

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Aurimas GEDMINAS

# MEMBRANINIŲ SKAIČIAVIMŲ TECHNOLOGIJA DAIKTŲ INTERNETO ĮRENGINIŲ SAVITVARKAI

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,  
ELEKTROS IR ELEKTRONIKOS INŽINERIJA (T 001)

Disertacija rengta 2015–2020 metais Vilniaus Gedimino technikos universitete.

### **Vadovas**

prof. dr. Dalius NAVAKAUSKAS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, elektros ir elektronikos inžinerija – T 001).

Vilniaus Gedimino technikos universiteto Elektros ir elektronikos inžinerijos mokslo krypties disertacijos gynimo taryba:

### **Pirmininkas**

prof. dr. Šarūnas PAULIKAS (Vilniaus Gedimino technikos universitetas, elektros ir elektronikos inžinerija – T 001).

### **Nariai:**

prof. dr. Algirdas BAŠKYS (Vilniaus Gedimino technikos universitetas, elektros ir elektronikos inžinerija – T 001),

dr. Ofer HADAR (Negevo Ben-Guriono universitetas, Izraelis, elektros ir elektronikos inžinerija – T 001),

doc. dr. Olga KURASOVA (Vilniaus universitetas, informatikos inžinerija – T 007),

prof. dr. Vytautas URBANAVIČIUS (Vilniaus Gedimino technikos universitetas, elektros ir elektronikos inžinerija – T 001).

Disertacija bus ginama viešame Elektros ir elektronikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje **2020 m. rugpjūčio 28 d. 9 val.** Vilniaus Gedimino technikos universiteto senato posėdžių salėje.

Adresas: Saulėtekio al. 11, LT-10223 Vilnius, Lietuva.

Tel. +370 5 274 4956; fax +370 5 270 0112; el. paštas: doktor@vgtu.lt

Pranešimai apie numatomą ginti disertaciją išsiųsti 2020 m. liepos 27 d.

Disertaciją galima peržiūrėti VGTU talpykloje <http://dspace.vgtu.lt> ir Vilniaus Gedimino technikos universiteto bibliotekoje (Saulėtekio al. 14, LT-10223 Vilnius, Lietuva) bei Lietuvos mokslų akademijos Vrublevskių bibliotekoje (Žygimantų g. 1, LT-01102 Vilnius, Lietuva).

VGTU leidyklos TECHNIKA 2020-024-M mokslo literatūros knyga

© VGTU leidykla TECHNIKA, 2020

© Aurimas Gedminas, 2020

*aurimas.gedminas@vgtu.lt*

VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Aurimas GEDMINAS

# MEMBRANE COMPUTING TECHNOLOGY FOR SELF-ORGANIZATION OF DEVICES IN THE INTERNET OF THINGS

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES,  
ELECTRICAL AND ELECTRONIC ENGINEERING (T 001)



Vilnius LEIDYKLA TECHNİKA 2020

Doctoral dissertation was prepared at Vilnius Gediminas Technical University in 2015–2020.

### **Scientific supervisor**

Prof. Dr Dalius NAVAKAUSKAS (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001).

The Dissertation Defense Council of Scientific Field of Electrical and Electronic Engineering of Vilnius Gediminas Technical University:

### **Chairman**

Prof. Dr Šarūnas PAULIKAS (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001).

### **Members:**

Prof. Dr Algirdas BAŠKYS (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001),

Dr. Ofer HADAR (Ben-Gurion University of the Negev, Israel, Electrical and Electronic Engineering – T 001),

Assoc. Prof. Dr Olga KURASOVA (Vilnius University, Informatics Engineering – T 007),

Prof. Dr Vytautas URBANAČIUS (Vilnius Gediminas Technical University, Electrical and Electronic Engineering – T 001).

The dissertation will be defended at the public meeting of the Dissertation Defense Council of Electrical and Electronic Engineering in the Senate Hall of Vilnius Gediminas Technical University at **9 a. m. on 28 August 2020**.

Address: Saulėtekio al. 11, LT-10223 Vilnius, Lithuania.

Tel. +370 5 274 4956; fax +370 5 270 0112; e-mail: doktor@vgtu.lt

A notification on the intend defending of the dissertation was send on 27 July 2020.

A copy of the doctoral dissertation is available for review at VGTU repository <http://dspace.vgtu.lt> and at the Library of Vilnius Gediminas Technical University (Saulėtekio al. 14, LT-10223 Vilnius, Lithuania) and the Wroblewski Library of the Lithuanian Academy of Sciences (Žygimantų st. 1, LT-01102, Vilnius, Lithuania).

# Reziumė

Disertacijoje nagrinėjamos membraninių skaičiavimų ir daiktų interneto technologijos bei su jomis susieti metaeuristikos algoritmai, optimizavimo skruzdžių kolonijomis algoritmai, daiktų interneto informacijos valdymo ir skirstymo algoritmai. Membraniniai skaičiavimai – tai kompiuterių mokslų sritis, kurioje skaičiavimo idėjos ir modeliai abstrahuojami iš gyvų ląstelių struktūros. Daiktų internetas – nepriklausomų įrenginių rinkinys, kuris vartotojams pateikiamas kaip viena darni sistema (struktūra) skirta išmaniaisiais įrankiais jungtis prie jutiklių ir daugybės prietaisų, siekiant juos valdyti arba gauti pagrindinę jų informaciją. Daiktų interneto savitvarkos duomenų valdymo ir skirstymo problemos sprendimas yra aktualus siekiant, kad sistemos veiktų nenutrūkstamai ir darniai.

Šios disertacijos tyrimo objektas yra daiktų interneto savitvarka. Darbo tikslas – pasiūlyti originalų metaeuristika grįstą P sistemos sprendimą, sukuriant ir ištiriant membraninių skaičiavimų imitavimo ir testavimo sistemą daiktų interneto sąlygomis. Darbas skirtas prisidėti vystant darnias daiktų interneto sistemas. Atliekama analitinė membraninių skaičiavimų ir daiktų interneto literatūros apžvalga, pagrindžiant darbe iškelto problemos aktualumą ir galimus jos sprendimo būdus tolimesniems tyrimams. Remiantis atlikta apžvalga bei pritaikant natūralioje gamtoje vykstančius procesus pasiūloma nauja daiktų interneto savitvarkos max-min skruzdžių P sistema, metodika. Pasiūlytas metodas ištiriamas MATLAB™ programinėje įrangoje sukuriant virtualų tinklą imituojantį daiktų internetą.

Disertaciją sudaro įvadas, trys skyriai ir bendrosios išvados. Pirmame skyriuje pristatomos esminės teorinės žinios apie membraninius skaičiavimus ir daiktų internetą, analizuojamas saviorganizuojantis daiktų internetas, metaeuristikos algoritmai, formuluojami disertacijos uždaviniai, apibrėžiama daiktų interneto informacijos skirstymo ir valdymo problema. Antrame skyriuje pristatoma saviorganizuojančio daiktų interneto metodika grįsta gamtoje vykstančiais procesais, pateikiamos P sistemos evoliucijos taisyklės ir programos pseudokodas. Trečiame skyriuje yra suformuluojami kokybiniai vertinimo kriterijai, MATLAB™ programinėje įrangoje atliekami eksperimentai ištiriant daiktų interneto informacijos skirstymo ir valdymo max-min P skruzdžių sistemą ir pateikiami eksperimentinio tyrimo rezultatai.

Pagrindiniai disertacijos rezultatai paskelbti 5 mokslinėse publikacijose – 3 iš jų atspausdintos recenzuojamuose mokslo žurnaluose, 2 – konferencijų medžiagoje. Rezultatai viešinti 6 mokslinėse konferencijose.

# Abstract

In the thesis is analyzed membrane computing, Internet of Things technologies and related metaheuristic algorithms, ant colony optimization algorithms, Internet of Things information management, and distribution algorithms. Membrane computing is a field of computer science in which computational ideas and models are abstracted from the structure of biological cells. The Internet of Things is a set of independent devices that are presented to users as a single sustainable system (structure) for connecting smart tools to sensors and multiple devices to control them or obtain their basic information. In order for the systems to operate continuously and sustainably, it is necessary to solve the problems of self-management data routing in the Internet of Things.

The object of research of this dissertation is the self-organizing Internet of Things network. The aim of the thesis is to offer an original metaheuristic based P system solution by creating and investigating a system for simulating and testing membrane computing in the Internet of Things. The thesis is designed to contribute to the development of sustainable Internet of Things systems. An analytical review of the literature on membrane computing and the Internet of Things is presented, substantiating the relevance of the problem raised in the work and possible solutions for further research. Based on the presented literature review and adapting the processes taking place in nature, a new max-min ants' P system algorithm and methodology for self-management of the Internet of Things is proposed. The proposed method is investigated in MATLAB™ software by creating a virtual network simulating the Internet of Things.

The dissertation consists of an introduction, three chapters, and general conclusions. The first chapter presents essential theoretical knowledge about membrane computing and the Internet of Things, analyzes the self-organizing Internet of Things, metaheuristic algorithms, formulates thesis tasks, defines the problem of Internet of Things data routing. The second chapter introduces the methodology of self-organizing the Internet of Things based on natural processes, presents the rules of the evolution of the P system, and the pseudocode of the program. In the third chapter, qualitative evaluation criteria are formulated, in MATLAB™ software are performed experiments by investigating the max-min ants' P system for the distribution and management of information on the Internet of Things and presents the results of the experimental research.

The main results of the thesis were published in 5 scientific publications: three of them were printed in peer-reviewed scientific journals, two articles – in conference proceedings. The research results were presented in 6 scientific conferences.

---

# Žymėjimai

## Simboliai

### Bendrieji

- $a$  – skaliaras (skaičius);
- $\mathbf{a}$  – vektorius (skaičių stulpelis);
- $\mathbf{A}$  – matrica (daugiamatis skaičių masyvas);
- $\mathcal{A}$  – rinkinys (elementų aibė);
- $\alpha$  – klasė (kriterijus);
- $\cdot^{\diamond}$  – siekiama (tikroji) vertė;
- $\cdot^{\star}$  – geriausia (apskaičiuota) vertė;
- $\cdot^{\top}, \cdot^{\perp}$  – maksimumas ir minimumas.

### Dalykinės sritys

- $a_{ch}, b_{ch}$  – orkestratoriaus koordinatės;
- $d_{ij}$  – atstumas tarp mazgų  $i$  ir  $j$ ;
- $\Delta\tau_{ij}^{(r)}$  –  $r$ -toje iteracijoje tarp mazgų  $i$  ir  $j$  paliekamo feromono kiekis;
- $\Delta\tau_{ij}^{\star}$  – iteracijos arba eigos geriausios skruzdės paliekamo feromono kiekis;
- $E_{ij}$  – energijos sąnaudos perduoti duomenis tarp  $i, j$  mazgų;
- $E_{\text{init}}$  – pradinė jutiklio mazgo energija;

$\eta(\vartheta)$	– skruzdės jautrumas ankstesnei judėjimo kryptčiai;
$\eta_{ij}$	– euristinė informacija tarp miestų $i$ ir $j$ ;
$EL$	– jutiklio mazgo eksploatavimo laikas;
$L^\diamond$	– (tikrojo) trumpiausio kelio ilgis;
$L^{(l)}$	– $l$ -tosios skruzdės rasto kelio ilgis;
$L^\star$	– iki šiol rasto trumpiausio kelio ilgis;
$\mu$	– membranos struktūra;
$m$	– membranos;
$R_m$	– membranos $m$ struktūros sričių evoliucijos taisyklių rinkiniai;
$u, v$	– objektų aibės;
$\mathcal{N}_m^k$	– nelankytų mazgų aibė pasiekama $k$ -tajai skruzdei iš $m$ -tojo mazgo;
$p^\top, p^\perp$	– viršutinė ir apatinė tikimybės ribos;
$p_{ij}^{(l)}$	– tikimybė skruzdei iš mazgo $i$ pasirinkti mazgą $j$ ;
$p_R^{(l)}$	– skruzdės dauginimosi (reprodukcijos) tikimybė;
$II$	– P sistemos konstruktas;
$\varrho$	– skruzdės traukos feromonui lygis;
$\sigma$	– daiktų interneto mazgų grafas;
$\sigma$	– standartinis nuokrypis;
$\bar{\sigma}$	– vidutinis absoliutus nuokrypis;
$t_{ij}$	– vėlavimas tarp $ij$ mazgų;
$\tau^\top, \tau^\perp$	– viršutinė ir apatinė feromono kiekio ribos;
$\tau_0$	– pradinis feromono kiekis;
$\tau_{ij}, \tau(x, y)$	– tarp $ij$ mazgų ar $(x, y)$ taške esantis feromono kiekis;
$\vartheta$	– skruzdės judėjimo kryptties pokytis;
$w$	– verčių svoris;
$\zeta(\tau)$	– skruzdės jautrumas feromono koncentracijos lygiui.

## Santrumpos

CH	– orkestratorius (angl. <i>cluster head</i> );
D2D	– įrenginys-įrenginiui (angl. <i>device-to-device</i> );
D2H	– įrenginys-žmogui (angl. <i>device-to-Human</i> );
DI	– daiktų internetas;
DIS	– daiktų interneto savitvarka;
DNR	– deoksiribonukleorūgštis;
KPU	– keliaujančio prekeivio uždavinys;
LEACH	– mažai energijos reikalaujanti adaptacinė klasterių hierarchija (angl. <i>low-energy adaptive clustering hierarchy</i> );
MMSPS	– max-min skruzdžių P sistema;



- MMSS – max-min skruzdžių sistema;
- MS – membraniniai skaičiavimai;
- NP – nedeterministinis polinominis laikas (angl. *nondeterministic polynomial time*);
- OSK – optimizavimas skruzdžių kolonijomis;
- RDA – radijo dažnio atpažinimas;
- RFID – atpažinimas radijo dažniu (angl. *radio-frequency identification*);
- RPL – maršruto protokolas mažos galios ir nuostolingiems tinklams (angl. *routing Protocol for Low-Power and Lossy Networks*).

## Reikšminiai žodžiai<sup>1</sup>

<b>IEEE Taxonomy</b>	Part(s)
<ul style="list-style-type: none"> <li>– Communications technology               <ul style="list-style-type: none"> <li>– Communication systems                   <ul style="list-style-type: none"> <li>– Internet – Internet of Things</li> <li>– Machine-to-machine communications</li> <li>– Protocols – Routing protocols</li> </ul> </li> </ul> </li> <li>– Computational and artificial intelligence               <ul style="list-style-type: none"> <li>– Computational intelligence                   <ul style="list-style-type: none"> <li>– Evolutionary computation</li> </ul> </li> </ul> </li> <li>– Mathematics               <ul style="list-style-type: none"> <li>– Algorithms                   <ul style="list-style-type: none"> <li>– Heuristic algorithms</li> <li>– Ant colony optimization</li> </ul> </li> </ul> </li> <li>– Computers and information processing               <ul style="list-style-type: none"> <li>– Computer Science                   <ul style="list-style-type: none"> <li>– Programming                       <ul style="list-style-type: none"> <li>– Automatic programming</li> <li>– Performance analysis</li> </ul> </li> </ul> </li> <li>– Software                   <ul style="list-style-type: none"> <li>– Software packages – MATLAB</li> </ul> </li> </ul> </li> </ul>	<p>1.1 1.2 2.4 3.2</p> <p>1.1 1.2.1 1.2.2</p> <p>1.2 1.2.1 1.1 2.4 3.2</p> <p>1.2.3 2.1 2.1.1 2.1.2 2.3</p> <p>2.1 2.1.1 2.1.2 2.2</p> <p>1.2.3 2.1 2.1.1 2.2</p> <p>3.1</p> <p>3.2 3.3</p> <p>3.1 3.2 3.3</p>

<sup>1</sup>Reikšminiai žodžiai atskiriems disertacijos skyriams pateikiami anglų kalba siekiant tikslumo ir remiantis visuotinai pripažinta 2014 metų IEEE taksonomija (prieiga per internetą <[https://www.ieee.org/documents/taxonomy\\_v101.pdf](https://www.ieee.org/documents/taxonomy_v101.pdf)>).



---

# Turinys

ĮVADAS .....	1
Problemos formulavimas .....	1
Darbo aktualumas .....	2
Tyrimų objektas .....	3
Darbo tikslas .....	3
Darbo uždaviniai .....	3
Tyrimų metodika .....	4
Darbo mokslinis naujumas .....	4
Darbo rezultatų praktinė reikšmė .....	5
Ginamieji teiginiai .....	5
Darbo rezultatų aprobavimas .....	5
Disertacijos struktūra .....	6
Padėka .....	6
1. DAIKTŲ INTERNETO IR MEMBRANINIŲ SKAIČIAVIMŲ TECHNOLOGIJŲ APŽVALGA .....	7
1.1. Daiktų internetas .....	7
1.2. Savitvarka daiktų internete .....	12
1.2.1. Informacijos valdymas .....	14
1.2.2. Duomenų skirstymo algoritmų klasifikavimas .....	15
1.2.3. Bio-grįštas globalus optimizavimas .....	17

1.3. Membraninių skaičiavimų technologijos ir P sistema . . . . .	20
1.3.1. Membraniniai skaičiavimai – gamtinių skaičiavimų dalis . . . . .	22
1.3.2. P sistema . . . . .	23
1.3.3. Membraninių skaičiavimų taikymas . . . . .	25
1.4. Pirmojo skyriaus išvados ir disertacijos uždavinių formulavimas . . . . .	28
2. DAIKTŲ INTERNETO SAVITVARKOS METODIKA . . . . .	29
2.1. Tiriamų skruzdžių kolonijų matematiniai modeliai . . . . .	30
2.1.1. Optimizavimas skruzdžių kolonijomis . . . . .	31
2.1.2. Max-min skruzdžių sistema . . . . .	33
2.2. Nauja max-min skruzdžių P sistema . . . . .	34
2.3. Evoliucijos taisyklės daiktų interneto duomenims skirstyti . . . . .	37
2.4. Daiktų interneto savitvarka grįsta max-min skruzdžių P sistema . . . . .	44
2.5. Antrojo skyriaus išvados . . . . .	48
3. DAIKTŲ INTERNETO SAVITVARKOS TYRIMAS . . . . .	49
3.1. Pasiruošimas imitaciniams eksperimentams . . . . .	49
3.2. Daiktų interneto savitvarkos įgyvendinimo tyrimas . . . . .	51
3.3. Daiktų interneto savitvarkos ribinėmis sąlygos įgyvendinimo tyrimas . . . . .	60
3.4. Trečiojo skyriaus išvados . . . . .	64
BENDROSIOS IŠVADOS . . . . .	65
LITERATŪRA IR ŠALTINIAI . . . . .	67
AUTORIAUS MOKSLINIŲ PUBLIKACIJŲ DISERTACIJOS TEMA SAŖAŠAS . . . . .	73
SUMMARY IN ENGLISH . . . . .	75
DALYKINĖ RODYKLĖ . . . . .	91
PRIEDAI <sup>1</sup> . . . . .	93
A priedas. Disertacijos autoriaus sąžiningumo deklaracija . . . . .	95
B priedas. Bendraautorijų sutikimai teikti publikacijose skelbtą medžiagą disertacijoje . . . . .	96
C priedas. Autoriaus mokslinių publikacijų disertacijos tema kopijos . . . . .	104

---

<sup>1</sup>Priedai pateikiami pridėtoje kompaktinėje plokštelėje.

---

# Contents

INTRODUCTION .....	1
Problem Formulation .....	1
Relevance of the Thesis .....	2
The Object of the Research .....	3
The Aim of the Thesis .....	3
The Objectives of the Thesis .....	3
Research Methodology .....	4
Scientific Novelty of the Thesis .....	4
Practical Value of the Research Findings .....	5
The Defended Statements .....	5
Approval of the Research Findings .....	5
Structure of the Dissertation .....	6
Acknowledgements .....	6
1. INTERNET OF THINGS AND MEMBRANE COMPUTING	
TECHNOLOGIES REVIEW .....	7
1.1. Internet of Things .....	7
1.2. Self-Healing in Internet of Things .....	12
1.2.1. Information Control .....	14
1.2.2. Classification of Data Routing Algorithms .....	15
1.2.3. Bio-Inspired Global Optimization .....	17

1.3. Membrane Computing and P System .....	20
1.3.1. Membrane Computing – a Part of the Natural Computing .....	22
1.3.2. P System .....	23
1.3.3. Applications of Membrane Computing .....	25
1.4. Conclusions of the First Chapter and Formulation of the Thesis Objectives .....	28
2. SELF ORGANIZING INTERNET OF THINGS METHODOLOGY .....	29
2.1. Mathematical Models of Ant Colonies under Investigation .....	30
2.1.1. Ant Colony Optimization .....	31
2.1.2. Max-Min Ant System .....	33
2.2. A New Max-Min Ant P System .....	34
2.3. Evolutional Rules for IoT Data Routing .....	37
2.4. Internet of Things Self-Organization based on Max-Min Ant P System .....	44
2.5. Second Chapter Conclusions .....	48
3. AN INVESTIGATION OF INTERNET OF THINGS SELF-ORGANIZATION TECHNOLOGY .....	49
3.1. Preparation for the Simulation Experiments .....	49
3.2. Investigation of the Internet of Things Self-Organization Implementation .....	51
3.3. Investigation of the Internet of Things Self-Organization in Marginal Conditions Implementation .....	60
3.4. Third Chapter Conclusions .....	64
GENERAL CONCLUSIONS .....	65
REFERENCES .....	67
LIST OF SCIENTIFIC PUBLICATIONS BY THE AUTHOR ON THE TOPIC OF THE DISSERTATION .....	73
SUMMARY IN ENGLISH .....	75
ANNEXES <sup>1</sup> .....	93
Annex A. Declaration of Academic Integrity .....	95
Annex B. The Co-authors' Agreement to Present Publications Material in the Dissertation .....	96
Annex C. The Copies of Scientific Publications by the Author on the Topic of the Dissertation .....	104

---

<sup>1</sup>The annexes are supplied in the enclosed compact disc.

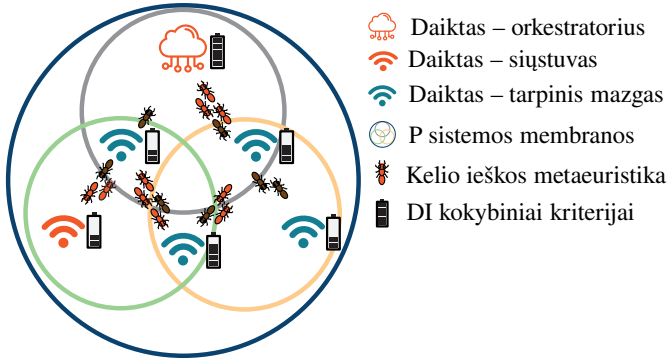
## Problemos formulavimas

Membraniniai skaičiavimai – tai kompiuterių mokslo sritis, kurioje skaičiavimo idėjos ir modeliai abstrahuojami iš gyvų ląstelių struktūros, funkcijų ir būdų, kuriais ląstelės jungiasi į audinius ar aukštesnes struktūras. Membraninių skaičiavimų teorijoje centrinę vietą užima P (Gheorghe Păun) sistema (Păun *et al.* 2010). P sistemos hierarchinė struktūra yra pamatinė ir puikiai atsiskleidžia ją pateikiant Eurelio (Euler 1761) ar Venno (Venn 1880) schemomis. Pati P sistemos formuluotė ir vykdomų skaičiavimų lygiagretumas suponuoja idealų membraninių skaičiavimų pritaikomumą didelio skaičiaus įrenginių (dažniausiai ribotų galimybių), siekiančių minimizuoti savuosius išteklius, tuo pačiu oportunistiniais ir kooperacijos principais (dažniausiai smarkiai kintančiuose aplinkos sąlygose) siekiančių užtikrinti bendruosius arba lokalius tikslus.

Daiktų interneto (DI) sistemų imperatyvi savybė yra autonomiškumas. Savitvarka – vienas iš svarbiausių autonominių skaičiavimų sudėtinių dalių. Todėl siekiant užtikrinti išmaniųjų daiktų sistemos darną, tikslinga tirti evoliucijoje patikrintų veiklos procesų pritaikymo DI galimybes.

Mokslinėje literatūroje nėra žinių apie tai, kuriuos DI principus tikslinga grįsti P sistema (Kumar *et al.* 2018). Kertinis iššūkis yra tinkamai parinkti ir susieti P sistemą su DI technologija – tai leistų pasiūlyti analogų neturinčią daiktų interneto savitvarkos dirbtinės gyvybės formą (0.1 pav.). Būtent todėl, šiame darbe

sprendžiama taikomoji mokslinė problema – racionalus daiktų interneto P sistema grįstos savitvarkos įgyvendinimas.



**0.1 pav.** Daiktų interneto savitvarkos idėjos, taikant P sistemą ir metaeuristinį algoritmą, iliustracija

**Fig. 0.1.** Idea sketch – IoT self-organization based on P system and metaheuristic algorithm

## Darbo aktualumas

Šiuo metu greičiausiai besivystančios technologijos yra susijusios su informacijos rinkimu, kaupimu ir sklaida. Šių technologijų plėtrai ir taikomumui didžiausią įtaką turi daiktų internetas. Taikant DI sistemas turi būti užtikrintos aukštos kokybės, didelio saugumo ir privatumo paslaugos, kurios bus atsparios triktims, prisitaikančios prie kintančios aplinkos (vietos ar klientų skaičiaus). Taigi šios sistemos privalo turėti savitvarkos savybes ir tai yra vienas iš pirmaeilų uždavinių.

Viena svarbiausių daiktų interneto savitvarkos savybių – patikimas duomenų valdymas ir skirstymas, kuris užtikrina atsparumą triktims. Sistemose kuriose įdiegti savitvarkos principai pasižymi geresniu apkrovos balansavimu, tai leidžia sutaupyti energiją bei sumažinti skaičiavimų apimtį (Wang *et al.* 2016).

Kuriant savitvarkes daiktų interneto valdymo priemones, geriausia remtis evoliuciniais ar metaeuristiniais algoritmais (Gubbi *et al.* 2013), nes gamta ir gyvybė išstobulino priemones bei procesus, kurie geba atsinaujinti ir prisitaikyti prie besikeičiančių sąlygų, o tai, remiantis Das, Dai (2007), yra vienas pamatinių kompiuterių mokslo tikslų. Taigi nenuostabu, kad šiuo metu pasaulyje aktualiomis mokslo šakomis tapo: dirbtinė gyvybė, skaitinės sintezės biologija ir gamtiniai skaičiavimai, apibendrintai pavadintos infobiotikos mokslu (Manca 2013).



Atlikta literatūros analizė patvirtina, kad P sistemos yra nepakankamai ištyrinėtos ir nėra taikytos DI savitvarkos uždaviniams spręsti. Pirminės P sistemų sąsajos su metaeuristiniais algoritmais, siekiant apjungti gerąsias abiejų paradigmu savybes, randamos mokslinėse publikacijose nuo 2010 m. (Ge-xiang Zhang, Marian Gheorghe 2010), o pirmoji monografijos dalis išleista tik prieš 3 metus. (Zhang *et al.* 2017). Optimizavimo skruzdžių kolonijomis (OSK) algoritmas yra vienas iš literatūroje plačiai nagrinėjamų spėčiaus sumanumu grįstų metaeuristinių algoritmu.

Siekiant išspręsti aktualią – racionalaus daiktų interneto P sistema grįstos savitvarkos įgyvendinimo problemą, darbe išsikelta ir įrodyta mokslinė hipotezė – taikant metaeuristinį algoritmą kartu su P sistema galima užtikrinti geresnę daiktų interneto savitvarkos kokybę.

## Tyrimų objektas

Darbo tyrimų objektas – daiktų interneto savitvarka. Tiriamos P sistemos ir metaeuristinio algoritmo taikymo galimybės daiktų interneto savitvarkoje siekiant padidinti jos efektyvumą.

## Darbo tikslas

Pasiūlyti originalų metaeuristika grįstą P sistemos sprendimą, sukuriant ir ištyriant membraninių skaičiavimų imitavimo ir testavimo sistemą daiktų interneto sąlygomis.

## Darbo uždaviniai

Tikslui pasiekti darbe reikia spręsti šiuos uždavinius:

1. Remiantis P sistemos teorija ir literatūros apžvalga, pasiūlyti originalius sprendimus atrinktam metaeuristiniam algoritmui aprašyti P sistemos konceptais siekiant greitesnio šio algoritmo veikimo.
2. Sukurtą metaeuristinio algoritmo P sistemą pritaikyti daiktų interneto savitvarkai, siekiant pagerinti kokybines charakteristikas greitaveikos, energijos sąnaudų ir kt. kriterijų pagrindu.
3. Imitavimo ir testavimo įrankiais ištyri sukurtą membraninių skaičiavimų sistemą vertinant kokybines charakteristikas.

## Tyrimų metodika

Disertacijos tikslas suformuotas atsižvelgiant į esamą ir plačiai naudojamą daiktų interneto technologiją ir aktyviai vystomas metaeuristinių algoritmų ir membranių skaičiavimų teorijas.

Darbe siūlomi technologiniai sprendimai kurti atsižvelgiant į šiuo metu visuomenei aktualius, praktiškai svarbius, nepakankamai moksliskai ištirtus uždavinius.

Disertacijoje naudojami neginčijami nusistovėję metodai, o keliamos hipotezės tikrinamos empiriniais tyrimais. Siekiama gauti kiekybiškai geresnius nei egzistuojantys rezultatus pagal vieną arba kelis vertinimo kriterijus (tinklo mazgo energijos suvartojimą, duomenų maršruto paieškos trukmę, duomenų paketo perdavimo delsą, tinklo gyvavimo trukmę ir likutinę energiją).

Darbe pasiūlyta nauja daiktų interneto savitvarkos metodika, taikant:

- komunikacijų sistemą: įrenginys-įrenginiui ryšiai (angl. *D2D*);
- membraninius skaičiavimus – P (Păun) sistema;
- evoliucinius skaičiavimus: max-min skruzdžių sistema.

Disertacijoje pateiktas naujas max-min skruzdžių P sistemos MMSPS algoritmas įgyvendintas taikant programos MATLAB™ R2020a versiją.

Moksliniai rezultatai vertinami informacijos paieškos, lyginamosios analizės, sisteminimo ir apibendrinimo būdais. Eksperimentinių tyrimų statistinė rezultatų analizė ir apibendrinimas atlikti remiantis eksperimentinio tyrimo metodu.

## Darbo mokslinis naujumas

1. Sukurtas pirmasis P sistemos matematinis aprašas, kurį tikslinga taikyti daiktų interneto savitvarkai ir jis įgyvendintas programinėje įrangoje.
2. Sukurta nauja daiktų interneto informacijos skirstymo ir valdymo, grįsta max-min skruzdžių P sistema, metodika. Šios metodikos taikymas leidžia automatizuoti daiktų interneto savitvarkos principus.
3. Įgyvendinant daiktų interneto savitvarkos principus gauti kiekybiškai geresni rezultatai pagal kelis vertinimo kriterijus (tinklo mazgo energijos suvartojimą, duomenų maršruto paieškos trukmę, duomenų paketo perdavimo delsą, tinklo gyvavimo trukmę ir likutinę energiją) turi teigiamą socialinį ir ekonominį efektą lyginant su atrinkuoju metaeuristiniu algoritmu.

## Darbo rezultatų praktinė reikšmė

Sukurtą daiktų interneto informacijos skirstymo ir valdymo max-min P skruzdžių sistemą galima taikyti optimalaus duomenų paketų kelio radimo uždaviniui spręsti. Šio uždavinio sprendimas tampa patogesnis, nes sistema automatiškai, atsižvelgiant į kokybines charakteristikas: greitaveikos, delsos ir energijos sąnaudų, parenka optimalų informacijos perdavimo kelią.

Turimų rezultatai buvo taikomi vykdant šiuos projektus:

- Intelektualiaisiais metodais grįstų ir lauku programuojamų loginių matricų integriniuose grandynuose įgyvendinamų elektroninių sistemų tyrimas. Priemonė Nr. 01.2.1-MITA-K-824 „Inovaciniai čekiai“.
- Šiuolaikinio ypač mažų matmenų, didelės duomenų perdavimo spartos šarvuotų durų užrakto stebėjimo sistemos maketo kūrimas. Priemonė Nr. 01.2.1-MITA-K-824-02 „Inovaciniai čekiai“.

## Ginamieji teiginiai

1. Sukurtą naują max-min skruzdžių P sistemą tinka taikyti daiktų interneto savitvarkoje.
2. Taikant max-min skruzdžių P sistemos algoritmą daiktų interneto duomenų maršruto parinkimui, sprendinys randamas 25 % greičiau nei atrinktuoju max-min skruzdžių sistemos (MMSS) algoritmu.
3. Atliktų Max-min skruzdžių P sistemos ribinėmis sąlygomis, esant sudėtingiems sprendiniams, imitacinių tyrimų rezultatai pagrindžia, kad siūlomos DIS metodikos taikymas lyginant su MMSS 15 % sumažina DI įrenginių tinklo energijos sąnaudas ir taip pailgina mobilaus tinklo gyvavimo trukmę.

