

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIU MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Daumantas Petkevičius

DINAMINIUOSE VERSLO PROCESUOSE NAUDOJAMŲ IŠTEKLIŲ OPTIMIZAVIMAS SIEKIANT APIBRĖŽTOS JŲ BŪSENOS

OPTIMISATION OF DYNAMIC BUSINESS PROCESS RESOURCES TO PURSUE THEIR DEFINITE STATE

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos programa, valstybinis kodas 621E15002
Informatikos inžinerijos studijų kryptis

Vilnius, 2018

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIU MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU

Katedros vedėjas

_____
(Parašas)

_____
(Vardas, pavardė)

_____
(Data)

Daumantas Petkevičius

DINAMINIUOSE VERSLO PROCESUOSE NAUDOJAMŲ IŠTEKLIŲ OPTIMIZAVIMAS SIEKIANT APIBRĖŽTOS JŲ BŪSENOS

OPTIMISATION OF DYNAMIC BUSINESS PROCESS RESOURCES TO PURSUE THEIR DEFINITE STATE

Baigiamasis magistro darbas

Informacinių sistemų programų inžinerijos programa, valstybinis kodas 621E15002

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	lekt. Regina Žukienė (Moksl. laipsnis/pedag. vardas, vardas, pavardė)	_____ (Parašas)	_____ (Data)

Vilnius, 2018

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIU MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

TVIRTINU

Katedros vedėjas

Informatikos inžinerijos studijų kryptis

Informacinių sistemų programų inžinerijos studijų programa, valstybinis kodas 621E15002

_____
(Parašas)

prof. dr. Dalius Mažeika
(Vardas, pavardė)

_____
(Data)

BAIGIAMOJO MAGISTRO DARBO UŽDUOTIS

.....Nr. .....

Vilnius

Studentui Daumantui Petkevičiui

Baigiamojo darbo tema: Dinaminiuose verslo procesuose naudojamų išteklių optimizavimas siekiant apibrėžtos jų būsenos.

patvirtinta 2017 m. spalio 25 d. dekano potvarkiu Nr. 437 fm

Baigiamojo darbo užbaigimo terminas 2018 m. birželio 8 d.

BAIGIAMOJO DARBO UŽDUOTIS:

1. Atlikti dinaminių verslo procesų ir juose naudojamų išteklių literatūros analizę;
2. Išnagrinėti dinaminių verslo procesų optimizavimo būdus;
3. Remiantis analize pasiūlyti metodą, kurį naudojant būtų galima optimizuoti dinaminius verslo proceso išteklius siekiant apibrėžtos jų būsenos;
4. Atlikti pasiūlyto metodo eksperimentinį tyrimą, pateikti eksperimento rezultatus ir jų analizę, suformuoti galutines išvadas.

Baigiamojo darbo rengimo konsultantai: Tadas Vysockis
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

Lietuvių kalbos konsultantas: lekt. Regina Žukienė
(Moksl. laipsnis/pedag. vardas, vardas, pavardė)

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

Užduotį gavau

.....
(Parašas)

Daumantas Petkevičius
(Vardas, pavardė)

.....
(Data)

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių sistemų katedra

ISBN ISSN
Egz. sk. .....
Data .....-.....-.....

Antrosios pakopos studijų **Informacinių sistemų programų inžinerijos** programos magistro baigiamasis darbas
Pavadinimas **Dinaminiuose verslo procesuose naudojamų išteklių optimizavimas siekiant apibrėžtos jų būsenos**
Autorius **Daumantas Petkevičius**
Vadovas **Olegas Vasilecas**

Kalba: lietuvių

Anotacija

Baigiamajame magistro darbe nagrinėjamas dinaminių verslų procesų optimizavimas atsižvelgiant į siekiamą išteklių būseną ir naudojant kompiuterinę simuliaciją. Darbe aptariamos verslo taisyklės ir procesai, procesų simuliacija, analizė ir optimizacija. Pasiūlytas optimizavimo metodas, kuris remiasi verslo proceso simuliacinio modelio parametrizavimu kartu su euristinių paieškos metodų kaip simplekso ir genetinės paieškos taikymu. Metodo veikimo principas pagrįstas geriausio parametrų reikšmių rinkinio paieška, su kuria atlikus proceso simuliaciją, gaunama siekiama išteklių būsena.Metodas realizuotas praplečiant VGTU sukurtą „DBPSim“ procesų simuliacijos įrankį.

Praktinėje dalyje atliktas eksperimentinis tyrimas, kurio tikslas – optimizuoti ligoninės priimamojo verslo proceso vykdymo trukmę ir kainą. Šiame tyrime parametrų reikšmės, kurioskeičiamos optimizacijos metu, yra procese dalyvaujančių pareigybių kiekiai. Gauti rezultatai parodė, kad abu realizuoti paieškos algoritmai geba surasti tokias parametrų reikšmes, kurias naudojant gaunama mažesnė proceso vykdymo trukmė ir kaina lyginant su tuo atveju, kai optimizacija netaikoma ir naudojamos fiksuotos parametrų reikšmes.

Darbo apimtis 67 p. teksto be priedų, 22 pav., 12 lentelių, 37 bibliografinių šaltinių. Atskirai pateikiami darbo priedai.

Prasminiai žodžiai: dinaminiai verslo procesai, išteklių optimizacija, simuliacija, simuliacinio modelio parametrizacija, genetinė paieška, simplekso paieška

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Systems

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

Master Degree Studies **Information Systems Software Engineering** study programme Master Graduation Thesis

Title	**Optimisation of Dynamic Business Process Resources to Pursue their Definite State**
Author	**Daumantas Petkevičius**
Academic supervisor	**Olegas Vasilecas**

Thesis language: Lithuanian

Annotation

The final master thesis examines the optimization of dynamic business processes in relation to the definite state of resources and using computer-based simulation. The paper describes concepts of business rule and process, process simulation, analysis and optimisation. The proposed optimization method is based on the parametrization of business process simulation model along with the use of heuristic search methods such as simplex algorithm and genetic search.The proposed method operates by seeking the best input parameters values configuration which is used in process simulation and is capable to produce the predefined state of resources. The method is implemented as an extension of „DBPSim“ simulation tool.

Practical part of the paper consists of an experimental research which goal is to optimize the duration and cost of hospital business process. The parameters which are being changed during the optimisation corresponds to the quantities of medical posts. The obtained results verify that both implemented search algorithms are capable of finding such parameters values that result in less process execution time and cost compared to the case where optimisation is not applied.

Thesis consists of 67 pages of text without appendixes, 22 pictures, 12 tables and 37 references. Appendixes are supplied separately.

Keywords: dynamic business process, optimisation of resources, simulation, parametrization of a simulation model, genetic search, simplex search

Vilniaus Gedimino technikos universiteto egzaminų sesijų ir baigiamųjų darbų rengimo bei gynimo organizavimo tvarkos aprašo
2 priedas

(Baigiamojo darbo sąžiningumo deklaracijos forma)

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Daumantas Petkevičius, 20160769
(Studento vardas ir pavardė, studento pažymėjimo Nr.)
Fundamentinių mokslų fakultetas
(Fakultetas)
Informacinių sistemų programų inžinerija, ISIfm-16
(Studijų programa, akademinė grupė)

BAIGIAMOJO DARBO (PROJEKTO)

SĄŽININGUMO DEKLARACIJA

2018 m. birželio 08 d.

Patvirtinu, kad mano baigiamasis darbas tema „Dinaminiuose verslo procesuose naudojamų išteklių optimizavimas siekiant apibrėžtos jų būsenos“ patvirtintas 2017 m. spalio 25 d. dekano potvarkiu Nr. 437fm, yra savarankiškai parašytas. Šiame darbe pateikta medžiaga nėra plagijuota. Tiesiogiai ar netiesiogiai panaudotos kitų šaltinių citatos pažymėtos literatūros nuorodose.

Parenkant ir įvertinant medžiagą bei rengiant baigiamąjį darbą, mane konsultavo mokslininkai ir specialistai: Tadas Vysockis. Mano darbo vadovas prof. dr. Olegas Vasilecas.

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

(Parašas) Daumantas Petkevičius (Vardas ir pavardė)

TURINYS

PAVEIKSLŲ SĄRAŠAS **9**

LENTELIŲ SĄRAŠAS **10**

SANTRUMPOS **11**

ĮVADAS **12**

Tyrimo objekas . . . 12
Darbo tikslai ir uždaviniai . . . 12
Temos naujumas . . . 13
Temos aktualumas . . . 13
Tyrimo metodika . . . 13
Mokslinė darbo vertė . . . 13
Darbo struktūra . . . 14

1. VERSLO TAISYKLĖS IR PROCESAI **15**

1.1. Verslo taisyklės . . . 15
1.1.1. Verslo taisyklių apibrėžtis . . . 15
1.1.2. Verslo taisyklių klasifikavimas . . . 15
1.2. Verslo procesai . . . 17
1.2.1. Verslo procesų apibrėžtis . . . 17
1.2.2. Verslo procesų klasifikavimas . . . 18
1.2.3. Dinaminiai verslo procesai . . . 18
1.2.4. Verslo procesų valdymas . . . 19
1.3. Verslo procesų modeliavimas . . . 20
1.3.1. Verslo procesų modeliavimo apibrėžtis . . . 20
1.3.2. BPMN modeliavimo kalba . . . 21
1.4. Verslo procesų simuliacija . . . 22
1.4.1. Verslo procesų simuliacijos apibrėžtis . . . 22
1.4.2. Simuliacijos privalumai ir trūkumai . . . 23
1.4.3. Diskrečiųjų įvykių simuliacija . . . 24
1.4.4. Verslo procesų ištekliai ir jų simuliacija . . . 25
1.4.5. Tikslo siekiančių dinaminių verslo procesų simuliacinis modelis . . . 28
1.4.6. Verslo procesų simuliacijos įrankiai . . . 29
1.5. Apibendrinimas . . . 31

2. VERSLO PROCESŲ OPTIMIZAVIMAS **32**

2.1. Verslo procesų optimizavimo apibrėžtis . . . 32
2.2. Verslo procesų optimizavimas vykdant simuliaciją . . . 33

2.2.1. Simplekso algoritmas 35
2.2.2. Genetinis algoritmas 39
2.3. Struktūrinė optimizacija 40
2.4. Apibendrinimas 43

3. DINAMINIUOSE VERSLO PROCESUOSE NAUDOJAMŲ IŠTEKLIŲ OPTIMIZAVIMO METODAS 44
3.1. Metodo bendra apžvalga 44
3.2. Įvesties parametrų reikšmių keitimas 46
3.3. Tikslo įvertinimo funkcija 48
3.4. Apibendrinimas 50

4. IŠTEKLIŲ OPTIMIZAVIMO PROTOTIPO REALIZACIJOS APRAŠYMAS 51
4.1. Išplėstiniai komponentai 51
4.2. Kiti komponentai 54
4.3. Apibendrinimas 55

5. EKSPERIMENTINIS TYRIMAS 56
5.1. Tiriamas verslo procesas 56
5.2. Eksperimentinis tyrimas 57
5.3. Apibendrinimas 62

IŠVADOS 63

TOLIMESNIŲ TYRIMŲ KRYPTYS 64

LITERATŪRA 65

PRIEDAI 68

PAVEIKSLŲ SĄRAŠAS

1 pav. Verslo taisyklių klasifikacija 16
2 pav. Verslo proceso schema 17
3 pav. Verslo procesų valdymo gyvavimo ciklas 20
4 pav. Verslo proceso BPMN modelio pavyzdys 21
5 pav. Diskrečiųjų įvykių simuliacijos algoritmas 25
6 pav. Simuliacinio modelio optimizavimas taikant euristinę paiešką 34
7 pav. Simplekso pavyzdžiai 35
8 pav. Simplekso algoritme atliekamų veiksmų atvaizdavimas dvimatėje erdvėje . . 37
9 pav. Simplekso algoritmo veiksmų diagrama 38
10 pav. Genetinio algoritmo veiksmų diagrama 40
11 pav. Proceso scenarijaus formavimas taikant PCA algoritmą 43
12 pav. Verslo proceso optimizacija naudojant simuliaciją 45
13 pav. Įvesties parametrų reikšmių keitimas naudojant simplekso algoritmą 47
14 pav. Įvesties parametrų reikšmių keitimas naudojant genetinę paiešką 48
15 pav. Išteklių būsenų erdvė taikant skirtingus būsenų vertinimo metodus 50
16 pav. Prototipo dalies išplėstinių komponentų UML klasių diagrama 52
17 pav. Skubios pagalbos suteikimo ligoninėje verslo procesas 56
18 pav. Įvertinimo funkcijos S_2 verčių priklausomybė nuo iteracijų skaičiaus taikant genetinę paiešką ir esant 60 pacientų kiekio atvejui 59
19 pav. Proceso vykdymo trukmės ir kainos priklausomybė nuo pacientų skaičiaus . 60
20 pav. Verslo proceso simuliacijos procesas 68
21 pav. Verslo proceso optimizacijos rodikliai 68
22 pav. Veiklos aprašo pavyzdys, kuriame apibrėžiami įvesties parametrai su pradinėmis reikšmėmis 69

LENTELIŲ SĄRAŠAS

1 lentelė. Veiklų-išteklių matrica 42
2 lentelė. Būsenos keitiklio klasės elementų aprašymas 53
3 lentelė. Algoritmo klasės elementų aprašymas 54
4 lentelė. Taisyklių šablono klasės papildomos funkcijos 55
5 lentelė. Skubios pagalbos suteikimo ligoninėje verslo proceso veiklų detalės 57
6 lentelė. Skubios pagalbos suteikimo ligoninėje verslo proceso išlaidos 57
7 lentelė. Simuliacinio modelio pradinės parametrų reikšmės 58
8 lentelė. Proceso vykdymo trukmės ir kainos priklausomybė nuo paciento skaičiaus esant fiksuotoms įvesties parametrų reikšmėms (optimizacija netaikoma) 58
9 lentelė. Siekiamia proceso vykdymo trukmė ir kaina priklausomai nuo pacientų skaičiaus 59
10 lentelė. Proceso vykdymo trukmės ir kainos priklausomybė nuo paciento skaičiaus taikant simplekso algoritmą 60
11 lentelė. Proceso vykdymo trukmės ir kainos priklausomybė nuo paciento skaičiaus taikant genetinę optimizaciją 60
12 lentelė. Proceso vykdymo trukmės ir kainos priklausomybė nuo paciento skaičiaus naudojant artimiausią tikslui gautą reikšmių rinkinį 61

SANTRUMPOS

DVP – Dinaminis verslo procesas (angl. *Dynamic Business Process*)

VT – Verslo taisyklė (angl. *Business Rule*)

OCL – Objektų ribojimo kalba (angl. *Object Constraint Language – OCL*)

UML – Suvienodinta modeliavimo kalba (angl. *Unified Modeling Language – UML*)

BPMN – Verslo procesų modelis ir žymėjimas (angl. *Business Process Model and Notation – BPMN*)

ĮVADAS

Kiekvienoje verslo organizacijoje vyksta įvairūs procesai. Iš esmės bet kokį verslą galime įsivaizduoti kaip sudėtingą tarpusavyje susijusių verslo procesų rinkinį. Sėkmingai veikiančios įmonės skiria daug dėmesio verslo procesų valdymui, nes jų analizė leidžia optimizuoti įmonės veiklą ir didinti efektyvumą. Vienas iš svarbiausių kiekvieno verslo proceso komponentų yra ištekliai. Kiekvienai verslo organizacijai ypač svarbu organizuoti savo procesus atsižvelgiant į ribotą išteklių kiekį, nes neefektyvus jų valdymas dažnai lemia ženklius finansinius nuostolius. Pavyzdžiui, gamybos įmonė, kuri nesugeba prognozuoti kiek, kada ir kokių žaliavų jai reikės, tikėtina jų pritrūks arba sandėliuos nereikalingai didelį kiekį. Tačiau toks verslo procesų, juose naudojamų išteklių optimizavimas nėra lengvas uždavinys, ypač atsižvelgiant į tai, kad dažnai verslo procesams būdingas ne griežtas apibrėžimas, o dinamiškumas, kitimas priklausomai nuo aplinkos sąlygų. Panašiems uždaviniams spręsti vis plačiau taikoma kompiuterinė verslo procesų simuliacija. Simuliacijos pagalba galima imituoti procesų pakeitimus ir kiekybiškai įvertinti jų įtaką išvengiant realiame pasaulyje egzistuojančių rizikų. Tačiau simuliacijos pritaikymas dinaminių verslo procesų išteklių optimizacijai yra sudėtingas uždavinys, turintis savo specifinių problemų, todėl bet koks indėlis į šio uždavinio sprendimą būtų labai aktualus mokslui ir verslo organizacijoms.

Šiame darbe tiriama problema: nėra metodų, kurie leistų optimizuoti dinaminiuose verslo procesuose naudojamus išteklius tokiu būdu, kad įvykdžius procesą būtų gauta iš anksto apibrėžta išteklių rinkinio būsena.

Tyrimo objekas

Tyrimo objektas – dinaminiuose verslo procesuose naudojamų išteklių optimizavimas siekiant apibrėžtos išteklių rinkinio būsenos.

Darbo tikslai ir uždaviniai

Darbo tikslas – patobulinti dinaminiuose verslo procesuose naudojamų išteklių optimizaciją pasiūlant metodą, kuris leistų gauti iš anksto apibrėžtą išteklių būseną.

Darbo tikslui pasiekti apibrėžiami šie uždaviniai:

1. Atlikti dinaminių verslo procesų ir juose naudojamų išteklių literatūros analizę;
2. Išnagrinėti dinaminių verslo procesų optimizavimo būdus;
3. Remiantis analize pasiūlyti metodą, kurį naudojant būtų galima optimizuoti dinaminius verslo proceso išteklius siekiant apibrėžtos jų būsenos;
4. Atlikti pasiūlyto metodo eksperimentinį tyrimą, pateikti eksperimento rezultatus ir jų analizę, suformuoti galutines išvadas.

Temos naujumas

Efektyvus verslo valdymas kuria finansinę naudą, todėl verslo optimizavimo idėjos pačios savaime nėra naujos ir nagrinėjamos nuo verslo, kaip reiškinio, atsiradimo pradžios. Tačiau verslo procesams būdingas dinamiškumas yra palyginti šių laikų bruožas, kurį paskatino technologijų vystymasis ir spartesnis gyvenimo tempas. Šio darbo tematika taip pat yra nauja tuo, kad tiriamas dinaminiuose verslo procesuose naudojamų išteklių optimizavimas pritaikant kompiuterinės simuliacijos metodą. Nors verslas gali būti optimizuojamas įvairiais pjūviais ir taikant įvairius būdus, tačiau tobulėjant skaičiavimo įrangai kompiuterinė procesų simuliacija tampa vis patrauklesnis metodas kiekybiškai ir be jokios rizikos vertinti procesų elgseną. Visgi tuo pačiu ši tyrinėjimų kryptis yra pakankamai jauna, todėl simuliacijos taikymas verslo proceso optimizacijai kelia daug neatsakytų klausimų.

Temos aktualumas

Kiekvienai verslo organizacijai ypač aktualu organizuoti savo procesus atsižvelgiant į ribotą išteklių kiekį, nes jų trūkumas ar perteklius įprastai lemia finansinius nuostolius. Tačiau toks procesuose naudojamų išteklių optimizavimas nėra paprastas uždavinys, nes šiuolaikiniams verslo procesams būdingas ne griežtas apibrėžimas, o dinamiškumas, kitimas priklausomai nuo aplinkos sąlygų. Bet kokie bandymai realioje aplinkoje dažnai yra per daug rizikingi, todėl verslui labai svarbu iš anksto tiksliai žinoti, kokie pokyčiai turi būti įgyvendinti procese, kad būtų pasiektas norimas rezultatas. Dėl šių priežasčių ši tema yra aktuali bet kuriai verslo organizacijai.

Tyrimo metodika

Darbo literatūros analizė atlikta naudojant bibliotekinio tyrimo, lyginamosios – kontekstinės analizės, sintezės, sąvokų apibendrinimo metodus. Remiantis atlikta analize, taikomas eksperimentinis tyrimas pasiūlytam metodui įgyvendinti. Eksperimento rezultatai analizuojami naudojant kiekybinį, lyginamąjį vertinimą.

Mokslinė darbo vertė

Dinaminių verslo procesų optimizavimas atsižvelgiant į siekiamą išteklių būseną yra aktuali, tačiau menkai ištyrinėta problema. Šiai problemai spręsti darbe pasiūlytas metodas, kuris pagrįstas veiklų parametrizavimu ir euristinės paieškos taikymu. Taip pat atliktas eksperimentinis tyrimas, kuris parodo pasiūlyto metodo efektyvumą nagrinėjamo proceso atveju. Gauti rezultatai gali būti toliau naudojami plėtojant verslo procesų optimizavimo idėjas.

Darbo struktūra

Įvade aptariama nagrinėjama problema, temos naujumas bei aktualumas. Taip pat apibrėžiamas darbo tikslas ir jam pasiekti iškelti uždaviniai. Galiausiai pateikiama mokslinė darbo vertė.

Pirmajame skyriuje atliekama literatūros analizė, susijusi su verslo taisyklėmis, verslo procesais, jų modeliavimu ir simuliacija.

Antrajame skyriuje atliekama literatūros analizė, susijusi su verslo procesų analize ir optimizavimu.

Trečiajame skyriuje pasiūlomas dinaminiuose verslo procesuose naudojamų išteklių optimizavimo metodas.

Ketvirtajame skyriuje pateiktas pasiūlyto dinaminiuose verslo procesuose naudojamų išteklių optimizavimo metodo realizacijos aprašymas.

Penktajame skyriuje atliekamas eksperimentinis tyrimas naudojant pasiūlytą išteklių optimizavimo metodą. Pateikiami gauti rezultatai.

Pabaigoje pateikiamos darbo išvados, naudotos literatūros sąrašas, galimos tolimesnės tyrimų kryptys bei darbo priedai.

1. VERSLO TAISYKLĖS IR PROCESAI

1.1. Verslo taisyklės

1.1.1. Verslo taisyklių apibrėžtis

Kiekvienoje verslo organizacijoje yra taikomos verslo taisyklės (toliau VT). Bendrai kalbant, VT yra teiginys, kuriuo naudojantis galima apibrėžti ir suvaržyti tam tikrą verslo aspektą. Norint geriau suvokti, kas yra VT, reikia nuspręsti, iš kieno požiūrio taško jas nagrinėti – verslo atstovo ar informacinių sistemų kūrėjo.

