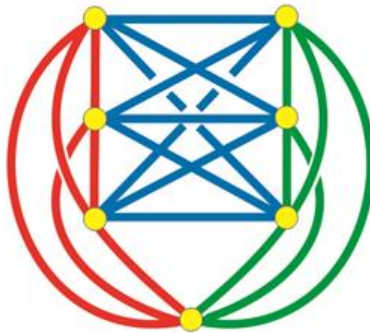


Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Інститут кібернетики імені В.М. Глушкова НАН України
Національний університет «Запорізька політехніка»
Запорізький національний університет
Інститут модернізації та змісту освіти
Міжнародний Європейський університет
Громадська організація «Системні дослідження»

Матеріали XXVI Міжнародного науково-практичного семінару

**КОМБІНАТОРНІ КОНФІГУРАЦІЇ
ТА ЇХНІ ЗАСТОСУВАННЯ**
присвяченого пам'яті професора Донця Г.П.

13-15 червня 2024 року



Кропивницький - Запоріжжя - Київ
2024

Редакційна колегія:

Леонід ГУЛЯНИЦЬКИЙ, член-кор. НАН України, д.т.н.;
Анна БАКУРОВА, д.е.н., професор;
Ігор КОЗІН, д.ф.-м.н., професор;
Марина СЕМЕНЮТА, к.ф.-м.н., доцент;
Василь ШЕНДЕРОВСЬКИЙ, д.ф.-м.н., професор.

Комбінаторні конфігурації та їхні застосування: Матеріали XXVI Міжнародного науково-практичного семінару, присвяченого пам'яті професора Донця Г.П., (Кропивницький–Запоріжжя–Київ, 13-15 червня 2024 року) / за ред. Л.Ф. Гуляницького – Кропивницький: ПП «Ексклюзив-Систем», 2024. – 228 с.

ISBN 978-617-7942-27-5

У збірнику подано матеріали XXVI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їхні застосування» присвяченого пам'яті професора Донця Г.П., який відбувся 13-15 червня 2024 року. Коло обговорених проблем: комбінаторні розміщення, графи і задачі оптимізації на них, автомати, скінченні алгебри, математична логіка і теорія множин, теорія ігор і теорія прийняття рішень та їхні застосування в різних галузях; реалізація математичних моделей на дискретних та неперервних множинах (автоматизовані системи керування, системи автоматизованого проектування, системи підтримки прийняття рішень); програмне забезпечення синтезу та аналізу комбінаторних конфігурацій; розпізнавання образів, нейромережі та методи машинного навчання; фрагментарні структури, алгоритми та метаевристики; топологія та історія розвитку української науки; кодування, інформаційна безпека та захист інформації.

Для науковців, викладачів вищих навчальних закладів, а також аспірантів, студентів і фахівців із застосування комбінаторних конфігурацій.

Програмний комітет конференції

Леонід ГУЛЯНИЦЬКИЙ	член-кореспондент НАН України, д.т.н., старший науковий співробітник, завідувач відділом методів комбінаторної оптимізації та інтелектуальних інформаційних технологій Інституту кібернетики імені В.М. Глушкова НАН України, голова програмного комітету;
Марина СЕМЕНЮТА	к. ф.-м. н., доцент, доцент кафедри вищої математики та фізики Центральноукраїнського національного технічного університету, співголова програмного комітету;
Анна БАКУРОВА	д.е.н., професор, професор кафедри «Системного аналізу та обчислювальної математики» НУ «Запорізька політехніка»; співголова програмного комітету;
Каміль АЙДА-ЗАДЕ	д. ф.-м. н., професор, завідувач лабораторії Інституту систем управління, Міністерство освіти і науки Азербайджанської Республіки, член-кореспондент Національної академії наук Азербайджану;
Олександр КОНОВАЛОВ	доктор наук, лектор, науковий співробітник Інституту стійкості програмного забезпечення, Школа комп'ютерних наук, Університет Сент-Ендрюса, Об'єднане Королівство;
Пітер КАМЕРОН	професор, Школа математики та статистики, Університет Сент-Ендрюса, Об'єднане Королівство;
Озгюр АКГЮН	доктор, лектор, Школа комп'ютерних наук, Університет Сент-Ендрюс, Об'єднане Королівство;
Ілля САФРО	доктор філософії, завідувач кафедри аспірантури та досліджень, доцент кафедри комп'ютерних та

Улугбек НАРЗУЛЛАЕВ	інформаційних наук, Університет Делавер, США; кандидат фізико-математичних наук, доцент, декан факультету телекомунікаційних технологій і професійної освіти Самаркандського філіалу Ташкентського університету інформаційних технологій;
Ігор КОЗІН	д.ф.-м.н., професор, професор кафедри економічної кібернетики Запорізького національного університету;
Сергій КРИВИЙ	д.ф.-м.н., професор, професор кафедри інтелектуальних програмних систем Київського національного університету імені Т. Шевченка;
Віктор КУЗЬМЕНКО	к. ф.-м. ., старший науковий співробітник, завідувач відділом економічної кібернетики Інституту кібернетики імені В.М. Глушкова НАН України
Микола МАЛІЯР	д.т.н., професор, декан факультету математики та цифрових технологій Ужгородського національного університету;
Слизова МЕЛІШКО	д.т.н., професор, доцент кафедри кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету;
Тетяна РОМАНОВА	д.т.н., професор, провідний науковий співробітник відділу нелінійної механіки та математичного моделювання Інституту проблем машинобудування ім. А.М. Підгорного НАН України;
Олексій СМІРНОВ	д.т.н., професор, завідувач кафедри кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету;
Петро СТЕЦЮК	член кореспондент НАН України, д.ф.- м.н., старший науковий співробітник,

- Надія ТИМОФІЄВА завідувач відділом методів негладкої оптимізації Інституту кібернетики імені В.М. Глушкова НАН України; д.т.н., старший науковий співробітник відділу комплексних досліджень інформаційних технологій Міжнародного науково-навчального центру інформаційних технологій та систем НАН та МОН України;
- Валентина ТУРЧИНА к.ф.-м.н., доцент, завідувач кафедри обчислювальної математики та математичної кібернетики Дніпровського національного університету імені О.Гончара;
- Василь ШЕНДЕРОВСЬКИЙ д.ф.-м.н., професор, віце-президент Українського фізичного товариства;
- Тетяна ШМЕЛЬОВА д.т.н., професор, професор кафедри аеронавігаційних систем Національного авіаційного університету;
- Зоя ШЕРМАН к. ф.-м. н., доцент кафедри фундаментальних та медико-профілактичних дисциплін Міжнародного європейського університету;
- Сергій ЯКОВЛЕВ член-кореспондент НАН України, д. ф.-м. н., професор, професор-дослідник Інституту математики Лодзьського політехнічного університету

Організаційний комітет:

- Володимир КРОПІВНИЙ професор, к.т.н., ректор Центральноукраїнського національного технічного університету, голова організаційного комітету;
- Сергій ЯКИМЕНКО к.ф.-м.н., доцент, завідувач кафедри вищої математики та фізики Центральноукраїнського національного технічного університету, заступник голови організаційного комітету;
- Ірина ФЛІМОНІХІНА к.ф.-м.н., доцент, доцент кафедри вищої математики та фізики

- Людмила ДЖУМА Центральноукраїнського національного технічного університету, секретар організаційного комітету;
к.т.н., доцент, доцент кафедри фізико-математичних дисциплін та застосування інформаційних технологій в авіаційних системах Льотної академії НАУ;
- Галина КОЗИНА к.ф.-м.н., доцент, доцент кафедри захисту інформації НУ «Запорізька політехніка»;
- Олександр НЕСТЕРЕНКО д. т. н., професор, завідувач кафедри інформаційних технологій Міжнародного Європейського університету;
- Володимир ПЕТРЕНЮК к.ф.-м.н., доцент, доцент кафедри кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету;
- Дмитро ПЕТРЕНЮК к.ф.-м.н., молодший науковий співробітник відділу економічної кібернетики Інституту кібернетики імені В.М. Глушкова НАН України;
- Антон РЯБЕНКО, к.ф.-м.н., доцент, доцент кафедри «Системний аналіз та обчислювальна математика НУ «Запорізька політехніка»;
- Еліна ТЕРЕЩЕНКО к.ф.-м.н., доцент, в.о. завідувача кафедри «Системний аналіз та обчислювальна математика НУ «Запорізька політехніка»;
- Дмитро ШИРОКОРАД к.ф.-м.н., доцент, ст викладач кафедри «Системний аналіз та обчислювальна математика НУ «Запорізька політехніка»;
- Громадська організація «Системні дослідження

ЗМІСТ

Вітальні слова учасникам семінару	12
Доповіді	
¹Бакурова А.В., ¹Гринченко П.В., ¹Грицкевич А.А., ²Зарицький О.В., ¹Терещенко Е.В., ¹Ширококоряд Д.В.	
<i>НУ «Запорізька політехніка», Національний авіаційний університет</i>	
ОЦІНЮВАННЯ ВРАЗЛИВОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ОСНОВІ FUZZY-МАТЕМАТИКИ	20
Peter Cameron	
<i>School of Mathematics and Statistics, University of St Andrews, UK</i>	
BEAUTIFUL GRAPHS FROM SIMPLE GROUPS	25
Вінтенко Б.Ю., Смірнова Т.В., Смірнов О.А.	
<i>Центральноукраїнський національний технічний університет</i>	
ДОСЛІДЖЕННЯ ВИМОГ ДО КІБЕРБЕЗПЕКИ КОМП'ЮТЕРНИХ СИСТЕМ УПРАВЛІННЯ АЕС	28
Олександр Глухов	
<i>Національний авіаційний університет</i>	
ПРО КВАЗИВИПАДКОВІ ГРАФИ НА ОСНОВІ 3-РЕБЕРНО ЗВ'ЯЗНИХ ГРАФІВ	31
Василь Горбачук, Дмитро Ніколенко, Віктор Годлюк, Дмитро Рибачок	
<i>Інститут кібернетики імені В.М. Глушкова НАН України</i>	
ПРО ЦІЛЬОВІ РЕКОНФІГУРАЦІЇ МЕРЕЖ	34
Леонід Гуляницький	
<i>Інститут кібернетики імені В.М. Глушкова НАН України</i>	
ГЕТЕРОГЕННІ ТА ІЄРАРХІЧНІ КОМБІНАТОРНІ КОНФІГУРАЦІЇ	44
Artem Gurin	
<i>National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"</i>	
AN APPLICATION OF THE PROBLEM OF A MATHEMATICAL SAFE IN THE FIELDS OF REDUNDANCIES TO THE CONSTRUCTION OF A SYMMETRIC ENCRYPTION SYSTEM BASED ON THE VIGENERE METHOD	50

Адам Доровці, Микола Маляр, Маріанна Шаркаді <i>ДВНЗ «Ужгородський національний університет»</i> АЛГОРИТМ НЕЧІТКОГО ВИВОДУ ДЛЯ ВИЗНАЧЕННЯ РІВНЯ ЕКОНОМІЧНОГО ДОБРОБУТУ НАСЕЛЕННЯ	56
¹Задорожний Б.О., ^{2,3}Романова Т.Є., ^{1,4}Стецюк П.І. <i>¹Ужгородський національний університет</i> <i>²Інститут проблем машинобудування імені А.М. Підгорного</i> <i>НАН України</i> <i>³Leeds University Business School, University of Leeds, UK</i> <i>⁴Інститут кібернетики імені В.М. Глушкова НАН України</i>	
ПОКРАЩЕННЯ ЕВРИСТИЧНОГО АЛГОРИТМУ ПАКУВАННЯ НЕРІВНИХ КРУГІВ ІЗ ЗАСТОСУВАННЯМ <i>r</i> - АЛГОРИТМУ ШОРА	59
Калашник Г.А., Калашник-Рибалко М.А. <i>Льотна академія Національного авіаційного університету</i> МЕТОД ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОЇ СТІЙКОСТІ ІНТЕГРОВАНОГО КОМПЛЕКСУ БОРТОВОГО ОБЛАДНАННЯ В ДЕСТАБІЛІЗУЮЧИХ УМОВАХ	66
Ігор Козін, Олег Сардак, Артур Алексєєв <i>Запорізький національний університет</i> <i>Навчально-науковий інститут ВВП "Дніпровський</i> <i>гуманітарний університет"</i>	
ПОБУДОВА ВИПАДКОВИХ ГРАФІВ НА БАЗІ МОДЕЛІ ЕРДЕША-РЕНЬЇ	72
Вячеслав Корольов, Максим Огурцов, Олександр Ходзінський <i>Інститут кібернетики імені В. М. Глушкова НАН України</i>	
АЛГОРИТМ НАВІГАЦІЇ ГРУПИ СІЛЬСЬКОГОСПОДАРСЬКИХ БПЛА ЗА ВІДСУТНОСТІ GPS	77
Сергій Кривий <i>Київський національний університет імені Тараса Шевченка</i> МАТЕМАТИЧНИЙ СЕЙФ, ГРАЦІОЗНІ ДЕРЕВА ТА ЗАХИСТ ІНФОРМАЦІЇ	84
Сергій Курапов, Максим Давидовський <i>Запорізький національний університет, КЗ «Запорізький</i> <i>обласний інститут післядипломної педагогічної освіти» ЗОР</i>	
ІНТЕГРАЛЬНИЙ ІНВАРІАНТ ГРАФУ	90
Литвиненко А.В. <i>НУ «Запорізька політехніка»</i> КОМПАРАТИВНИЙ АНАЛІЗ ГРАФІЧНИХ БІБЛІОТЕК ДЛЯ ВІЗУАЛІЗАЦІЇ ГРАФІВ	99

Маліснко О.О., Коваленко Є.О. <i>Дніпровський національний університет імені Олеся Гончара</i> ДОСЛІДЖЕННЯ ВПЛИВУ ПЕРЕРИВАНЬ НА ВИНИКНЕННЯ АНОМАЛІЙ У ЗАДАЧАХ ПАРАЛЕЛЬНОГО УПОРЯДКУВАННЯ	103
Мелешко Є. <i>Центральноукраїнський національний технічний університет,</i> МЕТОД ВИЯВЛЕННЯ ТА НЕЙТРАЛІЗАЦІЇ МЕРЕЖІ БОТІВ У РЕКОМЕНДАЦІЙНІЙ СИСТЕМІ	108
Мелешко Є.В., Варченко І.В <i>Центральноукраїнський національний технічний університет,</i> <i>м. Кропивницький</i> ДОСЛІДЖЕННЯ МЕТОДІВ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У КІБЕРБЕЗПЕЦІ	115
Володимир Петренюк, Дмитро Петренюк <i>Центральноукраїнський національний технічний університет,</i> <i>Інститут кібернетики імені В.М. Глушкова НАН України</i> КОРИСНІ ВЛАСТИВОСТІ ПЛОЩИННИХ ПІДГРАФІВ ГРАФІВ-ОБСТРУКЦІЙ ОБМЕЖЕНОГО НЕОРІЄНТОВАНОГО РОДУ	120
Дмитро Петренюк <i>Інститут кібернетики імені В.М. Глушкова НАН України</i> Т-ФАКТОРИЗАЦІЯ ПОВНОГО ГРАФА ТА ГРАЦІОЗНІСТЬ ДЕРЕВ	12
Юрій Прокопчук <i>Institute of Technical Mechanics of the NAS and SSA,</i> <i>Ukrainian State University of Science and Technologies</i> "КОГНІТИВНІ ЧИСЛА" ЯК ПРИКЛАД "ЖИВОЇ СТРУКТУРИ"	134
Антон Рябенко, Еліна Терещенко <i>Національний університет "Запорізька політехніка" Інститут</i> <i>кібернетики імені В. М. Глушкова НАН України</i> ТЕОРЕТИКО-ГРАФОВА ПОСТАНОВКА ЗАДАЧІ ФОРМУВАННЯ ЦІЛЬОВИХ ГРУП НА ОСНОВІ ФАКТОРА ЗМАГАННЯ	140
Ірина Рясна, Олександр Сенько <i>Інститут кібернетики імені В. М. Глушкова НАН України</i> НЕЧІТКА МОДЕЛЬ КЛАСТЕРІВ НА ОСНОВІ МУЛЬТИНЕЧІТКИХ МНОЖИН	144

Марина Семенюта, Сергій Якименко <i>Центральноукраїнський національний технічний університет</i>	
ДИСТАНЦІЙНІ МАГІЧНІ РОЗМІТКИ І ГРАФИ КЕЛІ	148
Марина Семенюта, Зоя Шерман, Олександр Нестеренко <i>Центральноукраїнський національний технічний університет, Міжнародний європейський університет</i>	
ПРО ВЕРХНІ І НИЖНІ ОЦІНКИ ХРОМАТИЧНОГО ЧИСЛА ДЛЯ ПЛАНАРНИХ І НЕПЛАНАРНИХ ГРАФІВ	152
Олексій Сергєєв <i>Національний технічний університет «Дніпровська політехніка</i>	
АСПЕКТИ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДВОЕТАПНОЇ ЗАДАЧІ АКТИВАЦІЇ-РОЗМІЩЕННЯ ДЛЯ ПОКРАЩЕННЯ МЕДИЧНОЇ ЛОГІСТИКИ	158
Oleksandr Smishko <i>National University «Zaporizhzhia Polytechnic</i>	
IMPACT OF THE CHOICE OF THE TECHNOLOGY STACK ON THE INNOVATION AND COMPETITIVENESS OF COMPANIES: SAP CLOUD VS SAP ON-PREMISE	165
Михайло Соломко <i>Національний університет водного господарства та природокористування</i>	
НЕСТАНДАРТНЕ ПОЧИНАННЯ СПРОЩЕННЯ БУЛЕВИХ ФУНКЦІЙ АНАЛІТИЧНИМ МЕТОДОМ	170
¹ Стецюк П.І., ² Романова Т.Є., ³ Тиводар С.Р. <i>¹ Інститут кібернетики імені В.М. Глушкова НАН України ² Інститут проблем машинобудування імені А.М. Підгорного НАН України ³ Ужгородський національний університет</i>	
ВИКОРИСТАННЯ СОЛВЕРА BARON ДЛЯ РОЗВ'ЯЗАННЯ КВАДРАТИЧНОЇ ЗАДАЧІ ОПТИМАЛЬНОЇ УПАКОВКИ НЕРІВНИХ КРУГІВ	179
Тимофієва Н. К. <i>Міжнародний науково-навчальний центр інформаційних технологій та систем НАН та МОН України</i>	
МОДЕЛЮВАННЯ ДИНАМІКИ МИСЛЕННЯ ЛЮДИНИ З ВИКОРИСТАННЯМ КОМБІНАТОРИКИ	188

¹Shmelova T., ²Sikirda Yu., ³Yatsko M.

^{1,3}National Aviation University

²Flight Academy of the National Aviation University

METHOD OF COLLABORATIVE DECISION-MAKING
MODELING BY OPERATORS OF AIR NAVIGATION SYSTEM
CONSIDERING THE FACTORS' PRIORITY 193

Ірина Філімоніхін

Центральноукраїнський національний технічний університет

ЗАСТОСУВАННЯ ГРАФІВ ДО РОЗВ'ЯЗАННЯ СЛАР 204

Артем Ховрат, Володимир Кобзєв

Харківський національний університет радіоелектроніки

ВИЯВЛЕННЯ СФАЛЬСИФІКОВАНОЇ ІНФОРМАЦІЇ ЗА
ДОПОМОГОЮ НАЇВНОГО БАССОВОГО КЛАСИФІКАТОРА 208

Sergiy Yakovlev, Oksana Pichugina

*Institute of Mathematics, Lodz University of Technology, Lodz,
Poland*

National Aerospace University "Kharkiv aviation institute", Kharkiv

COMBINATORIAL POINT CONFIGURATIONS AND THEIR
PROPERTIES 216

Олена Полевецька, Василь Шендеровський

Інститут фізики НАНУ

МИРОН ЗАРИЦЬКИЙ – ФУНДАТОР МАТЕМАТИЧНОЇ
НАУКИ НА ЗАХІДНОУКРАЇНСЬКИХ ЗЕМЛЯХ 219

Резолюція XXVI Міжнародного науково-практичного

семінару, присвяченого пам'яті професора Донця Г.П. 224

Доповіді закордонних учасників семінару

Özgür Akgün

School of Computer Science, University of St Andrews, UK

CONSTRAINT PROGRAMMING WITH CONJURE INSIDE
JUPYTER NOTEBOOKS

Peter Cameron

School of Mathematics and Statistics, University of St Andrews, UK

BEAUTIFUL GRAPHS FROM SIMPLE GROUPS

Olexandr KONOVALOV

School of Computer Science, University of St Andrews, UK

EXPLORING GRAPHS WITH GAP

можно подивитися за посиланням:

<https://www.youtube.com/watch?v=RZ9zcLf3Hvg>

ВІТАЛЬНІ СЛОВА УЧАСНИКАМ СЕМІНАРУ









*Привітання від Центральноукраїнського національного
технічного університету*

Шановні колеги!!!

Дозвольте привітати Вас від імені всього науково-педагогічного складу Центральноукраїнського національного технічного університету з початком роботи **XXVI Міжнародного науково-практичного семінару "КОМБІНАТОРНІ КОНФІГУРАЦІЇ ТА ЇХНІ ЗАСТОСУВАННЯ"**, присвяченого пам'яті видатного вченого доктора ф-м.н., професора Донця Георгія Панасовича!

Разом з Анатолієм Яковичем Петренюком, Георгій Панасович є засновниками семінару. Перший семінар відбувся в Держаній льотній академії України, якщо я не помиляюся у 2006 році. Перші семінари проводилися двічі на рік.

У 2009 році семінар продовжив роботу в Кіровоградському національному технічному університеті. Потім на деякий час він повернувся в льотну академію і в цьому році знов розпочав роботу на базі Центральноукраїнського національного технічного університету. Його життя продовжується завдяки активної участі колег кафедри системного аналізу та обчислювальної математики Національного університету «Запорізька політехніка», вони підтримали семінар під час пандемії і продовжили співпрацювати далі під час війни. Зараз семінар проводиться по черзі в Кропивницькому і Запоріжжі але разом і завдяки підтримки Інституту кібернетики імені В.М. Глушкова НАН України.

Георгій Панасович був не тільки одним з організаторів семінару, його активним учасником, але і надихав і підтримував інших колег.

Бажаю натхнення і плідної дискусії, творчих успіхів, здоров'я, миру!

Дякую, за увагу.

Марина Семенюта, доцент кафедри вищої математики та фізики Центральноукраїнського національного технічного університету

Привітання від Запорізького національного університету

Шановні колеги!

Від імені колективу Запорізького національного університету щиро вітаємо учасників XXVI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їхні застосування», присвяченого пам'яті видатного українського науковця, патріота та непересічної людини – доктора ф-м.н., професора Донця Георгія Панасовича!

Усе життя Георгія Панасовича було тісно пов'язаним із містом Запоріжжя та запорізькою землею. У м. Запоріжжя пройшли його шкільні роки, а потім він з радістю та бажанням повертався сюди до рідних, а також на конференції, семінари та інші заходи, вкладаючи все нові і нові внески в збагачення науки, що розвивалася на математичному та економічному факультетах Запорізького національного університету. Не можна не відзначити його участь у підготовці нових кваліфікованих кадрів – спочатку як вчений секретар спеціалізованої вченої ради в Інституті кібернетики НАНУ, а потім – як офіційний опонент кандидатських та докторських дисертацій, як багаторічний член спеціалізованої вченої ради із захисту дисертацій у Запорізькому національному університеті.

Багатий науковий досвід Георгія Панасовича та плідні спільні з доктором фіз.-мат. наук, професором Віталієм Перепелицею наукові пошуки в галузі теорії графів та економічної кібернетики сприяли функціонуванню наукової школи «Дискретна математика та її застосування в економіко-математичному моделюванні та інформаційних технологіях», що заснована проф. Перепелицею В.О. та розвивається в Запорізькому національному університеті його учнями.

Вітаючи учасників семінару, одним із засновником та натхненником якого був Георгій Донець, незважаючи на важку ситуацію, яку ми всі переживаємо, бажаємо отримати корисні для кожного з Вас нові знання в галузі комбінаторної оптимізації та її застосувань, надихнутися новими цікавими ідеями, отримати радість дискусій та обміну думками з колегами!

Нехай нові отримані знання сприятимуть розвитку нашої науки та йдуть на користь українській державі!

Наталія МАКСИШКО, доктор економічних наук, професор, завідувач кафедри економічної кібернетики Запорізького національного університету,

Ігор КОЗІН, доктор фізико-математичних наук, професор, професор кафедри економічної кібернетики Запорізького національного університету.

*Привітання від Інституту кібернетики імені В.М.Глушкова
НАН України*

Шановні колеги, учасники семінару!

Проведення Двадцять шостого семінару – це важлива подія для нашої наукової спільноти, яка збирає під своїм дахом провідних вчених, дослідників, викладачів та студентів, об'єднаних спільною пристрастю до математики.

Також маю честь і задоволення передати вітання від директора Інституту кібернетики ім.В.М.Глушкова НАН України академіка І.В.Сергієнка та академіка-секретарі Відділення інформатики НАН України академіка О.М.Хіміча.

Наш семінар присвячено пам'яті проф. Георгія Опанасовича Донця – відомого українського математика, який вніс суттєвий вклад у розвиток комбінаторного аналізу, перш за все, теорії графів. Його участь у попередніх наших семінарах сприяла розвитку теорії комбінаторики та суміжних областей. Отримані ним результати знаходять розвиток і продовження у працях учнів і колег.

Семінар вже давно став платформою для обміну новими ідеями, досягненнями та відкриттями в різних галузях математики. Завдяки нашим зустрічам створюються нові наукові колаборації, народжуються проекти та відкриваються нові горизонти в дослідженнях. Саме на них, в атмосфері взаємоповаги та відкритості, відбувається плідний обмін знаннями та досвідом, що сприяє розвитку нашої науки.

Цього року програма семінару включає широкий спектр тем, що охоплюють як фундаментальні, так і прикладні аспекти математики. Ми маємо честь слухати доповіді відомих вчених, а також надавати слово молодим дослідникам, які тільки починають свій науковий шлях. Така різноманітність дозволяє створити

багатогранну картину сучасного стану математичних досліджень у сфері комбінаторного аналізу та оптимізації.

Особливо хочу відзначити зусилля Оргкомітету, члени якого зробили все можливе, щоб цей семінар пройшов на високому рівні навіть у нинішніх непростих умовах.

Шановні учасники, ваші ідеї та дослідження є найважливішою складовою нашого семінару. Ваша активна участь додають заходу особливого значення та стимулюють на подальші зусилля в організації подібних зустрічей. Сподіваюся, що цей семінар стане для вас джерелом натхнення, нових знань та наукових контактів.

Бажаю всім плідної роботи, цікавих дискусій та незабутніх вражень від нашого семінару. Нехай кожна доповідь і кожне обговорення сприятимуть вашому професійному та особистому зростанню.

Леонід Гуляницький, член-кор. НАН України, Інститут кібернетики імені В.М.Глушкова НАН України

ASSESSMENT OF THE VULNERABILITY OF THE INFORMATION SYSTEM BASED ON FUZZY MATHEMATICS

Anna V. Bakurova¹, Oleh V. Zaritskyi², Anatoliy A. Gritskevich¹,
Pavlo V. Hrynchenko¹, Elina V. Tereschenko¹ and Dmytro
V. Shyrokorad¹

¹ National University "Zaporizhzhia Polytechnic"

² National Aviation University

ОЦІНЮВАННЯ ВРАЗЛИВОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ НА ОСНОВІ FUZZY-МАТЕМАТИКИ

Abstract. A fuzzy-mathematics approach for assessing the vulnerability of information systems has been proposed. This approach is based on systems analysis, viewing the information system as an interaction between subjects and objects governed by access control policies. The key elements are Assessing the vulnerability of objects, Monitoring anomalies, Evaluating risk levels, and Adjusting access policies. This approach allows for dynamic responses to changes in the system and adaptation of security policies, thereby enhancing the overall level of protection. The methodology entails the deployment of advanced tools and software, including Intrusion Detection Systems (IDS), fuzzy testing, User and Entity Behavior Analytics (UEBA), User Activity Monitoring (UAM), Software Bill of Materials (SBOM), and machine learning techniques. Integral to the methodology are relevant libraries and databases such as the CIS Benchmark, Common Vulnerabilities and Exposures (CVEs), Common Platform Enumeration (CPE) Dictionary, and Common Vulnerability Scoring System (CVSS). These components ensure the standardization and integration of the methodology with other approaches and methods for the control and monitoring of information systems. In this article, the example of calculation of vulnerability estimation for model Object-Subject was considered.

The team of authors has developed a methodology for building access control systems based on fuzzy mathematics. The grounds of the approach are outlined in the work [1]. The methodology involves a combination of modern tools and methods of data analysis and information system states in order to determine the risk level of the access control system. The influence of factors in the "subject-object" system, caused by various aspects, including incorrect actions of the user, is considered. The methodology is based on FUZZY mathematics with the following implementation in the MATLAB package [2]. The presented material is an approbation of the previously developed methodology. It

considers the "subject-object" model of a particular information system, which reflects the structure of relationships between the indicators of the access control testing system to this information system (Fig. 1). We will assume that the subject is an entity that interacts with the system and is endowed with certain rights to perform actions with system objects, for example, a system user or a process. The entity is characterized by the degree of trust in its qualifications and actions in the system. An object is an entity (often a resource) represented by elements of the information system, which also have their own attributes. Software, file system, service, and equipment are considered as objects.

All subjects are characterized by the ability to interact with objects (change, add, update, etc.). The access control policy defines the possibility of the subject's influence on the object, described by the Access Control List (ACL) indicator.

The output parameter is the Access control system risk level, assessed on a scale of 0 to 1. This indicator is calculated using fuzzy derivation and mathematical calculations, using the values of the parameters characterizing both the Object and the Subject.

We will describe the "subject-object" interaction using model (1). Using the Algebra of Statements, we will describe the events according to the state of the indicators, which is presented in Fig. 1.

$$\begin{aligned} AL &= PRM \wedge PR \vee NT \wedge AV \vee AC \wedge AML \vee AR \wedge AML \vee EM, \\ OVL &= VC \wedge ISL \vee (VA \wedge OAF \vee VA) \vee LOI, \\ ACSR &= AL \wedge OVL, \end{aligned} \quad (1)$$

where a possible event for the Subject indicator group will be presented as

$$AML = PML \vee SCA.$$

$$PML = RAL \wedge mPA \wedge ALD \wedge MPA \wedge EPH \wedge ALT \wedge MPL \wedge PCR \wedge SPE,$$

where the entered indicators are described in the standard [3].

Let's introduce linguistic variables for indicators of both Object and Subject corresponding to model (1). The input data of the production system of fuzzy output are the facts of certain states of the system, obtained at certain discrete moments, and dynamic, taking into account the results of constant monitoring of anomalies in the system, which are provided by modern tools and software, such as intrusion detection systems (IDS), fuzzy testing, User and Entity Behavior Analytics (UEBA), User Activity Monitoring (UAM), SBOM. Fuzzification of input data considers the predefined permission level access control list. The knowledge base is formed taking into account standardized

requirements for the security of information systems, such as CIS Benchmark, Common Vulnerabilities and Exposures (CVEs), Common Platform Enumeration (CPE) Dictionary, and Common Vulnerability Scoring System (CVSS). The use of appropriate libraries and databases provides standardization and integration of the methodology with other approaches and methods of control and monitoring of information systems.

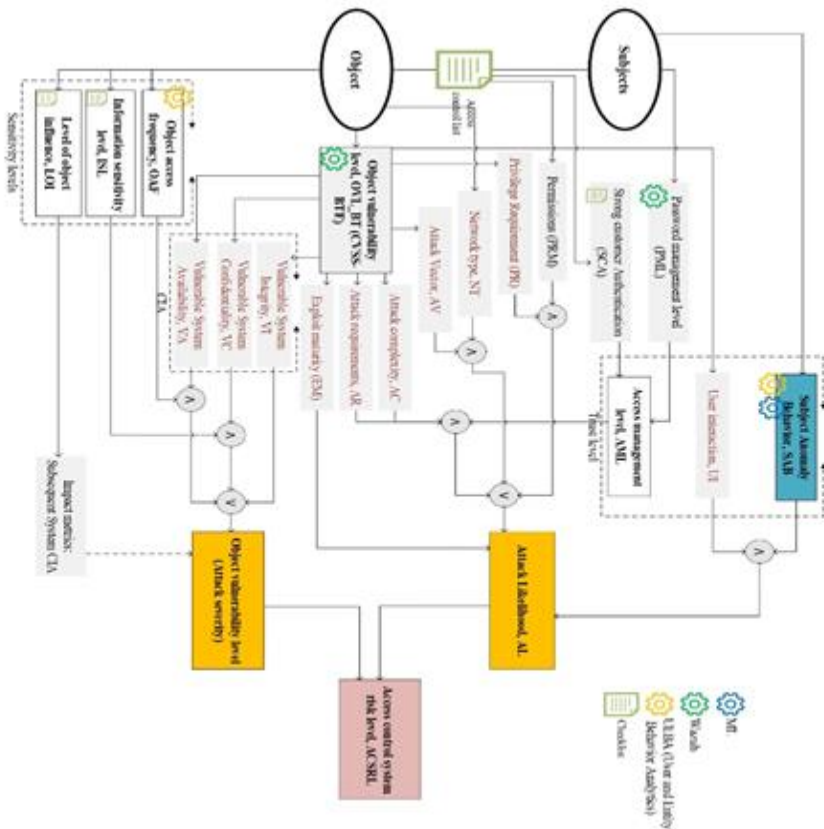


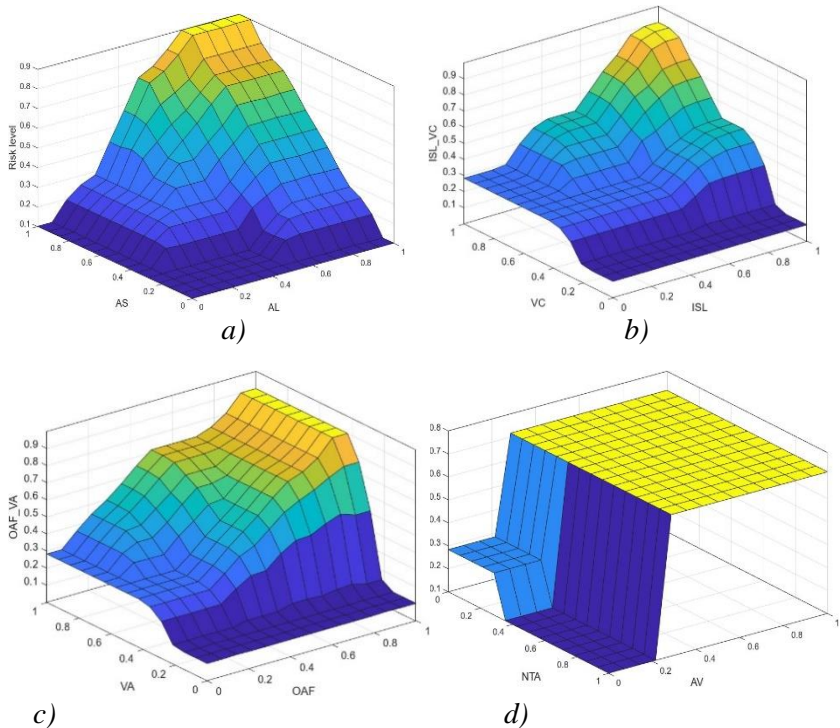
Fig. 1 – Structure of relationships between indicators of the access control testing system

The output linguistic variable of the production system is the ACSRL risk level, the terms and functions of which are defined in the table. 1, according to [4].

Table 1. Linguistic variables Access control system risk level (ACSRL)

Терми ACSRL	Функція приналежності термів
Very high (VH)	$s(0,70\ 0,90)$
High (H)	$P_i(0,50\ 0,65\ 0,82\ 0,90)$
Substantial (S)	$P_i(0,30\ 0,42\ 0,58\ 0,70)$
Possible (P)	$P_i(0,10\ 0,18\ 0,35\ 0,50)$
Slight (S)	$s(0,10\ 0,30)$

The fuzzy inference production system has a hierarchical structure corresponding to model (1). Fig. 2 b)-i) presents some response surfaces of this hierarchical structure, and Fig. 2 a) presents the ACSRL response surface.



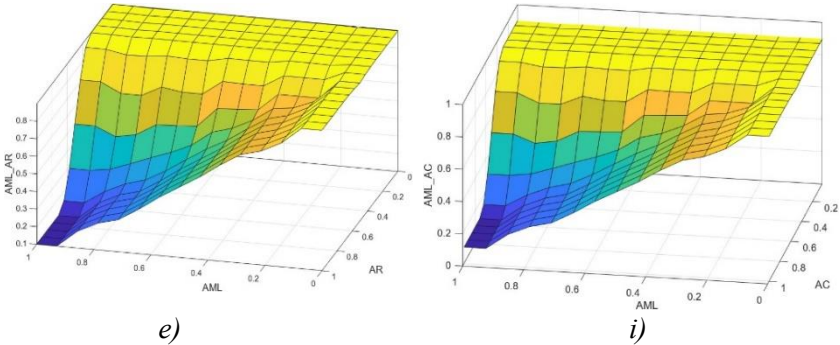


Fig.2 – Surface of fuzzy output a) ACSRL depend on AL & AS, b) VC&ISL, c) OAF&VIA, d) NTA &AV, e) AML&AR, i) AML&AC.

This is an example of system operation for a vector of input data under the conditions of operation of a real "subject-object" system using the built-in programming language of the Matlab package. The subject is the user, described by binary characteristics, and the object is some device characterized by phased data.

```
Subjects(1)=struct("RAL", 1,"mPA", 1,"ALD", 1,"MPA", 1,"EPH", 1,"ALT", 1,"MPL", 1,"PCR", 1,"SPE", 1);
```

```
Objects(2)=struct("Name", "Switch", "OVL", "High", "OAF", "Constantly", "ISL", "High", "LOD", "Low", "LOI", "High", "OVLP", [OVLP(2)]).
```

The vulnerability characteristic of the object looks like this:

```
OVLP(2)=struct ("CVE", "CVE-2020-35221", "VALUE", 8.8, "EM", "Not-Defined", "AV", "Adjacent", "AC", "Low", "PR", "None", "UI", "None", "VC", "High", "VA", "High", "VI", "High", "AR", "None").
```

Characteristics of the system as a whole:

```
NT=struct("NTA", "CAN", "NTS", "Open", "SPI", "Average").
```

The result of the system is a risk assessment ACSRL=0.64922.

Conclusions. The real example shows the operation of the developed system, which allows you to test the object of the system against the unauthorized influence of some subject and determine the assessment of the vulnerability of the "subject-object" connection.

The work was performed as part of the research project "Development of research methods of complex socio-economic systems based on intelligent technologies", registration number SR 0121U113264, in the Department of System Analysis and Computational Mathematics of the National University "Zaporizhzhia Polytechnic", Ukraine, and the Horizon Europe project TELEMETRY.

References

1. S. Parkinson, S. Khana, Identifying high-risk over-entitlement in access control policies using fuzzy logic, *Cybersecurity* 5:6 (2022) 1-17. [doi:10.1186/s42400-022-00112-1](https://doi.org/10.1186/s42400-022-00112-1).
2. MathWorks, 2024. URL: <https://www.mathworks.com/products/matlab.html>.
3. Common Vulnerability Scoring System version 4.0. User Guide. FIRST, 2023. URL: <https://www.first.org/cvss/>.
4. [International standard. Risk management, Risk assessment techniques, IEC 31010, Edition 2.0, 2019.](#)

BEAUTIFUL GRAPHS FROM SIMPLE GROUPS

Peter J. Cameron, University of St Andrews

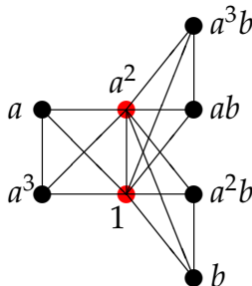
The commuting graph of a group

Let G be a finite group. The commuting graph of G is the graph with vertex set G , in which x and y are joined if $xy = yx$.

Here are the commuting graphs of the two non-abelian groups of order 8:

$$D_8 = \langle a, b : a^4 = 1, b^2 = 1, b^{-1}ab = a^{-1} \rangle \text{ and}$$

$$Q_8 = \langle a, b : a^4 = 1, b^2 = a^2, b^{-1}ab = a^{-1} \rangle.$$



We see that the commuting graph doesn't determine the group; but it carries a lot of information.

In 1955, Brauer and Fowler used the commuting graph to prove that there are only a finite number of simple groups of even order with a given centraliser of an involution (element of order 2).

With hindsight, this was the first step on the road to the Classification of Finite Simple Groups. A huge amount of work was done determining the simple groups with a given involution centraliser.

It is interesting that the paper doesn't contain the word "graph"; graph theory was not mainstream mathematics at the time.

A number of further graphs on groups have been discussed.

Among them are

- ▶ the power graph: x and y joined if one is a power of the other;
- ▶ the enhanced power graph: x and y joined if they are both powers of an element z (equivalently, $\langle x, y \rangle$ is cyclic).

I will talk about the difference graph $D(G)$, in which x and y are joined if $\{x, y\}$ is an edge of the enhanced power graph but not of the power graph.

I believe that similar results will be found for many of the other graph types, but this has not been studied.

Vertices x and y of a graph are twins if they have the same neighbours, possibly except for one another.

If a graph contains twins, then the information it contains can be expressed more compactly, since we only have to give the neighbour set of x and also whether or not x and y are joined.

So it is natural to apply twin reduction: delete one of a pair of twins, and continue until no twins remain.

Theorem. The result of twin reduction is (up to isomorphism) independent of the way the reduction is performed.

A graph is a cograph if it has no induced subgraph which is a four-vertex path.

Theorem. Applying twin reduction to a graph G yields the trivial 1-vertex graph if and only if G is a cograph.

Twin reduction preserves many graph-theoretic properties. For example, if H is a twin-free graph, then H is an induced subgraph of G if and only if H is an induced subgraph in the graph obtained from G by twin reduction.

Connected components which are not cographs remain connected after twin reduction.

Graphs defined on groups often have many pairs of twins. For example, if x and y generate the same cyclic subgroup (that is, if $y = x^k$ where k is coprime to the order of x), then x and y are twins.

So it is interesting to apply twin reduction to these graphs.

First, of course, we want to know whether the graph is a cograph!

Question

For each type T of graph on groups, determine the class of groups G for which $T(G)$ is a cograph.

Despite a lot of work, most cases are not completely resolved!

It turns out, that for small finite simple groups, $D(G)$ is a cograph. For slightly more complicated examples, $D(G)$ may not be very interesting; for example, the twin reduction may consist of a large number of small isomorphic graphs. When we get to the Mathieu group M_{11} , something more interesting happens. The twin reduction of $D(M_{11})$ is a beautiful graph:

- ▶ it is bipartite, with bipartite blocks of sizes 165 and 220;
- ▶ it is semiregular, with valencies 4 in one block and 3 in the other;
- ▶ it has diameter 10 and girth 10;
- ▶ its automorphism group is just M_{11} .

Note that $165 + 220 = 385$ is much smaller than $|M_{11}| = 7920$.

All these computations were done in GAP, using the package GRAPE for graphs and groups.

It is straightforward to construct the graph and perform twin reductions on it. Then many properties of the resulting graph can be read out by simple GRAPE commands.

This is as far as the exploration has gone so far, but I would be delighted if you would join me in pushing the analysis further!

See Sucharita Biswas, Peter J. Cameron, Angsuman Das, Hiranya Kishore Dey, On difference of enhanced power graph and power graph of a finite group, arXiv 2206.12422.

For background, see Peter J. Cameron, Graphs defined on groups, Internat. J Group Theory 11 (2022), 43–124

ДОСЛІДЖЕННЯ ВИМОГ ДО КІБЕРБЕЗПЕКИ КОМП'ЮТЕРНИХ СИСТЕМ УПРАВЛІННЯ АЕС

Вінтенко Б.Ю., Смірнова Т.В., Смірнов О.А.

Центральноукраїнський національний технічний університет

STUDY OF REQUIREMENTS FOR CYBER SECURITY OF COMPUTER SYSTEMS OF NPP MANAGEMENT

Abstract. In order to counter cryptoattacks on elements of critical infrastructure, in particular on computer control systems of nuclear power plants, this work achieved the goal of analyzing the requirements for computer security (cyber security) of the software of this system, which are relevant to the stage of its design, development and operation. To achieve the goal, the article provides information on standards and identified requirements for software cyber security.

На сучасному етапі розвитку комп'ютерних систем управління (КСУ) АЕС процеси з забезпечення кібербезпеки КСУ АЕС є надзвичайно актуальними та є складними і комплексними. У статті [1] розглядаються процеси впровадження та підтримки політики, програми та плану кібербезпеки на АЕС.

Забезпечення кіберзахисту КСУ критичного застосування є комплексною задачею, яка передбачає поєднання законодавчих, культурних, технологічних, адміністративних заходів. В дослідженні [2] розроблено модель керування кібербезпекою (*Cyber Security Management Model*), в якій виділяються шість головних сфер, що є критичними для кібербезпеки:

- правове регулювання (Legal Regulation);
- розуміння керівництвом (Good Governance) важливості забезпечення кібербезпеки та мінімізації наслідків кіберінцидентів;
- керування ризиками (Risk Management);
- культура безпеки (Security Culture);
- керування технологіями (Technology Management);
- керування інцидентами (Incident Management).

Аналогічні сфери кібербезпеки КСУ критичного застосування відображаються в міжнародних стандартах та нормативних документах різних країн. Зокрема, вимоги до комп'ютерної безпеки (кібербезпеки) КСУ АЕС містяться в міжнародному стандарті ІЕС62645 [3] та галузевому стандарті України «НП 306.2.237-2022» [4].

Стандарти, що розглядаються, вводять класифікацію рівнів кіберзахисту КСУ, їх компонентів та ПЗ для визначення вимог до кібербезпеки (розділ 5.2.3.1 стандарту ІЕС62645, розділ II стандарту НП 306.2.237-2022). Ці рівні визначаються категоріями виконуваних функцій, важливих для безпеки (категорії «А», «В», «С» визначені у [5]). Визначаються три рівні: «S1» («К1») (виконання функцій категорії «А»), «S2» («К2») (виконання функцій категорії «В» та робота в реальному часі) та «S3» («К3») (виконання функцій категорії «С», а також допоміжних та обслуговуючих функцій). Зазначається, що визначення рівня не є жорстко регламентованим, і виконується на основі аналізу функцій КСУ.

Згідно дослідження вимог стандартів (розділ 5.1 стандарту ІЕС62645, розділ III-IV стандарту НП 306.2.237-2022), **основні загальні принципи забезпечення кіберзахисту** є наступними:

- наявність політики кіберзахисту;
- глибокошелонований кіберзахист;
- диференційований підхід;
- заходи щодо попередження шкідливих дій, виявлення та реагування на шкідливі дії, пом'якшення їх наслідків та відновлення;
- заходи щодо попередження внесення несанкціонованих змін, надлишкових функцій, некоректних даних тощо;
- заходи щодо культури кіберзахисту;
- заходи щодо мінімізації впливу кіберзахисту на якість виконання основних функцій КСУ;
- заходи щодо оцінювання та переоцінювання кіберзахисту КСУ, що знаходяться в експлуатації.

Згідно дослідження вимог стандартів щодо **забезпечення кіберзахисту на етапі розробки та експлуатації** (розділ 5.2.3.2.3 – 5.2.3.2.7, 6.4 «ІЕС62645», розділ V-VII – «НП 306.2.237-2022») ПЗ КСУ необхідно врахування основних нижченаведених аспектів:

- Процес та засоби розробки.
- Контроль за наявністю прихованих функцій.
- Мінімізація впливу засобів кіберзахисту на виконання функцій.
- Напрямок передачі інформації.
- Контроль потоку даних.
- Автентифікація та авторизація:
 - а) фізичний захист компонентів КСУ;
 - б) захист від несанкціонованого доступу.

Висновок. Дослідження стандартів кібербезпеки КСУ АЕС показують, що виконання вимог кібербезпеки є обов'язковим на всіх етапах життєвого циклу продукту. Необхідною умовою реалізації заходів кібербезпеки програмного забезпечення (ПЗ) є визначення його компонентів, які можуть бути об'єктами кібератак та містити вразливості, що можуть бути причиною відмов при виконанні функцій КСУ. Після визначення цих компонентів ПЗ в них має бути програмно реалізовані функції для забезпечення розглянутих аспектів кібербезпеки та проведено тестування цих функцій.

Література

1. Сімонов А.А., Клевцов О.Л., Трубочанінов С.А., Лазуренко О.П. «Комп'ютерна безпека інформаційних та керуючих систем АЕС: документи, що обґрунтовують комп'ютерну безпеку». Ядерна та радіаційна безпека, 2019, вип. 4(84), С. 73-81.

2. Limba, T.; Plêta, T.; Agafonov, K.; Damkus, M. «Cyber security management model for critical infrastructure». Entrepreneurship and Sustainability Issues, 2017, 4(4): pp. 559-573.

3. IEC62645-2014: Nuclear power plants – Instrumentation and control systems – Requirements for security programmes for computer-based systems. Geneva, International Electrotechnical Commission (IEC), 2014.

4. НП 306.2.237-2022. Вимоги до кіберзахисту інформаційних та керуючих систем атомних станцій для забезпечення ядерної та радіаційної безпеки. Київ, Держатомрегулювання України, 2022.

5. IEC61226-2009: Nuclear power plants – Instrumentation and control important to safety – Classification of instrumentation and control functions. Geneva, International Electrotechnical Commission (IEC), 2009.

6. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». Системи управління, навігації та зв'язку, 2023, вип. 2(72), С. 170-178.

7. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів IEC60880 та IEC62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки».

**ПРО КВАЗИВИПАДКОВІ ГРАФИ
НА ОСНОВІ 3-РЕБЕРНО ЗВ'ЯЗНИХ ГРАФІВ**

Глухов О.Д.

Національний авіаційний університет

**QUASI-RANDOM GRAPHS
BASED ON 3-EDGE CONNECTED GRAPHS**

Abstract. We address the problem of estimating the probability of connectivity in quasi-random graphs. We introduce the concepts of multiframework and connected graph polynomial. Additionally, we present a new connectivity estimate for quasi-random graphs based on a 3-edge connected graph.

Квазівипадкові графи.

Квазівипадкові графи є моделями дискретних систем, структура яких може змінюватись внаслідок випадкового розриву частини зв'язків [1-3].

В цій статті розглядається задача оцінки ймовірності зв'язності квазівипадкового графа $G(p)$ побудованого на основі - 3-реберно зв'язного графа G .

Означення 1. Нехай $G = G_n$ - зв'язний граф на n вершинах з множиною вершин G^0 і множиною ребер G_n^1 , $|G^0| = n$, $|G^1| = m$, квазівипадковим графом на основі графа G називається граф $G(p)$ з множиною $G^0(p) = G^0$ вершин і з випадковою множиною $U = G^1(p)$ ребер для якої виконуються умови:

$$Prob(u \in U) = p, \text{ якщо } u \in G^1;$$

$$Prob(u \in U) = 0, \text{ якщо } u \notin G^1.$$

Мультикаркаси графа і С-поліноми.

Нехай G - зв'язний граф, $U \subseteq G^1$ - деяка множина його ребер, через $G[U]$ будемо позначати мінімальний підграф графа G з множиною ребер $(G[U])^1 = U$.

Означення 2. Мультикаркасом графа G будемо називати сімейство множин $\{E_j\}_{j=1}^s$, яке задовольняє наступним умовам:

$$(1) \forall j E_j \subseteq G^1$$

(2) Якщо множина W ребер графа G задовольняє умові $\forall j W \cap E_j \neq \emptyset$, то граф $G[W]$ буде зв'язним факторграфом графа G (зв'язним підграфом, який містить усі вершини даного графа).

Означення 3. Нехай $M = \{E_j\}_{j=1}^s$ деякий мультикаркас графа G , $\xi_k(M) = |\{E \in M : |E| = k\}|$. C -поліномом або зв'язністним поліномом мультикаркаса графа G назвемо поліном, визначений наступним чином:

$$C(G, M, x) = \sum_k \xi_k x^k.$$

Важливість C -полінома впливає з наступного твердження.

Теорема 1 [4]. Якщо G зв'язний граф, M - деякий його мультикаркас, то для ймовірності P зв'язності квазівипадкового графа $G(p)$ на основі графа G має місце наступна оцінка:

$$P \geq 1 - C(G, M, q), \text{ де } q = 1 - p.$$

Гомоморфізми.

Означення 4. Регулярним гомоморфізмом f графа G на граф H будемо називати сюр'єктивне відображення множини вершин графа G на множину вершин графа H яке задовольняє наступним умовам:

1) прообраз кожної вершини графа G є зв'язним підграфом графа H ;

2) існує таке ін'єктивне відображення множини ребер графа H в множину ребер графа G , що

$$g(w) = u \Rightarrow \partial w = (f(x), f(y)) \& (\partial u = \{x, y\}) \& (f(x) \neq f(y))$$

Фактично граф H отримується з графа G шляхом ретракції (стягування) в вершини деяких його породжених зв'язних підграфів. Таким чином, прообразом кожної вершини графа H є породжений зв'язний підграф графа G , зокрема це може бути тривіальний прообраз, тобто підграф, який складається з однієї вершини.

Означення 5. Регулярний гомоморфізм f графа G на граф H будемо називати простим циклічним гомоморфізмом, якщо не

більше, ніж одна вершина графа H має нетривіальний прообраз, який є простим циклом графа G .

Лема 1. Нехай f - простий циклічний гомоморфізм графа G на граф H , $Z = f^{-1}(z)$ - єдиний нетривіальний прообраз цього гомоморфізму, $M(Z), M(H)$ - деякі мультикаркаси графів Z і H відповідно, то $M(Z) \cup M(H)$ буде мультикаркасом графа G .

Лема 2. Нехай f - простий циклічний гомоморфізм графа G на граф H , $Z_k = f^{-1}(z)$ - єдиний нетривіальний прообраз цього гомоморфізму, Z_k - простий цикл на k вершинах, $C(H, M', x)$ - деякий C -поліном графа H , тоді $C(G, M, x) = C(H, M', x) + \frac{k(k-1)}{2} x^2$ буде C -поліномом графа G .

Головний результат.

Теорема 2. Нехай $G_n(p)$ квазівипадковий граф на основі 3-реберно зв'язного графа G_n , з n вершинами, $q = 1 - p$. Тоді для ймовірності P зв'язності графа $G_n(p)$ має місце наступна оцінка:

$$P \geq 1 - (n \log_2 n) q^2.$$

Доведення. Проводиться індукцією по числу вершин графа, враховуючи твердження лем 1, 2 а також той відомий факт, що 3-реберно зв'язний граф G , $|G^0| = n$, містить простий цикл довжини не більше $2 \log_2 n$ [2].

Висновок. Якщо $G_n(p)$ квазівипадковий граф на основі 3-реберно зв'язного графа G_n , з n вершинами, $q = 1 - p$,

$q = \frac{\alpha}{\sqrt{n \log_2 n}}$, то для ймовірності P зв'язності графа $G_n(p)$ має місце наступна оцінка: $P \geq 1 - \alpha^2$.

Неформально, квазівипадковий граф $G_n(p)$ на основі 3-реберно зв'язного графа G_n на n вершинах з достатньо високою

ймовірністю залишитися зв'язним при видаленні випадкових k ребер, якщо k істотно менше, ніж $\sqrt{n/\log_2 n}$.

Література

1. Frieze A, Karoński M. Introduction to Random Graphs. Cambridge University Press; 2024. — 576 P.
2. Diestel R. Graph Theory, Springer-Verlag, Heidelberg, Graduate Texts in Mathematics, Volume 173, 2017. — 428 P.
3. Глухов А.Д., Квазислучайные графы и структурная устойчивость сложных дискретных систем // Электронное моделирование, т. 38, №5, 2016, с.35-41.
4. Глухов О.Д., Про зв'язність квазівипадкових графів. – Матеріали Міжнародного науково-практичного семінару імені А.Я. Петренюка «Комбінаторні конфігурації та їхні застосування» 14-16 червня 2023.-Запоріжжя-Кропивницький, 2023.- с.109-112.

ПРО ЦІЛЬОВІ РЕКОНФІГУРАЦІЇ МЕРЕЖ

Василь Горбачук, Дмитро Ніколенко, Віктор Годлюк, Дмитро Рибачок

Інститут кібернетики імені В.М.Глушкова НАН України

ON TARGET RECONFIGURATIONS OF NETWORKS

Abstract. Traveling through the transport network or sending packets over the Internet is implicitly based on game-theoretic considerations: a specific decision-maker, choosing his or her route, takes into account the probability of congestion depending on all decision-makers, that is, other routes. Based on similar considerations, it is possible to develop models for network traffic. These models explain some paradoxical observations where, under certain circumstances, increasing the capacity of a given network can slow down its traffic. Adding a new strategy may worsen the equilibrium payoff of each player. For example, in a Prisoner's Dilemma, if each player's only strategy is cooperation, then each player will benefit from that compared to a situation where each player's strategies are competition (distrust the other player, confession) and cooperation. Such a phenomenon seems paradoxical, because updating the network is intuitively associated with its improvement. Therefore, any update should correspond to a certain measurable criterion of social gain, which each person understands in his or her own way. The reconfiguration of any network involves a certain coordination of all its participants.

Поїздка транспортною мережею чи надсилання пакетів через Інтернет неявно ґрунтується на теоретико-ігрових міркуваннях: конкретна особа, яка приймає рішення (ОПР), вибираючи свій маршрут, зважає на ймовірність виникнення заторів, залежних від усіх ОПР, тобто від інших маршрутів [1]. Виходячи з подібних міркувань, можна розробляти моделі для мережевого трафіку [2]. Ці моделі пояснюють деякі парадоксальні спостереження, коли збільшення спроможності даної мережі може за певних обставин сповільнювати її трафік [3–5].

Спочатку розробимо модель транспортної мережі та дослідимо, як вона реагує на затори [6]. Представимо транспортну мережу орієнтованим графом [7], де ребрами є автомагістралі, а вузлами є виїзди (exits), де ОПР може виїхати з певної автомагістралі та заїхати на іншу. Припустимо, що кожна ОПР бажає проїхати від вузла A (скажімо, виїзду до передмістя) до вузла B (виїзду до центру міста) [8]. Слід враховувати велику кількість інших ОПР проїхати від A до B у годину пік [9]. Кожне ребро характеризується певним часом на дорогу, залежним від трафіку на цьому ребрі.

Для конкретності припустимо, що 4000 автомобілів (відповідних ОПР) дістаються від A до B маршрутами 1, 2. Рівноважний трафік становить 2000 автомобілів на кожний маршрут з часом поїздки 65 хвилин. Нехай на маршруті 1 є вузол C , а на маршруті 2 є вузол D , причому поїздка підмаршрутом $C-B$ займає 45 хвилин, як і поїздка підмаршрутом $A-D$: ці підмаршрути є нечутливими до заторів, а час поїздки кожним таким підмаршрутом не залежить від кількості автомобілів, які рухаються ним.

З іншого боку, припустимо, що час поїздки підмаршрутом $A-C$ або підмаршрутом $D-B$ дорівнює $x(A-C)/100$ хвилин та $x(D-B)/100$ хвилин відповідно, де $x(A-C)$ та $x(D-B)$ – кількість автомобілів, які рухаються підмаршрутом $A-C$ та $D-B$ відповідно. Тоді при $x(A-C) = 2000 = x(D-B)$ сумарний час поїздки маршрутом $A-C-B$ або $A-D-B$ становить $2000/100 + 45 = 65$ хвилин. Якщо ж $x(A-C) = 4000$, то сумарний час поїздки маршрутом $A-C-B$ становить $4000/100 + 45 = 85$ хвилин; аналогічно сумарний час поїздки маршрутом $A-D-B$ становить 85 хвилин при $x(D-B) = 4000$.

На мові теорії ігор тут ОПР (водії) є гравцями, а стратегією кожного з них є вибір маршруту $A-C-B$ або $A-D-B$. У більших мережах гравці мають більше стратегій. Виграш кожного гравця –

це мінус сумарний час поїздки вибраним маршрутом. Можна очікувати формування рівноважного трафіку, скажімо, рівноваги Неша (список 4000 стратегій, де стратегія кожного гравця $i = 1, \dots, 4000$ – це найкраща реакція цього гравця на задані списком стратегії всіх інших гравців $1, \dots, i-1, i+1, \dots, 4000$). Бувають домінуючі та змішані стратегії.

В цій грі з трафіком загалом немає домінуючої стратегії: будь-який маршрут має потенціал бути найкращим вибором для даного гравця, якщо решта гравців використовують інший маршрут. Проте ця гра має рівноваги Неша (Джон Неш (1928–2015) – Нобелівський лауреат 1994 р.): можна показати, що будь-який список стратегій, в якому поведінка водіїв веде до однакової кількості (2000) автомобілів на кожному маршруті, є єдиною рівновагою Неша. Рівний баланс (автомобілів) між маршрутами дає рівновагу Неша, бо за рівномірного балансу між двома маршрутами будь-який водій не має стимулу переключатися на інший маршрут.

Всі рівноваги Неша мають рівний баланс тому, що при $x(A-C) > 2000$, $x(A-D) = 4000 - x(A-C) < 2000$ сумарний час поїздки маршрутом $A-C-B$ становить $x(A-C)/100 + 45 > 65$ (хвилин) і перевищує сумарний час $x(A-D)/100 + 45 < 65$ (хвилин) поїздки маршрутом $A-D-B$, а тому водій на маршруті $A-C-B$ має стимул переключитися на інший маршрут $A-D-B$. Отже, будь-який список стратегій, в якому $x(A-C) \neq 2000$, не може бути рівновагою Неша; будь-який список стратегій, в якому $x(A-C) = 2000$, є рівновагою Неша.

Таким чином, на даній мережі корислива поведінка всіх водіїв змушує їх у рівновазі досконало балансувати між наявними маршрутами. Однак на практиці мережі розвиваються і змінюються. Нехай, наприклад, міська влада вирішила побудувати додаткову швидкісну автомагістраль від C до D , причому час поїздки через нове ребро $C-D$ дорівнює 0 хвилин (або достатньо мале число ε хвилин). Тоді, незважаючи на оновлення системи автомагістралей, час поїздки від A до B в стані рівноваги становитиме 80 хвилин і перевищуватиме час поїздки 65 хвилин до оновлення системи, бо всі автомобілі використовуватимуть підмаршрут $C-D$.

