VILNIAUS UNIVERSITETAS

LORETA SAVULIONIENĖ

SUSIETUMO TAISYKLIŲ PAIEŠKA DIDELĖSE DUOMENŲ BAZĖSE

Daktaro disertacija

Technologijos mokslai, informatikos inžinerija (07 T)

Vilnius, 2014

Disertacija parengta 2008 – 2013 metais Vilniaus universiteto Matematikos ir informatikos institute.

Mokslinis vadovas:

prof. habil. dr. Leonidas Sakalauskas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07 T).

VILNIUS UNIVERSITY

LORETA SAVULIONIENĖ

ASSOCIATION RULES SEARCH IN LARGE DATA BASES

Summary of Doctoral Dissertation

Technological Sciences, Informatics Engineering (07 T)

Vilnius, 2014

The doctoral dissertation was accomplished at Vilnius University Institute of Mathematics and Informatics in the period from 2008 to 2013.

Scientific Supervisor

Prof. Dr. Habil. Leonidas Sakalauskas (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

Padėka

Nuoširdžiai dėkoju moksliniam darbo vadovui prof. habil. dr. Leonidui Sakalauskui už vertingas mokslines konsultacijas, už patarimus, pastabas, pasiūlymus bei nuolatinį skatinimą tobulėti.

Dėkoju Matematikos ir informatikos instituto direktoriui prof. habil. dr. Gintautui Dzemydai už suteiktas sąlygas doktorantūros studijoms, visapusę paramą ir supratimą, doktorantams už bendradarbiavimą ir pagalbą.

Dėkoju disertacijos recenzentams doc. dr. Olgai Kurasovai ir dr. Virginijui Marcinkevičiui už vertingus patarimus, pasiūlymus bei kritines pastabas, padėjusias tobulinti disertaciją.

Dėkoju Vilniaus kolegijos Elektronikos ir informatikos fakulteto dekanui doc. dr. Romanui Tumasoniui bei Programinės įrangos katedrai už jų paramą, supratingumą. Taip pat nuoširdžiai dėkoju Aušrai Netikšienei už pagalbą rengiant disertacijos santraukos tekstą.

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

Už nuolatinį palaikymą, kantrybę, meilę ir supratingumą dėkoju dukrai Gabrielei, vyrui Dainiui, mamai, tėtei ir sesei Intai.

Loreta Savulionienė

TURINYS

Padėka 5

Turinys 6

Reziumė 8

Abstract 10

Žymėjimai 12

Santrumpos 15

Paveikslų sąrašas 16

Lentelių sąrašas 17

1 skyrius. Įvadas 18

1.1. Tyrimo sritis ir problemos aktualumas 18

1.2. Tyrimų objektas 21

1.3. Darbo tikslas ir uždaviniai 21

1.4. Mokslinis darbo naujumas 22

1.5. Tyrimo metodika 23

1.6. Darbo praktinė reikšmė 23

1.7. Ginamieji teiginiai 23

1.8. Darbo rezultatų aprobavimas 24

1.9. Disertacijos struktūra 26

2 skyrius. Susietumo taisyklių nustatymas 27

2.1. Įvadas 27

2.2. Susietumo taisyklės 28

2.3. Susietumo taisyklių nustatymas 35

2.4. Antrojo skyriaus apibendrinimas 37

3 skyrius. Dažnų posekių paieškos algoritmai 38

3.1. Tikslieji dažnų posekių paieškos algoritmai 38

3.1.1. Apriori algoritmas ir jo modifikacijos 41

3.1.2. GSP algoritmas 44

3.1.3. Rekursinis algoritmas 46
3.1.4. SPADE algoritmas 46
3.1.5. Tiksliųjų algoritmų greičio palyginimas 47
3.2. Apytiksliai dažnų posekių paieškos algoritmai 49
3.2.1. ApproxMAP algoritmas 49
3.2.2. Tikimybinis dažnų sekų nustatymo algoritmas ProMFS 52
3.2.3. Apytikslis atsitiktinės imties metodas RSM 54
3.3. Trečiojo skyriaus apibendrinimas ir išvados 55
4 skyrius. Stochastiniai dažnų posekių paieškos algoritmai 58
4.1. Pasiūlytas naujas stochastinis dažnų posekių paieškos algoritmas 58
4.2. Pasiūlytos naujo stochastinio dažnų posekių algoritmo modifikacijos 63
4.3. Pasiūlytas naujas stochastinis susietumo taisyklių paieškos algoritmas 65
4.4. Statistinės charakteristikos 69
4.5. Ketvirtojo skyriaus išvados 73
5 skyrius. Eksperimentiniai tyrimai 75
5.1. Eksperimentinės duomenų bazės 75
5.2. Naujai pasiūlytų stochastinio dažnų posekių, SDPA1 ir SDPA2 algoritmų tikslumo tyrimas 76
5.4. Baigiamųjų darbų temų duomenų bazės tyrimas 82
5.4. Transakcijų duomenų bazės tyrimas 88
5.5. Penktojo skyriaus išvados 97
Bendros išvados 99
Literatūros sąrašas 101
PRIEDAI 113
Priedas 1. Dažnų posekių skaičiaus kitimas 113
Priedas 2. Duomenų bazių grupės ir jų generavimo charakteristikos. 114
Priedas 3. Transakcijų duomenų bazėje išskirtos susietumo taisyklės. 115
Priedas 4. Transakcijų duomenų bazėje išskirtos apibendrintos susietumo taisyklės. 124

Reziumė

Informacinių technologijų įtaka neatsiejama nuo šiuolaikinio gyvenimo. Bet kokia veiklos sritis yra susijusi su informacijos, duomenų kaupimu ir saugojimu. Šiandien nebepakanka tradicinio duomenų apdorojimo bei įvairių ataskaitų formavimo. Duomenų tyrybos technologijų taikymas leidžia iš turimų duomenų išgauti naujus faktus ar žinias, kurios leidžia prognozuoti veiklą, pavyzdžiui, pirkėjų elgesį ar finansines tendencijas, diagnozuoti ligas ir pan. Disertacijoje nagrinėjami duomenų tyrybos algoritmai dažniems posekiams ir susietumo taisyklėms nustatyti. Pagrindinis disertacijos tikslas - sukurti naują apytikslį dažnų posekių paieškos algoritmą, jo modifikacijas ir susietumo taisyklių nustatymo algoritmą bei pateikti algoritmų paklaidų įvertinimą.

Disertaciją sudaro penki skyriai, išvados, literatūros sąrašas bei 4 priedai.

Pirmajame (įvadiniame) skyriuje nagrinėjamas problemos aktualumas, formuluojamas darbo tikslas ir uždaviniai, aprašomas mokslinis darbo naujumas, pristatomi autorės pranešimų ir publikacijų sąrašai, disertacijos struktūra.

Antrajame skyriuje apžvelgiamos susietumo taisyklės, susietumo taisyklių nustatymo metodai.

Trečiajame skyriuje apžvelgiami tikslieji ir apytiksliai dažnų posekių paieškos algoritmai.

Ketvirtajame skyriuje pateikiamas naujai sukurtas stochastinis dažnų posekių paieškos bei jo modifikacijos ir susietumo taisyklių nustatymo algoritmas, taip pat aprašomos algoritmų statistinės charakteristikos.

Penktajame skyriuje pateikiami pasiūlytų algoritmų eksperimentinių tyrimų rezultatai.

Darbo pabaigoje pateikiamos bendrosios išvados bei cituojamos literatūros sąrašas.

Bendra disertacijos apimtis – 125 puslapiai, 21 paveikslas ir 10 lentelių. Literatūros sąrašą sudaro 108 šaltiniai.

Tyrimų rezultatai publikuoti 7 recenzuojamuose mokslo leidiniuose. Rezultatai pristatyti ir aptarti 5 respublikinėse konferencijose ir 2 tarptautinėse mokslininkų konferencijose.

Abstract

The impact of information technology is an integral part of modern life. Any activity is related to information and data accumulation and storage, therefore, quick analysis of information is necessary. Today, the traditional data processing and data reports are no longer sufficient. The need of generating new information and knowledge from given data is understandable; therefore, new facts and knowledge, which allow us to forecast customer behaviour or financial transactions, diagnose diseases, etc., can be generated applying data mining techniques. The doctoral dissertation analyses modern data mining algorithms for estimating frequent subsequences and association rules. The research object is data mining algorithms for solving data analysis tasks and discovery associations rules. The aim of the dissertation is to propose a new approximate algorithm for mining frequent sub-sequences and association rules and its modifications, and to present the evaluation of the algorithm errors.

The doctoral dissertation consists of five chapters, conclusions, the list of referred resources and appendices.

In the first chapter (Introduction) the relevance of the analysed problem is explained, the aim and objectives of the dissertation are formulated, as well as the scientific novelty is stated, the author‘s presentations and publications and the structure of the dissertation are presented.

In the second chapter association rules and methods for discovering association rules are overviewed.

In the third chapter exact and approximate algorithms for mining frequent subsequences are reviewed.

In the fourth chapter new generated stochastic algorithm for mining frequent subsequences and its modifications are presented and statistical characteristics of algorithms are described.

The fifth chapter focuses on the research results of the efficiency of the algorithms.

Finally the conclusions and the list of referred resources are given.

The total scope of the dissertation is 125 pages, 21 figures and 10 tables. The list of referred resources includes 108 resources.

The results of the research are published in 7 reviewed scientific publications. The results were presented and discussed in 5 national and 2 international scientific conferences.

Žymėjimai

A	Hierarchinis atstumas tarp posekių X ir Y
$\tilde{A}$	Atstumų vidurkių matrica
a_{jv}	Atstumų vidurkių matricos $\tilde{A}$ elementas
B	Grafo briaunų (lankų) aibė
BetaInv	Beta skirstinio kvantilis
C	Įvesties seka
C_k	Elementas kandidatas
$\underline{C}$	Modelinė seka
c_r	Modelinės sekos $\underline{C}$ elementas
conf	Patikimumas
d	Hierarchijos lygmuo
D	Transakcijų duomenų bazė
dist	Hierarchinis atstumas
Density	Tankumas
eid	Įvykio unikalus numeris
E	Tikėtina susietumo taisyklės reikšmė
F_k	Dažna seka
f	Funkcija
g	Tolygiojo skirstinio, pagal kurį pasiskirstęs atstumas (tarpo) tarp dviejų analizuojamų posekių, parametras
G	Grafas
H_0	Nulinė hipotezė
H_1	Priešingas tvirtinimas nulinei hipotezei
I	Elementų aibė
i_k	k-atasis aibės elementas
id	Transakcijos numeris
INDEL()	Įterpimo arba pašalinimo operatorius

l	Posekio ilgis
min_supp	Minimalus dažnumas
min_conf	Minimalus patikimumas
N	Posekių (imčių) skaičius
N_k	k ilgio posekių skaičius
n	Elementų skaičius aibėje
Ñ	Dažniausių atstumų matrica
O	Operacijų skaičius
p_1 ir p_2	Pasikliovimo tikimybės rėžiai
$\overline{p}_n$	Empyrinis dažnumas
$P(i_j)$	Elemento i_j pasirodymo pagrindinėje sekoje tikimybė
$Pr(X)$	Tikimybė, kad visi X elementai bus išplėstinėje transakcijoje
$Q(i_j, c_r)$	Modelinės sekos $\underline{C}$ elemento c_r charakteristika
q	Tolygiojo skirstinio, pagal kurį pasiskirstęs posekio ilgis, parametras
R	Susietumo taisyklės įdomumo parametras
$REPL()$	Keitimo operatorius
s	Duomenų bazės skenavimo numeris
S_i	Posekių šablonai
$(s_1 \rightarrow s_2 \rightarrow ... \rightarrow s_k)$	Seka S
$supp\ X$	Rinkinio X dažnumas
sid	Įvesties sekos C unikalus numeris
T	Transakcija
u, z	Kriterijaus statistika
V	Grafo viršūnių (mazgų) aibė
v	Posekio ilgio kitimas
$V(i_j)$	Elementų i_j skaičius pagrindinėje sekoje
WS	Svertinė seka
$X, Y, ...Z$	Elementų aibės I poaibiai (rinkiniai)

$\overline{X}, \overline{Y}, ...$	Poaibių *X*, *Y*, ... tėvai
$x, y, ...$	Poaibių *X*, *Y*, ... elementai
$X \Rightarrow Y$	Susietumo taisyklė tarp rinkinio *X* ir *Y*
$\overline{X} \Rightarrow \overline{Y}$	Susietumo taisyklės $X \Rightarrow Y$ tėvas
$\bar{x}$	Elemento *x* tėvas
γ	Pasikliovimo tikimybė
η_i	Atsitiktinės sekos elementas
Ω	Atsitiktinis dydis

Santrumpos

Santrumpa	**Pilnas pavadinimas**
BetaInv	Beta skirstinio kvantilis
GSP (Generalized Sequential Pattern algorithm)	Apibendrintas sekų šablonų algoritmas
LAPIN (Last Position Induction algorithm)	Paskutinės pozicijos indukcijos algoritmas
MRA (Multi Resolution Analysis)	Daugybinių sprendimų analizė
PRISM (Prime – Encoding Based Sequence Mining)	Pirminiais skaičiais paremta sekų paieška
ProMFS (Probabilistic algorithm for mining frequent sequences)	Tikimybinis dažnų sekų nustatymo algoritmas
RSM (Random Sampling Method)	Atsitiktinės imties metodas
SQL (Structured Query Language)	Struktūrizuota užklausų kalba
SPADE (Sequential Pattern Discovery using equivalence classes algorithm)	Dažnų šablonų paieškos algoritmas, naudojant klasių ekvivalentiškumą
SPAM (Sequential Pattern Mining)	Dažnų šablonų paieškos algoritmas
SDPA1	Stochastinis modifikuotas dažnų posekių paieškos algoritmas 1
SDPA2	Stochastinis modifikuotas dažnų posekių paieškos algoritmas 2

Paveikslų sąrašas

1 pav. Duomenų tyrybos uždaviniai.....19
2 pav. Prekių hierarchijos modelis. 31
3 pav. Tiksliųjų algoritmų veikimo trukmė duomenų bazėje D1.....48
4 pav. Tiksliųjų algoritmų veikimo trukmė duomenų bazėje D2.....49
5 pav. Stochastinio dažnų posekių paieškos algoritmo veikimo schema.....62
6 pav. Stochastinio susietumo taisyklių paieškos algoritmo veikimo schema...68
7 pav. SDPA1 algoritmo vykdymo trukmės priklausomybė nuo parametro *g* reikšmės. 77
8 pav. SDPA1 algoritmo bendro dažnų posekių skaičiaus priklausomybė nuo parametro *g* reikšmės. 79
9 pav. Vidutinis Apriori, GSP, SPADE, rekursinio, ProMFS, stochastinio, SDPA1 ir SDPA2 algoritmų veikimo laikas. 80
10 pav. Baigiamųjų darbų temų skaičius. 83
11 pav. Apytikslių metodų greičio palyginimas. 84
12 pav. 2000 – 2013 m. baigiamųjų darbų temų pasirinkimas. 88
13 pav. Algoritmų laiko sąnaudų palyginimas. 89
14 pav. Algoritmų dažnų posekių skaičiaus palyginimas. 90
15 pav. SPADE algoritmo dažnų posekių skaičiaus palyginimas pagal posekių ilgį. 90
16 pav. Stochastinio algoritmo dažnų posekių skaičiaus palyginimas pagal posekių ilgį.....91
17 pav. SDPA1 algoritmo dažnų posekių skaičiaus palyginimas pagal posekių ilgį. 91
18 pav. SDPA2 algoritmo dažnų posekių skaičiaus palyginimas pagal posekių ilgį. 91
19 pav. Kriterijaus statistikos *u* kitimas. 92
20 pav. Kriterijaus statistikos *z* kitimas.....93
21 pav. Tikėtinumo funkcijos *P*(0) reikšmių kitimas.....94

Lentelių sąrašas

1 lentelė. Elementų priklausymo transakcijai lentelė. 30
2 lentelė. Dažni posekiai ir jų skaičius. 78
3 lentelė. Pasikliautinojo intervalo rėžiai. 81
4 lentelė. SDPA1 algoritmo pirmos ir antros rūšies klaidos. 82
5 lentelė. Posekiai ir jų dažnumas. 85
6 lentelė. 2 – elementų posekiai ir jų dažnumas. 85
7 lentelė. Susietumo taisyklės. 86
8 lentelė. Susietumo taisyklių skaičius. 95
9 lentelė. Susietumo taisyklių skaičius ir laikas. 95
10 lentelė. Apibendrintų susietumo taisyklių skaičius. 96

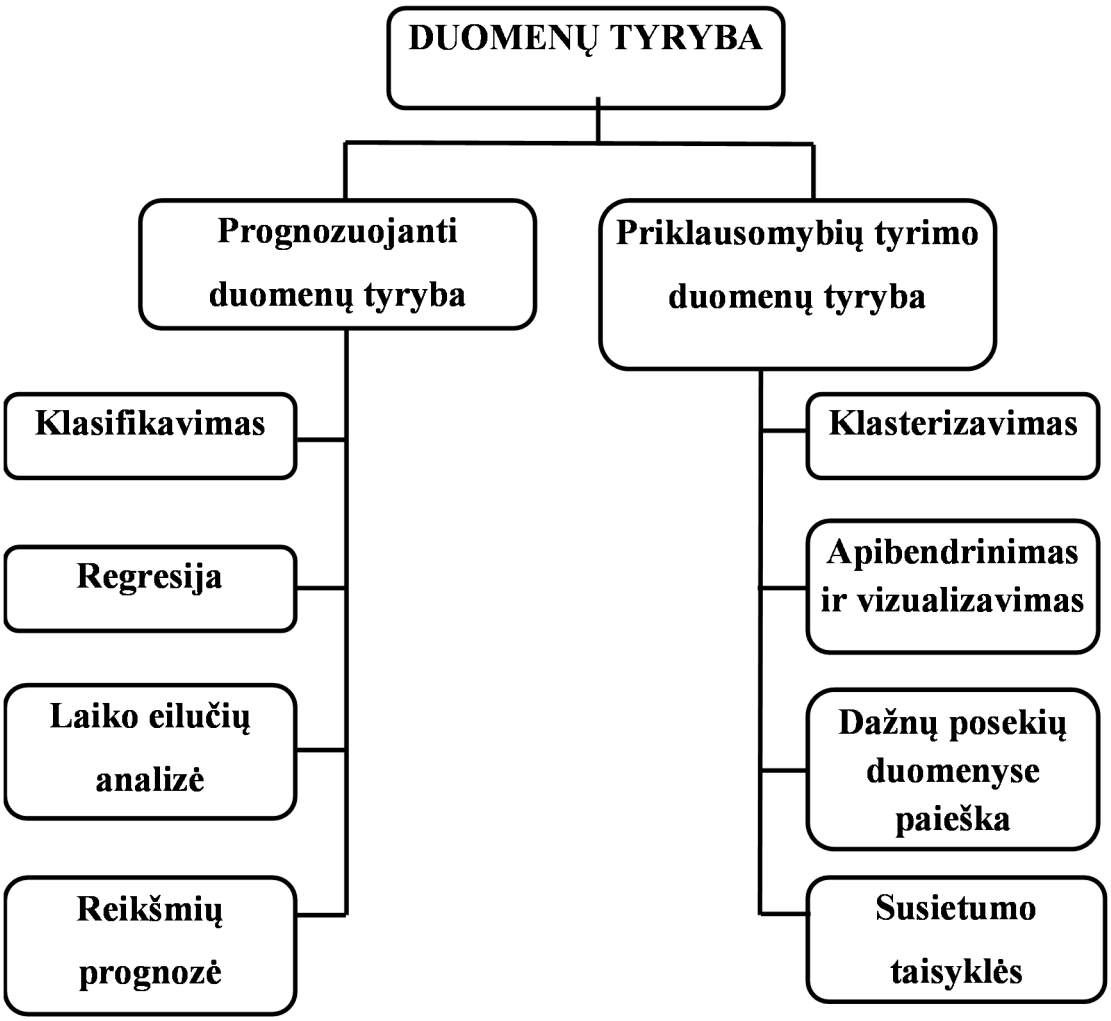
1 skyrius. Įvadas

Informacija yra mūsų eros valiuta, bet kokia yra jos vertė?
Informacija nėra žinios, ir tai tikrai nėra išmintis.
John Lippman

1.1. Tyrimo sritis ir problemos aktualumas

Bet kokios įmonės veikla šiandien susijusi su dideliais informacijos ir duomenų kiekiais. Informacija – tai žinios apie faktus, įvykius, daiktus, procesus, idėjas, sąvokas ir kitus objektus, kurios tam tikrame kontekste turi prasmę. Duomenys – tai formalizuotas informacijos vaizdinys, tinkamas perduoti kitiems, suvokti ir apdoroti. Dažnai teigiama: žinios – tai galia, žinios – tai pinigai ir pan. Daugelio naudojamas žinių apibrėžimas yra toks: žinios yra vertę turinti informacija. Žinios – tai visuma apdorotos informacijos bei kintančių sąryšių, kurių pagrindu gali veikti koks nors asmuo arba sistema. Vadinasi, nebepakanka turėti daug informacijos ar kaupti terabaitinius kiekius duomenų, svarbiausia iš turimos informacijos bei duomenų išgauti įžvalgas, sąryšius tarp duomenų ar naujas žinias. Tarp didelių informacijos kiekių slepiasi ir svarbi, ir niekinė informacija. Efektyvus informacijos, slypinčios duomenyse, atskleidimas ir panaudojimas [27] yra svarbiausias konkurencingumo didinimo veiksnys šiuolaikinėje dinamiškoje tyrimų ir verslo aplinkoje. Šioms problemoms spręsti taikoma duomenų tyryba (*angl. Data Mining*).

Duomenų tyryba yra šiuolaikinė informacijos analizės sritis, atsiradusi duomenų bazių technologijų, dirbtinio intelekto ir statistinės duomenų analizės sankirtoje. Duomenų tyryba yra labai plati sritis, apimanti daug metodų, algoritmų bei taikomųjų programinių sistemų. Duomenų tyrybos uždavinių klasifikavimas pateiktas 1 paveiksle.



1 pav. Duomenų tyrybos uždaviniai.

Dažnų posekių paieškos ir susietumo taisyklių (*angl. association rule*) nustatymo uždavinių pagrindas yra šablonų (*angl. pattern*), atvaizduojančių duomenų tarpusavio sąryšius, koncepcija. Šie šablonai atskleidžia vidinę duomenų struktūrą bei dėsningumus, būdingus duomenų poaibiams, kurie išreiškiami vartotojui suprantamu pavidalu – susietumo taisyklėmis. Svarbi duomenų tyrybos savybė yra ieškomų šablonų netrivialumas. Šablonai turi atvaizduoti neakivaizdžius, netikėtus (*angl*. *unexpected*) duomenų sąryšius, kurie vadinami paslėptomis žiniomis (*angl*. *hidden knowledge*). Garsi statistikos ir analizės bendrovė „Gartner Group“ duomenų tyrybos apibrėžimą formuluoja taip: Duomenų tyryba yra prasmingų šablonų, dėsningumų, modelių ir tendencijų radimo procesas dideliuose informacijos kiekiuose, panaudojant modelių atpažinimo, statistinius bei matematinius metodus.

G. Piatecki – Shapiro [30] duomenų tyrybą apibrėžia taip: Duomenų tyryba yra neapdorotų duomenų (*angl. raw data*) tyrinėjimo procesas siekiant nustatyti žinias, kurios yra naujos, prieš tai nežinotos; netrivialios; praktiškai naudingos; interpretuojamos bei būtinos sprendimams priimti pasirinktoje veiklos srityje. Duomenų tyryba apibūdinama kaip naujų prasmių duomenyse aptikimo, nustatymo, atradimo būdas.

Susietumo taisyklių paieška taikoma versle, finansinėse institucijose, medicinoje, nuotoliniame mokyme ir kitose srityse, kur tenka apdoroti didelius informacijos kiekius bei aptikti sąryšius tarp duomenų.

Šis procesas susideda iš kelių etapų:

- duomenų kaupimas (surinkimas);
- duomenų paruošimas apdorojimui;
- algoritmų taikymas;
- nustatytų sąryšių tarp duomenų pateikimas.

Nustatyti sąryšiai tarp duomenų padeda analitikams greičiau bei tiksliau priimti sprendimus, todėl susietumo taisyklių paieška yra svarbus uždavinys.

Vienas iš seniausių, bet ir šiuo metu labai aktualus verslui duomenų tyrybos uždavinys yra pirkinių krepšelio (*angl. market basket*) uždavinys [5, 9, 18, 26, 33, 75, 76, 77, 80, 86]. Prekybos tinklai turi didelius duomenų kiekius apie pirkimus ir jiems aktualu iš turimų duomenų išgauti naudingas naujas žinias sėkmingų sprendimų priėmimui bei verslo vystymui [71], todėl nuolatos aktualu žinoti, kokios prekės sudaro pagrindinį pirkinių krepšelį [18, 72], kaip jis kinta, ar priklauso nuo tam tikro sezono ir kitų veiksnių.

Panašius uždavinius formuluoja ir kitos veiklos sritys, pavyzdžiui kaip iš tam tikrų požymių nustatyti ligas, kokie dėsniai pasireiškia finansų rinkose, kokia investavimo strategija ir pan. Šiems uždaviniams spręsti taikomi tikslieji ir apytiksliai dažnų posekių paieškos, susietumo taisyklių nustatymo metodai ir algoritmai [79]. Tikslieji algoritmai daug kartų skaito duomenų bazę, todėl imlūs laikui ir taikomi, kur svarbus rezultatų tikslumas (medicinos, genetikos uždaviniai), o apytiksliai metodai yra greitesni, todėl jų taikymo sritis daug platesnė.

Nagrinėti anksčiau sukurti apytiksliai metodai analizuoja ne visą pradinę duomenų bazę, o tik atsitiktinę jos imtį ir išskiria dažnus posekius, tačiau nenustato susietumo taisyklių.

Susietumo taisyklių paieška didelėse duomenų bazėse yra vienas iš svarbiausių duomenų tyrybos uždavinių. Duomenų bazės skirstomos į keletą rūšių: mažos duomenų bazės (*angl. small database*), vidutinės duomenų bazės (*angl. medium database*), didelės duomenų bazės (*angl. large database*), labai didelės duomenų bazės (*angl. very large database*) ir dideli duomenys (*angl. big data*). Didelės duomenų bazės apibrėžimas nuolatos kinta, kaip ir techninės bei programinės įrangos galimybės apdoroti didelius duomenų kiekius. Vieni apibrėžia didelės duomenų bazės dydį įrašų skaičiumi $\geq 10^7$ įrašų, kiti – duomenų dydžiu ≥40 GB.

Disertacijoje yra pasiūlytas naujas apytikslis dažnų posekių paieškos algoritmas, jo modifikacijos ir susietumo taisyklių paieškos algoritmas bei pateiktas algoritmų paklaidų įvertinimas. Algoritmų rezultatai buvo palyginti su kitais tiksliaisiais bei apytiksliais algoritmais.

1.2. Tyrimų objektas

Disertacijos tyrimo objektas yra duomenų tyrybos algoritmai ir metodai, skirti dažnų posekių ir susietumo taisyklių nustatymo uždaviniams spręsti. Disertacijoje aprašytuose tyrimuose naudojamos imituotos ir realios duomenų bazės.

1.3. Darbo tikslas ir uždaviniai

Disertacijos tikslas yra sudaryti naują apytikslį dažnų posekių paieškos algoritmą, jo modifikacijas ir susietumo taisyklių nustatymo algoritmą bei pateikti algoritmų paklaidų įvertinimą.

Siekiant įgyvendinti užsibrėžtą tikslą sprendžiami šie uždaviniai:

- Išnagrinėti dažniausiai naudojamus duomenų tyrybos metodus ir algoritmus, skirtus dažnų posekių bei susietumo taisyklių nustatymui.
- Sukurti naują apytikslį dažnų posekių paieškos algoritmą.
- Įvertinti naujo algoritmo tikslumą, greitį bei statistines charakteristikas.
- Palyginti sudarytą apytikslį algoritmą su Apriori, GSP, SPADE, rekursiniu ir tikimybiniu dažnų sekų nustatymo ProMFS algoritmu.
- Realizuoti sukurto algoritmo modifikacijas algoritmo tikslumui padidinti.
- Realizuoti sukurto algoritmo modifikaciją susietumo taisyklėms nustatyti.
- Sukurti programinę įrangą eksperimentams atlikti.
- Atlikti eksperimentus su imituotais ir realiais duomenimis bei palyginti su kitais tiksliaisiais ir apytiksliais algoritmais.

1.4. Mokslinis darbo naujumas

Šioje disertacijoje yra nagrinėjamas aktualus susietumo taisyklių paieškos uždavinys, kurio sprendimui pasiūlytas naujas stochastinis dažnų posekių paieškos algoritmas, jo modifikacijos ir stochastinis susietumo taisyklių paieškos algoritmas, įvertintos šių algoritmų tikimybinės charakteristikos. Šie dažnų posekių ir susietumo taisyklių paieškos algoritmai yra apytiksliai, kurie vieno duomenų bazės skenavimo metu nustato dažnus posekius bei susietumo taisykles. Naujai pasiūlytų algoritmų paklaidų įvertinimas atliktas naudojantis statistiniais metodais. Šių algoritmų veikimas greitesnis lyginant su tiksliaisiais ir su nagrinėtais apytiksliais dažnų posekių paieškos algoritmais.

1.5. Tyrimo metodika

Pagrindiniai tyrimo metodai taikomi disertacijoje – informacijos paieška, duomenų imitavimas, informacijos sisteminimas, analizė, lyginamoji analizė, apibendrinimas, statistinė analizė, žvalgomasis tyrimas, eksperimentinis tyrimas. Analizuojant kitų autorių mokslinius ir eksperimentinius pasiekimus dažnų posekių ir susietumo taisyklių paieškos srityje, buvo naudoti informacijos paieškos, duomenų imitavimo, sisteminimo, analizės, lyginamosios analizės, žvalgomojo tyrimo ir apibendrinimo metodai. Sudarytiems algoritmams įvertinti buvo naudotas eksperimentinio tyrimo metodas ir statistinė analizė.

1.6. Darbo praktinė reikšmė

Darbe sukurtas stochastinis dažnų posekių paieškos algoritmas, jo modifikacijos SDPA1, SDPA2 ir stochastinis susietumo taisyklių paieškos algoritmas. Sukurta programinė įranga, kurioje realizuoti Apriori, GSP, SPADE, rekursinis, ProMFS, stochastinis dažnų posekių paieškos, SDPA1, SPDA2 bei stochastinis susietumo taisyklių paieškos algoritmai. Programinė įranga buvo naudojama eksperimentams atlikti ir yra parengta realiam naudojimui.

1.7. Ginamieji teiginiai

- Darbe sukurti stochastinis dažnų posekių paieškos, SDPA1, SDPA2 ir stochastinis susietumo taisyklių paieškos algoritmai yra apytiksliai, tačiau yra pakankamai tikslūs ir greiti.
- SDPA1 algoritmas yra stochastinio dažnų posekių paieškos algoritmo modifikacija, kuri padidina algoritmo tikslumą, kai pasirenkamas parametras $g \in [0{,}6;\ 1]$.

- SDPA2 algoritmas yra stochastinio dažnų posekių paieškos algoritmo modifikacija, kuri naudoja vieno elemento dažnus posekius, nustatytus pasirinktu tiksliuoju dažnų posekių paieškos algoritmu. SPDA2 algoritmas tikslesnis už stochastinį dažnų posekių paieškos ir SPDA1 algoritmus.

1.8. Darbo rezultatų aprobavimas

Pagrindiniai tyrimo rezultatai atspausdinti 7 mokslinėse publikacijose, rezultatai pristatyti 2 tarptautinėse mokslininkų konferencijose ir 5 respublikinėse konferencijose.

Pranešimai skaityti šiose konferencijose:

- Respublikinė konferencija: LOTD 3-oji jaunųjų mokslininkų konferencija: Operacijų tyrimai verslui ir socialiniams procesams, Lietuva, Vilnius, 2010 m. spalio 1 d.
- Respublikinė konferencija: IV respublikinė mokslinė – praktinė konferencija: Mokslo taikomųjų tyrimų įtaka šiuolaikinių studijų kokybei, Lietuva, Vilnius, 2011 m. gegužės 5 d.
- Respublikinė konferencija: LOTD 4-oji jaunųjų mokslininkų konferencija: Operacijų tyrimai versle, inžinerijoje ir informacinėse technologijose, Lietuva, Kaunas, 2011 m. rugsėjo 30 d.
- Konferencija: Kompiuterininkų dienos – 2013, Lietuva, Šiauliai, 2013 m., rugsėjo 19-21 d.
- Respublikinė konferencija: VI respublikinė mokslinė – praktinė konferencija: Mokslo taikomųjų tyrimų įtaka šiuolaikinių studijų kokybei, Lietuva, Vilnius, 2013 m. gegužės 6 d.
- Tarptautinė konferencija: Distributed Systems and Big Data – Towards New Horizons, NORDUGRID 2013, Lietuva, Šiauliai, 2013 m. birželio 4-6 d.

- Tarptautinė konferencija: The 19th International Conference on Information and Software Technologies (ICIST 2013), Lietuva, Kaunas, spalio 10-11 d., 2013.

Disertacijos autorės publikacijų sąrašas:

- Savulionienė, L. Dažnų posekių paieškos algoritmai ir jų rezultatai. *IV respublikinės mokslinės – praktinės konferencijos mokslinių straipsnių rinkinys*. ISSN 2029-2279, 2011, p. 107-113.
- Savulionienė, L., Sakalauskas, L. Statistinis dažnų posekių paieškos algoritmas. *Informacijos mokslai*, Vol. 58. ISSN 1392-0561, 2011, p.126-143.
- Savulionienė, L., Sakalauskas, L. Stochastinis dažnų posekių paieškos algoritmas. *Jaunųjų mokslininkų darbai*, Vol. 4(33). ISSN 1648-8776, 2011, p. 138-145.
- Savulionienė, L. Stochastinis modifikuotas dažnų posekių paieškos algoritmas. *VI respublikinės mokslinės – praktinės konferencijos mokslinių straipsnių rinkinys*. ISSN 2029-2279, 2013, p. 80-91.
- Savulionienė, L., Sakalauskas, L. Modifikuoto stochastinio dažnų posekių paieškos algoritmo tikimybinės charakteristikos. *XVI Kompiuterininkų konferencijos mokslo darbai*. ISBN 978-9986-34-293-9, 2013, p. 75-87.
- Savulionienė, L., Sakalauskas, L. Modified Stochastic Algorithm for Mining Frequent Subsequences. *Information and Software Technologies*, Vol. 403. ISBN: 978-3-642-41946-1 (Print), 978-3-642-41947-8 (Online), 2013, p. 222-235.
- Savulionienė, L., Sakalauskas, L. Stochastic frequent set search algorithm for association rules discovery. *Information technology and control*. ISSN:1392-124X (Print); ISSN: 2335-884X (Online). Priimtas (bus atspausdintas 2014 m. birželio mėn., T.43 Nr.2).

1.9. Disertacijos struktūra

Disertaciją sudaro 5 skyriai, bendrosios išvados, literatūros sąrašas. Disertacijos skyriai: įvadas, susietumo taisyklių nustatymas, dažnų posekių paieškos algoritmai, stochastiniai dažnų posekių paieškos algoritmai, tyrimo rezultatai. Disertacijos apimtis 125 puslapiai, 10 lentelių, 21 paveikslas, 4 priedai. Disertacijoje remtasi 108 šaltiniais.

2 skyrius. Susietumo taisyklių nustatymas

2.1. Įvadas

Techninių ir programinių priemonių spartus vystymasis, jų naudojimas įvairiose veiklos srityse yra neatsiejamas nuo duomenų kaupimo. Pasak informacinių technologijų analitikų, 2013 metais visame pasaulyje sukurtų ir padaugintų duomenų kiekis siekė apie 1,8 mlrd. terabaitų. Moore dėsnis teigia, kad kompiuterių galingumas padvigubėja apytiksliai kas 18 mėnesių, o tuo tarpu sukaupta informacija pasaulyje padvigubėja apytiksliai kas 40 mėnesių, o nuo 2012 m. kiekvieną dieną sukaupiamas papildomas kvintilijonas baitų informacijos. Tačiau informacija pati savaime yra kaip naudingieji ištekliai po žeme – kol jie neišgauti, neapdirbti, tol jie tiesiog yra ir neteikia jokios naudos. Taip pat ir su informacija – jos neapdorojus, neišgavus naujų įžvalgų iš turimų duomenų, neišnaudojamas teikiamas potencialas, kuris yra be galo reikalingas efektyviam procesų valdymui, sprendimų priėmimui ir pan. Įvairių rūšių sukaupti ir neapdoroti duomenys yra sunkiai suvokiami žmogui, todėl negali būti tikslingai panaudoti. Šiandien nebepakanka spręsti įvairius paieškos ar statistinius uždavinius, šiandien reikia žinoti, kas bus rytoj, kokių prekių reikės rytoj ar po keleto dienų ir t.t. Duomenų tyrybos technologijos apima daugybę matematinių metodų, kurie skirti surasti objektyvius ir neobjektyvius dėsningumus ir nustatyti ryšius tarp duomenų. Šiuolaikinių informacijos apdorojimo reikalavimų specifika turi įvertinti tai, kad:

- Duomenys yra labai didelės apimties;
- Duomenys yra įvairiarūšiai (tekstiniai, skaitiniai, loginiai, grafiniai ir kt.);
- Duomenų apdorojimo rezultatai turi būti aiškūs bei konkretūs;
- Duomenų apdorojimo priemonės ir įrankiai turi būti paprasti naudojimui.