## Darbo rezultatų aprobavimas

Darbo rezultatai pateikti 5 straipsniuose, iš kurių 3 straipsniai yra paskelbti recenzuojamose mokslo žurnaluose. Rezultatai viešinti 6 mokslinėse konferencijose:

- Gedminas, A.; Navakauskas, D. 2019. Preliminary results of Max-Min Ant P system for IoT Data Routing. AIEEE 2019. Ryga, Latvija.
- Kulakovskis, D.; Gedminas, A.; Navakauskas, D. 2018. Intravenous Glucose Tolerance Test Implementation in FPGA using Metabolic P System. KTU PhD Week 2018. Kaunas, Lietuva.
- Abromavičius, A.; Gedminas, A. 2017. Investigating Quality of Experience

using Spectral EEG Features in Immersive Audiovisual Communications. eStream 2017. Vilnius, Lietuva.

- Kulakovskis, D.; Gedminas, A. 2017. Metabolines P sistemos LPLM igyvendinimu skirtingais metodais tyrimas. Jaunujų mokslininkų konferencija „Mokslas – Lietuvos ateitis: Elektronika ir elektrotechnika“. Vilnius, Lietuva.
- Kulakovskis, D.; Sledevič, T.; Gedminas, A.; Navakauskas, D. 2016. Alternative Implementations of Metabolic P System in FPGA. AIEEE 2016. Vilnius, Lietuva.
- Gedminas, A. 2016. Mokslinių tyrimų membraninių skaičiavimų technologijų daiktų Internetui tematikoje analizė. Jaunujų mokslininkų konferencija „Mokslas – Lietuvos ateitis: Elektronika ir elektrotechnika“. Vilnius, Lietuva.

## Disertacijos struktūra

Disertaciją sudaro: įvadas, trys skyriai, bendrosios išvados, literatūros sąrašas su atskirai pateiktomis autoriaus publikacijomis ir trys priedai.

Darbo apimtis yra 94 puslapiai, kuriuose yra pateikta: 51 formulė, 34 paveiksai, 8 lentelės ir 5 algoritmai. Disertacijoje remtasi 74 kitų autorių literatūros šaltiniais.

## Padėka

Didelis ačiū darbo vadovui prof. dr. Daliui Navakauskui už kantrybę, motyvaciją ir metodinę pagalbą rašant straipsnius bei rengiant disertaciją.

Ačiū Vilniaus Gedimino technikos universiteto Elektronikos fakulteto ir ypač Elektroninių sistemų katedros darbuotojams už motyvaciją, vertingus patarimus bei pastabas. Ačiū doc. dr. Andriui Ušinskui už sudominimą moksline veikla. Dėkoju dr. Tomyslav Sledevič, dr. Vytautui Abromavičiui, dr. Dariui Kulakovskui, dr. Eldar Šabanovič, dr. Rokui Žalnėravičiui ir Liudui Duobai už bendrą mokslinę veiklą.

Už palaikymą ir paskatinimą studijuojant doktorantūroje širdingai dėkoju būsimai žmonai Rūtai Bernotaitei, šeimai ir draugams.

---

# Daiktų interneto ir membraninių skaičiavimų technologijų apžvalga

Siekiant pasiūlyti originalų metaeuristika grįstą P sistemos sprendimą šiame skyriuje apžvelgiamos daiktų interneto ir membraninių skaičiavimų technologijos, jų vystymosi kryptys ir tendencijos. Pirmajame poskyryje didžiausias dėmesys skiriamas daiktų interneto technologijoms ir autonomiškumo užtikrinimo galimybės. Antrajame poskyryje apžvelgiamos membraninių skaičiavimų technologijos ir jų taikymo daiktų internete galimybės. Trečiame poskyryje apžvelgiama pamatinė membraninių skaičiavimų sistema vadinama – P sistema. Skyriaus pabaigoje apibendrinamos žinios ir suformuluojamos išvados.

Dalis skyriuje pateiktos medžiagos skelbta autoriaus mokslinėse publikacijose (Gedminas 2019; Žalnėravičius *et al.* 2019) ir pristatyta tarptautinėje AIEEE (Ryga, 2016) ir respublikinėse „Mokslas – Lietuvos ateitis: Elektronika ir elektrotechnika“ (Vilnius, 2016, 2017) konferencijose.

## 1.1. Daiktų internetas

Daiktų internetas (DI) tai nauja paradigma, kurios koncepcija grindžiama visur esančiais objektais – jutikliais, radijo dažnių atpažinimo (RDA) žymomis, mobiliaisiais prietaisais ir t. t., bendraujančiais vieni su kitais, bendrų tikslų siekimui naudojant unikalius adresus (Atzori *et al.* 2010). DI yra labai jautrus programišių atakoms, daugumoje komunikacijos yra belaidės, dauguma komponentų turi ribotus išteklius (energijos ir duomenų pralaidos), todėl įmanoma juos atakuoti (Mus-

tafa *et al.* 2018). Atsižvelgiant į tai, kad DI turės informacijos apie beveik viską, duomenų saugumas, privatumas ir jų perdavimo keliai yra pagrindiniai DI tyrimuose susirūpinimą keliantys klausimai (Xu *et al.* 2014). Daiktų interneto technologija reikšmingai prisideda prie visuomenės, ekonomikos skaitmeninimo ir ketvirtosios pramonės revoliucijos (ITU 2012), nes individai ir daiktai komunikacijos tinklais sujungiami ir praneša apie juos supančią aplinką ar būseną. Taigi DI taikymo sritis labai plati, o jo sukuriamas duomenų kiekis per metus skaičiuojamas zetabaitais (Liton 2018).

Tyrimų apžvalgoje (Xu *et al.* 2014) teigiama, kad DI saugumo tyrimai yra būtini šios technologijos tolimesniai masiniam taikymui. Gubbi *et al.* (2013) išryškino savisaugos būtinumą buitiniuose įrenginiuose, teigdami, kad visi būtiniai įrenginiai bus prijungti prie bendros sistemos, ir ją reikės apsaugos nuo įsibrovėlių. Pagal Roman *et al.* (2011) bus labai svarbus atsparumas gedimui. Pažeidžiamų sistemų ir atakų skaičius didės, todėl yra poreikis plėtoti įsilaužimo aptikimo ir prevencijos sistemas, siekiant apsaugoti DI komponentus.

Internetas ir kompiuteriniai įrenginiai tapo įprasti mūsų gyvenime – nuo išmaniųjų telefonų ir kompiuterių, kuriuos naudojame patys, iki informacinių sistemų ir jutiklių, kurių mes nepastebime, bet kurie supa mus kasdieniniame gyvenime. Transportas, sveikatos priežiūra, energijos ir vandens tiekimas, prekyba, srautų valdymas, medijos, sauga ir pan. – praktiškai visose gyvenimo srityse ir pramonės sektoriuose valdymas ir stebėseną vyksta naudojant internetą ir įterptines sistemas, sąveikaujančias per interneto tinklą.

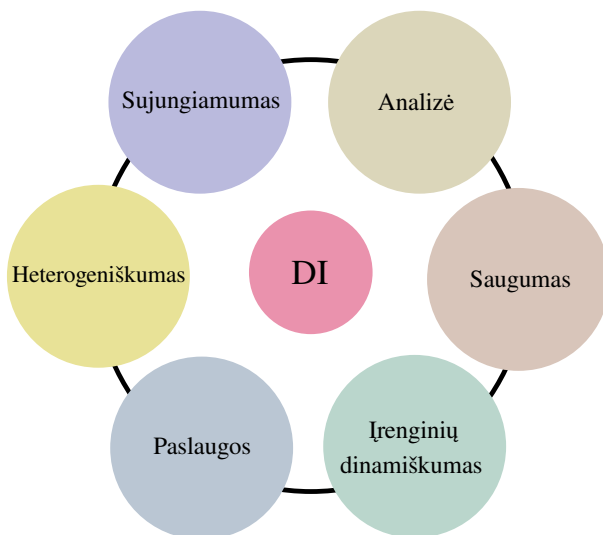
Duomenys internetu perduodami ne tik naudojant sąveiką „žmogus-kompiuteris“, bet ir per jutiklius sąveikaujant tarpusavyje „įrenginys–įrenginys“, dar vadinamiems interneto daiktams. Taip susiformavo nauja sąvoka – Daiktų internetas, kaip nauja informacinių tinklų paradigma, apimanti fizinių objektų komunikaciją ir sąveiką internete.

Daiktų internete daiktu laikomas fizinis įrenginys arba virtualusis įrenginys, kuris gali būti identifikuotas pagal *IP* ar *MAC* adresą ir integruotas į ryšių tinklus. Informacija apie fizinius įrenginius gali būti gauta per jutiklius, jie gali būti paveikti – stimuliuojami. Fizinių įrenginių pavydžiais gali būti mus supanti natūrali aplinka, prekės ir įvairūs jutikliai.

Virtualieji įrenginiai egzistuoja informaciniame pasaulyje ir gali būti kuriami ir tvarkomi virtualiai. Virtualiųjų įrenginių pavyzdžiu gali būti multimedijos turinys, įvairios sistemos ar programinė įranga. Fizinių įrenginių informacijos pasaulyje gali valdyti vienas ar daugiau virtualiųjų daiktų (sistemų ar programų), o virtualusis įrenginys gali egzistuoti ir be su juo susijusio fizinio įrenginio.

Daiktų interneto sistemų kūrimas dažniausiai prasideda nuo išmaniųjų biologinių jutiklių kūrimo (Žalnėravičius *et al.* 2019). Mažėjantys šių jutiklių matmenys ir noras pritaikyti juos visuomenės poreikiams patenkinti paskatino aktyviai tirti daiktų interneto sistemas.

DI gali būti suprantamas kaip globalioji infrastruktūra, teikianti pažangiausias paslaugas, sujungiant (fizinis ir virtualiuosius) įrenginius esamų ir besivystančių informacinių ir ryšių technologijų pagrindu (Serbanati *et al.* 2011). Identifikuojant, renkant duomenis, juos apdorojant ir naudojantis ryšių galimybėmis, DI leidžia visiškai panaudoti „daiktus“ įvairiausioms paslaugoms, tuo pačiu metu užtikrinant, kad būtų laikomasi saugumo ir privatumo reikalavimų.



**1.1 pav.** Daiktų interneto požymiai

**Fig. 1.1.** Internet of Things features

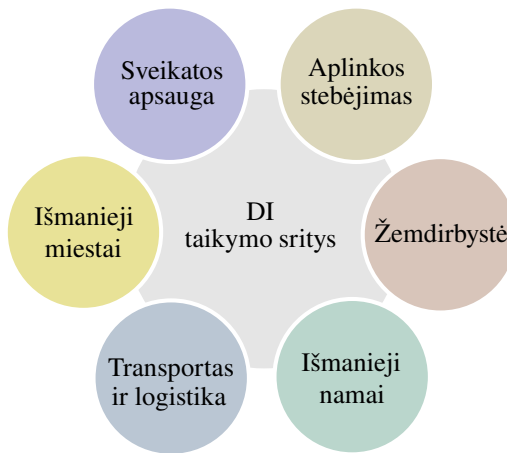
Pagrindiniai DI požymiai (1.1 pav.) leidžiantys atskirti jį nuo įprasto interneto ar žmogaus tiesiogiai valdomų IT prietaisų (Savukynas, Marcinkevičius 2017):

- Sujungiamumas – ryšio tarp visų daiktų tinkle užmezgimas per pasaulio komunikacijos ir ryšių infrastruktūrą, siekiant DI įrenginiams užtikrinti patikimą, saugų ir abipusį ryšį.
- Su daiktais susijusios paslaugos: DI gali teikti su daiktais susijusias paslaugas, atsižvelgiant į daiktų apribojimus, pvz., privatumo apsaugą.
- Analizė – sujungus daiktus į tinklą, reikia realiuoju laiku analizuoti surinktus duomenis ir priimti reikiamus sprendimus.
- Heterogeniškumas – DI įtaisai yra nevienalyčiai, nes jie remiasi įvairiomis platformomis ir tinklais. Jie gali sąveikauti su kitais įtaisais arba paslaugų platformomis per skirtingus interneto tinklus.
- Įrenginių dinamiškumas – įtaisų būseną keičiasi dinamiškai, pavyzdžiui,

jie užmiega ir atsibunda, įjungiami ir (arba) atjungiami, taip pat keičiasi jų vieta ir greitis. Be to, dinamiškai keistis gali ir įtaisų skaičius.

- Apimtis – administruojamų ir tarpusavyje bendraujančių įtaisų skaičius bus kelis kartus didesnis nei prie dabartinio interneto prijungtų įtaisų skaičius. Dar svarbesnė galimybė tvarkyti generuojamus duomenis ir juos interpretuoti, siekiant valdyti įrenginius.
- Bendrosios saugumo galimybės – autorizavimas, duomenų konfidencialumo ir vientisumo apsauga, privatumo apsauga ir saugumo auditas.
- Specifinės saugumo galimybės – susijusios su konkrečios taikomosios programos reikalavimais, pvz., mobiliųjų mokėjimų saugumo reikalavimais.

DI atsirado kaip rezultatas tolesnio interneto vystymo. Kaip ir dauguma kuriamų technologijų, DI skirtas patogesniai žmonių gyvenimui. Galima paminėti tik kai kurias galimybes, atsirandančias su DI, ir jų teikiamus privalumus (1.2 pav.).



1.2 pav. Daiktų interneto taikymo skritis

Fig. 1.2. Applications of the Internet of Things

**Aplinkos stebėjimas.** DI taikymas aplinkos stebėsenoje yra platus – aplinkos apsauga, ekstremalių oro sąlygų stebėjimas, vandens kokybės stebėseną, nykstančių rūšių apsauga, komercinis ūkininkavimas ir kita. Šiose srityse jautikliai nustato ir matuoja bet kokius aplinkos pokyčius ir mūsų poveikį aplinkai.

**Žemdirbystė.** Ūkininkavimas yra dažnai žmonių nepastebima daiktų interneto taikymo sritis. Tačiau ūkininkai naudodamiesi intelektualiausias ūkininkavimo sprendimais gali optimizuoti kiekvieną žemdirbystės procesą nuo gyvulių stebėsenos iki siekio sumažinti atliekų kiekį ir padidinti žemės ūkio našumą, pavyzdžiui



jutiklių pagalba nustatydami geriausią derliaus nuėmimo laiką ar realaus laiko dirvožemio tyrimais nustatydami dirvožemio maistinių medžiagų bei drėgmės koncentracijas bei sudaryti tinkams trąšos programas.

Transportas. Automobiliuose ir keliuose sumontuoti įvairūs jutikliai (kameros ir pan.), gali padėti geriau organizuoti eismą gatvėse, pvz. sumažinti spūsčių skaičių ir išmetamo anglies dvideginio kiekį. Įvairiais jutikliais aprūpinti išmanūs automobiliai jau gali savarankiškai judėti gatvėmis, kontroliuoti automobilio greitį pagal eismo ir aplinkos sąlygas, o įvykus nelaimei patys pakviesti pagalbą.

Viešasis saugumas. Kiekvienos šalies vyriausybės vienas pagrindinių ir svarbiausių uždavinių yra užtikrinti visuomenės saugumą. Duomenų prieinamumas realiu laiku yra labai svarbus įvairiose situacijose. Nepriklausomai ar tai žemės drebėjimas, automobilio vagystė, užpuolimas arba skubi medicininė pagalba, DI įrenginiai, kurie pavojaus metu gali greitai išsiųsti ir gauti informaciją, gali padėti padaryti pasaulį saugesnį. Per visuotinį ryšių tinklą sujungti prietaisai gali vykdyti žmogaus ar fizinio objekto paiešką realiu laiku. DI paplitimo mastas sąlygoja ir tai, kad nusikalstama veikla neišvengiamai palieka vienokius ar kitokius pėdsakus elektroninėje erdvėje.

Sveikatos apsauga. DI sveikatos apsaugoje taikomas įvairiose srityse: nuo nuotolinio paciento stebėjimo pažangiais ir išmaniais jutikliais iki įvairios įrangos operacinėse, reanimacijoje ir intensyvios slaugos palatose. Tai pagreitina ir supaprastinta pacientų priežiūrą daugiau skiriant laiko duomenų analizei ir tinkamo gydymo paskyrimui. Taip pat DI įrenginiai gali nedelsdami pranešti gydytojui apie nelaimę pavyzdžiui, astmos priepuolį, infarktą ir kt. Pavyzdžiui, kiekviena uždelsta minutė reaguojant į kieno nors staigų širdies sustojimą mažina tikimybę išgyventi 5,5 % (Larsen *et al.* 1993).

Išmanieji miestai Ši koncepcija skirta miesto planavimui ir viešajam administravimui, kūrybiškai ir tvariai automatizuojant paslaugas miestiečiams ar jo sveičiams. Išmanieji miestai apima visus aukščiau išvardintus sektorius transporto ir eismo valdymą, energetiką, viešąjį saugumą, sveikatos apsaugą, novatorišką miesto žemės ūkį ir atliekų tvarkymą. Tai padeda taupyti gamtinius, techninius, laiko, pinigų ir žmonių išteklius.

Išmanioji tiekimo grandinė. Krovinių judėjimo, skirstymo, užsakymų bei transporto stebėseną leidžia automatizuoti daugelį logistikos procesų taip sumažinant žmogiškosios klaidos tikimybę, sutrumpinti pristatymo terminus bei kuo racionaliau panaudoti turimus išteklius.

Išmanieji namai ir butai. Tai tikriausiai lengviausiai suprantama ir dažniausiai sutinkama taikymo sritis. Išmaniųjų namų vizija yra autonomiškai valdyti įvairių namuose naudojamą įrangą ir užtikrinti namų saugumą. Namų savininkas išmaniojo telefono ar kompiuterio pagalba gali stebėti ir valdyti namuose esančius įrenginius nuo elektros lemputės iki vėdinimo (šildymo) sistemų.

## 1.2. Savitvarka daiktų internete

Pirmą kartą terminą autonominė kompiuterija pavartojo kompanija „Tarptautinio verslo mašina“ (IBM) 2001 m. Jis buvo skirtas apibūdinti savarankiškas valdymo sistemas (Kephart, Chess 2003). Terminas autonominis yra kilęs iš biologijos, kur, pvz., autonominė žmogaus kūno nervų sistema rūpinasi daugeliu organizmo funkcijų, pašalindama iš sąmonės poreikį koordinuoti visas organizmo funkcijas.

IBM manifeste (Kreger 2001) teigiant, kad sudėtingos sistemos privalo turėti autonomines savybes, išskirtos keturios savarankiškai valdančių sistemų savybės:

- Savarankiška konfigūracija – savybė yra aptinkama sistemose, sugėbančiose savarankiškai instaliuoti ir savarankiškai nustatyti vartotojų poreikius.
- Savarankiška optimizacija – savybė yra aptinkama sistemose, galinčiose padaryti kai kuriuos pakeitimus, aktyviai siekiant pagerinti visos sistemos veikimą.
- Savarankiškas atsikūrimas – Savarankiško atsikūrimo savybė yra aptinkama sistemose, aptinkančiose ir diagnozuojančiose problemas. Svarbu, kad atsikūrimo sistemos būtų atsparios gedimams.
- Savarankiškas saugumas – aptinkamas sistemose, apsisaugančiose nuo kenkėjiškų atakų. Autonominė sistema prisitaiko, pasiūlydama apsaugą, privatumą ir duomenų apsaugą.

Savaiminį organizavimąsi galima įvardyti kaip prietaisų komunikacijos užmezgimo procesą tinkle po to, kai įvyksta komunikacijos klaida. DI technologijoje savaiminis organizavimasis turi keletą privalumų. Jis apima kitų aplinkų gedimų laipsnišką prevenciją, jei dėl aplinkoje esančių trikdžių duomenų tiekimas sutrinka (Evans 2013). Esant trikdžiams, šie prietaisai teikia duomenis, kuriuos galima naudoti aplinkai stebėti, operacijoms kontroliuoti ir valdyti. Be to, savaiminis organizavimas yra gyvybiškai svarbus ir dėl to, kad jo dėka tinklas palaiko DI naudojimą net ir esant trikdžiams, o tai reiškia, kad net ir tokiomis sąlygomis bendrasis daiktų sąveikos tikslas yra pasiekimas.

Gedimų nustatymas ir jų šalinimas savarankiškai save valdančiose sistemose vyksta šiais etapais. Stebėjimo etape vyksta duomenų rinkimas iš jutiklių ir šių duomenų perdavimas analizės etapui. Analizės fazė yra atsakinga už apdorotų duomenų gavimą ir galimų problemų nustatymą sistemoje. Planavimo etape yra organizuojami reikiami veiksmai, nustatytų analizės etape problemų pataisymui. Planuojami veiksmai yra įgyvendinami vykdymo etape.

Egzistuoja penki daiktų interneto savaiminio organizavimosi būdai. Jie atlieka specifinius vaidmenis, bet kartu jie užtikrina sklandų daiktų interneto veikimą. Šie komponentai apima aplinkinių daiktų atpažinimą, vidinę prieigos kontrolę, energijos valdymą ir paslaugų atkūrimą, daiktų interneto objektų savaiminio organizavimosi komponentų integravimą. Vidutinė prieigos kontrolė užtikrina, kad joks

kitas mazgas neužmegs ryšio su kanalu tuo metu, kai jį naudoja tinklo mazgas. Šalia esančių prietaisų atpažinimo funkcija leidžia prietaisams atpažinti arčiausiai esančius prietaisus, kurie savaiminio organizavimosi proceso dėka gali veikti ir koordinuotis (Bendavid *et al.* 2013).

DI savaiminio organizavimosi procesas labai pažengė pirmyn. Buvo atlikta daug savaiminio organizavimosi studijų (Banzhaf 2009), buvo pasiūlyta naujų algoritmų tinklo aplinkoms ir specifiniam pritaikymui. Kai kurie iš jų apima jutiklių tinklų savaiminį organizavimąsi, pažangiuosius elektros energijos tinklus, intelektualiųjų skaitiklių infrastruktūrą ir mobiliuosius ad-hoc tinklus. Buvo atlikta (da Silva, Albini 2014) įvairių savaiminio organizavimosi tyrimų, kurie koncentravosi į mobiliuosius ad-hoc ir jutiklių tinklus. Savaiminio organizavimosi tikslas – užtikrinti kiekvieno mazgo jungiamumą prie duomenis renkančio įrenginio. Tinklo mazgai sugrupuoti klasteriais ir jungiasi per šiuos klasterius su duomenis renkančiu įrenginiu.

Mažai energijos reikalaujanti adaptacinė klasterių hierarchija (LEACH) – tai bevielų jutiklių tinklų grupavimo schema. Esant plačiam daugelio radijo platformų ir technologijų spektrui, DI ji padės pasirinkti pagrindinius klasterius. DI savaiminio organizavimosi procesų metu tai padeda spręsti energijos valdymo problemas. Tinklo paskirstymo prigimčiai patobulinti galima naudoti bevelius jutiklinius tinklus. Šiems beveliams jutikliniams tinklams kurti pasitelkiamas heterogeniškumas. Sugedus jutikliniam maršrutizatoriui, šie tinklai pasižymi savaiminio atkūrimo savybėmis. Ši savybė yra gyvybiškai svarbi, kai žemutinio tinklo sluoksnio klientai naudojami kitais maršrutizatoriais: taip užtikrinama, kad interneto ryšys neprarandamas. Jei nutiktų taip, kad netoliese nebūtų maršrutizatoriaus, išjungtų netoliese esančių įrenginių atradimo, kelio kūrimo ir lokalaus jungiamumo komponentai ir sukurtų maršrutus iki veikiančio maršrutizatoriaus (Atzori *et al.* 2012).

Išplėtota ir daugelio tinklų taikymo, persidengiančių sluoksnių architektūra. Ejaz *et al.* (2020) atskleista, kad tinklo patikimumą mažina tinklo apkrova, kai tarpusavyje sąveikauja daugelis tinklo sluoksnių. Decentralizuoto daugiamazgio tinklo (angl. *mesh network*) savaiminio organizavimosi struktūra buvo išvystyta intelektualiųjų skaitiklių tinklams, siekiant paspartinti maršrutizatorių protokolą tinklui, esant žemam energijos suvartojimui. RPL modifikacijos pasitarnavo puikiu įžvalgų šaltiniu apie tai, kaip vietos jungiamumą, paslaugų atkūrimo vadybą, netoliese esančių prietaisų suradimą ir jungties nustatymą galima įgyvendinti vienoje sistemoje. Be to, kitų komponentų izoliavimas pasirodė esantis tinkamas taikyti įvairioms tinklo aplikacijoms. Vis dėlto DI yra skirtingų tarpusavyje sąveikaujančių interneto programų sistema ir nėra skirtas vienai tinklo aplikacijai (Kulkarni, Sathe 2014). Taigi paslaugoms ir tinklo jungiamumui atsistatyti būtinas savaiminis organizavimasis, kurį palaikytų įvairūs įrenginiai. Todėl reikalinga vieninga sistema, integruojanti DI savaiminio organizavimosi sprendimus.

### 1.2.1. Informacijos valdymas

Daiktų internetas – programinės ir techninės įrangos technologija, skirta išmaniaisiais įrankiais jungtis prie jutiklių ir daugybės prietaisų, siekiant juos valdyti arba gauti pagrindinę jų informaciją (Roberto Minerva, Abyi Biru 2015). Šios technologijos turi bendrą tikslą – generuoti duomenis, naudojamus išmaniajai aplinkai kurti Bello, Zeadally (2016). Vis dėlto šie tarpusavyje sąveikaujantys įrenginiai yra svarbiausieji DI sistemos komponentai. Panašiai taip kaip žmonės yra interneto vartotojai, įrenginiai (daiktai) yra DI vartotojai. Įrenginiai inicijuoja tarpusavio ryšius (Bello, Zeadally 2013). Baterijos veikimo laikas, duomenų apdorojimo galia ir atminties talpa turi apribojimų ir dažnai veikia nepastovioje belaidėje aplinkoje, todėl susiduriama su daugybe tinklo problemų, kurių negalima išspręsti tradiciniais informacijos skirstymo protokolais (Nasser *et al.* 2017). Jutiklių tinklų pažanga paskatino atsiradimą įrenginių kurie jaučia, skaičiuoja ir bendrauja vieni su kitais. Kai kurios technologijos, skirtos informacijos skirstymui ir valdymui, egzistuoja kelis dešimtmečius, tačiau skirtos DI įrenginiams dar nėra tokios brandžios. Šios technologijos suteikia pamatus DI sistemų veikimui. Jungiamumui ir komunikacijai užtikrinti tam tikrų tipų tinklai geba sąveikauti vieni su kitais, kad bet kuriam DI sistemos įrenginiui būtų tiekiamas tolygus, visur esantis ir nenutrūkstamas ryšys.

Daugeliui DI įrenginių būtina jutiklių duomenis perduoti į centrinę stotį, kurioje jie būtų analizuojami ar saugojami. Duomenų perdavimas tarp įrenginių turi vykti naudojant veiksmingus protokolus. Veiksmingas komunikacijos protokolas privalo turėti veiksmingą duomenų perdavimą ir būti energetiškai efektyvus (Machado *et al.* 2013). Tačiau tipinei įrenginys-įrenginiui komunikacijai būdingos savybės kelia daug iššūkių DI ir tradiciniai informacijos skirstymo protokolai šių iššūkių nepajėgūs išspręsti.

DI egzistuoja įvairių tipų komunikacija: įrenginys įrenginiui (D2D), įrenginio su paskirstyta saugykla, įrenginio su žmogumi (D2H) ir atvirkščiai. Komunikacija gali vykti tame pačiame tinkle ir tarp skirtingų tinklų. Svarbu paminėti, kad komunikacija gali vykti tiesiogiai su bazine stotimi, arba per kitus įrenginius jeigu nepavyksta tiesiogiai susisiekti su bazine stotimi. Taip pat tarpiniame komunikacijos lygmenyje gali būti reikalingas žmogaus įsikišimas, siekiant priimti sprendimą. Be žmogaus įsikišimo įrenginiai gali komunikuoti, kai reikia perduoti informaciją žmogui arba ją gauti tiesiogiai iš žmogaus sprendimams priimti. Įrenginiai gali bendrauti su duomenų saugykla, kai reikia perduoti užfiksuotus duomenis, atnaujinti saugomus duomenis arba nuskaityti saugomus duomenis tam, kad dirbtinis intelektas galėtų priimti sprendimus.

Remiantis De Poorter *et al.* (2011), svarbiausias DI keliamas reikalavimas – nenutrūkstamas ryšys su bazine stotimi. Tai labai svarbu užtikrinti D2D komunikaciją. Įrenginiai, kurie vadinami išmaniais daiktais gali būti nedideli jutikliai,

pasyvios arba aktyvios RFID žymos, kurios užfiksuoja fizinius duomenis ir gali savarankiškai atlikti užduotis ar priimti sprendimus. Įrenginys gali būti ir mobilusis prietaisas, pvz., išmanusis telefonas. Tam tikrais atvejais vietoj įrenginių, kurie keičiasi informacija, gali būti naudojamos skirtingos ryšių technologijos (De Poorter *et al.* 2011), kurios bendravimo procesui kelia iššūkių. Dauguma šių technologijų naudoja patentuotus protokolus, iš kurių kiekviena skirtingai įgyvendinta fiziniu, duomenų perdavimo ir tinklo lygmeniu. Taigi norint, kad DI veiktų nenutrūkstamai, komunikacijos procesas turi palaikyti informacijos skirstymą ir valdymą tarp skirtingų įrenginių ir skirtinguose tinkluose.

### 1.2.2. Duomenų skirstymo algoritmų klasifikavimas

Daugelyje DI taikymo sričių būtina perduoti įrenginių ar jų grupės jutiklių duomenis į centrinę stotį, kur jie būtų analizuojami ar saugojami. Duomenų perdavimas tarp įrenginių turėtų vykti naudojant efektyvius protokolus. Efektyvus komunikacijos protokolas turėtų palaikyti patikimą duomenų perdavimą ir efektyvų energijos vartojimą. Tačiau tipinei D2D komunikacijai būdingos savybės kelia daug iššūkių DI ir tradiciniai skirstymo protokolai šių iššūkių nepajėgūs išspręsti. Išskiriamos šios D2D komunikacijos savybės:

- Įrenginių heterogeniškumas – įrenginiai yra įvairūs, todėl skirtingai veikia ir skirtingai pritaikomi.
- Įrenginių bendradarbiavimas – vienu metu šalia egzistuoja daugybė įvairių įrenginių, kurie gali jungtis vieni su kitais ir bet kuriuo metu komunikuoti bei bendradarbiauti.
- Skirtingi tinklai ir tinklų standartai – veikdami prietaisai naudos skirtingas komunikacijos technologijas (*Bluetooth, zigbee, LoRaWAN, Sigfox* ir pan.).
- Įrenginių ribotumai – baterijos veikimo laikas, atminties talpa, duomenų apdorojimo galia ir pan.
- Savarankiška konfigūracija, organizavimasis ir autonomiškumas – dauguma D2D tinklų patys konfigūruojasi, organizuojasi ir yra autonomiški. Todėl įrenginiai patys valdys trukdžius, skirtingų protokolų vertimo procesą ir komunikaciją.
- Nenuspėjamas prietaisų judėjimas – DI sudaro mobilūs arba stacionarūs įrenginiai. Judėdami mobilieji įrenginiai gali atsijungti nuo tinklo, taigi gali atsirasti ryšio pertrūkių, todėl D2D tinkluose tinklo topologijos pokyčių neįmanoma prognozuoti.

Dėl išvardytųjų charakteristikų išmaniąjai D2D komunikacijai DI būtina išrasti naujus metodus. Be to, DI gali veikti daug įrenginių, todėl įrenginių perduodamų duomenų gausa gali turėti įtakos D2D tinklų patikimui ir komunikacijos spartai.

Galima teigti, kad didinant tinklo pajėgumą, padidėjusio duomenų perdavimo intensyvumo problema bus išspręsta. Tačiau dėl tokio sprendimo gali padidėti veiklos sąnaudos, be to, jis nėra pritaikomas prie poreikių. Tikrąjį sprendimą galima rasti per tinklų kūrimo procesą, nustatant informacijos skirstymo ir valdymo maršrutus (Liotta 2013). Siekiant atskleisti visus DI funkcionalumus, D2D ryšiui reikia intelektualių, veiksmingų ir kintamo dydžio maršruto nustatymo protokolų ar algoritmų, kurie galėtų prisitaikyti prie skirtingų tinklo dydžių scenarijų ir variantų bei būtų pajėgūs rasti optimalius maršrutus. Šie protokolai gali užtikrinti sklandžią D2D komunikaciją sudėtingoje ir nevienalytėje interaktyviosios terpės aplinkoje. Pavyzdžiui, pagal pastatų automatizavimo scenarijų, kartu egzistuoja ir ribotų išteklių įmontuoti tie įrenginiai, ir didelės galios valdiklių įrenginiai, todėl išmanusis protokolas privalo užtikrinti ryšį tarp šių skirtingų įrenginių taip, kad prietaisų asimetrija būtų nepastebima ir netrikdytų D2D komunikacijos. Žemiau pateikiama DI maršruto parinkimo algoritmų ir protokolų klasifikaciją ir jų savybės.