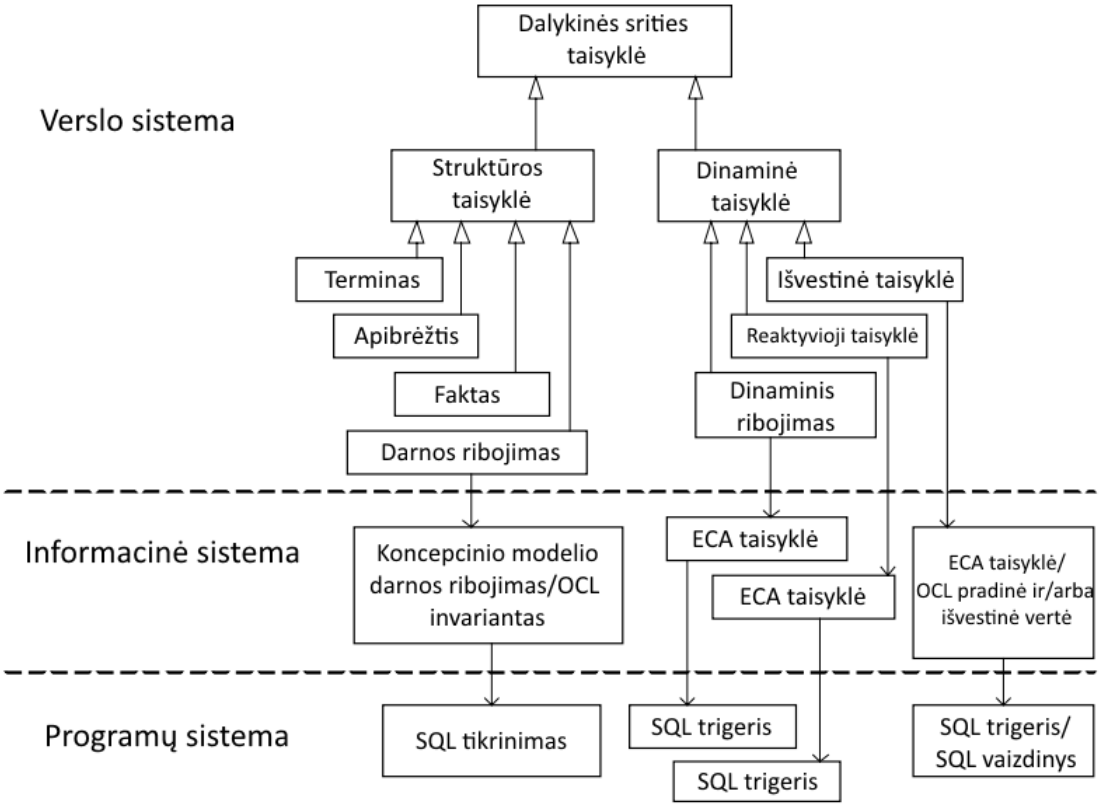
Verslo atstovo požiūriu VT yra nuoroda, lemianti arba vadovaujanti verslo elgsenai ir funkcionavimui [1]. Šiuo atveju VT yra tiesiogiai susijusios su žmonių veikla ir joms būdingi du svarbūs bruožai: norint taikyti VT turi būti dedamos sąmoningos pastangos, ir turi būti suvokiamos pasekmės, kylančios iš VT nesilaikymo. Jeigu nagrinėsime VT iš informacinių sistemų pusės, tokiu atveju tai yra teiginys, kuris apibrėžia verslo struktūrą ir kontroliuoja verslo procesus [2]. Remiantis šiuo požiūriu, reikia pabrėžti, kad VT realizuojamos informacinėje sistemoje, kurioje saugomi duomenys apie verslo procesuose dalyvaujančias esybes, ir todėl jos yra taikomos duomenims apdoroti.

1.1.2. Verslo taisyklių klasifikavimas

Nors VT galima klasifikuoti įvairiai, tačiau šiame darbe daugiau į jas žvelgsime iš informacinių sistemų kūrėjo požiūrio ir toliau, remdamiesi šaltiniuose pateikta medžiaga [3, 4], aptarsime kelis iš galimų klasifikavimo variantų.

VT gali būti skirstomos į tris abstrakcijos lygius:

- Verslo sistema (angl. *Business system*). Verslo sistema apibūdina dalykinę sritį, kurioje VT yra naudojamos. Šiame lygyje VT yra suvokiamos kaip teiginiai, kurie apibrėžia verslo ribojimus. Jos gali būti suformuluotos naudojant sintaksę, kuri yra panaši į natūralią kalbą t. y. neformaliai. Pavyzdžiui, „`Klientas negali gauti nuolaidos pirkinių krepšeliui, jeigu krepšelyje esančių pirkinių kiekis mažesnis už 10`";
- Informacinė sistema (angl. *Information system*). Šiame lygyje VT yra užrašomos formaliai ir apibūdina informacijos apdorojimą informacinėje sistemoje. Pavyzdžiui, formalia objektų ribojimo kalba OCL (*angl. Object Constraint Language*) užrašyta taisyklė „`context Turnynas inv: self.gautiMinimaluZaidėjuSkaicių() > 10`" apriboja turnyre dalyvaujančių minimalų žaidėjų skaičių. Tokios taisyklės yra vienareikšmės ir nepalieka vietos interpretacijoms;
- Programų sistema (angl. *Software system*). Šiame lygyje VT įgauna programinio kodo formą. Paprasčiausias pavyzdys yra trigeris duomenų bazių valdymo sistemoje kaip Microsoft SQL Server.



1 pav. Verslo taisyklių klasifikacija [3] (iš anglų kalbos versta autoriaus)

Taip pat VT galima klasifikuoti pagal jų įgyvendinimo būdą:

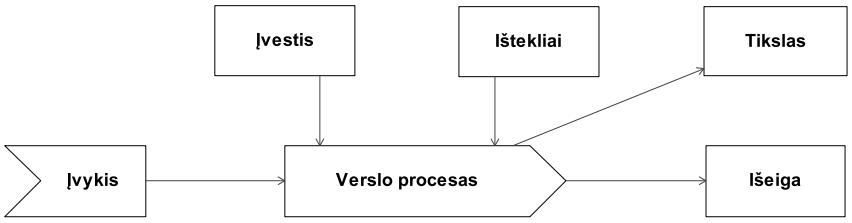
- Struktūros taisyklės (angl. *Structural rules*). Jas sudaro esybės, esybių apibrėžtys, faktai bei darnos ribojimai. Šio tipo VT įgyvendinamos jau semantinio modeliavimo lygyje, pavyzdžiui, kuriant esybių–ryšių (ER) ar UML klasių modelius. Dėl to šių taisyklių įgyvendinimas yra griežtai apibrėžtas ir jas lengviau suvokti kaip konceptus, verslo žodyno sąvokas:
  - Terminai (angl. *Terms*);
  - Apibrėžtys (angl. *Definitions*);
  - Faktai (angl. *Facts*);
  - Darnos ribojimai (angl. *Integrity constraints*).
- Dinaminės taisyklės (angl. *Dynamic rules*). Šios taisyklės įprastai yra išreiškiamos ECA forma. ECA forma, kaip galima suprasti iš pavadinimo, yra sudaryta iš trijų dalių: įvykio (angl. *event*), sąlygos (angl. *condition*) ir veiksmo (angl. *action*). Įvykus įvykiui tikrinama, ar yra tenkinama sąlyga. Jeigu sąlyga yra tenkinama, tada atliekamas veiksmas. Kaip minėta aukščiau, vienas iš būdų tokias taisykles apibrėžti programų sistemos lygyje yra naudojant trigerius. Tokios taisyklės dar gali būti skirstomos į:
  - Išvestines taisyklės (angl. *Derivation rules*). Šių taisyklių pagalba gaunamos naujos žinios iš jau turimų žinių taikant matematinius skaičiavimus ar darant loginias išvadas iš faktų;

– Reaktyviąsias taisykles (angl. *Reaction rules*). Šios taisyklės, priklausomai nuo to, ar yra tenkinamos tam tikros sąlygos, įgalioja atlikti tam tikrus veiksmus;
– Dinaminius ribojimus (angl. *Dynamic constraints*). Šie ribojimai draudžia sistemai pereiti iš vienos būsenos į kitą.

1.2. Verslo procesai

1.2.1. Verslo procesų apibrėžtis

Kiekvienoje verslo įmonėje ar organizacijoje vyksta įvairūs procesai. Šiuo metu yra keletas plačiau žinomų ir visuotinai priimtų verslo proceso apibrėžimų. Pavyzdžiui, pagal Thomasą Hayesą Davenportą verslo procesas yra tam tikra tvarka išrikiuotų veiklų rinkinys, turintis pradžią, pabaigą, įvestį ir išeigą, kurį įvykdžius gaunamas konkretus, į klientą orientuotas rezultatas. Tuo tarpu Michaelas Hammeris ir Jamesas Champas verslo procesą apibūdina kaip veiklų rinkinį, kuris naudoja įvesties duomenis bei išteklius ir sukuria kliento požiūriu vertingą išeigą. Kaip galima pastebėti, apibrėžimai yra panašūs ir jiems būdingi tam tikri pasikartojantys konceptai.



2 pav. Verslo proceso schema

2 paveiksle pavaizduota apibendrinto verslo proceso schema, toliau paaiškinamos minėti konceptai ir proceso savybės [5]:

1. Turi aiškiai apibrėžtą tikslą (angl. *Goal*). Tikslas yra suprantamas kaip nauda, kurios siekiama gauti įgyvendinus verslo procesą;
2. Turi įvestį (angl. *Input*). Įvestis yra suprantama kaip informacija, kurios pagalba galima atlikti veiklas. Vienas iš įvesties bruožų tai, kad ji yra nesunaudotina;
3. Turi išvestį (angl. *Output*). Išvestis yra suprantamas kaip verslo proceso rezultatas – verslo požiūriu vertę turintis objektas, kuris gali būti fizinis ar koncepcinis;
4. Naudoja išteklius (angl. *Resource*). Ištekliai yra suprantami kaip fiziniai objektai, kurie, priešingai nei įvestis, gali būti baigtiniai ir sunaudoti įgyvendinus verslo procesą;
5. Sudarytas iš tam tikros veiklų sekos, kurios atliekamos pagal tam tikrą tvarką;
6. Gali apimti daugiau kaip vieną įmonės organizacinį vienetą;
7. Lemia klientui naudingą rezultatą.

Šiame darbe, interpretuodami darbus [6, 7], teigsime, kad verslo procesas yra logiškai ar priežastiniais ryšiais išrikiuotų veiklų seka, turinti įvestį, išeigą, tikslą bei naudojanti išteklius,

kurios atlikimas lemia klientui naudingą rezultatą.

1.2.2. Verslo procesų klasifikavimas

Verslo procesai gali būti klasifikuojami įvairiai. Toliau pateiksime jų galimą klasifikaciją remiantis darbais [6, 8, 9]. Iš verslo pusės, procesai gali būti skiriami į šiuos tipus:

- Vidiniai verslo procesai (angl. *Internal business processes*).
  - Palaikantys verslo procesai (angl. *Supporting business processes*). Tai tokie procesai, kurių klientai nepastebi, tačiau padeda atlikti pagrindines verslo funkcijas, pavyzdžiui, apskaita;
  - Organizaciniai verslo procesai (angl. *Organization Management business processes*). Tai tokie procesai, kurie yra svarbūs siekiant palaikyti efektyvų verslo valdymą, pavyzdžiui, strategijos, tikslų planavimas.
- Išoriniai verslo procesai (angl. *External business processes*).
  - Pirminiai verslo procesai (angl. *Primary business processes*). Tai tokie procesai, kurie sukuria galutinį produktą ar paslaugą ir yra orientuoti į klientų poreikių tenkinimą. Kitaip sakant, šie procesai atlieka pagrindines verslo funkcijas ir kuria pridetinę vertę, pavyzdžiui, pardavimai, gamyba.

Šio darbo kontekste svarbesnis verslo procesų klasifikavimas pagal lankstumą:

- Statiniai procesai (angl. *Static processes*). Tai tokie procesai, kurie nekinta ir juos sudaro iš anksto griežtai apibrėžta, priežastiniais ryšiais susieta veiklų seka;
- Dinaminiai procesai (angl. *Dynamic processes*). Tai tokie procesai, kurie kinta eigos metu, pavyzdžiui:
  - Taisyklėmis grindžinami dinaminiai verslo procesai (angl. *Rule based dynamic processes*);
  - Kontekstui jautrūs dinaminiai verslo procesai (angl. *Context sensitive dynamic processes*).

1.2.3. Dinaminiai verslo procesai

Norint modeliuoti ir simuliuoti dinaminius verslo procesus (toliau DVP), svarbu juos tinkamai apibrėžti. Nors šiuo metu tokiems procesams skiriamas dėmesys auga, kol kas jų apibrėžtis nėra vienareikšmė. Darbe [10] autoriai yra suformulavę reikalavimus, kuriuos turėtų tenkinti DVP. Pagal šiuos reikalavimus, DVP turi būti būdingas jį sudarančių komponentų kitimas jo vykdymo metu. Pavyzdžiui, tokį procesą sudarančių veiklų vykdymo seka negali būti apibrėžta iš anksto, o turi būti suformuota ir keičiama proceso eigos metu. Be to, DVP turėtų kisti priklausomai nuo kintančio konteksto. Paties konteksto kitimo trukmė turėtų būti žymiai trumpesnė, lyginant su visa proceso trukme. Svarbiausia, kad DVP kitimas gali būti sukeltas bet kuriuo laiko momentu.

Šiame darbe naudosime Justino Trinkūno suformuluotą DVP apibrėžimą: dinaminiai verslo procesai realizuoja tokius verslo procesų modelius, kuriuos sudarantys komponentai (sąlygų rinkinys, veiklų rinkinys, veiklą sudarantis turinys, veiklų seka, sprendimų mazgai, dalyviai), esant konteksto kitimui, gali kisti (jeigu reikia, pakankamai greitai procesų eigos metu) [11] (iš anglų kalbos versta autoriaus).

Be to, [8, 12] darbuose aptariami verslo procesų dinamiškumo lygiai, kurie skirtingai realizuojami informacinėse sistemose. Dinamiškumo lygiai pateikti nuo žemiausio iki aukščiausio:

1. Žmogus ar automatizuota sistema, naudodama sprendimų mazgus, pasirenka, kokią veiklą reikia atlikti;
2. Veiklų rinkinys keičiamas naudojant skirtingus galimus šablonus;
3. Procesas yra orientuotas į tikslą ir veiklų rinkinys keičiamas automatiškai proceso eigos metu.

Kontekstui jautrūs DVP gali kisti proceso eigos metu priklausomai nuo kintančio konteksto. Kontekstas yra tam tikra informacija, turinti įtakos verslo proceso vykdymui. Formaliai kontekstą galima apibrėžti kaip minimalų kiekį kintamųjų, kurie apibūdina visą verslo procesų projektavimui ir vykdymui svarbią informaciją [8]. Kontekstas gali būti skirstomas į:

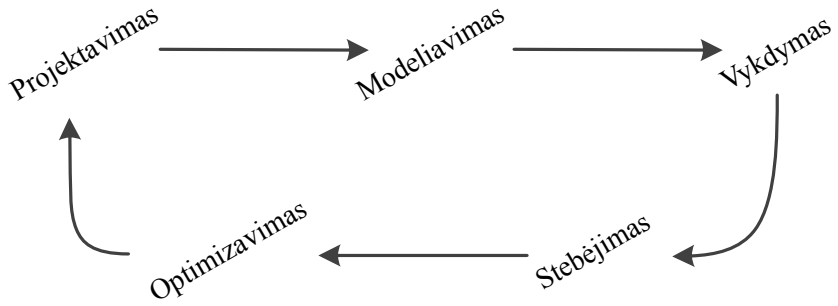
- Vidinį kontekstą (angl. *Internal context*). Vidinį kontekstą apibūdina sistemos ar organizacijos ištekliai, jų būsena;
- Išorinį kontekstą (angl. *External context*). Išorinį kontekstą apibūdina sistemos ar organizacijos aplinka, kuri išreiškiama naudojant kintamuosius ar taisykles. Išorinio konteksto pavyzdys – verslo organizacijos pelno priklausomybė nuo oro sąlygų ar šalies mokestinės politikos.

Verslo procesas, jį sudarančios veiklos gali daryti įtaką vidiniam, bet ne išoriniam kontekstui.

1.2.4. Verslo procesų valdymas

Verslo procesų valdymą galima apibrėžti kaip sistemingą organizacijoje vykstančių procesų stebėjimą, įgyvendinimą ir tobulinimą. Kiekvienos verslo įmonės kritinis sėkmės veiksnys yra gebėjimas apjungti žmones, informaciją ir procesus. Ši užduotis tampa dar sudėtingesnė, jeigu informacija apdorojama keliose sistemose, o procesai nėra statiniai ir kinta eigos metu. Tinkamas verslo procesų valdymas padeda verslo atstovams ir IT specialistams kartu bendradarbiauti ir sukurti efektyviai veikiančias informacines sistemas, kuriose būtų integruoti minėti elementai [13].

Galima išskirti penkis etapus, kurie tarpusavyje sudaro ciklą ir kartu apibūdiną verslo procesų valdymą:



3 pav. Verslo procesų valdymo gyvavimo ciklas

1. Projektavimas (angl. *Design*). Šiame etape identifikuojami esami procesai ir numatomi nauji procesai;
2. Modeliavimas (angl. *Model*). Šiame etape analizuojamas identifikuotų verslo proceso veikimo principas ir kūriami nauji procesai;
3. Vykdymas (angl. *Execute*). Šiame etape verslo procesai yra įgyvendinami;
4. Stebėjimas (angl. *Monitor*). Šiame etape analizuojamas verslo procesų vykdymas siekiant identifikuoti jų trūkumus;
5. Optimizavimas (angl. *Optimize*). Šiame etape verslo procesai yra tobulinami siekiant pašalinti identifikuotus trūkumus.

Tokiu būdų verslo procesai gali būti nuolatos perprojektuoti ar automatizuoti, o tai ypač svarbu žinant, kad šiuolaikinis verslas yra lankstus ir jo reikalavimai kinta.

1.3. Verslo procesų modeliavimas

1.3.1. Verslo procesų modeliavimo apibrėžtis

Modelis yra abstrakcija, kuri leidžia susikoncentruoti į esmines, labiausiai dominančias nagrinėjamų objektų savybes, tuo metu ignoruoti tai, kas nereikšminga ar bendra su kitais objektais. Modeliai dažniausiai apibrėžiami taisyklių rinkiniu, todėl jų dėka tampa žymiai paprasčiau perteikti informaciją kitiems asmenims. Verslo analitikas, norėdamas pagerinti komunikaciją su verslininku, kuria verslo procesų modelius. Šie modeliai yra atvaizduojami grafiniu pavidalu ir yra suprantami abiem pusėms. Tokiu būdu užtikrinama, kad verslo procesai visų šalių bus interpretuojami vienodai, palengvinamas procesų optimizavimas bei išvengiama projektavimo klaidų [7]. Šiame darbe laikysime, kad verslo procesų modeliavimas – tai tam tikrų verslo organizacijoje vykstančių procesų, juos sudarančių veiklų srautų grafinis vaizdavimas siekiant juos analizuoti ar tobulinti.

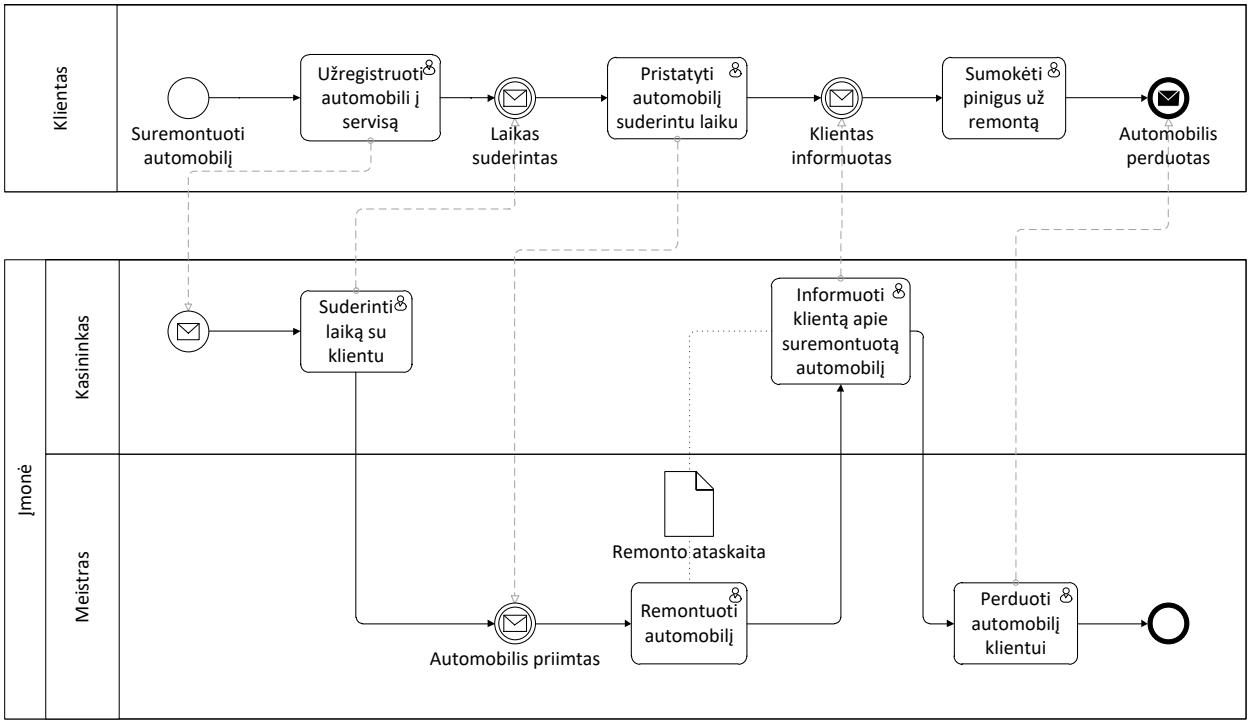
Realybėje vykstančių verslo procesų modeliavimas nėra paprastas uždavinys. Daugeliui šiuolaikinių verslo procesų būdingas dinamiškumas ir priklausomybė nuo aplinkos sąlygų, o ne griežtas apibrėžimas, todėl tokių procesų modeliavimas ar simuliacija tampa komplikuota.

Dėl to siekiama apibrėžti tokią verslo procesų modeliavimo kalbą, kuri gebėtų išreikšti sudėtingus procesus, tačiau kartu išliktų suprantama ir verslo atstovams [14]. Toliau pateiksime pagrindinius kriterijus, kuriuos turėtų atitikti modeliavimo kalba:

- Kalba turi būti pakankamai raiški. Kalbos raiškumas yra suvokiamas kaip gebėjimas atvaizduoti mus dominantį objektą ar įvykį;
- Kalba turi būti intiutyviai suprantama. Kuo kalba yra suprantamesnė, tuo lengviau perteikti informaciją specifinių modeliavimo žinių neturinčiam asmeniui, pavyzdžiui, verslo atstovui;
- Kalba turi būti formali ir vienareikšmė;
- Kalba turi būti pakankamai paplitusi ir naudojama.

1.3.2. BPMN modeliavimo kalba

Viena iš populiariausių verslo procesų modeliavimo kalbų yra BPMN (angl. *Business Process Model and Notation*), kurios pagalba galima grafiškai išreikšti verslo proceso veiklų srautą, jų atlikimo logiką. Ši kalba buvo sukurta taip, kad jos notaciją suprastų tiek verslininkai, tiek informacinių sistemų kūrėjai, tačiau kartu išliktų pakankamai formali ir raiški. Kitas privalumas – BPMN susieja procesų analizę su procesų įgyvendinimu, nes palaiko vidinį formalų modelį, užtikrinantį vykdomųjų procesų kalbų, pavyzdžiui BPEL (angl. *Business Process Execution Language*), generavimą. Dėl šių priežasčių BPMN paplito bei tapo verslo procesų modeliavimo kalbų *de facto* standartu.



4 pav. Verslo proceso BPMN modelio pavyzdys

BPMN modeliavimo kalboje naudojami elementai yra lengvai atskiriami vienas nuo kito.

Patys elementai suskirstyti į bazines kategorijas, kurios išreiškia skirtingas tų pačių elementų variacijas. Pagrindiniai elementai yra šie:

- Srauto objektai (angl. *Flow objects*).
  - Įvykis (angl. *Event*). Skiriami į tris tipus: pradžios, tarpinį ir pabaigos įvykį. Žymimas apskritimu;
  - Užduotis (angl. *Task*). Užduotis yra smulkiau neskaidoma (atominė) veikla, kuri atliekama proceso vykdymo metu. Žymimas stačiakampu suapvalintais kampais;
  - Procesas (angl. *Process*). Procesas, skirtingai nuo užduoties, yra sudarytas iš daugiau kaip vienos veiklos. Žymimas taip pat stačiakampiu suapvalintais kampais, tačiau turi pliuso ženklą;
  - Sprendimas (angl. *Gateway*). Sprendimas yra vieta, kurioje srautas išsišakoja į keletą ar daugiau alternatyvių kelių arba, atvirkščiai – susilieją į vieną srautą. Žymimas taisyklingu rombu;
- Sujungimo objektai (angl. *Connecting objects*).
  - Sekos srautas (angl. *Sequence flow*). Sekos srautas nurodo tvarką, pagal kurią atliekamos procesą sudarančios veiklos. Žymimas ištisine linija su juoda rodykle;
  - Pranešimų srautas (angl. *Message flow*). Pranešimų srautas nurodo pranešimų siuntimą ir gavimą tarp kelių proceso esybių. Žymimas punktyrine linija su balta rodykle;
  - Asociacijos srautas (angl. *Association flow*). Asociacijos srautas naudojamas prijungti artefaktą prie srauto objektų bei nurodyti veiklų įėjimus ir išėjimus. Žymimas taškine juoda rodykle.
- Takeliai (angl. *Swimlanes*). Takeliai naudojami proceso dalyviams ir veiklų rinkiniams atskirti;
- Artefaktai (angl. *Artifacts*). Artefaktai yra fiziniai ar abstraktūs objektai, kurie teikia papildomą informaciją apie procesą, pavyzdžiui, dokumentas, produktas, duomenys.

