



VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Roberta Poderytė

**TIKSLO SIEKIANČIŲ VERSLO PROCESŲ MODELIAVIMO IR  
SIMULIACIJOS TYRIMAS**

**MODELING AND SIMULATION ANALYSIS OF GOAL-ORIENTED  
BUSINESS PROCESSES**

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos studijų programa, valstybinis kodas 6211BX017

Informacinių sistemų specializacija

Informatikos inžinerijos studijų kryptis

Vilnius, 2019

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU  
Katedros vedėjas

\_\_\_\_\_  
(Parašas)

\_\_\_\_\_  
(Vardas, pavardė)

\_\_\_\_\_  
(Data)

Roberta Poderytė

**TIKSLO SIEKIANČIŲ VERSLO PROCESŲ MODELIAVIMO IR  
SIMULIACIJOS TYRIMAS**

**MODELING AND SIMULATION ANALYSIS OF GOAL-ORIENTED  
BUSINESS PROCESSES**

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos studijų programa, valstybinis kodas 6211BX017

Informacinių sistemų specializacija

Informatikos inžinerijos studijų kryptis

**Vadovas**

prof. dr. Olegas Vasilecas  
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

\_\_\_\_\_  
(Parašas)

\_\_\_\_\_  
(Data)

**Konsultantas**

Tadas Vysockis  
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

\_\_\_\_\_  
(Parašas)

\_\_\_\_\_  
(Data)

**Lietuvių kalbos konsultantas**

dr. Aušra Žemienė  
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

\_\_\_\_\_  
(Parašas)

\_\_\_\_\_  
(Data)

Vilnius, 2019

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS  
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS  
INFORMACINIŲ SISTEMŲ KATEDRA

Informatikos inžinerijos studijų kryptis  
Informcinių sistemų programų inžinerijos studijų programa, valstybinis kodas  
6211BX017  
Informcinių sistemų specializacija

TVIRTINU  
Katedros vedėjas

\_\_\_\_\_  
(Parašas)

\_\_\_\_\_  
(Vardas, pavardė)

\_\_\_\_\_  
(Data)

**BAIGIAMOJO MAGISTRO DARBO  
UŽDUOTIS**

.....Nr. ....  
Vilnius

Studentui (ei) Robertai Poderytei

Baigiamojo darbo tema: Tikslų siekiančių verslo procesų modeliavimo ir simuliacijos tyrimas

(Modeling and Simulation Analysis of Goal-oriented Business Processes)

Baigiamojo darbo (projekto) užbaigimo terminas 2017 m. birželio 10 d.

**BAIGIAMOJO DARBO (PROJEKTO) UŽDUOTIS:**

1. Atlikti literatūros, susijusios su tikslų siekiančių verslo procesų modeliavimu ir simuliacija, analizę.
2. Išnagrinėti verslo procesų modeliavimo ir simuliacijos metodus bei įrankius.
3. Pasiūlyti tikslų siekiančių verslo procesų simuliacijos metodą.
4. Atlikti pasiūlyto metodo eksperimentinį tyrimą, pateikti eksperimento rezultatus, suformuluoti galutines išvadas ir pateikti tolimesnes mokslinių tyrimų kryptis.

Baigiamojo bakalauro darbo (projekto) konsultantai:

Tadas Vysockis  
(Moksl. laipsnis, vardas, pavardė)

dr. Aušra Žemienė  
(Moksl. laipsnis, vardas, pavardė)

Vadovas .....  
(Parašas)

prof. dr. Olegas Vasilecas  
(Moksl. laipsnis, vardas, pavardė)

Užduotį gavau

.....  
(Parašas)

Robertai Poderytei

.....  
(Vardas, pavardė)

.....  
(Data)

Vilniaus Gedimino technikos universitetas Fundamentinių mokslų fakultetas Informacinių sistemų katedra		ISBN ..... ISSN ..... Egz. sk. .... Data .....-.....-.....
Antrosios pakopos studijų <b>Informacinių sistemų programų inžinerijos</b> programos magistro baigiamasis darbas Pavadinimas <b>Tikslo siekiančių verslo procesų modeliavimo ir simuliacijos tyrimas</b> Autorius <b>Roberta Poderytė</b> Vadovas <b>Olegas Vasilecas</b>		
<p style="text-align: right;"><b>Kalba:</b> lietuvių</p>		
<p><b>Anotacija</b></p> <p>Šiame darbe nagrinėjamas tikslo siekiančių verslo procesų modeliavimas ir simuliacija. Simuliacija – vienas iš būdų, leidžiančių atlikti įvairių verslo procesų vykdymo scenarijų analizę, priklausančią nuo įvairių kintamųjų. Daugybė veiksmų gali lemti verslo procesų simuliacijos rezultatus, o jų prognozės yra svarbios verslo tikslų įgyvendinimui.</p> <p>Darbe analizuojami šaltiniai, tiriantys tikslo siekiančių verslo procesų modeliavimo bei simuliacijos galimybes, o pasiūlytas ir realizuotas metodas leidžia analizuoti galimas proceso baigtis bei parinkti parametrus, didinant naudą verslui.</p> <p>Remiantis darbe atlikta analize, sukurtam simuliaciniam modeliui buvo naudojamas genetinio algoritmo veikimas. Atliktas eksperimentinis tyrimas, parodantis tinkamumo funkcijos svarbą simuliacinio modelio kūrimo metu bei įrodantis genetinio algoritmo gebėjimą siūlyti parametrus proceso tobulinimui.</p> <p>Darbą sudaro: įvadas, tikslo siekiančių verslo procesų, modeliavimo ir simuliacijos analizė, tikslo siekiantiems verslo procesams taikomo simuliacijos metodo analizė, realizacijos aprašas, eksperimentinis tyrimas, išvados, tolimesnių tyrimų kryptys, literatūros sąrašas.</p> <p>Darbo apimtis – 65 p. teksto be priedų, 29 iliustr., 6 lent., 53 bibliografiniai šaltiniai.</p>		
<p><b>Prasminiai žodžiai:</b> verslo procesai, tikslo siekiantys verslo procesai, modeliavimas, simuliacija, tikslo siekiančių verslo procesų simuliacija, genetinis algoritmas, tikslo funkcija.</p>		

ISBN	ISSN
Copies No. ....	
Date .....-.....-.....	

Title	<b>Modelling and Simulation Analysis of Goal-Oriented Business Process</b>
Author	<b>Roberta Poderytė</b>
Academic supervisor	<b>Olegas Vasilecas</b>

### Annotation

Thesis consist of: 65 p. text without appendixes, 29 pictures, 6 tables, 53 bibliographical entries.

**Keywords:** business processes, goal-oriented business processes, modeling, simulation, genetic algorithm, fitness function.

**(Baigiamojo darbo sąžiningumo deklaracijos forma)**

**VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS**

Roberta Poderytė, 20133700

(Studento vardas ir pavardė, studento pažymėjimo Nr.)

Fundamentinių mokslų fakultetas

(Fakultetas)

Informacinių sistemų programų inžinerija, ISIfm-17

(Studijų programa, akademinė grupė)

**BAIGIAMOJO DARBO (PROJEKTO)**

**SĄŽININGUMO DEKLARACIJA**

2019 m. birželio 03 d.

Patvirtinu, kad mano baigiamasis darbas tema „Tikslo siekiančių verslo procesų modeliavimo ir simuliacijos tyrimas“ patvirtintas 2018 m. spalio 26 d. dekanų potvarkiu Nr. 413fm, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Prenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą, mane konsultavo mokslininkai ir specialistai: Tadas Vysockis. Mano darbo vadovas profesorius daktaras Olegas Vasilecas.

Kitų asmenų indėlio į parengtą baigiamąjį darbą nėra. Jokių įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs (-usi).

(Parašas)

Roberta Poderytė  
(Vardas ir pavardė)

# Turinys

SANTRUMPOS .....	11
ĮVADAS .....	12
Tyrimo objektas .....	12
Darbo tikslas ir uždaviniai .....	12
Temos naujumas .....	13
Temos aktualumas .....	13
Tyrimo metodika .....	13
Darbo rezultatai .....	13
Darbo struktūra .....	13
1. TIKSLO SIEKIANČIŲ VERSLO PROCESŲ, MODELIAVIMO IR SIMULIACIJOS ANALIZĖ .....	15
1.1. Verslo procesai .....	15
1.2. Verslo procesų modeliavimas .....	16
1.3. Statiniai ir dinaminiai verslo procesai .....	22
1.4. Dinaminių verslo procesų simuliacija .....	25
1.5. Genetiniai algoritmai .....	28
1.5.1. Natūralios atrankos principai .....	28
1.5.2. Genetinių algoritmų fazės .....	29
1.5.3. Daugiatiksliai genetiniai algoritmai .....	32
1.5.3.1. NSGA-II algoritmas .....	33
1.5.3.2. SPEA ir SPEA 2 algoritmai .....	34
1.5.3.3. PESA algoritmas .....	35
1.5.3.4. NSGA-II, SPEA, SPEA2 ir PESA algoritmų lyginimas .....	36
1.6. Apibendrinimas ir rezultatai .....	38
2. TIKSLO SIEKIANTIEMS VERSLO PROCESAMS TAIKOMO SIMULIACIJOS METODO ANALIZĖ .....	39
2.1. Verslo apibrėžtų tikslų siekimas .....	39
2.2. Tikslų siekiančių verslo procesų simuliacijos modelis .....	39
2.2.1. Genetinio algoritmo vykdymas .....	40
2.3. Metodo apibendrinimas .....	43
3. TIKSLO SIEKIANTIEMS VERSLO PROCESAMS TAIKOMO SIMULIACIJOS METODO REALIZACIJOS APRAŠAS .....	44
3.1. Priemonės simuliacijai vykdyti .....	44
3.2. Simuliacijos elementai .....	44
4. EKSPERIMENTINIS TYRIMAS .....	47
4.1. Verslo proceso analizė .....	47
4.2. Verslo proceso simuliacija .....	50

4.3. Genetinio algoritmo pritaikymas.....	51
IŠVADOS .....	59
TOLIMESNIŲ TYRIMŲ KRYPTYS .....	60
LITERATŪROS SĄRAŠAS.....	61



## Iliustracijos

1 pav. Ryšys tarp SysML ir UML (sudaryta autoriaus, remiantis Pearce 2013).....	17
2 pav. SysML diagramų tipai (sudaryta autoriaus, remiantis Friedenthal 2009) .....	18
3 pav. BPMN diagramos jungčių tipai .....	20
4 pav. Proceso raida nuo statinio iki dinaminio (sudaryta autoriaus, remiantis Rusinaitė et al., 2015) .....	23
5 pav. Dinaminio verslo proceso simuliacijos eiga (sudaryta autoriaus, remiantis Vasilecas 2018) .....	26
6 pav. Veiklos parinkimo subproceso eiga (sudaryta autoriaus, remiantis Vasilecas 2018).....	27
7 pav. Genas, chromosoma ir populiacija .....	29
8 pav. Kryžminis taškas .....	30
9 pav. Palikuonių kūrimas, keičiant genus iki kryžminio taško .....	31
10 pav. Naujų palikuonių prijungimas prie populiacijos .....	31
11 pav. Genų mutacija.....	31
12 pav. „Kuprinės“ uždavinys.....	36
13 pav. Algoritmų rezultatai, kai $m = 2$ (sudaryta autoriaus, remiantis Zitzler et al., 2001).....	37
14 pav. Algoritmų rezultatai, kai $m = 3$ (sudaryta autoriaus, remiantis Zitzler et al., 2001).....	38
15 pav. Taikomo metodo modelis .....	39
16 pav. Submodelis „Vykdėti genetinį algoritmą“ .....	41
17 pav. Resursų paieška naudojant genetinį algoritmą (sudaryta autoriaus, remiantis Si et al. 2018).....	42
18 pav. Metodo realizacijai naudojamos klasės .....	46
19 pav. IT pagalbos centro skambučių priėmimo procesas.....	49
20 pav. Tinkamumo funkcijos koeficiento kitimas (2 parametrai) .....	52
21 pav. IT skambučių centro resursai (2 parametrai).....	53
22 pav. Vidutinis laukimo laikas (2 parametrai) .....	53
23 pav. Klientų atsiliepimo balai (2 parametrai).....	54
24 pav. Tinkamumo funkcijos koeficiento kitimas (3 parametrai) .....	55
25 pav. Klientų skundų procentas (3 parametrai).....	55
26 pav. IT skambučių centro resursai (3 parametrai).....	56
27 pav. Tinkamumo funkcijos koeficiento kitimas (4 parametrai) .....	57
28 pav. IT skambučių centro resursai (4 parametrai).....	57
29 pav. IT skambučių centro resursų panaudojamumas (4 parametrai).....	58

## Lentelės

1 lentelė. BPMN diagramos elementų grupės .....	19
2 lentelė. <i>HelpDeskProcess</i> klasė.....	44
3 lentelė. <i>SimulationResult</i> klasė.....	45
4 lentelė. <i>GAResults</i> klasė .....	45
5 lentelė. Simuliacijos parametrai .....	50
6 lentelė. Klientų atsiliepimo balai pagal laiką .....	51

## SANTRUMPOS

BPD – BPMN notacijos apibrėžiamos verslo proceso diagramos (angl. *Business Process Diagram*).

BPMN – verslo procesų modeliavimas ir notacija (angl. *Business Process Model and Notation*).

BPS – verslo procesų simuliacija (angl. *Business Process Simulation*).

DBP – dinaminiai verslo procesai (angl. *Dynamic Business Processes*).

KPI – indikatoriai, skirti našumui pasverti (angl. *Key Performance Indicator*).

L1 – pirmasis IT pagalbos skambučių centro lygmuo.

L2 – antrasis IT pagalbos skambučių centro lygmuo.

L3 – trečiasis IT pagalbos skambučių centro lygmuo.

MBSE – modeliais grįsta sistemų inžinerija (angl. *Model Based Systems Engineering*).

MOPSO – genetinio algoritmo rūšis (angl. *Multi-Objective Particle Swarm Optimisation*).

NSGA-II – genetinio algoritmo rūšis (angl. *Non-dominated Sorting Genetic Algorithm II*).

OMG – organizacija, sukūrusi darbe minimus tarptautiniais ISO standartais pripažintus produktus (BPMN, UML, SysML) (angl. *Object Management Group*).

SPEA – genetinio algoritmo rūšis (angl. *Strength Pareto Evolutionary Algorithm*).

SPEA2 – genetinio algoritmo rūšis (angl. *Strength Pareto Evolutionary Algorithm 2*).

UML – modeliavimo ir specifikacijų kūrimo kalba (angl. *Unified Modeling Language*).

## **IVADAS**

Procesai mūsų aplinkoje sutinkami visose srityse. Jais apibrėžiama įvykio ar reiškinio eiga, tam tikras vyksmas. Procesų modeliavimas – tai proceso atvaizdavimas, perkeliant jį iš realybės į supaprastintą ir konkretų modelį. To reikia norint manipuluoti ir simuliuoti galimus procesų scenarijus ir jų baigtimis ne realybėje, o modeliuose. Modeliuojant procesus galima išvengti didžiosios dalies klaidų, netikslumų ir nesusikalbėjimų tiek analizuojamoje srityje, tiek komunikuojant su išore.

Analizuojant tikslo siekiančius verslo procesus, geriausių rezultatų gaunama kuo tiksliau modeliuojant jų eigą, aplinkos veiksnius, galimus procesų vykdymo scenarijus. Netinkamai apibrėžus proceso eigą ar baigtį, atitinkamai kenčia ir priimamų sprendimų teisingumas, funkcionalumo įgyvendinimas. Be viso to, šių laikų verslas, norėdamas išlikti rinkoje, privalo gebėti greitai ir lanksčiai prisitaikyti prie besikeičiančio konteksto. Tai reiškia, jog organizacijos viduje vykstantys procesai turi būti ne tik struktūrizuoti, apibrėžti ir tinkamai sumodeliuoti, bet ir būti simuliuojami skirtingomis aplinkybėmis bei pagal įvairius scenarijus, numatant įvairias baigtis.

Šiame darbe sprendžiama problema yra aktuali beveik kiekvienam verslui – kaip sumodeliuoti verslo procesus ir pritaikyti efektyvias simuliacijas bei jų parametrus, kad verslo vykdoma veikla būtų maksimaliai optimizuota ir teiktų geriausius rezultatus? Visiems svarbu numatyti savo vykdomų procesų galimas baigtis pagal skirtingus scenarijus bei išorės ar vidaus veiksnių daromą įtaką. Tik sužinojus galimą baigtį, įmanoma kurti atitinkamų sprendimų planą ar strategijas.

### **Tyrimo objektas**

Šio baigiamojo magistro darbo tiriamasis objektas – tikslo siekiantys verslo procesai. Analizuojami ir aprašomi jų modeliavimo ir simuliacijos metodai, gauti rezultatai ir išvados.

### **Darbo tikslas ir uždaviniai**

Darbo tikslas – praplėsti tikslo siekiančių verslo procesų simuliacijos būdus, atliekant tyrimą ir ieškant našaus būdo galimoms baigtims numatyti. Tikslui įgyvendinti keliama uždaviniai:

- Atlikti literatūros, susijusios su tikslo siekiančių verslo procesų modeliavimu ir simuliacija, analizę.
- Išnagrinėti verslo procesų modeliavimo ir simuliacijos metodus bei įrankius.
- Pasiūlyti tikslo siekiančių verslo procesų simuliacijos metodą.
- Atlikti pasiūlyto metodo eksperimentinį tyrimą, pateikti eksperimento rezultatus, suformuluoti galutines išvadas ir pateikti tolimesnes mokslinių tyrimų kryptis.

## **Temos naujumas**

Tema plačiai nagrinėjama mokslinėje literatūroje, diskutuojama mokslinių konferencijų metu ir sprendžiama praktiniuose projektuose. Tačiau visi verslų procesai yra labai skirtingi ir sunku rasti vieną bendrą metodologiją, kuria remiantis ši problema būtų sprendžiama universaliai. Nėra vienos bendros išvados ar būdo, kuris užtikrintai teiktų geriausius rezultatus. Dėl to ši tema yra gana nauja, aktuali, tad yra daug erdvės diskusijoms, skirtingoms nuomonėms ir tyrimams.

## **Temos aktualumas**

Tikslo siekiančių verslo procesų modeliavimas ir simuliacija yra aktualūs tiek verslo atstovams dėl konkrečių įmonės poreikių tenkinimo, tiek sistemų inžinieriams, ieškantiems efektyviausių būdų, kaip procesus optimizuoti ar vėliau pritaikyti galimus sprendimus.

## **Tyrimo metodika**

Analitinėje darbo dalyje taikomi bibliotekinio tyrimo ir analizės metodai, apžvelgiant išleistoje literatūroje pateikiamus atvejus ir teorijas. Praktinei darbo daliai (tikslo siekiančių verslo procesų modeliavimui ir simuliacijai atlikti) taikomi konstravimo ir eksperimentinės analizės metodai.

## **Darbo rezultatai**

Atlikus literatūros analizę, apžvelgiamos bendros žinios apie procesus, jų prigimtį, modeliavimo ir simuliacijos metodus. Susipažįstama su darbe nagrinėjamomis sąvokomis, teoriniais faktais, sukuriamas pagrindas, kuriuo remiantis toliau atliekama praktinė dalis – tikslo siekiančių verslo procesų simuliacijos metodo pasiūlymas, jo realizacija ir eksperimentas.

Analizuojant siūlomo metodo galimybes, nusprendžiama apibrėžti tinkamumo funkcijas, reikalingas verslo procesams optimizuoti, bei pritaikyti genetinio algoritmo veikimą, simuliuojant procesų išteklius. Pasiūlyti įrankiai metodo realizacijai, kurie leidžia atlikti eksperimentinį tyrimą ir įrodyti genetinio algoritmo pritaikymo galimybes tikslo siekiančių verslo procesų simuliacijoms.

Darbas apibendrinamas išvadomis ir pasiūlymais, kaip nagrinėjama tema galėtų būtų analizuojama plačiau.

## **Darbo struktūra**

Įvade formuojami darbo tikslai ir uždaviniai, aptariamas darbo aktualumas ir struktūra.

Pirmame šio darbo skyriuje pateikiama literatūros analizė pagal pasirinktus šaltinius baigiamojo darbo temai nagrinėti.

Antrasis skyrius paaiškina siūlomą tikslo siekiančių verslo procesų simuliacijos metodą, kuris pritaikomas darbo tikslui pasiekti. Nusprendžiama pritaikyti genetinio algoritmo veikimą pagal apibrėžtas tinkamumo funkcijas.

Trečiajame skyriuje aprašoma metodo realizacija: koki įrankiai panaudojami metodo įgyvendinimui ir kaip įrankių funkcionalumas pritaikomas simuliacijų vykdymui.

Ketvirtajame skyriuose nagrinėjama eksperimentinė darbo dalis, gilinamasi į gautus rezultatus ir sprendžiamas genetinio algoritmo tinkamumas simuliacijos būdų praplėtimui

Paskutiniuose skyriuose pateikiamos darbo išvados bei teikiami siūlymai tolesniems tyrimams.

# 1. TIKSLO SIEKIANČIŲ VERSLO PROCESŲ, MODELIAVIMO IR SIMULIACIJOS ANALIZĖ

## 1.1. Verslo procesai

Prieš pradedant gilintis į procesų modeliavimą, tikslinga išnagrinėti jo paties sąvoką, galimą klasifikavimą bei metamodelių reikšmę. Pradedant nuo seniausių šaltinių, procesų modeliavimo reikšmė buvo identifikuota jau ir seniau (Roland 1998). Pagrindiniai procesų modeliavimo tikslai yra šie:

- Užfiksuoti, kas tiksliai vyksta proceso metu ir kuo tiksliau tai apibrėžti;
- Pažvelgti iš išorinio stebėtojo perspektyvos ir pastebėti galimus proceso patobulinimus rezultatų našumui ir efektyvumui pasiekti;
- Apibrėžti tam tikras taisykles, nurodymus, elgsenos scenarijus, kurie (jeigu jais bus naudojamosi) atneštų pageidaujamus rezultatus efektyviausiu būdu;
- Pateikti logiškus argumentus, dėl ko tam tikro proceso optimizacija yra reikalinga ir kodėl jis turėtų būti optimizuojamas;
- Rasti tiesioginį ryšį tarp kuriamų modelių ir keliamų reikalavimų.

Nors procesus vienija bendri pirmiau minėti tikslai, jų įvairovė gali būti labai plati. Dėl šios priežasties procesus galima skaidyti į skirtingas grupes pagal skirtingus požiūrius. Vienas iš apžvelgiamos literatūros autorių Sianipar savo 2014 metų publikacijoje pateikė galimą procesų klasifikavimą pagal orientaciją:

- **Orientuoti į veiklą** – tai susijusių, organizuotų ir turinčių savo vyksmo eilę veiklų rinkinys, suburtas bendram tikslui realizuoti;
- **Orientuoti į produktą** – tai veiklų serija, kuri transformuoja produktą ir keičia jo būsenas, kol pasiekama galutinė pageidaujama produkto versija;
- **Orientuoti į sprendimus** – rinkinys tarpusavyje susijusių sprendimų, skirtų galutiniam tikslui pasiekti;
- **Orientuoti į kontekstą** – kontekstų seka, kurios rezultatas veda į sėkmingą produkto transformaciją, darant įtaką priimtiems sprendimams kontekste;
- **Orientuoti į strategiją** – leidžia kurti modelius, kurie reprezentuoja daugiau galimų metodų ir skirtingų būdų procesui detalizuoti.