Duomenų tyrybos algoritmai leidžia duomenis nagrinėti sudėtingesniu lygiu bei išgauti svarbius faktus ir nustatyti įvairius sąryšius. Duomenų tyryba

– tai nežinomų, netrivialių bei praktiškai naudingų ir lengvai interpretuojamų žinių aptikimas chaotiškuose duomenyse. Šios žinios reikalingos sprendimų priėmimui įvairiose veiklos srityse. Informacija, kuri randama taikant duomenų tyrybos metodus, nežinoma iš anksto. Žinias aprašo nauji savybių ryšiai, nusakantys vienų požymių reikšmes pagal kitus nustatytus požymius. Nustatytos naujos žinios turi būti taikomos naujai informacijai sudaryti su tam tikru patikimumo laipsniu. Naujos žinios turi būti suprantamos vartotojui. Pavyzdžiui, žmogus lengvai supranta loginę konstrukciją „Jeigu ..., tai...“. Be to, tokios taisyklės gali būti naudojamos įvairiose duomenų bazių valdymo sistemose kaip SQL užklausos [74, 48]. Tokių uždavinių sprendimui naudojami įvairūs duomenų tyrybos metodai ir algoritmai, t.y. dirbtiniai neuroniniai tinklai, sprendimų medžiai, klasterizavimas, susietumo taisyklių nustatymas ir kiti.

2.2. Susietumo taisyklės

Susietumo taisyklės (*angl. association rules*) leidžia nustatyti sąryšį tarp įvykių ar procesų dėsningumo arba kitaip tariant susieja įvairius įvykių faktus [1, 9, 60, 84]. Susietumo taisyklių paieška yra vienas iš svarbiausių ir plačiai tyrinėjamų duomenų tyrybos uždavinių, kurį pirmasis suformulavo ir pradėjo tyrinėti R. Agrawal [9, 87]. Šio uždavinio tikslas iš duomenų bazių ar kitų duomenų saugyklų išgauti įdomius sąryšius tarp duomenų, dažnus šablonus, duomenų struktūrų susietumą ar atsitiktinumą. Susietumo taisyklių paieška pradinėje duomenų bazėje dažnai vadinama dažnų rinkinių paieška [102], t.y. šioje paieškoje nėra svarbi elementų tvarka rinkinyje.

Pavyzdžiui, pirkėjas, kuris pirko duoną, pirks ir pieną su 72 % tikimybe. Susietumo taisyklė: jei faktas *A* (pirko duoną) yra įvykio dalis, tai yra tikimybė *X* (72 %), kad faktas *B* (pirks pieną) bus to paties įvykio dalis.

Pirmiausiai susietumo taisyklių paieška buvo pritaikyta tipinio pirkėjo krepšelio nustatymui [5, 9, 17, 26, 31, 33, 75, 81, 86], tačiau šis uždavinys

nagrinėjamas ir dabar [76, 90]. Susietumo taisyklių paieška taikoma įvairiuose verslo, finansinių rinkų, draudimo, vartotojų ar klientų elgsenos, telekomunikacijų, genetikos, medicinos ir kt. uždaviniuose.

Tarkime, aibė $I=\{i_1, i_2, \ldots, i_n\}$ sudaryta iš n elementų, D – transakcijų duomenų bazė, kur kiekvieną transakciją T sudaro aibė elementų, priklausančių I, t.y. $T \subseteq I$. Elementų aibė $X \subseteq I$, priklauso transakcijai T tada ir tik tada, jeigu $X \subseteq T$. Elementų aibė X vadinama rinkiniu (*angl. itemset*). Rinkinys, sudarytas iš k elementų vadinamas k – rinkiniu. Rinkinio X dažnumas (*angl. support*), žymimas *supp X*, yra skaičius transakcijų, kuriose šis rinkinys yra transakcijų poaibis. Rinkinys vadinamas dažnu, jei jo pasikartojimas yra nemažesnis už nurodytą minimalų dažnumą *min_supp*.

Apibrėžimas 1. Susietumo taisykle vadinama išraiška $X \Rightarrow Y$, kur rinkiniai $X \subseteq I$, $Y \subseteq I$ ir $X \cap Y = \varnothing$.

Apibrėžimas 2. Susietumo taisyklės dažnumu vadinamas dydis $supp(X \Rightarrow Y) = \frac{supp(X \cup Y)}{|I|}$.

Susietumo taisyklės dažnumo $supp(X \Rightarrow Y)$ reikšmė parodo, kokia elementų aibės I dalis priklauso $X \cup Y$. Dažnumas gali būti išreiškiamas procentais [22].

Apibrėžimas 3. Susietumo taisyklės $X \Rightarrow Y$ patikimumu (*angl. confidence*) vadinamas dydis $conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$, t.y. sąlyginė tikimybė, kad transakcijoje yra rinkinys Y su sąlyga, kad joje yra rinkinys X.

Nustatant susietumo taisykles, apibrėžiami šie dydžiai [50, 64]: minimalus susietumo taisyklės dažnumas *min_supp* ir minimalus patikimumas *min_conf*.

Pavyzdys 1. Turime elementų bei transakcijų aibes. Elementų priklausomybė transakcijoms pateikta 1 lentelėje. Elementų aibę I sudaro 5 elementai.

1 lentelė. Elementų priklausymo transakcijai lentelė.

Transakcija/ Prekė	Vynas	Sausainiai	Pienas	Dribsniai	Sūris
T_1	1	0	0	0	1
T_2	0	1	1	1	0
T_3	1	0	1	1	1
T_4	1	1	1	0	1
T_5	0	1	1	1	1

Susietumo taisyklių dažnumas:

$$supp\,(\{\text{Vynas} \Rightarrow \text{Sūris}\}) = \frac{3}{5};$$

$$supp(\{\text{Sausainiai, Dribsniai}\} \Rightarrow \{\text{Pienas}\}) = \frac{supp\,(\{\text{Sausainiai, Dribsniai}\} \cup \{\text{Pienas}\})}{|I|} =$$

$$= \frac{supp\,\{s_2, s_5\}}{5} = \frac{2}{5}.$$

Susietumo taisyklės patikimumas:

$$conf\,(\{\text{Sausainiai , Dribsniai }\} \Rightarrow \{\text{Pienas }\}) =$$

$$= \frac{supp\,(\{\text{Sausainiai , Dribsniai }\} \cup \{\text{Pienas }\})}{supp\,(\{\text{Sausainiai , Dribsniai }\})} = \frac{supp\,(\{s_2, s_5\})}{supp\,(\{s_2, s_5\})} = 1.$$

2.2.1. Apibendrintos susietumo taisyklės

Nustatant susietumo taisykles laikoma, kad visi analizuojami elementai vienodi (vienarūšiai). Pavyzdžiui, pirkinių krepšelį sudaro prekės (vienodi elementai), kurios turi tuos pačius atributus [37], išskyrus pavadinimą. Jei transakciją papildytume informacija apie prekės priskyrimą tam tikrai prekių grupei, tai būtų galima sudaryti prekių hierarchijos [23] modelį, kurio pavyzdys pateiktas 2 paveiksle.

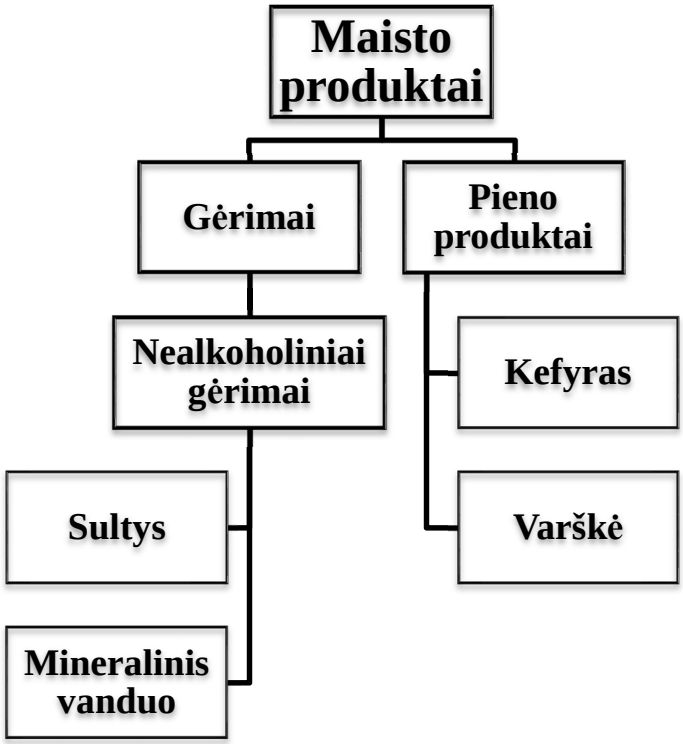
Apibrėžimas 4. Grafas yra aibių pora $G = (V, B)$, kur V – viršūnių (mazgų) aibė, B – viršūnių nesutvarkytų porų $(x, y) = (y, x)$, $x, y \in V$ arba

briaunų (lankų) aibė. Kai poros (*x, y)* laikomos sutvarkytomis, grafas *G* vadinamas orientuotu grafu.

Apibrėžimas 5. Mišku vadinamas neturintis ciklų grafas.

Apibrėžimas 6. Medžiu vadinamas jungus neturintis ciklų grafas arba, medis yra jungus miškas.

Apibrėžimas 7. Elementų hierarchija vadinamas miškas, kurį sudaro orientuoti medžiai.



2 pav. Prekių hierarchijos modelis.

Tarkime, $I=\{i_1; i_2, \ldots, i_n\}$ – elementų aibė, *G* – orientuotų vienakrypčių medžių miškas. *G* lankai nurodo priklausomybes tarp aibės *I* elementų. Aibės *I* elementai išdėstyti pagal hierarchiją. Jeigu lankas nukreiptas iš viršūnės i_k į viršūnę i_m, tai viršūnė i_k vadinama tėvu, o i_m – vaiku (pvz. i_k yra „Pieno produktai“ – atitinka elementų grupę, o i_m yra „Kefyras“ – patį elementą).

Apibrėžimas 8. Išplėstine transakcija vadinama transakcija, kuri papildyta visų aibės *I* elementų tėvais, priklausančiais transakcijai.

Apibrėžimas 9. Apibendrinta susietumo taisykle (*angl. generalized association rule*) vadinama išraiška $X \Rightarrow Y$, kur $X \subset I$, $Y \subset I$ ir $X \cap Y = \varnothing$, ir nei vienas elementas, priklausantis rinkiniui Y nėra nei vieno elemento, priklausančio rinkiniui X, tėvas. Dažnumas ir patikimumas apskaičiuojamas remiantis apibrėžimu 2 ir apibrėžimu 3.

Nagrinėjant išplėstines transakcijas [24] galima nustatyti susietumo taisykles, kurios susieja produktų grupes bei atskirus elementus su produktų grupėmis ir t.t. [101, 103]. Pavyzdžiui, jeigu pirkėjas pirko prekę, kuri priklauso „Nealkoholiniams gėrimams“, tai jis pirko ir prekę, kuri priklauso „Pieno produktams“ arba „Sultims“ ir „Pieno produktams“.

Apibendrintas susietumo taisykles sudaro elementai, priklausantys skirtingiems hierarchijos lygiams. Apibendrintose susietumo taisyklėse elementas $\bar{i}_k$ vadinamas elemento i_k tėvu ir atvirkščiai, elementas i_k vadinamas elemento $\bar{i}_k$ vaiku.

Papildomos informacijos apie elementų grupavimą įvedimo privalumai:

1. Galima nustatyti susietumo taisykles ne tik tarp atskirų elementų, bet ir tarp elementų grupių.
2. Atskiri elementai gali būti nedažni, bet grupė, kuriai priklauso tas elementas gali būti dažna.

Norint nustatyti apibendrintas susietumo taisykles, kiekvieną transakciją reikia papildyti informacija apie elementų grupes, o tai sąlygoja šias problemas:

1. Aukštesnių hierarchijos lygių elementai turi dideles dažnumo reikšmes, lyginant su žemesnių hierarchijos lygių elementais.
2. Transakcijų papildymas grupėmis didina atributų skaičių ir rezultatų aibę. Visa tai uždavinį daro sudėtingesniu bei nulemia didesnį taisyklių skaičių bei perteklinių taisyklių atsiradimą.
3. Perteklinių taisyklių atsiradimas gali prieštarauti apibendrintoms susietumo taisyklėms. Pavyzdžiui, Sultys $\Rightarrow$ Gėrimai. Akivaizdu, kad nėra praktinės tokios taisyklės naudos.

Grupuoti elementus galima ne tik pagal produktų grupes, bet ir pagal kitas charakteristikas, pavyzdžiui pagal kainą, gamintoją ir t.t.

2.2.2. Įdomios apibendrintosios susietumo taisyklės

Sukurta daug algoritmų, kurie skirti nustatyti susietumo taisykles [14, 34, 35, 104]. Susietumo taisyklių skaičiaus nustatymas yra esminė problema [47]. Dažnai surandamas didelis susietumo taisyklių skaičius, kurios gali būti akivaizdžios arba visiškai nenaudingos, t.y. neįdomios [12, 25, 34, 35], todėl gali būti naudojamas apibendrintų susietumo taisyklių įdomumo (*angl. interest*) parametras.

Pavyzdys 2. D – transakcijų aibė, I – aibė elementų, kurie sudaro hierarchinę struktūrą. Reikia nustatyti apibendrintas susietumo taisykles $X \Rightarrow Y$, kai $supp(X \Rightarrow Y) \geq min\_supp$ ir $conf(X \Rightarrow Y) \geq min\_conf$.

Spendžiant šį uždavinį bus surasta daug [38] visiškai nenaudingų apibendrintų susietumo taisyklių. Norint išvengti šios problemos, reikia įvesti naują parametrą taisyklės įdomumo lygmeniui [36, 37, 55, 56, 83, 106] nustatyti.

Tegul $Pr(X)$ – tikimybė, kad visi rinkinio X elementai priklauso vienai išplėstinei transakcijai, $Pr(X \cup Y)$ – tikimybė, kad visi elementai, priklausantys $X \cup Y$ yra transakcijoje. Tada:

$$supp\,(X \Rightarrow Y) = Pr\,(X \cup Y),$$

$$conf\,(X \Rightarrow Y) = Pr\,(Y / X).$$

Pažymime, X, Y – dažnų elementų rinkiniai, x, y – dažni elementai. Tegul $\{x, y\}$ – rinkinys, $\bar{x}$ yra x tėvas, $\bar{y}$ yra y tėvas. Jeigu rinkinio $\{x, y\}$ dažnumas tenkina apibrėžtą minimalų dažnumą *min_supp*, tai rinkinių $\{\bar{x}, y\}, \{x, \bar{y}\}, \{\bar{x}, \bar{y}\}$ dažnumai taip pat tenkins nustatytą minimalaus dažnumo *min_supp* reikšmę. Jeigu susietumo taisyklės $X \Rightarrow Y$ dažnumo ir patikimumo reikšmės tenkina nustatytas *min_supp* ir *min_conf* reikšmes, tai tik susietumo

taisyklės $X \Rightarrow \overline{Y}$ dažnumo ir patikimumo reikšmė tikrai tenkins reikšmes *min_supp* ir *min_conf* [83]. Tuo tarpu taisyklės $\overline{X} \Rightarrow Y$ ir $\overline{X} \Rightarrow \overline{Y}$ gali tenkinti *min_supp*, bet netenkinti *min_conf* reikšmės [83].

Tegul $\overline{X}$ – aibės *X* tėvas, kur *X* ir $\overline{X}$ – aibės elementų, priklausančių hierarchijai (*X*, $\overline{X} \subseteq I$). $\overline{X}$ yra *X* tėvas tik tuo atveju, jei $\overline{X}$ galima gauti iš *X* vieną arba keletą elementų pakeitus jų tėvais, be to *X* ir $\overline{X}$ turi tą patį elementų skaičių [83].

Pavyzdžiui, gali būti dvi aibės (žr. 2 pav.): *X*={Sultys, Kefyras}, $\overline{X}$ = ={Gėrimai, Pieno produktai}. Susietumo taisyklės Gėrimai $\Rightarrow$ Kefyras $(\overline{X} \Rightarrow Y)$, Sultys $\Rightarrow$ Pieno produktai $(X \Rightarrow \overline{Y})$, Gėrimai $\Rightarrow$ Pieno produktai $(\overline{X} \Rightarrow \overline{Y})$ vadinamos susietumo taisyklės Sultys $\Rightarrow$ Kefyras ($X \Rightarrow Y$) tėvais.

Apibrėžimas 10. Susietumo taisyklė $\overline{X} \Rightarrow \overline{Y}$ vadinama susietumo taisyklės $X \Rightarrow Y$ tėvu.

Analogiškus apibrėžimus galima suformuluoti ir susietumo taisyklėms $\overline{X} \Rightarrow Y$ ir $X \Rightarrow \overline{Y}$.

Nagrinėjama susietumo taisyklė $X \Rightarrow Y$. Tegul $Z = X \cup Y$. Elementų rinkinio *Z* dažnumas yra lygus susietumo taisyklės $X \Rightarrow Y$ dažnumui. Tegul reikšmė $E_{\overline{Z}}[Pr(Z)]$ yra tikėtina tikimybės *Pr*(*Z*) reikšmė atitinkanti $Pr(\overline{Z})$. Tegul $Z=\{z_1, \ldots, z_n\}$, $\overline{Z} = \{\overline{z}_1, \ldots, \overline{z}_j, z_{j+1}, \ldots, z_n\}$, $1 \le j \le n$, kur $\overline{z}_i$ yra z_i tėvas. Galima apibrėžti [83]:

$$E_{\overline{Z}}[Pr(Z)] = \frac{Pr(z_1)}{Pr(\overline{z}_1)} \times \ldots \times \frac{Pr(z_j)}{Pr(\overline{z}_j)} \times Pr(\overline{Z}).$$

Analogiškai apibrėžiama, $E_{\overline{X} \Rightarrow \overline{Y}}\left[Pr(Y \mid X)\right]$ tikėtina susietumo taisyklės $X \Rightarrow Y$ patikimumo reikšmė atitinkanti taisyklę $\overline{X} \Rightarrow \overline{Y}$. Tegul $Y=\{y_1, \ldots, y_n\}$, $\overline{Y} = \{\overline{y}_1, \ldots, \overline{y}_j, y_{j+1}, \ldots, y_n\}$, $1 \le j \le n$, kur $\overline{y}_i$ yra y_i tėvas. Galima apibrėžti [83]:

$$E_{\overline{X} \Rightarrow \overline{Y}}[Pr(Y \mid X) = \frac{Pr(y_1)}{Pr(\overline{y}_1)} \times \ldots \times \frac{Pr(y_j)}{Pr(\overline{y}_j)} \times Pr(\overline{Y} \mid \overline{X}),$$

be to $E_{\overline{X} \Rightarrow Y}[Pr(Y \mid X) = Pr(Y \mid \overline{X})$.

Apibrėžimas 8. Susietumo taisyklė $X \Rightarrow Y$ vadinama R – įdomia taisyklės tėvo $\overline{X} \Rightarrow \overline{Y}$ atžvilgiu, jeigu taisyklės $X \Rightarrow Y$ dažnumas yra R kartų didesnis už šios taisyklės tikėtiną dažnumą taisyklės tėvo $\overline{X} \Rightarrow \overline{Y}$ atžvilgiu arba taisyklės $X \Rightarrow Y$ patikimumas yra R kartų didesnis už šios taisyklės tikėtiną patikimumo reikšmę taisyklės tėvo $\overline{X} \Rightarrow \overline{Y}$ atžvilgiu.

Bendru atveju įdomių susietumo taisyklių suradimo uždavinys formuluojamas taip:

Tegul D – transakcijų aibė, I – aibė elementų, kurie susieti hierarchinėmis priklausomybėmis. Reikia rasti išraiškas $X \Rightarrow Y$, kurios yra apibendrintos susietumo taisyklės, kurių dažnumas yra ne mažesnis nei nustatyta minimali dažnumo reikšmė *min_supp*, patikimumas ne mažesnis nei nustatyta minimali patikimumo reikšmė *min_conf* bei taisyklė $X \Rightarrow Y$ yra įdomi.

2.3. Susietumo taisyklių nustatymas

Susietumo taisyklių paieška [34, 85, 91, 92, 93, 108] susideda iš šių etapų:

1. Dažnų elementų ar posekių aibės suradimas, t.y. elementų ar posekių, kurių dažnumas ne mažesnis nei nurodyta *min_supp* reikšmė, aibės radimas.
2. Susietumo taisyklės sudarymas pagal nustatytas dažnų elementų ar posekių aibes.

Parametrų *min_supp* ir *min_conf* reikšmės [6] parenkamos taip, kad apribotų nustatytų susietumo taisyklių skaičių [32, 34, 35]. Jeigu šių parametrų reikšmės yra labai didelės [83, 90], tai algoritmai suras tokias susietumo taisykles, kurios yra akivaizdžios ir gerai žinomos. Jeigu šių parametrų reikšmės bus labai mažos, tai bus generuojamas didelis kiekis susietumo taisyklių [6, 83], o tai reikalauja didelių techninių bei laiko resursų. Taigi, esminė problema – susietumo taisyklių skaičiaus nustatymas [19, 32, 83].

Susietumo taisyklių radimo uždavinys nėra trivialus [73, 98, 99, 100, 107]. Viena iš problemų – dažnų elementų ar požymių rinkinių radimo algoritmų sudėtingumas, nes didėjant elementų ar požymių skaičiui eksponentiškai [20] didėja potencialių elementų ar požymių rinkinių skaičius [12, 13, 39, 97].

Dažnų elementų ar požymių rinkinių nustatymas – tai daug skaičiavimų reikalaujanti operacija [97]. Pats paprasčiausias šio uždavinio sprendimo būdas – tai visų galimų elementų ar požymių rinkinių nustatymas. Tai reikalauja, kad būtų atlikta $O(2^n)$ operacijų, kur n – elementų ar požymių skaičius. Todėl taikoma viena iš dažnumo savybių – bet kurio elementų rinkinio dažnumas negali būti didesnis už minimalų bet kurio poaibio dažnumą.

Pavyzdžiui, trijų elementų rinkinio {Sausainiai, Pienas, Dribsniai} dažnumas visada bus mažesnis arba lygus dviejų elementų rinkinių {Sausainiai, Pienas}, {Sausainiai, Dribsniai}, {Pienas, Dribsniai} dažnumui. Be to, kiekvienoje transakcijoje, kurioje yra elementų rinkinys {Sausainiai, Pienas, Dribsniai}, turi būti ir šie elementų rinkiniai {Sausainiai, Pienas}, {Sausainiai, Dribsniai}, {Pienas, Dribsniai}, tačiau priešingas teiginys nėra teisingas. Ši savybė vadinama antimonotoniškumo savybe.

Antimonotoniškumo savybė gali būti formuluojama ir taip: didėjant elementų skaičiui rinkinyje, rinkinio dažnumas mažėja arba išlieka toks pat. Taigi, bet koks k – elementų rinkinys bus dažnas tada ir tik tada, jei visi jį sudarantys (k-*1*) elementų rinkiniai bus dažni. Visus vieno elemento rinkinius galima įsivaizduoti kaip gardelę, kuri prasideda tuščia aibe, po to pirmame lygyje išsidėstę visi vieno elemento rinkiniai, antrame lygyje išsidėstę *2* – elementų rinkiniai ir t.t., k-ajame lygyje išsidėstę visi k – elementų rinkiniai, kurie yra (k-*1*) – elementų rinkinių poaibis.

k – elementų kandidatų aibės rinkiniai generuojami naudojant (k-*1*) elementų rinkinius. Kandidatų generavimo algoritmas sudarytas iš dviejų žingsnių:

1. Apjungimas. Kiekvienas C_k kandidatas formuojamas iš dažnų (k-1) – elementų rinkinių.
2. Pašalinimas. Perteklinių rinkinių pašalinimui naudojama antimonotoniškumo savybė, t.y. panaikinami iš aibės C_k visi rinkiniai, kuriuose buvo bent vienas nedažnas (k-1) – elementų rinkinys.

Susietumo taisyklės – tai naujos žinios iš sukauptų duomenų, kurios yra suprantamos ir naudingos vartotojui [3, 39].

2.4. Antrojo skyriaus apibendrinimas

Susietumo taisyklių paieška turi platų praktinį pritaikymą įvairiuose genetikos, medicinos, marketingo, finansų ir t.t. uždaviniuose.

Susietumo taisyklių privalumas – tai vartotojui suprantamu pavidalu iš sukauptų duomenų pateiktos naujos ir naudingos žinios.

Susietumo taisyklės gali susieti tiek vienarūšius duomenis, tiek duomenis, priklausančius skirtingiems hierarchijos lygiams.

Susietumo taisyklių paieškoje svarbūs parametrai minimalus taisyklės dažnumas *min_supp* ir minimalus taisyklės patikimumas *min_conf*, kurių parinkimas įtakoja susietumo taisyklių skaičių bei jų naudingumą, tačiau nėra vieningos metodikos šių parametrų reikšmių parinkimui.

Apibendrintų susietumo taisyklių paieškoje gali būti naudojamas įdomumo parametras, kuris leidžia išvengti beprasmių ir neįdomių taisyklių išskyrimo.

Susietumo taisyklių paieška remiasi antimonotoniškumo savybe, t.y. bet kuris k – elementų rinkinys bus dažnas tada ir tik tada, jeigu visi jį sudarantys (k-1) elementų rinkiniai bus dažni.

Susietumo taisyklių paieška susideda iš dviejų žingsnių, t.y. dažnų posekių aibės suradimo ir susietumo taisyklių sudarymo.

3 skyrius. Dažnų posekių paieškos algoritmai

Dažnų posekių ir susietumo taisyklių paieškos algoritmai suskirstyti į dvi grupes: tikslieji algoritmai ir apytiksliai algoritmai [46]. Tikslieji dažnų posekių paieškos algoritmai buvo pradėti kurti nuo 1995 m. Šie algoritmai veikdami turi nuodugniai keletą kartų nuskaityti pradinę duomenų bazę, o tai lemia dideles laiko sąnaudas, tačiau tikslieji algoritmai nepakeičiami, kai reikia pateikti tikslius rezultatus. Šie algoritmai taikomi tokių uždavinių sprendimui, kurių pagrindinis tikslas yra rezultato tikslumas, o ne laiko sąnaudos.

Sparčiai didėjant duomenų bazėms, daugėja dažnų posekių paieškos uždavinių, kuriuose priimtina algoritmų daroma paklaida. Dažnai pagrindinis uždavinio kriterijus yra laikas, tai šią sąlygą gali išpildyti tik apytiksliai algoritmai. Daugelyje veiklos sričių aktualu kuo greičiau gauti atsakymą į klausimą „Koks objektas (ar objektai) yra dažnas?“, o ne gauti tikslų dažno objekto (ar objektų) skaičių duomenų bazėje, todėl gali būti aukojamas tikslumas dėl ženkliai didesnio rezultatų gavimo greičio.

3.1. Tikslieji dažnų posekių paieškos algoritmai

Tikslieji dažnų posekių (rinkinių) paieškos algoritmai kelis kartus skaito pradinę duomenų bazę. Šie algoritmai sudaryti naudojant vieną iš šių metodologijų:

- Apriori principą;
- FP – growth principą;
- Eclat principą.

Transakcijų duomenų bazę sudaro daug skirtingų elementų, iš kurių galima sudaryti didelį kiekį posekių (elementų rinkinių). Agrawal ir Srikant [5] taikė posekių sudarymui nukreiptą žemyn (*angl. downward*) sudarymo principą, t.y. *k* – elementų rinkinys yra dažnas, jei jį sudaro dažni (*k-1*) – elemento posekiai. Šis dažnų rinkinių sudarymo būdas vadinamas Apriori principu. Dažni vieno elemento posekiai nustatomi skenuojant duomenų bazę,

po to iš jų generuojami dviejų elementų posekiai kandidatai bei skenuojama duomenų bazė kandidatų dažnumui nustatyti. Šis procesas kartojamas tol, kol nebegalima iš dažnų k – elementų posekių sugeneruoti dažnų (k+1) – elemento posekių kandidatų.

Apriori algoritmas sudomino mokslininkus, kurie pasiūlė įvairių šio algoritmo patobulinimų: kandidatų generavimui naudoti maišos funkcijas (*angl. hash function*) ir maišos medžius (*angl. hash tree*) [64, 65], dalijimą [75], atrankos metodą [86], dinaminį rinkinių skaičiavimą [17, 18], rinkinių ilgio apribojimą [22], atmetimo metodus [2, 8, 22, 64, 65, 101] ir kt. Apriori algoritmo modifikacijos sumažino duomenų bazės skenavimų skaičių, o tai sutrumpino algoritmo veikimo laiką.

Naudojant Apriori principą, sumažinamas sekų – kandidačių dydis, tačiau generuojama daug sekų – kandidačių bei daug kartų skenuojama pradinė duomenų bazė, norint nustatyti ar seką – kandidatę priskirti dažnų posekių aibei.

GSP (*angl. Generalized Sequence Patterns*) algoritmas [82, 83] paremtas Apriori metodologija, t.y. sekų – kandidatų generavimo, testavimo ir atmetimo strategija. GSP algoritmas yra populiarus dažnų sekų paieškoje bei turi daug patobulinimų, kurie sumažino algoritmo veikimo laiką.

FP – growth (*angl. Frequent pattern - growth*) metodologija [44] dažnų posekių nustatymui nenaudoja sekų – kandidačių generavimo principo. Ši metodologija paremta „skaldyk ir valdyk“ (*angl. divide – and – conquer*) strategija. Pirmojo duomenų bazės skenavimo metu sudaromas dažnų elementų sąrašas, kuriame elementai išdėstyti dažnumo mažėjimo tvarka. Remiantis dažnų elementų sąrašu, duomenų bazė yra paverčiama dažnų posekių medžiu, t.y. FP – medžiu (*angl. FP – tree*), kuriame saugoma elementų susietumo informacija [43, 45]. FP – medis pradedamas sudaryti nuo kiekvieno vieno – elemento ilgio dažno posekio (tai pradinė struktūros dalis), sudaromos priešdėlinės sekos (tai subduomenų bazė, kurią sudaro priešdėlinės sekos ir sąryšiai su pradine struktūra FP – medyje), konstruojamas sąlyginis

FP – medis, kuriame atliekama rekursinė tyryba. Dažni posekiai yra nustatomi iš sąlyginio FP – medžio, apjungus priešdėlines sekas su rekursinės tyrybos rezultatais. FP – growth metodologija dažnų ilgų posekių paieškos uždavinį suskaido į trumpesnių dažnų posekių paieškos uždavinius, o po to apjungia nustatytus dažnus posekius. FP – growth metodologijos taikymas sumažina pradinės duomenų bazės skenavimų skaičių, o tai sutrumpina dažnų posekių paieškos laiką. FP – growth metodologijos algoritmai sekų saugojimui naudoja horizontalų duomenų formatą.

PrefixSpan algoritmo [68] veikimas pagrįstas FP – growth metodologija [67]. Algoritmas identifikuoja priešdėlines sekas (*angl. prefix*), tada visas duomenų bazės sekas suskirsto pagal identifikuotas priešdėlines sekas ir apskaičiuoja sekų dažnumą [46].

LAPIN (*angl. Last Position Induction algorithm*) algoritmas [100] paremtas FP – growth metodologija. Šiame algoritme paskutinė elemento *x* pozicija yra pagrindinis kriterijus, kuris apsprendžia ar dažną *k* – ilgio posekį papildyti elementu *x*, dažnam (*k+1*) posekiui gauti [53].

Zaki [104] pasiūlyta Eclat metodologija tiria duomenis saugomus vertikaliu formatu. Pirmo duomenų bazės skenavimo metu nustatomi visi dažni vieno – elemento posekiai. (*k+1*) – ilgio posekiai sudaromi naudojant *k* – ilgio dažnus posekius, t.y. Apriori principą, tačiau naudojama paieškos į gylį metodologija panaši į FP – growth. Sudarant (*k+1*) – ilgio posekius atsižvelgiama į jo priklausomybę transakcijai.

SPADE (*angl. Sequential Pattern Discovery using equivalence classes algorithm*) algoritme [105] naudojama Eclat metodologija, t.y. vertikalus dažnų sekų saugojimas, todėl sumažėja pradinės duomenų bazės skenavimų skaičius.

SPAM (*angl. Sequential Pattern Mining*) algoritme [4] sekoms išsaugoti naudojamas vertikalus bitmap formatas, o tai nulemia efektyvesnį sekų dažnumo nustatymą pradinėje duomenų bazėje. Šis algoritmas remiasi Eclat metodologija.

PRISM (*angl. Prime – Encoding Based Sequence Mining*) algoritmas [40, 41] naudoja vertikalią blokinę sekų saugojimo ir dažnių skaičiavimo metodologiją, kuri paremta pirminių skaičių faktorizacija.

Šioje disertacijoje eksperimentiniuose tyrimuose buvo naudojami Apriori, GSP, rekursinis ir SPADE algoritmai.

3.1.1. Apriori algoritmas ir jo modifikacijos

Apriori algoritmas, kurį R. Agrawal ir R. Srikant pasiūlė 1994 m. yra vienas pirmųjų dažnų posekių paieškos algoritmų. Šis algoritmas [5] paremtas potencialiai dažnų elementų rinkinių generavimo ir rinkinių kandidatų dažnumo nustatymo strategija.

Susietumo taisyklių paieškos algoritmų pagrindas – dažnų duomenų rinkinių analizė. Pirmiausiai ieškoma dažnų elementų, o po to iš šių elementų generuojami rinkiniai – kandidatai. Norint sutrumpinti susietumo taisyklių paiešką naudojama apriorišKumo savybė [7, 8], t.y. jeigu rinkinys X yra nedažnas, tai šio rinkinio papildymas bet kokiu nauju elementu i_k šio rinkinio X nepadaro dažnu.

Apriori algoritmas tapo vienu iš populiarių posekių paieškos algoritmų [5]. Taip pat yra sukurta Apriori algoritmo patobulinimų [59] bei įvairių taikymų [3, 11]. Pirmojo Apriori algoritmo žingsnio metu nustatomi dažni vieno elemento rinkiniai. Vykdant šį algoritmo žingsnį skenuojama visa duomenų bazė ir nustatoma, kiek kartų kiekvienas elementas yra sutinkamas duomenų bazėje ir tolimesniam apdorojimui naudojami tik tie elementai, kurie tenkina nustatytą minimalų pasirodymų dažnumą *min_supp*.

Kiti algoritmo žingsniai susideda iš dviejų dalių: potencialiai dažnų elementų rinkinių generavimo, kurie vadinami kandidatais ir rinkinių kandidatų dažnumo nustatymo [4].

Apriori algoritmas generuoja kito žingsnio elementų rinkinius kandidatus tik iš rastų dažnų rinkinių prieš tai atliktame žingsnyje, t.y.

taikomas Apriori principas, kad bet kuris dažnas elementų rinkinio poaibis turi būti dažnas rinkinys. Todėl rinkiniai kandidatai, sudaryti iš k elementų, generuojami sujungiant dažnus elementų rinkinius, turinčius (k–1) elementų, kurie tenkina minimalų pasikartojimų skaičių.

Šiame algoritme svarbi kandidatų generavimo funkcija. Vykdant kandidatų generavimą, nesikreipiama į duomenų bazę. Norint gauti k – elementų rinkinius, naudojami (k-1) – elementų rinkiniai, kurie buvo dažni ankstesniame žingsnyje. Kiekvienas kandidatas C_k konstruojamas papildant dažną (k-1) – elementų rinkinį kitu dažno (k-1) – elementų rinkinio elementu [7, 9].