Оновленій мережі відповідає інша єдина рівновага Неша з часом поїздки 80 хвилин від A до B . У стані рівноваги кожний водій використовує маршрут $A-C-D-B$, якому відповідає

сумарний час поїздки $x(A-C)/100+0+x(D-B)/100=4000/100+0+4000/100=80$ хвилин при $x(A-C)=4000=x(D-B)$: якщо водій замість $A-C-D-B$ використовує $A-C-B$, то сумарний час поїздки дорівнює $x(A-C)/100+45=4000/100+45=85$ (хвилин); якщо водій замість $A-C-D-B$ використовує $A-D-B$, то сумарний час поїздки дорівнює $45+x(D-B)/100=45+4000/100=85$ (хвилин). Створення згаданого ребра $C-D$ фактично зробило відповідний підмаршрут $C-D$ доміантною стратегією для кожного водія: незалежно від поточної схеми руху, кожний водій виграє при переключенні на підмаршрут $C-D$. Тоді цей підмаршрут притягує до себе всіх водіїв, в результаті чого всі вони програють: при заданій корисливій поведінці водіїв, в оновленій мережі немає способу повернутися до рішення рівномірного балансу попередньої мережі.

Дітріх Браесс (здобув науковий ступінь доктора наук з математики (квантова теорія) Університету Гамбурга (заснованого у 1919 р.) у 1964 р. за дисертацію «Вплив взаємодії кінцевого стану на електророзщеплення дейтрона біля порогу») вперше продемонстрував приклад, коли збільшення ресурсів транспортної мережі не гарантує підвищення її продуктивності в рівновазі, а при певних комбінаціях параметрів знижує її продуктивність [3]. Таке явище називають парадоксом Браесса. На користь цього явища свідчить кейс у місті Сеул (Корея), де будівництво громадського парку на місці шосе з шістьма смугами руху (six-lane highway) фактично зменшило час поїздки в місто та з нього за приблизно однакового обсягу трафіку [10].

Вищенаведений приклад мережі з вузлами A , B , C , D пояснює парадокс Браесса: додання нової стратегії може погіршувати рівноважний вигравш кожного гравця. Наприклад, якщо в дилемі в'язня єдиною стратегією кожного гравця є кооперація (довіра до іншого гравця, незізнання), то кожний гравець від цього виграватиме порівняно із ситуацією, коли стратегіями кожного гравця є конкуренція (недовіра до іншого гравця, зізнання) і кооперація. Подібне явище виглядає парадоксальним, бо оновлення мережі інтуїтивно асоціюється з її поліпшенням. Тому будь-якому оновленню має відповідати певний вимірюваний критерій суспільного вигравшу, який кожна людина розуміє по-своєму.

Парадокс Браесса фактично є відправною точкою для великої проблематики з теоретико-ігрового аналізу мережевого трафіку. Наприклад, можна вивчати питання вимірювання рівня оновлення

мережі, рівня зміни виграшу кожного гравця, рівня зміни суспільного виграшу тощо. Можна припускати, що час поїздки ребром $A-C$ є лінійною функцією $ax+b$, де a, b – деякі параметри (у вищезгаданому прикладі $a=100^{-1}$, $b=0$). Для таких функцій на ребрах Єва Гардош (здобула наукові ступені магістра і доктора наук з математики Університету Етвеша Лоранда (заснованого у 1635 р. у м.Будапешт (Угорщина)) у 1981 р. і 1984 р. відповідно) та її учень Тім Рафгарден (здобув наукові ступені бакалавра наук і магістра наук Стенфордського університету (заснованого у 1885 р.) у 1997 р. і 1998 р. відповідно, доктора філософії з комп'ютерних наук Корнельського університету (заснованого у 1864 р.) у 2002 р. за дисертацію «Егоїстична маршрутизація») показали, що після додання ребер до мережі з рівноважним шаблоном трафіку завжди існуватиме рівновага оновленої мережі з часом поїздки, який перевищує рівноважний час поїздки попередньої мережі не більше, ніж на третину [5, 11, 12].

У вищезгаданому прикладі до оновлення мережі рівноважний час поїздки маршрутом $A-C-B$ або $A-D-B$ дорівнює 65 хвилин, а після оновлення рівноважний час поїздки маршрутом $A-C-D-B$ дорівнює 80 хвилин, то $(80-65)/65=15/65=3/13 < 1/3$. Якщо до оновлення мережі рівноважний час поїздки маршрутом $A-C-B$ або $A-D-B$ становить $2000/100+40=60$ хвилин (поїздка підмаршрутом $C-B$ займає 40 хвилин, як і поїздка підмаршрутом $A-D$), а після оновлення рівноважний час поїздки маршрутом $A-C-D-B$ дорівнює 80 хвилин, то $(80-60)/60=20/60=1/3$. Коли час поїздки ребром є нелінійною функцією, то відношення часу поїздки після оновлення до часу поїздки перед оновленням може перевищувати $4/3$.

Проблематика з теоретико-ігрового аналізу мережевого трафіку теж включає способи проектування мереж [13], щоб запобігати виникненню небажаних рівноваг, і способи регулювання мереж, щоб уникати появи небажаних рівноваг [14], наприклад, через розумне використання тарифів за проїзд у певних частинах мережі [15]. Питанням мереж присвячена книга [6], перший автор якої Девід Ізлі здобув науковий ступінь доктора наук з економіки Північно-Західного університету (заснованого у 1851 р.) у 1979 р., а другий автор Йон Клейнберг здобув наукові ступені бакалавра інформатики і математики Корнельського університету у 1993 р., магістра і доктора філософії з комп'ютерних наук Массачусетського

технологічного інституту (заснованого у 1861 р.) у 1994 р. і 1996 р. відповідно.

Парадокс Браесса є одним з аспектів того загального явища, що мережевий трафік у рівновазі може не бути соціально оптимальним [16]. Важливо вимірювати відстань між рівноважним трафіком і соціально оптимальним трафіком для загальної мережі, якою може бути будь-який орієнтований граф. Є набір гравців (ОПР, водіїв), які перебувають у загалом різних початкових положеннях і мають загалом різні пункти призначення (destinations).

Кожне ребро (edge e) характеризується своєю функцією $T_e(x)$ часу поїздки, яка задає час (time), потрібний усім водіям, щоб проїхати ребром e , коли x водіїв їдуть цим ребром. Така функція відіграє роль мітки даного ребра. Припустимо, що кожна функція часу поїздки є лінійною за обсягом трафіку: $T_e(x) = a_e x + b_e$, де a_e , b_e – деякі невід’ємні параметри. Нехай $T_{AC}(x) = a_{AC}x + b_{AC} = x = a_{DB}x + b_{DB} = T_{TB}(x)$, $T_{CB}(x) = a_{CB}x + b_{CB} = 5 = T_{AD}(x) = a_{AD}x + b_{AD}$.

Схема дорожнього руху (шаблон трафіку; traffic pattern) – це вибір шляху кожним водієм, а соціальна вартість даної схеми дорожнього руху – це сумарний час поїздок усіх водіїв, потрібний їм за використання цієї схеми. Нехай є 4 гравці, $T_{CD}(x) = a_{CD}x + b_{CD} = 0$. Тоді є схема 1, де всі гравці використовують маршрут $A-C-D-B$ (кожний гравець потребує для поїздки $x+0+x=4+0+4=8$ (хвилин)), і є схема 2, де 2 гравці використовують маршрут $A-C-B$ (кожний гравець потребує для поїздки $x+5=2+5=7$ (хвилин)) і 2 гравці використовують маршрут $A-D-B$ (кожний гравець потребує для поїздки $5+x=5+2=7$ (хвилин)). Соціальна вартість схеми 1 (єдиної рівноваги Неша) становить $4 \times 8 = 32$, а соціальна вартість схеми 2 (соціально оптимальної схеми) становить $4 \times 7 = 28$. Схему (дорожнього) руху, що досягає мінімально можливих соціальних витрат, називають соціально оптимальною (соціальним оптимумом). Оскільки в цій мережі є різні схеми руху, що також досягають соціальної вартості 28 (хвилин), то є кілька соціально оптимальних схем руху. Соціально оптимальні схеми руху максимізують суспільний добробут цієї гри з трафіком, бо сума вигравів водіїв дорівнює мінус соціальна вартість.

Основні питання – довести існування рівноважної схеми руху в довільній мережі з лінійними функціями часу поїздки (є приклади

ігор, де не існує рівноваги в чистих стратегіях) і довести існування рівноважної схеми руху, соціальна вартість якої достатньо близька до соціального оптимуму [12]. Результати [12] були узагальнені в роботі [11] (перший автор цієї роботи є учнем Йона Клейнберга і здобув науковий ступінь доктора філософії з комп'ютерних наук Корнельського університету у 2005 р.; другий автор цієї роботи є учнем Джона Хопкрофта (здобув наукові ступені магістра і доктора філософії з електротехніки Стенфордського університету у 1962 р. і 1964 р. відповідно; один з авторів відомої книги [17], яка перекладалася) і здобув науковий ступінь доктора філософії з комп'ютерних наук Корнельського університету у 2006 р.; п'ятий автор цієї роботи є учнем Єви Тардош і здобув науковий ступінь доктора філософії з комп'ютерних наук Корнельського університету у 2005 р.). Фактично кожна рівноважна схема руху має суспільні витрати, які не перевищують оптимальні більше, ніж на третину. Тому за лінійної функції поїздки на кожному ребрі соціальна вартість ситуації парадоксу Браесса гірша ситуації соціального оптимуму не більше, ніж на третину. Простіше доводити те, що соціальна вартість ситуації парадоксу Браесса гірша ситуації соціального оптимуму не більше, ніж вдвічі.

У питанні схеми руху за рівноваги спочатку доведемо існування власне рівноваги через конструктивну процедуру її пошуку. Крок 0: Процедура починається з будь-якої схеми руху. Крок 1: Якщо ця схема руху є рівновагою, то процедура завершується. Крок 2: Якщо ж ця схема руху не є рівновагою, то є принаймні один водій, найкращою реакцією якого, з урахуванням дій решти водіїв, є деякий альтернативний шлях, що забезпечує менший час поїздки. Крок 3: Обираємо одного такого водія і переключаємо його на цей альтернативний шлях, дістаючи нову схему руху. Переходимо на Крок 1.

Ця процедура називається динамікою найкращої реакції, оскільки вона динамічно реконфігурує стратегії гравців, постійно спонукаючи якогось гравця шукати свою найкращу реакцію на поточну ситуацію. Зупин цієї процедури означає, що кожний гравець фактично реалізує свою найкращу реакцію на поточну ситуацію, тобто має місце рівноважна схема руху. Тому достатньо показати, що динаміка найкращої реакції, яка починається з будь-якої схеми руху, має завершитися в стані рівноваги [1].

Наведемо приклад гри з нескінченною динамікою, яка не має рівноваги. Нехай кожний з гравців 1, 2 має монету і показує іншому гравцю орел (head) або решку (tail) своєї монети; гравці діють

одночасно. Якщо обидва одночасно показують орла (обирають стратегію Н), то виграш гравця 2 становить +1, а виграш гравця 1 становить -1 (гра є антагоністичною); якщо обидва одночасно показують решку (обирають стратегію Т), то виграш гравця 2 становить +1, а виграш гравця 1 становить -1; в інших випадках (одночасних дій) виграш гравця 2 становить -1, а виграш гравця 1 становить +1. Ця проста гра відома як зіставлення сторін монет (Matching Pennies). Стратегії та відповідні виграші гравців можна записати у матричному вигляді.

Зіставлення сторін монет є простим прикладом великого класу важливих ігор з тією властивістю, що сума виграшів гравців завжди дорівнює нулю. Тому такі ігри атаки-захисту називають іграми з нульовою сумою. Прикладом подібної атаки вважають висадку союзників (156 тис. військових, зокрема 73 тис. від США, 61,715 тис. від Великобританії, 21,4 тис. від Канади) через Ла-Манш у Європі влітку 1944 р. під час Другої світової війни, які могли обрати місце висадки – місто Кале, найближче до Англії місто Франції (на відстані 34 км), чи береги Нормандії (і Бретані). В ролі гравця 2 виступала Німеччина, яка на місці висадки могла посилювати свої сили, а в ролі гравця 1 виступали союзники [18, 19]. Для цього кейсу розроблено десятки комп'ютерних ігор. 6 червня 2024 р. у Франції союзники відзначають 80-річчя висадки у Нормандії.

У грі зіставленні сторін монет немає пари стратегій, кожна з яких є найкращою реакцією на стратегію іншого гравця: для будь-якої пари стратегій, гравець, який отримує виграш -1, має стимул покращити свій виграш до +1 шляхом зміни своєї стратегії. Тоді уваги заслуговує гібрид цієї гри з іншою базовою грою координації, яку за Руссо (1712–1778) називають полюванням на оленя (Stag Hunt) [20]. Якщо кожний мисливець на полюванні вибирає стратегію кооперації, то мисливці разом можуть зловити оленя; якщо мисливець на полюванні вибирає стратегію конкуренції, то може зловити зайця; якщо гравець 1 обирає стратегію кооперації, а гравець 2 – стратегію конкуренції, то гравець 2 може зловити зайця, а гравець 1 самостійно не зможе зловити оленя і не ловитиме зайця.

Простим важливим прикладом є гра координації (Coordination), де два співавтори готують слайди для спільної презентації проекту, але співавтор 1 не може зв'язатися із співавтором 2 по телефону, коли починає працювати над слайдами і має прийняти рішення про використання Microsoft PowerPoint або Apple Keynote для підготовки слайдів. Будь-яке рішення є прийнятним, але набагато легше об'єднувати слайди, коли обидва співавтори використовують

однакове ПЗ. Якщо вони обирають однакове ПЗ, то виграш кожного дорівнює 1; якщо ж вони обирають різне ПЗ, то виграш кожного дорівнює 0. Таку ситуацію називають грою координації тому, де спільна мета гравців полягає у координації їхніх стратегій. Складність полягає у тому, що ця гра має дві рівноцінні рівноваги Неша, але обидва гравці програють за відсутності координації.

Реконфігурація будь-якої мережі передбачає певну координацію всіх її учасників, а також деякі регулювання, відомі в енергомережах, мережах автомобільного, залізничного, річкового, морського, повітряного транспорту, телефонного зв'язку, Інтернет-покриття тощо [21]. Серед таких регулювань важливі міжнародно визнані стандарти та відповідні професійні організації.

Література

1. Горбачук В., Дунаєвський М., Сулейманов С.-Б. Підхід до обчислення рівноважних потоків продукції конкурентних фірм через мережі ланцюгів постачання при ресурсних обмеженнях і ризикованих умовах. *Фізико-математичне моделювання та інформаційні технології*. 2023. № 36. С. 58–62.

2. Gorbachuk V. On the transportation subproblem of Euro-2012. *Network Science*. Kyiv: V.M. Glushkov Institute of Cybernetics; Texas A&M University; University of Florida, 2010. P. 6–7.

3. Braess D. Uber ein paradoxon aus der verkehrsplanung. *Unternehmensforschung*. 1968. 12. Z. 258–268.

4. Горбачук В.М., Гаркуша Н.І. Парадокс Браесса й умови його врегулювання. *PDMU-2010* (4–8 жовтня 2010 р., Новий Світ, Україна) Київ: Київський національний університет імені Тараса Шевченка, 2010. С. 52–53.

5. Горбачук В.М., Гаркуша Н.І. Втрата ефективності в рівновагах Неша. *Математичне моделювання в економіці*. 2013. 3. С. 79–89.

6. Easley D., Kleinberg J. *Networks, crowds, and markets: reasoning about a highly connected world*. Cambridge, UK: Cambridge University Press, 2010. 819 p.

7. Semeniuta M.F., Donets G.A. Group labeling of some graphs. *Cybernetics and Systems Analysis*. 2020. 56 (5). P. 701–709.

8. Ермольев Ю.М., Мельник И.М. *Экстремальные задачи на графах*. К.: Наукова думка, 1968. 176 с.

9. Gorbachuk V., Chumakov B. The analytic approach to city traffic network control. *Modelare matematica, optimizare si tehnologii informationale*. Chisinau: Evrica, 2010. P. 94–99.

10. Baker L. Removing roads and traffic lights speeds urban travel. *Scientific American*. 2009, February. P. 20–21.
11. Anshelevich E., Dasgupta A., Kleinberg J.M., Tardos E., Wexler T., Roughgarden T. The price of stability for network design with fair cost allocation. *45-th IEEE Symposium on Foundations of Computer Science* (October 17–19, 2004, Rome, Italy). Piscataway, NJ: IEEE, 2004. P. 295–304.
12. Roughgarden T., Tardos R. How bad is selfish routing? *Journal of the ACM*. 2002. 49 (2). P. 236–259.
13. Кривонос Ю.Г., Кириченко М.Ф., Крак Ю.В., Донченко В.С., Куляс А.І. *Аналіз і синтез ситуацій в системах прийняття рішень*. Київ: Наукова думка, 2009. 365 с.
14. Горбачук В.М. Постіндустріальна організація державних замовлень у розвитку AUTODIN, ARPANET, PRNET, NSFNET та Інтернету. *Вісник Одеського національного університету. Економіка*. 2016. Т. 21. Вип. 8. С. 116–122.
15. Roughgarden T. *Selfish Routing and the Price of Anarchy*. MIT Press, 2019. 240 p.
16. Горбачук В.М., Шулінок Г.О., Морозов О.О. Оцінювання ефективності мереж. *Теорія оптимальних рішень*. 2016. С. 154–161.
17. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. *Структуры данных и алгоритмы*. М.: Вильямс, 2000. 384 с.
18. Crawford V.P. Lying for strategic advantage: Rational and boundedly rational misrepresentation of intentions. *American Economic Review*. 2003. 93 (1). P. 133–149.
19. Mylovanov T. Veto-based delegation. *Journal of Economic Theory*. 138 (1). 2008. P. 297–307.
20. Skyrms B. *The Stag Hunt and Evolution of Social Structure*. Cambridge, UK: Cambridge University Press, 2003. 166 p.
21. Горбачук В.М. Організація функціонування мережних галузей. *Інноваційна економіка*. 2014. № 4 (53). С. 320–328.

ГЕТЕРОГЕННІ ТА ІЄРАРХІЧНІ КОМБІНАТОРНІ КОНФІГУРАЦІЇ

Леонід Гуляницький

Інститут кібернетики імені В. М. Глушкова НАН України

HETEROGENEOUS AND HIERARCHICAL COMBINATORIAL CONFIGURATIONS

Abstract. The concepts of heterogeneous and hierarchical combinatorial configurations are introduced, which are based on the formation of generalized combinatorial configurations associated with abstract basic sets. These concepts significantly expands the possibilities of formalizing practically important combinatorial optimization problems, including those on infinite spaces.

Введене К.Бержем поняття конфігурації [1] дозволило строго означити поширені конструкції з елементів скінченної множини, такі як перестановки, розміщення, комбінації, розбиття та деякі інші об'єкти, утворені на основі урнної схеми Але поза рамками цієї схеми залишилися інші комбінаторні об'єкти, наприклад, цілочислові вектори, гіперграфи, перестановочні матриці та ін. На основі підходу, запропонованого в [2,3], було здійснено розширену формалізацію комбінаторних об'єктів, названих узагальненими комбінаторними конфігураціями [4]. Далі пропонуються нові типи комбінаторних конфігурацій, які виникають у практичних задачах комбінаторної оптимізації.

Нехай спочатку $U = \{1, \dots, m\}$, а V – деяка скінченна лінійно впорядкована множина (ланцюг).

Означення 1. Під *конфігурацією* у сенсі Бержа розуміється відображення $\varphi: U \rightarrow V$, яке задовольняє певній системі обмежень Λ .

Розвитком схеми К. Бержа став підхід до формалізації узагальнених комбінаторних конфігурацій, що дозволило породжувати та класифікувати більш широке коло комбінаторних структур.

Нехай задані скінченна множина $Y = \{1, \dots, m\}$, Z – дискретний, зокрема, скінченний простір (назвемо його твірним), $\varphi: Y \rightarrow Z$ – відображення, що задовольняє деякій системі обмежень.

Означення 2. *Узагальненою комбінаторною конфігурацією* κ називається пара $\kappa = (\varphi, \Omega, \tilde{X})$, де $\varphi: Y \rightarrow \tilde{X}$, \tilde{X} – вибраний базовий простір, а Ω – система обмежень, що накладені на відображення.

Принагідно зауважимо, що величину t доречно назвати розмірністю породжуваної узагальненої комбінаторної конфігурації.

Конкретизацією виду базового простору можна породжувати узагальнені комбінаторні конфігурації різного типу, які класифікуємо так.

Означення 3. Назвемо *узагальненими комбінаторним конфігураціями 1-го порядку* такі комбінаторні конфігурації, у яких базовий простір співпадає з твірним, тобто $\kappa = (\varphi, \Omega, X_{(1)})$, де $X_{(1)} \equiv Z$, а відображення $\varphi: Y \rightarrow Z$ задовольняє обмеженням Ω .

Неважко переконатися, що якщо Z – це скінченний ланцюг, то такі комбінаторні конфігурації співпадають із конфігураціями у сенсі Бержа.

Означення 4. *Узагальненими комбінаторним конфігураціями k -го порядку ($k > 1$)* назвемо комбінаторні конфігурації

$$\kappa = (\varphi, \Omega, X_{(k)}), \text{ де } X_{(k)} = X_{(k-1)} \cup Z^k, \varphi: Y \rightarrow X_{(k)}. \quad (1)$$

У подальшому під комбінаторною конфігурацією будемо розуміти лише узагальнену комбінаторну конфігурацію, якщо не сказано інше.

Нагадаємо, що відображення $\varphi: Y \rightarrow Z$ називається *ін'єктивним*, якщо для $y', y'' \in Y$, таких що $y' \neq y''$, справедливо $\varphi(y') \neq \varphi(y'')$. Ін'єктивне відображення, при якому для кожного елемента $z \in Z$ існує таке $y \in Y$, що $z = \varphi(y)$, називається взаємно однозначним або *бієкцією*. *Строго монотонними* відображеннями називають такі відображення, що $\varphi(i) < \varphi(j)$, коли $i < j$. *Багатозначне відображення* $\varphi: Y \rightarrow Z$ – це таке відображення, яке кожному $y \in Y$ може ставити у відповідність не лише один, а декілька елементів $\varphi(y) \subseteq Z$, причому не виключаються випадки, коли $\varphi(y) = \emptyset$ (див., наприклад, [5]).

Розглянемо окремі класи відображень, що задовольняють конкретним обмеженням Ω і можуть бути використані для породження поширених комбінаторних конфігурацій.

1. Обмеження на відображення відсутні: $\Omega^1 = \emptyset$. Тоді кожному з таких відображень, враховуючи строгу упорядкованість множини Y , відповідає комбінаторна конфігурація, що визначає розміщення обсягу t із елементів Z із повтореннями.

2. Обмеження Ω^2 визначає ін'єктивні відображення, тоді породжуючі комбінаторні конфігурації задають розміщення обсягу t із елементів Z без повторень.

3. Обмеження Ω^3 описує при $m = \|Z\|$ множину бієкцій. Такі відображення формують перестановки m елементів Z .

4. Обмеження Ω^4 виокремлює строго монотонні відображення, а породжені такими відображеннями комбінаторні конфігурації відповідають комбінаціям (сполукам).

5. Обмеження Ω^5 охоплює багатозначні відображення $z_i = \varphi(y_i)$, $z_i \subseteq Z$, $i=1, \dots, m$, які задовольняють умови: $z_i \cap z_j = \emptyset$, $i \neq j$, $\bigcup_{i=1}^m z_i = Z$.

Такі відображення визначають *псевдорозбиття* Z на m *псевдоблоків* (класів), тобто таке розбиття, у якому можуть бути і пусті блоки. Як відомо, власне розбиття визначається додатковою умовою, що $z_i \neq \emptyset$.

Зрозуміло, що цей ряд можна доповнювати і іншим класами обмежень.

Зазначимо, що конфігурації у сенсі Бержа можна назвати у певному сенсі однорідними. Використання ж запропонованої схеми породження узагальнених комбінаторних конфігурацій порядків $k > 1$ може призводити до появи "неоднорідних" базових просторів. Наприклад, при $k=2$ такий простір може складатися як із окремих елементів твірного простору, так і з пар таких елементів (див. формулу (1)), що дозволяє, наприклад, в принципі, описувати графи.

Але для більш адекватного подання у подібних випадках пропонується ввести поняття гетерогенних комбінаторних конфігурацій.

Нехай маємо не один, а $r > 1$ базових просторів, причому деякі із яких можуть складатися із певних комбінаторних конфігурацій.

Означення 5. Гетерогенною комбінаторною конфігурацією рангу $r > 1$ будемо називати множину комбінаторних конфігурацій $(\varphi, \Omega_1, X_1) \cup \dots \cup (\varphi, \Omega_r, X_r)$, де $\varphi: Y_i \rightarrow X_i$, $\Omega_i \in \{\Omega^1, \dots\}$, $Y_i = \{1, \dots, m_i\}$, $m_i > 1$, $i=1, \dots, r$.

Наприклад, при породженні комбінаторних конфігурацій k -го порядку, $k > 1$, (див. означення 4) базовий простір можна подати як об'єднання виду

$$X_{(k)} = X_{(1)} \cup X_{(2)} \cup \dots \cup X_{(k-1)},$$

де $X_{(i)}$ – це базовий простір для комбінаторної конфігурації порядку i , $i=1, \dots, k$.

Тоді, до прикладу, подання гетерогенною комбінаторною конфігурацією гіперграфів із b вершинами, ребра яких включати не більше трьох вершин, буде базуватися на таких параметрах:

$m_1=b$, $Y_1=\{1,\dots,b\}$, $X_1=\{1,\dots,b\}$, $\Omega_1=\Omega^2$, тобто φ_1 – це ін'єктивне відображення, $\varphi_1: Y_1 \rightarrow X_1$;

$m_2=C_b^2$ (кількість комбінацій із b по 2), $Y_2=\{1,\dots,l\}$, l – кількість ребер гіперграфа, які інцидентні рівно двом його вершинам, $X_2=\{1,\dots,m_2\}$ – множина пар із елементів X_1 , $\Omega_2=\Omega^3$ із додатковою умовою, що вибрані пари елементів множини X_2 містять лише вершини, які сформовані відображенням φ_1 , тобто φ_2 – це строго монотонне відображення, яке задовольняє зазначеним обмеженням;

$m_3=C_b^3$ (кількість комбінацій із b по 3), $Y_3=\{1,\dots,h\}$, h – кількість ребер гіперграфа, які інцидентні рівно трьом його вершинам, $X_3=\{1,\dots,m_3\}$ – множина трійок із різних елементів X_1 , $\Omega_2=\Omega^3$ із додатковою умовою, що вибрані трійки елементів множини X_3 містять лише вершини, які сформовані відображенням φ_1 ; отже φ_3 – це також строго монотонне відображення, яке задовольняє зазначеним обмеженням.

Оскільки граф є частинним випадком гіперграфа, то для нього третій варіант буде відсутній ($X_3=\emptyset$). Аналогічно, якщо гіперграф не має ребер, інцидентних рівно двом вершинам, то $X_2=\emptyset$.

Ряд комбінаторних конструкцій, що виникають на практиці, потребують більш диференційованих схем породження.

Нехай маємо послідовність відображень $(\varphi, \Omega_1), \dots, (\varphi, \Omega_r)$.

Означення 6. Назвемо *ієрархічною комбінаторною конфігурацією* рівня s комбінаторну конфігурацію $\kappa^s = (\varphi, \Omega^s, X^s)$, у якій $X^s = \kappa^{s-1}(X^{s-1})$ – базовий простір, утворений на основі базового простору X^{s-1} комбінаторних конфігурацій $\kappa^{s-1}(\varphi, \Omega^{s-1}, X^{s-1})$, а ієрархічна комбінаторна конфігурація першого рівня – це узагальнена комбінаторна конфігурація, яка задається означеннями 1–4.

Таким чином, ієрархічні комбінаторні конфігурації породжуються послідовним застосуванням заданої множини відображень – у той час як при формуванні гетерогенних комбінаторних конфігурацій відображення можуть застосовуватися паралельно, тобто по суті використовується декомпозиція базових просторів.

Розглянемо приклад використання введених типів комбінаторних конфігурацій в одному класі задач планування місій з обстеження чи обслуговування [6].

Розглядається ситуація, коли є гібридна транспортна система: автомобіль чи інший транспортний засіб, що може послідовно

переміщуватися від початкового місця базування (бази) до інших заданих на маршруті баз, маючи на борту один БПЛА. Завдання БПЛА, який може злітати з транспортного засобу і повертатися на нього (чи поруч з ним), – облетіти задану сукупність цільових пунктів (цілей, місць розташування споживачів), враховуючи часові обмеження на польотний ресурс та умови синхронізації з рухом транспортного засобу за заданим маршрутом. Поповнення польотного ресурсу шляхом заміни або дозарядження акумулятора чи заправлення паливом БПЛА (як і його дозавантаження за потреби) у процесі виконання місії відбувається тільки у зазначених базах на маршруті транспорту.

Отже, за потреби у поповненні польотного ресурсу БПЛА повинен повернутися на певну базу на маршруті транспортного засобу. У такому разі відповідний фрагмент маршруту (підмаршрут) БПЛА завершується цією базою, враховується час обслуговування БПЛА для відновлення ресурсу (та дозавантаження за потреби) і продовжується процес формування наступного підмаршруту чи переміщення транспортного засобу до наступної бази.

Планування місії полягає у такому визначенні підмаршрутів загального маршруту БПЛА із усіх чи частини баз на маршруті та часових інтервалів його обслуговування на цих базах, щоб провести обстеження цілей за мінімальний час.

Аналіз постановки задачі показує, що розв'язок формується за три етапи: розбиття множини цілей Z на псевдоблоки, упорядкування цих псевдоблоків і, нарешті, упорядкування (перестановка) елементів у кожному із цих псевдоблоків.

Схему формування комбінаторних конфігурацій, які відображають розв'язок задачі, можна описати так.

На першому кроці формуються узагальнені комбінаторні конфігурації виду: $\kappa^1 = (\varphi, \Omega^5, X^1)$, де $Y = \{1, \dots, m\}$, m – кількість баз на маршруті транспортного засобу, $X^1 = Z$. Вони і подають розбиття множини Z на псевдоблоки.

На другому виникають ієрархічні комбінаторні конфігурації $\kappa^2 = (\varphi, \Omega^3, \kappa^1(X^1))$, де $\kappa^1(X^1)$ – множина m номерів псевдоблоків, утворених на першому кроці.

Нарешті, на завершальному кроці виникають гетерогенні комбінаторні конфігурації виду $\kappa_i = (\varphi, \Omega^3, X_i^3)$, де X_i^3 – множина номерів цілей, які ввійшли в i -й псевдоблок, $i=1, \dots, m$.

Введені поняття гетерогенних комбінаторних та ієрархічних комбінаторних конфігурацій пов'язані, в загальному випадку, із

абстрактними базовими просторами, що дозволяє використовувати наведені вище результати при формалізації широких класів практичних задач, включивши до цього класу і деякі оптимізаційні задачі, які визначені на нескінченних просторах.

Важливим напрямом подальших досліджень може стати розроблення підходів до формалізації нечітких комбінаторних конфігурацій, як це зроблено щодо узагальнених комбінаторних конфігурацій [3]. Цікавим також є розвиток цих досліджень у напрямі Евклідових комбінаторних конфігурацій [7].

Література

1. Berge C. (1971). Principles of combinatorics. N.Y.: Academic Press. 176 p.

2. Гуляницький Л.Ф. (2008). До формалізації та класифікації задач комбінаторної оптимізації. *Теорія оптимальних рішень, № 7*: 45–49.

3. Hulianytskyi L.F., Riasna I.I. (2017). Formalization and classification of combinatorial optimization problems. *Optimization Methods and Applications (eds. Butenko S., Pardalos P. M., Shylo V.)*. Cham: Springer International Publishing AG: 239–250. https://doi.org/10.1007/978-3-319-68640-0_11

4. Гуляницький Л.Ф. (2021). Узагальнені комбінаторні конфігурації: теорія та застосування. *Мат. XXVI Міжн. наук. конф. "Сучасні проблеми прикладної математики та комп'ютерних наук АРАМС-2021" (Львів, 27–28 вересня 2021 р.)*. Львів, Львівський національний університет ім.І.Франка: 91–94.

5. Sachkov V.N. (1996). Combinatorial methods in discrete mathematics. Cambridge University Press. 306 p.

6. Horbulin, V.P., Hulianytskyi, L.F. & Sergienko, I.V. (2023). Planning of Logistics Missions of the “UAV+Vehicle” Hybrid Systems. *Cybernetics and Systems Analysis*, **59**, 733–742. <https://doi.org/10.1007/s10559-023-00609-8>

7. Yakovlev S., Pichugina O., Koliechkina L. (2023). Combinatorial point configurations and polytopes. Lodz: Wydawnictwo Uniwersytetu Lodzkiego. 231 p.

AN APPLICATION OF THE PROBLEM OF A MATHEMATICAL SAFE IN THE FIELDS OF REDUNDANCIES TO THE CONSTRUCTION OF A SYMMETRIC ENCRYPTION SYSTEM BASED ON THE VIGENERE METHOD

Artem Gurin

*National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"*

Abstract. The task of practical application of the mathematical safe on matrices problem in message encryption is considered. For the cryptosystem, the Vigenere symmetric encryption method was chosen, where the input data is stored in the numerical order of the storage elements state vector on graphs, and the method for solving the mathematical safe problem is parametric.

The practical value and prospects of applying the mathematical safe problem are in solving the problem of creating fast and tamper-resistant cryptosystems.

There are a number of cryptosystems designed for a wide range of users to encrypt messages in fast messengers for active user communication. Many of them are affordable and convenient for the average user [1].

It is well known that these cryptosystems use symmetric, asymmetric, and hybrid (asymmetric + symmetric) types of encryption [2].

In symmetric encryption (standard algorithms RC4, AES, DES, 3DES and QUAD, Vigenere), one key is used to encrypt and decrypt data. The asymmetric type of encryption (standard algorithms: RSA, Diffie-Hellman, ECC, El Gamal and DSA) uses a pair of keys. These keys are known as the "public key" and the "private key" and are the encryption and decryption keys [3].

The practical significance of using the algorithms of the mathematical safe problem in the field of information security and cryptography was first formulated in detail for information exchange systems using symmetric encryption in the works of S. L. Krivoy [4,5], where the TSS method was proposed to solve the encryption problem. On its basis, a system of commutative rings with one unit is considered as a means of building a symmetric encryption system, and an information exchange tool based on isotropic rings is proposed and described, and a table hiding encryption algorithm is proposed as a tool for exchanging information in messengers where it is necessary to ensure

a rather mediocre resistance to hacking with a high speed of decoding and encrypting information.

From the security point of view, asymmetric encryption, which provides authentication, is probably better. However, symmetric encryption algorithms are always needed because performance is an aspect that cannot be ignored.

In work [6], in general, the k -a associative commutative ring of the k -o order is constructed in one line using the algorithms for constructing the ring operation tables $ADD-TAB-AKK1(I, k)$ and $MUL-TAB-AKK1(I, k)$. The time complexity of the first algorithm is $O(k^2 \log k)$, and the second is $O(k \log k)^2$, where k is the order of the ring. The paper also proposes to modify the RG-EN encryption and RG-DE decryption algorithms.

Consider the Vigenère encryption method, where the input data will be stored in the numerical order of the elements of the vector of storage state on graphs, and the method for solving the mathematical safe problem is parametric [7]. The Vigenere problem is quite common in practice. It is known that the performance parameters of a cryptographic system are ensured by the fact that, unlike multi-alphabet substitution or table “hamming”, the key of the Vigenère encryption algorithm has a finite length. Therefore, there is no need to encrypt each letter of the plaintext with a separate key value.

The Vigenere encryption system is defined as follows, where VIG :

$$(x_0, x_1, \dots, x_{n-1}) \rightarrow (y_0, y_1, \dots, y_{n-1}) = [(x_0 + k_0) \bmod N, (x_1 + k_1) \bmod N, \dots, (x_{n-1} + k_{n-1}) \bmod N],$$

where N is the number of letters in the alphabet, and n is the length of the Vigenere key with components k_i ($i = 0, 1, \dots, n$). As, a cyclic sequence of keys of the same length is superimposed on the original text X , and the corresponding elements of the key sequence are added modulo N , resulting in the original ciphertext Y .

For the first time, the problem of a mathematical safe on graphs was considered in the work of Donets G.P. [8], where a direct method for solving graphs of certain types of topologies was formulated.

Consider the problem of a mathematical safe on graphs modulo $K \geq 2$ a prime number using the method of summation and parametric method.

To encrypt the text, we will use a substitution G of the form:

$$(s_1, s_2, \dots, s_N) \rightarrow (y_1, y_2, \dots, y_N) = [(s_1 - x_1) \bmod K, \dots, (s_2 - x_2) \bmod K, \dots, (s_N - x_N) \bmod K]$$

according to which we will subtract the set of plaintexts $S = [s_1, s_2, \dots, s_N]$

along the entire length of the solution to the mathematical safe problem $X = (x_1, x_2, \dots, x_N)$ with the initial state vector $B = (b_1, b_2, \dots, b_N)$.

The plaintexts will be stored in the numerical order of the elements of the vault state vector on graphs with “ladder” topologies (Fig. 1). $K = 7$, and $b = (1, 2, 3, 4, 5, 6, 7, 8)$.

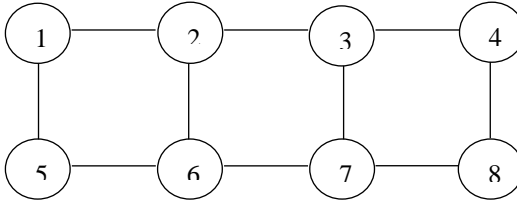


Fig. 1. The problem of a mathematical safe in the graph “ladder”

Then solve the problem by the method of summation. Write systems (1) and (2) for a given graph:

$$\left. \begin{array}{l}
 x_1 + x_2 + \dots + x_5 \dots \equiv -1 \\
 x_1 + x_2 + x_3 + \dots + x_6 \dots \equiv -2 \\
 \dots + x_2 + x_3 + x_4 + \dots + x_7 \equiv -3 \\
 \dots + x_3 + x_4 + \dots + x_8 \equiv -4 \\
 x_1 + \dots + x_5 + x_6 \dots \equiv -5 \\
 \dots + x_2 + \dots + x_5 + x_6 + x_7 \dots \equiv -6 \\
 \dots + x_3 + \dots + x_6 + x_7 + x_8 \equiv -7 \\
 \dots + x_4 + \dots + x_7 + x_8 \equiv -8
 \end{array} \right\} \pmod{K}. \quad (1)$$

$$\left. \begin{array}{l}
 d_1 + d_2 + \dots + d_5 \dots \equiv d \\
 d_1 + d_2 + d_3 + \dots + d_6 \dots \equiv d \\
 \dots + d_2 + d_3 + d_4 + \dots + d_7 \equiv d \\
 \dots + d_3 + d_4 + \dots + d_8 \equiv d \\
 d_1 + \dots + d_5 + d_6 \dots \equiv d \\
 \dots + d_2 + \dots + d_5 + d_6 + d_7 \dots \equiv d \\
 \dots + d_3 + \dots + d_6 + d_7 + d_8 \equiv d \\
 \dots + d_4 + \dots + d_7 + d_8 \equiv d
 \end{array} \right\} \pmod{K}. \quad (2)$$

Comparing the fourth and eighth equations in system (2), we obtain the equality $d_3=d_7$. By analogy, from the first and fifth equations follows $d_2=d_6$ from the third and seventh equations $d_4=d_8$ from the second and sixth equations $d_1=d_5$.

So, we obtain $d_1=d_4=d_5=d_8$, a $d_2=d_3=d_6=d_7$.

Comparing to the first and second equations from (2), we obtain $d_5=2d_2$. Let us assume that $d_2=1$. Then $d_3=d_6=d_7=1$, $d_1=d_4=d_5=d_8=2$. That is $d=5$. In consequence,

$$5S = -\sum_{i=1}^n d_i b_i = -54 = 2(\text{mod } 7), \text{ a } S = 6(\text{mod } 7).$$

Now find $x_i, i=1,2,\dots,n$, from system (1), using the representation of variables through the parameter. Adding the fourth and sixth equations, we get $S - x_1 = -10$. So $x_1 = S + 10 = 16 = 2(\text{mod } 7)$. Similarly, from the second and eighth equations we get $x_5 = 2(\text{mod } 7)$, from the first and seventh get $x_4 = 0(\text{mod } 7)$, from the third and fifth $x_8 = 0(\text{mod } 7)$. Substituting the obtained values into the equation of system (1), we obtain $x_2 = 2(\text{mod } 7)$, $x_3 = 0(\text{mod } 7)$, $x_6 = -2(\text{mod } 7)$, $x_7 = -1(\text{mod } 7)$. As a result, we obtain the general solution $X = (2, 2, 3, 0, 2, -2, -1, 0) (\text{mod } 7)$.

We need to encrypt the following set of characters: $S = (0, 6, 5, 4, 2, 1, 0, 1)$.

Since, $S = (s_1, s_2, \dots, s_N) - X \rightarrow Y = (y_1, y_2, \dots, y_N)$ then $\rightarrow Y = (-2, 4, 2, 4, 0, 3, 1, 1)$.

Consider an example for a “fan” graph (Fig. 2) for $n = 7, k = 3, K = 5, b_i = i$.

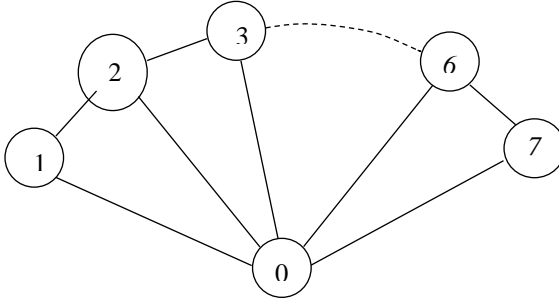


Fig. 2. The problem of a mathematical safe of the “fan” type

Then solve the problem by the parametric method. The system of equations will take the form:

$$\left. \begin{array}{rcl}
 x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 & \equiv & 0 \\
 x_0 + x_1 + x_2 & \equiv & -1 \\
 x_0 + x_1 + x_2 + x_3 & \equiv & -2 \\
 x_0 + x_2 + x_3 + x_4 & \equiv & -3 \\
 x_0 + x_3 + x_4 + x_5 & \equiv & -4 \\
 x_0 + x_4 + x_5 + x_6 & \equiv & -5 \\
 x_0 + x_5 + x_6 + x_7 & \equiv & -6 \\
 x_0 + x_6 + x_7 & \equiv & -7
 \end{array} \right\} \pmod{5}$$

Two parameters are enough for this graph. Let us denote by $x_0 = \alpha$, $x_1 = \beta$. From the first equation $x_2 = -1 - \alpha - \beta$. Then $x_3 = -2 - \alpha - \beta - (-1 - \alpha - \beta) = -1$.

Now:

$$x_4 = -3 - x_0 - x_2 - x_3 = -3 - \alpha - (-1 - \alpha - \beta) - (-1) = -1 + \beta,$$

$$x_5 = -4 - x_0 - x_3 - x_4 = -4 - \alpha - (-1) - (-1 + \beta) = -2 - \alpha - \beta,$$

$$x_6 = -5 - x_0 - x_4 - x_5 = -5 - \alpha - (-1 + \beta) - (-2 - \alpha - \beta) = -2,$$

$$x_7 = -6 - x_0 - x_5 - x_6 = -6 - \alpha - (-2 - \alpha - \beta) - (-2) = -2 + \beta.$$

Substituting these values into the 0th equation, we get:

$$\alpha + \beta + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 0 \Rightarrow$$

$$\alpha + \beta + (-1 - \alpha - \beta) + (-1) + (-1 + \beta) + (-2 - \alpha - \beta) + (-2) + (-2 + \beta) = 0 \Rightarrow$$

$$-\alpha + \beta = 9$$

Subtracting the 7th equation from 6th, we obtain
 $x_5 = +1 = -2 - \alpha - \beta$. From here we find:
 $\alpha = -1 \pmod{5}$, $\beta = 3 \pmod{5}$.

As a result, we get the solution $X = (-1, 3, 2, -1, 2, 1, -2, 1)$.

We need to encrypt the following set of characters:
 $S = (2, 3, 1, 0, 2, 2, 0, 3)$.

Since, $S = (s_1, s_2, \dots, s_N) - X \rightarrow Y = (y_1, y_2, \dots, y_N)$ then
 $\rightarrow Y = (3, 0, -1, 1, 0, 1, 2, 2)$.

In this case, in order to recover the matrix B , you need to go through the K^{mn} variants.

It should be noted that the proposed encryption scheme is focused on information that loses its value within a short time after transmission. Generally, in practice, this corresponds to the tasks of encrypting messages by mobile phones with small decryption software devices or the operation of inexpensive messengers to encrypt the exchange of ordinary messages.

Literature

1. Kuznetsov O.O., Evseev S.P., Korol O.G. Information security in information systems. Kh.: KNEU Publishing House, 2011. 512 c.
2. Khoroshko V.A., Chekatkov A.A. Methods and means of information protection. K.: Junior, 2003. 504 c.
3. Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996. 816 p.
4. Kryvyi S.L. Cryptosystem based on abelian groups and rings // Problems of programming. - 2020. - . 2-3. - P. 270-277.
5. Kryvyi S.L. Application of commutative rings with unit for the construction of a symmetric encryption system // Cybernetics and System Analysis. - 2022. - No. 3. - P. 3-16.
6. Kryvyi S.L., Opanasenko V.M., Hrynenko O.O., Northman Y.O. Symmetric information exchange system based on isotropic rings // Cybernetics and System Analysis. - 2023. - №. 3. - P. 3-15.
7. Gurin A.L., Grashchenko I.S., Savchenko L.M. Parametric method for solving problems of a mathematical safe on graphs // Problems of Control and Informatics. - 2021. - №. 2. - P. 5-10.
8. Donets G.P., Zhang Bin. Problems of a mathematical safe on graphs // Cybernetics and System Analysis. - 2006. - №. 5. - P. 84-93.

АЛГОРИТМ НЕЧІТКОГО ВИВОДУ ДЛЯ ВИЗНАЧЕННЯ РІВНЯ ЕКОНОМІЧНОГО ДОБРОБУТУ НАСЕЛЕННЯ

Доровці Адам Федорович, Маляр Микола Миколайович,

Шаркаді Маріанна Миколаївна

ДВНЗ «Ужгородський національний університет»

FUZZY INFERENCE ALGORITHM FOR DETERMINING THE LEVEL OF ECONOMIC WELL-BEING OF THE POPULATION

Abstract. In this work, we propose an algorithm for determining the economic well-being of the population. Economic well-being is decomposed into three parts: income, housing, and work. Based on the available statistical data, we have built the corresponding fuzzy inference system for each of the three components. Based on the results obtained, we have constructed a new fuzzy inference system that gives the level of economic well-being. In our research, we applied the Mamdani inference algorithm. We have built a rule base that describes the operation of the system, which consists of rules of the type "IF - THEN". After defuzzification, for which we used Centroid or Center of Gravity methods, we obtain the level of economic well-being of the population.

Однією зі складових добробуту населення є економічний добробут. У даній роботі пропонується алгоритм оцінки рівня економічного добробуту. У нещодавно опублікованій роботі [1] було перераховано ряд індикаторів, які можуть бути використані для оцінки, наприклад, економічного добробуту. Ці показники також використовуються ОЕСР у власному так званому індексі кращого життя [2]. Пропонуються три системи нечіткого виведення для перевірки економічного добробуту. Перша – дохід, де використовуємо вхідні дані: чисте багатство на одну особу та чистий наявний дохід. Друга система нечіткого виводу - це житло. Тут розглядаються три вхідні дані: «м²» на особу, витрати на житло (у відсотках) та житло з базовими зручностями. Третя - це робота з двома вхідними даними: рівень зайнятості та довготривале безробіття.

Для того, щоб отримати рівень економічного добробуту з цих систем нечіткого виводу, було використано властивість, доступну в програмному пакеті Matlab, яка називається FIS-дерево. FIS-дерево - це ієрархічна система систем нечіткого виводу. У структурі FIS-дерева виходи нечітких систем низького рівня служать входами для нечітких систем високого рівня. FIS-дерево є обчислювально ефективнішим і простішим для розуміння, ніж окрема система

нечіткого виводу з такою ж кількістю входів [3]. Побудоване нами FIS-дерево показано на рисунку 1.

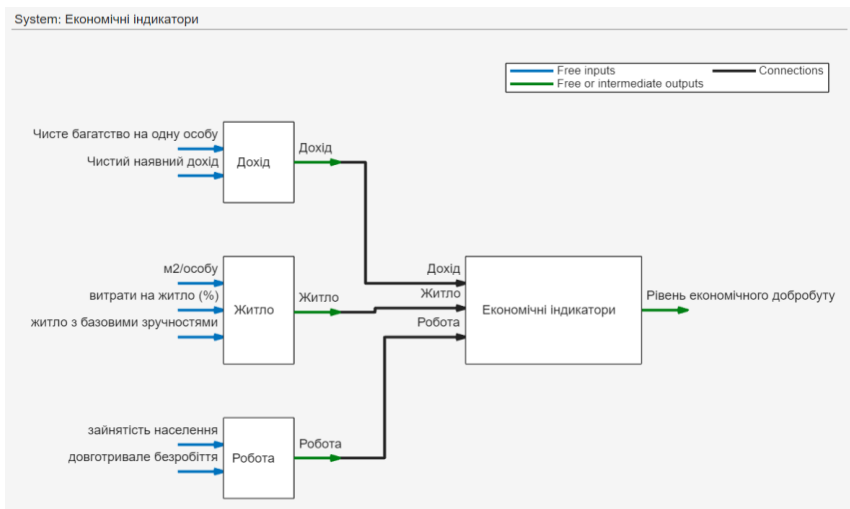


Рис.1 – FIS- дерево для індикаторів економічного добробуту.

Після побудови FIS-дерева запропоновано лінгвістичні змінні для кожної системи нечіткого виводу «низький», «середній» та «високий». Звичайно, цим лінгвістичним змінним були визначені різні значення для кожного типу входних даних. На рисунку 2 показано приклад використання лінгвістичних змінних на одному з входних індикаторів системи нечіткого виведення "Дохід".

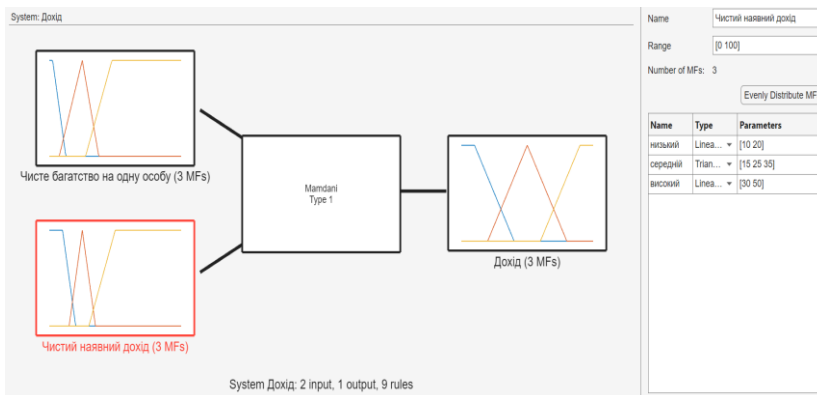


Рис.2 – Приклад використання лінгвістичних змінних.

Після присвоєння відповідних лінгвістичних змінних кожній вхідній та вихідній змінній, побудовано базу знань для кожної системи нечіткого виводу. В базі знань зберігається інформація про залежність $Y = f(X)$ у вигляді нечітких правил типу “якщо - то”. Така база знань показана на рисунку 3.

System: Дохід

Add All Possible Rules Clear All Rules

	Rule	Weight	Name
1	If Чисте багатство на одну особу is низький and Чистий наявний дохід is низький then Дохід is низький	1	rule1
2	If Чисте багатство на одну особу is середній and Чистий наявний дохід is низький then Дохід is середній	1	rule2
3	If Чисте багатство на одну особу is високий and Чистий наявний дохід is низький then Дохід is середній	1	rule3
4	If Чисте багатство на одну особу is низький and Чистий наявний дохід is середній then Дохід is середній	1	rule4
5	If Чисте багатство на одну особу is середній and Чистий наявний дохід is середній then Дохід is середній	1	rule5
6	If Чисте багатство на одну особу is високий and Чистий наявний дохід is середній then Дохід is високий	1	rule6
7	If Чисте багатство на одну особу is низький and Чистий наявний дохід is високий then Дохід is середній	1	rule7
8	If Чисте багатство на одну особу is середній and Чистий наявний дохід is високий then Дохід is високий	1	rule8
9	If Чисте багатство на одну особу is високий and Чистий наявний дохід is високий then Дохід is високий	1	rule9

Рис.3. – База знань для системи нечіткого виводу «Дохід».

У даному випадку для дефазифікації було використано метод центроїда площі, або центру тяжіння.

Даний підхід було застосовано до інших систем нечіткого виводу. Після отримання значень кожної системи нечіткого виводу використовували їх як вхідні дані для наступної системи нечіткого виводу (див. рис. 1.) для отримання рівня економічного добробуту суспільства.

Список використаних джерел:

1. Маляр М.М. Аналіз індикаторів впливу на рівень добробуту населення України / М.М. Маляр, А.Ф. Доровці, І.І. Половко // Наукові перспективи, випуск № 2(44). – Київ 2024. С. 784-796. [https://doi.org/10.52058/2708-7530-2024-2\(44\)-784-796](https://doi.org/10.52058/2708-7530-2024-2(44)-784-796)

2. Офіційний сайт: Організація економічного співробітництва та розвитку [Електронний ресурс] Режим Доступу: <https://www.oecdbetterlifeindex.org/#/50525555555>

3. Офіційний сайт: Matlab Documentation. [Електронний ресурс] Режим Доступу: <https://www.mathworks.com/help/fuzzy/fuzzy-trees.html>

ПОКРАЩЕННЯ ЕВРИСТИЧНОГО АЛГОРИТМУ ПАКУВАННЯ НЕРІВНИХ КРУГІВ ІЗ ЗАСТОСУВАННЯМ r - АЛГОРИТМУ ШОРА¹

¹Задорожний Б.О., ^{2,3}Романова Т.Є., ^{1,4}Стецюк П.І.

bohzador@gmail.com

¹Ужгородський національний університет

²Інститут проблем машинобудування імені А.М. Підгорного
НАН України

³Leeds University Business School, University of Leeds, UK

⁴Інститут кібернетики імені В.М. Глушкова НАН України

IMPROVED HEURISTIC FOR PACKING UNEQUAL CIRCLES USING SHOR'S r -ALGORITHM

Abstract. The paper studies a heuristic algorithm for solving the competitive problem "Dense packing of circles into a circle of minimal radius". Two approaches using Shor's r -algorithm with a dichotomy step are proposed to improve the heuristic. The obtained results are analyzed with respect to two r -algorithm parameters: the space dilation coefficient and the step multiplier reduction coefficient. The Rust 1.70.0 programming language is used for software implementation.

1. Вступ. У період з 1 листопада по 21 грудня 2022 року проводився міжнародний конкурс «Щільна упаковка кругів в круг мінімального радіусу» [1]. Організатори конкурсу – Інститут кібернетики імені В.М. Глушкова НАН України (Київ, Україна), Ужгородський національний університет (Ужгород, Україна), Інститут проблем машинобудування імені А.М. Підгорного НАН України (Харків, Україна) та Дрезденський технічний університет (Дрезден, Німеччина).

Конкурсною була така оптимізаційна задача: задано набір з N кругів з радіусами r_i , $i = 1, 2, \dots, N$, та зовнішній круг з центром в

¹ Робота підтримана Volkswagen Foundation (грант № 97775)

точці $(0,0)$ та змінним радіусом R . Потрібно знайти такі центри кругів $(x_i; y_i)$, $i = 1, 2, \dots, N$ та відповідний радіус R^* зовнішнього круга, що:

- 1) кожен з кругів $i = 1, 2, \dots, N$ має повністю знаходитись всередині зовнішнього круга (круги можуть торкатися межі зовнішнього круга);
- 2) для будь-якої пари $(i; j)$ кругів, де $i, j \in \{1, \dots, N\}$ та $i < j$, круги i, j не перетинаються (їм дозволено торкатися один одного);
- 3) радіус R зовнішнього круга має бути якомога меншим.

Друге місце в змаганні зайняв перший автор цієї статті за критерієм оцінки значення радіусу зовнішнього круга для кожного із конкурсних тестових прикладів, які знаходив запропонований евристичний алгоритм із застосуванням відповідного програмного забезпечення. Однак, розміщення кругів відповідно до значення цільової функції, у більшості прикладів містили «надлишковий простір». Тому предметом цієї статті є розробка підходу для покращення зазначеної евристики із застосуванням методів негладкої оптимізації.

Матеріал статті викладено у такому порядку. У розділі 2 описано евристичний алгоритм та його властивості. У розділі 3 наведено два підходи до покращення розв'язків, знайдених за допомогою евристичного алгоритму, із застосуванням r -алгоритму Шора з дихотомією кроку. У розділі 4 наведено результати обчислювальних експериментів за точністю знаходження радіуса зовнішнього круга.

2. Евристичний алгоритм. Розглянемо опис алгоритму розв'язання задачі пакування кругів в круг мінімального радіусу.

Етап 1. Встановлюємо допустимі межі радіусу зовнішнього круга: ліва межа дорівнює $R_{low} = \max_{i=1, \dots, N} r_i$, а права – $R_{up} = \sum_{i=1}^N r_i$.

Етап 2. Визначаємо стартове значення радіусу зовнішнього круга як $R^0 = 0.5(R_{low} + R_{up})$.

Етап 3. Пакуємо круги за такими правилами:

1. Встановлюємо центр першого круга з набору $i = 1, 2, \dots, N$ в точку $(0, R^0 - r_1)$.
2. Генеруємо шар кругів. За годинниковою стрілкою розміщуємо кожний наступний круг так, щоб він торкався попереднього круга та межі зовнішнього круга. Використовуючи координати

центра попереднього круга, обчислюємо координати центра наступного круга: встановлюємо центр наступного i -го круга в точці $(0, R^0 - r_i)$; розраховуємо кут, на який радіус-вектор потрібно повернути за годинниковою стрілкою. Отриманий кут уточнюємо за допомогою бінарного пошуку (Рис. 1а).

3. Розміщуємо наступні круги між межею зовнішнього круга та першим шаром кругів (Рис. 1б).
4. Розміщуємо круги між кругами попереднього шару або між кругами через один (Рис. 1в).

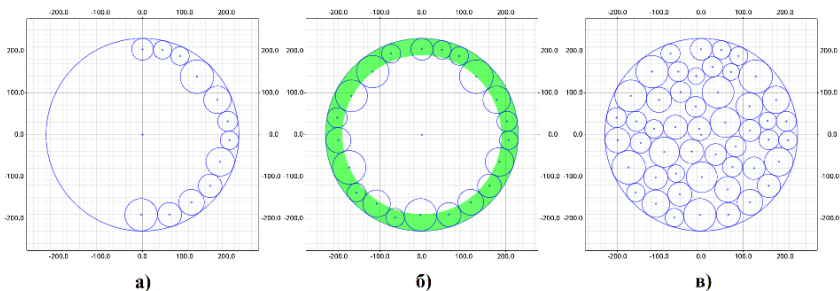


Рис. 1 – *Етап 3 евристичного алгоритму*

Етап 4. Якщо розміщення кругів та радіус зовнішнього круга задовольняють вимогам 1) та 2), то встановлюємо $R_{up} := R^0$ та запам'ятовуємо центри розміщених кругів. Інакше, встановлюємо $R_{low} := R^0$.

Етап 5. Якщо $R_{up} - R_{low} > \Delta$, то переходимо до етапу 2, де Δ – точність знаходження розв'язку.

Етап 6. Порівнюємо розв'язок отриманий на етапі 4 з найкращим знайденим. Якщо розміщення кругів задовольняє вимогам 1) та 2) задачі та має менший радіус зовнішнього круга за найкращий розв'язок, то останній встановлюємо рівним знайденому. Якщо досягнена максимальна кількість ітерацій, то алгоритм закінчує роботу. Інакше застосуємо “swar”-стратегію для розміщення пари випадкових розміщених кругів та переходимо до етапу 1.

Алгоритм не завжди може гарантувати розміщення кругів, задовольнивши обмеження 1) і 2), тому потрібно обрати кількість ітерацій при якій вдасться досягнути хоча б один допустимий розв'язок.

Варто зауважити, що оскільки алгоритм базується на певному

наборі правил розміщення кругів, то при малій кількості ітерацій залишається “вільний простір”, який можна зменшити, якщо пакувати круги щільніше.

3. Покращення евристичного розв’язку за допомогою r -алгоритму. Використовуючи точку, знайдену евристичним алгоритмом як стартову, та розв’язуючи задачу оптимізації за допомогою r -алгоритму з дихотомією кроку [2], можна отримати кращі розв’язки за значенням радіусу зовнішнього круга.

Для покращення результатів, отриманих евристичним алгоритмом, будемо використовувати r -алгоритм Шора [3].

Сформулюємо задачу пакування кругів так:

$$R^* = \min_{R,x,y} R, \quad (1)$$

за обмежень

$$x_i^2 + y_i^2 \leq (R - r_i)^2, i = 1, \dots, N, \quad (2)$$

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i + r_j)^2, 1 \leq i < j \leq N, \quad (3)$$

$$R \geq R_{low}, \quad (4)$$

де $R_{low} = \max_{i=1,\dots,N} r_i$, $x = (x_1, \dots, x_N)$, $y = (y_1, \dots, y_N)$.

Для пошуку локального мінімуму задачі (1) – (4) використаємо метод негладких штрафних функцій, який зводить задачу нелінійного програмування (1) – (4) до задачі безумовної мінімізації негладкої функції [3].