1.4. Verslo procesų simuliacija

1.4.1. Verslo procesų simuliacijos apibrėžtis

Verslo procesų simuliacija – tai veiksmas arba procesas, kuris sudaro sąlygas taikyti verslo procesų modelį konkrečiam tikslui arba tyrimui atlikti [6]. Šio darbo kontekste simuliacija yra verslo proceso vyksmo atkartojimas naudojant programinius įrankius, kurios tikslas yra įvertinti proceso elgseną. Darbe [7] yra pateiktas keletas simuliacijos panaudojimo pavyzdžių kaip proceso trukmės ar kainos įvertinimas, personalo darbotvarkės koregavimas, proceso pokyčio stebėjimas priklausomai nuo kintančių įvesties parametrų, ar skirtingų procesų bei proceso scenarijų tarpusavio lyginimas.

Norėdami simuliuoti verslo procesą, reikia atsižvelgti į įvairius parametrus, pavyzdžiui, planuojamą vykdymo laiką, naudojamus išteklius, o pati simuliacija, siekiant gauti patikimus rezultatus, turi būti vykdoma daug kartų [15]. Tam turi būti sukurtas simuliacinis modelis, kuriam įprastai būdingi trys esminiai komponentai [16]:

- Valdymo srautas (angl. *Control-flow*). Šis komponentas apibūdina procesą sudarančių veiklų loginę atlikimo tvarką;
- Duomenys, taisyklės (angl. *Data, rules*). Šis komponentas atsakingas už informacijos apsikeitimą tarp veiklų, sprendimų priėmimą;
- Ištekliai (angl. *Resources*). Šis komponentas apibūdina išteklių parametrus, jų paskirstymą per atitinkamas veiklas.

Kadangi simuliacija yra proceso vyksmo atkartojimas, tai dažniausiai simuliaciniai modeliai yra labai detalūs ir apima daug įvairių aspektų. Darbe [7] yra aptarti reikalavimai, kurie turėtų būti taikomi verslo procesų simuliacijos varikliui. Svarbiausi iš jų:

- Ištekliai atvaizduojami grafiškai ir jų būsenos stebimos bet kuriuo metu;
- Ištekliui galima priskirti kainą, kiekį, vaidmenį, prieinamumą;
- VT turėtų apibrėžti išteklių priskyrimą veikloms. Vienai veiklai galima priskirti keletą išteklių;
- Veiklai galima priskirti jų vykdymo prioritetus ir trukmes. Prioritetai nustatomi remiantis VT;
- Simuliacijos variklis turi teisingai interpretuoti įvykius bei kontekstą.

Verslo procesų simuliacija gali būti skiriama į šiuos tipus [17]:

- Dinaminė simuliacija (angl. *Dynamic simulation*). Tai tokia simuliacija, kurioje naudojami kintamieji kinta laike priklausomai nuo įvesties;
- Stochastinė simuliacija (angl. *Stochastic simulation*). Tai tokia simuliacija, kurioje naudojami kintamieji yra atsitiktiniai dydžiai, kurie įgyjami su tam tikra tikimybe;
- Diskrečiųjų įvykių simuliacija (angl. *Discrete-event simulation*). Tai tokia simuliacija, kuri imituoja proceso kitimą laike kaip atskirų įvykių seką.

1.4.2. Simuliacijos privalumai ir trūkumai

Nors šiuo metu verslo procesų simuliacija dažniausiai kol kas taikoma tik didelėse organizacijose, šio metodo populiarumas auga. Tai iš dalies lemia ir tai, kad vis labiau prieinami tampa simuliacijos įrankiai, o dalis jų yra nemokami. Galima teigti, kad verslo procesų simuliacija vis dar yra pradiniame etape, o jos tikrasis potencialas ir nauda atsiskleis ateityje. Šaltiniuose [17, 18] yra pateikti simuliacijos privalumai ir trūkumai, į kuriuos reikėtų atsižvelgti prieš sprendžiant, ar šį metodą verta taikyti:

- Simuliacija yra lanksti. Sunku būtų rasti tokį reiškinį, kurio nebūtų galima bent jau apytiksliai įvertinti naudojant simuliaciją. Žinoma, gautų rezultatų tikslumas priklauso nuo užduoties sudėtingumo ir sukurto simuliacino modelio tikslumo. Tobulėjant skaičiavimo technikai vis daugiau problemų, kurios anksčiau atrodė neišsprendžiamos, simuliacijos dėka tampa įveikiamos;
- Problemos sprendimas analitiniu būdu taikant matematines išraiškas ne visada yra praktiškas ar įmanomas. Kartais tokių problemų sprendimas skaitmeniškai naudojant simuliaciją yra paprastesnis būdas, teikiantis pakankamai tikslius rezultatus;
- Simuliacijos pagalba galima sąlyginai greitai įvertinti įvairius nagrinėjamos problemos aspektus, metrikas. Prieš pokyčius įgyvendinant tikrame pasaulyje, jie gali būti analizuojami atliekant simuliaciją. Simuliacija suspaudžia laiką – tai, kas realybėje užtruktų keletą mėnesių, simuliacijos dėka gali būti įvertinta per kelias sekundes;
- Simuliacijos pagalba paprasta perteikti informaciją. Jeigu simuliacija atliekama naudojant programinę įrangą, dažnai kartu su rezultatais pateikiamas grafinis nagrinėjamos problemos atvaizdavimas.

Simuliacijos trūkumai:

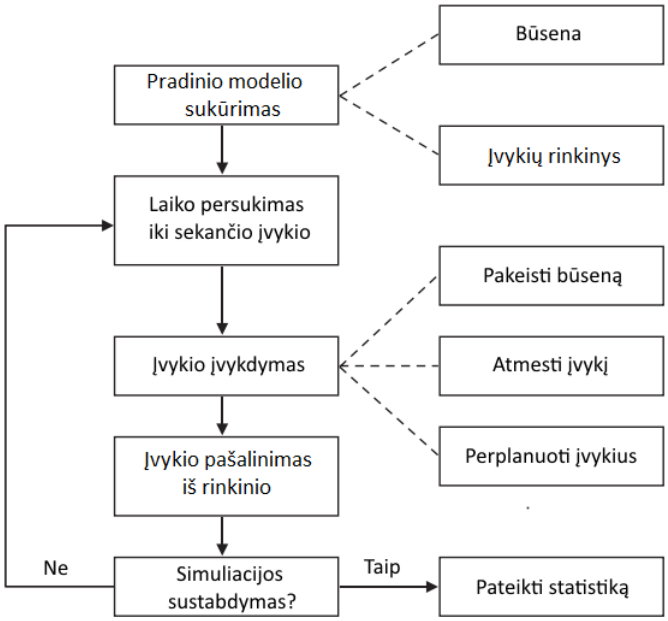
- Norint gauti pakankamai tikslius simuliacijos rezultatus, reikia įdėti daug pastangų. Pavyzdžiui, simuliaciją reikia kartoti ne vieną kartą. Taip pat paties simuliacinio modelio sukūrimas užima daug laiko;
- Ne visada pavyksta paaiškinti rezultatus, kurie gauti naudojant simuliaciją. Kartais gali būti daromos klaidingos prielaidos, kurios tikrovėje gali atnešti didelių nuostolių;
- Simuliacijos rezultatai gali visai neatikti realybės. Vyrauja nuomonė, kad simuliaciniai modeliai supaprastina tikrovę. Pavyzdžiai, simuliuoti žmogaus įtaką verslo procesui dažnai tampa pernelyg sudėtingas uždavinys;
- Verslo organizacijos, ypač mažesnės, linkusios skeptiškai vertinti simuliaciją kaip pernelyg brangų laiko ir kainos atžvilgių metodą.

1.4.3. Diskrečiųjų įvykių simuliacija

Diskrečiųjų įvykių simuliacija – tai yra procesas, kurio metu sistemos elgsena apibrėžiama kaip išrikiuota diskrečių įvykių seka [19]. Taikant diskrečiųjų įvykių simuliaciją verslo procesu atveju, laikoma, kad laikas kinta diskrečiai, o procesų būsenos keičiasi šuoliškai. Čia stebimos verslo procesą apibrėžiančių kintamųjų reikšmės laikui bėgant ir jų pokytis, kurį sukelia įvykstantys įvykiai. Kintamieji, pavyzdžiui, gali apibrėžti išteklių prieinamumą ar kiekį. Dėl to norint užtikrinti efektyvų simuliacijos vykdymą ne tiek svarbu fiksuoti laiko tėkmę, bet reikia įsiminti įvykusius bei eilėje stovinčius įvykius. Taip pat derėtų iš anksto nustatyti simuliacijos pradžios ir pabaigos įvykius. Diskrečiųjų įvykių simuliacijos pagalba galime „suspausti laiką“, nes tik įvykiai gali keisti verslo proceso būseną, o laiko tarpai tarp jų mūsų nedomina [17].

Diskrečiųjų įvykių simuliacijos modeliai pagal įgyvendinimą gali būti klasifikuojami [20]:

- Tikimybinis (angl. *Stochastic*). Tai toks modelis, kuriame naudojamos tikimybių pasiskirstymo funkcijos, teikiančios atsitiktines reikšmes;
- Deterministinis (angl. *Deterministic*). Tai toks modelis, kuris prie vienodų modelio įvesties parametrų teikia vienodas reikšmes.



5 pav. Diskrečiųjų įvykių simuliacijos algoritmas [17] (iš anglų kalbos versta autoriaus).

Toliau, remiantis [17], yra pateiktas galimas diskrečiųjų įvykių simuliacijos algoritmas (5 pav.):

1. Pradinio simuliacinio modelio sukūrimas ir įvykių rinkinio apibrėžimas;
2. Laiko persukimas iki sekančio įvykio. Kaip minėta anksčiau, diskrečiųjų įvykių simuliacija „suspaudžia laiką“, laiko tarpas tarp įvykių yra nesvarbus;
3. Įvykio įvykdymas. Po įvykio įvykdymo gali prireikti pakeisti dabartinę proceso būseną, atmesti ar perplanuoti kitus, eilėje laukančius įvykius;
4. Įvykio pašalinimas iš rinkinio;
5. Galimas simuliacijos sustabdymas, jeigu pasiekta tam tikra sąlyga, pavyzdžiui, iteracijų skaičius, simuliacijos laikas;
6. Svarbų simuliacijos rezultatų, pavyzdžiui, statistikos, pateikimas ir analizė, duomenų išsaugojimas ir t. t.

1.4.4. Verslo procesų ištekliai ir jų simuliacija

Bet kurių verslo procesų veiklų vykdymas dažniausiai reikalauja tam tikrų išteklių. Ištekliai yra susieti su konkrečiomis verslo proceso veiklomis ir be jų realybėje neįmanomas

joks verslo proceso įgyvendinimas. Verslo proceso išteklius galime įvardinti kaip bet kokius objektus, naudojamus verslo organizacijoje proceso vykdymo metu ir lemiančius išeigą. Kaip minėta anksčiau, ištekliai apibūdina vidinį organizacijos kontekstą, gali būti baigtiniai ir sunaudoti proceso vykdymo metu.

Remiantis [20, 21, 22] darbais, ištekliai gali būti klasifikuojami:

- Sunaudojami ištekliai. Tai tokie ištekliai, kurių kiekis proceso vykdymo metu mažėja:
  - Energija;
  - Žaliavos.
- Panaudojami ištekliai. Tai tokie ištekliai, kurie gali būti užimti tam tikros veiklos, o po jos įvykdymo – atlaisvinami ir toliau naudojami kitose veiklos. Šių išteklių trūkumas gali ženkliai daryti įtaką galutinei proceso kainai ar trukmei:
  - Žmogiškieji ištekliai;
  - Įrenginiai.
- Dalomi ištekliai:
  - Fiziškai dalomi ištekliai. Tai tokie ištekliai, kurie dalomi į dalis keičiant fizinę struktūrą;
  - Funkciškai dalomi ištekliai. Tai tokie ištekliai, kurių funkcijos gali būti dekomponuotos.
- Nedalomi ištekliai.
- Diskretūs ištekliai. Tai tokie ištekliai, kuriuos galima įvertinti tik kiekybiškai.
- Nuolatiniai ištekliai. Tai tokie ištekliai, kurių neįmanoma įvertinti kiekybiškai. Įprastai jų būsena apibrėžiama naudojant ribotą skaičių sąvokų;
- Pavieno naudojimo išteklius. Tai tokie ištekliai, kurie tuo pat metu gali būti naudojami tik vienos veiklos;
- Lygiagretaus naudojimo išteklius. Tai tokie ištekliai, kurie tuo pat metu gali būti naudojami kelių veiklų ar procesų;
- Neriboto prieinamumo ištekliai. Tai tokie ištekliai, kurių naudojimas nėra apribotas;
- Riboto prieinamumo ištekliai. Tai tokie ištekliai, kurie tam tikru metu negali būti naudojami.

Remiantis [20, 21, 22] darbais, ištekliams būdingos šios charakteristikos:

- Ištekliaus tipas. Pavyzdžiui, sunaudojamas ar panaudojamas.
- Ištekliaus matavimo vienetas:
  - Kiekybinis matavimo vienetas;
  - Kokybinis matavimo vienetas.
- Ištekliaus kiekis:

  - Suskaičiuojamas kiekis. Tai toks kiekis, kuris gali būti įvertinimas kiekybiškai;
  - Nesuskaičiuojamas kiekis. Tai toks kiekis, kuris gali būti įvertinimas kokybiškai.
- Ištekliaus kaina;
- Ištekliaus prieinamumas laiko atžvilgiu;
- Ištekliaus būsena. Ištekliaus būsena priklauso nuo jo prigimties ir dalykinės srities, kurioje jis yra naudojamas. Galimų būsenų pavyzdžiai simuliaciniame modelyje [20]:
  - Laisvas. Tai yra pradinė išteklio būsena;
  - Užimtas. Esant šioje būsenoje ištekliu atlieka užduotį;
  - Atlaisvinamas. Šiuo atveju ištekliui nustatomas perėjimo laikas tarp veiklų.
- Alternatyvus išteklius. Jeigu esamas išteklius yra neprieinamas, jis gali būti pakeistas alternatyviu ištekliu.
- Išteklius kaip komponentas. Išteklius gali būti fizinė arba funkcinė kito išteklio sudedamoji dalis;
- Ištekliaus vaidmuo. Atsižvelgiant į susietą veiklą, ištekliu gali atlikti skirtingus vaidmenis, pavyzdžiui, produktas, įrankis, žaliava, finansininkas (žmogiškųjų išteklių atveju);
- Ištekliaus priorietatas. Apibrėžiama su ištekliu susietų veiklų vykdymo tvarka pagal svarbumą;
- Ištekliaus vykdymo trukmė. Laiko tarpas, kuris skiriamas su ištekliu susietos veiklos vykdymu;
- Ištekliaus paruošimo trukmė. Laiko tarpas, kuris turi būti skirtas išteklio paruošimui prieš veiklos vykdymą.

Išteklių, ypač žmogiškųjų, apibrėžimas verslo procesų simuliaciniame modelyje kelia įvairių problemų [7, 16]:

- Išteklis gali dalyvauti keliuose procesuose. Dauguma simuliacijos įrankių vienu metu gali vykdyti tik vieną procesą ir neleidžia modeliuoti tokios situacijos, kai išteklis naudojamas tik tam tikrą laiko dalį;
- Simuliaciniai įrankiai negali įvertinti, kad tas pats išteklis gali dalyvauti keliuose to paties verslo proceso egzemplioriuose [20];
- Kai ištekliams, pavyzdžiui, žmonėms, nėra būdinga dirbti pastoviu tempu. Esant skirtingam paros metui ar žmogui susirgus, jo našumas taip pat gali būti skirtingas, o simuliacijos įrankiai to neįvertina;
- Ištekliai gali dirbti dalimis. Pavyzdžiui, kai kurie darbuotojai linkę atsakyti į elektroninius laiškus iš karto, o kiti – laukia, kol jų prisikaups ir skiria tam konkretų laiką. Tuo tarpu didžiojoje dalyje simuliaciniuose įrankiuose ištekliai yra naudojami iš karto atsiradus veiklai;

- Ištekliai, įprastai žmonės, yra aktyvūs ir dažnai savo veiklas paskirsto pagal prioritetus. Simuliaciniai įrankiai negali įvertinti, kuri veikla ar procesas yra svarbesnis ir atliekamas pirmiau, o kuris – nereikšmingas.

1.4.5. Tikslo siekiančių dinaminių verslo procesų simuliacinis modelis

Šiame darbe svarbu apžvelgti ir tikslo siekiančių verslo procesų simuliaciją. DVP naudojamų išteklių optimizavimas siekiant tam tikros jų būsenos yra konkretus išsikeltas tikslas. Darbe [23] yra pateiktas apibendrintas tikslo siekiančių DVP simuliacinis modelis. Šiame modelyje simuliacijos naudotojos turi apibrėžti verslo proceso tikslą ir veiklų rinkinį. Kiekviename simuliacijos žingsnyje turi būti įvykdytos tinkamiausios, artinančios prie iškelto tikslo, veiklos, kurias nulemia atitinkami įvykiai. Tuo metu vykdant veiklas yra naudojami ištekliai ir to pasekoje atnaujinamas kontekstas.

Verslo proceso tikslas yra tam tikra simuliacinio modelio būsena ar būsenų rinkinys [24]:

$$G = (p, A, T, C, Q)$$

, kai tikslas yra visų galimų būsenų poaibis $G \subset S$, o $S = \{s_1, s_2, ..., s_n\}$. Čia p – procesas, A – veiklų aibė, T – užduočių aibė, C – kontekstas, Q – eilė, s_i – galima būsena.

- Procesas yra sudarytas iš veiklų $p = (a \in A)$, kurios simuliacijos metu gali būti pridėtos ar pašalintos;
- Kontekstas yra sudarytas iš elementų $c \in C$, kai $c = (k, v)$. Čia k – konteksto elemento pavadinimas, v – elemento vertė. Šio darbo atveju kontekstą galima laikyti tam tikra išteklių būsenų aibę, o siekiama išteklių būsena yra konkretus tos aibės elementas;
- Veikla yra išreiškiama kaip $a = (r, T_a \in T)$, kai T_a yra užduočių, tenkinančių taisyklę r, aibė. Simuliaciniame modelyje iš esmės veiklų nėra, bet yra užduotis. Kelios užduotis gali sudaryti vieną veiklą;
- Užduotis $t \in T_a$ yra funkcija, kuri gali keisti kitų kintamųjų kaip veiklų, konteksto reikšmes;
- Eilė yra sudaryta iš elementų $Q \in Q$, kai $q = (t, d)$. Čia t – užduotis, kuri toliau turi būti vykdoma, d – užduoties vykdymo pradžios laikas. Kadangi dinaminiai verslo procesai pasižymi kitimu, tai eilė, ją sudarantys elementai, jų tvarka galėtų kisti simuliuojant tokį procesą.

Tikslo siekiančių DVP užduočių seka turi būti sudaryta ir keičiama simuliacijos vykdymo metu. Kiekviename procese užduotis yra susietos su ištekliais (kontekstu), tačiau dažniausiai jie yra riboti ir todėl jų trūkumas gali daryti įtaką proceso vykdymui. Dėl šios priežasties simuliaciniai modeliai turi gebėti reaguoti į besikeičiantį kontekstą.

1.4.6. Verslo procesų simuliacijos įrankiai

Šiame skyriuje pateikta verslo procesų simuliacijai skirtų įrankių apžvalga. Vienas iš įrankių bus naudojamas atliekant tolimesnius tyrimus.

„IBM WebSphere Business Modeler“ – tai įrankis, skirtas verslo procesų modeliavimui bei simuliacijai. Sukūrus verslo proceso modelį, galima pridėti dokumentaciją bei apibrėžti įvairias metrikas ir KPI (angl. *Key Performance Index*). Siekiant geriau suprasti procesą, „IBM WebSphere Business Modeler“ leidžia atlikti ir statistinę analizę. Norint atlikti verslo proceso simuliaciją, būtina susieti išteklius su užduotimis, apibrėžti jų trukmę bei sprendimo mazgų šakoms suteikti tikimybinius įverčius [25].

Kiti „IBM WebSphere Business Modeler“ įrankio privalumui [25, 26]:

- Leidžia rinktis tarp bazinio ir pažangaus modeliavimo režimo, todėl yra draugiškas naudotojui;
- Galimybė modeliuoti tiek laisva forma, tiek naudojant takelius;
- Vizualiai atskiriamos bei dokumentuojamos žmonių atliekamos užduotis;
- Integruojamos VT į verslo proceso modelį;
- Identifikuoja, jeigu procesas gali pasiekti netinkama būseną bei nurodo, kokie veiksmai gali būti atlikti siekiant grįžti į tinkamą būseną;
- Verslo procese naudojamų elementų kaip dokumentai, produktai modeliavimas;
- Galimas įvairaus tipo išteklių modeliavimas;
- Galimybė generuoti ir analizuoti ataskaitas, palaiko „Microsoft Excel“ formatą.

„ARIS“ yra programinių įrankių rinkinys, skirtas visam verslo valdymui palaikyti ir gali būti taikomas tiek verslo procesų modeliavimui, tiek simuliacijai. Procesų modeliavimui naudojamos BPMN diagramos, kurios vykdant simuliaciją yra transformuojamos į simuliacinį modelį. „ARIS“ fiksuoja įvairias metrikas ir KPI rodiklius, kurie padeda spręsti apie, pavyzdžiui, veiklų atlikimo trukmę, uždelsimą, išteklių naudojimą. Simuliacija bet kuriuo metu gali būti sustabdyta, pertraukta ar vykdoma pažingsniai.

Kiti „ARIS“ įrankio privalumai:

- Galimybė analizuoti ir testuoti verslo procesų modelius;
- Galimybė optimizuoti verslo proceso modelį identifikuojant silpnąsias vietas;
- Palaiko detalų išteklių planavimą ir simuliaciją;
- Daug dėmesio skiriama rizikų valdymui;
- Galimybė generuoti ataskaitas, palaiko „Microsoft Excel“ formatą.

„Simprocess“ – tai JAVA ir XML pagrindu veikiantis įrankis, skirtas verslo procesų analizei ir simuliacijai, kuriame taikomas procesų žymėjimas (angl. *process mapping*), diskrečių įvykių simuliacija, veiklomis grįstas kainos įvertinimas. „Simprocess“ gali atlikti statistinę analizę ir pagal tai tikrinti, ar proceso modelis gali būti pagerintas [27].

Kiti „Simprocess“ įrankio privalumai:

- Lengvai suprantama vartotojo sąsaja;
- Galimybė išsaugoti procesų, veiklų šablonus ir panaudoti pakartotinai;
- Palaiko išteklių simuliaciją ir seka jų būseną, leidžia apibrėžti jų tarpusavio priklausomybes.

„Simul8“ – tai sistemų simuliacijai ir optimizavimui skirtas įrankis. „Simul8“ yra vertinamas už savo lankstumą. Simuliacinį modelį galima kurti tiek vizualiai braižant schemas ir naudojant standartines piktogramas, tiek išplečiant modelį su Visual Basic kalba (šia kalba parašytas ir pats simuliacijos variklis). „Simul8“ remiasi diskrečių įvykių simuliacija. Šis įrankis taip pat palaiko išteklių simuliaciją. Čia ištekliai gali būti dalinami, nes persiskirsto tarp veiklų esant tokiam poreikiui [28].

Kiti „Simul8“ privalumai:

- Ištekliams galima priskirti prieinamumą laiko atžvilgiu;
- Galimybė apibrėžti minimalų išteklio kiekį tam tikrom veiklom. Didesnis nei reikia kiekis gali trumpinti veiklos vykdymo trukmę;
- Suderinamas su „Microsoft Excel“, „Microsoft Visio", „Microsoft Access“;
- Palaiko XML ir OLE formatą, gali naudoti SQL duomenų bazę.

