šešis kompiuterius-darbininkus, mažai skiriasi lyginant su gautais antro sprendimo metu. Galima teigti, kad SP jau antro sprendimo metu apsimokė spręsti uždavinį naudojant tokį kompiuterių-darbininkų skaičių. Be to, sprendžiant antra strategija sprendimų priėmėjas apsimoko greičiau. Kai uždavinys sprendžiamas 12 ir 24 kompiuteriais-darbininkais, iki penkto sprendimo gauti rezultatai labai panašūs, o vėliau skiriasi, ir geresni rezultatai gaunami naudojant 24 kompiuterius-darbininkus bei gerėja iki dešimto (paskutinio) sprendimo. Taigi sprendimų priėmėjai interaktyviai spręsdami daugiakriterinį optimizavimo uždavinį ilgiau apsimoko, kai sprendimui naudojamas arba mažas (1 ir 6) arba didelis (24) kompiuterių–darbininkų skaičius, o greičiausiai apsimoko naudojant 12 kompiuterių-darbininkų, uždavinį sprendžiant antra strategija.
- Lyginant patyrusio ir nepatyrusių sprendimų priėmėjų gautus rezultatus, patyrusio SP gautų jungtinių kriterijų reikšmių procentų vidurkiai ν yra mažesni, negu vidurkiai, kai uždavinį sprendžia nepatyrę sprendimų priėmėjai. Taigi patyręs SP sugeba geriau išnaudoti didesnį kompiuterių-darbininkų skaičių.

Ištyrus pasiūlytą interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdą, apjungiantį SKS ir ASKS metodus, galima daryti šias išvadas:

- Sprendžiant uždavinį SKS metodu, didinant svorinių koeficientų reikšmių kitimo žingsnio s reikšmę gaunamų sprendinių skaičius didėja, tačiau didelė dalis sprendinių sutampa, arba jie yra labai arti vienas kito, todėl Pareto aibė nėra tolygiai padengiama. Kai $s = 100$, Pareto aibės taškų skaičius $n' = 66$, pasiskirstymo įverčio reikšmė $\varphi = 0{,}110$.
- Kai uždavinys yra sprendžiamas ASKS metodu, randami sprendiniai

tose Pareto aibės srityse, kuriose nebuvo rasta SKS metodu. Kai $s = 100$, pasiskirstymo įverčio reikšmė $\varphi = 0{,}053$ ir yra žymiai mažesnė nei gauta SKS metodu, nors rastų Pareto aibės taškų skaičius $n' = 52$ yra mažesnis. Be to, ASKS metodu, kai Pareto aibės taškų skaičius $n' = 45$, gauta pasiskirstymo įverčio reikšmė $\varphi = 0{,}08$ yra mažesnė nei gauta SKS metodu, kai Pareto aibės taškų skaičius $n' = 66$.

- Pradinių reikšmių, sudarančių vektorių X_0, parinkimas turi įtakos gaunamiems sprendiniams.
- Sprendžiant dviejų kriterijų pašarų sudėties optimizavimo uždavinį, daugiausiai sprendinių gaunama vienoje Pareto aibės dalyje, kai $f_1 \in (0{,}05;\ 0{,}55)$ ir $f_2' \in (0{,}004;\ 0{,}23)$. Kriterijaus $f_1(X)$ reikšmės kinta didesniame intervale negu kriterijaus $f_2'(X)$ reikšmės. Taigi nedaug sumažėjus maistinių charakteristikų leistinų normų pažeidimų sumai savikaina (f_1) ženkliai padidėja.
- Analizuojant savikainos (f_1) ir maistinių charakteristikų leistinų normų pažeidimų sumos (f_2') priklausomybę nuo svorinių koeficientų reikšmių, savikainai mažėjant tiesiškai, pažeidimų suma auga eksponentiškai.
- Pasiūlytas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas leidžia rasti tolygiai pasiskirsčiusius sprendinius iš visos Pareto aibės; SP peržiūrėjęs gautus sprendinius gauna informacijos apie siekiamas kriterijų reikšmes, į kurias atsižvelgia interaktyviai spręsdamas daugiakriterinį optimizavimo uždavinį. Tokiu būdu SP turi eilę pradinių sprendinių iš visos Pareto aibės tolimesnei jį tenkinančio sprendinio paieškai.

5

Bendrosios išvados

Atlikti tyrimai atskleidė pasiūlyto interaktyvaus daugiakriterinių optimizavimo uždavinių sprendimo būdo, apjungiančio svertinės kriterijų sumos ir adaptyvaus svertinės kriterijų sumos metodus, ir jį įgyvendinančios sprendimų paramos sistemos galimybes. Eksperimentinių tyrimų rezultatai leido daryti šias išvadas:

1. Pasiūlytas interaktyvus daugiakriterinių optimizavimo uždavinių sprendimo būdas leidžia rasti tolygiai pasiskirsčiusius sprendinius iš visos Pareto aibės. Naudojant sukurtą sprendimų paramos sistemą, sprendimų priėmėjas peržiūrėjęs gautus sprendinius gauna informacijos apie siekiamas kriterijų reikšmes, į kurias atsižvelgia interaktyviai spręsdamas daugiakriterinį optimizavimo uždavinį. Sprendimų priėmėjas turi eilę pradinių sprendinių iš visos Pareto aibės tolimesnei jį tenkinančio sprendinio paieškai.
2. Uždavinį sprendžiant šiame darbe pasiūlyta strategija, atsižvelgiant į rezultatus, gautus spendimo pradžioje (iki 10-os sprendimo minutės),

jungtinio kriterijaus reikšmių, gautų naudojant 12 ir 24 kompiuterius-darbininkus, vidurkis sudaro atitinkamai 52 % ir 71 % jungtinio kriterijaus reikšmių, gautų naudojant šešis kompiuterius-darbininkus, vidurkio. Naudojant didesnį kompiuterių-darbininkų skaičių (>24), sprendimų priėmėjas nespės vertinti gaunamų sprendinių ir kompiuterių prastovos neišvengiamos.