Розглянемо два алгоритми покращення евристичного розв’язку [4], де алгоритм Шора застосується з наперед заданими параметрами за винятком стартової точки X в n -вимірному просторі та величини кроку h в напрямку локального екстремуму.

Алгоритм 1 має такий вигляд.

Етап 1. Задаємо $h = h_0$, де h_0 – початкова величина кроку, встановлюємо значення для h_{min} – мінімальне значення кроку та EPS – відсоток відносного покращення величини радіуса головного круга.

Етап 2. Використаємо r -алгоритм зі стартової точки X і знаходимо локальний мінімум задачі (1) – (4).

Етап 3. Якщо відносне покращення радіусу зовнішнього круга:

- більше за значення EPS , то точка локального мінімуму стає стартовою точкою;
- менше за значення EPS , то величину кроку h зменшуємо вдвічі.

Етап 4. Якщо величина кроку більша ніж мінімальне значення кроку, то переходимо до етапу 2. Інакше закінчуємо роботу алгоритму з покращенням розв'язку.

Алгоритм 2 відрізняється від алгоритму 1 формулюванням етапу 3: якщо відносно покращення радіусу зовнішнього круга більше ніж значення EPS , то точка локального мінімуму стає стартовою точкою і $h = h_0$, якщо менше ніж значення EPS , то величину кроку h зменшуємо вдвічі.

4. Результати обчислювальних експериментів. Алгоритми 1 та 2 реалізовано мовою програмування Rust версії 1.70.0. У таблиці 1 наведено порівняльні результати обох реалізацій при $EPS = \{0; 10^{-5}; 10^{-4}\}$. Тут n_iter – середнє арифметичне кількості ітерацій r -алгоритму, що були в середньому використані для кожного тесту задачі, n – кількість тестів.

Таблиця 1

Порівняння роботи двох алгоритмів уточнення розв'язку, знайденого евристичним алгоритмом (4832,04 балів)

Варіант	EPS	Кількість балів	n_iter	Покращення в балах
1	10^{-4}	4 905,47	11 543,00	73,43
2	10^{-4}	4 908,10	12 006,31	76,07
1	10^{-5}	4 906,02	11 759,53	73,98
2	10^{-5}	4 907,86	12 199,83	75,83
1	0	4 906,75	11 225,86	74,71
2	0	4 912,64	11 923,91	80,6

Наведені результати свідчать про ефективність роботи обох алгоритмів, причому другий показує більші покращення в балах. Зменшення EPS призводить до суттєвих покращень роботи обох алгоритмів.

Ефективність обох алгоритмів залежить від параметрів r -алгоритму: α – коефіцієнту розтягу простору та q_1 – коефіцієнта зменшення крокового множника. Отримані покращення евристичних результатів при $EPS = 0$ для алгоритму 2 наведено на рисунку 1. При розрахунках використовувались такі параметри зупинки: $\varepsilon_x \approx 10^{-6}$, $\varepsilon_g \approx 10^{-7}$.

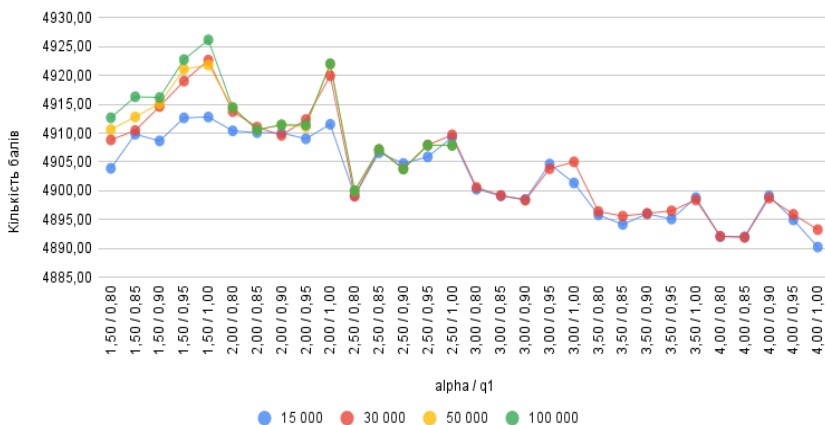


Рис. 2 – Отримана кількість балів для алгоритму 2 при $EPS = 0$ залежно від максимальної кількості ітерацій r -алгоритму

З рисунка 2 бачимо, що при збільшенні α кількість балів зменшується. Найкращі результати досягаються при $\alpha = 1,5$ та $q_1 = 0,95 - 1$.

На рисунку 3 наведено роботу евристичного алгоритму при великій кількості ітерацій та роботу алгоритмів 1 та 2 для $\alpha = 1,5$ і $q_1 = 0,95$.

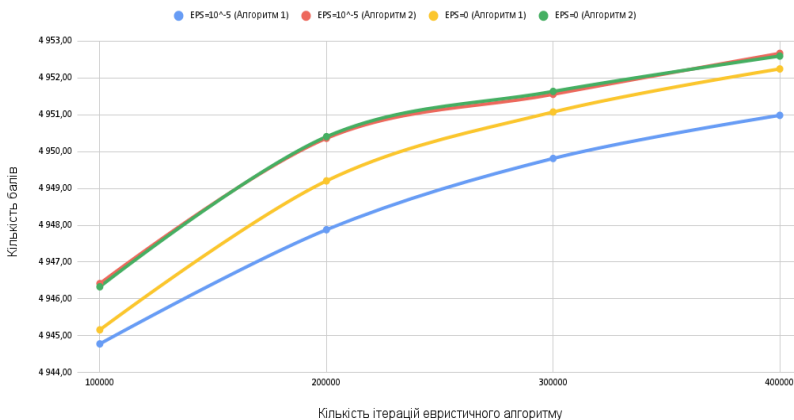


Рис. 3 – Порівняння евристичного алгоритму з алгоритмами 1 та 2

З рисунка 3 бачимо, що алгоритм 1, у порівнянні з алгоритмом 2, дає гірші результати, незалежно від значення EPS . Алгоритм 2 при різних EPS дає приблизно однакові результати.

Для $\alpha = 1,5$ та $q_1 = 1$ спостерігаються схожі результати. При використанні алгоритму 2 при збільшенні ітерацій від 100000 ця різниця майже відсутня.

Висновки. Використання евристичного алгоритму у сполученні з r -алгоритмом для уточнення евристичних розв'язків є ефективним. Значення коефіцієнта розтягу простору $\alpha = 1,5$ та коефіцієнта зменшення крокового множника $q_1 = 0,95 \div 1,0$ зарекомендували себе як найкращі параметри r -алгоритму.

Для евристичного алгоритму при кількості ітерацій 100000 та 400000 кращий результат (на декілька десятих бала) отримується при використанні $EPS = 10^{-5}$, причому при 400000 ітерацій – це найкращий результат загалом. Це означає, що в деяких випадках краще використовувати $EPS = 10^{-5}$.

Обираючи відповідні значення EPS та параметрів r -алгоритму можна отримати бажане покращення відповідно до часових обмежень на роботу програми.

Література

1. Щільна упаковка кругів в круг мінімального радіусу. Eolymp. URL: <https://packing-circles.eolymp.io> (дата звернення: 23.05.2024).
2. Задорожний Б.О., Міца О.В., Стецюк П.І. Про покращення евристичного алгоритму упаковки кругів в круг мінімального радіусу. Cybernetics and Computer Technologies. 2023. № 2. С. 32–45.
URL: <https://doi.org/10.34229/2707-451X.23.2.4> (дата звернення: 23.05.2024).
3. Субградієнтні алгоритми та задачі на комбінаторних конфігураціях. / П. І. Стецюк та ін. Київ: Університетське видавництво ПУЛЬСАРИ, 2019. 235 с.
4. Задорожний Б.О., Стецюк П.І. Покращення локального екстремуму для задачі упаковки кругів в круг мінімального радіусу. XXI міжнародна науково-практична конференція “Математичне та програмне забезпечення інтелектуальних систем (MPZIS-2023)”: тези доп., 22-24 лист. 2023 р., Дніпро, Дніпро: ДНУ, 2023. С. 128–129.

МЕТОД ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОЇ СТІЙКОСТІ ІНТЕГРОВАНОГО КОМПЛЕКСУ БОРТОВОГО ОБЛАДНАННЯ В ДЕСТАБІЛІЗУЮЧИХ УМОВАХ

Калашник Г.А., Калашник-Рибалко М.А.

Льотна академія Національного авіаційного університету

THE METHOD OF ENSURING THE FUNCTIONAL STABILITY OF THE INTEGRATED COMPLEX OF ON-BOARD EQUIPMENT IN DESTABILIZING CONDITIONS

Abstract. The authors have presented the method of ensuring the functional stability of aircraft avionics complexes under the influence of destabilizing factors. It is based on the procedure of maintaining valid knowledge bases of individual modules of the avionics complex through the organization of dynamic verification of such knowledge bases during their application on the principle of the probabilistic diagnostic core with system recovery without significant restructuring of verification means.

Головною особливістю функціонально стійкого інтегрованого комплексу бортового обладнання повинна бути здатність проведення самодіагностування технічного стану, прогнозування виникнення відмов і своєчасне парирування їх наслідків [1]. Для забезпечення функціональної стійкості інтегрованого комплексу бортового обладнання зростає значення засобів автоматизації управління польотом.

Математичну модель інтегрованого модульного комплексу авіоніки надаємо на основі опису елементів структури за допомогою орієнтованого графа та представляємо у вигляді розподіленої інтелектуальної системи:

$$G(V, L), V = \{v_i\}, L = \{l_{ij}\}, i, j = 1, 2, \dots, n, \quad (1)$$

У ній множина вершин V графа $G(V, L)$ відповідає множині інформаційних модулів комплексу авіоніки з розмірністю n , а множина ребер L – множині зв'язків l_{ij} між інформаційними модулями (i, j) . Ця модель, на відміну від існуючих, враховує випадкову структуру перевірочних зв'язків між модулями системи. Це дає можливість перевіряти окремі модулі комплексу авіоніки

іншими суміжними модулями в довільний час. Це дозволяє представити загальну діагностичну модель розподіленої бази даних комплексу авіоніки у вигляді орієнтованого графа $G(V, L)$, набір вершин $V = \{v_i\}$ якого відповідає модулям розподіленої бази даних системи комплексу авіоніки, а орієнтовані ребра $L = \{v_i, v_j\}$ – елементарним зв'язкам між інтелектуальними модулями.

Наведемо граф моделі комплексу авіоніки у вигляді матриці суміжності:

$$A = \{a_{ij} \mid i, j = 1, 2, \dots, N\}, \quad (2)$$

де a_{ij} – елементи матриці суміжності діагностичного графа, які мають значення 1, якщо є ребро:

$$l_{ij} = \{v_i, v_j\} \in L, \quad (3)$$

та 0 – якщо ребра не існує; v_i, v_j – модулі розподіленої системи комплексу авіоніки, об'єднанні інформаційними лініями зв'язку l_{ij} .

Математичні визначення необхідної та достатньої умов функціональної стійкості (ФС), ознак та критеріїв функціональної стійкості інтегрованого модульного комплексу авіоніки літака, структурних індикаторів запасу функціональної стійкості інтегрованого комплексу бортового обладнання (КБО) літака ми обґрунтували в науковій роботі [2].

Метод забезпечення функціональної стійкості інтегрованого комплексу авіоніки включає три послідовні етапи: виявлення відмови; її діагностування та розпізнавання через верифікацію розподіленої бази даних і своєчасне парування відмови.

Етап виявлення відмови ідентичний етапу оцінки технічного стану [3]. Після виявлення відмови визначається функція приналежності μ_i для окремих модулів $v_i, i = 1, \dots, n$, сукупності розподілених інформаційних модулів V комплексу авіоніки з базового набору M на основі операцій з бульовими векторами коректності модуля X_i , отриманих шляхом перевірки розподілених інформаційних модулів:

$$\forall_i \mid v_i \in V, i = 1, \dots, |V|,$$

$$\exists \mu_i | \mu_i \in M, \quad (4)$$

$$M = \left\{ \mu_i | \mu_i = \frac{1}{|X_i|} \sum_{j=1}^{|x_i|} x_{ij}, x_{ij} \in X_i, i=1,2,\dots,n \right\}$$

Верифікація розподіленої бази даних комплексу авіоніки – це процес визначення відповідності модулів специфікаціям, що описують процес під час його роботи, шляхом організації тестової діагностики на основі ймовірнісного діагностичного ядра, яке визначає випадкову структуру верифікаційних зв'язків із виконанням основних перевірок. у випадкові моменти часу:

$$M_D \in S, \quad S \setminus M_D \neq \emptyset, \quad M_i \leftarrow t_{ij}, \quad (5)$$

де M_D – набір апаратних компонентів ядра верифікації;

S – набір апаратних компонентів розподіленої бази даних комплексу авіоніки;

t_{ij} – остання елементарна перевірка в системі;

M_i – i -та апаратна складова комплексу авіоніки, яка виконувала останню перевірку.

Для організації оперативної самодіагностики динамічних систем найчастіше використовується система оцінки Препарата. Процес організації динамічної верифікації розподілених баз знань КБО для забезпечення його функціональної стійкості включає етапи [4]:

1) перевірки систем та окремих елементів комплексу авіоніки від інших модулів випадковим чином у випадкові моменти часу;

2) обмін і накопичення діагностичної інформації в пам'яті елементів авіоніки;

3) визначення параметрів необхідної надійності діагностування елементів комплексу авіоніки та формування перевірок для досягнення заданої надійності;

4) визначення достатності діагностичної інформації для її аналізу з використанням імовірності синдрому як ознаки достатності $SD = \{sd_{ij}\}$ (де SD – синдром, sd_{ij} – один результат тесту);

5) відстеження стану структури тестових зв'язків і результуючого синдрому по кожному з коректних модулів системи;

6) розшифровка діагностичної інформації шляхом аналізу синдрому та визначення місця та типу відмови елементів

інтегрованого КБО.

З накопиченням достатньої інформації про стан елементів у розподіленій базі даних комплексу авіоніки коректні модулі остаточно діагностують синдром $SD = \{sd_{ij}\}$. Розшифровка синдрому $SD = \{sd_{ij}\}$ дозволяє локалізувати відмову і визначити її характеристики. Висока достовірність результату перевірки забезпечується накопиченням інформації про результати перевірки тільки в коректних модулях X_i . При визначенні коректного модуля X_i розподіленої бази даних, здатного розшифрувати синдром, використовується метод умовної передачі результатів елементарних тестів.

Для забезпечення ФС інтегрованого комплексу авіоніки літака здійснюється перерозподіл його наявних ресурсів за принципом «синергетичного» відновного управління. Загалом завдання парування відмови полягає у визначенні нечіткого логічного висновку у вигляді:

$$Y = f(x_1, x_2, \dots, x_n). \quad (6)$$

Парування відмов і підтримання ФС інтегрованого комплексу авіоніки здійснюється шляхом відновлення розподіленої системи без істотної реорганізації засобів верифікації. Таким чином, в умовах збурень комплекс авіоніки продовжує функціонувати, вірогідно, з деяким погіршенням характеристик, але з виконанням мінімально необхідних функцій.

Оптимізація структури інтегрованого модульного комплексу авіоніки (ІМА – integrated modular avionic) літака вимагає одночасного вибору оптимальної структури та визначення основних змінних параметрів для забезпечення найкращих показників ефективності його роботи. Вирішення цієї проблеми можливе шляхом застосування методів оптимізації структури за критерієм максимуму узагальненого показника функціональної стійкості $F_{IMA \text{ AIRCRAFT}}$, обраного в якості критерію переваги з обмеженнями вартості на побудову системи:

$$F_{IMA \text{ AIRCRAFT}} = f(P_{ij}) = \sum_{i=1}^N \sum_{j=1}^N b_{ij} \cdot P_{ij},$$

де $F_{IMA \text{ AIRCRAFT}}$ – функціонал якості системи;

b_{ij} – вагові коефіцієнти;

P_{ij} – ймовірність зв'язності біполярного графа з вершиною-витоком v_i та вершиною стоком v_j ;

N – кількість вершин графа.

Вагові коефіцієнти b_{ij} визначаються на основі методу експертних оцінок. На основі модернізованих оцінок Поліського розраховано ймовірність зв'язності біполярного графа з вершиною-витоком v_i і вершиною стоком v_j .

Необхідно визначити оптимальну структуру розподіленої системи ІМА, яка характеризується максимальним рівнем функціональності $F_{IMA \text{ AIRCRAFT}}$, що залежить від ймовірності зв'язності при заданих обмеженнях. Усі комунікаційні лінії вихідної структури $G_0(V, L)$ повинні зберігатися в синтезованих структурах $G_\varepsilon(V, L)$. Для таких умов математична модель синтезу комплексу ІМА літака має вигляд:

$$F_{IMA \text{ AIRCRAFT}} = f(P_{ij}) \rightarrow \max, \quad i, j = 1, \dots, N, \quad i \neq j \quad (7)$$

з такими обмеженнями:

$$C_\varepsilon = \sum_i \sum_j C_{ij}(l_{ij}, \rho_{ij}, q_{ij}) \leq C_{A \varepsilon},$$

$$\forall \pi_{ij} \quad P_{ij} \geq P_{req},$$

$$\chi(G) \geq 2; \quad \lambda(G) \geq 3,$$

$$\{\omega(G) \geq 2 \cap k > 1\} \cup \{\mu(G) \geq 2 \cap k > 1\},$$

$$G_0(V, L) \subseteq G_\varepsilon(V, L),$$

$$\rho_{ij} > q_{ij},$$

$$\tau_{aver} \leq T_{\max},$$

де N – кількість блоків обміну інформацією комплексу авіоніки, які синтезуються;

$F_{IMA \text{ AIRCRAFT}}$ – функціонал якості системи, який максимізується;

P_{ij} – ймовірність зв'язності між парою (i, j) елементів системи;

P_{req} – ймовірність необхідного з'єднання;

p_{ij} – усі маршрути з імовірністю значення зв'язності p_{ij} ;

C_ε – встановлена вартість оптимізації структури;

C_{ij} – витрати на оптимізацію структури системи обміну інформацією між елементами (i, j) ;

C_{A_ε} – допустима встановлена вартість для оптимізації структури; $\lambda(G)$ – ступінь реберної зв'язності;

$\omega(G)$ – ω - з'єднання – максимальна кількість вершин незалежних маршрутів між будь-якою парою вершин графа $G(V, L)$;

$\mu(G)$ – це максимальна кількість незалежних від ребра маршрутів між будь-якою парою вершин графа $G(V, L)$;

k – кількість елементів оптимального руйнування $\omega(\mu)$ - зв'язного графа;

l_{ij} – довжина лінії з'єднання елементів (i, j) ;

ρ_{ij} – пропускна здатність одного каналу передачі інформації;

q_{ij} – інтенсивність обміну інформацією між елементами v_i, v_j ;

τ_{aver} – середній час затримки сповіщення, що передається від одного елемента до іншого;

T_{max} – максимально допустимий час виконання заданих часових характеристик;

ε – індекс заданого значення $\varepsilon=1, 2, \dots, n$, який передбачає знаходження кількох структур заданого значення C_{a_ε} для оптимізації та збільшення структури системи обміну інформацією.

У результаті розв'язання задачі (7) буде синтезовано оптимально розподілену систему комплексу ІМА з елементів із певним значенням матриці суміжності інформаційних зв'язків її графа. Визначення раціонального варіанта моделі комплексу ІМА можна здійснити за критерієм максимуму узагальненого показника ФС, обраного як критерій переваги з обмеженням вартості побудови системи. У такій постановці граф структури авіаційного комплексу ІМА має детерміновану кількість вузлів і випадкову кількість зв'язків, і задачу можна розв'язати за допомогою використання принципу максимуму Понтрягіна або іншими класичними методами.

Література

1. Mashkov O., Chumakevych V., Sokulsky O., & Chyrun L. Features of determining controlling effects in functionally-stable systems with the recovery of a control. *Mathematical Modeling and Computing*. 2019. 6, 80–86.

2. Калашник Г.А., Калашник-Рибалко М.А. Ознаки та критерії функціональної стійкості інтегрованого комплексу бортового обладнання сучасного повітряного судна та перспективні напрямки його розвитку. *Збірник наукових праць Харківського національного університету Повітряних сил*. 2021. №2(42). С.7-15.

3. Калашник-Рибалко М.А., Калашник Г.А. Удосконалення методу синтезу оптимально розподіленої структури інтегрованого комплексу бортового обладнання літального апарата за показниками функціональної стійкості. *Наука і техніка Повітряних Сил Збройних сил України*. 2021. №1(42). С.41-49.

4. Kalashnyk G.A., Kalashnyk-Rybalko M.A. Methodology for Ensuring the Functional Stability of Aircraft Integrated Modular Avionics Complex. *Наука і техніка Повітряних Сил Збройних сил України*. 2023. №4(53). С.30-40.

ПОБУДОВА ВИПАДКОВИХ ГРАФІВ НА БАЗІ МОДЕЛІ ЕРДЕША-РЕНЬЇ

Ігор Козін, Олег Сардак, Артур Алексєєв
Запорізький національний університет
Навчально-науковий інститут ВВП "Дніпровський
гуманітарний університет"

CONSTRUCTION OF RANDOM GRAPHS BASED ON THE ERDOS RENY MODEL

Abstract. Випадкові графи використовують для моделювання різних процесів у різних прикладних областях. Дослідженню випадкових графів присвячено безліч наукових праць як теоретичного, і прикладного характеру[1,2]. Крім того, випадкові графи можуть застосовуватися при створенні баз тестових прикладів для перевірки якості метаевристичних для широкого спектру прикладних задач, які зводяться до екстремальних задач на графах. Тому актуальним є дослідження алгоритмів побудови випадкових графів із заданими властивостями. Наприклад, випадкових дерев, регулярних графів або графів, що містять гамільтонів цикл.

Модель Ердеша-Реньї. Алгоритм побудови випадкового графу

Найбільш відомою моделлю генератора випадкових графів є модель Ердеша-Реньї [3], в якій кожне ребро графа генерується з однаковою заданою ймовірністю. Розглянемо задачу побудови випадкових графів з урахуванням використання саме цієї моделі.

Алгоритм генерації графів у моделі Ердеша-Реньї використовує лише один параметр $\alpha \in (0,1)$. При цьому ймовірність генерації графа $G = (V, E)$ із заданим числом вершин $n = |V|$ і заданим числом ребер $m = |E|$ визначається формулою:

$$P(G) = \alpha^m (1 - \alpha)^{\frac{n(n-1)}{2} - m}.$$

Нехай всі вершини графа G занумеровані від 1 до n і матриця суміжності графа G має вигляд $A = (a_{ij})_{i,j=1,2,\dots,n}$, де

$$a_{ij} = \begin{cases} 1, & \text{якщо ребро } (i, j) \text{ належить множині ребер графа } G \\ 0, & \text{в іншому випадку} \end{cases}$$

Наведемо алгоритм генерації випадкової матриці суміжності в моделі Ердеша-Реньї на мові C++:

```
for (var i = 1; i <= n; i++)
{
    for (var j = i + 1; j <= n; j++)
    {
        // random(x, y) повертає випадкове число
        на відрізку [x, y]
        if (random(0, 1) <= alpha)
        {
            c[i, j] = 1;
        }
        else
        {
            c[i, j] = 0;
        }
    }
}
```

Алгоритм побудова випадкового зв'язаного графу

Випадковий граф, який побудовано за наведеним алгоритмом моделі Ердеша Реньї не обов'язково буде зв'язаним. Якщо потрібно

згенерувати саме зв'язаний граф, то алгоритм потрібно буде змінити. Один з методів побудови випадкового зв'язаного графу складається з двох етапів. На першому етапі генерується випадкове кістякове дерево графу G . На другому етапі до цього дерева додаються ребра, що йому не належать за алгоритмом моделі Ердеша-Реньї. Тобто кожне нове ребро додається з однаковою ймовірністю α .

Розглянемо один з методів побудови випадкового кістякового дерева. Множину номерів $I = \{1, 2, \dots, n\}$ вершин графу розіб'ємо на дві частини $I = I_1 \cup I_2$. На початковому кроці $I_1 = \{ \}$, $I_2 = \{1, 2, \dots, n\}$. На першому кроці вибирається випадково два індекси $i_0, j_0 \in I_2$ і переносяться із множини I_2 в множину I_1 . Відповідно в дерево, що будується, додається ребро (i_0, j_0) . На черговому кроці випадково вибираються два номери $i \in I_1, j \in I_2$. В дерево, що будується додається ребро (i, j) . Після цього номер j вилучається із множини I_2 та доється до множини I_1 . Алгоритм закінчує роботу, коли множина I_2 стає порожньою. Наведемо одну із можливих реалізацій цього алгоритму на мові C++.

```

var I1[n] = { };
var I2[n] = {1, 2, 3, ..., n };
// випадково обирається ребро
var i0 = I2[random(1, I2.Length)];
I2.Remove(i0);
var j0 = I2[random(1, I2.Length)];
I1.Add(i0);
I1.Add(j0);
c[i0, j0] = 1;
do
{
    var i = I1[random(1, I1.Length)];
    var j = I2[random(1, I2.Length)];
// ребро додається до множини ребер графу
    c[i1, i2] = 1;
    I2.Remove(i2);
    I1.Add(i2);
}
while (I2.Length != 0);

```

В результаті роботи наведеного алгоритму буде отримана матриця суміжності графу, який є кістяковим деревом.

Алгоритм побудови випадкового графу, що має гамільтоновий цикл

Розглянемо тепер алгоритм побудови випадкового графу, який має гамільтоновий цикл. Гамільтоновий цикл однозначно визначається перестановкою вершин графу. Таким чином на першому етапі алгоритму буде побудований випадковий простий цикл, що містить всі вершини графу. На другому етапі до цього циклу будуть додані нові ребра відповідно моделі Ердеша Реньї.

Для побудови випадкової перестановки чисел $1, 2, \dots, n$ скористуємося алгоритмом тасування Фішера-Йетса [4]. Реалізація цього алгоритму на мові C++ має наступний вигляд:

```
// в масиві a[n] формується випадкова
перестановка чисел
    1, 2, ..., n
var a[n] = {1, 2, ..., n};
    //round(x) округляє число x до найближчого
цілого
for (var i = 1; i <= n; i++)
    {
    // round(x) округляє число x до найближчого
цілого
        var j = 1 + round(random(0, 1) * (n - 1));
        // swap(x, y) обмінює значеннями змінні x та
у
        swap(a[i], a[j]);
    }
```

Якщо отримана перестановка чисел i_1, i_2, \dots, i_n , то до графу додаються відповідні ребра $(i_1, i_2), (i_2, i_3), \dots, (i_{n-1}, i_n), (i_n, i_1)$, що утворюють цикл гамільтона.

Алгоритм побудови випадкового дводольного графу

Розглянемо задачу побудови випадкового дводольного графу. Нехай перша доля містить n вершин, а друга m вершин. Занумеруємо вершини першої долі числами $1, 2, \dots, m$, а другої долі $1, 2, \dots, n$. Дводольний граф будемо визначати матрицею $\{b_{ij}\}_{i=1, 2, \dots, m, j=1, 2, \dots, n}$, де

$$b_{ij} = \begin{cases} 1, & \text{якщо вершина } i, \text{ першої доли з'єднана ребром з вершиною } j \text{ другої доли} \\ 0, & \text{в іншому випадку} \end{cases}$$

Алгоритм побудови випадкового графу за моделлю Ердеша Реньї має наступний вигляд:

```

for (var i = 1; i <= n; i++)
{
    for (var j = i + 1; j <= m; j++)
    {
        // random(x, y) повертає випадкове число
на відрізьку [x, y]
        if (random(0, 1) <= alpha)
        {
            c[i, j] = 1;
        }
        else
        {
            c[i, j] = 0;
        }
    }
}

```

Випадкові графи з заданим частковим графом

Виходячи з попередніх прикладів, можна запропонувати спосіб побудови випадкового графа, який містить заданий частковий граф. Алгоритм побудови такого графа має такий вигляд: на першому етапі будується випадковий граф, який має задану властивість. Далі випадково додаються відсутні ребра відповідно до моделі Ердеша Реньї.

Наприклад, розглянемо принцип побудови випадкового графа з n вершинами, який містить повний подграф з вершинами k . Вершини графа нумеруються числами $1, 2, \dots, n$. На першому етапі випадково вибираються k чисел з множини $1, 2, \dots, n$ і додаються всі ребра, що з'єднують пари вершин з номерами з обраного набору чисел. Далі, кожне з ребер (i, j) , які ще не увійшли до множини ребер додається до цієї множини із заданою ймовірністю.

Література

1. M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, 1979, 338p.

2. Beasley J.E. “OR-Library: distributing test problems by electronic mail”, *Journal of the Operational Research Society*, 1990 no. 41, pp. 1069-1072.

3. Erdos P. and Spencer J. “Probabilistic methods in combinatorics”, Academic Press, New York, 1974, pp. 1375-1383.

4. D. Knuth, The Art of Computer Programming, Volume 2: Seminumerical Algorithms, Addison-Wesley, Boston, Third edition, 1997, 650p.

АЛГОРИТМ НАВІГАЦІЇ ГРУПИ СІЛЬСЬКОГОСПОДАРСЬКИХ БПЛА ЗА ВІДСУТНОСТІ GPS

Вячеслав Корольов, Максим Огурцов, Олександр Ходзінський
Інститут кібернетики імені В.М. Глушкова НАН України

NAVIGATION ALGORITHM FOR A GROUP OF AGRICULTURAL UAVS IN THE ABSENCE OF GPS

Abstract. The use of unmanned aerial vehicles (UAVs) is rapidly expanding, which is driven by a shift in focus from single UAVs to utilizing swarms of UAVs for increased efficiency. These swarms can be employed in various applications, such as agricultural surveys and irrigation.

The objective of this research was to develop a navigation algorithm for a group of agricultural UAVs in the absence of GPS (or its bad quality) to maintain a stable formation during group movement.

That problem has been formalized. Possible methods for representing the graph formed by the UAV group in two-dimensional Euclidean space using a base triangle method to create a relative coordinate system have been considered. The method of multilateration was used to determine the coordinates of the remaining vertices of the graph on the plane and to improve the accuracy of local positioning accordingly. Initial practical experiments have demonstrated the effectiveness of the proposed approach.

Вступ. Безпілотні літальні апарати (БПЛА) все активніше інтегруються у різні галузі, постійно розширюючи свої функціональні можливості. Замість використання окремих БПЛА, все більше уваги приділяється їхнім групам, оскільки для багатьох

завдань, таких як моніторинг сільськогосподарських угідь або зрошення, групове застосування виявляється значно ефективнішим та продуктивнішим [1]. Підвищення ефективності досягається шляхом значного скорочення кількості операторів, потрібних для керування тією ж кількістю БПЛА у складі групи, на відміну від традиційної схеми "один оператор – один БПЛА" [2]. Ба більше, використання груп дозволяє впроваджувати групове керування з елементами штучного інтелекту, яке у багатьох завданнях значно перевершує можливості та швидкість реакції людини. Втім, існують і певні проблеми, зокрема, недоступність GPS у ряді випадків для агродронів, що виконують обробку сільськогосподарських угідь від шкідників або моніторинг посівів [3].

Мета. Метою даної роботи є розробка алгоритму навігації групи сільськогосподарських БПЛА за відсутності (або поганої якості сигналу) GPS для утримання сталої структури під час руху групи. Передбачається, що групою, незалежно від кількості БПЛА, керує один оператор. Розроблений алгоритм повинен не тільки забезпечувати ефективне керування, але й запобігати потенційним зіткненням БПЛА під час повітряних маневрів.

Зазвичай у подібних задачах не застосовують підхід із розробкою системи локального взаємного позиціонування БПЛА, який ми пропонуємо в цій роботі.

Аналіз попередніх розробок та публікацій. У наукових роботах, присвячених подібним задачам [1-7], варто відзначити підходи, що базуються на парадигмі Інтернету речей та штучного інтелекту [4]. Незважаючи на їхню інноваційність, вони вимагають великих обчислювальних ресурсів, що обмежує їхнє практичне застосування. Застосування паралельних генетичних алгоритмів та інших методів комбінаторної оптимізації [5] також викликає інтерес, але автори вважають, що у багатьох випадках виникаючі задачі можна розв'язувати простішими методами.

Інший підхід полягає в попередньому плануванні маршрутів групи БПЛА [6], що значно звужує можливості їх використання. Одним із найпоширеніших методів є оптимізація роєм частинок [6-7], проте цей підхід також є досить ресурсоемним.

Основна частина. Розглянемо, як задача може бути розв'язана в двовимірній постановці, яка цілком застосовна за умови, що група рухається на однаковій висоті і різницю їх висот можна ігнорувати. Це також підходить для наземних безпілотних апаратів, що взаємодіють у зоні прямої видимості. Вважатимемо елементи групи

гомогенними, хоча з додаванням додаткових параметрів та обмежень можна використовувати й гетерогенні БПЛА.

Маємо набір БПЛА, які формують групу. У цій роботі розглядається група до 30 БПЛА. Кожен БПЛА оснащений датчиками (GPS, компас, акселерометр, висотомір тощо), що дозволяють визначати взаємне положення та підтримувати структуру групи. Завдання полягає в тому, щоб досягти необхідної конфігурації групи в мінімальний час і підтримувати цю конфігурацію з мінімальними витратами ресурсу.

Алгоритм повинен працювати ітеративно з частотою повторення не менше разу на кілька секунд, щоб враховувати можливі зміни положення елементів групи БПЛА. Конфігурація групи повинна забезпечувати однакові відстані між сусідніми БПЛА.

Змістовна постановка задачі.

Вхідні дані:

1. Кількість БПЛА.
2. Координати GPS.
3. Напрямок руху.
4. Швидкість.
5. Кут нахилу.
6. Висота.

Параметри:

1. Максимальна швидкість руху БПЛА.
2. Мінімумально допустима висота польоту БПЛА.
3. Максимальна дальність польоту БПЛА.
4. Мінімумально допустима відстань між БПЛА.

Критерій:

Мінімізація сумарного додаткового руху усіх БПЛА групи, потрібного для відновлення заданої конфігурації.

Вихідні дані:

GPS координати точок, куди має переміститись кожен БПЛА для відновлення заданої конфігурації групи.

Граф, що утворюється групою, спочатку поданий у тривимірному Евклідовому просторі [8], а потім має бути приведений до двовимірного графу.

Позначення

$G(V,E)$ – заданий повний граф, що відображає усі БПЛА,
 $n=|V|$ – кількість вершин графа,

$V=\{v_1, \dots, v_n\}$ – множина вершин, кожна із яких подає елемент групи, (окремий БПЛА), положення якого визначається трьома координатами,

$(x_i, y_i, z_i), i=1, n$ – координати вершин (БПЛА), де x_i, y_i невідомі, а z_i відомі.

$E=\{e_{12}, e_{13}, \dots, e_{n-1, n}\}$ – відомі довжини ребер.

Відображення графа з тривимірного Евклідова простору на площину

З рисунка 1 видно, що довжина проєкції ребра e_{ij} між вершинами v_i, v_j на площину XOY:

$$e'_{ij} = \sqrt{e_{ij}^2 - (z_j - z_i)^2}.$$

Для створення необхідної конфігурації вершин (елементів групи БПЛА) на основі відомих координат, слід виконати такі кроки: Вибір базового трикутника, фіксація базового трикутника на площині, Мультилатерація для визначення координат інших вершин графа.

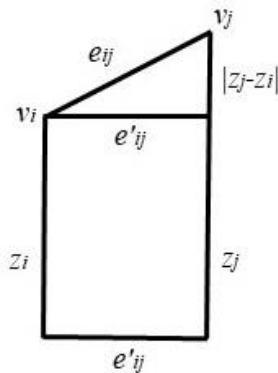


Рис. 1 – Зведення тривимірної задачі до задачі на площині

Вибір базового трикутника. Базовий трикутник – це трикутник, який можна сформувати з вершин графа.

Використання базового трикутника дозволяє формувати потрібну конфігурацію всередині нього, не збільшуючи максимальну відстань між будь-якими двома вершинами.

Критерії вибору базового трикутника:

1. Максимальна кількість елементів:

Базовий трикутник обирається з урахуванням максимальної кількості елементів групи, які можна розмістити всередині нього.

2. Максимальна площа трикутника.

Після вибору базового трикутника, можна формувати потрібну конфігурацію вершин всередині нього, не порушуючи обмеження на максимальну відстань між двома вершинами (елементами групи).

Фіксація базового трикутника на площині. Після вибору базового трикутника зберігається кілька ступенів свободи при розміщенні проекції графа на площині. Виберемо такі: початок системи координат (перша вершина трикутника), напрямок осі абсцис (друга вершина трикутника), напрямок осі ординат (третя вершина трикутника).

1. Зафіксуємо координати вершини v_b у якості початку координат:

$$v_b = (0,0)$$

2. Сторона $v_b v_c$ нехай буде частиною осі $OX: v_c = (e'_{bc}, 0) = (a, 0)$

3. y_a – координату вершини v_a визначимо з площі трикутника:

$$S = \sqrt{p(p-a)(p-b)(p-c)} = \frac{ay_a}{2}$$

4. x_a – координату вершини v_a визначимо за теоремою Піфагора:

$$x_a = \sqrt{c^2 - y_a^2}$$

Завдяки доступності даних супутникового позиціонування, завдання підтримки структури групи БПЛА може бути спрощено до підтримки позиції кожного БПЛА у заданій локації геометричної фігури, використовуючи абсолютні значення координат.

Обмеження супутникового позиціонування. При малих відстанях між БПЛА точність супутникового позиціонування може бути недостатньою. Це може призвести до порушення мінімальної відстані, встановленої для запобігання зіткненням БПЛА.

Мультилатерація. Для визначення координат інших вершин графа можна використати метод мультилатерації [9]. Це метод локального позиціонування, який не залежить від супутникових систем глобального позиціонування (GPS) і є більш точним [10].

Алгоритм трilaterації:

Спрощений алгоритм трilaterації складається з наступних кроків:

Етап 0: Калібрування відстаней за потужністю:

Вхідні дані: Таблиця відстаней, отримана в результаті калібрування.

Вихідні дані: Відкалібровані відстані між вузлами.

Етап 1: Збір даних про потужність сигналу:

Кожний вузол мережі БПЛА отримує інформацію про потужність сигналу від інших вузлів і передає їм свої дані.

Етап 2: Розрахунок відстаней та координат:

Кожний вузол для всіх інших досяжних вузлів розраховує відстані за таблицею з результатами калібрування та розв'язує систему рівнянь трилатерації для отримання своїх локальних координат.

Отримані значення координат надсилаються всім вузлам мережі [11].

Важливо зазначити, що точність мультилатерації залежить від щільності розташування вузлів мережі та якості приймально-передавального обладнання. Тому комбінування мультилатерації з супутниковим позиціонуванням може забезпечити ще більш високий рівень точності локального позиціонування БПЛА.

Початкові практичні експерименти показали застосовність пропонованого підходу, при цьому акцент робився на виконанні обчислень на одноплатних комп'ютерах типу Raspberry Pi 4B на борту БПЛА [12].

Висновки. В рамках виконання роботи поставлено задачу формування та підтримання структури та конфігурації групи сільськогосподарських БПЛА в двовимірній постановці. Виконано формалізацію поставленої задачі. Розглянуті можливі методи відображення графа, що утворюється роєм БПЛА, у двовимірний Евклідов простір методом формування базового трикутника для створення відносної системи координат. Для визначення координат решти вершин графа на площині та відповідного підвищення точності локального позиціонування застосований метод мультилатерації. Розглянуті варіанти застосування алгоритму трилатерації та триангуляції. Початкові практичні експерименти показали пропонованого підходу, дозволяючи визначити локальне позиціонування елементів групи, що містить від двох до тридцяти елементів.

Напрямом подальших досліджень є розробка алгоритму, що дозволяє на основі визначених локальних координат ліквідувати збурення, внесені в структуру групи через внутрішні чи зовнішні чинники, наприклад, втрату одного з елементів групи [13].

Література

1. Y. Zhou, R. Bin, W. Wei, UAV swarm intelligence: Recent advances and future trends / *Ieee Access* 8, 2020. Pp. 183856-183878.
2. M. Champion, R. Prakash, F. Saleh, UAV swarm communication and control architectures: a review / *Journal of Unmanned Vehicle Systems*, 2018. 7.2. Pp. 93-106.
3. Сучасні методи цифрового моніторингу в рослинництві: монографія/ Зозуля О.Л. та ін. К.: КНУ імені Тараса Шевченка, 2023. 254 с.
4. X. Wang, G. Chen, H. Gong, J. Jiang, UAV swarm autonomous control based on Internet of Things and artificial intelligence algorithms, *Journal of Intelligent & Fuzzy Systems*, 40(4), 2021. Pp. 7121-7133.
5. V. Roberge, M. Tarbouchi, G. Labonté, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Transactions on industrial informatics*, 9(1), 2012. Pp. 132-141.
6. S. Shao, Y. Peng, C. He, Y. Du, Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. *ISA transactions*, 97, 2020. Pp. 415-430.
7. M. D. Phung, Q. P. Ha, Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. *Applied Soft Computing*, 107, 2021. P. 107376.
8. D. Sikeridis, E. EleniTsiropoulou, M. Devetsikiotis, S. Papavassiliou, Self-adaptive energy efficient operation in UAV-assisted public safety networks / 2018 IEEE 19th international workshop on signal processing advances in wireless communications (SPAWC) 2018, June, IEEE. 2018. Pp. 1-5.
9. S. N. Othman, Node positioning in zigbee network using trilateration method based on the received signal strength indicator (RSSI). *European Journal of Scientific Research*, 46(1), 2010. Pp. 48-61.
10. F. He, T. Zhou, W. Xiong, S. M. Hasheminnasab, A. Habib, Automated aerial triangulation for UAV-based mapping, *Remote Sensing*, 10(12), 2018. P. 1952.
11. B. Yang, E. Yang, L. Yu, A. Loeliger, High-precision UWB-based localisation for UAV in extremely confined environments, *IEEE Sensors Journal*, 22(1), 2021. Pp. 1020-1029.
12. M. Ogurtsov, V. Korolyov, O. Khodzinskyi, Improving the Productivity of UAV Operations Based on Granular Computing and Fuzzy Sets / 2021 IEEE 6th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD) 19-21 Oct. 2021. 2021, Pp. 33-36.

13. V.P. Horbulin, L.F. Hulianytskyi, I.V. Sergienko, Planning of Logistics Missions of the “UAV+Vehicle” Hybrid Systems, Cybernetics and System Analysis, 59, 2023. Pp. 733–742.

МАТЕМАТИЧНИЙ СЕЙФ, ГРАЦІОЗНІ ДЕРЕВА ТА ЗАХИСТ ІНФОРМАЦІЇ

Кривий С. Л.

Київський національний університет імені Тараса Шевченка

MATHEMATICAL SAFE, GRACEFUL TREES AND PROTECTION OF INFORMATION

Kryvyi S. L.

Taras Shevchenko national university of Kyiv

Abstract. Considered some applications of mathematical safe model and graceful trees to protection of software. Some approaches are proposed to solution of such problems. Corresponding algorithms have polynomial time complexity. Ref. 3 titles.

1. Вступ

Робота присв'ячується світлій пам'яті **Георгія Панасовича Донця**, відомого вченого і чудову людину, яка внесла вагомий вклад в області дискретної математики, теорії графів, математичних ігор, теорії оптимізації та інші області. В даній роботі розглядаються методи захисту програмного забезпечення, які ґрунтуються на моделях математичного сейфа [1] та граціозних дерев [2], які були розроблені Г.П. Донцем та досліджувалися його учнями та колегами.

2. Математичний сейф

Перше формулювання задачі про математичний сейф було дано в роботі [1]. Математичним сейфом називається система $Z=(z_1, z_2, \dots, z_n)$ взаємопов'язаних засувів така, що коли виконується поворот ключа в одному із засувів, то такий же поворот виконується і у всіх засувах, які пов'язані з даним. Математичний сейф може задаватися двома способами: за допомогою прямокутної матриці, елементи якої відповідають засувам, і за допомогою графа, вершини якого відповідають засувам. При матричному заданні математичного сейфа кожний засув z_{ij} пов'язаний з тими засувами, які розміщені в i -му рядку і в j -му стовпчику. А при заданні математичного сейфа за допомогою графа, пов'язаними з даним

засувом є ті, які розміщені у суміжних вершинах. Далі в роботі розглядатимуться тільки графові сейфи, які задаються стовпчиком початкових позицій засувів $b=|b_i|$, $i=0,1,\dots,k-1$. Кожний засув може знаходитися в одній із декількох позицій. Всіх можливих позицій скінченне число: $0,1,\dots, k-1$. Засув відкритий, якщо він знаходиться у позиції 0. У довільній іншій позиції засув вважається закритим. При цьому, якщо у якомусь засуві виконується поворот ключа, то всі засуви, пов'язані з даним засувом, збільшують свої позиції на одиницю за модулем k .

Необхідно розв'язати таку задачу: виходячи з початкових позицій засувів сейфа, заданих вектором $b=|b_i|$, де $b_i \in \{0,1,\dots, k-1\}$, знайти таку послідовність засувів і число поворотів ключа в них, щоб сейф перейшов у положення відкритого, тобто коли всі його засуви знаходяться у позиції 0.

3. Задання математичного сейфа графом

Нагадаємо формулювання задачі про математичний сейф за допомогою графа. У вершинах графа $G=(V,E)$ знаходяться засуви, які можуть знаходитися у одній з позицій із множини $\{0,1,\dots,k-1\}$. Якщо у вершині u засув знаходиться у позиції i , то поворот ключа в цій вершині переводить її засув у позицію $(i+1)(\text{mod } k)$ а також всі засуви у вершинах, які суміжні з вершиною u . Початкові позиції засувів у вершинах задаються вектором b .

Г. П. Донець запропонував чудову інтерпретацію задачі про математичний сейф. Ця інтерпретація зводила розв'язання задачі про математичний сейф до розв'язання системи лінійних однорідних рівнянь (СЛОР) $Ax+b=0$ за модулем k над вибраною областю D (а ця область є або кільцем або полем лишків за модулем k). Застосовуючи TSS-алгоритм [3] для розв'язання такого типу систем, дістаємо розв'язки СЛОР, які відкривають сейф. Проілюструємо сказане прикладом графового сейфа, який заданий неорієнтованим графом, який наведено на рисунку 1.

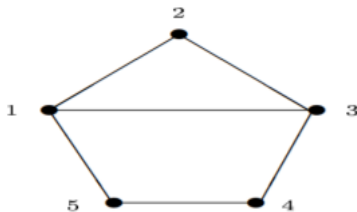


Рис. 1 Неорієнтований граф

Будується СЛОДР $Ax+b=0 \pmod{k}$, де $b=(1,2,1,1,3)$. Яка відповідає даному графовому сейфу:

$$A = \begin{cases} 1 & 2 & 3 & 4 & 5 & b \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 2 & 1 & 1 & 1 & 0 & 0 & 2 \\ 3 & 1 & 1 & 1 & 1 & 0 & 1 \\ 4 & 0 & 0 & 1 & 1 & 1 & 1 \\ 5 & 1 & 0 & 0 & 1 & 1 & 3 \end{cases} = 0 \pmod{5},$$

де верхній рядок СЛОДР це нумерація стовпчиків, а лівий стовпчик – нумерація рядків СЛОДР. Застосовуючи TSS-алгоритм до розв'язання цієї системи, дістаємо такий розв'язок $a=(0,1,2,1,1,1)$, який відкриває сейф. Дійсно, додаючи до стовпчика b , відповідні стовпчики матриці, помножені на координати вектора a , дістаємо:

$$\begin{array}{cccccc} 1 & 2 & 1 & 1 & 3 \\ & a_2 & = & 1 & \end{array}$$

$$\begin{array}{cccccc} 2 & 3 & 2 & 1 & 3 \\ & a_3 & = & 2 & \end{array}$$

$$\begin{array}{cccccc} 4 & 0 & 4 & 3 & 3 \\ & a_4 & = & 1 & \end{array}$$

$$\begin{array}{cccccc} 4 & 0 & 0 & 4 & 4 \\ & a_5 & = & 1 & \end{array}$$

$$0 \ 0 \ 0 \ 0 \ 0.$$

4. Граціозна нумерація дерев

Нехай дано неорієнтований граф $G=(V,E)$ з n ребрами та занумерованими вершинами. *Індукованою міткою ребра* називається абсолютна величина різниці між номерами (мітками) двох його кінців. Граціозною нумерацією неорієнтованого графа G називається бієктивна відповідність φ між множиною вершин графа G та множиною $\{0,1,\dots,n-1\}$ така, що всі індуковані мітки ребер є різними. Іншими словами, нумерація φ граціозна, якщо $\varphi: V \rightarrow \{0,1,\dots,|E|\}$ є ін'єкцією і всі ребра G мають різні номери з множини $\{1,\dots,|E|\}$. Граф називається *граціозним*, якщо він

допускає граціозну нумерацію. Далі будуть розглядатися графи, які є деревами. Граціозною нумерацією дерева T з n вершинами називають присвоєвання його вершинам номерів $1, 2, \dots, n$ при якому абсолютні різниці номерів кінців ребра є різними та утворюють послідовність $1, 2, \dots, n-1$. Граціозними деревами є зірка, ланцюг, гусениця, оливкове дерево, бананове дерево тощо [2].

Важливу властивість дерев висвітлює така теорема.

Теорема 1. З будь якої граціозної нумерації дерева порядку n можна отримати іншу граціозну нумерацію шляхом перекодування $x_i = n + 1 - x_i$. Таку нумерацію називають двоїстою нумерацією [2].

5. Задача захисту інформації

Проблема захисту інформації на даний час є однією з пріоритетних в інформаційних технологіях. Кожний користувач, який хоче обмежити небажаним особам несанкціонований доступ до своєї приватної інформації, потребує засобів захисту та засобів надання права доступу до такої інформації. Наприклад, кожен користувач потребує захисту своєї **web**-сторінки. Таким простим і разом з тим ефективним може стати використання моделей математичного сейфу та граціозного дерева. Для пояснення ідеї скористаємося прикладом найпростішого типу дерева – ланцюгом, який має 9 вершин і таку граціозну нумерацію: 1,9,2,8,3,7,4,6,5.

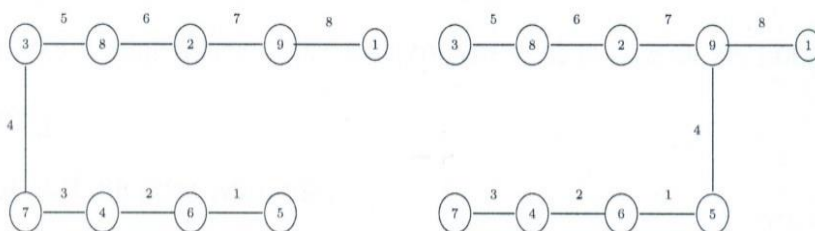


Рис. 2. Граціозні дерева

На кожному з цих ланцюгів, як на графі, можна визначити математичний сейф з такими початковими станами засувів 0,8,7,6,5,4,3,2,1, які є мітками ребер графа. Наявність декількох граціозних ланцюгів дає можливість варіації вибору ланцюга. А це призводить до зміни рівнянь у СЛЮДР і до зміни розв'язків системи в цілому. Тоді, для доступу до конфіденційної інформації особі небажаний потрібно знати топологію графа, його граціозну нумерацію і модуль кільця чи поля лишків, у якому потрібно шукати

послідовність перемикання засувів з метою відкриття сейфа. Доступ до інформації відкривається тільки тоді, коли сейф відкрито.

Для математичного сейфа, заданого графом (зліва) на рисунку 2, отримуємо таку СЛОДР

$$A = \begin{cases} 1 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & b \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 8 \\ 3 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 7 \\ 4 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 6 \\ 5 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 5 \\ 6 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 4 \\ 7 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 3 \\ 8 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 2 \\ 9 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{cases} = 0 \pmod{9},$$

розв'язання якої дає таку послідовність перемикання засувів $a=(4,8,2,1,5,8,3,6,5,1)$, що відкриває сейф. Зрозуміло, що коли задати математичний сейф графом справа, показаним на рисунку 2, то СЛОДР буде мати інший розв'язок. В цьому і полягає варіація граціозним деревом.

Для отримання доступу до інформації особа, якій належить дана інформація, надає доступ абонентам до цієї інформації шляхом повідомлення топології графа, для якого абонентам відомі порядок графа та його граціозна нумерація. Опираючись на властивості дерев, які впливають з теореми 1, дістаємо можливість варіації граціозною нумерацією графа. Крім того, можна варіювати правилами перемикання засувів у вершинах. Наприклад, нехай правила перемикання засувів є такими: коли виконується поворот ключа у засуві 2, то його позиція збільшується на одиницю за модулем k , а у суміжних вершинах позиції збільшуються на 3, поворот ключа в засуві 6 збільшує його позицію на одиницю, а у суміжних вершинах позиції збільшуються на 2 одиниці, поворот ключа у засуві 8 збільшує його позицію на одиницю, а в суміжних вершинах позиції збільшуються на 5 одиниць. Тоді СЛОДР для такого сейфа набуває вигляду:

$$A = \begin{pmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & b \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 8 \\ 3 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 7 \\ 4 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 6 \\ 5 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 5 \\ 6 & 0 & 0 & 0 & 2 & 2 & 1 & 0 & 0 & 0 & 4 \\ 7 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 3 \\ 8 & 0 & 5 & 5 & 0 & 0 & 0 & 0 & 1 & 0 & 2 \\ 9 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} = 0 \pmod{9},$$

розв'язком якої є така послідовність перемикання засувів $b=(1,6,3,6,0,3,0,3,8,3)$, яка відкриває сейф.

Зауважимо, що для відкриття доступу до інформації потрібна сумісність відповідної СЛОП. У полі лишків для цього потрібна лінійна незалежність рівнянь системи, а у кільці лишків – лінійна незалежність рівнянь і значення детермінанта є дільником одиниці [3].

Для передачі ключа доступу абонентам, при відомій топології граціозного дерева, передається в даному прикладі трійка $2(3),6(2),8(5)$.

Зрозуміло, що це не єдина можлива варіація і таких варіацій існує принаймні $k!$.

Література

1. Г. П. Донець. Задачі про сейф на $(0,1)$ -матрицях. Ж. Кібернетика та системний аналіз.- 2002.- № 1.- С. 98-105.
2. Г.П. Донець, Д.А. Петренюк Граціозна нумерація дерев. Київ. Національна академія наук України. Інститут кібернетики ім. В.М. Глушкова. – 2007. – 142 с
3. С.Л. Кривий Лінійні Діофантові обмеження та їх застосування. Київ: Інтерсервіс. - 2021. – 260 с.

ІНТЕГРАЛЬНИЙ ІНВАРІАНТ ГРАФУ

Сергій Курапов, Максим Давидовський

Запорізький національний університет, КЗ «Запорізький обласний інститут післядипломної педагогічної освіти» ЗОР

INTEGRAL GRAPH INVARIANT

Abstract. The paper examines the properties of various graph invariants that have polynomial computational complexity. It is shown that the invariant of edge cuts of a graph and the invariant of edge cycles of a graph, if applied separately to the solution of the problem of recognizing graph isomorphism, cannot be considered a complete invariant of the graph. The paper presents an integral graph invariant based on the joint application of the edge cuts invariant and the graph edge cycle invariant. The integral graph invariant is based on the existence of a relation between the subspaces of cuts and cycles of the space of subgraphs. To illustrate the properties of the integral invariant, an example is given of constructing a direct sum of subspaces of cuts $S(G)$ and cycles $C(G)$ in the case of graphs that do not have intersecting elements in the subspaces. The structure of the digital invariant of the edge graph $L(G)$ is considered and the relation with the integral invariant of the graph is revealed. It is shown that the integral invariant is a complete invariant of the graph.

На сьогоднішній день відомі наступні інваріанти графів, що застосовуються для створення алгоритмів розпізнавання ізоморфізму двох неорієнтованих несепарабельних графів G і H :

1. Вектор локальних степенів графа $\rho_v(G)$.
2. Цифровий інваріант реберного графа $IL(G)$.
3. Інваріант реберних розрізів графа $IS(G)$.
4. Інваріант реберних циклів графа $IC(G)$.

У даній роботі розглядаються питання застосування цих інваріантів для вирішення задачі визначення ізоморфізму двох несепарабельних графів G і H . Алгоритми розпізнавання ізоморфізму таких графів G і H мають поліноміальну обчислювальну складність. Відомо, що інваріант, який ґрунтується на векторі локальних степенів графа, не є повним інваріантом для визначення ізоморфізму графів [1–2,5]. Інваріант реберних розрізів графа також не є повним інваріантом графа з точки зору розпізнавання ізоморфізму [3,4]. Як приклад наведемо пару неізоморфних графів, що мають однакове значення ваг в інваріантах реберних розрізів (рис. 1).

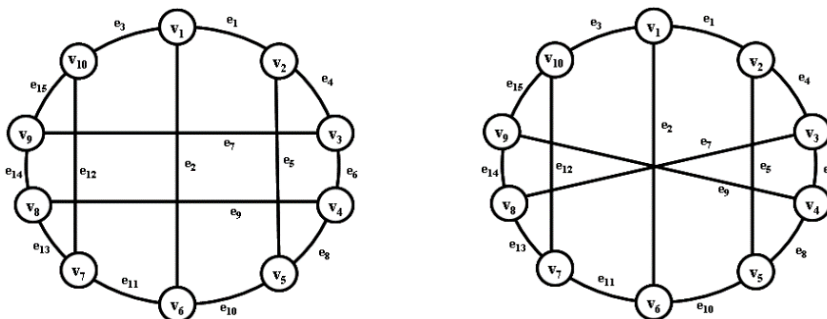


Рис. 1 – Діаграми неізоморфних графів G_1 (ліворуч) та G_2 (праворуч)

Вектор ваг ребер графа: $F_{es}(G_1) = (12, 12, 12, 12, 12, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14)$; вектор ваг вершин графа: $F_{vs}(G_1) = (40, 40, 40, 40, 40, 40, 40, 40, 40, 40)$; вектор ваг ребер графа: $F_{es}(G_2) = (12, 12, 12, 12, 12, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14)$; вектор ваг вершин графа: $F_{vs}(G_2) = (40, 40, 40, 40, 40, 40, 40, 40, 40, 40)$.

Однак кількість ізометричних циклів у графах G_1 та G_2 різна. Відповідно, різні й спектри реберних циклів графів. Кількість ізометричних циклів у графі $G_1 = 7$. Ізометричні цикли у графі G_1 у вигляді ребер:

$$c_1 = \{e_1, e_2, e_5, e_{10}\}; c_2 = \{e_1, e_3, e_4, e_7, e_{15}\}; c_3 = \{e_2, e_3, e_{11}, e_{13}\};$$

$$c_4 = \{e_4, e_5, e_6, e_8\};$$

$$c_5 = \{e_6, e_7, e_9, e_{14}\}; c_6 = \{e_8, e_9, e_{10}, e_{11}, e_{12}\}; c_7 = \{e_{12}, e_{13}, e_{14}, e_{15}\}.$$

Вектор ваг ребер для спектру реберних циклів: $F_{ec}(G_1) = (6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7)$; вектор ваг вершин: $F_{vc}(G_1) = (20, 20, 20, 20, 20, 20, 20, 20, 20, 20)$. Кількість ізометричних циклів у графі $G_2 = 15$. Ізометричні цикли у графі G_2 , подані у вигляді ребер:

$$c_1 = \{e_1, e_2, e_5, e_{10}\}; c_2 = \{e_1, e_3, e_4, e_7, e_{12}, e_{13}\}; c_3 = \{e_1, e_3, e_4, e_6, e_9, e_{15}\};$$

$$c_4 = \{e_1, e_3, e_4, e_7, e_{14}, e_{15}\}; c_5 = \{e_2, e_3, e_{11}, e_{13}\}; c_6 = \{e_1, e_3, e_5, e_8, e_9, e_{15}\};$$

$$c_7 = \{e_1, e_2, e_4, e_7, e_{11}, e_{12}\}; c_8 = \{e_4, e_5, e_6, e_8\}; c_9 = \{e_6, e_7, e_9, e_{14}\};$$

$$c_{10} = \{e_4, e_5, e_7, e_{10}, e_{11}, e_{12}\}; c_{11} = \{e_8, e_9, e_{10}, e_{11}, e_{12}, e_{14}\};$$

$$c_{12} = \{e_2, e_3, e_8, e_9, e_{10}, e_{15}\};$$

$$c_{13} = \{e_8, e_9, e_{10}, e_{11}, e_{13}, e_{15}\}; c_{14} = \{e_6, e_7, e_8, e_{10}, e_{11}, e_{12}\};$$

$$c_{15} = \{e_{12}, e_{13}, e_{14}, e_{15}\}.$$

Вектор ваг ребер для спектру реберних циклів: $F_{ec}(G_2) = (8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8)$ вектор ваг вершин: $F_{vc}(G_2) = (24, 24, 24, 24, 24, 24, 24, 24, 24, 24)$.

Інваріант реберних циклів також не може служити повним

інваріантом для розпізнавання ізоморфізму графів [4]. Як приклад можна навести два неізоморфні графи G_3 і G_4 , що мають рівне значення параметрів в інваріанті реберних циклів.

Кількість ізометричних циклів у графі $G_3 = 17$; $Fec(G_3) \& Fvc(G_3) = (4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,11,11,12,12,13,13,13,13,15,15,17,17) \& (14,14,14,14,28,28,30,30,34,34,36,36,44,44,44,44)$.

Кількість ізометричних циклів у графі $G_4 = 17$; $Fec(G_4) \& Fvc(G_4) = (4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,11,11,12,12,13,13,13,13,15,15,17,17) \& (14,14,14,14,28,28,30,30,34,34,36,36,44,44,44,44)$.

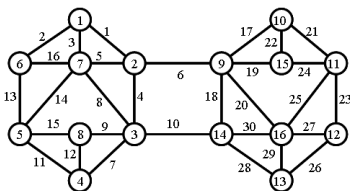


Рис. 2 – Граф G_3

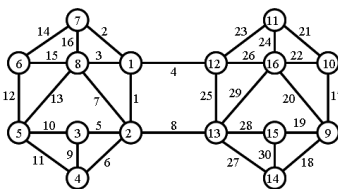


Рис. 3 – Граф G_4

Однак неізоморфні графи G_3 і G_4 відрізняються значеннями ваг елементів графа в спектрах реберних розрізів.

