

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ВОДНОГО ГОСПОДАРСТВА ТА ПРИРОДОКОРИСТУВАННЯ

ISSN 2522-1957



Національний університет
водного господарства
та природокористування

ВІСНИК

НАВЧАЛЬНО-НАУКОВОГО ІНСТИТУТУ АВТОМАТИКИ, КІБЕРНЕТИКИ ТА
ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ НУВГП

Збірник наукових праць



Випуск 8

Національний університет
водного господарства
та природокористування

Рівне-2021

У збірнику опубліковано наукові статті студентів і викладачів Навчально-наукового інституту автоматики, кібернетики та обчислювальної техніки НУВГП.

Редакційна колегія:

Мартинюк П.М., д.т.н., професор, головний редактор (Національний університет водного господарства та природокористування); **Бомба А.Я.**, д.т.н., професор (НУВГП); **Власюк А.П.**, д.т.н., професор (Національний університет «Острозька академія»); **Древецький В.В.**, д.т.н., професор (НУВГП); **Іващук Я.Г.**, к.ф.-м.н., доцент (НУВГП); **Круліковський Б.Б.**, к.т.н., доцент (НУВГП); **Маланчук Є.З.**, д.т.н., професор (НУВГП); **Матус С.К.**, к.т.н., доцент (НУВГП); **Степанченко О.М.**, к.т.н., доцент (НУВГП); **Сафоник А.П.**, д.т.н., професор (НУВГП); **Тадеев П.О.**, к.ф.-м.н., д.пед.н., професор (НУВГП); **Грицюк П.М.**, д.е.н., професор (НУВГП); **Турбал Ю.В.**, к.ф.-м.н., д.т.н., професор (НУВГП).



Національний університет
водного господарства
та природокористування

Матеріали збірника розглянуті і рекомендовані до видання на Вченій раді Національного університету водного господарства та природокористування
28 травня 2021 р., протокол № 6.

Адреса редколегії: 33028, м. Рівне, вул. Соборна, 11, НУВГП.
© Національний університет водного господарства та природокористування, 2021

ЗМІСТ

Соломко М. Т., Пашко С. І.	ВІЗУАЛЬНО-МАТРИЧНА ФОРМА АНАЛІТИЧНОГО МЕТОДУ ДЛЯ МІНІМІЗАЦІЇ БУЛЕВИХ ФУНКЦІЙ	7
Соломко М. Т., Горобець В. Я.	МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ БЛОК-СХЕМ ЗА ДОПОМОГОЮ ІДЕАЛЬНИХ КІЛЬЦЕВИХ В'ЯЗАНОК	13
Рощенюк А., Дідушок А.	ВАЖЛИВІСТЬ СПРИЯТЛИВИХ ВЗАЄМОДІЙ ВИКЛАДАЧА ТА СТУДЕНТА	20
Соломко М. Т., Шрамко Т. Ю.,	МІНІМІЗАЦІЯ БУЛЕВИХ ФУНКЦІЙ У ПОЛІНОМНОМУ ТА ЗМІШАНОМУ БАЗИСАХ МЕТОДОМ ОБРАЗНИХ ПЕРЕТВОРЕНЬ	24
Іванчук Н. В., Шайнюк К. С.	ПОРІВНЯННЯ ПРОТОКОЛІВ TCP ТА UDP	30
Сидор А.І., Опанасюк Е.М.,	ПІДМІНА ОБЛИЧЧЯ У ВІДЕОРОЛИКАХ “DEEPFAKE” НА ОСНОВІ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ	37
Сидор А.І., Щепковський В. В.,	ДОСЛІДЖЕННЯ ТА ВИКОРИСТАННЯ SAAS-ПЛАТФОРМ ДЛЯ WEB-РЕСУРСІВ	46
Сидор А.І., Кватерчук М. О., Жищук М. В.,	ТЕСТУВАННЯ ТА АНАЛІЗ ОСНОВНИХ ХАРАКТЕРИСТИК СИСТЕМ УПРАВЛІННЯ БАЗАМИ ДАНИХ	54
Козлюк С. М., Подобайло А. Л.	СИСТЕМА GPS-МОНІТОРИНГУ ОБ'ЄКТІВ З ФУНКЦІЄЮ ВІДДАЛЕНОГО КЕРУВАННЯ	62
Сидор А. І., Алтунін Є. А.,	АНАЛІЗ ТЕХНІЧНИХ СПОСОБІВ КЕРУВАННЯ ПРИСТРОЯМИ ПІДТРИМКИ МІКРОКЛІМАТУ В ТЕПЛИЦІ	70
Козлюк С. М., Ярута А.Г.	АНАЛІЗ РОЗДРОБЛЕНСОТІ МЕСЕНДЖЕРІВ	79
Соломко М. Т., Глінчук М. В.	СУМАТОР БІНАРНИХ КОДІВ ДЛЯ РЕАЛІЗАЦІЇ АЛГЕБРИЧНИХ ОПЕРАЦІЙ	84
Шевченко І. М., Опанасюк М. Ю.	ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ОРГАНІЗАЦІЇ КОНТРОЛЮ ЗНАТЬ ЗА ДОПОМОГОЮ ТЕСТІВ З ДИСЦИПЛІНИ	90
Соломко М. Т., Патійчук Н. Б.	СПРОЩЕННЯ БУЛЕВИХ ФУНКЦІЙ В ІМПЛІКАТИВНОМУ БАЗИСІ ЗА ДОПОМОГОЮ ОБРАЗНИХ ПЕРЕТВОРЕНЬ	96
Бойчура М. В., Абдулалі А. А.	ОДИН ПІДХІД ДО ІДЕНТИФІКАЦІЇ КОЕФІЦІЄНТА ФІЛЬТРАЦІЇ ШАРУВАТОГО СЕРЕДОВИЩА ЧИСЛОВИМИ МЕТОДАМИ КВАЗІКОНФОРМНИХ ВІДОБРАЖЕНЬ	105
Соломко М. Т., Кирикович В. О.	ОПТИМАЛЬНЕ ЧЕРГУВАННЯ ЛОГІЧНИХ ОПЕРАЦІЙ ПРИ МІНІМІЗАЦІЇ БУЛЕВИХ ФУНКЦІЙ	114