3. Lyginant tarpusavyje pasiūlytos ir pirmos strategijos, sukurtos kitų autorių, rezultatus, nustatyta, kad pasiūlyta strategija yra pranašesnė už pirmą ypač uždavinio sprendimo pradžioje, kadangi atsižvelgiant į rezultatus iki 10-os sprendimo minutės, jungtinio kriterijaus reikšmių, gautų naudojant 12 ir 24 kompiuterius-darbininkus pasiūlyta strategija, vidurkis sudaro atitinkamai 56 % ir 40 % jungtinio kriterijaus reikšmių, gautų naudojant šešis kompiuterius-darbininkus pirma strategija, vidurkio.
4. Tiriant sprendimų priėmėjo apsimokymą spręsti uždavinį naudojant sukurtą interaktyvią sprendimų paramos sistemą, nustatyta, kad sprendimų priėmėjai greičiausiai apsimoko, kai sprendimui naudojama 12 kompiuterių–darbininkų, bet naudojant didesnį jų skaičių (24) sprendimo priėmėjo apsimokymas trunka ilgai, tačiau uždavinio sprendimo pabaigoje pasiekiami geresni rezultatai jungtinio kriterijaus prasme.
5. Lyginant patyrusio ir nepatyrusių sprendimų priėmėjų gautus rezultatus, patyrusio sprendimų priėmėjo gautų jungtinių kriterijų reikšmių procentų vidurkiai yra mažesni, negu vidurkiai, kai uždavinį sprendžia nepatyrę sprendimų priėmėjai. Vadinasi patyręs sprendimų priėmėjas sugeba geriau išnaudoti didesnį kompiuterių-darbininkų skaičių.

Literatūra ir šaltiniai

Abraham, A.; Jain, L. C. 2005. Evolutionary Multiobjective Optimization. In A. Abraham; L. C. Jain; R. Goldberg, *Evolutionary Multiobjective Optimization – Theoretical Advances and Applications* (1–6). London: Springer-Verlag.

Alter, S. L. 1975. *A Study of Computer Aided Decision Making in Organizations. Doctoral Dissertation*. Cambridge: Massachusetts Institute of Technology.

Alter, S. L. 1980. *Decision Support Systems: Current Practice and Continuing Challenge*. Boston: Addison-Wesley.

Alves, M. J.; Climaco, J. 2000. A Note on a Decision Support System for Multiobjective Integer and Mixed-Integer Programming Problems. European Journal of Operational Research, 155, 258–265.

Angehrn, A.; Luthi, H. 1990. Intelligent Decision Support Systems: A Visual Interactive Approach. Interfaces, 20(6), 17–28.

Antunes, C. H.; Alves, M. J.; Silva, A. L.; Clımaco, J. 1992. An integrated MOLP method based package – A guided tour of TOMMIX. Computers & Operations Research, 1(4), 609–625.

Belton, V.; Vickers, S. 1990. Use of a Simple Multi-Attribute Value Function Incorporating Visual Interactive Sensitivity Analysis for Multiple Criteria Decision Making. In C. A. Bana e Costa, *Readings in Multiple Criteria Decision Aid* (319–334). Berlin: Springer-Verlag.

Benayoun, R.; de Montgolfier, J.; Tergny, J.; Larichev, O. I. 1971. Linear Programming with Multiple Objective Functions: Step Method (STEM). Mathematical Programming, 1(1), 366–375.

Byrd, R. H.; Hribar, M. E.; Nocedal, J. 1999. An Interior Point Algorithm for Large-Scale Nonlinear Programming. SIAM Journal on Optimization, 9(4), 877–900.

Blasco, X.; Herrero, J. M.; Sanchis, J.; Martínez, M. 2008. A New Graphical Visualization of n-Dimensional Pareto Front for Decision-Making in Multiobjective Optimization. Information Sciences, 178(20), 3908–3924.

Branke, J.; Deb, K.; Miettinen, K.; Słowiński, R. 2008. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin: Springer.

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.

Bringmann, K.; Friedrich, T. 2010. An Efficient Algorithm for Computing Hypervolume Contributions. Evolutionary Computation, 18(3), 383–402.

Bringmann, K.; Friedrich, T. 2010. Tight Bounds for the Approximation Ratio of the Hypervolume Indicator. *In Proccedings of the 11th International Conference on Parallel Problem Solving From Nature*, Springer-Verlag. 607–616.

Buchanan, J. T. 1997. A Naive Approach for Solving MCDM Problems: The GUESS Method. Journal of the Operational Research Society, 48(2), 202–206.

Bui, L. T.; Alam, S. 2008. *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*. London: Information Science Reference.

Burstein, F.; Holsapple, C. W. 2008. *Handbook on Decision Support Systems*. Berlin: Springer-Verlag.

Caballero, R.; Luque, M.; Molina, J.; Ruiz, F. 2002. An Interactive System for Multiobjective Programming. Information Technologies and Decision Making, 1, 635–656.