Вектор ваг ребер графа G_3 :

$Fes(G_3) = (4,4,12,12,12,12,13,13,14,14,14,14,15,15, 15,15,18,18,18,18,19,19,19,19,19,19,21,21,23,23) = (2 \times 4, 4 \times 12, 2 \times 13, 4 \times 14, 4 \times 15, 4 \times 18, 6 \times 19, 2 \times 21, 2 \times 23)$; вектор ваг вершин графа G_3 : $Fvs(G_3) = (37,37,37,37,37,37,45,45,55,55,80,80,81,81,100,100) = (6 \times 37, 2 \times 45, 2 \times 55, 2 \times 80, 2 \times 81, 2 \times 100)$; вектор ваг ребер графа G_4 : $Fes(G_4) = (4,4,12,12,12,12,13,13,14,14,14,14,15,15,15,15,16,17, 17,18,18,19,19,20,20,20,20,21,21,28) = (2 \times 4, 4 \times 12, 2 \times 13, 4 \times 14, 4 \times 15, 1 \times 16, 2 \times 17, 2 \times 18, 2 \times 19, 4 \times 20, 2 \times 21, 1 \times 28)$; вектор ваг вершин графа G_4 : $Fvs(G_4) = (37,37,38,38,38,38,44,44,55,55,72,72,80,80,108,108) = (2 \times 37, 4 \times 38, 2 \times 44, 2 \times 55, 2 \times 72, 2 \times 80, 2 \times 108)$.

Таким чином, якщо значення ваг елементів графа в спектрі реберних розрізів неізоморфних графів збігаються, то, можливо, не збігаються значення ваг елементів у спектрі реберних циклів графів. І навпаки, якщо значення ваг елементів у спектрі реберних циклів неізоморфних графів збігаються, то можуть не збігатися значення ваг елементів у спектрі реберних розрізів графів. Відповідно до

такої логіки міркувань, об'єднаємо інваріанти, засновані на спектрі реберних розрізів графа і на спектрі реберних циклів графа.

Означення 1. *Інтегральним інваріантом графа* називається результат об'єднання інваріанту, заснованого на спектрі реберних розрізів графа, та інваріанту, заснованого на спектрі реберних циклів графа.

Як приклад розглянемо інтегральний інваріант для графа G_4 :

$$F_{es}(G_4) \& F_{vs}(G_4) \& F_{ec}(G_4) \& F_{vc}(G_4) = (8 \times 4,10 \times 5,2 \times 11,2 \times 12,4 \times 13,2 \times 15,2 \times 17) \& (4 \times 14,2 \times 28,2 \times 30,2 \times 34,2 \times 36,4 \times 44) \& (2 \times 4,4 \times 12,2 \times 13,4 \times 14,4 \times 15,1 \times 16,2 \times 17,2 \times 18,2 \times 19,4 \times 20,2 \times 21,1 \times 28) \& (2 \times 37,4 \times 38,2 \times 44,2 \times 55,2 \times 72,2 \times 80,2 \times 108).$$

Відповімо на таке запитання: чи існують однакові значення ваг елементів в інтегральному інваріанті для двох неізоморфних неорієнтованих несепабельних графів? Відомо, що підпростір циклів $C(G)$ і підпростір розрізів $S(G)$ належать простору суграфів графа. Базис підпростору розрізів $S(G)$ завжди можна скласти з елементів множини центральних розрізів графа з потужністю рівною рангу графа. У свою чергу, базис підпростору циклів $C(G)$ завжди можна скласти з елементів множини ізометричних циклів з потужністю, що дорівнює цикломатичному числу графа. Наведемо деякі теореми, що встановлюють зв'язок між підпростором циклів та підпростором розрізів.

Теорема 1. Будь-який суграф у підпросторі циклів графа G має парне число спільних ребер із будь-яким суграфом у підпросторі розрізів графа G [2].

Теорема 2. Суграф графа G належить підпростору циклів графа G , якщо він має парне число спільних ребер із будь-яким суграфом у підпросторі розрізів графа G [2].

Теорема 3. Підпростори циклів та розрізів графа ортогональні [5].

Теорема 4. Підпростори циклів $C(G)$ та розрізів $S(G)$ графа є ортогональними доповненнями тоді і тільки тоді, коли $C(G) \cap S(G)$ – нульовий вектор [5].

Нехай $C(G)$ та $S(G)$ – ортогональні доповнення. Це означає, що кожен вектор у просторі суграфів \mathcal{L}_G графа G можна представити кільцевою сумою $w_i + w_j$, де вектор w_i належить простору $C(G)$, а вектор w_j – простору $S(G)$. Іншими словами, кожен суграф графа G можна представити кільцевою сумою двох суграфів, один з яких належить підпростору циклів, а інший – підпростору розрізів.

Зокрема, сам граф G може бути представлений у такий самий спосіб.

Припустимо, що $C(G)$ та $S(G)$ не є ортогональними доповненнями. Тоді, очевидно, існує такий суграф, який не можна представити як кільцеву суму суграфів у просторах $C(G)$ та $S(G)$. Якщо простір \mathcal{L}_G розкладається в пряму суму підпросторів циклів і розрізів, то підпростори циклів і розрізів тісно пов'язані між собою і є цілісною системою. Тобто, неможливо побудувати простір суграфів \mathcal{L}_G використовуючи підпростір розрізів одного графа і підпростір циклів іншого графа. Якщо графи G і H ізоморфні, то повинні існувати ізоморфні підпростори розрізів та ізоморфні підпростори циклів цих графів, які породжують елементи простору суграфів \mathcal{L}_G і \mathcal{L}_H . Як приклад, що демонструє утворення прямої суми просторів, розглянемо граф G^* , представлений на рис. 4.

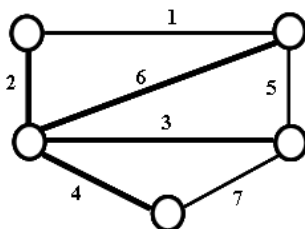


Рис. 4 – Граф G^*

Фундаментальна система циклів графа:

$$c_1^* = (0,1,0,0,0,1,1) = \{1,2,6\}; c_2^* = (0,1,1,0,1,0,0) = \{3,5,6\};$$

$$c_3^* = (1,0,0,1,1,0,0) = \{3,4,7\}.$$

Фундаментальна система розрізів графа:

$$s_1^* = (0,0,0,0,0,1,1) = \{1,2\}; s_2^* = (1,0,1,0,1,0,0) = \{3,5,7\};$$

$$s_3^* = (1,0,0,1,0,0,0) = \{4,7\}; s_4^* = (0,1,1,0,0,0,1) = \{1,5,6\}.$$

Утворимо підпростір квазіциклів $C(G^*)$:

$$c_1 = c_1^* = (0,1,0,0,0,1,1) = \{1,2,6\}; c_2 = c_2^* = (0,1,1,0,1,0,0) = \{3,5,6\};$$

$$c_3 = c_3^* = (1,0,0,1,1,0,0) = \{3,4,7\}; c_4 = c_1^* \oplus c_2^* = (0,0,1,0,1,1,1) = \{1,2,3,5\};$$

$$c_5 = c_1^* \oplus c_3^* = (1,1,0,1,1,1,1) = \{1,2,3,4,6,7\};$$

$$c_6 = c_2^* \oplus c_3^* = (1,1,1,1,0,0,0) = \{4,5,6,7\};$$

$$c_7 = c_1^* \oplus c_2^* \oplus c_3^* = (1,0,1,1,0,1,1) = \{1,2,4,5,7\}.$$

Утворюємо підпростір розрізів $S(G^*)$:

$$s_1 = s_1^* = (0,0,0,0,0,1,1) = \{1,2\}; s_2 = s_2^* = (1,0,1,0,1,0,0) = \{3,5,7\};$$

$$\begin{aligned}
s_3 &= s_3^* = (1,0,0,1,0,0,0) = \{4,7\}; s_4 = s_4^* = (0,1,1,0,0,0,1) = \{1,5,6\}; \\
s_5 &= s_1^* \oplus s_2^* = (1,0,1,0,1,1,1) = \{1,2,3,5,7\}; \\
s_6 &= s_1^* \oplus s_3^* = (1,0,0,1,0,1,1) = \{1,2,4,7\}; \\
s_7 &= s_1^* \oplus s_4 = (0,1,1,0,0,1,0) = \{2,5,6\}; \\
s_8 &= s_2^* \oplus s_3^* = (0,0,1,1,1,0,0) = \{3,4,5\}; \\
s_9 &= s_2^* \oplus s_4^* = (1,1,0,0,1,0,1) = \{1,3,6,7\}; \\
s_{10} &= s_3 \oplus s_4^* = (1,1,1,1,0,0,1) = \{1,4,5,6,7\}; \\
s_{11} &= s_1^* \oplus s_2^* \oplus s_3^* = (0,0,1,1,1,1,1) = \{1,2,3,4,5\}; \\
s_{12} &= s_1^* \oplus s_2^* \oplus s_4^* = (1,1,0,0,1,1,0) = \{2,3,6,7\}; \\
s_{13} &= s_1^* \oplus s_3^* \oplus s_4^* = (1,1,1,1,0,1,0) = \{2,4,5,6,7\}; \\
s_{14} &= s_2^* \oplus s_3^* \oplus s_4^* = (0,1,0,1,1,0,1) = \{1,3,4,6\}; \\
s_{15} &= s_1^* \oplus s_2^* \oplus s_3^* \oplus s_4^* = (0,1,0,1,1,1,0) = \{2,3,4,6\}.
\end{aligned}$$

Складемо елементи прямої суми цих двох підпросторів $S(G^*)$ та $C(G^*)$:

$$\begin{aligned}
\alpha_0 &= c_0 \oplus s_0 = \emptyset; \alpha_1 = s_{10} \oplus c_6 = \{1\}; \alpha_2 = s_{13} \oplus c_6 = \{2\}; \\
\alpha_3 &= s_1 = \{1,2\}; \alpha_4 = s_3 \oplus c_3 = \{3\}; \alpha_5 = s_4 \oplus c_2 = \{1,3\}; \\
\alpha_6 &= s_7 \oplus c_2 = \{2,3\}; \alpha_7 = s_6 \oplus c_3 = \{1,2,3\}; \alpha_8 = s_{11} \oplus c_4 = \{4\}; \\
\alpha_9 &= s_{12} \oplus c_5 = \{1,4\}; \alpha_{10} = s_9 \oplus c_5 = \{2,4\}; \alpha_{11} = s_8 \oplus c_4 = \{1,2,4\}; \\
\alpha_{12} &= s_5 \oplus c_7 = \{3,4\}; \alpha_{13} = s_{15} \oplus c_1 = \{1,3,4\}; \alpha_{14} = s_{14} \oplus c_1 = \{2,3,4\}; \\
\alpha_{15} &= s_2 \oplus c_7 = \{1,2,3,4\}; \alpha_{16} = s_6 \oplus c_7 = \{5\}; \alpha_{17} = s_7 \oplus c_1 = \{1,5\}; \\
\alpha_{18} &= s_4 \oplus c_1 = \{2,5\}; \alpha_{19} = s_3 \oplus c_7 = \{1,2,5\}; \alpha_{20} = s_1 \oplus c_4 = \{3,5\}; \\
\alpha_{21} &= s_{13} \oplus c_5 = \{1,3,5\}; \alpha_{22} = s_{10} \oplus c_5 = \{2,3,5\}; \alpha_{23} = c_4 = \{1,2,3,5\}; \\
\alpha_{24} &= s_2 \oplus c_3 = \{4,5\}; \alpha_{25} = s_{14} \oplus c_2 = \{1,4,5\}; \alpha_{26} = s_{15} \oplus c_2 = \{2,4,5\}; \\
\alpha_{27} &= s_5 \oplus c_3 = \{1,2,4,5\}; \alpha_{28} = s_8 = \{3,4,5\}; \alpha_{29} = s_9 \oplus c_6 = \{1,3,4,5\}; \\
\alpha_{30} &= s_{12} \oplus c_6 = \{2,3,4,5\}; \alpha_{31} = s_{11} = \{1,2,3,4,6\}; \alpha_{32} = s_1 \oplus c_1 = \{6\}; \\
\alpha_{33} &= s_{13} \oplus c_7 = \{1,6\}; \alpha_{34} = s_{10} \oplus c_7 = \{2,6\}; \alpha_{35} = c_1 = \{1,2,6\}; \\
\alpha_{36} &= s_6 \oplus c_5 = \{3,6\}; \alpha_{37} = s_7 \oplus c_4 = \{1,3,6\}; \alpha_{38} = s_4 \oplus c_4 = \{2,3,6\}; \\
\alpha_{39} &= s_3 \oplus c_5 = \{2,4,5\}; \alpha_{40} = s_8 \oplus c_2 = \{4,6\}; \alpha_{41} = s_9 \oplus c_3 = \{1,4,6\}; \\
\alpha_{42} &= s_{12} \oplus c_3 = \{2,4,6\}; \alpha_{43} = s_{11} \oplus c_2 = \{1,2,4,6\}; \alpha_{44} = s_2 \oplus c_6 = \{3,4,6\}; \\
\alpha_{45} &= s_{14} = \{1,3,4,6\}; \alpha_{46} = s_{15} = \{2,3,4,6\}; \alpha_{47} = s_5 \oplus c_6 = \{1,2,3,4,6\}; \\
\alpha_{48} &= s_3 \oplus c_6 = \{5,6\}; \alpha_{49} = s_4 = \{1,5,6\}; \alpha_{50} = s_7 = \{2,5,6\};
\end{aligned}$$

$$\begin{aligned}
\alpha_{51} &= s_6 \oplus c_6 = \{1,2,5,6\}; \alpha_{52} = c_2 = \{3,5,6\}; \alpha_{53} = s_{10} \oplus c_3 = \{1,3,5,6\}; \\
\alpha_{54} &= s_{13} \oplus c_3 = \{2,3,5,6\}; \alpha_{55} = s_1 \oplus c_2 = \{1,2,3,5,6\}; \alpha_{56} = s_5 \oplus c_5 = \{4,5,6\}; \\
\alpha_{57} &= s_{15} \oplus c_4 = \{1,4,5,6\}; \alpha_{58} = s_{14} \oplus c_4 = \{2,4,5,6\}; \\
\alpha_{59} &= s_2 \oplus c_5 = \{1,2,4,5,6\}; \alpha_{60} = s_{11} \oplus c_1 = \{3,4,5,6\}; \\
\alpha_{61} &= s_{12} \oplus c_7 = \{1,3,4,5,6\}; \alpha_{62} = s_9 \oplus c_7 = \{2,3,4,5,6\}; \\
\alpha_{63} &= s_8 \oplus c_1 = \{1,2,3,4,5,6\}; \alpha_{64} = s_5 \oplus c_4 = \{7\}; \alpha_{65} = s_{15} \oplus c_5 = \{1,7\}; \\
\alpha_{66} &= s_{14} \oplus c_5 = \{2,7\}; \alpha_{67} = s_2 \oplus c_4 = \{1,2,7\}; \alpha_{68} = s_{11} \oplus c_7 = \{3,7\}; \\
\alpha_{69} &= s_{12} \oplus c_1 = \{1,3,7\}; \alpha_{70} = s_9 \oplus c_1 = \{2,3,7\}; \alpha_{71} = s_8 \oplus c_7 = \{1,2,3,7\}; \\
\alpha_{72} &= s_3 = \{4,7\}; \alpha_{73} = s_4 \oplus c_6 = \{1,4,7\}; \alpha_{74} = s_7 \oplus c_6 = \{2,4,7\}; \\
\alpha_{75} &= s_6 = \{1,2,4,7\}; \alpha_{76} = c_3 = \{3,4,7\}; \alpha_{77} = s_{10} \oplus c_2 = \{1,3,4,7\}; \\
\alpha_{78} &= s_{13} \oplus c_2 = \{2,3,4,7\}; \alpha_{79} = s_1 \oplus c_3 = \{1,2,3,4,7\}; \alpha_{80} = s_8 \oplus c_3 = \{5,7\}; \\
\alpha_{81} &= s_9 \oplus c_2 = \{1,5,7\}; \alpha_{82} = s_{12} \oplus c_2 = \{2,5,7\}; \alpha_{83} = s_{12} \oplus c_3 = \{1,2,5,7\}; \\
\alpha_{84} &= s_2 = \{3,5,7\}; \alpha_{85} = s_{14} \oplus c_6 = \{1,3,5,7\}; \alpha_{86} = s_{15} \oplus c_6 = \{2,3,5,7\}; \\
\alpha_{87} &= s_5 = \{1,2,3,5,7\}; \alpha_{88} = s_1 \oplus c_7 = \{4,5,7\}; \alpha_{89} = s_{13} \oplus c_1 = \{1,4,5,7\}; \\
\alpha_{90} &= s_{10} \oplus c_1 = \{2,4,5,7\}; \alpha_{91} = c_7 = \{1,2,4,5,7\}; \alpha_{92} = s_6 \oplus c_4 = \{3,4,5,7\}; \\
\alpha_{93} &= s_7 \oplus c_5 = \{1,3,4,5,7\}; \alpha_{94} = s_4 \oplus c_5 = \{2,3,4,5,7\}; \\
\alpha_{95} &= s_3 \oplus c_4 = \{1,2,3,4,5,7\}; \alpha_{96} = s_2 \oplus c_2 = \{6,7\}; \alpha_{97} = s_{14} \oplus c_3 = \{1,6,7\}; \\
\alpha_{98} &= s_{15} \oplus c_3 = \{2,6,7\}; \alpha_{99} = s_5 \oplus c_2 = \{1,2,6,7\}; \alpha_{100} = s_8 \oplus c_6 = \{3,6,7\}; \\
\alpha_{101} &= s_9 = \{1,3,6,7\}; \alpha_{102} = s_{12} = \{2,3,6,7\}; \alpha_{103} = s_{11} \oplus c_6 = \{1,2,3,6,7\}; \\
\alpha_{104} &= s_6 \oplus c_1 = \{4,6,7\}; \alpha_{105} = s_7 \oplus c_7 = \{1,4,6,7\}; \alpha_{106} = s_4 \oplus c_7 = \{2,4,6,7\}; \\
\alpha_{107} &= s_3 \oplus c_1 = \{1,2,4,6,7\}; \alpha_{108} = s_1 \oplus c_5 = \{3,4,6,7\}; \\
\alpha_{109} &= s_{13} \oplus c_4 = \{1,3,4,6,7\}; \alpha_{110} = s_{10} \oplus c_4 = \{2,3,4,6,7\}; \\
\alpha_{111} &= c_5 = \{1,2,3,4,6,7\}; \alpha_{112} = s_{11} \oplus c_5 = \{5,6,7\}; \alpha_{113} = s_{12} \oplus c_4 = \{1,5,6,7\}; \\
\alpha_{114} &= s_9 \oplus c_4 = \{2,5,6,7\}; \alpha_{115} = s_8 \oplus c_5 = \{1,2,5,6,7\}; \\
\alpha_{116} &= s_5 \oplus c_1 = \{3,5,6,7\}; \alpha_{117} = s_{15} \oplus c_7 = \{1,3,5,6,7\}; \\
\alpha_{118} &= s_{14} \oplus c_7 = \{2,3,5,6,7\}; \alpha_{119} = s_2 \oplus c_1 = \{1,2,3,5,6,7\}; \\
\alpha_{120} &= c_6 = \{4,5,6,7\}; \alpha_{121} = s_{10} = \{1,4,5,6,7\}; \alpha_{122} = s_{13} = \{2,4,5,6,7\}; \\
\alpha_{123} &= s_1 \oplus c_6 = \{1,2,4,5,6,7\}; \alpha_{124} = s_3 \oplus c_2 = \{3,4,5,6,7\}; \\
\alpha_{125} &= s_4 \oplus c_3 = \{1,3,4,5,6,7\}; \alpha_{126} = s_7 \oplus c_3 = \{2,3,4,5,6,7\}; \\
\alpha_{127} &= s_6 \oplus c_2 = \{1,2,3,4,5,6,7\}.
\end{aligned}$$

Сума цих підпросторів є простір суграфів: $\dim \mathfrak{L}_G = \dim C + \dim S = 3+4=7$.

Означення 2. Підпростори розрізів графів G і H вважатимемо ізоморфними, якщо ізоморфні його суграфи (розрізи).

Означення 3. Підпростори циклів графів G і H вважатимемо ізоморфними, якщо ізоморфні його суграфи (цикли).

Таким чином, для графів, які не мають елементів, що перетинаються у підпросторах циклів і розрізів, можна стверджувати наступне.

Лема 1. Несепарабельні неорієнтовані графи G і H , що мають однакову кількість вершин і ребер і не мають елементів, що перетинаються в підпросторах циклів і розрізів, ізоморфні тоді і тільки тоді, коли ізоморфні їх суграфи простору суграфів \mathcal{L}_G і \mathcal{L}_H .

Доведення. Необхідність. Якщо ізоморфні графи G і H , існує бінарна відповідність між суграфами просторів суграфів \mathcal{L}_G і \mathcal{L}_H .

Достатність. Якщо ізоморфні суграфи просторів суграфів \mathcal{L}_G і \mathcal{L}_H графів G і H , то ізоморфні суграфи підпросторів циклів і розрізів графів G і H , тому що кожен суграф підпросторів циклів і розрізів може бути утворений як результат розкладання прямої суми просторів \mathcal{L}_G і \mathcal{L}_H . Оскільки суграфи підпросторів ізоморфні, то ізоморфні базиси підпросторів розрізів та циклів. *Лема доведена.*

Таким чином, у разі відсутності перетинів елементів у підпросторі циклів та підпросторі розрізів, інтегральний інваріант графа однозначно характеризує єдине представлення графа. Отже, для графів, що не мають елементів, що перетинаються в підпросторі циклів і підпросторі розрізів, інтегральний інваріант є повним інваріантом графа. Слід зазначити, що в цій парадигмі для графів, що мають елементи, які перетинаються, в підпросторі циклів і підпросторі розрізів, інтегральний інваріант не може вважатися повним. Тому розглянемо докладніше структуру цифрового інваріанту реберного графа $L(G)$. Цифровий інваріант $L(G)$ є повним інваріантом графа G , оскільки побудований на підставі теореми Вітні про ізоморфізм графів. Цифровий інваріант складається з наступних множин: множини трійок, що характеризують центральні розрізи графа G , множини ізометричних циклів графа G , множини суграфів графа G довжиною чотири. Зауважимо, що базові реберні розрізи будуються на основі множини центральних розрізів. Множина суграфів довжиною чотири є частина простору суграфів, яка може бути утворена лінійною комбінацією суграфів підпросторів циклів і розрізів графа. Тому можна припустити, що в загальному випадку інтегральний інваріант є іншою формою запису цифрового інваріанту реберного графа. Задля перевірки припущення розглянемо граfi G_3 і G_4 . Відмінність ваг ребер у

спектрі реберних розрізів видно вже на етапі формування базових реберних розрізів за однакового значення ваг ребер у спектрі реберних циклів. Вектор ваг ребер рівня 1 графа G_3 : $(4 \times 4,8 \times 5,8 \times 6,8 \times 7,2 \times 8)$; вектор ваг ребер рівня 1 графа G_4 : $(4 \times 4,8 \times 5,9 \times 6,6 \times 7,3 \times 8)$. Ця відмінність і визначає різницю ваг ребер і вершин у цифрових інваріантах реберних графів. Вектор ваг ребер у цифровому інваріанті: $F_{eL}(G_3) = (2 \times 4,2 \times 5,2 \times 6,6 \times 7,8 \times 10,2 \times 11,4 \times 12,2 \times 13,2 \times 16)$; вектор ваг вершин у цифровому інваріанті: $F_{vL}(G_3) = (6 \times 21,2 \times 23,2 \times 33,2 \times 41,2 \times 58,2 \times 62)$; вектор ваг ребер у цифровому інваріанті: $F_{eL}(G_4) = (2 \times 4,2 \times 5,2 \times 6,7 \times 7,6 \times 10,2 \times 11,4 \times 12,3 \times 13,2 \times 16)$; вектор ваг вершин у цифровому інваріанті: $F_{vL}(G_4) = (6 \times 21,2 \times 23,2 \times 33,2 \times 38,2 \times 61,2 \times 62)$. Якщо використовувати аналогію між цифровим інваріантом реберного графа і інтегральним інваріантом графа, то можна встановити ступінь участі елементів спектра реберних розрізів і елементів спектра реберних циклів у структурі інваріантів.

Означення 4. Інваріантом реберних розрізів графа $IS(G)$ називається склад ваг ребер і вершин, що визначаються першими двома рівнями спектра реберних розрізів графа.

Означення 5. Інваріантом реберних циклів графа $IC(G)$ називається склад ваг ребер і вершин, що визначаються першим рівнем спектра реберних циклів графа.

З вищевикладеного, інтегральний інваріант графа можна розглядати як повний інваріант графа.

Література

1. Ємелічев В. Лекції з теорії графів. М. : Наука, 1990. 384 с.
2. Зиков А. Теорія кінцевих графів. 1990. 384 с.
3. Kurapov S., Davidovsky M. Computational methods for determining graph invariants. *INJOIT*. 2021. Vol. 9, no. 2. P. 1–8.
4. Курапов С., Давидовський М. Алгоритмічні методи кінцевих дискретних структур. Запоріжжя ЗНУ, 2021. 214 с.
5. Свами М., Тхуласіраман К. Графи, мережі та алгоритми. М. : Мир, 1984. 455 с.

КОМПАРАТИВНИЙ АНАЛІЗ ГРАФІЧНИХ БІБЛІОТЕК ДЛЯ ВІЗУАЛІЗАЦІЇ ГРАФІВ

Артем Литвиненко

Національний університет “Запорізька політехніка”

COMPARATIVE ANALYSIS OF GRAPH LIBRARIES FOR GRAPH VISUALIZATION

Abstract. The author conducted a comparative analysis of modern graphics libraries that provide flexibility, convenience and scaling when visualizing graphs D3.js, Matplotlib, SFML, Qt Graphics, Graphviz.

Візуалізація даних є важливим етапом аналізу та інтерпретації інформації. Однією з проблем, з якою зустрічаються дослідники, є вибір правильних інструментів для створення зображень на основі вхідних даних [1]. Попередні дослідження були сфокусовані на аналізі методів пошуку зв'язків між елементами певного набору даних. Наступний крок - вибір графічної бібліотеки для зчитування та візуалізації даних за допомогою графів. Вибір правильної бібліотеки є важливим кроком, оскільки він впливає на ефективність та якість візуалізації. У даному контексті розглядаються сучасні графічні бібліотеки, які забезпечують гнучкість, зручність і масштабування при візуалізації графів D3.js, Matplotlib, SFML, Qt Graphics, Graphviz..

Метою роботи є аналіз інструментів у вигляді графічних бібліотек для зчитування вхідних даних та побудови якісного зображення. У якості тестових вхідних даних виступатиме текстовий файл, всередині якого зберігається масив вхідних даних, які матимуть в собі 2 розділи даних. Першим розділом вхідних даних є список елементів з унікальним числовим ідентифікатором та назвою. Список елементів зберігатиметься під ключем “nodes”. Другий розділ — набір інструкцій, за допомогою яких буде побудовано зв'язки між елементами та зберігатиметься під ключем “edges”. Всередині ключа зберігається масив даних, в якому описані пари елементів для відображення зв'язків. У якості ідентифікації використовуються числові значення елементів, які описані у масиві “nodes” у полі під назвою “id”. Отже, якщо масив має вигляд {"from": 1, "to": 2}, то це означатиме зв'язок між елементами з ідентифікаторами 1 та 2. Згідно з описаною інструкцією, вхідні дані матимуть такий вигляд:

```
{ "nodes": [  
  {"id": 1, "label": "Test Node 1"},
```

```

        {"id": 2, "label": "Test Node 2"},
        .....
    ],
    "edges": [
        {"from": 1, "to": 2},
        {"from": 1, "to": 3},
        .....
    ]
}

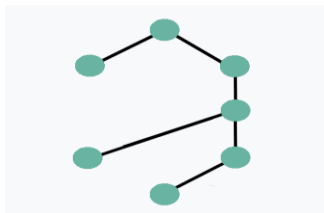
```

У роботі було досліджено та налаштовано такі бібліотеки, як D3.js, Matplotlib, SFML, Qt Graphics, Graphviz. Кожна з бібліотек вимагала окремих налаштувань робочого середовища через те, що кожен бібліотеку було реалізовано різними мовами програмування.

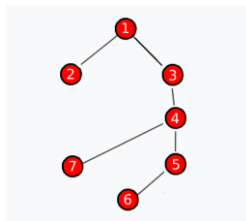
Переваги і недоліки кожної з бібліотек, можна узагальнити:

1. Бібліотека *D3.js* (рис.1,а) потребує базових знань з налаштування веб-серверу через реалізацію бібліотеки за допомогою мови JavaScript, яка взаємодіє лише з браузерами [2]. Незважаючи на це, сама бібліотека має зрозумілу документацію, де описані розповсюджені випадки та багато прикладів. Методи для використання та побудови зображення є досить простими тож бібліотека підійде для початківців. Але, варто зауважити, що дане рішення не підійде для використання з великим масивом вхідних даних через обмеження налаштування веб-серверу.

2. Бібліотека *matplotlib* (рис.1,б) є доволі популярною бібліотекою для Python, яка забезпечує простоту використання та підходить для створення складних візуалізацій. Для роботи потребує базових знань мов Python. Також має багато прикладів, які можна запозичити для реалізації власних ідей [3]. Але у випадку використання для побудови веб-іграшок графів виникають обмеження у вигляді масштабування та зручності розташування елементів, особливо у випадках великої кількості елементів.



a)

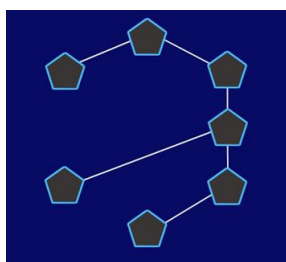


b)

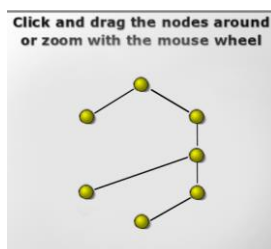
Рис.1 – Демонстрація роботи бібліотеки а) *D3.js*, б) *matplotlib*

3. Бібліотека *SFML* (рис.2,a) є потужною бібліотекою, адже вона реалізована мовою C++, що дозволяє максимально ефективно розпоряджатися ресурсами комп'ютера. Сама бібліотека є ігровим двигуном, але завдяки базовому набору інструментів та наявності реалізованих функцій для роботи з фігурами з'являється можливість створювати будь-які зображення та візуальні рішення, одним з яких є граф [4]. Також сама бібліотека дає повний контроль для реалізації будь-яких ідей, адже логіка побудування програмується власноруч [5]. Також демонструється наявність обширної документації та багато прикладів. Але, вагомим мінусом є необхідність знання самої мови C++, яка є доволі складною та при неправильному розподіленні ресурсів можна перевантажити систему. Тож, дану бібліотеку варто обирати лише досвідченим користувачам.

4. Бібліотека *QtGraphics* (рис.2,b) являється схожою на *SFML* бібліотекою, яка також реалізована за допомогою мови C++. Вона також дає можливість розпоряджатися ресурсами комп'ютера, але одразу після встановлення в середовище надає більш широкий спектр можливостей та інструментів. Завдяки прозорій документації та великій кількості прикладів, розміщених на існуючих репозиторіях, дана бібліотека легко дозволяє реалізувати ідеї навіть початківцям [6,7]. Тож, дана бібліотека є більш універсальною, що робить її універсальним вибором для досліджень будь-якої складності.



a)



b)

Рис.2 – Демонстрація роботи бібліотеки a) *QtGraphics*, b) *SFML*

5. Бібліотека *Graphviz* (рис.3) є однією з найстарших бібліотек, адже вона реалізована мовою C. Це також дає можливість повністю керувати ресурсами під час побудови зображення. Але, важливим моментом, на який варто звернути увагу, що дана бібліотека

розрахована саме на побудову будь-яких графів, тож одразу після встановлення ми маємо доступ до заготовлених алгоритмів розміщення елементів графу, зручні інструменти для створення самих елементів графу, починаючи від форми елемента, закінчуючи деталями та інформацією [8]. Крім цього, Graphviz має ряд бібліотек для інтеграції з іншими мовами програмування, такими як Python, Java, та C++[9].

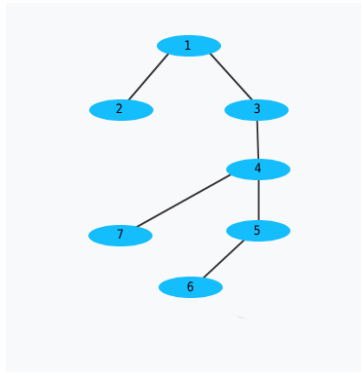


Рис.3 – Демонстрація роботи бібліотеки Graphviz

Це дозволяє використовувати Graphviz у різноманітних середовищах розробки та інтегрувати його в існуючі проекти для візуалізації графічних структур, що робить дану бібліотеку універсальним та потужним рішенням.

Завдяки отриманим зображенням можна підсумувати, що за допомогою будь-якої бібліотеки можна реалізувати побудову графу з малою кількістю елементів та розмістити всі елементи для зручного зчитування інформації. Тож, вибір бібліотеки залежить більше від кількості елементів графу [10]. Якщо ж це, наприклад, невелика кількість елементів та в бібліотеці не реалізовано масштабування або зручне розташування, то можна обрати *D3.js* або *matplotlib*. Якщо ж розглядається рішення для великого об'єму даних, то для реалізації необхідний більший контроль над ресурсами та більше можливостей для розташування елементів, вибір кольорів, форм тощо. Для реалізації рішень такого типу краще скористатися бібліотеками *SFML*, *QtGraphics* або *Graphviz*, кожна з яких дає можливість реалізувати гнучке рішення та надає максимальний контроль.

ЛІТЕРАТУРА

1. Lanum C. Visualizing Graph Data. O'Reilly Media, Inc., 2016. 344с.
2. Мюррей С. Інтерактивна візуалізація даних для Інтернету. O'Reilly, 2013. 474 с.
3. Tosi S. Matplotlib for Python Developers. Packt Publishing, 2009. 292 с.
4. Williams, L., Booth, P. Practical Video Game Bots: Automating Game Design with SFML and Modern C++. CRC Press, 2019.
5. Moreira A., Guedes A. SFML Game Development By Example. Packt Publishing, 2015. 356 с.
6. Blanchette J., Summerfield M. C++ GUI Programming with Qt 5. Second Edition. Prentice Hall, 2016. 768 с.
7. Dalheimer, M. K., "Programming with Qt". O'Reilly Media, 2020.
8. Ellson J., Gansner E.R., Koutsofios L., North S.C., Woodhull G. Graphviz and Dynagraph - Static and Dynamic Graph Drawing Tools. Graph Drawing Software, Springer-Verlag, 2004. 127-148 с.
9. Mutton P., Graphviz - A Graph Visualization Software. *SoftVis'20 Proceedings of the 2020 ACM Symposium on Software Visualization*, 2020. 3-4 с.
10. Bondy, J. A., Murty, Graph Theory with Applications. North-Holland, 2020.

ДОСЛІДЖЕННЯ ВПЛИВУ ПЕРЕРИВАНЬ НА ВИНИКНЕННЯ АНОМАЛІЙ У ЗАДАЧАХ ПАРАЛЕЛЬНОГО УПОРЯДКУВАННЯ

Малієнко О.О., Коваленко Є.О.

Дніпровський національний університет імені Олеся Гончара

STUDY OF THE INFLUENCE OF INTERRUPTS ON THE OCCURRENCE OF ANOMALIES IN PARALLEL ORDERING PROBLEMS

Abstract. In parallel ordering problems, anomalies often arise, consisting in an unexpected increase in the length of the execution of tasks when the initial conditions are improved. This work investigates the effect of interrupts on the occurrence of such anomalies and establishes the conditions under which enabling interrupts can avoid them. An algorithm of priority dynamic

redistribution is proposed, which takes into account the priority of tasks, the possibility of interruption, and the dynamic updating of the queue of tasks to achieve the minimum length of ordering.

У задачах паралельного упорядкування дослідження часто зосереджуються на випадках виникнення аномалій [1], тобто неочікуваного збільшення значення цільової функції при покращенні початкових умов, що повинні призводити до його зменшення.

Виникає питання, пов'язане з впливом переривань на виникнення даних аномалій, і, зокрема, встановлення умов при яких дозвіл переривань дозволяє уникнути їх.

Нехай задано оргграф $G = (V, U)$, список пріоритетів $L = (i_1, \dots, i_n)$ та ваги вершин $T = (\tau_1, \dots, \tau_n)$ [2]. Вводимо додаткові початкові умови: обмеження на ширину упорядкування множиною значень h_i та дозвіл на переривання при виконанні робіт [3].

Побудоване паралельне упорядкування повинно задовольняти список пріоритетів та мати мінімальну довжину.

Твердження. Дозвіл переривань деяких завдань для випадків, коли аномалії існують, дозволяє уникнути їх виникнення за наступних умов:

1. Пріоритетність: список пріоритетів має бути побудований згідно до алгоритму побудови оптимального списку пріоритетів [4]; завдання з високим пріоритетом можуть переривати виконання завдань з нижчим пріоритетом.

Ця умова гарантує виконання більш важливих завдань у першу чергу.

2. Переривання: перерване завдання може бути продовжене на будь-якому кроці, але згідно зі списком пріоритетів.

Це дозволяє виконувати більш пріоритетні завдання на крайніх лівих допустимих місцях [5].

3. Динамічна актуалізація: на кожному кроці виконання завдання можливе переривання виконання роботи та перерозподіл робіт.

Цією умовою враховуються зміни у доступності виконавців та стані виконання завдань у реальному часі.

Опишемо алгоритм вирішення даної задачі. Алгоритм пріоритетного динамічного перерозподілу:

1. Ініціалізація: всі виконавці позначаються як вільні.

2. Призначення роботи: коли нова робота i стає доступною для виконання, вона призначається вільному виконавцю.

3. Перерозподіл: на кожному кроці виконання роботи i перевіряється доступність виконання більш пріоритетних робіт. Якщо знайдена більш пріоритетна робота j , яка може стає доступною до виконання на даному кроці, виконання роботи i переривається, а робота j призначається даному виконавцю.

У даному алгоритмі особлива увага приділяється розподілу пріоритетів та ефективному управлінню перериваннями, що дозволяє отримати упорядкування оптимальної довжини.

Проілюструємо введення додаткових умов та використання алгоритму пріоритетного динамічного перерозподілу на прикладі.

Приклад 1. Нехай задано граф G (рис.1), ширина упорядкування $h = 3$ список пріоритетів $L = (1,2,3,4,5,6,7,8,9,10,11,12,13,14)$ та час виконання завдань $T = (4,2,2,2,2,4,2,2,1,1,6,3,1,4,2)$.

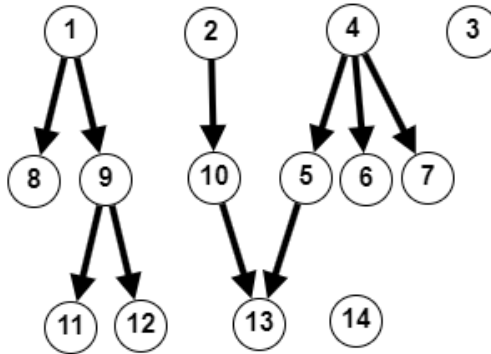


Рис.1 – Граф G

Знайдемо оптимальне упорядкування та його довжину.

Таблиця 1. Оптимальне упорядкування вершин графа G при $h = 3$

1	1	1	1	5	5	5	5	8	11	11	11
2	2	4	4	6	6	7	7	9	12	14	14
3	3	10	10	10	10	10	10	13	13	13	13

Довжина $l = 12$. Введемо множину h_i як початкову умову. Нехай множина $h_i = (4,4,4,3, \dots 3)$. Отримаємо упорядкування (табл. 2) з довжиною $l = 15$.

Таблиця 2 Оптимальне упорядкування вершин графа G при h_i

1	1	1	1	7	9	11	11	11			13	13	13	13
2	2	5	5	5	5	12	14	14						
3	3	6	6	8	10	10	10	10	10	10				
4	4	7												

Змінюємо список пріоритетів згідно до алгоритму побудови оптимального списку пріоритетів.

Знаходимо загальний час виконання усіх шляхів графа: $l_{1-8} = 5, l_{1-11} = 8, l_{1-12} = 6, l_{2-13} = 12, l_{4-13} = 10, l_{4-6} = 4, l_{4-7} = 4, l_3 = 2, l_{14} = 2$.

Оновлюємо список пріоритетів:

$$L' = (2,10,13,4,5,1,9,11,12,8,6,7,3,14).$$

Будуємо оптимальне упорядкування, не змінюючи множину h_i (табл. 3).

Таблиця 3. Оптимальне упорядкування вершин графа G при L'

2	2	10	10	10	10	10	10	10	13	13	13	13
4	4	5	5	5	5	12	8	6	14	14		
1	1	1	1	9	11	11	11	7	7			
3	3	6										

Довжина $l = 12$, тобто дорівнює довжині оптимального упорядкування та довжині критичного шляху.

Таким чином, використання алгоритму пріоритетного динамічного перерозподілу дозволяє ефективно розподіляти роботи між виконавцями, враховуючи пріоритети та можливість переривання завдань. Це мінімізує загальний час виконання робіт і запобігає виникненню аномалій, які можуть виникати внаслідок зміщення виконання більш пріоритетних завдань менш пріоритетними.

На практиці дана постановка задачі може бути застосована у різних галузях, таких як: виробничі процеси, обробка даних, логістика, управління запасами та інші сфери, де необхідно

розподіляти завдання між виконавцями з однаковими здібностями. Наприклад, у хмарних середовищах алгоритм може використовуватися для динамічного розподілу обчислювальних завдань між серверами з урахуванням їх навантаження та пріоритетів (розподілені обчислення). Або у сфері охорони здоров'я: розподіл завдань між медичними працівниками в операційних відділеннях, з урахуванням терміновості операцій та можливості переривання планових операцій у разі надзвичайних ситуацій.

Подальшого дослідження потребують питання впливу різних стратегій переривання на оптимальність значення цільової функції та ефективність алгоритму для розподілених систем.

Література

1. Graham R. L. Bounds on multiprocessing timing anomalies // *SIAM J. Appl. Math.*, 1969. V. 17. P. 416–429.

2. Челпанова О.О. Узагальнення аномальних випадків у задачах упорядкування. / О.О. Челпанова, В.А. Турчина. // *Питання прикладної математики та математичного моделювання: зб. наук. пр. Дніпро*, 2021. – Вип. 21. – С. 220-226.

3. Коваленко Є.О. Про один частковий випадок задачі паралельного упорядкування. / Є.О. Коваленко, В.А. Турчина. // *Theoretical and empirical scientific research: concept and trends: Oxford, February 2, 2024. V.6. P. 222-226.*

4. Maliienko O.O. Conditions for length stabilization in parallel ordering problems. / O.O. Maliienko, V.A. Turchyna. // *Theoretical and empirical scientific research: concept and trends: Oxford, February 2, 2024. V. 6. P. 216-221.*

5. Турчина В.А. Алгоритми перерахування всіх паралельних упорядкувань фіксованої довжини. / В.А. Турчина, Ю.С. Зозуля, А.К. Підаш. // *Питання прикладної математики та математичного моделювання: зб. наук. пр. Дніпро*, 2013. С. 256-261.

МЕТОД ВИЯВЛЕННЯ ТА НЕЙТРАЛІЗАЦІЇ МЕРЕЖІ БОТІВ У РЕКОМЕНДАЦІЙНІЙ СИСТЕМІ

Мелешко Є.В.

*Центральноукраїнський національний технічний університет,
м. Кропивницький*

METHOD FOR DETECTING AND NEUTRALIZING A BOT NETWORK IN A RECOMMENDATION SYSTEM

Abstract. Information attacks on recommendation systems are often carried out for marketing purposes to increase the ratings of the attacking party's products or to decrease the ratings of competitors' products. Also, attacks on recommendation systems may aim to spread informational influence during information confrontations, for example, for political purposes. This paper proposes a method for detecting and neutralizing a bot network in a recommendation system based on graph clustering and user behavior analysis. A series of experiments were conducted to investigate the developed method's performance, demonstrating its high accuracy.

На сьогоднішній день у сфері інформаційних технологій для просування товарів та послуг все частіше використовують так звані рекомендаційні системи – інструменти автоматичної генерації рекомендацій на основі вивчення персональних потреб користувачів веб-сайтів чи додатків [1-4]. Рекомендаційні системи застосовують, напр., такі популярні веб-сайти як Netflix, Amazon, Spotify, YouTube, Facebook тощо.

У цій роботі запропоновано метод виявлення та нейтралізації мережі ботів у рекомендаційній системі на основі графової кластеризації [5, 6] та аналізу дій користувачів.

Найчастіше інформаційні атаки на рекомендаційні системи здійснюються у маркетингових цілях з метою підвищення рейтингів товарів атакуючої сторони або зниження рейтингів товарів конкурентів. Хоча атаки на рекомендаційні системи можуть мати на меті також і поширення інформаційних впливів під час інформаційних протиборств, напр., у політичних цілях. Через соціальні мережі часто здійснюються різні інформаційні впливи [7-8], і рекомендаційні системи, як їх складова, стали однією з цілей для інформаційних атак з метою здійснення таких впливів [9-12]. Виконавши успішну атаку на рекомендаційну систему соціальної мережі або агрегатора новин, можна змінити наповнення та порядок

показу об'єктів у стрічках новин користувачам системи. Це можна використати не тільки в маркетингових, а й політичних чи шахрайських цілях. Для реалізації атак на рекомендаційні системи використовуються мережі ботів, так як тільки певна сукупність профілів у системі здатна вплинути на формування рекомендацій своїми згуртованими діями [9-12]. Переважно в існуючих дослідженнях [3, 12-16] пропонується вважати виявлення інформаційної атаки на рекомендаційну систему тотожним виявленню профілів ботів. Оскільки виявлення профілів ботів досить ресурсномістка задача, у даній роботі пропонується розділити задачу захисту рекомендаційної системи від інформаційних атак на дві частини: 1) виявлення ознак атаки та 2) виявлення і нейтралізація профілів ботів.

Виявлення атаки може бути менш ресурсозатратною задачею і полягати у відслідковуванні динаміки рейтингів об'єктів системи, усіх або тільки критично важливих з точки зору інформаційної безпеки. Це може бути реалізовано у вигляді автоматичного моніторингу змін рейтингів усіх або важливих об'єктів. Підозрілими (такими, що підлягають перевірці) можна вважати занадто швидкі або нетипові зміни у рейтингах об'єктів.

Задачу виділення бот-мережі серед профілів користувачів системи можна звести до задачі пошуку підграфу у соціальному графі рекомендаційної системи, вершинами якого будуть профілі пов'язані між собою деякими спільними діями, що мали вплив на зміну трендів рейтингів усіх або більшості об'єктів з множини ймовірних цілей інформаційної атаки [3, 17-19]. Якщо виявлена атака на рекомендаційну систему та зафіксована множина цілей ботів G , що могло бути здійснено як при візуальному спостереженні так і під час автоматизованого моніторингу системи, логічним буде дослідити профілі всіх користувачів, які вплинули на зміну трендів рейтингів об'єктів з цієї множини G , що і пропонується зробити у даній роботі для пошуку серед них профілів ботів.

Для вирішення задачі ідентифікації профілів ботів було розроблено метод заснований на графовій кластеризації та аналізі дій користувачів, з використанням коефіцієнтів «недовіри» (для реалізації використані такі програмні засоби – мова програмування Python та графова база даних Neo4j):

Етап 1. Формуємо множину підозрілих профілів користувачів S , в яку поміщаємо профілі, що ставили цільові оцінки r_i об'єктам з множини G .

Етап 2. Привласнюємо кожному користувачу з множини G

мітку *:suspicious* у базі даних та коефіцієнт недовіри, розрахований за наступною формулою:

$$k_{d,i} = \sum_{j \in G} \frac{E_{r_{t,i,j}}}{n_g}, \quad (1)$$

де $E_{r_{t,i,j}}$ – наявність цільової оцінки r_t від користувача i об'єкту j , що належить множині ймовірних цілей атаки G , приймає значення 1, якщо цільова оцінка є та 0 – при відсутності такої оцінки від користувача i об'єкту j ; n_g – кількість об'єктів у множині можливих цілей атаки G .

Етап 3. Для кожної пари користувачів i_1 та i_2 з множини S , де $k_{d,i1} \geq q$ та $k_{d,i2} \geq q$, створюємо ребро між ними з міткою *:BotNet*.

Етап 4. Виконуємо графову кластеризацію для підграфу, що містить вершини з мітками *:User* і *:suspicious* та ребра з міткою *:BotNet*. Результати роботи такої кластеризації будуть наступними – усі боти потраплять до одного великого кластеру, якщо бот-мережа одна, або до декількох великих кластерів – якщо бот-мереж декілька; аутентичні користувачі потраплять у різні кластери, кожний з таких кластерів буде містити одного користувача або невелику кількість користувачів. Можливі й випадки, коли до кластеру з ботами потрапить деяка кількість аутентичних користувачів або деяка група схожих між собою та активних аутентичних користувачів утворить окремий кластер у разі, якщо їх дії зсунули рейтинги деяких об'єктів.

Етап 5. Визначаємо найбільші кластери, що складаються з $(N_{cr} - e)$ користувачів, де N_{cr} – мінімальна кількість користувачів, що може вплинути на результати роботи рекомендаційної системи (залежить від параметрів конкретної системи), e – приблизне значення похибки при розділенні профілів користувачів на кластери. Такий кластер (або кластери) вважаємо можливою бот-мережею (бот-мережами). У користувачів, що не потрапляють до даних кластерів прибираємо ребра з міткою *:BotNet*. Користувачів, що потрапили до підграфу BotNet треба додатково перевірити, проаналізувавши статистичні характеристики їх профілів, напр., за допомогою запропонованого у попередньому розділі способу з використанням нейронних мереж [20]. Також для додаткової перевірки профілів з підграфу BotNet можна здійснити пошук у них певних характерних для ботів ознак, напр., погано заповнених анкетних даних або надзвичайно високої активності (характерні особливості ботів залежать від конкретної системи і можуть ставати відомими в процесі збору статистичних даних під час її роботи).

Одними з загальних ознак ботів для багатьох систем можуть бути: відмінність значення дисперсії оцінок та дисперсії часових інтервалів між виставленнями оцінок у профілях ботів від середньостатистичних значень відповідних дисперсій у профілях користувачів системи. Для конкретної системи ознаками ботів можуть бути: особливості реєстрації профілю, особливості наповнення профілю особистою інформацією, стиль написання та зміст коментарів, список друзів користувача тощо. Після перевірки статистичних даних окремих профілів користувачів, які потрапили у підграф BotNet, треба його скоректувати, видаливши з нього користувачів, розпізнаних за статистичними даними як аутентичні. Якщо є кластер, у якому всі користувачі визнані автентичними – слід перестати вважати його бот-мережею.

Етап 6. Коректуємо множину G після аналізу оцінок користувачів з підграфу BotNet. Перевіряємо, яким об'єктам користувачі з бот-мережі скоординовано виставляли цільові оцінки. Видаляємо з G об'єкти, які не одержували взагалі або одержали незначний процент цільових оцінок від користувачів, ідентифікованих як боти. Додаємо до множини G об'єкти, які одержали цільові оцінки від усіх (або великого проценту) ботів.

Було проведено серію експериментів для тестування розробленого методу. Набори даних для експериментів генерувалися у розробленій програмній імітаційній моделі рекомендаційної системи [21]. Формат та статистичні особливості даних генерувалися максимально наближеними до відповідних характеристик відкритого набору даних MovieLens Datasets [22]. Атаки моделювалися за допомогою популярної, випадкової та середньої моделей атак [3, 10]. У якості алгоритму графової кластеризації використовувався алгоритм Label Propagation.

У проведеній серії експериментів було згенеровано 10% ботів, усі інші користувачі системи – аутентичні. У різних експериментах у ботів була різна кількість цілей для атаки: 1, 5, 10, 15, 20, 25, 30, 35, 40 та 45 цілей. Також серед аутентичних користувачів було згенеровано 20% профілів з високим рівнем активності у рекомендаційній системі, які виставляли значно більше оцінок, ніж середньостатистичні користувачі. Об'єкти системи були згенеровані таким чином, щоб відноситися за своїми властивостями до одного з 19 кластерів. Алгоритм пошуку ботів для досліджуваної рекомендаційної системи мав наступні параметри:

– $q = 0.05$ – тобто, користувач виставив цільові оцінки не менше 5% об'єктів, які визначені підсистемою інформаційної безпеки як

ймовірні цілі атаки. Порогове значення невелике, тому що невідомо наскільки вірно розпізнані цілі атаки, можливо там багато об'єктів, у яких змінилися рейтинги природнім чином.

– Відсіювання профілів користувачів з підграфу BotNet здійснюється на основі значень дисперсії оцінок та дисперсії часових інтервалів між виставленням оцінок. Аутентичними користувачами, які помилково потрапили до підграфу BotNet, вважаються такі, дані з профілів яких відповідають правилу:

$$|D_r - D_{r,avr}| < 0.15 \text{ AND } (D_{t,r} > 72 \text{ AND } D_{t,r} < 600),$$

де D_r – дисперсія оцінок у профілі користувача, $D_{r,avr}$ – усічене середнє дисперсії оцінок у профілі користувачів системи (відсікалося 30% крайніх значень, тобто, гранично можливий процент ботів у розглядуваній системі), $D_{t,r}$ – дисперсія часових інтервалів між виставленнями оцінок. Тобто, дисперсія оцінок у профілях аутентичних користувачів несуттєво відрізняється від середньостатистичної дисперсії (атакуючий систему це значення знати не може, він може лише приблизно його оцінити, тому не може вірно відтворити дану характеристику при створенні профілів ботів). А перевірка дисперсії часових інтервалів між виставленнями оцінок має наступний сенс – слід вважати підозрілими користувачів, які роблять занадто однакові або занадто різні інтервали між виставленнями оцінок, в той же час дисперсія часових проміжків між виставленнями оцінок у автентичних користувачів знаходяться, як правило, в певному діапазоні значень.

Для оцінки якості роботи розробленого методу було обрано наступні метрики:

– Точність (Precision):

$$precision = \frac{tp}{tp+fp}, \quad (2)$$

де tp – правильно виявлений бот; fp – аутентичний користувач невірно ідентифікований як бот.

– Повнота (Recall, Sensitivity):

$$recall = \frac{tp}{tp+fn}, \quad (3)$$

де tp – правильно виявлений бот; fn – бот, який невірно ідентифікований як аутентичний користувач.

– Середньоквадратична помилка (RMSE):

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}}, \quad (4)$$

Як показала серія експериментів з табл. 6.4, точність розпізнавання ботів розробленим методом в середньому становить 0.72 для випадкової атаки, 0.81 для середньої атаки, а також 0.71 для

популярної атаки. Також було більш детально досліджено точність розпізнавання ботів розробленим методом для популярної моделі атаки. Найгірші результати одержувалися, коли ціль для атаки ботів була одна, тоді точність розробленого методу падала до 0.57 у середньому. Найвища точність спостерігалася, коли цілей атак було 25-30 шт., в такому разі вона сягала значення 0.78. При вдалій атаці ботів, що зсунула рейтинги цільових об'єктів, наявність бот-мережі та значний процент ботів з неї завжди було виявлено.

Література

1. Jones M. Recommender systems, Part 1. Introduction to approaches and algorithms. Learn about the concepts that underlie web recommendation engines. *Official Web-site of IBM company*. 2013. URL: https://www.ibm.com/developerworks/opensource/library/os-recommender1/index.html?S_TACT=105AGX99&S_CMP=CP
2. Jones M. Recommender systems, Part 2. Introducing open source engines. Explore software for building a recommendation capability. *Official Web-site of IBM company*. 2013. URL: https://www.ibm.com/developerworks/library/os-recommender2/index.html?S_TACT=105AGX99&S_CMP=CP
3. Ricci F., Rokach L., Shapira B., Kantor P.B. (Editors) Recommender Systems Handbook. *Springer*. 2011. 842 p. DOI: <https://doi.org/10.1007/978-0-387-85820-3>
4. Мелешко Є.В., Семенов С.Г., Хох В.Д. Дослідження методів побудови рекомендаційних систем в мережі Інтернет. *Системи управління, навігації та зв'язку*. 2018. Вип. 1(47). С. 131-136.
5. Мелешко Є.В. Методи кластеризації графів соціальних мереж для побудови рекомендаційних систем. *Системи управління, навігації та зв'язку*. 2019. Т. 2 (54). С. 129-134.
6. Мелешко Є.В. Дослідження засобів кластеризації графів у графовій СУБД Neo4j для виявлення співтовариств у соціальних мережах. *Збірник тез VIII Міжнародної наукової конференції «Інформація. Комунікація. Суспільство», смт. Чинадієво, 16-18 травня 2019 р. Львів: Видавництво Львівської політехніки*. 2019. С. 19-20.
7. Ulichev O.S., Meleshko Ye.V., Sawicki D., Smailova S. Computer modeling of dissemination of informational influences in social networks with different strategies of information distributors. *Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments, Wilga, Poland*. 2019. 111761T

8. Богуш В.М., Юдін О.К. Інформаційна безпека держави. "МК-Прес", Київ. 2005. 432 с.
9. Горбулін В.П., Додонов О.Г., Ланде Д.В. Інформаційні операції та безпека суспільства: загрози, протидія, моделювання: монографія. *Інтертехнологія*, Київ. 2009. 164 с.
10. Gunes I., Kaleli C., Bilge A., Polat H. Shilling attacks against recommender systems: a comprehensive survey. *Artificial Intelligence Review*. Vol. 42. 2014. P. 767-799. DOI: <https://doi.org/10.1007/s10462-012-9364-9>
11. Kaur P., Goel S. Shilling attack models in recommender system. *International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore. 2016. P. 1-5. URL: <https://ieeexplore.ieee.org/document/7824865/>
12. Kumari T., Punam B. A Comprehensive Study of Shilling Attacks in Recommender Systems. *IJCSI International Journal of Computer Science Issues*. 2017. Vol. 14, Iss. 4. URL: <https://www.ijcsi.org/articles/A-comprehensive-study-of-shilling-attacks-in-recommender-systems.php>
13. O'Mahony M.P., Hurley N.J., Silvestre G.C.M. Promoting recommendations: An attack on collaborative filtering. *DEXA, Lecture Notes in Computer Science*. 2002. Vol. 2453. P. 494-503.
14. Williams A. C., Mobasher B., Burke R. Defending recommender systems: detection of profile injection attacks. *Service Oriented Computing and Applications*. 2007. P. 157-170.
15. Zhou W., Wen J., Koh Y.S., Alam S., Dobbie G. Attack detection in recommender systems based on target item analysis. *International Joint Conference on Neural Networks (IJCNN 2014)*, Beijing. 2014. P. 332-339. URL: <https://ieeexplore.ieee.org/document/6889419>
16. Zhou W., Wen J., Qu Q., Zeng J., Cheng T. Shilling attack detection for recommender systems based on credibility of group users and rating time series. *PLoS ONE*. 2018. Vol. 13(5): e0196533. DOI: <https://doi.org/10.1371/journal.pone.0196533>
17. Мелешко Є.В. Аналіз структури соціальної мережі з точки зору інформаційної безпеки. *Збірник тез XVIII міжнародного науково-практичного семінару "Комбінаторні конфігурації та їх застосування"*, м. Кіровоград, 15-16 квітня 2016. Кіровоград: Кіровоградський національний технічний університет. 2016. С. 93-97.
18. Мелешко Є.В. Дослідження методів динамічного аналізу віртуальних соціальних мереж з точки зору інформаційної безпеки. *Матеріали Всеукраїнської науково-практичної конференції*

"Кібербезпека в Україні: правові та організаційні питання", м. Одеса, 21 жовтня 2016 р. Одеса: ОДУВС. 2016. С. 154-155.

19. Мелешко Є.В., Хох В.Д., Улічев О.С. Дослідження відомих моделей атак на рекомендаційні системи з колаборативною фільтрацією. *Системи управління, навігації та зв'язку*. 2019. № 5(57). С. 67-71.

20. Meleshko Ye., Drieiev O., Drieieva H. Method of identification bot profiles based on neural networks in recommendation systems. *Advanced Information Systems*. 2020. Vol. 4, No. 2 P. 24-28.

21. Meleshko Ye. Computer model of virtual social network with recommendation system. *Innovative technologies and scientific solutions for industries*. 2019. №2(8). P. 80-85.

Harper F.M., Konstan J.A. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiS)*. 2015. 19 p. DOI: <https://doi.org/10.1145/2827872>

ДОСЛІДЖЕННЯ МЕТОДІВ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У КІБЕРБЕЗПЕЦІ

Мелешко Є.В., Варченко І.В

*Центральноукраїнський національний технічний університет,
м. Кропивницький*

RESEARCH ON METHODS OF USING ARTIFICIAL INTELLIGENCE IN CYBERSECURITY

Abstract. This paper examines the use of artificial intelligence (AI) in cybersecurity, highlighting its advantages and disadvantages. It describes the most popular applications of AI in cybersecurity and outlines drawbacks associated with its use. The research includes a review of AI's capabilities in detecting unknown threats, managing large data volumes, and improving overall security, providing a comprehensive overview of AI's impact on cybersecurity. Additionally, the paper underscores the importance of addressing the ethical and legal considerations involved in deploying AI technologies.

Штучний інтелект (ШІ) і машинне навчання стали критично важливими технологіями в інформаційній безпеці, оскільки вони здатні швидко аналізувати мільйони подій і визначати багато різних типів загроз – від зловмисного програмного забезпечення, що

використовує вразливості нульового дня, до виявлення ризикованої поведінки, яка може призвести до вдалої фішингової атаки або завантаження шкідливого коду. Ці технології з часом навчаються, спираючись на минуле, щоб зараз ідентифікувати нові типи атак.

Завдяки використанню машинного навчання можна визначити, як виглядає нормальна поведінка в системі, і виділити будь-які випадки, які відхиляються від норми. Це може допомогти ідентифікувати потенційні атаки та виявляти системи, які не працюють належним чином і є вразливими до атак.