Хмельничий Т. В.	ПРОЕКТУВАННЯ ТА СТВОРЕННЯ WEB-РЕСУРСУ ДЛЯ МОНІТОРИНГУ ЯКОСТІ ОСВІТИ У ЗАКЛАДАХ ВИЩОЇ ОСВІТИ	121
Жакун А.В., Мічута О.Р.	РОЗРОБКА ЧАТ-БОТА З ВИКОРИСТАННЯМ TELEGRAM VOT API ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ НАВЧАННЯ	126
Остапчук О. П., Стрілець П. М., Бойко В. В.	ВИКОРИСТАННЯ ТРИАНГУЛЯЦІЇ ДЕЛОНЕ ПРИ РОЗРОБЦІ ІГРОВИХ ПРОГРАМНИХ ПРОДУКТІВ	131
Берник В. О., Матус О. В.	РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНІТОРИНГУ ОСНОВНИХ ХАРАКТЕРИСТИК ЛОКАЛЬНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ	140
Кондратюк К. М.	ОГЛЯД ОСНОВНИХ ПІДХОДІВ ДО МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ІНФЕКЦІЙНИХ ЗАХВОРЮВАНЬ	145
Бабич Т.Ю., Ящук Д.А,	НЕЙРО-НЕЧІТКІ МЕРЕЖІ В ПРИЙНЯТТІ РІШЕНЬ У ФІНАНСОВІЙ ДІЯЛЬНОСТІ	151
Макарець М., Мічута О.Р.	ВИКОРИСТАННЯ СУЧАСНОГО ФРЕЙМВОРКУ FLUTTER ДЛЯ СТВОРЕННЯ МОБІЛЬНИХ ДОДАТКІВ ТА ІНТЕГРАЦІЯ З БАЗОЮ ДАНИХ FIREBASE В РЕАЛЬНОМУ ЧАСІ	160
Рощенюк А., Юрєвич І.	ДИСТАНЦІЙНЕ НАВЧАННЯ У ВИЩІЙ ШКОЛІ	165



Національний університет
водного господарства
та природокористування

CONTENT

Solomko M. T., Pashko S. I.	VISUAL-MATRIX FORM OF ANALYTICAL METHOD FOR MINIMIZATION OF BOOL FUNCTIONS	7
Solomko M. T., Gorobets V. Ya	MATHEMATICAL MODELING OF BLOCK-DESIGN'S WITH THE PERFECT OF NUMERICAL BUNDLES	13
Roshcheniuk A., Didushok A.	THE IMPORTANCE OF FAVORABLE INTERACTIONS BETWEEN TEACHER AND STUDENT	20
Solomko M. T., Shramko T. Y.	MINIMIZATION OF BOOLEAN FUNCTIONS IN POLYNOMIAL AND MIXED BASES BY THE METHOD OF FIGURATIVE TRANSFORMATIONS	24
Ivanchuk N. V., Shainiuk K. S.	COMPARISON OF TCP AND UDP PROTOCOLS	30
Sydor A.I., Opanasyuk E.M.	FACE CHANGE IN "DEEPFAKE" VIDEOS BASED ON THE USE OF NEURAL NETWORKS	37
Sydor A.I., Shchepkovsky V.V.	RESEARCH AND USE OF SAAS-PLATFORMS FOR WEB- RESOURCES	46
Sydor A.I., Kvaterchuk M.O.	TESTING AND ANALYSIS OF THE BASIC CHARACTERISTICS OF DATABASE MANAGEMENT SYSTEMS	54
Kozlyuk S.M., Podobailo A.L.	GPS-MONITORING SYSTEM OF OBJECTS WITH REMOTE CONTROL FUNCTION	62
Sydor A.I., Altunin Y.A.	ANALYSIS OF TECHNICAL METHODS OF CONTROL OF MICROCLIMATE SUPPORT DEVICES IN THE GREENHOUSE	70
Kozlyuk S.M., Yaruta A.H.	ANALYSIS OF MESSAGE FRAGMENTATION	79
Solomko M. T., Glinchuk M. V.	BINARY CODE ADDER FOR REALIZATION OF ALGEBRIC OPERATIONS	84
Shevchenko I. M., Opanasyuk M.Yu.	INFORMATION SYSTEM FOR THE ORGANIZATION OF KNOWLEDGE CONTROL USING TESTS ON A SUBJECT	90
Solomko M. T., Patiichuk N. B.,	SIMPLIFICATION OF BOOL FUNCTIONS IN THE IMPLIATIVE BASIS WITH THE HELP OF IMAGE TRANSFORMATIONS	96
Boichura M. V., Abdulali A. A.,	ONE APPROACH TO IDENTIFICATION OF FILTRATION COEFFICIENT OF THE LAYERED MEDIA USING NUMERICAL QUASICONFORMAL MAPPING METHODS	105
Solomko M. T., Kyrykovych V. O.	OPTIMAL SEQUENCE OF FIGURATIVE TRANSFORMATIONS AT MINIMIZING BOOLEAN FUNCTIONS	114
Khmelnychy T. V.	DESIGN AND CREATION OF WEB-RESOURCE FOR MONITORING THE QUALITY OF EDUCATION IN HIGHER EDUCATION INSTITUTIONS	121