Chankong, V.; Haimes, Y. Y. 1983. *Multiobjective Decision Making: Theory and Methodology*. New York: Elsevier Science Publishing Co.

Chankong, V.; Haimes, Y. Y. 1978. The Interactive Surrogate Worth Trade-Off (ISWT) Method for Multiobjective Decision Making. In S. Zionts, *Multiple Criteria Problem Solving* (42–67). Berlin: Springer.

Charnes, A.; Cooper, W. W. 1977. Goal Programming and Multiple Objective Optimization. European Journal of Operational Research, 1(1), 39–54.

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

Charnes, A.; Cooper, W. W.; Ferguson, R. O. 1955. Optimal Estimation of Executive Compensation by Linear Programming. Management Science, 1(2), 138–151.

Cohon, J. L. 1985. Multicriteria programming: Brief review and application. Design Optimization, 163–191.

Coleman, T. F.; Li, Y. 1996. An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds. SIAM Journal on Optimization, 6(2), 418–445.

Coleman, T. F.; Li, Y. 1994. On the Convergence of Reflective Newton Methods for Large-Scale Nonlinear Minimization Subject to Bounds. Mathematical Programming, 67(2), 189–224.

Čiegis, R. 2005. *Lygiagretieji algoritmai ir tinklinės technologijos*. Vilnius: Technika.

Danėnas, P.; Garšva, G. 2011. SVM and XBRL Based Decision Support System for Credit Risk Evaluation. *In Proccedings of the 17th International Conference on Information and Software Technologies*, Kaunas: Technologija. 1–12.

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.

Das, I.; Dennis, J. E. 1998. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. SIAM Journal on Optimization, 8(3), 631–657.

Deb, K.; Kumar, A. 2007. Interactive Evolutionary Multi-Objective Optimization and Decision-Making Using Reference Direction Method. *In Proccedings of the 9th annual conference on Genetic and evolutionary computation (GECCO'07)*. New York: ACM Press.

Deb, K.; Sinha, A.; Korhonen, P. J.; Wallenius, J. 2010. An Interactive Evolutionary Multiobjective Optimization Method Based on Progressively Approximated Value Functions. IEEE Transactions on Evolutionary Computation, 14(5), 723–739.

Dell, R. F.; Karwan, M. H. 1990. An Interactive MCDM Weight Space Reduction Method Utilizing a Tchebycheff Utility Function. Naval Research Logistics, 37, 263–277.

Diwekar, U. 2008. *Introduction to Applied Optimization, Second Edition*. Cambridge: Springer-Verlag.

Dyer, J. S. 1973. A Time-Sharing Computer Program for the Solution of the Multiple Criteria Problem. Management Science, 19(12), 1379–1383.

Dowhań, L.; Wymysłowski, A.; Dudek, R. 2009. Multi-Objective Decision Support System in Numerical Reliability Optimization of Modern Electronic Packaging. Microsystem Technologies, 15(12), 1777–1783.

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

Dzemyda, G.; Šaltenis, V. 1994. Multiple Criteria Decision Support System: Methods, User's Interface and Applications. Informatica, 5(1–2), 31–42.

Dzemydienė, D. 1998. Decision Support System for Transport Management. *In Proccedings of the IFORS SPC8 "Organisational Structures, Management, Simulation of Business Sectors and Systems"*, Kaunas: Technologija. 188–193.

Dzemydienė, D. 2000. Ekspertinių žinių įgijimo metodai sprendimų priėmimo sistemose. *In Proccedings of the Lietuvos mokslas ir pramonė – Informacinės technologijos'2000*, Kaunas: Technologija. 211–217.

Dzemydienė, D. 2006. *Intelektualizuotų informacinių sistemų projektavimas ir taikymas*. Vilnius: Mykolo Romerio universitetas.

Dzemydienė, D. 1996. Sprendimų priėmimo sistemos projektavimo ypatumai operatyviai kintančioje taikomojoje srityje. *In Proccedings of the Informacinės technologijos 96*, Kaunas: Lietuvos mokslas ir pramonė. 41–44.

Dzemydienė, D.; Dzindzalieta, R. 2009. Development of Decision Support System for Risk Evaluation of Transportation of Dangerous Goods Using Mobile Technologies. Technological and Economic Development of Economy, 16(4), 654–671.

Dzemydienė, D.; Kazemikaitienė, E. 2005. Ontology-Based Decision Support System for Crime Investigation Processes. In A. Čaplinskas; O. Vasilecas; W. Woitkowsky; S. Wrycza; J. Zupancic, *Information Systems Development: Advances in Theory, Practice and Education* (245–256). Berlin: Springer.

Dzemydienė, D.; Maskeliūnas, S. 2008. An Approach for Decision Dupport System Design for Evaluation of Water Contamination Processes. *In Proccedings of the Modelling of Business, Industrial and Transport Systems*, Riga: Transport and Telecommunications institute. 197–203.

Dzemydienė, D.; Naujikienė, R. 2006. Intelligent Decision Support for Sustainable Development Processes Evaluation. *In Proccedings of the Citizens and Governance for Sustainable Development (CIGSUD'2006)*, Vilnius: Technika. 275–280.

Eddy, J.; Lewis, K. 2001. Effective Generation of Pareto Sets Using Genetic Programming. *In Proccedings of the ASME International Design Engineering Technical Conference*. Pittsburgh.

Edgeworth, F. 1881. *Mathematical Psychics*. London: P. Keagan.