Навіть поведінка користувача, яка може бути проблемою, наприклад, випадковий витік даних або ексфільтрація, потенційно може бути виявлена за допомогою ШІ.

Основні області застосування ШІ у кібербезпеці:

- Виявлення та запобігання вторгненням (IDS/IPS) шляхом аналізу аномальної поведінки мережевого трафіку.

- Системи управління інформаційною безпекою (SIEM) для автоматичного аналізу логів і виявлення підозрілої активності в реальному часі.

- Управління ідентифікацією та доступом (IAM) – допомога у створенні та контролі складних моделей аутентифікації та авторизації для забезпечення безпеки доступу до ресурсів.

- Захист кінцевих точок (Endpoint Security) – виявлення та блокування шкідливого програмного забезпечення та інших загроз на кінцевих пристроях.

- Кіберрозвідка (Cyber Threat Intelligence) – ШІ може аналізувати дані з різних джерел для виявлення нових загроз і тенденцій у кіберпросторі.

- Моніторинг та аналіз мережевого трафіку для виявлення аномалій.

- Захист від DDoS-атак для забезпечення стабільної роботи комп'ютерних мереж та веб-сайтів.

Ось основні переваги та можливості використання штучного інтелекту у кібербезпеці:

1. Штучний інтелект може виявляти крім відомих й невідомі загрози. Виявлення всіх потенційних загроз для компанії може бути надзвичайно складним через постійну зміну тактики хакерів. Тому вкрай важливо застосовувати сучасні рішення, такі як штучний інтелект, для ефективного виявлення та запобігання невідомим загрозам, які можуть завдати серйозної шкоди, якщо їх

не виявити.

2. ШІ може обробляти дуже велику кількість трафіку та перевіряти на наявність потенційних загроз. Мережа компанії генерує величезну кількість трафіку, що ускладнює персоналу служби кібербезпеки перевірку вручну всіх дій на наявність потенційних загроз. Використання штучного інтелекту дозволяє автоматично сканувати та визначати приховані загрози, спрощуючи процес виявлення та покращуючи захист.

3. ШІ дозволяє реагувати у режимі реального часу і може донавчатися у процесі роботи. ШІ використовує машинне навчання та глибоке навчання, методи аналізу поведінки мережі та виявлення відхилень або інцидентів безпеки від норми. Це дозволяє негайно реагувати та посилює майбутні заходи безпеки, блокуючи потенційні загрози з подібними характеристиками. Постійний процес навчання ШІ ускладнює хакерам задачу вторгнення в систему та зменшує їх можливості по обходу існуючих захистів.

4. Пошук та тестування вразливих місць комп'ютерної мережі. Перспективним є використання ШІ в кібербезпеці для керування вразливими місцями мережі, враховуючи щоденні загрози, з якими стикаються компанії. Він аналізує існуючі заходи безпеки, щоб виявити слабкі місця, що дозволяє компаніям зосередитися на критичних завданнях безпеки. Це покращує здатність вирішувати проблеми та захищає бізнес-системи швидше, ніж персонал із кібербезпеки.

5. Прогнозування майбутніх загроз. Аналізуючи журнали подій та історичні дані про кібернапади, ШІ може прогнозувати ймовірні майбутні атаки та допомагати організаціям підготуватися до них.

6. Зменшення дублюючих та рутинних процесів. Одним із основних випадків використання штучного інтелекту в кібербезпеці є те, що він виконує монотонні та повторювані завдання безпеки, які можуть викликати перевантаження персоналу кібербезпеки. Він регулярно виявляє та запобігає основним загрозам безпеці та проводить ретельний аналіз для виявлення потенційних прогалин у безпеці. Завдяки штучному інтелекту підприємства можуть забезпечити послідовне впровадження найкращих практик безпеки мережі без ризику людської помилки чи втоми.

7. Прискорення виявлення атак та часу відповіді на них. Інтеграція ШІ з кібербезпекою дозволяє швидко виявляти загрози та реагувати на них, захищаючи компанію від незворотних збитків. ШІ сканує всю систему, рано виявляє загрози та спрощує завдання

безпеки порівняно з людьми.

8. Додатковий захист автентифікації. Веб-сайти з функціями облікових записів користувачів або контактними формами, що містять конфіденційну інформацію, потребують додаткового рівня безпеки та захисту. Штучний інтелект забезпечує цей рівень безпеки за допомогою таких інструментів, як розпізнавання обличчя, САРТСНА та сканери відбитків пальців, щоб забезпечити автентифікацію під час спроб входу. Це допомагає виявити шахрайські спроби входу та запобігти викраденню облікових даних і атакам грубої сили, які можуть призвести до потенційного порушення безпеки у комп'ютерній мережі.

9. Автоматизований аналіз та класифікація великих об'ємів текстових даних. Завдяки цьому алгоритми ШІ можуть ефективно розпізнавати фішингові повідомлення та спам, захищаючи користувачів від шахрайства.

10. Автоматизоване виявлення ботів. Боти становлять зростаючу загрозу кібербезпеці, їх використовують для зловмисних дій, як-от розповсюдження зловмисного програмного забезпечення та викрадення даних. Штучний інтелект може розпізнавати та блокувати ботів, визначаючи їхні шаблони, створюючи більш безпечні капчі та розгортаючи приманки для їхнього захоплення.

Хоча штучний інтелект має величезний потенціал, для його відповідального та етичного впровадження необхідно розуміти і враховувати його недоліки та проблеми:

1. Складність. Розробка систем кібербезпеки з використанням ШІ вимагає спеціальних знань та навичок їх створення, налаштування й поточного обслуговування.

2. Високі витрати на впровадження. Впровадження та підтримка ШІ-систем може бути досить дорогою, включаючи витрати на апаратне забезпечення, програмне забезпечення, навчаючі вибірки даних та навчання персоналу.

3. Потреба у великих обсягах якісних даних для навчання. ШІ потребує великих обсягів якісних даних для навчання моделей. Недостатність або низька якість даних може призвести до неточних результатів і зниження ефективності системи.

4. Вразливість до специфічних атак. ШІ-системи самі можуть стати мішенню атак, таких як атаки на моделі машинного навчання (наприклад, отруєння даних). Зловмисники можуть змінювати вхідні дані для введення системи в оману.

5. Дотримання етичних норм. Використання ШІ у кібербезпеці піднімає етичні та правові питання, включаючи

конфіденційність даних, прозорість прийняття рішень та відповідальність за помилки системи. Встановлення й дотримання чітких етичних принципів є досить важливими.

6. Дотримання правових норм. Правові вимоги, пов'язані зі штучним інтелектом і кібербезпекою, складні й постійно змінюються.

Висновок. З використанням штучного інтелекту у кібербезпеці є можливість значно підвищити рівень безпеки комп'ютерних мереж та систем, швидкість часу відгуку та адаптивність систем захисту. Водночас важливо враховувати етичні аспекти та можливі ризики, пов'язані з використанням штучного інтелекту, враховувати його недоліки та обмеження.

Література

1. Shutenko V. III in Cybersecurity: Exploring the Top 6 Use Cases [Електронний ресурс] / Victoria Shutenko. – 2023. – Режим доступу до ресурсу: <https://www.techmagic.co/blog/ai-in-cybersecurity/>.

2. Watkins O. 4 use cases for III in cyber security [Електронний ресурс] / Owen Watkins. – 2024. – Режим доступу до ресурсу: <https://www.redhat.com/en/blog/4-use-cases-ai-cyber-security>.

3. III in Cybersecurity Uses & Applications [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://www.engati.com/blog/ai-in-cybersecurity#toc-how-does-ai-in-cybersecurity-help-prevent-threats->.

4. III in Cyber Security: Pros and Cons, and What it Means for Your Business [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.terrano vasecurity.com/blog/ai-in-cyber-security>.

5. Шарма Н. 10 найбільших проблем використання III у сфері кібербезпеки [Електронний ресурс] / Нідгі Шарма – Режим доступу до ресурсу: <https://aw.club/global/uk/blog/10-most-critical-ai-challenges-in-cybersecurity>.

6. Штучний Інтелект. Частина 1 – переваги та недоліки III [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: <https://ardenis.com.ua/blog/shtuchnyj-intelekt-shi-perevagy-ta-nedoliky-chasty-na-1/>.

КОРИСНІ ВЛАСТИВОСТІ ПЛОЩИННИХ ПІДГРАФІВ ГРАФІВ-ОБСТРУКЦІЙ ОБМЕЖЕНОГО НЕОРІЄНТОВАНОГО РОДУ

В. І. Петренюк, Центральноукраїнський нац. технічний університет

Д.А. Петренюк, ІК НАН України

SOME PROPERTIES OF PLANAR SUBGRAPHS OF THE GRAPHS -OBSTRUCTIONS WITH LIMITED NONORIENTABLE GENUS.

Abstract. The task will be to study the structural properties of graphs - obstructions G of an nonorientable limited genus $\gamma(G)$, presented as the ϕ -image of a planar graph G_1 with a given set of points $X_1 = \{x_{1j}\}_{j=1}^{|X_1|}$ and a simple star $St(x_0)$ with a set of hanging vertices $X_2 = \{x_{2j}\}_{j=1}^{|X_2|}$, when identifying pairs of points from sets of joining points (x_{1j}, x_{2j}) , $j=1,2,\dots,|X_i|$, where by a point we mean either a vertex or an arbitrary point of the edge of the graph G . For sets of connection points X_i with a given reachability number t_i , $t_i \geq 1$, which is equal to the smallest number of the faces on the borders of which all points of the set X_i are located, and known characteristic $\theta(X_i)$ are used, and two new cells distances of a given set of points X and the minimum of a planar graph with respect to a given cells distance are determined distances of the set of points X . The main result is Theorems 1 and 2.

Основні поняття та позначення узяті із [1]-[3]. Під граф-обструкцією розумітимемо такий простий граф G у якого кожне ребро є суттєвим відносно роду $\gamma(G)$ при операції видаленні ребра u , т. то $\gamma(G \setminus u) = \gamma(G) - 1$. **Задача** полягатиме у вивченні структури площинних підграфів графів обструкцій обмеженого неорієнтованого роду.

Визначення 1. Для множини точок M , $M = \{a_i\}_{i=1}^m$, площинного графа G з числом досяжності $t_G(M, \Sigma_0)$, де $t_G(M, \Sigma_0) = t$, $t \geq 2$, будемо називати клітковою довжиною $D_G(M, t)$, де $D_G(M, t) = \min_{\forall f \in fGN} \sum_{\forall s_i, s_j \in T} d_G(s_i, s_j, f)$, між підмножинами L_i, L_j множини

M , $M = L_i \cup L_j$, $L_i \cap L_j = \emptyset$, розташованими на границях довільних кліток $s_i, s_j, \{s_i, s_j\} \subset \mathcal{S}_G(\Sigma_0, f)$, що реалізують число досяжності $t_G(M, \Sigma_0)$, де fGN - множина всіх неізоморфних вкладень графа G до Σ_0 , що реалізують число досяжності $t_G(M, \Sigma_0)$ множини точок M .

Визначення 2. Будемо називати множини M , $M = \{a_i\}_{i=1}^m$, точок графа G з числом досяжності t , де $t_G(M, \Sigma_0) = t$, критичною відносно кліткової довжини $D_G(M)$ при операції видалення довільного елемента a_i , якщо має місце нерівність $D_G(M \setminus a_i) < D_G(M)$, чи відносно операції стискання ребра $u = (ab)$ в точку a' (якщо $\{a, b\} \subset M$, то замість M розглядатимемо множину $M' = (M \setminus (a, b)) \cup \{a'\}$), якщо має місце нерівність $D_{Gu}(M') < D_G(M)$.

Визначення 3. Будемо називати граф G мінімальним відносно $D_G(M)$ при операції видалення або стискання в точку довільного ребра u , якщо , або $D_{Gu}(M) + 1 = D_G(M)$, або $D_{Gu}(M') < D_G(M)$, де Gu - граф зі стиснутим в точку a' ребром $u = (a, b)$ та $M' = (M \setminus \{a, b\}) \cup a'$.

Теорема 1. Нехай задано площинний граф G із заданою множиною M точок суттєвих при операції видалення з множини відносно кліткової відстані $D_G(M)$, $D_G(M) = d, d \geq 2$, з числом досяжності $t_G(M)$, $t_G(M) = t \geq 2$, а множина ребер графа є суттєвою відносно кліткової відстані $D_G(M)$ при операції видалення чи стискання в точку. Мають місце наступні твердження:

1. Якщо $D_G(M) = 2$ та $t_G(M) = 2$, то граф G ізоморфний одному з перших 6-ти графів, тобто є кільцем з двома невивірженими простими циклами z_i , довжини $l(z_i)$, де $z_1 \cap z_2 = \emptyset$, $l(z_1) \geq l(z_2) > 3$, що задовольняють співвідношенню д) та одній з наступних умов:

- а) $l(z_1) = 6, l(z_2) = 3$; б) $l(z_1) = 5, l(z_2) = 3$; в) $l(z_1) = 4, (l(z_2) = 4) \vee (l(z_2) = 3)$;
 г) $l(z_1) = 3, (l(z_2) = 4) \vee (l(z_2) = 3)$;

д) вершини циклів z_1, z_2 складають множину M та пари вершин (v_1, v_2) циклів z_1, z_2 , відповідно, що з'єднані ребром, належать простим циклам довжини не більше 5.

2. Якщо множину M розбито на $t - 1$ непорожніх частин M_1, M_i, M_j , де $M_i \cap M_j = \emptyset$, $D_G(M_1 \cup M_i) = 2$ для всіх, $t_G(M) = t$, $t > 2$, то граф G є ϕ -образом k копій, принаймні двох графів H_{1i}, H_{1j} , ізоморфних одному з перших 6-ти графів H_{1i} , $H_{1i} = G(M_1, M_i)$, наведених на рис. 2, та заданим на парі простих ланцюгів $L_{H_{1i}}, L_{H_{1j}}$ графів H_{1i}, H_{1j} , де $i < j$, $i, j = 1, 2, \dots, k$, $k \leq t$.

Доведення твердження 1. Нехай виконуються умови теореми 1. Покажемо, що граф G ізоморфний одному з перших 6-ти графів, тобто є або кільцем з двома зовнішнім та внутрішнім циклами, або клітковим ланцюгом з двома кінцевими невиродженими простими циклами z_i , $z_1 \cap z_2 = \emptyset$, з довжинами $l(z_i)$, де $l(z_1) \geq l(z_2) \geq 3$, що задовольняють одній з умов а),..г).

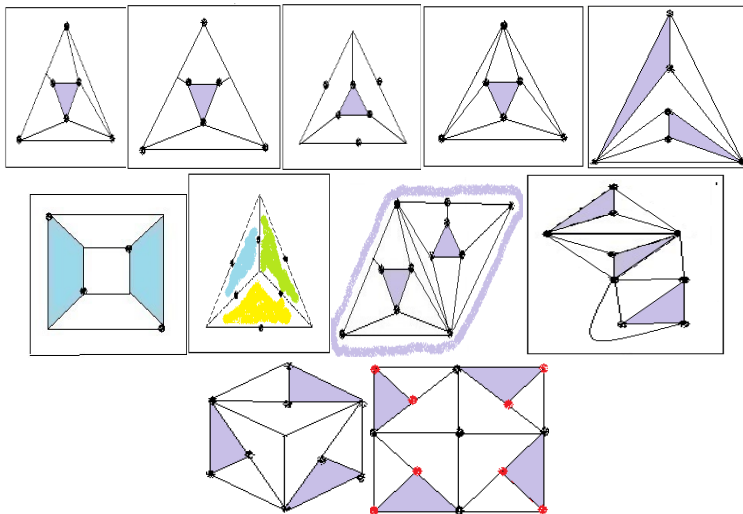


Рис 1. На перших шести картах є площинні підграфи графів $B_7, C_3,$

$C_4, D_2, K_{1,2,2,2}, K_7 \setminus C_4, K_{4,5} \setminus 4K_2$, відповідно, з виділеними жирним вершинами із множин M з клітковою відстанню 2; після приєднання до них висячих вершини простої зірки отримаємо згадані вище графи. На другому ряду перша карта 1-підрозділений граф K_4 є підграфом E_2 , має множину M з точок 1-підрозділення ребер, де $(t=3) \wedge (\theta=1)$, далі три 3-клітково мінімальні та один 4-клітково мінімальний граф

Доведемо, що обидва цикли графа G не можуть бути довжиною більше 6. Припустимо що один з двох простих циклів z_i має довжину більше 6. Тоді на ньому має бути не менше чотирьох елементів множини M та, принаймні, чотири вершини з $G^0 \setminus M$. Оскільки $t_G(M) = 2$, то на кожному z_i має бути щонайменше по одному елементу v_i множини M . Видалимо v_1 та розглянемо цикли на яких розміщено множину $M \setminus v_1$. Згідно умови суттєвості операції видалення з множини відносно кліткової відстані $D_G(M)$, $D_G(M) = d$, $d = 2$, матимемо множину $M \setminus v_1$ розміщену на двох простих циклах довжиною, принаймні 3, та однією спільною вершиною. Отже, або цикл z_i містить три елементи множини M і не більше, або містить ребро $u = (v, w)$ з двома кінцевими вершинами степені 2 з множини M , яке є несуттєвим при операції стискання в точку. Обидва випадки суперечать умові мінімальності множини M відносно $D_G(M)$, або умові мінімальності множини ребер відносно $D_G(M)$. Припущення неможливе. Тим самим доведено, що один цикл має довжину 6, а інший довжину 3 та містять по три елементи множини M . Твердження 1 доведене.

Теорема 2. Нехай площинний граф G - мінімальний відносно кліткової довжини 2 підграфом графа-обструкції проективної площини та задано його підмножину точок M з суттєвих вершин відносно чисел $D_G(M)$ і $t_G(M) = t$, де $t = 2$, $D_G(M) = 2$, при операції видалення точки з множини M . Якщо задано φ -перетворення графа G та квазіірки $S_G(H)$ з центром-графом H , H - центр квазіірки гомеоморфний одному з графів множини $\{K_{2,3}, K_4, K_5 \setminus e, \bar{K}_r\}$, $r \geq 1$,

наступним чином:

$\varphi : (G + St(H), \sum_{i=1}^{|M|} (a_i + g_i)) \rightarrow (DH, \{a_i^*\}_{i=1}^{|M|})$, де множина M ,

$M = \{a_i\}_{i=1}^{|M|}$, точок має характеристики $t = 2$, $\theta = 0$, квазіірка $St(H)$ із множиною висячих вершин $\{g_i\}_{i=1}^{|M|}$, тоді мають місце наступні співвідношення:

1. Якщо $r = 1$, ($t = 2$, $\theta = 0$) чи ($t = 3$, $\theta = 1$), то граф DH є графом-обструкцією для проективної площини, причому образи пари висячих ребер квазіірки не породжують підграфів

гомеоморфних графу K_4 з двома видаленими несуміжними ребрами;

2. Якщо граф DH є графом–обструкцією для поверхні Клейна, то образи пар висячих ребер квазізірки породжують не більше двох різних підграфів гомеоморфних K_4 та можуть бути несуттєвими відносно роду при операції стягування висячих ребер в точку;

3. Якщо замінити в співвідношенні 1 рівність для t на $t=3$, то граф DH матиме неорієнтований рід не менше 4, причому φ -образи пари висячих ребер квазізірки породжують не більше трьох підграфів гомеоморфних K_4 .

Доведення. Нехай виконуються умови твердження 5 та співвідношення 1. Покажемо виконання умови суттєвості при операції видалення кожного ребра u графа DH відносно неорієнтованого роду 2. Можливі наступні три випадки:

1. $u \in G^1$; 2. $u \in H^1$; 3. $u \in St^1(H) \setminus H^1$.

Нехай має місце випадок 1. Видаливши ребро матимемо згідно умови суттєвості множини M відносно числа кліткової відстані $D_G(M)$ матимемо зменшення на 1 цього числа. Тоді матимемо можливість такого вкладення графа G в площину з меншим на 1 числом приклеєних лент Мебіуса яке розміщуватиме множину M на границі однієї псевдоклітки, в яку вкладемо квазізірку $St(H)$.

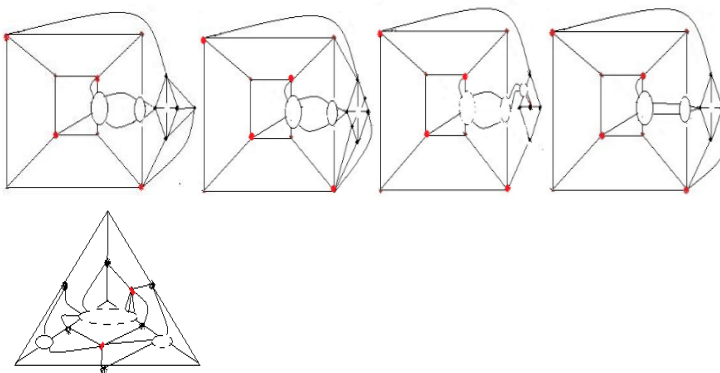


Рис.2. На перших чотирьох картах наведені приклади прототипів 3-зв'язних графів обструкцій неорієнтованого роду 3 на 5-й карті граф не 3-зв'язний.

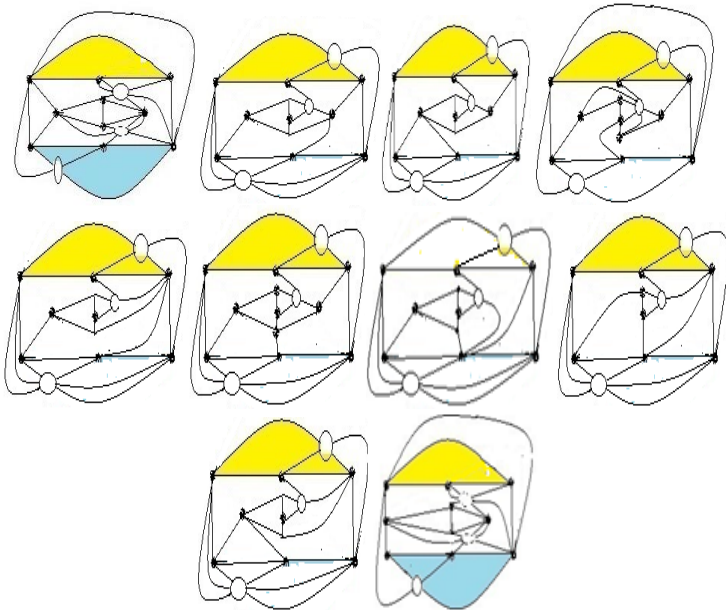


Рис.3. Приклади n -зв'язних прототипів графів-обструкцій неорієнтованого роду 3, $n > 2$. В другому ряду наведені мінори поверхні Клейна.

Висновок. Отримано математичне забезпечення для алгоритмів побудови мінорів для поверхні заданого неорієнтованого роду більше 2 шляхом φ -перетворення пари:

1. площинних графів мінімальних з заданою клітковою довжиною певної множини точок та зірки,
2. 3-мінімальних площинних з множиною всіх вершин та висячих вершинк вазізірок з центром K_4 чи $K_{2,3}$.

Література

1. Хоменко М. П. φ -перетворення графів. Препринт ИМ АНУ. Київ. 1973. 383 с.
2. Хоменко М. П. Топологические аспекты теории графов. Препринт ИМ АНУ. Київ. 1970. 299 с.
3. Mohar B., Thomassen C. Graphs on Surfaces. Johns Hopkins University Press, 2001. 412 p. <https://www.sfu.ca/~mohar/Book.html>
4. Hur S. The Kuratowski covering conjecture for graphs of order less than 10. Phd, Ohio State University, 2008. http://rave.ohiolink.edu/etdc/view?acc_num=osu1209141894

5. Archdeacon D., Huneke P. A Kuratowski Theorem for Nonorientable Surfaces. *Journal of combinatorial theory, Series B.* 1989. **46**. P. 173–231.
 6. Петренюк В.І. Про структуру площинних підграфів графів-обструкцій неорієнтованої поверхні заданого роду. *Фізико математичне моделювання та інформаційні технології.* 2021. № 33. С. 105–109. Google Scholar.
 7. Anna Flötotto. Embeddability of graphs into the Klein surface. Dissertation, University Bielefeld, 2010, -174 pp.
 8. P.Skoda. Obstructions for embedding graphs into surfaces, Simon Frazer University, PhD dissertation, 2012.-133 p.
 9. Bienstock D., Dean N. On obstructions to small face covers in planar graphs, *J. Combin. Theory Ser. B.* 1992. **55**. P. 163–189. <https://doi.org/10.1016/0095-8956%2892%2990040-5>
 10. Bienstock D., Monma C.L. On the complexity of covering vertices by faces in a planar graph. *SIAM J. Comput.* 1988. **17**. P. 53–76. <https://doi.org/10.1137/0217004>
 11. Mohar B. Face Covers and the Genus Problem for Apex Graphs. *Journal of Combinatorial Theory, Series B.* 2001. **82**. P. 102–117. <https://doi.org/10.1006/jctb.2000.2026>
 12. Mohar B. Apex graphs with embeddings of face-width three. *Discrete Mathematics.* 1997. **176**. P. 203–210. [https://doi.org/10.1016/S0012-365X\(96\)00363-9](https://doi.org/10.1016/S0012-365X(96)00363-9)
- [1] 13. В.І. Петренюк, Д.А. Петренюк, О.В. Оришака. Структура проєктивно площинних підграфів графів-обструкцій заданої поверхні. *Кібернетика та комп'ютерні технології.* 2022, № 2. С. 1-20 DOI:10.34229/2707-451X.22.2.0
- [2] 14. В.І. Петренюк, Д.А. Петренюк. Про алгоритм побудови 2-зв'язних мінорів поверхні Клейна. *Фізико-математичне моделювання та інформаційні технології,* № 37 (2023): 72-74. <http://www.fmmit.lviv.ua/index.php/fmmit/article/view/308>
- [3] 15. Edwin R. van Dam, Jack H. Koolen, Hajime Tanaka. Distance-regular graphs, E-JC, DS22: Apr 15, 2016, <https://www.combinatorics.org/ojs/index.php/eljc/issue/view/Surveys>
- [4] 16. Bojan Mohar, Petr Škoda Excluded minors for the Klein bottle I. Low connectivity case, *Journal of Combinatorial Theory, Series B,* Volume 164, January 2024, Pages 299-320. <https://doi.org/10.1016/j.jctb.2023.10.002>

- [5] 17. Bojan Mohar, Petr Škoda Excluded minors for the Klein bottle II. Low connectivity case, Journal of Combinatoria Theory, Series B, Volume 166, May 2024, Pages 80-108. <https://doi.org/10.1016/j.jctb.2023.12.006>

***T*-факторизація повного графа та граціозність дерев**

Петренюк Д.А.

dmytrotheukrainian@ukr.net

Інститут кібернетики ім. В.М. Глушкова НАН України

Розклади графів мають велику кількість різноманітних застосувань, яка неухильно зростає. Серед таких застосувань ортогональні оптичні коди, синхронний кратний доступ до каналів, групове тестування та суперпозиційні коди, аутентифікаційні коди, пружні та кореляційно-захисні функції, схеми каналів та мережі зв'язку, тестування програмного забезпечення, розмітка дисків та нанесення доріжок. Одним з важливих різновидів розкладів графів є *T*-факторизація повного графа, тісний зв'язок якої з популярною у математичних колах та все ще відкритою гіпотезою про граціозність дерев робить цей вид розкладів особливо цікавим.

***T*-факторизація повного графа**

Розкладом графу H на підграфи з даного сімейства $G = \{g_1, g_2, \dots, g_k\}$, або (H, G) -*розкладом*, називають розбиття множини ребер графу H на підграфи (*компоненти розкладу*), кожен з яких ізоморфний одному з елементів множини G . Підграф G_1 називають *фактором* графу G , якщо множина вершин підграфу G_1 співпадає з множиною вершин графу G . *Факторизацією* графу називають такий розклад цього графу, компонентами якого є фактори графу H . Факторизацію, всі компоненти якої ізоморфні дереву T , називають *T-факторизацією* [1].

Одним з методів побудови *T*-факторизацій є *півобертвий метод*. *T*-факторизації, отримані з його допомогою, також називають *півобертковими*. Вказаний метод застосовується для так званих *півсиметричних дерев*. *Півсиметричне дерево* [1] містить центральне ребро, після видалення якого воно розпадається на дві ізоморфних компоненти (*симетричні половини*). Симетричні половини являють собою кореневі дерева, корені яких – кінці центрального ребра відповідного півсиметричного дерева.

Для побудови T -факторизації використовується шаблон, який являє собою коло, розбите $n=2k$ точками на рівні дуги (назвемо ці дуги *елементарними*). Точки поділу позначають $0, 1, \dots, n-1$. Півсиметричне дерево порядку $n=2k$ називають *правильно вписаним* у це коло (рис.1), якщо: точки поділу є вершинами дерева T ; ребра дерева T зображаються хордами кола; для кожної допустимої довжини хорди рівно два нецентральних ребра, симетричних відносно центра кола, мають таку довжину. Під *довжиною хорди* розуміємо кількість елементарних дуг у меншій з дуг, на які ця хорда розбиває коло.

Якщо T – півсиметричне дерево порядку $n=2k$, правильно вписане в коло, то можна виконати T -факторизацію графа K_n на k компонент, ізоморфних дереву T . Для цього досить повернути вписане дерево T на h елементарних дуг і одержати дерево T_h ($h=0, 1, \dots, k$). Дерева $T_0=T, T_1, \dots, T_k$ разом складають шукану T -факторизацію. Іншими словами, T -факторизація утворюється в результаті півоберту твірної компоненти навколо центра кола. Звідси і назва методу півобертвий.

Таким чином, задача побудови T -факторизації зводиться до правильного вписування дерева в шаблон. Беручи до уваги те, що півсиметричне дерево однозначно визначається своєю симетричною половиною, останню задачу можна, звести до такого вписування в півколо симетричної половини (тобто кореневого дерева порядку k), при якому присутні усі довжини ребер від 1 до $k-1$, без повторень (рис. 2). Ця умова забезпечує відсутність спільних ребер у компонент розкладу, а отже, інша симетрична половина вписується в протилежну половину кола, і цикли не утворюються.

Петренко А. Я. у 2001 році висунув наступну гіпотезу:

Гіпотеза 1: будь-яке півсиметричне дерево T допускає півобертovu T -факторизацію.

Приклад застосування T -факторизації повного графа

Одну з практичних проблем, які приводять до задачі отримання T -факторизації повного графа, є проблема прокладання безпроводових мереж з самоорганізацією (wireless ad hoc networks) [2]. Розглянемо радіомережу мобільних пристроїв зв'язку протипожежного відділення, частина з яких є стаціонарними, решта – мобільними. Кожен пристрій дозволяє виходити на зв'язок з будь-яким іншим пристроєм, але зв'язок між деякими парами пристроїв може тимчасово втрачатися через топографічні особливості місцевості, якою рухаються пожежники. Для ефективності дій

команди необхідна можливість швидко відновити зв'язок, а також підтримувати оптимальні параметри мережі: малу кількість ретрансляцій (малий діаметр мережі) та розподіл навантаження на пристрої (малий максимальний степінь вершини). Кожний пристрій має свій унікальний ідентифікаційний номер і працює як приймач, передавач та маршрутизатор (тож маршрутизація підтримується самими пристроями).

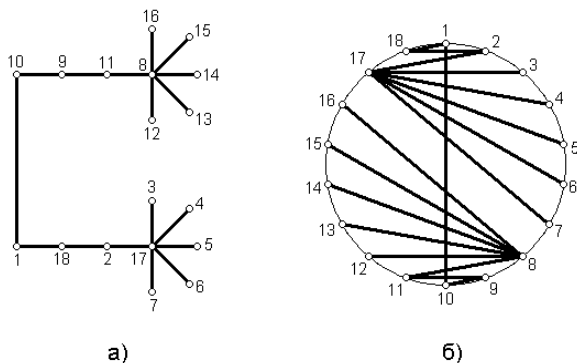


Рис. 1. Півсиметричне дерево порядку 18 (а) і його правильне вписування в коло (б).

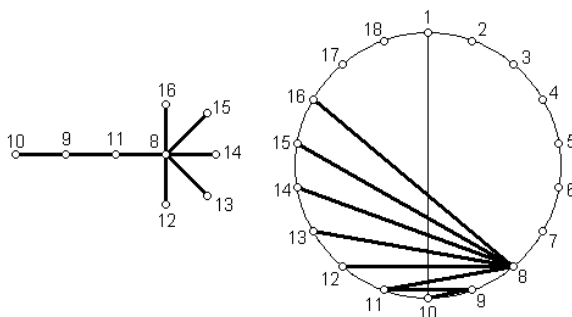


Рис.2. Приклад вписування симетричної половини в коло з дотриманням необхідних умов вписування.

В термінах теорії графів ця проблема звучить так: необхідно побудувати зв'язну мережу, яка не містить циклів, тобто дерево. Множина усіх можливих зв'язків утворює повний граф, де вершини графа – пристрої зв'язку, а ребра – радіозв'язок між парами цих пристроями. Кожна окрема мережа утворює в цьому повному графі остовне дерево (фактор), ізоморфне одній і тій самій структурі. Отримується набір ізоморфних факторів (структур мережі), які

утворюють сукупність мереж з однаковим діаметром та однаковим максимальним степенем. Жодні дві мережі не мають спільних ребер, тому пошкодження однієї окремої мережі не вплине ні на який інший фактор із загальної сукупності мереж. Замість відновлення одного пошкодженого зв'язку здійснюється перехід до іншої мережі. Перелік мереж може бути закладено у програмне забезпечення пристроїв зв'язку. У випадку, коли втрачено зв'язок між двома пристроями, від одного з них надходить радіоповідомлення, і проблема вирішується негайно шляхом переключення на одну з резервних мереж.

Гіпотеза про граціозність дерев

Неважко переконатися в тому, що для успішного розв'язання задачі правильного вписування k -вершинного дерева в півколо достатньо так занумерувати його вершини числами $x_i \in (1, 2, \dots, k)$, щоб абсолютні різниці номерів суміжних вершин склали множину $(1, 2, \dots, k-1)$. Така розмітка (на сьогодні відповідне англійське слово «labeling» зазвичай перекладають як «розмітка», тому автор вживає тут цей переклад замість слова «нумерація», яке вживалося автором раніше) була вперше розглянута А. Росою у 1976 році [3] і згодом отримала назву граціозної.

Граціозною розміткою дерева T з n вершинами називають присвоєння його вершинам номерів $1, 2, \dots, n$, при якому абсолютні різниці номерів кінців ребра є різними та утворюють послідовність $1, 2, \dots, n-1$. Дерево, яке допускає граціозну розмітку, називають, відповідно, *граціозним*.

Була сформульована знаменита гіпотеза Рінгеля-Коціга-Роси:

Гіпотеза 2 (Гіпотеза про граціозність дерев): усі дерева граціозні, тобто будь-яке дерево порядку k допускає таку розмітку його вершин натуральними числами $1, 2, 3, \dots, k$, при якій різниці номерів кінців ребер утворюють ряд натуральних чисел $1, 2, 3, \dots, k-1$ без повторень.

З моменту своєї появи у 1976 році і до сьогодні, гіпотеза про граціозність дерев є однією з найпопулярніших математичних гіпотез. Доведення цієї гіпотези автоматично довело би, зокрема, і гіпотезу про те, що всі півсиметричні дерева допускають півобертову T -факторизацію (Гіпотезу 1).

Існує два основних підходи до розв'язання проблеми істинності гіпотези про граціозність дерев. Перший з них полягає в доведенні граціозності та отриманні алгоритмів граціозної розмітки для окремих класів дерев. Якби вдалося продемонструвати граціозність

усіх можливих класів дерев, то гіпотезу було би доведено. На сьогодні граціозність доведено для зірок, ланцюгів, гусениць, оливкових дерев, деяких підкласів омарів (фесрверків, (2,2)-гусениць, омарів з досконалими паросполученнями), бананових та узагальнених бананових дерев та низки інших класів дерев (рис. 3). Знайдено значну кількість методів поєднання кількох граціозних дерев для отримання більшого граціозного дерева [4].

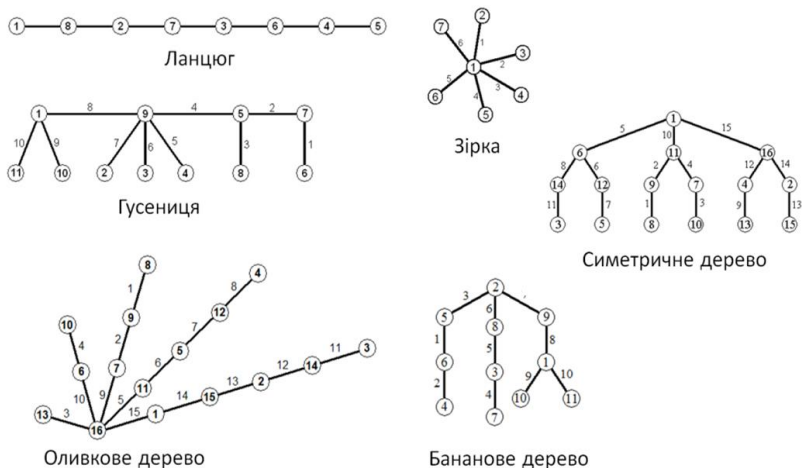


Рис.3. Деякі класи граціозних дерев

Інший підхід до відповіді на питання про граціозність усіх дерев полягає у використанні комп'ютера для перевірки граціозності дерев, кількість вершин яких не перевищує заданої скінченної величини. Такий підхід є додатковим джерелом алгоритмів граціозної розмітки дерев, а також може бути використаний для спростування гіпотези, якщо буде знайдено хоча б одне дерево, яке не допускає граціозної розмітки. Досі, щоправда, такого дерева не знайдено (як дотепно зауважив з цього приводу один дослідник, «віра в справедливість гіпотези про граціозність дерев настільки сильна, що, якби навіть було насправді знайдено дерево, яке не допускає граціозної розмітки, його, скоріше за все, не визнали б деревом» [5]). У 2010 році Фанг використав детермінований алгоритм пошуку з поверненням і довів, що всі дерева, які мають не більше 35 вершин, граціозні [6].

Обидва підходи до вивчення гіпотези про граціозність дерев досі давали лише підтвердження гіпотези. Проте деякі дослідники висловлюють серйозні сумніви щодо її вірності. Зокрема, А. Вістрі [7] зауважує, що переважну більшість методів побудови граціозної розітки було розроблено для дерев, які мають певні риси регулярності (зірки, симетричні дерева) або характеризуються досить простою структурою (гусениці, феєрверки), в той же час повністю відсутні докази граціозності у випадках достатньо нерегулярних дерев (навіть граціозність усіх омарів ще досі не доведено).

Втім, варто зазначити, що нещодавно було доведено одну з гіпотез, безпосередньо пов'язаних з гіпотезою про граціозність дерев. Свого часу Роса запропонував граціозну розмітку (він тоді назвав її β -оцінкою) як засіб доведення наступної гіпотези, висунутої у 1963 році:

Гіпотеза 3 (Гіпотеза Рінгеля) [8]: для кожного додатного числа m існує розклад графа K_{2m+1} на $2m+1$ ізоморфних дерев порядку $m+1$ (рис. 4). Ківаш та Штаден у квітні 2020 року [9] та, незалежно від них, Монтгомері, Покровський та Судаков у січні 2021 року [10] опублікували доведення гіпотези Рінгеля.

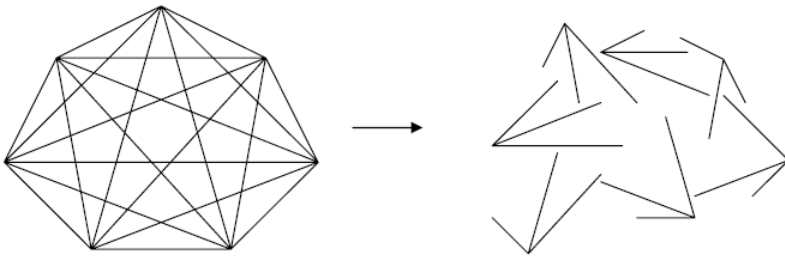


Рис. 1.1. Розклад графа K_7 на 7 ізоморфних дерев порядку 4.

Серед ефективних шляхів критики гіпотези можна назвати аналіз усіх можливих граціозних розміток деяких граціозних дерев, що дав би можливість оцінити вплив структури дерева на можливі номери його вершин. На думку Вістрі, деякі дерева зі складною будовою можуть накладати настільки серйозні обмеження на розмітку вершин, що граціозність цих дерев буде поставлено під сумнів [5]. Хоча знаходження граціозних розміток для усіх дерев з кількістю вершин до 35 включно визнається вагомим аргументом на користь правдивості гіпотези, але цей факт ніяким чином не

виключає можливості існування неграціозних дерев, що мають більше ніж 35 вершин.

Так чи інакше, сьогодні ще рано говорити про повне доведення або спростування гіпотези про граціозність дерев, незважаючи на те, що з часу її появи минуло майже півстоліття. Втім, переважна більшість дослідників досі схиляється до віри в справедливості гіпотези, беручи до уваги граціозність багатьох класів дерев, результати комп'ютерного пошуку та повну відсутність контрприкладів. Відповідно, відкритою залишається гіпотеза про те, що кожне півсиметричне дерево допускає півобертову T -факторизацію.

ЛІТЕРАТУРА

1. Петренюк А. Я. Півобертові деревні факторизації повних графів. *Український математичний журнал*. 2001. **53**. № 5. с.710–716.

2. Kovar P. Decompositions and Factorizations of Complete Graphs / Kovar P. // *Structural Analysis of Complex Networks* [Edited by Matthias Dehmer]. — 2010. — Springer.

3. A. Rosa, On certain valuations of the vertices of a graph, *Theory of Graphs* (Internat. Sympos. Rome, 1966) pp. 349-355. Eds. Gordon and Breach, New York; Dunod, Paris, 1967.

4. Донець Г. П., Петренюк Д. А. Граціозна нумерація дерев [монографія]. Київ, ПП, 2017. 144 с.

5. M. Alfalayleh, L. Brankovic, H. Giggins, M.Z. Islam. Towards The Graceful Tree Conjecture: A survey // In: *Proceed. AWOCA2004*, 7-9 July, Ballina, Australia (2004).

6. Fang W. A computational approach to the graceful tree conjecture [Електронний ресурс]. — Access Mode: arXiv:1003.3045v2 [cs.DM]

7. Vietri A. Sailing towards, and then against, the Graceful Tree Conjecture: some promiscuous results // *Bulletin of the ICA*. 2008. Vol 53, May. P. 31–46.

8. Gallian J. A. A Dynamic Survey of Graph Labeling. [Twenty-fifth edition, December 2, 2022]. *The Electronic Journal of Combinatorics*. 2022, #DS6.

9. Keevash P., Staden K. Ringel's tree packing conjecture in quasirandom graphs. In press, 2022. <https://arxiv.org/pdf/2004.09947>

10. Montgomery R., Pokrovskiy A., Sudakov B. A proof of Ringel's conjecture. *Geometric and Functional Analysis*. **31**. 2021. 663-720.

"КОГНІТИВНІ ЧИСЛА" ЯК ПРИКЛАД "ЖИВОЇ СТРУКТУРИ"

Юрій Прокопчук

*Institute of Technical Mechanics of the NAS and SSA,
Ukrainian State University of Science and Technologies*

"COGNITIVE NUMBERS" AS AN EXAMPLE OF "LIVING STRUCTURE"

Abstract. The paper considers possible cognitive mechanisms of formation of quantitative meanings in explicit and implicit processes of knowledge representation about the world. The specific properties of the formal structure "cognitive number" are analyzed. It is claimed that this entity belongs to the class of "living structures". Key words and topics: "cognitive number", "living structure", "intelligent mathematics", the nature of vagueness, sketch networks, multi-resolution descriptions, the meaning of information, symbolization, cognitive measurements, cognitive sensors, entanglement, wholeness, cognitive-affective processes, creative imagination, combinatorial complexity, principles of constructing world; A formal model approach.

У доповіді розглядаються можливі когнітивні механізми формування кількісних смислів у явних та неявних процесах репрезентації знань про світ. Аналізуються специфічні властивості формальної структури «когнітивне число» (ког-число / cog-number). У рамках парадигми граничних узагальнень (ПГУ) стверджується [1, 2, 3], що ця сутність, як і всі мережі начерків, належить до класу «живих структур». Розглянемо цю сутність більш детально.

Будь-яке число-значення будь-якого тесту має множинну інтерпретацію [1, 4, 5]. На рис. 1 в динаміці показана структурна (фрактало-подібна) інфляція - генерація сенсу - когнітивного числа x деякого тесту A (Mental Magnitude Representations; Temporal morphogenesis, Morpho-evolution with learning; Foundations of Quantitative Thought; Intelligent Mathematics: Number Understanding Modelling). Формальне представлення ког-числа τ . $Gv(\tau) = \{a \rightarrow_e b\}_\tau$ [Context]. Приклад: «Температура тіла = 38°C» \rightarrow «Темп. висока» \rightarrow «Темп. не Норма».

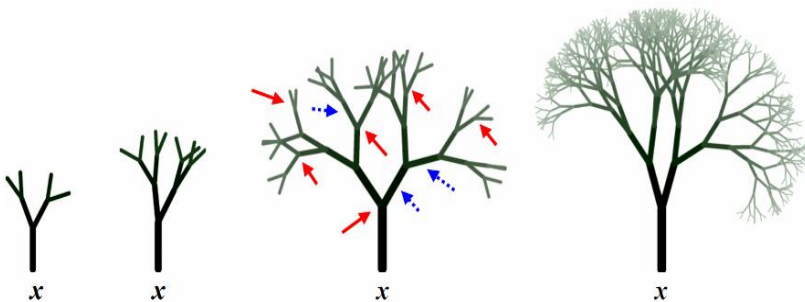


Рис.1 - Структурна інфляція - генерація смислу - когнітивного числа x

Вузли 'дерев' / мережі начерків задають внутрішню інтерпретацію вхідного числа / значення (Meaning of information: the interrelation between numerical and language processing; The Emergence of New Meanings in the Self). У процесі суб'єктивної інфляції ког-число 'зростає' як жива структура. Схожість додає динамічна циркуляція енергії / активності, зокрема, хвильове поширення енергії з усіма наслідками (резонанси, лавини, вихори, осцилятори). Для живої системи важливо економити енергію. Це приводить до концепції критичності ментальної сфери [1, 5]. Для реалізації критичності потрібна різноманітність начерків образів – як точних, так і грубих. Ця різноманітність забезпечується мережею начерків образу, різновидом якої є ког-число.

Примітка. Christopher Alexander (архітектор та дизайнер) виявив [6], що узгоджене ціле має вражаючий характер, званий живою структурою або цілісністю, яка є фізичною або математичною структурою, що складається з безлічі підструктур з властивою їм ієрархією. Ціле постійно і рекурсивно диференціюється на численні субцілі. Це не складання готових елементів у стилі LEGO, а зростання, подібне до ембріона, з постійною диференціацією та адаптацією до єдиного цілого [6, 7]. Alexander ввів поняття "degree of order" – це формальна характеристика, яку можна пов'язати з життєвістю, структурною красою, зв'язністю / цілісністю або гармонією. Мережа мереж начерків (модель ментальної сфери в рамках ПГУ), включно з ког-числами, відповідає цій характеристиці (Living Structure, the Degree of Structural / Cognitive Beauty or Livingness).

Поняття живої розумної структури безпосередньо стикається зі специфічними проблемами смислу, краси, розуміння в різних

аспектах (концепції критичності, заплутаності; «стиснення - це розуміння»: модель краще, якщо вона може пояснити більше з меншими витратами) [3, 4, 5]. У рамках ПГУ розробляється загальний підхід для ефективного виявлення у Розумі базової живої (розумної) структури та її підструктур з урахуванням принципів економії, холізму, адаптивності / живучості, духовності та краси. Духовність є важливим аспектом когнітивної краси.

Згідно з цим організмичним світоглядом (Whitehead, Alexander) [7], світ є когерентним цілим, а не фрагментованим на механічні частини, як це розуміється і сприймається в даний час в рамках картезіанської картини світу (і це переноситься на моделі Розуму). Концепція «когнітивних чисел» відповідає цілісному / холістичному підходу з огляду на «Онтологію духовності». Жодна з наук про складність досі не займається проблемою створення "духовних структур / сутностей". Без таких сутностей вони не зможуть зрозуміти, зокрема, метафору "Душа місця / образу / об'єкту / числа".

На середньому зображенні рис. 1 показані індуктори або евристики, які також дозрівають в процесі імпліцитного пізнання та накопичення досвіду і є частиною ког-числа. Причому індуктори є і для вузлів і зв'язків між значеннями (стрілки різних кольорів і типів). Індукторний простір забезпечує заплутаність (сильну зв'язність) або цілісність ментальної сфери. Індуктори реалізують глибоку інтуїцію (втільений прогноз), аудит та семіозис (означає процес інтерпретації знака або процес породження значення). Крім того, ког-число бере участь у тих чи інших спостереженнях / подіях і, отже, "стрілах часу" (формують Суб'єктивний Простір-Час-Дії - СПЧД: interconnected nature of remembering and imagining; memory and imagination as meaning-making processes; Cognitive computing systems; Extending dynamics with fractal like calculus; Livingness of Space). Останні разом з індукторами та наративами реалізують узагальнене заплутування ког-числа (теста), а також формують різноманітний контекст (досвід). Це узагальнення дозволяє аналізувати динаміку фрактало-подібних систем, вловлюючи їх складні та самоподібні властивості, зокрема, критичний стан. Мульти-формалізм, який застосовується в концепті «ког-числа» (загалом в мережах начерків), відкриває нові можливості для розуміння та моделювання складних та цілісних фрактало-подібних "розумних" обчислювальних структур ("Living Numbers": Computing the Structural / Cognitive Beauty of "Cognitive Number") [5].

Ментальна деталізація будь-якого значення τ ког-числа ґрунтується як на «контрольованій (контекстній) галюцинації / Controlled Hallucination» в межах мережі начерків (τ^\downarrow [Context]), так і на «творчій (контекстній) уяві / Creative Imagination» ($\tau^{\downarrow\downarrow}$ [Context]); контексти формуються «стрілами часу»; Imagination as higher mental function). Множина емоційних подій, в які включені ког-числа, відносить ког-числа до «Ontology of spirituality» (як і інші мережі начерків). Дійсно, кожне (екстремальне, не типове) значення ког-числа викликає з пам'яті (досвіду - СПЧД) множину емоційних подій, де виникало таке значення, а також множину пояснень асистентів (Remembering Through the Environment). В сукупності ці події формують «емоційну ауру» навколо того чи іншого значення.

Будь-яке значення/знак τ може відображати активність деякого нарису образу або радикала P , наприклад, дії. Формально цей зв'язок є когом $\text{Cog}(\tau, P)$, тому можемо записати: $Gv(\tau) = \{\text{Cog}(a, P_a) \rightarrow_e \text{Cog}(b, P_b)\}_\tau$ [Context]. Таке представлення розкриває взаємозв'язок між образним та знаковим мисленням. Спільно, всі аспекти представлення ког-числа формують «number sense». Мисленні обчислення та оцінки є базовими повсякденними навичками, які необхідні для реальних арифметичних операцій та розуміння чисел (Mental Computation and Estimation). Вправи з мисленням обчисленням-узагальненням сприяють розвитку математичних концепцій та навичок, таких як більш глибоке почуття чисел та численні когнітивні здібності, включаючи гнучкість вибору стратегій (наприклад, рівнів абстракції та аналогій), що використовуються у різних математичних, фізичних та інженерних областях.

Порівняйте концепції «ког-числа», «нечіткого числа», «р-адичного числа» та «сірого числа» (концепти «інтелектуальної математики»). Будь-який природний сенсор формує ког-числа, ког-знаки. На кожній стадії розвитку версії ког-числа формально різняться між собою (мають різну схему інтерпретації або схему смислового моделювання). З цієї причини ког-числа x близьких за змістом тестів різняться в різних суб'єктах.

Описуючи стан довільної динамічної системи вектором когнітивних чисел отримуємо ПГУ-концепцію багатомасштабного фазового простору. Кожен масштабний рівень формує свій малюнок поведінки динамічної системи з урахуванням своєї фізики. Всі масштаби існують одночасно.

Кожне ког-число в рамках заданого тесту має власну мережу начерків-інтерпретацій. Об'єднавши у кожен момент часу мережі начерків всіх чисел-значень тесту А, що розвиваються, отримаємо загальну мережу начерків тесту, яку, у ряді випадків, можна представити у вигляді мережі доменів тесту $G(A)$ (приклад на рис. 2; круги – це домени, знизу вгору йде узагальнення; базовий/нижній домен числовий; Temporal morphogenesis) [1].

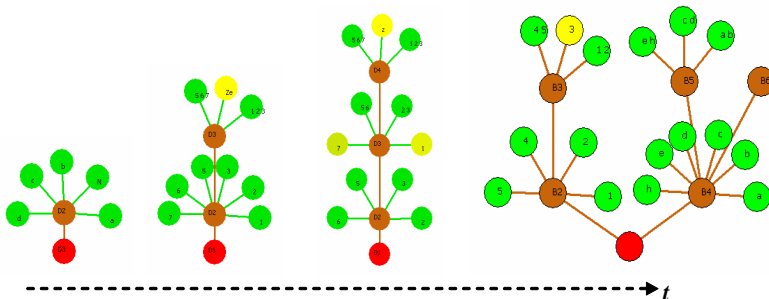


Рис.2 - Приклад стадій структурної інфляції мережі начерків тесту $G(A)$

«Мислення – це оперування Смыслами». Сукупність мереж доменів тестів разом з індукторним простором та СПЧД формують смислову систему координат будь якого дослідження або динамічної системи [4]. Концепція 'ког-числа' свідчить про те, що має місце спонтанна синтетична діяльність, яка дозволяє дитині зрозуміти або, швидше, поступово отримати інтуїцію, що в багатьох процесах сприйняття та мислення задіяна одна й та сама регулююча концепція – 'число' (Complexity of numbers: Phenomenology / Essence of Numbers).

Наведемо «Математичну модель породження смислу сигналу / змінної / ког-чисел» (sense-making) або «Модель адаптивних процесів та режимів усвідомлення сенсорних сигналів у когнітивних системах» (коментар в [1]):

```

Test [#TestX...] [{a/A}] [Context] {
  Dn [^Dn...] [#Dx...] { ; ; } [{On}_n] [{Sg}_n] [{S, R}_n] [{NN}_n]
  [{Ag}_n] [{LAoT}_n]
  ...
  D2 [^D2...] [#Dy...] { ; ; } [{On}_2] [{Sg}_2] [{S, R}_2] [{NN}_2]
  [{Ag}_2] [{LAoT}_2]

```

$$D1 [\wedge D1...] \{ ; ; \} \quad \{ \{ On \}_1 \} \{ \{ Sg \}_1 \} \{ \{ S, R \}_1 \} \{ \{ NV \}_1 \} \\ \{ \{ Ag \}_1 \} \{ \{ LAoT \}_1 \}, \\ \forall \{ ; ; \} = \{ z_1 [\wedge x_1...] [Int_1] \{ \{ g/\mu \}_1 \}; \dots ; z_k [\wedge x_k...] [Int_k] \{ \{ g/\mu \}_k \},$$

де D_j – задачі розрізнення, а $\{LAoT\}_j$ – досвід («стріли часу» СПЧД). Це загальна модель будь якого когнітивного сенсору в холістичній парадигмі.

Висновок. Концепції нескінченності і критичності представлень / знань та ресурсної гнучкості у задачах розрізнення дозволяє об'єктивно оцінити «живу (розумну) структуру» (мережі начерків, ког-числа) з погляду її краси чи життєвості: «стріли пізнання», критичні начерки; ситуативний простір можливостей, мульти-формалізм, предикативність, колективність, конкурентність, динамічність, ресурсна ефективність.

З банком тестів $\{Gv(z)\}$ безпосередньо пов'язана когнітивна складність, що визначається розмірністю та різноманітністю семантичного простору, яким оперує той чи інший суб'єкт/агент. Люди з більшою когнітивною складністю легше переносять кризові ситуації та навантаження, швидше знаходять вихід із них, завдяки більшому розмаїттю евристик у задачах розрізнення.

Література

1. Прокопчук Ю.А. (2022). Интуиция: опыт формального исследования. Днепр: ГВУЗ «ПГАСА». 724 с.
2. Прокопчук Ю.А. (2017). Набросок формальной теории творчества. Днепр: ГВУЗ «ПГАСА». 452 с.
3. Prokopchuk Y., Nosov P., Ben A. (2024). Problems of Meaning, Understanding, Computability and Adaptability in Artificial Cognitive Systems. Proceedings of the 16th Scientific and Practical Conference «Modern Information and Innovative Technologies in Transport (MINTT-2024)» (May 29-30, 2024, Odesa). Odesa, Ukraine: Kherson State Maritime Academy. P. 19 – 25.
4. Прокопчук Ю.О. (2023). Комбінаторне Узагальнення: Моделі Критичних Знань // Комбінаторні конфігурації та їхні застосування: Матеріали XXV Міжнародного науково-практичного семінару імені А.Я. Петренюка (Запоріжжя – Кропивницький, 14-16 червня 2023 року). Запоріжжя: НУ «Запорізька політехніка». С. 157 – 162.
5. Прокопчук Ю.О. (2023). Когнітивні Системи: Математичні Моделі Краси та Цілісності. Satellite conference "Information technologies and applications (IT&Is)" (Taras Shevchenko National

University of Kyiv, November 21, 2023). – Kyiv: Publishing House «Caravela». С. 334 – 337.

6. Alexander, C. (2002–2005). The Nature of Order: An Essay on the Art of Building and the Nature of the Universe. Berkeley, CA: Center for Environmental Structure.

7. Jiang, Bin & Huang, Ju Tzu. (2021). A New Approach to Detecting and Designing Living Structure of Urban Environments. Computers Environment and Urban Systems. 88(4). 10.1016/j.compenvurbsys.2021.101646.

ТЕОРЕТИКО-ГРАФОВА ПОСТАНОВКА ЗАДАЧІ ФОРМУВАННЯ ЦІЛЬОВИХ ГРУП НА ОСНОВІ ФАКТОРА ЗМАГАННЯ

Антон Рябенко, Еліна Терещенко
Національний університет “Запорізька політехніка”

GRAPH-THEORETICAL STATEMENT OF THE FORMING TARGET GROUPS PROBLEM ON THE BASIS OF THE COMPETITION FACTOR

Abstract. The authors conducted a study of a number of the best options for the formation of target groups with given parameters, taking into account the positive practical experience of interaction between participants.

Цільова група – це колектив, який виконує діяльність спільно-взаємодіючого типу для досягнення конкретної мети. Змістовна постановка завдання формування цільової групи (ЦГ) має таке формулювання. Для виконання конкретної цільової програми необхідно сформувати колектив (бригаду, екіпаж, творчий колектив) з деякої вже існуючої множини фахівців різного профілю. Виробнича діяльність ЦГ передбачає інформаційний обмін та взаємодію між її учасниками. Основна вимога до роботи колективу: ефективне виконання поставленого завдання [1-3].

Змістовна постановка полягає в тому, щоб провести серію змагань між командами виконавців, щоразу варіюючи склад цих команд.

Кількість учасників у командах має бути більшою, ніж кількість учасників у цільових групах, які ми плануємо сформувати

в результаті. Після проведення змагальної серії ми отримаємо інформацію про склади команд та їхні результати, ґрунтуючись на якій оцінимо потенційну успішність взаємодії кожної пари виконавців, що може бути використане при формуванні цільових груп.

Сформулюємо математичну постановку задачі формування цільових груп на повному неорієнтованому графі $G = (O, E)$.

Множина вершин $O = \{o_i | i = \overline{0, n-1}\}$ відповідає множині потенційних виконавців. Кожне ребро $e_{ij} \in E$ має чисельну вагу w_{ij} , яка відбиває якість взаємодії двох виконавців o_i та o_j .

Далі буде детально описана методика розрахунку цих значень. Допустимим розв'язком задачі є підграф $x = (O, E)$, $O \subset O, E \subset E$, який складається з максимально можливої кількості компонент зв'язності, кожна з яких має потужність k . Кожна така компонента є повним графом, і вона представляє групу виконавців. Якщо n не є кратним k , то до x не будуть входити r вершин, де r - остача від ділення n на k .

Множина усіх допустимих розв'язків (МДР) на графі $G = (O, E)$ позначимо через $X = X(G) = \{x\}$.

На МДР визначено векторну цільову функцію (ВЦФ):

$$F = (F_1, F_2),$$

що складається з таких критеріїв:

$$MAXSUM: F_1 = \sum_{e_{ij} \in E} w_{ij} \rightarrow \max,$$

$$MAXMIN: F_2 = \max_{e_{ij} \in E} w_{ij} \rightarrow \max.$$

Перший критерій оцінює загальну успішність. Другий критерій оцінює “найслабкішу ланку”, дозволяючи обрати варіант, при якому максимізуються мінімальні ваги ребер в кожній компоненті зв'язності.

ВЦФ визначає в МДР паретівську множину (ПМ) X^* , що складається з усіх паретівських оптимумів. Будемо розглядати алгоритмічну проблему знаходження повної множини альтернатив (ПМА) X^0 , яка є підмножиною ПМ мінімальної потужності і при цьому виконується умова $F(X^0) = F(X^*)$ [4].

Серія змагань буде складатися з окремих змагань, спланованих таким чином, щоб кількість переможців була не менше третини загальної кількості команд. Наприклад, якщо маємо 12 виконавців, то для команд з чотирьох учасників рекомендується проводити змагання з однією командою-переможцем. А якщо треба формувати

команди з трьох учасників, тоді більш доцільно обирати двох переможців в двох парах команд. Це робиться для того, щоб доля перемог та програшів була порівняною.

При плануванні серії змагань важливо створити склади команд таким чином, щоб кожна можлива пара виконавців була декілька разів в одній команді. Тобто при такому плануванні треба враховувати ресурс часу — скільки серій змагань є можливість провести. За певних умов можна відмовитися від цього обмеження. Зауважимо, що дослідження такого планування має окремий науковий інтерес.

В таблиці 1 наведено розподіл 8 гравців за двома командами, таким чином, щоб кожна пара гравців грала за одну команду не менше двох разів. Гравці позначені індексами від 0 до 7 включно. Серія складається з 8 змагань. Також зауважимо що в цьому прикладі максимальна кількість входжень в одну команду для пари (o_0, o_2) дорівнює 5, а середнє значення кількості дорівнює 3.43.