Stochastiniai/tikimybių algoritmai. Šie algoritmai skirti optimizuoti maršruto parinkimą. Jie sukurti siekiant nustatyti informacijos skirstymo maršruto tikimybes ir optimizuotų tinklo išteklius: energijos sunaudojimą, ryšio parametrus (pvz., signalo ir triukšmo santykį, klaidų lygį) ir pan. Remiantis tinklo sąlygomis, šie tinklo ištekliai yra pasirinkti ir apibrėžti kaip optimalumo kriterijai. Bet koks pasirinktas maršrutas yra tas, kuris maksimaliai pagerina šiuos kriterijus. Du pagrindiniai metodai, kuriuos šie algoritmai naudoja optimizavimui, yra optimizavimas realiuoju laiku ir išankstinis optimizavimas (Lott, Teneketis 2006). Taikant optimizavimo realiuoju laiku metodą, informacija apie dominančius kriterijus atskleidžiama tuo metu, kai duomenys perduodami iš vieno įrenginio į kitą, taigi maršrutai kuriami dinamiškai, remiantis turima informacija. Šis metodas naudoja daug energijos ir skaičiavimo išteklių. Išankstinio optimizavimo atveju, sprendimas priimamas iš anksto ir jis konstruojamas remiantis tikėtinumo kriterijais, taigi kuriami maršrutai gali būti ne optimalūs. Taigi stochastiniai maršruto parinkimo algoritmai yra tinkamesni, kai būtina palaikyti nenuspėjamą mobiliųjų prietaisų judėjimą DI vykstančioje D2D komunikacijoje.

Hierarchiniai algoritmai. Hierarchiniai algoritmai gali būti grindžiami medžiu arba klasteriu.

- Medžiu grindžiami algoritmai – įrenginiai kurie veikia medžiu grindžiamais algoritmais duomenis į bazinę stotį perduoda per kitus įrenginius esančius tame pačiame tinkle. Daugelio šuolių medis konstruojamas duomenų maršrutui nustatyti. Ši funkcija riboja jų bendrą naudojimą DI, nes tarp įrenginių egzistuoja skirtingi komunikacijos būdai. Pavyzdžiui, D2D tinkle komunikacija gali vykti tarp pavienių įrenginių, pavieniai įrenginiai gali komunikuoti su daugeliu, daugelis įrenginių gali komunikuoti su vienu arba daugelis su daugeliu (Machado *et al.* 2013).

- Klasteriu grindžiami algoritmai – klasteriu grindžiami algoritmai klasifikuoja įrenginius į hierarchines grupes (klasterius). Atsižvelgiant į savo vietą hierarchijoje, įrenginiai atlieka skirtingus vaidmenis. Aukštesnę vietą hierarchijoje turintis tam tikros grupės įrenginys yra klasterio arba grupės orkestratorius (angl. *cluster head* – CH). Paprastai CH yra atsakingas už klasteryje surinktos informacijos perdavimą kitam klasteriui arba į bazinę stotį/saugojimo įrenginį. CH parinkimas yra svarbiausias šio tipo algoritmų uždavinys. Yra daugybė algoritmų, remiantis kuriais literatūroje siūloma išrinkti CH, bet šie algoritmai turi įtakos duomenų perdavimo spartai (Machado *et al.* 2013).

Kontekstą vertinantys algoritmai. Kontekstas – tai bet kokia informacija, kuri gali būti naudojama objekto (žmonių ar daiktų) būklei apibūdinti (Abowd *et al.* 1999). Prietaisų atveju, kontekstas paaiškina vidinę ar išorinę jų būseną ir gali būti išgaunamas iš įrenginio, jo aplinkos ar gautų pranešimų turinio. Įrenginio kontekstas gali būti jo akumuliatoriaus būklė, judėjimo greitis, duomenų apdorojimo sparta, atminties talpos (saugyklos) dydis ir būseną. Iš gautų pranešimų gautas kontekstas apima pranešimo šaltinį, pranešimo paskirties vietą, prioriteto lygį ir pristatymo terminą (Mirco Musolesi 2006). Aplinkos kontekstas apima atstumą tarp įrenginių, trukdžių lygius arba paketų praradimo greitį bei tinklo topologiją.

Kontekstas gali būti naudojamas teikiant maršruto paslaugą įrenginiui, kai įrenginio užduotis – perduoti duomenis. Taigi kontekstas išmaniajai D2D komunikacijai yra naudingas priimant sprendimus apie maršrutą (Abowd *et al.* 1999). D2D komunikacijoje, kontekstą vertinantys maršruto parinkimo algoritmai gauna reikiamą informaciją apie situaciją aplink tinkle esančius įrenginius ir, remdamiesi gautu kontekstu, parenka maršrutus, kuriais perduodama informacija. Kontekstą vertinantys algoritmai gali reaguoti į besikeičiantį įrenginio kontekstą ir keisti jo maršruto parinkimo parametrus realiuoju laiku, kaip to reikalauja tinklas.

### 1.2.3. Bio-grįstas globalus optimizavimas

Šios klasės algoritmai sprendžia dažnas ir reikšmingas problemas didelio masto tinklams, kuriems būdinga sudėtinga ir heterogeniška architektūra, dinaminė saviorganizacijos prigimtis, riboti išteklių ir centralizuotų valdymo bei infrastruktūros nebuvimas (Dressler, Akan 2010). Biotechnologiniai algoritmai imituoja biologinių sistemų valdymo dėsnius ir jų dinamiką. Biotechnologiniai algoritmų pavyzdžiai – spiečiaus logika grindžiami algoritmai, pvz., skruzdžių kolonijos optimizavimo (angl. *the ant colony optimization, OSA*) algoritmas. Spiečiaus logika grindžiama decentralizuotų ir savarankiškų organizacijų, pvz., skruzdžių kolonijų, žuvų pulkų, bičių spiečių ar paukščių būrių, kolektyvinio elgesio stebėjimu (Di Caro *et al.* 2005).

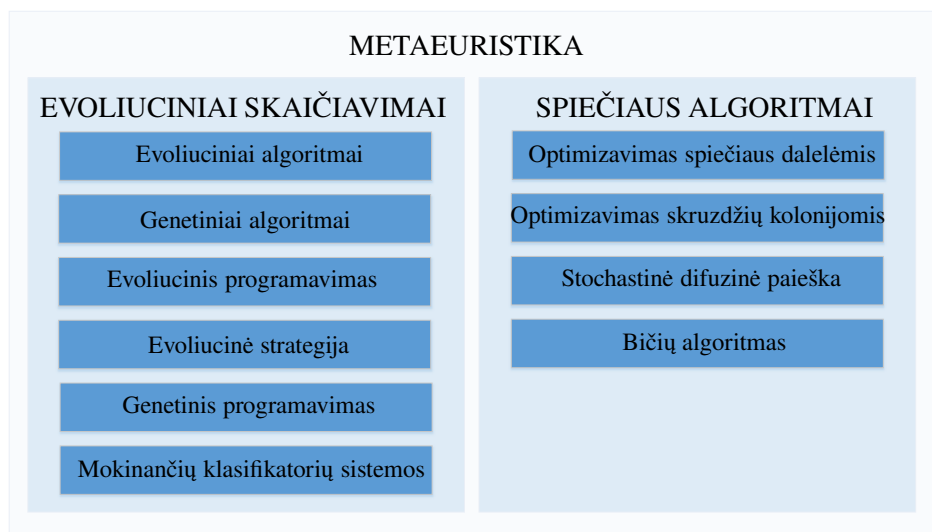
Pagrindiniai OSK maršrutizavimo algoritmų veikimo principai:

1. Maisto paieškos – tai skruzdžių naudojamas procesas, kai norima atrasti trumpiausią kelią nuo lizdo iki maisto šaltinio. Jis vyksta kolonijų lygmeniu – skruzdės kartu dalyvauja kelio mokymosi ir atradimo procesuose, siekdamos atrasti trumpiausią kelią, jungiantį jas su maisto šaltiniu.
2. Stigmergija – procesas, kai skruzdės lokaliai keičia atrastus takus ir reaguoja į šias modifikacijas – taip kuriamas tam tikras visuotinis koordinavimas. Tai paskirstytojo mokymosi ir valdymo forma, grindžiama netiesiogine skruzdžių sąveika ir bendradarbiavimu. Stigmerginis koordinavimas yra pagrindinis būdas užtikrinti sistemai būdingą elgseną. Remdamiesi stigmerginiu procesu, OSK metodai užtikrina veiksmingą maršrutą dideliuose tinkluose (Pablo López-Matencio, Javier Vales-Alonso 2017).

Taigi šių gamtoje vykšančiais procesais grįstų algoritmų privalumai DI informacijos skirstyme ir valdyme yra šie:

1. Jie pajėgūs susidoroti su DI veikiančių įrenginių galimybių ir technologijų heterogeniškumu ir asimetrija: skruzdžių kolonijas sudaro pavienės skruzdės, besiskiriančios savo pajėgumais ir gebėjimais, tačiau, nepaisant šio būdingo heterogeniškumo, skruzdžių kolonijos pasitelkia kolektyvinę intelektą, kad optimizuotų kelio pasirinkimą (Klugl 2001).
2. Jie gali palaikyti savarankišką organizavimąsi, savarankišką konfigūravimąsi ir bendradarbiavimą, o tai palengvina autonomiškos, nuo infrastruktūros nepriklausomos D2D komunikacijos įdiegimą DI aplinkoje (Atakan, Akan 2007): stigmerginis procesas leidžia skruzdžių kolonijoms bendradarbiauti ir organizuotis savarankiškai, be centrinio koordinavimo.
3. Gamta grįsti algoritmai gali dinamiškai prisitaikyti, kad užtikrintų įrenginių komunikaciją: daiktų interneto D2D tinklas yra dinamiškas, įrenginių mobilumas jame yra nenusėjamas, taigi informacijos skirstymo algoritmai gali veikti pagal maisto paieškų modelį. Maisto paieška leidžia skruzdžių kolonijai veiksmingai prisitaikyti prie kintančių aplinkybių, kad išgyventų.
4. Prisitaikomumas prie kintančių aplinkos sąlygų: naujomis sąlygomis biologinės sistemos gali lengvai mokytis ir vystytis. Taigi ir gamtos grįsti algoritmai gali padėti įrenginiams spręsti neprognozuojamumo problemą, kuri gali kilti daiktų internete dėl aplinkinių prietaisų, dalyvaujančių D2D komunikacijoje.
5. Tvirtumas ir atsparumas: jie tvirti ir atsparūs gedimams, įvykstantiems dėl vidinių ar išorinių veiksnių; gali veikti sudėtingomis aplinkos sąlygomis.
6. Efektyvus ribotų išteklių valdymas: maisto paieškų procesas skruzdžių kolonijose įkvepia daugybę efektyviai naudojamų tinklo kūrimo procesų.

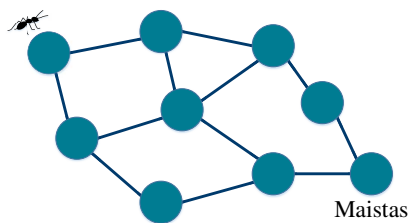




**1.3 pav.** Metaeuristikos algoritmų klasifikavimas (Laptik 2009)

**Fig. 1.3.** Classification of Metaeuristic Algorithms (Laptik 2009)

Metaeuristikoje (1.3 pav.) spiečiaus algoritmai yra sritis, kurioje daugiausia dėmesio skiriama kolektyviniam intelektui, kurį sudaro didelės grupės mažiau protingų agentų. Agentų, tokių kaip skruzdės, paukščiai, bakterijos ir bitės, kolektyvinis intelektas pasiekiamas per jų bendradarbiavimą. Šie agentai sąveikauja natūraliai, norėdami patenkinti pagrindinį maisto poreikį ir poreikį persikelti: skruzdės – feromonais, bitės – šokdamos, o bakterijos – cheminiais gradientais.



**1.4 pav.** Skruzdžių kolonijos algoritmas optimaliam keliui rasti

**Fig. 1.4.** Ant Colony Algorithm for Solving the optimal Path Planning problem

Spiečiaus intelektas buvo tiriamas pasitelkus skruzdžių kolonijos optimizavimą, dalelių virpėjimo optimizavimą, bičių algoritmus ir bakterijų paruošimo opti-

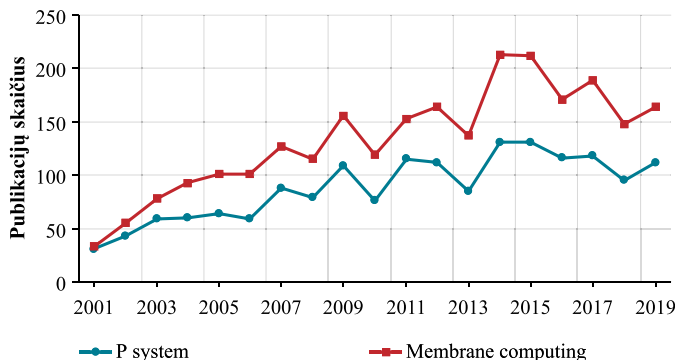
mizavimą. Skruzdžių kolonijos optimizavimas (Gao *et al.* 2016) – tai tikimybinių algoritmų, pagrįstų maisto ieškančių skruzdžių elgesiu, tyrimas ir šių algoritmų įgyvendinimu paieškos ir optimizavimo srityje. Skruzdės paskleidžia feromonus kelyje, kuriuo jos keliauja iš kolonijos link maisto šaltinio 1.4 paveiksle feromonus jos skleidžia tiek keliaudamos pirmyn, tiek atgal. Aplinkoje feromonai suyra. Kiekviename kelių susikirtime yra tikimybė pasirinkti vieną iš kelių. Tuo atveju, kai visi tikėtini maršrutai yra nauji ir nėra viename tų maršrutų feromonų nėra paskleista, vieno ar kito maršruto pasirinkimo tikimybė yra lygi. Kadangi skruzdės juda pirmyn ir atgal, priklausomai nuo kelionės dažnumo, feromonų sąnašų kiekis trumpesniam kelyje išauga.

Apibendrinant galima teigti, kad nors visi spiečiaus klasės algoritmai yra pritaikomi, vis dėlto išmaniajai D2D komunikacijai DI labiausiai tinka skruzdžių kolonijų algoritmai. Šie algoritmai išmaniai komunicuoja (D2D) ir geba pasinaudoti tinklo ištekliais, užtikrindami sklandžią sąveiką.

### 1.3. Membraninių skaičiavimų technologijos ir P sistema

Jau ne kartą buvo įrodyta, kad plačiau analizuojant ir tyrinėjant biologinius procesus, juos galima tikslingai pritaikyti kompiuterių moksle (Hunter 1993). Biologija tirdama gyvuosius organizmus, jų sandarą, rūšis, jų tarpusavio santykius ir sąveiką su aplinka padeda suprasti supančią gyvąją aplinką ir tai, kaip galima ją pasitelkti šiuolaikiniams iššūkiams įveikti. Per daugybę milijardų metų gamta ir gyvybė iš-tobulino priemones ir procesus, kurie geba atsinaujinti ir prisitaikanti prie besikeičiančių sąlygų. O tai yra ir vienas iš pamatinių kompiuterių mokslų tikslų (Das, Dai 2007), todėl kuriant DI informacijos skirstymo ir valdymo algoritmus tikslinga atsižvelgti į biologinius procesus. Membraniniai skaičiavimai (MS) tai kompiuterių mokslų sritis, kurioje skaičiavimo idėjos ir modeliai abstrahuojami iš gyvų ląstelių struktūros, funkcijų ir būdų, kuriais ląstelės jungiasi į audinius ar aukštesnes struktūras. Membranų sistema kurią toliau vadiname P sistema (Guo *et al.* 2017) nagrinėja paskirstytuosius ir lygiagrečiuosius skaičiavimo modelius, naudojamus lokalizuotai apdoroti simbolių aibės (evoliucijos taisyklės ir besivystantys objektai sujungti į sritis, kurias vienas nuo kitų skiria membranos), o svarbiausias vaidmuo tenka tarp sričių vykstančiai komunikacijai. MS pirmą kartą pradėti tyrinėti 1998 m., o galutinė šio straipsnio versija buvo paskelbta kaip tyrimo ataskaita 2000 m. (Păun 2000). Šiuo metu yra paskelbta virš 2500 straipsnių (1.5 pav.), daugiau negu 100 disertacijų ir 50 knygų. Juose yra daug iškeltų ir neišspręstų optimizavimo problemų (Zhang *et al.* 2014) tokių kaip natūrali atranka, molekulinė genetika bei decentralizuotų ir savarankiškai organizuotų sistemų kolektyvinis

elgesys, kurias tiria didelė grupė mokslininkų iš viso pasaulio.



**1.5 pav.** Raktinių žodžių „Membraniniai skaičiavimai“ ir „P sistema“ analizė Web of Science duomenų bazėje

**Fig. 1.5.** Analysis of keywords “Membrane computing” and “P system” in Web of Science database

Gamtiniai skaičiavimai – pastangos susisteminti kompiuterių mokslui naudingų idėjų, modelių ir paradigmu iš natūralios gamtos, o ypač iš gyvybės (Beale, Jackson 1990). Gamtoje vystantys natūralūs procesai gali būti atkartoti kuriant naujus matematinius modelius. Klasikinė, biologijos grįsta, gamtinių skaičiavimų šaka yra genetikos algoritmai ir dirbtinė gyvybė. Abi patvirtina, kad verta mokytis iš biologijos ir grindžia optimistinę nuostatą, kad per daugybę milijardų metų gamta ir gyvybė išstobulino tam tikras priemones ir procesus, kuriuos tinkamai pritaikius kompiuterių mokslo reikmėms, galėtų būti stebėtinai naudingi ir įvairiai pritaikomi. Gyvybė tiesiogiai susijusi su ląstelėmis. Ląstelė yra mažiausias objektas, kurį visuotinai laikome gyvu. Ji labai maža, sudėtingos struktūros, atlieka sudėtingas funkcijas, jos viduje vyksta sudėtingi procesai, o išoriškai ji sąveikauja su šalia esančiomis ląstelėmis ir aplinka. Per milijardus evoliucijos metų ji išvystė būdą organizuoti ir valdyti biocheminius, ir informacinius procesus. Membranos leidžia medžiagoms judėti iš vieno skyriaus į kitą selektyviai (Ciobanu *et al.* 2005). Ši selekcija gali būti atliekama vien pagal dydį, kai juda mažos molekulės, arba ji gali vykti kur kas sudėtingiau – per baltymų kanalus, kurie molekules ne vien atrenka, bet ir leidžia joms judėti iš žemos koncentracijos į didesnę koncentraciją, galbūt joms jungiantis, kai vyksta simportiniai ar antiportiniai procesai. Be to, ląstelių membranos neatskiria vien tų jungčių, kurių viduje vyksta tam tikros cheminės reakcijos: daugelis ląstelėje vykstančių reakcijų gali vykti pačių membranų viduje, o membranose esantys baltymai veikia kaip katalizatoriai.

Esminė P sistemos sudedamoji dalis yra membranų struktūra, kur membranos gali būti išsidėsčiusios hierarchine tvarka kaip medis arba kaip membranų tinklas. Membranos idėja primena trimatę vezikulę iš biologijos, tačiau pati jos sąvoka yra apibendrinta: membrana interpretuojama kaip skyriklis dviejų sričių (Euklido erdvės): ribotos vidinės erdvės ir neribotos išorinės, sudarantis galimybę šioms dviem sritims selektyviai komunikuoti.

Iš biologijos pasisemtų idėjų įvairovė ir galimybių apibrėžti membranų pagrindu sukurtų duomenų apdorojimo mechanizmų architektūrą ir funkcijas spektras yra praktiškai beribis, todėl jau ir dabar MS literatūra pateikia daugybę įvairių modelių. Taigi MS nėra su konkrečiu modeliu siejama teorija: ji veikiau yra struktūra, naudojama kategorizuotiems modeliams kurti. Sritis yra pakankamai jauna (Pāun 2000), ir svarbiausias jos bruožas, grindžiamas ir biologijos kontekstu bei naudojamu matematinio formalioju metodu. Šiuo metu pasiūlyta daug P sistemų tipų, tačiau P sistemų lankstumas ir įvairiapusiškumas yra beribis. Sparčiai didėjantis šios srities tyrimų skaičius patvirtina, kad MS neįmanoma pristatyti glaustai ir tiksliai. Vis dėlto egzistuoja visa eilė idėjų, žymėjimo sistemų ir modelių, kurie jau tapo standartiniais, stabiliais ir laikytini kertiniais MS elementais.

### 1.3.1. Membraniniai skaičiavimai – gamtinių skaičiavimų dalis

Gaminiai skaičiavimai – pastangos pasisemti kompiuterių mokslui naudingų idėjų, modelių ir paradigmų iš natūralios gamtos, o ypač iš gyvybės. Gamtoje vystantys natūralūs procesai gali būti atkartoti kuriant naujus matematinius modelius. Klasikinė biologijos grįstų gamtinių skaičiavimų šaka yra genetikos algoritmai ir dirbtinė gyvybė. Abi kryptys patvirtina, kad verta mokytis iš biologijos, ir palaiko optimistinę nuostatą, kad per daugybę milijardų metų gamta ir gyvybė išstobulino tam tikras priemones ir procesus, kuriuos tinkamai juos pritaikius kompiuterių mokslo reikmėms, galėtų būti stebėtinais naudingi ir įvairiai pritaikomi.

Gyvybė tiesiogiai susijusi su ląstelėmis. Ląstelė yra mažiausias objektas, kurį visuotinai laikome gyvu. Ji labai maža, sudėtingos struktūros, atlieka sudėtingas funkcijas, jos viduje vyksta sudėtingi procesai, o išoriškai ji sąveikauja su šalia esančiomis ląstelėmis ir aplinka. Per milijardus evoliucijos metų ji išvystė būdą organizuoti ir valdyti biocheminius, ir informacinius procesus.

Ląstelės – tai membranos. Pačią ląstelę apibrėžia ir nuo aplinkos atskiria išorinė membrana. Ląstelės viduje keletas membranų sudaro sritis, kuriose vyksta biocheminiai procesai. O būtent, membranose yra branduoliai sudaryti iš eukariotinių ląstelių, kuriuose saugoma genetinė medžiaga. Pro membranose esančias vezikules molekulių grupės gali būti transportuojamos iš vienos ląstelės dalies (pvz., iš Goldžio komplekso) į kitą ląstelės dalis tokiu būdu, kad šio transportavimo metu molekulės nėra „prieinamos“ jas supančiai terpei.

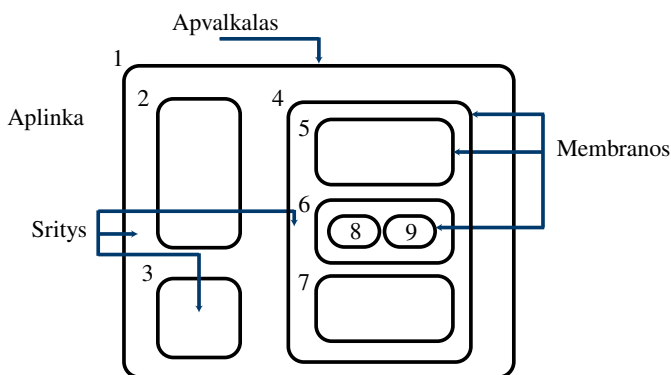
Membranos leidžia medžiagoms judėti iš vieno skyriaus į kitą selektyviai (Al-

berts *et al.* 2002). Ši selekcija gali būti atliekama vien pagal dydį, kai juda mažos molekulės, arba ji gali vykti kur kas sudėtingiau – per baltymų kanalus, kurie molekulės ne vien atrenka, bet ir leidžia joms judėti iš žemos koncentracijos į didesnę koncentraciją, galbūt joms jungiantis, kai vyksta simportiniai ar antiportiniai procesai. Be to, ląstelių membranos neatskiria vien tų jungčių, kurių viduje vyksta tam tikros cheminės reakcijos: daugelis ląstelėje vykstančių reakcijų gali vykti pačių membranų viduje, o membranose esantys baltymai veikia kaip katalizatoriai (Rubin 1999). Sakoma, kad tada, kai jungčių skaičius yra per didelis, kad jame vykstantys biocheminiai procesai būtų veiksmingi, gyvybė suformuoja membranas, kad sukurtų mažesnius reaktorių – pakankamai nedidelius, kad vykstant Brauno judėjimui, bet kuriuos dvi šalia esančios ląstelės galėtų sąveikauti, kitaip tariant tarp jų pakankamai dažnai vykėtų reakcijos ir kad susikurtų nauji „reakcijos paviršiai“ (Ioan 2002).

Egzistuoja ląstelės, kurios gyvena pavieniui (vienaląsčiai organizmai), bet paprastai ląstelės jungiasi į audinius, organus ir organizmus. Visais šiais atvejais egzistuoja sąveika, kuri prasideda nuo tiesioginio kaimyninių ląstelių komunikacijos (bendradarbiavimo), o baigiasi įvairaus lygmens jų sąveika su aplinka. Šis faktas kelia daug idėjų, kurios yra įdomios matematinio požiūriu ir galimai yra naudingos informatikos požiūriu. Kai kurias iš šių idėjų jau ištyrinėjo MS, bet dauguma jų dar laukia tyrimų rezultatų.

### 1.3.2. P sistema

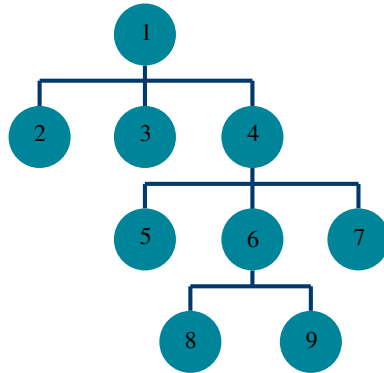
Viena iš pamatinių P sistemos sudedamųjų dalių yra membranų struktūros. Šios idėjos reikšmę atspindi 1.6 paveikslas.



1.6 pav. Membranų struktūra (Paun *et al.* 2006)

Fig. 1.6. A membrane structure (Paun *et al.* 2006)

Membranos struktūrą 1.6 paveiksle sudaro rinkinys membranų, hierarchiškai išsidėsčiusių tam tikroje išorinėje membranoje, atitinkančioje plazmos membraną ir paprastai vadinamoje apvalkalo membrana. Apvalkalo membranos viduje gali būti keletas membranų – jos atitinka ląstelėje esančias membranas. Aplink branduolį esanti membrana, kurios viduje nėra kitų membranų, laikytina elementariąja membrana. Kiekviena membrana turi sritis, kurios yra atskirtos nuo kitų, esančių virš jos ir žemiau jos. Galime pastebėti, kad membrana ir sritis tiksliai atitinka viena kitą – štai kodėl kartais vartojame sukeičiamumo sąvoką.



**1.7 pav.** 1.6 paveikslo membranų struktūros medis (Paun *et al.* 2006)

**Fig. 1.7.** The tree of the membrane structure from Fig. 1.6 (Paun *et al.* 2006)

Membranų hierarchinę struktūrą galima atvaizduoti išsisknijusiu medžiu – 1.7 paveiksle pateiktas medis, atitinkantis 1.6 paveiksle pateiktą membranų struktūrą. Medžio šaknys vaizduoja apvalkalo membraną, o lapai – elementariąsias membranas. Taip buvo sukurta daug grafinių teorinių idėjų, pvz., medžio atstumai, membranos lygis, membranos struktūros aukštis (gylis), taip pat tokie terminai, kaip motininė (dukterinė) membrana, įpėdinis ir pan. Iš medžio atvaizdo tiesiogiai kyla simbolinė membranos struktūros išraiška taikant skliaustų ir membranos indeksų seką. 1.7 paveiksle pateikto medžio struktūrą atitinka ši seka:

$$[1[2]2[3]3[4]5]5[6[8]8[9]9]6[7]7]4]1. \quad (1.1)$$

P sistema yra membranos struktūra su objektais membranose ir evoliucijos taisyklėmis, taikomomis objektams. Objektai evoliucionuoja pagal evoliucijos taisykles, kurios taip pat yra lokalizuotos ir susietos su membranų struktūros sritimis. Iš tiesų yra trys pagrindinių taisyklių tipai:

1. Aibių taisyklės (dažniausiai vadinamomis evoliucijos taisyklėmis).

2. Komunikacijos taisyklės.
3. Membranų tvarkymo taisyklės.

Šios taisyklės atitinka chemines reakcijas, kurios gali vykti ląstelėse, taigi jų forma yra  $u \rightarrow v$ , čia  $u$  ir  $v$  yra objektų aibės. Tačiau norėdami, kad sritys bendradarbiautų, turime priversti objektus judėti pro membranas, todėl prie objektų, atsiradusių pagal nurodytą taisyklę (objektų, atsiradusių iš rinkinio  $v$ ), pridėdame tikslo funkcijas. Šios funkcijos yra *čia*, *vidun* ir *išorėn*. Jos reiškia, kad su indikacija *čia* asocijuojamas objektas išlieka toje pačioje srityje, su indikacija *vidun* asocijuojamas objektas persikelia į gretimą žemesnę membraną, kuri nėra pasirenkama pagal konkrečius kriterijus, o *išorėn* reiškia, kad objektas turi palikti membraną, taigi tampa ją supančios aplinkos objektu.

P sistemos aprašo pavyzdys:

$$\Pi = (O, C, \mu, w_1, w_2, \dots, w_m, R_1, R_2, \dots, R_m, i_o), \quad (1.2)$$

čia  $O$  – (baigtinė ir netuščia) objektų abėcėlė;  $C \subset O$  – katalizatorių rinkinys;  $\mu$  – membranos struktūra  $(1, 2, \dots, m)$ , sudaryta iš  $m$  membranų;  $w_1, w_2, \dots, w_m$  – sekos, didesnės už  $O$ , žyminčios membranos struktūros sričių  $1, 2, \dots, m$  objektų aibės;  $R_1, R_2, \dots, R_m$  – membranos struktūros sričių baigtiniai evoliucijos taisyklių rinkiniai;  $i_o$  – viena iš žymų  $1, 2, \dots, m$ , kai atitinkama sritis yra sistemos rezultato sritis arba 0, kai skaičiavimo rezultatas gaunamas iš sistemos aplinkos.

Apibendrinant pažymėtina, kad P sistemos konfigūraciją apibūdina membranos struktūra ir jos skyriuose esančių objektų aibė. Pradinė konfigūracija nurodoma apibūdinant membranos struktūrą tuo metu, kai skaičiavimai pradedami srityse esančių objektų sekomis, tai yra  $\mu, w_1, \dots, w_m$ . Sistemai evoliucionuojant ir taikant taisykles, gali keistis ir objektų aibės, ir membranos struktūra (Păun 2000).

### 1.3.3. Membraninių skaičiavimų taikymas

MS buvo sukurti siekiant surasti kompiuterijos mokslo idėjas, modelius, tikslinius įrankius ląstelės struktūros ir funkcionavimo atkūrimui, išskyrus ląstelių modeliavimą realiame laike. Tačiau, atsiribojant nuo ląstelių biochemijos, buvo sukurta nauja modeliavimo sistema (pradedant nuo naujos programavimo kalbos, koncepcijų komplektų ir programinių įrankių), kuri yra naudinga ne tik biologinių procesų, bet ir lingvistinių faktų, valdymo aspektų modeliavimui. Žinomos kelios programos, padedančios išspręsti su kompiuteriais susijusias problemas, pvz., rūšiuojant ir klasifikuojant problemas, imituojant tinklą su kryžminiu kanalų dubliavimu. Ciobanu *et al.* (2005) knygoje nagrinėjama didžioji šių taikomųjų programų dalis.

Kai kuriose iš programinių priemonių naudojama MS kalba. Ji sąlygoja ilgą koncepcijų sąrašą (naujai įdiegtų ar susietų nauju būdu šioje srityje) ir naują būdą pateikti ląstelių struktūrą. Šis pateikimo būdas yra patrauklus biologams: Eulerio-

Veno diagramos su srityse išsidėsčiusiais daugybiniais objektais (cheminėmis medžiagomis) ir su jų normų rinkiniais (išdėstytais srityse, perrašomų normų atvejais, arba prie membranų), todėl galima manyti, kad jos yra susijusios su membranomis.

Kitas lygmuo – tai MS programinių įrankių, technologijų, rezultatų naudojimas. Biologų keliamų problemų sprendimas kitomis sąlygomis ir kitoje sistemoje gali tapti išpūdingu pasiekimu, ir tai yra natūraliausias vystymo būdas, bet nebūtinai efektyviausias, bent jau pradiniam etape. Naudojant naujus įrankius gali kilti naujų problemų, kurios negali būti suformuluotos kitais skaičiavimų metodais. Pirmojo tipo problemos (jau išnagrinėtos biologų, daugiausia eksperimentiniame lygyje) apima, pavyzdžiui, procesų koreliaciją, tam tikrų cheminių medžiagų buvimą arba nebuvimą, jų daugialypumą (koncentraciją, populiaciją) tam tikrame objekte, tuo tarpu antrojo tipo problemos yra susijusios su biosistemų evoliucija, kai jos yra sumodeliuotos kaip dinaminės sistemos (pvz., susijusios su periodiškumu ir trajektorijų asimptotinėmis savybėmis, arba tam tikrų konfigūracijų pasiekiamumu).