„Bonita BPM“ – tai atviro kodo verslo procesų valdymo įrankis, kurį sudaro trys komponentai: „Bonita Studio“ taikomas verslo procesų modeliavimui BPMN kalba, „Bonita Engine“ yra JAVA pagrįstas simuliacijos variklis, kuris leidžia vykdyti sukurtus BPMN modelius, „Bonita Portal“ yra vartotojo sąsaja, leidžianti atlikti procesų valdymą. Remiantis [20], simuliacijos variklis nebeatnaujinamas.

Kiti „Bonita BPM“ privalumai:

- Galimybė prisijungti prie išorinių CRM ar ERP sistemų, duomenų bazių;
- Daug dėmesio skiriama verslo procesų valdymui;
- Naudojamas BPMN 2.0 standartas.

„DBPSim“ – tai Vilniaus Gedimino technikos universitete informacinių sistemų laboratorijoje sukurtas įrankis. Šiame įrankyje realizuotas konteksto, veiklų logikos keitimas simuliacijos metu, todėl yra galimybė vykdyti DVP. Įrankį sudarantys komponentai:

- Įvykių valdymo modulis;
- Taisyklių valdymo modulis;
- Simuliacijos vykdymo modulis;
- Grafinė vartotojo sąsaja.

Įrankis yra atviro kodo, todėl gali būti kūriami plėtiniai ir taikomas moksliniuose tyrimuose. Pats įrankio programinis kodas parašytas Microsoft C# kalba ir naudojant .NET 4.5 bibliotekų rinkinį. Daugiau naudingos techninės informacijos apie įrankį yra pateikta [8] darbe.

Kuriant verslo proceso simuliacinį modelį „DBPSim“ įrankiu, būtina aprašyti įvykius ir veiklas. Veiklos aprašomas remiantis ECA taisyklių šablonu. Veiklos aprašymas yra sudėtingas, nes tam naudojama Visual Basic programavimo kalba, tačiau tai suteikia lankstumo. Veikla ir įvykis gali keisti kontekstą, jį sudarančius faktus, o patį simuliacijos procesą iššaukia pirminis įvykis, kuriam įvykus tikrinamos visų veiklų sąlygos (taisyklės). Veiklos, kurių sąlygos esamuoju momentu yra tenkinamos, vykdomos. Tai kartojama tol, kol nebėra galimų vykdyti veiklų. Tokiu atveju simuliacijos atvejis užbaigiamas, iššaukiamas naujas sekantis įvykis ir ciklas yra kartojamas. Taip pat yra galimybė kviesti įvykius pagal tikrąjį laiką.

Kadangi įrankis sukurtas naudojant vidinius universiteto išteklius, jame trūksta įvairių funkcijų, kurios dažnai realizuotos komerciniuose produktuose, o pats įrankio naudojimas yra sudėtingas. Tačiau atviras kodas, išplėtimo galimybė ir motyvacija tobulinti universiteto sukurtą produktą lemia tai, kad „DBPSim“ įrankis bus taikomas šio darbo tolesniame moksliniame tyrime.

1.5. Apibendrinimas

Verslo taisyklės ir verslo procesai yra esminės sąvokos, be kurių detalesnės apžvalgos negalima kalbėti apie procesų optimizavimą. Verslo taisyklių paskirtis yra nustatyti verslo organizacijos elgesį ir vykstančius procesus. Verslo taisyklių apibrėžtis nėra standartizuota ir priklauso nuo požiūrio taško, iš kurio taisyklės nagrinėjamos. Informacinių sistemų atžvilgiu taisykles patogu nagrinėti pagal įvykis–sąlyga–veiksmas šabloną.

Tuo tarpu verslo procesai sudaryti iš veiklų rinkinio, kurios atliekamos tam tikra tvarka. Tvarką ar kitą veiklų vykdymo logiką gali nusakyti verslo taisyklės. Kiekvienas verslo procesas turi tikslą ir dažniausiai jo įgyvendinimui reikia skirti išteklių. Šio tyrimo tikslas yra susijęs su procesą sudarančių veiklų optimizavimu atsižvelgiant į išteklius.

Verslo procesų modeliavimas yra vienas iš svarbiausių verslo valdymo metodų. Procesų modelių pagalba tampa žymiai lengviau perteikti informaciją apie įmonėje vykstančius procesus, taip pat jų analizė sudaro sąlygas didinti procesų efektyumą. Tačiau toks efektyvumo didinimas gali būti atliktas tik rankiniu būdu ir žymiai priklauso nuo analitikų patirties. Dažnai sunku prognozuoti, kas įvyks įgyvendinus pokyčius realybėje, jeigu pakeitimai buvo suplanuoti tik ant poperiaus analizuojant verslo procesų modelius.

Dėl šios priežasties vis dažniau taikoma verslo procesų simuliacija. Šio darbo kontekste verslo procesų simuliacija yra proceso elgsenos imitavimas taikant programinę įrangą ir siekiant įvertinti proceso charakteristikas. Nors simuliacija turi daug privalumų, tačiau simuliacinio modelių, kurie teiktų „pakankamai“ tikslius rezultatus, sukūrimas yra sudėtingas uždavinys. Šiame tyrime simuliacija atlieka labai svarbų vaidmenį, nes yra pagrindinis metodas, skirtas įvertinti verslo procesų elgseną.

2. VERSLO PROCESŲ OPTIMIZAVIMAS

2.1. Verslo procesų optimizavimo apibrėžtis

Verslo proceso optimizavimas – tai automatiniu būdu atliekamas verslo proceso patobulinimas, kuris paremtas kiekybinių įverčių naudojimu [29]. Verslo proceso optimizavimas įprastai prasideda nuo proceso modelio analizės ir šio darbo kontekste patobulinimas yra suprantamas kaip verslo proceso modelyje naudojamų išteklių pertvarkymas atsižvelgiant į siekiamą jų būseną.

Priklausomai nuo to, kokią modeliavimo kalbą naudojant yra sukurtas modelis, gali būti taikomi atitinkami proceso analizės metodai [29]:

- Diagramų modeliai (angl. *Diagrammatic models*), pavyzdžiui, blokinė srautų diagrama (angl. *flowchart*). Šiomis kalbomis sukurti verslo procesų modeliai tinka greitam ir daugeliui suprantamam procesų atvaizdavimui, tačiau pagrindinis jų trūkumas – formalaus pagrindo nebuvimas, todėl jų analizė bei kartu optimizacija yra subjektyvi ir žymiai priklauso nuo analitiko įgūdžių. Čia įprastai taikomas paprasčiausias analizės metodas – peržiūra (angl. *observational analysis*). Atlikus analizę, optimizavimas gali būti vykdomas panaikinant veiklas, kurios nekuria pridėtinės vertės, suprastinant sudėtingas veiklas, automatizuojant veiklas [29];
- Matematiniai/formalūs modeliai (angl. *Mathematical/formal models*), pavyzdžiui, Petri net tinklai. Šiomis kalbomis sukurti procesų modeliai yra pagrįsti matematika ir matematine logika, todėl įmanomas kiekybinis įvertinimas bei objektyvi analizė. Dėl šios priežasties tokie modeliai yra tikslūs, tačiau dažnai jų kūrimas ir palaikymas reikalauja daugiau darbo, tuo metu verslo organizacijoms paprastai užtenka tik grafinio proceso atvaizdavimo t. y. diagramų modelio. Remiantis [29] darbu, formaliems procesų modeliams gali būti taikomi šie analizės metodai:
  - Validacija (angl. *Validation*);
  - Verifikacija (angl. *Verification*);
  - Vykdymo įvertinimas (angl. *Performance evaluation*).
- Verslo procesų kalbos modeliai (angl. *Business process languages models*). Šiomis kalbomis sukurti procesų modeliai gali būti vykdomi, nes yra pagrįsti XML standartu. Tokios kalbos pavyzdys yra anksčiau aptartas BPMN, kuris susietas su BPEL procesų vykdymo kalba. Šiuo atveju analizė gali būti atlikta vykdant modelio simuliaciją, kas leidžia kiekybiškai įvertinti tam tikrus proceso parametrus kaip kainą ar trukmę. Pagrindinė problema taikant simuliaciją yra proceso modelio apribojimai, pavyzdžiui, išteklių, veiklų konkurencijos atžvilgiu [7].

2.2. Verslo procesų optimizavimas vykdant simuliaciją

Kaip minėta anksčiau, vienas iš verslo procesų optimizavimo būdų yra simuliacijos taikymas. Naudojant simuliaciją galima greitai išbandyti įvairias to paties verslo proceso konfigūracijas t. y. keisti veiklų seką, išteklių kiekį ar kitus galimus kintamuosius. Gauti rezultatai lyginami tarpusavyje. Šis metodas nesunkiai realizuojamas, jeigu nagrinėjamas verslo procesas yra paprastas ir neturi daug galimų keitimo variantų. Tačiau jeigu verslo procesas sudėtingas, tai dažniausiai paieškos erdvė yra per didelė, kad būtų galima simuliacijos būdu išbandyti visus jo vykdymo scenarijus.

Šiuo atveju verslo procesą siekiama optimizuoti naudojant „protingą“, o ne pilną paiešką. Kiekvieną verslo proceso simuliacinį modelį galima laikyti funkcija, kuri turi įvestį ir generuoją išvestį (rezultatą). Įprastai šios funkcijos išreikštinis pavidalas nėra žinomas, todėl ji gali būti optimizuojama tik skaitmeniškai. Optimizacijos tikslas – surasti tokią įvestį, kuriai esant tiriama funkcija nustatytose ribose įgytų minimalią arba maksimalią reikšmę.

Knygoje [17] išskiriamos dvi galimos verslo proceso optimizavimo kryptis:

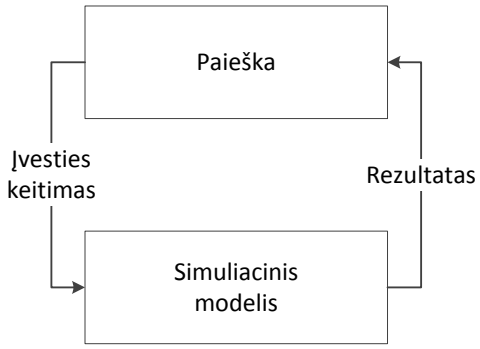
- Struktūrinė (angl. *Structural*) optimizacija, kurios metu ieškoma geriausio proceso vykdymo scenarijaus;
- Operacinė (angl. *Operational*) optimizacija, kurios metu ieškoma konkretaus proceso vykdymo scenarijaus geriausios konfigūracijos.

Verslo procese naudojamų technologijų pasirinkimas paprastai susijęs su struktūriniais pakeitimais, o pasirinktos technologijos naudojimo tvarkos apibrėžimas – operacinis pasirinkimas. Pavyzdžiui, jeigu nagrinėsime automobilių remontą apibūdinantį verslo procesą, operacinis pakeitimas galėtų būti susijęs su meistrų skaičiaus ar maksimalaus remonto termino nustatymu, o struktūrinis pakeitimas – naujos automobilių kėbulo tiesinimo technologijos taikymas.

Dinaminių verslo proceso struktūrinių pakeitimų atlikimas yra sudėtinga užduotis. Tai iš esmės yra procesą sudarančių veiklų sekos keitimas, veiklų pridėjimas ar pašalinimas proceso vykdymo metu. Tuo metu operacinė optimizacija yra paprasčiau įgyvendinama, nes verslo proceso simuliacinio modelio užduotys dažnai turi apibrėžtus parametrus, kurie, pavyzdžiui, nusako tam tikrų išteklių kiekį ar galimybę juos naudoti konkrečiu laiku. Tokiu atveju reikia keisti atitinkamų parametrų reikšmes.

Norint „protingai“ atlikti parametrų reikšmių keitimą, galima taikyti euristinius paieškos algoritmus. Euristinė paieška yra vienas iš dirbtinio intelekto metodų, kurių pagalba galima žymiai greičiau rasti apytikslį uždavinio sprendimą. Čia netaikomas pilnas būsenų (visų galimų scenarijų) perrinkimas, o naudojama euristinė įvertinimo funkcija, kuri suteikia informacijos apie paieškos algoritmo vykdomą sekantį žingsnį ir padeda sumažinti galimų pasirinkimų skaičių. Euristinė funkcija yra paremta apytiksliais įverčiais ir generuoja apytikslę reikšmę, todėl rezultato tikslumas priklauso nuo euristinės funkcijos aprašymo ir neužtikrina globalaus ekstremumo taško radimo, ypač jeigu paieška vykdoma nepakankamai ilgai. Dažnai pernelyg sudėtinga pasakyti, kokio tipo sprendinys yra gautas. Tačiau euristinės

paieškos pagalba galima gauti pakankamai tikslius sprendimus, kurie tenkina lūkesčius. Taikant euristinį verslo procesų optimizavimą simuliacinis modelis yra laikomas juodąja dėže (6 pav.). Įvesties reikšmė kiekvieną kartą įvykdžius simuliaciją yra keičiama atsižvelgiant į



6 pav. Simuliacinio modelio optimizavimas taikant euristinę paiešką [17]. Interpretuota autoriaus

siekiamą tikslą. Dažniausiai naujoji įvesties reikšmė gaunama kaip kelių anksčiau naudotų įvesčių modifikacijos rezultatas. Paieška yra sustabdoma pasiekus tam tikrą kriterijų, pavyzdžiui, iteracijų skaičių, paieškos laiką ar rezultatų skirtumą tarp dvejų paskutinių iteracijų rezultatų. Tokiu principu veikiantys keitimo algoritmai dar žinomi kaip evoliuciniai. Evoliucinių paieškos algoritmų pavyzdžiai yra genetinis algoritmas, tabu, simuliuojamo atkaitinimo paieška.

Norint optimizuoti DVP simuliacinį modelį keičiant įvestį, reikia:

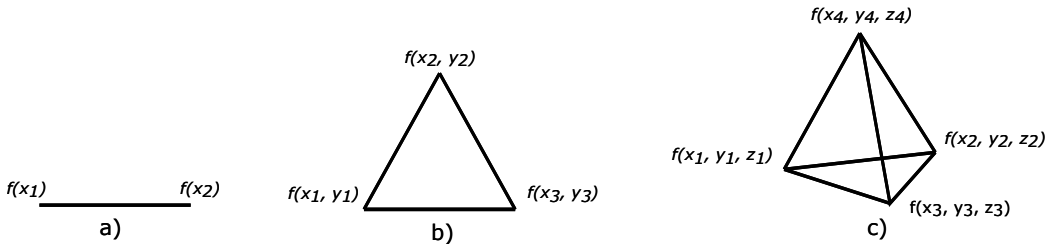
- Apibrėžti įvestį, pavyzdžiui, kaip parametrus ir jiems priskirti pradines reikšmes. Šios reikšmės vykdant paiešką yra nuolatos keičiamos. Parametrai gali apibūdinti, pavyzdžiui, darbuotojų skaičių, paslaugos kainą;
- Apibrėžti ribojimus. Ribojimai gali išreikšti sąryšius tarp skirtingų parametrų, pavyzdžiui, nustatyti, kiek daugiausiai darbuotojų gali dirbti tam tikrą darbo valandą. Ribojimas gali būti išreikštas kaip baudžiamoji funkcija (angl. *penalty function*), kuri neleistinų reikšmių naudojimą padaro pernelyg brangų. Ribojimų naudojimas priklauso nuo sprendžiamos problemos, todėl jie nėra būtini modelyje [17];
- Apibrėžti tikslą. Tai matematinė tikslo išraiška, pavyzdžiui, sumažinti proceso vykdymo laiką. Tokiu atveju optimizacija būtų tokio įvesties parametrų reikšmių rinkinio suradimas, su kuriuo tikslo funkcija įgytų nustatytą vertę;
- Užtikrinti pakankamai didelį iteracijų skaičių siekiant, kad gauti rezultatai būtų kuo tikslesni. Kadangi tikrovė yra nedeterministinė, sunkiai nuspėjama, tai turėtų atsispindėti ir simuliaciniame modelyje. Tai reiškia, kad kelis kartus įvykdžius modelį su identiškomis įvesties parametrų reikšmėmis gausime rezultatų pasiskirtymą tam tikrame intervale. Tai turi įtakos paieškos efektyvumui ir norint daryti išvadas, paieška turi būti kartojama.

Taip pat galima atlikti DVP simuliaciją kaip įvestį naudojant iš anksto apibrėžtų diskrečių reikšmių rinkinį. Jeigu šios reikšmės yra parinktos tolygiai ir priklauso fiksuotam intervalui, galime nubrėžti tam tikrą paviršių. Taikant statistinius metodus šis paviršius gali būti aproksimuotas ir tokiu būdu gautas jo išreikštinis pavidalas, iš kuriuo galime bandyti įžvelgti optimalias vertes ar dydžių tarpusavio priklausomybes.

Toliau aprašysime paieškos algoritmus, kurie gali būti naudojami simuliacinio modelio įvesties keitimui.

2.2.1. Simplekso algoritmas

Simplekso (arba Nelder-Mead) algoritmas – tai yra euristinės paieškos metodas, naudojamas gauti tiriamos funkcijos minimalią ar maksimalią vertę. Simplekso algoritmas dažnai taikomas spręsti uždaviniams, kuriuose nėra žinomas išreikštinis optimizuojamos funkcijos ir jos išvestinių pavidalas (skaitmeninis sprendimas), tačiau funkcija privalo būti tolydi ir gali turėti daugiau nei vieną įvesties parametrą. Tokio tipo funkcija galima aproksimuoti simuliacinį modelį. Pagrindinis šio algoritmo konceptas yra simpleksas (angl. *simplex*), kuris apibrėžiamas kaip n-matės erdvės objektas, turintis plokščias sienas. N-matėje erdvėje simpleksas turi $n+1$ skaičių viršūnių. Pavyzdžiui, vienmatėje erdvėje simpleksas yra linija, dvimatėje – trikampis, trimatėje – tetraedras ir t. t. (7 pav.). Patį erdvės dimensijų skaičių nusako optimizuojamos funkcijos įvesties parametrų kiekis. Simplekso viršūnės – tai optimi-



7 pav. Simplekso pavyzdžiai vienmatėje (a), dvimatėje (b), trimatėje (c) erdvėse

zuojamos funkcijos reikšmės esant skirtingiems įvesties parametrų reikšmių rinkiniams.

Simplekso algoritme atliekamų veiksmų aprašymas dvimatėje erdvėje [30, 31]:

- Pradinio simplekso generavimas. Pradinio simplekso dydis priklauso nuo sprendžiamos problemos prigimties ir yra individualus, todėl nėra reikalavimo, kaip turėtų būti gautos jo viršūnės. Paprastai tai atliekama turint nuspėjamą pradinį funkcijos parametrų reikšmių rinkinį (viršūnę) ir kiekvienos dimensijos atžvilgiu tam tikru vienodu atstumu gaunant kitas viršūnes. Visgi per mažas pradinis simpleksas gali sukelti išankstinį paieškos konvergavimą;
- Geriausios, geros (antros pagal gerumą) ir blogiausios simplekso viršūnių identifikavimas (šias viršūnes atitinkamai žymėsime B, G ir W raidėmis). „Gerumas“ nusako artimumą siekiamam tikslui – pavyzdžiui, jeigu tikslas yra gauti kuo mažesnę tiriamos

funkcijos vertę, tai geriausia viršūnė laikoma ta, su kurios parametrų reikšmių rinkiniu gaunama mažiausia tos funkcijos vertė;

- Vidurio taško tarp geriausios ir geros viršūnių apskaičiavimas:

$$\langle \overrightarrow{x} \rangle = \frac{1}{n} \sum_{i=1}^{n} \overrightarrow{x_i} \tag{1}$$

Dvimatėje erdvėje tai būtų:

$$M = \frac{B+G}{2} = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \tag{2}$$

- Atspindys (8a pav.). Šis veiksmas atliekamas remiantis prielaida, kad optimizuojama funkcija tolygiai gerėja jai pereinant nuo blogiausios link geriausiosios viršūnės. Atspindžio viršūnė R apskaičiuojama tokiu pat atstumu nuo vidurio taško kaip ir blogiausioji viršūnė bei fiksuojama tiesėje, nubrėžtoje per blogiausią viršūnę ir vidurio tašką:

$$\overrightarrow{x_R} = \langle \overrightarrow{x} \rangle + \alpha \left(\langle \overrightarrow{x} \rangle - \overrightarrow{x_W} \right) \tag{3}$$

Čia α yra atspindžio koeficientas, kurio įprastinė vertė yra 1.

- Ekspansija (8b pav.). Šis veiksmas atliekamas įvertinus gautą atspindžio viršūnę. Jeigu optimizuojama funkcija įgavo geresnę vertę atspindžio viršūnėje nei blogiausioje viršūnėje, daroma prielaida, kad gauta atspindžio vertė gali būti „pagerinta“ papildomai padidinus jos atstumą nuo vidurio taško. Tokiu būdu gaunama ekspansijos viršūnė E:

$$\overrightarrow{x_E} = \langle \overrightarrow{x} \rangle + \beta \left(\overrightarrow{x_R} - \langle \overrightarrow{x} \rangle \right) \tag{4}$$

Čia β yra ekspansijos koeficientas, kurio įprastinė vertė yra 2.

- Sutraukimai:
  - Pirmojo tipo sutraukimas (8c pav.). Šis veiksmas atliekamas, jeigu optimizuojama funkcija neįgavo geresnės vertės atspindžio viršūnėje. Tokiu atveju atliekamas simplekso sutraukimas, kuris galimas dviem kryptimis. Dėl šios priežasties skaičiuojami du susitraukimo taškai C_1 ir C_2, bei įvertinama optimizuojamos funkcijos reikšmė kiekvienam iš taškų. Tada geresnės C vertės link atliekamas simplekso sutraukimas:

$$\overrightarrow{x_{C1}} = \langle \overrightarrow{x} \rangle + \gamma \left(\overrightarrow{x_R} - \langle \overrightarrow{x} \rangle \right) \tag{5}$$

$$\overrightarrow{x_{C2}} = \langle \overrightarrow{x} \rangle - \gamma \left(\overrightarrow{x_R} - \langle \overrightarrow{x} \rangle \right) \tag{6}$$

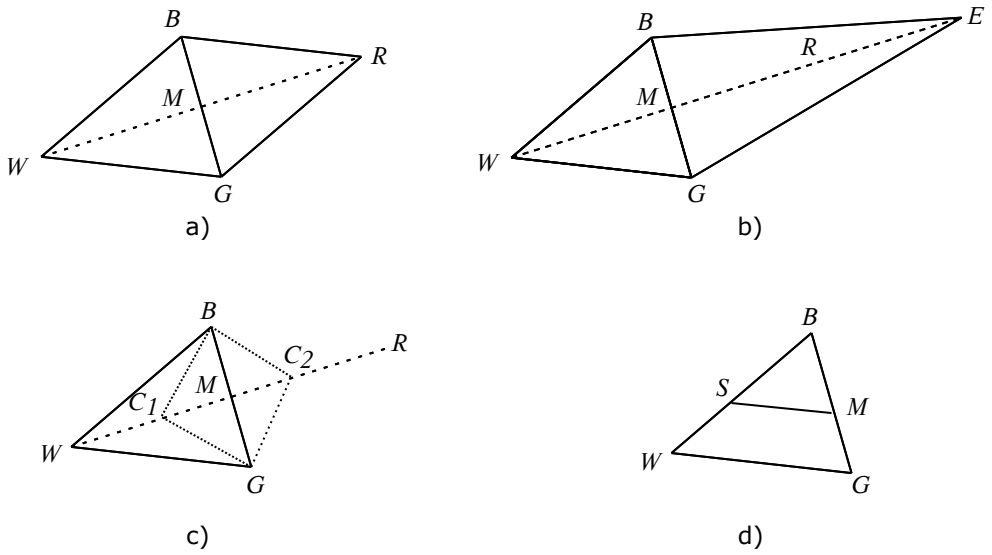
Čia γ yra pirmojo tipo sutraukimo koeficientas, kurio įprastinė vertė yra $\frac{1}{2}$;

  - Antrojo tipo sutraukimas (8d pav.). Šis veiksmas atliekamas tokiu atveju, jeigu optimizuojamos funkcijos reikšmės C taškuose nėra geresnės už funkcijos reikšmę blogiausioje viršūnėje. Šio tipo sutraukimas atliekamas link artimiausios tikslui viršūnės ir anksčiau buvęs vidurio taškas tampa naująja blogiausiąja viršūne. Naująją gerąja viršūne tampa vidurio taškas tarp geriausios ir senosios blogiausios

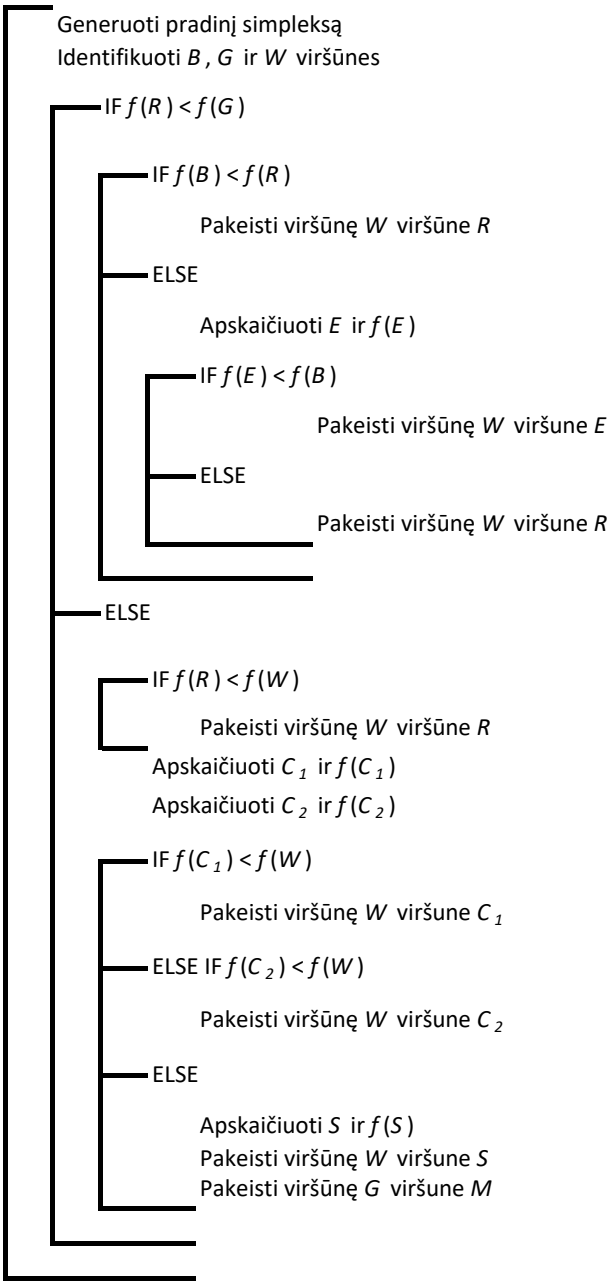
viršūnės. Bendru atveju, visi taškai, išskyros geriausiąjį, yra pakeičiami pagal šią formulę:

$$\overrightarrow{x_i} = \overrightarrow{x_B} + \delta\,(\overrightarrow{x_{0i}} - \overrightarrow{x_B}) \tag{7}$$

Čia δ yra antrojo tipo sutraukimo koeficientas, kurio įprastinė vertė yra $\frac{1}{2}$. Sutraukimo veiksmai yra brangesni lyginant su kitais veiksmais, nes jų vykdymo metu apskaičiuojama daugiau nei viena nauja viršūnė.