Eom, S. B.; Lee, S. M.; Kim, J. K. 1993. The Intellectual Structure of Decision Support Systems (1971–1989). Decision Support Systems, 10(1), 19–35.

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.

Ferguson, R. L.; Jones, C. H. 1969. A Computer Aided Decision System. Management Science, 15(10), B550–B561.

Figueira, J.; Greco, S.; Ehrgott, M. 2005. *Multiple Criteria Decision Analysis: State of the Art Surveys* (Vol. 78). Springer.

Fishburn, P. C. 1974. Lexicographic Orders, Utilities and Decision Rules: A Survey. Management Science, 20(11), 1442–1471.

Flavell, R. B. 1976. A New Goal Programming Formulation. Omega, 4(6), 731–732.

Fleischer, M. 2003. The Measure of Pareto Optima Applications to Multi-Objective Metaheuristics. *In Proccedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization (EMO'03)*, Berlin: Springer-Verlag. 519–533.

Floudas, A. C.; Pardalos, P. M. 2009. *Encyclopedia of Optimization* (2). New York: Springer.

Friedrich, T.; Bringmann, K.; Voß, T.; Igel, C. 2011. The Logarithmic Hypervolume Indicator. *In Proccedings of the 11th ACM Foundations of Genetic Algorithms*, ACM. 81–92.

Gass, S.; Saaty, T. 1955. The Computational Algorithm for the Parametric Objective Function. Naval Research Logistics Quarterly, 2(2), 39–45.

Geoffrion, A.; Dyer, J.; Feinberg, A. 1972. An Interactive Approach for Multi-Criterion Optimization, with an Application to the Operation of an Academic Department. Management Science, 19(4), 357–368.

Gill, P. E.; Murray, W.; Wright, M. H. 1981. *Practical Optimization*. London: Academic Press.

Ginevičius, V.; Podvezko, R. 2008. Multicriteria Graphical-Analytical Evaluation of the Financial State of Construction Enterprises. Technological and Economic Development of Economy, 14(4), 452–461.

Grosan, C.; Abraham, A. 2008. Generating Uniformly Distributed Pareto Optimal Points for Constrained and Unconstrained Multicriteria Optimization. Science And Technology, 73–77.

Guliashki, V.; Toshev, H.; Korsemov, C. 2009. Survey of Evolutionary Algorithms Used in Multiobjective Optimization. Problems of Engineering Cybernetics and Robotics, 60, 42–54.

Hackathorn, R. D.; Keen, P. G. 1981. Organizational Strategies for Personal Computing in Decision Support Systems. MIS Quarterly, 5(3), 21–26.

Haimes, Y. Y.; Lasdon, L. S.; Wismer, D. A. 1971. On a Bicriterion Formulation of the Probems of Integrated System Identification and System Optimization. IEEE Transactions on Systems, Man, and Cybernetics, 296–297.

Han, S. P. 1977. A Globally Convergent Method for Nonlinear Programming. Journal of Optimization Theory and Applications, 22(3), 297–309.

Holt, C. C.; Huber, G. P. 1969. A Computer Aided Approach to Employment Service Placement. Management Science, 15(11), 573–594.

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.

Hurwicz, L. 1958. Programming in Linear Spaces. In K. J. Arrow; L. Hurwicz; H. Uzawa, *Studies in Linear and Non-Linear Programming* (38–102). Stanford: Stanford Univiversity Press.

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

Ignizio, J. P. 1983. A Note on Computational Methods in Lexicographic Linear Goal Programming. The Journal of the Operational Research Society, 32(6), 539–542.

Isermann, H. 1974. *Lineare Vektoroptimierung (in German)*. Regensburg: Hain.

Yu, P. L. 1973. A Class of Solutions for Group Decision Problems. Management Science, 936–946.

Yu, P. L. 1974. Cone Convexity, Cone Cxtreme Points and Nondominated Solutions in Decision Problems with Multiobjectives. Journal of Optimization Theory and Applications, 14, 319–377.

Jao, S. 2010. *Decision Support Systems*. Rijeka: InTech.

Jaszkiewicz, A.; Slowinski, R. 1994. The Light Beam Search Over a Non-Dominated Surface of a Multiple-Objective Programming Problem. *In Proccedings of the Multiple Criteria Decision Making*, Berlin: Springer. 87–99.

Kaklauskas, A.; Zavadskas, E. K. 2010. *Intelektinė ir biometrinė sprendimų parama*. Vilnius: Technika.

Kaklauskas, A.; Zavadskas, E. K. 2002. *Internetinė sprendimų parama*. Vilnius: Technika.

Kaklauskas, A.; Zavadskas, E. K.; Raslanas, S. 2005. Decision Support Systems in Lithuania. Transport and Telecommunication, 6(1), 11–17.

Kaklauskas, A.; Zavadskas, E. K.; Seniut, M.; Krutinis, M.; Dzemyda, G.; Ivanikovas, S.; et al. 2008. Web-BasedBiometric Mouse Decision Support System for User's Emotional and Labour Productivity Analysis. *In Proccedings of the 25th International Symposium on Automation and Robotics in Construction*, Vilnius: Technika. 69–75.

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

Kaliszewski, I. 2006. *Soft Computing for Complex Multiple Criteria Decision Making*. Springer.

Keen, P. G.; Scott Morton, M. S. 1978. *Decision Support Systems: An Organizational Perspective*. Boston: Addison-Wesley.