Michuta O.R., Zhakun A.V.	DEVELOPMENT OF A CHAT BOT USING TELEGRAM BOT API TO ASSESS THE QUALITY OF LEARNING.	126
Ostapchuk O. P., Strilets P.N., Boiko V.V.	THE USE OF DELAUNAY TRIANGULATION IN THE DEVELOPMENT OF GAMES	131
Bernik V.O., Matus O.V.	DEVELOPMENT OF SOFTWARE FOR MONITORING THE BASIC CHARACTERISTICS OF THE LOCAL AREA NETWORK	140
Kondratiuk K. M., Boiko V.V.	OVERVIEW OF MAIN APPROACHES TO MATHEMETICAL MODELING OF INFECTIOUS DISEASES	145
Babych T. Yu., Yashchuk D.A.	NEURO-FUZZY NETWORKS IN FINANCIAL ACTIVITY DECISION-MAKING	151
Мічута О.Р., Макарець М.	ВИКОРИСТАННЯ СУЧАСНОГО ФРЕЙМВОРКУ FLUTTER ДЛЯ СТВОРЕННЯ МОБІЛЬНИХ ДОДАТКІВ ТА ІНТЕГРАЦІЯ З БАЗОЮ ДАНИХ FIREBASE В РЕАЛЬНОМУ ЧАСІ	160
Roshcheniuk A., Yurevych I.	DISTANCE LEARNING IN HIGHER SCHOOL	165



Національний університет
водного господарства
та природокористування

Соломко¹ М. Т., к.т.н, доцент; Пашко С. І., студент (Національний університет водного господарства та природокористування, м. Рівне,
¹m.t.solomko@nuwm.edu.ua)

ВІЗУАЛЬНО-МАТРИЧНА ФОРМА АНАЛІТИЧНОГО МЕТОДУ ДЛЯ МІНІМІЗАЦІЇ БУЛЕВИХ ФУНКЦІЙ

Проведеними дослідженнями встановлена можливість збільшення ефективності візуально-матричної форми аналітичного методу для мінімізації булевих функцій шляхом виявлення резервів більш складнішого алгоритму проведення логічних операцій поглинання та супер-склеювання змінних у термах функцій булевого запису.

Ключові слова: мінімізація булевих функцій, образні перетворення, логічна операція поглинання та супер-склеювання змінних.

Вступ. Аналітичний метод, в основу якого покладено рівносильні перетворення за допомогою законів і тотожностей булевої алгебри, ефективний при спрощенні порівняно простих булевих функцій. Розмова йде про те, щоб перейти від ДДНФ (ДКНФ) до ДНФ (КНФ) з мінімальним числом термів. Кількість літералів у кожному термі теж має бути мінімальна. Недоліком аналітичного методу є невизначеність послідовності логічних операцій при спрощенні функцій, і, отже, відсутність алгоритму мінімізації булевих функцій. У свою чергу відсутність алгоритму не завжди гарантує, що отриманий вираз булевої функції буде мінімальним і подальше його спрощення неможливе.

Дві форми інформації, які визначаються рефлексивним та континуальним мисленням, представлено у роботі [1].

У першому випадку людина отримує інформацію словами, думає словами, та інколи перетворює їх в образи. Такий спосіб передачі інформації (вербальний) має малу інформаційну ємність, вимагає активної участі мозкових структур для розшифровки, переробки та доповнення прийнятої інформації.

При континуальній свідомості мислення відбувається не словами, а образами. Образне мислення характеризується великим надходженням інформації за одиницю часу, неспівставним з вербальним.

У роботі [2] зазначається, що «інформацію несуть не тільки усипані буквами листи книги або людська мова, але і сонячне світло, складки гірського хребта, шум водоспаду, шелест трави».

Візуально-матричні методи мінімізації булевих функцій розглядалися у працях [3–6] та ін.

Вербальна форма інформації – найбільш поширений варіант представлення опису предметної області. Будь-який алгебричний вираз – це насамперед текст, утворений за визначеними правилами. Алгебричний спосіб мінімізації булевих функцій є вербальною процедурою. До ілюстрацій звертаються у випадку коли є намагання пояснити те, що важко викласти текстом.

Ілюстративний (образний) опис наглядний, що дозволяє подати одночасно систему відношень між окремими змінними задачі. Характерною особливістю образу є його смислова ємність, здатність передавати великий об'єм інформації невеликою кількістю знаків і, як наслідок, забезпечення імпліціювання значної частини цієї інформації [7].

Таким чином, образна форма інформації у вигляді комбінаторних об'єктів має забезпечити більше шансів визначити алгоритм аналітичного методу мінімізації булевих функцій. Комбінаторними об'єктами у цьому випадку будуть двовимірні бінарні матриці та присутність у структурі таблиці істинності повних $2-(n, b)$ -design, або неповних $2-(n, x/b)$ -design систем з повторенням і суть власне комбінаторних образів [8–10]. У підсумку вербальні процедури алгебричних перетворень замінюються рівносильними образними перетвореннями [8].

Візуально-матричний спосіб мінімізації булевих функцій – це до певної міри, завершений і самостійний метод, який ґрунтується на використанні деяких властивостей візуального сприйняття інформації [3–6]. Найбільш компактною формою представлення інформації у цьому випадку є двовимірна матриця. Порядок взаємного розташування елементів матриці, однаковий при алгебричному підході, відіграє суттєву роль при візуальному сприйнятті двовимірних даних. Потенційні можливості мінімізації булевих функцій, представлених за допомогою матриці, забезпечуються властивостями періодичності та симетрії зображення [3; 5].

Хоча візуально-матричні методи стали відомі з кінця 1940 – початку 1950-х рр., візуальний метод, який ґрунтується на бінарних комбінаторних системах з повторенням $2-(n, b)$ -design, $2-(n, x/b)$ -design, як дещо окреме від періодичності та симетрії матричного зображення, продукт конкретного припущення, започаткований і розвивається з 2017 р. [8].