8 pav. Simplekso algoritme atliekamų veiksmų atvaizdavimas dvimatėje erdvėje



9 pav. Simplekso algoritmo veiksmų diagrama [30]

2.2.2. Genetinis algoritmas

Genetinis algoritmas – tai vienas iš optimizavimo metodų, kuris pagrįstas natūralios gyvybės evoliucijos ir prisitaikymo gamtoje pavyzdžiu. Pagrindinis šio algoritmo konceptas yra populiacija, kuri yra sudaryta iš tam tikro skaičiaus chromosomų (individų). Tuo metu kiekviena chromosoma yra sudaryta iš genų. Genetinis algoritmas iteratyviai gerina populiaciją. Tai pasiekiama atliekant atrankos (angl. *selection*), paveldėjimo (angl. *crossover*) ir mutacijos (angl. *mutation*) veiksmus.

Prieš pradedant spręsti problemą taikant genetinį algoritmą, reikia apibrėžti esminius komponentus [32]:

- Pradinę populiaciją (sprendinius) ir jos generavimo būdą. Populiaciją sudarančių chromosomų skaičius priklauso nuo sprendžiamos problemos, jos sudėtingumo;
- Tinkamą chromosomas sudarančių genų atvaizdavimą. Tradiciškai kaip genai yra naudojami dvejatainiai skaičiai, tačiau tai priklauso nuo sprendžiamos problemos prigimties;
- Chromosomų artimumo tikslui įvertinimo funkciją (angl. *fitness function*);
- Genetiniame algoritme taikomus veiksmus. Pavyzdžiui, nuo genų atvaizdavimo būdo gali priklausyti, kaip turėtų būti atlikti paveldėjimo ir mutacijos veiksmai;
- Kintamųjų ar koeficientų vertes kaip pavyzdžiui iteracijų skaičius, populiacijos dydis, mutacijos tikimybė ir t. t. Pavyzdžiui, esant nepakankamam pradiniam populiacijos dydžiui gali kilti pirmalaikio konvergavimo problema.

Genetiniame algoritme atliekamų veiksmų aprašymas:

- Atranka. Šio veiksmo metu remiantis įvertinimo funkcija su tam tikra tikimybe iš pradinės populiacijos atrenkamas fiksuotas skaičius tinkamiausių chromosomų. Kuo artimesnė tikslui įvertinimo funkcijos reikšmė, tuo turėtų būti didesnė tikimybė atrinkti atitinkamą chromosomą. Tikimybinis įvertinimas reikalingas siekiant išvengti ankstyvo chromosomų supanašėjimo sekančiose kartose;
- Paveldėjimas. Tai yra pagrindinė genetinio algoritmo operacija, kurios metu sukuriami naujos kartos individai naudojant anksčiau atrinktas chromosomas kaip tėvus. Paveldėjimas gali būti atliktas įvairiai, tačiau paprasčiausias būdas yra naudoti vieną ar kelis paveldėjimo taškus. Pavyzdžiui, jeigu turime šias atrinktas tėvines chromosomas:

$$(a_1, a_2, a_3, a_4, a_5, a_6) \tag{8}$$

$$(b_1, b_2, b_3, b_4, b_5, b_6) \tag{9}$$

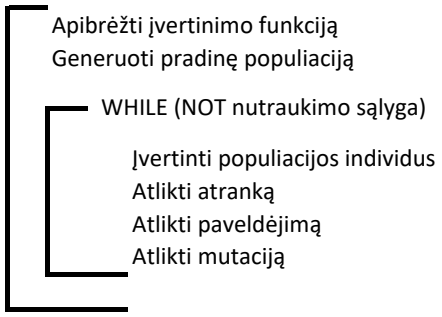
  Paveldėjimo taškas gali būti nuo 1 iki 5 (įskaitant) ir yra parenkamas atsitiktiniu būdu. Tokiu atveju, jeigu paveldėjimo taškas yra lygus 4, gaunami naujos kartos individai:

$$(b_1, b_2, b_3, b_4, a_5, a_6) \tag{10}$$

$$(a_1, a_2, a_3, a_4, b_5, b_6) \tag{11}$$

- Mutacija. Mutacijos pagalba palaikomas populiacijos įvairumas ir gaunami tokie individai, kurie galimai nepasiekiami taikant tik paveldėjimo metodą. Kiekviena chromosoma, esant tam tikrai tikimybei, gali pakeisti vieno ar kelių genų reikšmes. Mutacijos pavyzdys:

$$(a_1, a_2, a_3, a_4, a_5) \xrightarrow{Mutacija} (a_1, a_2, a_3, x, a_5) \tag{12}$$



10 pav. Genetinio algoritmo aukšto lygmens veiksmų diagrama

Genetinis algoritmas gali būti taikomas sprendžiant įvairias problemas, pavyzdžiui, optimizuoti funkcijas, kurios yra netolydžios ar nediferencijuojamos. Darbe [32] teigiama, kad algoritmas naudojamas sprendžiant keliaujančio pirklio, tvarkaraščių sūdarymo, schemų braižymo (angl. *wire-routing*) uždavinius.

2.3. Struktūrinė optimizacija

Vykdant verslo proceso modelio struktūrinę optimizaciją, ieškoma optimaliausio proceso vykdymo scenarijaus t. y. keičiama procesą sudarančių veiklų seka. Kaip minėta anksčiau, tai yra sudėtingas uždavinys.

Vienas iš bandymų išspręsti šią problemą buvo pasiūlytas [33] darbe. Šio darbo autoriai sukūrė PLM srauto įrankį (angl. *PLM flow tool*), kuris geba keisti proceso veiklų seką vykdymo metu. Minėtas įrankis pagrįstas idėja, kad VT rinkinys turi būti lengvai modifikuojamas bei atskirtas nuo verslo proceso. Tik VT rinkinio pakeitimai gali lemti skirtingą procesą sudarančių veiklą tvarką. PLM įrankis sudarytas iš dviejų komponentų:

- VT ir šablonų bazė, naudojama aprašyti įvairią verslo logiką. Čia VT išreiškiamos ECA forma;
- Srauto mašina, kuri naudojama formuojant proceso vykdymo scenarijų.

Verslo proceso modelis sudaromas atliekant šiuos žingsnius:

1. Sudaromos galimų atgalinių veiklų (angl. *backward-chain*) sekų rinkinys. Seka sudaroma analizuojant taisykles, kurios yra susietos su konkrečiomis veiklomis. Atgalinės taisyklės gali būti duomenų ribojimai – prieš vykdant veiklą turi būti pasiekiami tam tikri duomenys, arba srauto ribojimai – prieš vykdant veiklą turi būti įvykdytos tam tikros kitos veiklos. Atgalinės veiklų sekos sudarymo procesas:

1.1. Pradžioje kaip algoritmo įvestis naudojama galutinė verslo proceso veikla (angl. *target task*). Toliau algoritmas rekursyviai naudojamas kiekvienai naujai veiklai iki tol, kol atgalinės–grandinės taisyklės nesugeneruoja naujų veiklų;
1.2. Jeigu kurioje nors atgalinės–grandinės taisyklėje apibrėžtas `OR` operatorius, tokiu atveju gali būti suformuotos kelios galimos veiklų sekos;
1.3. Jeigu kurioje nors suformuotoje veiklų sekoje susidaro uždaras veiklų ciklas, tokios sekos yra atmetamos.

2. Sudaromas galimų priekinių veiklų (angl. *forward-chain*) sekų rinkinys. Proceso scenarijus, suformuotas taikant atgalinės–grandinės taisykles, gali būti papildytas naujomis veiklomis taikant priekinės–grandinės taisykles. Šiuo atveju analizuojamos taisyklės, kurios apibrėžia veiklų tarpusavio priklausomybes ir reikalauja, kad, jeigu bus vykdoma veikla A, tai vėliau turės būti įvykdyta ir veikla B. Čia taisyklių analizė yra sudėtingesnė, kiekviena taisyklės sąlyga išskaidoma į sprendimų medį:
   - Jeigu taisyklės sąlyga yra tenkinama, proceso scenarijus papildomas su analizuota taisykle susieta veikla;
   - Jeigu taisyklės sąlyga nėra tenkinama, tikrinamas taisyklių tarpusavio prieštaravimas. Jeigu proceso veiklų sekoje aptinkami prieštaravimai tarp tarp priekinės–grandinės ir galinės–grandinės ar dviejų priekinės–grandinės taisyklių, daroma išvada, kad nėra galimybių pasiekti galutinės veiklos ir toks proceso scenarijus atmetamas.
3. Pasirenkamas geriausias proceso scenarijus;
4. Pasirinktas proceso scenarijus yra vykdomas.

Kitas verslo procesų struktūrinės optimizacijos metodas pateiktas [34] darbe. Šis sprendimas pagrįstas teiginiu, kad kiekviena verslo proceso veikla turi įvestį ir išvestį bei priskirtus atributus. Minėto darbo kontekste įvestis ir išvestis yra suprantama kaip išteklius, kurio prigimtis nėra svarbi. Tokiu būdu verslo proceso veiklos tarpusavyje yra susietos per išteklius. Tuo metu atributai charakterizuoja pačią veiklą, pavyzdžiai, gali nusakyti veiklos kainą ar trukmę.

Laikantis šių prielaidų, [34] darbe siūloma apibrėžti dvi matricas:

- Veiklų-išteklių matrica (1 lentelė). Kiekvienas matricos langelis nusako konkrečios veiklos ir išteklio sąryšį. Jeigu išteklius yra veiklos įvestis, tada langelio reikšmė lygi 1; jei veikla generuoja išteklių kaip išvestį – 2; jei veikla išteklio nenaudoja – 0. Ši matrica naudojama įvertinti, ar suformuotas proceso scenarijus yra galimas;
- Veiklų-atributų matrica. Šioje matricoje pateiktos kiekvienos veiklos atributų reikšmės. Reikšmės gali būti agreguojamos skaičiuojant galutines proceso scenarijaus charakteristikas.

Minėto darbo autoriai teigia, kad naudojant veiklų-išteklių matricą ir jų sukurtą PCA

1 lentelė. Veiklų-išteklių matrica [34]

	Išteklius 1	**Išteklius 2**	**Išteklius 3**	**Išteklius 4**	**Išteklius 5**
Veikla 1	1	2	0	0	0
Veikla 2	0	0	0	1	0
Veikla 3	0	0	2	0	0
Veikla 4	0	1	0	0	0

(angl. *Process Composition Algorithm*) algoritmą, galima suformuoti verslo proceso vykdymo scenarijus. Sudaryti proceso scenarijai yra pateikiami grafiškai bei įvertinamas jų galimumo laipsnis (angl. *Degree of Infeasibility*). Scenarijus laikomas galimu, jeigu yra tenkinamos šios sąlygos:

1. Visi ištekliai, kuriuos reikalaujama naudoti proceso scenarijuje, yra naudojami vienos ar kelių suformuoto scenarijaus veiklų;
2. Visi ištekliai, kuriuos reikalaujama generuoti proceso scenarijuje, yra generuojami vienos ar kelių suformuoto scenarijaus veiklų;
3. Suformuoto scenarijaus veiklos tarpusavyje turi būti susietos per įvesties ar išvesties išteklius, arba per kitas veiklas. Proceso scenarijaus pirmosios veiklos įvestis ir paskutinės veiklos išvestis yra griežtai apibrėžta.

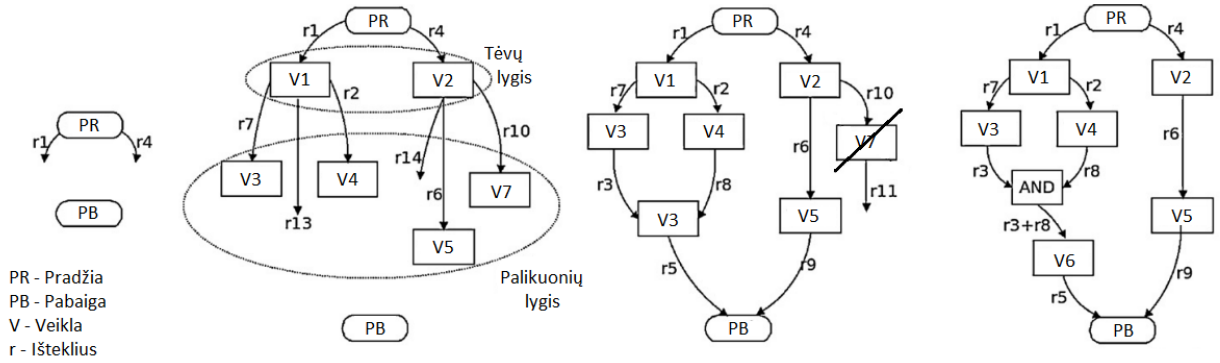
Pats PCA algoritmas generuoja šias išvestis:

- Suformuotas proceso scenarijus, pateiktas grafiniu pavidalu;
- Atnaujintų scenarijaus veiklų rinkinys. Algoritmas geba atnaujinti veiklas, remdamasis taisyklėmis:
  - Jeigu suformuotame proceso scenarijuje yra nepanaudotų veiklų, jos taip pat turi būti panaikintos iš veiklų-išteklių matricos;
  - Formuojant proceso scenarijų pirmines veiklas gali pakeisti naujai pridėtos veiklos, kurios nėra apibrėžtos veiklų-išteklių matricoje. Šios veiklos saugomos bazėje ir naudojamos, jeigu reikia pagerinti proceso galimumo laipsnį. Tokiu atveju taip pat turi būti atnaujinta veiklų-išteklių matrica.
- Suformuoto proceso scenarijaus galimumo laipsnio reikšmė.

Galimumo laipsnis apskaičiuojamas pagal formulę:

$$D = 1 \cdot n_{in} + 5 \cdot r_{out} + 3 \cdot r_{in} \tag{13}$$

čia n_{in} yra proceso scenarijuje panaudotų naujų veiklų skaičius, r_{out} – nesugeneruotų išteklių skaičius, r_{in} – nepanaudotų išteklių skaičius. Geriausios proceso scenarijus yra tas, kuri D vertė lygi 0, tačiau darbo [34] autoriai teigia, kad suformuoti tokį proceso scenarijų yra labai sudėtinga.



11 pav. Proceso scenarijaus formavimas taikant PCA algoritmą [34]

2.4. Apibendrinimas

Efektyvaus verslo procesų optimizacijos metodo sukūrimas šiuo metu yra siekiamybė, kuri leistų verslo organizacijoms ženkliai taupyti savo išteklius. Priklausomai nuo to, kaip aprašytas verslo proceso modelis, optimizacija gali būti atlikta įvairiai. Jeigu verslo procesas apibrėžtas naudojant griežtą matematinį modelį, tada optimizacija įgyvendinama ieškant ekstremumo taško tam tikroje tiriamoje funkcijoje. Tačiau tokių matematinių verslo procesų modelių sukūrimas yra brangus ir ne visada galimas. Praktiškesnis būdas – naudoti kompiuterinę procesų simuliaciją. Tam reikia išreikštinai apibrėžti proceso įvestį ir ją keisti atsižvelgiant į siekiamą tikslą ir gaunamą rezultatą. Jeigu rezultatas patenka į nustatytas tenkinamas ribas, teigiama, kad optimizacija įgyvendinta. Tokiu atveju optimizacija gali apimti proceso vykdymo scenarijaus pertvarkymą (struktūrinė) arba geriausios konfigūracijos paiešką (operacinė).

3. DINAMINIUOSE VERSLO PROCESUOSE NAUDOJAMŲ IŠTEKLIŲ OPTIMIZAVIMO METODAS

3.1. Metodo bendra apžvalga

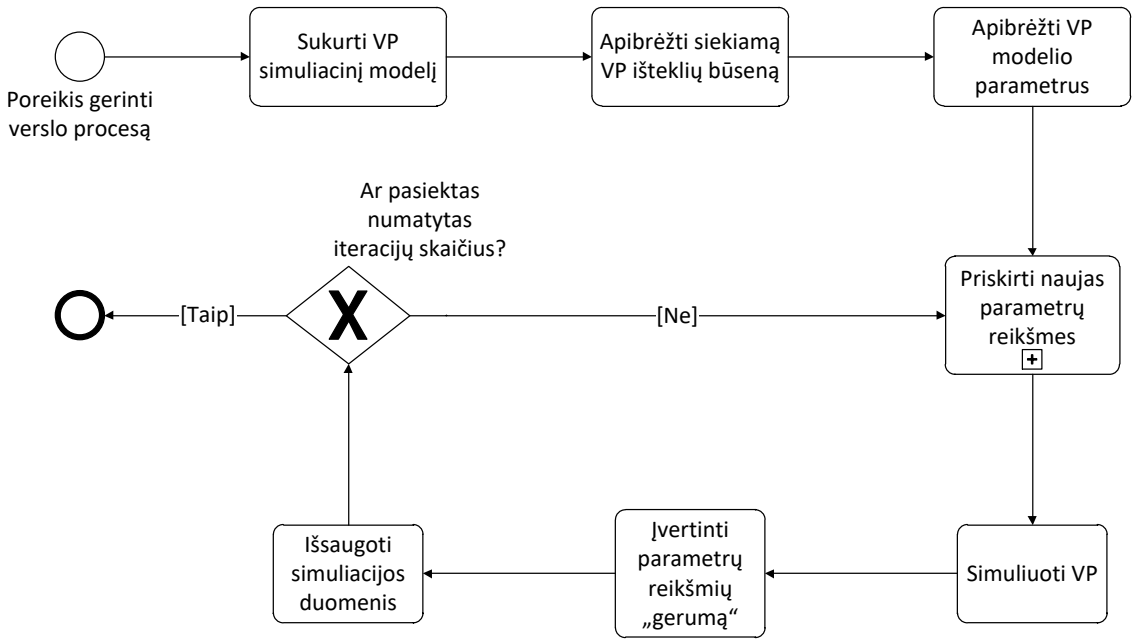
Šio darbo kontekste verslo proceso tikslu yra laikoma tam tikra išteklių būsena – DVP reikia pertvarkyti tokiu būdu, kad jį įvykdžius pasiektume šią iš anksto apibrėžtą išteklių rinkinio būseną. Kaip minėta anksčiau, gali būti taikomi šie optimizacijos metodai:

- Daugiakriterinė optimizacija (angl. *Multi-objective optimization*). Čia būtina sudaryti formalų–matematinį verslo proceso simuliacinį modelį: ištekliai, jų pokyčiai, tarpusavio sąryšiai turi būti išreikšti matematinėmis lygtimis. Taikant šį metodą galima gauti matematine prasme optimalų sprendimą;
- Tikslo siekiančių DVP simuliacija. Taikant šį metodą reikia stebėti, kaip kinta išteklių būsena priklausomai nuo proceso įvesties pokyčio. Kiekviename sekančiame žingsnyje įvestis turėtų būti keičiama tokia kryptimi, kuri leistų priartėti prie siekiamo tikslo būsenos. Čia bandoma gauti ne visiškai optimalų, o pakankamai gerą, mus tenkinantį sprendimą, kuris patenka į siekiamą iš anksto apibrėžtą intervalą. Toliau detaliau nagrinėsime šį būdą ir pateiksime pirminį principą, kaip turėtų būti įgyvendinta tokio tipo optimizacija.

Naudojant verslo procesų simuliaciją minėtai išteklių optimizacijai pasiekti, svarbiausias ir kartu sudėtingiausias veiksmas yra simuliacinio modelio keitimas. Derėtų taip keisti proceso modelį, kad jį įvykdžius išteklių būsena priartėtų prie siekiamos. Pavyzdžiui, darbe [23] teigiama, kad atliekant tikslo siekiančių DVP simuliaciją, gali būti keičiama veiklų seka atsižvelgiant į konkrečių veiklų naudą (tai galima įvertinti, pavyzdžiui, iš istorinių duomenų) ir aprašytas VT (ribojimus). Šiame darbe laikysime, kad modelio keitimas susietas su naujų parametrų reikšmių priskyrimu t. y. simuliacinis modelis yra parametrizuojamas:

- Verslo proceso simuliacinis modelis gali turėti tam tikrus įvesties parametrus, nuo kurių priklauso simuliacijos rezultatas. Pavyzdžiui, gali būti naudojamas įvesties parametras „Užsakymo kiekis“. Priklausomai nuo to, kokia parametro reikšmė yra naudojama, tai gali turėti skirtingą įtaką galutinei išteklių būsenai;
- Jeigu procesas turi skirtingas galimas veiklų atlikimo tvarkas, kurios gali lemti skirtingas išteklių būsenas, tada simuliuoti visus galimus scenarijus. Tai galima įgyvendinti taip aprašant simuliacinį modelį, kad skirtingi scenarijai būtų vykdomi priklausomai nuo to, kokios yra įvesties parametrų reikšmės.

12 paveiksle aprašytas siūlomas optimizacijos procesas:



12 pav. Verslo proceso optimizacija naudojant simuliaciją

.