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

Kim, E. M.; de Weck, O. 2006. Adaptive Weighted Sum Method for Multiobjective Optimization: a New Method for Pareto Front Generation. Structural and Multidisciplinary Optimization, 31(2), 105–116.

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.

Korhonen, P. 1988. A Visual Reference Direction Approach to Solving Discrete Multiple Criteria Problems. European Journal of Operational Research, 34(2), 152–159.

Korhonen, P. 1987. VIG – A Visual Interactive Support System for Multiple Criteria Decision Making. Belgian Journal of Operations Research, Statistics and Computer Science, 27(1), 3–15.

Korhonen, P.; Laakso, J. 1986. A Visual Interactive Method for Solving Multiple Criteria Problems. European Journal of Operational Research, 24, 277–287.

Kuhn, H. W.; Tucker, A. W. 1951. Nonlinear Programming. *In Proccedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley: University of California Press. 481–492.

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

Lapin, K. 2010. SKY-Scanner: Time-Critical Decision Support System Surveilling Aircraft Landing and Take-Off. *In Proccedings of the 9th Innovative Research Workshop and Exhibition*, France: EUROCONTROL. 19–26.

Lapin, K.; Čyras, V.; Savičienė, L. 2011. Visualization of Aircraft Approach and Departure Procedures in a Decision Support System for Controllers. *In Proccedings of the Frontiers in Artificial Intelligence and Applications*. *224*, 408–421. Amsterdam: IOS Press.

Lee, S. 1972. *Goal Programming for Decision Analysis*. Philadelphia: Auerbach Publications.

Lewandowski, A.; Kreglewski, T.; Rogowski, T.; Wierzbicki, A. 1989. Decision Support Systems of DIDAS family. In A. Lewandowski, *Aspiration Based Decision Support Systems: Theory, Software and Applications* (21–27). Berlin: Springer.

Licklider, J. C. 1960. Man-Computer Symbiosis. IRE Transactions on Human Factors in Electronics, HFE-1(1), 4–11.

Lieberman, E. R. 1991. Soviet Multi-Objective Mathematical Programming methods: An Overview. Managment Science, 37(9), 1147–1165.

Little, J. D. 1970. Models and Managers: The Concept of a Decision Calculus. Management Science, 16(8), B466–B485.

Liu, S.; H. B. Duffy, A. H.; Whitfield, R. I.; Boyle, I. M. 2010. Integration of Decision Support Systems to Improve Decision Support Performance. Knowledge and Information Systems, 22(3), 261–286.

Luque, M.; Jian-Bo Yang Wong, B. 2009. PROJECT Method for Multiobjective Optimization Based on Gradient Projection and Reference Points. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 39(4), 864–879.

Luque, M.; Ruiz, F.; Miettinen, K. 2011. Global Formulation for Interactive Multiobjective Optimization. OR Spectrum, 33(1), 27–48.

Mackutė-Varoneckienė, A. 2006. *Optimizavimo metodai separavimo procesų uždaviniams, kai tikslo funkcijoms ir leistinosioms sritims būdingi nereguliarumai. Daktaro disertacija*. Kaunas: Vytauto Didžiojo universiteto leidykla.

Marler, R. T.; Arora, J. S. 2004. Survey of Multi-Objective Optimization Methods for Engineering. Structural and Multidisciplinary Optimization, 26(6), 369–395.

Messac, A. 1996. Physical Programming: Effective Optimization for Computational Design. AIAA Journal, 34(1), 149–158.

Messac, A.; Mattson, C. A. 2002. Generating Well-Distributed Sets of Pareto Points for Engineering Design Using Physical Programming. Optimization and Engineering, 3(4), 431–450.

Messac, A.; Mattson, C. A. 2004. Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier. AIAA Journal, 42(10), 2101–2111.

Messac, A.; Ismail-Yahaya, A.; Mattson, C. A. 2003. The Normalized Normal Constraint Method for Generating the Pareto Frontier. Structural and Multidisciplinary Optimization, 25(2), 86–98.

Miettinen, K. 1999. *Nonlinear Multi-Objective Optimization*. Boston: Kluwer Academic Publishers.

Miettinen, K.; Mäkelä, M. M. 1993. An Interactive Method for Nonsmooth Multiobjective Optimization with an Application to Optimal Control. Optimization Methods and Software 2, 31–44.

Miettinen, K.; Mäkelä, M. M. 1995. Interactive Bundle-Based Method for Nondifferentiable Multiobjective Optimization: NIMBUS. Optimization, 34(3), 231–246.

Miettinen, K.; Mäkelä, M. M. 2000. Interactive Multiobjective Optimization System WWW-NIMBUS on the Internet. Computers & Operations Research, 27, 709–723.

Miettinen, K.; Makela, M. M. 2006. Synchronous Approach in Interactive Multiobjective Optimization. European Journal Of Operational Research, 170(3), 909–922.

Miettinen, K.; Eskelinen, P.; Ruiz, F.; Luque, M. 2010. NAUTILUS method: An Interactive Technique in Multiobjective Optimization Based on the Nadir Point. European Journal of Operational Research, 206(2), 426–434.

Miettinen, K.; Mäkelä, M. M.; Mäkinen, R. A. 1996. Interactive Multiobjective Optimization System NIMBUS Applied to Nonsmooth Structural Design Problems. *In Proccedings of the System Modelling and Optimization*, London: Chapman & Hall. 379–385.

Mir, S. A.; Quadri, S. M. 2009. Decision Support Systems: Concepts, Progress and Issues – A Review. Sustainable Agriculture Reviews, 2, 373–399.