Atlikus elementų rinkinių generavimą, tikrinama, kurie nauji kandidatai tenkina nustatytą minimalų pasirodymų dažnį. Akivaizdu, kad dažnų elementų rinkinių gali būti labai daug, todėl reikalingas efektyvus būdas šiems rinkiniams suskaičiuoti. Pats paprasčiausias būdas – sekos elementus lyginti su kiekvienu sugeneruotu kandidatu. Tačiau toks sprendimas užima daug laiko. Greitesnis ir efektyvesnis sprendimas – tai grafų, medžių naudojimas [15, 16, 29, 42, 52].

Maišos medis (*angl. hash tree*) [49, 65, 66, 78] konstruojamas kiekvieną kartą, kai generuojami kandidatai. Iš pradžių medis turi tik šaknį (šaknis apibrėžiama pirmajame lygmenyje), kuri tampa medžio lapu ir neturi jokių kandidatų rinkinių. Medžio mazgas gali būti rinkinių sąrašas (mazgo lapas) arba lentelė (vidinis mazgas). Kiekvienas vidinis mazgas, kurio lygmuo d, turi šakas į kitus medžio mazgus, kurių lygmuo d+1. Kai generuojamas naujas kandidatas, jis prijungiamas prie mazgo. Kai mazgo lapų skaičius viršija nurodytą slenkstį, mazgas paverčiamas maišos lentele (vidiniu mazgu) ir šiam mazgui sukuriami lapai. Visi kandidatai pasiskirsto mazguose pagal elementų, įeinančių į rinkinį, maišos reikšmę. Kiekvienas naujas kandidatas generuojamas vidiniame mazge, o saugomas mazgo lape. Taip sukuriamas elementų sekų kandidatų medis. Naudojant šį medį nesunku rasti kiekvieno kandidato pasikartojimų skaičių. Pradedama nuo šaknies ir randami visi kandidatai, kurie sutampa su transakcijos T_i elementais, t.y. $C_k \cap T_i = C_k$.

Pirmajame lygmenyje, t.y. medžio šaknyje maišos funkcija taikoma kiekvienam transakcijos elementui. Antrajame lygmenyje maišos funkcija taikoma antrajam elementui ir t.t., *k* – ajame lygmenyje maišos funkcija taikoma *k*-elementui [51]. Tai atliekama tol, kol pasiekiamas medžio lapas. Po to, kai kiekviena transakcija „pereina“ per maišos medį, tikrinama, ar gauta reikšmė tenkina nustatytą minimalų pasirodymų dažnį. Kandidatai, kurie tenkina nustatytą minimalų pasirodymų dažnį, priskiriami dažnoms sekoms [4].

Apriori algoritmas turi daug patobulinimų ir modifikacijų [57, 62, 63]. Algoritmas AprioriAll kiekvieną kartą, kai analizuojama duomenų bazė, naudoja posekius, kurie gauti priešpaskutiniame analizavime [61]. Tada generuojamos sekos – kandidatai bei atliekamas jų dažnio skaičiavimas, analizuojant duomenų bazę. Dažni rinkiniai, kurie aptinkami tikrinant pirmą kartą yra vieno elemento posekiai. Dažnai šis procesas yra vadinamas inicializacija [96]. Šis algoritmas formuoja sekas iš visų galimų ilgių posekių. Tačiau jeigu tam tikro ilgio posekių dažnis yra mažesnis už nustatytą minimalų dažnumą *min_supp*, tai šio ilgio posekiai praleidžiami. Algoritmas, kaip parametrą naudoja posekių ilgius, kurie buvo analizuojami prieš tai buvusiame žingsnyje ir grąžina posekių ilgius, kurie bus analizuojami kitame žingsnyje [50, 95].

Pažymime, duomenų bazės skenavimo numerį – *s*, posekio ilgį – *k*, tai tada $k(s+1)=k(s)+v$. Tai reiškia, kad kitame žingsnyje bus analizuojami posekiai, kurie yra ilgesni *v* dydžiu. Tuo atveju, kai $v=1$, tai algoritmas yra analogiškas Apriori algoritmui, t.y. bus analizuojami visų ilgių posekiai. AprioriAll algoritmo tikslas – apibrėžti kokio ilgio posekiai gali būti praleisti.

Jeigu *k* – ajame tikrinime bus nustatyta, kad dalinių sekų posekis F_{k-1} yra tuščias, tai suformuoti galimų kandidatų posekį C_k bus neįmanoma. Tada norint suformuoti C_k galima naudoti kandidatų aibę C_{k-1} [80, 81].

Algoritmas AprioriSome duomenų bazės analizavimo metu apdoroja tik apibrėžto *k* ilgio posekius [6, 8].

Algoritmo DymanicSome inicializavimo metu apskaičiuojami visi k ilgio posekiai kandidatai. Po to apdorojami posekiai, kurių ilgis yra k kartotinis [86].

Apriori DynamicSome algoritmas, kai yra maža *min_supp* reikšmė, generuoja labai daug sekų kandidačių. Šio algoritmo laiko sąnaudos yra didesnės negu AprioriSome algoritmo [83]. Visų Apriori algoritmų vykdymo laikas didėja, kai mažinamas dažno posekio dažnumas *min_supp*, nes tuo atveju didėja dažnų posekių kiekis. Pagrindinis algoritmo AprioriSome pranašumas lyginant su algoritmu AprioriAll yra tai, kad išvengiama trumpesnių nei nurodyto ilgio posekių skaičiavimo [83]. Tačiau šis privalumas sumažėja dėl dviejų priežasčių. Pirmiausiai, kandidatai C_k yra aibės F_{k-1} poaibiai ir kandidatų skaičius, kurį generuoja algoritmas AprioriSome gali būti didesnis. Antra priežastis, nors algoritmas AprioriSome praleidžia kai kurių ilgių posekių analizę, mažesnio ilgio nei nurodytas ilgis posekiai vis tiek generuojami.

3.1.2. GSP algoritmas

GSP algoritmas [82] buvo pasiūlytas Rakesh Agrawal ir Ramakrishnan Srikant 1996 m. GSP algoritmas veikia pagal Apriori principą, t.y. norint gauti k – elementų rinkinius, naudojami (k-1) elementų rinkiniai, kurie buvo dažni ankstesniame žingsnyje ir tenkina vartotojo nustatytą minimalų dažnumą *min_supp*. Kiekvienas kandidatas C_k konstruojamas papildant dažną (k-1) elementų rinkinį kitu dažno (k-1) elementų rinkinio elementu. Šio algoritmo pagrindinis uždavinys nustatyti, kokios sekos yra tikrai nedažnos ir jų toliau netikrinti. Taigi, GSP algoritmas sudarytas iš šių pagrindinių žingsnių:

- kandidatų generavimas,
- kandidato dažnumo tikrinimas,
- kandidato priskyrimas dažnų posekių aibei arba jo atmetimas.

Tarkime, kad aibė $I=\{i_1, i_2, \ldots, i_n\}$, sudaryta iš n elementų.

Nagrinėjama duomenų bazė *D*, kuri sudaryta iš įvairių aibės *I* elementų kombinacijų. Reikia rasti dažnus posekius. Pirmiausia tikrinami pirmojo lygmens posekiai. Tokių posekių yra *n*: (i_1, i_2, …, i_n). Nustačius jų dažnius, tikrinami antrojo lygmens posekiai. Šių posekių skaičius yra n^2: (i_1i_1, i_1i_2,…i_1i_n, i_2i_1, …, i_2i_n, …, i_ni_1, …i_ni_n). Tačiau dabar tikrinami ne visi posekiai. Kuriuos posekius tikrinti, sprendžiama pagal prieš tai buvusį lygmenį. Jeigu į antrojo lygmens posekį įeina nedažnas pirmojo lygmens posekis, tai antrojo lygmens posekis irgi yra nedažnas ir jį galima atmesti toliau netikrinant.

Taip pereinama prie kito lygmens, kuris buvo sukurtas iš prieš tai buvusio antrojo lygmens. Trečiajame lygmenyje bus n^3 kombinacijų, tačiau tikrinami ne visi posekiai, o tik tie, kuriuose nėra nedažnų antrojo lygmens posekių ir t.t. Vadinasi, bus tikrinamos ne visos kombinacijos, bet tik tos kombinacijos, kurios yra prieš tai buvusio lygmens dažnų sekų posekiai [86, 87].

GSP algoritmas tapo populiariu ir naudojamu algoritmu dažnų sekų paieškoje [61] bei sulaukė daugelio patobulinimų, sumažinusių algoritmo veikimo laiką. GSP algoritmo patobulinimas galimas naudojant maišos medžius [64, 65], dinaminį posekių skaičiavimą [18] ir kitas metodologijas. Nors daugelis iš siūlomų patobulinimų žymiai sumažino generuojamų pagal Apriori principą kandidatų kiekį, tačiau GSP algoritmas vis dar turi du pagrindinius trūkumus:

1. Generuojamas didelis kiekis posekių – kandidatų, naudojant Apriori principą;

2. Pakartotinai skenuojama duomenų bazė ir tikrinamas minimalus posekio – kandidato dažnumas, o tik tada posekis priskiriamas dažnų posekių aibei arba atmetamas. Tai reikalauja daugybinio duomenų bazės nuskaitymo, o tai nulemia dideles laiko sąnaudas.

3.1.3. Rekursinis algoritmas

Rekursija yra plačiai naudojama apibrėžiant matematinius objektus, sudarant bei realizuojant įvairiausius algoritmus. Rekursiją [54] sudaro du pagrindiniai žingsniai:

- Objektas, kuris priklauso nuo parametro, yra apibrėžiamas naudojant tą patį objektą ar objektus, tik su kitomis parametro reikšmėmis.
- Nurodoma rekursijos pabaiga ir nustatomos pradinės sąlygos. Šių sąlygų skaičius sutampa su rekursijos gyliu.

Rekursinis algoritmas sekų – kandidačių saugojimui naudoja vertikalų formatą. Pirmajame žingsnyje iš dažnų vieno elemento posekių sudaroma seka $\{i_1, i_2, ... , i_n\}$. Iš šios sekos kiekvieno elemento generuojami antrojo lygmens posekiai $\{i_1i_1, i_1i_2, ..., i_1i_n, i_2i_1, i_2i_2, ..., i_2i_n, ..., i_ni_1, i_ni_2, ..., i_ni_n\}$. Tada nustatomas kiekvieno sugeneruoto posekio dažnumas. Jeigu posekis yra nedažnas lygyje n, tai iš jo daugiau negeneruojama $n+1$ lygio sekų-kandidačių.

3.1.4. SPADE algoritmas

SPADE algoritmas naudoja vertikalų transakcijų numerių *id* saugojimo formatą [105]. Tarkime, kad aibė $I=\{i_1, i_2, ... , i_n\}$, sudaryta iš n elementų, kurie išdėstyti abėcėlės tvarka. Įvykiu vadinama netuščias, nesutvarkytas leksografine tvarka elementų i_k rinkinys. Seka yra sutvarkytas įvykių rinkinys. Įvykis žymimas $(i_1\ i_2\ ...i_k)$, kur i_j yra aibės I elementai. Seka S žymima taip: $(s_1 \rightarrow s_2 \rightarrow ... \rightarrow s_k)$, kur s_i yra įvykis. Seka, sudaryta iš k elementų yra vadinama k – seka. Jeigu įvykis s_i įvyko anksčiau nei įvykis s_j, tai kiekvienai sekai S_k žymime $s_i < s_j$. Jeigu seka S_i yra kitos sekos S_j posekis, tai žymima $S_i \leq S_j$, tai egzistuoja funkcija f, išsauganti sąryšį tarp įvykių posekyje S_i ir sekoje S_j, t.y. $s_i \subseteq f(s_i)$ ir jeigu $s_i < s_j$, tai $f(s_i) < f(s_j)$.

Duomenų bazė *D* yra sudaryta iš įvesties sekų rinkinių. Kiekviena įvesties seka *C* duomenų bazėje *D* turi unikalų identifikatorių *sid*, ir kiekvienas įvykis turi taip pat unikalų identifikatorių *eid*.

Įvesties seka *C* turi posekį S_i, jeigu $S_i \leq C$. Sekos S_i dažnumas *supp* yra lygus įvesties sekų, kurių posekis yra S_i, skaičiui duomenų bazėje D. Dažna k – seka yra F_k. Dažna seka yra maksimalaus ilgio, jei ji nėra jokios kitos dažnos sekos posekis.

SPADE algoritmo pirmojo žingsnio metu skenuojama duomenų bazė ir nustatomi dažni vieno – elemento posekiai. Antrojo algoritmo žingsnio metu sudaromos dviejų – elementų sekos horizontalų sekų saugojimo formatą transformuojant į vertikalų formatą. Sudaromos sekų ir įvykių poros (*sid, eid*), pagal kurias sukuriamos įvesties sekos bei nustatomas jų dažnumas [105].

Kitų algoritmų žingsnių metu formuojamos dažnos *n* – elementų sekos, sujungiant pagal transakcijų numerių *id* sąrašą dažnas (*n*-1) – elementų sekas, *id* sąrašo dydis nurodo jam priklausančių sekų skaičių. Jei šis skaičius yra didesnis už nurodytą *min_supp* reikšmę, tai seka yra dažna. Algoritmo vykdymas baigiamas, kai sudarytos *n* – elementų sekos netenkina nurodytos *min_supp* reikšmės. SPADE algoritmas gali naudoti tiek paieškos į gylį (*angl. depth – first search*), tiek paieškos į plotį (*angl. breadth – first search*) metodus dažnų sekų paieškai [104, 105].

3.1.5. Tiksliųjų algoritmų greičio palyginimas

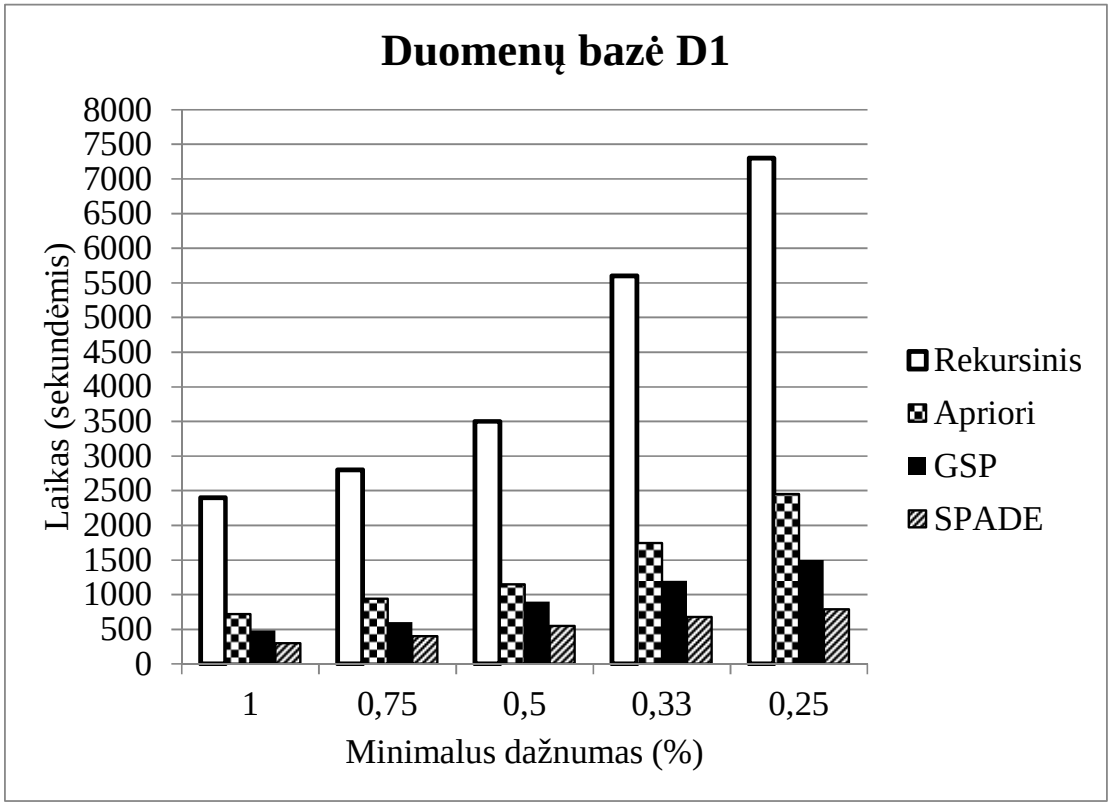
Apriori, GSP, SPADE ir rekursinio algoritmų greičio palyginimui naudojamos dvi sugeneruotos duomenų bazės D1 ir D2.

Duomenų bazę D1 sudaro 200000 simbolių. D1 sudaro elementai A, E, I, N, S, kurie pasiskirstę su vienodomis tikimybėmis.

Duomenų bazę D2 sudaro 200000 simbolių. D2 sudaro elementai A, E, I, N, kurie pasiskirstę su vienodomis tikimybėmis.

Algoritmų veikimo greičio palyginimo eksperimentui naudotas kompiuteris su 2.5 GHz Intel(R) Core(TM) i5-3210 procesoriumi, 4GB RAM atminties, HDD 320 GB. Visi eksperimente lyginami algoritmai realizuoti Object Pascal programavimo kalba.

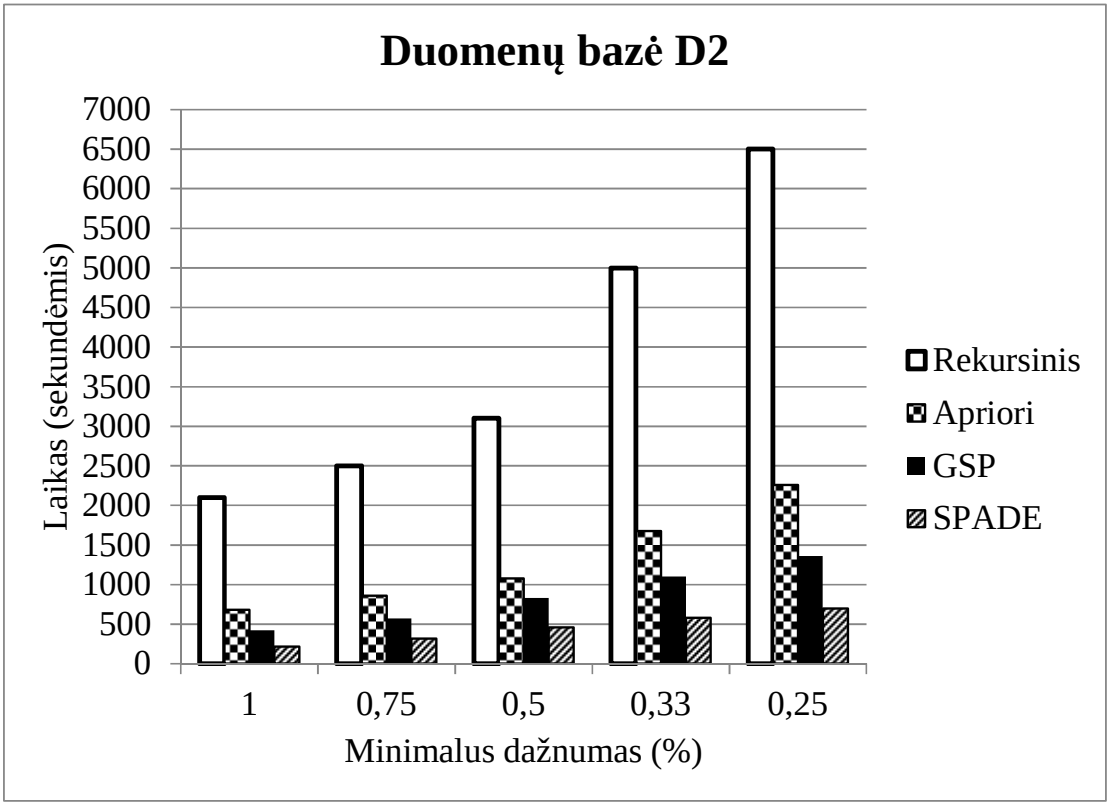
Duomenų bazės D1 ir D2 buvo apdorotos Apriori, GSP, SPADE ir rekursiniu algoritmais, kai *min_supp* reikšmės 1%, 0,75%, 0,5%, 0,33%, 0,25% (*min_supp* = 2000, 1500; 1000; 650; 500).



3 pav. Tiksliųjų algoritmų veikimo trukmė duomenų bazėje D1.

3 paveiksle pateiktas algoritmų laiko palyginimas, esant skirtingoms minimalaus dažnumo reikšmėms *min_supp* duomenų bazėje D1, o 4 paveiksle pateiktas algoritmų laiko palyginimas, esant skirtingoms minimalaus dažnumo reikšmėms *min_supp* duomenų bazėje D2.

Pagal apdorojimo trukmę greičiausias yra SPADE algoritmas, o ilgiausiai veikia rekursinis algoritmas. Visų algoritmų veikimo laikas trumpėja didėjant minimalaus dažnumo reikšmei *min_supp* bei mažėjant skirtingų duomenų bazės elementų skaičiui.



4 pav. Tiksliųjų algoritmų veikimo trukmė duomenų bazėje D2.

3.2. Apytiksliai dažnų posekių paieškos algoritmai

Apytiksliai dažnų posekių paieškos algoritmai yra greitesni už tiksliuosius paieškos algoritmus, nes dažniausiai taikomi ne visai duomenų bazei, bet daug mažesnei duomenų bazės imčiai. Apytikslių algoritmų strategijos remiasi MRA (*angl. Multi Resolution Analysis*), PAC (*angl. Probably Approximate Correct*) bei Shannon pasirinkimo teoremomis [21].

3.2.1. ApproxMAP algoritmas

ApproxMAP algoritmas (*angl. Approximate Multiple Alignment Pattern mining*) [58] vietoj tikslių dažnų posekių paieškos pradinėje duomenų bazėje, identifikuoja posekius, kurie dažnai naudojami daugelyje kitų posekių.

Šis algoritmas kelis kartus skaito pradinę duomenų bazę, ieškodamas apytikslių posekių.

Pažymime $S_1 = \{s_{11}, s_{12}, ..., s_{1n}\}$ ir $S_2=\{s_{21}, s_{22}, ..., s_{2m}\}$ – posekių šablonai, kur $s_{11}, s_{12}, ..., s_{1n}$ ir $s_{21}, s_{22}, ..., s_{2m}$ – visi posekių elementai.

Apibrėžiamas hierarchinis atstumas A tarp posekių šablonų S_1 ir S_2 taip:

$$A(i, j) = \min\begin{cases} A(i-1, j) + INDEL(s_{1i}) \\ A(i, j-1) + INDEL(s_{2j}) \\ A(i-1, j-1) + RELP(s_{1i}, s_{2j}), \end{cases}$$

kai $1\leq i \leq n$, $1\leq j \leq m$, $INDEL()$ – įterpimo arba pašalinimo operatorius, $REPL()$ – keitimo operatorius.

Normalizuotas hierarchinis atstumas tarp S_1 ir S_2 apibrėžiamas taip:

$$dist(S_1, S_2) = \frac{A(n,m)}{\max\{\|S_1\|, \|S_2\|\}}.$$

Naudojant normalizuotą hierarchinį atstumą, posekių grupavimui gali būti taikomas tankumu pagrįstas grupavimo algoritmas. Posekis vadinamas tankiu, jei duomenų bazėje D yra daug į jį panašių posekių. Tarkime, kiekvienam duomenų bazės posekiui S_i reikšmės $d_1, d_2, ..., d_k$ sudaro posekį, kai $dist(S_i, S_j)$ reikšmės yra mažiausios bei $S_i \neq S_j$.

Tankumas apibrėžiamas taip:

$$Density(S_i) = \frac{n}{\|s\|d},$$

kur $d = max\{d_1, d_2, ..., d_k\}$, $n=||S_j \in |dist(S_i, S_j) \leq d|\}||$, n yra k – artimiausių posekių skaičius.

ApproxMAP algoritmas susideda iš šių žingsnių [58]:

1. Duomenų bazės posekių grupavimas pagal panašumą. Grupavimo algoritmo įvestis yra posekis $S=\{S_i\}$ ir apibrėžtas k – artimiausių posekių skaičius. Algoritmo išvesties reikšmė yra posekių grupės $\{C_i\}$, kur kiekviena grupė yra posekių rinkinys. Grupavimą sudaro trys žingsniai:

a. Kiekvienas posekis apibrėžiamas kaip grupė. Kiekvieno posekio S_i tankumas prilyginimas grupės tankumo reikšmei;

b. Apjungiami artimiausi tankūs posekiai į grupę bei įvertinamas naujos grupės tankis.

c. Apjungiamos grupės pagal tankumą. Visi posekiai, kurie neturi artimiausių posekių, kurių tankumas didesnis už sekos S_i tankumą, bet yra artimos sekai S_j, kurios tankumas lygus S_i, apjungiamos į dvi grupes C_{S_j} ir C_{S_i}. Šioms grupėms priklauso visos sekos, kurios tenkina šią nelygybę:

$$Density\,(C_{S_j}) > Density\,(C_{S_i}).$$

2. Posekių palyginimas ir šablonų generavimas. Sugrupavus posekius, kurie yra panašūs vieni į kitą, atliekamas grupės posekio šablono nustatymas. Pirmame algoritmo žingsnyje buvo atliktas posekių tankumo apskaičiavimas, todėl dabar reikia visus posekius išrikiuoti mažėjimo tvarka pagal jų tankumą. ApproxMAP algoritmas išskiria dažnai aptiktą svertinę seką $WS=\langle s_{11}{:}v_1, s_{12}{:}v_2, ..., s_{1l}{:}v_l\rangle{:}n$, kur n – sekos svorio reikšmė. Seka v_i, sudaryta iš elementų s_{1i}, kai $1\le i\le l$. Palyginamas elementas s_{ji} yra išreiškiamas tokia forma: $s_{ji}=(s_{j1}{:}\ w_{j1}, s_{j2}{:}\ w_{j2}, ..., s_{jm}{:}\ w_{jm},)$.

3. Kiekvienos grupės ilgiausio apytikslio posekio šablono generavimas. Norint išskirti šablonus, sudaroma svertinė seka kiekvienai posekių grupei. Tada sukuriamas geriausiai grupės posekius atspindintis maksimalaus ilgio šablonas, naudojant seką *WS*. Šablonai taip pat gali būti generuojami išrenkant posekių dalį, kuri yra daugumoje tos grupės posekių. Sekai

$WS=\langle(s_{11}{:}w_{11}, s_{12}{:}w_{12}, ..., s_{1m}{:}w_{1m}){:}v_1, ..., (s_{l1}{:}w_{l1}, s_{l2}{:}w_{l2}, ..., s_{lm}{:}w_{lm}){:}v_l\rangle{:}n$,

i-tojo elemento $s_{ij}{:}w_{ij}$ ilgis l apibrėžiamas taip: $\frac{s_{ij}}{n}\cdot 100\%$.

Žinoma, elementas, kurio ilgis didžiausias, yra keliuose tos grupės posekiuose. Gali būti nurodomas ilgis ($0 \le min\_l \le 1$). Šablonas gali būti gaunamas iš sekos *WS* pašalinus elementus, kurių ilgiai mažesni nei nurodyta *min_l* reikšmė.

3.2.2. Tikimybinis dažnų sekų nustatymo algoritmas ProMFS

Vienas iš apytikslių tikimybinių dažnų posekių paieškos algoritmų yra ProMFS (*angl. probabilistic algorithm for mining frequent sequences*) [88, 89].

ProMFS algoritmas, remdamasis tikimybinėmis charakteristikomis, kurios apibūdina elementų pozicijas pagrindinėje sekoje generuoja naują žymiai trumpesnę modelinę seka, kuri analizuojama GSP algoritmu bei nustato dažnus posekius naujojoje sekoje ir daro išvadas apie dažnus posekius pradinėje duomenų sekoje.

Tikimybinis dažnų sekų nustatymo algoritmas ProMFS [88, 89] paremtas šiomis statistinėmis pagrindinės sekos charakteristikomis:

1. Elemento pasirodymo pagrindinėje sekoje tikimybė.

$P(i_j) = \frac{V(i_j)}{l}$ yra elemento i_j pasirodymo pradinėje sekoje tikimybė, kur $i_j \in I$, $j=1, \ldots, n$; $I=\{i_1, i_2, \ldots, i_n\}$ yra aibė, kuri sudaryta iš n skirtingų elementų; $V(i_j)$ – elementų i_j skaičius pagrindinėje sekoje S; l – sekos ilgis. Be to, $\sum_{j=1}^{n} P(i_j) = 1.$

2. Sąlyginė tikimybė, kad vienas elementas eis po kito.

$P(i_j \mid i_v)$ – sąlyginė tikimybė, kad elementas i_v pasirodys po elemento i_j, $i_j, i_v \in I$; $j, v = 1, \ldots, n$. Be to, $\sum_{j=1}^{n} P(i_j \mid i_v) = 1$ visiems $j = 1, \ldots, m$.

3. Atstumo tarp dviejų elementų pagrindinėje sekoje vidurkis.

$D(i_j \mid i_v)$ – atstumas tarp elemento i_j ir i_v , kur $i_j, i_v \in I$, $j, v = 1,..., n$. $D(i_j \mid i_v)$ yra elementų skaičius tarp i_j ir pirmojo surasto elemento i_v , ieškant nuo i_j iki pagrindinės sekos pabaigos. Atstumas tarp dviejų kaimyninių elementų sekoje yra lygus vienetui.

4. Atstumų vidurkių matrica $\tilde{A}$. Atstumų vidurkių matrica sudaryta iš elementų a_{jv}=*Average* ($D(i_j \mid i_v), i_j, i_v \in I$), $j, v = 1,..., n$.

Išvardintos tikimybinės charakteristikos apibūdina elementų pozicijas pagrindinėje sekoje, todėl remiantis šiomis charakteristikomis bei atstumų vidurkių matrica $\tilde{A}$, generuojama nauja žymiai trumpesnė modelinė seka $\underline{C}$, kurios ilgis lygus l. Modelinės sekos $\underline{C}$ elementai žymimi c_r, $r = 1,...,l$. Modelinė seka $\underline{C}$ sudaryta iš visų aibės I elementų. Kiekvienam modelinės sekos $\underline{C}$ elementui c_r apibrėžiama skaitinė charakteristika $Q(i_j, c_r)$, $r = 1,...,l$, $j = 1,...,m$. Iš šių skaitinių charakteristikų sudaroma matrica, kurios pradinės reikšmės yra lygios nuliams. Tada atliekama pagrindinės sekos analizė, algoritmą papildant funkcija $f(c_r, a_{rj})$. Ši funkcija padidina charakteristikų $Q(i_j, c_r)$ reikšmes vienetu. Elementas $c_r \in I$, $r = 1, ..., l$ modelinėje sekoje $\underline{C}$ nustatomi pagal maksimalią reikšmę $max(P(i_j))$, $i_j \in L$. Tada vykdoma funkcija:

$f(c_r, a_{rj}) \Rightarrow Q(i_j, 1+a_{rj}) = Q(i_j, 1+a_{rj})+1$, $j = 1, ..., m$, $r = 1, ..., l$. Jeigu su tam tikromis reikšmėmis p ir t gaunama, kad $Q(i_p, c_r) = Q(i_t, c_r)$, tai elementas c_r parenkamas pagal maksimalią sąlyginių tikimybių reikšmę, t.y. $max(P(c_{(r-1)}|i_p), P(c_{(r-1)}|i_t))$.

Jeigu $P(c_{(r-1)}|i_p) > P(c_{(r-1)}|i_t)$, tai $c_r=i_p$, ir $c_r=i_t$, jeigu $P(c_{(r-1)}|i_p)<P(c_{(r-1)}|i_t)$.

Jeigu $P(c_{(r-1)}|i_p)=P(c_{(r-1)}|i_t)$, tai $c_r=max(P(i_p), P(i_t))$.

Po to gauta nauja seka analizuojama GSP algoritmu arba kokiu nors kitu tiksliu algoritmu. Gauti dažni posekiai modelinėje sekoje bus dažni posekiai ir pagrindinėje sekoje. Šiame algoritme galima vietoj vidurkių matricos $\tilde{A}$ naudoti dažniausių atstumų matricą Ñ.

ProMFS algoritmas remiasi tik empiriniais bandymais ir stebėjimais [88, 89], kaip algoritmas veikia skirtingose duomenų bazėse, tačiau neturi teorinio algoritmo daromų paklaidų įvertinimo.

3.2.3. Apytikslis atsitiktinės imties metodas RSM

Atsitiktinės imties metodas RSM (*angl. Random Sampling Method*) [69, 70] analizuoja ne visą pradinę duomenų seką, o daug trumpesnę jos atsitiktinę imtį. Generuojama atsitiktinio dydžio su vienodomis tikimybėmis seka.

Pradinės sekos atsitiktinė imtis $\underline{S}$ sudaroma taip:

1. Generuojamos atsitiktinio dydžio η_i, įgyjančio reikšmes 1, 2, ..., n su vienodomis tikimybėmis $\frac{1}{n}$, realizacijų seka η_1, η_2, ..., η_n.
2. Ieškant pirmojo lygio, t.y. vieno elemento dažnų posekių, atsitiktinė imtis $\underline{S}$ elementams i_i yra $S_{\eta_1}, S_{\eta_2}, ..., S_{\eta_k}$. Antrojo lygio atsitiktinė imtis elementų poroms $i_i i_j$ yra $(S_{\eta_1}, S_{\eta_1+1}), (S_{\eta_2}, S_{\eta_2+1}), ..., (S_{\eta_k}, S_{\eta_k+1})$. k-ojo lygio atsitiktinė imtis elementų rinkiniams $i_i ... i_k$ yra tokia:

$(S_{\eta_1}, ..., S_{\eta_1+k-1}), (S_{\eta_2}, ..., S_{\eta_2+k-1}), ..., (S_{\eta_n}, ..., S_{\eta_n+k-1})$ ir t.t. Tokia imtis yra sudaryta grąžintiniu ėmimu, nes kai kurie skaičiai η_i gali pasikartoti. Negrąžintiniu ėmimu sudaryta atsitiktinė imtis formuojama iš pasikartojančių skaičių η_i pašalinant visus pasikartojančius skaičius bei papildomai generuojant naujus skaičius, kol bus gautas nesikartojančių skaičių rinkinys η_1, η_2, ..., η_n.

Tada pasinaudojus bet kuriuo tiksliuoju dažnų posekių paieškos algoritmu nustatomi dažnų posekių $s_{i_1}, s_{i_2}, ..., s_{i_k}$ empyriniai dažniai atsitiktinėje imtyje $\underline{S}$:

$$\overline{p}_n(s_{i_1}, s_{i_2}, ..., s_{i_k}) = \frac{k\{j : S_{\eta_j} = s_{i_1}, S_{\eta_j+1} = s_{i_2}, ..., S_{\eta_j+k-1} = s_{i_k}\}}{n}.$$

Pasirenkamas skaičius $\delta > 0$ ($0 < \varepsilon - \delta < \varepsilon + \delta < 1$), k=1, 2, ... Posekiai $s_{i_1}, s_{i_2}, ..., s_{i_k}$ skirstomi į tris grupes:

1) dažni posekiai, jeigu $\overline{p}_n(s_{i_1}, s_{i_2}, ..., s_{i_k}) \geq \varepsilon + \delta$;

2) reti posekiai, jeigu $\overline{p}_n(s_{i_1}, s_{i_2}, ..., s_{i_k}) \leq \varepsilon + \delta$;

3) tarpiniai posekiai, jeigu $\overline{p}_n(s_{i_1}, s_{i_2}, ..., s_{i_k}) \in (\varepsilon - \delta, \varepsilon + \delta)$.

RSM metodas turi teorinį paklaidų įvertinimą [69, 70]. Fiksuojamas koks nors posekis $s_{i_1}, s_{i_2}, ..., s_{i_k}$. Galimos dviejų rūšių klaidos:

1) posekis priskirtas dažnų posekių klasei, tačiau iš tikrųjų jis yra retas;

2) posekis priskirtas retų posekių klasei, tačiau iš tikrųjų jis yra dažnas.

Pažymima $\overline{p}_n = \overline{p}_n(s_{i_1}, s_{i_2}, ..., s_{i_k})$ ir $p = p(s_{i_1}, s_{i_2}, ..., s_{i_k})$. Tada pirmos rūšies klaidos tikimybė neviršija $P(\overline{p}_n - p) > \delta$, o antros rūšies klaidos tikimybė neviršija $P(\overline{p}_n - p) < -\delta$. Vertinant šias tikimybes, pasinaudojama atsitiktiniais dydžiais, kurie apibrėžiami taip: atsitiktinis dydis Ω_1=1, jeigu $S_{\eta_i} = s_{i_1}, S_{\eta_i+1} = s_{i_2}, ..., S_{\eta_i+k-1} = s_{i_k}$, i =1, ..., n, priešingu atveju Ω_i = 0.