1. Sukurti verslo proceso simuliacinį modelį: apibrėžti įvykius, veiklas (užduotis) ir jų vykdymo sąlygas, išteklius ir kitus simuliacijai reikalingus elementus;
2. Aprašyti siekiamą verslo proceso išteklių būseną: išteklių bazėje (kuri gali būti realizuota įvairiai) apibrėžti siekiamas išteklių charakteristikas, pavyzdžiui, minimalias, maksimalias vertes. Šias charakteristikas turėtų naudoti tikslo vertinimo funkcija lyginant galinę išteklių būseną (kuri gauta vykdant simuliacija su konkrečiu parametrų reikšmių rinkiniu) su šia aprašyta siekiama būsena;
3. Aprašyti simuliacinio modelio įvesties parametrus: priskirti parametrų tipus, pavadinimus, pradines reikšmes. Pradinės reikšmės naudojamos kaip paieškos atskaitos taškas, pavyzdžiui, generuojant pradinę populiaciją;
4. Priskirti naujas parametrų reikšmes. Jeigu simuliacija vykdoma pirmą kartą, tai būtų pradinių reikšmių naudojimas, priešingu atveju – taikant tam tikrą algoritmą priskiriamos naujos reikšmės. Nuo šio žingsnio pradedamas parametrų reikšmių keitimo-paieškos ciklas, kurio metu ieškoma tokio reikšmių rinkinio, kuris leistų pasiekti apibrėžtą išteklių būseną;
5. Vykdyti verslo proceso simuliaciją. Kiekvienos simuliacijos metu priskiriamos vis naujos simuliacinio modelio parametrų reikšmės. Kaip minėta anksčiau, skirtingų parametrų reikšmių rinkiniai turėtų lemti skirtingas gaunamas galines išteklių būsenas;
6. Įvertinti parametrų reikšmių „gerumą“. Šis dydis gaunamas naudojant tikslo įvertinimo funkciją, kurios argumentas yra išteklių būsena. Tuo metu pati išteklių būsena gaunama atlikus simuliaciją su tiriamu parametrų reikšmių rinkiniu;
7. Išsaugoti simuliacijos ir optimizacijos duomenis, pavyzdžiui, naudotą parametrų reikšmių rinkinį, gautą galinę išteklių būseną, tikslo įvertinimo funkcijos reikšmę. Išsaugoti

duomenys vėliau gali būti naudojami analizei, grafikų braižymui;

8. Priklausomai nuo to, ar pasiekta minėto ciklo sustabdymo sąlyga, pavyzdžiui, iteracijų skaičius ar paieškos laikas, toliau keisti parametrų reikšmes arba baigti procesą.

Vykdant tokią optimizaciją, jos efektyvumas žymiai priklauso nuo naudojamų parametrų reikšmių keitimo algoritmo ir tikslo įvertinimo funkcijos. Šiuos du elementus aptarsime sekančiuose skyriuose.

3.2. Įvesties parametrų reikšmių keitimas

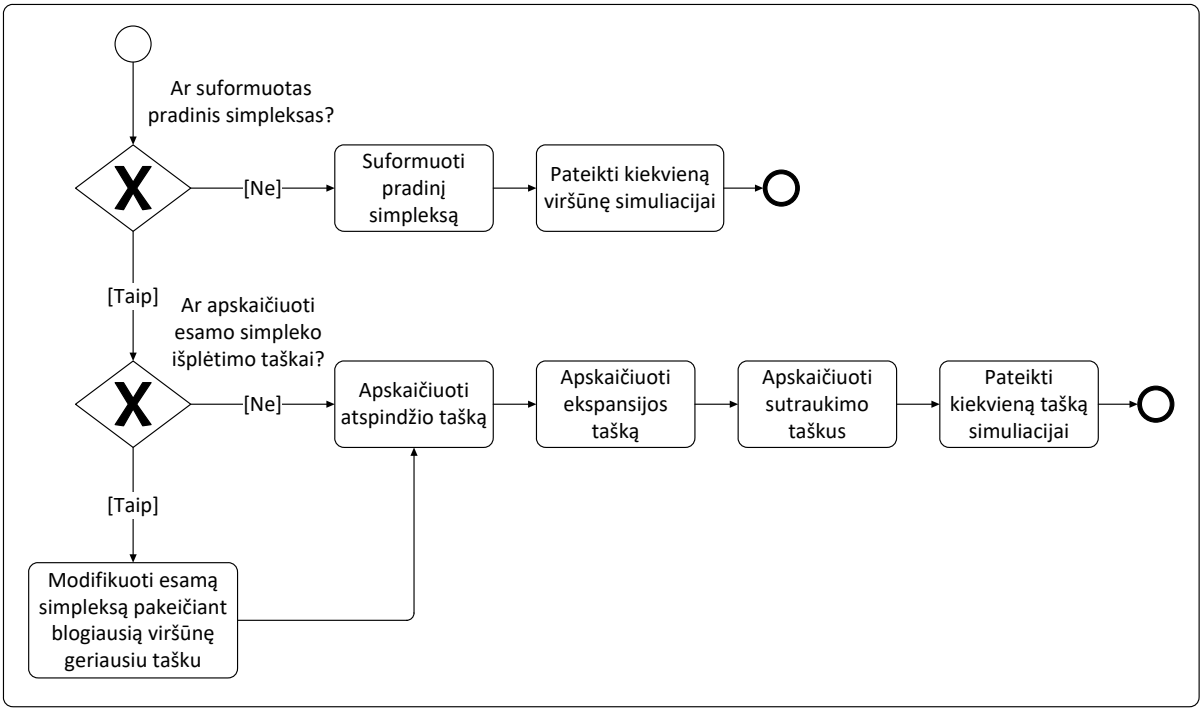
Simuliacinio modelio įvesties parametrų reikšmės gali būti keičiamos naudojant:

- „Akląją“ paiešką. Iš anksto apibrėžiamos galimų įvesties parametrų reikšmių aibės ir tokiu būdu išbandomas visos galimos reikšmių kombinacijos. Formaliai toks simuliacinio modelio keitimas negalėtų būti vadinamas optimizacija, tačiau tai leidžia sąlyginai greitai ištestuoti simuliacinio modelio elgseną tam tikrame parametrų reikšmių intervale;
- Euristinius paieškos algoritmus. Ši paieška dar vadinama informuotąja, nes, kaip minėta anksčiau, čia naudojama pagalbinė tikslo įvertinimo funkcija, nusakanti, kaip arti siekiamos išteklių būsenos yra konkrečiu simuliacijos vykdymo atveju gauta galinė išteklių būsena. Tokiu būdu suteikiama informacija apie sekantį paieškos žingsnį, o leidžia keisti parametrų reikšmes atsižvelgiant į siekiamą tikslą.

Šiame darbe verslo proceso išteklių optimizacijai naudosime simplekso algoritmą bei genetinę euristines paieškas.

Simplekso algoritmo pasirinkimo priežastis – algoritmo gebėjimas optimizuoti funkcijas, kurios turi daugiau nei vieną įvesties parametrą. Simuliacinį modelį galima aproksimuoti tokio tipo funkcija. Tačiau šio algoritmo trūkumas yra tas, kad optimizuojama funkcija turi būti tolydi, todėl jį sudėtinga taikyti tokių uždavinių sprendimui, kurių įvesties ar rezultatų aibė yra ribota (angl. *constrained optimization*). Toliau pateiksime įvesties parametrų reikšmių keitimo proceso aprašymą, kuris atliekamas naudojant simplekso algoritmą (13 pav.):

- Taikant šį metodą laikoma, kad simplekso viršūnės ir išplėtimo taškai atitinka konkrečius įvesties parametrų reikšmių rinkinius (simpleksas apibrėžtas šių reikšmių erdvėje);
- Pirmieji simuliacijai pateikiami reikšmių rinkiniai yra pradinio simplekso viršūnės. Su kiekviena simplekso viršūne (reikšmių rinkiniu) vykdoma simuliacija, kad jiems būtų priskirta galinė išteklių būsena ir įvertintas viršūnių „gerumas“;
- Atlikus simuliaciją su visomis pradinio simplekso viršūnėmis, generuojami simplekso praplėtimo ir sutraukimo taškai. Apskaičiavus šiuos taškus, su jais taip pat vykdoma simuliacija;



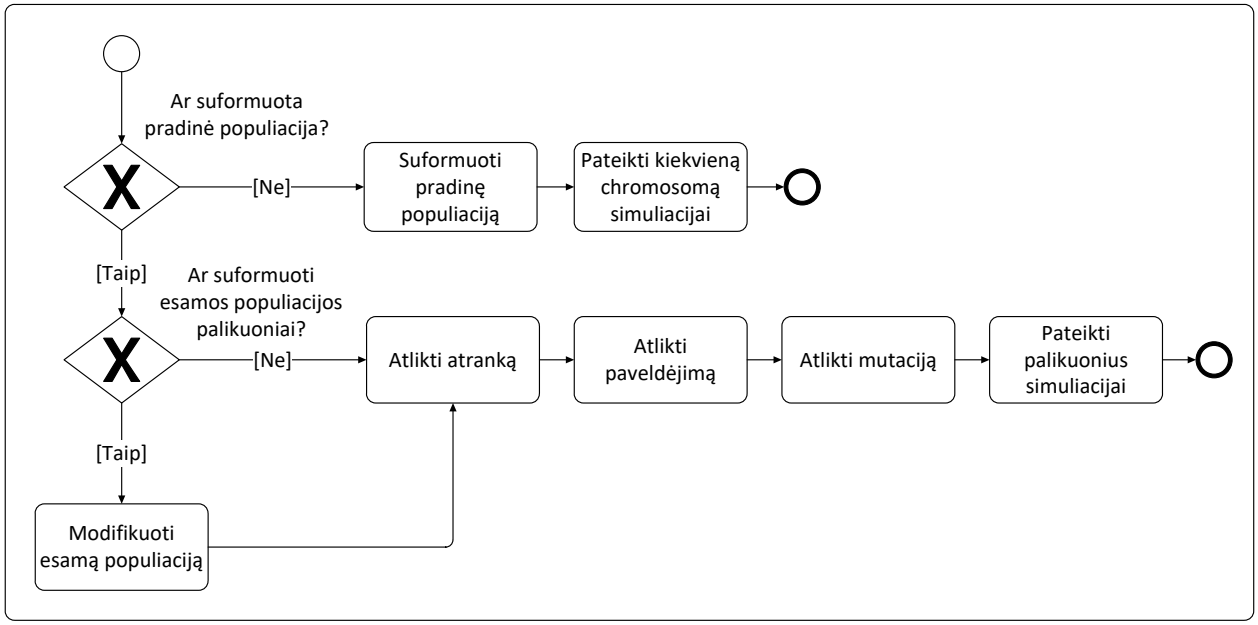
13 pav. Įvesties parametrų reikšmių keitimas naudojant simplekso algoritmą

.

- Įvertinus išplėtimo taškų „gerumą“, esamas simpleksas modifikuojamas atliekant atspindį, ekspansiją ar sutraukimą. Bet kuriuo veiksmo atveju blogiausią simplekso viršūnę pakeičia geriausias išplėtimo taškas.

Genetinio paieškos algoritmo taikymas verslo proceso optimizacijai minimas įvairiuose darbuose, pavyzdžiui, [17, 34]. Genetinį algoritmą įprastai nesunku įgyvendinti, paieška dažnai nesustoja ties lokalių sprendimų radimu, todėl jis taikomas skirtingose srityse, pavyzdžiui, ekonomikoje, mašininiame mokyme. Trūkumai – labai svarbu tinkamai apibrėžti pradinę populiaciją ir atrankos, paveldėjimo, mutacijos veiksmus, be to, gali prireikti daug iteracijų. Įvesties parametrų reikšmių keitimo procesas taikant genetinę paiešką:

- Taikant šį metodą laikoma, kad chromosoma yra konkretus simuliacijoje naudojamas įvesties parametrų reikšmių rinkinys (atitinkamai genas yra įvesties parametro reikšmė);
- Pirmieji simuliacijai pateikiami reikšmių rinkiniai atitinka pradinės populiacijos chromosomas. Ši pradinė populiacija generuojama naudojant etaloninę chromosomą, kiekvieną jos geną a_i dauginant iš atsitiktinai sugeneruotos vertės r_i ir mastelio keitimo keoficiento α_i. Taigi populiacijos naujos chromosomas genas $b_i = a_i \cdot r_i \cdot \alpha_i$;
- Atlikus simuliaciją su visomis pradinės populiacijos chromosomomis, paeiliui vykdomos atrankos, paveldėjimo ir mutacijos veiksmai:
  - Atrankos metu kiekvienai chromosomai priskiriama tikimybė būti pasirinktai, kuri proporcinga chromosomos „gerumui“. Atranka gali būti įgyvendinta taikant ruletės parinkimo metodą;



14 pav. Įvesties parametrų reikšmių keitimas naudojant genetinę paiešką

.

  - Paveldėjimas įgyvendintas tarpusavyje sukeičiant anksčiau parinktų chromosomų genus. Keitimo taškas parenkamas atsitiktinai;
  - Mutacija atliekama atrinktoms chromosomom po paveldėjimo ir su tam tikra tikimybe. Mutacijos genas parenkamas atsitiktinai. Genas dauginamas iš atsitiktinės vertės ir mastelio keitimo koeficiento $a_i^{'} = a_i \cdot r_i \cdot \alpha_i$.

- Atlikus minėtus veiksmus, geresnioji iš naujai gautų chromosomų pridedama į populiaciją pakeisdama prasčiausią populiacijos narį.

3.3. Tikslo įvertinimo funkcija

Kiekvieną kartą pakeitus įvesties parametrų reikšmių rinkinį, įvykdoma verslo proceso simuliacija ir gaunama galinė išteklių būsena. Ši galinė išteklių būsena yra lyginama su siekiama išteklių būsena. Apskritai kalbant, lyginimo logika gali būti įvairi ir priklauso nuo sprendžiamos problemos, tačiau bet kuriuo atveju ji turi nusakyti, kaip arti gauta išteklių būsena yra siekiamos būsenos atžvilgiu.

Norint įvertinti, ar pasiekta siekiama išteklių būsena, reikia apibrėžti būsenos apibrėžime naudojamas ištekliaus charakteristikas. Šiame metode siūloma naudoti tokį charakteristikų rinkinį:

- Unikalus naudojimo identifikatorius;
- Įvesties parametrų reikšmių rinkinio identifikatorius. Apibrėžia, kokiam reikšmių rinkiniui esant gauta esama ištekliaus būsena;
- Ištekliaus identifikatorius ar pavadinimas;
- Atributo identifikatorius ar pavadinimas, pavyzdžiui, ištekliaus kiekis;

- Ištekliaus vertė (toliau žymėsime r_i). Pagrindinis dydis, naudojamas apibrėžiant rėžius, kuriuos turi tenkinti pakankamai geras sprendimas;
- Ištekliaus vertės tipas, kuris gali būti:
  - Pradinė (r_{0_i});
  - Galinė (r_i);
  - Siekiama-Minimali (r_{min_i});
  - Siekiama-Maksimali (r_{max_i});
  - Siekiama-Ideali (r_{f_i}).

Papraščiausiu atveju išteklių būsenos tikslo vertinimas gali būti atliekamas pagal išteklių skaičių, kuris patenka į nustatytas ribas. Kiekvienam ištekliui r_i, priklausiančiam verslo proceso išteklių aibei $r_i \subset R$, priskiriama minimali r_{min_i} ir maksimali r_{max_i} išteklio siekiamos būsenos toleravimo ribos. Tokiu atveju apibrėžiame funkciją, kuri vertina ištekliaus patekimą į šias ribas:

$$f_1(r_i) = \begin{cases} 0, & r_{min_i} \leq r_i \leq r_{max_i} \\ 1, & r_i < r_{min_i} \text{ arba } r_i > r_{max_i} \end{cases} \tag{14}$$

Jeigu išteklius patenka į nustatytas ribas, grąžinamos funkcijos reikšmė lygi 0, priešingu atveju lygi 1 (atstumo prasmė). Bendras rezultatas gaunamas atlikus sumavimą visų išteklių atžvilgiu:

$$S_1 = \sum_i^n \alpha_i \cdot f_1(r_i) \tag{15}$$

Čia papildomai gali būti naudojamas išteklio svarbumo koeficientas α. Tačiau toks galinės ir siekiamos išteklių būsenų lyginimo būdas yra labai grubus – nors patenkančių į nustatytas ribas išteklių skaičius gali ir sutapti, tačiau mažai tikėtina, kad šios išteklių būsenos yra identiškos, viena iš jų tikriausiai yra artimesnė tikslui už kitą. Esant tokiam vertinimo būdui, algoritmo paieškos kelias gali būti labai neefektyvus.

Išteklių būsenas galima lyginti pagal jų santykinį nuotolį nuo fiksuoto taško (idealios vertės), esančio siekiamos išteklių būsenos ribose. Tokiu atveju reikia apibrėžti fiksuotą tašką r_{f_i}, kuriam taip pat galioja sąlyga $r_{min_i} \leq r_{f_i} \leq r_{max_i}$. Tada santykinio nuotolio vertinimas:

$$f_2(r_i) = \frac{|r_i - r_{f_i}|}{r_{max_i} - r_{min_i}} \tag{16}$$

Šiuo atveju bendras rezultatas gali būti atlikus sumavimą visų išteklių atžvilgiu:

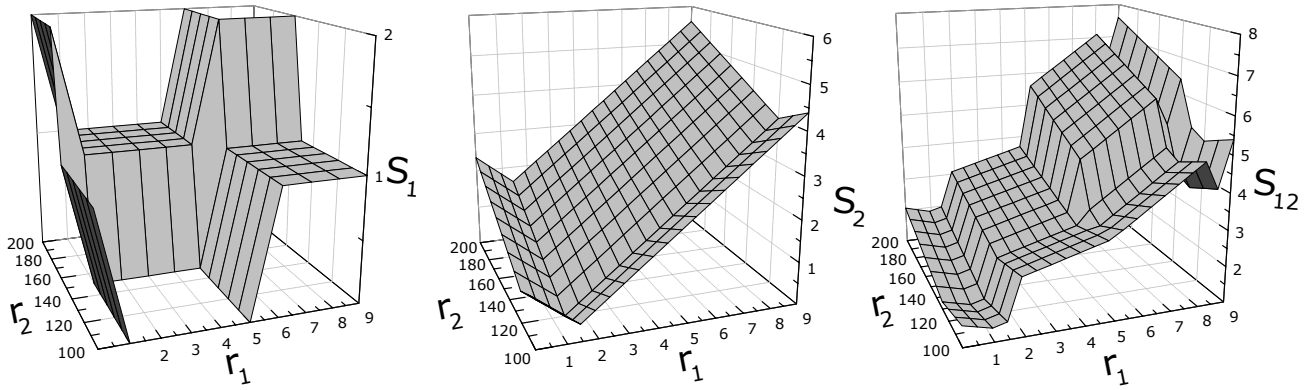
$$S_2 = \sum_i^n \alpha_i \cdot f_2(r_i) \tag{17}$$

Taikant 17 formulę, nepriklausomai nuo to, ar vienodas išteklių kiekis patenka į siekiamas ribas, jų būsenos vis vien gali būti lyginamos tarpusavyje. Taip pat galima bandyti taikyti įvairius kombinuotus metodus, pavyzdžiui:

$$S_{12} = S_1 + \frac{1}{n} \cdot S_2 \tag{18}$$

Antrasis 18 išraiškos narys apibrėžia santykinių išteklių nuotolių nuo siekiamos būsenos vidurkį.

Toliau 15 paveiksle pateiktas pavyzdys, kuriame nagrinėjami išteklių r_1 ir r_2 būsenų erdvė taikant skirtingus vertinimo metodus. Apibrėžti šie siekiamos išteklių būsenos parametrai: $\alpha_1 = \alpha_2 = 1$, $r_{min_1} = 2$, $r_{max_1} = 6$, $r_{f_1} = 4$; $r_{min_2} = 100$, $r_{max_2} = 150$, $r_{f_2} = 120$. Paieškos algoritmas šiose erdvėse ieško minimumo taško, o mūsų apibrėžtas vertinimo metodas turėtų užtikrinti, kad šis taškas atitiktų siekiamą išteklių būseną.



15 pav. Išteklių būsenų erdvė taikant skirtingus būsenų vertinimo metodus

3.4. Apibendrinimas

Šio darbo tikslas yra optimizuoti DVP naudojamus išteklius atsižvelgiant į siekiamą jų būseną. Norint tai įgyvendinti nuspręsta parametrizuoti verslo proceso simuliacinį modelį. Skirtingos parametrų reikšmės lemia skirtingas išteklių būsenas įvykdžius procesą. Parametrų reikšmių rinkiniai, kurie leidžia gauti siekiamą išteklių būseną, turi būti gauti taikant euristinę paiešką. Šiame darbe siūloma naudoti simplekso ir genetinę paieškas. Taip pat būtina įvesti tikslo vertinimo funkcija, kurios pagalba nusakoma, kaip arti gauta galinė išteklių būseną, naudojant konkretų parametrų reikšmių rinkinį, yra siekiamos būsenos atžvilgiu.

4. IŠTEKLIŲ OPTIMIZAVIMO PROTOTIPO REALIZACIJOS APRAŠYMAS

Prototipas realizuotas praplečiant Vilniaus Gedimino technikos universiteto sukūrtą „DBP-Sim“ simuliacijos įrankį. Pats įrankis sukurtas naudojant C# programavimo kalbą ir .NET 4.6.1 karkasą, todėl šios technologijos buvo naudojamas ir toliau jį vystant. Kaip minėta anksčiau, šis įrankis naudojamas dėl to, jog yra nemokamas, atviro kodo, gali vykdyti dinaminių verslo procesų simuliaciją. Įrankyje šiuo metu nėra įgyvendinta jokio verslo procesų optimizacijos galimybė. Daugiau informacijos apie patį įrankį ir jo architektūrą pateikta darbe [35].