Mirrazavi, S. K.; Jones, D. F.; Tamiz, M. 2003. MultiGen: An Integrated Multiple Objective Solution System. Decision Support Systems, 36, 177–187.

Mockus, J. B.; Mockus, L. J. 1991. Bayesian Approach to Global Optimization and Application to Multiobjective and Constrained Problems. Journal of Optimization Theoty and Applications, 70(1), 157–172.

Mockus, J. 2011. On the Pareto Optimality in the Context of Lipschitzian Optimization. Informatica, 22(4), 521–536.

Nakayama, H. 1995. Aspiration Level Approach to Interactive Multi-Objective Programming and Its Applications. In P. M. Pardalos; Y. Siskos; C. Zopounidis, *Advances in Multicriteria Analysis* (147–174). Dordrecht: Kluwer Academic Publishers.

Nakayama, H.; Sawaragi, Y. 1984. Satisficing Trade-Off Method for Multiobjective Programming. In M. Grauer; A. P. Wierzbicki, *Interactive Decision Analysis* (113–122). Heidelberg: Springer.

Narula, S. C.; Kirilov, L.; Vassilev, V. 1994. Reference Direction Approach for Solving Multiple Objective Nonlinear Programming Problems. IEEE Transactions on Man, Systems, and Cybernetics, 24(5), 804–806.

Niu, L.; Lu, J.; Zhang, G. 2009. Decision Making and Decision Support Systems. Studies in Computational Intelligence, 238, 3–18.

Nocedal, J.; Wright, S. J. 2000. *Numerical Optimization*. Berlin: Springer.

Ogryczak, W.; Stuchinski, K.; Zorychta, Z. 1992. DINAS: A Computer-Assisted Analysis System for Multiobjective Transshipment Problems with Facility Location. Computers and Operations Research, 19, 637–648.

Ojalehto, V.; Miettinen, K.; Makela, M. M. 2007. Interactive Software for Multiobjective Optimization: IND-NIMBUS. WSEAS Transactions on Computers, 6(1), 87–94.

Osyczka, A. 1988. Computer Aided Multicriterion Optimization System (CAMOS). In H. A. Eschenauer; G. Thierauf, *Discretization Methods and Structural Optimization* (263–270). Berlin: Spinger-Verlag.

Osyczka, A. 1992. *Computer Aided Multicriterion Optimization System (CAMOS): Software Package in FORTRAN*. Cracow: International Software Publishers.

Pareto, V. 1906. *Manuale di Economia Politica*. Milano: Societa Editrice Libraria.

Petkus, T. 2001. *Kompiuterių tinklo panaudojimas interaktyviame optimizavime. Daktaro disertacija*. Vilnius: Vilniaus pedagoginis universitetas.

Petkus, T. 2001. Pradinių duomenų parinkimas interaktyviam daugiakriteriniam optimizavimui. Informacijos mokslai, 34, 321–325.

Pomerol, J.; Barba-Romero, S. 1993. *Choix multicritère dans l 'entreprise; principe et pratique (in French)*. Paris: Hermes.

Powell, M. J. 1978. A Fast Algorithm for Nonlinearly Constrained Optimization Calculations. In G. A. Watson, *Lecture Notes in Mathematics* (Vol. 630, 144–157). Berlin: Springer.

Power, D. J. 2007. *A Brief History of Decision Support Systems*. Retrieved December 2, 2011, from DSSResources.COM: http://DSSResources.COM/history/dsshistory.html

Power, D. J. 2004. Decision Support Systems: From the Past to the Future. *In Proccedings of the 2004 Americas Conference on Information Systems*, New York. 2025–2031.

Power, D. J. 1998. Web-based Decision Support Systems. DSstar, The On-Line Executive Journal for Data-Intensive Decision Support, 2(33).

Power, D. J.; Sharda, R. 2009. Decision Support Systems. In S. Y. Nof, *Springer Handbook of Automation* (1539–1548). Berlin: Springer-Verlag.

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

Radziukyniene, I.; Zilinskas, A. 2008. Comparison of Several Methods for Pareto Set Generation in Multi-Criteria Portfolio Optimization. International Journal of Computing, 7(3), 22-29.

Raymond, R. C. 1966, April. Use of the Time-Sharing Computer in Business Planning and Budgeting. Management Science, 12(8), B363–B381.

Rangaiah, G. P. 2008. *Multi-objective Optimization: Techniques and Applications in Chemical Engineering*. World Scientific.

Ravindranath, B. 2009. *Decision Support Systems and Data Warehouses*. New Age Books.

Ritzel, B. J.; Wayland Eheart, J.; Ranjithan, S. 1994. Using Genetic Algorithms to Solve a Multiple Objective Groundwater Pollution Containment Problem. Water Resources Research, 30(5), 1589–1603.

Sakawa, M. 1982. Interactive Multiobjective Optimization by the Sequential Proxy Optimization Technique (SPOT). IEEE Transactions on Reliability, R-31(5), 461–464.

Sawaragi, Y.; Nakayama, H.; Tanino, T. 1985. *Theory of Multiobjective Optimization*. Salt Lake City: Academic Press.

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

Schott, J. R. 1995. *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*. Cambridge, Massachusetts: Massachusetts Institute of Technology.

Scott Morton, M. S. 1967. *Computer-Driven Visual Display Devices – Their Impact on the Management. Doctoral Dissertation*. Boston: Harvard Business School.

Shan, S.; Wang, G. G. 2005. An Efficient Pareto Set Identification Approach for Multiobjective Optimization on Black-Box Functions. Journal of Mechanical Design, 127(5).