Procesai gali būti skirstomi pagal apimtį (Roland 1993):

- **Strateginiai procesai** – nagrinėja alternatyvius procesų modeliavimo būdus, dažnai reikalauja kūrybingumo ir žmogaus įsikišimo;
- **Taktiniai procesai** – padeda siekti plano, daugiausiai koncentruoti į faktinio plano taktiką, o ne į plano kūrimą;
- **Realizacijos procesai** – tai žemiausios grandies procesai, tiesiogiai atsakingi už detales, kas ir kaip bus daroma vykdant planą.

Procesų pasitaiko beveik visose įmanomose gyvenimo srityse ir pasižymi didele įvairove. Šiame darbe toliau nagrinėjami būtent tikslo siekiantys (angl. *goal-oriented*) verslo procesai. Tai procesai, skirti užsibrėžtiems tikslams pasiekti ar tam tikroms užduotims, priklausomai nuo verslo specifikos, vykdyti.

## 1.2. Verslo procesų modeliavimas

Geriau suvokus galimas proceso prigimtis ir kryptis, toliau galima analizuoti jų modeliavimo galimybes. Kai modeliuojami verslo procesai, dažniausiai tai reiškia, kad yra poreikis šį procesą optimizuoti ar pakeisti našesniu. Modeliuojant stengiamasi identifikuoti pagrindines problemas ir trikdžius, kurie turėtų būti koreguojami. Tuo tikslu gali būti naudojamos įvairios pažangios technologijos, kurias siūlo daug tiekėjų. Dažniausiai pasirenkami patikimi ir patikrinti metodai: verslo procesų modeliavimas ir notacija (angl. *Business Process Model and Notation*, BPMN), kuris įgauna vis pilnesnio vykdymo galimybių ir artėja prie realybės, taip pat gerai žinomos technologijos kaip UML kalba (angl. *Unified Modeling Language*) ar jos pogrupis SysML, modeliais grindžiama architektūra (angl. *model-driven architecture*) ar į paslaugas orientuota architektūra (angl. *service-oriented architecture*).

Modernių sistemų sudėtingumas šiais laikais auga dideliu tempu. Tai lemia keli skirtingi veiksniai: didėjantys sistemų komponentų skaičiai ir tipai, augantys grafinių sąsajų (angl. *interface*) kiekiai, sistemos funkcionalumo pasiekiamumas už sistemos ribų ir kt. Šiems veiksniams suvaldyti naudojama modeliais grįsta sistemų inžinerija (angl. *Model Based Systems Engineering*, MBSE) (Bajaj *et al.*, 2017).

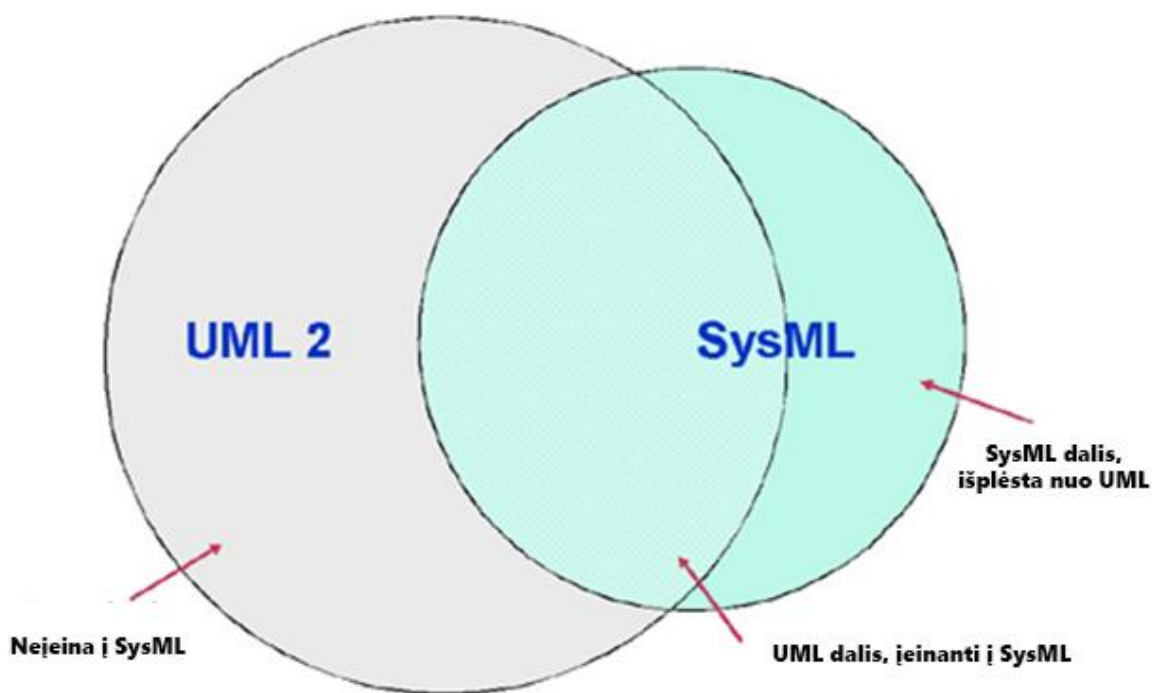
Plačiau nagrinėjant verslo procesams modeliuoti skirtas modeliavimo kalbas, OMG organizacija (angl. *Object Management Group*) yra sukūrusi jau paminėtus plačiai naudojamus, tiksliai dokumentuotus ir tarptautiniais ISO standartais pripažintus produktus: BPMN notaciją, UML modeliavimo kalbą ir jos plėtinį SysML (OMG 2017).

Šios kalbos pasižymi savo diagramų įvairove ir notacijos konkretumu – standartai aiškiai dokumentuoti ir prieinami visiems vartotojams, o modeliavimo kalbos gana lanksčios, kad galima



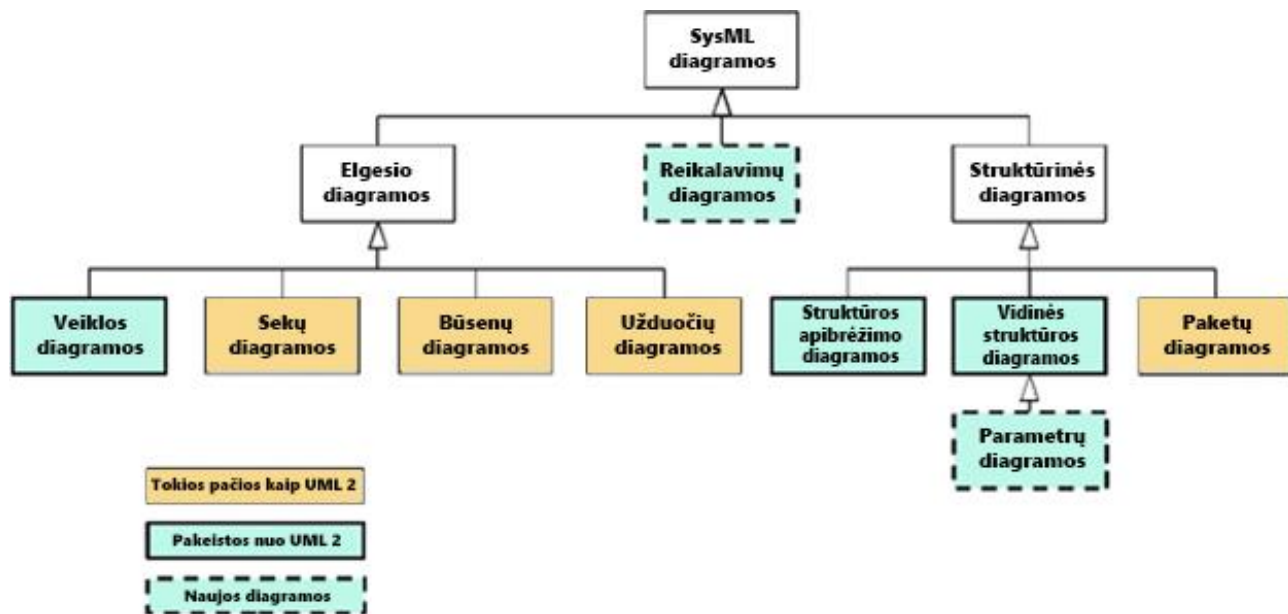
būtų vaizduoti procesus ir tiksliai, ir orientuojantis į kliento poreikius, taip pat suprasti braižomus modelius.

SysML naudojimas pasiteisina kuriant sudėtingas sistemas, kuriose būtinas tikslumas: ligoninių, valdžios, karinių padalinių sistemų. Literatūroje aprašomas ne vienas atvejis, kada SysML modeliavimo kalbos naudojimas davė geriausių rezultatų, pvz., kariniam jūrų laivynui, kurį sudaro daugiau negu keturiasdešimt posistemų, apimančių didelę funkcijų įvairovę – nuo ginklų laikymo ir tvarkymo iki šaldymo ir hidraulikos užtikrinimo (Pearce 2013). SysML dar vadinamas UML 2 kalbos dialektu dėl savo panašių komponentų (žr. 1 pav.).



**1 pav.** Ryšys tarp SysML ir UML (sudaryta autoriaus, remiantis Pearce 2013)

Taip pat glaustai apžvelgiant šių modeliavimo kalbų galimybes, galima sulyginti jų diagramų skirtumus ir panašumus, akcentuojant SysML kalbos diagramų klasifikavimą (Friedenthal 2009) (žr. 2 pav.).



**2 pav.** SysML diagramų tipai (sudaryta autoriaus, remiantis Friedenthal 2009)

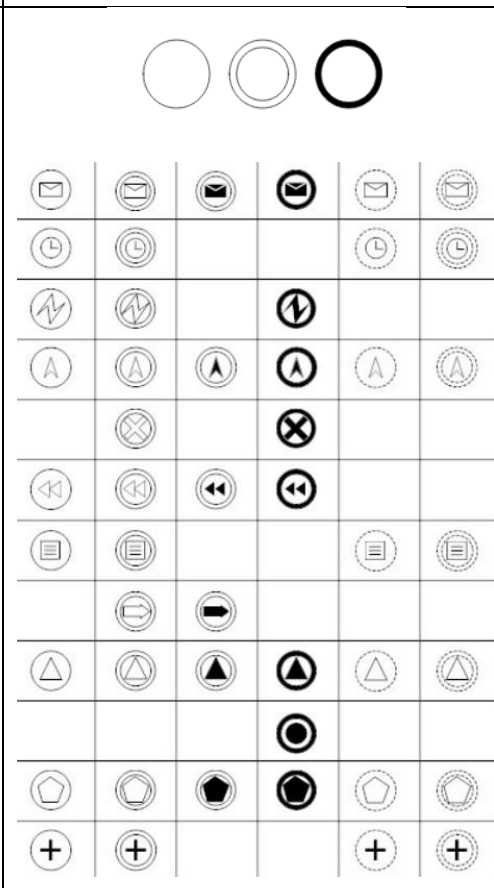
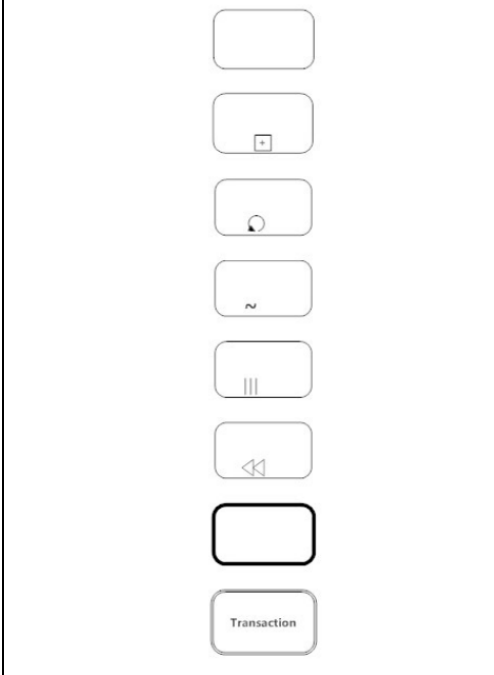
Kad ir kaip būtų, BPMN yra vienas iš lyderiaujančių standartų daugumai kasdienių operacijų ar atvejų. Naudojantis šaltinio <https://bpmnmatrix.github.io> duomenimis, galima identifikuoti daugiau nei 60 oficialių platformų, kurios naudojamos BPMN notacijai realizuoti, atsižvelgiant į šioje matricoje pateiktą suderinamumą su operacinėmis sistemomis, naujumą, kainą. Taip pat randama ir naršyklės versijų (angl. *online*) įrankių, kuriais diagramos gali būti nesunkiai braižomos gyvoje aplinkoje, o didžiosios korporacijos ar svarbūs elektroniniai valdžios organai kuria vidinius, konkrečiai sričiai pritaikytus variklius vartotojui palankiems brėžiniams kurti.







Pirmoji BPMN specifikacijos versija buvo išleista 2004 m., kurią pristatė Stephenas A. White'as, atstovavęs IBM komandai (White 2004). Iškeltas tikslas – pateikti rinkoje notaciją, kuri būtų lengvai suprantama visiems verslo proceso dalyviams: verslo analitikams, braižantiems pradinis proceso karkasus; programuotojams, atsakingiems už diagramose minimo funkcionalumo įdiegimą sistemoje; verslo produkto vartotojams, kurie naudosis procesais ir juos kontroliuos. BPMN notacija apibrėžia verslo proceso diagramas (angl. *Business Process Diagram*, BPD) naudojimą ir grafinių modelių sintaksę.

Prieš 2.0 BPMN versijos išleidimą buvo beveik neįmanoma lengvai perkelti modelių iš vieno įrankio į kitą. Patvirtinus šią versiją, specifikacija apibrėžia standartizuotą apsikeitimo formatą – tai leidžia modelius perkelti net ir į kitų programinės įrangos leidėjų siūlomus įrankius ar sistemas. Taip pat į šią versiją įtraukiamas metamodelio funkcionalumas ir procesų automatizacijos vykdymo specifikacija (Allweyer 2016).

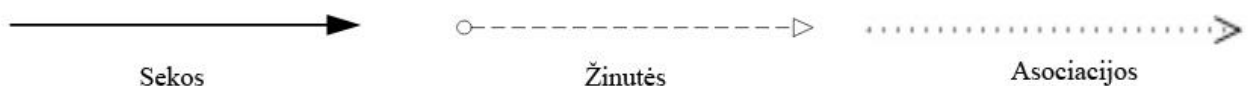
BPD vaizdavimui naudojamos trys elementų grupės: įvykiai (angl. *events*), veiklos (angl. *activities*) ir sąlygos (angl. *gateway*). Lentelėje apžvelgiama kiekvienos grupės elementų paskirtis ir pateikiama keletas kiekvienos grupės elementų pavyzdžių (žr. 1 lentelė) (SmartDraw 2019).

**1 lentelė.** BPMN diagramos elementų grupės

Elementas	Paskirtis	Vaizdavimas
Įvykis	Įvykis vaizduojamas apskritimu. Verslo procese įvykis dažniausiai turi savo priežastį ir rezultatą. Įvykis gali būti trijų tipų: pradžios, tarpinis ir pabaigos. Apskritimai yra tušti, tačiau gali turėti papildomą paskirtį, kuri vaizduojama apskritimo viduje (žinutė, laikmatis, klaida, signalas ar kita rūšis, numatyta specifikacijoje).	
Veikla	Veikla vaizduojama stačiakampiu su apvaliais kampais ir nusako verslo vykdomas užduotis. Veikla gali susidaryti ir iš smulkesnių jungtinių veiklų – tai pažymima subprocesu (angl. <i>subtask</i> ) pridedant pluso ženklą stačiakampyje. Yra ir daugiau skirtingų veiklų žymėjimų.	

Sąlygos	Deimanto formos figūra BPMN diagramoje pažymi proceso sąlygą, išsišakojimą, suliejimą, sujungimą, o vidiniai žymėjimai gali numatyti skirtingą elgseną (BPMB 2011).	     
---------	---	--

Šias tris elementų grupes jungia proceso tėkmės jungtys – sekos, žinutės arba asociacijos (žr. 3 pav.).



**3 pav.** BPMN diagramos jungčių tipai

Egzistuoja ir kitų papildomų žymėjimų, kurie praplečia pirmiau minėtos sintaksės aiškumą.

Ši notacija yra viena iš plačiausiai naudojamų procesų vaizdavimo srityje, tad taip pat bus naudojama ir šiame darbe kaip pagalbinė priemonė darbo tikslui pasiekti.

Dėl didelės procesų įvairovės sunku pritaikyti būtent tam procesui tinkamiausius metodus, modeliavimo kalbas, priemones. Tapo svarbu identifikuoti modelio kokybę. Tad modelių kokybės apibrėžčiai ir sampratai buvo sukurtos šabloninės sistemos – karkasai (angl. *framework*). Viena iš pavyzdžių – Q-Me, kuris suteikia gerai apibrėžtų kokybės savybių rinkinį ir procedūras, kaip objektyviai šias savybes įvertinti (Hommes 2000). Toliau autorius Hommes apibrėžia tokias kokybės savybes (kurios tiesiogiai susijusios su verslo proceso modeliavimo technikomis):

- **Ekspresyvumas** – laipsnis, kuriuo modeliavimo technika geba apibrėžti modelius skirtingo skaičiaus ir kiekio aplikacijoms;
- **Įnoringumas** – laisvės laipsnis, turimas modeliuojant vieną vienetą;
- **Tinkamumas** – laipsnis, kuriuo pasirinkta modeliavimo technika pritaikyta modeliuoti būtent tą pasirinktą vienetą pagal jo specifiką;

- **Supratingumas** – lengvumas, stengiantis vartotojams suprasti pasirinktą modeliavimo techniką, ir lengvumas dirbant su ja;
- **Nuoseklumas** – lygis, koku modeliai kaip atskiros mažos dalys sudaro bendrą nuoseklią visumą;
- **Išbaigtumas** – lygis, kuriuo visi reikalingi konceptai gali būti reprezentuojami modeliais;
- **Našumas** – kiek, modeliuojant procesą, sunaudojama tokių resursų, kaip laikas ar žmogiškieji ištekliai;
- **Efektyvumas** – laipsnis, kuriuo procesų modeliavimas gali pasiekti pradžioje iškeltą tikslą.

Be paliovos atliekama daug tyrimų ir simuliacijų, norint rasti kokybiškų modelių kūrimo optimumą, tačiau sunku apibrėžti kokybės ribas. Nors ir kaip sudėtinga, remiantis Mendling'o šaltiniu, išskirtos keturios skirtingos apibrėžtys kokybės sistemos iš viršaus į apačią (angl. *top-down quality frameworks*), statistikos kokybės aspektai iš apačios į viršų (angl. *bottom-up metrics related to quality aspects*), empirinės apklausos modeliavimo technikos (angl. *empirical surveys related to modelling techniques*) ir pragmatinės gairės (angl. *pragmatic guidelines*) (Mendling 2010).

Vertinant koncepcinio modelio kokybę, dažniausiai atsižvelgiama į lingvistines jo modeliavimo kalbos savybes ir kaip jos kalba bei semantika taikoma konkrečiame modelyje. Taip pat egzistuoja kokybės sistema iš viršaus į apačią SEQUAL (angl. *top-down quality framework SEQUAL*). Ji apibrėžia kelis kokybės aspektus, paremtus ryšiu tarp modelio, žinių, modeliavimo kalbos. Ši sistema neapibrėžia skirtingų kokybės lygių, bet yra dažnai naudojama modeliuojant verslo procesus. Remiantis Krogstie (2006) atliktais tyrimais, yra keletas lygių procesų modelių kokybei vertinti:

- **Sintaksės kokybė** – įvertina, kiek tiksliai modelis yra sukurtas, laikantis pasirinktos modeliavimo kalbos gramatikos;
- **Semantikos kokybė** – kaip tiksliai modelis atitinka užsakovo reikalavimus;
- **Pragmatikos kokybė** – kiek lengvai modelis gali būti suprantamas išorės veikėjų, klientų, užsakovų, neturinčių pasirinktos modeliavimo kalbos žinių;
- **Fizinė kokybė** – ar sukurtas modelis yra pastovus ir ar tikslinė auditorija yra pajėgi jį suprasti;

- **Empirinė kokybė** – ar kuriant modelį buvo atsižvelgta į modeliavimo kalbą kontroliuojančius apribojimus;
- **Socialinė kokybė** – kiek rezultatyvus yra tarpusavio susitarimas su klientas, modeliuojant dalykinę sritį.

SEQUAL turi ir neigiamą pusę – vertinimo sistema yra per daug statinė vertinant semantinę kokybę ir modelius tarpusavyje, o pragmatikos apibrėžtis taip pat ganėtinais siaura.

Apibendrinant pagrindinius veiksnius, kurie lemia verslo procesų modeliavimo kokybę, svarbiausia yra atsižvelgti į kuriamos sistemos sudėtingumą bei apimtį ir nuspręsti, kuri modeliavimo kalba projektuojant galėtų atnešti geriausių rezultatų. Modeliuojant svarbu, kad diagramos ir juose vaizduojami poreikiai bei funkcionalumas būtų ekspresyvūs, išbaigti, gerai suprantami tiek inžinieriams, tiek užsakovams, jog galima būtų užtikrinti reikalavimų vienareikšmiškumą bei tarpusavio susikalbėjimą. Taip pat reikalingas tikslumas, apibrėžtumas ir aiškumas tinkamai realizacijai ir funkcijų implementacijai.

### 1.3. Statiniai ir dinaminiai verslo procesai

Kiti autoriai pateikia kitokį požiūrį. Jie nurodo, jog galima apibrėžti dinامينius verslo procesus – tokius, kurie neturi iš anksto apibrėžtos eigos ir reikalauja kitokio požiūrio pritaikant simuliacinius metodus (Kalibatienė et al., 2016).

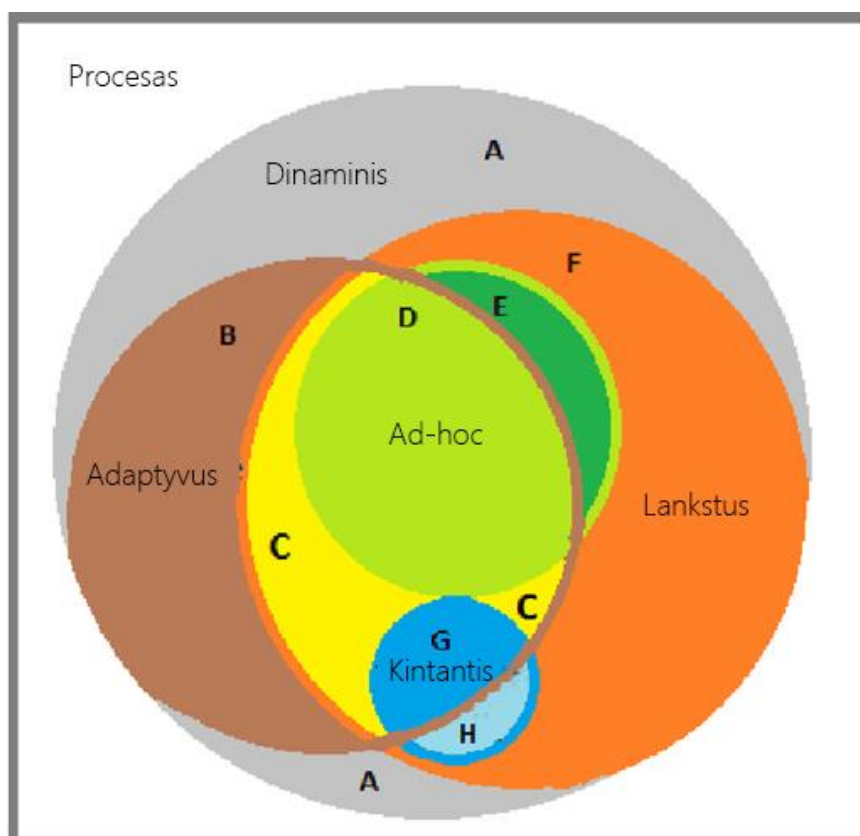
Simuliuojant dinamiškai besikeičiančius procesus, galima analizuoti įvairius scenarijus skirtingomis aplinkybėmis, o tai padeda identifikuoti kritinius proceso kelius. Verslo procesų simuliacija (angl. *Business Process Simulation*, BPS) taip pat suteikia verslo analitikams galimybę verifikuoti, ar siūlomi pakeitimai verslo procesuose gebės patenkinti kylančius poreikius dar prieš jų įdiegimą produkcinėje aplinkoje (Vasilecas 2018).