Еволюція аналітичного методу спрощення логічних функцій та його алгоритмізація є результатом невпинної оптимізації, тому актуальними залишаються дослідження направлені, зокрема, на внесення оновлення до алгоритму мінімізації аналітичним методом, який розглянутий у роботі [12]. Це дасть змогу забезпечити належну ефективність до раніш не врахованих варіантів спрощення булевих функцій аналітичним методом, зокрема у класі ДДНФ та ДКНФ представленні, а також оптимізувати вартість технології мінімізації булевих функцій аналітичним методом.

1. Особливості застосування логічних операцій поглинання та суперсклеювання змінних. Логічні операції зі змінними на бінарних і алгебричних структурах функцій, до певної міри, будемо виділяти кольором. Це забезпечить кращу дидактику методу.

Для аналітичного методу мінімізації булевих функцій поглинання змінних з дублюванням елементарних кон'юнкцій може мати такий вигляд:



$$\begin{aligned}
 & \left| \begin{array}{ccc|c} 1 & & & 1 \\ 2 & & & 0 \\ 3 & 1 & & \\ 4 & 1 & 0 & \\ 5 & & 0 & 1 \end{array} \right| = \left| \begin{array}{ccc|c} 1 & & & 1 \\ 1 & & & 0 \\ 1 & 1 & & \\ 1 & & 0 & 0 \\ 1 & & 0 & 1 \\ & & 0 & 0 & 1 \end{array} \right| = \\
 & = \left| \begin{array}{ccc|c} 1 & & & 1 \\ 1 & & & 0 \\ 1 & 1 & & \\ 1 & & 0 & 0 \\ & & 0 & 0 & 1 \end{array} \right| = \left| \begin{array}{ccc|c} 1 & & & 1 \\ 1 & & & 0 \\ 1 & 1 & & \\ 1 & & 0 & 0 \\ 1 & & 0 & 1 \\ & & 0 & 0 & 1 \end{array} \right| = \\
 & = \left| \begin{array}{ccc|c} 1 & & & 1 \\ 1 & & & 0 \\ 1 & 1 & & \\ 1 & & 0 & 0 \\ & & 0 & 0 & 1 \end{array} \right| = \left| \begin{array}{ccc|c} 1 & & & 1 \\ 1 & & & 0 \\ 1 & 1 & & \\ 1 & & 0 & 0 \\ & & 0 & 0 & 1 \end{array} \right| = \\
 & = \left| \begin{array}{ccc|c} 1 & & & 1 \\ 1 & & & 0 \\ 1 & 1 & & \\ & & 0 & 0 & 1 \end{array} \right|.
 \end{aligned}$$



$$\begin{aligned}
 & \overline{X_1 X_4} + \overline{X_1 X_5} + \overline{X_1 X_2} + \overline{X_1 X_3} + \overline{X_2 X_3 X_5} = \\
 & = \overline{X_1 X_4} + \overline{X_1 X_5} + \overline{X_1 X_2} + \overline{X_1 X_3 X_4} + \overline{X_1 X_3 X_4} + \overline{X_2 X_3 X_5} = \\
 & = \overline{X_1 X_4} + \overline{X_1 X_5} + \overline{X_1 X_2} + \overline{X_1 X_3 X_4} + \overline{X_2 X_3 X_5} = \\
 & = \overline{X_1 X_4} + \overline{X_1 X_5} + \overline{X_1 X_2} + \overline{X_1 X_3 X_4 X_5} + \\
 & + \overline{X_1 X_3 X_4 X_5} + \overline{X_2 X_3 X_5} = \overline{X_1 X_4} + \overline{X_1 X_5} + \overline{X_1 X_2} + \\
 & + \overline{X_1 X_3 X_4 X_5} + \overline{X_2 X_3 X_5} = \overline{X_1 X_4} + \overline{X_1 X_5} + \overline{X_1 X_2} + \\
 & + \overline{X_1 X_2 X_3 X_4 X_5} + \overline{X_1 X_2 X_3 X_4 X_5} + \overline{X_2 X_3 X_5} = \\
 & = \overline{X_1 X_4} + \overline{X_1 X_5} + \overline{X_1 X_2} + \overline{X_2 X_3 X_5}.
 \end{aligned}$$

Дублювання конституант з наступною логічною операцією простого склеювання змінних може мати такий, наприклад, вигляд

$$\left| \begin{array}{cc|c} 0 & 0 & \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & & 1 \end{array} \right| = \left| \begin{array}{ccc|c} 0 & 0 & 0 & \\ 0 & 1 & 0 & \\ 0 & 1 & 1 & \\ 1 & 0 & 0 & \\ 1 & & 1 & \end{array} \right| = \left| \begin{array}{cc|c} 0 & 1 & \\ & 0 & 0 \\ 1 & & 1 \end{array} \right|. \quad (1)$$

Отримати інший мінімальний вираз стає можливим при дублюванні конституанти для нижнього рядка першої бінарної матриці (1).

Дублювання конститuant з наступною операцією супер-склеювання змінних може мати такий, наприклад, вигляд:



Національний університет
водного господарства
та природокористування

$$\begin{array}{c}
 \left| \begin{array}{cccc} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{array} \right| = \left| \begin{array}{cccc} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{array} \right| = \\
 \\
 \left| \begin{array}{cccc} 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{array} \right|
 \end{array}$$