Shukla, P. K. 2007. On the Normal Boundary Intersection Method for Generation of Efficient Front. Lecture Notes in Computer Science, 4487, 310-317.

Sindhya, K.; Ruiz, A. B.; Miettinen, K. 2011. A Preference Based Interactive Evolutionary Algorithm for Multi-objective Optimization: PIE. Lecture Notes in Computer Science, 6576, 212–225.

Skūpienė, J. 2010. *Algoritmų ir juos realizuojančių programų vertinimas informatikos olimpiadose. Daktaro disertacija*. Vilnius: Vilniaus universitetas.

Sprague, R. H.; Carlson, E. D. 1982. *Building Effective Decision Support Systems*. New Jersey: Englewood Cliffs, Prentice-Hall.

Sprague, R. H.; Watson, H. 1979. Bit by Bit: Toward Decision Support Systems. California Management Review, 22(1), 60–68.

Stasytytė, V. 2011. *Investicijų portfelio sprendimų paramos sistema. Daktaro disertacija*. Vilnius: Technika.

Steponavičė, I. 2010. *Tolygaus sprendinių išdėstymo Pareto aibėje algoritmai ir jų taikymai rizikos valdyme. Daktaro disertacija*. Kaunas: Vytauto Didžiojo universitetas.

Steuer, R. E. 1985. *Multiple Criteria Optimization: Theory, Computation and Application*. New York: John Wiley & Sons.

Steuer, R. E.; Choo, E. U. 1983. An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming. Mathematical Programming, 26, 326–344.

Straffin, P. D. 1996. *Game Theory and Strategy*. Washington: The Mathematical Association of America.

Šarka, V. 2008. *Sprendimų paramos sistema statyboje taikant daugiakriterinius sintezės metodus*. Vilnius: Technika.

Tabucanon, M. 1988. *Multiple Criteria Decision Making in Industry*. Amsterdam: Elsevier Science.

Thiele, L.; Miettinen, K.; Korhonen, P.; Molina, J. 2009. A Preference-Based Evolutionary Algorithm for Multi-Objective Optimization. Evolutionary Computation, 17(3), 411–436.

Trinkūnienė, E. 2006. *Statybos rangos sutarčių vertinimo sprendimų paramos sistema internete. Daktaro disertacija*. Vilnius: Technika.

Turban, E. 1967. The Use of Mathematical Models in Plant Maintenance Decision Making. Management Science, 13(6), B342–B358.

Turskis, Z. 2008. Multi-Attribute Contractors Ranking Method by Applying Ordering of Feasible Alternatives of Solutions in Terms of Preferability Technique. Technological and Economic Development of Economy, 14(2), 224–239.

Urban, G. L. 1967. SPRINTER: A Tool for New Products Decision Makers. Industrial Management, 8(2), 43–54.

Utyuzhnikov, S. V.; Fantini, P.; Guenov, M. 2009. A Method for Generating a Well-Distributed Pareto Set in Nonlinear Multiobjective Optimization. Journal of Computational and Applied Mathematics, 223(2), 820–841.

Van Veldhuizen, D. A. 1999. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Ohio: Air Force Institute of Technology, Wright-Patterson AFB.

Van Veldhuizen, D. A.; Lamont, G. B. 1998. *Multiobjective Evolutionary Algorithm Research: A History and Analysis*. Air Force Institute of Technology, Department of Electricaland Computer Engineering. Ohio: Wright-Patterson AFB.

Vassilev, V.; Atanassov, A.; Sgurev, V.; Kichovitch, M.; Deianov, A.; Kirilov, L. 1993. Software Tools for Multiciteria Programing. *In Proccedings of the User-Oriented Methodology and Techniques of Decision Analysis and Support*, Berlin: Springer. 288–294.

Vassilev, V.; Genova, K.; Vassileva, M. 2006. A Multicriteria Decision Support System Multidecision-1. Information Theories & Applications, 13, 203–209.

Vassilev, V.; Genova, K.; Vassileva, M.; Narula, S. 2003. Classification-Based Method of Linear Multicriteria Optimization. International Journal on Information Theories and Applications, 10(3), 266–270.

Vassilev, V.; Narula, S.; Vladimirov, P.; Djambov, V. 1997. MOIP: A DSS for Multiple Objective Intege Programming Problems. In J. Climaco, *Multicriteria analysis* (259–268). Berlin: Springer.

Vassilev, V.; Vassileva, M.; Staykov, B.; Genova, K.; Andonov, F.; Chongova, P. 2008. MultiDecision-2: A Multicriteria Decision Support System. Information Technologies and Knowledge, 2(3), 203–211.

Wallenius, H. 1991. *Implementing Interactive Multiple Criteria Decision Methods in Public Policy*. Jyväskylä: University of Jyväskylä.

Wallenius, J.; Zionts, S. 1976. Some Tests of an Interactive Programming Method for Multicriteria Optimization and an Attempt at Implementation. In H. Thiriez; S. Zionts, *Multiple Criteria Decision Making* (319–331). Berlin: Springer.

Waltz, R. A.; Morales, J. L.; Nocedal, J.; Orban, D. 2006. An Interior Algorithm for Nonlinear Optimization that Combines Line Search and Trust Region Steps. Mathematical Programming, 107(3), 391–408.

Weistroffer, H. R.; Smith, C. H.; Narula, S. 2005. Multiple Criteria Decision Support Software. In J. Figueira; S. Greco; E. M., *Multiple Criteria Decision Analysis: State of the Art Surveys* (989–1009). Springer.