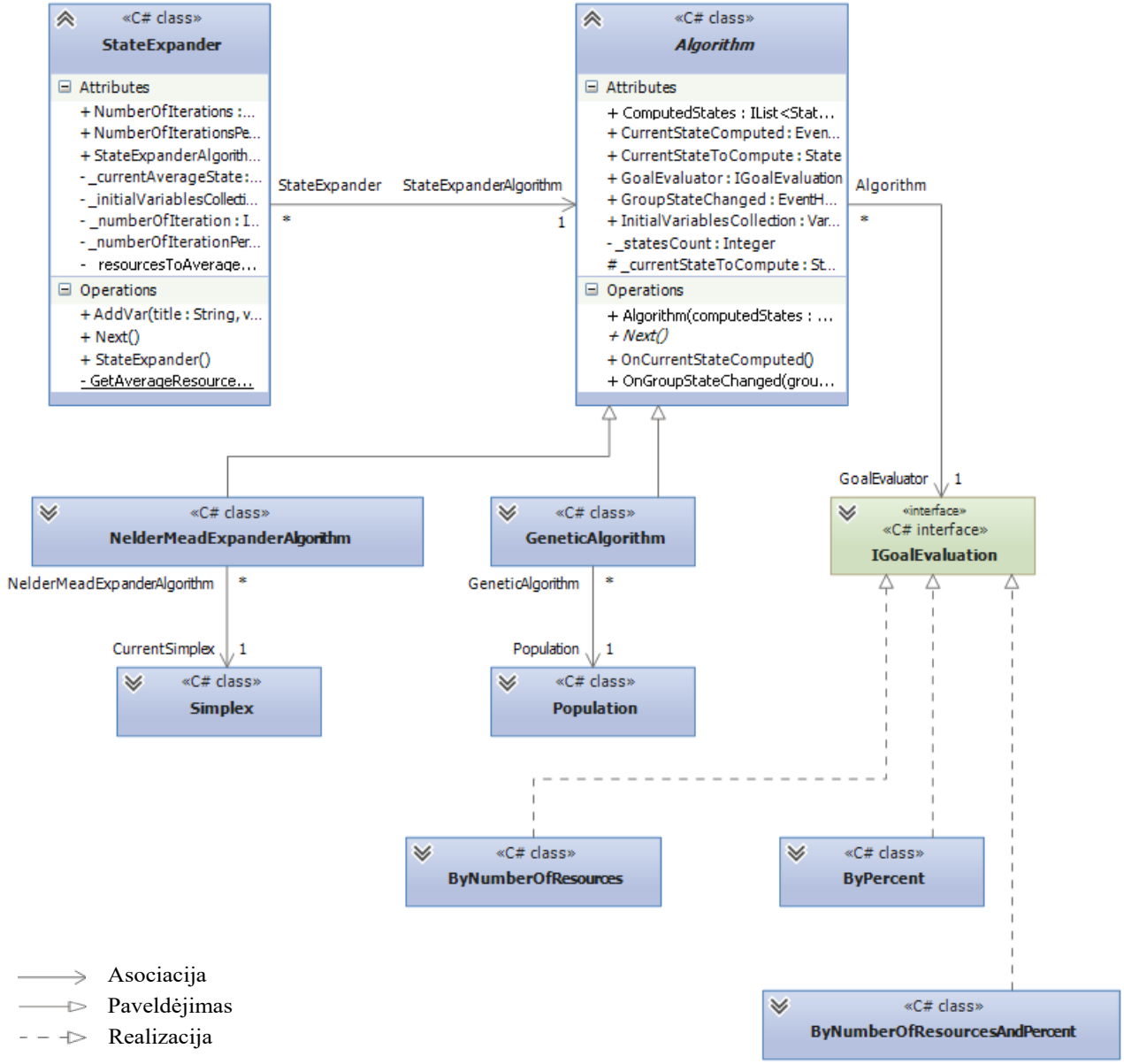
4.1. Išplėstiniai komponentai

Sekančiame paveiksle pavaizduota UML klasių diagrama, kurioje pateikti svarbiausi išplėstiniai komponentai:

- Būsenų keitiklio klasė (diagramoje *StateExpander*). Ši klasė atsakinga už paieškos valdymą: paieškos nustatymų kaip algoritmas, iteracijų skaičius saugojimas, įvesties parametrų reikšmių priskyrimas, paieškos vykdymas pažingsniui;
- Algoritmo abstrakčioji klasė (*Algorithm*). Ši klasė naudojama kaip tėvinė klasė konkretiems paieškos algoritmams. Paieškos klasės, kurios paveldi šią klasę, privalo apibrėžti sekančio paieškos žingsnio operaciją. Taip pat šioje klasėje apibrėžti elementai, kuriuose saugomi paieškos rezultatai: įvesties parametrų reikšmių rinkiniai kartu su jiems gautomis galinėmis išteklių būsenomis, sekantis skaičiavimui pateikas parametrų reikšmių rinkinys;
  - Simplekso algoritmo klasė (*NelderMeadExpanderAlgorithm*). Šioje klasė aprašyta logika, kuri reikalinga atlikti paiešką pagal anksčiau aprašytą simplekso paieškos algoritmą. Kartu papildomai įvedama simplekso (*Simplex*) klasė, kurioje apibrėžtos simplekso transformavimui reikalingos operacijos;
  - Genetinio algoritmo klasė (*GeneticAlgorithm*). Šioje klasėje aprašyti genetiniui algoritmui būdingi veiksmai kaip atranka, paveldėjimas ir mutacija. Taip pat apibrėžta populiacijos (*Population*) klasė.
- Tikslo vertinimo sąsaja (*IGoalEvaluation*). Ši sąsaja naudojama siekiant įvertinti, kaip arti gauta galinė išteklių būsena yra siekiamos būsenos atžvilgiu. Šiuo metu paieškos logika aprašyta tokiu būdu, kad sąsajos vertinimo metodas *EvaluateGoalValue* turi grąžinti realų skaičių aibėje $\in [0; +\infty)$. Kuo grąžinamas skaičius mažesnis, tuo gauta galinė išteklių būsena yra arčiau siekiamos būsenos (atstumo prasmė). Kadangi tikslo vertinimo logika dažnai priklauso nuo sprendžiamos problemos ir gali kisti, naudojama

sąsaja, kuri užtikrina lengvą papildomų vertinimo būdų išplėtimą. Toliau pateikti prototipe įgyvendinti vertinimo būdai:

– Vertinamo pagal išteklių skaičių klasė (*ByNumberOfResources*);
– Vertinamo pagal santykinius nuotolius klasė (*ByPercent*);
– Vertinamo pagal išteklių skaičių ir santykinius nuotolius klasė (*ByNumberOfResourcesAndPercent*);



16 pav. Prototipo dalies išplėstinių komponentų UML klasių diagrama

.

Asociacija – vienos klasės objektai naudoja kitos klasės objektus; paveldėjimas nusako hierarchiją – palikuonių klasė paveldi tėvinės klasės atributus ir operacijas; realizacija – ryšys tarp sąsajos ir konkrečios klasės, kuri įgyvendina sąsajoje aprašytus atributus ir operacijas.

Detalūs būsenų keitiklio (diagramoje *StateExpander*) ir algoritmo klasių aprašymai pateikti atitinkamai 2 ir 3 lentelėse:

2 lentelė. Būsenos keitiklio (diagramoje *StateExpander*) klasės elementų aprašymas

Elementas	**Aprašymas**
NumberOfIterationsPerAverage	Apibrėžiama, kiek iteracijų reikia atlikti su identiškomis įvesties parametrų reikšmėmis prieš atliekant gautų rezultatų vidurkinimą.
StateExpanderAlgorithm	Apibrėžiamas paieškos algoritmas.
_currentAverageState	Saugoma naujausia būsena, kuri gaunama atlikus būsenų vidurkinimą per apibrėžtą iteracijų skaičių. Būsena yra sudaryta iš įvesties parametrų reikšmių rinkinio, išteklių būsenos (išteklių rinkinio) ir tikslo įverčio. Vidurkinamų būsenų įvesties parametrų reikšmių rinkinys yra identiškas, todėl vidurkinimas atliekamas tik išteklių būsenų atžvilgiu. Atlikus vidurkinimą, remiantis gauta išteklių būsena priskiriamas tikslo įvertis.
_initialVariablesCollection	Saugomi apibrėžti įvesties parametrai.
_numberOfIterations	Apibrėžiamas bendras iteracijų skaičius, kuris atliekamas su unikaliomis įvesties parametrų reikšmėmis (nemaišyti su iteracijų skaičiumi per vidurkinimą).
_resourcesToAverage	Laikinai saugomi ištekliai iki kol atliekamas būsenų vidurkinimas.
AddVar(<...>)	Metodas, kuris naudojamas priskirti simuliacinio modelio veiklų aprašyme apibrėžtus parametrus į _initialVariablesCollection elementą.
Next()	Klasės metodas, kuris nurodo tolesnį paieškos žingsnį: • Jeigu vidurkinumui skirtų iteracijų skaičius nepasiektas, atlikti verslo proceso simuliaciją su esamu įvesties paramatrų reikšmių rinkiniu; • Jeigu iteracijų skaičius pasiektas, remiantis paieškos algoritmu gauti naują įvesties parametrų reikšmių rinkinį.
GetAverageResources(<...>)	Statinis klasės metodas, kuris naudojamas atlikti išteklių būsenos vidurkinimą.

3 lentelė. Algoritmo klasės elementų aprašymas

Elementas	**Aprašymas**
ComputedStates	Saugomos visos apskaičiuotos būsenos.
CurrentStateToCompute	Saugoma sekanti skaičiavimui pateikta būsena. Tokia būsena turi priskirtas įvesties parametrų reikšmes, tačiau neturi išteklių būsenos. Atlikus verslo proceso simuliaciją gaunamos galinės išteklių rinkinio vertės, kurios priskiriamos šiai būsenai.
GoalEvaluator	Apibrėžiamas išteklių būsenos vertinimo metodas.
Next()	Vykdomas sekantis paieškos algoritmo žingsnis ir pateikiama nauja būsena (turinti naują įvesties parametrų reikšmių rinkinį) skaičiavimui.
OnCurrentStateComputed()	Įvykis, pranešantis, kad būsena pilnai apskaičiuota. Naudojama duomenų įrašymui į failą ar pateikimui į ekraną.

4.2. Kiti komponentai

Be 16 paveiksle pavaizduotų klasių buvo sukurti ir kiti komponentai:

- Parametro klasė. Parametrui suteikiamas pavadinimas ir reikšmė. Taip pat apibrėžtos parametrų tarpusavio matematinės operacijas, pavyzdžiui, sudėtis, atimtis ir t. t. Šias operacijas galima atlikti tik su parametrais, turinčiais tuos pačius pavadinimus;
- Parametrų rinkinio klasė. Parametrų rinkinį sudaro aibė parametrų, kurių pavadinimai turi būti skirtingi. Ši klasė realizuota naudojant *IList* sąsają siekiant, kad parametrų rinkinį būtų galima naudoti cikluose;
- Būsenos klasė. Būsenos klasė naudoja parametrų rinkinių, išteklių rinkinio bei tikslo įvertinimo funkcijos reikšmės duomenų tipus. Tokiu būdu su konkrečiu parametrų reikšmių rinkiniu susiejama konkretus išteklių rinkinys (arba būsena) bei įvertinimo funkcijos vertė. Kaip minėta anksčiau, įvertinimo funkcijos vertė apskaičiuojama naudojant išteklių būseną;
- Rezultatų įrašymo klasė. Naudojama įrašyti simuliacijos rezultatus į išorinį failą.

Darbe [36] buvo sukurtas išteklių duomenų modelis, realizuotas MS SQL duomenų bazėje ir pritaikytas „DBPSim“ simuliacijos įrankiui. Šiame darbe naudojant minėtą išteklių modelį apibrėžiami naudojami ištekliai, priskiriami atributai ir pradinės reikšmės. Taip pat modifikuota šio modelio *ResourceAttribute* reliacija, sukuriant papildomą atributą *ResourceAttributeState*. Šiuo atributu nusakomos ištekliaus būsenos tipas – pradinė, galinė ar siekiama būsena.

Taip pat simuliacijos įrankio taisyklių šablono klasėje buvo aprašytos papildomos funk-

cijos. Taisyklių šablono funkcijos yra naudojamos aprašant veiklas. Toliau pateikta naujai aprašytų funkcijų paskirtis:

4 lentelė. Taisyklių šablono klasės papildomos funkcijos

Funkcija	**Aprašymas**
UseVar(<...>)	Jeigu vykdoma pirmoji iteracija, tada sukuria apibrėžtą parametrą ir priskiria jį parametrų rinkiniui (naudoja būsenų keitiklio AddVar(<...>) funkciją). Kaip įvestis nurodamas parametro pavadinimas, pradinė reikšmė bei ar tai sveikoji vertė. Jeigu tai nėra pirmoji iteracija, grąžina parametro reikšmę (naudoja algoritmo klasės CurrentStateToCompute kintamąjį)
LoadNext()	Paprastai kviečiama paskutiniame simuliacinio modelio veiklos apraše. Priskiria simuliacijos metu gautą galinę išteklių būseną konkrečiam parametrų reikšmių rinkiniui ir pakeičia parametrų rinkinio reikšmes (naudoja būsenų keitiklio Next() funkciją).
ClearVars()	Kviečiama tik tuo atveju, jeigu reikia panaikinti simuliacijoje naudojamą parametrų reikšmių rinkinį.
GetResourceValue(<...>)	Naudojama gauti tam tikro ištekliaus vertę. Funkcijos įvestis yra ištekliaus pavadinimas ir atributas.
UseResourceValue(<...>)	Naudojama sunaudoti tam tikro ištekliaus vertę. Funkcijos įvestis yra ištekliaus pavadinimas, atributas ir sunaudojama vertė.
SupplyResourceValue(<...>)	Naudojama papildyti tam tikro ištekliaus vertę. Funkcijos įvestis yra ištekliaus pavadinimas, atributas ir pildoma vertė.

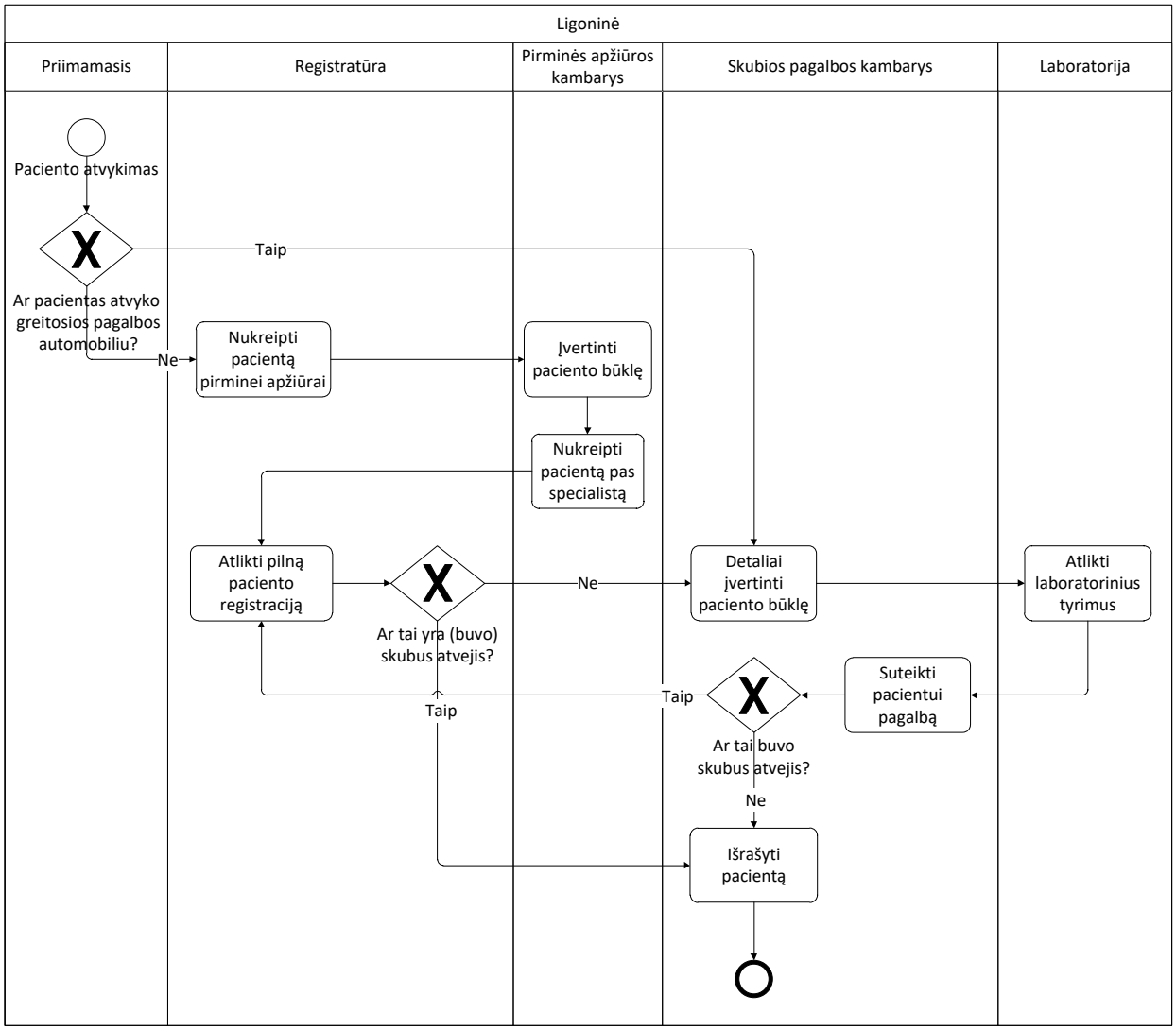
4.3. Apibendrinimas

Siūlomas metodas realizuotas praplečiant „DBPSim“ simuliacijos įrankį. Prototipui sukurti buvo naudojama C# programavimo kalba, .NET 4.6.1 karkaso technologijos. Esminiai išplėstiniai komponentai: paieškos valdymo klasė, algoritmo abstrakčioji klasė su palikuonių klasėmis, tikslo vertinimo sąsaja su realizacijos klasėmis. Išplėstiniams komponentams palaikyti buvo sukurtos papildomos pagalbinės klasės bei modifikuota taisyklių šablono klasė.

5. EKSPERIMENTINIS TYRIMAS

5.1. Tiriamas verslo procesas

Šiame skyriuje atliekamas dinaminio verslo proceso optimizavimo tyrimas. Kaip minėta anksčiau, verslo proceso simuliacinis modelis yra parametrizuojamas ir vykdoma išteklių optimizacija atsižvelgiant į siekiamą jų būseną. Optimizacija realizuojama naudojant 4–ąjame skyriuje aprašytą prototipą. Tyrime naudojamas verslo procesas aprašo skubios pagalbos suteikimo procedūrą ligoninėje (17 pav.). Šis procesas suformuotas remiantis kitų autorių darbuose naudojamu panašiu procesu [37].



17 pav. Skubios pagalbos suteikimo ligoninėje verslo procesas

.

Proceso vykdymas prasideda nuo paciento atvykimo. Pacientai skirstomi į grupes priklausomai nuo jų būklės: savarankiškai į priėmimo skyrių atvykę pacientai (toliau tokius pacientus žymėsime A) ir sunkesnės būklės pacientai, atvykę greitosios pagalbos automobiliu (žymėsime B). Paciento būklė lemia verslo proceso veiklų vykdymo tvarką – B tipo

pacientams gydymas ir tyrimai atliekami iš karto, tuo metu A tipo pacientai pirmiausia apžiūrimi seselės ir užregistruojami pas reikalingą specialistą. Toliau 5 lentelėje pateikti minėto proceso vaidmenys ir veiklų atlikimo trukmės:

5 lentelė. Skubios pagalbos suteikimo ligoninėje verslo proceso veiklų detalės

Veikla	**Vaidmuo**	**Aptarnavimo laikas (min)**
Nukreipti pacientą pirminei apžiūrai	Administratorius	5-15
Įvertinti paciento būklę	Seselė	10-20
Nukreipti pacientą pas specialistą	Seselė	5-15
Atlikti pilną paciento registraciją	Administratorius	10-20
Detaliai įvertinti paciento būklę	Gydytojas, seselė	15-30
Atlikti laboratorinius tyrimus	Laborantas	30-60
Suteikti pacientui pagalbą	Gydytojas, seselė	45-120

Simuliuojant procesą laikoma, kad vienas darbuotojas vienu metu gali aptarnauti tik vieną pacientą. Jeigu esamu momentu nėra laisvų darbuotojų, pacientas turi laukti, kol jie atsilaisvins. Toliau 6 lentelėje pateiktos verslo proceso vaidmenų (pareigybių) kainos už darbo valandą:

6 lentelė. Skubios pagalbos suteikimo ligoninėje verslo proceso išlaidos

Vaidmuo	**Kaina per valandą (Eur)**
Administratorius	15
Seselė	15
Gydytojas	45
Laborantas	20

5.2. Eksperimentinis tyrimas

Šio eksperimentinio tyrimo tikslas yra minimalizuoti anksčiau aprašyto verslo proceso vykdymo trukmę ir kainą naudojant kompiuterinę simuliaciją ir optimizaciją. Tyrimas apima skirtingų pacientų skaičiaus atvejų nagrinėjimą. Simuliacinis modelis yra parametrizuotas – parametrai nusako procese dalyvaujančių vaidmenų (pareigybių) kiekį. Šių vaidmenų pradinės reikšmės apibrėžtos 7 lentelėje.

Pradžioje atliekama modelio simuliacija be optimizacijos t. y. įvesties parametrų reikšmės nekinta. Tokiu būdu siekiama įvertinti, kokia yra įprastinė proceso vykdymo trukmė ir kaina. Kiekvienu pacientų skaičiaus atveju simuliacija vykdoma 200 kartų ir gauti rezultatai vidurkinami. Šio tyrimo rezultatai pateikti 8 lentelėje.

7 lentelė. Simuliacinio modelio pradinės parametrų reikšmės

Vaidmuo	Pradinis kiekis
Seselė	7
Gydytojas	3
Laborantas	4
Administratorius	4

8 lentelė. Proceso vykdymo trukmės ir kainos priklausomybė nuo paciento skaičiaus esant fiksuotoms įvesties parametrų reikšmėms (optimizacija netaikoma)

A tipo pacientų sk.	B tipo pacientų sk.	Seselių sk.	Gydytojų sk.	Laborantų sk.	Admin. sk.	Išlaidos (Eur.)	Trukmė (val.)
20	2	7	3	4	4	10241	23,47
40	4					20982	48,15
60	6					32007	72,88
80	8					41134	95,74
100	10					52477	119,19

Iš gautų rezultatų matyti, kad vidutinė proceso vykdymo trukmė ir kaina tiesiškai priklauso nuo pacientų skaičiaus. Taip pat remiantis šiais rezultatais apibrėšime siekiamas proceso vykdymo trukmės ir kainos ribas, kurias norime gauti atlikus optimizaciją. Šio darbo tyrime teigsime, kad norime sumažinti proceso vykdymo trukmę ir kainą 10 procentų. Tokiu atveju suapvalintos ribos pateiktos 9-oje lentelėje.

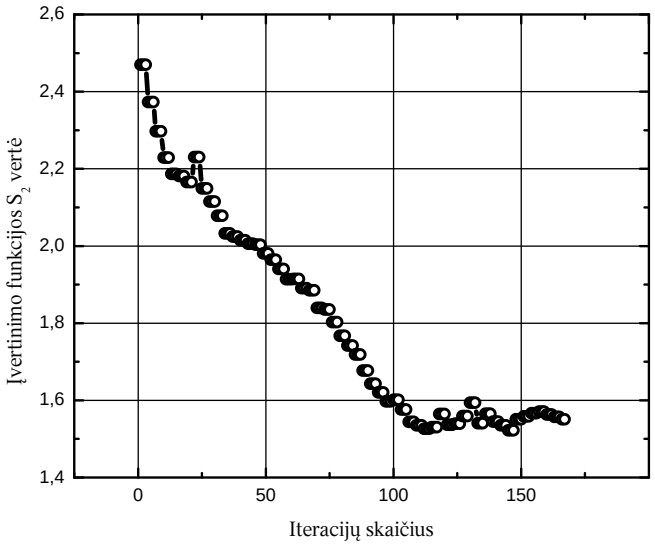
Optimizacijos metu keičiamos pradinės įvesties parametrų reikšmės (pareigybių kiekiai) siekiant surasti tokią jų konfigūraciją, kuri leistų pasiekti minėtas ribas. Kuo didesnis pacientų skaičius ir kuo mažesnis pareigybių skaičius, tuo proceso vykdymo trukmė yra ilgesnė. Tokiu būdu vykdymo trukmę galima sumažinti padidinus pareigybių skaičių. Iš kitos pusės, didesnis pareigybių skaičius lemia didesnes išlaidas. Taigi tokio proceso optimizacija nėra akivaizdi.

Verslo proceso optimizacija atliekama naudojant simplekso ir genetinės paieškos algoritmus bei taikant anksčiau aprašytą S_2 tipo tikslo įvertinimo funkciją. Tyrumo rezultatai pateikti 10, 11 lentelėse bei 19 paveiksle. Remiantis gautais rezultatais, galima teigti, kad įgyvendintas optimizacijos būdas iš tiesų leidžia sumažinti proceso vykdymo trukmę ir kainą. Nors atlikus optimizaciją ne visada ištekliai patenka į apibrėžtas ribas, tačiau bet kokiu atveju pokytis gaunamas siekiama kryptimi (18 pav.).