Taikymas biologijoje bendrai sukuria tokio tipo scenarijų: viena sistema tikrina realumą, susijusį ląstelių biochemija, kita sukuria P sistemą, modeliuojančią atitinkamą procesą, kita surašo šią sistemą imituojančią programą (arba naudojančią vieną iš daugelio esamų programų), kita atlieka eksperimentus su programa, keičiant tam tikrus parametrus ir ieškant sistemos įvertinimo (paprastai tam tikrų objektų populiacijai). Tokiu būdu buvo ištirtas bakterijų kvėpavimas (Ardelean, Cavaliere 2003), fotosintezė (Nishida 2002), su imunine sistema susiję procesai (Franco, Manca 2004; Paun *et al.* 2006) ir kiti procesai (Ciobanu *et al.* 2003). Bet kokiu atveju, tyrimai yra preliminarūs, tačiau pažanga yra akivaizdi, ir tikimasi artimiausioje ateityje sukurti didelį susidomėjimą biologų tarpe turėsiančias taikomas programas. Taip pat tampa prieinama vis daugiau P sistemos įvairių klasių imitacijų (įgyvendinimų), su aiškesnėmis vartotojų sąsajomis, užtikrinančiomis palankią sąveiką su programa.

Ląstelę bandoma matematiškai sumodeliuoti – ypač žinant, kad biologams reikalingi matematiniai įrankiai, kurie galėtų padėti jų tyrimuose, ypač jie tikisi globalių ląstelės modelių. Kyla svarbus klausimas – kokius matematinius skaičiavimus reikia naudoti? Tradiciškai, matematiniai modeliai yra lyginami su testiniais matematiniais modeliais, ypač (dalinėmis) diferencialinėmis lygtimis – tai taikoma ir biologijoje. Rezultatai yra tik iš dalies patenkinami: ląstelė yra per daug sudėtinga, kad ją galima būtų modeliuoti kaip visumą, tuo tarpu diferencines lygtis biologams sunku išspręsti, suprasti, apskaičiuoti.

Beveik prieš pusę amžiaus panaši situacija buvo susidariusi lingvistikoje, kai netikėtai paaiškėjo, kad tinkamas požiūris yra ne tas, kuris remiasi testiniais matematiniais skaičiavimais, bet paremtas diskretiniais algoritminiais modeliais. Biologijoje yra dar vienas ypatumas: sukauptos didelės duomenų bazės (pvz., genomo projektas), kurios dar turi būti ištirtos. Būtinybė naudoti kompiuterius yra akivaizdi. Be to, dauguma šių duomenų yra diskretūs. MS pateikia ir nagrinėja tiek dis-



krečius, tiek algoritminius modelius – dvi patrauklias ir laiku apsvarstytas savybes. Ne kartą buvo bandoma pateisinti diskrečių modelių naudojimą nagrinėjant ląstelę, kaip visumą, kadangi biologams jie yra lengvai suprantami, lengvai išplečiami, efektyviai imituojami kompiuteriuose.

Tokiu atveju, jeigu nėra kitų prižasčių, P sistemos gali būti vertinamos kaip galimi ląstelės modeliai bent jau dėl tos akivaizdžios priežasties, kad tai yra pirmoji modelių klasė, tiesiogiai paremta ląstelės struktūra ir funkcionavimu, tuo tarpu anksčiau naudotos ląstelių modeliavimo priemonės buvo pasiskolintos iš kitų sričių, paprastai fizikos ir kompiuterijos mokslo, ir reikalavo daug pastangų siekiant pritaikyti jas biologinėje realybėje.

Pirminis MS tikslas siejamas su DNR skaičiavimu. Kodėl MS naudojama kaip kompiuterių mokslo šaka? Cavaliere *et al.* (2014) šį klausimą suformulavo daug tiksliau: kuo MS yra geresnė už kitas kompiuterių mokslo šakas? Atsakymas vėlgi gali būti pateikiamas keliuose lygiuose. Pirmia, ląstelė yra toks išskirtinis, milijardus metų „gludinamas“ mechanizmas, suteikiantis puikią galimybę rasti joje, jos architektūroje, medžiagų ir informacijos apdorojimo būduose įvairias kompiuterių mokslui naudingas idėjas, technologijas ir modelius. Kiti natūralaus skaičiavimo metodai yra gana patrauklūs. MS – tai sisteminis skaičiavimo modelių tyrimas, paremtas ląstelės struktūra ir funkcionavimu, tuo tarpu kitose kompiuterių mokslo šakose tokie tikslai nėra keliami. Visų pirma – priešingai nei DNR skaičiavimuose – MS skirtas tiesioginiam ląstelės suskaidymui.

P sistemose egzistuojantis skaičiavimų paskirstymas (lygiagretumas) yra esminis šios sistemos privalumas. Gamta beveik visada „skaičiuoja“ lygiagrečiai, nuo ląstelės dalių iki audinių, smegenų, organizmų, populiacijos. Technologiniu požiūriu nėra sudėtinga sujungti didelį skaičių elektroninių procesorių ar kompiuterių (kaip pavyzdį galima pateikti internetą); sudėtinga yra juos naudoti, koordinuoti, sinchronizuoti ir t. t. (paskirstytuose ar lygiagrečiuose skaičiavimuose komunikacijos kompleksiskumas yra vienodai svarbus ir prieštaraujantis klasikiniam laiko ir erdvės kompleksiskumui). Gamta išsprendė šią problemą tokiu būdu, kurį dar reikia atrasti ir inkorporuoti į mūsų sukurtas „dirbtinio“ skaičiavimo mašinas. „Natūralaus skaičiavimo“, ląstelės vientisumo, jos gyvybingumo išlaikymo būdų pagrindinė sąvoka yra decentralizacija, atsitiktiniai pasirinkimai, nedeterminuotas pasirinkimas, laisva kontrolė, asinchronizacija, skatinimo (slopinimo) kontrolės būdai, tačiau tai yra bendras iššūkis skaičiavimų srityje.

Aukštesniame lygyje yra įrodytas MS tinkamumas kompiuterių mokslo srityje dėl dviejų šiai sričiai bendrų dalykų: Tiuringo mašinos universalumo ir greito sudėtingų problemų sprendimo. Riba tarp universalumo ir neuniversalumo yra pagrindinė, daugeliu atveju netriviali, skaičiavimo problema. Ląstelė, išskirta MS sąlygomis, paverčiama labai galingu „kompiuteriu“, daugeliu atveju universaliu, įskaitant stebinantį ir filosofiniu požiūriu svarbų skaičiavimo metodą tik komunikacijos būdu P sistemose. Be to, daug pastangų buvo skiriama sudėtingumo kla-

sėms pagal skirtingus P sistemų tipus, kur galima sukurti eksponentinę erdvę linijiniame laike (pvz. taikant gerai žinomą biologinį procesą – membranos padalijimą). Rezultatas – tai įrodymas, kad tam tikro tipo P sistemos gali išspręsti tam tikrą problemą laike, tačiau buvo pateikti ir įdomesni rezultatai. Jau buvo užsiminta apie galimybę numatyti PSPACE ir charakterizuoti  $P \neq NP$  problemą. Kitas uždavinys susijęs su sistemų, skirtų problemų sprendimui, sudarymu. Dažniausiai Tiuringo mašina šias sistemas kuria vienodu režimu (pradedant nuo pavyzdžių dydžio), esant polinominiam veikimo laikui. Laisvesnės sistemos atveju leidžiama pusiau tolygi konstrukcija: ją Tiuringo mašina vykdo polinominiu veikimo laiku, tačiau pradedant nuo paties sprendžiamo pavyzdžio (galimybė dirbti polinominiu laiku užtikrina kūrimo „sąžiningumą“: problemos sprendimas nepateikiamas kūrimo etape). Ryšys tarp sudėtingumo klasių, paremtų tolygiomis ir pusiau tolygiomis konstrukcijomis iki šiol nėra pilnai išaiškintas. Be to, šioje srityje egzistuoja kelios sudėtingos hipotezės, susijusios su savybėmis, kurios yra būtinos norint pasiekti tam tikrą skaičiavimo efektyvumo lygį.

#### 1.4. Pirmojo skyriaus išvados ir disertacijos uždavinių formulavimas

Atlikus daiktų interneto ir membraninių skaičiavimų technologijų apžvalgą galima daryti šias išvadas pagrindžiant pagrindinius darbo uždavinius:

1. Siekiant pagerinti daiktų interneto savitvarkos kokybines charakteristikas greಿತaveikos, energijos sąnaudų ir kt. kriterijų pagrindu reiktų išspręsti daiktų interneto informacijos valdymo ir skirstymo uždavinį tam, kad sistemos veiktų nenutrūkstamai ir darniai.
2. Kuriant savitvarkius daiktų interneto valdymo algoritmus, reikia remtis biotechnologiniais algoritmais, nes gamta ir gyvybė išstobulino priemones ir procesus, kurie geba atsinaujinti ir prisitaikyti prie besikeičiančių sąlygų.
3. Remiantis moksline literatūra optimizavimo skruzdžių kolonijomis (OSK) algoritmai dažniausiai literatūroje nagrinėjamų biotechnologinių metaeuristinių algoritmų optimizavimo problemoms spręsti, todėl kuriant originalią daiktų interneto savitvarkos metodiką verta remtis šiuo algoritmu.
4. Membraninių skaičiavimų taikymo galimybės daiktų internete nėra tirtos, nors susidomėjimas daiktų internetu ir membraniniais skaičiavimais kasmet auga. Taigi verta ištirti galimybę įgyvendinti skruzdžių sistema ir membraniniais skaičiavimais grįsta daiktų interneto savitvarkos metodiką.

---

## Daiktų interneto savitvarkos metodika

Į vieningą tinklą susijungiant vis daugiau aplink mus esančių įrenginių, DI tinklas tampa svarbia mūsų kasdienio gyvenimo dalimi. Šio tinklo savaiminių organizavimąsi galima įvardyti kaip prietaisų komunikacijos užmezgimo proceso optimalų parinkimą. DI savaiminis organizavimasis turi keletą privalumų. Jis apima kitų aplinkų gedimų laipsnišką prevenciją, jei dėl aplinkoje esančių trikdžių duomenų tiekimas sutrinka (Evans 2013). Esant triktims, prietaisai teikia duomenis ir įgalina stebėti aplinką, kontroliuoti ar valdyti operacijas. Be to, savaiminis organizavimas yra gyvybiškai svarbus, nes jo dėka galime naudoti DI net esant trikdžiams.

DI savitvarkai labiausiai tinka gamta grįsti algoritmai (Gedminas 2019). Jie palaiko išmaniosios D2D komunikacijos charakteristikas, geba tinkamai pasinaudoti tinklo ištekliais realiuoju laiku, užtikrindami sklandžią komunikaciją. Optimizavimo skruzdžių kolonijomis (OSK) algoritmai yra vieni iš literatūroje plačiai nagrinėjamų spiečiaus algoritmų (Zedadra *et al.* 2018), žinoma daug šio algoritmo modifikacijų, o jo taikymo sričių ratas platus.

Šiame skyriuje, sprendžiant DI savitvarkos problemą, pateikiamos OSK algoritmo idėjos ir matematiniai modeliai. Siekiant OSK algoritmą pagreitinti siūloma taikyti membraninių skaičiavimų metodiką. Jos pagrindu pristatomas naujas algoritmas – max-min skruzdžių P sistema (MMSPS).

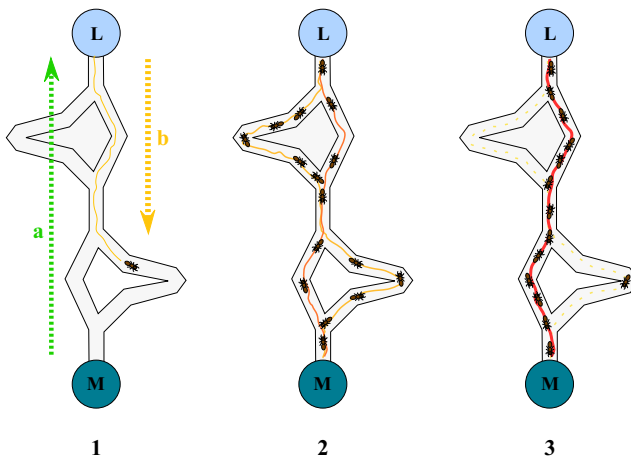
Skyriuje pateikti rezultatai skelbti autoriaus mokslinėse publikacijose (Gedminas *et al.* 2020; Gedminas 2019). Pagrindiniai rezultatai pristatyti tarptautinėse AIEEE (Ryga, 2016, 2019) ir respublikinėse „Mokslas – Lietuvos ateitis: Elektronika ir elektrotechnika“ (Vilnius, 2016, 2017) konferencijose.

## 2.1. Tiriamų skruzdžių kolonijų matematiniai modeliai

Pagrindinė OSK algoritmo idėja yra stigmergijos sąvoka gamtoje (Mirjalili 2018). Stigmergija nurodo, kad biologiniai organizmai gali manipuluoti aplinka, kad galėtų bendrauti tarpusavyje. Tokį bendravimą daro unikaliu tai, kad individai bendrauja netiesiogiai. Ryšys taip pat yra vietinis, o tai reiškia, kad asmenys turėtų būti šalia manipuluojamos zonos, kad galėtų ją pasiekti (Dorigo *et al.* 2000).

Skruzdės kolonijoje nuolat atsitiktine tvarka ieško maisto šaltinių aplink lizdą. Pratt *et al.* (2002) įrodė, kad skruzdė suradusi maisto šaltinį keliai iki jo žymi feromonu. Feromono kiekis labai priklauso nuo maisto šaltinio kokybės ir kiekio. Kuo maisto šaltinis didesnis, tuo stipresnis ir labiau koncentruotas feromonas paliekamas. Kitoms skruzdėms suvokus feromono buvimą, jos taip pat seka feromono taku, kad pasiektų maisto šaltinį. Gavusios dalį maisto, skruzdės jį nugabena į lizdą ir pažymi savo kelią į kitą vietą. Įdomu tai, kad ši paprasta veikla išsprendžia optimizavimo problemą, kaip skruzdei rasti artimiausią kelią nuo skruzdėlyno iki maisto šaltinio.

Tuo tarpu skruzdei paliekant feromoną take link maisto šaltinio, vyksta garavimo procesas. Tačiau laikotarpis, kurio metu skruzdė palieka feromoną prieš jam išgaruojant, yra ilgesnis, atvirkščiai proporcingas kelio ilgiui. Tai reiškia, kad feromonas trumpiausiam kelyje tampa labiau koncentruotas, nes daugiau skruzdžių pritraukia stipriausias feromonas (2.1 pav.).



2.1 pav. Skruzdės vykdoma maisto paieška (Salem, Sleit 2018)

Fig. 2.1. Ant seeking food (Salem, Sleit 2018)

Pirmuoju etapu skruzdės pradeda maisto paiešką. Antruoju etapu skruzdės atsitiktine tvarka pasklinda visuose keliuose. Trečiuoju etapu radusios maistą skruzdės grįžta trumpiausiu keliu, o kitos skruzdės seka jų paliktu feromono pėdsaku. Kituose keliuose feromonas išgaruoja ir visos skruzdės eina tuo pačiu – trumpiausiu keliu.

Pirmoji optimizavimo teorija, taikanti skruzdžių modelius, vadinama skruzdžių sistema, pasiūlyta italo Marco Dorigo, kuris 1992 m. apgynė pirmąją šios srities disertaciją. Šioje srityje taip pat yra ir kitų darbų, pvz., 1996 m. pasiūlyta patobulinta šio algoritmo versija „Max-Min Skruzdžių sistema“ (Dorigo, Gambardella 1996), o 1997 m. pasiūlytas optimizavimas skruzdžių kolonijomis (Dorigo, Gambardella 1997). OSK algoritmo matematinis modelis ir pseudo-kodas pateikiami tolesniuose poskyriuose.

### 2.1.1. Optimizavimas skruzdžių kolonijomis

Kaip aptarta anksčiau, OSK algoritmas iš pradžių buvo pasiūlytas kombinatorinių problemų sprendimui (Mullen *et al.* 2009). Šių problemų tikslas yra rasti optimalų poaibį iš duotosios baigtinės aibės. Dauguma kombinatorinių problemų yra NP sunkios (angl. *NP-hard*), todėl metaeuristiniai metodai visada buvo naudingi ir populiarūs.

Siekiant išspręsti kombinatorines problemas, buvo pasiūlyta OSK vykdyti trimis pagrindiniais žingsniais: pirminių parametrų atranka, feromono pėdsakų atnaujinimas ir vietinė paieška. OSK algoritmo pseudokodas ir pagrindinių fazių seka pateikta 2.1 algoritme.

#### 2.1 algoritmas (OSK algoritmo pseudokodas)

- 1 **Inicijavimas**
- 2 **While** (galutinė sąlyga nėra įvykdyta) **do**
- 3 Pirminių parametrų atranka 2.1
- 4 Rasti galimus sprendinius 2.3
- 5 Feromono pėdsakų atnaujinimas 2.4
- 6 **End**
- 7 **Geriausio sprendimo gražinimas**

Pirminių parametrų atrankos žingsnyje dirbtinės skruzdės kuriamos iš baigtinių  $n$  galimų sprendimų (Dorigo, Stutzle 2003). Kiekviena skruzdė apibrėžiama kaip verčių rinkinys, paimtas iš pagrindinio baigtinio rinkinio. Krašto parinkimo procesas atliekamas remiantis tikimybėmis. Pavyzdžiui, sprendžiant keliaujančio prekeivio uždavinį (KPU) kiekvienoje iteracijoje skruzdė gali pasirinkti visus nea-

plankytus kraštus, prieinamus iš dabartinio miesto. Šis pasirinkimo procesas atliekamas remiantis tikimybe:

$$p_{i,j} = \frac{\tau_{i,j}^{(\alpha)} \eta_{i,j}^{(\beta)}}{\sum \tau_{i,j}^{(\alpha)} \eta_{i,j}^{(\beta)}}, \quad (2.1)$$

čia  $\tau_{i,j}$  – pradinis  $i, j$  krašto feromono kiekis;  $\alpha$  ir  $\beta$  – santykinė feromono ir euristinės informacijos įtaka;  $\eta_{i,j}$  – skruzdei  $k$  iš miesto  $i, j$  pasiekiami nelankyti miestai.

Krašto pasiekiamumą apibrėžia svorių funkcija. Ši funkcija dažniausiai yra metaeuristinė ir priskiria vertę, kuri parodo, koks geras yra parinktas kraštas. Jei atstumas turėtų būti sumažintas,  $\eta_{i,j}$  jis gali būti apibrėžiamas taip:

$$\eta_{i,j} = \frac{1}{d_{i,j}}, \quad (2.2)$$

čia  $d_{i,j}$  –  $i, j$  krašto ilgis.

Feromono pėdsakų atnaujinimo žingsnis yra pagrindinis veiksmas, užtikrinantis ryšį tarp dirbtinių skruzdžių ir padedantis joms priimti sprendimus (Stützle 2009). Feromono garavimas ir feromono pėdsakai yra pagrindinės šio žingsnio fazės. KPU problemos sprendimui, pavyzdžiui, feromono kiekiui kraštui tarp  $i, j$  miestų yra apskaičiuojamas taip:

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j}, \quad (2.3)$$

čia  $\rho$  – feromono garavimo greitis;  $\Delta\tau_{i,j}$  – bendras feromono kiekis, nusėdęs ant  $i, j$  krašto, apibrėžto taip:

$$\Delta\tau_{i,j} = \begin{cases} \frac{1}{L_k}, & k\text{-oji skruzdė keliauja kraštu } i, j; \\ 0, & \text{kitu atveju,} \end{cases} \quad (2.4)$$

čia  $L_k$  – kelio, kuriuo keliauja  $k$ -oji skruzdė, ilgis (bendroji vertė).

Bendroji feromono atnaujinimo lygtis, kurioje feromonų reikšmės apibrėžia visos skruzdės, užrašoma taip:

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \sum_{k=1}^m \Delta\tau_{i,j}^{(k)}, \quad (2.5)$$

čia  $m$  – skruzdžių skaičius;  $\rho$  – feromono garavimo greitis;  $\Delta\tau_{i,j}^{(k)}$  – feromono kiekis, kurį nusodino  $k$ -oji skruzdė ant  $i, j$  krašto.

## 2.1.2. Max-min skruzdžių sistema

Max-min skruzdžių sistema (MMSS) – atnaujinta skruzdžių sistema pasižyminti didesniu našumu daugeliui optimizavimo problemų Stützle, Hoos (2000). MMSS papildomai analizuoja rastus geriausius kelius. Iteracijos geriausiai arba geriausią sprendinį radusiai iki šiol skruzdėi šis algoritmas leidžia palikti feromono pėdsaką. Tokia strategija lemia tai, kad visos skruzdės labai greitai pasirenka vieną ir tą patį optimalų kelią. Strategija įgyvendinama įvedant feromono ribas  $[\tau^\perp, \tau^\top]$ . Pradine feromono reikšme parenkama  $\tau^\top$  ir tai leidžia algoritmo vykdymo pradžioje geriau iširti sprendinius. Tačiau, jeigu per nustatytą skaičių iteracijų nepavyksta rasti geresnio sprendinio nei iki šio rastas sprendinys, feromono pėdsakams iš naujo priskinamos pirminės reikšmės.

### 2.2 algoritmas (Max-min OSK algoritmo pseudokodas)

- 1 **Max-min skruzdžių sistemos inicializavimas**
- 2 **While** (galutinė sąlyga nėra įvykdyta) **do**
- 3   Max-min parametrų nustatymas 2.7
- 4   Pirminių parametrų atranka 2.1
- 5   Rasti galimus sprendinius 2.3
- 6   Feromono pėdsakų atnaujinimas 2.6
- 7 **End**
- 8 **Geriausio sprendimo gražinimas**

Feromono garavimo apskaičiavimas vykdomas pagal įprastą skruzdžių sistemos lygtį (2.3). Kuomet visos skruzdės randa sprendinius, atnaujinamas feromono pėdsakas:

$$\tau_{i,j} \leftarrow \left[ (1 - \rho)\tau_{i,j} + \Delta\tau_{i,j}^\star \right]_{\tau^\perp}^{\tau^\top}, \quad (2.6)$$

čia naudojamos dvi strategijos kurios tarpusavyje yra keičiamos. Šioje formulėje  $\Delta\tau_{i,j}^\star = 1/C^\star$  geriausia skruzdė, kuriai leidžiama palikti feromono pėdsaką arba naudojant kitą strategiją  $\Delta\tau_{i,j}^\star = 1/C^\star$  geriausią iki šiol kelią radusi skruzdė, čia  $C^{ib}$  – kelio ilgis šioje iteracijoje. Šios strategijoje max-min skruzdžių sistemoje tarpusavyje keičiamos. Jeigu dažniau naudojama geriausios skruzdės strategija paieška vyksta greitai, apima mažą galimų sprendinių skaičių, tačiau nukenčia sprendinio kokybę. Taikant iteracijos geriausios skruzdės strategiją, feromonas yra paskirstomas didesnioje sprendinių srityje, paieška tampa lėtesnė, bet leidžia surasti geresnį sprendinį.  $\tau^\top$  yra viršutinė feromono riba,  $\tau^\perp$  rodo žemiausią feromono

ribą,  $[x]_{\tau_{\perp}}^{\tau_{\perp}}$  apibrėžiamas taip:

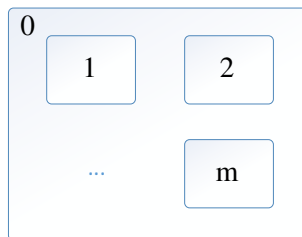
$$[x]_{\tau_{\perp}}^{\tau_{\perp}} \leftarrow \begin{cases} \max, & x > \max; \\ \min, & x < \min; \\ x, & \text{kitaip.} \end{cases} \quad (2.7)$$

Jei pėdsakas tampa žemesnis už minimalią pėdsako ribą, pėdsakui nustatoma minimali vertė ir, jei pėdsakas yra didesnis už maksimalią pėdsako ribą, pėdsakas nustatomas maksimaliai. Po pėdsako atnaujinimo skruzdės konstruoja naują sprendimų grupę ir procesas kartojasi. Jei aptinkama algoritmo stagnacija, pėdsakai atstatomi į pradines vertes.

Iki šiol Zhou (2009) atlikti eksperimentiniai tyrimai parodė, kad esant nesunkiems KPU, kuomet yra nedidelis miestų skaičius, patartina naudoti iteracijos geriausios skruzdės strategiją, nes sprendinių skaičius nėra didelis. Tačiau didinant miestų skaičių, patartina taikyti geriausios skruzdės strategiją, taip randant tikslesnį sprendinį.

## 2.2. Nauja max-min skruzdžių P sistema

Šiame poskyryje pateikiama modifikuota ląstelių tipo P sistemų ir jų evoliucijos taisyklių hierarchinė struktūra, o parametrizuojamas tikimybinis modelis, t. y. feromonų modelis, OSK, nurodomas siekiant patikslinti max-min skruzdžių P sistemos (MMSPS) algoritmą. Objektai susideda iš skruzdžių arba DI įrenginių maršrutų grafų. Evoliucijos taisyklės, kurios yra atsakingos už sistemos evoliuciją ir geriausios skruzdės parinkimą, apima maršrutų konstravimą ir transformacijos (komunikacijos) taisykles, įgyvendintas naudojant vietines ir globalias feromonų atnaujinimo taisykles.



2.2 pav. Membranos struktūra

Fig. 2.2. Membrane structure



Tikslesnis P sistemos (2.2 pav.) aprašas:

1. Vieno lygio membranos struktūra  $\mu = [0[1]_1, [2]_2, \dots, [m]_m]_0$  su  $m + 1$  regionais kuriuos riboja  $m$  pradinės membranos, ir membranos apvalkalas, žymimas 0.
2. Kodų rinkinys, apimantis visas skruzdės.
3. Terminuotų simbolių rinkinys, DI konstrukcijos grafai.
4. Pradinės daugialypės terpės  $|w_0 = \lambda, w_1 = A_1 A_2 \dots A_{M_1}, w_2 = A_{M_1+1} \times A_{M_1+2} \dots A_{M_2}, w_m = A_{M_{(m-1)+1}} A_{M_{(m-1)+2}} \dots A_{M_m}$ .
5. Taisyklės kategorizuojamos taip:
  - Maršrutų konstrukcijos taisyklės nuo 0 iki  $m$ ; tai yra į transformaciją panašios taisyklės, kurios sukuria maršrutus skruzdėms.
  - Komunikacijų taisyklės, kuriose feromonų reikšmės naudojamos atnaujinti DI grafų ribas. Yra trys komunikacijos taisyklių lygiai. Pirmasis lygis, atitinkantis OSK vietinės feromono atnaujinimo strategiją, naudojamas keistis informacija tarp dabartinės skruzdės ir būsimos skruzdės. Antrasis ir trečiasis komunikacijos taisyklių lygiai kilę iš globalios feromono atnaujinimo strategijos OSK. Antrasis įgyvendina informacijos apsikeitimą tarp geriausios skruzdės ir skruzdžių, likusių tam tikroje membranoje. Trečiasis atlieka komunikaciją tarp pradinių membranų skruzdžių ir apvalkalo membranoje esančių skruzdžių.

MMSPS pradinė skruzdžių kolonija yra išsklaidyta po membranų struktūrą. Pradinę koloniją sudaro daugialypės terpės  $w_1, \dots, w_m$ . Kiekviena iš pirminių membranų skruzdžių nuosekliai kuria savo maršrutus naudodamos (a) tipo taisyklės. Atlikdama  $N$  (įrenginių skaičius) žingsnius, skruzdė suplanuoja visą kelią, skirtą  $N$  įrenginiams. Jei visos skruzdės turi savo kelią, dabartinė karta yra vertinama pagal tinkamumo principą, siekiant pasirinkti tinkamiausią skruzdę kiekvienai pradinei membranai. Geriausia skruzdė naudojama norint sureguliuoti feromonų vertes DI įrenginių grafe, kad būtų galima komunikuoti su kitomis tos pačios pradinės membranos skruzdėmis. Kiekvienos  $g_i (i = 1, 2, \dots, m)$  sekcijos geriausia skruzdė siunčiama į membranos išorę. Tokiu būdu  $m$  skruzdės iš  $m$  pradinės membranos sudaro pradinius objektus apvalkalo membranoje. Šios skruzdės savarankiškai vystosi  $g_0$  kartoms tam, kad išrinktų geriausią skruzdę, kuri geriausiai komunikuotų su skruzdėmis kiekvienoje iš pradinių membranų. Procesas bus sustabdytas pagal iš anksto nustatytas nutraukimo sąlygas, pvz., tam tikrą iteracijų skaičių. MMSPS pseudokodas pateiktas 2.3 algoritme.

## 2.3 algoritmas (MMSPS algoritmo pseudokodas)

```

1 Pradžia
2  $t \leftarrow 1$ 
3 While (nenutrukstamumo sąlyga) do.
4   Padalinamos skruzdės po pradines membranas
5   Nustatomos kiekvienos pradinės membranos iteracijas
6   for  $i = 1, 2, \dots, m$  do
7     Inicijuojami parametrai  $t = 0, N_c = 0, \Delta\tau_{ij}(0) = 0$ 
8     for  $i, j = 1, 2, \dots, m$  do
9       nustatoma pradinė feromono vertė  $\tau_{ij} = \text{const}$ 
10    end
11    for  $N_c < N^\top$  do
12       $N_c \leftarrow N_c + 1$ 
13      for  $k = 1, 2, \dots, m$  do
14        if  $L \neq D$  then
15          Pasirenkamas kitas mazgas pagal tikimybę (2.1)
16          Pakeičiamas Tabu
17        end
18      end
19      Išsaugoma  $p_{ij}$  reikšmė
20      Atnaujinamas feromonas pagal (2.3) formulę
21      if  $\Delta\tau_{ij} < \tau^\perp$  then
22         $\Delta\tau_{ij} = \tau^\perp$ 
23      end
24      if  $\Delta\tau_{ij} > \tau^\top$  then
25         $\Delta\tau_{ij} = \tau^\top$ 
26      end
27    end
28     $p_i \leftarrow p_{ij}^\star$ 
29  end
30  Apvalkalo membranoje suformuojama skruzdžių kolonija
31  Atliekami MMSS (6–29 eilutės) skaičiavimai apvalkalo membranoje
32  Pradedama globali komunikacija
33   $t \leftarrow t + 1$ 
34  end
35 Pabaiga

```

Žemiau aprašomi svarbiausi algoritmo etapai:

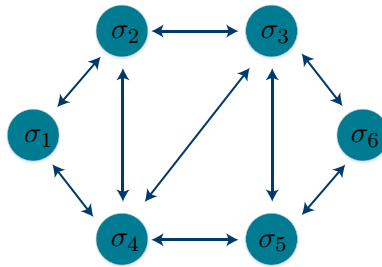
- Antrajame etape yra sukurta vieno lygio membranos struktūra, pavaizduota 2.2 paveiksle.
- Ketvirtuoju etapu  $M$  skruzdės, sudarančios koloniją, yra išsklaidytos po  $m$  pradinės membranas tokiu būdu, kuris garantuoja, kad kiekvienoje pradinėje membranoje yra bent dvi skruzdės. Tai naudinga įgyvendinant antrąjį komunikacijos proceso lygį. Taigi, skruzdžių skaičius pradinėje membranoje svyruoja nuo  $M - 2$  iki  $m + 2$ .
- Penktasis žingsnis nustato kiekvienos pradinės membranos iteracijų skaičių, kad būtų galima atlikti nepriklausomą MMSS. Konkrečiau,  $i$  pradinės membranos iteracijų skaičius  $g_i (i = 1, 2, \dots, m)$  generuojamas atsitiktinai tarp  $g^+$  ir  $g^+$ , t. y.  $g_i = g^+ + \lfloor \text{rand}(0,1) \cdot (g^+ - g^+) \rfloor$ , čia  $g^+$  ir  $g^+$  yra žemesnės ir viršutinės pradinės membranos iteracijų ribos, atitinkamai; tai yra funkcija, suapvalinanti elementą iki artimiausio mažesnio sveiką skaičių.
- Nuo 6 iki 29 etapo kiekvienoje  $m$  pradinėje membranoje, max-min algoritmas (2.2) atliekamas savarankiškai, t. y. maršrutų konstrukcija, vietinis feromonų atnaujinimas ir visuotinis feromonų atnaujinimas yra nuosekliai atliekami  $g_i (i = 1, 2, \dots, m)$  iteracijų atveju.
- Trisdešimtam etape skruzdžių kolonija apvalkalo membranoje sudaroma naudojant geriausias pradinių membranų skruzdes. Kiekviena sekcija siunčia geriausią skruzdę į apvalkalo membraną, todėl iš viso yra  $m$  skruzdžių.
- Trisdešimt pirmuoju etapu globali komunikacija naudojama siekiant keistis tam tikra informacija tarp skruzdžių, esančių apvalkalo membranoje ir skruzdžių, esančių pradinėse membranose. Geriausia apvalkalo membranoje esanti skruzdė atnaujinama feromonų vertes DI įrenginių diagramose kiekvienoje pradinėje membranoje. Ši operacija turi teigiamą poveikį skruzdžių sekcijoje siekiant atlikti greitesnę skaičiavimą bei gauti trumpesnę kelią.