Tradicioniai verslo simuliacijos standartai reikalauja, jog simuliacinis modelis būtų išsamiai apibrėžtas dar prieš simuliaciją, galimybės keisti modelį simuliacijos metu yra labai stipriai apribotos, priešingai nei prieš tai minėti dinaminiai verslo procesai (angl. *Dynamic Business Processes*, DBP), kurie gali kisti bet kuriuo metu tiek verslo taisyklių, tiek konteksto sferoje.

Dinaminių verslo procesų seka negali būti iš anksto numatyta, tad simuliacijos metu turėtų būti leidžiamos modelio, verslo taisyklių ar konteksto modifikacijos (Vasilecas et al., 2016).

Verslo procesų dinamiškumo kitimas nuo statinio iki dinaminio per kelerius metus turėjo įvairių apibrėžtų formų: *ad-hoc*, kintamasis, adaptyvus, lankstus, dinaminis procesas (žr. 4 pav.) (Rusinaitė et al., 2015):

- **Ad-hoc procesas:** verslo procesas yra besikeičiantis kiekvienu specifiniu atveju. Nėra jokios iš anksto suformuotos proceso apibrėžties.
- **Adaptyvus procesas:** galimybė verslo procesui reaguoti į išskirtines aplinkybes (kontekstus), kurie gali būti ir nenumatyti ir paveiktų vieną ar kelis procesus.
- **Kintantis procesas:** verslo procesas pritaikomas kaip atsakas į vidinę ir išorinę besikeičiančią aplinką.
- **Lankstus procesas:** gebantis reaguoti arba prisitaikyti prie kintančios sistemos aplinkos arba keičiant jos reikalavimus.
- **Deklaratyvus procesas:** tokia proceso eiga fokusuojama siekiant suprasti, kas turėtų būti atlikta, kad būtų galima pasiekti verslo tikslus, neapsiribojant, kaip ir koku būdu konkrečiai galutinė būsena pasiekama.



**4 pav.** Proceso raida nuo statinio iki dinaminio (sudaryta autoriaus, remiantis Rusinaitė et al., 2015)

Taip procesai palaipsniui evoliucionavo nuo statinių iki dinaminių. Nagrinėjant deklaratyviuosius procesus detaliau, jų vaizduojami modeliai savo notacija skiriasi nuo standartinių tuo, jog pradžios ir pabaigos taškai nebūtinai turi būti sujungti su pabaigos taškais. Tokio modelio pagrindinis tikslas – apibrėžti proceso vyksmo taisykles. Visi alternatyvūs keliai yra netiesiogiai specifikuojami ir apibrėžiami kaip keliai, nepažeidžiantys verslo taisyklių. Deklaratyvaus modeliavimo principai, valdant verslo procesus, gali būti taikomi nuo paprastų ECA taisyklių (taisyklės, kurios traktuojamos

taip, jog jei įvyksta įvykis ir sąlyga yra tenkinama, tai atliekamas veiksmas, angl. *Event-condition-action*) iki deklaratyvių procesų modeliavimo kalbų ConDec ir BPCN (Vasilecas 2018).

Reikalavimai, keliami dinaminiam verslo procesui, yra šie:

1. Kiekvienas proceso komponentas turėtų būti laisvai keičiamas, įskaitant sąlygų rinkinius, veiklas, veiklų sekas, sprendimų mazgus ir dalyvius.
2. Veiklos sekos formuojasi per visą proceso eigą ir nėra iš anksto apibrėžtos.
3. Turi būti galimybė atlikti proceso kontekstų keitimus tiek iš vidinių, tiek iš išorinių resursų.
4. Konteksto keitimo trukmė yra kur kas trumpesnė nei viso proceso trukmė.
5. Proceso kitimai gali būti inicijuojami bet kurio vaidmens, bet kuriuo metu su mažu delsos laiku, palyginti su vis proceso trukme.

**Vidiniai ir išoriniai dinaminių verslo procesų kontekstai.** Išoriniu kontekstu gali būti laikomas kintamųjų ir taisyklių rinkinys, kuris apibrėžia tam tikrą aplinkos būseną. Vidiniu kontekstu laikoma esama sistemos resursų būsena. Dinaminis verslo procesas turėtų gebėti reaguoti į besikeičiančius abiejų tipų kontekstus.

**Tam tikros dalys ar visa struktūra.** Tam tikros dalies keitimai turėtų ir paveikti tik tą konkrečią dinaminio verslo proceso dalį, o struktūros kitimai daro įtaką visoms po jos einančioms dalims. Pavyzdžiu galime laikyti veiklų sekos ar komponento keitimą, veiklos trynimą arba pridėjimą į seką.

**Taisyklių dinamika.** Pakeitimai turi būti įdiegti pagal apibrėžtas verslo taisykles. Kiekviena kita veikla turėtų būti parenkama pagal pasirenkamas (angl. *triggered*) ECA taisykles. Turėtų būti leidžiama pridėti naują, keisti esamą arba trinti verslo taisyklę vykdant dinaminio verslo proceso dalį.

**Pakeitimų realizavimas.** Dinaminių verslo procesų pakeitimai turėtų būti atliekami vykdant atitinkamą dalį arba po jos einančią su mažiausia delsa. Bet kuris procese dalyvaujantis vaidmuo neturėtų būti apribotas nuo galimo keitimo.

**Duomenų analizė.** Visi atliekami veiksmai vykdant procesą turėtų būti pateikiami atitinkamuose žurnaluose (angl. *process log file*). Turėtų būti įmanoma atskirti blogas praktikas nuo gerų, kad gerosios būtų pakartotinai naudojamos įvairioms verifikacijoms ar eksperimentams. Laiko, kainos ir kitos proceso tikslų dimensijos turėtų būti apskaičiuojamos ir pateikiamos kiekvienai įvykdytai dinaminio verslo proceso daliai. Blogosios praktikos pavyzdžiai turėtų būti atrinkti ir vėlesniems atvejams nebenaudojami.



#### 1.4. Dinaminių verslo procesų simuliacija

Svarbiausias sėkmingos verslo proceso simuliacijos žingsnis – sumodeliuoti tikslų simuliacijos modelį. Simuliacija naudojama norint suprasti dinaminę proceso eigseną laiko perspektyvoje – kaip procesas reaguos į tam tikrus pakeitimus, įtaką iš aplinkos, parametrų keitimą ar kt. (Vasilecas et al. 2016).

Simuliacinis modelis – tai kompiuterizuotas verslo proceso modelis, kuris įkūnija verslo proceso charakteristikas ir reprezentuoja dinaminę sistemą. Norint simuliacinį modelį taikyti dinaminiam procesams, modeliujama turi būti taip, kad vykdant galėtų būti pritaikomi pakeitimai. Kai pasirenkamos simuliacijos veiklos, proceso eigos kontrolei palaikyti turi veikti eiliškumo mechanizmas (Vasilecas et al. 2015).

Simuliacija apibrėžiama aibe  $S = (p, A, T, C, Q)$ :

- Procesas  $p$  yra veiklų rinkinys.  $p = (a \in A)$ , kai  $A$  yra apibrėžtų įmanomų veiklų procese rinkinys arba dinamiškai pridėtų / išimtų simuliacijos metu;
- Veikla  $a = (r, T_a \in T)$  yra aibė, čia  $T_a$  – užduočių rinkinys, kurį būtina įvykdyti, jei taisyklės sąlyga  $r$  tenkinama.  $T$  yra visų įmanomų užduočių simuliacijoje rinkinys;
- Taisyklė  $r$  yra funkcija, kurios rezultatai nustato, ar visos veiklos vykdymo sąlygos yra patenkinamos;
- Užduotis  $t \in T_a$  yra funkcija, galinti keisti bet kurį simuliacijos būsenos elementą (veiklas, resursus, kontekstą). Užduotis – mažiausias nedalomas ir nepertraukiamas simuliacijos atomas;
- Kontekstas  $C$  yra elementų rinkinys  $c \in C$ , čia  $c = (k, v)$ , kai  $k$  – konteksto elemento vardas, o  $v$  – elemento reikšmė. Kontekstas naudojamas resursams modeliuoti, simuliacijos statistikai ir kitiems parametrų;
- Eilė  $Q$  yra  $q \in Q$  elementų rinkinys, čia  $q = (t, d)$ .  $t$  yra vykdyti paruošta užduotis, o  $d$  – laikas, kai užduotis bus vykdoma.

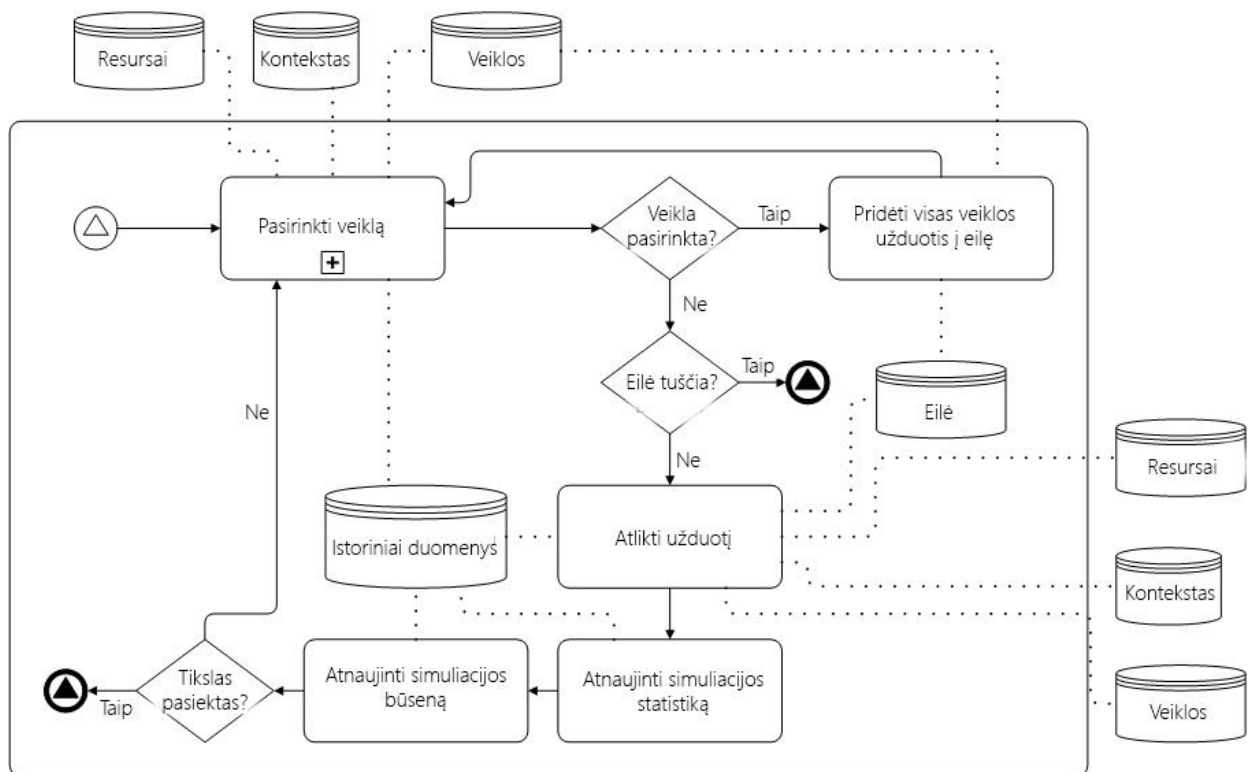
Pagrindiniai dinaminio verslo proceso simuliacijos žingsniai yra tokie (Rus et al. 2014):

1. Simuliacinių duomenų, pavyzdžiui, kaip konteksto, išteklių, veiklų įkėlimas. Šie duomenys galėtų būti importuoti iš išorinių programinės įrangos šaltinių, tokių kaip ERP sistema.
2. Simuliacijos procesas. Simuliacija orientuota į tikslą, o tikslas atsispindi kontekste – kiekviename etape veiklos parenkamos pagal palankiausią tikslui scenarijų.

Simuliacija atliekama tol, kol tikslas pasiekiamas ir eilėje nebėra veiklų, kurios turėtų būti įvykdytos.

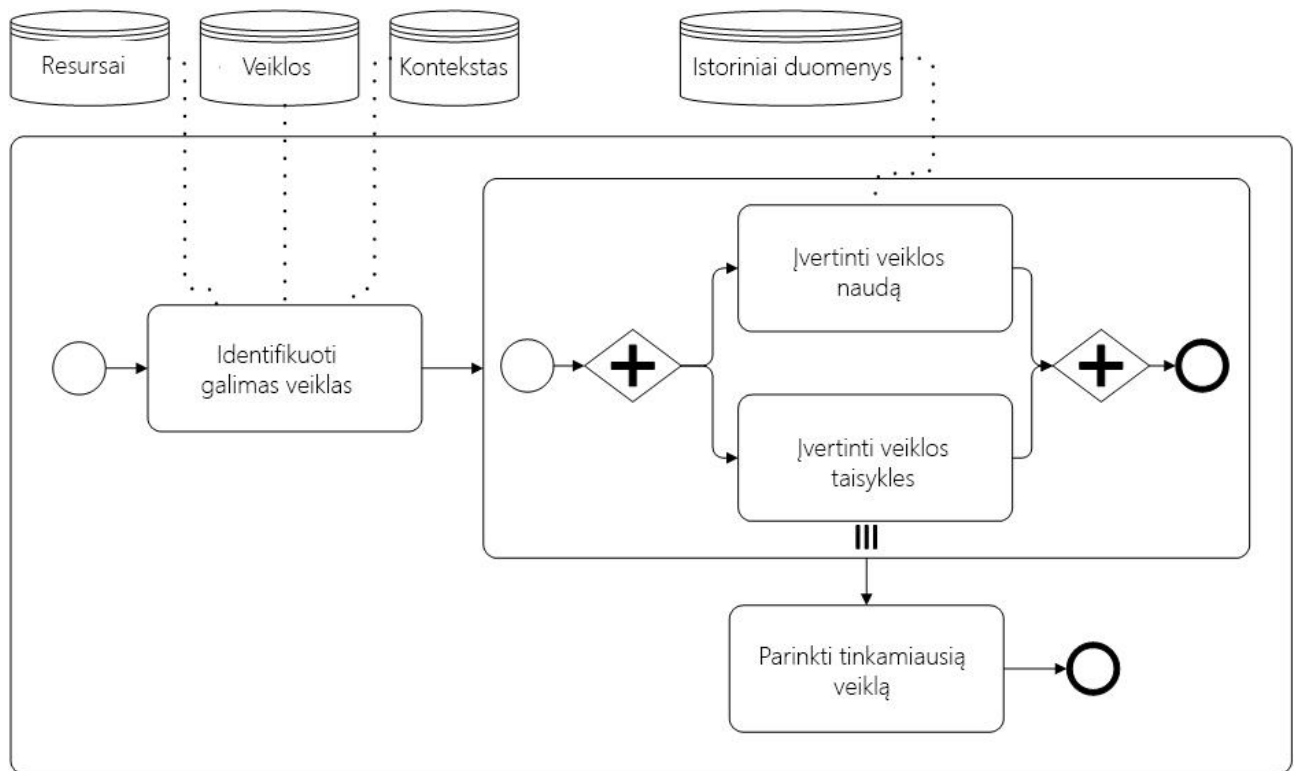
3. Simuliacijos rezultatų analizė. Šiame etape sugeneruojami atliktos simuliacijos rezultatai ir eksportuojama atitinkama vizualizacija.

Pirmasis ir trečiasis etapai yra bendri taikant visus simuliacinius įrankius. Dinaminio verslo proceso simuliacija atliekama antrame žingsnyje (žr. 5 pav.). Kai simuliacinis variklis įvertina visą informaciją (resursus, kontekstą, veiklą ir įrašų žurnalą), jis parenka tinkamiausią veiklą atlikimui. Kai veikla parenkama, joje apibrėžtos užduotys skiriamos vykdymo eilei. Procedūros kartojamos tol, kol nėra pasirenkama jokia veikla ir eilė yra tuščia. Jei eilė tuščia, simuliacijos variklis laukia, kol kontekstas pasikeičia, ir tęsia simuliaciją, kai tik aptinkami pasikeitimai.



**5 pav.** Dinaminio verslo proceso simuliacijos eiga (sudaryta autoriaus, remiantis Vasilecas 2018)

*Pasirinkti veiklą* skaidoma į subprocesą (žr. 6 pav.). Simuliacinis variklis parenka galimų veiklų rinkinį, remiantis turima informacija esamame simuliacijos žingsnyje. Veikla pradeda vertinti, kai tik gaunamas galimų veiklų rinkinys. Vertinimas skaidomas į taisyklių tikrinimą, žiūrint, ar veikla gali būti apskritai atliekama, ir į galimos naudos pasvėrimą. Nauda matuojama remiantis istoriniais duomenimis. Vertinimo pabaigoje simuliacija parenka veiklas, kurios gali būti vykdomos ir kurios yra naudingiausios bendram procesui vykdyti (Vasilecas 2018).



**6 pav.** Veiklos parinkimo subproceso eiga (sudaryta autoriaus, remiantis Vasilecas 2018)

Dauguma šiuolaikinių bandymų optimizuoti verslo procesus atliekami rankomis, netaikant jokios automatizuotos metodologijos. Kai kurie tyrėjai (Vergidis 2007) verslo procesams siūlo naudoti daugiafunkcės optimizacijos karkasus (angl. *framework for multi-objective optimisation*). Karkase naudojamas formaliai apibrėžtas bendras verslo proceso modelis, nurodantis kainą ir trukmę kaip tinkamumo funkcijas. Verslo proceso modelis suprogramuojamas ir įtraukiamas į programinės įrangos platformą, kur atrinkti daugiafunkciai optimizacijos algoritmai naudojami daugeliui testų, įtraukiant ir realius pavyzdžius.

Verslo procesai gali varijuoti tarp skirtingų sudėtingumo lygių ir taip pat gali būti optimizuojami plačiai naudojamais genetiniais algoritmais: NSGA-II (angl. *Non-dominated Sorting Genetic Algorithm II*), SPEA (angl. *Strength Pareto Evolutionary Algorithm*), SPEA2 (angl. *Strength Pareto Evolutionary Algorithm 2*), MOPSO (angl. *Multi-Objective Particle Swarm Optimisation*). Rezultatai rodo, kad nors verslo procesų optimizacija yra stipriai suvaržyta dėl fragmentuotos paieškos erdvės, tokie daugiafunkciai optimizacijos algoritmai, kaip NSGA-II ir SPEA2, teikia nemažą skaičių alternatyvų verslo proceso optimizacijai (Khare et al., 2003). Nors ir kaip būtų, optimizacijos algoritmų stipriai mažėja sudėtingėjant procesų modeliams.

Kadangi tarp procesų simuliacijos ir optimizacijos įrankių integracijos nėra nustatytos ir apibrėžtos formalios struktūros, tai trukdo simuliaciniam procesui balansuoti tarp tokių pageidaujamų savybių, kaip našumas, bendrumas, skaidrumas, didelės apimtys. Viena iš galimų parinkčių – Zeiglerio karkaso (angl. *Zeigler's framework*) implementacija (Zeigler et al., 1994). Svarbus šio

karkaso konceptas yra jo eksperimentiniai rėmai, kurie paaiškina atskirtį tarp modelio ir bet kokių duomenų surinkimo (pavyzdžiui, statistinių rodiklių) bei kontrolės (pavyzdžiui, simuliacijos paleidimo ir stabdymo), kurie nėra atliekami realioje sistemoje. Tam, kad šios funkcijos būtų įgyvendintos, konceptas formalizuotas trimis komponentais: generatoriumi (angl. *generator*), keitikliu (angl. *transducer*) ir akceptoriumi (angl. *acceptor*) (Zeigler et al., 2008).

Verslo procesui sukūrus tinkamą simuliacinį modelį remiantis pirmiau pateikta metodika, toliau ieškoma būtų, kaip efektyviau keisti galimus simuliacijos parametrus, tikintis artėjimo prie iškelto tikslo. Tam gali padėti genetinių algoritmų pritaikymas.

## **1.5. Genetiniai algoritmai**

### **1.5.1. Natūralios atrankos principai**

Remiantis Čarlzo Darvino teorija nuo 1831 metų, gamtoje vyksta natūrali atranka. Informacinių technologijų specialistai rėmėsi šia teorija pritaikydami natūralios atrankos principą kaip optimizacijos techniką ir suformuoti visa tai į genetinius algoritmus.

Natūralios atrankos principas teigia, jog prie aplinkos labiau prisitaikę organizmai bus atrinkti ir sudarys daugumą visos populiacijos dalies. Natūralios atrankos sąlygos susidaro dėl genetinių skirtumų tarp individų, o genetiniai skirtumai atsiranda dėl atsitiktinių mutacijų (Abdalla 2017). Šiam procesui sudaryti keli reikalavimai:

- Populiacijai priklausantys objektai turi gebėti perduoti savo charakteristikas.
- Charakteristikų įvairovė turi būti tam, jog jos toliau galėtų kisti ir prisitaikyti (arba ne).
- Turi egzistuoti selekcijos procesas, kuriuo remiantis atrenkami populiacijos objektai, turintys tinkamiausias charakteristikas. Tai vadinama tinkamiausių išlikimu (angl. *survival of the fittest*).

Bandant šį principą perkelti į algoritmų logiką, galimi šie žingsniai:

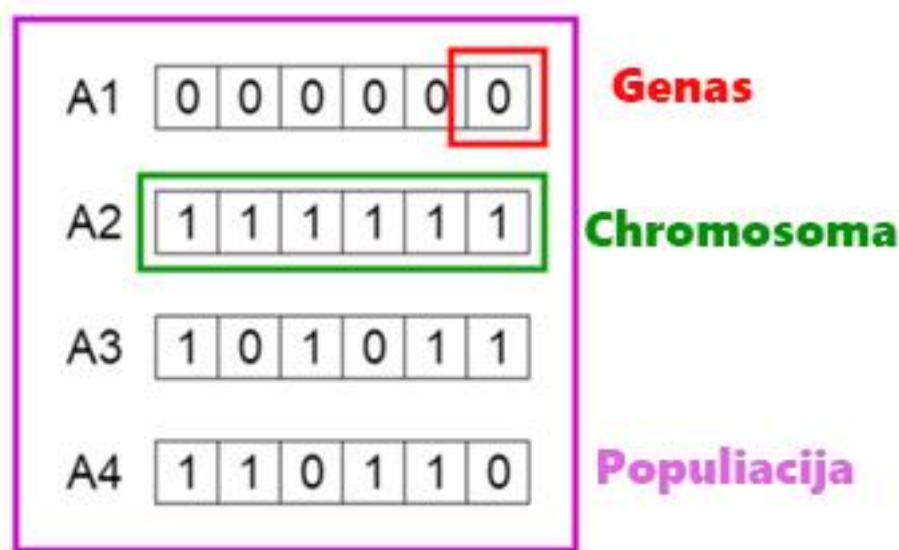
- Apibrėžti objektų populiaciją;
- Skaičiuoti kiekvieno objekto atitiktį selekcijos procesui ir surūšiuoti juos nuo labiausiai iki mažiausiai tinkamų;
- Palikti populiacijoje tinkamiausius ir išmesti mažiausiai tinkamus;
- Įvesti atsitiktines kai kurių objektų mutacijas;
- Kartoti žingsnius nuo skaičiavimo, kol pasiekiamas reikalingas atitikimo lygmuo (Martin 2001).