(2)



$$\begin{aligned}
 & x_1 x_2 x_3 x_5 x_6 + x_1 x_2 x_3 x_5 x_6 + x_1 x_2 x_3 x_4 x_5 x_6 + \\
 & + x_1 x_2 x_3 x_4 x_5 x_6 + \\
 & + x_1 x_2 x_3 x_4 x_5 x_6 + x_1 x_2 x_3 x_4 x_5 x_6 + x_1 x_2 x_3 x_4 x_5 x_6 + \\
 & + x_1 x_2 x_3 x_4 x_5 x_6 = x_1 x_2 x_3 x_5 x_6 + x_1 x_2 x_3 x_5 x_6 + \\
 & + x_1 x_2 x_3 x_4 x_5 x_6 + x_1 x_2 x_3 x_4 x_5 x_6 + x_1 x_2 x_3 x_4 x_5 x_6 + \\
 & + x_1 x_2 x_3 x_4 x_5 x_6 + x_1 x_2 x_3 x_4 x_5 x_6 + x_1 x_2 x_3 x_4 x_5 x_6 + \\
 & + x_1 x_2 x_3 x_4 x_5 x_6 + x_1 x_2 x_3 x_4 x_5 x_6 = x_1 x_2 x_3 x_5 x_6 + \\
 & + x_1 x_2 x_3 x_5 x_6 + x_1 x_2 x_3 x_4 (x_5 x_6 + x_5 x_6 + x_5 x_6 + x_5 x_6) + \\
 & + x_1 x_2 x_3 x_4 (x_5 x_6 + x_5 x_6 + x_5 x_6 + x_5 x_6) = \\
 & = x_1 x_2 x_3 x_5 x_6 + x_1 x_2 x_3 x_5 x_6 + x_1 x_2 x_3 x_4 + x_1 x_2 x_3 x_4.
 \end{aligned}$$

(3)

Результати матричного (2) та алгебричного (3) способу мінімізації логічного виразу співпадають.

Представлені порівняно складніші алгоритми застосування логічних операцій поглинання та супер-склеювання змінних розширюють варіанти їхнього використання, що дає збільшення ефективності процедури мінімізації булевих функцій аналітичним методом.

2. Результати мінімізації булевих функцій візуально-матричною формою аналітичного методу. Протокол з відносно складним алгоритмом застосування операцій поглинання та супер-склеювання змінних (п. 2), стек логічних операцій у першій матриці визначає новий, порівняно з [11], стандарт мінімізації булевих функцій. Це збільшує ефективність процедури, що дає змогу, зокрема, спростувати логічні функції з відносно більшим числом вхідних змінних ручним способом.

Приклад 1. Мінімізувати логічну функцію $F(x_1, x_2, x_3, x_4, x_5)$ аналітичним

методом, яку представимо у канонічній формі [12].

$$F = \Sigma (0, 2, 5, 6, 7, 8, 10, 16, 19, 20, 23, 27, 31). \quad (4)$$

Рішення:



Національний університет
водного господарства
та природокористування

$$F(x_1, x_2, x_3, x_4, x_5) =$$

0	0	0	0	0	0	$= \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix} =$
2	0	0	0	1	0	
5	0	0	1	0	1	
6	0	0	1	1	0	
7	0	0	1	1	1	
8	0	1	0	0	0	
10	0	1	0	1	0	
16	1	0	0	0	0	
19	1	0	0	1	1	
20	1	0	1	0	0	
23	1	0	1	1	1	
27	1	1	0	1	1	
31	1	1	1	1	1	

$$= \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{vmatrix} \quad (5)$$

У табл. 1 представлені результати мінімізації функції $F(x_1, x_2, x_3, x_4, x_5)$ (4) за допомогою Карти Карно, методом Квайна – Мак-Класкі, методом невизначених коефіцієнтів, [12] та аналітичним методом.

Таблиця 1

Результат мінімізації функції $F(x_1, x_2, x_3, x_4, x_5)$ (13)

Картою Карно, методом Квайна-Мак-Класкі, методом невизначених коефіцієнтів	Аналітичним методом
$F = \overline{x_1 x_2 x_3 x_5} + \overline{x_1 x_2 x_4 x_5} + \overline{x_1 x_3 x_4 x_5} + \overline{x_1 x_2 x_4 x_5} + \overline{x_1 x_3 x_4 x_5} + \overline{x_1 x_4 x_5}$	$F = \overline{x_1 x_3 x_5} + \overline{x_1 x_2 x_3 x_5} + \overline{x_1 x_2 x_3 x_4} + \overline{x_1 x_2 x_4 x_5} + \overline{x_1 x_4 x_5}$

Споглядаючи табл. 1, бачимо, що результатом мінімізації функції $F(x_1, x_2, x_3, x_4, x_5)$ (14) аналітичним методом є мінімальна функція, що містить п'ять мінтермів. Це на один мінтерм менше ніж у [12].

Результат мінімізації функції $F(x_1, x_2, x_3, x_4, x_5)$ (4) за допомогою програмного забезпечення Logic Friday 1.1.4 [26] є такий:

$$F = A'B'DE' + A'C'E' + ADE + AB'D'E' + A'B'CE. \quad (6)$$

Мінімальна функція (6) вміщує 11 інвертованих змінних. Це на одну інвертовану змінну більше ніж у (5).

Висновки

1. Порівняно складніші алгоритми застосування логічних операцій поглинання та супер-склеювання змінних розширюють варіанти їхнього застосування, що дозволяє збільшувати ефективність процедури мінімізації булевих функцій.

2. Візуально-матрична форма налаштовує до напівінтуїтивного підходу мінімізації булевих функцій, що ґрунтується на деяких властивостях візуального сприйняття бінарних матриць. Саме такі методи потрібні для ручної мінімізації булевих функцій [3].

3. Ефективність візуально-матричної форми аналітичного методу для мінімізації булевих функцій демонструється прикладом 1 [12] – мінімізація 5-розрядних булевих функцій;

За результатами порівняння встановлено, що ефективність візуально-матричної форми аналітичного методу для мінімізації булевих функцій дає підставу для доцільності застосування її у процедурах мінімізації логічних функцій, оскільки візуально-матрична форма аналітичного методу спроможна:

– забезпечити оперативний вибір стеку логічних операцій у першій та другій бінарних матрицях, що у підсумку дає оптимальний сценарій мінімізації логічних функцій;

– збільшити ефективність процедури мінімізації логічних функцій за рахунок реалізації відносно складних алгоритмів застосування логічних операцій поглинання та супер-склеювання змінних.