Для розрахунку реберної ваги w_{ij} будемо використовувати формулу:

$$w_{ij} = \frac{\bar{N}_{ij}}{N_{ij}},$$

де \bar{N}_{ij} - кількість перемог команд, до складу яких входили виконавці o_i та o_j ,

N_{ij} - загальна кількість команд, до складу яких входили виконавці o_i та o_j .

Таблиця 1

Розподіл гравців за 2 командами

№	Склад команди 1	Склад команди 2	Переможець
1	0, 2, 3, 4	1, 5, 6, 7	команда 2
2	0, 2, 3, 6	1, 4, 5, 7	команда 2
3	0, 1, 2, 3	4, 5, 6, 7	команда 2
4	0, 1, 3, 5	2, 4, 6, 7	команда 2
5	0, 2, 6, 7	1, 3, 4, 5	команда 1
6	0, 1, 4, 6	2, 3, 5, 7	команда 1
7	0, 1, 2, 5	3, 4, 6, 7	команда 2
8	0, 4, 5, 7	1, 2, 3, 6	команда 2

Наприклад $w_{26} = 0.750$, тому що пара (o_2, o_6) брала участь в чотирьох змаганнях (за номерами 2, 4, 5 та 8), з яких 3 були для неї переможні (4, 5 та 8).

Якщо наша мета в цьому прикладі полягала в створенні двох цільових груп по три виконавця з наданої множини восьми кандидатів (тобто $n = 8, k = 3$), то для формування МДР нам потрібно сформувати перелік усіх можливих варіантів формування такої структури. Таких варіантів 280, що визначає потужність МДР.

Після цього треба виконати розрахунки значень ВЦФ для усіх елементів МДР. Візуалізація цих розрахунків надана на рис. 1.

Маємо два розв'язки, які утворюють ПМ та ПМА: $x_{229} = ((o_1, o_2, o_7), (o_4, o_5, o_6))$, $x_{260} = ((o_1, o_5, o_6), (o_2, o_4, o_7))$. На рис. 1 виділено маркери, що відповідають знайденому ПМА.

Сформовано множину кращих варіантів формування цільових груп із заданими параметрами з урахуванням позитивного практичного досвіду взаємодії між учасниками.

Розроблена математична модель дає змогу формалізувати вибір варіантів цільової групи з найбільш продуктивно працюючих учасників.

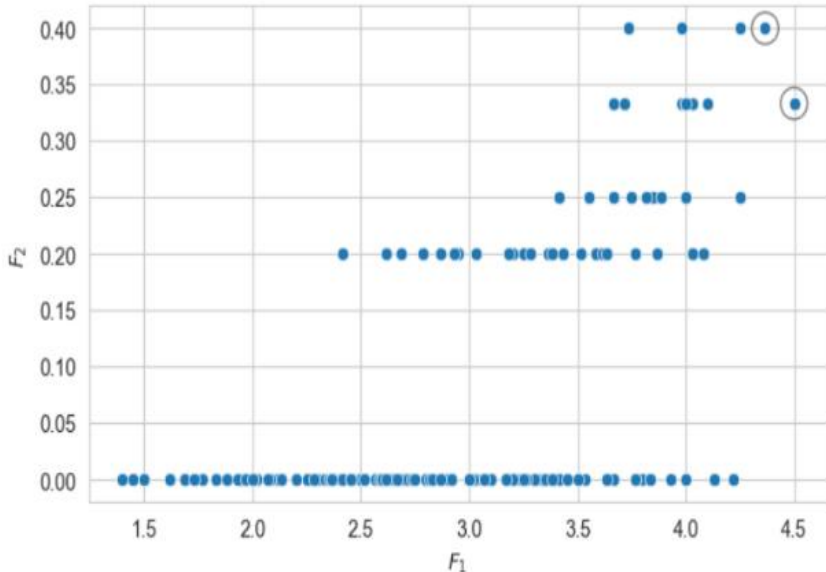


Рис.1 – Значення ВЦФ для МДР ($n = 8, k = 3$)

Література

1. Перепелица В. А. Многокритериальные модели и методы для задач оптимизации на графах. Saarbrücken: LAP LAMBERT Academic Publishing, 2013. 336 с.
2. Перепелица В. О., Рябенко А. Є. Оцінки надійності в задачах формування цільових груп. *Вісник Запорізького державного університету. Фізико-математичні науки. Біологічні науки.* 2000. № 1. С. 90–93.
3. Бакурова А.В. Самоорганізація соціально-економічних систем: моделі і методи: монографія. Запоріжжя: КПУ, 2010. 328 с.
4. Рябенко А.Е., Терещенко Э.В. 3-критериальная модель формирования целевых рабочих групп. *Бизнес Информ.* – 2016. - №10. – С.129-134.

НЕЧІТКА МОДЕЛЬ КЛАСТЕРІВ НА ОСНОВІ МУЛЬТИНЕЧІТКИХ МНОЖИН

Ірина Рясна, Олександр Сенько

Інститут кібернетики імені В. М. Глушкова НАН України

FUZZY CLUSTER MODEL BASED ON MULTI-FUZZY SETS

Abstract. The paper considers the problem of clustering under qualitative information. Based on the theory of fuzzy sets, the problem is analyzed and its fuzzy cluster model is proposed. The formal concept of a multi-fuzzy cluster is introduced. An approach to clustering a set of qualitative data based on fuzzy combinatorial configurations using a fuzzy similarity relation, a fuzzy kernel, and the concept of multi-fuzzy sets is proposed.

Нехай задано множину об'єктів $X = \{x_i\}_{i=1}^n$ та множину $W = \{w_v\}_{v=1}^m$ їхніх якісних ознак, яка не містить характеристичну ознаку множини X .

Нехай задана відповідність $\psi: X \rightarrow W$, що визначає матрицю «об'єкт-ознака» Ψ . Вважаємо, що кожен $x_i \in X$ має непустий образ $\psi(x_i) \subseteq W$, а кожна ознака $w_v \in W$ має непустий прообраз $\psi^{-1}(w_v) \subseteq X$. Вочевидь, що $\psi(x_i)$ – чітка підмножина множини W , $i \in \{1, \dots, n\}$.

Відповідність ψ породжує відношення еквівалентності \approx на $X \times X$:

$$x_i \approx x_j \Leftrightarrow \psi(x_i) = \psi(x_j), \quad (1)$$

тобто якщо $x_i, x_j \in X$ мають одну і ту ж сукупність ознак, то вони є еквівалентними.

Уведемо бінарне відношення $T = \Psi \times \Psi'$, де Ψ' – транспонована матриця, \times – добуток матриць, $t(x_i, x_j) = |\psi(x_i) \cap \psi(x_j)|$ – потужність перетину образів об'єктів $x_i, x_j \in X$. Відношення T – рефлексивне та симетричне, а $t(x_i, x_i) \geq t(x_i, x_j)$; $t(x_i, x_i)$ – кількість (ненульових) ознак об'єкту x_i ; $t(x_i, x_j) = 0 \Leftrightarrow \psi(x_i) \cap \psi(x_j) = \emptyset$.

Покладемо $\pi(x_i, x_j) = t(x_i, x_j) / |\psi(x_i) \cup \psi(x_j)|$, $i, j \in \{1, \dots, n\}$; тоді маємо R_π : $(x_i, x_j) \in R_\pi \Leftrightarrow \pi(x_i, x_j) > 0$; $\pi: X \times X \rightarrow [0, 1]$.

Вочевидь, що відношення R_π є відношенням подібності, а функція $\mu_i(x_j) = \pi(x_i, x_j)$ є мірою подібності елементів $x_j \in X$, $j \in \{1, \dots, n\}$, з елементом $x_i \in X$.

Нечітку підмножину $C_{x_i} = \left\{ \left(x_j, \mu_{C_{x_i}}(x_j) \right) \right\}_{j=1}^n = \left\{ \left(x_j, \mu_i(x_j) \right) \right\}_{j=1}^n$, $\mu_i(x_j) = \pi(x_i, x_j)$, $i \in \{1, \dots, n\}$, назовемо нечітким C_{x_i} -кластером.

Позначимо $\mathbf{C} = \{C_{x_i}\}_{i=1}^n$. Задамо відображення $\xi: \xi(x_i) = C_{x_i}$, $\xi^{-1}(C_{x_i}) = x_i$, $x_i \in X$, $C_{x_i} \in \mathbf{C}$. Неважко показати, якщо ядра кластерів C_{x_i} , C_{x_j} перетинаються, то $C_{x_i} = C_{x_j}$, і $x_i \approx x_j$, тобто

$$x_i \approx x_j \Leftrightarrow C_{x_i} = C_{x_j} \quad x_i, x_j \in X. \quad (2)$$

Згідно класифікації нечітких комбінаторних конфігурацій [1, 2] отримуємо нечіткі комбінаторні конфігурації другого порядку третього типу, для яких твірною множиною є векторнозначна нечітка множина $\tilde{X} = \left\{ (x_i, \bar{\mu}_i) \right\}_{i=1}^n$, де $\bar{\mu}_i = \left(\mu_i(x_j) \right)_{j=1}^n$.

Множину C подаємо у вигляді мультимножин таким чином. Нехай $K = \{K_l\}_{l=1}^p$ визначена за (1) множина класів еквівалентності множини X . Введемо відображення $\xi'(K_l)$, таке, що $\xi'(K_l) = \{C_{x_i} \mid x_i \in K_l\} = Q_l$, де Q_l – образ класу K_l , який є множиною рівних між собою C_{x_i} -кластерів. Позначимо $|K_l| = k_l$, тоді з урахуванням (2) має місце рівність: $|K_l| = |Q_l|$. Нехай u_l – еталон класу еквівалентності K_l , позначимо як $U = \{u_l\}_{l=1}^p$ множину обраних еталонів отриманих класів еквівалентності; $u_l \neq u_q$, $l \neq q$. Множину U назовемо універсумом мультимножин. Позначимо $a_{lq} = \pi(x_i \in K_l, x_j \in K_q) = \pi(u_l, u_q)$, $a_{lq} \in [0, 1]$, $l, q \in \{1, \dots, p\}$. Вочевидь, якщо $l \neq q$, то $a_{lq} = a_{ql}$; $a_{ll} > a_{lq}$. Матрицю (a_{lq}) , $l, q \in \{1, \dots, p\}$, назовемо нечіткою матрицею подібності елементів універсума U .

Нехай $x_i \in K_l$, тоді

$$C_{(x_i \in K_l)} = \left\{ \left(k_q \square (u_q, a_{lq}) \right) \right\}_{q=1}^p = \left\{ k_1 \square (u_1, a_{l1}), \dots, k_p \square (u_p, a_{lp}) \right\},$$

$$k_1 + \dots + k_p = n.$$

Таким чином, кластер подано як нечітку мультимножину на універсумі U .

Уведемо такі операції:

$$C_{x_i} \cup C_{x_l} = \left\{ \left(x_j, \max(\mu_i(x_j), \mu_l(x_j)) \right) \right\}_{j=1}^n, \quad \forall x_i, x_l \in X;$$

$$C_{x_i} + C_{x_l} = \left\{ \left(x_j, \mu_i(x_j) \right), \left(x_j, \mu_l(x_j) \right) \right\}_{j=1}^n, \quad \forall x_i, x_l \in X;$$

$$k \square C_{x_i} = \left\{ k \square \left(x_j, \mu_i(x_j) \right) \right\}_{j=1}^n, \quad \forall x_i \in X.$$

Подамо кластер як мультинечітку множину [3, 4].

$$\text{Покладемо } k_q \square (u_q, a_{lq}) = (u_q, (k_q \square a_{lq})), \text{ тоді } \forall C_{(x_i \in K_l)} \in Q_l \quad C_{(x_i \in K_l)} =$$

$$= \left\{ k_1 \square (u_1, a_{l1}), \dots, k_q \square (u_q, a_{lq}), \dots, k_p \square (u_p, a_{lp}) \right\} = \left\{ (u_1, (k_1 \square a_{l1})), \dots, (u_p, (k_p \square a_{lp})) \right\}.$$

$$\text{Позначимо } \vec{k} = (k_1, \dots, k_p), \quad \vec{a}_l = (a_{l1}, \dots, a_{lq}, \dots, a_{lp});$$

$C_{(x_i \in K_l)} = (u_l, \bar{k} \otimes \bar{a}_l)$, де \otimes – операція, яка визначає “покомпонентний добуток” векторів:

$$\bar{k} \otimes \bar{a}_l = (k_1, \dots, k_q, \dots, k_p) \otimes (a_{l1}, \dots, a_{lq}, \dots, a_{lp}) = (k_1 \sqcap a_{l1}, \dots, k_q \sqcap a_{lq}, \dots, k_p \sqcap a_{lp}).$$

$$\text{Тоді, } Z_{Q_l} = \sum_{x_i \in K_l} C_{x_i} = k_l \sqcap (u_l, \bar{k} \otimes \bar{a}_l); \quad |Z_{Q_l}| = \left| \sum_{x_i \in K_l} C_{x_i} \right| = k_l \sum_{q=1}^p k_q a_{lq},$$

а множину мультинечітких множин $\mathbf{C} = \{C_{x_i}\}_{i=1}^n$ запишемо так:

$$Z = \{Z_{Q_l}\}_{l=1}^p = \{k_l \sqcap (u_l, \bar{k} \otimes \bar{a}_l)\}_{l=1}^p.$$

Нечітким ядром рівня α C_{x_i} -кластера назовемо нечітку множину $\Omega_{C_{x_i}}^\alpha = \{(x_j, \mu_i(x_j)) \mid \mu_i(x_j) \geq 1 - \alpha, \alpha \in [0, 1]\}$.

Межами рівня β C_{x_i} -кластера з нечітким ядром рівня α назовемо множину $Bound_\beta C_{x_i} = \{x_j \in X \mid 0 < \beta \leq \mu_{C_{x_i}}(x_j) < 1 - \alpha\}$.

Нехай $x_i \in K_l$, $Q_l = \xi'(K_l)$, $C_{x_i} \in Q_l$, $Ker_\alpha C_{x_i} = \{x_t \mid \mu_{C_{x_i}}(x_t) \geq 1 - \alpha\}$ – множина елементів у носії нечіткого ядра рівня α C_{x_i} -кластера; J_i – перелік номерів елементів, і $r(i)$ – відповідно, кількість елементів у носії нечіткого ядра C_{x_i} -кластера; $t \in J_i = \{j_{i_1}, \dots, j_{i_{r(i)}}\}$.

Мультинечітким кластером назовемо нечітку множину $S_i(\alpha, \beta) = \bigcup_{t \in J_i} C_{x_t}$, $x_t \in Ker_\alpha C_{x_i}$, $0 < \beta < 1 - \alpha$, функцію належності якої обчислюємо за формулою

$$\mu_{S_i(\alpha, \beta)}(x_j) = \begin{cases} 1, & 1 - \alpha \leq \mu_{C_{x_i}}(x_j) \leq 1, \\ \max_j \{\mu_{C_{x_t}}(x_j), t \in J_i\}, & \beta \leq \mu_{C_{x_i}}(x_j) < 1 - \alpha, \\ 0, & 0 \leq \mu_{C_{x_i}}(x_j) < \beta, t \in J_i. \end{cases}$$

Висновки. Потужність ядра заданого рівня α мультинечіткого кластера обчислюємо за формулою $|\Omega_{C_{x_i} \in Q_l}^\alpha| = \sum_{q=j_{i_1}}^{j_{i_{r(i)}}} k_q a_{lq}$. Потужність носія нечіткого ядра рівня α мультинечіткого кластера

– за формулою: $\left| \text{Ker}_\alpha C_{(x_i \in Q_i)} \right| = \sum_{q=j_h}^{j_{r(i)}} k_q$. Подання моделі нечітких кластерів на основі мультинечітких множин використовуємо у подальших дослідженнях з метою адекватного визначення нечітких понять неформальних моделей при формалізації та розв'язанні задач кластеризації з урахуванням подібності та потужності класів еквівалентності.

Література

1. Гуляницький Л.Ф., Рясна І.І. До формалізації задач комбінаторної оптимізації на нечітких множинах. *Теорія оптимальних рішень*. Київ: Ін-т кібернетики імені В.М. Глушкова НАН України. 2016. С. 17 – 25.
2. Hulianytskyi L.F., Riasna I.I. Formalization and classification of combinatorial optimization problems. *Optimization Methods and Applications (eds. Butenko S., Pardalos P. M., Shylo V.)*. Cham: Springer International Publishing AG, 2017. P. 239 – 250.
3. Sebastian S., Ramakrishnan T.V. Multi-fuzzy sets: an extension of fuzzy sets. *Fuzzy Inf. Eng.* 2011. 1. P. 35–43. DOI 10.1007/s12543-011-0064-y.
4. Hong-Yu Wang, Jie-Sheng Wang, Guan Wang. Multi-fuzzy clustering validity index ensemble: A Dempster-Shafer theory-based parallel and series fusion. *Egyptian Informatics Journal*. 2023. v 24. № 4. 100417. <https://doi.org/10.1016/j.eij.2023.100417G>

ДИСТАНЦІЙНІ МАГІЧНІ РОЗМІТКИ І ГРАФИ КЕЛІ

Марина Семенюта, Сергій Якименко

Центральноукраїнський національний технічний університет

DISTANCE MAGIC LABELINGS AND KELI GRAPHS

Abstract. . We have found conditions for the existence of distance magic markings for certain types of Kelly graphs on the diedra group D_{2n} .

Існують різні типи магічних розміток графів, їх опис і результати відносно рішення задач існування, побудови і переліку можна знайти в [1]. Для деяких розміток використовували різні терміни або навпаки різні розмітки мають однакову назву. В даній роботі розглядатимемо розмітку, яку ввів V. Vilfred у 1994 році і

назвав її Σ -розміткою [2], у 2003 цю розмітку незалежно від нього запропонували М. Miller, С. Rodger, R. Simanjuntak, використавши для неї термін «1-вершинно-магічна вершинна» [3], а у 2009 році – S. Arumugam, D. Froncek, N. Kamatchi – під назвою «дистанційна магічна» [4,5]. Останній термін набув найбільшої популярності і вживається в науковій літературі.

Будемо розглядати звичайні графи, тобто скінчені неорієнтовані графи, що не містять кратних ребер та петель. Нехай задано граф $G = (V, E)$ і множина $D \subseteq \{0, 1, 2, \dots, \text{diam}(G)\}$, де $\text{diam}(G)$ – діаметр графа G . Задано відображення $f: V(G) \rightarrow \{0, 1, 2, \dots, |V|\}$, яке називають розміткою графа G . D -околом вершини v називають множину $N_D(v)$, її елементами є вершини, що знаходяться на відстані $d \in D$ від вершини v . Під D -вагою $w(v)$ (або $w_f(v)$) вершини v при розмітці f розуміємо суму міток вершин D -околу вершини v , тобто $w(v) = \sum_{u \in N_D(v)} f(u)$, де $v \in V(G)$.

Граф G порядку n називають D -дистанційним магічним, якщо існує бієкція $f: V(G) \rightarrow \{1, 2, \dots, n\}$ і така стала k , що для кожної вершини $v \in V(G)$ її D -вага дорівнює k , тобто $w(v) = k$. Число k називають D -дистанційною (або дистанційною) магічною сталою розмітки f [6].

При $D = \{1\}$ розмітку f графа G називають *дистанційною магічною* [5].

Одним з найбільш розповсюджених прикладів застосування дистанційної магічної розмітки є організація неповних кругових турнірів [7].

В останні роки теорія графів Келі розвивається як окрема частина алгебраїчної теорії графів. Графи Келі знаходять застосування в теорії груп і теорії графів при розв'язанні задач класифікації, ізоморфізму груп і графів, розфарбовування графів, переліку. Крім того, існує багато цікавих застосувань графів Келі, зокрема їх використання в біології представлено в роботі [8], а в інформатиці – в [9].

Мотивацією для аналізу графів Келі стали роботи [10-12]. Метою даної роботи є розв'язання задачі існування дистанційної магічної розмітки для окремих типів графів Келі на групі діедра D_{2n} .

Розглянемо скінчену групу \mathbf{G} з породжувальною множиною $S \subset \mathbf{G}$, де множина S не містить одиничний елемент і є симетричною, тобто $S = S^{-1}$, $S^{-1} = \{s^{-1} | s \in S\}$.

В графі Келі $\Gamma = \text{Cay}(\mathbf{G}, S) = (V, E)$ вершини відповідають елементам групи, а ребра – добутку елемента групи справа на

елемент S . В даному випадку Γ – звичайний граф, він є $|S|$ -регулярним і вершинно-транзитивним.

Група діедр (діедральні група) D_{2n} – циклічна група порядку $2n$, фактично маємо

$$D_{2n} = \{a, b | a^n = b^2 = e, bab = a^{-1}\} = \{e, a, a^2, \dots, a^{n-1}, b, ab, \dots, a^{n-1}b\}.$$

Нехай $S \subset D_{2n}$. Розглянемо приклади графів Келі $\Gamma = \text{Cay}(D_{2n}, S)$ для $|S| = 1; 2; 3$ і дослідимо їх на наявність дистанційної магічної розмітки.

Приклад 1. $|S| = 1$, тоді граф Келі $\Gamma = \text{Cay}(D_{2n}, S) = nK_2$ не є дистанційним магічним.

Приклад 2. $S = \{a, a^{-1}\}$, де $a \neq a^{-1}$, маємо $\Gamma = \text{Cay}(D_{2n}, S) = \frac{2n}{m} C_m$ і $o(a) = m$.

Приклад 3. $S = \{a, b\}$, де $a^2 = b^2 = e$, маємо $\Gamma = \text{Cay}(D_{2n}, S) = \frac{2n}{m} C_{2m}$, і $o(ab) = m$.

У прикладах 2 та 3 граф Γ буде дистанційним магічним, коли кожна його компонента є циклом C_4 [13]. З цього випливає, що n повинно бути парним.

Приклад 4. $D_{2n} = \langle a, b | a^n = b^2 = e, bab = a^{-1} \rangle$, $S = \{a, a^{-1}, b\}$, де $2n \geq 6$, маємо $\Gamma = \text{Cay}(D_{2n}, S) = K_2 \times C_n$ (рис.1).

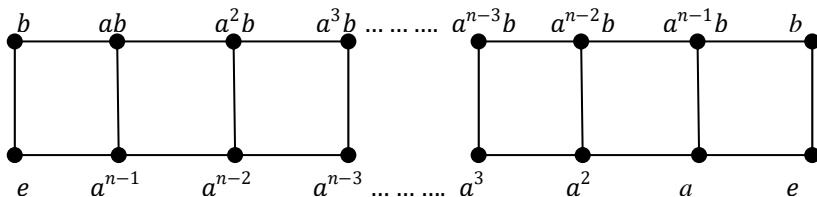


Рис.1 – $\text{Cay}(D_{2n}, S) = K_2 \times C_n$

Розглянемо, множини суміжності вершин a^{n-2} і ab графа $\Gamma = K_2 \times C_n$: $N(a^{n-2}) = \{a^{n-1}, a^{n-3}, a^2b\}$, $N(ab) = \{b, a^2b, a^{n-1}\}$. Одержимо

$$|N(a^{n-2}) \cap N(ab)| = \deg(a^{n-2}) - 1 = \deg(ab) - 1 = 2,$$

де $\deg(a^{n-2})$, $\deg(ab)$ – степені відповідних вершин. Отже, цей граф не є дистанційним магічним [3].

Приклад 5. $S = \{a, a^{-1}, b\} \subset D_{2n}$, де $a \neq a^{-1}$, $b^2 = e$, тоді маємо $\Gamma = \text{Cay}(D_{2n}, S) = \frac{n}{m} (K_2 \times C_n)$.

Приклад 6. $S = \{a, b, c\} \subset D_{2n}$, де $n \geq 3$, $|S| = 3$ і $a^2 = b^2 = c^2 = e$, тоді маємо $\Gamma = Cay(D_{2n}, S) = K_2 \times C_n$.

Для будь якого графа з прикладів 5 і 6 знайдуться дві вершини x і y , для яких $|N(x) \cap N(y)| = \deg(x) - 1 = \deg(y) - 1 = 2$, тому ці графи не є дистанційним магічним [3].

Таким чином, нами доведено наступне твердження.

Твердження. Графи $\Gamma = Cay(D_{2n}, S)$ при

1) $|S| = 1$;

2) $S = \{a, a^{-1}, b\} \subset D_{2n}$, де $a \neq a^{-1}$, $b^2 = e$;

3) $S = \{a, b, c\} \subset D_{2n}$, де $n \geq 3$, $|S| = 3$ і $a^2 = b^2 = c^2 = e$

не є дистанційними магічними.

Граф $\Gamma = Cay(D_{2n}, S)$, де $S = \{a, a^{-1}\}$, $a \neq a^{-1}$ і $o(a) = m$ є дистанційним магічними тоді і тільки тоді, коли $m = 4$ і n – парне.

Граф $\Gamma = Cay(D_{2n}, S)$, де $S = \{a, b\}$, $a^2 = b^2 = e$ і $o(ab) = m$ є дистанційним магічними тоді і тільки тоді, коли $m = 2$ і n – парне.

Подальші дослідження можуть бути пов'язані з пошуком умов існування D -дистанційних магічних розміток графа Келі на групі діедра.

Література

1. J.A. Gallian, A dynamic survey of graph labeling. The Electronic Journal of Combinatorics. DS6: Dec 1, 2023. 644 p.

2. V. Vilfred, Σ -labelled graph and Circulant Graphs, Ph.D. Thesis, University of Kerala, Trivandrum, India, 1994.

3. M. Miller, C. Rodger, R. Simanjuntak, Distance magic labelings of graphs. *Australasian Journal of Combinatorics*. 2003. Vol. 28. P. 305-315.

4. S. Arumugam, D. Froncek, N. Kamatchi, Distance magic graphs – a survey. *Journal of the Indonesian Mathematical Society. Special Edition*. 2011. P. 11-26.

5. K.A. Sugeng, D. Froncek, M. Miller, J. Ryan and J. Walker, On distance magic labeling of graphs, *J. Combin. Math. Combin. Comput.*, 71 (2009), 39–48.

6. A. O'Neal, P. Slater, An introduction to distance D magic graphs. *Journal of the Indonesian Mathematical Society. Special Edition*. 2011. P. 89-107.

7. М.Ф. Семенюта, З.А. Шерман, О.Н. Дмитриев Неполные турниры и магические типы разметок. Управляющие системы и машины. 2018. №5 (277). С.13-24.

8. V. Bafna, P. A. Pevzner, Genome rearrangements and sorting by reversals. SIAM Journal on computing, vol. 25, no. 2, pp. 272-289, 1996.

9. A. Kelarev, J. Ryan, J. Yearwood, Cayley graphs as classifiers for data mining: the influence of asymmetries, Discrete mathematics, vol. 309, no. 17, pp. 5360-5369, 2009.

10. P. Gregor and P. Kovar, Distance magic labelings of hypercubes. Electronic Notes in Discrete Mathematics, 1:145–149, 2013.

11. H. Pfeiffer, Distance Magic Labelings. Master thesis. Dept. of Theoretical Computer Science and Mathematical Logic, faculty of mathematics and physics Charles University.

12. M. Semeniuta, V. Shulhin, Matrices Associated with D-Distance Magic Graphs and Their Properties. Cybernetics and Systems Analysis Volume 55, pages 441–448, (2019).

13. M. Jinnah, On Σ -labelled graphs, In Technical Proceedings of Group Discussion on Graph Labeling Problems eds. B.D. Acharya and S.M. Hedge, (1999), 71-77.

ПРО ВЕРХНІ І НИЖНІ ОЦІНКИ ХРОМАТИЧНОГО ЧИСЛА ДЛЯ ПЛАНАРНИХ І НЕПЛАНАРНИХ ГРАФІВ

¹Семенюта М., ²Шерман З., ²Нестеренко О.

¹Центральноукраїнський національний технічний університет,

²Міжнародний європейський університет

DISTANCE MAGIC LABELINGS AND KELI GRAPHS

Abstract. We considered the problem of estimating the chromatic number of a pair of graphs G and G_1 , satisfying the following conditions: G is not planar; G contains one largest clique K_n of order $n \geq 3$; after removing one edge e in the clique, K_n the graph $G_1 = G - e$ is transformed into a planar one.

Будемо розглядати звичайні графи, тобто скінчені неорієнтовані графи, що не містять кратних ребер та петель. Правильним розфарбуванням графа $G = (V, E)$ називається відображення $c: V \rightarrow N$, таке, що $c(v) \neq c(u)$, якщо $u, v \in V$ і $\{u, v\} \in E$, де $N = \{1, 2, 3, \dots\}$ – множина фарб. Хроматичне число $\chi(G)$ графа G – це мінімальна кількість фарб, яку використано для правильного розфарбування.

Задача пошуку правильного розфарбування графа є однією з основних задач теорії графів. Існують різні методи та алгоритми її розв'язку [1-4]. Одним з напрямів досліджень є обчислення або оцінка хроматичного числа графа за допомогою інших, більш зручних характеристик графа.

Нехай задано зв'язний граф $G = (V, E)$, який задовольняє наступним умовам:

G не є планарним;

G містить одну найбільшу кліку K_n порядку $n \geq 3$;

після видалення одного ребра e в кліці K_n граф $G_1 = G - e$ перетворюється на планарний.

Приклад такого графа G наведено на рис.1. Впевнімося, що він не є планарним. Для цього виконаємо послідовно операцію стягнення ребер у підграфі G' (рис.2) графа G :

стягнемо ребро $\{4, 13\}$, нову вершину позначимо 4;

стягнемо ребро $\{1, 11\}$, нову вершину позначимо 1;

стягнемо ребро $\{2, 10\}$, нову вершину позначимо 10.

Отримали граф K_5 (рис.3). За критерієм Вагнера G не є планарним графом.

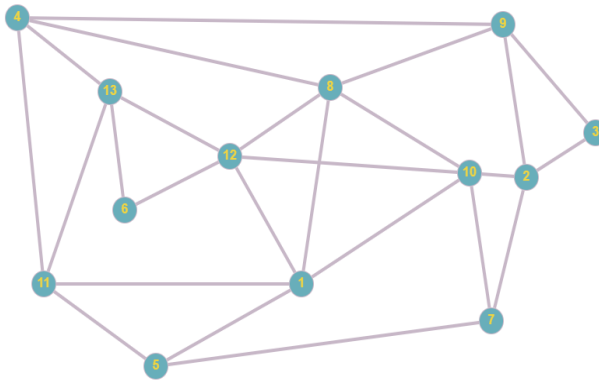


Рис.1. – Граф $G = (V, E)$

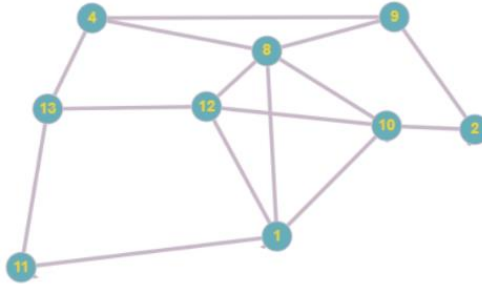


Рис.2. – Підграф G' графа G

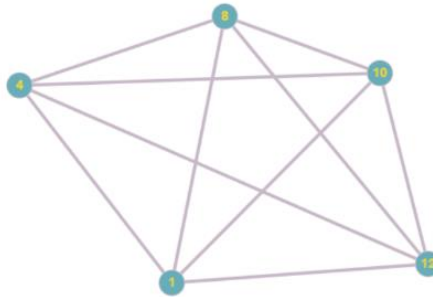


Рис.4. – Граф K_5

Після видалення з графа G одного ребра $\{10,12\}$ (або $\{1,8\}$) одержимо граф $G_1 = G - e$. Він є планарним так як має плоску укладку.

Оцінки хроматичного числа графів G і G_1 наведено в таблиці 1.

Таблиця 1. Нижні і верхні оцінки хроматичного числа графів G і G_1

Граф G	
Нижні оцінки хроматичного числа	Верхні оцінки хроматичного числа
1) Клікове число (або щільність) графа G дорівнює $\varphi(G) = 4$. Тоді для графа G справедливо: $\chi(G) \geq \varphi(G) = 4$.	Так як G – зв'язний граф максимальної степені 5, то за теоремою Брукса $\chi(G) \leq 5$.
$\chi(G) \geq \left\lceil \frac{p}{\alpha(G)} \right\rceil = 3$, де $\alpha(G)$ – число незалежності графа G .	

$\chi(G) \geq \frac{p^2}{p^2 - 2q}; \chi(G_1) \geq 2.$	
Граф G_1	
1) Клікове число (або щільність) графа G_1 дорівнює $\varphi(G_1) = 3$. Тоді для графа G_1 справедливо: $\chi(G_1) \geq \varphi(G_1) = 3.$	G_1 – планарний граф, тоді для його верхньої оцінки одержимо обмеження: $\chi(G_1) \leq 4.$
2) $\chi(G_1) \geq \left\lceil \frac{p}{\alpha(G_1)} \right\rceil = 3$, де $\alpha(G_1)$ – число незалежності графа G_1 .	
3) $\chi(G_1) \geq \frac{p^2}{p^2 - 2q}; \chi(G_1) \geq 2.$	

Отримали $4 \leq \chi(G) \leq 5, 3 \leq \chi(G_1) \leq 4.$

Якщо $A = (a_{ij})$ – матриця суміжності графа порядку p , то $\deg(v_i) = \deg_i = \sum_{j=1}^p a_{ij}$ називають валентністю (степенем) вершини v_i , а $\deg^{(k)}(v_i) = \deg_i^{(k)} = \sum_{j=1}^p a_{ij} \cdot \deg_j^{(k-1)}$ – k -кроковою валентністю вершини v_i .

Для розфарбування графів G і G_1 , використаємо модифікацію алгоритму НП-упорядкування вершин на основі k -крокових валентностей (рис. 5 і 6). НП-алгоритм відноситься до паралельних локальних алгоритмів, так як використовується інформація з локальної пам'яті суміжних вершин і один його крок складається в одночасному та незалежному переобчисленні міток на всіх вершинах графа на основі відомостей, отриманих на попередньому кроці.

Виконаємо упорядкування вершин графів G і G_1 за НП-алгоритмом в порядку незростання їх валентностей. Результати представлені в таблицях 2 та 3.

Таблиця 2. Вершини та їх k -крокові валентності графа G

p	\deg_i	$\deg_i^{(2)}$	p	$\deg_i^{(3)}$	НП-упорядкування вершин
1	5	22	1	94	1
8	5	12	10	92	10
10	5	22	12		12
12	5	21	8		8
2	4	14	4		4

4	4	17	11		11
9	4	15	9	62	13
11	4	16	13	63	9
13	4	15	2		2
5	3	12	5	50	5
7	3	12	7	48	7
3	2	8	6		6
6	2	9	3		3

Таблиця 2. Вершини та їх k -крокові валентності графа G_1

p	$\text{deg}(i)$	$\text{deg}^{(2)}(i)$	p	$\text{deg}^{(3)}(i)$	НП- упорядкування вершин
1	5	20	1		1
8	5	21	8		8
2	4	13	10		10
4	4	17	12		12
9	4	15	4		4
10	4	21	11		11
11	4	16	9		9
12	4	20	13		13
13	4	14	2		2
5	3	12	5		5
7	3	11	7		7
3	2	8	3	28	6
6	2	8	6	34	3

Отже, маємо $\chi(G) = \chi(G_1) = 4$.

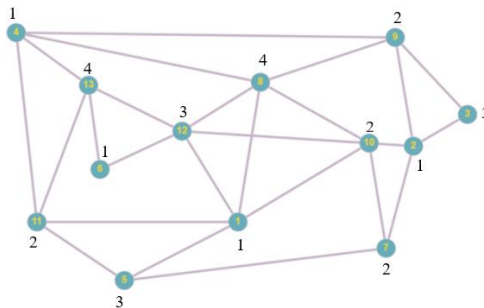


Рис. 5. – Правильне розфарбування граф G за модифікованим алгоритмом з НП-упорядкуванням вершин

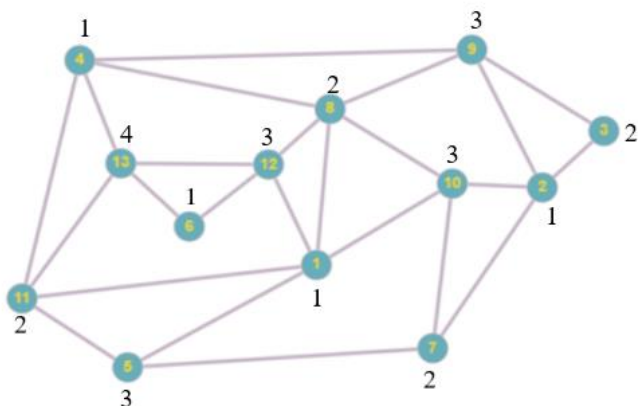


Рис.6. – Правильне розфарбування граф G_1 за модифікованим алгоритмом з НП-упорядкуванням вершин

Постає питання чи для кожної пари графів G і G_1 , що задовольняють вище наведеним умовам, виконується рівність $\chi(G) = \chi(G_1)$.

Література

1. Tutte W. T. Graph Theory. Cambridge University Press, 2001 – 333 p.
2. F. Harary Graph Theory. Addison-Wesley, Reading, Mass., 1969.
3. U. S. R. Murty, Adrian Bondy, Graph Theory: An Advanced Course Springer-Verlag London 2008
4. Reinhard Diestel. Graph Theory. Electronic Edition 2005 Springer-Verlag Heidelberg, New York 2005.
5. Ian Bogle, George M. Slota, Erik G. Boman, Karen D. Devine, Sivasankaran Rajamanickam, Parallel graph coloring algorithms for distributed GPU environments. *Parallel Computing*. Volume 110, May 2022, <https://doi.org/10.1016/j.parco.2022.102896>

АСПЕКТИ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ДВОЕТАПНОЇ ЗАДАЧІ АКТИВАЦІЇ-РОЗМІЩЕННЯ ДЛЯ ПОКРАЩЕННЯ МЕДИЧНОЇ ЛОГІСТИКИ

Олексій Сергєєв

*Національний технічний університет «Дніпровська
політехніка*

ASPECTS OF THE SOFTWARE IMPLEMENTATION OF THE TWO-STAGE ACTIVATION-PLACEMENT TASK FOR IMPROVING MEDICAL LOGISTICS

Abstract. The necessity of studying medical logistics in critical situations arises from the need to adapt to unpredictable crises such as pandemics and natural disasters. Improving the transportation network is a central element for the efficient distribution of medications and medical supplies during emergencies. Historical examples, such as the challenges during the COVID-19 pandemic and the aftermath of military actions, highlight the consequences of logistical shortcomings in the distribution of medicines, whereas successful models demonstrate the effectiveness of well-structured transportation systems.

This study analyzes the current medical logistics system in Ukraine to identify weaknesses and potential improvements. To implement them, a mathematical model of a multi-stage activation-placement problem is proposed. This model aims to optimize the logistics infrastructure by combining evolutionary approaches and the theory of optimal set partitioning. For practical application, the proposed approach is implemented in a software solution consisting of three main modules: a user interface, modules for operating a Genetic Algorithm (GA) and modules for center placement.

The software integrates with existing information systems to enhance real-time information exchange among all participants in the medical logistics chain. The proposed model and software implementation offer a robust tool for improving the efficiency of medical resource distribution during crises.

Вступ. Необхідність вивчення питання медичної логістики в критичних ситуаціях полягає в вимогах адаптації до непередбачуваних криз, таких як пандемії та стихійні лиха. Вдосконалення транспортної мережі є центральним елементом для ефективного розподілу ліків та виробів медичного призначення під час надзвичайних ситуацій. Історичні приклади, такі як виклики під час пандемії COVID-19 та наслідків військових дій, підкреслюють логістичні недоліки у розподілі ліків.

Сучасна система медичної логістики в Україні має кілька слабких місць, які потребують уваги та вдосконалення. Перш за все,

існує значна затримка в ухваленні рішень через відсутність централізованої інформаційної системи для оперативного обміну даними між Міністерством охорони здоров'я (МОЗ) та регіональними департаментами охорони здоров'я (ДОЗ). Це призводить до бюрократичних затримок у затвердженні наказів та розподілі ресурсів, що є особливо критичним під час надзвичайних ситуацій. Існує недостатня кількість субрегіональних центрів та центрів дистрибуції, а також слабка їх підготовка до кризових ситуацій. Відсутність автоматизації процесів управління запасами та транспортуванням додатково ускладнює ефективне функціонування системи.

У статті [1] досліджується інтеграція технології інтелектуального виявлення вторгнень у систему медичної логістики з використанням публічної хмари. У роботі [2] розглядається методологія оптимізації розміщення об'єктів, що враховує переваги користувача. Використовується інтерактивний еволюційний алгоритм, який знаходить найбільш прийнятні рішення на Парето-фронті, спираючись на переваги користувача. Необхідність покращення системи медичної логістики в Україні обумовлена кількома ключовими факторами. У своїй роботі ми розглядаємо систему медичної логістики у кризових ситуаціях на предмет можливих покращень процесу. Пропонуємо подальшу та більш глибоку інтеграцію з наявними інформаційними системами для оперативного обміну інформацією між усіма учасниками медичної логістики. Також пропонуємо предметну постановку задачі та відповідну математичну модель, яка враховує потреби системи. Для вирішення цієї проблеми пропонуємо програмну реалізацію та описуємо її аспекти, що сприяють підвищенню ефективності розподілу медичних ресурсів під час кризових ситуацій. Ця робота є продовженням [3], де авторами було розглянуто математичну модель багатоетапної задачі розміщення об'єктів з обмеженням на максимальну кількість одночасно активованих підприємств і запропоновано підхід до її розв'язування, який базується на використанні евристичних підходів та теорії оптимального розбиття множин.

Математична модель.

Позначимо:

- Ω – область, де розподіляються медичні товари між споживачами;
- Ω_i – область обслуговування центра дистрибуції, $i = \overline{1, N}$;
- N – необхідна кількість центрів дистрибуції (ЦД);
- M – загальна кількість доступних для активації субрегіональних центрів (СРЦ);
- K – відома кількість регіональних центрів (РЦ);
- L – заданий ліміт на кількість можливо активованих СРЦ;
- b_i^I – попит i -го центру дистрибуції, $i = \overline{1, N}$;
- b_j^{II} – потужність j -го субрегіонального центру, $j = \overline{1, M}$;
- b_k^{III} – запаси k -го регіонального центра.
- I – множина відомих РЦ $I = \{\tau_1^{III}, \tau_2^{III}, \dots, \tau_K^{III}\}$.
- J – множина доступних для активації субрегіональних центрів: $J = \{\tau_1^{II}, \tau_2^{II}, \dots, \tau_M^{II}\}$.
- $c_i^I = c(x, \tau_i^I)$ – вартість транспортування медичних товарів від центра дистрибуції з координатами τ_i^I до споживача $x \in \Omega$, $i = \overline{1, N}$;
- $c_{ij} = c(\tau_i^r, \tau_j^p)$ – вартість транспортування медичних товарів від центра з координатами τ_j^r , r -го етапу до центра з координатами τ_i^p , p -го етапу, де $r, p \in \{I, II, III\}, r \neq p$;
- $\rho(x)$ – попит на ліки та ВМП в точці x в області Ω ;
- A_j – витрати на активацію j -го субрегіонального центру, $j = \overline{1, M}$;
- $\tau_i^r = (\tau_{i1}^r, \tau_{i2}^r)$ – координати центра дистрибуції ($r=I$); субрегіонального центру ($r=II$) або регіонального центра ($r=III$);
- v_{ij}^I – кількість медичних товарів, що транспортується від субрегіонального центра j до центра дистрибуції i , $i = \overline{1, N}, j = \overline{1, M}$.
- v_{kj}^{II} – кількість медичних товарів, що транспортується від регіонального центра k до субрегіонального j , $k = \overline{1, K}, j = \overline{1, M}$.
- $\theta_j = \begin{cases} 1, & \text{якщо СРЦ } j \text{ активований,} \\ 0, & \text{інакше} \end{cases}, j = \overline{1, M}$.

Тоді математична модель має вид:

$$\min_{\theta(\cdot) \in \Gamma_2, \tau^I \in \Omega^N, v \in R_{NM}^+} \sum_{j=1}^M A_j \theta_j + \sum_{i=1}^N \int_{\Omega_i} c_i^I(x, \tau_i^I) \rho(x) dx + \sum_{i=1}^N \sum_{j=1}^M c_{ij} v_{ij}^I + \sum_{k=1}^K \sum_{j=1}^M c_{kj} v_{kj}^{II}, \quad (1)$$

за таких обмежень

$$\sum_{j=1}^M v_{ij}^l \theta_j = \int_{\Omega_i} \rho(x) dx, \quad i = \overline{1, N}, \quad (2)$$

$$\sum_{i=1}^N v_{ij}^{ll} \leq b_j^{ll}, \quad j = \overline{1, M}, \quad (3)$$

$$\sum_{i=1}^N v_{ij}^l \leq b_j^{ll}, \quad j = \overline{1, M}, \quad (4)$$

$$\sum_{j=1}^M \theta_j \leq L, \quad (5)$$

$$\cup_{i=1}^N \Omega_i = \Omega, \quad (6)$$

$$\text{mes}(\Omega_i \cap \Omega_j) = 0, \quad i \neq j, \quad i, j = \overline{1, N}, \quad (7)$$

$$v_{ij}^l \geq 0, \quad \theta_j \in \{0; 1\}, \quad i = \overline{1, N}, \quad j = \overline{1, M}, \quad (8)$$

$$\tau^l = (\tau_1^l, \tau_2^l \dots \tau_N^l), \quad \tau^l \in \Omega^N. \quad (9)$$

Процес розв'язання задачі за допомогою генетичного алгоритму розпочинається з ініціалізації популяції, де кожне можливе рішення кодується на основі пріоритетів. Далі оцінюється придатність кожної хромосоми шляхом розв'язання задачі із застосуванням підходів з теорії оптимального розбиття множин та врахуванням плану перевезень на інших етапах. В результаті отримаємо розташування центрів, їх потужності та план перевезень. Відбір хромосом здійснюється за методом рулетки, після чого ці хромосоми піддаються кросоверу для отримання нового покоління. Деякі хромосоми мутуються для забезпечення варіативності. Нова популяція формується з нащадків і деяких особин з поточної популяції. Процес повторюється до досягнення умови зупинки. Після цього алгоритм завершується, декодується найефективніша хромосома, і повертається ефективний розв'язок.

Програмна реалізація. Перейдемо до опису програмної реалізації для запропонованого підходу. У розробці застосовувались різні технології та бібліотеки. Для інтерфейсу користувача, генетичного алгоритму використовувались: Python3, PyQt5 (для інтерфейсу користувача), Numpy (для реалізації генетичного алгоритму), Concurrent (пакет для підтримки багатопоточності), Multiprocessing (пакет для організації роботи між модулями реалізації). Для розв'язання неперервної задачі розміщення: C++; Eigen (для швидких математичних перетворень при реалізації чисельних методів недиференційованої оптимізації), spdlog (для логування), Qt6 (для візуалізації канвасу розміщення).

При розробці програмної реалізації, автором було додано функцію візуалізації карти, використовуючи сервіс OpenStreetMap. Ця функція дозволяє користувачам переглядати та взаємодіяти з

географічними даними, пов'язаними з центрами дистрибуції (ЦД) та іншими ресурсами на реальній карті. Користувачі можуть збільшувати та переміщувати карту, надаючи просторовий контекст для розподілу ресурсів та транспортних маршрутів. Відображення та управління картою здійснюється через відповідні компоненти інтерфейсу користувача.

Для візуалізації модельних задач використовується інший компонент, що надає область для графічного представлення результатів ГА. Існує можливість представляти розбиття, центри та плани перевезень у інтерактивній формі. Канвас динамічно оновлюється на основі даних моделі, дозволяючи користувачам взаємодіяти та аналізувати візуальні елементи. Також наявна функціональність, що дозволяє користувачам завантажувати завдання моделі з вхідних файлів, полегшуючи налаштування та ініціалізацію вхідних даних ГА та неперервних задач розміщення. Це дозволяє швидко почати роботу з попередньо визначеними конфігураціями або раніше збереженими станами. Завантаження та збереження файлів керується через відповідні контролери. Для забезпечення повторного аналізу та зберігання розв'язків передбачено підтримку їх серіалізації та десеріалізації. Користувачі можуть зберігати поточний розв'язок у файл та завантажувати його пізніше для подальшого аналізу. В інтерфейсі користувача передбачена можливість ітеративного переміщення по різних поколіннях рішення ГА. Можна переходити до наступного або попереднього покоління і переглядати зміни та оновлення розв'язку з часом. Важливою функцією є візуалізація оптимальних розбиттів для центрів дистрибуції. Це дозволяє користувачам бачити, як ресурси та зони обслуговування розподілені, що важливо для оптимізації стратегій розподілу. Спрощений варіант UML діаграми для представлення зв'язків між головними компонентами наведено на рис. 1.

Розглянемо діаграму послідовності дій та побудову взаємодії між модулями. Її подано на рис. 2.

Спочатку користувач завантажує файл з даними вхідної моделі. Модуль інтерфейсу користувача передає данні модулю ГА, де ініціалізуються параметри. При розв'язанні задачі, ГА модуль надсилає запити до модуля розміщення на відповідне отримання координат ЦД. Модуль розміщення на основі теорії оптимального розбиття розміщує ЦД та повертає відповідні локації. Під час розв'язання постійно оновлюється інтерфейс користувача з метою відображення змін. За запитом, користувач може серіалізувати або

десеріалізувати розв’язок. На виході отримуємо розв’язок та візуалізацію для користувача.

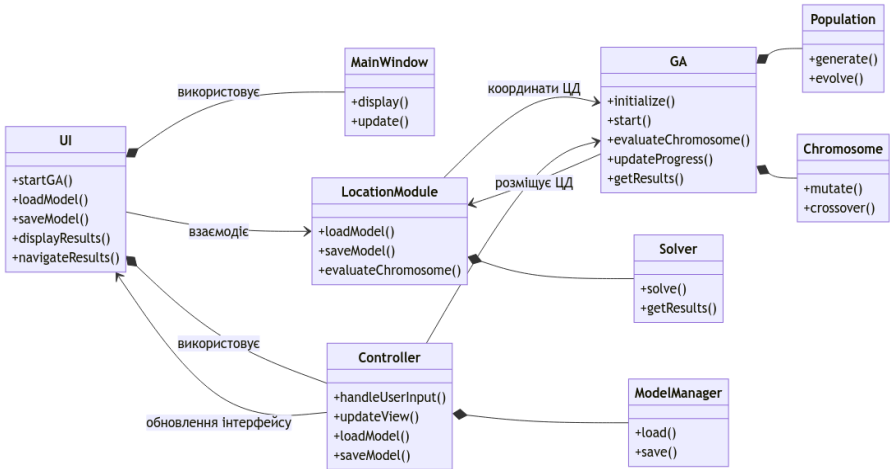


Рис.1 – Спрощена UML діаграма взаємодії між головними модулями

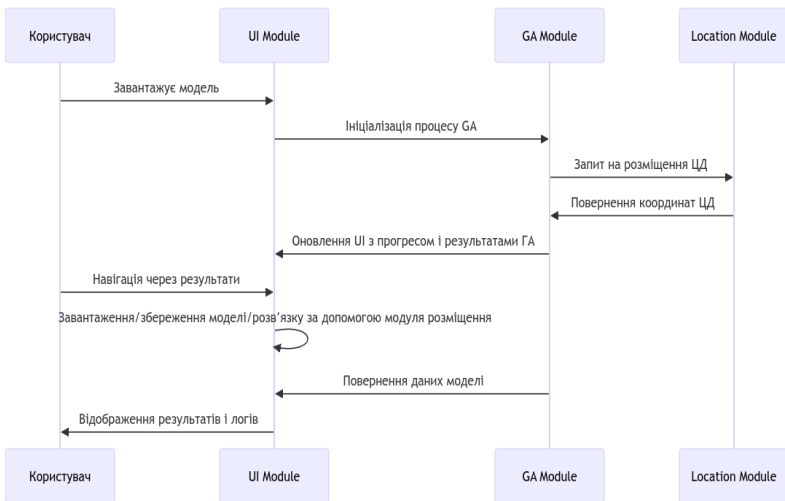


Рис.2 – Діаграма послідовності дій взаємодії користувача та модулів програмної реалізації

Висновки. В роботі розглянуто аспекти практичної імплементації рішення для двоетапної задачі розміщення-активації. Було розроблено програмну реалізацію, що складається з модулів, які забезпечують ефективне розв'язання задач оптимального розміщення центрів дистрибуції, а саме: інтерфейс користувача для взаємодії та візуалізації результатів, генетичний алгоритм для оптимізації, підтримку багатопоточності та міжмодульної взаємодії, чисельні методи недиференційованої оптимізації для розв'язання неперервних задач, логування процесу, візуалізацію мапи та модельних задач, а також функціональність серіалізації та десеріалізації для збереження та завантаження розв'язків. Імплементація дозволяє користувачам взаємодіяти з географічними даними, збільшуючи та переміщуючи мапу, аналізувати візуальні елементи розміщення та плани перевезень у інтерактивній формі. Користувачі можуть завантажувати вхідні моделі з файлів, що полегшує налаштування та ініціалізацію даних для генетичного алгоритму та неперервних задач. Імплементація також підтримує ітеративне переміщення по поколіннях генетичного алгоритму, що дозволяє користувачам переглядати зміни та оновлення у розв'язку з часом.

Робота є складовою частиною досліджень за держбюджетною науковою темою 0123U100011 «Задачі аналізу, моделювання та оптимізації технологічних процесів у складних системах різної природи», що виконуються в НТУ «Дніпровська політехніка».

Література

1. Barbati M., Corrente S., Greco S. Multiobjective Combinatorial Optimization with Interactive Evolutionary Algorithms: the case of facility location problems. *EURO Journal on Decision Processes*. 2024. P. 100047. URL: <https://doi.org/10.1016/j.ejdp.2024.100047>.
2. Lu W. Application cost of intelligent intrusion detection in medical logistics management under public cloud environment. *Computers and Electrical Engineering*. 2023. Vol. 112. P. 109014. URL: <https://doi.org/10.1016/j.compeleceng.2023.109014>.
3. Svitlana U., Oleksii S. An algorithm for solving a two-stage continuous-discrete location problem for medical logistics optimization. *System technologies*. 2024. Vol. 5, no. 148. P. 71–85. URL: <https://doi.org/10.34185/1562-9945-5-148-2023-07>.

IMPACT OF THE CHOICE OF THE TECHNOLOGY STACK ON THE INNOVATION AND COMPETITIVENESS OF COMPANIES: SAP CLOUD VS SAP ON-PREMISE

Oleksandr Smishko

National University «Zaporizhzhia Polytechnic»

Abstract. The article clarifies the positioning of the technological stack in the development of modern companies that simplify and accelerate the processes that take place in modern conditions. Processes and stages in which technologies play an important role are analyzed and systematized. The international experience of creating and developing IT technologies and their use has been described. The characteristics and features of using database management systems such as SAP Cloud and SAP On-Premise are highlighted. It was found that these systems facilitate access to information, contribute to the improvement and optimization of many business processes in companies. It has been proven that modern companies need technological innovations. The conclusions highlight the importance and global impact of the choice of technological stack on the competitiveness and innovation of companies, which directly affects competition, contributing to the blurring of market boundaries, the emergence of new industries and the emergence of business ecosystems.

In today's business environment, companies are constantly growing and competing with each other, and therefore need to monitor problems and innovate in data processing on an ongoing basis. In particular, an important aspect is that the identified problems should not only be monitored but also resolved.

SAP HANA is a database management system that allows you to create and deploy intelligent applications for processing large amounts of data [4].

SAP HANA, a virtually unlimited database for any workload, supports document repositories, geographic data, graphics, and models (from vectors to time series). The model offers high-quality data processing, high performance, and complete security, shifting developers' focus from administrative tasks to creating innovations.

Understanding and implementing SAP Cloud and SAP On-Premise contributes to the continuous improvement of innovation processes and the competitiveness of companies in the modern world [3]. These systems are innovative platforms that allow companies to quickly implement new functionality and continue to innovate. Projects involving the implementation of SAP Cloud and SAP On-Premise typically include business process development, utilizing SAP's best practices, and leveraging the enterprise management layer for SAP

S/4HANA. These projects enable organizations to rapidly deploy working systems based on ready-made business processes. Implementing these systems requires a consistent adoption of cloud thinking to maximise enterprise value. This approach ensures readiness for cloud operations, facilitates access to test environments, and provides isolated software environments based on the SAP Cloud Application Library. It is important to emphasise the benefits of SAP Cloud and SAP On-Premise:

- **Business Process Development.** This methodology is based on the use of practical systems built on the basis of ready-made business processes provided by SAP with two key resources provided by SAP, namely best practices (available to all customers) and the enterprise management layer for SAP S/4HANA. Both of these packages are the starting point for implementing SAP S/4HANA Cloud and SAP S/4HANA On-Premise, enabling organisations to get up and running quickly. This is done through a working system built on the basis of ready-made business processes. The project team employs a standards compliance approach to ensure adherence to standard business processes and to identify the delta requirements necessary to customize the solution for the organization's unique needs. Delta requirements refer to the specific changes or additional requirements needed to bridge the gap between the standard solution and the organization's specific demands. These typically fall into categories such as configuration, scalability, analysis, integration, data security, and access.
- **Readiness for cloud operations.** This principle applies not only to companies implementing solutions in the cloud, but also to project teams implementing SAP S/4HANA. In these cases, cloud technologies can be used to access test environments or to provide project teams with an isolated software environment based on the SAP Cloud Application Library. Organisations deploying solutions in the cloud can take advantage of cloud thinking in methodology and detailed information on how to request, access, activate, and use cloud solutions.

These systems were developed by SAP to create new and expanded existing applications in a secure cloud computing environment. Currently, there are one On-Premise and two cloud-based systems

deployed on the basis of cloud technologies. On-Premises systems are installed locally on the client's server. All work on such systems is performed by the company's employees. In particular, a private cloud actually looks like a user's own server somewhere in an external data centre. The user has full access to the system and can develop and modify the system code within the permitted limits.

Since 2023, only SAP Private Cloud Edition is available for new customers in Ukraine. SAP Cloud Public Edition is only available outside of Ukraine as a software model. The SAP On-Premise model (not a cloud solution) is not available for new customers in Ukraine and is only available for existing customers who are migrating from an older version of the SAP system to the new version of S/4 HANA.

It is quite a complex process to predict how the business will develop. It is also important to implement this prediction correctly. That is why planning is especially important for businesses and modern companies. In many of them, it is important to ensure that budgets, goals, and long-term strategies are properly met, and ideally, that the business is running. To support the planning process in spreadsheets with visual and functional SAP Analytics Cloud is responsible for a full-fledged planning component and setting up the process of monitoring its implementation [2].

SAP Process Control's special problem management functions provide an effective way to monitor and manage problems that go beyond regular control assessments. With this functionality, you can identify, track, and manage problems at various levels:

- organisational;
- administrative;
- regulatory.

When migrating to SAP S/4HANA, several critical decisions must be made. These decisions encompass selecting the type of system environment and defining the primary SAP S/4HANA functionalities to be utilized. The options for deployment include using SAP S/4HANA on self-owned hardware, hosting the software with a cloud service provider, or adopting a hybrid approach where part of the business processes are migrated to the cloud while retaining the remainder on-premises. It is also important to decide whether to use the migration process to redesign business processes or to maintain existing business processes.

Speaking of SAP On-Premise, this operating model involves the use of software purchased and its customers that runs on company-owned hardware. The software allows the company to have full control over its

hardware and software. This means that customers using local systems can decide whether to update their software or not. The product availability matrix allows them to see when SAP maintenance ends. It also provides maximum flexibility for customised developments and integration with other systems (internal solutions or external). Depending on the product, this may also include changes for which the company takes full responsibility. If the company decides to make changes, it is responsible for maintaining the functionality and planning the necessary adjustments in case of updates [3].

In addition to the cost of hardware and software, powerful and complex systems require additional costs for IT staff to implement, manage and maintain the software, which is important for companies to consider.

Businesses do not have the ability to manage and control the software beyond the user configuration settings. In addition to SAP S/4HANA Cloud, this category also includes the following SAP solutions: SAP SuccessFactors, SAP Cloud for Customer, SAP Ariba. The provider is responsible for managing and controlling the underlying cloud infrastructure, and the company manages applications built on the basis of the provided development environment. An example of this model is the SAP Business Technology Platform. In this model, companies can manage the planning and maintenance of their applications, but not the tools and services used by the applications.

The company manages the applications and operating systems used and often installs them as well. However, the provider is still responsible for monitoring and managing the cloud infrastructure. Examples include hyperscalers such as Amazon Web Services (AWS), Google Cloud Services (GCS), and Microsoft Azure [1]. There are three types of cloud deployment models:

1. Public clouds. In a public cloud, services and applications are publicly available and accessible to everyone. Users usually share the cloud infrastructure resources provided by the provider. The public version of SAP S/4HANA Cloud falls into this category.
2. Private cloud. In a private cloud, the cloud infrastructure is distributed between separate companies or only one company. Cloud infrastructure can be deployed internally or by an external provider. The same applies to private clouds. When migrating, you need to take into account the flexibility and scalability of the IT infrastructure, the way business applications are accessed via the Internet, and the regularity of automatic application updates.

The private version of SAP S/4HANA Cloud falls into this category.

3. Hybrid cloud. A combination of public and private clouds is called a hybrid cloud. In this case, part of the IT infrastructure operates locally, while other services are provided by an external public cloud provider [3].

Conclusions. Sustainable development and competitiveness of companies are more important than ever. All data operations and processes must be transparent. In order to manage sustainability for reporting, climate change, circular economy and social responsibility, it is necessary to use modern technology stacks, namely SAP Cloud or SAP On-Premise.

SAP solutions provide company-wide functionality and industry-specific functions that can be used to embed sustainability across the business by integrating operations, expertise and financial insights into core business processes.