Wierzbicki, A. P. 1982. A Mathematical Basis for Satisficing Decision Making. Mathematical Modelling, 3, 391–405.

Wierzbicki, A. P. 1999. Reference Point Approaches. In T. Gal; T. J. Stewart; T. Hanne, *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications* (9–39). Dordrecht: Kluwer.

Wierzbicki, A. P. 1980. The Use Of Reference Objectives in Multiobjective Optimization. In G. Fandel; T. Gal, *Multiple Criteria Decision Making, Theory and Applications* (468–486). Berlin: Springer.

Zadeh, L. 1963. Optimality and Non-Scalar-Valued Performance Criteria. IEEE Transactions on Automatic Control, 8(1), 59–60.

Zavadskas, E. K.; Turskis, Z. 1997. Multicriteria Decision Macing System for Building Returbishment. Statyba, 62–67.

Zavadskas, E. K.; Simanauskas, L.; Kaklauskas, A. 1999. *Sprendimų paramos sistemos statyboje*. Vilnius: Technika.

Zavadskas, E. K.; Zakarevicius, A.; Antucheviciene, J. 2006. Evaluation of Ranking Accuracy in Multi-Criteria Decisions. Informatica, 17(4), 601–618.

Zeleny, M. 1973. Compromise Programming. In J. L. Cochrane; M. Zeleny (Ed.), *Multiple Criteria Decision Making* Columbia: University of South Carolina Press. 262–301.

Zeleny, M. 1974. *Linear Multiobjective Programming*. Berlin: Springer-Verlag.

Zhou, A.; Jin, Y.; Zhang, Q.; Sendhoff, B.; Tsang, E. 2006. Combining Model-Based and Genetics-Based Offspring Generation for Multi-objective Optimization Using a Convergence Criterion. *In Proccedings of the 2006 IEEE Congress on Evolutionary Computation*, Vancouver. 892–899.

Zionts, S. 1989. Multiple Criteria Mathematical Programming: An Updated Overview and Several Approaches. In B. Karpak; S. Zionts (Ed.), *Multiple criteria decision making and risk analysis using microcomputers* (Vol. 56, 7–60). Berlin, New York: Springer-Verlag.

Zionts, S.; Wallenius, J. 1976. An Interactive Programming Method for Solving the Multiple Criteria Problem. Management Science, 22(6), 652–663.

Zitzler, E.; Thiele, L. 1998. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In A. E. Eiben (Ed.), *Conference on Parallel Problem Solving from Nature V* Amsterdam: Springer-Verlag. 292–301.

Zujevs, A.; Eiduks, J. 2011. New Decision Maker Model for Multiobjective Optimization Interactive Methodc. *In Proccedings of the INFORMATION TECHNOLOGIES' 2011*, Kaunas: Technologija. 51–58.

Žilinskas, A. 2000. *Matematinis programavimas*. Kaunas: Vytauto Didžiojo universiteto leidykla.

Autoriaus publikacijų sąrašas disertacijos tema

Straipsniai recenzuojamuose periodiniuose mokslo žurnaluose

A 1. Petkus T.; Filatovas E. 2008. Decision Making to Solve Multiple Criteria Optimization Problems in Computer Networks. *Information Technology and Control*, Vol. 37(1), 63–68. ISSN 1392-124X. (ISI Web of Science, VINITI, INSPEC). (Impact Factor 2010: 0,638).

A 2. Petkus T.; Filatovas E.; Kurasova O. 2009. Investigation of Human Factors while Solving Multiple Criteria Optimization Problems in Computer Network. *Technological and Economic Development of Economy*, Vol. 15(3), 464–479. ISSN 1392-8619. (ISI Web of Science).

A 3. Ivanikovas S.; Filatovas E.; Žilinskas J. 2009. Experimental Investigation of Local Searches for Optimization of Grillage-Type Foundations. In: R. Čiegis, D. Henty, B.Kågström, J. Žilinskas

(Eds.), *Parallel Scientific Computing and Optimization*, *Springer Optimization and Its Applications*, Vol. 27, 103–112. ISSN 1931-6828. (ISI Web of Science (Conference Proceedings Citation Index), SpringerLink, INSPEC, Zentralblatt MATH).

A 4. Filatovas E.; Kurasova O. 2011. A Decision Support System for Solving Multiple Criteria Optimization Problems. *Informatics in Education*, Vol. 12(2), 1–12. ISSN 1648-5831. (VINITI, INSPEC).

Straipsniai kituose mokslo leidiniuose

B 1. Filatovas E.; Kurasova O. 2009. Decision Support System for the Optimal Selection of Feed Ingredients. *Proceedings of Euro Mini Conference “Knowledge-Based Technologies and OR methodologies for Strategic Decisions of Sustainable Development”* (KORSD-2009), 58–63, Vilnius: Technika. ISBN 978-9955-28-482-6.

Ernestas Filatovas

DAUGIAKRITERINIŲ OPTIMIZAVIMO UŽDAVINIŲ SPRENDIMAS INTERAKTYVIUOJU BŪDU

Daktaro disertacija

Technologijos mokslai,
Informatikos inžinerija (07T)

Ernestas Filatovas

SOLVING MULTIPLE CRITERIA OPTIMIZATION PROBLEMS IN AN INTERACTIVE WAY

Doctoral Dissertation

Technological Sciences,
Informatics Engineering (07T)