1. Налимов В. В. В поисках иных смыслов. *Сер. Библиотека журнала «Путь»*. М. : Издательская группа «Прогресс», 1993. 280 с. 2. Глушков В. М. Кибернетика. Вопросы теории и практики. М. : Изд. «Наука», 1986. 488 с. 3. Закревский А. Д. Визуально-матричный метод минимизации булевых функций. *Автоматика и телемеханика*. Томск, 1960. Вып. 3. Том 21. С. 369–373. 4. Плехль О. Электромеханическая коммутация и коммутационные аппараты. (Контактные схемы и аппараты). Введение в теорию и расчет / под ред. А. Н. Юрасова. М.; Л. : Госэнергоиздат, 1959. 288 с. 5. Свобода А. Графико-механические средства анализа и синтеза контактных цепей. *Stroje na zpracování informací : Симпозиум по машинам для обработки информации (на чешском языке)*. IV. Прага : Чехословацкая академия наук, Научно-исследовательский институт математических машин, 1956. С. 9–22. 6. Karnalgh M. The Map Method for Synthesis of Combinatorial Logic Circuits. *Trans. A TEE*. V. 72. Part I. 1953. 7. Донец С. Истоки скрытой информативности образа. *Філологічні науки*. 2015. Вип. 21. С. 107–112. 8. Різник В. В., Соломко М. Т. Мінімізація булевих функцій комбінаторним методом. *Технологічний аудит та резерви виробництва*. 2017. Вип. 4/2 (36). С. 49–64. 9. Різник В. В., Соломко М. Т. Застосування алгебричної операції супер-склеювання змінних для мінімізації булевих функцій комбінаторни методом. *Технологічний аудит та резерви виробництва*. 2017. Вип. 6/2 (38). С. 60–76. 10. Різник В. В., Соломко М. Т. Дослідження протоколів мінімізації 5-розрядних

булевих функцій комбінаторним методом. *Технологічний аудит та резерви виробництва*. 2018. Вип. 4/2 (42). С. 41–52. **11.** Різник В. В., Соломко М. Т., Тадеєв П. О., Назарук В. Д., Зубик Л. В., Волошин В. С. Алгоритм мінімізації булевих функцій методом оптимального комбінування послідовності образних перетворень. *Східно-Європейський журнал передових технологій*. 2020. № 3/4 (105). С. 43–51. **12.** Матеріал из Национальной библиотеки им. Н.Э. Баумана. Методы минимизации функций алгебры логики. 28 февраля 2016. URL: https://ru.bmstu.wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4%D1%8B_%D0%BC%D0%B8%D0%BD%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D1%8B_%D0%BB%D0%BE%D0%B3%D0%B8%D0%BA%D0%B8 (дата звернення: 25.05.2021).

Solomko M. T., Candidate of Engineering (Ph.D.), Associate Professor; Pashko S. I., Senior Student (National University of Water and Environmental Engineering, Rivne)

VISUAL-MATRIX FORM OF ANALYTICAL METHOD FOR MINIMIZATION OF BOOL FUNCTIONS

The research established the possibility of increasing the efficiency of the visual-matrix form of the analytical method of minimizing Boolean functions by identifying reserves of a more complex algorithm for logical operations of absorption and super-gluing of variables in terms of Boolean functions.

Keywords: minimization of Boolean functions, figurative transformations, logical operation of absorption and super-gluing of variables.

УДК 621.382:519

Соломко¹ М. Т., к.т.н, доцент; Горобець В. Я., студент (Національний університет водного господарства та природокористування, м. Рівне, ¹m.t.solomko@nuwm.edu.ua)

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ БЛОК-СХЕМ ЗА ДОПОМОГОЮ ІДЕАЛЬНИХ КІЛЬЦЕВИХ В'ЯЗАНОК

У статті розглянуто метод побудови комбінаторних неповних зрівноважених блок-схем за допомогою ідеальних кільцевих в'язанок. Досліджено ефективність розробленого методу та алгоритму, обґрунтовано його переваги та можливості практичного застосування.

Ключові слова: комбінаторна конфігурація, моделювання, блок-схема, кільцева в'язанка, синтез.

Вступ. Одним з перспективних методів математичного моделювання та

оптимізації системних рішень є застосування апарату комбінаторного аналізу для побудови абстрактних моделей як базової структури досліджуваної системи, яка приводить до оптимального рішення. При цьому послуговуються загальними поняттями комбінаторної конфігурації і схеми інцидентності, що лежать в основі уявлення про комбінаторні розміщення. В широкому розумінні комбінаторна конфігурація – це будь-яка система підмножин скінченної множини, елементи якої розміщені з деякою регулярністю. Сучасне уявлення цієї теорії включає вивчення класів спеціальних матриць та інваріантів, що становлять основу таблично-схемних методів комбінаторного аналізу.

Великим класом комбінаторних конфігурацій, які активно досліджуються, є блок-схеми. До них належать латинські квадрати і прямокутники, системи трійок Штейнера та інші різновиди блок-схем. Перші опубліковані результати в цій області належать Ейлеру (1782 р.). Потім з'явилися роботи Вулхауса (1844 р.), Кіркмана (1847 р.) і Штейнера (1853 р.) та інші публікації.

На початку 70-х років 20 ст. сформувався математичний апарат алгебраїчної комбінаторики на основі підходу, який розробив Дельсарт. У своїй фундаментальній праці [1] він вперше показав можливість використання схем відношень як базової структури для розгляду широкого класу комбінаторних проблем в теорії кодування, теорії блок-схем та ін. Початки цього підходу прослідковуються в роботах ряду авторів 50-х років.

1. Загальна характеристика і властивості блок-схем. Блок-схеми є одним з основних типів комбінаторних конфігурацій. Нехай $X = \{x_i\}_{i=1}^v$ – скінченна множина “точок”, а $D = \{B_j\}_{j=1}^b$ – скінченна сукупність k -елементних підмножин (k -підмножин з X), які називаються «блоками». Тоді $D \in t-(v,k,l)$ – конфігурація (коротко t -конфігурація), якщо кожна t -підмножина з X міститься рівно в l блоках сукупності D [5].