Although the development vector of SAP products has been focused on cloud solutions for the last decade, SAP Business Suite remains the reliable basis of the information environment of many large and medium-sized enterprises. The technological platform SAP Netweaver, developed for many years, is stable and still meets the requirements of most companies risk being caught up in tomorrow's technological development, which is unacceptable in the era of information technologies and the fifth information revolution. For such companies, SAP offers "ready-to-use" system integration solutions. As a typical example, a company with an SAP ERP accounting system can implement a new SAP C/4HANA cloud system that can be accessed from anywhere with Internet access, is easy for users to understand and learn, and can be accessed from a mobile phone or tablet. SAP offers a solution package that integrates these two products. Business partners, trade names, orders, and other data are transmitted online by a two-way inter-accounting system ERP and CRM system C/4HANA. By leveraging this comprehensive solution, integration costs can be significantly reduced while providing clients with a thoroughly tested and efficient system, along with ongoing updates from SAP.

References:

1. Nepyivoda M., Rekalov O., Mychuda L., Korobeinikova T. Web applications technological stack. Modern Engineering and

- Innovative Technologies. 2022. №1 (22-01). С. 104-112. DOI: <https://doi.org/10.30890/2567-5273.2022-22-01-038>
2. Abassin S. SAP Analytics Cloud. Sap PRESS, 2024. 421 p. URL: https://www.sap-press.com/sap-analytics-cloud_5753/
 3. Densborn F., Finkbohner F., Hoft M., Rubarth B., Klob P., Mathab K. Migrating to SAP S/4HANA. Reinwerk Publishing, 2024. 633 p. URL: https://www.sap-press.com/migrating-to-sap-s4hana_5816/
 4. What is SAP HANA? SAP. URL: <https://www.sap.com/central-asia-caucasus/products/technology-platform/hana/what-is-sap-hana.html>

НЕСТАНДАРТНЕ ПОЧИНАННЯ СПРОЩЕННЯ БУЛЕВИХ ФУНКЦІЙ АНАЛІТИЧНИМ МЕТОДОМ

Михайло Соломко

*Національний університет водного господарства та
природокористування*

NON-STANDARD START OF SIMPLIFYING BOOLEAN FUNCTIONS USING THE ANALYTICAL METHOD

Abstract. The solved problem is the effectiveness of the method of simplifying Boolean functions to obtain optimal structures of logic circuits. A new theorem of a non-standard system of simplification of Boolean functions is formulated, according to which to obtain a minimal function it is enough to perform all non-redundant operations of simple and/or super-gluing of variables, which ultimately provides a minimal function in the main basis without using an implicant table. Thus, the problem of simplifying Boolean functions to the simplest normal equivalent is solved in one step. The interpretation of the result is that the properties of 2-(n, b)-design combinatorial systems make it possible to reproduce the definition of logical operations of super-gluing variables, to represent logical operations in a different way, and vice versa. This, in turn, ensures the establishment of the locations of equivalent transformations on the binary structure of the truth table and the implication of a systematic procedure for simplifying Boolean functions by an analytical method. The peculiarity of the obtained results is that unambiguous identification of the locations of equivalent transformations is possible even when different intervals of the Boolean space containing the 2-(n, b)-design systems have common blocks, intersections.

Гонка за фізичне зменшення розмірів транзисторів для цифрових пристроїв припинилась, проте продуктивність чипів зростає. Це, зокрема, зумовлено застосуванням 3-вимірної інтеграції, фотонних кристалів, у тому числі й результатами проектування на основі логічних функцій. Заходи зі спрощення логічних функцій для проектування цифрових схем будуть вживані завжди, оскільки мала потужність електроживлення, дешевизна та швидкість обчислення має значення, та є параметрами універсальними і актуальними, незалежно від динаміки промисловості, коли зменшуються проєктні норми чи впроваджуються виробничі програми, що реалізуються за межами закону Мура.

Логічні операції аналітичного методу для рівносильного перетворення булевих функцій не є ні систематичними, ні придатними для комп'ютерної реалізації. Крім зазначеного такий спосіб потребує більше часу для спрощення.

Можливі заходами подолання проблеми реалізації спрощення булевих функцій аналітичним методом полягає у застосуванні герменевтики [1], метаданих [2], встановленні локацій рівносильних перетворень [3], імплікуванні алгоритму спрощення булевих функцій. У підсумку вербальні процедури алгебричних перетворень замінюються на рівносильні образні перетворення.

Образна форма інформації, зокрема, у вигляді комбінаторних об'єктів переходить можливо випадково з тенденції причинності у необхідно-дійсну тенденцію послідовності нестандартної процедури, яка у підсумку має забезпечити більше шансів визначити алгоритм спрощення булевих функцій, коли вони представлені візуально-матричною формою. Комбінаторними об'єктами у цьому випадку будуть двовимірні повні 2 -(n , b)-design, та/або неповні 2 -(n , x/b)-design системи з повторенням і суть власне комбінаторні образи. Модель нестандартного спрощення булевих функцій, що потрібно представити, допустити особливість того, що є деяка аналогія алгоритму, який перетворює «безладну» складність процедури спрощення аналітичним методом у складний порядок образних перетворень.

У зв'язку з цим актуальним аспектом теоретичних наукових досліджень нестандартної системи спрощення булевих функцій залишаються дослідження направлені, зокрема на:

– отримання нового результату під час спрощення булевих функцій;

– вимірювання та оцінку результатів впровадження нестандартної системи спрощення в реальний синтез цифрових компонентів з подальшим їх застосуванням у цифрових технологіях;

Визначення логічних операцій супер-склеювання змінних комбінаторними системами 2-(n, b)-design

Враховуючи те, що повна ДДНФ функції:

– однозначно визначає повну комбінаторну систему з повторенням 2-(n, b)-design і навпаки [4];

– скорочення повної ДДНФ функції завжди дає одиницю, наприклад скорочення 3-розрядної повної ДДНФ булевої функції має вигляд:

$$\begin{aligned} & \overline{x_1 x_2 x_3} + \overline{x_1 x_2 x_3} + \overline{x_1 x_2 x_3} + \overline{x_1 x_2 x_3} + \overline{x_1 x_2 x_3} + \overline{x_1 x_2 x_3} + \overline{x_1 x_2 x_3} + \overline{x_1 x_2 x_3} = \\ & = \overline{x_1 x_2} (\overline{x_3} + \overline{x_3}) + \overline{x_1 x_2} (\overline{x_3} + \overline{x_3}) + \overline{x_1 x_2} (\overline{x_3} + \overline{x_3}) + \overline{x_1 x_2} (\overline{x_3} + \overline{x_3}) = \\ & = \overline{x_1 x_2} + \overline{x_1 x_2} + \overline{x_1 x_2} + \overline{x_1 x_2} = \overline{x_1} (\overline{x_2} + \overline{x_2}) + \overline{x_1} (\overline{x_2} + \overline{x_2}) = \overline{x_1} + \overline{x_1} = 1, \end{aligned}$$

це дає підставу подати логічну операцію супер-склеювання змінних повною комбінаторною системою з повторенням 2-(n, b)-design за таким вербальним концептом:

$$\begin{aligned} 2-(n,b)-design & \equiv \overline{x_1 x_2 \dots x_n} + \overline{x_1 x_2 \dots x_n} + \dots + \overline{x_1 x_2 x_3} = \\ & = \overline{x_1 x_2 \dots} (\overline{x_n} + \overline{x_n}) + \overline{x_1 x_2 \dots} (\overline{x_n} + \overline{x_n}) + \dots + \overline{x_1 x_2 \dots} (\overline{x_n} + \overline{x_n}) = \\ & = \overline{x_1 \dots x_{n-1}} + \dots + \overline{x_1 x_{n-1}} = \overline{x_1 \dots} (\overline{x_{n-1}} + \overline{x_{n-1}}) + \dots = 1. \end{aligned}$$

Таким чином система 2-(n, b)-design представляє логічну операцію супер-склеювання змінних ($n > 1, b > 3$).

Для $n=1, b=2$ комбінаторна система 2-(n, b)-design подає логічну операцію простого склеювання змінних:

$$2-(1,2)-design \equiv \overline{x_1} + x_1 = 1.$$

Тоді візуально-матрична форма для операції супер-склеювання змінних може мати такий, наприклад, вигляд:

– перше правило супер-склеювання для 4-розрядної булевої функції:

x_1	x_2	x_3	x_4
0	0	0	x_4
0	0	1	x_4
0	1	0	x_4
0	1	1	x_4
1	0	0	x_4
1	0	1	x_4
1	1	0	x_4
1	1	1	x_4

$$= x_4; \quad (1)$$

Літерали, що повторюються у наборах змінних ДДНФ функції (1) являють собою прості імпліканти для скороченої форми булевої функції. Отже x_4 проста імпліканта у скороченій формі булевої функції (1).

Загальне правило склеювання на інтервалах булевого простору, що утримують комбінаторні системи 2-(n, b)-design можна сформулювати у такий спосіб: склеюванню підлягають набори змінних у кількості, що є степенем числа 2. Новий отриманий елементарний добуток (проста імпліканта) визначається як добуток змінних, які не змінюють свого значення на всіх наборах, що склеюються. Число m змінних, що залишається в елементарному добутку, визначається за формулою:

$$m = n - \log_2 M,$$

де n — число змінних функції; M — число наборів, що склеюється. Отже, для функції (1) число m становить $m = n - \log_2 M = 4 - \log_2 8 = 1$.

Трійковий вектор візуально-матричної форми (1), що представляє інтервал булевого простору $I(0001, 1111)$ має вигляд:

$$(- - - 1).$$

Десятковий відповідник аналітичного запису першого правила (1) має такий вигляд:

$$x_4 \Rightarrow (---1) \Rightarrow \begin{array}{c} \left| \begin{array}{cccc} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{array} \right| \Rightarrow (1,3,5,7,9,11,13,15). \end{array}$$

– друге правило супер-склеювання для 4-розрядної булевої функції:

x_1	x_2	x_3	x_4
0	0	x_3	x_4
0	1	x_3	x_4
1	0	x_3	x_4
1	1	x_3	x_4

$$= x_3 x_4; \quad (2)$$

Число m для функції (2) становить $m = n - \log_2 M = 4 - \log_2 4 = 2$.

Трійковий вектор візуально-матричної форми (2), що представляє інтервал булевого простору $I(0011, 1111)$ має вигляд:

$$(- - 11).$$

Десятковий відповідник аналітичного запису другого правила (2) має такий вигляд:

$$x_3 x_4 \Rightarrow (- - 11) \Rightarrow \begin{array}{c} \left| \begin{array}{cccc} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{array} \right| \Rightarrow (3,7,11,15). \end{array}$$

– третє правило супер-склеювання для 4-розрядної булевої функції:

x_1	x_2	x_3	x_4
0	x_2	x_3	x_4
1	x_2	x_3	x_4

$$= x_2 x_3 x_4. \quad (3)$$

Число m для функції (3) становить $m = n - \log_2 M = 4 - \log_2 2 = 3$.

Трійковий вектор візуально-матричної форми (3), що представляє інтервал булевого простору $I(0111, 1111)$ має вигляд:

$$(-111).$$

Десятковий відповідник аналітичного запису третього правила (3) має такий вигляд:

$$x_2 x_3 x_4 \Rightarrow (-111) \Rightarrow$$

$$\Rightarrow \left| \begin{array}{cccc} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{array} \right| \Rightarrow (7,15).$$

Перше правило супер-склеювання змінних використовує систему 2-(3, 8)-design. Друге правило використовує систему 2-(2, 4)-design. Третє правило використовує систему 2-(1, 2)-design. Правило (3) проявляє себе як просте склеювання змінних та є частковим випадком правил (1) і (2).

Змінні, що утворюють повну комбінаторну систему з повторенням 2-(n , b)-design, можуть займати будь-які розряди логічної функції.

Десяткові відповідники аналітичних записів (1)–(3) безпосередньо представляють потрібні інтервали булевого простору для проведення рівносильних перетворень при спрощенні булевих функцій.

Теорема нестандартної системи спрощення булевих функцій

Спрощення булевих функцій здійснюється рівносильними перетвореннями, зокрема, за допомогою наступних логічних операцій:

– просте склеювання змінних:

$$xy + \bar{x}y = y; \quad (4)$$

$$(x + y)(\bar{x} + \bar{y}) = y.$$

– супер-склеювання змінних:

$$xyz + \bar{x}yz + x\bar{y}z + x\bar{y}\bar{z} = (x + \bar{x})yz + (x + \bar{x})\bar{y}z = yz + \bar{y}z = (y + \bar{y})z = z; \quad (5)$$

$$(x + y + z)(\bar{x} + \bar{y} + \bar{z})(x + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z) = (y + z)(\bar{y} + z) = z.$$

Теорема. (без доведення) Якщо у досконалій диз'юнктивній нормальній формі булевої функції провести всі операції простого та/або супер-склеювання змінних (4), (5), то у підсумку буде отримана мінімальна ДНФ булевої функції.

Примітка. Операції склеювання змінних (4), (5) можуть бути надлишковими. Це означає присутність у бінарній структурі таблиці істинності інших мінімальних ДНФ заданої функції. Тому для отримання мінімальної функції потрібно провести всі ненадлишкові операції простого та/або супер-склеювання змінних. Це забезпечить отримання мінімальної функції в основному базисі без застосування імплікантної таблиці.

Скорочену ДНФ визначають за допомогою диз'юнктивного об'єднання кон'юнкцій за умови, що всі набори змінних таблиці істинності булевої функції, що вміщують системи 2-(n, b)-design, повністю покривають ту її частину наборів, на яких булева функція повертає одиницю. При цьому допускається перетин наборів змінних, що утримують системи 2-(n, b)-design.

Приклад. Нестандартною системою отримати мінімальну булеву функцію чотирьох змінних $f(x_1, x_2, x_3, x_4)$, що задана досконалою диз'юнктивною нормальною формою [5]:

$$f(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1\bar{x}_2x_3\bar{x}_4 + \bar{x}_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3\bar{x}_4 + \bar{x}_1x_2x_3x_4 + x_1\bar{x}_2\bar{x}_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_3x_4 + x_1\bar{x}_2x_3\bar{x}_4 + x_1\bar{x}_2x_3x_4 + x_1x_2\bar{x}_3\bar{x}_4 + x_1x_2\bar{x}_3x_4 + x_1x_2x_3\bar{x}_4 + x_1x_2x_3x_4. \quad (6)$$

Розв'язок.

$$f_{\text{МДНФ}}(x_1, x_2, x_3, x_4) =$$

№	x_1	x_2	x_3	x_4
1	0	0	0	1
2	0	0	1	0
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
8	1	0	0	0
9	1	0	0	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

$$=$$

x_1	x_2	x_3	x_4
		0	1
0		1	0
	1		0
1		0	
1	1		

$$=$$

$$= x_1x_2 + x_1x_3 + x_2x_4 + x_1x_3x_4 + x_3x_4.$$
(7)

У першій матриці (7), що представляє ДДНФ $f(x_1, x_2, x_3, x_4)$ проведено наступні дії:

– до блоків 1, 5, 9, 13; 4, 6, 12, 14; 8, 9, 12, 13; 12, 13, 14, 15, кожен з яких утворює відповідний інтервал булевого простору, що вміщує повну комбінаторну систему 2–(2, 4)-design, застосовано операцію супер-склеювання змінних [4].

– до блоків 2, 6, які утворюють відповідний інтервал булевого простору, що вміщує комбінаторну систему 2–(1, 2)-design та містять загальний літерал « x_2 » застосовано операцію простого склеювання змінних [4].

Результати ненадлишкових логічних операцій склеювання змінних записуються до другої матриці виразу (7).

Десятковий відповідник аналітичного запису мінімальної функції (7) має вигляд:

$$f(x_1, x_2, x_3, x_4) = \{(1,5,9,13), (4,6,12,14), (8,9,12,13), (12,13,14,15), (2,6)\}.$$

Шостий блок змінних «0110» у першій матриці виразу (7) є загальним для двох систем – $\Sigma_m(4,6,12,14)$, $\Sigma_m(2,6)$ та однієї операції простого склеювання змінних, локалізація якої визначається відповідними інтервалами булевого простору, що

вміщують комбінаторні системами 2-(2, 4)-design та 2-(1, 2)-design [3].

Дванадцятий блок змінних «1100» у першій матриці виразу (7) є загальним для трьох систем – $\Sigma_m(4,6,12,14)$, $\Sigma_m(8,9,12,13)$, $\Sigma_m(12,13,14,15)$ та однієї операції супер-склеювання змінних, локалізація якої визначається відповідними інтервалами булевого простору, що вміщують комбінаторні системами 2-(2, 4)-design [3].

Тринадцятий блок змінних «1101» у першій матриці виразу (7) є загальним для трьох систем – $\Sigma_m(1,5,9,13)$, $\Sigma_m(8,9,12,13)$, $\Sigma_m(12,13,14,15)$ та однієї операції супер-склеювання змінних, локалізація якої визначається відповідними інтервалами булевого простору, що вміщують комбінаторні системами 2-(2, 4)-design.

Чотирнадцятий блок змінних «1110» у першій матриці виразу (7) є загальним для двох систем – $\Sigma_m(4,6,12,14)$, $\Sigma_m(12,13,14,15)$ та однієї операції супер-склеювання змінних, локалізація якої визначається відповідними інтервалами булевого простору, що вміщують комбінаторні системи 2-(2, 4)-design.

У підсумку за один крок, без застосування імплікантної таблиці, отримано мінімальну функцію (7), яка на два літерали менша, порівняно зі скороченою функцією [5] ($f^2 = \overline{x_1 x_3 x_4} + \overline{x_3 x_4} + \overline{x_2 x_3} + \overline{x_2 x_4} + x_1 x_2 + x_1 x_3$).

Література

1. Різник, В.В., Соломко М.Т. Мінімізація кон'юнктивних нормальних форм булевих функцій комбінаторним методом // Технологічний аудит та резерви виробництва. 2018 – № 5/2 (43). – С. 42 – 55. URL: <http://journals.urau.com/tarp/article/view/146312> (дата звернення: 15. 12. 2024).

2. Соломко, М. Т., Хомюк, Н. Л., Івашук, Я. Г., Назарук, В. Д., Рейнська, В. Б., Зубик, Л. В., Попова А. М. Впровадження методу образних перетворень для мінімізації функції Шеффера // Східно-Європейський журнал передових технологій. 2020 – № 5/4 (107). – С. 19 – 34. URL: <http://surl.li/sjiuh> (дата звернення: 20.04.2024)

3. Mykhaylo Solomko, Mykola Antoniuk, Ihor Voitovych, Yuliia Uliianovska, Nataliia Pavlova, Viacheslav Biletskyi. Implementation of the method of figurative transformations to minimizing partially defined boolean functions // Eastern-European Journal of Enterprise Technologies. –Vol. 1, No 4(121). 2023. – pp 6 – 25. DOI: 10.15587/1729-4061.2023.273293. URL: <http://surl.li/sjgyc> (дата звернення: 20. 04. 2024).

4. Різник, В.В., Соломко, М.Т. Застосування алгебричної операції супер-склеювання змінних для мінімізації булевих функцій комбінаторни методом // Технологічний аудит та резерви виробництва. – Вип. 6/2 (38), 2017. С. 60 – 76. URL: <http://surl.li/sjggtt> (дата звернення: 20. 04. 2024).

5. Закревский, А.Д., Потгосин, Ю.В., Черемисинова, Л.Д. Логические основы проектирования дискретных устройств / А.Д. Закревский, Ю.В. Потгосин, Л.Д. Черемисинова. – М.: ФИЗМАТЛИТ, 2007. – 592 с. – ISBN 978-5-9221-0811-9. URL: <http://surl.li/siddo>

ВИКОРИСТАННЯ СОЛВЕРА BARON ДЛЯ РОЗВ'ЯЗАННЯ КВАДРАТИЧНОЇ ЗАДАЧІ ОПТИМАЛЬНОЇ УПАКОВКИ НЕРІВНИХ КРУГІВ²

¹ Стецюк П.І., ²Романова Т.Є., ³Тиводар С.Р.

¹ Інститут кібернетики імені В.М. Глушкова НАН України

² Інститут проблем машинобудування імені А.М. Підгорного
НАН України

³ Ужгородський національний університет

USING THE BARON SOLVER FOR SOLVING THE QUADRATIC PROBLEM OF OPTIMAL PACKING OF UNEQUAL CIRCLES

Abstract. The paper considers a quadratic nonconvex extremal problem of packing a small number of unequal circles into a minimum-radius outer circle. The influence of additional linear constraints to speed up the BARON solver time for solving the packing problem is studied. Computational experiments are performed for six benchmark instances that are freely available on the website <https://www.packomania.com/>. The number of unequal circles is varied from

² Робота підтримана Volkswagen Foundation (грант № 97775)

five to ten. The BARON solver couldn't find the global solutions for the set of instances for the time-limit of 7200 seconds. Two propositions, aimed at reducing the number of possible solutions of the quadratic problem by means of extending the quadratic model with two corresponding families of additional linear constraints, are formulated. The computational results are provided. The BARON solver found proved global solutions for the first four instances within 0.37, 0.61, 26.41, 309.91 seconds, and significantly improved the lower bounds of the outer circle radius for the fifth and sixth instances.

Нехай задано сімейство m кругів C_i з радіусами r_i , $i = 1, \dots, m$ та зовнішній круг C_0 з центром в точці $(0, 0)$ та змінним радіусом R . Будемо вважати, що всі радіуси кругів є різними. Оптимальною упаковкою сімейства кругів в зовнішній круг називають таку їх упаковку, для якої R^* – радіус зовнішнього круга є мінімальним за умов, що круги C_i , $i = 1, \dots, m$ повністю знаходяться всередині зовнішнього круга та попарно не перетинаються між собою.

Якщо позначити (x_i, y_i) – невідомий центр круга C_i , $i = 1, \dots, m$, то задачі оптимальної упаковки сімейства кругів відповідає багатоекстремальна квадратична задача [1], яка має такий вигляд:

$$R^* = \min_{R, x, y} R \quad (1)$$

за обмежень

$$x_i^2 + y_i^2 \leq (R - r_i)^2, \quad i = 1, \dots, m, \quad (2)$$

$$(x_i - x_j)^2 + (y_i - y_j)^2 \geq (r_i + r_j)^2, \quad 1 \leq i < j \leq m, \quad (3)$$

$$R \geq \max_{i=1, \dots, m} r_i, \quad (4)$$

де $x = (x_1, \dots, x_m)$, $y = (y_1, \dots, y_m)$.

Цільова функція (1) є лінійною функцією та пов'язана з мінімізацією радіуса зовнішнього круга C_0 . Квадратичні обмеження (2) означають, що кожний круг C_i , $i = 1, \dots, m$, належить зовнішньому кругу C_0 . Квадратичні обмеження (3) гарантують, що ніякі два круги із сімейства кругів C_i , $i = 1, \dots, m$ не перетинаються. Лінійне обмеження (4) забезпечує додатне значення R – радіуса зовнішнього круга необхідне для коректності квадратичних обмежень (2).

Задача (1) – (4) є багатоекстремальною задачею нелінійного (квадратичного) програмування і для знаходження її розв'язку можна використовувати солвери, які призначені для розв'язання

задач глобальної оптимізації. На NEOS-сервері [2] представлено ряд таких солверів, серед яких є відомий солвер BARON [3,4].

В таблиці 1 представлено результати розв'язання задачі (1) – (4) за допомогою солвера BARON для шести тестових прикладів, які є у вільному доступі на веб-сайті <https://www.packomania.com/>, де $m \in \{5, 6, 7, 8, 9, 10\}$ та радіуси кругів C_i , $i = 1, \dots, m$ є рівними відповідним цілим додатним числам, тобто $r_i = i$. Відводився граничний час в 7200 секунд.

Таблиця 1

BARON – отримані розв'язки задачі (1) – (4) для шести тестових прикладів

m	N_{itm}	N_{nodes}	N_{nodes}^*	R_{low}	R_{up}	R^*
5	6796413	1535	849	8.55369	9.00140	9.00140
6	5819686	2502	2502	10.5063	11.0570	11.0570
7	2842333	18880	147157	12.7890	13.4621	13.4621
8	825944	88276	406705	13.8261	16.2217	16.2217
9	490757	136297	312999	12.0193	19.3517	19.2332
10	354724	103893	144784	14.3897	22.1176	22.0002

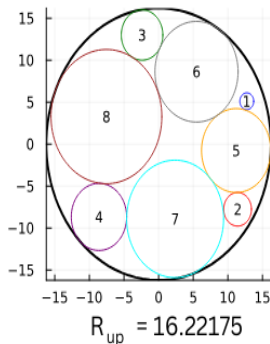
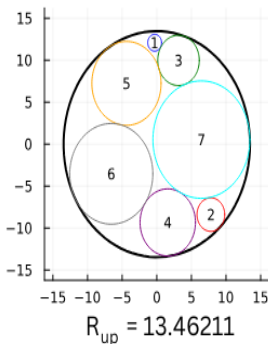
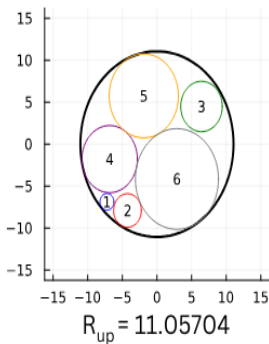
В таблиці використовуються такі позначення: N_{itm} – кількість ітерацій солвера BARON, N_{nodes} – максимальна кількість задіяних вершин, N_{nodes}^* – номер вершини, на якій знайдено найкраще значення радіусу зовнішнього круга C_0 , R_{low} – нижня границя на радіус, R_{up} – верхня границя на радіус, R^* – найменший радіус круга C_0 , взятий з сайту <https://www.packomania.com/>.

З таблиці 1 видно, що BARON не знайшов глобальний розв'язок ні для одного тестового прикладу за 7200 секунд. Для перших чотирьох тестових прикладів він знайшов R_{up} – верхню границю радіуса, яка співпадає з мінімальним радіусом круга C_0 . Найбільший розрив між верхньою та нижньою границями складає 15% для четвертого тестового прикладу, а найменший – 5% для перших трьох тестових прикладів. Для п'ятого та шостого прикладу верхня границя радіуса зовнішнього круга є більшою за його

мінімальне значення R^* . Розрив між границями складає 38% для п'ятого та 35% для шостого тестових прикладів.

Для тестових прикладів $m \in \{6, 7, 8\}$ на рисунку 1 наведені розв'язки, знайдені солвером BARON, та розв'язки з сайту <https://www.packomania.com/>.

BARON



Packomania

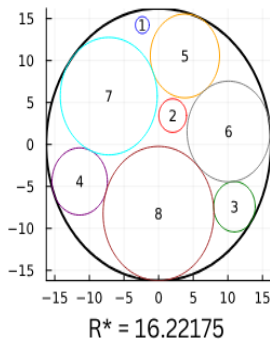
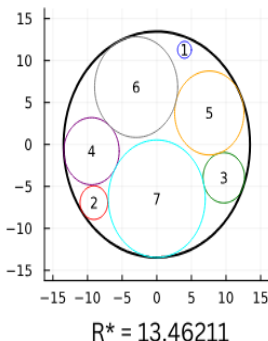
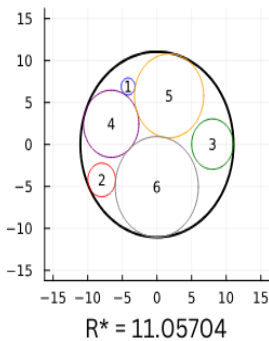


Рис. 1 – BARON та Packomania: розв'язки задачі (1) –(4) для $m \in \{6, 7, 8\}$

Із рисунка 1 видно, що розв'язки задачі (1)-(4) для прикладів $m \in \{6, 7, 8\}$, знайдені солвером BARON та взяті з сайту «Packomania», сильно відрізняються розміщенням центрів кругів C_i , $i = 1, \dots, m$. Це пояснюється наступним фактом, який сформулюємо таким твердженням.

Твердження 1. Нехай $x^* = \{x_i^*\}_{i=1}^m$, $y^* = \{y_i^*\}_{i=1}^m$, R^* – розв’язок задачі (1) – (4), а кут φ – такий, що $0 \leq \varphi \leq 2\pi$. Тоді $x^{**} = \{x_i^{**}\}_{i=1}^m$, $y^{**} = \{y_i^{**}\}_{i=1}^m$, $R^{**} = R^*$ буде розв’язком задачі (1) – (4), якщо центри кругів C_i , $i = 1, \dots, m$ обчислюються за формулами $x_i^{**} = \cos \varphi \cdot x_i^* + \sin \varphi \cdot y_i^*$, $y_i^{**} = -\sin \varphi \cdot x_i^* + \cos \varphi \cdot y_i^*$, $i = 1, \dots, m$.

Доведення. Покажемо що, якщо квадратичні нерівності (2) та (3) виконуються для $x^* = \{x_i^*\}_{i=1}^m$, $y^* = \{y_i^*\}_{i=1}^m$, R^* , то вони будуть виконуватися і для $x^{**} = \{x_i^{**}\}_{i=1}^m$, $y^{**} = \{y_i^{**}\}_{i=1}^m$, $R^{**} = R^*$.

Для нерівностей (2) це випливає з такого ланцюга співвідношень:

$$\begin{aligned} (R^{**} - r_i)^2 &\geq (x_i^{**})^2 + (y_i^{**})^2 = (\cos \varphi \cdot x_i^* + \sin \varphi \cdot y_i^*)^2 + (-\sin \varphi \cdot x_i^* + \cos \varphi \cdot y_i^*)^2 = \\ &= \cos^2 \varphi \cdot (x_i^*)^2 + 2 \cos \varphi \sin \varphi \cdot x_i^* \cdot y_i^* + \sin^2 \varphi \cdot (y_i^*)^2 + \sin^2 \varphi \cdot (x_i^*)^2 - 2 \cos \varphi \sin \varphi \cdot x_i^* \cdot y_i^* + \\ &+ \cos^2 \varphi \cdot (y_i^*)^2 = \cos^2 \varphi \cdot (x_i^*)^2 + \sin^2 \varphi \cdot (y_i^*)^2 + \sin^2 \varphi \cdot (x_i^*)^2 + \cos^2 \varphi \cdot (y_i^*)^2 = \\ &= (\cos^2 \varphi + \sin^2 \varphi) \cdot (x_i^*)^2 + (\sin^2 \varphi + \cos^2 \varphi) \cdot (y_i^*)^2 = (x_i^*)^2 + (y_i^*)^2 \leq (R^* - r_i)^2, \quad i = 1, \dots, m. \end{aligned}$$

Для нерівностей (3) це випливає з такого ланцюга співвідношень:

$$\begin{aligned} (r_i + r_j)^2 &\leq (x_i^{**} - x_j^{**})^2 + (y_i^{**} - y_j^{**})^2 = (\cos \varphi \cdot x_i^* + \sin \varphi \cdot y_i^* - \cos \varphi \cdot x_j^* - \sin \varphi \cdot y_j^*)^2 + \\ &+ (-\sin \varphi \cdot x_i^* + \cos \varphi \cdot y_i^* + \sin \varphi \cdot x_j^* - \cos \varphi \cdot y_j^*)^2 = (\cos \varphi \cdot (x_i^* - x_j^*) + \sin \varphi \cdot (y_i^* - y_j^*))^2 + \\ &+ (-\sin \varphi \cdot (x_i^* - x_j^*) + \cos \varphi \cdot (y_i^* - y_j^*))^2 = \cos^2 \varphi \cdot (x_i^* - x_j^*)^2 + 2 \cos \varphi \cdot \sin \varphi \cdot (x_i^* - x_j^*) \times \\ &\times (y_i^* - y_j^*) + \sin^2 \varphi \cdot (y_i^* - y_j^*)^2 + \sin^2 \varphi \cdot (x_i^* - x_j^*)^2 - 2 \sin \varphi \cdot \cos \varphi \cdot (x_i^* - x_j^*) \cdot (y_i^* - y_j^*) + \\ &+ \sin^2 \varphi \cdot (y_i^* - y_j^*)^2 = (\cos^2 \varphi + \sin^2 \varphi) \cdot (x_i^* - x_j^*)^2 + (\sin^2 \varphi + \cos^2 \varphi) \cdot (y_i^* - y_j^*)^2 = \\ &= (x_i^* - x_j^*)^2 + (y_i^* - y_j^*)^2 \geq (r_i + r_j)^2, \quad 1 \leq i < j \leq m. \end{aligned}$$

Твердження 1 доведено. \square

Для того, щоб зменшити кількість розв’язків, які для задачі (1)–(4) мають місце згідно твердження 1, достатньо для довільного із кругів C_i , $i = 1, \dots, m$ зафіксувати одну з компонент його центра так, щоб круг не виходив за межі зовнішнього круга C_0 . Для цього оберемо найбільший круг C_i з індексом $i = m$ та зафіксуємо його x -ову компоненту рівною \bar{x} за допомогою лінійної рівності

$$x_m = \bar{x}, \text{ де } -R^* + r_m \leq \bar{x} \leq R^* - r_m. \quad (5)$$

Якщо до задачі (1) – (4) додати обмеження (5), то в новій задачі розв’язків стане менше, що дозволить прискорити роботу солвера BARON. Про це свідчать результати розрахунків для тестових прикладів, які наведені в таблиці 2.

Таблиця 2

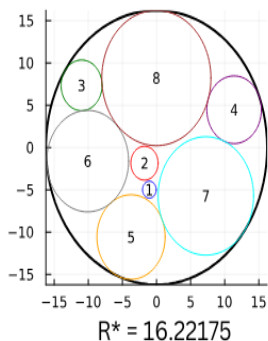
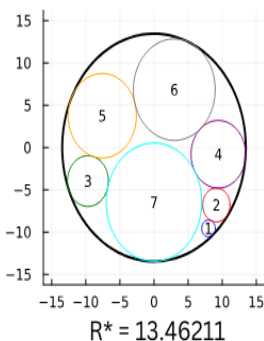
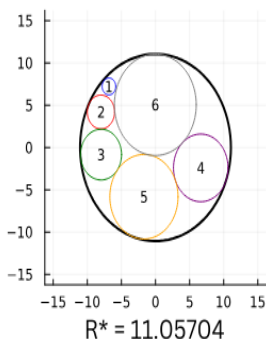
BARON – отримані розв’язки задачі (1) – (5) для шести тестових прикладів

m	x_m	N_{in}	N_{nodes}	N_{nodes}^*	$time$	R_{low}	R_{up}
5	-0.01535	33	5	-1	0.78	9.00139	9.00140
6	-0.02365	333	15	-1	2.38	11.0570	11.0570
7	0.00055	12589	292	11223	48.87	13.4621	13.4621
8	-0.02348	101679	2079	15446	638.57	16.2217	16.2217
9	-0.02009	757457	53254	113331	7200	16.8822	19.2332
10	-0.02110	440500	79418	68323	7200	14.5351	22.0002

В таблиці використовуються два нових позначення: x_m – значення зафіксованих x -ових координат центрів відповідних найбільших кругів C_i , $i = m$, які взяті з сайту «Раскоmania», $time$ – час розв’язання тестових задач (1) – (5). Всі інші позначення такі, як в таблиці 1.

З таблиці 2 видно, що солвер BARON успішно розв’язав перших чотири тестових прикладів, затративши невеликий час – менше секунди для першого прикладу, трохи більше двох секунд – для другого, близько 49 секунд – для третього та трохи більше десяти хвилин – для четвертого прикладу. П’ятий та шостий тестові приклади він не розв’язав, але для них покращив розрив між верхньою та нижньою границями на радіус зовнішнього круга C_0 – для $m=9$ розрив становить 12%, а для $m=10$ – 34%. Зауважимо, що значення верхніх границь на радіус зовнішнього круга C_0 в таблиці 2 (колонка R_{up}) співпадає зі значеннями відповідного мінімального радіусу зовнішнього круга в таблиці 1 (колонка R^*).

BARON



Ракomania

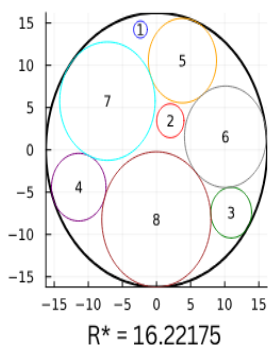
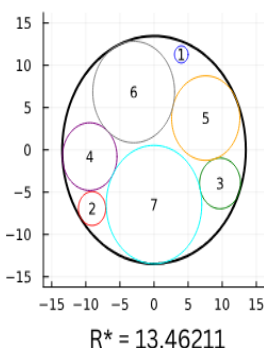
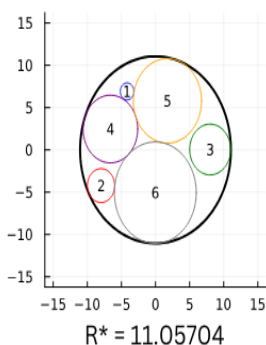


Рис. 2 – *BARON* та *Ракomania*: розв'язки задачі (1) – (5) для $m \in \{6, 7, 8\}$

На рисунку 2 наведені розв'язки задачі (1) – (5) для тестових прикладів $m \in \{6, 7, 8\}$, які знайдені солвером *BARON*, та розв'язки з сайту <https://www.rakomania.com/>. Із рисунка 2 видно, що розв'язки тестових прикладів не так сильно відрізняються за розміщенням центрів кругів C_i , $i = 1, \dots, m$, як це було на рисунку 1 з розв'язками тестових задач (1) – (4). Для тестових прикладів $m = 6$ та $m = 8$ найбільші круги розміщені у верхній частині зовнішнього круга C_0 . Для того щоб розмістити їх в нижній частині зовнішнього круга C_0 можна використати таке твердження.

Твердження 2. Нехай $x^* = \{x_i^*\}_{i=1}^m$, $y^* = \{y_i^*\}_{i=1}^m$, R^* – розв'язок задачі (1) – (4). Тоді $x^{***} = \{x_i^{***}\}_{i=1}^m$, $y^{***} = \{y_i^{***}\}_{i=1}^m$, $R^{***} = R^*$ – розв'язок задачі (1) – (4), якщо центри кругів C_i , $i = 1, \dots, m$ обчислюються за

формулами $x_i^{***} = x_i^*$, $y_i^{***} = -y_i^*$, $i = 1, \dots, m$.

Доведення. Аналогічно, як це було при доведенні твердження 1, покажемо що, якщо нерівності (2) та (3) виконуються для $x^* = \{x_i^*\}_{i=1}^m$, $y^* = \{y_i^*\}_{i=1}^m$, R^* , то вони будуть виконуватися і для $x^{***} = \{x_i^{***}\}_{i=1}^m$, $y^{***} = \{y_i^{***}\}_{i=1}^m$, $R^{***} = R^*$.

Для нерівностей (2) це впливає з такого ланцюга співвідношень

$$\begin{aligned} (R^{***} - r_i)^2 &\geq (x_i^{***})^2 + (y_i^{***})^2 = (x_i^*)^2 + (-y_i^*)^2 = \\ &= (x_i^*)^2 + (y_i^*)^2 \leq (R^* - r_i)^2, \quad i = 1, \dots, m. \end{aligned}$$

а для нерівностей (3) – з ланцюга співвідношень

$$\begin{aligned} (r_i + r_j)^2 &\leq (x_i^{***} - x_j^{***})^2 + (y_i^{***} - y_j^{***})^2 = (x_i^* - x_j^*)^2 + (-y_i^* + y_j^*)^2 = \\ &= (x_i^* - x_j^*)^2 + (y_i^* - y_j^*)^2 \geq (r_i + r_j)^2, \quad 1 \leq i < j \leq m. \end{aligned}$$

Твердження 2 доведено. □

Якщо до задачі (1) – (5) додати обмеження (лінійна нерівність)

$$y_m \leq 0, \tag{6}$$

то це дозволить прискорити роботу солвера BARON. Про це свідчать результати розрахунків для тестових прикладів, які наведені в таблиці 3.

Таблиця 3

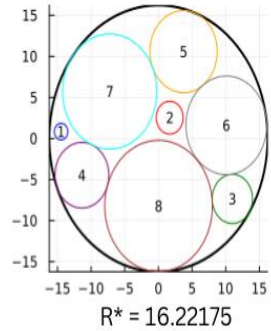
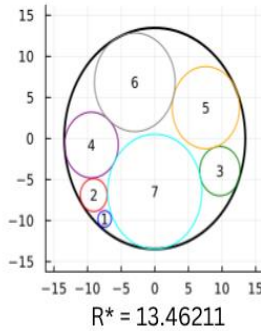
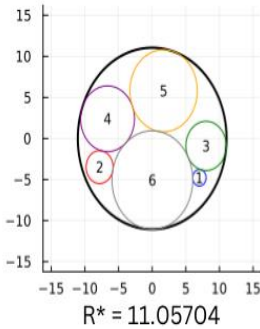
BARON – отримані розв'язки задачі (1) – (6) для шести тестових прикладів

m	x_m	N_{itm}	N_{nodes}	N_{nodes}^*	$time$	R_{low}	R_{up}
5	-0.01535	13	3	-1	0.37	9.00139	9.00140
6	-0.02365	21	5	5	0.61	11.0570	11.0570
7	0.00055	6000	150	159	26.41	13.4621	13.4621
8	-0.02348	48299	903	4317	309.91	16.2217	16.2217
9	-0.02009	319676	22524	230328	7200	17.0606	19.2332
10	-0.02110	176855	33998	176179	7200	16.8858	22.0002

З таблиці 3 видно, що BARON зміг покращити результат по часу для перших чотирьох тестових прикладів, затративши на їх

розв'язання відповідно 0.37, 0.61, 26.41 та 309.91 секунд. П'ятий та шостий тестові приклади він не розв'язав, але для них знову покращив розрив між верхньою та нижньою границями на радіус зовнішнього круга C_0 . Так, для $m = 9$ розрив в границях становить 11%, а для $m = 10$ – 23%.

BARON



Packomania

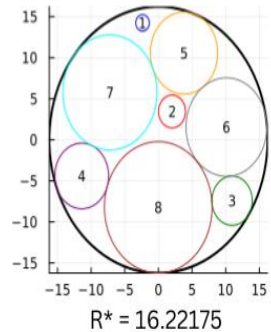
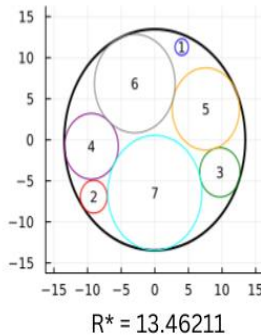
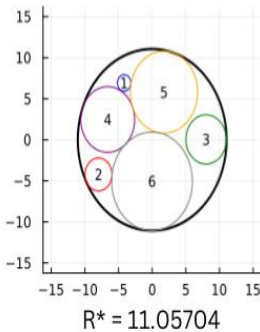


Рис. 3 – *BARON* та *Packomania*: розв'язки задач (1) – (6) для $m \in \{6, 7, 8\}$

На рисунку 3 наведені розв'язки задачі (1) – (6) для тестових прикладів $m \in \{6, 7, 8\}$, які знайдено за допомогою солвера *BARON*, та розв'язки, взяті з сайту <https://www.packomania.com/>. Із рисунка видно, що розв'язки не відрізняються за розміщенням центрів більших кругів C_i , $i = m$.

Література

1. Stetsyuk P., Romanova T., Scheithauer G. On the global minimum in a balanced circular packing problem. *Optimization Letters*. 2016. № 10. P. 1347–1360.
2. NEOS Solver. <https://neos-server.org/> (accessed: 10.06.2024).
3. Tawarmalani M., Sahinidis N.V. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*. 2005. 103(2). P. 225–249.
4. Sahinidis N.V., BARON 21.1.13: Global Optimization of Mixed-Integer Nonlinear Programs, User's manual, 2021.
5. Packomania. <https://www.packomania.com/>
(звернення: 09.08.2023)

МОДЕЛЮВАННЯ ДИНАМІКИ МИСЛЕННЯ ЛЮДИНИ З ВИКОРИСТАННЯМ КОМБІНАТОРИКИ

Тимофієва Н. К.

Міжнародний науково-навчальний центр інформаційних технологій та систем НАН та МОН України

MODELING OF THE DYNAMICS OF HUMAN THINKING USING COMBINATORICS

Abstract. It is shown that the process of the dynamics of human thinking has a combinatorial nature. The electroencephalogram, which conveys the processes of activation of thinking, is given by such a combinatorial configuration as placement with repetitions. The method of modeling the dynamics of human thinking using the significant information space, which exists in two states: convolute and deployed, is considered.

Вступ. В роботі описано спосіб моделювання динаміки мислення людини. Показано, що цей процес має комбінаторну природу, тому для його моделювання використано комбінаторний аналіз. Динаміка мислення пояснюється з використанням знакового інформаційного простору, який як і знаковий комбінаторний, існує в спокої та динаміці.

Постановка задачі. В літературі описано багато способів моделювання природного інтелекту людини та динаміки мислення. Відомі підходи використовують для створення штучного інтелекту.

Але, незважаючи на великі розробки в цій галузі, теорії штучного інтелекту пока не створено.

Підхід, що пропонується. З використанням знакового інформаційного простору, який задається аксіомами, що характерні знаковим комбінаторним просторам, побудовано математичну модель динаміки мислення людини. Інтуїція розглядається як спосіб прийняття рішень в умовах невизначеності за правилами, які формалізувати досить складно.

Про штучний інтелект. Для створення штучного інтелекту необхідно виявити властивості природного інтелекту та розробити спосіб його моделювання. В літературі подано багато означень штучного інтелекту, але точного визначення цієї науки ще немає. Як відомо, під інтелектом розуміють здатність пізнавати навколишній світ та вирішувати різноманітні проблеми. Як синонім користуємося поняттям розум, яке виражає здатність мислити: аналізувати та робити висновки. Одним з понять штучного інтелекту вважають формалізацію проблем та завдань, які подібні до дій, що виконує людина. Різні автори природний інтелект моделюють по-різному. Наприклад у [1] штучний інтелект визначають як властивість цифрової обчислювальної машини реагувати на інформацію, яка поступає на її вхідні пристрої, майже так, як реагує в тих же інформаційних умовах певна людина. Такий підхід ґрунтується на принципі самоорганізації моделі і його називають евристичний. У роботі [2] інтелект людини розглядається як інтуїтивна система. Тобто, під інтуїцією розуміють процес оптимального прийняття рішень по відношенню до зовнішнього середовища. Отже, інтелект людини розглядають як систему, яка сприймає зовнішнє середовище, аналізує інформацію і в залежності від цього приймає рішення про виконання. Але процес мислення в людини може проходити без зовнішньої дії та без обов'язкових вказівок до виконання рішення певним органам. Процес творчості супроводжується різними проявами емоцій, а прийняття рішень в природному інтелекті проводиться в умовах невизначеності різного типу. В задачах цього класу вона пов'язана з: 1) неповною вхідною та поточною інформацією; 2) нечіткою вхідною інформацією; 3) нечітко розробленими правилами обробки та оцінки інформації. В залежності від типу невизначеності наведемо класифікацію природного інтелекту. Вважатимемо, що з поняттям інтелекту пов'язані такі операції як обробка та оцінка інформації. Тоді інтелект людини умовно розділимо на такі рівні [3]:

I рівень. Людина виконує правила, які чітко сформульовано і описано в книгах та підручниках або передаються від учителів без аналізу на їхню точність. Назвемо їх *правилами навчання*. На сьогоднішній день існує багато програм і пристроїв-роботів, які працюють за чітко розробленими правилами. Якщо не враховувати емоції, характерні для вищих живих організмів, то інтелект I-го рівня на сучасному рівні частково реалізовано.

II рівень. В процесі життєдіяльності індивідуум аналізує інформацію на точність і розробляє свої правила поведінки за різних умов, які ґрунтуються на попередньому власному досвіді. Ці правила можуть бути і неточними. Назвемо їх *правилами самонавчання*. Інтелект 2-го рівня завдяки самоналагоджувальним алгоритмам частково реалізовано.

III рівень. Здатність до незалежного від існуючих правил аналізу, обробки та оцінки інформації на точність, розроблення нових точних правил поведінки в умовах невизначеності, які максимально ураховують прогноз майбутніх результатів. Прийняття рішення в умовах невизначеності проводиться на рівні інтуїції за правилами, які складно формалізувати. Вважаємо, що інтуїція – точні правила мислення, які неможливо формалізувати і які закладено на генетичному рівні. Таким особливим мисленням володіє обмежена категорія людей. Назвемо ці правила *правилами інтуїції*.

Інтелект III го рівня реалізувати досить складно, хоча вважають, що цей рівень є справжнім інтелектом. Цей рівень може реалізувати людина з найвищим рівнем інтелекту.

Знаковий інформаційний простір та динаміка мислення людини

В процесі мислення людини вибираються певні об'єкти з пам'яті, тобто спостерігаємо перебір варіантів. Там де є перебір, там має місце комбінаторна природа даного процесу. Змодельємо динаміку мислення людини з використанням знакових комбінаторних просторів, зокрема знакових інформаційних просторів. Наведемо аксіоми, які властиві цим просторам [4].

1. Знакові комбінаторні простори існують в двох станах: спокої (згорнутий) та динаміці (розгорнутий).

2. Згорнутий простір задається інформаційним знаком $\mathfrak{R} = \langle A, T, P, \Xi \rangle$, який містить властивості розгорнутого простору певного типу, де A – одна або кілька базових множин, з елементів $a_i \in A_i \subset A$ яких утворюються розгорнуті комбінаторні простори,

$j \in \{1, \dots, n\}$, $l \in \{1, \dots, \tilde{q}\}$, \tilde{q} – кількість базових множин; T – тип комбінаторного простору; P – правила його розгортання; Ξ – правила згортання знакового комбінаторного простору.

3. Утворення із згорнутого розгорнутих комбінаторних просторів проводиться за рекурентними правилами. Точкою розгорнутого простору є комбінаторна конфігурація певного типу. Розгортанню комбінаторного простору характерна властивість періодичності, яка впливає з рекурентного способу утворення та впорядкування комбінаторних конфігурацій.

4.. Згортання знакового комбінаторного простору певного типу проводиться з точок як одного так і кількох просторів. Згорнутий простір має властивості просторів, з яких він згорнувся.

Точкою знакових комбінаторних просторів є різні типи комбінаторних конфігурацій, зокрема вибірки, які можуть бути як упорядковані так і неупорядковані. Неупорядковані це – сполучення без повторень або з повтореннями. Упорядковані це – розміщення з повтореннями або без повторень.

Оскільки точкою знакового інформаційного простору є розміщення з повтореннями, то ясність мислення проявляється для упорядкованої їхньої множини, де присутнє “золоте” число. Якщо правила розгортання ґрунтуються на строгих законах, то знаковий розгорнутий комбінаторний простір є структуризований. Якщо правила розгортання простору не підпорядковані строгим законам, то розгорнутий простір утворюється безладно. Хаотично впорядковані множини вносять безлад у мислення людини.

Сформулюємо математичну модель інформаційного простору [5].

Як і знаковий комбінаторний, так і знаковий інформаційний простір має комбінаторну природу та існує в двох станах: спокої (згорнутий) і динаміці (розгорнутий). Згорнутий задається інформаційним знаком $\mathfrak{R} = \langle A, T, P, \Xi \rangle$, який містить властивості розгорнутого простору, де $A = \{A_1, \dots, A_n\}$ – одна або кілька базових множин, T – тип комбінаторної конфігурації (точка простору), P – система правил, за якою він розгортається (за строгими законами або хаотично), Ξ – правила його згортання з розгорнутого (одного або кількох).

Інформація пов’язана з функціонуванням людського мозку і перебуває в підсвідомості чи свідомості у вигляді образів, фрагментів мовлення тощо. Вважатимемо, що згорнутий

інформаційний простір – це підсвідомість, а елементи a_{lj} базових множин $A_l \subset A$ – образи, фрагменти мовлення тощо, $l = \overline{1, q}$, q – кількість базових множин, $j = \overline{1, n}$, n – кількість елементів у A_l . Активізується підсвідомість мисленням – системою правил P , завдяки яким із елементів базових множин розгортається частково розгорнутий інформаційний простір – свідомість, що характеризується поняттями, думкою, а комбінаторна конфігурація в ньому є розміщення з повтореннями. Передача інформації (думки) проводиться за допомогою розгорнутого інформаційного простору через мовленнєвий простір, завдяки жестам, рухам, за допомогою письма, графічних зображень. Згортання інформаційного простору із розгорнутих мовленнєвого та різних звукових просторів проводиться слуховим апаратом, а образів – зоровим апаратом.

Інформаційний простір, який існує поза межами людського організму та створений людиною, назвемо штучним інформаційним простором. Він також існує в двох станах: спокої та динаміці. Книги, рукописи, електронні бібліотеки – це штучний згорнутий інформаційний простір. Для його розгортання необхідно знати певні правила (правила читання, доступу до електронних бібліотек тощо). Якщо знаковий інформаційний простір розгортається за строгими правилами, то отримаємо чітке (впорядковане) мислення. В іншому разі прийняття оптимального рішення може бути нечітким (невпорядкованим) по відношенню до законів природи.

В літературі описано експерименти, пов'язані з динамікою мислення. При активізації мислення піддослідної миші спостерігається утворення кругових хвиль навколо точки певної ділянки мозку. Генеруванню ж множин комбінаторних конфігурацій характерна властивість періодичності, а їхнє утворення можна описати концентричними колами з центром, який задається інтервалом нульового рангу

Висновок. Отже, процес мислення людини має комбінаторну природу. Для моделювання його динаміки використовуються знакові комбінаторні простори. Із описаного знакового інформаційного простору видно, що розгорнутий простір задається комбінаторною конфігурацією – розміщення з повтореннями. Вона одночасно є скінченною і нескінченною та визначає нечіткість певної інформації.

Література

1. Амосов Н.М. *Алгоритмы разума*. Киев. Наук. думка, 1979. – 223 с.
2. Косяков Ю.Б. *Мой мозг. Строение, принципы работы, моделирование*. М.: СИНТЕГ, 2001.– 160 с.
3. Надія Тимофієва Класифікація природного та штучного інтелекту. *Фізико-математичне моделювання та інформаційні технології*. Центр математичного моделювання Інституту прикладних проблем механіки і математики. Львів, 2023, випуск 37, С. 128–132.
4. Тимофієва Н.К. Знакові комбінаторні простори та штучний інтелект. *Штучний інтелект*. 2015. № 1-2 (67-68). С.180–189..
5. Тимофієва Н.К. Знаковий інформаційний простір та “золоте” число. *Кибернетика и системный анализ*. 2021. № 5. С. 35– 42.

METHOD OF COLLABORATIVE DECISION-MAKING MODELING BY OPERATORS OF AIR NAVIGATION SYSTEM CONSIDERING THE FACTORS' PRIORITY

¹Shmelova T., ²Sikirda Yu., ³Yatsko M.

^{1,3}National Aviation University

²Flight Academy of the National Aviation University

Abstract. According to statistical data from the Transportation Safety Board of Canada, the significant shares of the incidents are engine failure (13%) and smoke/fire on the aircraft (11%). In time, correct and concerted joint actions of aviation specialists to prevent the development of a situation toward catastrophe is the relevant task. The collaborative decision-making (CDM) models of the Air Navigation System's operators in flight emergencies with and without taking into account objective factors' priority with the help of a decision matrix based on the Wald, Laplace, and Hurwitz criteria are built. An example of the decision-making of a pilot in a real situation in the emergency “Cargo fire during the climb” (route Najaf – Madinah) is presented. Calculations of optimal CDM in the case of cargo fire during the climb by taking into account objective factors' priorities are presented. The obtained CDM models can be used in the Intelligent Decision Support System to enlarge the efficiency of coordinated actions of aviation staff.

Introduction. Today, approximately 80 percent of aircraft accidents occur due to human errors and 20 percent – due to technical failures [1].

Most human factors-related aviation accidents are the errors of the pilots (about 70 percent): deviations from the standard procedures of piloting; fatigue, health issues; errors in adverse meteorological conditions; errors in conflicting instrumental clauses; flying in unfamiliar territory disorientation; disrupting crew interactions; insufficient qualifications for such aircraft. Various ground personnel are related to another 30 percent of errors: incorrect handling, operation, repair, and maintenance of aircraft; air traffic controller operator (ATCO) errors, etc. These facts motivate to carry of scientific research on the reduction in the number of aircraft accidents due to human factors.

To increase the level of flight safety, practical and scientific research on the problems of interaction of aviation experts is increasingly being carried out. Collective labor research in aviation was originally initiated by the National Aeronautics and Space Administration (NASA) to improve interactions among flight crew members. Later, this method was further developed and became one of the most successful tools for avoiding human errors [2]. Using the collaborative decision-making (CDM) model [3] is associated with the effectiveness of the solution, according to the extended requirements of ICAO. Today, as part of the Airport Collaborative Decision-Making (A-CDM) concept, specific solutions that coordinate the interests of all parties on the ground and in the sky involved (airports, aircraft, air traffic, ground handling, maintenance staff, emergency services, etc.) are being implemented. The A-CDM concept is based on the principles of transparency and information exchange; it aims at improving air traffic and controlling airport capacity through reduced delays, improved predictability of situations, and optimization of resource usage [3]. Furthermore, the required day-to-day operational efficiencies can be achieved through the mechanism of Flight and Flow Information for a Collaborative Environment (FF-ICE) [4]. The FF-ICE concept defines navigation information requirements for flight planning, air traffic, flow, and trajectory management; it is the basis for a performance-based Air Navigation System (ANS) [4].

When there is flight emergency (FE), a lot of ANS operators interact with the pilot: the ATCO (issues timely, competent, and substantiated recommendations, instructions, and permits); the flight dispatcher (flight route, alternate aerodromes choice, calculation of the fuel amount for definite flights, etc.); the maintenance staff (issues recommendations for troubleshooting on the aircraft); emergency and rescue services personnel (eliminates the consequences of FE); ground services specialists (organize service in the case of flight delay); units of the state security

(negotiates with terrorists, etc.). To improve the organization of CDM by all aviation operators, based on general information about the flight process and the characteristics of FE, Artificial Intelligence can be used as an additional participant to obtain a hybrid (Human Intelligence + Artificial Intelligence = Man-Machine Intelligence) optimal solution [5]. The synergy of a group of aviation experts can have both a positive effect – counteracting the development of FE, forecasting, and prediction of emergency, and a negative effect – the flight situation developing in the direction of deterioration because there is a limit of time for DM but there are many participants/instructions/solutions, and many factors influencing DM. Since the joint primary goal is to complete the flight plan, in the CDM process operators analyze the current situation based on a common set of factors, albeit from different perspectives. However, the final decision in flight is made by the pilot of the aircraft but with effective support from the main specialist and invited specialists (Fig. 1).

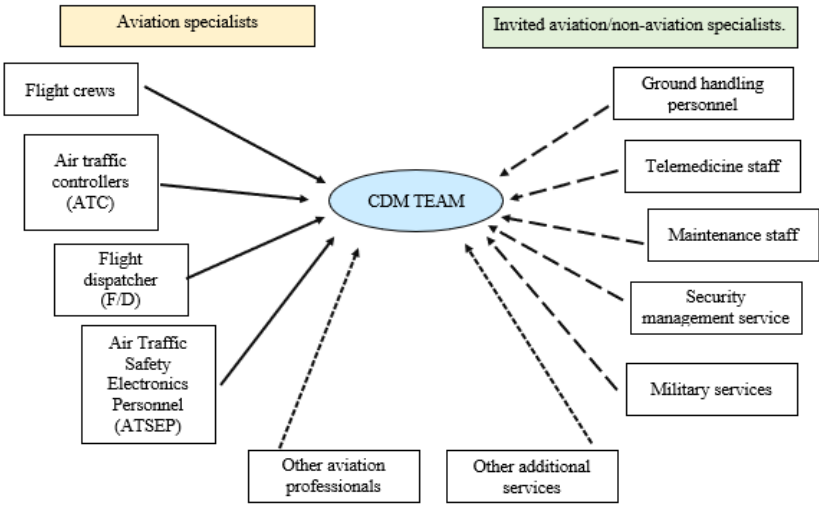


Fig. 1 – CDM team of specialists (aviation and invited)

Every ANS operator plays an important role at a different point in time, because flying safely does not just start with takeoff. Aviation specialists strictly adhere to the manuals and regulatory documents approved in their professional field of activity. Very often, the complexity, content, and characteristics of the documents regulating the activities of individual aviation specialists are different, which does not

allow the development of a common algorithm of action for all aviation personnel under certain conditions, especially in FE, there is uncertainty, lack of information and time to DM. At the same time, conflicts exist between the decisions and actions of participants involved in the CDM process.

In [6], the synchronization of Pilot Flying and Pilot Monitoring operational procedures during cross-monitoring in FE is considered. In [7–9], the deterministic, stochastic, non-stochastic, and neural-network modeling, optimization, and intellectualization of CDM by the teams of ANS operators (pilot, UAV operator, ATCO, flight dispatcher, rescuer, engineer, etc.) in various FE are studied.

Nevertheless, the problem of ANS operator interaction in FE and the low-level formalization of the CDM process under influencing objective and subjective factors that do not allow the application of performance-based approaches for improvement remains unresolved.

The purpose of this work is to build the individual and CDM models of ANS operators in the FE “Cargo fire during the climb” considering the priority of objective factors influencing the development of the emergency.

Main part. *Algorithm of nondeterministic CDM models in FE based on the factors’ priority:*

1. To obtain a team for CDM by ANS operators (pilot, UAV operator, ATCO, flight dispatcher, rescuer, engineer, etc.) according to the situation.

2. Matrices of decision-making in uncertainty – each operator composed a matrix of decisions, where alternative solutions are alternative aerodromes for the route.

3. Each operator considers the same factors in the current situation but with different priorities. For example, for the current situation: fuel reserve on board of the aircraft (always controlled); meteorological situation (of departure, destination, alternate aerodromes, an-route too); capabilities of aircraft, aerodrome, crew (crew operating minima), etc.; location of obstacles in approach, missed approach, and departure sectors too; air situation (intensity of air traffic control sector, radio frequency overload, etc.); commercial conditions.

4. The optimal solution is based on the Wald (W), Laplace (L), and Hurwitz (H) criteria.

5. The priorities of the factors for decision-making in a specific emergency are determined using the opinion of experts and the Expert

Judgment Method, on statistical data, and in the presence of Big Data using an Artificial Intelligence System.

The decision-making matrix for l -operator O_l in uncertainty is presented in Table 1. Analogically for all specialists.