Dėl sekos η_1, η_2, ..., η_n sudarymo būdo atsitiktiniai dydžiai Ω_1, Ω_2, ..., Ω_n yra tarpusavyje nepriklausomi ir vienodai pasiskirstę, jų vidurkis $E\Omega_i = p$, o dispersija $D\Omega_i = p(1 - p)$.

Klaidų tikimybės įvertinamos standartiniais matematinės statistikos metodais, t.y. remiantis binominio skirstinio savybėmis grąžintinės imties atveju ir hipergeometrinio skirstinio savybėmis negrąžintinės imties atveju bei centrine ribine teorema.

3.3. Trečiojo skyriaus apibendrinimas ir išvados

Šiame skyriuje apžvelgti tikslieji Apriori, AprioriAll, AprioriSome, Apriori DymanicSome, GSP, rekursinis, SPADE algoritmai ir apytiksliai ApproxMAP, tikimybinis dažnų sekų nustatymo ProMFS algoritmai bei atsitiktinės imties metodas RSM.

Tikslieji algoritmai sudaryti naudojant vieną iš šių metodologijų: Apriori principą; FP – growth principą arba Eclat principą.

Tikslieji dažnų sekų paieškos algoritmai veikimo metu turi bent keletą kartų nuskaityti pradinę duomenų bazę, o tai lemia dideles laiko sąnaudas, tačiau tikslieji algoritmai nepakeičiami uždaviniuose, kuriuose reikia pateikti tikslius rezultatus. Šie algoritmai taikomi genetinių, biologinių, medicinos ir pan. uždavinių sprendimui, kurių pagrindinis tikslas yra rezultato tikslumas, o ne laiko sąnaudos.

Apytiksliai algoritmai minimalų skaičių kartų nuskaito duomenų bazę, nes analizuojama ne visa duomenų bazė, o tam tikra metodika sudaryta duomenų bazės imtis.

Spendžiant verslo, finansinių rinkų, draudimo, vartotojų ar klientų elgsenos, telekomunikacijų ir pan. uždavinius priimtina apytikslių algoritmų daroma paklaida, nes dažnai pagrindinis uždavinio kriterijus yra laikas ir atsakymas į klausimą „Koks objektas (ar objektai) yra dažnas?“, o ne gauti tikslų dažno objekto (ar objektų) skaičių duomenų bazėje, todėl dažnai gali būti aukojamas tikslumas dėl ženkliai didesnio rezultatų gavimo greičio.

ApproxMAP algoritmo autoriai H. C. Kum, J. Pei, W. Wang ir D. Duncan nepateikia algoritmo greičio tyrimų, tik tikslumo tyrimus, kurie buvo atlikti naudojant tam tikro tipo duomenų bazę.

Apytikslio ProMFS algoritmo autoriai R. Tumasonis ir G. Dzemyda pateikia algoritmo greičio tyrimus. ProMFS algoritmas analizuoja tam tikru būdu sudarytą iš pradinės duomenų bazės modelinę seką. Šio algoritmo tikslumas paremtas empiriniais bandymais skirtingose duomenų bazėse.

Atsitiktinės imties RSM metodo autoriai J. Pragarauskaitė ir G. Dzemyda pateikia algoritmo greičio tyrimus. RSM algoritmas analizuoja tam tikru būdu sudarytą pradinės duomenų bazės atsitiktinę imtį, ir remiantis šia analize daromos statistinės išvados apie dažnus posekius pradinėje duomenų bazėje. Algoritmo tikslumas įvertinamas statistiniais metodais.

Eksperimentų rezultatai duomenų bazėse D1 ir D2 parodė, kad greičiausiai veikia SPADE algoritmas, antroje vietoje yra GSP, o didžiausios laiko sąnaudos yra rekursinio algoritmo.

Visų algoritmų laiko sąnaudos didėja, mažinant minimalaus dažnumo reikšmę *min_supp*.

4 skyrius. Stochastiniai dažnų posekių paieškos algoritmai

Ankščiau aprašyti metodai naudojami dažnų posekių nustatymui. Tikslieji metodai dažnų posekių paieškai generuoja didelį skaičių sekų – kandidačių bei daug kartų skenuoja duomenų bazę, todėl netinkami naudoti didelėse duomenų bazėse. Apytiksliai metodai nagrinėja atsitiktinę pradinės duomenų bazės imtį, iš kurios daromos išvados apie dažnus posekius pradinėje duomenų bazėje, tačiau neišskiria susietumo taisyklių.

Naujai pasiūlyti stochastiniai algoritmai yra apytiksliai. Šių algoritmų tikslas – nustatyti dažnus posekius didelėse duomenų bazėse bei išskirti susietumo taisykles. Šių algoritmų privalumas – veikimo metu duomenų bazė skenuojama vieną kartą ir atsitiktinai išrenkami ir praleidžiami atsitiktinio ilgio posekiai. Pasirenkamų ir praleidžiamų posekių ilgiai yra pasiskirstę pagal tolygųjį skirstinį. Šie algoritmai leidžia suderinti du svarbius kriterijus, t.y. laiką ir tikslumą. Darbe sukurtų stochastinių algoritmų paklaidų tikimybės įvertinamos naudojant standartinius statistinius metodus.

4.1. Pasiūlytas naujas stochastinis dažnų posekių paieškos algoritmas

Nagrinėjama duomenų bazė *D*. Dažniems posekiams nustatyti yra analizuojami atsitiktinai pasirinkti atsitiktinio ilgio posekiai. Kadangi nėra jokios informacijos apie tai, kad kurioje nors duomenų bazės vietoje kai kurie elementai pasitaiko dažniau nei kiti, tai natūralu priimti, kad bet kuris elementas gali būti dažno posekio elementu su tikimybe, kuri yra vienoda visiems duomenų bazės elementams. Ši tikimybė yra q. Nesunku pastebėti, kad tokiu atveju analizuojamų posekių skaičius yra pasiskirstęs pagal tolygųjį skirstinį su parametru q, o tarpų tarp dviejų analizuojamų posekių ilgiai taip pat pasiskirstę pagal tolygųjį skirstinį su parametru g [28].

Posekio dažnumas yra lygus jo dažnumui tarp visų peržiūrėtų posekių. Santykinis posekio dažnumas yra lygus jo dažnumui tarp visų peržiūrėtų to paties ilgio posekių.

Tegul analizuojant duomenų bazę D atsitiktinai pasirinkta N (imčių skaičius) įvairaus ilgio posekių s_k, kurie sugrupuojami pagal posekių ilgį. Atitinkamo ilgio k posekių dažnumai $supp$ (s_k) apskaičiuojami pagal šią formulę:

$$supp(s_k) = \frac{N_k}{N}, \text{ kur } k = 1, 2, \ldots, n, \quad (1)$$

N_k – atitinkamo ilgio posekių skaičius, N – visų posekių skaičius, k – posekio ilgis, n – maksimalus posekio ilgis.

Posekis priskiriamas dažnų posekių aibei, jei jo dažnumas viršija tam tikrą nustatytą minimalią dažnumo reikšmę min_supp, t.y. $supp(s_k) \geq min\_supp$.

Stochastinis dažnų posekių paieškos algoritmas yra apytikslis, todėl galimos pirmos ir antros rūšies klaidos.

Pirmos rūšies klaida reiškia, kad posekis yra dažnas, tačiau stochastinio algoritmo neaptiktas ir nepriskirtas dažnų posekių aibei.

Antros rūšies klaida reiškia, kad posekis yra nedažnas, o stochastinio algoritmo priskirtas dažnų posekių aibei.

Pasirenkame statistikas p_1, p_2, kurios tenkina šią nelygybę: $P(p_1 \leq p \leq p_2) = \gamma$. Intervalas $[p_1; p_2]$ vadinamas parametro p pasikliautinuoju intervalu. Skaičius γ vadinamas pasikliovimo lygmeniu. Stochastinio algoritmo tikslumo kriterijus – tai posekio radimo pasikliautinojo intervalo rėžiai.

Pasikliautinojo intervalo rėžiai įvertinami pagal šias formules [28]:

$$p_1 = 1 - BetaInv(\frac{1-\gamma}{2}, n-k, k+1); \quad (2)$$

$$p_2 = 1 - BetaInv((1-\frac{1-\gamma}{2}), n-k+1, k); \quad (3)$$

kur p_1 ir p_2 – pasikliautinojo intervalo rėžiai, n – visų posekių skaičius, k – tam tikro posekio pasirodymų skaičius, *BetaInv* – beta skirstinio kvantilis, γ – pasikliovimo lygmuo.

Posekis yra dažnas, jei jo pasikliautinojo intervalo apatinis rėžis p_1 viršija γ.

Pirmos rūšies klaida yra tada, kai $p_2 < \gamma$

Antros rūšies klaida yra tada, kai $p_1 > \gamma$.

Bendra stochastinio dažnų posekių paieškos algoritmo schema:

1 žingsnis. Duomenų bazės failo nuskaitymas.

2 žingsnis. Įvedamas maksimalus imties posekio dydis n. Šis dydis nurodo maksimalų posekio ilgį, kuris gali būti paimtas tolimesniam nagrinėjimui.

3 žingsnis. Įvedama pradinė q reikšmė. Kintamojo q reikšmė priklauso intervalui [0;1].

4 žingsnis. Sukuriamas tuščias rezultatų failas. Šiame faile bus saugomi visi algoritmo veikimo metu atrinkti posekiai. Posekiai šiame faile saugomi surikiuoti pagal posekių ilgius bei dažnumą, kuris apskaičiuojamas pagal (1) formulę.

5 žingsnis. Generuojama kintamojo g reikšmė. Šis dydis pasiskirstęs pagal tolygųjį skirstinį ir priklauso intervalui [0;1].

6 žingsnis. Tikrinama ar reikšmė q yra didesnė už reikšmę g, t.y. ar nelygybė $q > g$ yra teisinga.

7 žingsnis. Jeigu nelygybė $q > g$ yra teisinga, tai apskaičiuojamas imties posekio ilgis $l = round\,(q \cdot n)$, t.y. skaičius, kuris nurodo kokio ilgio posekį reikia paimti. Imties ilgis priklauso intervalui [1; n]. Jeigu eilutėje likusių transakcijos elementų skaičius yra mažesnis už sugeneruotą imties posekio ilgį, tai imami visi tos transakcijos elementai ir pereinama į kitą transakciją.

8 žingsnis. Apskaičiuojamas praleidžiamo posekio ilgis $t = round\,(g \cdot n)$, kuris nurodo kokio ilgio elementų posekis bus praleidžiamas. Šis

skaičius priklauso intervalui [1; n]. Šis dydis yra naudojamas nustatyti praleidžiamo posekio ilgį prieš kitą iteraciją. Jeigu likęs transakcijos elementų skaičius yra mažesnis už apskaičiuotą praleidžiamo posekio ilgį, tai praleidžiami visi likę transakcijos elementai ir pereinama į kitą transakciją.

9 žingsnis. Jeigu g reikšmė didesnė už parametro q reikšmę, t.y. nelygybė $q > g$ yra neteisinga, tai nepraleidžiamas nė vienas elementas. Šiuo atveju praleidžiamo posekio ilgis lygus nuliui.

10 žingsnis. Paimamas apskaičiuoto ilgio posekis s_k. Šio posekio ilgis buvo apskaičiuotas 7 – ajame žingsnyje.

11 žingsnis. Tikrinama, ar rezultatų sąraše yra paimtas posekis s_k.

12 žingsnis. Jeigu posekis s_k yra rezultatų sąraše – tai posekių skaičiaus skaitliuką padidiname vienetu.

13 žingsnis. Jeigu posekio s_k nėra rezultatų sąraše – tai sąrašą papildome nauju posekiu ir posekio skaitliuką prilyginame vienetui.

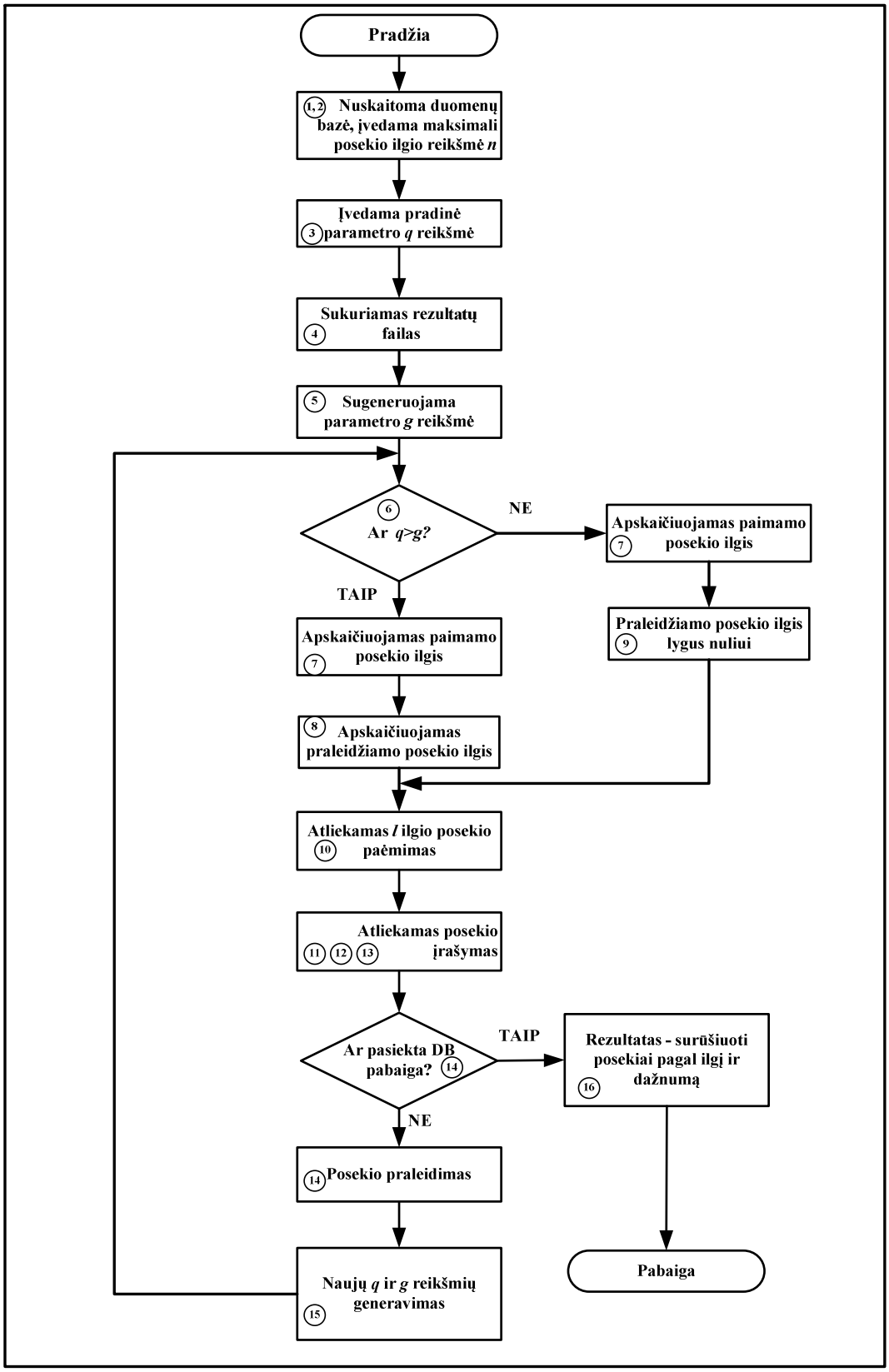
14 žingsnis. Tikrinama, ar nepasiekta duomenų bazės failo pabaiga. Jei nepasiekta failo pabaiga, tai praleidžiamas apskaičiuoto ilgio posekis. Praleidžiamo posekio ilgis buvo apskaičiuotas 8 žingsnyje arba prilyginamas nuliui 9 žingsnyje.

15 žingsnis. Po posekio praleidimo veiksmo generuojamos naujos q ir g reikšmės, kurios pasiskirstę pagal tolygųjį skirstinį ir priklauso intervalui [0, 1]. Po to grįžtama į 6 žingsnį ir kartojami visi algoritmo žingsniai iki 14 žingsnio.

16 žingsnis. Jeigu pasiekta duomenų bazės failo pabaiga, tai sutvarkomas gautas rezultatų failas – posekiai faile išrikiuojami pagal posekių ilgį bei dažnumą, kuris apskaičiuojamas pagal (1) formulę. Atitinkamo ilgio posekių skaičius yra lygus posekių skaitliuko reikšmei, kurios aprašytos 12 ir 13 žingsniuose.

17 žingsnis. Stochastinio dažnų posekių algoritmo rezultatas yra posekiai, kurie išrikiuoti pagal posekio ilgį bei dažnumą, t.y. *Rezultatas:*= $\cup_k s_k$, kur k =1, 2, ..., n.

Stochastinio dažnų posekių paieškos algoritmo veikimo schema pateikta 5 paveiksle.



5 pav. Stochastinio dažnų posekių paieškos algoritmo veikimo schema.

4.2. Pasiūlytos naujo stochastinio dažnų posekių algoritmo modifikacijos

Stochastiniam dažnų posekių paieškos algoritmui buvo atliktos šios modifikacijos:

1. Stochastinio dažnų posekių paieškos algoritmo parametras q arba g įvedamas ir nekinta algoritmo veikimo metu. Jeigu būtų fiksuojami abu parametrai, tai nelygybė $q > g$ visada būtų teisinga ir algoritmo veikimo metu niekada nebūtų praleidžiamas posekis, kurio ilgis lygus nuliui, arba nelygybė $q > g$ visada būtų neteisinga, tai nebūtų praleidžiamas nei vienas posekis, t.y. tiriama visa duomenų bazė, kurios elementai suskaidyti į tam tikro ilgio posekius. Šią stochastinio dažnų posekių paieškos algoritmo modifikaciją pažymime SDPA1.
2. Stochastinio dažnų posekių paieškos algoritmu analizuojami atsitiktinai pasirinkti atsitiktinio ilgio l posekiai, kuriuose yra bent vienas tiksliuoju dažnų posekių paieškos algoritmu nustatytas dažnas vieno – elemento posekis. Šią stochastinio dažnų posekių paieškos algoritmo modifikaciją pažymime SDPA2.

Stochastinių modifikuotų algoritmų SDPA1 ir SDPA2 klaidų įvertinimui naudojama ta pati metodika, kuri aprašyta skyriuje 4.1.

Stochastinio modifikuoto dažnų posekių paieškos SDPA1 algoritmo schema (kai fiksuota q reikšmė/ kai fiksuota g reikšmė):

1 – 4 žingsnis. Nesikeičia (žr. 4.1. skyrius).

5 žingsnis. Nesikeičia / Įvedama kintamojo g reikšmė, kuri nesikeičia algoritmo veikimo metu. Šis skaičius priklauso intervalui [0;1].

6 žingsnis. Nesikeičia / Tikrinama ar įvesta reikšmė g yra didesnė už reikšmę q, t.y. ar nelygybė $q > g$ yra teisinga.

7 – 14 žingsnis. Nesikeičia (žr. 4.1. skyrius).

15 žingsnis. Po posekio praleidimo veiksmo naudojama ta pati q reikšmė, kuri priklauso intervalui [0, 1]. Po to grįžtama į 5 žingsnį ir kartojami visi algoritmo žingsniai iki 14 žingsnio / Po posekio praleidimo veiksmo generuojama nauja q reikšmė, kuri priklauso intervalui [0, 1]. Po to grįžtama į 5 žingsnį ir kartojami visi algoritmo žingsniai iki 14 žingsnio.

16 – 17 žingsnis. Nesikeičia (žr. 4.1. skyrius).

Stochastinio modifikuoto dažnų posekių paieškos SDPA2 algoritmo schema:

1 žingsnis. Nesikeičia (žr. 4.1. skyrius).

2 žingsnis. Įvedama pradinė q reikšmė. Kintamojo q reikšmė priklauso intervalui [0;1]. Tada įvedami dažni vieno – elemento posekiai, kurie nustatyti kokiu nors tiksliuoju dažnų posekių paieškos algoritmu.

3 – 9 žingsnis. Nesikeičia (žr. 4.1. skyrius).

10 žingsnis. Tikrinama ar numatytame paimti posekyje s_k yra nors vienas dažnas vieno – elemento posekis, kuris buvo nurodytas 2 žingsnyje. Jei ši sąlyga tenkinama, tai atliekamas apskaičiuoto ilgio posekio s_k paėmimas. Šio posekio ilgis buvo apskaičiuotas 7 – ajame žingsnyje. Priešingu atveju, vykdomas 14 žingsnis.

11 – 16 žingsnis. Nesikeičia (žr. 4.1. skyrius).

17 žingsnis. Stochastinio modifikuoto dažnų posekių algoritmo SDPA2 rezultatas – posekiai, kuriuose yra bent vienas dažnas vieno – elemento posekis, nustatytas tiksliuoju dažnų posekių paieškos algoritmu. Rezultatų faile šie posekiai išrikiuoti pagal posekio ilgį bei dažnumą, t.y. *Rezultatas:* $= \cup_k s_k$, kur k=1, 2, ..., n.

4.3. Pasiūlytas naujas stochastinis susietumo taisyklių paieškos algoritmas

Susietumo taisyklių paieška susideda iš dviejų pagrindinių etapų: dažnų posekių nustatymo ir susietumo taisyklių sudarymo iš dažnų posekių aibės. Stochastinis dažnų posekių paieškos algoritmas papildytas funkcijomis, kurios iš dažnų posekių sudaro susietumo taisykles. Šis algoritmas pavadintas stochastiniu susietumo taisyklių paieškos algoritmu.

Stochastinio susietumo taisyklių paieškos algoritmo klaidų įvertinimui naudojama ta pati metodika, kuri aprašyta skyriuje 4.1.

Stochastinio susietumo taisyklių paieškos algoritmo schema:

1 žingsnis. Duomenų bazės failo nuskaitymas.

2 žingsnis. Įvedamas maksimalus imties posekio dydis *n*. Šis dydis nurodo maksimalų posekio ilgį, kuris gali būti paimtas tolimesniam nagrinėjimui.

3 žingsnis. Įvedama pradinė *q* reikšmė. Kintamojo *q* reikšmė priklauso intervalui [0;1].

4 žingsnis. Įvedama minimalaus posekio dažnumo reikšmė *min_supp*.

5 žingsnis. Sukuriamas tuščias rezultatų failas. Šiame faile bus saugomi visi algoritmo veikimo metu atrinkti posekiai. Posekiai šiame faile saugomi surikiuoti pagal posekių ilgius bei dažnumą, kuris apskaičiuojamas pagal (1) formulę.

6 žingsnis. Sukuriamas susietumo taisyklių rezultatų failas. Šiame faile bus išsaugotos sudarytos susietumo taisyklės.

7 žingsnis. Generuojama kintamojo *g* reikšmė. Šis dydis pasiskirstęs pagal tolygųjį skirstinį ir priklauso intervalui [0;1].

8 žingsnis. Tikrinama ar reikšmė *q* yra didesnė už reikšmę *g*, t.y. ar nelygybė $q > g$ yra teisinga.

9 žingsnis. Jeigu nelygybė $q > g$ yra teisinga, tai apskaičiuojamas imties posekio ilgis $l = round\ (q \cdot n)$, t.y. skaičius, kuris nurodo kokio ilgio

posekį reikia paimti. Imties ilgis priklauso intervalui [1; n]. Jeigu transakcijos elementų skaičius yra mažesnis už apskaičiuotą imties posekio ilgį, tai imami visi tos transakcijos elementai ir pereinama į kitą transakciją.

10 žingsnis. Apskaičiuojamas praleidžiamo posekio ilgis $t = round\,(g \cdot n)$, kuris nurodo kokio ilgio elementų posekis bus praleidžiamas. Šis skaičius priklauso intervalui [1; n]. Šis dydis yra naudojamas nustatyti praleidžiamo posekio ilgį prieš kitą iteraciją. Jeigu likęs transakcijos elementų skaičius yra mažesnis už apskaičiuotą praleidžiamo posekio ilgį, tai praleidžiami visi likę transakcijos elementai ir pereinama į kitą transakciją.

11 žingsnis. Jeigu g reikšmė didesnė už parametro q reikšmę, t.y. nelygybė $q > g$ yra neteisinga, tai nepraleidžiamas nė vienas elementas. Šiuo atveju praleidžiamo posekio ilgis lygus nuliui.

12 žingsnis. Atliekamas apskaičiuoto ilgio posekio s_k paėmimas. Šio posekio ilgis buvo apskaičiuotas 9 – ajame žingsnyje.

13 žingsnis. Tikrinama, ar rezultatų sąraše yra paimtas posekis s_k.

14 žingsnis. Jeigu posekis s_k yra rezultatų sąraše – tai posekio skaičiaus skaitliuką padidiname vienetu.

15 žingsnis. Jeigu posekio s_k nėra rezultatų sąraše – tai sąrašą papildome nauju posekiu ir posekio skaitliuką prilyginame vienetui.

16 žingsnis. Tikrinama, ar nepasiekta duomenų bazės failo pabaiga. Jei nepasiekta failo pabaiga, tai praleidžiamas apskaičiuoto ilgio posekis. Praleidžiamo posekio ilgis buvo apskaičiuotas 10 – 11 žingsniuose.

17 žingsnis. Po posekio praleidimo veiksmo generuojamos naujos q ir g reikšmės, kurios pasiskirstę pagal tolygųjį skirstinį ir priklauso intervalui [0, 1]. Po to grįžtama į 8 žingsnį ir kartojami visi algoritmo žingsniai iki 16 žingsnio.

18 žingsnis. Jeigu pasiekta duomenų bazės failo pabaiga, tai sutvarkomas gautas rezultatų failas – posekiai faile išrikiuojami pagal posekių ilgį bei dažnumą, kuris apskaičiuojamas pagal (1) formulę.

19 žingsnis. Išsaugomi faile posekiai, kurie išrikiuoti pagal posekio ilgį bei dažnumą, t.y. *Rezultatas:* = $\cup_k s_k$, kur k=1, 2, ..., n.

20 žingsnis. Atrenkami visi posekiai, kurių dažnumo reikšmė ne mažesnė nei nurodyta *min_supp* reikšmė.

21 žingsnis. Iš kiekvieno posekio sudaromos susietumo taisyklės. Susietumo taisyklė sudaroma taip: jei k ilgio posekis X yra k+1 ilgio posekio Y poaibis, tai sudaroma susietumo taisyklė $X \Rightarrow Y$ ir $Y \Rightarrow X$.

22 žingsnis. Apskaičiuojamas kiekvienos sudarytos susietumo taisyklės dažnumas procentais pagal šią formulę:

$$supp(X \Rightarrow Y) = \frac{supp(X \cup Y)}{|I|} \cdot 100\%.$$

23 žingsnis. Apskaičiuojamas kiekvienos sudarytos susietumo taisyklės patikimumas procentais pagal šią formulę:

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \cdot 100\%.$$

24 žingsnis. Rezultatas – susietumo taisyklės, susietumo taisyklių dažnumas ir patikimumas išsaugomi 6 žingsnyje sukurtame rezultatų faile.

Stochastinio susietumo taisyklių paieškos algoritmo veikimo schema pateikta 6 paveiksle.

Pavyzdys 3. Elementų aibė I={A, B, C, D, E}. Transakcijos:

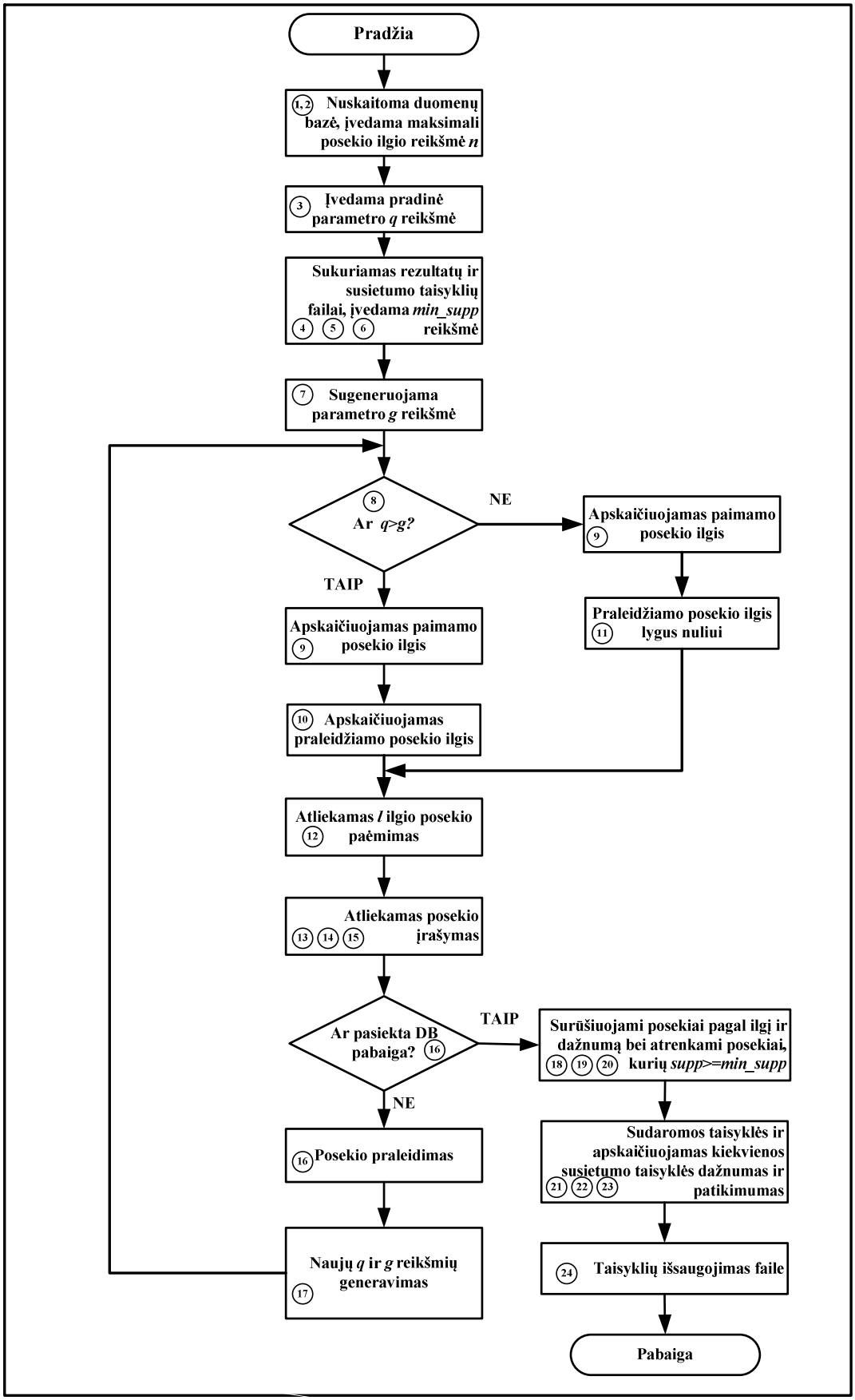
T_1={A, B},

T_2={A, C, D, E},

T_3={B, C, D}, T_4={A, B, C, D},

T_5={A, B, C}.

Nustatyti stochastiniu susietumo taisyklių paieškos algoritmu susietumo taisykles, kurių *min_supp*=40% ir *min_conf*=50%.



6 pav. Stochastinio susietumo taisyklių paieškos algoritmo veikimo schema.

Stochastinis susietumo taisyklių algoritmas nustatė šiuos posekius: *A, B, C, D, AB, AC, BC, CD, ABC*.

Posekių dažnumas: *A* (*supp*=80%), *B* (*supp*=80%), *C* (*supp*=80%), *D* (*supp*=60%), *AB* (*supp*=60%), *AC* (*supp*=60%), *BC* (*supp*=60%), *CD* (*supp*=60%); *ABC* (*supp*=40%).

Stochastinis susietumo taisyklių algoritmas sudarė šias susietumo taisykles:

A $\Rightarrow$ B (dažnumas 60%, patikimumas 75%);

B $\Rightarrow$ A (dažnumas 60%, patikimumas 75%);

A $\Rightarrow$ C (dažnumas 60%, patikimumas 75%);

C $\Rightarrow$ A (dažnumas 60%, patikimumas 75%);

B $\Rightarrow$ C (dažnumas 60%, patikimumas 75%);

C $\Rightarrow$ B (dažnumas 60%, patikimumas 75%);

C $\Rightarrow$ D (dažnumas 60%, patikimumas 75%);

D $\Rightarrow$ C (dažnumas 60%, patikimumas 100%);

A $\Rightarrow$ BC (dažnumas 40%, patikimumas 50%);

BC $\Rightarrow$ A (dažnumas 40%, patikimumas 66,67%);

B $\Rightarrow$ AC (dažnumas 40%, patikimumas 50%);

AC $\Rightarrow$ B (dažnumas 40%, patikimumas 66,67%);

C $\Rightarrow$AB (dažnumas 40%, patikimumas 50%);

AB $\Rightarrow$ C (dažnumas 40%, patikimumas 66,67%).

4.4. Statistinės charakteristikos

Statistinė hipotezė – tai bet kuris tvirtinimas apie nagrinėjamos aibės požymių visumos pasiskirstymą arba apie pasiskirstymo parametrus.

Statistinės hipotezės kriterijus – tai taisyklė, kuri nurodo hipotezės atmetimo atvejus, remiantis atsitiktinės imties reikšmėmis.

Pagrindinis tvirtinimas apie pasiskirstymo dėsnį bei jo parametrus yra vadinamas nuline hipoteze H_0. Alternatyvus tvirtinimas, priešingas nulinei hipotezei žymimas H_1.

Siekiant nustatyti, kuri iš hipotezių yra pagrįsta, naudojami įvairūs statistiniai metodai. Šiuo atveju pasirinktas kriterijaus statistikų vertinimas.

4.4.1. Kriterijaus *u* ir *z* statistikos

Sudarome posekių dažnumų seką ir nustatome jos statistinių savybių pasikeitimo momentą.

Tegul nagrinėjamos dvi nepriklausomos skirtingo ilgio posekių imtys, kurių dydžiai yra n_1 ir n_2. Pirmojoje imtyje n_1 dažniausias posekis pasitaikė k_1 kartų, o antrojoje – k_2 kartų. Tuomet nulinė hipotezė tvirtina, kad dažnų posekių proporcijos imtyse yra vienodos, o alternatyva tvirtina, kad tos proporcijos nelygios:

$$H_0: r_1 = r_2;$$

$$H_1: r_1 \neq r_2.$$

Tuomet iš antimonotoniškumo taisyklės išplaukia, kad esant teisingai nulinei hipotezei nagrinėjami posekiai yra dažni, o iš alternatyvos išplaukia, kad ilgesnysis iš tų posekių yra nedažnas.

Kriterijaus hipotezėje H_0 statistika gali būti įvertinama įvairiais būdais. Kriterijaus statistika u konstruojama taip, kad hipotezei H_0 esant teisingai, ji būtų pasiskirsčiusi pagal standartinį normalųjį skirstinį. Kriterijaus statistika u apskaičiuojama pagal šią formulę [28]:

$$u = \frac{d_1 - d_2}{\sqrt{\left(\frac{k_1 + k_2}{n_1 + n_2}\right) \cdot \left(1 - \frac{k_1 + k_2}{n_1 + n_2}\right) \cdot \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}$$

kur $d_1 = k_1/n_1$ ir $d_2 = k_2/n_2$.

Pažymime, kad $d = (k_1 + k_2) / (n_1 + n_2)$, tai gaunama tokia formulė:

$$u = \frac{d_1 - d_2}{\sqrt{d \cdot (1 - d) \cdot \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}.$$

Kriterijaus statistiką z galima įvertinti ir taip [10, 94]:

$$z = \left(2\arcsin\sqrt{d_1} - 2\arcsin\sqrt{d_2}\right)\cdot\sqrt{\frac{n_1 \cdot n_2}{n_1 + n_2}}\,.$$

4.4.2. Prielaidų vertinimas

Posekių charakteristikų pasikeitimo momentui nustatyti sukurtas modelis, kuriame dažniausiai pasitaikantis posekis (pvz.: pirkinių krepšelis) yra nustatomas didžiausio tikėtinumo būdu. Dažno posekio ilgis nustatomas remiantis monotoniškumo taisykle – dažnų posekių poaibis yra dažnas posekis. Sekų charakteristikų pasikeitimo momentų nustatymo modelyje yra apskaičiuojamas dažniausio posekio dažnis, priklausomai nuo jo ilgio. Toliau pasinaudojama tuo:

- tiriamo posekio ilgis yra mažesnis už tam tikrą ilgį, tai nagrinėjamo posekio dažnis yra beveik pastovus;
- tiriamo posekio ilgis yra didesnis už tam tikrą ilgį, tai nagrinėjamo dažniausio posekio tikimybė pradeda mažėti.