Сукупність, що складається з усіх k -підмножин множини X , є $k-(v,k,1)$ – конфігурація, що називається повною або тривіальною блок-схемою. Блок-схема, або в більш загальному випадку структура інцидентності повністю визначається через свою матрицю інцидентності $A = (a_{ij})_{v \times b}$ задану наступним чином:

$$a_{ij} = \begin{cases} 1, & \text{якщо } X_i \in B_j, \\ 0, & \text{якщо } X_i \notin B_j. \end{cases}$$

Число $|B_j|$ елементів інцидентних блоку B_j позначається через k_j , а число блоків, інцидентних елементу a_i – через r_i . Через λ_{ij} позначається число пар $\{B_j: a_i \in B_j, a_l \in B_j\}$.

Якщо блок-схема складається не з усіх можливих k -підмножин, то вона називається неповною.

Цілі числа v – загальне число елементів, b – загальна кількість блоків, r_i, k_j, λ_{ij} ($i, l=1, \dots, v; j=1, \dots, b$) називається параметрами блок-схеми. Якщо $r_i=r$ – кількість блоків, що містить один i той же елемент, для всіх $i=1, \dots, v$, $k_j=k$ – кількість елементів в одному блоці, для $j=1, \dots, b$, а $\lambda_{ij}=\lambda$ – число випадків появи у блоках будь-якої пари

неоднакових елементів, то така конфігурація є зрівноваженою неповною блок-схемою або ВІВ-схемою.

Зрівноважену неповну блок-схему в комбінаторній математиці часто називають просто блок-схемою [3].

Інколи про блок-схеми говорять як про конфігурації, зрівноважені (або збалансовані) відносно пар елементів, оскільки вивчаються також конфігурації, зрівноважені відносно трійок, четвірок елементів і т.д. [4].

Зрівноважену неповну блок-схему з параметрами b, v, r, k, l , ще називають (b, v, r, k, l) – конфігурацією.

Параметри блок-схеми задовольняють співвідношенням [6]

$$bk=vr, \tag{1}$$

$$r(k-1)=l(v-1). \tag{2}$$

Нехай серед чисел l_i ($i, l=1, \dots, v$) трапляється рівно m однакових l_1, \dots, l_m , і нехай на елементах множини V уведено m симетричних відношень, пов'язаних так, що виконуються наступні умови:

а) множина V^2 всіх пар елементів множини V розбивається на m непересічних підмножин $V^2_1, V^2_2, \dots, V^2_m$, причому, якщо $(a, a') \in V^2_j$, то говорять, що елементи a і a' пов'язані;

б) $|\{B_j: a \in B_j, a' \in B_j, (a, a') \in V^2_j\}| = l_j$;

в) $|\{a: \exists a' (a, a' \in V^2_i)\}| = n_i, i=1, \dots, m$;

г) $|\{a'': (a'', a) \in V^2_j, (a'', a') \in V^2_k, (a, a') \in V^2_i\}| = p_{jk}^i$, причому в силу симетричності

$$p_{jk}^i = p_{kj}^i \quad (i, j, k=1, \dots, m).$$

Блок-схема з властивостями а) – г) називається частково зрівноваженою блок-схемою з m типами зв'язків, або РВІВ(m)-схемою. Для параметрів РВІВ(m)-схеми справедливі рівняння (1), (2) [3] і наступні співвідношення

$$\sum_{i=1}^m n_i = v - 1;$$

$$\sum_{i=1}^m \lambda_i n_i = r(k - 1);$$

$$\sum_{k=1}^m p_{jk}^i = \begin{cases} n_j, & i \neq j, \\ n_{j-1}, & i = j, \quad i, j = 1, \dots, m; \end{cases}$$

$$n_i p_{jk}^i = n_j p_{ik}^j = n_k p_{ij}^k, \quad i, j, k = 1, \dots, m.$$

Цілочисельність параметрів накладає обмеження на існування ВІВ-схем. Так, наприклад, не існує блок-схеми з $l=1, k=3, v=5$ або 6. Дійсно, із залежності (2) при $v=6$ випливає, що $2r=5$. Якщо $v=5$, тоді $r=2$ і згідно (1) маємо $3b=10$, що свідчить про неможливість існування блок-схеми з вказаними параметрами.

Якщо у зрівноваженій неповній блок-схемі число блоків дорівнює числу елементів ($b=v$), тоді число випадків появи кожного елемента в блоках схеми дорівнює розміру блока: $r=k$, і навпаки. Блок-схема з такою властивістю називається

симетричною [7].

Нижче наведено приклад симетричної (b,v,r,k,l) – конфігурації з параметрами $b=v=7, r=k=3, l=1$:

$$B^{(1)}=(1,2,4) \quad B^{(2)}=(2,3,5) \quad B^{(3)}=(3,4,6) \quad B^{(4)}=(4,5,7);$$

$$B^{(5)}=(1,5,6) \quad B^{(6)}=(2,6,7) \quad B^{(7)}=(1,3,7).$$

Легко перевірити, що тут кожен із семи ($b=7$) блоків $B^{(1)}, B^{(2)}, \dots, B^{(7)}$ містить три ($k=3$) елементи, і кожен з цих ($v=7$) елементів $1, 2, \dots, 7$ належить рівно трьом ($r=3$) блокам, причому кожен з елементів утворює рівно одну ($l=1$) пару з кожним із усіх шести ($v-1=6$) решти елементів.

Зі співвідношення (2) та властивості симетричної блок-схеми випливає залежність

$$k(k-1)=l(v-1) \quad (3)$$

У частковому випадку, коли $l=1$, симетрична блок-схема називається скінченною проєктивною площиною $(k-1)$ -го порядку [3; 7].