### 1.5.2. Genetinių algoritmų fazės

Genetiniai algoritmai apibrėžiami ir konkretinami penkiomis skirtingomis fazėmis:

- Pradinės populiacijos apibrėžimas;
- Tinkamumo funkcijos apibrėžimas;
- Selekcijos vykdymas;
- Kryžminimas;
- Mutacija.

Procesas pradedamas objektų rinkinio apibrėžimu – populiacija (angl. *population*). Kiekvienas jos vienetas yra problemos, kurią norima spręsti, sprendimas. Kiekvienas šis vienetas charakterizuojamas parametrais (kintamaisiais), kurie įvardijami kaip genai (angl. *genes*). Genai yra jungiami tarpusavyje į eilutę (angl. *string*) ir taip suformuojami į chromosomas (angl. *chromosome*) – tai prilyginama sprendimui (žr. 7 pav.). Dažniausiai naudojamos binarinės reikšmės (Mallawaarachchi 2017b).



7 pav. Genas, chromosoma ir populiacija

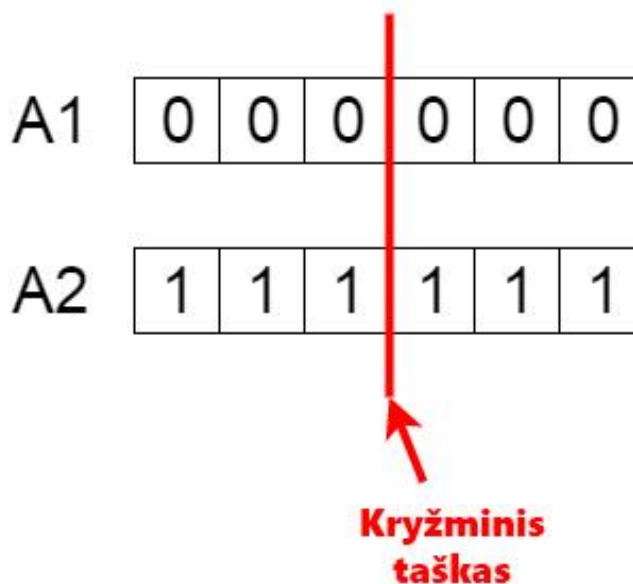
Genetiniuose algoritmuose pagrindinė funkcija, atliekanti atranką, vadinama tinkamumo funkcija (angl. *fitness*). Ji vertina, kaip arti prie optimumo taško atitikties objektas yra. Kiekvienas objektas reprezentuojamas kaip eilė binarinių skaičių, kuri vadinama chromosoma. Tinkamumo funkcija turi būti aiškiai apibrėžiama ir jos naudotojas privalo aiškiai suvokti, kaip tinkamumo reitingas bus skaičiuojamas. Tinkamumo funkcija taip pat turi būti tinkamai įdiegta ir nestabdyti viso algoritmo veikimo. Netinkamai pritaikyta tinkamumo funkcija gali būti pagrindiniu viso sklandaus proceso stabdžiu. Ji turėtų grąžinti intuityvius rezultatus, įvardydamas geriausius ir blogiausius kandidatus su kiekybiškai pamatuotais rezultatais (Mallawaarachchi 2017a).

Algoritmai veikia rezultatyviai, jei jie parenkami tinkamai problemai spręsti. Tinkamumo funkcija turi būti pritaikyta konkrečios problemos sprendimui ir ne visada tinka visiems atvejams. Tai bene sunkiausia algoritmo formavimo dalis. Kaip ir kaip būtų, informacinių technologijų specialistai bando taikyti tam tikras funkcijas skirstydami problemas pagal tam tikras grupes.

Klasifikavimo užduotims vertinti dažnai renkamasi Euklidinio ar Manheteno atstumo principai, taikomi tinkamumo funkcijoms formuoti. Optimizavimo atvejams spręsti gali būti taikomos eilinės sumos funkcijos iš apskaičiuotų parametrų rinkinių, susijusių su problema.

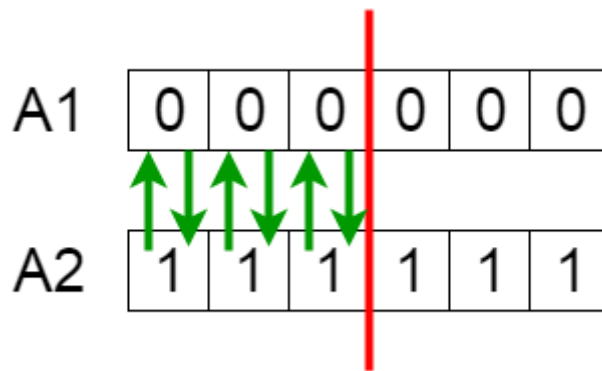
Selekcijos fazės metu tinkamiausi objektai yra paliekami ir jų genai toliau gali būti perduodami kitai kartai. Vienetai reprodukcijai parenkami pagal geriausius tinkamumo rodiklius.

Kryžminimo fazė yra viena iš reikšmingiausių algoritmo procese. Kiekvienai tėvinei porai atsitiktinai parenkamas kryžminis taškas (angl. *crossover point*). Pateiktame pavyzdyje (žr. 8 pav.) kryžminio taško reikšmė yra 3 (Mallawaarachchi 2017b).



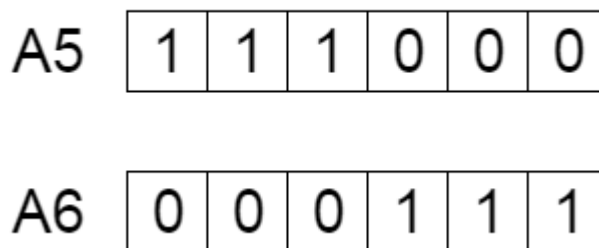
8 pav. Kryžminis taškas

Palikuonys (angl. *offspring*) kuriami keičiant genus tarpusavyje, kol pasiekiamas kryžminis taškas (žr. 9 pav.).



**9 pav.** Palikuonių kūrimas, keičiant genus iki kryžminio taško

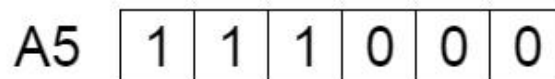
Atlikus kryžminimą, nauji palikuonys prijungiami prie populiacijos (žr. 10 pav.).



**10 pav.** Naujų palikuonių prijungimas prie populiacijos

Tam tikruose palikuonyse su maža atsitiktine tikimybe genai mutuoja. Bitų eilutėje bitai gali būti apverčiami – tai atliekama įvairovei populiacijoje palaikyti (žr. 11 pav.)

#### Prieš mutaciją



#### Po mutacijos



**11 pav.** Genų mutacija

Algoritmas nutraukiamas, jei populiacijoje palikuonys nebėra vedami turint akivaizdžius tarpusavio skirtumus, lyginant su praėjusia karta. Tada nutariama, jog genetinis algoritmas pateikė iškelto problemos sprendimų rinkinį (Mallawaarachchi 2017b).

Populiacijos dydis yra fiksuotas. Erdvė naujiems palikuonims yra suformuojama, kai mažiausią tinkamumo rodiklį turintys vienetai yra panaikinami iš populiacijos.

Genetiniai algoritmai ne visada randa optimalų sprendimą dėl savo atsitiktinumo prigimties. Tai, kaip kokybiškai algoritmas gebės rasti sprendimą, dažnai priklauso ne tik nuo tinkamumo funkcijos, bet ir nuo būdo, kaip duomenys koduojami. Kai tai atliekama tinkamiausiu metodu, tokie algoritmai gerai tinka tam tikroms sekoms optimizuoti, pavyzdžiui darbuotojams priskirti tvarkaraščiuose arba oro linijų tvarkaraščiuose (Koepec 2018).

NASA genetinius algoritmus naudoja radijo antenų automatizacijoje, taip pat tokių algoritmų naudojimas gerai žinomas *Wall Streeto* akcijų prekybos taisyklėse, kurios taikomos kartu su neuroniniais tinklais išteklių selekcijai ir prekybai identifikuoti. Naudojant genetinių algoritmų simuliacijas, kontroliuojami ir eismo veiksniai arba iliustracijų bei paviršių atpažinimo funkcijos (Cheu 2004). Tai nurodo genetinių algoritmų naudojimo įvairovę ir svarbą praktikoje.

### 1.5.3. Daugiatiksliai genetiniai algoritmai

Sudėtingesniems atvejams naudojama daugiatikslė optimizacija (angl. *multi-objective optimization*) – tai optimizacijos sritis, grindžiama sprendimų priėmimu pagal kelis kriterijus. Ji sutelkta į matematinės optimizacijos problemas, turinčias daugiau nei vieną tikslo funkciją optimizacijai tuo pačiu metu. Tikslo funkcija (angl. *objective function*) – tiesinio programavimo funkcija, skirta skaitinėms reikšmėms maksimizuoti arba minimizuoti (Courses PSU 2016).

Tai skatina naudoti daugiatikslius genetinius (evoliucinius) algoritmus. Keli problemos tikslai kelia poreikį gauti ne vieną optimalų sprendimą, bet optimalių sprendimų rinkinį (kitaip žinomą kaip Pareto optimalūs sprendimai). Klasikiniai optimizavimo metodai siūlo skaidyti daugiatiksles problemas į vienatiksles, gaunant vieną optimalų Pareto sprendimą kiekvienai iš jų.

Pareto optimalus rinkinys dažniausiai apibrėžiamas dviem tarpusavyje besipriešinančiais tikslais – distancija iki optimumo turi būti minimizuota ir sugeneruotų sprendimų įvairovė turi būti maksimizuota. Šiame kontekste, modeliuojant daugiatikslį genetinį algoritmą, yra du fundamentalūs veiksniai – poravimo selekcija ir aplinkos selekcija. Pirmasis prioritetas tiesiogiai orientuotas į klausimą – kaip valdyti paiešką siekiant Pareto optimalaus sprendimo. Turint individų aibę, tinkamumo funkcijos turi būti taikomos renkant, kurie individai tarpusavyje turėtų versti palikuonis. Procedūra dažniausiai atliekama atsitiktiniu būdu. Antrasis veiksnys sprendžia klausimą, kuriuos individus išlaikyti evoliucijos procese. Dėl ribotų laiko ir vietos resursų tik dalis individų kartos gali būti perkelta į kitą kartą.

Moderniuose evoliuciniuose algoritmuose šie du kriterijai realizuojami tokiu būdu:

- **Poravimo selekcija.** Vertinama dviem etapais. Pirmiausia individai lyginami pagal Pareto dominavimą (ang. *Pareto dominance relation*), kuris apibrėžia dalinę tvarką ir



sureitinguoja individus. Po to reitingas patikslinamas atsižvelgiant į tankumą. Tankumui ir nišos dydžiui, kurioje individai išsidėstę, nustatyti naudojama daug skirtingų technikų.

- **Aplinkos selekcija.** Be populiacijos, egzistuoja ir aplinka – archyvas, kuriame saugoma nedominuojanti aibė po visų iki tol įvykdytų sprendimų. Archyvo narys išmetamas, jei: 1) rastas sprendimas, kuris dominuoja archyvo nario atžvilgiu; 2) pasiektas maksimalus archyvo dydis ir lokacija, kurioje laikomas individas, yra perkrauta. Būti kopijuojamam į archyvą yra bene vienintelis būdas individui išgyventi kelias kartas ir pagal atsitiktinumą turėti galimybę pratęsti giminę. Ši technika taikoma norint neprarasti tam tikrų nedominuojančių narių rinkinių dėl atsitiktinių procesų poveikio.

Abi selekcijos schemas iš pricipo yra nepriklausomos viena nuo kitos. Tačiau aplinkos selekcija yra paremta Pareto principu, o poravimo – svorio skaičiavimu įvertinant tinkamumo reikšmes. Kad ir kaip būtų, abi iš jų daugumos evoliucinių algoritmų realizuojamos panašiu būdu (Courses PSU 2016).

#### 1.5.3.1. NSGA-II algoritmas

Vienas iš daugiausia nagrinėjamų daugiatislių genetinių algoritmų yra NSGA-II (angl. *Non-dominated Sorting Genetic Algorithm II*). Remiantis Deb (2002), jis turi pranašumą daugiatislių genetinių algoritmų įvairovėje, nes paneigia tris pagrindinius kitų algoritmų trūkumus:

- $O(MN^3)$  sudėtingumą, čia  $M$  yra lygus tikslų skaičiui, o  $N$  – populiacijos dydžiui;
- Elitizmo nebuvimą. Elitizmas – geriausių chromosomų suradimo procesas tam tikroje populiacijos kartoje ir išlaikantis jas nepaveiktas kitai kartai. Tai atliekama norint neprarasti geriausių chromosomų kryžminimo ir mutacijų metu (Wyld 2011);
- Poreikį specifikuoti besidalijamąjį parametą.

Simuliacijos rezultatai rodo, kad NSGA-II algoritmas daugumoje situacijų yra pajėgus rasti geresnį sprendimų paskirstymą ir geresnį artėjimą prie optimalaus Pareto sprendimo (angl. *Pareto optimal solution*) nei kiti elitizmą palaikantys algoritmai. Sprendimas vadinamas optimaliu Pareto, kai nė viena iš tiesinių funkcijų negali būti patobulinta nemažinant kitų tiesinių reikšmių (IGI Global 2018).

NSGA-II geba rasti kelis optimalius Pareto sprendimus vykdant vieną simuliaciją. Jis dažnai lyginamas su PAES (angl. *Pareto-archived evolution strategy*), SPEA (angl. *strength-Pareto EA*) ir SPEA2 (angl. *strength-Pareto EA 2*), tačiau kai kurių tyrimų metu pateikta argumentų dėl jo pranašumo (Deb 2002).

### 1.5.3.2. SPEA ir SPEA 2 algoritmai

Nagrinėjant dažnai su pirmiau minėtu algoritmu lyginamą SPEA algoritmą, pradinė jo versija buvo pristatyta 1998 m. (Zitzler et al. 1998). Jis gerokai viršijo tuo metu egzistavusių neelitistinių algoritmų našumą ir iškilo tarp labiausiai diskutuojamų ir tarpusavyje lyginamų algoritmų. SPEA naudoja įprastą populiaciją ir archyvą. Pradedant nuo pirminės populiacijos ir tuščio archyvo, vykdomos iteracijos. Pirmiausia visa nedominuojanti populiacija nukopijuojama į archyvą. Visi dominuojantys individai arba dublikatai (atsižvengiant į reikiamas reikšmes) pašalinami iš archyvo atnaujinimo metu. Jei atnaujinto archyvo dydis viršija numatytą limitą, kiti archyvo nariai šalinami remiantis klasterizavimo (angl. *clustering*) technika, kuri išsaugo nedominuojančio fronto charakteristikas (Gro 2002). Paskui pritaikomos tinkamumo reikšmės tiek archyvo, tiek populiacijos nariams:

- Kiekvienam individui  $i$  archyve priskiriama stiprumo reikšmė  $S(i) \in [0, 1]$ , kuri tuo pačiu metu reprezentuoja savo tinkamumo funkcijos  $F(i)$  reikšmę.  $S(i)$  yra skaičius populiacijos narių  $j$ , kurie yra dominuojami arba lygūs  $i$ , atsižvelgiant į tikslines reikšmes, padalintas iš populiacijos dydžio ir pridėdant vieną.
- Individo  $j$  populiacijoje tinkamumas  $F(j)$  apskaičiuojamas sumuojant visų archyvo narių  $i$  stiprumo reikšmes  $S(i)$ , kurie dominuoja arba yra lygūs prieš  $j$ , pridėdant vieną pabaigoje.

Kitas žingsnis – poravimo selekcijos fazė, kur individai iš populiacijos ir archyvo junginio parenkami pagal binarines reikšmes. Tinkamumas šiame etape yra minimizuojamas – kiekvienas individas archyve turi didesnę tikimybę būti pasirinktas nei individai iš populiacijos. Po rekombinacijos ir mutacijos senoji populiacija pakeičiama palikuonių populiacija (Zitzler et al. 2001).

Analizuojant šį algoritmą ir pritaikant skirtingiems eksperimentams, atrastos silpnosios vietos:

- **Tinkamumo priskyrimas.** Individai, kurie nudominuoti tų pačių archyvo narių, turi identiškas tinkamumo reikšmes. Tai reiškia, jog tokiu atveju, kai archyve bus tik vienas individas, visi populiacijos nariai turės tą patį reitingą nepriklausomai nuo to, ar jie dominuoja vienas kito atžvilgiu, ar ne. Tokiu atveju SPEA algoritmas elgsis kaip atsitiktinės paieškos algoritmas.
- **Tankumo įvertinimas.** Jei daug individų esamoje kartoje yra sąlyginai panašūs, t. y. nedominuoja vienas kito atžvilgiu, informacijos rikiuojant populiaciją išgaunama labai mažai. Tokia situacija yra tikėtina, kai egzistuoja daugiau nei du tikslai. Tada tankumo informacija naudojama efektyvesnės paieškos tikslais. Klasterizacija tokiu atveju gali būti naudinga tik archyvui, bet ne populiacijai.

- **Archyvo mažinimas.** Nepaisant to, kad klasterizacijos funkcija leidžia sumažinti nedominuojantį rinkinį neprarandant reikalingų charakteristikų, įmanoma prarasti papildomus sprendimus, kurie turėtų būti kaupiami archyve ir suteikti galimybę turėti didelę nedominuojančių sprendimų įvairovę.

Išleidus NSGA-II, SPEA algoritmas nublanko atlikus tam tikrus testus. Atsižvelgiant į trūkumus, SPEA algoritmas buvo toliau tobulinamas ir išleista nauja jo variacija SPEA2. Pagrindiniai dalykai, skiriantys SPEA nuo SPEA2:

- Patobulinta tinkamumo priskyrimo schema, kuri įtraukia kiekvieną individą ir nustato, kelių individų atžvilgiu jis dominuoja ir kiek individų dominuoja jo atžvilgiu.
- Pritaikoma artimiausio kaimyno tankumo apskaičiavimo technika, kuri leidžia tiksliau laviruoti paieškoje.
- Atnaujinti archyvo mažinimo metodai garantuoja geresnį ribinių sprendimų išsaugojimą archyve.

### 1.5.3.3. PESA algoritmas

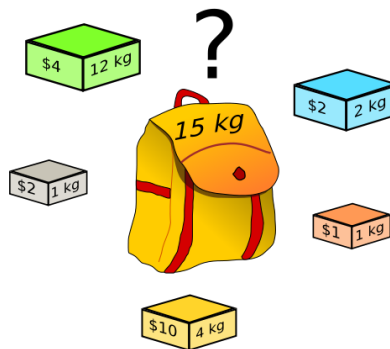
PESA (angl. *Pareto Envelope-Based Selection Algorithm*) algoritme poravimo selekcija yra atliekama tik archyve, kuriame yra nedominuojantis rinkinys (Corne et al. 2000). Tankio matavimai gali būti klasifikuojami histograminėmis technikomis, o tai leidžia paskirstyti archyvo narius pagal tankumo lygį. Sugeneruoti palikuonys, sudarantys tikrąją populiaciją, pagal tai įtraukiami į archyvą. Individai, kurie nepatenka į archyvą, yra pašalinami prieš pradedant kitos kartos ciklą. Šiuo požiūriu poravimo ir aplinkos selekcija yra identiška pagal selekcijos kriterijus ir skiriasi tik selekcijos procesu (atsitiktinis prieš deterministinį). Toks požiūris taikomas ir NSGA-II algoritmui – individai išskirstomi į skirtingus frontus pagal Pareto dominavimo principą. Nariai, kurių reitingas aukščiausias, bus priskirti pirmajam frontui, antri pagal reitingą – antrajam ir t. t. Kiekvienas reitingas, reprezentuojantis distanciją tarp dviejų arčiausių kiekvieno tikslo kaimynų sumą, padeda sudaryti rūšiuotą eilę tarp individų. Remiantis šiuo reitingu atliekama tiek poravimo, tiek aplinkos selekcija. Kombinuojant tėvinę (kuri gali būti laikoma archyvu) ir palikuonių populiaciją bei ištrinant 50 proc. blogiausiųjų, individų atsargos yra sumažinamos. Paskui atliekamos binarinės atrankos tarp likusių individų (archyvo narių) kitai palikuonių kartai generuoti.

Naudojant PESA algoritmą, archyve galima rasti tiek nedominuojančių, tiek dominuojančių individų rinkinių. Vadovaujantis PESA principu archyvas gali būti užpildomas iš dalies, o NSGA-II algoritmu archyvas visada bus užpildomas iki galo (Zitzler et al. 2001).

#### 1.5.3.4. NSGA-II, SPEA, SPEA2 ir PESA algoritmų lyginimas

NSGAI, SPEA, SPEA2 ir PESA algoritmų elgesys dažnai analizuojamas ir lyginamas tarpusavyje, vykdam testus, turinčius skirtingas funkcijas. Nagrinėjant vieną iš Zitlerio aprašytų testų (Zitzler et al., 2001), apžvelgiami gauti rezultatai ir kiekvieno algoritmo elgsena.

Taikomas „kuprinės“ uždavinio pavyzdys (angl. *knapsack problem*) – tai kombinatorinio optimizavimo užduotis, kai, turint daiktų rinkinį, kurio kiekvienas objektas turi savo svorį ir vertę, reikia apibrėžti kiekvieno daikto skaičių, kuris turėtų būti įtrauktas į pasirinkimus, kad svorio suma neviršytų leistino limito ir kad vertė būtų kuo didesnė. Pavyzdžiui, kurios dėžės turėtų būti pasirinktos, kad pinigų kiekis būtų kuo didesnis neviršijant 15 kilogramų (žr. 12 pav.).



12 pav. „Kuprinės“ uždavinys

„Kuprinės“ uždavinio sąlygoje bus naudojama 750 daiktų, turint 2 ir 3 tikslus. Individai reprezentuojami kaip bitų eilutės, kur kiekvienas bitas atitinka vieną sprendimo kintamąjį. Du individai kryžminami pagal vieną kryžminimo tašką. Mutacija įvyksta, kai bito apsisvertimo tikimybė yra 0,006. Populiacijos ir archyvo dydis – 250, kai  $m = 2$ , ir 300, kai  $m = 3$ . Kiekvienos problemos sunkumas nustatomas turint 100 sprendimo kintamųjų. Naudojamas sferos modelis (angl. *Sphere Model*) SPH- $m$  ir imami dideli režiai, siekiant ištestuoti algoritmų gebėjimą didelėje erdvėje aptikti Pareto optimalius rinkinius. SPH- $m$  funkcijai priskiriami režiai  $[-10^3; 10^3]^n$ , o tikslo funkcija apibūdinama kaip  $f_j(x) = \sum_{1 \leq i \leq n, i \neq j} (x_i)^2 + (x_j - 1)^2$ , kai  $1 \leq j \leq m$ ;  $m = 2; 3$ .