### 2.3. Evoliucijos taisyklės daiktų interneto duomenims skirstyti

Daiktų interneto informacijos valdymo ir skirstymo problema gali būti matematiškai modeliuota grafu  $G = (V, E)$ , čia  $V = \{0, 1, \dots, n\}$  – mazgų, atstovaujančių serveriui (0) ir siuntėjams  $\{1, \dots, n\}$  rinkinys;  $E = \{(i, j) | i, j\}$  – kelio rinkinys, kai kiekvienas turi minimalų laiką  $t_{ij}$ , reikalingą duomenims perduoti iš  $i$  mazgo į  $j$  mazgą (Astudillo, Kadoch 2019). Problemos sprendimo tikslas – rasti geriausią

kelią, tenkinantį tris skirtingus parametrus: delną, energijos suvartojimą ir tinklo balansą. Kadangi DI susideda iš nevienalyčių sistemų, šis tikslas turėtų būti pasiektas kiekvienam tinklo tipui, įskaitant ir iš dalies sutampančias sritis.

Kadangi P sistema yra pagrįsta grafu, prilyginame ją daiktų interneto informacijos skirstymo ir valdymo problemos grafui  $G = (V, E)$ , sudarytame iš mazgų ir kelių rinkinių. Pateikus paprastą P sistemą  $\Pi = (O, K, \delta)$ , čia  $\delta$  yra DI mazgų grafas, nustatome  $E \simeq \{(u, v), (v, u) | (u, v) \in \delta\}$ . Šiuo atveju, paieškos grafas yra simetriškas ir kiekvienas iš mazgų perduoda duomenis pirmyn ir atgal – turi dvipusio ryšio kanalus.



2.3 pav. P sistemos daiktų interneto grafas

Fig. 2.3. P system graph of Internet of things

Sekantis etapas identifikuoti kiekvieno mazgo kaimynus, t. y. sudaryti tinklo topologiją. Kiekvienas mazgas siunčia savo numerį visiems kaimynams ir išimena iš kaimynų atsiųstus mazgų numerius. Tai pradinis etapas, kuris yra identiškasis tiek kelių, tiek mazgų identifikavimui.

### 2.1 lentelė. Kaimynų paieška P sistemoje

Table 2.1. Neighbors search in the P system

Mazgas	Kaimynai	Keliai
$\sigma_1$	$\{\sigma_2, \sigma_4\}$	$\{n_2, n_4\}$
$\sigma_2$	$\{\sigma_1, \sigma_3, \sigma_4\}$	$\{n_1, n_3, n_4\}$
$\sigma_3$	$\{\sigma_2, \sigma_4, \sigma_5, \sigma_6\}$	$\{n_2, n_4, n_5, n_6\}$
$\sigma_4$	$\{\sigma_1, \sigma_2, \sigma_3, \sigma_5\}$	$\{n_1, n_2, n_3, n_5\}$
$\sigma_5$	$\{\sigma_3, \sigma_4, \sigma_6\}$	$\{n_3, n_4, n_6\}$
$\sigma_6$	$\{\sigma_3, \sigma_5\}$	$\{n_3, n_5\}$

Objektai, naudojami šiame etape yra  $\{a, k, z\} \cup \{g_j, u_j, n_j\}$ . Šie objektai turi tokias reikšmes:  $a$  – mazgas, pasiekiamas iš  $\sigma_S$ ;  $k$  – mazgo šaltinis;  $z$  – tikslo

mazgas;  $n_j$  rodo, kad  $\sigma_j$  – esamo mazgo kaimynas;  $g_j, u_j$  rodo, kad  $\sigma_j$  – tikslo mazgas;  $g_j$  – tik šaltinio mazge, o  $u_j$  šio apribojimo neturi.

Pradžioje mazgo šaltinis  $s_s$  turi  $g_j$  kopiją, nurodančią, kad tikslinio mazgo  $\sigma_j$  numeris ir kiti mazgai yra tušti. Visi mazgai praddami su pradine būseną  $s_0$ . Kiekvienas pasiekiamas mazgas progresuoja per būsenas  $s_0, s_1, s_2, s_3$  ir  $s_4$  pagal žemiau pateikiamas taisyklės.

$s_0$  būsenos taisyklės:

$$s_0 g_i \rightarrow \min s_0, \quad (2.8a)$$

$$s_0 g_j \rightarrow \min s_1 a k(u_j), \quad (2.8b)$$

$$s_0 u_i \rightarrow \min s_1 a z(u_i), \quad (2.8c)$$

$$s_0 u_i \rightarrow \max s_1, \quad (2.8d)$$

$$s_0 u_j \rightarrow \min s_1 a(u_j). \quad (2.8e)$$

$s_1$  būsenos taisyklė:

$$s_1 a \rightarrow \min s_2 a(n_i). \quad (2.9)$$

$s_2$  būsenos taisyklė:

$$s_2 a \rightarrow \min s_3 a. \quad (2.10)$$

$s_3$  būsenos taisyklės:

$$s_3 a \rightarrow \min s_4 a, \quad (2.11a)$$

$$s_3 u_j \rightarrow \max s_4. \quad (2.11b)$$

Pavyzdžiui kaip taisyklės yra įgyvendinamos  $\sigma_1$  mazge:

$$s_0 g_1 \rightarrow \min s_0, \quad (2.12a)$$

$$s_0 g_1 \rightarrow \min s_1 a k(u_1), \quad (2.12b)$$

$$s_0 g_2 \rightarrow \min s_1 a k(u_2), \quad (2.12c)$$

$$\vdots$$

$$s_0 g_n \rightarrow \min s_1 a k(u_n). \quad (2.12d)$$

Mazgo atliktų būsenų perėjimų  $\sigma_i, i \in \{1, 2, \dots, n\}$  aprašymas:

- $s_0 \rightarrow s_1$ : jeigu  $\sigma_i$  turi  $g_j$  jis tampa šaltinio mazgu. Šaltinio mazgas  $u_j$  objektą perduoda visiems savo kaimynams. Gavęs  $u_j$  objektą,  $\sigma_i$  mazgas tampa tikslo mazgu, jei  $i = j$ ; arba, kitaip, tarpiniu mazgu. Toliau kiekvienas mazgas perduoda vieną iš gautų  $u_j$  objektų visiems savo kaimynams.
- $s_1 \rightarrow s_2$ :  $\sigma_i$  mazgas perduoda  $n_i$  visiems savo kaimynams. Be to,  $\sigma_i$  suranka  $n_j$  objektus iš kaimynų.

- $s_2 \rightarrow s_3$ :  $\sigma_i$  mazgas surenka tolesnius  $n_j$  objektus iš kaimynų.
- $s_3 \rightarrow s_4$ :  $\sigma_i$  mazgas surenka tolesnius  $n_j$  objektus iš kaimynų. Be to,  $\sigma_i$  pašalina nereikalingus  $e_j$  objektus.

Sudarius esamo DI tinklo topologiją sukuriamas kraštų tarp mazgų P sistemos aprašas. Sudarant P sistemą atsižvelgiama į tinklo duomenų šaltinius ir tikslą – duomenų gavėją.

Sistemos įvestis: P sistema  $\Pi = (O, K, E, \delta)$ , kurioje tinklo šaltinio mazgas  $\sigma_s \in K$  turi žymeklį  $t_t$  identifikuojantį duomenų gavėjo numerį –  $\sigma_t \in K$ .

Sistemos išvestis: Jei  $S \neq t$ , kiekvienas mazgas  $\sigma_i \in K$  turi objektą  $P_i = \{p_j | (j, i) \text{ ir objektų rinkinį } C_i = \{c_j | (i, j)\}$ , kuris nusako krašto atsiribojimo maršrutus nuo  $\sigma_s$  iki  $\sigma_t$ , kur yra pasitelkiami šie kintamieji:

- Duomenų srautai:  $c_i \notin C_i, p_i \notin P_i, c_j \in C_i \Leftrightarrow p_i \in P_j$  ir  $c_j \in C_i \Rightarrow j \in \delta(i) \cup \delta^{-1}(i)$
- Duomenų šaltinis ir tikslas:  $P_s = \emptyset$  ir  $C_t = \emptyset$ .
- Įeinantis srautas = išeinantis srautas: jei  $i \notin \{s, t\}$ , tuomet  $|C_i| = |P_i|$ .
- Duomenų maršrutai:  $S(I) = U_{i \in I} \{j | j \in C_i\}$ ,  $S^{n-1}(I) = S(S(\dots S(I) \dots)) = 0$ .

Dėl tinklo srauto savybių įvedame sąlygą  $|C_s| = |P_t|$ , kuri nusako didžiausią atskirtų maršrutų skaičių.

Siūlomas įgyvendinimas turi du etapus:

1 etapas (aptikimo), naudojant  $s_0, s_1, \dots, s_3$  būsenas aprašytas aukščiau,

2 etapas, aprašytas žemiau nuosekliai naudojant  $s_4, s_5, \dots, s_{13}$  būsenas.

2.2 lentelėje pavaizduota algoritmo išvestis, P sistemai su tinklo struktūra, atitinkančią 2.3 paveikslą.

Kaimyniniai tinklo mazgų objektai aprašomi taip:  $N_i = \{n_j | j \in \delta(i) \cup \delta^{-1}(i)\}$ , čia  $i \in \{1, 2, \dots, n\}$ , nustatyti pirmu etapu.

## 2.2 lentelė. P sistemos maksimalių kelių vaizdavimas

**Table 2.2.** A representation of maximum paths for P system

Mazgas	$N_i$	$P_i$	$C_i$
$\sigma_1$	$\{n_2, n_4\}$	$\emptyset$	$\{c_2, c_4\}$
$\sigma_2$	$\{n_1, n_3, n_4\}$	$\{p_1\}$	$\{c_3\}$
$\sigma_3$	$\{n_2, n_4, n_5, n_6\}$	$\{p_2, p_4\}$	$\{c_5, c_6\}$
$\sigma_4$	$\{n_1, n_2, n_3, n_5\}$	$\{p_1\}$	$\{c_3\}$
$\sigma_5$	$\{n_3, n_4, n_6\}$	$\{p_3\}$	$\{c_6\}$
$\sigma_6$	$\{n_3, n_5\}$	$\{p_3, p_5\}$	$\emptyset$

Be pirmame etape naudojamų objektų rinkinio, antrame etape papildomai naudojamas objektų rinkinys  $\{b_j, c_j, d_j, e_j, f_j, h_j, m_j, p_j, q_j, r_j, t_j, x_j, y_j\} \cup \{v, w\}$ .

Mazge  $\sigma_i$ , šie objektai turi tokias reikšmes:

- $b_j$  rodo postūmį, gautą iš  $\sigma_i$  srauto perėmėjo  $\sigma_j$ ;
- $e_j$  įrašo postūmio siuntėją, jei  $\sigma_i$  dar nepasiektas;
- $h_j$  įrašo postūmio siuntėją, jei  $\sigma_i$  jau pasiektas;
- $c_j$  rodo, kad  $\sigma_j$  yra  $\sigma_i$  srauto perėmėjas;
- $p_j$  rodo, kad  $\sigma_j$  yra  $\sigma_i$  srauto pirmtakas;
- $t_j$  įrašo  $\sigma_i$  išsiųstą postūmį savo srauto pirmtakui  $\sigma_j$ ;
- $t_j$  įrašo atgalinio maršruto užklausa, po nesėkmingo postūmio  $\sigma_j$ ;
- $d_j$  rodo, kad  $\sigma_j$  yra  $\sigma_i$  paieškos perėmėjas;
- $w_j$  rodo, kad  $\sigma_j$  yra  $\sigma_i$  paieškos pirmtakas;
- $f_j$  rodo bandytą paieškos išplėtimą, gautą iš  $\sigma_j$ ;
- $m_j$  įrašo, kad  $\sigma_j$  buvo sėkmingai išbandytas;
- $x_j$  rodo, kad  $\sigma_j$  atmetė srauto išplėtimą arba srauto postūmio bandymą;
- $x_j$  rodo, kad  $\sigma_j$  priėmė srauto išplėtimą arba srauto postūmio bandymą;
- $v$  prašo, kad  $\sigma_i$  atstatytų visų išbandytų ir aplankyto kaimynų įrašą;
- $w$  prašo, kad  $\sigma_i$  išliktų nenaudojamas vienam žingsniui.

Pradžioje  $\sigma_S$  šaltinis turi vieną  $g_j$  kopiją, parodančią tikslo mazgo  $\sigma_j$  numerį, o kiti mazgai yra tušti. Atsižvelgiant į pirmojo etapo taisyklės, kiekvienas pasiekiamas mazgas progresuoja būsenoje  $s_4$ , kuri yra antrojo etapo pradžia, kurio taisyklės yra pateikiamos žemiau.

Elemento  $\sigma_i$  taisyklės būsenoje  $s_4$ :

$$s_4k \rightarrow \min s_5k, \quad (2.13a)$$

$$s_4z \rightarrow \min s_6z, \quad (2.13b)$$

$$s_4a \rightarrow \min s_7a. \quad (2.13c)$$

Elemento  $\sigma_i$  taisyklės būsenoje  $s_5$ :

$$s_5an_j \rightarrow \min s_5d_j(f_i)_{\downarrow j}, \quad (2.14a)$$

$$s_5d_jy_j \rightarrow \min s_{12}ac_jww(v), \quad (2.14b)$$

$$s_5d_jx_j \rightarrow \min s_5am_j, \quad (2.14c)$$

$$s_5b_j \rightarrow \min(s_5x_i)_{\downarrow j}, \quad (2.14d)$$

$$s_5f_j \rightarrow \min(s_5x_i)_{\downarrow j}, \quad (2.14e)$$

$$s_5ak \rightarrow \min s_{13}aaww(a). \quad (2.14f)$$

Elemento  $\sigma_i$  taisyklės būsenoje  $s_6$ :

$$s_6 n_j f_j \rightarrow \min s_6 p_j (y_i)_{\downarrow j}, \quad (2.15a)$$

$$s_6 v \rightarrow \min s_{12} w w(v), \quad (2.15b)$$

$$s_6 a a z \rightarrow \min s_{13} a a w w(a). \quad (2.15c)$$

Elemento  $\sigma_i$  taisyklės būsenoje  $s_7$ :

$$s_7 v \rightarrow \min s_{12} w w(v), \quad (2.16a)$$

$$s_7 a a \rightarrow \min s_{13} a a w w(a), \quad (2.16b)$$

$$s_7 n_j f_j \rightarrow \min s_8 q_j, \quad (2.16c)$$

$$s_7 c_j b_j \rightarrow \min s_8 e_j, \quad (2.16d)$$

$$s_7 h_j \rightarrow \min s_{11} c_j (x_i)_{\downarrow j}, \quad (2.16e)$$

$$s_7 p_j q_k \rightarrow \min s_{10} p_j q_k, \quad (2.16f)$$

$$s_7 q_j \rightarrow \min s_7 m_j (x_i)_{\downarrow j}, \quad (2.16g)$$

$$s_7 f_j \rightarrow \min s_7 (x_i)_{\downarrow j}. \quad (2.16h)$$

Elemento  $\sigma_i$  taisyklės būsenoje  $s_8$ :

$$s_8 a n_j \rightarrow \min s_9 a d_j (f_i)_{\downarrow j}, \quad (2.17a)$$

$$s_8 a \rightarrow \min s_{10} a. \quad (2.17b)$$

Elemento  $\sigma_i$  taisyklės būsenoje  $s_9$ :

$$s_9 d_j y_j e_k \rightarrow \min s_7 c_j m_k (y_i), \quad (2.18a)$$

$$s_9 d_j y_j q_k \rightarrow \min s_7 c_j p_k (y_i), \quad (2.18b)$$

$$s_9 d_j x_j \rightarrow \min s_8 m_j, \quad (2.18c)$$

$$s_9 c_j b_j \rightarrow \min s_9 c_j (x_i)_{\downarrow j}, \quad (2.18d)$$

$$s_9 n_j f_j \rightarrow \min s_9 m_j (x_i)_{\downarrow j}. \quad (2.18e)$$

Elemento  $\sigma_i$  taisyklės būsenoje  $s_{10}$ :

$$s_{10} a p_j \rightarrow \min s_{11} a r_j (b_i)_{\downarrow j}, \quad (2.19a)$$

$$s_{10} a e_j \rightarrow \min s_7 a c_j (x_i)_{\downarrow j}, \quad (2.19b)$$

$$s_{10} a q_j \rightarrow \min s_7 a m_j (x_i)_{\downarrow j}. \quad (2.19c)$$



Elemento  $\sigma_i$  taisyklės būsenoje  $s_{11}$ :

$$s_{11}r_j y_j e_k \rightarrow \min s_7 m_j m_k(y_i), \quad (2.20a)$$

$$s_{11}r_j y_j q_k \rightarrow \min s_7 m_j p_k(y_i), \quad (2.20b)$$

$$s_{11}r_j x_j \rightarrow \min s_{10} t_j, \quad (2.20c)$$

$$s_{11}c_j b_j \rightarrow \min s_7 h_j, \quad (2.20d)$$

$$s_{11}n_j f_j \rightarrow \min s_{11} m_j (x_i)_{\downarrow j}. \quad (2.20e)$$

Elemento  $\sigma_i$  taisyklės būsenoje  $s_{12}$ :

$$s_{12}w \rightarrow \min s_{12}, \quad (2.21a)$$

$$s_{12}v \rightarrow \max s_{12}, \quad (2.21b)$$

$$s_{12}m_j \rightarrow \min s_{12} n_j, \quad (2.21c)$$

$$s_{12}t_j \rightarrow \min s_{12} p_j, \quad (2.21d)$$

$$s_{12}k \rightarrow \min s_5 k, \quad (2.21e)$$

$$s_{12}z \rightarrow \min s_6 z, \quad (2.21f)$$

$$s_{12}a \rightarrow \min s_7 a. \quad (2.21g)$$

Elemento  $\sigma_i$  taisyklės būsenoje  $s_{13}$ :

$$s_{13}w \rightarrow \min s_{13}, \quad (2.22a)$$

$$s_{13}a \rightarrow \max s_0, \quad (2.22b)$$

$$s_{13}t_j \rightarrow \min s_0 p_j, \quad (2.22c)$$

$$s_{13}n_j \rightarrow \min s_0, \quad (2.22d)$$

$$s_{13}m_j \rightarrow \min s_0. \quad (2.22e)$$

Žemiau aprašoma keletas kraštų algoritmo žingsnių, pavyzdžiui, tarpinių mazgų į priekį ir konsolidavimo režimai.

Būsenos  $s_8$  taisyklės apima tarpinio mazgo režimo į priekį bandymą per ne srautinį kraštą. Jei  $\sigma_i$  dar neišbandė kaimyninio  $\sigma_h \notin P_i \cup C_i$ , paieška yra tęsiama su  $\sigma_h$  kitu atveju paieška yra nutraukiama.

Būsenos  $s_9$  taisyklės apima konsolidavimo režimą tarpiniam mazge  $\sigma_i$ , kuris perėjo į priekį ne srautiniame krašte. Konsolidavimo proceso metu,  $\sigma_i$  veiksmas priklauso nuo to, ar  $\sigma_i$  buvo pasiektas per srauto kraštą arba ne srautinį kraštą:

- Jei  $\sigma_i$  buvo pasiektas per srauto kraštą  $(i, j)$ , atvirkštinėje kryptyje,  $\sigma_i$  pakeičia savo srauto perėmėją  $\sigma_j$  su  $\sigma_h$ , kur  $\sigma_h$  yra jo paieškos pirmtakas.
- Jei  $\sigma_i$  buvo pasiektas ne per srautinį kraštą  $(k, i)$ ,  $\sigma_i$  nustato  $\sigma_k$  kaip srauto pirmtaką ir  $\sigma_h$  kaip srauto perėmėją, kai  $\sigma_h$  ir  $\sigma_j$  atitinkamai yra  $\sigma_i$  paieškos pirmtakas ir paieškos perėmėjas.

Būsenos  $s_{10}$  taisyklės apima tarpinio mazgo  $\sigma_i$  bandymą eiti į priekį per postūmį. Jei  $\sigma_i$  dar neišbandė srauto pirmtako  $\sigma_h$ , paieška yra pratęsiama su  $\sigma_h$  (t. y. naudojamas srauto kraštas  $(h, i)$ , atvirktine kryptimi) kitu atveju paieška yra nutraukiama.

Taisyklės būsenoje  $s_{11}$  apima konsolidavimo režimą tarpiniam mazgui  $\sigma_i$ , kuriam pavyko į priekį pastumti iki  $\sigma_h$  (t. y., srauto kraštą  $(h, i)$ , atvirktine kryptimi).

Konsolidacijos proceso metu,  $\sigma_i$  veiksmas priklauso nuo to, ar  $\sigma_i$  buvo pasiektas per srauto kraštą, ar per ne srauto kraštą:

- Jei  $\sigma_i$  buvo pasiektas per srauto kraštą  $(i, j)$  atvirktinėje kryptyje,  $\sigma_i$  pašalina savo srauto pirmtaką  $\sigma_h$  ir srauto perėmėją  $\sigma_j$ .
- Jei  $\sigma_i$  buvo pasiektas per ne srautinį kraštą  $(k, i)$ ,  $\sigma_i$  pakeičia savo srauto pirmtaką  $\sigma_h$  su  $\sigma_i$ , kur  $\sigma_k$  yra  $\sigma_i$  paieškos pirmtakas.

## 2.4. Daiktų interneto savitvarka grįsta max-min skrudžių P sistema

Kompiuterinis intelektas – tai sritis, apibūdinanti subsimbolinius metodus, daugiausia dėmesio skiriant strategijai ir rezultatams. Skrudžių kolonijos optimizavimas (Blum 2005) – tai tikimybinių algoritmų, pagrįstų maisto ieškančių skrudžių elgesiu, tyrimas ir šių algoritmų įgyvendinimu paieškos ir optimizavimo srityje. Skrudės paskleidžia feromonus kelyje, kuriuo jos keliauja iš kolonijos link maisto šaltinio. Feromonus jos skleidžia tiek keliaudamos pirmyn, tiek atgal. Aplinkoje feromonai išgaruoja. Kiekviename kelių susikirtime yra tikimybė pasirinkti vieną iš kelių. Tuo atveju kai visi tikėtini maršrutai yra nauji ir nė viename tų maršrutų feromonų nėra paskleista, vieno ar kito maršruto pasirinkimo tikimybė yra viena. Kadangi skrudės judant pirmyn ir atgal, priklausomai nuo kelionės dažnumo, feromonų sąnašų kiekis trumpesniame kelyje išauga. Toks feromonų padaugėjimas padidina tam tikro maršruto ir kito krašto pasirinkimo tikimybę feromonams skleisti. Šias savybes galime panaudoti kuriant daiktų interneto autonominės savitvarkos metodiką.

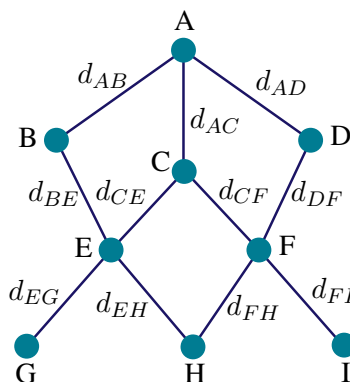
Terminas autonominis yra kilęs iš biologijos, kur, pavyzdžiui, autonominė žmogaus kūno nervų sistema rūpinasi daugeliu organizmo funkcijų, pašalindama iš sąmonės poreikį koordinuoti visas organizmo funkcijas. IBM manifeste (Kreger 2001) buvo teigiama, kad sudėtingos sistemos turėtų turėti autonomines savybes ir išskyrė keturias savarankiškai valdančių sistemų savybes: savarankiškos konfigūracijos, savarankiškos optimizacijos, atsikūrimo ir savarankiškos apsaugos. Plačiau šios savybės aprašytos 1.2 poskyryje.

Bendrinė max-min skrudžių sistema turi būti patobulinta siekiant ją pritaikyti konkrečiam daiktų interneto informacijos valdymo ir skirstymo uždaviniui.

Visų pirma turime padaryti prielaidą kad pasirinktoje daiktų interneto sistemoje esantys įrenginiai siunčia ir gauna skirtingą duomenų kiekį, todėl kiekvienos interakcijos metu turi būti parenkamas skirtingas duomenų siuntimo kelias. Teikiami sistemai duomenys yra ne vienodo dydžio. Sistemai parinkti operatorių seką, teikiamos duomenų siuntėjo ir gavėjo koordinatės.

Pirmiausia max-min skruzdžių sistemos algoritmas inicijuoja tolygų feromono pasiskirstymą visais keliais: informacijos šaltinis siunčia duomenų paketą. Tuo pačiu išgarina ir atnaujina feromono reikšmes. Tuomet skaičiavimai kartojasi, kol pasiekia nurodytą skaičių kartų.

Pagrindinė specializuotos max-min skruzdžių sistemos kūrimo idėja yra ta, kad skruzdės juda duomenų siuntimo operatorių ir funkcijų parametrų grafe. Kiekvienas šio grafo mazgas susietas su vieninteliu duomenų apdorojimo operatoriumi arba funkcijos parametru. Grafo briaunos – tai skruzdžių keliai, kuriais skruzdės kelias palikdamos feromoną ir taip pasirinkdamos geriausio kelio arba geriausios duomenų pralaidos sprendinį.



2.4 pav. Skruzdės kelio parinkimas

Fig. 2.4. Ant path selection

Šiam uždaviniui spręsti naudojame kitą, negu standartiniame MMSS algoritme, 2.4 paveiksle pažymėto kelio  $d$  skaičiavimo metodiką. Šiame metode kelio ilgio skaičiavimui yra naudojamas išmatuotų energijos verčių ir jungčių vėlavimų derinys:

$$d_{ij} = w_e E_{ij} + w_t t_{ij}, \quad (2.23)$$

čia  $E$  – mazgo normalizuota likutinė energija;  $t$  – mazgo  $k$  sugeneruotas vėlavimas;  $w_e$  ir  $w_t$  – šių verčių svoriai. Atitinkamai, feromonų atnaujinimas koreguoja-

mas pagal šią formulę:

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \frac{F}{P}; \quad (2.24)$$

$$P = \sum_{i=1}^n d_k, \quad (2.25)$$

čia  $F$  – feromono intensyvumas,  $\rho$  – feromono garavimo koeficientas.

Šiame metode naudojamas objektyvaus svėrimo būdas, vadinamas entropijos svoriu, kuomet nustatome briaunų svorius pradiniam etape. Šis metodas, skirtas nustatyti rodiklių svorį objektyviomis sąlygomis, atsižvelgiant į jų tinkamumą. Tai gali atspindėti paslėptą duomenų informaciją ir pagerinti rodiklių skiriamąją gebą, ir įvairovę. Entropijos svorio metodas nustato svorį pagal kiekvieno mazgo pateiktos informacijos entropijos dydį. Pagrindiniai apskaičiavimo veiksmai:

1. *Duomenų norminimas:*

$$Y_{ij} = \frac{X_{ij} - \min(X_i)}{\max(X_i) - \min(X_i)}. \quad (2.26)$$

2. *Informacijos entropijos skaičiavimas.* Pagal informacijos entropijos apibrėžimą informacijos teorijoje, duomenų rinkinio informacinė entropija yra:

$$H_j = -\log_2(n)^{-1} \sum_{i=1}^n p_{ij} \log_2 p_{ij}, \quad (2.27)$$

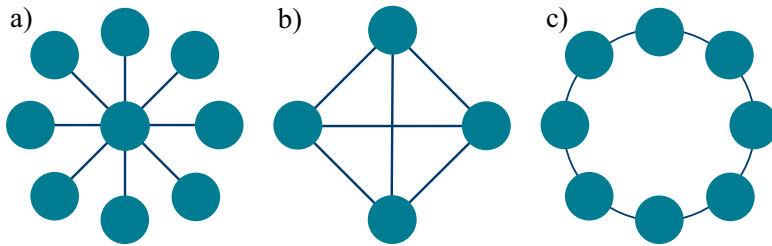
čia  $p_{ij}$  tikimybė apskaičiuojama taip:

$$p_{ij} = \frac{Y_{ij}}{\sum_{i=1}^n Y_{ij}}. \quad (2.28)$$

3. *Briaunų svorių apskaičiavimas.* Apskaičiuojame kiekvieno rodiklio svorį pagal informacijos entropiją:

$$w_i = \frac{1 - H_j}{k - \sum H_j}. \quad (2.29)$$

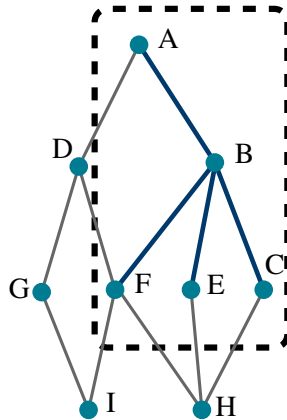
Norėdami sužinoti kiekvieno modeliuojamos DI sistemos įrenginio esamą baterijos talpą ir bendrą visų baterijų talpą, kiekvienos iteracijos metu atnaujiname skaičiavimus, o jų santykis yra likutinė galia procentais. Kita vertus, norint stebėti mazgo tinklo būklę, mazgas periodiškai tikrina tinklo būklę su savo kaimynais.



**2.5 pav.** Kaimynų padėties nustatymo topologijų pavyzdžiai: a) žvaigždės; b) tinklelio; c) žiedo

**Fig. 2.5.** Examples of neighborhood positioning topologies: a) star; b) mesh; c) ring

Naudojant 2.5 a paveiksle pavaizduotą kaimynų paieškos metodą DI tinklo mazgai atpažįsta tik informaciją apie kaimynus, o tai sumažina mazgo apkrovą. Už visuotinės informacijos suvokimą yra atsakingos dirbtinės skruzdės. Kaip parodyta 2.6 paveiksle, B mazgas renka informaciją apie kaimyninius mazgus B, C, E ir F. Tačiau apie D, G, H ir I informacija nėra renkama, nes B nėra tiesiogiai sujungtas su D, G, H.



**2.6 pav.** Vienos membranos sudarymas kaimynų atžvilgiu

**Fig. 2.6.** Single-membrane formation with neighbors

Pagrindinis skirtumas tarp siūlomo max-min skruzdžių algoritmo ir tradicinio skruzdžių algoritmo yra tas, kad siūlomas algoritmas nustato maksimalią ir mažiausią feromono vertę. Viena vertus, šis metodas neleidžia feromonams būti per dideliems ir lemia optimalų vietinį sprendinį. Kita vertus, tai neleidžia feromo-

nams būti per mažiems, o tai lemia greitą sprendinio radimą. Jie atsakingi tik už dirbtinių skruzdžių paketų persiuntimą ir feromonų įrašymą. Dvi skruzdžių būsenos yra apibrėžtos kaip į priekį ir atgal. Konvergencijos mazgas siunčia priekines skruzdes, kad pasiektų paskirties mazgą. Jie yra atsakingi už maršruto suradimą. Skruzdei pasiekus paskirties mazgą, ji tampa atgaline skruzde, ir grįžta pradiniu maršrutu.

## 2.5. Antrojo skyriaus išvados

1. Išnagrinėjus bendrinę skruzdžių sistemą, jos pagrindu aprašytos membraninių skaičiavimų P sistemos evoliucijos taisyklės.
2. Išnagrinėjus bendrinę max-min skruzdžių sistemą, jos pagrindu sukurtas naujas max-min skruzdžių P sistemos (MMSPS) algoritmas, grindžiamas membraniniais skaičiavimais.
3. Max-min skruzdžių P sistemos algoritmo pagrindu sukurta nauja daiktų interneto savitvarkos (DIS) metodika, kuri automatizuoja informacijos skirstymo ir valdymo procesus.