Tokiu atveju yra sudaromas binarinis procesas, kuriame nagrinėjami du gretimi tam tikro ilgio posekiai l_i *ir* l_{i+1}, $i \in [1, N]$. Binarinio proceso reikšmė yra:

- lygi vienetui, jei statistinis kriterijus neprieštarauja hipotezei apie dviejų gretimo ilgio posekių pasirodymo tikimybės sutapimą;
- lygi nuliui, jei dviejų gretimo ilgio posekių pasirodymo tikimybės reikšmingai skiriasi.

Šio binarinio proceso charakteristikų pasikeitimo momentas leidžia nustatyti dažniausių posekių ilgį (pirkinių krepšelio dydį, t.y. iš kiek elementų susideda dažniausias pirkinių krepšelis). Kadangi dviejų gretimo ilgio posekių pasirodymų skaičius imtyse yra pasiskirstęs pagal binominį dėsnį, galima sudaryti logaritminę tikėtinumo funkciją dažniausiai pasitaikančio posekio pasirodymo tikimybės pasikeitimui nustatyti:

$$f(k) = \frac{k!}{k_1! \cdot (k-k_1)!} \cdot \frac{(N-k)!}{(K-k_1)! \cdot (N-k-K+k_1)!} \cdot p_1^{k_1} \cdot (1-p_1)^{k-k_1} \cdot$$
$$\cdot p_2^{K-k_1} \cdot (1-p_2)^{N-k-K+k_1}.$$

Iš čia gauname logaritminę tikėtinumo funkciją:

$$\ln\big(f(k)\big) = \sum_{i=1}^{k} \ln i - \sum_{i=1}^{k_1} \ln i - \sum_{i=1}^{k-k_1} \ln i + \sum_{i=1}^{N-k} \ln i - \sum_{i=1}^{K-k_1} \ln i - \sum_{i=1}^{N-k-K+k_1} \ln i +$$
$$+ \sum_{i=1}^{k_1} \ln p_1 + \sum_{i=1}^{k-k_1} \ln(1-p_1) + \sum_{i=1}^{K-k_1} \ln p_2 + \sum_{i=1}^{N-k-K+k_1} \ln(1-p_2),$$

kur k – binarinio proceso charakteristikų pasikeitimo momentas, k_1 – tikimybių dažnių sutapimų skaičius iki pasikeitimo momento, k_2 – tikimybių dažnių sutapimų skaičius po pasikeitimo momento, N – maksimalus posekio ilgis, K – nagrinėjamo posekio ilgis. Dažniausio posekio ilgis atitinka logaritminės tikėtinumo funkcijos minimumo reikšmę. Minimizuojančią funkciją yra patogu įvesti kaip dviejų gretimų šios funkcijos reikšmių skirtumą, kuris yra lygus:

$$\ln(f(k)/f(k-1)) =$$
$$= \ln k - \ln(k-k_1) - \ln(N-k+1) + \ln(N-k-k_1+1) + \ln(1-p_1) +$$
$$+\ln(1-p_2),$$

jeigu k – oji binarinio proceso reikšmė yra lygi 0.

Jei k – oji binarinio proceso reikšmė lygi 1, tai gretimų tikėtinumo funkcijos reikšmių skirtumas yra lygus:

$$\ln(f(k)/f(k-1)) =$$
$$= \ln k - \ln k_1 - \ln(N-k+1) + \ln(K-k_1+1) + \ln p_1 + \ln p_2,$$

čia $p_1 = \frac{k_1}{k}, p_2 = \frac{k_2}{k}$.

Tikėtinumo funkcijos minimumas sutampa su pirmąja kintamojo k reikšme, kuriai dviejų gretimų šios funkcijos reikšmių skirtumas yra teigiamas. Skaičiavimai pradedami nuo pradinės reikšmės $k = 0$. Nesunku pastebėti, kad pradinė tikėtinumo funkcijos reikšmė yra:

$$\ln\big(f(0)\big) = \sum_{i=1}^{N} \ln i - \sum_{i=1}^{K} \ln i - \sum_{i=1}^{N-K} \ln i + K \cdot \ln p_2 + \ln(1-p_2) \cdot (N-K).$$

Apskaičiuojant logaritminės tikėtinumo funkcijos reikšmes, galima naudotis rekurentinėmis formulėmis:

- jei *k*-oji binarinio proceso reikšmė yra lygi nuliui, tai:

$$k_1(k+1) = k_1(k), \qquad k_2(k+1) = k_2(k);$$

- jei *k*-oji binarinio proceso reikšmė yra lygi vienetui, tai:

$$k_1(k+1) = k_1(k) + 1, \qquad k_2(k+1) = k_2(k) - 1.$$

Įvertinus kriterijaus statistiką bei tikėtinumo funkcijos reikšmes, atliekamas prielaidų tikimybių vertinimas. Kai alternatyva dvipusė *(H_1: $r_1 \neq r_2$)*, gautąją *u* reikšmę atitinkanti *p* – reikšmė apskaičiuota taip:

$$p = 2 \cdot \Big(1 - NORMSDIST\big(ABS(u)\big)\Big),$$

kur *NORMSDIST* – normaliojo skirstinio funkcija, *ABS* – modulio funkcija. *p* – reikšmė reiškia tikimybę rizikos, kad atmetant H_0 bus padaryta pirmos rūšies klaida, todėl H_0 pagrįstai atmesti galima tik tada, kaip *p* – reikšmė gaunama nedidelė, nežymi, mažesnė už įprastinius, tradicinius reikšmingumo lygmenis (0,1; 0,05; 0,01 ar 0,001). *p* – reikšmė išreiškia hipotezės H_0 tikėtinumą, t. y. tikimybę, kad joje išsakytas teiginys atitinka tikrovę, todėl kuo didesnė *p* – reikšmė, tuo labiau nulinė hipotezė pasikliautina.

4.5. Ketvirtojo skyriaus išvados

Šiame skyriuje pateiktas naujas stochastinis dažnų posekių paieškos algoritmas. Taip pat pateiktos modifikacijos SDPA1 ir SDPA2 bei stochastinis susietumo taisyklių paieškos algoritmas.

Darbe sukurtas stochastinis dažnų posekių paieškos algoritmas, jo modifikacijos SDPA1 ir SDPA2 bei stochastinis susietumo taisyklių paieškos algoritmas yra apytiksliai algoritmai.

Stochastiniai algoritmai nustatydami dažnus posekius ir susietumo taisykles duomenų bazę skaito vieną kartą, todėl yra tinkami naudoti didelėse duomenų bazėse.

Anksčiau aprašyti apytiksliai dažnų posekių paieškos algoritmai analizuoja ne visą duomenų bazę, o tam tikru būdu sudarytą pradinės duomenų bazės atsitiktinę imtį bei neatlieka susietumo taisyklių paieškos.

Darbe sukurtų stochastinių algoritmų klaidų tikimybės įvertintos naudojant standartinius statistinius metodus, todėl gali būti taikomi daugelyje sričių, kur algoritmo paklaidos yra priimtinos, pavyzdžiui pirkinių krepšelio, vartotojų elgsenos analizės uždaviniams spręsti, marketingo, finansinių duomenų analizei ir t. t.

5 skyrius. Eksperimentiniai tyrimai

Eksperimentiniai tyrimai disertacijoje sukurtiems stochastiniams algoritmams buvo atliekami naudojant dirbtinai sugeneruotus ir realius duomenis.

Eksperimentiniams tyrimams naudotas kompiuteris, kurio techninės charakteristikos šios: 2.50 GHz Intel(R) Core(TM) i5-3210 procesorius, 4 GB RAM atmintis, standusis diskas HDD 500 GB.

Eksperimentiniuose tyrimuose naudoti Apriori, GSP, rekursinis, SPADE, ProMFS algoritmai, stochastinis dažnų posekių paieškos algoritmas bei jo modifikacijos SDPA1, SDPA2 ir stochastinis susietumo taisyklių paieškos algoritmas realizuoti naudojant Object Pascal programavimo kalbą.

5.1. Eksperimentinės duomenų bazės

Eksperimentiniai tyrimai buvo atliekami naudojant realią pirkimų transakcijų duomenų bazę, realią baigiamųjų darbų temų duomenų bazę bei imitacines duomenų bazes.

Imitacinių duomenų bazių generavimui naudojama sukurta programinė įranga.

Eksperimentiniams tyrimams buvo naudojamos šios duomenų bazės:

1. Sugeneruotos duomenų bazės, kurių charakteristikos aprašytos 5.2. skyriuje.
2. Reali Vilniaus kolegijos Elektronikos ir informatikos fakulteto Programavimo kompiuteriams studijų programos baigiamųjų darbų 2000 – 2013 m. m. temų duomenų bazė. Duomenų bazę sudaro 1030 temų.
3. Reali UAB „Arsuna“ vienų metų pirkimų transakcijų duomenų bazė. Duomenų bazę sudaro 400 000 transakcijų.

5.2. Naujai pasiūlytų stochastinio dažnų posekių, SDPA1 ir SDPA2 algoritmų tikslumo tyrimas

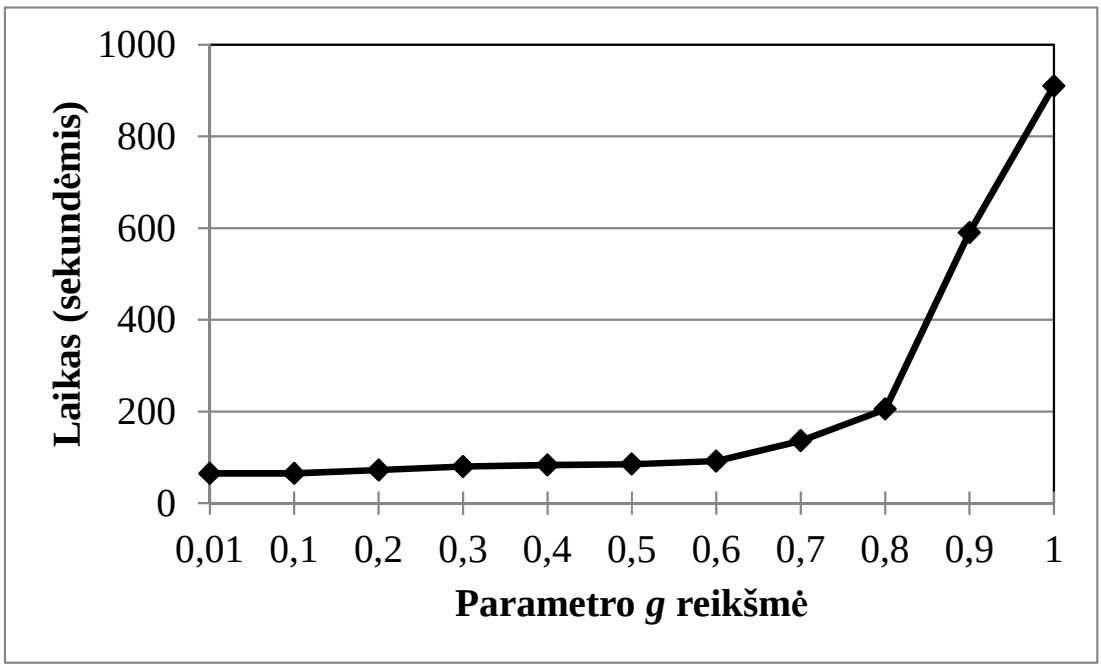
Eksperimentui buvo sugeneruota 100 duomenų bazių failų, kurių dydis 200 000 simbolių, t.y. 2000 eilučių, 100 simbolių eilutėje. Duomenų bazė sudaryta iš simbolių *A, E, I, N, S*, t.y. *I*={*A, E, I, N, S*}. Tarp šių simbolių buvo įterpiamas posekis *SIENA*. Duomenų bazės failai buvo generuojami su šiomis tikimybėmis:

- ✓ posekio *SIENA* įterpimo tikimybė – 0,2;
- ✓ simbolio *A* įterpimo tikimybė – 0,15;
- ✓ simbolio *E* įterpimo tikimybė – 0,15;
- ✓ simbolio *I* įterpimo tikimybė – 0,15;
- ✓ simbolio *N* įterpimo tikimybė – 0,15;
- ✓ simbolio *S* įterpimo tikimybė – 0,2.

Šie failai apdoroti naujai pasiūlytais stochastiniu dažnų posekių paieškos, SDPA1 ir SDPA2 algoritmais po 100 kartų. Visų eksperimentų metu buvo pasirinktas maksimalus posekio ilgis – 5 simboliai. Naujai pasiūlytuose stochastiniuose algoritmuose parametras *q* naudojamas apskaičiuojant posekio imties dydį, o parametras *g* – apskaičiuojant praleidžiamo posekio ilgį. Atlikus eksperimentus buvo įvertintas vidutinis vieno failo apdorojimo laikas. Taip pat eksperimentų metu buvo tiriama SDPA1 algoritmo rezultatų priklausomybė nuo parametro *g* reikšmės, t.y. esant fiksuotoms skirtingoms *g* reikšmėms (0,01; 0,1; 0,2, ..., 1). Algoritmo parametro *q* reikšmė – generuojamas atsitiktinis dydis, kuris pasiskirstęs pagal tolygųjį skirstinį.

Stochastinio dažnų posekių paieškos, SDPA1 ir SDPA2 algoritmų veikimo laikas bei tikslumas buvo palyginti su Apriori, GSP, SPADE, rekursinio, ProMFS algoritmų rezultatais.

SDPA1 algoritmo vidutinė vieno failo apdorojimo trukmė pateikta 7 paveiksle.



7 pav. SDPA1 algoritmo vykdymo trukmės priklausomybė nuo parametro *g* reikšmės.

Iš eksperimento rezultatų pastebima, kad kuo didesnė parametro *g* reikšmė, tuo sparčiau didėja algoritmo vykdymo laikas (žr. 7 pav.), nes dažniau neatliekamas atsitiktinio ilgio posekio praleidimas (žr. 4 ir 5 pav.). Failo apdorojimo laikas pradeda didėti, kai $g \geq 0{,}6$. Kai $g = 1$, tai failo apdorojimo laikas išauga, nes neatliekamas posekių praleidimas, o visa duomenų bazė suskaidoma posekiais.

Eksperimento metu buvo palyginti dažni posekiai nustatyti GSP, stochastiniu dažnų posekių paieškos, SDPA1 ir SDPA2 algoritmais, kai posekio ilgis lygus 5. Apdorojant duomenų bazių failus GSP algoritmu buvo pasirinktas minimalus dažnumas *min_supp*=1000 (0,5 %), o SDPA1 algoritmu – pasirinktas parametras $g = 0{,}5$, SDPA2 algoritme pasirinktas vieno elemento dažnų posekių nustatymui GSP algoritmas. Visi algoritmai nustatė tuos pačius dažnus posekius, skiriasi tik dažnų posekių skaičius, kuris pateiktas 2 lentelėje.

Stochastinio dažnų posekių paieškos ir SDPA1, kai pasirinkta parametro *g* reikšmė 0,5, algoritmų nustatytų dažnų posekių skaičius vidutiniškai 4 kartus mažesnis nei tiksliojo GSP algoritmo nustatytų dažnų posekių skaičius. SDPA2

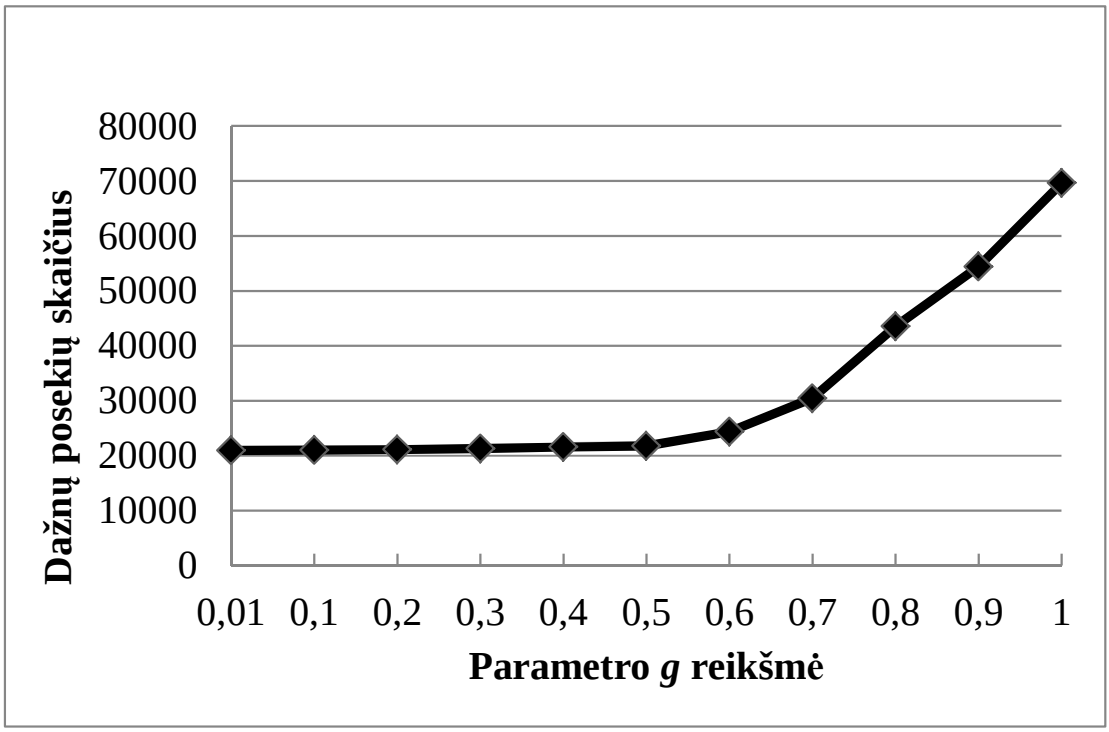
algoritmo nustatytų dažnų posekių skaičius vidutiniškai 2 kartus mažesnis nei tiksliojo GSP algoritmo nustatytų dažnų posekių skaičius.

2 lentelė. Dažni posekiai ir jų skaičius.

GSP algoritmas		**Stochastinis algoritmas**		**SDPA1 algoritmas**		**SDPA2 algoritmas**	
Posekis	**Kiekis**	**Posekis**	**Kiekis**	**Posekis**	**Kiekis**	**Posekis**	**Kiekis**
SIENA	21671	SIENA	5115	SIENA	5220	SIENA	10834
IENAS	10268	IENAS	2459	IENAS	2510	IENAS	5144
ASIEN	7563	ASIEN	1749	ASIEN	1785	ASIEN	3782
SSIEN	5943	SSIEN	1484	SSIEN	1515	SSIEN	2970
ENASI	4905	ENASI	1205	ENASI	1230	ENASI	2453
NASIE	4641	NASIE	989	NASIE	1010	NASIE	2320
IENAA	3499	IENAA	793	IENAA	810	IENAA	1749
ENASS	2938	ENASS	637	ENASS	650	ENASS	1466
IENAI	2618	IENAI	681	IENAI	695	IENAI	1311
ISIEN	2614	ISIEN	637	ISIEN	650	ISIEN	1304
ESIEN	2594	ESIEN	647	ESIEN	660	ESIEN	1297
NSIEN	2593	NSIEN	656	NSIEN	670	NSIEN	1297
IENAN	2585	IENAN	627	IENAN	640	IENAN	1293
IENAE	2520	IENAE	578	IENAE	590	IENAE	1267
ASSIE	2142	ASSIE	559	ASSIE	570	ASSIE	1068
SSSIE	1632	SSSIE	299	SSSIE	305	SSSIE	821
ENAAS	1607	ENAAS	412	ENAAS	420	ENAAS	806
NASSI	1504	NASSI	318	NASSI	325	NASSI	758
ENAIS	1236	ENAIS	333	ENAIS	340	ENAIS	617
ENANS	1217	ENANS	304	ENANS	310	ENANS	611
AASIE	1208	AASIE	299	AASIE	305	AASIE	606
ENAES	1188	ENAES	314	ENAES	320	ENAES	591
ENASA	1004	ENASA	220	ENASA	225	ENASA	507

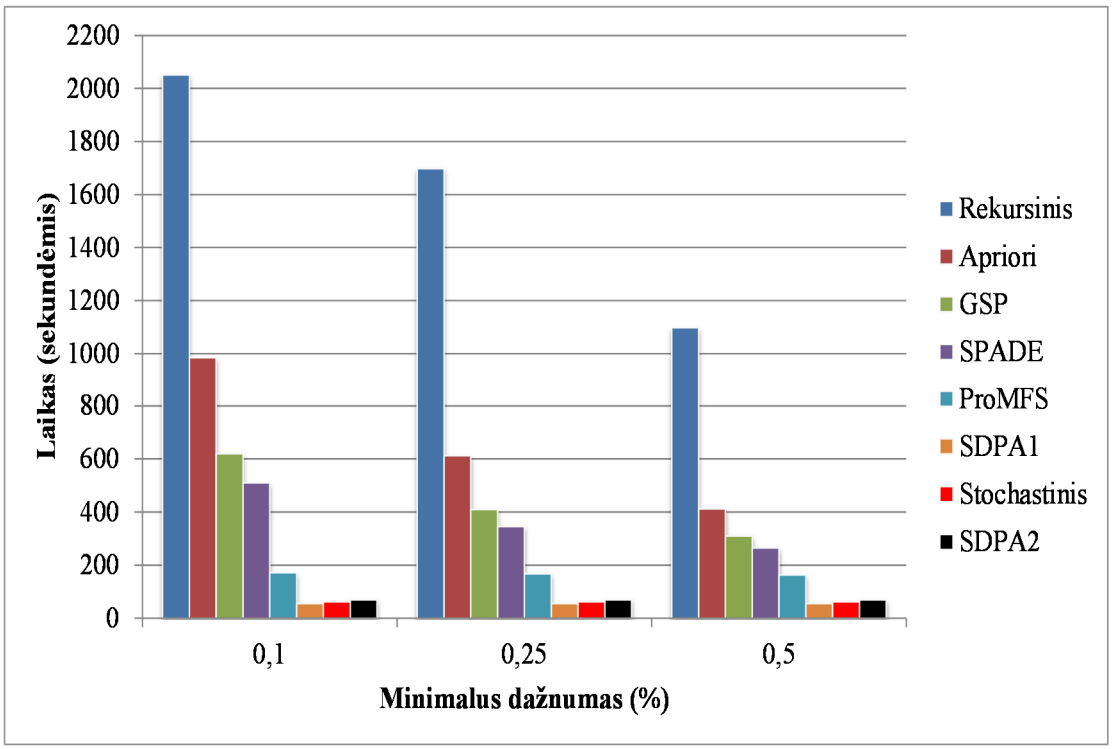
Bendras SPDA1 algoritmo nustatytų dažnų posekių, kurie pateikti 2 lentelėje, skaičiaus palyginimas, esant skirtingoms parametro *g* reikšmėms, pateiktas 8 paveiksle.

Kiekvieno SPDA1 algoritmo nustatyto dažno posekio, kurie pateikti 2 lentelėje, skaičiaus kitimo palyginimas, esant skirtingoms parametro *g* reikšmėms, pateiktas 1 priede.



8 pav. SDPA1 algoritmo bendro dažnų posekių skaičiaus priklausomybė nuo parametro *g* reikšmės.

Šios duomenų bazės buvo apdorotos Apriori, GSP, SPADE, rekursiniu, stochastiniu dažnų posekių paieškos, SDPA1, SPDA2 ir ProMFS algoritmais. Apriori, GSP, SPADE ir rekursiniame algoritmuose pasirinktas minimalus posekio dažnumas *min_supp*=200 (0,1%), *min_supp*=500 (0,25 %), *min_supp*=1000 (0,5%). ProMFS algoritmo sukurta modelinė seka buvo analizuojama GSP algoritmu, kai *min_supp*=6 (0,1%), *min_supp*=15 (0,25%), *min_supp*=30 (0,5%). SDPA1 algoritmo parametro *g* reikšmė buvo pasirinkta *g*=0,3. SDPA2 algoritme buvo naudojami vieno elemento posekiai, nustatyti GSP algoritmu. Algoritmų vidutinis duomenų bazių apdorojimo laikas pateiktas 9 paveiksle.



9 pav. Vidutinis Apriori, GSP, SPADE, rekursinio, ProMFS, stochastinio, SDPA1 ir SDPA2 algoritmų veikimo laikas.

Eksperimento rezultatai parodė, kad mažiausios laiko sąnaudos yra SDPA1 algoritmo, o didžiausias duomenų bazių apdorojimo laikas yra rekursinio algoritmo.

5.3. Stochastinio dažnų posekių paieškos, SDPA1 ir SDPA2 algoritmų tikslumo tyrimas

Eksperimento metu buvo sugeneruota 19 duomenų bazių grupių. Šios grupės ir jų generavimo charakteristikos pateiktos 2 priede. Kiekvieną failų grupę sudaro 100 failų. Visi 1900 failų apdoroti su stochastiniu dažnų posekių paieškos, SDPA1 ir SDPA2 algoritmais. Eksperimentas atliktas 80 kartų.

Eksperimento metu buvo įvertinamos stochastinių algoritmų pirmos ir antros rūšies klaidos.

Posekis priskiriamas dažnų posekių aibei, jei jo minimalus dažnumas $min\_supp \geq 0{,}07$.

Stochastinio dažnų posekių paieškos, SDPA1 ir SDPA2 algoritmų pasikliautinojo intervalo rėžiai p_1 ir p_2 buvo apskaičiuoti pagal (2) ir (3) formules kiekvienoje duomenų bazėje, kai γ=0,95.

Stochastinio dažnų posekių paieškos algoritmo pasikliautinojo intervalo rėžių vidurkių reikšmės yra p_1 = 0,95 ir p_2 = 0,99.

SDPA1 algoritmo pasikliautinojo intervalo rėžiai p_1 ir p_2 buvo apskaičiuoti kiekvienoje duomenų bazėje, esant skirtingoms algoritmo parametro *g* reikšmėms. 3 lentelėje pateiktos visų duomenų bazių pasikliautinojo intervalo rėžių p_1 ir p_2 vidurkių reikšmės.

3 lentelė. Pasikliautinojo intervalo rėžiai.

Parametro *g* reikšmė	p_1	p_2
0,01	0,9527	0,9993
0,1	0,9537	0,9993
0,2	0,9531	0,9993
0,3	0,9517	0,9993
0,4	0,9527	0,9992
0,5	0,9521	0,9993
0,6	0,9523	0,9992
0,7	0,9599	0,9992
0,8	0,9533	0,9992
0,9	0,9535	0,9992
1	0,9634	0,9992

Įvertinus eksperimento rezultatus nustatytas SDPA1 algoritmo pasikliautinasis intervalas yra [0,95; 0,99].

SDPA2 algoritmo pasikliautinojo intervalo rėžiai p_1 ir p_2 buvo apskaičiuoti kiekvienoje duomenų bazėje. Pasikliautinojo intervalo rėžių vidurkių reikšmės yra p_1 = 0,97 ir p_2 = 0,99, t.y. pasikliautinasis intervalas yra [0,97; 0,99].

Stochastinio dažnų posekių paieškos algoritmo vidutinė pirmos rūšies klaida yra 2,38 %, vidutinė antros rūšies klaida – 5,58 %.

SDPA1 algoritmo pirmos ir antros rūšies klaidos pateiktos 4 lentelėje.

4 lentelė. SDPA1 algoritmo pirmos ir antros rūšies klaidos.

SDPA1 algoritmo parametro *g* reikšmė	**Pirmos rūšies klaida,** %	**Antros rūšies klaida,** %
0,01	2,63	6,10
0,1	2,74	6,41
0,2	2,43	6,47
0,3	2,42	5,89
0,4	2,41	5,52
0,5	2,42	5,41
0,6	2,36	5,2
0,7	2,34	5,12
0,8	2,33	5,07
0,9	2,21	5,05
1	2,05	4,63

SDPA1 algoritmo pirmos rūšies klaida yra apytiksliai 2,4 %, antros rūšies klaida – 5,6 %.

SDPA2 algoritmo vidutinė pirmos rūšies klaida yra 1,3 %, vidutinė antros rūšies klaida – 3,12 %.

5.4. Baigiamųjų darbų temų duomenų bazės tyrimas

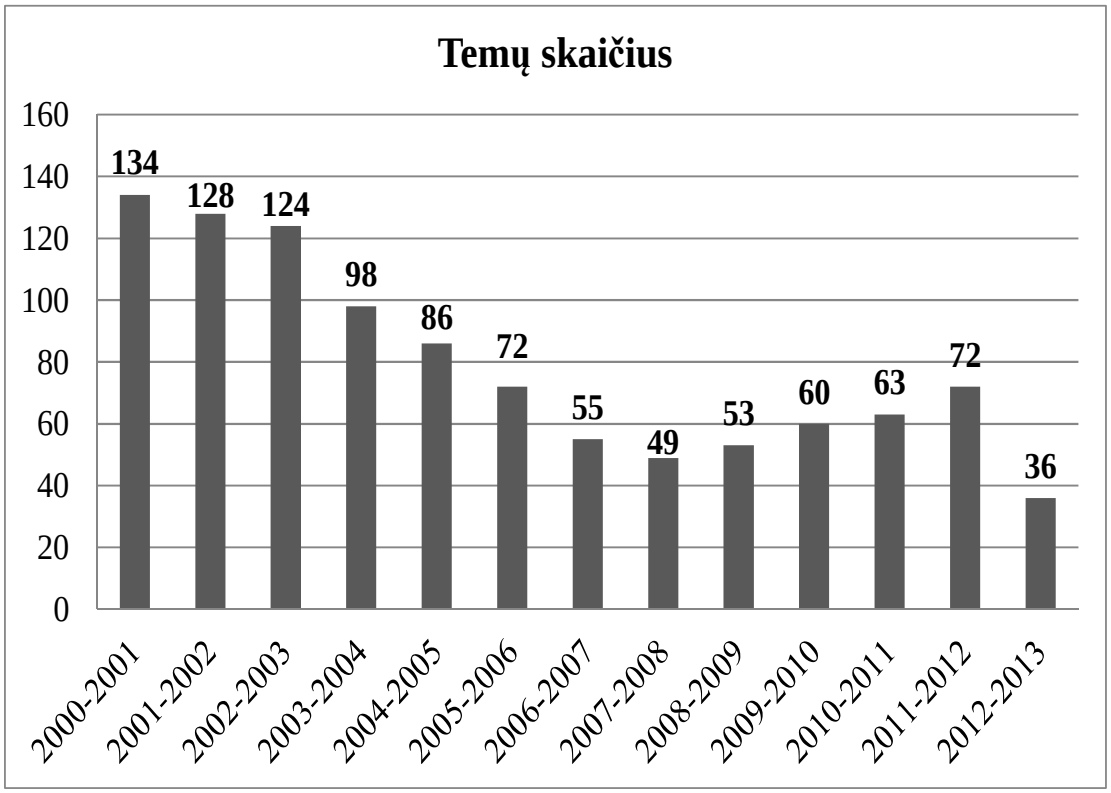
Eksperimento metu tiriama 2000 – 2013 m. m. (13 mokslo metų) Vilniaus kolegijos Elektronikos ir informatikos fakulteto Programavimo kompiuteriams studijų programos baigiamųjų darbų temų duomenų bazė. Duomenų bazę sudaro 1030 temų.

Eksperimento tikslas – nustatyti, kokie dažniausi žodžiai aptinkami baigiamųjų darbų pavadinimuose bei nustatyti susietumo taisykles tarp dažniausių žodžių.

Eksperimento metu vienas žodis buvo laikomas nedalomu elementu. Pavyzdžiui, tema UAB „Skado medis” interneto svetainė, tai i_1=UAB, i_2=Skado, i_3=medis, i_4=interneto, i_5=svetainė.

Eksperimento metu baigiamųjų darbų temų duomenų bazė analizuota tiksliuoju SPADE algoritmu bei apytiksliais ProMFS, stochastiniu dažnų posekių paieškos, SDPA1 ir stochastiniu susietumo taisyklių paieškos algoritmais, norint nustatyti dažnus vieno elemento posekius. Eksperimento metu buvo pasirinkta SDPA1 algoritmo parametro *g* reikšmė lygi 0,5. Šiame etape duomenų bazės nebuvo analizuojamos SDPA2 algoritmu, nes šis algoritmas naudoja dažnus vieno elemento posekius, kurie nustatyti kitais algoritmais.

Baigiamųjų darbų temų skaičius pasirinktais mokslo metais pateiktas 10 paveiksle.



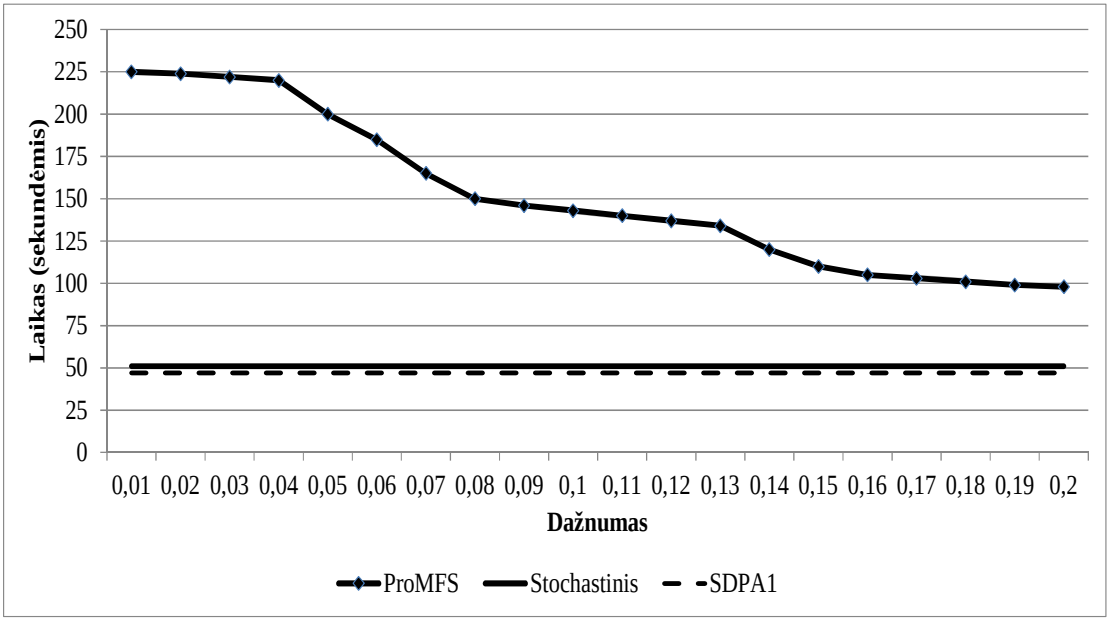
10 pav. Baigiamųjų darbų temų skaičius.

Eksperimento metu SPADE, ProMFS, stochastinis dažnų posekių paieškos, SDPA1 ir stochastinis susietumo taisyklių paieškos algoritmai, nustatė šiuos dažniausius elementus (žodžius):

- Programa;
- UAB;
- Sistema;

- Svetainė;
- Tvarkymo;
- Modulis;
- Apdorojimo;
- Žaidimas.

Apytikslių ProMFS, stochastinio susietumo taisyklių paieškos, SDPA1 algoritmų greičio palyginimas Programavimo kompiuteriams studijų programos baigiamųjų darbų temų duomenų bazėje priklausomai nuo nustatyto minimalaus dažnumo *min_supp* pateiktas 11 paveiksle.



11 pav. Apytikslių metodų greičio palyginimas.

Eksperimento rezultatai patvirtino ProMFS algoritmo autoriaus teiginį: „*Galima padaryti išvadą, kad ProMFS algoritmas yra efektyvus (žymiai mažesnės laiko sąnaudos), esant sąlyginai dideliam minimaliam dažnumui*“. [88]. Stochastinių algoritmų efektyvumui nustatytas minimalus dažnumas įtakos neturi, nes dažni posekiai atrenkami atsitiktiniu būdu. Minimalaus dažnumo reikšmė įtakoja tik sudaromų susietumo taisyklių skaičių.

Dažniausiai sutinkami žodžiai 2000 – 2013 m. baigiamųjų darbų temų pavadinimuose yra *UAB*, *Programa*, *Sistema*, *Svetainė*, *Tvarkymo*, *Modulis*,

Apdorojimo, Žaidimas. Šių posekių dažnumas, kai *min_supp* = 0,01, pateiktas 5 lentelėje.

5 lentelė. Posekiai ir jų dažnumas.

Posekis	**Dažnumas**
UAB	0,34
Programa	0,53
Sistema	0,23
Svetainė	0,14
Tvarkymo	0,12
Modulis	0,06
Apdorojimo	0,08
Žaidimas	0,03

Dažni 2 – elementų posekiai ir jų dažnumas, kai *min_supp* = 0,01 pateiktas 6 lentelėje.

6 lentelė. 2 – elementų posekiai ir jų dažnumas.