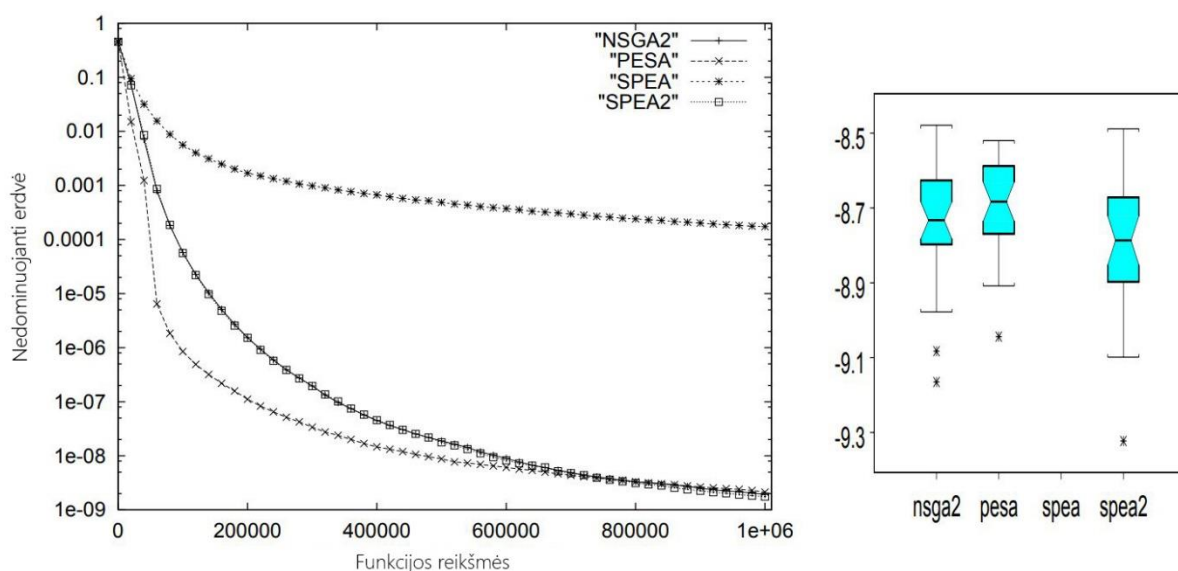
Kiekvienam algoritmui ir kiekvienai problemai buvo leidžiama 30 atvejų. Kokybei ir našumui matuoti taikomas kiekybės požiūris su švelniomis modifikacijomis. Tikslas – minimizuoti erdvės dalį, kuri nėra dominuojama nė vieno galutinio archyvo nario. Tai bus laikoma tinkamiausiu skaliariniu indikatoriumi, nes kombinuojamas tiek sprendimų atstumas, tiek pasiskirstymas. Kiekvieno paleidimo metu matuojamas normalizuotas nedomnuojančio objekto erdvės dydis, suteikiant 30 reikšmių (tiek paleidimų buvo vykdoma eksperimento metu).

Problema, su kuria algoritmas susiduria apdorodamas funkciją, yra Pareto optimalaus rinkinio erdvės lokacija, kuri, priklausomai nuo sprendimo kintamųjų rango, gali suformuoti labai mažą erdvę

paieškos lauke. Artėjant prie Pareto optimumo, tikslumas turi būti reguliuojamas tinkamai, tačiau artėjant tai padaryti vis sunkiau, nes pasisėkimo rodiklis krinta labai greitai (Laumanns et al 2001).

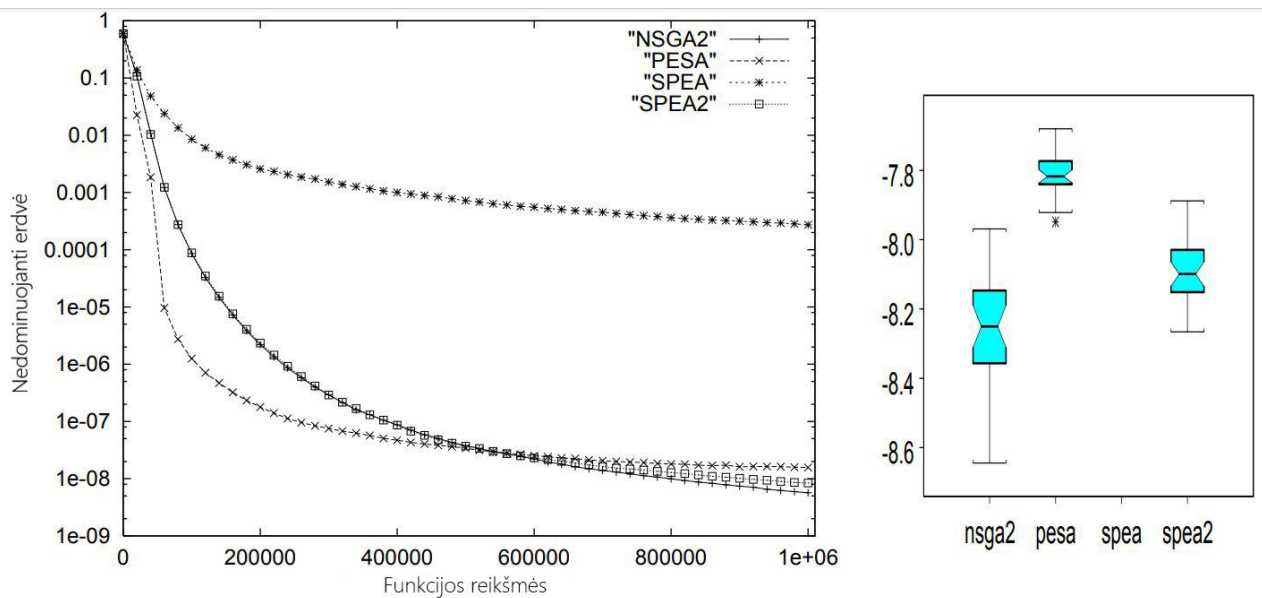
Iš atlikto eksperimento rezultatų matoma, kad SPEA algoritmas abiem sąlygomis nekonverguoja link Pareto optimalių sprendimų regiono. Taip nutinka dėl tinkamumo priskyrimo strategijos. Archyvas sudarytas iš vieno individo, kuris dominuoja visų kitų atžvilgiu ir suteikia tokią pačią tinkamumo reikšmę visiems kitiems ir tą pačią selekcijos tikimybę.

PESA algoritmas, palyginti su likusiais dviem, yra spartesnis pirmoje leidimų serijoje, aptinkant Pareto optimalių sprendimų rinkinį. Tai greičiausiai nutinka dėl intensyvesnio elitizmo, nes šiame etape archyvas sudarytas tik iš kelių punktų. Esant arti Pareto optimalaus fronto, SPEA2 ir NSGA-II algoritmai aplenkia PESA algoritmą. Tai galima pastebėti kairėje diagramų pusėje, kuriose vaizduojamos algoritmų vidutinės gautos reikšmės laiko atžvilgiu (žr. 13 pav. ir 14 pav.).



**13 pav.** Algoritmų rezultatai, kai  $m = 2$  (sudaryta autoriaus, remiantis Zitzler et al., 2001)

Galutinės bandymo reikšmės rodo, jog kai  $m = 2$ , SPEA2 algoritmas yra šiek tiek pranašesnis už NSGA-II algoritmą pagal reikšmių pasiskirstymą po visų galutinių paleidimų, tačiau kai  $m = 3$ , situacija apsiverčia. Šiuos rodiklius galime stebėti diagramose dešinėje pusėje (žr. 13 pav. ir 14 pav.).



**14 pav.** Algoritmų rezultatai, kai  $m = 3$  (sudaryta autoriaus, remiantis Zitzler et al., 2001)

Apibendrinant gautus rezultatus, darytina išvada, kad SPEA2 algoritmo geriausi rezultatai, kai  $m = 2$  plačiausių reikšmių pasiskirstymo atžvilgiu, tačiau kai  $m = 3$ , geriausi rezultatai pasireiškia naudojant NSGA-II algoritmą. Vis dėlto, NSGA-II laiko požiūriu nepasižymi našiausiais vykdymo rodikliais.

Literatūroje minima ir daugiau atvejų, kokiomis funkcijomis, be Sferos modelio, bandoma atlikti testus, tačiau įprastai NSGA-II ir SPEA2 algoritmai leidžia pasiekti apylygius rezultatus, kurie yra geresni naudojant SPEA ir PESA algoritmus (Digalakis 2002).

## 1.6. Apibendrinimas ir rezultatai

Nuosekliai išnagrinėjus verslo procesų, modeliavimo, simuliacijos sąvokas, galimybes ir praktikoje taikomus metodus, genetinius algoritmus ir jų taikymą, toliau galima plėsti paieškas iškeltoms problemoms spręsti. Jau atlikti ir aprašyti tyrimai padeda suprasti dabartinę situaciją ir progresą verslo procesų modeliavimo ir simuliacijos srityje, tad išanalizuota literatūra galima remtis kaip pagrindu, taikant jau naudotus algoritmus ir ieškant sprendimų pagerinti jau pritaikytus metodus. Tai, jog kai kurie autoriai analizuoja simuliacijos ir optimizacijos metodų tarpusavio integracijos galimybes ir ieško būdų, kaip tai atlikti, rodo, jog problema iš tikrųjų egzistuoja, yra sprendžiama bei nagrinėjama.

## 2. TIKSLO SIEKIANTIEMS VERSLO PROCESAMS TAIKOMO SIMULIACIJOS METODO ANALIZĖ

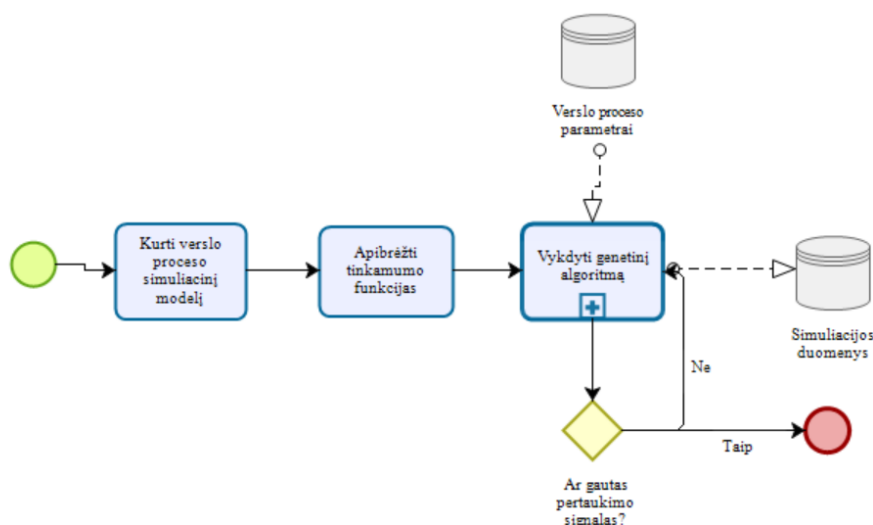
### 2.1. Verslo apibrėžtų tikslų siekimas

Išanalizavus literatūros šaltinius ir genetinių algoritmų pritaikymo galimybes, toliau galima nuosekliai ieškoti iškeltos problemos sprendimo būdų ir apibrėžto tikslo realizacijos. Kaip žinoma, visų pelno siekiančių verslų pagrindinis tikslas – didinti pelną, iš kurio vedami antriniai tikslai: didinti klientų pasitenkinimą teikiama paslauga ar preke, kelti savo įmonės reputaciją, mažinti išlaidas, paslaugos ar produkto kūrimo trukmę ir savikainą, kitaip tobulinti verslo procesą siekiant sukurti palankesnes sąlygas didesniai pelnui ir pridėtinei vertei generuoti (Laguna et al. 2019).

Tikslo siekiančių verslo procesų simuliacija sudaro galimybes detaliau analizuoti resursų pasiskirstymą dar prieš įvykdant procesus. Dėl to rinkoje reikalingi našūs simuliacijos ir jų optimizavimo būdai, lengvinantys verslo procesų resursų valdymą ir artinantys įmonę prie savo siekiamų tikslų. Tačiau rezultatams vertinti ne visada tinka paprasti metodai – jei įmonė turi kelis indikatorius savo našumui pasverti (angl. *Key Performance Indicator*, KPI) ir verslo procesas yra sudėtingas tiesioginei analizei, realu, kad, pritaikant tam tikrą sprendimą, turėtų būti atsižvelgiama į kelių sėkmės indikatorių rodiklius nustatytu santykiu (Kanri 2016) ir simuliuojama naudojantis sudėtinga schema. Čia gali padėti genetiniai algoritmai, kurie sėkmingai taikomi gamybos, matematikos, inžinerijos srityse. Genetinių algoritmų naudojimas ir tyrimai rodo, jog jie galėtų būti naudojami ir verslo procesų schemoms optimizuoti.

### 2.2. Tikslo siekiančių verslo procesų simuliacijos modelis

Kadangi šio darbo siekis – praplėsti tikslo siekiančių verslo procesų simuliacijos būdus ir ieškoti našiausio sprendimo tam atlikti, toliau pateikiamas siūlomas metodo modelis (žr. 15 pav.).



15 pav. Taikomo metodo modelis

Bandant siekti apibrėžtų tam tikros įmonės ar organizacijos tikslų, reikia turėti sukurta verslo proceso modelį, nusakantį vykdomą veiklą. Tinkamai apibrėžus kertinius procesus, darančius įtaką našesnių rezultatų siekimui, modeliai suteikia galimybę simuliuoti galimas baigtis dar prieš pradėdant jas vykdyti, taikant skirtingus parametrus.

Pasirinktos srities verslo procesas turi būti analizuojamas ir detaliam specifikuojamas, bendraujant su tiesiogiai veiklą vykdančiais žmonėmis. Analitiko vaidmenį atliekantis asmuo turėtų gilintis į pagrindines verslo užduotis, joje naudojamas konstantas ir kintančius parametrus, kurie galėtų būti optimizuojami. Svarbu tinkamai identifikuoti proceso sritis, kurios daro tiesioginę įtaką rezultatams, siekiant apibrėžto tikslo. Analizės metu turėtų būti iškelti ne tik svarbiausi reikalavimai ir surinkti duomenys, bet ir atvaizduojama verslo proceso diagrama, kuria remiantis būtų įmanoma patvirtinti tinkamą tarpusavio supratimą ir vienareikšmiškumą. Vizuali informacija ir aiškiai dokumentuoti reikalavimai turėtų būti patvirtinami tiek verslo atstovo, tiek analitiko.

Struktūrizavus modelį ir pateikus jį viena iš modeliavimo kalbų (šiuo atveju naudojant BPMN), verslo procesą galima simuliuoti pateikiant modeliui skirtingus kintančius parametrus ir stebint, kaip kiekvienos simuliacijos metu pagal tą patį verslo procesą rezultatai kinta priklausomai nuo skirtingų parametrų (žr. 15 pav.).

Sukurtam modeliui turėtų būti apibrėžiamos tinkamumo funkcijos – funkcijos, pagal kurių reikšmės bus skaičiuojamas rezultatų tinkamumas problemai spręsti. Matematiškai apibrėžus rodiklį, kuris padeda spręsti, ar, keičiant parametrus ir vykdant simuliacijas, artėjama tikslo link ir algoritmo naudojimas įgauna prasmę. Turint tinkamumo funkcijas, taip pat ir apibrėžtis, kada tinkamumo funkcijos bus tenkinamos, galima pradėti vykdyti genetinį algoritmą (Mallawaarachchi 2017).

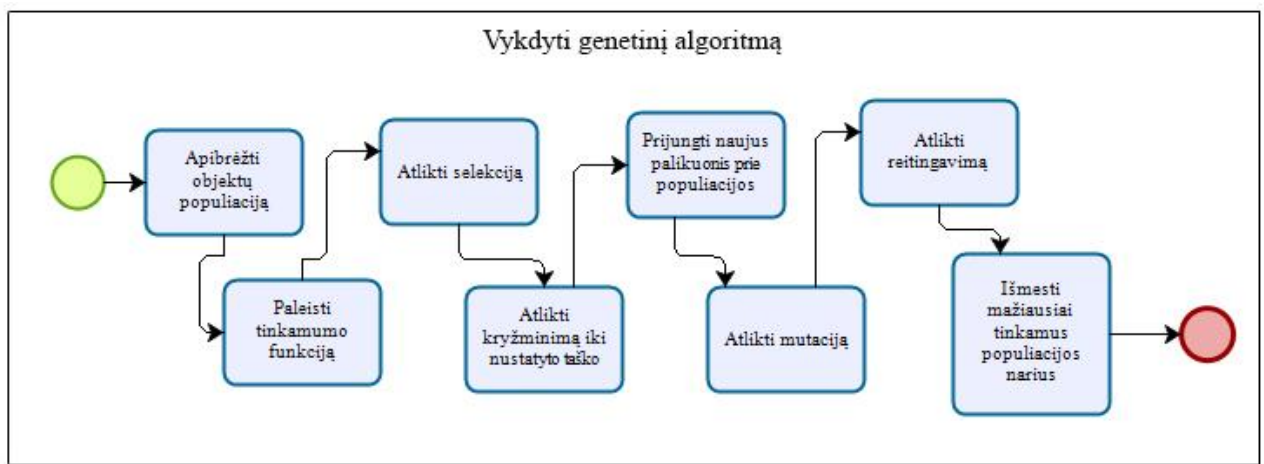
Algoritmas naudoja skirtingus pateikiamus verslo proceso parametrus ir kiekviena sugeneruota karta yra kaupiama bei išsaugoma kaip įvykusios simuliacijos duomenys, o genetinis algoritmas vykdomas tol, kol gaunamas pertraukimo signalas. Pertraukimo sąlyga turi būti numatyta dar prieš pradėdant skaičiavimus – svarbu nustatyti kriterijų, kada algoritmas nustos veikti. Dažniausiai procesas nutraukiamas maksimaliai priartėjus prie optimalaus sprendimo ir kai algoritmas pradeda vesti tarpusavyje labai panašias kartas arba po tam tikro vykdymų skaičiaus (žr. 15 pav.).

### **2.2.1. Genetinio algoritmo vykdymas**

Pavaizduotame taikomo metodo modelyje (žr. 15 pav.) procesas „Vykdyti genetinį algoritmą“ yra sudėtinis ir skaidomas į smulkesnius vidinius procesus (angl. *sub-task*) (žr. 16 pav.). Ši užduočių eiga taikoma remiantis išnagrinėta literatūra ir analizuotais pavyzdžiais. Pirmasis žingsnis – objektų populiacijos apibrėžimas, iš kurios bus vedamos kitos kartos. Naudojant gaunamus verslo sferos parametrus ir pritaikant juos suformuotam verslo proceso simuliaciniam modeliui, tikrinama



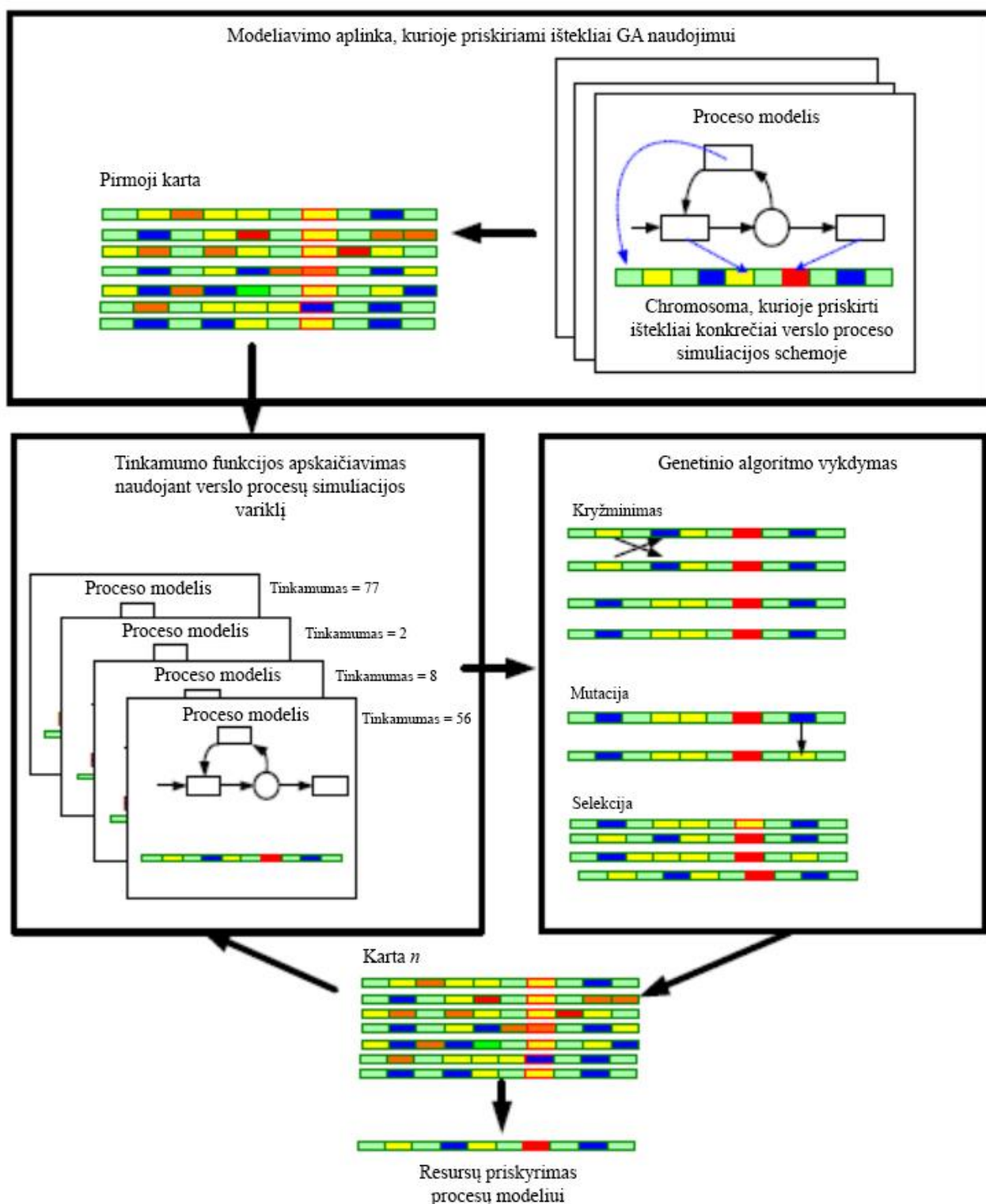
apibrėžtoji tinkamumo funkcija. Ją naudojant grąžinti rezultatai atrenkami selekcijos metu – iš esamos kartos išrenkamos chromosomos, kurios bus naudojamos naujosios kartos generacijai. Toliau atrinktose chromosomose įvykdomas kryžminimas iki nustatytojo kryžminimo taško – parenkamos dvi tėvinės chromosomos ir, imant jų genus bei iki kryžminio taško juos keičiant, sudaromi nauji palikuonys, kurie bus prijungiami prie naujosios populiacijos. Generacijos įvairovei užtikrinti vykdoma genų mutacija – atsitiktinis genų reikšmės pakeitimas naujai formuojamoje kartoje. Šiems procesams įvykus, toliau populiacija gali būti reitinguojama nuo tinkamiausio generacijos nario iki mažiausiai tinkamo, kurie išmetami iš naujosios kartos pagal nustatytą maksimalų populiacijos skaičių (žr. 16 pav.).