---

## Daiktų interneto savitvarkos tyrimas

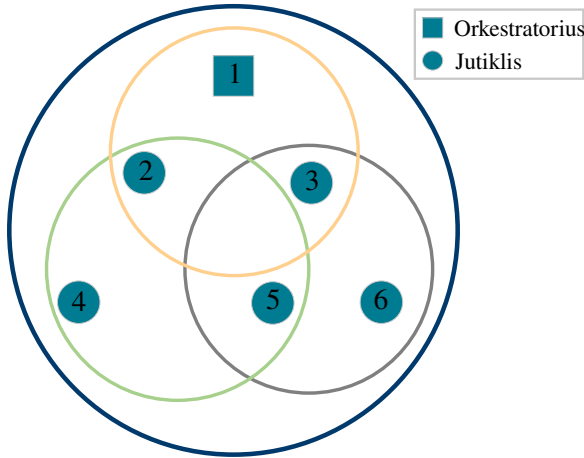
Daiktų internetas yra nepriklausomų įrenginių rinkinys, kuris vartotojams pateikiamas kaip viena darni sistema. Keitimosi duomenimis maršruto parinkimas ir jo tikslumas yra svarbiausias informacijos valdymo ir skirstymo algoritmo vertinimo kriterijus. Tam užtikrinti pasitelkta intelektualiaisiais metodais. Visų pirma, reikia išsiaiškinti, kurie duomenų srautai yra svarbūs ir reikalauja kokybinių paramet-rų (vėlinimas ir energijos vartojimas) užtikrinimo, o kuriems užtenka minimalių resursų. Srautus identifikuoti galime naudodamiesi informacija iš įrenginių ant-raščių. Identifikavus srautus reikia užtikrinti patikimiausią keitimosi duomenimis maršruto kelią. Tai gali būti pasiekta taikant sukurtą max-min skruzdžių kolonijų P sistemos (MMSPS) algoritmą. Siekiant parodyti algoritmo veikimą, modeliuo-ta situacija virtualiame tinkle, kuriame veikia šeši duomenų srautus generuojantys šaltiniai.

Skyriuje pateikti rezultatai skelbti autoriaus mokslinėje publikacijoje (Gedmi-nas *et al.* 2020). Pagrindiniai rezultatai pristatyti tarptautinėje AIEEE (Liepoja, 2019) konferencijoje.

### 3.1. Pasiruošimas imitaciniams eksperimentams

Eksperimentui atlikti MATLAB™ programinėje įrangoje Sukurta virtuali daiktų in-terneto sistema, kurią sudaro 6 mazgai. Tinklas plokštumoje pavaizduotas 3.1 pa-veiksle. Šiame paveiksle 4–6 yra jutiklio mazgai tinklo krašte; 2–3 yra tarpiniai

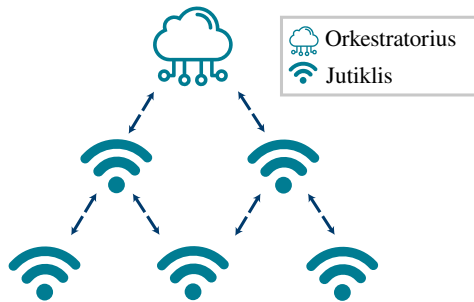
mazgai kurie atlieka duomenų perdavimo funkciją; 1 yra orkestratoriaus mazgas, o jo funkcija yra keistis duomenimis su internetu.



**3.1 pav.** Tiriama DI tinklo schema

**Fig. 3.1.** Investigated IoT network scheme

Šį tinklą galima atvaizduoti grafu – 3.2 paveikslas, kuris atitinka 3.1 paveiksle pateiktą tinklo struktūrą.



**3.2 pav.** DI tinklo topologijos grafas

**Fig. 3.2.** Graph of IoT network topology

Kiekvienam mazgui yra nustatytos pradinės reikšmės: energijos kiekis, duomenų pralaida. Kiekviename mazge yra aprašytas MMSPS algoritmas ir jis gali įgyvendinti pranešimų siuntimą, gavimą ir persiuntimą. Papildomai įdiegtas klientas, kuris gali išanalizuoti, saugoti ir apskaičiuoti gautą informaciją. Įdiegiant 3.2 pa-



veiksle pateiktą topologiją, tinkle naudojame jungiamąjį metodą. Tai leidžia įrenginiams prisijungti tik nurodytoje topologijoje. Siekdami imituoti DI tinklo aplinką, modeliudami naudojame mažą pralaidumą ir ilgą delką. Kalbant apie energijos suvartojimą, nustatome kiekvieno mazgo pradinę energiją lygią 100 %, o kiekvienos užmegztos komunikacijos metu naudojamo mazgo energiją sumažiname 0,01 %.

Eksperimentų grupės – tikslas ir valdymo priemonės. Įvesties duomenys – tinklo struktūra.

Esant skirtingoms jutiklio duomenų sąlygoms, duomenų perdavimo nuostoliai taip pat skiriasi. Kai atstumas tarp jutiklio mazgo ir bazinės stoties yra mažas, visų maršruto algoritmų duomenų perdavimo delsa yra panaši. Didėjant atstumui, pamažu didėja duomenų pristatymo delsa. Taip yra daugiausia todėl, kad didėjant atstumui nuo bazinės stoties padidėja duomenų perdavimui reikalingas šuolių skaičius ir atitinkamai sumažėja jų pristatymo greitis.

## 3.2. Daiktų interneto savitvarkos įgyvendinimo tyrimas

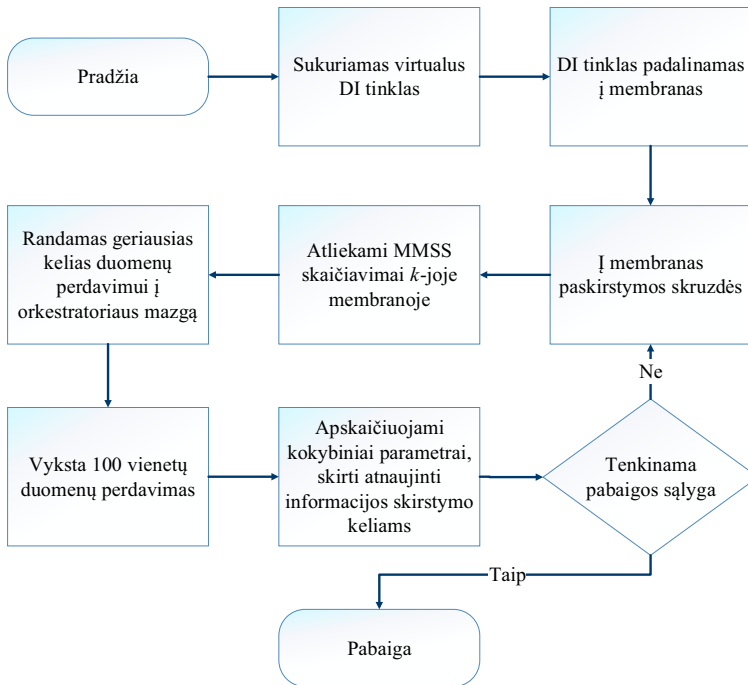
Siūlomas metodas turi tradiciniuose tinkluose retai pasitaikančią savybę – analizuoti viso tinklo topologijos būseną bei turėti visą informaciją apie ją. Tai padeda automatiškai rasti ir pritaikyti geriausią perduodamų duomenų srautų konfigūraciją be sistemų administratoriaus įsikišimo.

Skirtingai nuo dabar esamų informacijos valdymo ir skirstymo mechanizmų, metodas suteikia galimybę dinamiškai reaguoti į pasikeitusias tinklo charakteristikas ir geriau išnaudoti turimus DI tinklo resursus.

Kadangi metodas viso tinklo žemėlapių sudaro duomenų siuntimo ir gavimo metu, siūloma taupant mazgų energijos išteklius atsisakyti transliavimo (angl. *broadcast*) duomenų paketų srauto. Galiniams įrenginiams reikalingą informaciją siūloma perduoti centralizuotai tiesiai iš orkestratoriaus mazgo ir taip sumažinti tokio tipo duomenų paketų plitimą DI tinkle.

3.3 paveiksle pavaizduotoje eksperimento eigos blokinėje schemoje veikimą galima suskirstyti į šias pagrindines dalis:

1. Pradinės būsenos nustatymas.
2. DI grafas padalinamas į membranas.
3. MMSPS algoritmo taikymas trumpiausio kelio paieškai.
4. Duomenų perdavimas.
5. Parametrų stebėjimas.



**3.3 pav.** Eksperimento eigos blokinė schema  
**Fig. 3.3.** Block diagram of the experiment steps

Pradinės būsenos nustatymo metu vienas iš pirmųjų uždavinių yra sudaryti esamo tinklo topologiją – sukurti virtualų DI tinklą. Iš visų DI įrenginių ir tarpusavio jungčių sudaromas grafas ir kiekvienam mazgui yra nustatomos pradinės reikšmės: energijos kiekis, duomenų pralaida.

DI tinklo padalinimo į membranas etapu kaimyniniai mazgai apjungiami į bendrą struktūrą. Kiekvienas mazgas žino informaciją (baterijos talpą, ir duomenų pralaidumą) apie savo kaimynus.

Vykdamas max-min skruzdžių P sistemos 2.3 algoritimą surandamas trumpiausias kelias nuo duomenų siuntėjo iki gavėjo per tarpinius mazgus.

Duomenų perdavimo dalyje atliekamas duomenų siuntimas rastu trumpiausiu keliu. Atlikus duomenų perdavimą iš karto apskaičiuojami kokybiniai parametrai, skirti atnaujinti informacijos skirstymo keliams, geriau atitinkantiems duotųjų srautų kokybinius poreikius. Pagal gautus duomenis kiekvienam srautui sudaromi pagrindiniai ir alternatyvieji duomenų perdavimo keliai.

Parametrų stebėjimo procesas yra nuolatinis: stebima esama tinklo situacija bei reaguojama į įvykčius pasikeitimus. Pvz., padidėjus tam tikro sujungimo vė-

linimui ir viršijus nustatytą kažkuriam srautui ribą tam srautui bus ieškoma alternatyvaus kelio, kuris atitiks reikalaujamas kokybines charakteristikas.

Igyvendinant daiktų interneto savitvarkos duomenų maršrutus, kuriais nuo siuntėjo iki gavėjo turi nukeliauti duomenys, tuo pačiu metu stebima tinklo apkrova, ar ne visi duomenų paketai seka tuo pačiu maršrutu. Siekiant subalansuoti tinklą buvo įdiegtas informacijos skirstymo algoritmas, kuriuo yra parenkamas geriausias kiekvieno paketo pristatymas. Kadangi tinklo sąlygos kinta beveik visą laiką, kiekvieno maršruto energijos kaštai atitinkamai skiriasi, nes skiriasi duomenų perdavimui reikalingas šuolių skaičius, todėl maršrutizatoriai turi būti apie tai informuojami. Taigi, atnaujinimo pranešimų serija yra nuolat paskirstoma tarp mazgų, siekiant atnaujinti maršruto lentelėse saugomą informaciją. Taip pat įgyvendinant DI savitvarką turime atsižvelgti į tokius parametrus:

1. Tinklo balansavimas.
2. Atsparumas triktims.

Tinklo balansavimas – optimizavimas, kurį siūlomas algoritmas taiko siekiant prailginti tinklo gyvavimo laiką, keičiant duomenų maršrutus ir vienodai išnaudojant visus tinklo mazgų pajėgumus. Atsparumas triktims yra gebėjimas atsiradus gedimui prisitaikyti prie kintančios aplinkos ir sukurti naują patikimą ryšį.

Kalbant apie eksperimente taikytos skruzdžių kolonijos parametrus jos populiacijos dydis yra vienas iš pagrindinių parametrų, turinčių įtakos iteracijos vykdymo greičiui ir trukmei. Kuo daugiau skruzdžių yra membranoje, tuo ilgiau trunka viena iteracija. Jeigu kolonijos populiacijos dydis sudaro daugiau nei 30 % maršruto ilgio, tai sukelia persegmentavimą – kelių kolonijų buvimą viename kelyje. Kai populiacijos dydis yra per mažas, skaičiavimai atliekami greitai, bet duomenų kiekis yra nepakankamas ir kai kurie keliai lieka nepilnai ištirti. Didesnis jautrumas feromonui  $\alpha$  verčia skruzdės keliauti aplink su didesne tikimybe praleisti grafo viršūnę. Esant žemam feromono jautrumui, skruzdės juda pagal trumpiausią kelią, varžymasis mažėja, kiti keliai nepilnai dengiami feromonu.

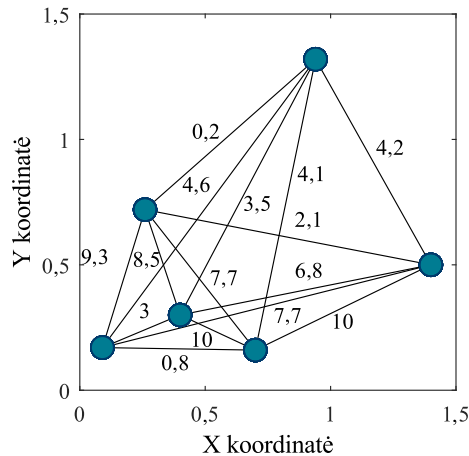
Su maža feromono jautrumo riba, esant aukštam feromono lygiui, skruzdės išsibarsto ir užima aplink esančius kelius. Su didele feromono jautrumo riba, skruzdės stipriai traukiamos feromono ir maršrutai išskiriami neteisingai.

Greitas feromono garavimas lemia žemą feromono kiekį, ir viena kolonija gali lengvai užimti kitos kolonijos kelią. Lėtas garavimas lemia aukštą bendrą feromono kiekį ir dėl to skruzdės traukiamos žemesnio feromono kiekio link.

Sprendžiant DI duomenų valdymo ir skirstymo uždavinį, galimų sprendimų skaičius priklauso nuo mazgų – įrenginių tinkle skaičiaus  $m$ . Tuomet galimų kombinacijų skaičius, tarus, kad duomenų paketas gali tik vieną kartą aplankyti konkretų mazgą, lygus  $(n - 1)!$ . Perrinkimo metodas garantuoja geriausią sprendinį, nes patikrinama kiekviena kombinacija. Taikant skruzdžių sistemos algoritmą mažuose tinkluose, geriausio sprendinio radimas gali būti netikslus. Taigi šis algoritmas

taikomas esant dideliam įrenginių tinkle skaičiui.

Prieš atliekant tyrimus, reikia įvertinti uždavinio sudėtingumą ir pateikti pirminius max-min skruzdžių sistemos parametrus. Kadangi skruzdės gali aplankyti jau lankytus mazgus, galimų sprendimų skaičius gali būti  $n^{l^T}$ . Žinant, kad skruzdžių skaičius  $m$  yra lygus mazgų skaičiui  $n$ , per vieną iteraciją, max-min skruzdžių sistemos modelis gali patikrinti  $n$  sprendinių.



**3.4 pav.** Atstumas tarp mazgų  $d$  išreiškiamas sunkumu

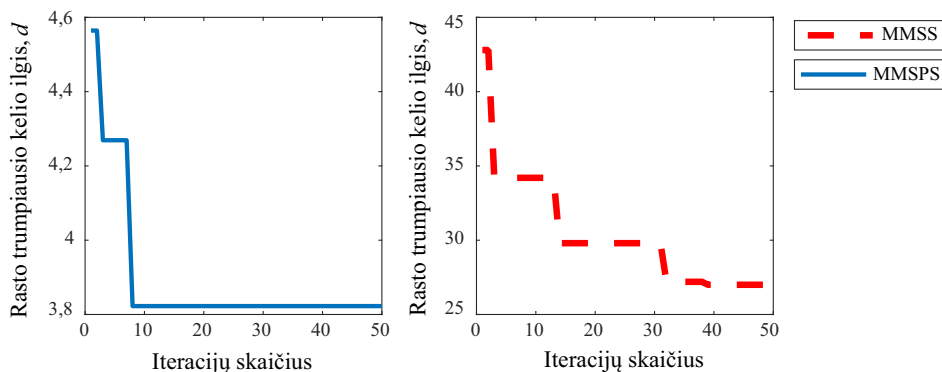
**Fig. 3.4.** Distance between nodes  $d$  is expressed by difficulty

3.4 paveiksle tarp mazgų pavaizduotos vertės yra vienas pagrindinių geriausio maršruto paieškos parametrų – atstumas tarp mazgų  $d$ , kadangi pasiūlytas metodas kelią tarp mazgų apskaičiuoja remdamasis ne geografine padėtimi, o daiktų interneto įrenginių energijos likutine verte ir vėlinimu, parametraž  $d$  vadiname sunkumu įveikti kelią ir apskaičiuojame šia lygtimi:

$$d_{ij} = w_e E_{ij} + w_t t_{ij}, \quad (3.1)$$

čia  $E$  – mazgo normalizuota likutinė energija;  $t$  – mazgo  $k$  sugeneruotas vėlavimas;  $w_e$  ir  $w_t$  – šių verčių svoriai. Šiame metode naudojamas objektyvaus svėrimo metodas, vadinamas entropijos svoriu, kuomet nustatome svorius pradiniam etape. Šis metodas skirtas nustatyti rodiklių svorį objektyviomis sąlygomis, atsižvelgiant į jų tinkamumą. Tai gali atspindėti paslėptą duomenų informaciją ir pagerinti rodiklių skiriamąją gebą ir įvairovę, kad būtų išvengta analizės sunkumų dėl nedidelių pasirinktų rodiklių skirtumų, atspindinčių įvairių rūšių informaciją.

Naudojant tą patį duomenų rinkinį ir tinklo topologiją atliekame skaičiavimus pritaikius membraninius skaičiavimus ir anksčiau aptartą kelio tarp taškų metriką.



**3.5 pav.** Iteracijų skaičių po kurių didžiausias feromono kiekis nusėda

**Fig. 3.5.** The number of iterations after which the pheromone settles

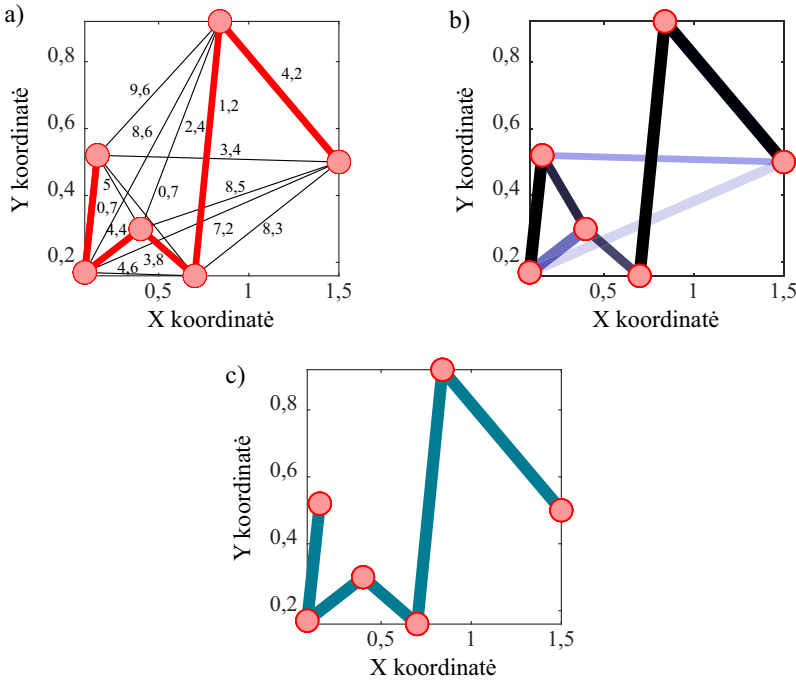
3.5 paveiksle matome, kad jau po apytiksliai pirmų 10 iteracijų feromonas nusėda ant geriausio kelio, o kituose keliuose jis išgaruoja. Atlikus šį eksperimentą patvirtinimas metodo tikslumas ir greitaveika. Kompiuterinio eksperimento rezultatai pateikti 3.1 lentelėje.

**3.1 lentelė.** DI maršruto paieškos trukmė, iteracijų skaičius

**Table 3.1.** Average duration for path searching, iterations

Ekspimento numeris	MMSS	MMSPS
1–5	29,6	27,6
6–10	31,4	19,8
11–15	38,2	27,4
16–20	31,0	11,8
21–25	31,0	22,0
26–30	39,6	26,8
31–35	25,8	13,4
36–40	41,4	37,6
41–45	14,6	12,4
46–50	21,8	19,6
<b>Vidurkis</b>	<b>30</b>	<b>22</b>
<b>Standartinis nuokrypis</b>	<b>8,77</b>	<b>8,22</b>

Imitacinis eksperimentas parodė (3.6 pav.), kad vidutiniškai po 22 iteracijų feromonas nusėda geriausiame kelyje (išgaruoja ant kitų kelių), ir tai yra 25% greičiau nei skruzdžių kolonijų sistemos rezultatas. Šis eksperimentas patvirtina metodo svarbą greitaveikos atžvilgiu.



**3.6 pav.** Geriausia (apskaičiuota) trumpiausio kelio  $d^*$  vertė po 22 iteracijų: a) siekiama (tikroji) trumpiausio kelio  $d^\diamond$  vertė; b) geriausia (apskaičiuota) MMSS trumpiausio kelio  $d^*$  vertė; c) geriausia (apskaičiuota) MMSPS trumpiausio kelio  $d^*$  vertė

**Fig. 3.6.** Best (calculated) shortest path  $d^*$  value after 22 iterations: a) known shortest path  $d^\diamond$ ; b) best (calculated) shortest path  $d^*$  value by MMAS; c) best (calculated) shortest path  $d^*$  value by MMAPS

Atliktų eksperimentų skaičius leido fiksuoti eksperimentų statistiką: sprendinio vidurkį ir standartinį nuokrypį. Parametrai buvo parenkami taip, kad bent vienas sprendinys atitiktų trumpiausią kelią. Buvo sudarytas toks uždavinys, kad būtų žinomas trumpiausias kelias esant duotam mazgų skaičiui. Šio uždavinio trumpiausio kelio ilgio priklausomybė nuo mazgų skaičiaus galima aproksimuoti tiese. Stebint sprendinių vidurkį, galima spręsti apie sistemos parametrų tinkamumą ir atitinkamai juos keisti.

Kitas svarbus sistemos veikimo stebėjimo rodiklis yra standartinis nuokrypis. Standartinio sprendinių nuokrypio stebėjimas leidžia laiku keisti parametrus. Staugus standartinio nuokrypio padidėjimas esant šešių tinklo mazgų optimizavimo uždaviniui, rodė nepakankamą skruzdžių skaičių. Padidinus skruzdžių skaičių nuo 9 iki 16, standartinis nuokrypis sumažėjo. Standartinis nuokrypis gali būti naudo-

jamas sprendinio kokybei įvertinti – kai nuokrypis tampa lygiu nuliui, stochastinis algoritmas tampa determinuotu. Standartinio nuokrypio vertė gali būti maža dėl:

- Uždaviniui taikyta per didelė skruzdžių kolonija – sprendinių vidurkis labai arti arba sutampa su geriausiu sprendiniu.
- Neteisingai parinkti skruzdžių kolonijos parametrai – kolonija dažnai užstringa tame pačiame (lokalaus minimumo) sprendinyje. Didėjanti standartinio nuokrypio vertė rodo algoritmo efektyvumo mažėjimą – sprendinio paieška tampa visiškai atsitiktinė, t. y. algoritmo sprendinius mažai ar visai neįtakoja euristinė ir feromono informacija.
- Sprendžiamam uždaviniui taikyta per maža skruzdžių kolonija – sprendinių vidurkis toli nuo geriausio sprendinio.
- Parinkti prasti skruzdžių kolonijos parametrai – kolonija nesugeba ieškoti greta geresnių sprendinių.

Optimizavimo algoritmai negarantuoja geriausio sprendinio radimo (didelės apimties uždaviniui geriausias sprendinys nebus rastas), bet rasti sprendiniai bus arti geriausio ir algoritmo tikslas – tinklo balansavimas, bus pasiektas.

Siekiant, kad siūlomas metodas tinkamai veiktų ir būtų galima įvertinti, kaip perduoti duomenis, reikalingas kokybinių parametrų stebėjimas.

Pagrindinis šio eksperimento tikslas – patikrinti modeliuojamos sistemos veikimą siekiant sumažinti energijos sąnaudas ir prailginant tinklo eksploatavimo laiką. Tai galime pasiekti subalansavus tinklą. Toliau pateikiami kokybiniai parametrai stebimi šiame darbe:

1. Duomenų perdavimo vėlinimas – parametras parodo kiek laiko duomenys yra perduodami iš duomenų siuntėjo į orkestratoriaus mazgą.
2. Tinklo veikimo laikas – galima apibrėžti kaip laiką nuo sistemos veikimo pradžios iki kol pirmasis jutiklis nebegali perduoti informacijos savo kaimynams.
3. Energijos sunaudojimas – nurodo mazgo, sunaudotos energijos kiekį perduodant duomenų paketą gavėjui.
4. Standartinis energijos nuokrypis – parodo visų mazgų vidutinę energijos lygio dispersiją.

Duomenų perdavimo vėlinimas skaičiuojamas, kai feromonas nusistovi geriausiame kelyje pagal MMSS ir MMSPS algoritmus. Jis apskaičiuojamas susumuojant visus tarpinius kelius maršrute pagal šią lygtį:

$$T = \sum_{k=1}^n t_{ij}^{(k)} \quad (3.2)$$

apskaičiuojami MMSS ir MMSPS algoritmų kokybiniai parametrai  $T_1$  ir  $T_2$ .

Pasiūlytas naujas duomenų valdymo algoritmas raskamas geriausią maršrutą naudojant MMSPS algoritmą sumažina viso kelio delką vidutiniškai 27 %. Eksperimento rezultatai pateikiami 3.2 lentelėje.

**3.2 lentelė.** Duomenų maršruto delsa, ms

**Table 3.2.** Data path delay, ms

<b>Eksperimento numeris</b>	<b>MMSS</b>	<b>MMSPS</b>
1–5	22,30	17,14
6–10	27,72	17,56
11–15	21,62	17,26
16–20	25,24	16,90
21–25	25,36	20,84
26–30	25,18	16,56
31–35	25,24	15,84
36–40	31,92	25,28
41–45	23,56	17,02
46–50	23,48	19,12
<b>Vidurkis</b>	<b>25,16</b>	<b>18,15</b>

DI mazgų energijos suvartojimas daugiausia susijęs su duomenų perdavimo ir priėmimo sistemos aktyviu režimu. Tiesą sakant, miego ir tuščiosios eigos režimuose sunaudojama labai mažai energijos, palyginti su perdavimo ir priėmimo režimais. Tarkime jutiklio mazgas  $k$  turi  $N$  bitų paketų, kuriuos reikia perduoti ar priimti aktyviu režimu (per laiko vienetą). Atitinkamai, visas  $k$  sunaudotas energijos kiekis gali būti apskaičiuojamas taip:

$$E(k) = E^{\text{Tx}}(k) + E^{\text{Rx}}(k). \quad (3.3)$$

Viso tinklo veikimo laikas jutiklių tinkle priklauso nuo daugelio veiksnių, įskaitant baterijų talpą, užklausų kiekį, pranešimų dydį ir agregavimo schemas. Todėl DI turėtų parinkti tinkamus duomenų maršrutus, kad pailgėtų jutiklių tinklo eksploatavimo laikas. Atitinkamai buvo įvertintas tarnavimo laikas remiantis energijos suvartojimo modeliu. Taigi tinklo eksploatavimo laikas deklaruojamas kaip laikas, praleistas nuo sistemos veikimo pradžios iki kol pirmasis jutiklis ar jo dalis nebegali perduoti informacijos savo kaimynams, nes jis išsekvojo savo energiją. Todėl jutiklio mazgo eksploatavimo laikas –  $T_{\text{EL}}$ , pagrįstas lygtimi, apskaičiuojamas taip:

$$T_{\text{EL}} = \frac{E_{\text{init}}}{EnC_{\text{Pkt}}(k)}, \quad (3.4)$$

čia  $E_{\text{init}}$  – pradinė jutiklio mazgo energija,  $En$  – energijos sąnaudos mazge  $k$ ,  $C_{\text{Pkt}}$  – paketo dydis. Tai reiškia, kad norint maksimaliai padidinti viso tinklo eksp-



loatavimo laiką reikia optimizuoti mazgo energijos sąnaudas, nes šis mazgas daro tiesioginį poveikį visam DI tinklo gyvavimui jo veikimo laikotarpyje. Šią problemą gali apibūdinti ši mažinimo funkcija:

$$T_{(EL)}^{\star}(k) = \min(\max(EnC_{Pkt}(k))) . \quad (3.5)$$

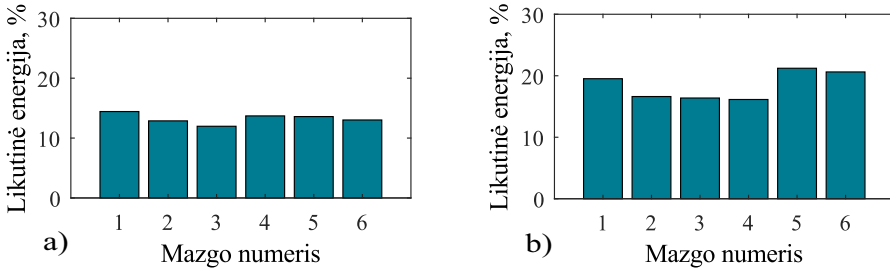
Ši lygtis rodo, kad maksimalus tinklo eksploatavimo laikas yra funkcija, kuria siekiama sumažinti energijos suvartojimo jutiklių mazgų energijos išsekvojimo greitį. Todėl siūlomas sprendimas, kuris užkerta kelią greitai išsikrauti vienam mazgui, sunaudojančiam daugiausia energijos, subalansuodami energijos suvartojimą DI viduje, kad paketai būtų efektyviai nukreipiami kitu keliu. Tiriama tinklo gyvavimo laikas pateikiamas 3.3 lentelėje.

**3.3 lentelė.** DI tinklo gyvavimo trukmė, s

**Table 3.3.** IoT network lifetime, s

Eksperimento numeris	MMSS	MMSPS
1–5	1232	1721
6–10	1416	1544
11–15	1225	1468
16–20	1142	1486
21–25	1132	1328
26–30	1155	1449
31–35	1211	1612
36–40	1178	1481
41–45	1274	1423
46–50	1316	1748
<b>Vidurkis</b>	<b>1228</b>	<b>1526</b>

Taikant siūlomą daiktų interneto savitvarkos metodą, kai šaltinis perduos duomenis į kitą siųstuvą per duotos sesijos laikotarpį, siųstuvą pradės siųsti užklausą ir atnaujinti informaciją apie dabartinius naudojamus maršrutus ir patikrinti, ar nėra kitų geresnių maršrutų. Taip pat periodiškai tikrina DI pokyčius, atsirandančius dėl nutrūkusios jungties – dėl jutiklio mazgų judėjimo arba jutiklio mazgų energijos išsekvojimo turimame kelyje. Atnaujina feromono informaciją, kad skruzdės galėtų sekti naujus duomenų perdavimo maršrutus, paskleidžiant feromono vertę paskleidimo pranešimu, per kaimynus ir tikrindamos energijos lygį, kad apskaičiuotų kelio kokybės funkciją alternatyviais maršrutais. Tai padeda išvengti duomenų paketo maršruto nukreipimo visą laiką tuo pačiu keliu ir jutiklio mazgo išsikrovimą labai greitai, todėl pailgės tinklo eksploatavimo laikas. Taigi šis metodas padidina galimybę pasirinkti geresnį maršrutą iš keleto kelių, kuriais galima perduoti paketus gavėjui.



3.7 pav. Likutinės energijos mazguose palyginimas: a) MMSS; b) MMSPS

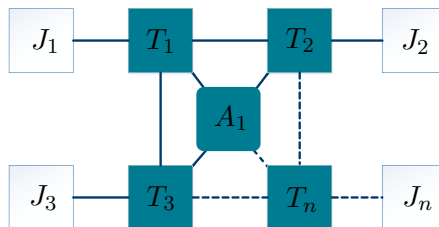
Fig. 3.7. The distribution of residual energy: a) MMSS; b) MMSPS

Veikiant DI tinklui, jutiklių mazgo likutinė energija kinta. Čia MMSPS algoritmo energijos sąnaudos yra palyginti nedidelės, o likutinė energija yra santykinai didelė, daugiausia dėl to, kad pasirenkamas tinkamas membraninių skaičiavimų algoritmas mazgų su keliais žingsniais duomenims perduoti. Tai dar labiau optimizuoja maršruto sudarymo procesą, kad būtų išlaikytas viso DI tinklo energijos balansas. Palyginamieji rezultatai pateikiami 3.7 paveiksle.

### 3.3. Daiktų interneto savitvarkos ribinės sąlygos įgyvendinimo tyrimas

Siekiant nustatyti ribines sąlygas ir patikrinti 2.2 poskyryje sukurtą teorinį daiktų interneto informacijos skirstymo ir valdymo savitvarkos algoritmą, atlikti imitaciniai eksperimentiniai tyrimai.