Table 1
The decision-making matrix for participants (l -operator O_l)

	Priority of factors influencing DM in FE						Solutions		
	f_1	f_2	...	f_j	...	f_n	W	L	H, α
R_j	R_1	R_2	...	R_j	...	R_n	-	-	-
c_j	c_1	c_2	...	c_j	...	c_n	-	-	-
γ_j	γ_1	γ_2	...	γ_j	...	γ_n	-	-	-
The DM matrix with the priority of factors									
A_i	$\gamma_1 u'_{i1}$	$\gamma_2 u'_{i2}$...	$\gamma_j u'_{ij}$...	$\gamma_n u'_{in}$	W_1	L_1	H_1, α
...
A_i	$\gamma_1 u'_{i1}$	$\gamma_2 u'_{i2}$...	$\gamma_j u'_{ij}$...	$\gamma_n u'_{in}$	W_i	L_i	H_i, α
...
A_m	$\gamma_1 u'_{m1}$	$\gamma_2 u'_{m2}$...	$\gamma_j u'_{mj}$...	$\gamma_n u'_{mn}$	W_m	L_m	H_m, α

In Table 1: R_j are the ranks of the factors; c_j are the intermediate grades; γ_j are the weight coefficients; f_j are the factors influencing decision-making in FE; A_i are the alternative decisions based on factors' priority; A'_i are the alternative decisions without factors' priority; u'_{ij} are the expected outcomes. The optimal solution is based on the Wald (W), Laplace (L), and Hurwitz (H) criteria. The priorities of the Factors for DM in a specific emergency are determined using the opinion of experts and the Expert Judgment Method, on statistical data, and in the presence of Big Data using an Artificial Intelligence System.

According to statistical data from the Transportation Safety Board of Canada collected for the period from 2007 to 2017 [10], the significant shares of the incidents are engine failure (13%) and smoke/fire on the aircraft (11%). In-flight fire is one of the most dangerous events for an aircraft crew. A fire on an aircraft can quickly lead to the catastrophic waste of the aircraft if the crew does not take aggressive action. Once a fire does break out, it is a low probability that the crew will be able to extinguish it. In the event of a fire on board, the pilot has an average of about 17 minutes to bring the aircraft to the ground. Unlimited fire can destroy an aircraft in as little as 20 minutes [11]. Fire can fully destroy a

smoky cabin in 6-10 minutes [12]. Time is of the essence when fighting fires in flight. Aircraft fires can happen in different places and for different reasons. Aircraft fires are generally divided into three types: engine fire, cabin fire, and concealed fire [13].

The crew cabin, passenger's salon, baggage, and cargo cabin fire start in flight for many reasons, for example, malfunction of electrical components, lithium-ion batteries, wiring, and circuit protection. Many cabin fires are caused by human error (for example, improper storage of lithium-ion batteries in gadgets, transportation of dangerous goods, or terrorism). Concealed fire is difficult to reach and extinguish (for example, fires behind panels on sidewalls or in upper areas). Overhead is the aircraft's entertainment system, numerous wiring harnesses, control surface cables – components of the air conditioning system, oxygen system, etc. Failure of any of these components could result in a fire in the upper compartment. Often, electric arcs along bundles of wires cause hidden fires in flight. The electric arc burns other surrounding materials.

Consequences of the fire in the air [14; 15]:

1. Smoke and fumes. Smoke can deteriorate visibility inside the aircraft. An electrical fire usually produces a lot of white thick smoke that can blind the crew, unable to see the flight panel or see out the glass. So, the crew will not be able to control the aircraft unless the smoke is cleared. Smoke and fumes are highly toxic and irritating to the eyes and lungs of crew and passengers. Thus, smoke can quickly incapacitate a crew if protective measures are not taken.

2. Warmth (rise of temperature). The warmth from the fire will affect the aircraft's systems and ultimately affect the aircraft's structural integrity, resulting in the loss of control.

There is presented an example of CDM in the FE “Cargo fire during the climb”. After departure from Najaf airport during the climb phase after passing flight level 300 “AFT CARGO FIRE” warning lights were illuminated. The crew performed the instructions from QRH “Cargo fire”. The decision was to divert to Baghdad airport. All emergency services were informed. The landing was successful. Nobody was injured.

Initial data:

1. Flight: FBA2219, 03.02.2023.

2. Aircraft: B737-900ER

3. Flight-technical characteristics of B737-900ER:

Maximum structural weights:

- maximum takeoff mass – 85139 kg;
- maximum landing mass – 71350 kg;
- maximum zero fuel mass – 67721 kg;
- maximum ramp mass – 85366 kg.

Airplane length – 42.11 m.

Fuselage width – 3.76 m.

Actual masses:

- crew – 8, passengers – 211, baggage – 3286 kg;
- dry operation mass (average) – 44600 kg;
- payload (average) – 20100 kg;
- fuel (average) – 10000 kg;
- takeoff mass – 74700 kg.

4. Route: Najaf (ORNI) (A_1) – Madinah (OEMA) (A_2).

5. Flight stage: climb, passing flight level 300.

6. Alternate aerodromes:

- Baghdad (ORBI) (A_3);
- Jeddah (OEJN) (A_4).

7. There are three operators in the CDM process: pilot (O_1), ATCO (O_2), and specialist of ground services (O_3). Factors influencing decision-making for each operator:

- $\{x_j\}$ – factors taken into account by operator O_1 (pilot);
- $\{y\}$ – factors taken into account by operator O_2 (ATCO);
- $\{z_j\}$ – factors taken into account by operator O_3 (specialist of ground services).

For rational CDM, each operator analyzed and considered the current flight situation. The operators composed the decision-making matrices, where alternative solutions are the departure aerodrome Najaf, destination aerodrome Madinah, and alternative aerodromes Baghdad and Jeddah. Each operator took into account the same factors in the current situation but with different advantages.

The general objective factors for each operator (x_j, y_j, z_j) are taken into account while decision-making:

- x_1, y_1, z_1 – fuel reserve on board;
- x_2, y_2, z_2 – remoteness of the alternate aerodrome;
- x_3, y_3, z_3 – technical characteristics of the runway;
- x_4, y_4, z_4 – meteorological conditions;
- x_5, y_5, z_5 – approach lighting system;
- x_6, y_6, z_6 – available approach system;
- x_7, y_7, z_7 – available navigation aids;
- x_8, y_8, z_8 – performance characteristics of the aircraft;
- x_9, y_9, z_9 – radio communication;
- x_{10}, y_{10}, z_{10} – air traffic intensity;
- x_{11}, y_{11}, z_{11} – commercial point;
- x_{12}, y_{12}, z_{12} – firefighting, search, and rescue category.

As the main alternative variants, the aerodromes Baghdad and Najaf are considered. Both airports have firefighting, search, and rescue categories available for this aircraft (Baghdad preferable – higher category). Aircraft mass was around 3400 kg more than the maximum landing mass (consider overweight landing). The landing distance available is higher in Baghdad than in Najaf (Baghdad is preferable):

- Baghdad (ORBI) runway length = 4000 m (ILS33R – 3666 m), runway width is 60 m;
- Najaf (ORNI) runway length = 3000 m (ILS28 – 2704 m), runway width is 45 m.

Baghdad is the main base; technical equipment and qualified personnel are there. All other services are provided in Baghdad. Both airports were closed to the point of the event. Expected outcomes considered by the pilot (operator O_1) in FE “Cargo fire during the climb” with and without considering objective factors’ priority are represented in Table 2. The decision-making matrices for other operators (ATCO (O_2) and specialist of ground services (O_3)) are built analogically.

Table 2

The decision-making matrix for operator O_1 (pilot)

	Factors												Solutions			
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	W	L	$H, \alpha=0.5$	
	1, 2, 3, 4, 5				6, 7, 8, 9, 10, 11, 12						1, 2, 3, 4, 5					
R_j	3	3	3	9	9	9	9	9	9	9	3	3	-	-	-	
c_j	0.8	0.8	0.8	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.8	0.8	-	-	-	
γ_j	2	2	2	7	7	7	7	7	7	7	2	2	-	-	-	
γ_j	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.1	-	-	-	
γ_j	4	4	4	5	5	5	5	5	5	5	4	4	-	-	-	
A_i																
A_1	0.9	0.9	0.8	0.4	0.3	0.3	0.2	0.4	0.4	0.4	0.8	0.8	0.2	0.5	0.60	
A_2	5	5	2	1	2	6	7	1	1	1	2	2	7	8		
A_3	0.8	0.8	0.8	0.4	0.3	0.3	0.3	0.4	0.4	0.4	1.2	0.6	0.3	0.5	0.80	
A_4	2	2	2	1	2	6	2	1	1	1	3	8	2	8		
A_3	0.5	0.5	1.3	0.4	0.4	0.4	0.4	0.4	0.3	0.4	1.2	1.3	0.3	0.6	0.90	
A_4	5	5	6	1	5	5	1	5	6	1	3	6	6	7		
A_4	0.8	0.8	0.8	0.4	0.3	0.3	0.3	0.4	0.3	0.3	0.8	0.6	0.3	0.5	0.60	
A_4	2	2	2	1	6	2	2	1	6	6	2	8	2	4		
A'_i																
A'_1	7	7	6	9	7	8	6	9	9	9	6	6	6.0	7.4	7.50	
A'_2	6	6	6	9	7	8	7	9	9	9	9	5	5.0	7.5	7.00	
A'_3	4	4	10	9	10	10	9	10	8	9	9	10	0	0		
A'_4	6	6	6	9	8	7	7	9	8	8	6	5	0	8.5	7.00	
A'_4	6	6	6	9	8	7	7	9	8	8	6	5	0	0		
A'_4	6	6	6	9	8	7	7	9	8	8	6	5	5.0	7.0	7.00	
A'_4	6	6	6	9	8	7	7	9	8	8	6	5	0	8		

In Table 2: R_j are the ranks of the factors (1 – very bad; 10 – very good); c_j are the intermediate grades; γ_j are the weight coefficients; A_i are the alternative decisions based on factors’ priority (A_1 is Najaf, A_2 is Madinah, A_3 is Baghdad, A_4 is Jeddah); A'_i are the alternative decisions without factors’ priority (A'_1 is Najaf, A'_2 is Madinah, A'_3 is Baghdad, A'_4 is Jeddah). Optimal solutions are based on the Wald (W), Laplace (L), and Hurwitz (H) criteria. Factors’ priority: 1, 2, 3, 11, 12. The optimal landing aerodrome follows the pilot's decision with factors’ priority by all criteria is Baghdad (A_3); without taking into account factors’ priority based on the Wald (W) and Hurwitz (H) criteria – is Najaf (A_1), Laplace criterion (L) – is Baghdad (A_3). To determine the consistency of operators, the collective decision-making matrix is constructed (see Table 6), in which the objective factors for the operators (pilot (O_1), ATCO (O_2), specialist of ground services (O_3)) are identical, and the solutions of the operators are taken from individual matrices. In the CDM

matrix, the operators' opinions (subjective factors) are used. The optimal CDM for operators in FE "Cargo fire during the climb" based on the Wald (W), Laplace (L), and Hurwitz (H) criteria is represented in Table 3.

Table 3

The CDM matrix for all operators (pilot, ATCO, specialist of ground services)

	Operators/Solutions											
	O_1	O_2	O_3	W	O_1	O_2	O_3	L	O_1	O_2	O_3	$H, \beta = 0.5$
A_1^*	0.27	0.27	0.27	0.27	0.58	0.63	0.56	0.49	0.61	0.68	0.61	0.65
A_2^*	0.32	0.36	0.27	0.27	0.58	0.69	0.53	0.52	0.77	0.80	0.55	0.67
A_3^*	0.36	0.36	0.36	0.36	0.67	0.64	0.64	0.56	0.86	0.86	0.86	0.86
A_4^*	0.32	0.32	0.32	0.32	0.54	0.60	0.54	0.49	0.57	0.70	0.57	0.64

The optimal CDM is determined by objective and subjective factors taking into account objective factors' priority to follow the decisions of all operators (pilot, ATCO, specialist of ground services) by all criteria (Wald, Laplace, and Hurwitz) is landing at the aerodrome Baghdad (A_3).

Conclusion. The non-stochastic individual and CDM models of the ANS operators in a real situation in the FE "Cargo fire during the climb" (route Najaf – Madinah) with and without taking into account objective factors' priority with the help of a decision matrix are built. The optimal CDM based on objective factors' priority by Wald, Laplace, and Hurwitz criteria is determined. The obtained CDM models can be used in the Intelligent Decision Support System to enlarge the efficiency of coordinated actions of aviation staff.

The authors took part in a new approach to the practical training for aviation specialists using CDM in education – "Collaborative Decision-Making – Education" (CDM-E) for effective solutions in training on examples of emergencies [16; 17].

Literature

1. Rankin W. MEDA investigation process. *AERO*. Vol. 26. 2017. P. 15–21.
2. J. A. Wise, V. D. Hopkin, D. J. Garland (Eds.), Handbook of aviation human factors, 2nd ed., CRC Press, Florida, USA, 2016. <https://doi.org/10.1201/b10401>.
3. Airport-Collaborative Decision Making: IATA Recommendations. Canada, Montreal: IATA, 2018. 20 p.

4. Liang D., Cropf K., Sherwin R., Porter G., Masarky S., Sutton F. Operational evaluation of FF-ICE/R2. *Proceedings of the IEEE 2019 Integrated Communications, Navigation and Surveillance Conference (ICNS-2019)*. USA, VA, Herndon, 2019. P. 649–658. <https://doi.org/10.1109/ICNSURV.2019.8735320>.
5. Artificial Intelligence Roadmap: A human-centric approach to AI in aviation. Germany, Cologne: European Union Aviation Safety Agency, 2020. 33 p.
6. Socio-Technical Decision Support in Air Navigation Systems: Emerging Research and Opportunities / T. Shmelova, Yu. Sikirda, N. Rizun, A.-B. M. Salem, Yu. Kovalyov. USA, Hershey: IGI Global Publ., 2018. 305 p. <https://doi.org/10.4018/978-1-5225-3108-1>.
7. Shmelova T., Chialastri A., Sikirda Yu., Yatsko M. Decision making models by the pilot in flight emergency “Engine failure during take-off”. *CEUR Workshop Proceedings*. Vol. 3101. 2021. P. 347–365.
8. Shmelova T., Sikirda Yu., Yatsko M., Kasatkin M. Collective models of the aviation human-operators in emergency for Intelligent Decision Support System. *CEUR Workshop Proceedings*. Vol. 3156. 2022. P. 160–174.
9. Shmelova T., Yatsko M., Sikirda Yu. Collaborative-factor models of decision making by operators of the Air Navigation System in conflict or emergency situations. *Communications in Computer and Information Science (CCIS)*. Vol. 1635. 2022. P. 391–409. https://doi.org/10.1007/978-3-031-38082-2_22.
10. Statistical summary: Air transportation occurrences in 2017. *Transportation Safety Board of Canada*. URL: <https://www.bst-tsb.gc.ca/eng/stats/aviation/2017/ssea-ssao-2017.html>. (date of request: 18.05.2024).
11. Fire in the air. *SKYbrary*. <https://www.skybrary.aero/articles/fire-air>. (date of request: 18.05.2024).
12. How common are commercial airplane fires? *FireSystems Inc*. URL: <https://firesystems.net/2019/12/08/how-common-are-commercial-airplane-fires/>. (date of request: 18.05.2024).
13. Rossier R. Prevention strategies and what to do if it happens to you. *AOPA*. URL: <https://www.aopa.org/training-and-safety/students/flighttestprep/skills/aircraft-fires>. (date of request: 18.05.2024).
14. Protecting aircraft and passengers from cargo fires. *AIRBUS*. URL: <https://safetyfirst.airbus.com/protecting-aircraft-and-passengers-from-cargo-fires/>. (date of request: 18.05.2024).

15. In-flight fire: Guidance for flight crews. *SKYbrary*. URL: <https://www.skybrary.aero/articles/flight-fire-guidance-flight-crews>. (date of request: 18.05.2024).

16. Shmelova T., Sikirda Yu., Yatsko M., Kolotusha V. Intelligent Integrated Training System for the Aviation Specialists “Collaborative Decision-Making – Education” (CDM-E). *CEUR Workshop Proceedings*. Vol. 3538. 2023. P. 168–180.

17. Shmelova T., Sikirda Yu., Yatsko M., Stratonov V. Integration of Deterministic, Stochastic, and Non-Stochastic Models to Obtain an Optimal Collaborative Decision in the Flight Emergency. *Lecture Notes in Networks and Systems (LNNS)*. Vol. 992. 2024. P. 205–223. https://doi.org/10.1007/978-3-031-60196-5_17.

Застосування графів до розв’язування СЛАР

Філімоніхіна І.І.

Центральноукраїнський національний технічний університет

Applications of graces to solving systems of Linear algebraic equations

Filimonikhina I.I.

Central Ukrainian national technic university

Abstract. Considered some applications of graces to solving systems of linear algebraic equations (SLAE). Any SLAE can be representing by oriental grace. By transforming nodes corresponding to unknown variables into simple cascade nodes, one-variable first-order equations are obtained. The procedure for such transformation is described and examples are given.

Графи широко застосовуються в багатьох розділах прикладної математики, в програмуванні, електротехніці, в теорії планування та управління [1-4]. Досить часто математичні моделі в цих галузях описуються системами лінійних алгебраїчних рівнянь (СЛАР). Існує багато традиційних методів розв’язування СЛАР [5].

Розглянемо, як можна представляти і розв’язувати СЛАР засобами теорії графів.

СЛАР можна зобразити у вигляді орієнтованого графа наступним чином:

- праві частини вважаємо стоками;
- невідомі змінні вважаємо вершинами графа;

- коефіцієнти при невідомих вважаємо ребрами графа.

Суть розв'язування СЛАР за допомогою графів полягає в послідовному перетворенні вузлів, що відповідають певним змінним, в прості каскадні вузли і поступовому виключенні змінних. В кінці такої процедури залишається лише одна змінна. Якщо СЛАР n -го порядку, то необхідно здійснити $n-1$ крок. По остаточному графу записують рівняння першого порядку з однією змінною.

При перетворенні графів діють такі правила:

- 1) декілька паралельних ребер замінюються одним, причому вага ребра дорівнює сумі ваг ребер;
- 2) декілька послідовних ребер замінюються одним в вагою ребра, що дорівнює добутку ваг ребер;
- 3) якщо джерело і стік поміняти місцями, то вага ребра виражається числом, оберненим до початкової ваги ребра;
- 4) якщо міняємо напрям ребра, то ваговий коефіцієнт змінює знак на протилежний.

За допомогою графів можна розв'язувати СЛАР будь-якого порядку.

В процесі розв'язання поступово переносять сток в одну з вершин, що відповідає деякій невідомій, потім цю вершину виключають з графа. Після цього переходять до цієї ж процедури стосовно наступної вершини.

Розглянемо декілька прикладів розв'язання СЛАР за цією методикою.

Приклад 1. Розв'язати систему:

$$\begin{cases} 2x + y = 12, \\ 3x - 5y = 5. \end{cases}$$

На рисунку 1 а) зображено граф по початковій системі. На рисунку 1 б) зображено граф після перенесення стоку в вершину x . На останньому рисунку маємо остаточний граф для знаходження змінної y .

Запишемо рівняння для цього графа:

$$1 \cdot \frac{3}{2} + \frac{13}{2} \cdot y = 8$$

$$\frac{13}{2} y = \frac{13}{2}$$

$$y = 1$$

Щоб знайти змінну x , по рисунку 1 б) запишемо рівняння для вузла, що містить x .

$$x = \frac{3}{2}y + \frac{1}{2}$$

Щоб знайти x , підставляємо в рівняння значення y :

$$x = \frac{3}{2} \cdot 1 + \frac{1}{2} = 2.$$

Відповідь: (2;1)

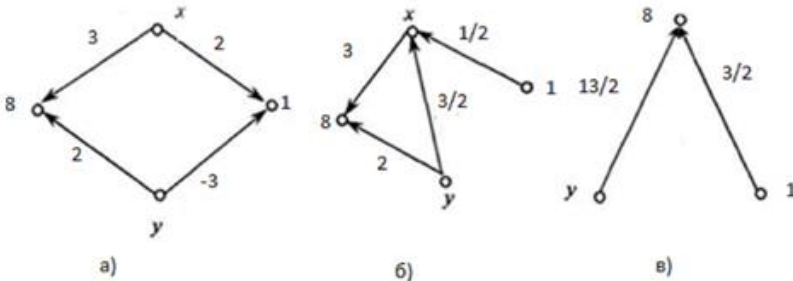


Рисунок 1.

Приклад 2. Розв'язати систему;

$$\begin{cases} 3x + 2y + 5z = 10, \\ 2x + 3y + 4z = 9, \\ 4x + 7y + 3z = 14. \end{cases}$$

На рисунку 2а) зображено граф по початковій системі. На рисунку 2б) зображено граф після перенесення стоку в вершину y . На рисунку 2г) зображено граф після перенесення стоку в вершину x . На останньому рисунку маємо остаточний граф для знаходження змінної z .

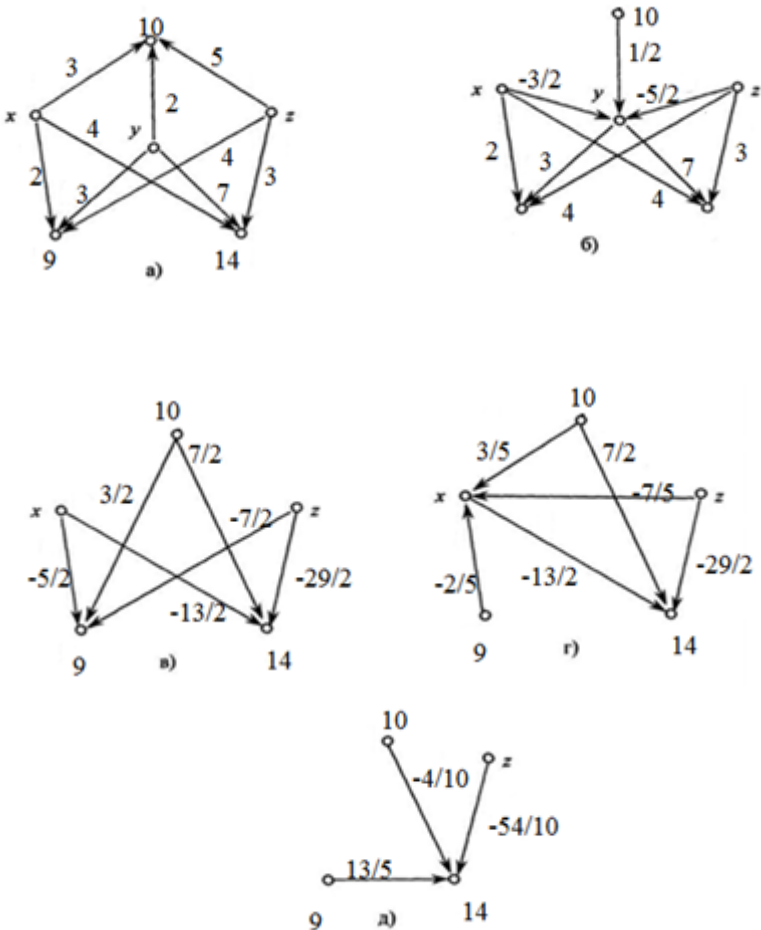


Рисунок 2

Література

1. Бардачов Ю.М., Соколова Н.А., Ходаков В.Э. Дискретна математика. – К.: Вища школа, 2002.
2. Оре О. Графы и их применение. – М.: Мир, 2002.
3. Ядренко М.Й. Дискретна математика. – К.: КНУ, 2002.
4. Семенюта М.Ф., Шульгин В.А. Матрицы, ассоциированные с D-дистанционными магическими графами, и их свойства. Кібернетика і системний аналіз. 2019. Т. 55, №3. С.112-120.

5. Лисенко В.І. Розв'язування системи лінійних рівнянь за допомогою графів // У світі математики, 1974, № 5, с. 48-56.

ВИЯВЛЕННЯ СФАЛЬСИФІКОВАНОЇ ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ НАЇВНОГО БАЄСОВОГО КЛАСИФІКАТОРА

Артем Ховрат, Володимир Кобзєв

Харківський національний університет радіоелектроніки

DETECTION OF FALSIFIED INFORMATION USING NAÏVE BAYES CLASSIFIER

Abstract. The current work addresses the issue of information falsification in the context of artificial intelligence-based generation technologies. The research is a continuation of a series of studies on methods of classifying such data in the social of social instabilities. The naive Bayes classifier has been chosen as the target approach, and a set of features identified during the analysis of fake news characteristics has been proposed. To test the speed and accuracy of the model, an approach utilizing linear additive convolution with weighting coefficients has been determined. A hybrid neural network created during previous research has been selected as the model for comparison. It has been determined that the proposed probabilistic approach is more effective. However, considering the specificity of the comparison, the results serve as a basis for further study on the feasibility of using the selected family of models.

За останні декілька років технології, спроможні сфальсифікувати зображення та голос чи згенерувати текст, суттєво еволюціонували. На це вказують численні наукові дослідження вчених з усього світу [1]. І якщо проблема із піддробкою відео ще не гостро стоїть у суспільстві, оскільки відповідні засоби не є “ідеальними”, а спотворення аудіоінформації лише нещодавно досягло необхідного рівня [2], то видозміна тексту з шахрайською метою вже декілька років активно використовується в соціальних мережах. Гострішою проблема стає в умовах військово-політичної нестабільності, коли будь-яка заява сприймається крізь призму інтенсифікованих емоцій, що сповільнюють процес критичного мислення [3]. Окрім безпосередньо обману населення, використання подібних текстів може нести пряму небезпеку під час військових операцій [4].

Загальносвітова політична тенденція, спрямована на боротьбу з подібною інформацією, в цілому є позитивною [5], концентруючи свою увагу на боротьбі з проблемними “новинами”. У якості прикладу можна згадати діяльність Центру протидії дезінформації. Однак потрібно розуміти, що зі збільшенням кількості фейків загальна якість їх виявлення буде спадати. Окрім того можуть виникати ситуації, коли “викривач” подробиць є тим, хто спотворює дані на свій розсуд. Аби прибрати суб’єктивність та автоматизувати процес виявлення сфабрикованої інформації, було вирішено в рамках чинної роботи дослідити ефективність використання однієї з груп алгоритмів спрямованих на виявлення текстів-подробок.

Проведені раніше дослідження концентрували свою увагу на використанні нейронних мереж, зокрема поєднанні декількох згорткових та модернізованих рекурентних [6]. Однак ці підходи не є єдиними, хоча і наразі активно використовуються. Після ознайомлення з сучасними світовими науковими напрацюваннями в обраній сфері, було вирішено зупинитися на ймовірнісних моделях, зокрема тих, що наявні в системах виявлення спаму для поштових сервісів. Наприклад, наївний баєсів класифікатор. Однак ця модель, зважаючи на свою простоту, вимагає додаткових методів передобробки даних, які могли б покращити її ефективність.

Виходячи із зазначеного вище, метою цього дослідження є перевірка ефективності використання наївного баєсового класифікатора у поєднанні з алгоритмом передобробки даних. Задля досягнення цієї мети передбачаються наступні завдання:

- побудувати алгоритм передобробки даних та сформувані відповідну модель на основі наївного баєсового класифікатора ;
- сформувані набір критеріїв ефективності та загальне експериментальне оточення ;
- здійснити порівняння створеного підходу із розглянутою в ході попередніх досліджень гібридною нейронною мережею.

Здійснимо поступовий розгляд зазначених завдань.

Сутність наївного баєсового підходу полягає у використанні теореми Баєса для визначення приналежності спостереження до одного з класів при припущенні незалежності змінних. Зазначене обмеження можна вважати виправданим, оскільки цільові ознаки, описані нижче не впливають на наступні їхні значення.

Класична формула для теореми Баєса має наступний вигляд:

$$p(C_k|\mathbf{x}) = \frac{p(C_k)p(\mathbf{x}|C_k)}{p(\mathbf{x})}$$

де C_k – цільові класи даних (у чинному випадку передбачено два класи – інформація є сфабрикованою чи ні); \mathbf{x} – вектор ознак; $p(C_k)$ – апіорна ймовірність приналежності випадково обраного спостереження до класу C_k ; $p(\mathbf{x}|C_k)$ – ймовірність вектору ознак при заданому класі C_k ; $p(\mathbf{x})$ – апіорна ймовірність ознак; $p(C_k|\mathbf{x})$ – апостеріорна ймовірність приналежності обраного спостереження до класу C_k за умови вектору ознак \mathbf{x} .

Найважливішим при побудові відповідної моделі є формування вектору ознак. Задля формування набору цільових ознак було вирішено використати наступний набір особливостей фейкової інформації:

- використання неприродної кількості риторичних запитань, якщо мова йде про контекстуальне викривлення суспільно важливої інформації. Проведені лінгвістичні дослідження зазначають, що в офіційно-діловому та публіцистичному стилях, що мають використовуватися ЗМІ, подібний тип мовленнєвих конструкцій майже не з'являється [7]. Ця особливість характерна як для текстових новин, так і аудіо, і відео ;

- відсутність заперечувальних конструкцій задля зменшення когнітивного навантаження у поєднанні з песимістичним забарвленням обраних слів. Як приклад можна привести заміну слова “проблема” на “катастрофа” ;

- використання закликів та спонукань у недоречних формах. Необхідно зазначити, що у випадку контекстуального викривлення, що має на меті замінити реальні новини, подібні конструкції одразу вказують на неправдивість та некоректність вказаної інформації ;

- використання невинуватливої кількості займенників. Цей фактор у більшості випадків має місце для контекстуального викривлення, що наслідують публіцистичний стиль викладення.

Враховуючи зазначене вище сформовано наступний набір цільових ознак:

- кількість риторичних конструкцій по відношенню до загальної кількості речень у повідомленні ;

- кількість заперечувальних конструкцій по відношенню до загальної кількості речень у повідомленні ;

- частотно-емоційна характеристика 50 найбільш популярних новин за дату створення повідомлення. Цей показник дозволить

врахувати новинне зовнішнє середовище і відповідно скорегувати оцінку класифікатора ;

- показник підозрілості, визначений на основі переліку слів, які часто вживаються у фейковій інформації;

- вага повідомлення ;

- показник настрою повідомлення.

Заключний індиктор є найбільш важливим для подальшого дослідження. Аби здійснити перетворення отриманої текстової інформації у числове представлення скористаємось наступним алгоритмом:

- розбиваємо текст на речення та окремі слова і видаляємо з утвореного словника конструкції без суттєвого лексикографічного навантаження. Наприклад: “однак”, “це”, “або”, “тощо” ;

- здійснюємо операцію лематизації та стемінгу – визначення лем та основ усіх елементів словника ;

- визначаємо частотну характеристику кожного слова та його емоційного забарвлення за допомогою засобів Sentiment Analysis, збудованих у моделі nltk мови програмування Python3 ;

- агрегуємо та нормуємо частотно-емоційний показник для кожного речення.

Як зазначалося вище, описану модель необхідно порівняти зі створеною в ході попередніх досліджень нейромережею (RCNN), що є поєднанням згорткових (CNN) та рекурентних (RNN) підмереж. Аби уникнути звичних проблем зі “спадаючим” та “вибуховим” градієнтами при обробці тексту, було вирішено використати не класичну RNN архітектуру, а двонаправлену рекурентну нейромережу з підтримкою довго- та короткотривалої пам’яті. Таким чином, архітектуру RCNN можна подати у вигляді наведеному нижче (рис. 1).

Варто також зауважити, що для пришвидшення вказаної моделі вирішено використати технологію MapReduce на основі Hadoop, яка дозволяє розподілити навантаження на окремі вузли.

Перейдемо до опису експериментального середовища. Враховуючи специфіку запропонованого дослідження, було обрано метод контрольованого експерименту. Базове середовище виконання має наступний набір характеристик: CPU - Intel Core i5-1135G7; RAM - 16 Гб; VRAM - 4 Гб.

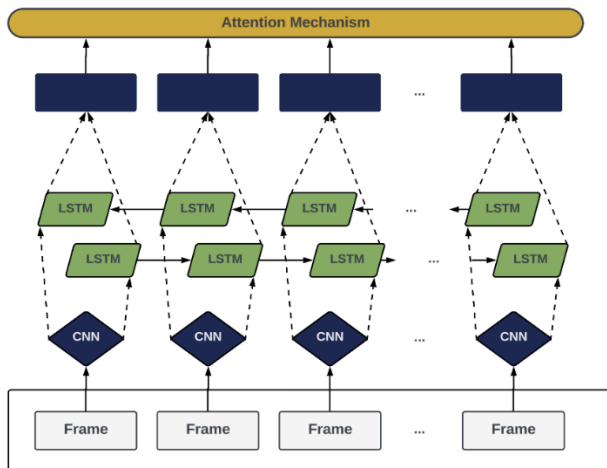


Рис. 1 – Схематичне зображення RCNN архітектури

З огляду на те, що розглядається задача класифікації для соціально гострих процесів, найбільш важливими критеріями є економія часу та точність класифікації. Вказані характеристики майже в повному обсязі були продубльовані на віртуальних вузлах, на яких планується здійснюватися процес частково обчислення (було зменшено RAM з 16 до 8). Їх кількість становитиме від 3 до 4 (2 у випадку паралелізації двонаправлених нейронних мереж).

У якості засобу обчислення часу виконання було обрано бібліотеку `datetime` для Python 3, що має точність до наносекунди. Аби базові обчислення не сповільнювали роботу програми вирішено використати бібліотеки `numpy` та `polars`. Для обробки природніх мов (враховуючи лематизацію, токенизацію та інші необхідні функції) було вирішено обрати python-версію бібліотеки `nlTK`. Задля реалізації нейронних мереж вирішено використати `tensorflow` із застосуванням інструментарію, що надає `pipeline` субмодуль.

У якості цільових наборів даних вирішено розглянути новини, що стосуються повномасштабного вторгнення Росії на територію України та присвячені виборчому процесу у 2019 році, який супроводжувався появою великої кількості неправдивої інформації. Кожна з цих вибірок ділитиметься у співвідношенні 80 до 20 на навчальну та тестову підвибірки, відповідно.

Загалом було обрано наступний перелік факторів:

- показник точності ;

- економія часу роботи моделі ;
- економія мінімально допустимого обсягу даних для досягнення точності у 80% (максимальна кількість – 500000 елементів).

Визначившись з критеріями, здійснимо опис відповідних шкал оцінювання.

Економія часу роботи моделі вимірюватиметься у секундах за допомогою вказаної вище бібліотеки. Сам показник при цьому не обмежуємо. Аби знизити вплив можливих похибок вимірювання, викликаних проблемами з точністю роботи часових модулів чи оточення, було вирішено проводити по 5 замірів для показників часу та перевірити точність прогнозування на двох вибірках даних.

Точність класифікації визначена за допомогою комбінації F1-score та Precision, нормалізована в межах від 0 до 1. Замір точності здійснюватимемо для двох вибірок і братимемо усереднене значення.

Аби визначити яка з моделей є найбільш ефективною за наведеними вище критеріями було вирішено застосувати принцип лінійної адитивної згортки з ваговими коефіцієнтами. Для визначення вагових коефіцієнтів було проведене експертне оцінювання серед аналітиків (кількість рецензентів склала 50 осіб). Отже, перейдемо до визначення вагових коефіцієнтів. У питанні класифікації важливим є показник точності. Загалом важливість можна описати наступним чином:

- для точності: 16 очок ;
- для економії часу роботи моделі: 2 очки ;
- для економія мінімального допустимого обсягу даних для досягнення потрібного рівня точності: 2 очки.

З огляду на це отримуємо наступні вагові коефіцієнти для кожного критерію:

- для точності: $16/20 = 0.8$;
- для економії часу роботи моделі: $2/20 = 0.1$;
- для економія мінімального допустимого обсягу даних для досягнення потрібного рівня точності: $2/20 = 0.1$.

Наступним важливим елементом експериментального середовища є визначення можливих похибок. Виходячи із описаного плану можна виокремити наступні фактори, що здатні вплинути на результат:

- під час перевірки економії часу: людський фактор та інструментальна похибка ;

– під час перевірки точності: проблема даних.

Аби пом'якшити зазначені невизначеності, як зазначалося, вимірювання показників здійснюватиметься декілька разів.

Агреговані результати проведеного експерименту наведені у таблиці нижче (табл. 1).

Таблиця 1
Результати експерименту

Підхід	Час роботи, мс	Точність	Мінімальний обсяг
RCNN	221	97%	10000
НБК	154	85%	8000

Переведемо вказані дані у єдину безрозмірну шкалу та нормалізуємо. Отримані значення наведено у табл. 2.

Таблиця 2
Нормалізовані та видозмінені результати експерименту

Підхід	Економія часу роботи	Точність	Економія обсягу
RCNN	0.00	0.97	0.980
НБК	1.00	0.85	0.984

Ураховуючи отримані результати та вагові коефіцієнти, можна встановити, що значення лінійної адитивної згортки для випадку RCNN дорівнює 0.874, а для випадку наївного баєсового класифікатора – 0.8784. Таким чином другий підхід в рамках запропонованих критеріїв є більш ефективним. Однак варто зауважити, що він істотно гірший з точки зору точності за RCNN і, фактично, отримане значення є наслідком недоскональності підходу з лінійною адитивною згортокою.

Чинна робота ставила на меті перевірити ефективність використання ймовірнісного підходу на основі наївного баєсового класифікатора для виявлення сфабрикованих текстових даних.

Задля виконання поставленої мети та завдань був сформований алгоритм передобробки даних, який дозволив врахувати настрої текстового представлення та зовнішній контекст. Задля порівняння результатів класифікації була визначена задача багатокритеріального вибору та використана, отримана в ході

попереднього дослідження, модель на основі поєднання декількох нейромереж.

У результаті ефективність ймовірнісного підходу доведена, однак питання щодо достовірності отриманих результатів при використанні іншого способу порівняння і подальшого вдосконалення алгоритму передобробки залишається відкритим, зокрема через відносно низьку точність моделі. Виходячи із зазначених уточнень можна вказати, що отриманий висновок відповідає загальносвітовій практиці.

Література

1. Anders M. “Fake News Detection. European Data Protection Supervisor”, available at: https://edps.europa.eu/press-publications/publications/techsonar/fake-news-detection_en (last accessed 27.05.2024).

2. Bansal, N., Aljrees, T., Yadav, D. P., Singh, K. U., Kumar, A., Verma, G. K., Singh, T. (2023), “Real-Time Advanced Computational Intelligence for Deep Fake Video Detection”, Applied Science, No. 13(5), Article 3095. DOI: 10.3390/app13053095.

3. Batailler, C., Brannon, S. M., Teas, P. E., Gawronski, B. (2023), “A Signal Detection Approach to Understanding the Identification of Fake News”, Perspectives on Psychological Science, No. 17(1), P. 78–98. DOI: 10.1177/1745691620986135.

4. Reis, J. C. S., Correia, A., Murai, F., Veloso, A., Benevenuto, F. (2019), “Supervised Learning for Fake News Detection”, IEEE Intelligent Systems, No. 34(2), P. 76–81. DOI: 10.1109/MIS.2019.2899143.2.

5. Giandomenico, D. D., Sit, J., Ishizaka, A., Nunan, D. (2021), “Fake news, social media and marketing: A systematic review”, Journal of Business Research, Vol. 124, P. 329–341. DOI: 10.1016/j.jbusres.2020.11.037.

6. Yakovlev, S., Khovrat, A., Kobziev, V. (2023), “Using Parallelized Neural Networks to Detect Falsified Audio Information in Socially Oriented Systems”. Information Technology and Implementation (IT&I 2023): 10th International Conference, Kyiv, 20-21 November 2023: CEUR Workshop Proceedings. No. 3624, P. 220–238, available at: https://ceur-ws.org/Vol-3624/Paper_19.pdf (last accessed: 09.06.2024).

7. Rocha, Y. M., de Moura, G. A., Desiderio, G. A., de Oliveira, C. H., Lourenço, F. D., de Figueiredo Nicolete, L. D. (2023), “The impact of fake news on social media and its influence on health during the

COMBINATORIAL POINT CONFIGURATIONS AND THEIR PROPERTIES

Sergiy Yakovlev, Oksana Pichugina

*Institute of Mathematics, Lodz University of Technology, Lodz, Poland
National Aerospace University “Kharkiv aviation institute”, Kharkiv,
Ukraine*

Abstract. The concept of a combinatorial point configuration is introduced based on Berge's theory of combinatorial configurations and their mapping to Euclidean space. The basic properties of combinatorial point configurations are described.

In general, the term "configuration" refers to the appearance, shape and relative arrangement of objects or their component parts. In mathematics, particularly in the research domain of projective geometry, configuration typically denotes a specific arrangement of a set of points and lines on a plane or surfaces in space. Point configurations are treated as collections of points within the Euclidean space. The focus is on studying the arrangement and properties of these points in the given space.

The modern classification of such configurations was provided by B. Grunbaum [1], distinguishing their three main classes – topological, geometric, and combinatorial. Grunbaum's classification framework provides a comprehensive understanding of different types of configurations in relation to their underlying mathematical structures.

A configuration is topological if it represents an arrangement of pseudo lines within the projective plane, along with their corresponding intersection points.

Geometric configurations are those in which the lines are considered in either the Euclidean or projective plane, and the points arise from the intersections of these lines. The primary focus of research in the theory of geometric configurations revolves around two main problems. The first is the selection and establishment of the existence of different classes of configurations. The second is determining the count of non-isomorphic configurations within the respective class. These challenges drive the exploration and analysis within the field of geometric configurations. The

primary focus of research in the theory of geometric configurations revolves around two main problems. The first is the selection and establishment of the existence of different classes of configurations. The second is determining the count of non-isomorphic configurations within the respective class. These challenges drive the exploration and analysis within the field of geometric configurations. A complete review of publications on geometric configurations can be found in [2].

Combinatorial configurations predominantly focus on geometric aspects while treating points and lines as abstract sets within a combinatorial context. Currently, the study of combinatorial configurations has become a separate area in the field of discrete mathematics, the founder of which is considered to be C. Berge [3].

Combinatorial configuration, according to C. Berge, is understood as a mapping of an initial set of elements, which can be of arbitrary nature, onto a finite abstract resulting set that has a certain structure. This mapping is subject to constraints that govern the relationships and arrangement of elements. By studying these configurations, researchers explore the interactions between constraints, structure, and mappings in combinatorics.

When studying combinatorial configurations, it was usually assumed that both the initial and the resulting sets are finite. In [4], the concept of a combinatorial configuration was further developed by relaxing the requirement that the result set be finite, which made it possible to make it countable. As a result, the definition of combinatorial objects and combinatorial objects of a certain order was introduced. This made it possible to significantly expand the range of real problems that can be formulated and analyzed using such a concept.

Any combinatorial configuration can be associated with a certain point in Euclidean space. Then the set of combinatorial configurations will correspond to a finite point configuration, which we called a combinatorial point configuration. It is obvious that there are an infinite number of such mappings, and for any dimension of Euclidean space.

We will therefore require that the combinatorial point configuration have a special structure. In this regard, we will use the concept of Euclidean combinatorial sets, first introduced by Yu. Stoyan in [5]. The elements of Euclidean combinatorial sets, according to Yu. Stoyan, differ in constituent items or their order. For such sets, a certain mapping is proposed, called immersion in Euclidean space. As a result, finite point configurations are formed that have a number of specific properties. This also made it possible to use the theory of combinatorial polytopes to study various classes of finite point configurations.

The goal of the presented paper research is the study of combinatorial point configurations [6], which are formed as a result of mapping a set of combinatorial configurations into Euclidean space. The identification of combinatorial point configurations into a special class became possible thanks to an integrated approach, which, on the one hand, is based on the properties of finite point configurations of Euclidean space, and on the other, on the properties of combinatorial configurations. Therefore, in the monograph, special attention was first paid to the description of these properties.

In combinatorics, the basic combinatorial configurations are permutation and partial permutation. It is obvious that permutation and partial permutation sets are Euclidean combinatorial sets. Therefore, the paper focuses on the finite point configurations corresponding to these very sets. We identified the classes permutation point configuration, multipermutation point configuration, partial permutation point configuration, and partial multipermutation point configuration and studied their properties.

We proposed a general approach to decomposing finite point configurations into hyperplanes and partitioning them into pairwise disjoint subsets. Based on the properties of convex surfaces, a typology of superficially located sets is proposed, including classes of spherically and super spherically located sets.

In the study of convex hulls of finite point configurations, the methods of polyhedral combinatorics have been further developed. Analytical forms for specifying the corresponding combinatorial polyhedra, including multi-level ones, are considered.

Approaches to the functional-analytic representation of various classes of finite point configurations are proposed and theoretically substantiated. Classes of vertex-located and polyhedral-surface sets are singled out and their properties are investigated. Decomposition of finite point configurations into vertex-located subsets are described.

Funding: The research is funded by the IMPRESS-U program within the framework of the project entitled "Modeling and forecasting the spread of infection in war and post-war period using epidemiological, behavioral and genomic surveillance data" (2023/05/Y/ST6/00263).

References

1. Grunbaum B. (2009). Configurations of Points and Lines. University of Washington, Seattle, WA. 399 p.
2. Gropp, H. (2004): Configurations Between Geometry and Combinatorics. *Discrete Applied Mathematics* 138(1), 79–88.

3. Berge C. (1971). Principles of Combinatorics. N.Y.: Academic Press. 176 p.
4. Hulianytskyi L.F., Riasna I.I. (2017). Formalization and classification of combinatorial optimization problems. Optimization Methods and Applications (eds. Butenko S., Pardalos P. M., Shylo V.). Cham: Springer International Publishing AG: 239–250.
5. Stoyan Yu. G. (1982) On mapping of combinatorial sets into Euclidean space. *Preprint Institute of Mechanical Engineering Problems*; 173. Kharkiv, 33 p.
6. Yakovlev S., Pichugina O., Koliechkina L. (2023). Combinatorial Point Configurations and Polytopes. Lodz: Wydawnictwo Uniwersytetu Lodzkiego. 231 p.

**МИРОН ЗАРИЦЬКИЙ –
ФУНДАТОР МАТЕМАТИЧНОЇ НАУКИ
НА ЗАХІДНОУКРАЇНСЬКИХ ЗЕМЛЯХ**
Олена Полевецька, Василь Шендеровський
Інститут фізики НАНУ

**MYRON ZARYTSKY -
THE FOUNDER OF MATHEMATICAL SCIENCE
ON WESTERN UKRAINIAN LANDS**

Abstract. The theses highlight the creative path of Myron Zarytskyi, one of the founders of Ukrainian mathematical science in the western Ukrainian lands.

The main scientific works of Myron Zarytskyi included the theory of sets and the theory of functions of a real variable. He prepared the monograph "Set Theory" for publication, but it was not printed. The main topic of his work was the theory of Boolean algebras, where he solved several problems.

Поступ науки не може бути успішним без вивчення її історії, творчого доробку окремих вчених і їхньої ролі у розвитку наукового знання. Тому в пізнавальному процесі важливу роль відіграють дослідження науковців, в яких висвітлено біографічні дані вчених і їхній внесок у скарбівню світової науки і техніки.

Метою цієї розвідки є висвітлення творчого шляху одного з фундаторів української математичної науки на західноукраїнських землях, дійсного члена Наукового товариства Шевченка Мирона Зарицького.

Народився Мирон Зарицький в родині українського сільського пароха. Був первістком у сім'ї, але ріс не дуже здоровою дитиною, хоча розвивався досить швидко. Згодом батьки переїхали до Нового Села Підволочиського району. Початкову школу малий Мирон закінчив у селі Кривому в свого діда Антона Слоневського. Коли хлопцеві виповнилося десять років, він вступає до Бережанської гімназії, перші два класи якої закінчує на «відмінно».

Гімназія відіграла помітну роль в національному відродженні Галичини, згуртувала видатних педагогів, випустила у світ багато визначних діячів. Тут працювали такі відомі діячі культури, як Лучаківський, Олександр Барвінський, Насальський, Богдан Лепкий, Томашівський, Зілінський, брати Антон і Мар'ян Крушельницькі. Жваву діяльність провадив у Бережанській гімназії таємний український гурток «Молода Україна», заснований Зеноном Кузелею 1900 року.

На третьому році навчання батьки перевели Мирона до Тернопільської української гімназії. Навчався хлопець добре. Наступного року він самостійно підготувався і здав екстерном екзамен за 6-й клас та вступив до 7-го класу класичної гімназії в Перемишлі.

Саме в цій славетній гімназії Мирон склав іспити на зрілість і написав батькам таку телеграму: «Матура з відзнакою, порожній гаманець в кишені».

1907 року Мирон Зарицький вступає до Віденського університету, де студіює природознавство та філософію. Проте через брак коштів на прожиття, після закінчення першого курсу університету, батьки не відпустили хлопця до Відня, і він вступив до Львівського університету. Тут він вивчає математику, фізику, філософію. Стає активним у студентському та громадському житті.

Ще більшу любов до математики прищеплюють йому провідні на той період у Львівському університеті математики професори Юзеф Пузина та Вацлав Серпінський, який сам був учнем видатного українського математика Георгія Вороного.

1912 року Мирон Зарицький закінчує навчання у Львівському університеті. Саме цього року австрійський уряд погодився (після домагань галицьких українців) заснувати український університет не пізніше початку 1916 року, але війна перешкодила реалізації цього проекту.

Через рік Мирон Зарицький складає учительський іспит і отримує звання вчителя середніх шкіл з математики і фізики. Розраховувати на можливість викладати у вищій школі українцеві в

умовах того часу було нереально. Отож Мирон Зарицький працює учителем у приватних українських гімназіях Белза та Збаража, а згодом – у державних гімназіях Коломиї та Тернополя. Нарешті, у 1925 році, він переїздить до Львова, де спочатку працює у польській, а згодом – в Українській академічній гімназії. Слід зауважити, що Мирон Зарицький не лише викладає в цій найстарішій українській гімназії Галичини, а й веде наукову роботу.

Про наукову зрілість Мирона Зарицького свідчить обрання його 1927 року дійсним членом Наукового товариства Тараса Шевченка. Він активно працює у математично-природописно-лікарській секції. Друкує свої наукові праці у «Збірнику НТШ», а за працю «Деякі основні поняття аналізу положення з точки зору алгебри логіки», надруковану в журналі «Fundamenta Mathematicae», Львівський університет 25 жовтня 1930 року присудив йому вчений ступінь доктора філософії.

Його обирають членом Німецького математичного товариства, Львівського польського математичного товариства. Він зближується з видатними польськими математиками Штейнгаузом і Банахом.

Саме до Банаха й прихилився Мирон Зарицький у ці роки. Він перекладає працю Банаха з французької на українську. Після смерті Банаха в 1945 році Мирон Зарицький стає деканом фізико-математичного факультету, а з 1948 року керує кафедрою загальної математики.

21 квітня 1945 року йому присвоюють звання професора, а 6 липня 1946 року – вчений ступінь кандидата фізико-математичних наук.

Мирон Зарицький у цей період активно працює як науковець і як організатор математичних кафедр у Львівському та Ужгородському університетах, читає лекції з вищої математики. Він був неперевершеним педагогом. На науку дивився як на «правду і красу, що вивищує людину духовно». Ще 1927 року на ювілейному зібранні, присвяченому 30-ій річниці математично-природописно-лікарської секції НТШ, Мирон Зарицький виголосив доповідь «Правда, краса і математика».

«Кого не манить ні краса, ні мистецтво, хто живе вбогим духовним життям, той нічого не дасть математиці. Поезія не різниться від математики вищим лотом уяви, а математик різниться від поета лиш тим, що все і всюди розуміє... Але як у мистецтві, так і в математиці лише твори гарні переживають століття і виховують цілі покоління», – писав Мирон Зарицький.

Наука для Мирона Зарицького була потребою і насолодою, працею і відпочинком, як він сам часто говорив.

Учений справді найбільше любив математику, був великим знавцем історії математики, надрукував кілька праць з історії науки, зокрема «Хрестоматію грецької математики» 1936 року польською мовою. Варто сказати, що він добре знав до десятка чужинецьких мов, серед них навіть старогрецьку, захоплювався астрономією і заохочував студентів університету спостерігати зоряне небо.

Основні наукові праці Мирона Зарицького стосувалися теорії множин і теорії функцій дійсної змінної. Він навіть підготував до друку монографію «Теорія множин», але вона не була надрукована. Головною ж тематикою його творчості була теорія Булевих алгебр, де він розв'язав кілька задач. Досягнення нашого вченого на терені математики досить вагомі. Про це свідчать хоча б спогади його сучасника професора, видатного математика і педагога Миколи Чайківського. Він пише, що про значення наукових здобутків Зарицького «може свідчити хоча б той факт, що на праці українського вченого посилаються або цитують їх окремі положення французькі та німецькі математики Фреше та Гільберт, колишній вчитель нашого земляка професор Вацлав Серпінський та інші».

Цікавими є спогади вчених, яким доводилося слухати лекції Мирона Зарицького. Професор Львівського університету Михайло Сеньків у своїх спогадах пише: «Професор Зарицький читав лекції виключно українською, ігноруючи наказ адміністрації про читання лекцій. Про його педагогічну майстерність сказано багато: треба бути педагогом найвищого класу, щоб поєднувати простоту і доступність викладу з високим науковим рівнем... Характерна деталь: більше половини студентів приїхало з Польщі (Краків, Познань, Варшава) і ніколи не чули української мови. Всі вони акуратно відвідували і конспектували лекції професора Зарицького... Він навчив нас більше, ніж теореми математичного аналізу, він навчив нас гуманности, любови до людей, тому, що сам їх любив...»

Інший слухач лекцій Мирона Зарицького Лянце згадує: «Головним була та атмосфера, яка панувала на лекціях Мирона Онуфрійовича. Я маю на увазі ту глибоку, майже релігійну, пошану до великого творіння людського духу, яким є математичний аналіз. Коли Мирон Онуфрійович називав великі імена – Архімеда, Ньютона, Лейбніца, Коші, Вейерштраса, Ейлера, Кантона,

Гільберта..., лише з одного того, як він про них говорив, ми розуміли, що справжні боги не ті, що в Кремлі (а тоді там був Сталін)...».

І трохи далі: «Мову Мирона Онуфрійовича, його українську мову, вирізняла якась кришталева чистота, дивовижна милозвучність, істинність, непідробність... Своїм способом буття він ніби підкреслював: мова народу – його найцінніше надбання, надбання, яке треба берегти, примножувати. Це – скарб не тільки даного народу, але й всього людства...»

Справді, Мирон Зарицький був не тільки великим вченим, а й людиною сильного духу. Дозволимо собі висловити припущення, що самого Мирона Зарицького від ув'язнення оборонив його власний науковий авторитет...

Очевидно, все пережите не додало здоров'я Мирону Зарицькому. На схилі віку він тяжко хворів. Відійшов у вічність 19 серпня 1961 року. Його поховано на Личаківському цвинтарі у Львові.

Своїм життєвим подвигом Мирон Зарицький ще раз ствердив, що закладені у свідомість з дитячих років національно-патріотичні ідеали не можуть бути знищені жодними життєвими незгодами, а подвижницька праця на користь свого народу є сенсом життя. Він творив добро для людей, навчаючи гуманності, любові до життя, любові до людей.

Література:

1. *Пташник Б. Й.* Поет математики // Аксіоми для нащадків / Упоряд. О. Романчук. Львів: Меморіал, 1992. 544 с.

2. Енциклопедія українознавства / Гол. ред. проф. д-р В. Кубійович. Париж–Нью-Йорк: Молоде життя, 1957. Т. 2. С. 755.

3. *Гусяк Д.* Незабутня // Тернопілля-96. Тернопіль: Збруч, 1996. С. 143–145.

4. *Шендеровський В.* Нехай не гасне світ науки. У 4 т. Київ: Рада, 2003. Т. 1. С. 103–110.

РЕЗОЛЮЦІЯ
XXVI Міжнародного науково-практичного семінару
"КОМБІНАТОРНІ КОНФІГУРАЦІЇ ТА ЇХНІ
ЗАСТОСУВАННЯ"
присвяченого пам'яті професора Донця Г.П.

Учасники XXVI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їхні застосування», організованого за підтримки Міністерства освіти і науки України, Центральнотукраїнським національним технічним університетом (кафедра вищої математики та фізики), Національним університетом «Запорізька політехніка» (кафедрою системного аналізу та обчислювальної математики) за підтримки Інституту кібернетики імені В.М. Глушкова НАН України, представили 32 навчальних заклади та наукових установ України та іноземні. В семінарі брали участь представники вищих навчальних закладів та наукових установ:

1. Центральнотукраїнський національний технічний університет
2. Національний університет «Запорізька політехніка»
3. Інститут кібернетики імені В. М. Глушкова НАН України
4. Leeds University Business School, University of Leeds, UK
5. ДВНЗ «Ужгородський національний університет»
6. Дніпровський національний університет імені Олеся Гончара
7. Запорізький національний університет
8. Запорізький обласний інститут післядипломної педагогічної освіти
9. Інститут проблем машинобудування імені А.М. Підгорного НАН України
10. Інститут технічної механіки НАН України
11. Інститут фізики НАН України
12. Київський національний університет імені Тараса Шевченка
13. Льотна академія Національного авіаційного університету
14. Міжнародний Європейський університет
15. Міжнародний науково-навчальний центр інформаційних технологій та систем НАН та МОН України
16. Навчально-науковий інститут ВВП "Дніпровський гуманітарний університет"
17. Національний авіаційний університет

18. Національний аерокосмічний університет «Харківський авіаційний інститут»
19. Національний університет водного господарства та природокористування
20. Національний технічний університет «Дніпровська політехніка»
21. Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
22. Національний університет «Львівська політехніка»
23. Український державний університет науки і технологій
24. Харківський національний університет радіоелектроніки
25. Харківський національний університет міського господарства імені О.Бекетова
26. Інститут математики Лодзьського політехнічного університету, Польща
27. Інститут систем управління, Азербайджан
28. Університет Делавер, США
29. Школа комп'ютерних наук, Університет Сент-Ендрюса, Об'єднане Королівство
30. Самаркандський філіал Ташкентського університету інформаційних технологій, Узбекистан
31. Громадська організація «Системні дослідження»
32. Українське фізичне товариство

Учасники семінару розглянули найбільш актуальні питання, пов'язані із комбінаторними конфігураціями та їхніми застосуваннями до вирішення сучасних проблем моделювання та прийняття рішень в управлінні складними системами та процесами за такими напрямками:

1. Комбінаторні конфігурації, графи та задачі оптимізації на них.

2. Автомати, скінченні алгебри, математична логіка і теорія множин, теорія ігор і теорія прийняття рішень, кодування та розпізнавання образів, їх застосування в різних галузях.

3. Математичні моделі дискретних процесів і систем різного призначення.

4. Програмне забезпечення синтезу та аналізу комбінаторних конфігурацій.

5. Графи знань, онтології, машинне навчання, графові бази даних, штучний інтелект.

6. Історія розвитку української науки.

Проблематика конференції привернула до себе увагу фахівців із різних галузей знань. У науковому заході взяли участь 100

учасників, серед яких: 30 докторів наук (фізико-математичних – 15, технічних – 12, педагогічних – 2, економічних – 1); 46 кандидатів наук (фізико-математичних – 26, технічних – 17, педагогічних – 1; економічних – 1); 9 викладачів та науковців без наукового ступеня; 15 студентів і аспірантів

Учасники конференції обговорили такі проблемні питання:

- введення нових типів комбінаторних конфігурацій;
- розв'язання двоетапної задачі активації-розміщення;
- алгоритмічного підходу до навігації групою БПЛА;
- методи захисту програмного забезпечення, які ґрунтуються

на моделях математичного сейфа та граціозних дерев;

- аналіз методів розмітки графів;
- розв'язання квадратичної задачі упаковки кругів;
- вирішення задачі ізоморфізму графів за допомогою інваріантів;

• методи прийняття рішень операторами аеронавігаційної системи;

- моделювання кластерів на основі мультинечітких множин;
- моделювання випадкових графів;

• методи виявлення та нейтралізації мережі ботів у рекомендаційних системах;

- методи виявлення сфальсифікованої інформації;
- вирішення задач пошуку ідентичних структур;
- задачі побудови графів та їх груп;
- історія розвитку української науки.

Підкреслили багатомірність розглянутих задач. Усвідомлюючи роль і значення розвитку фундаментальних і прикладних досліджень в галузях науки, де широко використовуються комбінаторні конфігурації, учасники конференції вважають за необхідне:

1. Продовжити наукові дослідження за обраними напрямками, враховуючи зростаючу актуальність цієї тематики. Відзначивши високу ефективність дистанційної форми проведення XXVI Міжнародного науково-практичного семінару «Комбінаторні конфігурації та їхні застосування», рекомендувати в майбутньому поєднувати дистанційну форму роботи семінару разом з традиційною очною присутністю (за можливістю).

2. Розвивати наукові зв'язки з вітчизняними та закордонними провідними вищими навчальними закладами та науковими установами для організації і реалізації спільних наукових та освітніх проєктів.

3. Рекомендувати для проведення наступних семінарів підготовку інформаційних матеріалів українською та англійською мовою.

4. Підтримувати залучення молоді до участі в семінарі.

5. Рекомендувати кращі доповіді семінару до публікації в збірнику наукових праць «Кібернетика та комп'ютерні технології».

6. Розмістити інформацію про проведений семінар та опублікувати резолюцію конференції і збірник матеріалів на сайтах Центральноукраїнського національного технічного університету та Національного університету «Запорізька політехніка».

7. Поширювати інформацію про результати семінару у спеціалізованих виданнях.

заступник голови організаційного комітету
зав. каф. ВМ та Ф

С.М.Якименко

Матеріали
XXVI Міжнародного науково-практичного семінару
"КОМБІНАТОРНІ КОНФІГУРАЦІЇ ТА ЇХНІ
ЗАСТОСУВАННЯ"
присвяченого пам'яті професора Донця Г.П.
13-15 червня 2024 року

Редакційна колегія не несе відповідальності за матеріали,
опубліковані в збірнику. Усі вони подані в авторській редакції та
виражають персональну позицію учасників семінару

Підписано до друку 12.06.2024 р.
Формат 60x84/16. Папір офсетний.
Гарнітура Times New Roman. Друк ризограф.
Ум. друк. арк. 14.25
Наклад 300 прим. Зам. №0780

Приватне підприємство «Ексклюзив-Систем»
Свідоцтво держ. реєстру ДК№4470 від 17.01.2013р.
25006, м. Кропивницький, вул. Шевченка, 25
тел. 050-92-20913