**16 pav.** Submodelis „Vykdėti genetinį algortimą“

Vienas iš pavyzdžių, kaip galėtų būti sudaromos populiacijos chromosomos, pateikiamas Yain Whar Si šaltinyje, kuriame taip pat prijungiami ir Petri tinklai (Si et al. 2018). Turint proceso modelį, kiekvienas chromosomoje esantis genas gali reprezentuoti priskirtus išteklius procese. Iš tokių chromosomų sugeneruojama pirmoji karta, kuri toliau duodama tinkamumo funkcijai vertinti, ir sekami tolesni anksčiau minėti procesai – selekcija, kryžminimas, mutacija. Kartos yra generuojamos, kol priartėjama prie našiausio sprendimo ir grąžinama tinkamiausia chromosoma iš paskutinės kartos su jai priskirtais ištekliais (žr. 17 pav.). Visas simuliacijos procesas kartojamas taikant tą patį modelį, tačiau su skirtingais parametrais, kol gaunamas tenkinantis sprendinys.

Populiacijos dydis, resursų kiekis, kartų dydis arba pertraukimo kriterijus, selekcijos ir eliminacijos reitingai gali būti keičiami. Algoritmas nustato, kad priartėta prie geriausio sprendimo, lyginant tinkamiausią chromosomą iš paskutinės ir naujai sugeneruotos kartos, ir pagal nustatytą tinkamumo slenkstį (angl. *threshold*) tikrinama, ar algoritmas grąžina panašias reikšmes.



**17 pav.** Resursų paieška naudojant genetinį algoritmą (sudaryta autoriaus, remiantis Si et al. 2018)

Kadangi visoms pelno siekiančioms įmonėms aktualu gauti kuo daugiau pelno ir naudos įgyvendinant įmonės tikslus, vienas iš svarbių siekių, darantis tiesioginę įtaką pelno didinimui, yra veiklos išlaidų minimizavimas. Tai padeda įmonei išvengti netinkamo pinigų paskirstymo ir optimizuoti nereikalingą pinigų nutekėjimą ten, kur galima sutaupyti, o tai atneša daugiau pelno.

Tokio siekio tinkamumo funkcija galėtų būti  $f(P) = \sum u_i x_i \rightarrow \min$ , čia parametras  $u_i$  nurodo veiklos  $a_i$  vykdymo kainą, o parametras  $x_i$  – ar veikla  $a_i$  įtraukta į vykdomą verslo procesą. Šios sandaugos sumos reikšmė turėtų būti minimizuojama (Vergidis et al. 2006).

Verslo modeliai, verčiami matematinėmis formulėmis, gali susidaryti iš daugybės diskrečių binarinių kintamųjų, o tai fragmentuoja paieškos erdvę ir sunkina procesą. Ir nors verslo modeliai gali būti lengvai suprantami, vizualizuojami ir iliustruojami, formuojant juos formaliais matematiniais apibrėžimais modeliai tampa griežtai ribojami. Tai gali kelti sunkumų ieškant optimalių sprendimų rinkinių. Dėl šios priežasties tikslinga naudoti daugiafunkcinius genetinius algoritmus.

### **2.3. Metodo apibendrinimas**

Pelno siekiančių įmonių tikslai tarpusavyje gali turėti nemažai panašumų: siekis minimizuoti išlaidas, savikainą, didinti pelną, taupyti laiką. Remiantis šiame darbe išnagrinėtu metodu, verslo procesų modeliams ir jų tinkamumo funkcijoms vykdyti bus taikomi genetiniai algoritmai, apibrėžiant pradinę populiaciją su išteklių chromosomomis ir toliau taikant tinkamumo funkciją. Po jos atliekama chromosomų selekcija, kurių tėvinės poros bus kryžminamos iki kryžminimo taško ir taip išvedami naujos populiacijos palikuonys. Generacijos įvairovei užtikrinti vykdoma genų mutacija, o po jos karta sureitinguojama – mažiausiai tinkami nariai eliminuojami. Visa tai kartojama iki pertraukio – iš anksto nustatyto slenksčio, kurį pasiekus tinkamiausi pagal tinkamumo funkciją nustatyti nariai naujose populiacijose nesikeičia.

### 3. TIKSLO SIEKIANTIEMS VERSLO PROCESAMS TAIKOMO SIMULIACIJOS METODO REALIZACIJOS APRAŠAS

#### 3.1. Priemonės simuliacijai vykdyti

Pasiūlyto metodo realizacija vykdoma naudojantis *Microsoft* kompanijos produktu – integruota kūrimo aplinka (angl. *Integrated Development Environment*) *Visual Studio Community 2017*. Naudojant *C#* programavimo kalbą bei *.NET Core 2.1* versijos karkasą, prijungiamos kelios bibliotekos: *SimSharp* (3.1.1 versija), *GeneticSharp* (2.5.1 versija), *CsvHelper* (12.1.2 versija).

*SimSharp* naudojama simuliacinio modelio kūrimui. Ši biblioteka sukurta *SimPy* sąvokų pagrindu, kuri pritaikyta naudojantis *Python* programavimo kalba, tačiau *SimSharp* palaikoma naudojantis *.NET* karkasais. Tai – procesais grįstas diskrečiųjų įvykių simuliacijos karkasas, leidžiantis vykdyti simuliacijas pasirinktu greičiu arba realiuoju laiku.

*GeneticSharp* suteikia galimybę lengviau vykdyti sekančius pasiūlyto metodo žingsnius. Tai biblioteka, praplečianti genetinių algoritmų panaudojimo galimybes. Naudojantis *GeneticSharp*, galima pritaikyti metodus reikalingiems žingsniams realizuoti – chromosomų, populiacijų ir generacijų kūrimui, selekcijai, kryžminimui, mutacijai, , pertraukimui.

Pridėjus *CsvHelper* biblioteką, gautus simuliacijos rezultatus galima įrašyti į CSV plėtinio failus, o tai įgalina toliau naudoti duomenis analizei bei pasirinkto metodo efektyvumo nustatymui.

Šie produktai buvo pasirinkti stengiantis teikti prioritetą genetinio algoritmo realizacijai, o visos paminėtosios priemonės atitinka įgyvendinimo lūkesčius. Reikalingos bibliotekos priskiriamos *C#* programavimo kalbai, o *Microsoft* produkcija yra plačiai naudojama, išvystyta, teikia nemokamas nekomercines versijas bei pakankamą kiekį pagalbinės medžiagos kuriant projektus.

#### 3.2. Simuliacijos elementai

*C#* kodo pagalba aplinkoje sukurtos simuliacijos vykdymui reikalingos klasės. Aptariant svarbiausias (žr. 18 pav.), *HelpDeskProcess* klasė yra svarbiausia charakterizuojant pasirinktą verslo sritį. Joje generuojami įmonėje dirbančių ir sistemos gedimus patiriančių vartotojų sąrašai (*Callers* klasė) pagal nustatytą dažnį ir, atsižvelgiant į klasei teikiamus parametrus, vykdomas trijų lygių IT pagalbos skambučių centro simuliacijos procesas (žr. 2 lentelė).

2 lentelė. *HelpDeskProcess* klasė

Klasė	Klasės elementas	Paskirtis
<i>HelpDeskProcess</i>	<i>callers</i>	IT pagalbos centrui skambinusiųjų sąrašas.
	<i>MinPatience, MaxPatience</i>	Vartotojų laukimo linijoje laukimo režiai.
	<i>MinTime, MaxTime</i>	Incidentų dažnis.
	<i>SimTime</i>	Simuliacijos trukmė.

	<i>GenerateCallers</i>	Metodas, skirtas skambinančių vartotojų generavimui.
	<i>HandleCaller,</i> <i>HandleSecondLevel,</i> <i>HandleThirdLevel</i>	Metodai, skirti kiekvieno pagalbos skambučių centro lygio procesui vykdyti (atitinkamai L1, L2, L3).
	<i>Simulate</i>	Metodas simuliacijos vykdymui.

*Callers* klasė suteikia galimybę saugoti simuliacijos statistikai reikalingus duomenis apie kiekvieną skambinusių. *SimulationStatistics* pagalba suskaičiuojami reikalingi rodikliai ir perduodami klasei *SimulationResult*. Šie rezultatai svarbūs vertinant IT pagalbos skambučių centro veiklos našumą (žr. 3 lentelė)

**3 lentelė.** *SimulationResult* klasė

Klasė	Klasės elementas	Paskirtis
<i>SimulationResult</i>	<i>AverageLevel</i>	Gražina reikšmę, kuri nurodo, kokiam IT pagalbos skambučių centro lygyje (L1, L2 ar L3) vidutiniškai baigiamas procesas visų skambinusių atžvilgiu.
	<i>AverageSatisfaction</i>	Vidutinis klientų atsiliepimo balas.
	<i>AverageTalkingTime</i>	Vidutinė skambučio trukmė.
	<i>AverageWaitingTime</i>	Vidutinė kliento laukimo linijoje trukmė.
	<i>FirstLevelResourceUsability,</i> <i>SecondLevelResourceUsability,</i> <i>ThirdLevelResourceUsability</i>	Atitinkamai: L1, L2, L3 lygio darbuotojų užimtumas.
	<i>NumberOfCalls</i>	Visų inicijuotų skambučių kiekis.
	<i>NumberOfEscalations</i>	Klientų skundų kiekis.
	<i>TotalTimeInFirstLevel,</i> <i>TotalTimeInSecondLevel,</i> <i>TotalTimeInThirdLevel</i>	Atitinkamai: L1, L2, L3 lygiuose priimtų skambučių trukmės sumos.

Genetinio algoritmo veikimą aprašo *GeneticAlgorithmForHelpDesk* klasė. Apdorotus duomenis galime panaudoti priimant sprendimą, kaip bus optimizuotas verslo procesas, parenkant tinkamą kiekį resursų, kuris atspindės *GAResults* klasės grąžinamuose rezultatuose.

**4 lentelė.** *GAResults* klasė

Klasė	Klasės elementas	Paskirtis
<i>GAResults</i>	<i>FirstLevelWorkers,</i> <i>SecondLevelWorkers,</i> <i>ThirdLevelWorkers</i>	Atitinkamai: L1, L2, L3 lygiams priskirtas darbuotojų skaičius.
	<i>FitnessFunctionResult</i>	Tinkamumo funkcijos rezultatas.
	<i>Generation</i>	Genetinio algoritmo kartos skaičius.
	<i>Group</i>	Siūlomų resursų eilutė

Visų šių klasių funkcionalumas apjungiamas į vientisą procesą (žr. 18 pav.). Naudojantis tokia struktūra, pasirinkta aplinka gebės simuliuoti IT pagalbos skambučių centro veiklą. Pritaikant

parametrus, naudojamus versle, ir keičiant įeigos duomenis, šis elementų junginys generuos reikiamus duomenis tolesnei analizei.



18 pav. Metodo realizacijai naudojamos klasės

## 4. EKSPERIMENTINIS TYRIMAS

### 4.1. Verslo proceso analizė

Trečiame skyriuje išanalizavus pasiūlytam metodui reikalingus realizuoti produktus, modelio tinkamumas bus vertinamas jį vykdant. Pateikto modelio žingsniai bus pritaikyti realiam verslo procesui ir eksperimento metu bus bandoma įrodyti genetinio algoritmo tinkamumą tikslo siekiančių verslo procesų simuliacijoms.

Daugelio įmonių vykdomos veiklos procesai vis labiau kompiuterizuojami. Darbuotojams vis dažniau tenka dirbti su įvairia kompiuterine įranga bei jų programomis. Tai reiškia, kad visos įmonės priklausomai nuo verslo dydžio turi apgalvoti techninės ir programinės įrangos palaikymo strategiją.

Jau dešimtis metų plėtojamos diskusijos ir vykdomi tyrimai, koks turėtų būti optimalus įmonės, kuriai reikalinga ši paslauga, darbuotojų ir IT pagalbos specialistų santykis. 2008 metų *Robert Half Technology* atlikta apklausa parodė, jog optimalus santykis turėtų būti vidutiniškai 82:1, o realiose situacijose vidutinis santykis buvo 136:1 (Livy 2018). Bėgant metams, IT paslaugų centrų procesai yra nuolatos tobulinami ir ieškoma automatizuotų procesų bendrojo pobūdžio problemoms spręsti. Tačiau taip pat tobulėja ir verslo procesų automatika, kompiuterizuotų užduočių skaičius, o tai lemia dar didesnę IT paslaugų poreikį ir platesnį jų spektrą skirtingų rūšių problemoms spręsti. Remiantis to paties šaltinio (Robert Half Technology) duomenimis, 2017 m. optimalusis santykis pakito iki 70:1, o jei įmonė dirba su kelių rūšių operacinėmis sistemomis arba neturi bendrų standartų techninei įrangai, santykis gali kristi iki maždaug 45:1 (Half 2017b).

IT pagalbos resursų skaičių lemia ir kiti veiksniai. Manoma, jog 75 žmonių įmonė, turinti padalinių trijose skirtingose vietose (tarus, jog biurai yra skirtinguose žemynuose), yra labiau decentralizuota nei 255 žmonių komanda, dirbanti tame pačiame pastate (Half 2017b). Tai gali lemti, kad reikia daugiau specialistų dėl laiko juostų skirtumo ir besiskiriančių pamainų, procesų nevienodumo ar kitų veiksnių, kurie yra tokio biurų pasklidimo priežastis.

Kitas rodiklis, lemiantis resursų santykį, yra galimų problemų sudėtingumo įvertis. Jei įmonės programinė ar techninė su IT susijusi veikla yra labai specifinė ir jai būdinga didelė galimų problemų įvairovė, tai gali kelti skirtingų lygių pagalbos centro poreikį. Tokiose įmonėse pirmojo lygio pagalbos centras dažniausiai sprendžia bendrojo pobūdžio smulkias problemas, o aukštesnis lygis (prireikus keli lygiai) bando padėti esant specifinėms, daugiau specialisto žinių reikalaujančioms problemoms. (Half 20017).

Skambučių centrai dažniausiai naudoja pamatuojamus indikatorius našumui vertinti (angl. *Key Performance Indicator*, KPI ). Vienas iš jų yra kiekvienos problemos fiksavimas atskiru įrašu –

„biletu“ (angl. *ticket*) – bei jų skaičiaus valdymas. Apibrėžiant kiekvieną problemą skirtingu įrašu, lengviau sekti jo progresą, statusą, sprendimo laiką, sudėtingumą. IT pagalbos darbuotojų skaičiaus poreikis yra tiesiogiai susijęs su vidutiniu gaunamu „biletų“ skaičiumi sistemoje (jas galima fiksuoti tam pritaikytose platformose, pavyzdžiui *ServiceNow* siūlomuose produktuose (*ServiceNow* 2019)). Tai gali būti vienas iš pagrindinių rodiklių, priimant sprendimą, kiek resursų įmonė turėtų skirti įmonės kompiuterizuotai veiklai optimizuoti. Kitas rodiklis – vidutinis laikas, praleistas sprendžiant vieno įrašo problemą (Livy 2018).

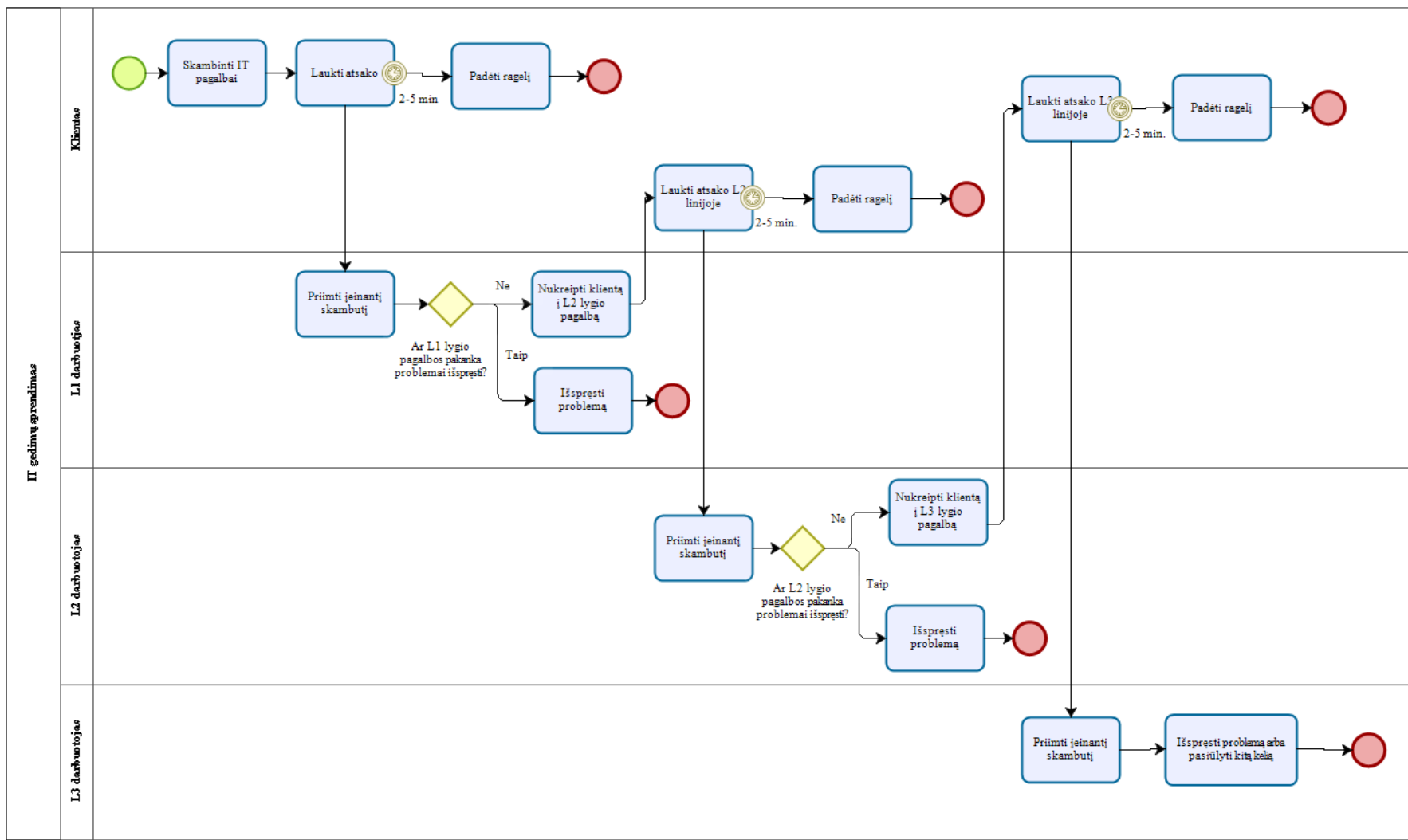
Kadangi IT pagalbos centro darbo optimizavimo tema yra plačiai diskutuojama, yra daug skirtingų tyrimų ir problema aktuali pasauliniu mastu, šiame darbe analizuosime realią IT pagalbos skambučių centro situaciją. Toliau pateikiamas išnagrinėtas veiklos procesas ir jo kintantys bei nekintantys parametrai.

Analizuojant verslo procesą, pavaizduotą diagramoje, pastebima, kad nagrinėjamoje įmonėje didelė dalis darbuotojo veiklos atliekama kompiuteriu. Jei vartotojas susiduria su sistemos gedimu ar kitu sunkumu, trukdančiu tęsti su darbo užduotimi susijusią veiklą, darbuotojas gali kreiptis į IT pagalbos centrą (procesas vaizduojamas 19 pav.). Vartotojas skambina į pagalbos liniją ir laukia atsako iš IT pagalbos darbuotojo. Jei žmogus, susidūręs su sistemos gedimu, nesulaukia atsako linijoje, jis padeda ragelį ir pateikia oficialų skundą, jog IT aptarnavimo centre pagalba jam nebuvo suteikta.

Jeigu tuo metu yra pakankamai resursų ir laisvas IT darbuotojas atsako į kliento skambutį, klientas turi paaiškinti savo problemą. Pirmojo lygio (L1) IT aptarnavimo darbuotojas skambučio metu sistemoje užfiksuoja gedimą ir sukuria naują incidento įrašą. Užfiksavęs problemą, IT darbuotojas turi nuspręsti, ar jis yra pajėgus padėti konkrečioje situacijoje. Jeigu taip, problema išsprendžiama ir darbuotojui patvirtinus, jog šis užfiksuotas gedimas yra pašalintas, darbas gali būti sėkmingai tęsiamas iki darbo dienos pabaigos.

Jei L1 lygio IT darbuotojas negali padėti vartotojui išspręsti kilusios problemos, jis privalo nukreipti įmonės darbuotoją į antro lygio (L2) pagalbos liniją. Su gedimu susidūręs darbuotojas laukia, kol L2 linijoje į jo skambutį bus atsakyta, ir turi tą pačią teisę padėti ragelį bei reikalauti iš centro problemos sprendimo. Jei L2 lygio specialistas atsako į skambutį, darbuotojas turi nurodyti problemą, kuri jau buvo užfiksuota L1 lygio darbuotojo sistemoje, ir pateikti papildomą informaciją, reikalingą problemai spręsti. Problema išsprendžiama, jei L2 lygyje pavyksta gauti atsakymą, tačiau jei pasirodo, jog gedimui pašalinti reikia dar daugiau specifinių žinių, klientas nukreipiamas į L3 liniją.





19 pav. IT pagalbos centro skambučių priėmimo procesas

Jei klientas nepadeda telefono ragelio ir laukia paslaugos iš teikėjui, L3 specialistui atsakius, gedimas turi būti apibūdinamas vartotojo su dar detalesniais aspektais. Šiame lygyje L3 lygio specialistas išsprendžia problemą arba privalo pateikti kitą sprendimą darbuotojui, jog šis galėtų tęsti savo darbinę veiklą arba imtis tam tikrų veiksmų.

Išanalizavus IT gedimo sprendimo telefonu procesą, toliau galima kurti simuliacinį modelį ir pritaikyti parametrus, kurie lems skirtingas proceso baigtis.

## 4.2. Verslo proceso simuliacija

Šio verslo proceso simuliacija bus atliekama darant prielaidą, kad kompanijos darbuotojai yra pasiskirstę apylygiai skirtingose laiko zonose ir tuo pačiu metu visą parą dirba panašus darbuotojų skaičius. Įmonė turi 140 000 vartotojų, turinčių asmeniškai priskirtą techninę bei programinę įrangą (kurios gali skirtis pagal darbuotojo lokaciją) ir nuolatos jomis besinaudojančių. Tarp dešimties ir trisdešimties sekundžių įmonėje vieno iš darbuotojų sistemą sutrikdo gedimas ir vartotojas skambina IT pagalbos darbuotojui. Minimalus vidutinis laikas, kurį klientas yra pasiryžęs laukti, kad jo skambutis būtų priimtas, yra 2 minutės. Maksimali kliento kantrybės riba – 5 minutės. Jei į skambutį atsakoma, vidutinė incidento sprendimo trukmė L1 IT pagalbos lygyje yra tarp 1 ir 5 minučių. L2 – nuo 3 iki 10 minučių, o L3 – nuo 3 iki 30 minučių. Tikimybė, kad, sistemos vartotojui susisiekus su pagalbos darbuotoju, jo problemai išspręsti nepakaks pirmojo lygio pagalbos, yra 25 proc. Taip pat yra tikimybė, jog L2 lygio taip pat neužteks galutiniam sprendimui užtikrinti, ir ji lygi 20 proc (žr. 5 lentelė).