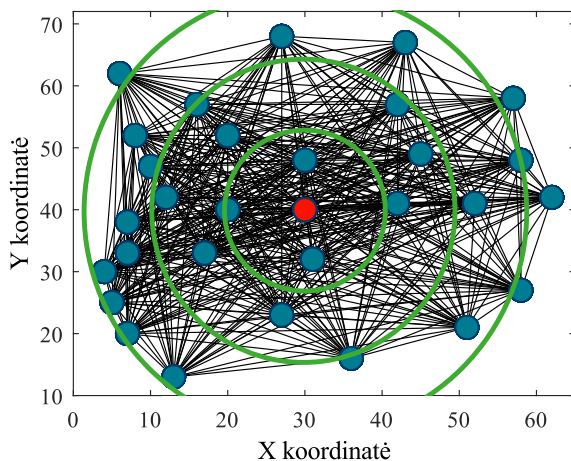
Daiktų interneto savitvarkos įgyvendinimo ribinėmis sąlygomis tyrimas buvo vykdomas 3.8 paveiksle pavaizduotoje DI tinklo topologijos schemoje kurią sudarė jutiklių ( $J_1$ – $J_n$ ) ir tarpiniai mazgai ( $T_1$ – $T_n$ ) bei vienas orkestratoriaus mazgas  $A_1$ .



3.8 pav. Tinklo topologija DI ribinėmis sąlygomis

Fig. 3.8. Network topology of IoT boundary conditions

Testavimo eksperimentą sudaro trys dalys. Pirmojo eksperimento metu tinkle ieškoma ribinių mazgų ir lyginamas šių mazgų kiekis tinkle, antruoju etapu vykdomas tinklo gyvavimo eksperimentas su 100 mazgų taikant MMSS ir MMSPS algoritmus, trečiuoju lyginamas DI tinklo gyvavimo laikas priklausomai nuo tinkle esančių mazgų skaičiaus. Eksperimentų metu iš jutiklių  $J_n$  siunčiami duomenis į orkestratoriaus mazgą  $A_1$  per tarpinius mazgus  $T_n$ . Pirmasis eksperimentas atliktas su 30 mazgų tinklu pavaizduotu 3.9 paveiksle.



**3.9 pav.** 30 mazgų DI tinklo topologija

**Fig. 3.9.** IoT network topology of 30 nodes

Norėdami testuoti šį procesą, turime surasti du ribinius jutiklius esančius DI tinklo kraštuose. Iš pradžių mes imame keturis taškus tinklo ribinėje plokštumoje  $R_u, R_d, R_l, R_r$ . Atitinkamai šių taškų koordinatės plokštumoje yra  $-(P^T/2, 0)$ ,  $(P^T/2, 0)$ ,  $(0, Q^T/2)$ ,  $(P^T, Q^T/2)$ . Taigi šie ribiniai mazgai yra mažiausiu Euklido atstumu nuo ribinių taškų  $R_u, R_d, R_l, R_r$ . Iš šių taškų pasirenkame keturis ribinius mazgus  $BN_u, BN_d, BN_l, BN_r$  ir tikriname ar jie turi ryšius su savo kaimynais. Jeigu šie mazgai neturi ryšio su savo kaimynais, pertvarkome DI tinklo grafą laikantis šių sąlygų:

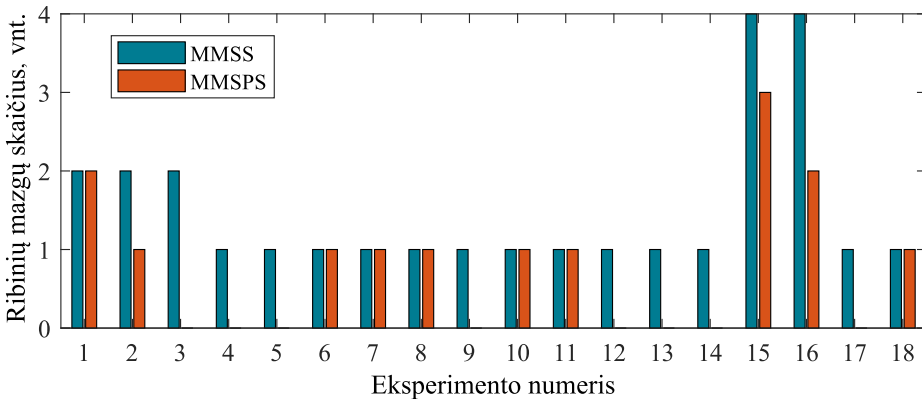
- Kaimyninis įtaisas turi turėti pilnų ryšių grafo šakas su savo kaimynais.
- Kaimyninio jutiklio likutinės energijos lygis turi būti didesnis nei nustatytas slenkščio lygis.

Grafo atnaujinimo procesas tarp ribinių įtaisų ir DI tinklo prasideda, kai tik gauname ribinius įrenginius. Apibūdiname dviejų tipų įrenginius  $JUT_{Pir}$  ir  $JUT_{Alt}$ .

$JUT_{\text{Pir}}$  įrenginiai yra pirminiame kelyje, o  $JUT_{\text{Alt}}$  įrenginiai yra alternatyviame kelyje. Kaimyninio įrenginio paieška remiasi šia lygtimi:

$$DF = \sqrt{(a_{ch} - a_i)^2 + (b_{ch} - b_i)^2}, \quad (3.6)$$

čia  $DF_j$  – atstumas tarp jutiklio ir orkestratoriaus;  $(a_{ch}, b_{ch})$  – orkestratoriaus koordinatės;  $(a_i, b_i)$  – jutiklio koordinatės. Šiame kaimynų paieškos algoritme yra naudojami įrenginiai, kurie yra budėjimo režime. Miego režime esantys įrenginiai neliečiami. Suradus artimiausią tinkamą kaimyną duomenų perdavimui yra taikomas pasiūlytas MMSPS algoritmas geriausio kelio paieškai ir šio algoritmo pagalba yra randamas pagrindinis ir alternatyvieji keliai. Duomenys yra perduodami pagrindiniu keliu, kol kuris nors kelyje esantis DI tinklo mazgas pasieks ribinį likutinės energijos lygį. Pasiekus šį lygį yra naudojamas surastas alternatyvus kelias, tol kol nutrūks visi galimi ryšiai, tuomet bus kartojamas šiame poskyryje aprašytas kaimyno paieškos algoritmas, kadangi jutiklinis mazgas taps ribiniu.

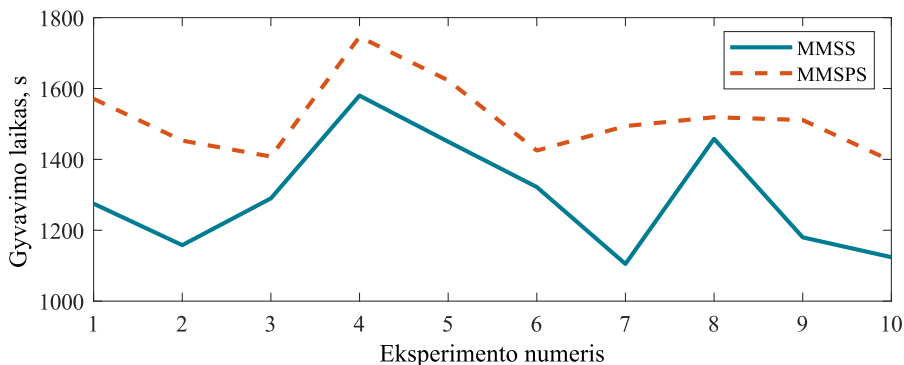


**3.10 pav.** Ribinių DI mazgų skaičius vykdant kaimynų paieškos algoritmą

**Fig. 3.10.** Number of Boundary IoT devices after using a neighbor search algorithm

3.10 paveiksle pateikiamas ribinių mazgų kiekis vykdant eksperimentus. Rezultatai rodo, kad vykdant MMSPS algoritmą, kaimynų paieškos algoritmas naudojamas rečiau, nes išsikrovusių mazgų skaičius tinkle yra mažesnis dėl tinklo balansavimo ir alternatyvių kelių naudojimo.

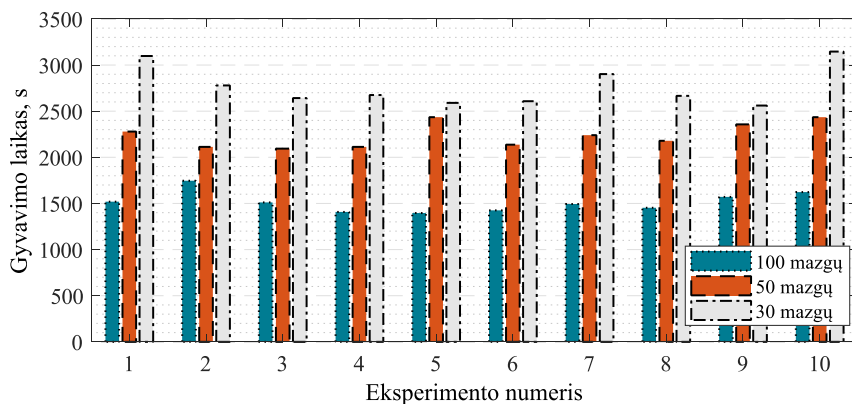
Gauti antrojo eksperimento rezultatai vertinant 100 mazgų tinklo gyvavimo laiką, kol išsikrauna pirmasis mazgas, pavaizduoti 3.11 paveiksle.



**3.11 pav.** 100 mazgų tinklo gyvavimo laikas

**Fig. 3.11.** Network lifetime of 100 nodes

Rezultatai patvirtina, kad nepriklausomai nuo mazgų skaičiaus MMSPS algoritmu vykdant geriausio kelio paiešką, tinklas gyvuoja ilgiau ir tai dar kartą patvirtina balansavimo sąlygą.



**3.12 pav.** DI tinklo gyvavimo trukmė priklausomai nuo mazgų skaičiaus tinkle

**Fig. 3.12.** Network lifetime depending on the number of nodes

Trečiuoju eksperimentu lyginame tinklo gyvavimo trukmę sekundėmis, priklausomai nuo DI tinkle esančių mazgų skaičiaus. Pagal 3.12 paveiksle pateiktus rezultatus matome, kad didinant mazgų skaičių DI tinklo gyvavimo laikas neproporcingai mažėja, nes perduodant duomenis iš jautiklių reikia naudoti daugiau šuolių per tarpinius mazgus, taip juos iškraunant.

Vykdamt eksperimentus buvo atsižvelgta į tai, kad internetinėje erdvėje dauguma įrenginių sąveikauja su fiziniais objektais (stebėjimas, matavimas ir kt.). Šie mazgai renka fizinius duomenis. DI mazgai neatpažįsta visuotinės maršruto informacijos ir dėl išteklių apribojimo sukelia nereikalingus energijos išteklių nuostolius. Atliktuose eksperimentuose naudojamas skruzdžių algoritmas, grįstas visuotiniu informacijos skirstymu, kad intelektualioms skruzdėms ieškant būtų galima rasti optimalų maršrutą. Eksperimentai rodo, kad šiame darbe priimta DI savitvarkos strategija turi mažesnę uždelsimą ir mažesnę energijos suvartojimą nei lyginamasis metodas, be to, jos našumas yra geresnis prailginant tinklo eksploatavimo laiką ir skaičiavimo efektyvumą. MMSPS algoritmas gali pagerinti daiktų interneto veikimą ir subalansuoti energijos suvartojimą ir kokybinius reikalavimus. Skirtingiems daiktų interneto poreikiams šį algoritmą galima lanksčiai patobulinti ir optimizuoti atsižvelgiant į įvairius išteklius.

### 3.4. Trečiojo skyriaus išvados

1. MMSPS algoritmo pagrindu grįsta nauja daiktų interneto savitvarkos (DIS) metodika įgyvendinta programoje MATLAB™. Gauti imitacinių eksperimentinių tyrimų rezultatai įrodo šios metodikos taikymo tikslingumą.
2. DIS metodikos išsamūs imitacinių eksperimentų rezultatai rodo, kad 25 % greičiau pavyko rasti optimalų informacijos perdavimo kelią nepriklausomai nuo keičiamų parametrų: a) DI įrenginių skaičiaus; b) DI tinklo ryšių parametrų; c) MMSPS algoritmo nustatomus parametrų; d) MMSPS algoritmo atsitiktinai parenkamus parametrų.
3. Tinklo ribinėmis sąlygomis atliktų DI imitacinių tyrimų rezultatai pagrindžia, kad DIS metodikos taikymas lyginant su MMSS 15 % sumažina DI įrenginių tinklo energijos sąnaudas ir taip prailgina mobilaus tinklo gyvavimo laiką.

---

## Bendrosios išvados

1. Tikslo funkcijos optimizavimo uždaviniams spręsti sukurtas naujas max-min skruzdžių P sistemos (MMSPS) algoritmas, grindžiamas membraniniais skaičiavimais.
2. Max-min skruzdžių P sistemos taikymas daiktų interneto duomenims apdoroti, esant sudėtingiems sprendiniams, randa sprendinį 25 % greičiau nei max-min skruzdžių sistema (MMSS).
3. Max-min skruzdžių P sistemos algoritmo pagrindu sukurta nauja daiktų interneto savitvarkos (DIS) metodika, efektyviai automatizuoja DI informacijos skirstymo ir valdymo procesus.
4. DIS metodikos išsamūs imitacinių eksperimentų rezultatai rodo, kad pavyko rasti optimalų informacijos perdavimo kelią nepriklausomai nuo keičiamų parametrų: a) DI įrenginių skaičiaus; b) DI tinklo ryšių parametrų; c) MMSPS algoritmo nustatomų parametrų; d) MMSPS algoritmo atsitiktinai parenkamų parametrų.
5. Siūloma DIS metodika ribinėmis sąlygomis subalansuoja tinklą ir lyginant su MMSS 15 % sumažina DI įrenginių tinklo energijos sąnaudas taip prailgindama jutiklinio tinklo gyvavimo laiką.





---

## Literatūra ir šaltiniai

Abowd, G. D.; Dey, A. K.; Brown, P. J.; Davies, N.; Smith, M.; Steggles, P. 1999. Towards a better understanding of context and context-awareness, in *International Symposium on Handheld and Ubiquitous Computing: Handheld and Ubiquitous Computing*, Springer Berlin Heidelberg, 304–307. ISBN 978-3-540-48157-7. [see p.]

Alberts, B.; Johnson, A.; Lewis, J.; *et al.* 2002. *Molecular Biology of the Cell*. New York: Garland Science. [see p.]

Ardelean, I. I.; Cavaliere, M. 2003. Modelling biological processes by using a probabilistic p system software, *Natural Computing* 2(2): 173–197. ISSN 1567-7818. [see p.]

Astudillo, G.; Kadoch, M. 2019. Directional graph-based energy model for iot wireless relay systems, in *2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, 251–258. [see p.]

Atakan, B.; Akan, O. B. 2007. Biologically-inspired spectrum sharing in cognitive radio networks, *2007 IEEE Wireless Communications & Networking Conference, Vols 1-9* 43–48. ISSN 1525-3511. [see p.]

Atzori, L.; Iera, A.; Morabito, G. 2010. The internet of things: A survey, *Computer Networks* 54(15): 2787 – 2805. ISSN 1389-1286. [see p.]

Atzori, L.; Iera, A.; Morabito, G.; Nitti, M. 2012. The social internet of things – when social networks meet the internet of things: Concept, architecture and network characterization, *Computer Networks* 56(16): 3594–3608. ISSN 1389-1286. [see p.]

Banzhaf, W. 2009. *Self-organizing Systems* *Self-organizing systems*. New York, NY: Springer New York. 8040–8050 p. ISBN 978-0-387-30440-3. [see p.]

- Beale, R.; Jackson, T. 1990. *Neural Computing - An Introduction*. Institute of Physics Publishing. 240 p. ISBN 0852742622. [see p.]
- Bello, O.; Zeadally, S. 2013. *Communication Issues in the Internet of Things (IoT)*, London: Springer London, 189–219. ISBN 978-1-4471-5164-7. [see p.]
- Bello, O.; Zeadally, S. 2016. Intelligent device-to-device communication in the internet of things, *Ieee Systems Journal* 10(3): 1172–1182. ISSN 1932-8184. [see p.]
- Bendavid, Y.; Wamba, S. F.; Barjis, J. 2013. Special issue on rfid - towards ubiquitous computing and the web of things: Guest editors introduction, *Journal of Theoretical and Applied Electronic Commerce Research* 8(2): III–XI. ISSN 0718-1876. [see p.]
- Blum, C. 2005. Ant colony optimization: Introduction and recent trends, *Physics of Life Reviews* 2(4): 353 – 373. ISSN 1571-0645. [see p.]
- Cavaliere, M.; Mazza, T.; Sedwards, S. 2014. *Statistical Model Checking of Membrane Systems with Peripheral Proteins: Quantifying the Role of Estrogen in Cellular Mitosis and DNA Damage*, Cham: Springer International Publishing, 43–63. ISBN 978-3-319-03191-0. [see p.]
- Ciobanu, G.; Dumitriu, D.; Huzum, D.; Moruz, G.; Tanas, B. 2003. *Client–Server P Systems in Modeling Molecular Interaction*, Berlin, Heidelberg: Springer Berlin Heidelberg, 203–218. ISBN 978-3-540-36490-0. [see p.]
- Ciobanu, G.; Păun, G.; J., M.; Pérez-Jiménez 2005. *Applications of Membrane Computing (Natural Computing Series)*. Springer-Verlag. ISBN 978-3-540-29937-0. [see p.]
- Das, M. K.; Dai, H. K. 2007. A survey of dna motif finding algorithms, *Bmc Bioinformatics* 8: 13. ISSN 1471-2105. [see p.]
- De Poorter, E.; Moerman, I.; Demeester, P. 2011. Enabling direct connectivity between heterogeneous objects in the internet of things through a network-service-oriented architecture, *EURASIP Journal on Wireless Communications and Networking* 2011(1): 61. ISSN 1687-1499. [see p.]
- Di Caro, G.; Ducatelle, F.; Gambardella, L. M. 2005. Anthocnet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks, *European Transactions on Telecommunications* 16(5): 443–455. [see p.]
- Dorigo, M.; Bonabeau, E.; Theraulaz, G. 2000. Ant algorithms and stigmergy, *Future Generation Computer Systems* 16(8): 851 – 871. ISSN 0167-739X. [see p.]
- Dorigo, M.; Gambardella, L. M. 1996. A study of some properties of ant-q, *Parallel Problem Solving from Nature PPSN IV Lecture Notes in Computer Science* 656–665. [see p.]
- Dorigo, M.; Gambardella, L. M. 1997. Ant colonies for the travelling salesman problem, *Biosystems* 43(2): 73–81. [see p.]
- Dorigo, M.; Stutzle, T. 2003. The ant colony optimization metaheuristic: Algorithms, applications, and advances, *Handbook of Metaheuristics International Series in Operations Research & Management Science* 250–285. [see p.]

- Dressler, F.; Akan, O. B. 2010. A survey on bio-inspired networking, *Computer Networks* 54(6): 881–900. ISSN 1389-1286. [see p.]
- Ejaz, M.; Kumar, T.; Ylianttila, M.; Harjula, E. 2020. Performance and efficiency optimization of multi-layer iot edge architecture, in *2020 2nd 6G Wireless Summit (6G SUMMIT)*, 104–117. [see p.]
- Euler, L. 1761. *Lettres a Une Princesse d'Allemagne*. 102–108 p. [see p.]
- Evans, D. 2013. The internet of things: How the next evolution of the internet, *Cisco Internet Business Solutions Group* 8. [see p.]
- Franco, G.; Manca, V. 2004. *A Membrane System for the Leukocyte Selective Recruitment*, Berlin, Heidelberg: Springer Berlin Heidelberg, 181–190. ISBN 978-3-540-24619-0. [see p.]
- Gao, S. C.; Wang, Y. R.; Cheng, J. J.; Inazumi, Y.; Tang, Z. 2016. Ant colony optimization with clustering for solving the dynamic location routing problem, *Applied Mathematics and Computation* 285: 149–173. ISSN 0096-3003. [see p.]
- Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. 2013. Internet of things (iot): A vision, architectural elements, and future directions, *Future Generation Computer Systems* 29(7): 1645 – 1660. ISSN 0167-739X. [see p.]
- Guo, P.; Zhang, M.; Chen, J. 2017. A family of ant colony p systems, in *Bio-inspired Computing: Theories and Applications*, 175–193. ISBN 978-981-10-7178-2. [see p.]
- Hunter, L. 1993. *Molecular biology for computer scientists*, American Association for Artificial Intelligence, 1–46. ISBN 0-262-58115-9. [see p.]
- Ioan, A. 2002. Molecular biology of bacteria and its relevance for p systems, in *International Workshop, WMC-CdeA*, 1–18. [see p.]
- ITU 2012. *Overview of the Internet of Things*. [see p.]
- Kephart, J. O.; Chess, D. M. 2003. The vision of autonomic computing, *Computer* 36(1): 41–50. ISSN 0018-9162. [see p.]
- Klugl, F. 2001. Swarm intelligence: From natural to artificial systems, *Jasss-the Journal of Artificial Societies and Social Simulation* 4(1): U153–U156. ISSN 1460-7425. [see p.]
- Kreger, H. 2001. *Web Services Conceptual Architecture*. Available online at: [http://users.cs.uoi.gr/~pitoura/courses/ds04\\_gr/webt.pdf](http://users.cs.uoi.gr/~pitoura/courses/ds04_gr/webt.pdf). [see p.]
- Kulkarni, A.; Sathe, S. 2014. Healthcare applications of the internet of things: A review, in *International Journal of Computer Science and Information Technologies*, 4. [see p.]
- Kumar, D.; Verma, N. K.; Singh, N. 2018. A review paper on deducting database in membrane computing, *Journal of Statistics and Management Systems* 21(4): 667–673. [see p.]
- Laptik, R. 2009. *Skruzdžių kolonijų technologijos vaizdams apdoroti: daktaro disertacija*. Vilniaus Gedimino technikos universitetas. Vilnius: Technika. 115 p. ISBN 978-9955-28-501-4. [see p.]

- Larsen, M. P.; Eisenberg, M. S.; Cummins, R. O.; Hallstrom, A. P. 1993. Predicting survival from out-of-hospital cardiac arrest: A graphic model, *Annals of Emergency Medicine* 22(11): 1652 – 1658. ISSN 0196-0644. [see p.]
- Liotta, A. 2013. *Why the Internet Needs Cognitive Protocols*. Available online at: <https://spectrum.ieee.org/computing/networks/why-the-internet-needs-cognitive-protocols>. [see p.]
- Liton, M. 2018. *How Much Data Comes From IoT Devices*. Available online at: <https://www.sumologic.com/blog/iot-data-volume/>. [see p.]
- Lott, C.; Teneketzi, D. 2006. Stochastic routing in ad-hoc networks, *Ieee Transactions on Automatic Control* 51(1): 52–70. ISSN 0018-9286. [see p.]
- Machado, K.; Rosario, D.; Cerqueira, E.; Loureiro, A. A. F.; Neto, A.; de Souza, J. N. 2013. A routing protocol based on energy and link quality for internet of things applications, *Sensors* 13(2): 1942–1964. ISSN 1424-8220. [see p.]
- Manca, V. 2013. Infobiotics, *Emergence, Complexity and Computation* 368. [see p.]
- Mirco Musolesi, C. M. 2006. Evaluating context information predictability for autonomous communication, in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, 495–499. [see p.]
- Mirjalili, S. 2018. Ant colony optimisation, *Studies in Computational Intelligence Evolutionary Algorithms and Neural Networks* 33–42. [see p.]
- Mullen, R.; Monekoso, D.; Barman, S.; Remagnino, P. 2009. A review of ant algorithms, *Expert Systems with Applications* 36(6): 9608 – 9617. ISSN 0957-4174. [see p.]
- Mustafa, G.; Ashraf, R.; Mirza, M.; Jamil, A.; , M. 2018. A review of data security and cryptographic techniques in iot based devices, in *The 2nd International Conference on Future Networks and Distributed Systems*, 1–9. [see p.]
- Nasser, N.; Karim, L.; Ali, A.; Anan, M.; Khelifi, N.; Ieee 2017. *Routing in the Internet of Things*, New York: Ieee. IEEE Global Communications Conference. ISBN 978-1-5090-5019-2. [see p.]
- Nishida, T. Y. 2002. Simulations of photosynthesis by a k-subset transforming system with membrane, *Fundamenta Informaticae* 49(1): 249–259. ISSN 0169-2968. [see p.]
- Pablo López-Matencio, E. C.-M., Javier Vales-Alonso 2017. Ant: Agent stigmergy-based iot-network for enhanced tourist mobility, *Mobile Information Systems* 2017: 15. [see p.]
- Paun, A.; Perez-Jimenez, M. J.; Romero-Campero, F. J. 2006. *Modeling Signal Transduction Using P Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg, 100–122. ISBN 978-3-540-69090-0. [see p.]
- Pratt, S.; Mallon, E.; Sumpter, D.; Franks, N. 2002. Quorum sensing, recruitment, and collective decision-making during colony emigration by the ant *leptothorax albipennis*, *Behavioral Ecology and Sociobiology* 52: 117–127. [see p.]

- Păun, G. 2000. Computing with membranes, *Journal of Computer and System Sciences* 61(1): 108 – 143. ISSN 0022-0000. [see p.]
- Păun, G.; Rozenberg, G.; Salomaa, A. 2010. *The Oxford Handbook of Membrane Computing*. Oxford University Press, Inc. 696 p. ISBN 0199556679, 9780199556670. [see p.]
- Roberto Minerva, D. R., Abyi Biru 2015. *Towards a Definition of the Internet of Things*. Available online at: [https://iot.ieee.org/images/files/pdf/IEEE\\_IoT\\_Towards\\_Definition\\_Internet\\_of\\_Things\\_Revision1\\_27MAY15.pdf](https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf). [see p.]
- Roman, R.; Najera, P.; Lopez, J. 2011. Securing the internet of things, *Computer* 44(9): 51–58. ISSN 0018-9162. [see p.]
- Rubin, H. 1999. The Touchstone of Life: Molecular Information, Cell Communication, and the Foundations of Life, *Annals of Internal Medicine* 131(1): 79–79. ISSN 0003-4819. [see p.]
- Salem, A.; Sleit, A. 2018. Analysis of ant colony optimization algorithm solutions for travelling salesman problem, *International Journal of Scientific & Engineering Research*. [see p.]
- Savukynas, R.; Marcinkevičius, V. 2017. Daiktų interneto objektų identifikavimo metodų palyginimas, *Informacijos mokslai* 78: 66. [see p.]
- Serbanati, A.; Medaglia, C. M.; Ceipidor, U. B. 2011. Building blocks of the internet of things: State of the art and beyond, in *Deploying RFID*, ed. by Turcu, C., Rijeka: IntechOpen, Chap. 20. [see p.]
- da Silva, E.; Albini, L. C. P. 2014. Middleware proposals for mobile ad hoc networks, *Journal of Network and Computer Applications* 43: 103 – 120. ISSN 1084-8045. [see p.]
- Stützle, T. 2009. Ant colony optimization, *Lecture Notes in Computer Science Evolutionary Multi-Criterion Optimization 2–2*. [see p.]
- Stützle, T.; Hoos, H. H. 2000. MAX-MIN ant system, *Future Generation Computer Systems* 16(8): 889 – 914. ISSN 0167-739X. [see p.]
- Venn, J. 1880. I. on the diagrammatic and mechanical representation of propositions and reasonings, *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 10(59): 1–18. Available online at: <https://doi.org/10.1080/14786448008626877>. [see p.]
- Wang, K.; Wang, Y.; Sun, Y.; Guo, S.; Wu, J. 2016. Green industrial internet of things architecture: An energy-efficient perspective, *IEEE Communications Magazine* 54(12): 48–54. [see p.]
- Xu, L. D.; He, W.; Li, S. 2014. Internet of things in industries: A survey, *IEEE Transactions on Industrial Informatics* 10(4): 2233–2243. ISSN 1551-3203. [see p.]

Zedadra, O.; Guerrieri, A.; Jouandea, N.; Spezzano, G.; Seridi, H.; Fortino, G. 2018. Swarm intelligence-based algorithms within iot-based systems: A review, *Journal of Parallel and Distributed Computing* 122: 173 – 187. ISSN 0743-7315. [see p.]

Zhang, G.; Pérez-Jiménez, M.; Gheorghe, M. 2017. *Real-life Applications with Membrane Computing*. Springer Berlin Heidelberg. 212 p. [see p.]

Zhang, G. X.; Gheorghe, M.; Pan, L. Q.; Perez-Jimenez, M. J. 2014. Evolutionary membrane computing: A comprehensive survey and new results, *Information Sciences* 279: 528–551. ISSN 0020-0255. [see p.]

Ge-xiang Zhang, J.-x. C., Marian Gheorghe 2010. An approximate algorithm combining p systems and ant colony optimization for traveling salesman problems, in *Eighth Brainstorming Week on Membrane Computing*, 321–340. [see p.]

Zhou, Y. 2009. Runtime analysis of an ant colony optimization algorithm for tsp instances, *IEEE transactions on evolutionary computation* 13(5): 1083–1092. ISSN 1089-778X. [see p.]

---

# Autoriaus mokslinių publikacijų disertacijos tema sąrašas

## **Straipsniai recenzuojamuose mokslo žurnaluose**

Gedminas, A; Duoba, L; Navakauskas, D. 2020. P System Framework for Ant Colony Algorithm in IoT Data Routing, *International Journal of Advanced Computer Science and Applications* 11(4): 28–34. ISSN 2156-5570.

Gedminas, A. 2019. Membraninių skaičiavimų ir daiktų interneto technologijos, *Mokslas, Mokslas – Lietuvos ateitis* 11. ISSN 2029-2341.

Žalnėravičius, R; Gedminas, A; Ruzgas, T; Jagminas, A. 2019. Nanoplatelet MoS<sub>2</sub> arrays decorated with Pt nanoparticles for non-enzymatic detection of hydrogen peroxide, *Journal of Electroanalytical Chemistry* 839: 274–282. ISSN 1572-6657.

## **Straipsniai kituose mokslo leidiniuose**

Abromavičius, V; Gedminas, A; Serackis, A. 2017. Detecting sense of presence changes in EEG spectrum during perception of immersive audiovisual content, in *Proceedings of Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 1–4.

Kulakovskis, D; Sledevič, T; Gedminas, A; Navakauskas, D. 2016. Alternative Implementations of Metabolic P System in FPGA, in *Proceedings of IEEE 4rd Workshop on advances in information, electronic and electrical engineering*, 1–5.





---

# Summary in English

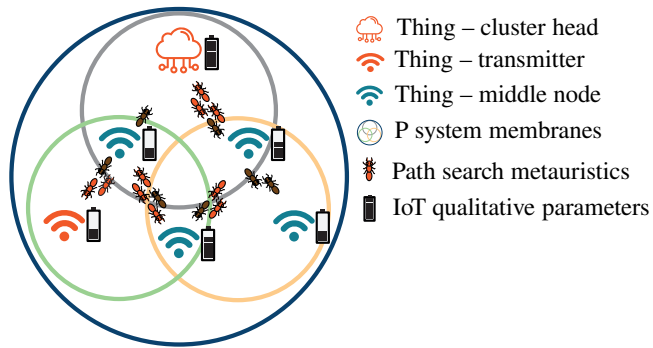
## Introduction

### Problem Formulation

In the theory of membrane computing, the P (Gheorghe Păun) system (Păun *et al.* 2010) takes a central place. P system is a hierarchically arranged set of membranes and its perfect shows in the schemes of Euler (Euler 1761) or Venn (Venn 1880). The formulation of the P system and the maximal parallelism of the performed calculations presuppose the ideal applicability to a large number of devices (usually limited resources) to minimize own resources, while using cooperative principles (usually in highly changing environmental conditions) to achieve goals in common organization and management.

The imperative feature of the Internet of Things (IoT) systems is autonomy. Self-healing is one of the most important components of autonomous systems. Therefore, in order to ensure sustainability, as well as security, it is appropriate to investigate the applicability of evolutionarily tested techniques.

The P systems theory offers only a general set of rules that do not specify how to apply this theory to the Internet of Things. There is no knowledge in the scientific literature which principles of self-organizing in the Internet of Things should be based on P system (Kumar *et al.* 2018). The key challenge is to properly select P system to the IoT technology – which would offer an artificial self-management of the Internet of Things (Fig. S0.1). That is why this work solves the *applied scientific problem* – the rational implementation of the Internet of Things self-organizing based on P system.



**Fig. S0.1.** Idea sketch – IoT self-organization based on P system and metaheuristic algorithm

## Relevance of the Thesis

Currently, the most rapidly evolving technologies are related to the collecting, processing, and analyzing of information. With the increasing use of IoT devices, there is a need for these devices to operate seamlessly and sustainably (Atzori *et al.* 2010). In addition, IoT must provide high-quality and privacy services that will be adaptable to a changing environment. So these systems must have self-organizing principles and this is one of the primary tasks.

Evolutionary or metaheuristic algorithms are often used to develop self-organizing IoT systems (Gubbi *et al.* 2013), because nature has developed tools and processes that are able to innovate and adapt to changing conditions. So nowadays these topics became very relevant: Artificial Life, Computational Synthetic Biology, and Natural Computing, collectively named *Infobiotics* science (Manca 2013).

The theory of P systems is relatively new, because it is been developing only for two decades. An analysis of the literature confirms that P systems are insufficiently researched and not applied to self-organizing IoT problems. The primary application of P systems with metaheuristic algorithms has been found in scientific publications since 2010 (Ge-xiang Zhang, Marian Gheorghe 2010), and the first part of the monograph was published only 3 years ago (Zhang *et al.* 2017).