Posekis	**Dažnumas**
{*UAB, Programa*}	0,11
{*UAB, Sistema*}	0,08
{*UAB, Svetainė*}	0,04
{*UAB, Tvarkymo*}	0,06
{*UAB, Modulis*}	0,02
{*UAB, Apdorojimo*}	0,03
{*Programa, Tvarkymo*}	0,01
{*Programa, Apdorojimo*}	0,02
{*Sistema, Tvarkymo*}	0,01
{*Sistema, Apdorojimo*}	0,01
{*Tvarkymo, Modulis*}	0,01
{*Apdorojimo, Modulis*}	0,01

Stochastinis susietumo taisyklių paieškos algoritmas nustatė susietumo taisykles bei apskaičiavo susietumo taisyklių dažnumą ir patikimumą. 7 lentelėje pateiktos susietumo taisyklės, kurių minimalus patikimumas *min_conf* = 2%.

2000 – 2001 m.m. 73,88 % baigiamųjų darbų temų buvo žodis *Programa*, 20,9 % baigiamųjų darbų temų buvo žodis *Sistema*, 3,73 % baigiamųjų darbų temų buvo žodis *Modulis.*

7 lentelė. Susietumo taisyklės.

Posekis	**Dažnumas**	**Patikimumas, %**
UAB ⇒ Programa	0,11	39,41
UAB ⇒ Sistema	0,08	18,53
UAB ⇒ Svetainė	0,04	5,88
UAB ⇒ Tvarkymo	0,06	5,29
UAB ⇒ Modulis	0,02	2,35
UAB ⇒ Apdorojimo	0,03	3,24
Programa ⇒ Tvarkymo	0,01	3,39
Sistema ⇒ Tvarkymo	0,01	5,21
Sistema ⇒ Apdorojimo	0,01	4,35
Tvarkymo ⇒ Modulis	0,01	6,67
Apdorojimo ⇒ Modulis	0,01	11,25

2001 – 2002 m.m. 66,41 % baigiamųjų darbų temų buvo žodis *Programa*, 28,91 % baigiamųjų darbų temų buvo žodis *Sistema*, 3,12 % baigiamųjų darbų temų buvo žodis *Modulis.*

2002 – 2003 m.m. 65,32 % baigiamųjų darbų temų buvo žodis *Programa*, 24,19 % baigiamųjų darbų temų buvo žodis *Sistema*, 5,64 % baigiamųjų darbų temų buvo žodis *Žaidimas*, 6,45 % baigiamųjų darbų temų buvo žodis *Modulis.*

2003 – 2004 m.m. 75,51 % baigiamųjų darbų temų buvo žodis *Programa*, 16,33 % baigiamųjų darbų temų buvo žodis *Sistema*, 7,14 % baigiamųjų darbų temų buvo žodis *Žaidimas*, 1,02 % baigiamųjų darbų temų buvo žodis *Modulis.*

2004 – 2005 m.m. 48,84 % baigiamųjų darbų temų buvo žodis *Programa*, 24,42 % baigiamųjų darbų temų buvo žodis *Sistema*, 16,27 % baigiamųjų darbų temų buvo žodis *Svetainė*, 5,81 % baigiamųjų darbų temų buvo žodis *Žaidimas*, 2,33 % baigiamųjų darbų temų buvo žodis *Modulis.*

2005 – 2006 m.m. 42,06 % baigiamųjų darbų temų buvo žodis *Programa*, 18,06 % baigiamųjų darbų temų buvo žodis *Sistema*, 26,39 % baigiamųjų darbų temų buvo žodis *Svetainė*, 8,33 % baigiamųjų darbų temų buvo žodis *Žaidimas*, 2,78 % baigiamųjų darbų temų buvo žodis *Modulis.*

2006 – 2007 m.m. 47,27 % baigiamųjų darbų temų buvo žodis *Programa*, 30,9 % baigiamųjų darbų temų buvo žodis *Sistema*, 16,36 % baigiamųjų darbų temų buvo žodis *Svetainė*, 3,64 % baigiamųjų darbų temų buvo žodis *Modulis.*

2007 – 2008 m.m. 36,73 % baigiamųjų darbų temų buvo žodis *Programa*, 38,78 % baigiamųjų darbų temų buvo žodis *Sistema*, 20,41 % baigiamųjų darbų temų buvo žodis *Svetainė.* Nebuvo pasirinktas žaidimų kūrimas.

2008 – 2009 m.m. 35,85 % baigiamųjų darbų temų buvo žodis *Svetainė*, 24,53 % baigiamųjų darbų temų buvo žodis *Programa*, 22,64 % baigiamųjų darbų temų buvo žodis *Sistema*, 11,32 % baigiamųjų darbų temų buvo žodis *Modulis.*

2009 – 2010 m.m. 33,33 % baigiamųjų darbų temų buvo žodis *Svetainė*, 28,33 % baigiamųjų darbų temų buvo žodis *Programa*, 18,33 % baigiamųjų darbų temų buvo žodis *Modulis.*

2010 – 2011 m.m. 30,16 % baigiamųjų darbų temų buvo žodis *Svetainė*, 28,57 % baigiamųjų darbų temų buvo žodis *Programa*, 19,05 % baigiamųjų darbų temų buvo žodis *Sistema*, 17,46 % baigiamųjų darbų temų buvo žodis *Modulis.*

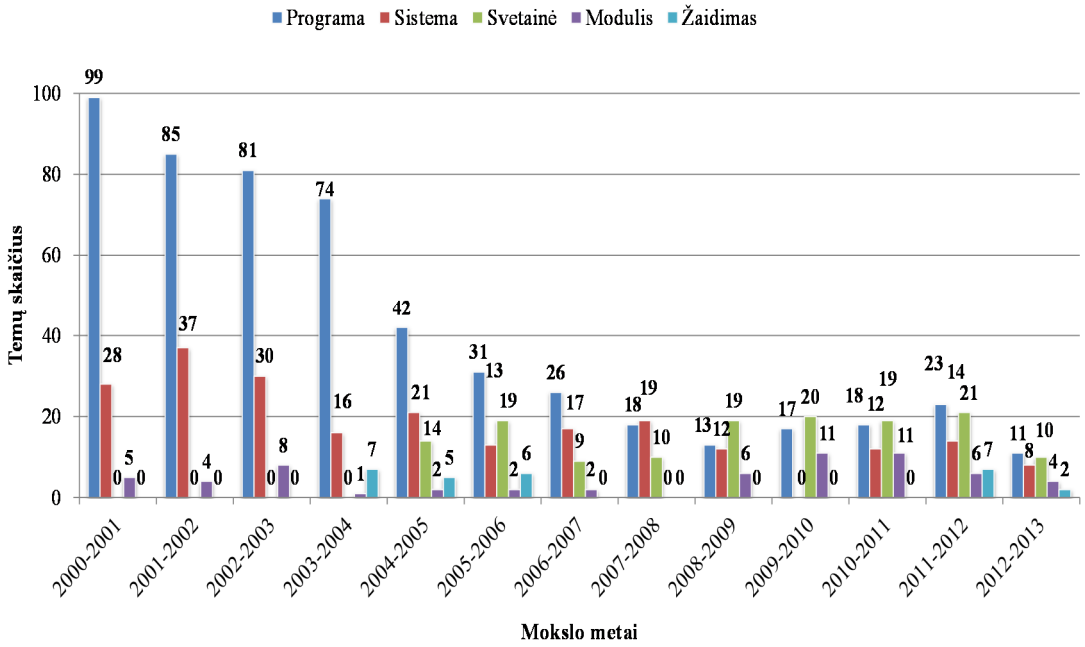
2011 – 2012 m.m. 31,94 % baigiamųjų darbų temų buvo žodis *Programa*, 29,16 % baigiamųjų darbų temų buvo žodis *Svetainė*, 19,44 % baigiamųjų darbų temų buvo žodis *Sistema*, 9,72 % baigiamųjų darbų temų buvo žodis *Žaidimas*, 8,33 % baigiamųjų darbų temų buvo žodis *Modulis.*

2012 – 2013 m.m. 30,56 % baigiamųjų darbų temų buvo žodis *Programa*, 27,78 % baigiamųjų darbų temų buvo žodis *Svetainė*, 22,22 %

baigiamųjų darbų temų buvo žodis *Sistema*, 11,11 % baigiamųjų darbų temų buvo žodis *Modulis*, 5,56 % baigiamųjų darbų temų buvo žodis *Žaidimas.*

Baigiamieji programavimo kompiuteriams studijų programos darbai turi praktinę vertę, nes buvo skirti konkrečioms įmonėms. Dažniau kuriamos programos ar sistemos, kurios skirtos konkrečiam vartotojui (-ams), tačiau nemažą dalį tarp baigiamųjų darbų užima ir internetinės svetainės. Rečiau renkamasi modulių ir žaidimų kūrimas.

2000 – 2013 m. baigiamųjų darbų temų pasirinkimas pavaizduotos diagramoje 12 paveiksle.



12 pav. 2000 – 2013 m. baigiamųjų darbų temų pasirinkimas.

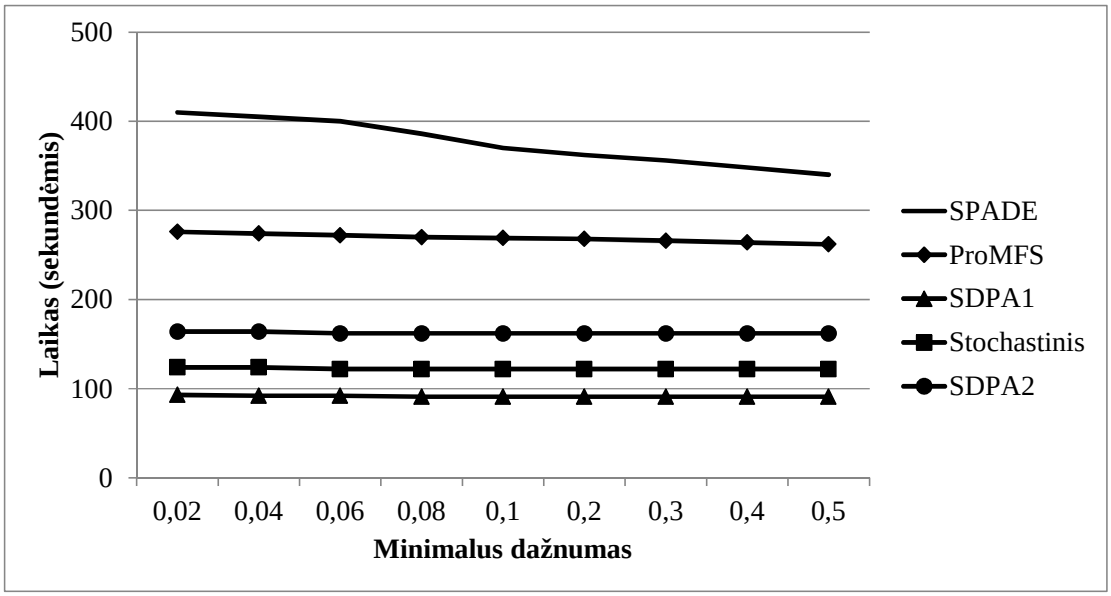
2000 – 2013 m. iš visų baigiamųjų darbų buvo sukurta 52,23 % įvairių taikomųjų *Programų*, 22,04 % įvairių *Sistemų*, 13,69 % internetinių *Svetainių*, 6,02 % įvairių *Modulių* bei 2,62 % *Žaidimų*, 3,4 % buvo pasirinktos kitos temos.

5.4. Transakcijų duomenų bazės tyrimas

Nagrinėjama pirkimų transakcijų duomenų bazė, kurią sudaro 400 000 transakcijų. Duomenų bazėje visi analizuojami elementai vienarūšiai.

Pirkinių krepšelį sudaro prekės, kurios turi tuos pačius atributus, išskyrus pavadinimą. Duomenų bazėje yra 25 skirtingų pavadinimų prekės.

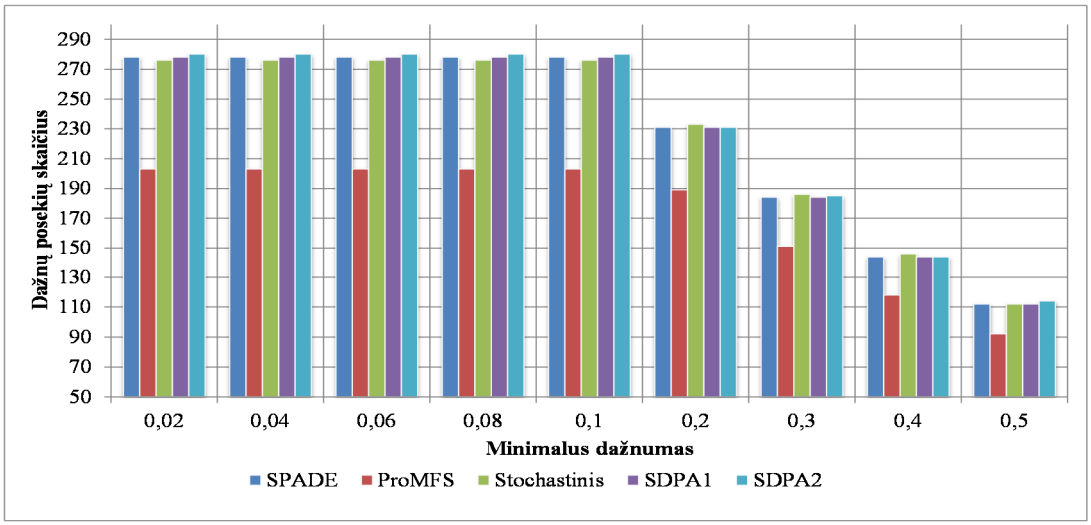
Transakcijų duomenų bazė apdorojama stochastiniu dažnų posekių paieškos, SDPA1, SDPA2, SPADE ir tikimybiniu dažnų sekų nustatymo ProMFS algoritmais. Eksperimento metu pasirinktos minimalaus dažnumo reikšmės *min_supp* = 0,02; 0,04; 0,06; 0,08; 0,1; 0,2; 0,3; 0,4; 0,5, o posekio ilgis yra intervale [1; 10], t.y. nustatomas ne didesnis nei 10 prekių pirkinių krepšelis. SDPA1 algoritmo parametro *g* reikšmė pasirinkta 0,5, o SDPA2 algoritme vieno – elemento dažni posekiai buvo nustatyti GSP algoritmu. ProMFS algoritmo sudarytos modelinės sekos ilgis 1342 elementai, šios sekos apdorojimui buvo naudojamas GSP algoritmas bei *min_supp* = 4; 6; 8; 10; 12; 14; 16; 18; 20. Buvo palygintos algoritmų laiko sąnaudos, nustatytų dažnų posekių skaičiaus kitimas nuo nustatyto minimalaus dažnumo. Stochastiniu susietumo taisyklių paieškos algoritmu nustatytos susietumo taisyklės šioje duomenų bazėje. Algoritmų laiko sąnaudų palyginimas esant skirtingoms minimalaus dažnumo reikšmėms pateiktas 13 paveiksle.



13 pav. Algoritmų laiko sąnaudų palyginimas.

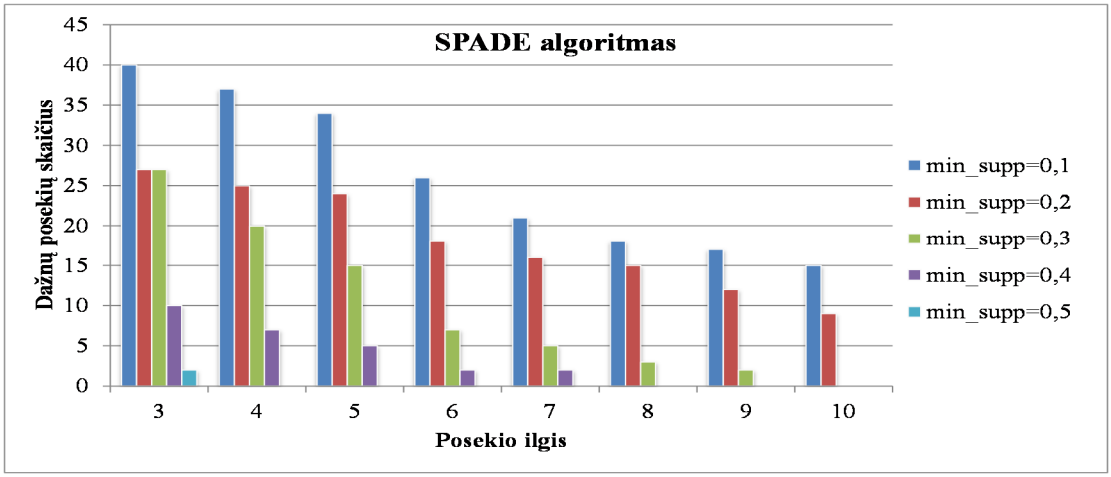
SPADE algoritmo dažnų posekių skaičius pradeda mažėti, kai *min_supp* = 0,2; 0,3; 0,4; 0,5, o ProMFS algoritmo – kai *min_supp* = 14; 16; 18; 20. Algoritmų dažnų posekių skaičiaus kitimas esant atitinkamoms skirtingoms

minimalaus dažnumo reikšmėms pateiktas 14 paveiksle. Stochastiniu dažnų posekių paieškos, SDPA1, SDPA2 ir SPADE algoritmais surasti tie patys dažni posekiai, tačiau stochastiniai algoritmai suranda mažesnį dažnų posekių pasikartojimų skaičių. Tikimybinio dažnų sekų nustatymo ProMFS algoritmas nustatė ne visus dažnus posekius. ProMFS algoritmo dažnų posekių praradimas, kai *min_supp* = 0,02; 0,04; 0,06; 0,08; 0,1 vidutiniškai 27 %, o kai *min_supp* = 0,2; 0,3; 0,4; 0,5 vidutiniškai 18 %.

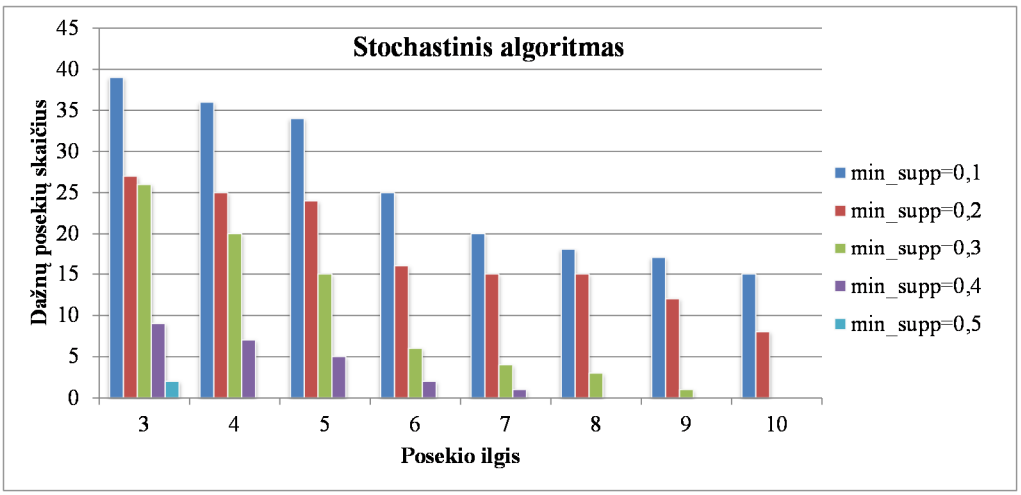


14 pav. Algoritmų dažnų posekių skaičiaus palyginimas.

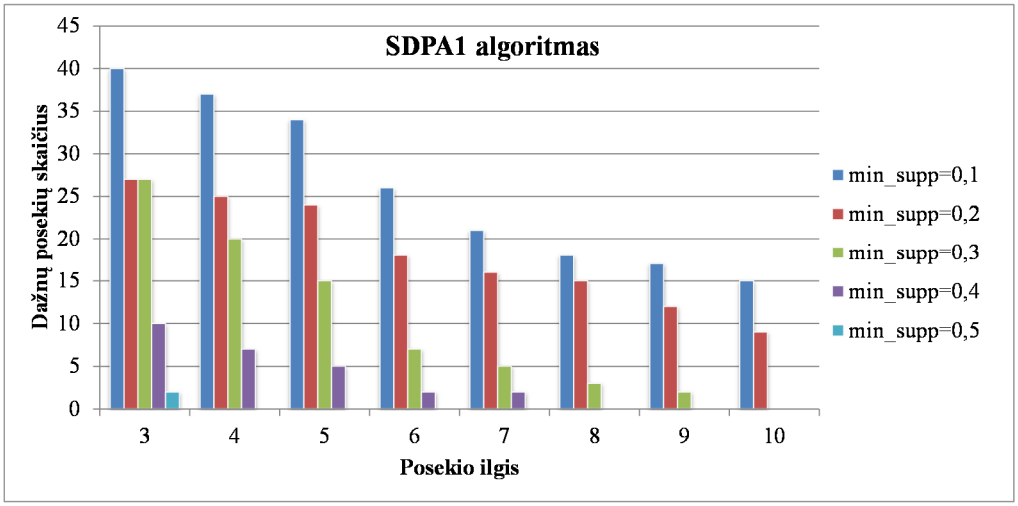
Eksperimente buvo įvertintas ir palygintas stochastinio dažnų posekių paieškos, SDPA1, SDPA2 ir SPADE algoritmų posekių skaičius kitimas pagal posekio ilgį. Tam tikro ilgio dažnų posekių skaičiaus palyginimas nuo nustatyto minimalaus dažnumo pateiktas 15 – 18 paveiksluose.



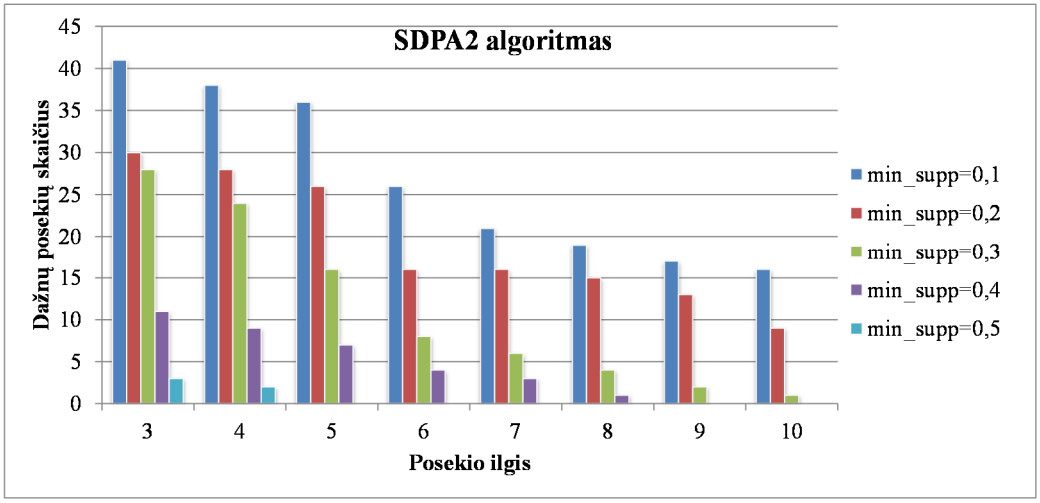
15 pav. SPADE algoritmo dažnų posekių skaičiaus palyginimas pagal posekių ilgį.



16 pav. Stochastinio algoritmo dažnų posekių skaičiaus palyginimas pagal posekių ilgį.



17 pav. SDPA1 algoritmo dažnų posekių skaičiaus palyginimas pagal posekių ilgį.

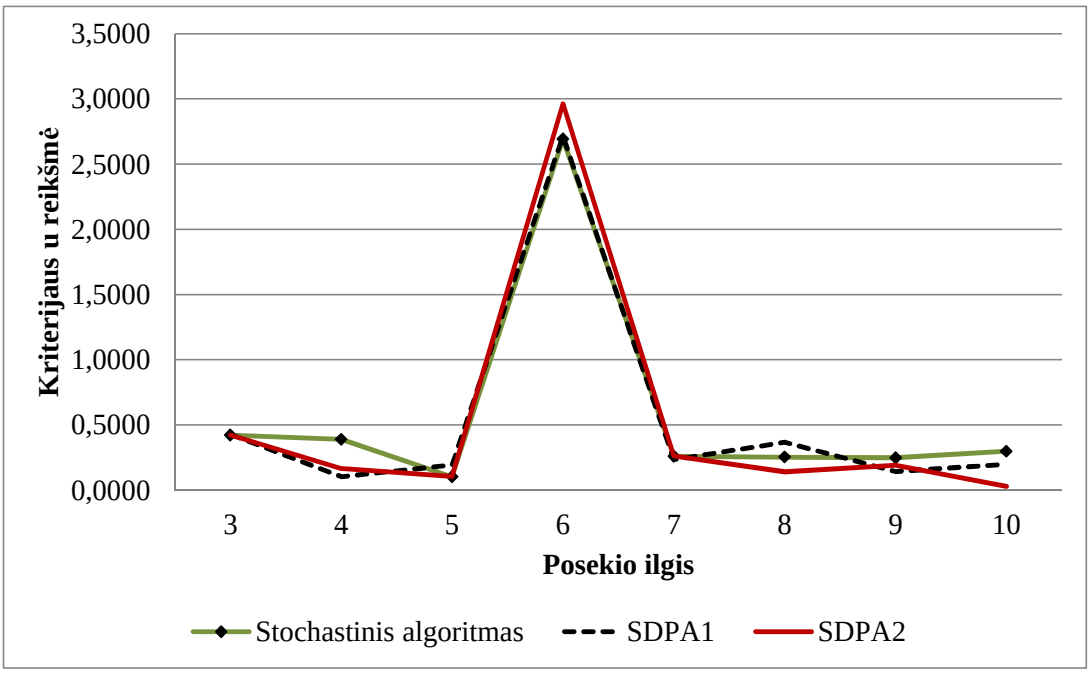


18 pav. SDPA2 algoritmo dažnų posekių skaičiaus palyginimas pagal posekių ilgį.

Kuo didesnis posekio ilgis, tuo mažiau yra dažnų posekių. Iš rezultatų matoma, kad 3 – elementų, 4 – elementų ir 5 – elementų dažnų posekių skaičius mažėja nežymiai, tačiau akivaizdus dažnų posekių skaičiaus sumažėjimas, kai posekio ilgis yra 5 ir daugiau elementų.

Norint nustatyti, koks yra didžiausias dažnų posekių ilgis, kai tokio posekių skaičius yra didžiausias galima pasinaudoti tikimybinių charakteristikų *u*, *z* ir tikėtinumo funkcijos *P*(0) reikšmių įvertinimu, kurie aprašyti 4.4. skyriuje.

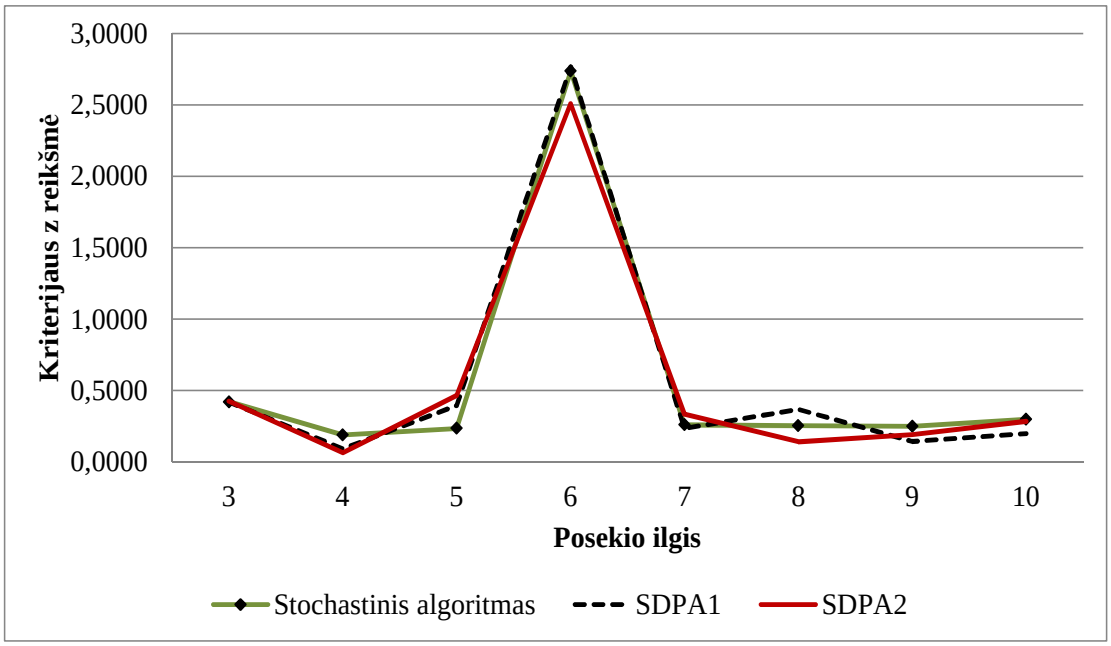
Eksperimento metu įvertintos stochastinio dažnų posekių paieškos, SDPA1, SDPA2 algoritmų tikimybinės charakteristikos *u*, *z* ir tikėtinumo funkcijos *P*(0) reikšmės. Šių charakteristikų reikšmių vertinimas leidžia išskirti, koks yra didžiausias dažnų posekių ilgis, kai tokių posekių skaičius yra didžiausias. Nagrinėjamos dvi nepriklausomos posekių imtys, jų dydžiai yra n_1 ir n_2 ir pirmojoje imtyje dažniausias posekis pasitaikė k_1 kartų, o antrojoje – k_2 kartų. Kriterijaus *u* statistika apibrėžia dviejų gretimo ilgio posekių dažnumo sutapimą. Kriterijaus *u* statistikos kitimas priklausomai nuo posekio ilgio pateiktas 19 paveiksle. Posekio ilgis kinta nuo 3 – elementų iki 10 – elementų.



19 pav. Kriterijaus statistikos *u* kitimas.

Įvertinus dviejų gretimo ilgio posekių dažnumo sutapimo kriterijaus statistiką *u* stochastiniu dažnų posekių paieškos, SDPA1, SDPA2 algoritmais nustatyta, kad dažniausias yra 5 – elementų posekis, t.y. dažniausią prekių krepšelį sudaro 5 elementai, nes kai posekio ilgis lygus 6, statistikos *u* reikšmė ženkliai padidėja, o tai reiškia, kad 6 – elementų posekio dažnumas žymiai mažesnis už 5 – elementų posekių dažnumą.

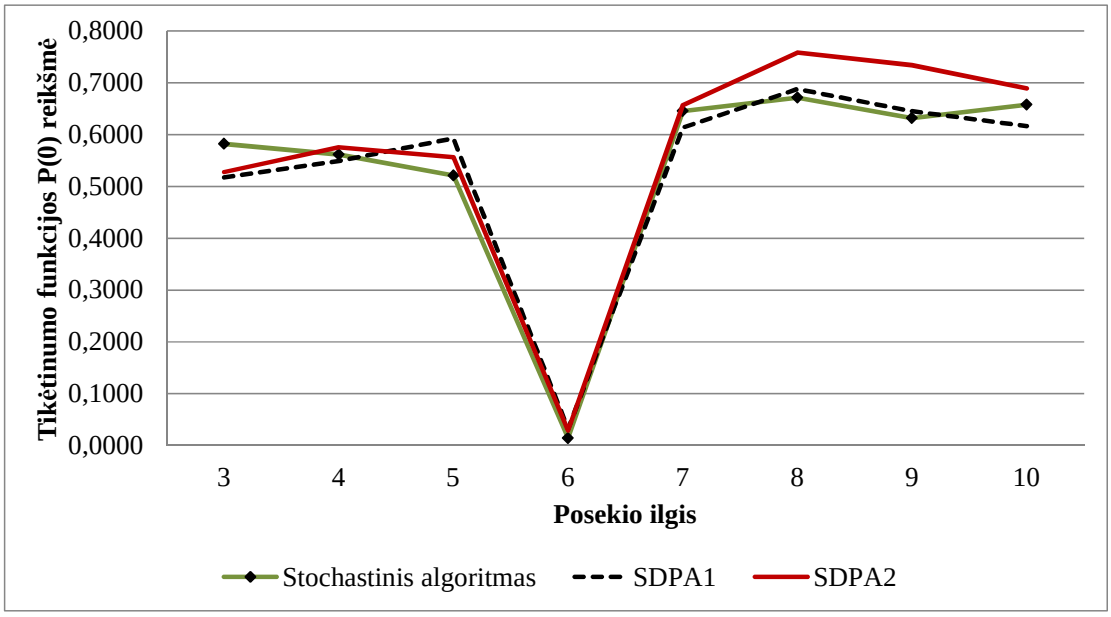
Kriterijaus *z* statistika taip pat apibrėžia dviejų gretimo ilgio posekių dažnumo sutapimą. Kriterijaus *z* statistikos kitimas, priklausomai nuo posekio ilgio, kai posekio ilgis kinta nuo 3 iki 10, pateiktas 20 paveiksle.



20 pav. Kriterijaus statistikos *z* kitimas.

Įvertinus dviejų gretimo ilgio posekių dažnumo sutapimo kriterijaus statistiką *z* stochastiniu dažnų posekių paieškos, SDPA1, SDPA2 algoritmais nustatyta, kad dažniausias yra 5 – elementų posekis, t.y. dažniausią prekių krepšelį sudaro 5 elementai, nes kai posekio ilgis lygus 6, statistikos *z* reikšmė ženkliai padidėja, o tai reiškia, kad 6 – elementų posekio dažnumas žymiai mažesnis už 5 – elementų posekių dažnumą.

Tikėtinumo funkcijos *P*(0) reikšmių kitimas, kai posekio ilgis kinta nuo 3 iki 10, pateiktas 21 paveiksle.



21 pav. Tikėtinumo funkcijos *P*(0) reikšmių kitimas.

Įvertinus tikėtinumo funkcijos *P*(0) reikšmes, stochastiniu dažnų posekių paieškos, SDPA1, SDPA2 algoritmais nustatyta, kad dažniausias yra 5 – elementų posekis, t.y. dažniausią prekių krepšelį sudaro 5 elementai, nes kai posekio ilgis lygus 6, tikėtinumo funkcijos *P*(0) reikšmė ženkliai sumažėja, o tai reiškia, kad 6 – elementų posekio dažnumas žymiai mažesnis už 5 – elementų posekių dažnumą.

Transakcijų duomenų bazė apdorota stochastiniu susietumo taisyklių paieškos algoritmu. Pasirinkus skirtingas susietumo taisyklių dažnumo ir patikimumo reikšmes stochastinis algoritmas nustatė susietumo taisykles bei apibendrintas susietumo taisykles. Apibendrintos susietumo taisyklės buvo sudaromos naudojant dvi prekių grupes. Stochastinio susietumo taisyklių paieškos algoritmo nustatytas susietumo taisyklių skaičius, esant skirtingoms dažnumo ir patikimumo reikšmėms, pateiktos 8 lentelėje.

Tiriamoje transakcijų duomenų bazėje, esant dažnumo reikšmei *min_supp* > 40% nebegalima sudaryti susietumo taisyklių. Esant dažnumo reikšmei *min_supp* > 25% nebegalima sudaryti susietumo taisyklių, kuriose būtų penki elementai. Susietumo taisyklės, kai *min_supp* = 25 %, *min_conf* = 70 % ir *min_supp* = 30 %, *min_conf* = 30 % pateiktos 3 priede.

8 lentelė. Susietumo taisyklių skaičius.

Taisyklės dažnumas, %	Taisyklės patikimumas, %									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
10	107236	107236	103780	93435	87363	82099	63853	44827	44663	44662
20	3056	3056	3056	2961	2739	2258	1905	1401	1237	1236
25	484	484	484	484	452	370	288	157	134	133
30	70	70	70	70	70	68	46	27	23	22
40	10	10	10	10	10	10	10	7	3	2
50	0	0	0	0	0	0	0	0	0	0

Stochastinis susietumo taisyklių paieškos algoritmas per vieną sekundę vidutiniškai sudaro 19 susietumo taisyklių. Susietumo taisyklių skaičius ir laikas pateiktas 9 lentelėje.

9 lentelė. Susietumo taisyklių skaičius ir laikas.