**5 lentelė.** Simuliacijos parametrai

Parametras	Reikšmė
Vartotojų sistemoje	140 000
Naujas incidentas įvyksta kas X sekundžių	10-30 sek.
Minimali kantrybės riba laukiant	2 min.
Maksimali kantrybės riba laukiant	5 min.
Vieno incidento sprendimo trukmė L1 lygyje	1-5 min.
Vieno incidento sprendimo trukmė L2 lygyje	3-10 min.
Vieno incidento sprendimo trukmė L3 lygyje	3-30 min.
Tikimybė, kad pranešimą apie incidentą L1 lygis perduos L2 lygiui	0,25
Tikimybė, kad pranešimą apie incidentą L2 lygis perduos L3 lygiui	0,2

Simuliacija bus atliekama aštuonias valandas ir bus stebima, kiek resursų naudojama kiekvieno lygio darbuotojų atžvilgiu. Proceso našumui vertinti bus matuojami skirtingi rodikliai, tiesiogiai nusakantys resursų optimalumą:

- **vidutinė laukimo trukmė** – nurodo viso laukto laiko linijoje sumą, dalijamą iš skambinusių asmenų skaičiaus;
- **skundų skaičius** – apibrėžia, kiek žmonių, skambinusių IT pagalbai, nesulaukė atsako ir padėjo ragelį;
- **skundų procentas** – nurodo, kiek procentų skambinusių asmenų kreipėsi į IT skambučių centrą;
- **klientų pasitenkinimas** – įvertis pagal penkiabalę sistemą, nurodantis klientų pasitenkinimo lygį pagal laukimo trukmę (žr. 6 lentelė);

Klientų atsiliepimams daro įtaką jų trukmė linijoje. Naudojantis lentele (žr. 6 lentelė), atsiliepimo balai priskiriami pagal penkiabalę sistemą remiantis atitinkamu sekundžių skaičiumi.

**6 lentelė.** Klientų atsiliepimo balai pagal laiką

Balai	Sekundės
5	$0 \leq$ ir $<30$
4	$30 \leq$ ir $<90$
3	$90 \leq$ ir $<180$
2	$180 \leq$ ir $<240$
1	$240 \leq$ ir $<300$
0	Visais kitais atvejais

Toliau simuliaciniam modeliui bus pritaikomas genetinis algoritmas ir juo remiantis bus keičiami parametrai, lemiantys resursų naudojimą kiekvienam IT pagalbos linijos lygiui.

### 4.3. Genetinio algoritmo pritaikymas

Sukurtajam simuliaciniam modeliui toliau pritaikomas genetinio algoritmo veikimas. Algoritmas ieškos tinkamiausių parametų pagal apibrėžtą tinkamumo funkciją – bus tikrinami L1, L2 ir L3 lygių resursų dydžiai, ieškant kuo optimalesnio rezultato ir stengiantis grąžinti tinkamiausią šių trijų lygių darbuotojų kombinaciją.

Viena iš svarbiausių genetinio algoritmo dalių – tinkamumo funkcija, pagal kurią bus skaičiuojami rezultatai. Dėl to labai svarbu identifikuoti svarbiausius rodiklius procese, kurie turėtų daugiausiai įtakos vertinant proceso našumą.

Eksperimento eigoje bus naudojamas standartinis genetinio algoritmo veikimas – tinkamiausios chromosomos paieška pagal tinkamumo funkciją, vykdant populiacijų generavimą, taikant kryžminimo procesą bei mutaciją.

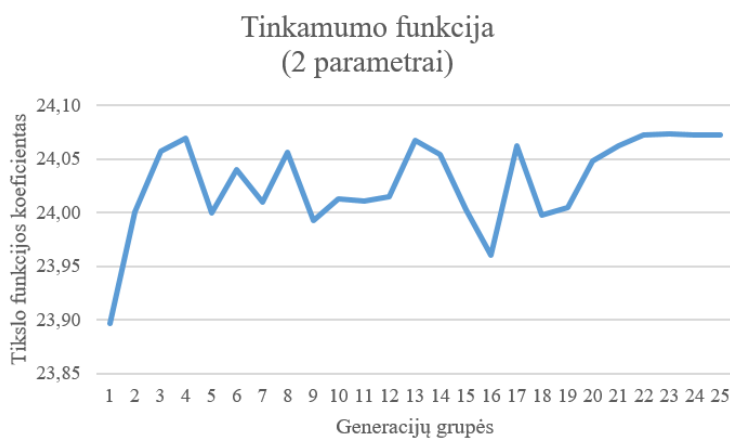
Parametrus, kurie naudojami tinkamumo funkcijose, bus suteikiami skirtingi svoriai pagal jų daromą įtaką rezultatams. Bus skaičiuojamas koeficientas, kurio maksimali reikšmė turėtų artėti link 50. Visų parametrų svoris toliau vykdomose eksperimentuose bus nustatomas pagal maksimaliai naudingas verslui parametrų ribas, bandant gauti maksimaliausio koeficiento vertę ir reguliuojant jų svorį pagal eksperimento keliu daromas išvadas. Kiekvienam atvejui bus kuriama 4000 generacijų ir skaidant jas į 25 grupes, stebint kaip kinta tinkamumo funkcijos koeficientas ir visų trijų lygių siūlomų resursų skaičius.

Pirmajam bandymui pasirinkta tinkamumo funkcija, kurią sudaro du parametrai. Jiems suteikiamas vienodas svoris skaičiuojant koeficientą – vidutinis klientų laukimo laikas linijoje (siekiant minimizuoti reikšmę) ir vidutinis klientų pasitenkinimo lygis (siekiant maksimizuoti reikšmę).

$$f(x_1, x_2) = \frac{a}{x_1} + b * x_2,$$

*kur*  $x_1$  - vidutinis laukimo laikas;  
 $x_2$  - vidutinis pasitenkinimo lygis;  
 $a, b$  - parametrui suteiktas svoris.

Darant prielaidą, jog palankiausias kompanijai vidutinis laukimo laikas galėtų būti 1 sek., o pasitenkinimo lygis maksimalus – 5, koeficientas artės link maksimalios vertės – 50. Svoriiui  $a$  suteikiama reikšmė 25;  $b$  – 5. Pritaikius šią tinkamumo funkciją, galime stebėti, jog tinkamumo funkcijos koeficientas labai nežymiai (tik šimtosiomis dalimis) kinta visų generacijų metu (žr. 20 pav.) ir neartėja link siekiamo geriausio rezultato.

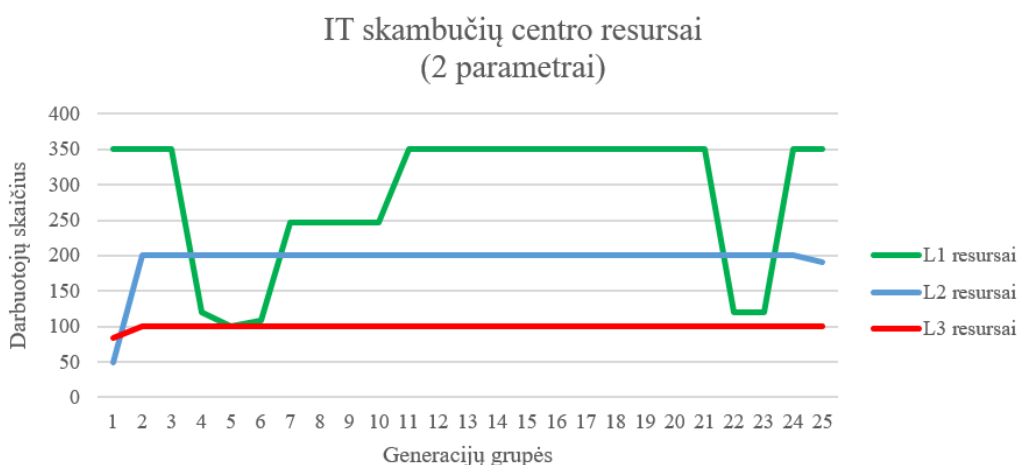


**20 pav.** Tinkamumo funkcijos koeficiento kitimas (2 parametrai)

Analizuojant resursų pasiskirstymą tokiu atveju, matome, jog labiausiai orientuojamasi į maksimalų resursų skaičių pirmame lygyje (350 darbuotojų), tačiau algoritmas pateikia ir kitų reikšmių (žr. 21 pav.).

Antrojo lygio resursų kitimas žymiai mažesnis – algoritmas pateikia minimalesnes reikšmes pirmosiose generacijose, tačiau vėliau taikomos tik maksimalios antrojo lygmens vertės (200 darbuotojų) ir rezultatas nekinta didėjant generacijų skaičiui (žr. 21 pav.).

Trečiojo lygmens resursų priskyrimo tendencija išlieka panaši kaip ir antrojo lygmens, kai taikoma tinkamumo funkcija su dvejais parametrais – pradžioje pasiūlęs mažesnę reikšmę, algoritmas išlaiko maksimalų (100 darbuotojų) trečiojo lygmens resursų skaičių (žr. 21 pav.).



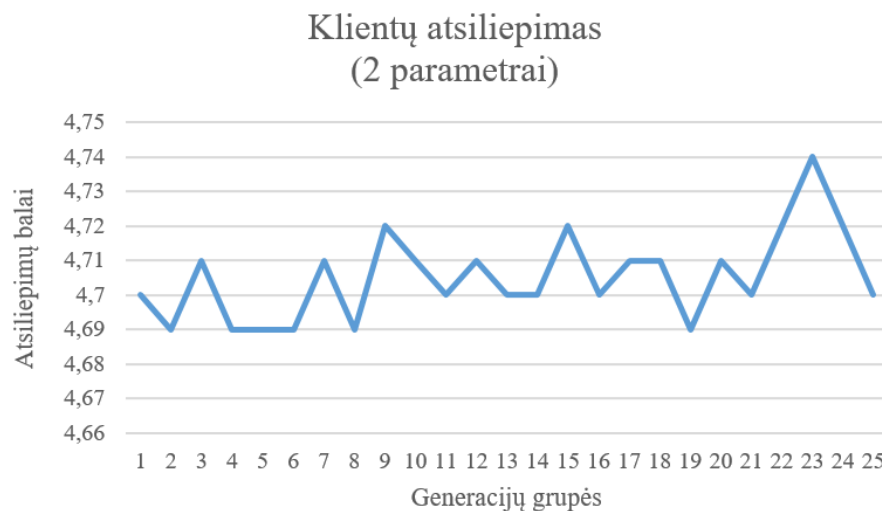
**21 pav.** IT skambučių centro resursai (2 parametrai)

Nagrinėjant, kokius rezultatus teikia genetinio algoritmo siūlomos reikšmės, matoma, jog tinkamumo funkcija atsižvelgia į optimalaus vidutinio laukimo laiko palaikymą. Kadangi, pagal pradiniu surinktus duomenis, klientas pasiryžęs laukti tarp 2 ir 5 minučių linijoje, vidutinis laukimo laikas tenkina šias ribas (žr. 22 pav.).



**22 pav.** Vidutinis laukimo laikas (2 parametrai)

Apžvelgiant klientų pasitenkinimo lygį, iš 5 maksimalių balų visų generacijų metu atsiliepimai svyravo ribose tarp 4,6 ir 4,8, tad rezultatai artėja prie optimalios verslui reikšmės (žr. 23 pav.).



**23 pav.** Klientų atsiliepimo balai (2 parametrai)

Tačiau, pritaikius tinkamumo funkciją su dvejais parametrais – vidutiniu laukimo laiku linijoje ir vidutiniu pasitenkinimo lygiu, galime daryti išvadą, jog šių parametrų nepakanka artėjant prie geriausių rezultatų. Bendrojo koeficiento reikšmė kinta tik šimtosiomis dalimis, kai bandoma varijuoti vien šiais parametrais, o siūlomas resursų skaičius, neatitinka lūkesčių – dauguma atvejų siūlomos maksimalios resursų reikšmės.

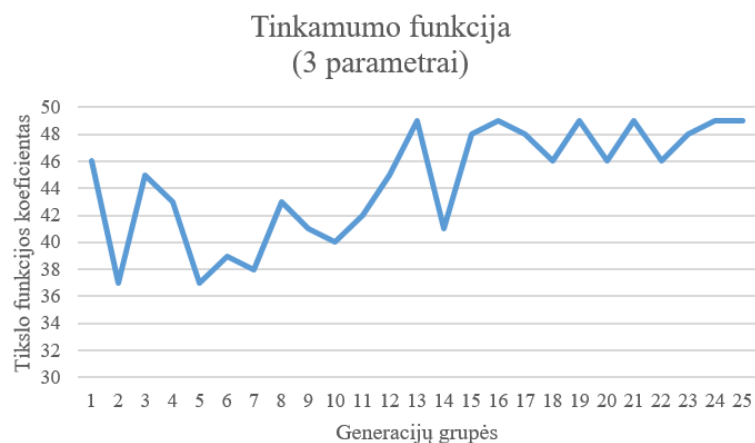
Kadangi šie rodikliai vis vien yra svarbūs verslo procese, tačiau nepakankami tikslesnėms išvadoms, prijungiamas dar vienas rodiklis, gaunamas simuliacijų metu – skundų procentas. Tai – klientų skundų ir visų skambučių santykis, kuris parodo, kiek iš visų gautųjų skambučių, IT pagalbos centras nepakėlė ragelio laiku. Tai yra svarbus rodiklis verslui, o laikantis teiginio, jog mūsų analizuojamą įmonę tenkinanti rodiklio riba yra 1 procentas (skaičiuojant pagal mūsų kompanijos dydį, skundų skaičius neturėtų būti didesnis nei 140), toliau formuojama nauja tinkamumo funkcija.

Šiam rodikliui suteikiamas didesnis svoris nei anksčiau pritaikytiems, tad pakoreguojamos konstantos funkcijoje, jog rezultatas artėtų prie koeficiento 50. Skundų procentas turi būti minimizuojamas.

$$f(x_1, x_2, x_3, x_4) = \frac{a}{x_1} + b * x_2 + \frac{x_3}{c * x_4}$$

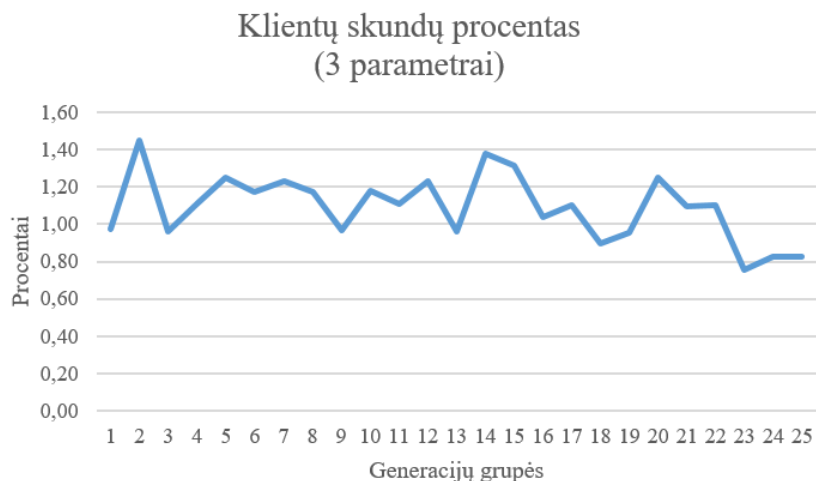
*kur*  $x_1$  - vidutinis laukimo laikas;  
 $x_2$  - vidutinis pasitenkinimo lygis;  
 $x_3$  - visi gauti skambučiai;  
 $x_4$  - visi skundai.  
 $a, b, c$  - parametrui suteiktas svoris.

Svoriui  $a$  suteikiama reikšmė 10;  $b$  – 2;  $c$  – 3,35. Pritaikę atnaujintą tinkamumo funkciją stebime, jog skirtingai nei prieš tai atliktame bandyme, tinkamumo funkcija ryškiau kinta simuliacijų metu (žr. 24 pav.).



**24 pav.** Tinkamumo funkcijos koeficiento kitimas (3 parametrai)

Papildžius tinkamumo funkciją parametru, kuris nurodo klientų skundų ir visų gautųjų IT pagalbos linijos skambučių santykį, matome, jog algoritmas ieško tenkinančių reikšmių, mažindamas šį procentą (žr. 25 pav.)

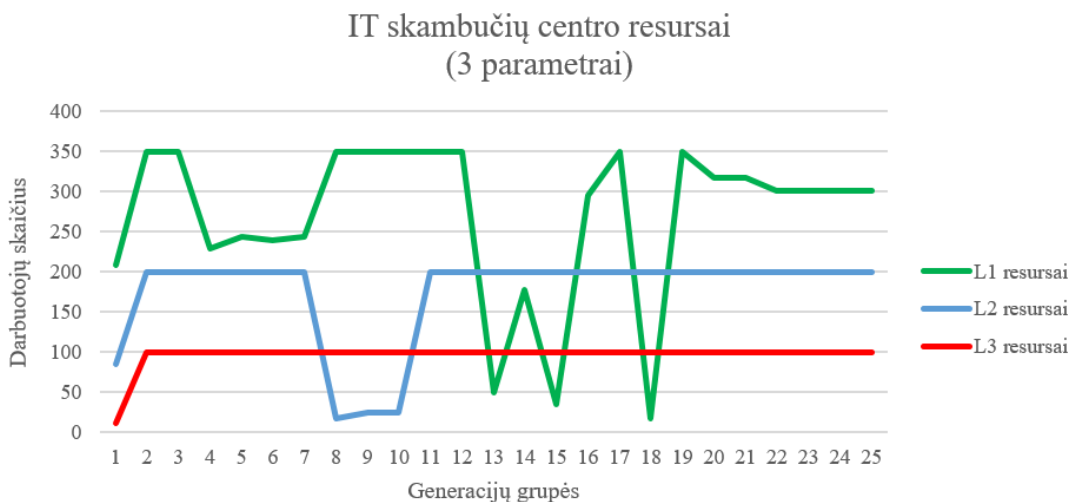


**25 pav.** Klientų skundų procentas (3 parametrai)

Tačiau, analizuojant atitinkamai siūlomus resursų dydžius, matoma, jog nors pirmojo lygmens resursų tam tikrose generacijose siūloma mažiau, algoritmas sugrįžta prie maksimalaus darbuotojų skaičiaus (žr. 26 pav.).

Antrojo lygmens resursų panaudos rezultatai gaunami pagal panašią tendenciją – nors algoritmas pasiūlo ir mažesnių reikšmių, koeficientui artėjant prie optimumo, resursų skaičius vis vien padidėja (žr. 26 pav.). Tai pat galime pastebėti, jog tam tikrose generacijose antrojo lygmens resursai viršija pirmojo lygmens. Sprendžiant iš to, jog pagal simuliacinį modelį yra tik 25 procentų tikimybė, jog skambinusiajam prireiks L2 pagalbos, antrojo lygmens resursai neturėtų viršyti pirmojo lygio darbuotojų skaičiaus. Tai rodo poreikį papildyti tinkamumo funkciją.

Peržvelgiant trečiojo lygmens resursų siūlymus (žr. 26 pav.) patvirtinama prieš tai keliama abejonė dėl resursų parinkimo skaičiaus – algoritmas siūlo beveik visais atvejais naudoti maksimalią L3 reikšmę, kuri tam tikrose generacijose viršija net ir L1 lygio resursų kiekį.



**26 pav.** IT skambučių centro resursai (3 parametrai)

Po šio bandymo galime padaryti išvadą, kad tinkamumo funkcija yra nepakankamai išbaigta. Turime tris svarbius rodiklius – vidutinį laukimo laiką, vidutinį pasitenkinimo lygį ir skundų procentą. Akivaizdu, jog genetinis algoritmas, bandydamas ieškoti geriausio koeficiento šių trijų rodiklių kombinacijai, maksimizuoja resursus, nes jam nėra nurodoma resursų panaudojamumo svarba. Tai reiškia, jog nepaisant to, jog bus labai daug laisvų ir neišnaudotų resursų, algoritmas pasiūlys maksimumą tam, jog būtų kuo mažesnis skundų bei laukimo laikas ir kuo didesnis pasitenkinimas.

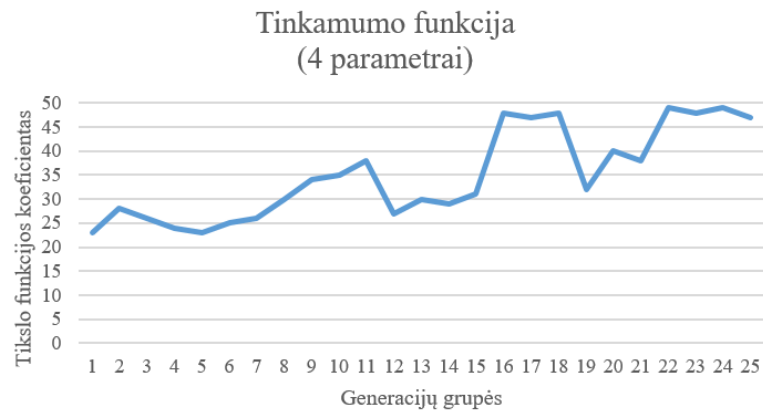
Kadangi mums reikalingi tikslesni duomenys, vedantys prie verslo tikslų siekimo, papildome tinkamumo funkciją ketvirtuoju rodikliu – resursų panaudojamumo procentais. Skaičiuojamas kiekvieno lygmens darbuotojų užimtumas ir šiems rodikliams suteikiamas didžiausias svoris tinkamumo funkcijoje, stengiantis pasiekti maksimalią darbuotojų užimtumo dalį.

$$f(x_1, x_2, x_3, x_4, x_5, x_6, x_7) = \frac{a}{x_1} + b * x_2 + \frac{x_3}{c * x_4} + d * (x_5 + x_6 + x_7)$$

*kur*  $x_1$  - vidutinis laukimo laikas;  
 $x_2$  - vidutinis pasitenkinimo lygis;  
 $x_3$  - visi gauti skambučiai;  
 $x_4$  - visi skundai;  
 $x_5$  - L1 lygio darbuotojų užimtumas;  
 $x_6$  - L2 lygio darbuotojų užimtumas;  
 $x_7$  - L3 lygio darbuotojų užimtumas.  
 $a, b, c, d$  - parametrui suteiktas svoris.