In order to solve the problem – rational implementation of the Internet of Things self-organizing based on P system *scientific hypothesis* has been raised and proved in the work – by applying a metaheuristic algorithm together with P system, it is possible to ensure better Internet of Things the quality of self-organizing.

## The Object of the Research

Object of work – self-organizing Internet of Things network. The following subjects are related to the research object are studied: possibilities of application of P system and metaheuristic algorithm and their efficiency

## The Aim of the Thesis

The aim of the thesis – to propose an original metaheuristic based P system solution by creating and investigating a system for simulating and testing membrane computing in the Internet of Things.

## The Objectives of the Thesis

In order to solve the stated problem and reach the aim of the thesis the following objectives are formulated:

1. Based on P system theory and insights from world practice, propose original solutions to the selected metaheuristic algorithm to describe P system concepts for faster operation of this algorithm.
2. To adapt developed metaheuristic algorithm P system to the self-organizing Internet of Things in order to improve the qualitative characteristics of speed, energy consumption, etc. criteria.
3. To investigate the developed membrane computation system in the conditions of the Internet of Things with simulation and testing tools.

## Research Methodology

The aim of the dissertation is realistic because it is formed taking into account the existing and widely used Internet of Things technology and actively developing theories of metaheuristic algorithms and membrane computing.

The aim is seeking clear qualitative results. The proposed technological solutions are developed taking into account the currently relevant, practically important, insufficiently scientifically researched tasks, the solution of which may have a positive social and economic effect.

The dissertation follows clear provisions, as undisputed methods are used and the hypotheses are tested by empirical research. The aim is to obtain quantitatively better results than the existing ones according to one or more evaluation criteria (IoT network node energy consumption, data path search time, data packet transmission delay, network lifetime, and residual energy).

The thesis proposes a new methodology of self-organizing Internet of Things, applying knowledge and theories of different fields of science: a) communication systems: Internet of Things (IoT), device-to-device communications (D2D); b) membrane computing – P (Păun) system; c) evolutionary computing: ant colonies optimization and max-min ant system.

The new algorithm of the max-min ant P system (MMAPS) has been implemented, the research of computer modeling of self-optimization of the Internet of Things under normal and boundary conditions has been performed, using MATLAB™ v. R2020a.

Scientific results are evaluated by search of information search, analysis, comparative analysis, systematization, and generalization. Statistical analysis and summarization of the results of experimental research were performed based on the method of experimental research.

### **Scientific Novelty of the Thesis**

1. The first mathematical description of the P system, which is expedient to apply to the self-organizing Internet of Things, has been developed and implemented in software.
2. A new methodology for the distribution and management of Internet of Things information based on the max-min ant P system has been developed. The application of this methodology allows to automate the principles of self-organizing Internet of Things.
3. The implementation self-organizing principles of the Internet of Things has resulted in quantitatively better results according to one or more evaluation criteria (network node energy consumption, data route search time, the data packet transmission delay, network lifetime, and residual energy). It also has a positive socio-economic effect compared to the selected metaheuristic algorithm.

### **Practical Value of the Research Findings**

The developed max-min ants P system can be applied to solve the problem of finding the optimal path of data. The solution to this task becomes more convenient, because the system automatically selects the optimal path of information transmission, taking into account the qualitative characteristics: speed, delay, and energy consumption.

The results of the research were applied in the following projects:

- Research of electronic systems implemented in integrated circuits of field programmable logic arrays based on intelligent methods. Support No. 01.2.1-MITA-K-824 „Innovation vouchers“.
- Development of a modern an ultra-small, high-speed armored door lock monitoring system model. Support No. 01.2.1-MITA-K-824-02 „Innovation vouchers“.

### **The Defended Statements**

1. The developed new max-min ant P system is suitable for application in the Internet of Things self-regulation.
2. Application of max-min ant P system in IoT data routing finds a solution 25% faster than selected max-min ant system (MMAS).
3. The results of simulation studies of the max-min ants P system performed under boundary conditions for complex solutions justify that the application of the proposed self-organizing in the Internet of Things methodology reduces the network energy consumption of IoT devices compared to MMSS by 15% and thus prolongs the life of the mobile network.

### **Approval of the Research Findings**

Five articles are published on the subject of the dissertation: 3 papers were published in scientific journals reviewed in international databases and 2 papers – in international

conference proceedings.

The results of the research described in the Thesis were presented at six scientific conferences in Lithuania and abroad::

- Gedminas, A.; Navakauskas, D. 2019. Preliminary results of Max-Min Ant P system for IoT Data Routing. AIEEE 2019. Riga, Latvia.
- Kulakovskis, D.; Gedminas, A.; Navakauskas, D. 2018. Intravenous Glucose Tolerance Test Implementation in FPGA using Metabolic P System. KTU PhD Week 2018. Kaunas, Lithuania.
- Abromavičius, A.; Gedminas, A. 2017. Investigating Quality of Experience using Spectral EEG Features in Immersive Audiovisual Communications. eStream 2017. Vilnius, Lithuania.
- Kulakovskis, D.; Gedminas, A. 2017. Investigation of metabolic P system FPGA implementation by different methods. Conference of Young Scientists „Science – Future of Lithuania: electronics and electrical engineering“. Vilnius, Lithuania.
- Kulakovskis, D.; Sledevič, T.; Gedminas, A.; Navakauskas, D. 2016. Alternative Implementations of Metabolic P System in FPGA. AIEEE 2016. Vilnius, Lithuania.
- Gedminas, A. 2016. Analysis of research membrane computing technologies in the field of Internet of Things. Conference of Young Scientists „Science – Future of Lithuania: electronics and electrical engineering“. Vilnius, Lithuania.

## Structure of the Dissertation

The scope of work is pages, which contain: 51 equations, 34 pictures, 8 tables, 5 algorithms. The dissertation was based on 74 literature from other authors.

## 1. Internet of Things and Membrane Computing Technologies Review

The Internet of Things (IoT) is a critical part of current information technology (Atzori *et al.* 2010). The International Telecommunication Union has reported (ITU 2012) that the IoT is changing content networking hence being diverse. Also, it has been indicated that the connection has to be safe (Mustafa *et al.* 2018) and always present: anytime, anywhere and among any devices. Consequently, if the sensor nodes are dispersed on edge and find essential to interact with the other nodes outside, such nodes are supposed to have reasonable routes for the interaction. Limited resources such as the computation speed, available amount of memory, remaining battery power or channel bandwidth, to name a few, of the device when designing the data routes must be considered.

The Ant System (AS) has been considered as a successful evolutionary model of swarm intelligence which has been derived from the common traits of bugs and other creatures. It was developed by Marco Dorigo (Dorigo, Gambardella 1996) to contribute to tackling several combinatorial optimization challenges. The AS imitates the qualities of

real ants which are utilizing pheromone tails to detect their routes to the food sources. The major AS algorithms in research are the following original ant colony structures: Max-Min Ant System (MMAS), Rank-Based Ant System, Hyper-Cube Ant System, and the Ant Colony System (ACS), the last being noted as one of the most robust. The Ant Colony System is successfully applied to different kinds of matters, e.g., traveling salesman problem (TSP) or distribution network. Therefore, the implementation of ACS for routing in the IoT has also been analyzed.

Over the last half a decade, the optimization of the bio-inspired algorithm has become a major issue of concern. The P system inspired by the nature of membrane processes not only simplifies a complex system behavior annotation but also delivers a good balance between performance, flexibility and scalability. According to (Guo *et al.* 2017), membrane computing models also known as the P-computing help to solve different optimization problems.

## 2. Self Organizing Internet of Things Methodology

Bio-inspired algorithms are most suitable for IoT self-organizing. The original ACO algorithm was adapted to solve combinatorial optimization problems, but many modifications of this algorithm are known to solve various other problems. This chapter presents the ideas and mathematical models of the algorithm for solving self-organizing IoT problems. In order to speed up the ACO algorithm, a membrane computing technology was proposed and a new algorithm methodology max-min ant P system (MMAPS) was introduced.

When applied to solve a practical problem, the initial step is usually to generate an artificial ant colony. Afterwards, it enables each ant to create either a partial or full solution (Mirjalili 2018). On the initial state the artificial ants start from transmitter, and then and select the second nodes to arrive until set up a full solution. The ant then releases a pheromone proportionally to the quality level of the solution it has found in the path, and each ant solves the problem by starting a new process until a satisfactory solution is found. ACO algorithm has obtained achievements through its application in the routing research (Gao *et al.* 2016). Basic ACO pseudocode is show in Algorithm 2.1.

### Algorithm 2.1 (ACO algorithm pseudocode)

```

36 Initializing
37 While (Termination conditions not met) do
38   Selection of primary parameters
39   Tour construction (S2.1)
40   Pheromone update (S2.2)
41 End
42 Returning best path

```

While initializing, different nodes,  $m$ , are selected at random, and send the signal between nodes  $i$  and  $j$ . The probability that signal  $k$  is transmitted from node  $i$  to node  $j$ ,

therefore:

$$p_{i,j} = \frac{\tau_{i,j}^{(\alpha)} \eta_{i,j}^{(\beta)}}{\sum \tau_{i,j}^{(\alpha)} \eta_{i,j}^{(\beta)}}, \quad (\text{S2.1})$$

the  $\tau_{i,j}$  – represents the pheromone value on edge  $i, j$ ;  $\alpha$  ir  $\beta$  – are two parameters that control the relative weight of the pheromone trail and heuristic value;  $\eta_{i,j}$  – for the ant  $k$  reachable nodes are accessible between node  $i$  and  $j$ .

After the  $n$  moments, signal  $k$  passes through all the nodes and forms a complete cycle. The quantity of pheromone (information) in all paths should be updated based on the equation:

$$\tau_{i,j} \leftarrow (1 - \rho)\tau_{i,j} + \sum_{k=1}^m \Delta\tau_{i,j}^{(k)}, \quad (\text{S2.2})$$

$m$  – represents number of ants;  $\rho$  – pheromone evaporation rate;  $\Delta\tau_{i,j}^{(k)}$  – value of pheromone settled on edge between  $i, j$ .

In this dissertation Max-Min Ant System mathematical model is used. The Max-Min Ant System (MMAS) was proposed by Stützle and Hoos (Stützle, Hoos 2000). The MMAS is an extension of the ACO with higher performance for many optimization problems. The MMAS has several advantages compared to the ACO. The MMAS investigates the best found paths and allows the ant which has found the best solution or which has been the best, to leave the pheromone. Such strategy means that very quickly all the ants choose only one path which is the best. When all the ants find solutions, the pheromone is renewed by applying the evaporation system to the ant, leaving a new amount of pheromone:

$$\tau_{i,j} \leftarrow [(1 - \rho)\tau_{i,j} + \Delta\tau_{i,j}^{\star}]_{\tau_{lb}}^{\tau_{ub}}. \quad (\text{S2.3})$$

In this formula  $\Delta\tau_{i,j}^{\star} = 1/C^{\star}$  is the best ant that is allowed to leave a pheromone.

$\tau^{\top}$  is the max limit of the pheromone,  $\tau^{\perp}$  is the min limit of the pheromone,  $[x]_{\tau^{\perp}}^{\tau^{\top}}$  is determined by:

$$[x]_{\tau^{\perp}}^{\tau^{\top}} = \begin{cases} \max, & x > \max; \\ \min, & x < \min; \\ x, & \text{otherwise.} \end{cases} \quad (\text{S2.4})$$

Then the ACO has been integrated into the P system as a sub-algorithm which ensures that the P systems have the evolution rules of the membrane computing. Ants evolve independently generations to elect a best one to communicate with the ants in each elementary membrane. Communication rules performs the communication between the ants in the elementary membranes and those in the skin membrane. According to (Ge-xiang Zhang, Marian Gheorghe 2010), a combination of the P systems and ACO is seen to perform better than ACO. Fist IoT network is divided in to elementary membranes then the developed Max-Min Ant P system (MMAPS) algorithm is used.

MMAPS algorithm pseudo code is presented in Algorithm 2.2 where each step is described as follows. The process will stop according to a preset termination condition, such as a certain number of iterations.

## Algorithm 2.2 (MMAPS algorithm pseudocode)

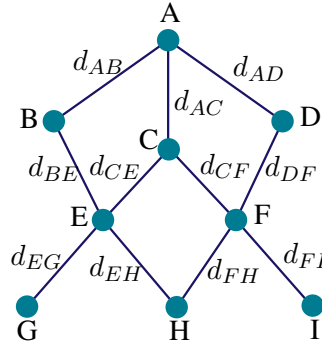
```

1 Begin
2  $t \leftarrow 1$ 
3 While (not termination condition) do.
4   Scatter ants into elementary membranes
5   Determine iterations for each of elementary membranes
6   for  $i = 1, 2, \dots, m$  do
7     Initializing parameters  $t = 0, N_c = 0, \Delta\tau_{ij}(0) = 0$ 
8     for  $i, j = 1, 2, \dots, m$  do
9       the initial value of the pheromone is determined  $\tau_{ij} = \text{const}$ 
10    end
11    for  $N_c < N^\top$  do
12       $N_c \leftarrow N_c + 1$ 
13      for  $k = 1, 2, \dots, m$  do
14        if  $L \neq D$  then
15          Another node is selected according to probability (S2.1)
16          Replace Tabu
17        end
18      end
19      Saving  $p_{ij}$  value
20      Update pheromone according to (S2.3)
21      if  $\Delta\tau_{ij} < \tau^\perp$  then
22         $\Delta\tau_{ij} = \tau^\perp$ 
23      end
24      if  $\Delta\tau_{ij} > \tau^\top$  then
25         $\Delta\tau_{ij} = \tau^\top$ 
26      end
27    end
28     $p_i \leftarrow p_{ij}^*$ 
29  end
30  Form a colony of ants in the skin membrane
31  Perform MMAS (lines 6–29) in the skin membrane
32  Execute global communication
33   $t \leftarrow t + 1$ 
34 end
35 End Begin

```



To specify the MMAPS algorithm the structure of P systems is combined with its evolution rules and the parameterized probabilistic model which is based on the pheromone model of ACO. The objects consist of IoT nodes (or ants) graphs. The tour is constructed by evolution rules, the same which are responsible to select the best ant, and communication rules implemented by using local and global pheromone update rules of MMAS.



**Fig. S2.1.** Ant path selection

The basic MMAS is upgraded to accommodate the specific task of the IoT self-organizing. First and foremost, we have to assume that the devices on the chosen IoT system send and receive different amounts of data, so each iteration must have a different data transfer path. Using the proposed technique, the path between two IoT nodes is evaluated according to its difficultness instead of path length calculated from geographic location of nodes. The path Fig. S2.1 difficultness  $d(k)_{ij}$  is estimated by an aggregation of estimated node energy consumption and link delays:

$$d_{ij} = w_e E_{ij} + w_t t_{ij}. \quad (\text{S2.5})$$

Here  $E_k$  denotes the normalized residual energy of the node  $k$ ,  $t_{ij}$  indicates the normalized  $i$  and  $j$  link delay,  $w_e$  and  $w_t$  denote the weights of corresponding variables and are selected during system training. Accordingly, pheromone updating is adjusted by the formula:

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \frac{F}{P}; \quad (\text{S2.6})$$

$$P = \sum_{k=1}^n d_k. \quad (\text{S2.7})$$

$F$  symbolizes the pheromone intensity, and  $P_{ij}$  denotes the total cost of the link between the  $i$  and  $j$  nodes. This technique also uses an objective weighting method called entropy weight, where the weights of edges are determined at the initial stage. It is intended to determine the weighting of the indicators under objective conditions taking into account their relevance and objectivity.

### 3. An Investigation of Internet of Things Self-Organization Technology

To confirm the quality evaluation criteria measurements a self-organization internet of things imitation and the testing system should be created and researched.

First a virtual 6 node IoT has been simulated with software package MATLAB™. The network topology is shown in Fig. S3.1.

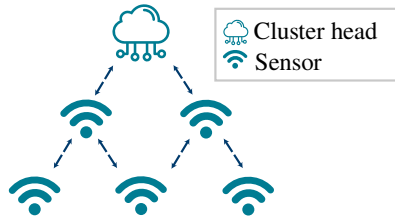


Fig. S3.1. Graph of IoT network topology

The main steps of the experiment are: 1) network initialization; 2) calculation of the best path using MMAPS; 3) corresponding data routing; 4) monitoring of qualitative parameters; 5) update of the node residual energy.

During initialization, one of the first tasks is to make the connections of the existing network Fig. S3.1 First, a graph is produced containing all nodes and interconnections. In the second step, this graph connects the senders and recipients of the information, or in other words, the things. The value between the sensors is path difficulty  $d$  which is calculated by measuring energy consumption values and link delays is shown in Fig. S3.2.

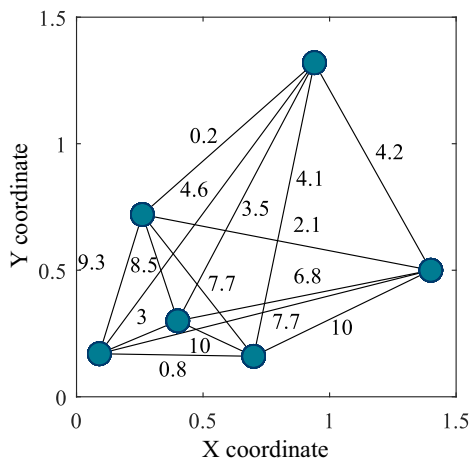
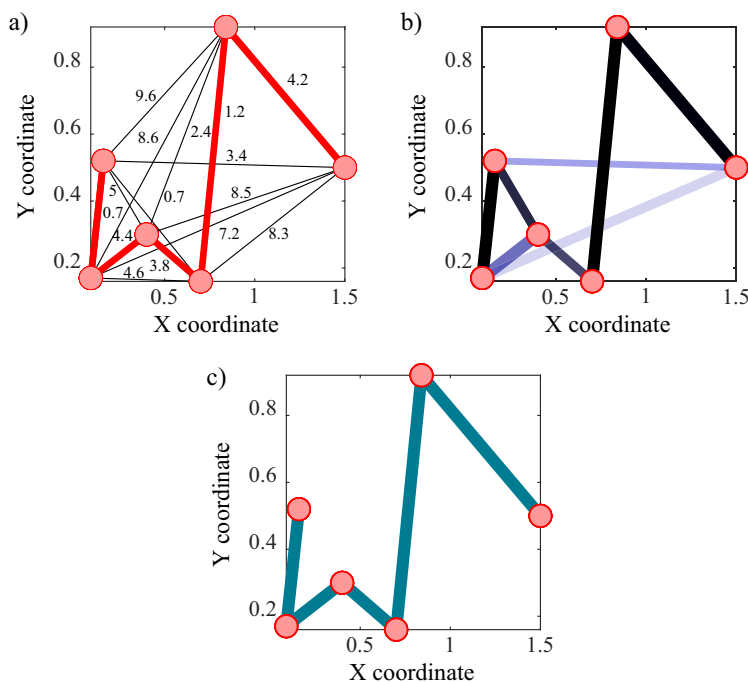


Fig. S3.2. Distance between nodes  $d$  is expressed by difficulty

Then by using proposed original metaheuristics Algorithm 2.2 based on the max-min ant colony and P system technology the best data path is calculated. Once the source node transfers data to another sensor node during given session period, it will start requesting and updating information on the current (used) routes and will also check for possibly better routes. Such process periodically checks for IoT changes caused by a broken connection due to the movement of sensor nodes or energy depletion in sensor nodes on the available path. It involves the same concept of path finding and updating of pheromone information so that ants can follow new data paths by spreading the value of the pheromone through a spread message, neighbors, and checking energy levels to calculate path quality.

After corresponding data routing starts the process of calculating qualitative parameters and it is continuous: monitoring the current network situation and responding to changes that have occurred. For example, increasing the delay for a given connection and exceeding the threshold for some traffic will enable finding an alternative path for the traffic that meets the required quality characteristics.

Finally update of the node residual energy because this parameter is used to search for the best path and defines the difficultness between the sensors.



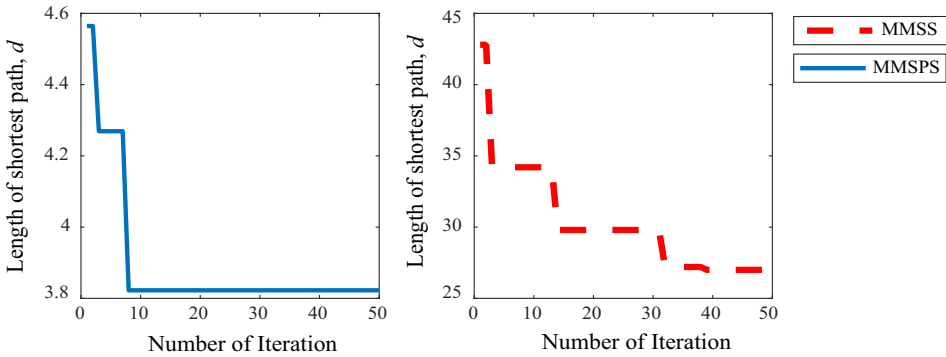
**Fig. S3.3.** Best (calculated) shortest path  $d^*$  value after 22 iterations: a) known shortest path  $d^*$ ; b) best (calculated) shortest path  $d^*$  value by MMAS; c) best (calculated) shortest path  $d^*$  value by MMAPS

**Table S3.1.** Average duration for path searching, iterations

Experiment number	MMAS	MMAPS
1–5	29.6	27.6
6–10	31.4	19.8
11–15	38.2	27.4
16–20	31.0	11.8
21–25	31.0	22.0
26–30	39.6	26.8
31–35	25.8	13.4
36–40	41.4	37.6
41–45	14.6	12.4
46–50	21.8	19.6
<b>Average</b>	<b>30</b>	<b>22</b>
<b>Standard deviation</b>	<b>8.77</b>	<b>8.22</b>

By using the same data set and the previously calculated path length between nodes, membrane computing calculations have been performed. After applying the proposed max-min ant P system algorithm, the best path search result is shown in Fig. S3.3.

Summarized simulation results in Table S3.1 indicate that after 22 iterations on average the pheromone settles on the best path (evaporates on the other roots) Fig. S3.4, and this is 25% faster than the result achieved by the max-min ant colony system. This experiment confirms the significance of the technique in terms of speed. The measured path

**Fig. S3.4.** The number of iterations after which the pheromone settles on the shortest path

delay parameter  $T$  for all the edges is obtained after the pheromone has settled on the best path for MMAS and MMAPS algorithms. The measured value is determined by:

$$T = \sum_{k=1}^n t_{ij}^{(k)}. \quad (\text{S3.1})$$

**Table S3.2.** Data path delay, ms

<b>Experiment number</b>	<b>MMAS</b>	<b>MMAPS</b>
1–5	22.30	17.14
6–10	27.72	17.56
11–15	21.62	17.26
16–20	25.24	16.90
21–25	25.36	20.84
26–30	25.18	16.56
31–35	25.24	15.84
36–40	31.92	25.28
41–45	23.56	17.02
46–50	23.48	19.12
<b>Average</b>	<b>25.16</b>	<b>18.15</b>

It is evident that the new approach to finding the best route by using MMAPS algorithm reduces delay of the overall path by 27% on average. Results shown in Table S3.2.

Half node dead duration value is shown in Table S3.3. It is the time to take the half of the nodes run out of energy. The experiment shows that the proposed algorithm has achieved better results. It takes 26% longer time for half of node run out of energy. It means that the higher number of nodes which has more remaining energy. The proposed algorithm provides a more balanced best path searching mechanism which has more nodes with the nearly equivalent energy.

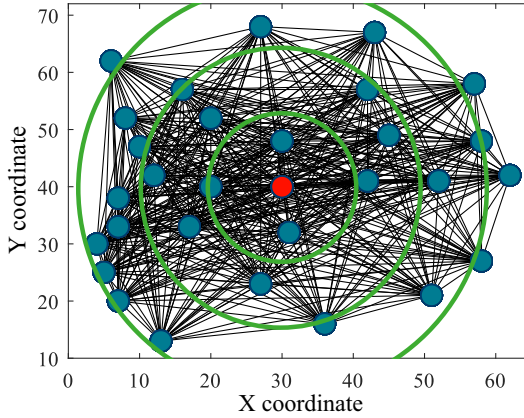
**Table S3.3.** IoT network lifetime, seconds

<b>Experiment number</b>	<b>MMAS</b>	<b>MMAPS</b>
1–5	1232	1721
6–10	1416	1544
11–15	1225	1468
16–20	1142	1486
21–25	1132	1328
26–30	1155	1449
31–35	1211	1612
36–40	1178	1481
41–45	1274	1423
46–50	1316	1748
<b>Average</b>	<b>1228</b>	<b>1526</b>

The MMAPS algorithm provides a more balanced best path searching mechanism to balance the energy between nodes.

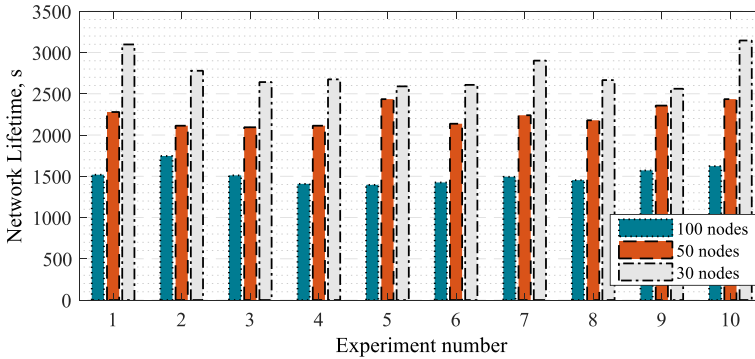
The last experiment has been made for the IoT network with nodes on the boundary. IoT network topology of 30 nodes with 4 nodes on the boundary shown in Fig. S3.5.

By using a neighbor search algorithm the IoT network topology has been redesigned.



**Fig. S3.5.** IoT network topology of 30 nodes

Once the nearest suitable neighbor is found, the proposed MMAPS algorithm for searching the best path is applied to the data transmission and the main and alternative paths are found.



**Fig. S3.6.** Network lifetime depending on the number of nodes in the IoT network

In the experiment, the lifetime of the network, depending on the number of nodes (devices) in the IoT network, has been compared. According to Fig. S3.6 the results presented show that increasing the number of nodes disproportionately decreases the lifetime of the IoT network because more steps through intermediate nodes need to be used.

## General Conclusions

1. To solve objective function optimization problems a new max-min ant P system (MMAPS) algorithm which is based on membrane computing has been developed.
2. The application of the max-min ant P system to process IoT data finds a best path 25% faster than the max-min ant system (MMAS).
3. On the basis of the Max-min ant P system algorithm, a new self-organizing Internet of Things methodology has been developed that effectively automates IoT data routing processes.
4. The detailed results of the IoT self-organizing methodology simulation experiments show that it was possible to find the optimal way of information transmission regardless of the parameters have been changed: a) the number of IoT devices; b) IoT network communication parameters; c) parameters determined by the MMAPS algorithm; d) MMAPS algorithm randomly selected parameters.
5. The proposed self-organizing Internet of Things methodology balances the network and under boundary conditions reduces the network energy consumption of the IoT devices network by 15% compared to MMAS, and thus prolongs the life of the mobile wireless sensor network.





---

# Dalykinė rodyklė

## Symbols

Žemdirbystė ..... 11

## A

Algoritmas

biotechnologinis ..... 17

evoliucinių skaiavimų ..... 18

hierarchinis ..... 16

kontekstą vertinantis ..... 17

max min skruzdžių kolonijų ..... 33, 44

tikimybių ..... 16

algoritmas

ACO ..... 76

MMSS ..... 33

OSK ..... 31

Aplinkos stebėjimas ..... 11

## B

Biologinis procesas ..... 18, 20, 22

## C

CH ..... 17

## D

D2D ..... 14–18, 20, 29

D2H ..... 14

Daiktų interneto tinklas ..... 29, 49, 50

DI . 1–3, 7–16, 18, 20, 29, 34, 35, 37, 38, 40, 46,  
47, 50–53, 55, 58–65

DIS ..... 5, 48, 64, 65

DNR ..... 27

Duomenų maršrutas ..... 15, 17, 44

Duomenų norminimas ..... 46

## E

Entropija ..... 46

Eurelio Venno diagrama ..... 1, 25

Evoliucijos taisyklės ..... 20, 24, 37, 39

aibių taisyklės ..... 24

komunikacijos taisyklės ..... 24

membranų tvarkymo taisyklės ..... 24

Evoliuciniai skaičiavimai

evoliucinė strategija ..... 18

evoliuciniai algoritmai ..... 18

evoliucinis programavimas ..... 18

genetiniai algoritmai ..... 18

genetinis programavimas ..... 18

gokinančių klasifikatorių sistemos ..... 18

## G

Gamtiniai skaičiavimai ..... 20, 22

Grafų teorija ..... 34, 37, 50

## I

Išmanieji miestai ..... 11

- Išmanieji namai ..... 11  
 Išmanusis protokolas ..... 15  
 Infobiotika ..... 2  
 IP ..... 8
- J**  
 Jutikliai ..... 2, 7, 8, 10, 50, 58
- K**  
 Kokybinės charakteristikos  
   atminties talpa ..... 57  
   delsos ..... 51  
   duomenų pralaidos ..... 7, 50, 51  
   energijos sunaudojimo ..... 7, 16, 45, 57  
   tinklo veikimo laiko ..... 57, 58  
 Kompanija  
   IBM ..... 12, 44  
   Sigfox ..... 15  
   Zigbee ..... 15  
 Komunikacija  
   įrenginys–įrenginiui ..... 14, 15, 29  
   įrenginys–žmogui ..... 14  
 KPU ..... 31, 32, 34
- L**  
 Ląstelė ..... 20  
 LEACH ..... 13
- M**  
 MAC ..... 8  
 Membraniniai skaičiavimai ..... 20, 22  
 Membranos struktūra ..... 23, 24, 34, 37  
   aplinka ..... 23  
   apvalkalas ..... 23  
   membranos ..... 23  
   sritis ..... 23  
 Metaeuristika ..... 18, 31  
 MMSPS ..... 4, 29, 34, 35, 48–51, 55–60, 62–65  
 MMSS ..... 5, 33, 37, 55–60, 65  
 MS ..... 20, 22, 23, 25–27
- N**  
 NP ..... 31
- O**  
 OSK ..... 3, 17, 18, 28–31, 34, 35  
   atigmergija ..... 17  
   feromonas ..... 18, 30  
   feromono garavimas ..... 30, 32, 33, 53  
   feromono pėdsakų atnaujinimas ..... 31–33  
   feromono riba ..... 33, 53  
   maisto paieška ..... 17, 30
- P**  
 P sistema ..... 1, 3, 22, 23, 25, 38, 40, 49  
 Programa  
   MATLAB ..... 49
- R**  
 RDA ..... 7  
 RFID ..... 14  
 RPL ..... 13
- S**  
 Savitvarka  
   Savarankiška konfigūracija ..... 12, 15, 44  
   Savarankiška optimizacija ..... 12, 15  
   Savarankiškas atsikūrimas ..... 12  
   Savarankiškas saugumas ..... 12  
 Spiečiaus algoritmai  
   Bičių algoritmas ..... 18  
   Optimizavimas skruzdžių kolonijomis ..... 17, 18,  
     29, 34  
   Optimizavimas spiečiaus dalelėmis ..... 18  
   Stochastinė difuzinė paieška ..... 18  
 Spiečiaus intelektas ..... 18  
 Sveikatos apsauga ..... 11
- T**  
 Tinklo topologijos ..... 17, 40, 46, 51, 60  
 Transportas ir logistika ..... 11
- V**  
 Virtualieji įrenginiai ..... 8

---

## Priedai<sup>1</sup>

**A priedas.** Disertacijos autoriaus sąžiningumo deklaracija

**B priedas.** Bendraautorių sutikimai teikti publikacijose skelbtą medžiagą disertacijoje

**C priedas.** Autoriaus mokslinių publikacijų disertacijos tema kopijos

---

<sup>1</sup>Priedai pateikiami pridėtoje kompaktinėje plokštelėje.

Aurimas GEDMINAS

MEMBRANINIŲ SKAIČIAVIMŲ TECHNOLOGIJA  
DAIKTŲ INTERNETO ĮRENGINIŲ SAVITVARKAI

Daktaro disertacija

Technologijos mokslai,  
elektros ir elektronikos inžinerija (T 001)

MEMBRANE COMPUTING TECHNOLOGY FOR  
SELF-ORGANIZATION OF DEVICES IN THE INTERNET OF THINGS

Doctoral Dissertation

Technological Sciences  
Electrical and Electronic Engineering (T 001)

2020 07 24. 9,5 sp. l. Tiražas 20 egz.  
Vilniaus Gedimino technikos universiteto  
leidykla „Technika“,  
Saulėtekio al. 11, 10223 Vilnius,  
<http://leidykla.vgtu.lt>  
Spausdino BĮ UAB „Baltijos kopija“,  
Kareivių g. 13B, 09109 Vilnius