Taisyklės dažnumas ir patikimumas, %	**Taisyklių skaičius**	**Laikas (sekundėmis)**	**Sudarytų taisyklių skaičius per 1 sekundę**
20; 100	1236	66,08	19
25; 10	484	27,67	17
25; 20	484	27,69	17
25; 30	484	27,69	17
25; 40	484	27,68	17
25; 50	452	25,68	18
25; 60	370	20,67	18
25; 70	288	20,66	14
25; 80	157	10,65	15
25; 90	134	9,65	14
25; 100	133	8,67	15
30; 10	70	3,27	21
30; 20	70	3,28	21
30; 30	70	3,27	21
30; 40	70	3,26	21
30; 50	70	3,19	22
30; 60	68	2,47	28
30; 70	46	2,17	21
30; 80	27	1,16	23
30; 90	23	1,16	20
30; 100	22	1,16	19
40; 10	10	0,54	19

Taisyklės dažnumas ir patikimumas, %	**Taisyklių skaičius**	**Laikas (sekundėmis)**	**Sudarytų taisyklių skaičius per 1 sekundę**
40; 20	10	0,54	19
40; 30	10	0,54	19
40; 40	10	0,54	19
40; 50	10	0,54	19
40; 60	10	0,52	19
40; 70	10	0,52	19
40; 80	7	0,46	15
40; 90	3	0,22	14
40; 100	2	0,12	17

Stochastinio susietumo taisyklių paieškos algoritmo nustatytos apibendrintos susietumo taisyklės pateiktos 10 lentelėje.

10 lentelė. Apibendrintų susietumo taisyklių skaičius.

Taisyklės dažnumas, %	**Taisyklės patikimumas, %**									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
10	1794	1703	1537	1212	1023	863	646	393	382	382
20	170	170	159	137	122	98	74	38	27	27
25	86	86	86	73	64	55	43	24	17	17
30	20	20	20	16	16	12	10	9	8	8
40	6	6	6	6	6	4	3	3	2	2
50	2	2	2	2	2	2	2	2	1	1
60	0	0	0	0	0	0	0	0	0	0

Tiriamoje transakcijų duomenų bazėje, esant dažnumo reikšmei didesnei nei 50 % nebegalima sudaryti apibendrintų taisyklių tenkinančių šią reikšmę.

Apibendrintos susietumo taisyklės, kai *min_supp* = 25 %, *min_conf* = 70 % ir *min_supp* = 30 %, *min_conf* = 30 % pateiktos 4 priede.

Tyrimo rezultatai parodė, kad svarbu tinkamai parinkti susietumo taisyklių dažnumo bei patikimumo reikšmes. Jei šios reikšmės labai mažos, tai sudaromas didelis skaičius susietumo taisyklių, nustačius dideles dažnumo bei patikimumo reikšmes bus sudaroma mažai taisyklių.

5.5. Penktojo skyriaus išvados

Naujai pasiūlyti stochastinis dažnų posekių paieškos, SDPA1 ir SDPA2 algoritmai tirtose duomenų bazėse nustato tuos pačius dažnus posekius kaip ir tikslieji algoritmai, skiriasi tik dažnų posekių pasikartojimų skaičius.

Naujai pasiūlyto SDPA1 algoritmo parametras *g* turi įtakos nustatytų dažnų posekių skaičiui, algoritmo vykdymo laikui. Kuo didesnė parametro *g* reikšmė, tuo didesnės algoritmo laiko sąnaudos ir didėja dažnų posekių skaičius, nes rečiau atliekamas atsitiktinio ilgio posekio praleidimas.

Naujai pasiūlytų stochastinio dažnų posekių paieškos, SDPA1 ir SDPA2 algoritmų greičiai buvo palyginti su Apriori, rekursiniu, GSP, SPADE ir ProMFS algoritmų veikimo greičiais. Eksperimentų rezultatai parodė, kad stochastiniai algoritmai veikia greičiausiai.

Eksperimentinėse duomenų bazėse naujai pasiūlytų stochastinių algoritmų vidutinė pirmos rūšies klaida yra apytiksliai 2,4 %, vidutinė antros rūšies klaida 5,6 %, o pasikliautinasis intervalas yra [0,95; 0,99].

Vilniaus kolegijos Elektronikos ir informatikos fakulteto Programavimo kompiuteriams studijų programos baigiamųjų darbų temų duomenų bazė analizuota tiksliuoju SPADE algoritmu bei apytiksliais ProMFS, stochastiniu dažnų posekių ir susietumo taisyklių paieškos, SDPA1 algoritmais. Visi algoritmai nustatė tuos pačius dažniausius posekius. Šioje duomenų bazėje stochastinio susietumo taisyklių paieškos ir SDPA1 algoritmo laiko sąnaudos mažesnės nei ProMFS algoritmo. Naujai pasiūlytas stochastinis susietumo taisyklių paieškos algoritmas nustatė 11 susietumo taisyklių, kurios tenkino pasirinktą minimalaus patikimumo reikšmę *min_conf* = 2%.

Transakcijų duomenų bazė buvo apdorojama stochastiniu dažnų posekių paieškos, SDPA1, SDPA2, SPADE ir tikimybiniu dažnų sekų nustatymo ProMFS algoritmais. Eksperimento rezultatai parodė, kad greičiausiai veikia SDPA1 algoritmas, nežymiai didesnės laiko sąnaudos buvo stochastinio dažnų posekių paieškos algoritmo, trečioje vietoje – SPDA2, ketvirtoje

vietoje – ProMFS algoritmas, o didžiausios laiko sąnaudos – SPADE algoritmo.

Naujai pasiūlyti stochastinis dažnų posekių paieškos, SDPA1, SPDA2 algoritmai nustatė tuos pačius dažnus posekius kaip ir tikslusis SPADE algoritmas.

Įvertinus stochastinio dažnų posekių paieškos, SDPA1, SDPA2 algoritmų tikimybines charakteristikas *u, z* ir tikėtinumo funkcijos *P(0)* reikšmes, nustatyta, kad dažniausias maksimalaus ilgio posekis yra sudarytas iš 5 elementų.

Naujai pasiūlytu stochastiniu susietumo taisyklių paieškos algoritmu nustatytos susietumo taisyklės, esant skirtingoms minimalaus dažnumo ir minimalaus patikimumo reikšmėms. Kai minimalaus dažnumo ir minimalaus patikimumo reikšmės labai mažos algoritmas sudaro didelį skaičių susietumo taisyklių, didinat šias reikšmes susietumo taisyklių skaičius mažėja. Pasirinkus minimalaus dažnumo ir minimalaus patikimumo reikšmes intervale [30; 70], algoritmas per vieną sekundę nustatė vidutiniškai 19 susietumo taisyklių. Šis algoritmas nėra imlus laikui, todėl tinkamas darbui su didelėmis duomenų bazėmis.

Bendros išvados

Šios disertacijos tyrimų objektas yra duomenų tyrybos algoritmai ir metodai, skirti dažnų posekių ir susietumo taisyklių nustatymo uždaviniams spręsti. Tyrimo metu buvo išnagrinėti populiariausi dažnų posekių radimo būdai ir algoritmai bei metodologijos. Dažnų posekių ir susietumo taisyklių paieškai didelėse duomenų bazėse buvo sukurtas apytikslis stochastinis dažnų posekių paieškos algoritmas bei šio algoritmo modifikacijos SDPA1, SDPA2 ir stochastinis susietumo taisyklių paieškos algoritmas. Naujai pasiūlytų stochastinių algoritmų pagrindinė strategija – vieno duomenų bazės skenavimo metu atrinkti atsitiktinio ilgio posekius, praleidžiant atsitiktinį simbolių skaičių. Paimamų posekių ilgis ir praleidžiamų simbolių skaičius yra pasiskirstę pagal tolygųjį skirstinį. Naujai pasiūlyti algoritmai leidžia suderinti du svarbius kriterijus, t.y. tikslumą ir laiką. Šių algoritmų eksperimentinis tyrimas buvo atliktas naudojant realias bei imitacines duomenų bazes. Eksperimentų rezultatai palyginti su kitas egzistuojančiais tiksliaisiais bei apytiksliais algoritmais. Šioje disertacijoje atlikti tyrimai leido padaryti tokias išvadas:

1. Palyginus tiksliuosius SPADE, GSP, Apriori, rekursinį dažnų posekių paieškos algoritmus rezultatai parodė, kad greičiausiai veikia SPADE algoritmas, o rekursinio algoritmo laiko sąnaudos yra pačios didžiausios.
2. Naujai pasiūlytų stochastinio dažnų posekių paieškos, SDPA1, SDPA2 algoritmų tikslumas ištirtas empyriniais eksperimentais. Eksperimentams naudotose duomenų bazėse stochastinių algoritmų vidutinė pirmos rūšies klaida yra apie 2,4 %, vidutinė antros rūšies klaida – apie 5,6 %, pasikliautinasis intervalas yra [0,95; 0,99].
3. Darbe sukurtų stochastinio dažnų posekių paieškos, SDPA1, SPDA2 algoritmų greičiui neturi įtakos skirtingų duomenų bazės elementų skaičius, nes duomenų bazės skenavimo metu pasirenkami atsitiktinio ilgio posekiai.

4. Eksperimentinėse duomenų bazėse naujai pasiūlyti stochastinis dažnų posekių paieškos, SDPA1, SDPA2 algoritmai nustatė tuos pačius dažnus posekius kaip ir tikslieji algoritmai.
5. Greičiausias iš visų pasiūlytų stochastinių algoritmų yra SDPA1 algoritmas, kai algoritmo parametro reikšmė $g \in [0; 0,6]$. Kai reikšmė $g \in (0,6; 1]$, tai algoritmų SDPA1 ir stochastinio dažnų posekių paieškos algoritmo vykdymo laikas apytiksliai vienodas. SDPA2 algoritmo laiko sąnaudos didesnės už SDPA1 ir stochastinio dažnų posekių paieškos algoritmų, tačiau didesnis dažnų posekių skaičius.
6. Eksperimentinėse duomenų bazėse stochastinis susietumo taisyklių paieškos algoritmas per vieną sekundę sudarė vidutiniškai 19 susietumo taisyklių.
7. Naujai pasiūlyti stochastinis dažnų posekių paieškos, SDPA1, SDPA2 ir stochastinis susietumo taisyklių paieškos algoritmai yra greiti ir pakankamai tikslūs.

Literatūros sąrašas

[1] Adamo, J.M. Data Mining for Association Rules and Sequential Patterns. Springer-Verlag New York, Inc., 2001.

[2] Aflori, C., Craus, M. Grid implementation of the Apriori algorithm. Advances in Engineering Software, vol. 38, issue 5, p. 295-300, 2007.

[3] Afrati, F., Gionis, A., Mannila, H. Approximating a collection of frequent item sets. Proceedings ACM SIGKDD conference, p. 12–19, 2004.

[4] Ayres, J., Flannick, J., Gehrke, J., Yiu, T. Sequential Pattern mining using a bitmap representation. Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 429–435. ACM Press, Ed-monton, Alberta, Canada, 2002.

[5] Agrawal, R., Srikant, R. Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, p. 487-499. Morgan Kaufmann, Santiago de Chile, Chile, 1994.

[6] Aggarwal, C.C., Yu, P.S. A new framework for itemset generation. Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of Database Systems, p.18-24, Seattle, Washington, USA, ACM Press, 1998.

[7] Agrawal, R., Srikant, R. Mining sequential patterns. Proceedings of the Eleventh International Conference of the 11th Int'l Conference on Data Engineering. IEEE, p. 3-14, 1995.

[8] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A. I. Fast discovery of association rules. Advances in Knowledge Discovery and Data Mining, p. 307-328, AAAI Press, 1996.

[9] Agrawal, R., Imielinski, T., Swami, A. Mining association rules between sets of items in large databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, p. 207–216. ACM Press, Washington, D.C., 1993.

[10] Айвазян,С. А., Мхитарян, В.С. Прикладная ятатистика и основы эконометрики. Москва, Издательство „Юнити“.

[11] Bayardo, R.J. Efficiently mining long patterns from databases. SIGMOD 1998, p. 85–93, 1998.

[12] Bayardo, R., Agrawal, R. Mining the most interesting rules. Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD '99), p. 145-154, San Diego, California, USA, 1999.

[13] Bayardo, R., Agrawal, R., Gunopulos, D. Constraint – based rule mining in large, dense databases. Data Mining and Knowledge Discovery, vol. 4 (2/3), p. 217-240, 2000.

[14] Basel A. Mahafzah, Amer F. Al-Badarneh, Mohammed Z. Zakaria. A new sampling technique for association rule mining. Journal Of Information Science, vol.35, p. 358-376, 2009.

[15] Bodon, F. A Fast Apriori Implementation. Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, vol. 90 of CEUR Workshop Proceedings, 2003.

[16] Borgelt, Ch., Kruse, R. Induction of Association Rules: Apriori Implementation. Proceedings in Computational Statistics, p. 395-400, Physica-Verlag HD, 2002.

[17] Brin, S., Motwani, R., Silverstein, C. Market Baskets: Generalizing Association Rules to Correlations. Proceedings ACM SIGMOD International Conference on Management of Data, p. 265–276. ACM Press, Tucson, Arizona, 1997.

[18] Brin, S., Motwani, R., Ullman, J., Tsur, S. Dynamic item-set counting and implication rules for market basket data. Proceedings of the ACM SIGMOD International Conference on Management of Data, vol. 26, ACM, p. 255–264, 1997.

[19] Burdick, D., Calimlim, M., Gehrke, J. Mafia: A maximal frequent itemset algorithm for transactional databases. ICDE 2001, Heidelberg, Germany, 2001.

[20] Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., Zanasi, A. Discovering Data Mining: Form Consepts to Implementation. Prentice Hall PTR. 1997.

[21] Cai-Yan, J., Xie-Ping, G. Multi-scaling sampling: an adaptive sampling method for discovering approximate association rules. Journal of Computer Science and Technology, vol. 20, p. 309-318. Kluwer Academic Publishers, 2005.

[22] Cheung, D. W., Han, J., Ng, V. T., Fu, A. W., Fu, Y. A Fast Distributed Algorithm for Mining Association Rules. Proceedings of of the Fourth International Conference on Parallel and Distributed Information Systems, p.31-43. IEEE Computer Society, Miami Beach, Florida, USA, 1996.

[23] Chiang, R.H.L., Huang Cecil, C.E., Lim, E.P. Linear correlation discovery in databases: a data mining approach. Data & Knowledge Engineering, vol. 53(3), p. 311–337, 2005.

[24] Choh Man Teng. A Comparison of Standard and Interval Association Rules. Proceedings of the Sixteenth International FLAIRS Conference, p.371-375, 2003.

[25] Cho; Ch.W., Wu; Y.H., Chen, A.L.P. Effective Database Transformation and Efficient Support Computation for Mining Sequential Patterns. Database Systems for Advanced Applications, vol. 3453, p. 163–174. Springer, Berlin, Heidelberg, 2005.

[26] Coenen, F., Goulbourne, G., Leng, P. Tree Structures for Mining Association Rules. Data Mining and Knowledge Discovery, vol. 8, p. 25–51. Kluwer Academ-ic Publishers, 2004.

[27] Cormen, T. H., Leiserson, Ch. E., Rivest, R. L., Stein, C. Introduction to Algorithms (3th edition). The MIT Press Cambridge, London, 2009.

[28] Čekanavičius, V., Murauskas, G. Statistika ir jos taikymai. TEV, Vilnius, 2000.

[29] Das, A., Ng, W.K., and Woon, Y, K. Rapid association rule mining. Proceedings of the tenth international conference on Information and knowledge management, p. 474-481, ACM press, 2001.

[30] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. From data mining to knowledge discovery in databases [interaktyvus]. AI Magazine 17, p. 37–54, 1996. [žiūrėta 2013 m. lapkričio 21 d.]. Prieiga per internetą: <http://www.kdnuggets.com/gpspubs/aimag-kdd-overview-1996-Fayyad.pdf>.

[31] Garofalakis, M., Rastogi, R., Shim, K. SPIRIT: Sequential pattern mining with regular expression constraints. VLDB 1999, p. 223–234, San Francisco, Morgan Kaufmann, 1999.

[32] Gharib, T. F., Nassar, H., Taha, M., Abraham, A. An efficient algorithm for incremental mining of temporal association rules. Data & Knowledge Engineering, vol. 69, p. 737-880. North Holland, 2010.

[33] Gyenesei, A., Teuhola, J. Probabilistic Iterative Expansion of Candidates in Mining Frequent Itemsets. Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, p.192-195. Melbourne, Florida, USA, 2003.

[34] Gyenesei, A., Teuhola, J. A Probabilistic Approach to Mining Fuzzy Frequent Patterns. Classification and Clustering for Knowledge Discovery, p. 73-89, 2005.

[35] Gyenesei, A., Teuhola, J. Interestingness Measures for Fuzzy Association Rules. Proceedings 5th European Conference, PKDD 2001, p. 152-164, Freiburg, Germany, 2001.

[36] Gyenesei, A. A Fuzzy Approach for Mining Quantitative Association Rules. Acta Cybernetica, vol.15, p.305-320, 2001.

[37] Gyenesei, A., Teuhola, J. A Probabilistic Approach to Mining Fuzzy Frequent Patterns. Classification and Clustering for Knowledge Discovery, p.73-89, 2005.

[38] Gyenesei, A. Mining Weighted Association Rules for Fuzzy Quantitative Items. Proceedings 4th European Conference, PKDD 2000, p. 416-423, Lyon, France, 2000.

[39] Gyenesei, A., Schlapbach, R., Stolte, E., Wagner, U. Frequent Pattern Discovery Without Binarization: Mining Attribute Profiles. Proceedings 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, p. 528-535, Berlin, Germany, 2006.

[40] Gouda, K., Hassaan, M., Zaki, M.J. Prism: A primal-encoding approach for frequent sequence mining. Data Mining. ICDM 2007. Seventh IEEE International Conference, p. 487-492, 2007.

[41] Gouda, K., Hassaan, M., Zaki, M.J. Prism: An effective approach for frequent sequence mining via prime-block encoding. Journal of Computer and System Sciences, vol. 76(1), p. 88-102, 2010.

[42] Grahne, G., Zhu, J. Efficiently using prefix-trees in mining frequent itemsets. Proceeding of the ICDM'03 international workshop on frequent itemset mining implementations (FIMI'03), Melbourne, p. 123–132, 2003.

[43] Han, J., Dong, G., Yin, Y. Efficient mining of partial periodic patterns in time series database. ICDE 1999, p. 106–115. Sydney, Australia, Mar, 1999.

[44] Han, J., Pei, J., Yin, Y. Mining frequent patterns without candidate generation. ACM SIGMOD Record, vol. 29, ACM, p. 1-12, 2000.

[45] Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. Proceedings of the 17th International Conference on Data Engineering, p. 215-224, 2001.

[46] Han, J., Cheng, H., Xin, D., Yan, X. Frequent pattern mining: current status and future directions. Data Mining and Knowledge Discovery, vol. 15(1), p. 55-86, 2007.

[47] Hipp, J., Güntzer, U., Nakhaeizadeh, G. Algorithms for Association Rule Mining – A Genereal Survey and Comparison. SIGKDD

Explorations, Newsletter, vol.2 (1), p. 58-64. ACM, New York, USA, 2000.

[48] Hipp, J., Güntzer, U., Grimmer, U. Integrating association rule mining algorithms with relational database systems. Proceedings of the 3rd International Conference on Enterprise Information Systems (ICEIS 2001), p. 130-137, Setubal, Portugal, 2001.

[49] Holt, J.D., Chung, S. M. Efficient Mining of Association Rules in Text Databases. CIKM'99, p. 234-242, Kansas City, USA, 1999.

[50] Huanyin, Z., Jinsheng, L. The Research of A-Priori Algorithm Candidates Based on Support Counts. International Conference on Information Technology and Computer Science, p.192-195. TBD, Kiev, Ukraine, 2009.

[51] Huang, Jin; Yin, Zhiben. Improvement of Apriori Algorithm for Mining Association Rules. Journal of University of Electronic Science and Technology of China. No.32(1), p..76-79, 2003.

[52] Inokuchi, A., Washio, T., Motoda, H. An Apriori-Based Algorithm for Mining Frequent Substructures from Graph Data. Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery, p. 13–23. Springer-Verlag, London, UK, 2000.

[53] Yang, Z., Wang, Y., Kitsuregawa, M. Effective sequential pattern mining algorithms for dense database. Japanese Data Engineering Workshop (DEWS), 2006.

[54] Juozapavičius, A. Duomenų struktūros ir efektyvūs algoritmai, Vilnius, TEV, 2007.

[55] Keith, C.C. Chan, Au, Wai-Ho. Mining Fuzzy Association Rules. Proceedings of CIKM Conference, p. 209-215, LasVegas, Nevada, USA, 1997.

[56] Klemetinen, L., Mannila, H., Ronkainen, P. Finding interesting rules from large sets of discovered association rules. Proceedings of the Third International Conference on Information and Knowledge Management, p. 401-407, Gaithersburg, USA, 1994.

[57] Kotasek, P., Zendulka, J. Comparison of Three Mining Algorithms for Association Rules. Proceedings of 34th Spring Int. Conf. on Modelling and Simulation of Systems (MOSIS'2000), Workshop Proceedings Information Systems Modelling (ISM'2000), p. 85-90, 2000.

[58] Kum, H. C., Pei, J., Wang, W., Duncan, D. ApproxMAP: Approximate Mining of Consensus Sequential Patterns. Proceedings of the 2003 SIAM International Conference on Data Mining (SIAM DM '03), p. 311–315, 2003.

[59] Lazcorrte, E., Botella, F., Fernandez-Caballero, A. Towards personalized recommendation by two-step modified Apriori data mining algorithm. Expert Systems with Applications, vol. 35, issue 3, p.1422-1429, 2008.

[60] Mannila, H., Toivonen, H., Verkamo, A. I. Discovering frequent episodes in sequences. KDD 1995, p. 210–215. Montreal, Quebec, Canada, 1995.

[61] Masseglia, F., Poncelet, P., Teisseire, M. Incremental mining of seqential patterns in large databases. Data& Knowledge Engineering, vol. 46, issue 1, p.97-121, 2003.

[62] Orlando, S., Palmerini, P., Parego, R. Enhancing the apriori algorithm for frequent set counting. Proceedings Third International Conference, DaWaK 2001, p. 71-82, Munich, Germany, 2001.

[63] Pallavi D. Association Rule Mining on Distributed Data. International Journal of Scientific & Engineering Research, vol. 3, p. 1-6. Research India Publications, 2012.

[64] Park, J. S., Chen, M. S., Yu, P. S. An effective hash-based algorithm for mining association rules. Proceedings of the 1995 ACM SIGMOD International Conference on Management of data, vol. 24, ACM, p. 175-186, 1995.

[65] Park, J. S., Chen, M. S., Yu, P. S. Efficient parallel data mining for association rules. Proceedings of the fourth international conference on Information and knowledge management, ACM, p. 31-36, 1995.

[66] Park, J., Chen, M., Yu, P. Using a Hash-based method with transaction trimming for mining association rules. IEEE Trans. Knowledge and Data Engineering, vol. 9, no. 5, p. 813–824, 1997.
[67] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C. Mining Sequential Patterns by Pattern-growth: The PrefixSpan Approach. TKDE, vol. 16, no 11, p. 1424-1440, 2004.
[68] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C. PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. Proceeding of the 2001international conference on data engineering (ICDE'01), Heidelberg, Germany, p. 215–224, 2001.
[69] Pragarauskaitė, J., Dzemyda, G. Probabilistic algorithm for mining frequent sequences. Proceedings ASMDA 2011, Sapienza University of Rome, Edizioni ETS, p. 1454-1460, 2011.
[70] Pragarauskaitė, J. Dažnų sekų analizė sprendimų priėmimui labai didelėse duomenų bazėse. (Disertacija). 2013.
[71] Rasoulian, M., Saeed, A. The Effect of Data Mining Based on Association Rules in Strategic Management. Journal of Basic and Applied Scientific Research, p. 1742-1748. TextRoad Publication, 2012.
[72] Raorane, A.A., Kulkarni, R.V., Jitkar, B.D. Association Rule – Extracting Knowledge Using Market Basket Analysis. Research Journal of Recent Sciences, vol. 1(2), p. 19-27, 2012.
[73] Sandhu, P.S., Dhaliwal, D.S., Panda, S.N. Mining utility-oriented association rules: An efficient approach based on profit and quantity. International Journal of the Physical Sciences, vol. 6(2), p. 301-307, 2011.
[74] Sarawagi, S., Thomas, S., Agrawal, R. Integrating association rule mining with relational database systems: Alternatives and implications. Data Mining and Knowledge Discovery, vol. 4(2), p. 89-125, 2000.
[75] Savasere, A., Omiecinski, E., Navathe, Sh. An Efficient Algorithm for Mining Association Rules in Large Databases. Proceedings of the 21st International Conference on Very Large Databases, p. 432–444. Zurich, Swizerland, 1995.

[76] Siebes, A., Vreeken, J., Leeuwen, M. Item Sets That Compress. Data Mining and Knowledge Discovery, vol. 23, p. 169-214, 2011.

[77] Silverstein, C., Brin, S., Motwani, R. Beyond Market Baskets: Generalizing Association Rules to Dependence Rules. Data Mining and Knowledge Discovery, vol. 2, p. 39-68, 1998.

[78] Soo, J., Chen, M.S., Yu, P.S. Using a Hash-Based Method with Transaction Trimming and Database Scan Reduction for Mining Association Rules. IEEE Transactions On Knowledge and Data Engineering, vol.5. p. 813-825, 1997.

[79] Sotiris, K., Kanellopoulos, D. Association Rules Mining: A Recent Overview. GESTS International Transactions on Computer Science and Engineering, vol.32 (1), pp. 71-82, 2006.

[80] Srikant, R., Agrawal, R. Mining generalized Association Rules. Proceeding VLDB ’95 Proceedings of the 21st International Conference on Very large Data Bases, p. 407-419. San Francisco, CA, USA, 1995.

[81] Srikant, R., Agrawal, R. Mining sequential patterns. ICDE '95 Proceedings of the Eleventh International Conference on Data Engineering, p.3-14. Taipei, Taiwan, 1995.

[82] Srikant, R., Agrawal, R. Mining sequential patterns: Generalizations and performance improvements. Advances in Database Technology EDBT'96, p. 1-17, 1996.

[83] Srikant, R., Agrawal, R. Mining generalized association rules. Future Generation Computer Systems, vol. 13, p. 161–180, 1997.

[84] Tien Dung Do, Siu Cheng Hui, Alvis Fong. Mining frequent itemsets with category Based Constraints. Lecture Notes in Computer Science, vol. 2843, p. 226-234, 2003.

[85] Tsau Young Lin. Sampling in association rule mining. Conference on Data mining and knowledge discovery: Theory, Tools, and Technology VI, vol. 5433, p. 161-167, 2004.

[86] Toivonen, H. Sampling Large Databases for Association Rules. Proceedings of the 22nd International Conference on Very Large Databases, p. 134–145. Mumbai, India, 1996.

[87] Toivonen, H., Klemettinen, M., Ronkainen, P., Hatonen, K., Mannila, H. Pruning and grouping discovered association rules. MLnet Workshop on Statistics, Machine Learning, and Discovery in Databases, p. 47-54, Heraklion, Crete, Greece, 1995.

[88] Tumasonis, R. Dažnų sekų paieška dideliuose duomenų masyvuose (Disertacija). 2007.

[89] Tumasonis, R; Dzemyda, G. The Probabilistic Algorihm for Mining Frequent Sequences. Proceedings ADBIS'04 Eight East-European Conference on Advances in Databases and Information Systems, p. 89–98, 2004.

[90] Tseng, M. C., Lin, W. Y. Efficient mining of generalized association rules with non – uniform minimum support. Data & Knowledge Engineering 62. Science Direct, p. 41-64, 2007.

[91] Umarani, V., Punithavalli, M. A study on effective mining of Association Rules from huge Databases. International Journal of Computer Science and Research, vol. 1, p. 30-34, 2010.

[92] Umarani, V., Punithavalli, M. Developing a Novel and Effective Approach for Association Rule Mining Using Progressive Sampling. Proceedings of 2nd International Conference on Computer and Electrical Engineering (ICCEE 2009), vol.1, p.610-614, 2009.

[93] Umarani, V., Punithavalli, M. On Developing an Effectual Progressive Sampling Based Approach for Association Rule Discovery. Proceedings of 2nd IEEE International Conference on Information and data Engineering (2nd IEEE ICIME 2010), Chengdu ,China, 2010.

[94] Урбах, В. Ю. Биометриеские методы. Москва, p. 93- 133, 1964.

[95] Wang, H., Liu, X. The Research of Improved Association Rules Mining Apriori Algorithm. Eighth International Conference on Fuzzy

Systems and Knowledge Discovery (FSKD), p. 961-964. IEEE, Shanghai, China, 2011.

[96] Wang, C., Tjortjis, C. Prices: An Efficient Algorithm for Mining Association Rules. Lecture Notes in Computer Science, vol. 2447, p. 77-83, 2002.

[97] Webb, G. Efficient search for association rules. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 99–107. Boston, MA, 2000.

[98] Xin, D., Han, J., Yan, X., Chend, H. Mining compressed frequent-pattern sets. Proceedings VLDB conference, p. 709–720, 2005.

[99] Yang, J., Zhao, C. Study on the Data Mining Algorithm Based on Positive and Negative Association Rules. Computer and Information Science, vol. 2, p. 103-106, 2009.

[100] Yang, Z., Wang, Y., Kitsuregawa, M. Effective sequential pattern mining algorithms for dense database. Japanese Data Engineering Workshop (DEWS), 2006.

[101] Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W. New Algorithms for Fast Discovery of Association Rules. Proceedings of the Third Int‘l Conference on Knowledge Discovery in Databases and Data Mining, p. 283-286, 1997.

[102] Zaki, M.J., Ogihara, M. Theoretical Foundations of Association Rules. 3rd SIGMOD’98 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD), p. 7:1-7:8, Seattle, WA, 1998.

[103] Zaki, M.J. Parallel and Distributed Association Mining: A Survey. IEEE Concurrency, special issue on Parallel Mechanisms for data Mining, vol. 7, no.4, p. 14-25, 1999.

[104] Zaki, M.J. Scalable Algorithms for Association Mining. IEEE Transactions on Knowledge and Data Engineering, vol. 12 (3), p. 372–390, 2000.

[105] Zaki, M.J. SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning, vol. 42, p. 31–60, 2001.

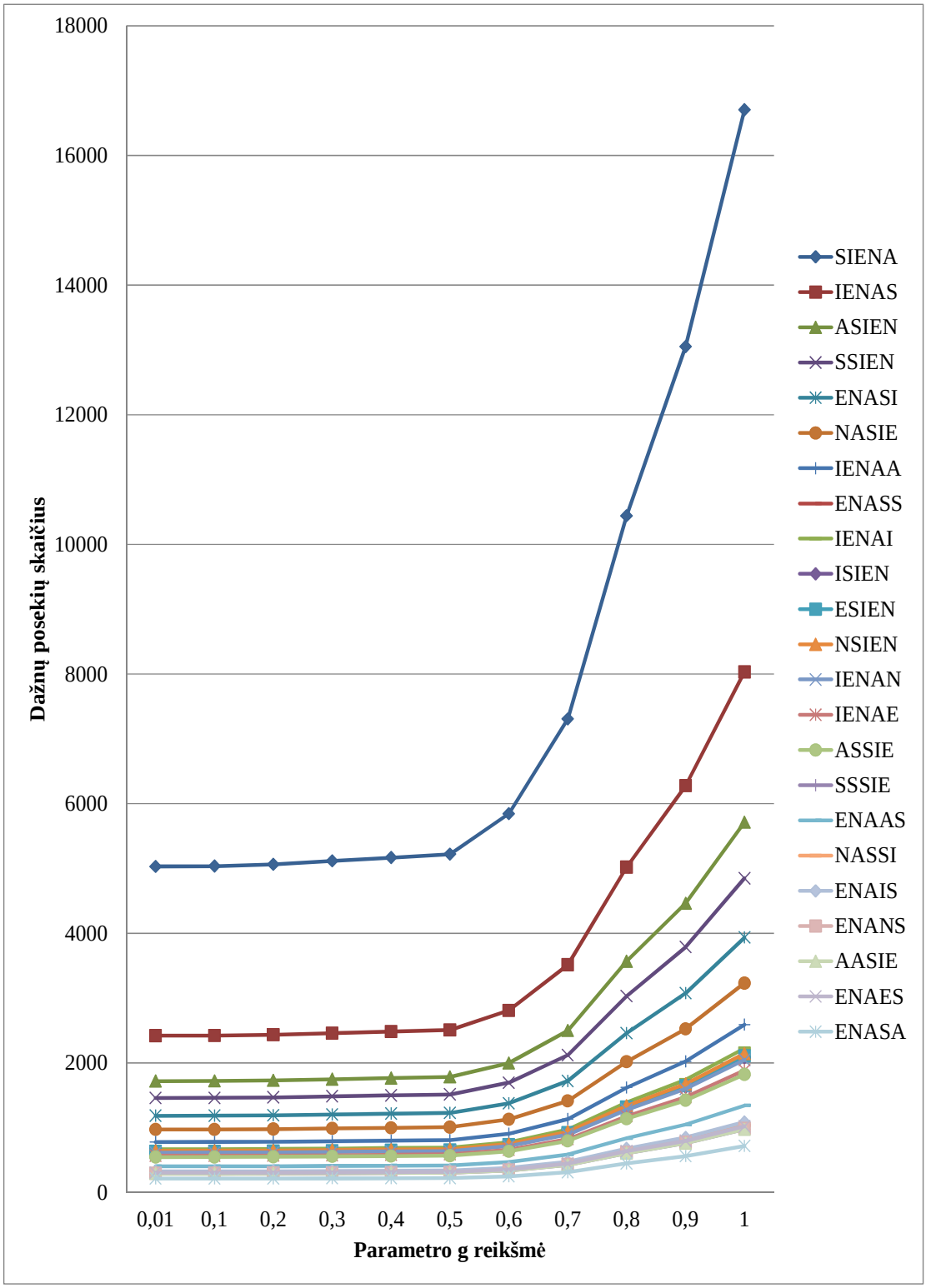
[106] Zhang, C., Zhang, S., Webb, G.I. Identifying Approximate Itemsets of Interest in Large Databases. Applied Intelligence, vol. 18, p. 91-104, 2003.

[107] Zhang, T. Association rules. Knowledge Discovery and Data Mining, Current Issues and New Applications. 4th Pacific-Asia Conference, PADKK 2000, Proceedings, Lecture Notes in Computer Science, vol. 1805. Springer, 2000.

[108] Zhao, Y., Zhang, C., Zhang, S. Efficient frequent itemsets mining by sampling. Proceedings of the fourth International Conference on Active Media Technology (AMT), p. 112-117, 2006.

PRIEDAI

Priedas 1. Dažnų posekių skaičiaus kitimas.



Priedas 2. Duomenų bazių grupės ir jų generavimo charakteristikos.

Kiekvienos duomenų bazės dydis yra 100 000 simbolių (1000 eilučių, kuriose yra po 100 simbolių), I={A, E, I, N, S}. Iš viso sugeneruota 1900 duomenų bazių, kuriose posekis SIENA yra įterptas tarp simbolių *A, E, I, N, S* su skirtingomis tikimybėmis. Pavyzdžiui, jei turime tokias generavimo charakteristikas SIENA – 0,1, A – 0,18, E – 0,18, I – 0,18, N – 0,18, S – 0,18, tai kiekvienoje eilutėje posekis SIENA pasikartos 10 kartų, visi kiti simboliai pasikartos atsitiktinai po 10 kartų. Lentelėje pateiktos generavimo charakteristikos.

Duomenų bazės pavadinimas	Simboliai ir jų generavimo tikimybės
0.1.*.txt	SIENA – 0,1; S – 0,18; I – 0,18; E – 0,18; N – 0,18; A – 0,18
0.09.*.txt	SIENA – 0,09; S – 0,19; I – 0,18; E – 0,18; N – 0,18; A – 0,18
0.08.*.txt	SIENA – 0,08; S – 0,19; I – 0,19; E – 0,18; N – 0,18; A – 0,18
0.07.*.txt	SIENA – 0,07; S – 0,19; I – 0,19; E – 0,19; N – 0,18; A – 0,18
0.06.*.txt	SIENA – 0,06; S – 0,19; I – 0,19; E – 0,19; N – 0,19; A – 0,18
0.05.*.txt	SIENA – 0,05; S – 0,19; I – 0,19; E – 0,19; N – 0,19; A – 0,19
0.04.*.txt	SIENA – 0,04; S – 0,2; I – 0,19; E – 0,19; N – 0,19; A – 0,19;
0.03.*.txt	SIENA – 0,03; S – 0,2; I – 0,2; E – 0,19; N – 0,19; A – 0,19
0.02.*.txt	SIENA – 0,02; S – 0,2; I – 0,2; E – 0,2; N – 0,19; A – 0,19
0.01.*.txt	SIENA – 0,01; S – 0,2; I – 0,2; E – 0,2; N – 0,2; A – 0,19
0.009.*.txt	SIENA – 0,009; S – 0,2; I – 0,2; E – 0,2; N – 0,2; A – 0,191
0.008.*.txt	SIENA – 0,008; S – 0,2; I – 0,2; E – 0,2; N – 0,2; A – 0,192
0.007.*.txt	SIENA – 0,007; S – 0,2; I – 0,2; E – 0,2; N – 0,2; A – 0,193
0.006.*.txt	SIENA – 0,006; S – 0,2; I – 0,2; E – 0,2; N – 0,2; A – 0,194
0.005.*.txt	SIENA – 0,005; S – 0,2; I – 0,2; E – 0,2; N – 0,2; A – 0,195
0.004.*.txt	SIENA – 0,004; S – 0,2; I – 0,2; E – 0,2; N – 0,2; A – 0,196
0.003.*.txt	SIENA – 0,003; S – 0,2; I – 0,2; E – 0,2; N – 0,2; A – 0,197
0.002.*.txt	SIENA – 0,002; S – 0,2; I – 0,2; E – 0,2; N – 0,2; A – 0,198
0.001.*.txt	SIENA – 0,001; S – 0,2; I – 0,2; E – 0,2; N – 0,2; A – 0,199

Priedas 3. Transakcijų duomenų bazėje išskirtos susietumo taisyklės.