Taikant simplekso metodą tolygiai optimizuojama tiek proceso vykdymo trukmė, tiek kaina. Ryškesnę tendenciją galima įžvelgti analizuojant genetinės paieškos teikiamus rezultatus – šio tipo paieška geba reikšmingiau sumažinti proceso vykdymo trukmę (nors kainos

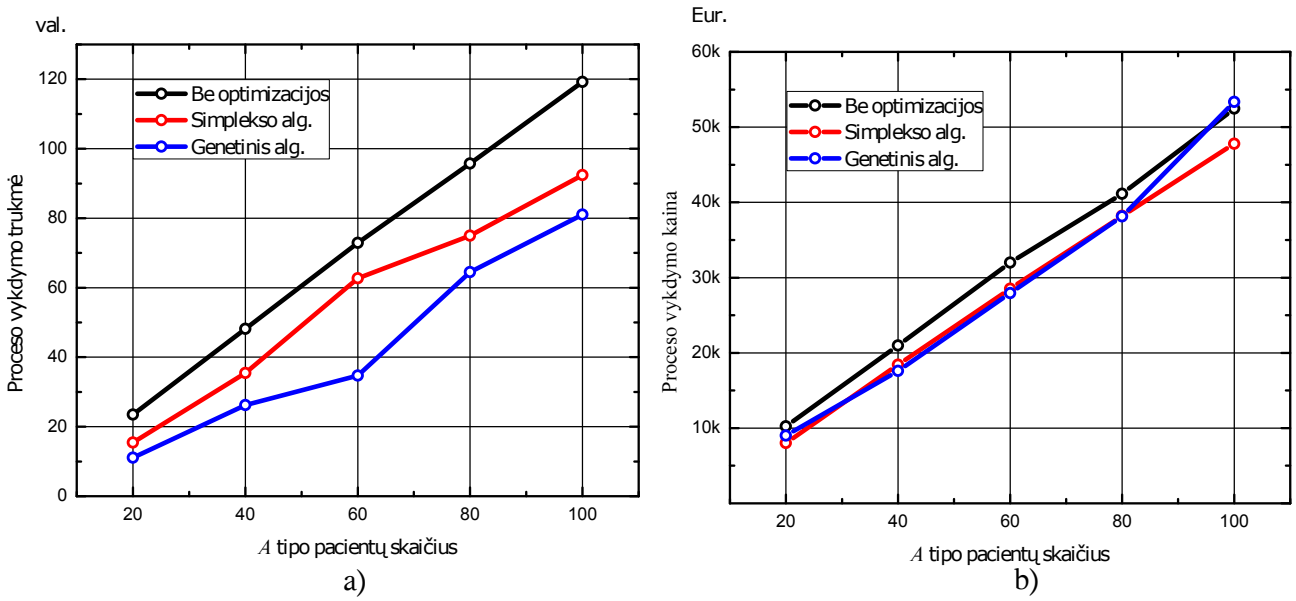
9 lentelė. Siekiamia proceso vykdymo trukmė ir kaina priklausomai nuo pacientų skaičiaus

A tipo pacientų sk.	**B tipo pacientų sk.**	**Siekiama proceso kaina (iki Eur.)**	**Siekiama proceso trukmė (iki val.)**
20	2	9200	21
40	4	19300	44
60	6	29700	67
80	8	37900	86
100	10	48100	110



18 pav. Įvertinimo funkcijos S_2 verčių priklausomybė nuo iteracijų skaičiaus taikant genetinę paiešką ir esant 60 pacientų kiekio atvejui

mažinimas išlieka panašaus efektyvumo, o vienu atveju ji viršija įprastinę). Todėl galima teigti, kad simplekso paieška labiau linkusi sustoti ties lokalių sprendimų radimu, kurie nėra tokie artimi tikslui kaip genetinės paieškos atveju. Simplekso paieškoje didelę įtaką turi pradinis parametrų reikšmių rinkinys, tuo metu genetinėje paieškoje turi įtakos atsitiktinumo faktorius. Pavyzdžiui, su tam tikra tikimybe atrenkami populiacijos nariai, atsitiktinai parenkamas paveldėjimo ir mutacijos (kuri irgi vyksta su tam tikra tikimybe) taškas. Tokiu būdu taikant genetinę paiešką pavyksta gauti įvairesnes parametrų reikšmių konfigūracijas, kurios simplekso algoritmui nepasiekiamos. Kartais ši suoptimizuota reikšmių konfigūracija atitinka didelį pradinį pareigybių skaičių, kuris leidžia greitai įvykdyti procesą. Tai aiškiausiai matosi lyginant optimizuotą seselių skaičių, kuris genetinės paieškos atveju yra žymiai didesnis. Tačiau toks ryškus pareigybių kiekių padidinamas taip pat didina riziką, kad ženkliai kils proceso vykdymo kaina. Kitaip sakant, genetinė paieška gali generuoti tiek labai



19 pav. Proceso vykdymo trukmės (a) ir kainos (b) priklausomybė nuo pacientų skaičiaus. Raudonai pažymėta simplekso, mėlynai – genetinė paieška.

10 lentelė. Proceso vykdymo trukmės ir kainos priklausomybė nuo paciento skaičiaus taikant simplekso algoritmą. Žalia spalva žymi išteklių reikšmių patekimą, raudona – nepatekimą į siekiamas ribas.

A tipo pacientų sk.	B tipo pacientų sk.	Seselių sk.	Gydytojų sk.	Laborantų sk.	Admin. sk.	Išlaidos (Eur.)	Trukmė (val.)
20	2	7	4	4	6	7999	15,47
40	4	7	4	4	5	18405	35,48
60	6	6	3	3	8	28528	62,69
80	8	7	4	4	5	38188	74,95
100	10	7	4	4	5	47793	92,42

11 lentelė. Proceso vykdymo trukmės ir kainos priklausomybė nuo paciento skaičiaus taikant genetinę optimizaciją

A tipo pacientų sk.	B tipo pacientų sk.	Seselių sk.	Gydytojų sk.	Laborantų sk.	Admin. sk.	Išlaidos (Eur.)	Trukmė (val.)
20	2	12	6	7	5	8991	11,09
40	4	14	4	4	8	17605	26,22
60	6	12	6	7	5	27943	34,72
80	8	10	4	4	8	38128	64,47
100	10	14	4	4	8	53342	81,03

artimus, tiek tolimus siekiamam tikslui rezultatus. Priklausomai nuo sprendžiamos problemos pobūdžio ir naudojamos būsenų įvertinimo funkcijos, jeigu tiriama erdvė turi daug lokalių minimumų, tokie genetinio algoritmo teikiami rezultatai gali būti didelis privalumas prieš simplekso paiešką. Apibendrinus gautus rezultatus, tyrimas parodė, kad optimizacijai taikant simplekso algoritmą proceso vykdymo trukmė vidutiniškai sumažėjo 24 %, o kaina 13 %. Tuo metu taikant genetinį algoritmą trukmė vidutiniškai sumažėjo 43 %, o kaina 9 %.

Turint šiuos rezultatus gali būti atliktas papildomas sekantis tyrimas, kuris pagrįstas prielaidomis, kad proceso vykdymo trukmė ir kaina tiesiškai priklauso nuo pacientų skaičiaus ir sukurtas optimizacijos metodas yra netobulas. Jeigu galioja pirmoji prielaida ir metodas būtų tobulas, tada visais atvejais, nepriklausomai nuo pacientų skaičiaus, turėtume gauti vienodą optimizuotą įvesties parametrų reikšmių rinkinį. Kaip matyti iš rezultatų, to paties algoritmo teikiamos vertės yra panašios, bet ne identiškos. Šio tyrimo tikslas – pagerinti gautus rezultatus naudojant artimiausią tikslui gautą optimizuotą parametrų reikšmių rinkinį visiems pacientų kiekiams tirti (toks rinkinys nustatomas vertinant santykinį nuotolį tarp įprastos ir optimizuotos verčių). Šiuo atveju tai rinkinys, gautas taikant genetinę paiešką ir esant 60 pacientų kiekiui. Tokio tyrimo rezultatai pateikti 12 lentelėje.

12 lentelė. Proceso vykdymo trukmės ir kainos priklausomybė nuo paciento skaičiaus naudojant artimiausią tikslui gautą reikšmių rinkinį

A tipo pacientų sk.	**B tipo pacientų sk.**	**Seselių sk.**	**Gydytojų sk.**	**Laborantų sk.**	**Admin. sk.**	**Išlaidos (Eur.)**	**Trukmė (val.)**
20	2	12	6	7	5	8991	11,09
40	4					17716	22,32
60	6					27943	34,72
80	8					36249	45,42
100	10					46730	59,79

Bendrai kalbant, tokio parametrų rinkinio naudojimas pasiteisino – tyrimo rezultatai yra artimesni tikslui už visus iki tol gautus rezultatus. Lyginant šį parametrų reikšmių rinkinį su pradiniu rinkiniu, matyti, kad seselių, gydytojų ir laborantų skaičius praktiškai padvigubėjo, o administratorių – pasikeitė minimaliai. To priežastis – administratoriai nedalyvauja pačiame gydymo procese. Jeigu jų kiekis ir būtų žymiai didesnis, tai pacientai greičiau susiburtų tik ties pirminės apžiūros kambariu (nes šiai veiklai vykdyti reikia seselių) ar skubios pagalbos kambariu (reikia gydytojų ir seselių), bet paties proceso vykdymo trukmė pakistų nežymiai. Daugiausiai laiko užima gydymo veiklos, pavyzdžiui, pagalbos suteikimas pacientui, kuriose dalyvauja seselės ir gydytojai, todėl šių pareigybių kiekis reikšmingai riboja viso proceso vykdymo trukmę. Iš kitos pusės, kaip minėta anksčiau, visų pareigybių skaičius

neproporcingai didinti negalima, nes tai augina proceso vykdymo kainą.

5.3. Apibendrinimas

Pasiūlytas metodas įgyvendintas praplečiant „DBPSim“ simuliacijos įrankį. Naudojant šį plėtinį, buvo atlikti eksperimentiniai tyrimai. Gauti rezultatai parodė, kad abu realizuoti euristiniai algoritmai geba optimizuoti įvesties parametrų reikšmes. Algoritmų efektyvumas yra skirtingas, tačiau jie turi savo privalumų: simplekso algoritmas šiuo proceso atveju geba tolygiai sumažinti proceso vykdymo trukmę ir kainą, tačiau genetinė paieška teikia įvairesnes galimas konfigūracijas, kas leidžia išvengti sustojimo ties lokaliais ekstremumais.

IŠVADOS

- Atlikus literatūros analizę paaiškėjo:
  - Vienas iš būdų optimizuoti verslo procesą ir jame naudojamus išteklius galimas tada, kai aprašyto proceso modelis remiasi griežtu matematiniu pagrindu. Tačiau tokie modeliai yra brangūs ir ne visada įmanomi;
  - Kitas verslo procesų optimizavimo būdas yra naudoti kompiuterinę simuliaciją;
- Remiantis literatūros analizę buvo pasiūlytas metodas, pagrįstas simuliacinio modelio parametrizavimu ir euristinės paieškos taikymu, kuris leidžia optimizuoti dinaminiuose verslo procesuose naudojamus išteklius atsižvelgiant į siekiamą išteklių būseną;
- Atliktas eksperimentinis tyrimas parodė, kad pasiūlytas metodas gali optimizuoti nagrinėjamo dinaminio verslo proceso išteklius. Optimizacijai taikant simplekso algoritmą proceso vykdymo trukmė vidutiniškai sumažėjo 24 %, o kaina 13 %. Tuo metu taikant genetinį algoritmą trukmė vidutiniškai sumažėjo 43 %, o kaina 9 %.

TOLIMESNIŲ TYRIMŲ KRYPTYS

1. Išbandyti kitus euristinius paieškos algoritmus ir tikslo vertinimo funkcijas;
2. Atlikti realaus verslo proceso išteklių optimizacijos tyrimą;
3. Patobulinti vartotojo sąsają suteikiant galimybę apibrėžti parametrus ir siekiamas išteklių būsenas atskirame lange.

LITERATŪRA

[1] B. von Halle, *Business Rules Applied: Building Better Systems Using the Business Rules Approach* (Wiley Publishing, 2002).

[2] A. Macioł, An Application of Rule-Based Tool in Attributive Logic for Business Rules Modeling, *Expert Systems with Applications* 34(3), 1825–1836 (2008).

[3] O. Vasilecas, D. Kalibatienė, G. Guizzardi, Towards a Formal Method for the Transformation of Ontology Axioms to Application Domain Rules, *Information Technology and Control* 38(4), 271–282 (2009).

[4] D. Kalibatienė, O. Vasilecas, Ontology Axioms for the Implementation of Business Rules, *Technological and Economic Development of Economy* 16(3), 471–486 (2010).

[5] The Business Process Model, Sparx Systems, Prieiga per internetą: http://www.sparxsystems.com/business_process_model.html [žiūrėta 2016-11-24].

[6] K. Stankevičius, O. Vasilecas, Research on Rules-Based Business Process Modelling and Simulation, *Mokslas - Lietuvos ateitis* 6(2), 147–150 (2014).

[7] A. Rima, O. Vasilecas, Verslo procesų simuliacija: verslo procesų ir išteklių modelių reikalavimai, *Mokslas - Lietuvos ateitis* 7(3), 291–295 (2015).

[8] O. Vasilecas, D. Kalibatienė, D. Lavbič, Rule- and Context-Based Dynamic Business Process Modelling and Simulation, *Journal Of Systems and Software* 122(C), 1–15 (2016).

[9] Business Process, Pega, Prieiga per internetą: http://www.pega.com/business-process [žiūrėta 2016-12-23].

[10] T. Rusinaitė, D. Kalibatienė, O. Vasilecas, Requirements of Dynamic Business Processes – a Survey, in *2015 IEEE 3rd Workshop on Advances in Information, Electronic and Electrical Engineering* (2015), 1–4.

[11] J. Trinkūnas, T. Rusinaitė, O. Vasilecas, Research on Improving Dynamic Business Procesess in HIS, in *24th International Conference on Information Systems Development* (2015), 1–11.

[12] M. Rouse, Dynamic BPM (Business Process Management), WhatIs.com, Prieiga per internetą: http://whatis.techtarget.com/definition/dynamic-BPM-business-process-management [žiūrėta 2017-01-06].

[13] M. Cooper, P. Patterson, Business Process Management (BPM) Definition and Solutions, CIO, Prieiga per internetą: http://www.cio.com/article/2439162/

business-process-management/business-process-management--bpm--definition-and-solutions.html [žiūrėta 2016-11-14].

[14] J. Holt, *A Pragmatic Guide to Business Process Modelling* (British Computer Society, 2009).

[15] R. Grašys, O. Vasilecas, Verslo procesų simuliavimo priemonių analizė ir vertinimas, *Mokslas - Lietuvos ateitis* 6(2), 157–161 (2014).

[16] W. M. P. van der Aalst, Business Process Simulation Revisited, in *Enterprise and Organizational Modeling and Simulation: 6th International Workshop* (2010), 1–14.

[17] M. Laguna, J. Marklund, *Business Process Modeling, Simulation, and Design* (Pearson/Prentice Hall, 2013).

[18] W. M. P. van der Aalst, *Business Process Simulation Survival Guide* (Springer Berlin Heidelberg, 2015), 337–370.

[19] K. Casier, A. Gouwy, S. Verbrugge, D. Colle, M. Pickavet, P. Demeester, Using Discrete Event Simulation to Reduce Operational Expenses for a Telecom Operator, in *Networks 2008 – The 13th International Telecommunications Network Strategy and Planning Symposium* (2008), 1–8.

[20] A. Rima, *Taisyklėmis grindžiamų veiklos procesų su bendrai naudojamais ištekliais modeliavimo ir simuliacijos tyrimas*, Daktaro disertacija, Vilniaus Gedimino technikos universitetas (2016).

[21] S. Smith, M. Becker, An Ontology for Constructing Scheduling Systems, in *Working Notes of 1997 AAAI Symposium on Ontological Engineering* (1997), 120–127.

[22] F. G. Fadel, M. S. Fox, M. Gruninger, A Generic Enterprise Resource Ontology, in *Proceedings of 3rd IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* (1994), 117–128.

[23] O. Vasilecas, T. Vysockis, T. Rusinaitė, On Goal-Oriented Business Process Simulation, in *2016 IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)* (2016), 1–4.

[24] O. Vasilecas, T. Savickas, K. Normantas, T. Vysockis, D. Kalibatienė, A Goal-Oriented Approach to Dynamic Business Process Simulation, in *2016 Databases and Information Systems IX : Conference Proceedings of 12th International Baltic Conference on Databases and Information Systems* (2016).

[25] M. Fasbinder, WebSphere Business Modeler Advanced Simulation, IBM, Prieiga per internetą: https://www.ibm.com/developerworks/websphere/techjournal/0708_fasbinder/0708_fasbinder.html [žiūrėta 2017-06-19].

[26] WebSphere Business Modeler Basic, IBM, Prieiga per internetą: https://www.ibm.com/support/knowledgecenter/en/SSBJCK_7.0.0/com.ibm.btools.modeler.basic.intro.doc/overview/benefits.html [žiūrėta 2017-06-19].

[27] Why Simprocess?, Simprocess, Preiga per internetą: http://simprocess.com/about-simprocess/why-simprocess/ [žiūrėta 2017-06-19].

[28] SIMUL8 Help Center: Resources, Simul8, Prieiga per internetą: https://www.simul8.com/support/help/doku.php?id=model_building_basics:simulation_objects:resources [žiūrėta 2017-06-19].

[29] K. Vergidis, A. Tiwari, B. Majeed, Business Process Analysis and Optimization: Beyond Reengineering, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38(1), 69–82 (2008).

[30] J. H. Mathews, K. D. Fink, *Numerical methods using MATLAB* (Simon & Schuster, 1998).

[31] F. Gao, L. Han, Implementing the Nelder-Mead Simplex Algorithm with Adaptive Parameters, *Computational Optimization and Applications* 51(1), 259–277 (2012).

[32] Z. Michalewicz, C. Z. Janikow, Genetic Algorithms for Numerical Optimization, *Statistics and Computing* 1(2), 75–91 (1991).

[33] L. Zeng, D. Flaxer, H. Chang, J.-J. Jeng, PLM Flow-Dynamic Business Process Composition and Execution by Rule Inference, in *TES* (2002), 141–150.

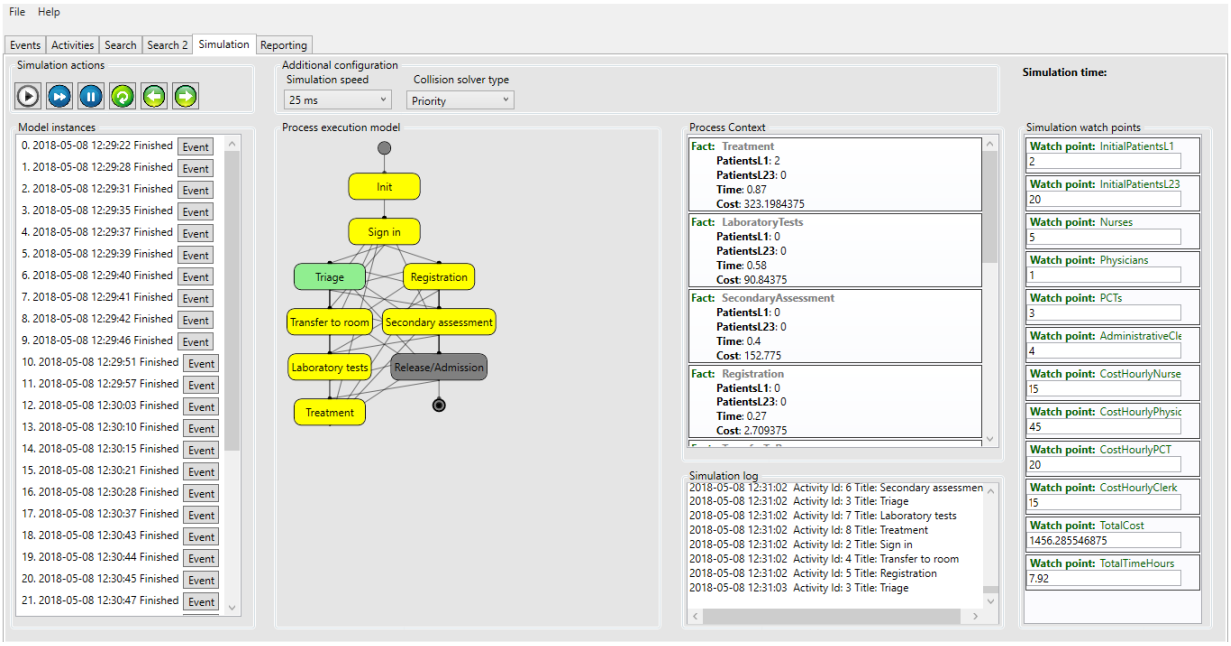
[34] K. Vergidis, D. Saxena, A. Tiwari, An Evolutionary Multi-Objective Framework for Business Process Optimisation, *Applied Soft Computing* 12(8), 2638–2653 (2012).

[35] M. Rameika, *Taisyklėmis grindžiamų dinaminių verslo procesų modeliavimo ir simuliacijos būdas*, Magistro tezės, Vilniaus Gedimino technikos universitetas (2014).

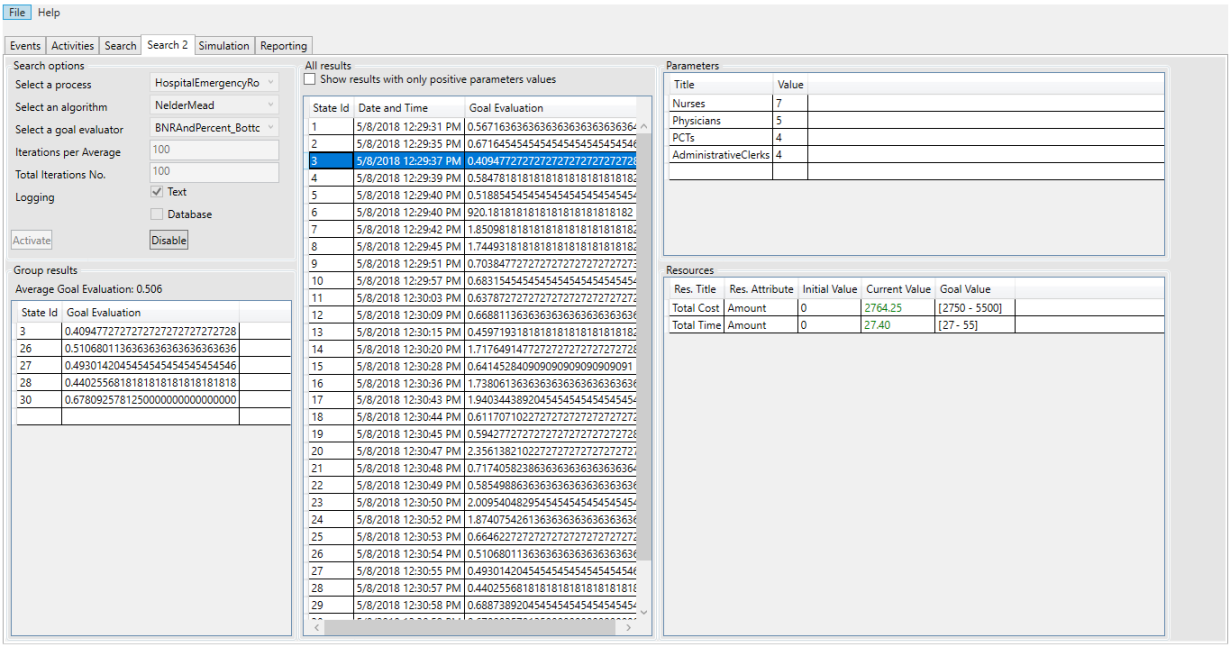
[36] T. Vysockis, *Dinaminių verslo procesų išteklių modeliavimas ir simuliavimas*, Magistro tezės, Vilnius: Vilniaus Gedimino technikos universitetas (2016).

[37] M. A. Ahmed, T. M. Alkhamis, Simulation Optimization for an Emergency Department Healthcare Unit in Kuwait, *European Journal of Operational Research* 198(3), 936–942 (2009).

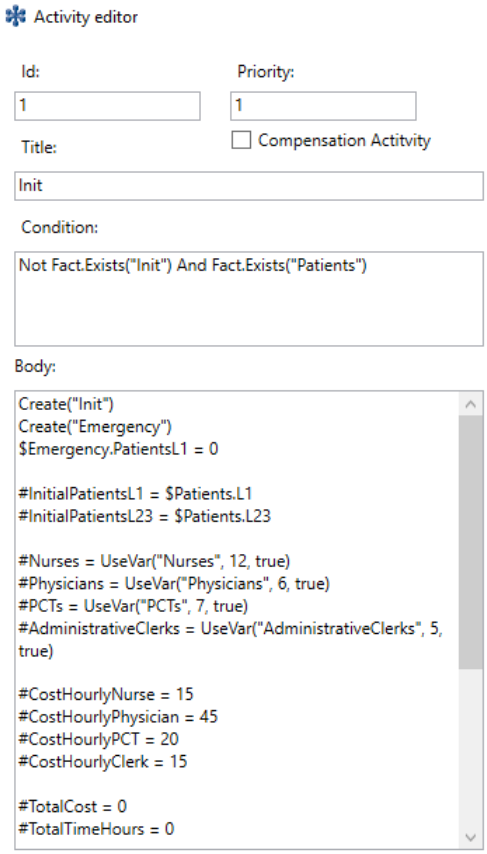
PRIEDAI



20 pav. Verslo proceso simuliacijos procesas



21 pav. Verslo proceso optimizacijos rodikliai



22 pav. Veiklos aprašo pavyzdys, kuriame apibrėžiami įvesties parametrai su pradinėmis reikšmėmis (naudojama *UseVar* funkcija)