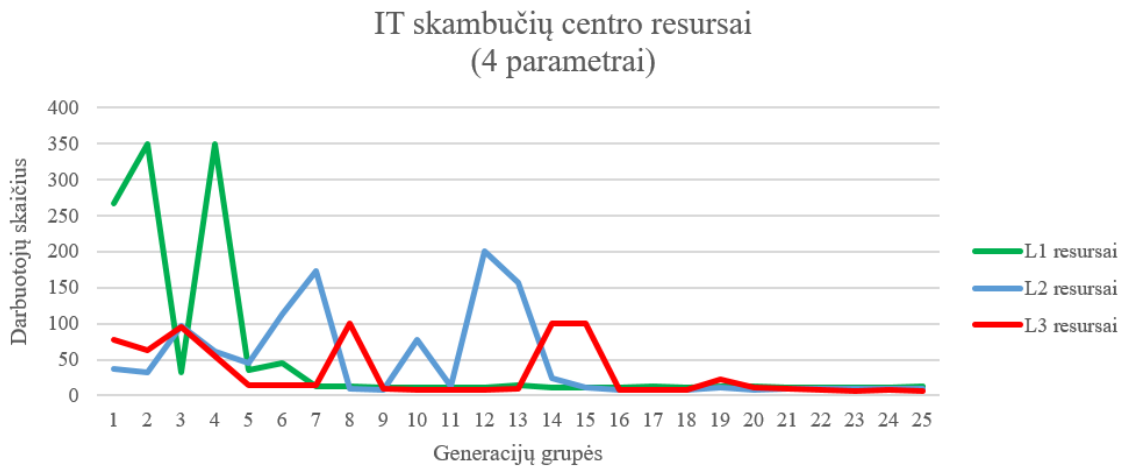
Svoriui  $a$  suteikiama reikšmė 5;  $b$  – 1;  $c$  – 6,7;  $d$  – 8. Pritaikius tokią funkciją matome, jog koeficientas nuosekliai artėja link optimalesnės reikšmės (žr. 27 pav.).





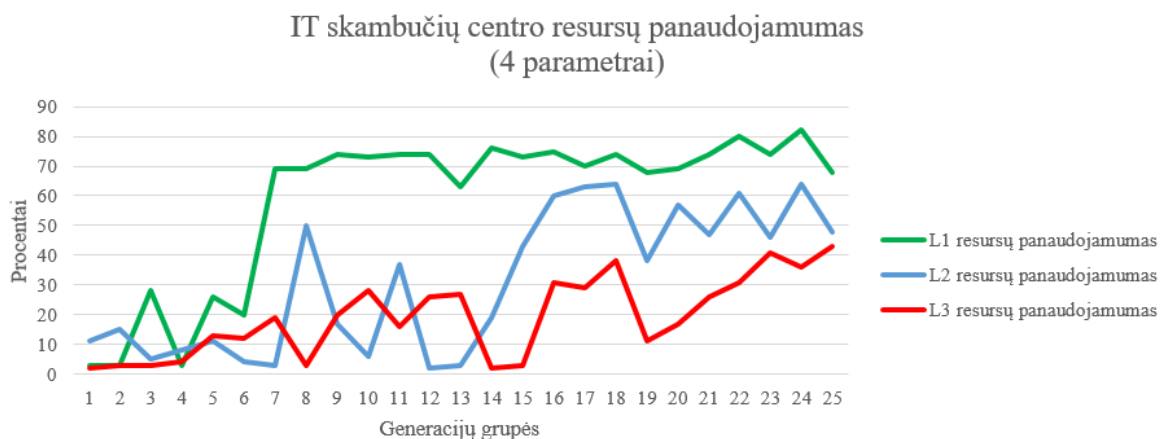
**27 pav.** Tinkamumo funkcijos koeficiento kitimas (4 parametrai)

Naudojant keturis parametrus, matome, jog genetinis algoritmas atsižvelgia į visų trijų lygių resursų panaudojimo rodiklį ir tinkamumo funkcijos rezultatas gerėja net ir išlaikant mažą pirmojo lygmens resursų kiekį (žr. 28 pav.).



**28 pav.** IT skambučių centro resursai (4 parametrai)

Tikslingai užpildomas ir darbuotojų užimtumas visuose lygiuose. Analizuojant šių simuliacijų rezultatus, paaiškėja, jog genetinis algoritmas atsižvelgia į tinkamumo funkcijoje nustatytą resursų panaudojamumo svorį ir pradinėse grupėse gavus mažą užimtumą, vėlesnėse generacijų grupėse užimtumas didinamas kiekviename lygyje, kol pasiekiamas optimalus resursų pasiskirstymo santykis (žr. 29 pav.).



**29 pav.** IT skambučių centro resursų panaudojumas (4 parametrai)

Atlikus bandymus su keturiais rodikliais, matome tikslesnius rezultatus. Generacijos su geriausiai tinkamumo funkcijos koeficientais siūlo naudoti 12 resursų pirmajam lygiui, 9 antrajam bei 8 trečiajam. Naudojant tokį kiekį darbuotojų, vidutinis laukimo laikas yra apytiksliai 1min. 8 sek. (rezultatas tenkinantis, nes apibrėžta kliento kantrybės riba laukiant atsako yra nuo 2 iki 5 minučių), vidutinis pasitenkinimo lygis yra 4,67 iš 5, skundų ir skambučių santykis yra 0,9% bei lygių užimtumai pasiskirto taip: L1 resursų panaudojumas – apie 82%, L2 – apie 64% ir L3 – apie 36%.

Galima daryti išvadą, jog algoritmo grąžinamų rezultatų tinkamumui ypatingai svarbu tiksli tinkamumo funkcija. Nepaisant to, jog algoritmas stengsis artėti link apibrėžto koeficiento, tai neužtikrina, jog rezultatai mus tenkins, jei tinkamumo funkcija bus suformuota netinkamai.

## IŠVADOS

Šio darbo metu apibrėžta darbo problematika, tikslai, uždaviniai ir analizuojamas objektas. Atlikus analizę bei eksperimentą ir aptarus kiekvieną baigiamojo darbo skyrių, galima daryti bendrąsias išvadas. Sugrįžtant prie apibrėžtų darbo uždavinių (žr. skyrių „Darbo tikslas ir uždaviniai“), kiekvienam iš jų formuluojamos išvados:

1. Atlikus su šia tema susijusių literatūros šaltinių analizę, galima daryti išvadą, jog procesai gali būti skirtingos prigimties ir analizuojami iš kelių skirtingų požiūrių, o jų įvairovė labai plati, priklausanti nuo verslo specifikos. Dėl šios priežasties ypač svarbu išsamiai išanalizuoti jo ypatumus ir sudaryti tikslų proceso modelį, jog darbo metu būtų gaunami kuo tikslesni rezultatai ir būtų lengviau numatomos įmanomos proceso baigtys, sudarant kuo skirtingesnes situacijas ir keičiant jų parametrus.
2. Išnagrinėjus verslo procesų modeliavimo ir simuliacijos metodus bei įrankius, paaiškėjo, jog modeliavimo būdų yra ne vienas, tačiau populiariausi, labiausiai integruoti į kasdienį įmonių naudojimą, būdami labiausiai sintaksiškai suderinti su kitais įrankiais, yra OMG organizacijos produktai (BPMN, UML, SysML). Išanalizavus simuliacijos metodus galima daryti prielaidą, jog vienas iš simuliacijai tinkamų būdų – genetinių algoritmų naudojamas. Jie padeda ne tik natūralioje aplinkoje ir gamtoje, bet ir įvairiose inžinerijos srityse bei kitiems apžvelgtiems praktiniams taikymams. Galima daryti išvadą, kad tokius algoritmus įmanoma pritaikyti ir verslo procesų simuliacijoms, o jų rezultatai gali padėti optimizuoti tam tikrų išteklių naudojimą.
3. Stengiantis pasiūlyti tikslo siekiančių verslo procesų simuliacijos metodą, buvo priimtas sprendimas kurti verslo proceso simuliacinį modelį ir apibrėžti konkrečiam verslui reikalingas tinkamumo funkcijas, kurios atspindėtų sprendžiamąją problematiką. Taip paaiškėjo, kad įmanoma toliau pritaikyti genetinio algoritmo vykdymą iki apibrėžto pertraukio signalo, naudojant reikalingus verslo sferos parametrus ir saugant simuliacijos duomenis tolesnėms analizėms.
4. Atlikus pasiūlyto metodo eksperimentinį tyrimą pasirinktame verslo procese, įrodyta, jog genetinis algoritmas gali padėti ieškant optimalaus parametrų rinkinio tikslui pasiekti. Šis teiginys patvirtintas pritaikius nagrinėtąjį metodą IT pagalbos skambučių centro procesui ir pasiūlius skirtingo lygio žmoniškųjų išteklių resursų derinius, artėjant prie verslui palankių sprendimų. Nustatyta, jog tinkamumo funkcijos apibrėžimas yra vienas iš svarbiausių žingsnių, artėjant prie optimalių ir tikslių rezultatų.

## TOLIMESNIŲ TYRIMŲ KRYPTYS

Pasiekus baigiamojo darbo tikslą (praplėsti tikslo siekiančių verslo procesų modeliavimo ir simuliacijos būdus, atliekant tyrimą ir ieškant našaus būdo galimoms baigtims numatyti), pateikiamos rekomendacijos tolesniems šių rezultatų tyrimams:

1. Eksperimentinės dalies simuliacinis modelis galėtų būti plečiamas atsižvelgiant į daugiau papildomų veiksnių, darančių įtaką verslo procesui ir varijuojančių galimas baigtis (pavyzdžiui, konkrečiu atveju – tarptautinės įmonės IT gedimų pasiskirstymas pagal skirtingose laiko juostose dirbančių vartotojų tankį ir kt.). Eksperimentinėje dalyje nagrinėjamo modelio optimizavimas galėtų būti praplėstas pridedant dar vieną ar kelias tinkamumo funkcijas pagal poreikį, bandant įrodyti ir daugiatislių genetinių algoritmų naudojimo galimybę.
2. Baigiamojo darbo pasiūlytą metodą galima pritaikyti naudojant kelis skirtingus genetinius (pavyzdžiui, literatūros analizės skyriuje išnagrinėtus ir tarpusavyje palygintus) algoritmus, lyginant rezultatus ir artėjant prie išvadų, kuris algoritmas pateikė tikslesnius, spartesnius ar platesnio spektro parametrų pasiūlymus.
3. Pritaikant genetinius algoritmus, toliau būtų galima analizuoti įvairias tinkamumo funkcijos formavimo metodikas. Kadangi tai yra vienas iš svarbių etapų, lemiančių galutinius rezultatus, galėtų būti gilinamos žinios apie tinkamumo funkcijos formavimą įvairiems uždaviniams spręsti ir atliekami tyrimai geriausioms praktikoms pritaikyti.

Pateikus keletą rekomendacijų verta paminėti, jog tai ne vienintelės sritys, kuriose galima išsamiau nagrinėti keliamus klausimus. Verslo procesų modeliavimas ir simuliacija yra plati tema, kurioje dar yra pakankamai erdvės tyrimams, analizei ir eksperimentams. Kadangi kiekvieno verslo procesų prigimtis yra skirtinga, sudėtinga rasti visiškai universalų būdą baigtims numatyti. Dėl to žinios apie jau esamas metodikas gali būti toliau plečiamos, o naujų būdų paieškos taip pat galėtų būti inicijuojamos.

## LITERATŪROS SĄRAŠAS

Abdalla S. (2017). *Genetic Algorithms: A Gentle Introduction*. Prieiga per internetą: [https://www.safaribooksonline.com/oriole/genetic\\_algorithms\\_a\\_gentle\\_introduction](https://www.safaribooksonline.com/oriole/genetic_algorithms_a_gentle_introduction) [žiūrėta 2018-06-07]

Allweyer, T. (2016). *BPMN 2.0 – Introduction to the Standard for Business Process Modeling*.

Bajaj, M., Waikar, M., Backhaus, J., Zwemer, D., Walden, T., Schreiber, C. (2017). Graph-Based Digital Blueprint for Model Based Engineering of Complex Systems. *27th Annual INCOSE International Symposium (IS 2017)*. Prieiga per internetą: [http://www.omg.sysml.org/Graphs\\_MBE\\_INCOSE\\_IS\\_Bajaj-et-al.pdf](http://www.omg.sysml.org/Graphs_MBE_INCOSE_IS_Bajaj-et-al.pdf) [žiūrėta 2017-12-01]

Baker, H. (2013). *Rightsizing Your Help Desk Team* Prieiga per internetą: <https://www.business2community.com/leadership/rightsizing-your-help-desk-team-part-3-0402710#qh2BhLtWWfcLzoEi.97> [žiūrėta 2019-05-09]

BPMB. (2011). *BPMN 2.0 – Business Process Model and Notation*. Prieiga per internetą: [http://www.bpmb.de/images/BPMN2\\_0\\_Poster\\_EN.pdf](http://www.bpmb.de/images/BPMN2_0_Poster_EN.pdf) [žiūrėta 2019-05-10]

Cheu, R. L., Wang, Y., Fwa, T. F. (2004). *Genetic Algorithm-Simulation Methodology for Pavement Maintenance Scheduling*. Prieiga per internetą: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8667.2004.00369.x> [žiūrėta 2018-06-07]

Corne, D. W., Knowles, J. D., Oates, M. J. (2000). The pareto envelope-based selection algorithm for multiobjective optimisation. *Parallel Problem Solving from Nature – PPSN VI*, 839–848.

Courses PSU. (2016). *What is the Objective Function?* Prieiga per internetą: [https://www.courses.psu.edu/for/for466w\\_mem14/Ch11/HTML/Sec1/ch11sec1\\_ObjFn.htm](https://www.courses.psu.edu/for/for466w_mem14/Ch11/HTML/Sec1/ch11sec1_ObjFn.htm) [žiūrėta 2018-06-07]

Deb, K. (2005). *Multi-objective NSGA-II code in C*

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation*, 6(2), 182-197

Digalakis, J.G., Margaritis, K. G. (2002). An Experimental Study of Benchmarking Functions for Genetic Algorithms. *International Journal of Computer Mathematics*, 79, 403-416.

Friedenthal, S., Moore. A., Steiner. R. (2009). *OMG Systems Modeling Language (OMG SysML™) Tutorial*. Prieiga per internetą: <http://www.omg.sysml.org/INCOSE-OMGSysML-Tutorial-Final-090901.pdf> [žiūrėta 2017-12-01]

Gro C., Dumitrescu, D. (2002). *A comparison of multiobjective evolutionary algorithms*. Prieiga per internetą: [http://auajournal.uab.ro/upload/53\\_474\\_Crina.pdf](http://auajournal.uab.ro/upload/53_474_Crina.pdf) [žiūrėta 2018-06-16]

Half, R. (2017a). *Do You Need More Help Desk Tier 2 Support Staff?* Prieiga per internetą: <https://www.roberthalf.com/blog/management-tips/do-you-need-more-help-desk-tier-2-support-staff> [žiūrėta 2019-05-09]

Half, R. (2017b). *How Many Help Desk Tier 1 Personnel Do You Need?* Prieiga per internetą: <https://www.roberthalf.com/blog/management-tips/how-many-help-desk-tier-1-personnel-do-you-need> [žiūrėta 2019-05-09]

Halim, R. A., Seck, M. D. (2011). *The Simulation-based Multi-objective Evolutionary Optimization (SIMEON) Framework*.

Hesse, M. (2019). *BPMN Tool Matrix*. Prieiga per internetą: <https://bpmnmatrix.github.io> [žiūrėta 2019-04-24]

Hommes, B. J., Reijswoud, V. (2000). Assessing the Quality of Business Process Modelling Techniques. *Proceedings of the 33rd Hawaii International Conference on System Science*. Prieiga per internetą: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.134.4530&rep=rep1&type=pdf> [žiūrėta 2018-01-03]

IGI Global. (2018) *What is Pareto Optimal Solution?* Prieiga per internetą: <https://www.igi-global.com/dictionary/pareto-optimal-solution/21879> [žiūrėta 2018-06-07]

Yusoff, Y., Ngadiman, M. S., Zain, A.M. (2011). Overview of NSGA-II for Optimizing Machining Process Parameters. *Procedia Engineering*, 15, 3978 -3983

Kalibatiene, D., Vasilecas, O., Savickas, T., Vysockis, T., Bobrovs, V. (2016). A New Approach on Rule and Context Based Dynamic Business Process Simulation. *Baltic J. Modern Computing*, 4(3), 408-419.

Kanri, H. (2016). *The Strategic Approach to Continuous Improvement*.

Khare, V., Yao, X., Deb K. (2003). Performance Scaling of Multi-objective Evolutionary Algorithms. *International Conference on Evolutionary Multi-Criterion Optimization*, 376-390.

Kopec, D. (2018). *Classic Computer Science Problems in Swift: Essential techniques for practicing programmers*. Prieiga per internetą: <https://www.safaribooksonline.com/library/view/classic-computer-science/9781617294891/> [žiūrėta 2018-06-07]

Krogstie, J., Sindre, G., Jørgensen, H. (2006). Process models representing knowledge for action: a revised quality framework. *European Journal of Information Systems*, 15, 91–102. Prieiga per internetą: <https://link.springer.com/content/pdf/10.1057%2Fpalgrave.ejis.3000598.pdf> [žiūrėta 2018-01-03]

Laguna, M., Marklund, J. (2019). *Business Process Modeling, Simulation and Design*.

Laumanns, M., Rudolph, G., Schwefel, H.P. (2001). Mutation control and convergence in evolutionary multi-objective optimization. *Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001)*, Brno, Czech Republic.

Livy, D. (2018). *What's the Average IT Service Desk to Employee Ratio?*

Mallawaarachchi V. (2017a). *How to define a Fitness Function in a Genetic Algorithm?* Prieiga per internetą: <https://towardsdatascience.com/how-to-define-a-fitness-function-in-a-genetic-algorithm-be572b9ea3b4> [žiūrėta 2018-06-07]

Mallawaarachchi V. (2017b). *Introduction to Genetic Algorithms—Including Example Code*. Prieiga per internetą: <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3> [žiūrėta 2018-06-07]

Martin, W. N., Spears, W., Martin, W. (2001). *Foundations of Genetic Algorithms 2001 (FOGA 6)*, Prieiga per internetą: <https://www.safaribooksonline.com/library/view/foundations-of-genetic/9781558607347/> [žiūrėta 2018-06-07]

Mendling, J., Reijers, H.A., Aalst, W.M.P. (2010). Seven process modeling guidelines (7PMG). *Information and Software Technology*, 52, 127–136. Prieiga per internetą: <https://pdfs.semanticscholar.org/5c4a/08a28b46fec62a0b45a26647330a0e69feb8.pdf> [žiūrėta 2018-01-03]

OMG 2017a. *What is SysML?* Prieiga per internetą: <http://www.omg.sysml.org/what-is-sysml.htm> [žiūrėta 2017-12-01]

Pearce, P., Friedenthal, S. (2013). A Practical Approach For Modelling Submarine Subsystem Architecture In SysML. *Submarine Institute of Australia Science, Technology & Engineering Conference* 2013. [http://www.omg.sysml.org/A\\_Practical\\_Approach\\_for\\_Modelling\\_Submarine\\_Subsystem\\_Architecture\\_in\\_SysML-Pearce\\_Friedenthal.pdf](http://www.omg.sysml.org/A_Practical_Approach_for_Modelling_Submarine_Subsystem_Architecture_in_SysML-Pearce_Friedenthal.pdf) [žiūrėta 2017-12-01]

Roland, C. (1993). *Modeling the Requirements Engineering Process*. Prieiga per internetą: [https://www.researchgate.net/publication/2824908\\_Modeling\\_the\\_Requirements\\_Engineering\\_Process](https://www.researchgate.net/publication/2824908_Modeling_the_Requirements_Engineering_Process) [žiūrėta 2018-01-03]

Roland, C. (1998). A Comprehensive View of Process Engineering. *International Conference on Advanced information Systems Engineering*. Prieiga per internetą: <https://hal.archives-ouvertes.fr/hal-00707940/document> [žiūrėta 2018-01-03]

Rus, I., Holger, N., Jürgen, M. (2014). *A systematic methodology for developing discrete event simulation models of software development processes*.

Rusinaite, T., Kalibatiene, D., Vasilecas, O. (2015). Requirements of dynamic business processes - a survey. *2015 IEEE 3rd Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)* Prieiga per internetą: <https://www.semanticscholar.org/paper/Requirements-of-dynamic-business-processes-a-survey-Rusinaite-Kalibatiene/45d22bf305dc816126a5a5eb84e2b6b78f11a163> [žiūrėta 2018-05-26]

ServiceNow. (2019). *IT Service Management*. Prieiga per internetą: <https://www.servicenow.com/products/it-service-management.html> [žiūrėta 2019-05-10]

Si, Y., Chan, V., Dumas, M., Zhang, D. (2018). A Petri Nets Based Generic Genetic Algorithm Framework for Resource Optimization in Business Processes. *Simulation Modelling Practice and Theory*, 86, 72-101.

Sianipar, C. P. M., Yudoko, G., Dowaki, K. (2014). Physiological Concept: Visible Modeling for Feasible Design. *Applied Mechanics and Materials*. 493. Prieiga per internetą: [https://www.researchgate.net/publication/259494630\\_Physiological\\_Concept\\_Visible\\_Modeling\\_for\\_Feasible\\_Design](https://www.researchgate.net/publication/259494630_Physiological_Concept_Visible_Modeling_for_Feasible_Design) [žiūrėta 2018-01-03]

SmartDraw. (2019). *Business Process Map*. Prieiga per internetą: <https://www.smartdraw.com/business-process-mapping/> [žiūrėta 2019-05-06]

Vasilecas, O. (2018). Dynamic Business Process and their Simulation: A Rule- and Context-Based Approach. *Proc. at Ben-Gurion University of the Negev*.

Vasilecas, O., Kalibatiene, D., Rima, A., Birzniece, I., Rudzajs, P. (2015). A Resource Model for the Rule-Based Dynamic Business Process Modelling and Simulation. *Proc. of the 29th European Simulation and Modelling Conference (ESM'2015)*, 36-41.

Vasilecas, O., Savickas, T., Normantas, K., Vysockis, T., Kalibatiene, D. (2016). A Goal-Oriented Approach to Dynamic Business Process Simulation. *Databases and Information Systems IX– Selected Papers*, 143-154.

Vergidis, K., Tiwari, A., Majeed, B. (2006). Business Process Improvement using Multi-objective Optimisation. *BT Technology Journal*, 24



Vergidis, K.; Tiwari, A.; Majeed, B.; Roy, R. (2007). Optimisation of Business Process Designs: An algorithmic approach with multiple objectives. *International Journal of Production Economics*, 109(1-2).

White, S. A. (2004). *Introduction to BPMN*. Prieiga per internetą: [https://www.omg.org/bpmn/Documents/Introduction\\_to\\_BPMN.pdf](https://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf) [žiūrėta 2019-04-24]

Wyld, D. C., Wozniak, M., Chaki, N., Meghanathan, N., Nagamalai, D. (2011). Advances in Computing and Information Technology. *First International Conference, ACITY 2011, Chennai, India, July 15-17, 2011, Proceedings*. Prieiga per internetą: <https://books.google.co.uk/books?id=JHgQBwAAQBAJ&dq> [žiūrėta 2018-06-07]

Zeigler, B. P., Sarjoughian, H. S. (2008). *DEVS Component-Based M&S Framework: An Introduction*. Prieiga per internetą: [https://grid.cs.gsu.edu/xhu/ClassArchive/CSC8840\\_Spring2008/ACIMS%20DEVSTut%20AIS200%20Final.pdf](https://grid.cs.gsu.edu/xhu/ClassArchive/CSC8840_Spring2008/ACIMS%20DEVSTut%20AIS200%20Final.pdf) [žiūrėta 2018-06-18]

Zeigler, B. P., Song, H. S., Kim, T. G., Praehofer H. (2005). DEVS framework for modelling, simulation, analysis, and design of hybrid systems. *International Hybrid Systems Workshop*, 529-551.

Zitzler, E., Laumanns, M., Thiele, L. (2001). *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Prieiga per internetą: <https://www.research-collection.ethz.ch/bitstream/handle/20.500.11850/145755/eth-24689-01.pdf> [žiūrėta 2018-06-18]

Zitzler, E., Thiele, L. (1998). *An evolutionary algorithm for multiobjective optimization: The strength pareto approach*.