Šiame priede pateiktos susietumo taisyklės, kai minimalus dažnumas *min_supp* = 25 % ir minimalus patikimumas *min_conf* = 70 % bei *min_supp* = 30 % ir *min_conf* = 30 %. Prekių pavadinimai žymimi A, B, ...

Susietumo taisyklių skaičius – 288 (*min_supp* = 25 % ir *min_conf* = 70).

(R) ⇒ (S) (dažnumas: 25%, patikimumas: 75%)
(S) ⇒ (L) (dažnumas: 29,17%, patikimumas: 77,78%)
(L) ⇒ (S) (dažnumas: 29,17%, patikimumas: 77,78%)
(R) ⇒ (L) (dažnumas: 25%, patikimumas: 75%)
(P) ⇒ (L) (dažnumas: 25%, patikimumas: 75%)
(L) ⇒ (M) (dažnumas: 29,17%, patikimumas: 77,78%)
(S) ⇒ (K) (dažnumas: 29,17%, patikimumas: 77,78%)
(R) ⇒ (K) (dažnumas: 25%, patikimumas: 75%)
(P) ⇒ (K) (dažnumas: 25%, patikimumas: 75%)
(M) => (K) (dažnumas: 41,67%, patikimumas: 90,91%)
(K) ⇒ (M) (dažnumas: 41,67%, patikimumas: 83,33%)
(L) ⇒ (K) (dažnumas: 37,5%, patikimumas: 100%)
(K) ⇒ (L) (dažnumas: 37,5%, patikimumas: 75%)
(I) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)
(I) ⇒ (K) (dažnumas: 29,17%, patikimumas: 87,5%)
(G) ⇒ (K) (dažnumas: 33,33%, patikimumas: 80%)
(I) ⇒ (G) (dažnumas: 25%, patikimumas: 75%)
(H) ⇒ (G) (dažnumas: 29,17%, patikimumas: 100%)
(G) ⇒ (H) (dažnumas: 29,17%, patikimumas: 70%)
(H) ⇒ (F) (dažnumas: 29,17%, patikimumas: 100%)
(F) ⇒ (H) (dažnumas: 29,17%, patikimumas: 77,78%)
(G) ⇒ (F) (dažnumas: 29,17%, patikimumas: 70%)
(F) ⇒ (G) (dažnumas: 29,17%, patikimumas: 77,78%)
(I) ⇒ (E) (dažnumas: 25%, patikimumas: 75%)
(H) ⇒ (E) (dažnumas: 29,17%, patikimumas: 100%)
(G) ⇒ (E) (dažnumas: 41,67%, patikimumas: 100%)
(E) ⇒ (G) (dažnumas: 41,67%, patikimumas: 83,33%)
(F) ⇒ (E) (dažnumas: 37,5%, patikimumas: 100%)
(E) ⇒ (F) (dažnumas: 37,5%, patikimumas: 75%)
(G) ⇒ (C) (dažnumas: 33,33%, patikimumas: 80%)
(F) ⇒ (C) (dažnumas: 29,17%, patikimumas: 77,78%)
(E) ⇒ (C) (dažnumas: 41,67%, patikimumas: 83,33%)
(C) ⇒ (E) (dažnumas: 41,67%, patikimumas: 76,92%)
(D) ⇒ (C) (dažnumas: 33,33%, patikimumas: 100%)
(D) ⇒ (B) (dažnumas: 25%, patikimumas: 75%)
(B) ⇒ (C) (dažnumas: 29,17%, patikimumas: 70%)

(D) ⇒ (A) (dažnumas: 29,17%, patikimumas: 87,5%)
(C) ⇒ (A) (dažnumas: 45,83%, patikimumas: 84,62%)
(A) ⇒ (C) (dažnumas: 45,83%, patikimumas: 78,57%)
(B) ⇒ (A) (dažnumas: 41,67%, patikimumas: 100%)
(A) ⇒ (B) (dažnumas: 41,67%, patikimumas: 71,43%)
(D) ⇒ (B, A) (dažnumas: 25%, patikimumas: 75%)
(D, A) ⇒ (B) (dažnumas: 25%, patikimumas: 85,71%)
(D, B) ⇒ (A) (dažnumas: 25%, patikimumas: 100%)
(B) ⇒ (C, A) (dažnumas: 29,17%, patikimumas: 70%)
(B, A) ⇒ (C) (dažnumas: 29,17%, patikimumas: 70%)
(C, B) ⇒ (A) (dažnumas: 29,17%, patikimumas: 100%)
(D) ⇒ (C, A) (dažnumas: 29,17%, patikimumas: 87,5%)
(D, A) ⇒ (C) (dažnumas: 29,17%, patikimumas: 100%)
(C, D) ⇒ (A) (dažnumas: 29,17%, patikimumas: 87,5%)
(D) ⇒ (C, B) (dažnumas: 25%, patikimumas: 75%)
(D, B) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(C, B) ⇒ (D) (dažnumas: 25%, patikimumas: 85,71%)
(C, D) ⇒ (B) (dažnumas: 25%, patikimumas: 75%)
(C, A) ⇒ (E) (dažnumas: 33,33%, patikimumas: 72,73%)
(E, A) ⇒ (C) (dažnumas: 33,33%, patikimumas: 100%)
(E, C) ⇒ (A) (dažnumas: 33,33%, patikimumas: 80%)
(E, C) ⇒ (F) (dažnumas: 29,17%, patikimumas: 70%)
(F) ⇒ (E, C) (dažnumas: 29,17%, patikimumas: 77,78%)
(F, C) ⇒ (E) (dažnumas: 29,17%, patikimumas: 100%)
(F, E) ⇒ (C) (dažnumas: 29,17%, patikimumas: 77,78%)
(F) ⇒ (H, E) (dažnumas: 29,17%, patikimumas: 77,78%)
(F, E) ⇒ (H) (dažnumas: 29,17%, patikimumas: 77,78%)
(H) ⇒ (F, E) (dažnumas: 29,17%, patikimumas: 100%)
(H, E) ⇒ (F) (dažnumas: 29,17%, patikimumas: 100%)
(H, F) ⇒ (E) (dažnumas: 29,17%, patikimumas: 100%)
(G, A) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(G, C) ⇒ (A) (dažnumas: 25%, patikimumas: 75%)
(E, A) ⇒ (G) (dažnumas: 25%, patikimumas: 75%)
(G, A) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(E, C) ⇒ (G) (dažnumas: 33,33%, patikimumas: 80%)
(G) ⇒ (E, C) (dažnumas: 33,33%, patikimumas: 80%)
(G, C) ⇒ (E) (dažnumas: 33,33%, patikimumas: 100%)
(G, E) ⇒ (C) (dažnumas: 33,33%, patikimumas: 80%)
(F) ⇒ (G, E) (dažnumas: 29,17%, patikimumas: 77,78%)
(F, E) ⇒ (G) (dažnumas: 29,17%, patikimumas: 77,78%)
(G) ⇒ (F, E) (dažnumas: 29,17%, patikimumas: 70%)
(G, E) ⇒ (F) (dažnumas: 29,17%, patikimumas: 70%)
(G, F) ⇒ (E) (dažnumas: 29,17%, patikimumas: 100%)

(H) ⇒ (G, E) (dažnumas: 29,17%, patikimumas: 100%)
(H, E) ⇒ (G) (dažnumas: 29,17%, patikimumas: 100%)
(G) ⇒ (H, E) (dažnumas: 29,17%, patikimumas: 70%)
(G, E) ⇒ (H) (dažnumas: 29,17%, patikimumas: 70%)
(G, H) ⇒ (E) (dažnumas: 29,17%, patikimumas: 100%)
(F) ⇒ (G, H) (dažnumas: 29,17%, patikimumas: 77,78%)
(H) ⇒ (G, F) (dažnumas: 29,17%, patikimumas: 100%)
(H, F) ⇒ (G) (dažnumas: 29,17%, patikimumas: 100%)
(G) ⇒ (H, F) (dažnumas: 29,17%, patikimumas: 70%)
(G, F) ⇒ (H) (dažnumas: 29,17%, patikimumas: 100%)
(G, H) ⇒ (F) (dažnumas: 29,17%, patikimumas: 100%)
(I) ⇒ (G, E) (dažnumas: 25%, patikimumas: 75%)
(I, E) ⇒ (G) (dažnumas: 25%, patikimumas: 100%)
(I, G) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(K, A) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(K, C) ⇒ (A) (dažnumas: 25%, patikimumas: 75%)
(E, A) ⇒ (K) (dažnumas: 25%, patikimumas: 75%)
(K, A) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(K, E) ⇒ (A) (dažnumas: 25%, patikimumas: 75%)
(E, C) ⇒ (K) (dažnumas: 33,33%, patikimumas: 80%)
(K, C) ⇒ (E) (dažnumas: 33,33%, patikimumas: 100%)
(K, E) ⇒ (C) (dažnumas: 33,33%, patikimumas: 100%)
(G, A) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(K, A) ⇒ (G) (dažnumas: 25%, patikimumas: 100%)
(K, G) ⇒ (A) (dažnumas: 25%, patikimumas: 75%)
(G) ⇒ (K, C) (dažnumas: 33,33%, patikimumas: 80%)
(G, C) ⇒ (K) (dažnumas: 33,33%, patikimumas: 100%)
(K, C) ⇒ (G) (dažnumas: 33,33%, patikimumas: 100%)
(K, G) ⇒ (C) (dažnumas: 33,33%, patikimumas: 100%)
(G) ⇒ (K, E) (dažnumas: 33,33%, patikimumas: 80%)
(G, E) ⇒ (K) (dažnumas: 33,33%, patikimumas: 80%)
(K, E) ⇒ (G) (dažnumas: 33,33%, patikimumas: 100%)
(K, G) ⇒ (E) (dažnumas: 33,33%, patikimumas: 100%)
(M, A) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(M, C) ⇒ (A) (dažnumas: 25%, patikimumas: 85,71%)
(M, C) ⇒ (E) (dažnumas: 25%, patikimumas: 85,71%)
(M, E) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(G, C) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)
(M, C) =⇒> (G) (dažnumas: 25%, patikimumas: 85,71%)
(M, G) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(M, E) ⇒ (G) (dažnumas: 25%, patikimumas: 100%)
(M, G) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(K, C) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)

(M, C) ⇒ (K) (dažnumas: 25%, patikimumas: 85,71%)
(K, E) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)
(M, E) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(K, G) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)
(M, G) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(I) ⇒ (M, K) (dažnumas: 25%, patikimumas: 75%)
(K, I) ⇒ (M) (dažnumas: 25%, patikimumas: 85,71%)
(M, I) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(M, K) ⇒ (L) (dažnumas: 29,17%, patikimumas: 70%)
(L) ⇒ (M, K) (dažnumas: 29,17%, patikimumas: 77,78%)
(L, K) ⇒ (M) (dažnumas: 29,17%, patikimumas: 77,78%)
(L, M) ⇒ (K) (dažnumas: 29,17%, patikimumas: 100%)
(P) ⇒ (L, K) (dažnumas: 25%, patikimumas: 75%)
(P, K) ⇒ (L) (dažnumas: 25%, patikimumas: 100%)
(L, P) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(L) ⇒ (S, K) (dažnumas: 29,17%, patikimumas: 77,78%)
(L, K) ⇒ (S) (dažnumas: 29,17%, patikimumas: 77,78%)
(S) ⇒ (L, K) (dažnumas: 29,17%, patikimumas: 77,78%)
(S, K) ⇒ (L) (dažnumas: 29,17%, patikimumas: 100%)
(S, L) ⇒ (K) (dažnumas: 29,17%, patikimumas: 100%)
(R) ⇒ (L, K) (dažnumas: 25%, patikimumas: 75%)
(R, K) ⇒ (L) (dažnumas: 25%, patikimumas: 100%)
(R, L) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(G, M) ⇒ (E, K) (dažnumas: 25%, patikimumas: 100%)
(G, K) ⇒ (E, M) (dažnumas: 25%, patikimumas: 75%)
(G, K, M) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(E, M) ⇒ (G, K) (dažnumas: 25%, patikimumas: 100%)
(E, K) ⇒ (G, M) (dažnumas: 25%, patikimumas: 75%)
(E, K, M) ⇒ (G) (dažnumas: 25%, patikimumas: 100%)
(E, G, M) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(E, G, K) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)
(G) ⇒ (E, F, H) (dažnumas: 29,17%, patikimumas: 70%)
(H) ⇒ (E, F, G) (dažnumas: 29,17%, patikimumas: 100%)
(H, G) ⇒ (E, F) (dažnumas: 29,17%, patikimumas: 100%)
(F) ⇒ (E, H, G) (dažnumas: 29,17%, patikimumas: 77,78%)
(F, G) ⇒ (E, H) (dažnumas: 29,17%, patikimumas: 100%)
(F, H) ⇒ (E, G) (dažnumas: 29,17%, patikimumas: 100%)
(F, H, G) ⇒ (E) (dažnumas: 29,17%, patikimumas: 100%)
(E, G) ⇒ (F, H) (dažnumas: 29,17%, patikimumas: 70%)
(E, H) ⇒ (F, G) (dažnumas: 29,17%, patikimumas: 100%)
(E, H, G) ⇒ (F) (dažnumas: 29,17%, patikimumas: 100%)
(E, F) ⇒ (H, G) (dažnumas: 29,17%, patikimumas: 77,78%)
(E, F, G) ⇒ (H) (dažnumas: 29,17%, patikimumas: 100%)

(E, F, H) ⇒ (G) (dažnumas: 29,17%, patikimumas: 100%)
(G, M) ⇒ (C, K) (dažnumas: 25%, patikimumas: 100%)
(G, K) ⇒ (C, M) (dažnumas: 25%, patikimumas: 75%)
(G, K, M) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(C, M) ⇒ (G, K) (dažnumas: 25%, patikimumas: 85,71%)
(C, K) ⇒ (G, M) (dažnumas: 25%, patikimumas: 75%)
(C, K, M) ⇒ (G) (dažnumas: 25%, patikimumas: 100%)
(C, G) ⇒ (K, M) (dažnumas: 25%, patikimumas: 75%)
(C, G, M) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(C, G, K) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)
(E, M) ⇒ (C, K) (dažnumas: 25%, patikimumas: 100%)
(E, K) ⇒ (C, M) (dažnumas: 25%, patikimumas: 75%)
(E, K, M) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(C, M) ⇒ (E, K) (dažnumas: 25%, patikimumas: 85,71%)
(C, K) ⇒ (E, M) (dažnumas: 25%, patikimumas: 75%)
(C, K, M) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(C, E, M) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(C, E, K) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)
(G, M) ⇒ (C, E) (dažnumas: 25%, patikimumas: 100%)
(E, M) ⇒ (C, G) (dažnumas: 25%, patikimumas: 100%)
(E, G, M) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(C, M) ⇒ (E, G) (dažnumas: 25%, patikimumas: 85,71%)
(C, G) ⇒ (E, M) (dažnumas: 25%, patikimumas: 75%)
(C, G, M) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(C, E, M) ⇒ (G) (dažnumas: 25%, patikimumas: 100%)
(C, E, G) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)
(G) ⇒ (C, E, K) (dažnumas: 33,33%, patikimumas: 80%)
(G, K) ⇒ (C, E) (dažnumas: 33,33%, patikimumas: 100%)
(E, K) ⇒ (C, G) (dažnumas: 33,33%, patikimumas: 100%)
(E, G) ⇒ (C, K) (dažnumas: 33,33%, patikimumas: 80%)
(E, G, K) => (C) (dažnumas: 33,33%, patikimumas: 100%)
(C, K) ⇒ (E, G) (dažnumas: 33,33%, patikimumas: 100%)
(C, G) ⇒ (E, K) (dažnumas: 33,33%, patikimumas: 100%)
(C, G, K) ⇒ (E) (dažnumas: 33,33%, patikimumas: 100%)
(C, E) ⇒ (G, K) (dažnumas: 33,33%, patikimumas: 80%)
(C, E, K) ⇒ (G) (dažnumas: 33,33%, patikimumas: 100%)
(C, E, G) ⇒ (K) (dažnumas: 33,33%, patikimumas: 100%)
(G, K) ⇒ (A, E) (dažnumas: 25%, patikimumas: 75%)
(E, K) ⇒ (A, G) (dažnumas: 25%, patikimumas: 75%)
(E, G, K) ⇒ (A) (dažnumas: 25%, patikimumas: 75%)
(A, K) ⇒ (E, G) (dažnumas: 25%, patikimumas: 100%)
(A, G) ⇒ (E, K) (dažnumas: 25%, patikimumas: 100%)
(A, G, K) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)

(A, E) ⇒ (G, K) (dažnumas: 25%, patikimumas: 75%)
(A, E, K) ⇒ (G) (dažnumas: 25%, patikimumas: 100%)
(A, E, G) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(G, K) ⇒ (A, C) (dažnumas: 25%, patikimumas: 75%)
(C, K) ⇒ (A, G) (dažnumas: 25%, patikimumas: 75%)
(C, G) ⇒ (A, K) (dažnumas: 25%, patikimumas: 75%)
(C, G, K) ⇒ (A) (dažnumas: 25%, patikimumas: 75%)
(A, K) ⇒ (C, G) (dažnumas: 25%, patikimumas: 100%)
(A, G) ⇒ (C, K) (dažnumas: 25%, patikimumas: 100%)
(A, G, K) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(A, C, K) ⇒ (G) (dažnumas: 25%, patikimumas: 100%)
(A, C, G) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(E, K) ⇒ (A, C) (dažnumas: 25%, patikimumas: 75%)
(C, K) ⇒ (A, E) (dažnumas: 25%, patikimumas: 75%)
(C, E, K) ⇒ (A) (dažnumas: 25%, patikimumas: 75%)
(A, K) ⇒ (C, E) (dažnumas: 25%, patikimumas: 100%)
(A, E) ⇒ (C, K) (dažnumas: 25%, patikimumas: 75%)
(A, E, K) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(A, C, K) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(A, C, E) ⇒ (K) (dažnumas: 25%, patikimumas: 75%)
(C, G) ⇒ (A, E) (dažnumas: 25%, patikimumas: 75%)
(C, E, G) ⇒ (A) (dažnumas: 25%, patikimumas: 75%)
(A, G) ⇒ (C, E) (dažnumas: 25%, patikimumas: 100%)
(A, E) ⇒ (C, G) (dažnumas: 25%, patikimumas: 75%)
(A, E, G) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(A, C, G) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(A, C, E) ⇒ (G) (dažnumas: 25%, patikimumas: 75%)
(B, C) ⇒ (D, A) (dažnumas: 25%, patikimumas: 85,71%)
(B, A, C) ⇒ (D) (dažnumas: 25%, patikimumas: 85,71%)
(D) ⇒ (B, A, C) (dažnumas: 25%, patikimumas: 75%)
(D, C) ⇒ (B, A) (dažnumas: 25%, patikimumas: 75%)
(D, A) ⇒ (B, C) (dažnumas: 25%, patikimumas: 85,71%)
(D, A, C) ⇒ (B) (dažnumas: 25%, patikimumas: 85,71%)
(D, B) ⇒ (A, C) (dažnumas: 25%, patikimumas: 100%)
(D, B, C) ⇒ (A) (dažnumas: 25%, patikimumas: 100%)
(D, B, A) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(K, A) ⇒ (E, G, C) (dažnumas: 25%, patikimumas: 100%)
(K, C) ⇒ (E, G, A) (dažnumas: 25%, patikimumas: 75%)
(K, C, A) ⇒ (E, G) (dažnumas: 25%, patikimumas: 100%)
(G, A) ⇒ (E, K, C) (dažnumas: 25%, patikimumas: 100%)
(G, C) ⇒ (E, K, A) (dažnumas: 25%, patikimumas: 75%)
(G, C, A) ⇒ (E, K) (dažnumas: 25%, patikimumas: 100%)
(G, K) ⇒ (E, C, A) (dažnumas: 25%, patikimumas: 75%)

(G, K, A) ⇒ (E, C) (dažnumas: 25%, patikimumas: 100%)
(G, K, C) ⇒ (E, A) (dažnumas: 25%, patikimumas: 75%)
(G, K, C, A) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(E, A) ⇒ (G, K, C) (dažnumas: 25%, patikimumas: 75%)
(E, C, A) ⇒ (G, K) (dažnumas: 25%, patikimumas: 75%)
(E, K) ⇒ (G, C, A) (dažnumas: 25%, patikimumas: 75%)
(E, K, A) ⇒ (G, C) (dažnumas: 25%, patikimumas: 100%)
(E, K, C) ⇒ (G, A) (dažnumas: 25%, patikimumas: 75%)
(E, K, C, A) ⇒ (G) (dažnumas: 25%, patikimumas: 100%)
(E, G, A) ⇒ (K, C) (dažnumas: 25%, patikimumas: 100%)
(E, G, C) ⇒ (K, A) (dažnumas: 25%, patikimumas: 75%)
(E, G, C, A) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(E, G, K) ⇒ (C, A) (dažnumas: 25%, patikimumas: 75%)
(E, G, K, A) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)
(E, G, K, C) ⇒ (A) (dažnumas: 25%, patikimumas: 75%)
(M, C) ⇒ (E, G, K) (dažnumas: 25%, patikimumas: 85,71%)
(K, C) ⇒ (E, G, M) (dažnumas: 25%, patikimumas: 75%)
(K, M, C) ⇒ (E, G) (dažnumas: 25%, patikimumas: 100%)
(G, C) ⇒ (E, K, M) (dažnumas: 25%, patikimumas: 75%)
(G, M) ⇒ (E, K, C) (dažnumas: 25%, patikimumas: 100%)
(G, M, C) ⇒ (E, K) (dažnumas: 25%, patikimumas: 100%)
(G, K) ⇒ (E, M, C) (dažnumas: 25%, patikimumas: 75%)
(G, K, C) ⇒ (E, M) (dažnumas: 25%, patikimumas: 75%)
(G, K, M) ⇒ (E, C) (dažnumas: 25%, patikimumas: 100%)
(G, K, M, C) ⇒ (E) (dažnumas: 25%, patikimumas: 100%)
(E, M) ⇒ (G, K, C) (dažnumas: 25%, patikimumas: 100%)
(E, M, C) ⇒ (G, K) (dažnumas: 25%, patikimumas: 100%)
(E, K) ⇒ (G, M, C) (dažnumas: 25%, patikimumas: 75%)
(E, K, C) ⇒ (G, M) (dažnumas: 25%, patikimumas: 75%)
(E, K, M) ⇒ (G, C) (dažnumas: 25%, patikimumas: 100%)
(E, K, M, C) ⇒ (G) (dažnumas: 25%, patikimumas: 100%)
(E, G, C) ⇒ (K, M) (dažnumas: 25%, patikimumas: 75%)
(E, G, M) ⇒ (K, C) (dažnumas: 25%, patikimumas: 100%)
(E, G, M, C) ⇒ (K) (dažnumas: 25%, patikimumas: 100%)
(E, G, K) ⇒ (M, C) (dažnumas: 25%, patikimumas: 75%)
(E, G, K, C) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)
(E, G, K, M) ⇒ (C) (dažnumas: 25%, patikimumas: 100%)

Susietumo taisyklių skaičius – 70 (*min_supp* = 30 % ir *min_conf* = 30 %).
(M) ⇒ (K) (dažnumas: 41,67%, patikimumas: 90,91%)
(K) ⇒ (M) (dažnumas: 41,67%, patikimumas: 83,33%)
(L) ⇒ (K) (dažnumas: 37,5%, patikimumas: 100%)
(K) ⇒ (L) (dažnumas: 37,5%, patikimumas: 75%)

(K) ⇒ (G) (dažnumas: 33,33%, patikimumas: 66,67%)
(G) ⇒ (K) (dažnumas: 33,33%, patikimumas: 80%)
(K) ⇒ (E) (dažnumas: 33,33%, patikimumas: 66,67%)
(E) ⇒ (K) (dažnumas: 33,33%, patikimumas: 66,67%)
(G) => (E) (dažnumas: 41,67%, patikimumas: 100%)
(E) ⇒ (G) (dažnumas: 41,67%, patikimumas: 83,33%)
(F) ⇒ (E) (dažnumas: 37,5%, patikimumas: 100%)
(E) ⇒ (F) (dažnumas: 37,5%, patikimumas: 75%)
(K) ⇒ (C) (dažnumas: 33,33%, patikimumas: 66,67%)
(C) ⇒ (K) (dažnumas: 33,33%, patikimumas: 61,54%)
(G) ⇒ (C) (dažnumas: 33,33%, patikimumas: 80%)
(C) ⇒ (G) (dažnumas: 33,33%, patikimumas: 61,54%)
(E) ⇒ (C) (dažnumas: 41,67%, patikimumas: 83,33%)
(C) ⇒ (E) (dažnumas: 41,67%, patikimumas: 76,92%)
(D) ⇒ (C) (dažnumas: 33,33%, patikimumas: 100%)
(C) ⇒ (D) (dažnumas: 33,33%, patikimumas: 61,54%)
(E) => (A) (dažnumas: 33,33%, patikimumas: 66,67%)
(A) ⇒ (E) (dažnumas: 33,33%, patikimumas: 57,14%)
(C) ⇒ (A) (dažnumas: 45,83%, patikimumas: 84,62%)
(A) ⇒ (C) (dažnumas: 45,83%, patikimumas: 78,57%)
(B) => (A) (dažnumas: 41,67%, patikimumas: 100%)
(A) ⇒ (B) (dažnumas: 41,67%, patikimumas: 71,43%)
(A) ⇒ (E, C) (dažnumas: 33,33%, patikimumas: 57,14%)
(C) ⇒ (E, A) (dažnumas: 33,33%, patikimumas: 61,54%)
(C, A) ⇒ (E) (dažnumas: 33,33%, patikimumas: 72,73%)
(E) ⇒ (C, A) (dažnumas: 33,33%, patikimumas: 66,67%)
(E, A) ⇒ (C) (dažnumas: 33,33%, patikimumas: 100%)
(E, C) ⇒ (A) (dažnumas: 33,33%, patikimumas: 80%)
(C) ⇒ (G, E) (dažnumas: 33,33%, patikimumas: 61,54%)
(E) ⇒ (G, C) (dažnumas: 33,33%, patikimumas: 66,67%)
(E, C) ⇒ (G) (dažnumas: 33,33%, patikimumas: 80%)
(G) ⇒ (E, C) (dažnumas: 33,33%, patikimumas: 80%)
(G, C) ⇒ (E) (dažnumas: 33,33%, patikimumas: 100%)
(G, E) ⇒ (C) (dažnumas: 33,33%, patikimumas: 80%)
(C) ⇒ (K, E) (dažnumas: 33,33%, patikimumas: 61,54%)
(E) ⇒ (K, C) (dažnumas: 33,33%, patikimumas: 66,67%)
(E, C) ⇒ (K) (dažnumas: 33,33%, patikimumas: 80%)
(K) ⇒ (E, C) (dažnumas: 33,33%, patikimumas: 66,67%)
(K, C) ⇒ (E) (dažnumas: 33,33%, patikimumas: 100%)
(K, E) ⇒ (C) (dažnumas: 33,33%, patikimumas: 100%)
(C) ⇒ (K, G) (dažnumas: 33,33%, patikimumas: 61,54%)
(G) ⇒ (K, C) (dažnumas: 33,33%, patikimumas: 80%)
(G, C) => (K) (dažnumas: 33,33%, patikimumas: 100%)

(K) ⇒ (G, C) (dažnumas: 33,33%, patikimumas: 66,67%)
(K, C) ⇒ (G) (dažnumas: 33,33%, patikimumas: 100%)
(K, G) ⇒ (C) (dažnumas: 33,33%, patikimumas: 100%)
(E) ⇒ (K, G) (dažnumas: 33,33%, patikimumas: 66,67%)
(G) ⇒ (K, E) (dažnumas: 33,33%, patikimumas: 80%)
(G, E) ⇒ (K) (dažnumas: 33,33%, patikimumas: 80%)
(K) ⇒ (G, E) (dažnumas: 33,33%, patikimumas: 66,67%)
(K, E) ⇒ (G) (dažnumas: 33,33%, patikimumas: 100%)
(K, G) ⇒ (E) (dažnumas: 33,33%, patikimumas: 100%)
(K) ⇒ (E, C, G) (dažnumas: 33,33%, patikimumas: 66,67%)
(G) ⇒ (E, C, K) (dažnumas: 33,33%, patikimumas: 80%)
(G, K) ⇒ (E, C) (dažnumas: 33,33%, patikimumas: 100%)
(C) ⇒ (E, G, K) (dažnumas: 33,33%, patikimumas: 61,54%)
(C, K) ⇒ (E, G) (dažnumas: 33,33%, patikimumas: 100%)
(C, G) ⇒ (E, K) (dažnumas: 33,33%, patikimumas: 100%)
(C, G, K) ⇒ (E) (dažnumas: 33,33%, patikimumas: 100%)
(E) ⇒ (C, G, K) (dažnumas: 33,33%, patikimumas: 66,67%)
(E, K) ⇒ (C, G) (dažnumas: 33,33%, patikimumas: 100%)
(E, G) ⇒ (C, K) (dažnumas: 33,33%, patikimumas: 80%)
(E, G, K) ⇒ (C) (dažnumas: 33,33%, patikimumas: 100%)
(E, C) ⇒ (G, K) (dažnumas: 33,33%, patikimumas: 80%)
(E, C, K) ⇒ (G) (dažnumas: 33,33%, patikimumas: 100%)
(E, C, G) ⇒ (K) (dažnumas: 33,33%, patikimumas: 100%)

Priedas 4. Transakcijų duomenų bazėje išskirtos apibendrintos susietumo taisyklės

Šiame priede pateiktos apibendrintos susietumo taisyklės, kai minimalus dažnumas *min_supp* = 25 % ir minimalus patikimumas *min_conf* = 70 % bei *min_supp* = 30 % ir *min_conf* = 30 %.

Apibendrintų susietumo taisyklių skaičius – 43 (*min_supp* = 25 % ir *min_conf* = 70). Prekių pavadinimai žymimi A, B, ... , prekių grupių pavadinimai – 1, 2.

(R) ⇒ (S) (dažnumas: 25%, patikimumas: 75%)
(I) ⇒ (M) (dažnumas: 25%, patikimumas: 75%)
(I) ⇒ (2) (dažnumas: 25%, patikimumas: 75%)
(H) ⇒ (2) (dažnumas: 29,17%, patikimumas: 100%)
(D) ⇒ (2) (dažnumas: 33,33%, patikimumas: 100%)
(D) ⇒ (B) (dažnumas: 25%, patikimumas: 75%)
(B) ⇒ (2) (dažnumas: 29,17%, patikimumas: 70%)
(S) ⇒ (1) (dažnumas: 29,17%, patikimumas: 77,78%)
(U) ⇒ (1) (dažnumas: 25%, patikimumas: 85,71%)
(T) ⇒ (1) (dažnumas: 33,33%, patikimumas: 80%)
(R) ⇒ (1) (dažnumas: 25%, patikimumas: 75%)
(P) ⇒ (1) (dažnumas: 33,33%, patikimumas: 100%)
(M) ⇒ (1) (dažnumas: 45,83%, patikimumas: 100%)
(I) ⇒ (1) (dažnumas: 29,17%, patikimumas: 87,5%)
(D) ⇒ (1) (dažnumas: 33,33%, patikimumas: 100%)
(2) ⇒ (1) (dažnumas: 54,17%, patikimumas: 86,67%)
(B) ⇒ (1) (dažnumas: 41,67%, patikimumas: 100%)
(D) ⇒ (B, 1) (dažnumas: 25%, patikimumas: 75%)
(D, 1) ⇒ (B) (dažnumas: 25%, patikimumas: 75%)
(D, B) ⇒ (1) (dažnumas: 25%, patikimumas: 100%)
(B) ⇒ (2, 1) (dažnumas: 29,17%, patikimumas: 70%)
(B, 1) ⇒ (2) (dažnumas: 29,17%, patikimumas: 70%)
(2, B) ⇒ (1) (dažnumas: 29,17%, patikimumas: 100%)
(D) ⇒ (2, 1) (dažnumas: 33,33%, patikimumas: 100%)
(D, 1) ⇒ (2) (dažnumas: 33,33%, patikimumas: 100%)
(2, D) ⇒ (1) (dažnumas: 33,33%, patikimumas: 100%)
(D) ⇒ (2, B) (dažnumas: 25%, patikimumas: 75%)
(D, B) ⇒ (2) (dažnumas: 25%, patikimumas: 100%)
(2, B) ⇒ (D) (dažnumas: 25%, patikimumas: 85,71%)
(2, D) ⇒ (B) (dažnumas: 25%, patikimumas: 75%)
(M, 2) ⇒ (1) (dažnumas: 29,17%, patikimumas: 100%)
(I) ⇒ (M, 1) (dažnumas: 25%, patikimumas: 75%)
(I, 1) ⇒ (M) (dažnumas: 25%, patikimumas: 85,71%)

(I, M) ⇒ (1) (dažnumas: 25%, patikimumas: 100%)
(B, 2) ⇒ (D, 1) (dažnumas: 25%, patikimumas: 85,71%)
(B, 1, 2) ⇒ (D) (dažnumas: 25%, patikimumas: 85,71%)
(D) ⇒ (B, 1, 2) (dažnumas: 25%, patikimumas: 75%)
(D, 2) ⇒ (B, 1) (dažnumas: 25%, patikimumas: 75%)
(D, 1) ⇒ (B, 2) (dažnumas: 25%, patikimumas: 75%)
(D, 1, 2) ⇒ (B) (dažnumas: 25%, patikimumas: 75%)
(D, B) ⇒ (1, 2) (dažnumas: 25%, patikimumas: 100%)
(D, B, 2) ⇒ (1) (dažnumas: 25%, patikimumas: 100%)
(D, B, 1) ⇒ (2) (dažnumas: 25%, patikimumas: 100%)

Apibendrintų susietumo taisyklių skaičius – 20 (*min_supp* = 30 % ir *min_conf* = 30 %).
(D) ⇒ (2) (dažnumas: 33,33%, patikimumas: 100%)
(2) ⇒ (D) (dažnumas: 33,33%, patikimumas: 53,33%)
(T) ⇒ (1) (dažnumas: 33,33%, patikimumas: 80%)
(1) ⇒ (T) (dažnumas: 33,33%, patikimumas: 40%)
(P) ⇒ (1) (dažnumas: 33,33%, patikimumas: 100%)
(1) ⇒ (P) (dažnumas: 33,33%, patikimumas: 40%)
(M) ⇒ (1) (dažnumas: 45,83%, patikimumas: 100%)
(1) ⇒ (M) (dažnumas: 45,83%, patikimumas: 55%)
(D) ⇒ (1) (dažnumas: 33,33%, patikimumas: 100%)
(1) ⇒ (D) (dažnumas: 33,33%, patikimumas: 40%)
(2) ⇒ (1) (dažnumas: 54,17%, patikimumas: 86,67%)
(1) ⇒ (2) (dažnumas: 54,17%, patikimumas: 65%)
(B) ⇒ (1) (dažnumas: 41,67%, patikimumas: 100%)
(1) ⇒ (B) (dažnumas: 41,67%, patikimumas: 50%)
(1) ⇒ (2, D) (dažnumas: 33,33%, patikimumas: 40%)
(D) ⇒ (2, 1) (dažnumas: 33,33%, patikimumas: 100%)
(D, 1) ⇒ (2) (dažnumas: 33,33%, patikimumas: 100%)
(2) ⇒ (D, 1) (dažnumas: 33,33%, patikimumas: 53,33%)
(2, 1) ⇒ (D) (dažnumas: 33,33%, patikimumas: 61,54%)
(2, D) ⇒ (1) (dažnumas: 33,33%, patikimumas: 100%)