Симетрична зрівноважена неповна блок-схема називається циклічною, коли для неї існує циклічний автоморфізм α , який полягає в тому, що зміна індексів $j \rightarrow j+1(\text{mod } v)$ при всіх k елементах j -го блоку блок-схеми приводить до утворення множини елементів $(j+1(\text{mod } v))$ -го блоку цієї ж блок-схеми, так що множина елементів окремого блоку повністю визначає всю блок-схему.

Повертаючись до вищезгаданого прикладу можна побачити, що наведена конфігурація є прикладом симетричної зрівноваженої блок-схеми, яка є одночасно скінченною проєктивною площиною другого ($k-1=2$) порядку і циклічною, оскільки зміна індексів $j \rightarrow j+1(\text{mod } 7)$ при всіх $k=3$ елементах j -го блоку ($j=1, 2, \dots, 7$) приводить до утворення множини елементів $(j+1(\text{mod } 7))$ -го блоку.

2. Характеристика і параметри в'язанок для моделювання блок-схем. Нехай задано множину $M_n=\{x_i\}, i=1, n$ елементів і вказана операція $*$, що здійснюється над елементами цієї множини. Тоді n -в'язанкою називається впорядкована сукупність із n елементів, на множині яких утворюється множина M_s нових елементів як результат здійснення вказаної операції між усіма впорядкованими наборами послідовно розміщених елементів даної сукупності. Отже n -в'язанка породжує множину $M=M_n \cup M_s$, базові елементи якої належать множині M_n , а новоутворені – множині M_s [2].

В'язанка, на якій множина всіх породжених нею чисел вичерпує значення, що є пропорційними елементами натурального ряду із заданою кількістю повторень кожного з елементів, називається одновимірною ідеальною числовою в'язанкою.

Велику групу складають неідеальні в'язанки. Останні за своїми комбінаторними властивостями можуть бути близькими до ідеальних, але з тих, чи інших причин не задовольняють усім вимогам, які ставлять до ідеальних в'язанок.

Якщо кожній j -й упорядкованій парі чисел (p_j, q_j) $p_j \times q_j$ – матриці, $p_j, q_j \in \{1, 2, \dots, n\}$ поставити у відповідність суму $S_j = S(p_j, q_j)$ на послідовності цілих додатніх чисел $k_n = (k_1, k_2, \dots, k_r, \dots, k_n)$, $k_r \in N_x$, де N_x – ряд натуральних чисел, у яких відповідно

$N_x = \{1, 2, \dots, x\}$,

$$S_j = S(p_j, q_j) = \sum_{i=p_j}^{q_j} k_i, p_j \leq q_j, \quad (4)$$

то упорядкована пара чисел (p_j, q_j) визначає суму $S(p_j, q_j)$ на послідовності чисел $i \in p$ по суті числовим кодом даної суми [2].

За геометричними ознаками в'язанки можна розділити на послідовності елементів, кожному з яких відповідає точка на лінії, площині або поверхні. Якщо в першому випадку елемент в'язанки можна задати одним числом, в другому – двома, то в третьому – сукупністю чисел (вектор багатовимірного простору). Такі в'язанки умовно називають одно-, дво-, та багатовимірними.

В свою чергу одновимірні в'язанки, які складаються з чисел, діляться на ідеальні і неідеальні в'язанки. До ідеальних в'язанок відносять ідеальні кільцеві в'язанки (ІКВ), ідеальні лінійки не більше третього порядку.

В'язанки в своїй сукупності можуть утворювати широку структуру внутрішніх зв'язків комбінаторних моделей.

Ідеальна кільцева в'язанка задається числом елементів n та кратністю R , яка визначає число способів утворення кільцевих сум з однаковими значеннями.

Ідеальній кільцевій в'язанці з параметрами n і R завжди відповідає неповна збалансована блок-схема з параметрами

$$\begin{aligned} v &= S_n = n(n-1)/R+1, \\ k &= n, \\ l &= R, \end{aligned}$$

яка є симетричною і циклічною.

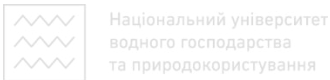
Для неідеальних в'язанок, таких як компактні кільцеві в'язанки, лінійки без розгалуження більше третього порядку, лінійки з розгалуженням відповідають частково зрівноважені блок-схеми (ЧЗ блок-схеми). Частково зрівноважена блок-схема характеризується тим, що для неї виконується лише рівняння (1). На відміну від неповних збалансованих блоків, у частково зрівноваженій блок-схемі допускається можливість появи пар елементів неоднакове число разів.

3. Кільцеві в'язанки як математичні моделі блок-схем. За результатами досліджень сучасна теорія блок-схем включає в себе багатий фактичний матеріал про величезну кількість різних типів блок-схем, який на даний момент ще недостатньо впорядкований з-за наявності численних інтерпретацій та перехресних зв'язків. Один з підходів подальшого дослідження та синтезу блок-схем полягає у використанні математичної моделі, яка б за своїми властивостями відповідала блок-схемам з різноманітними параметрами, спрощувала методику їх синтезу та відкривала можливості для знаходження нових результатів і практичних застосувань. В основу такого підходу покладено поняття числової в'язанки.

Відомо, що ідеальна кільцева в'язанка з параметрами n і R відповідає циклічній блок-схемі на множині елементів $\{b_j\} = \{j\}, j=1, 2, \dots, S_n$ [2]. Розглядаючи ІКВ як модель усієї блок-схеми, послідовності чисел $(k_1, k_2, \dots, k_j, \dots, k_n)$, що утворює ІКВ,

можна поставити у відповідність $S_n = \frac{n(n-1)}{R} + 1$ послідовностей:

$$\begin{aligned}
 B^{(1)} &= (b_1^{(1)}, b_2^{(1)}, \dots, b_1^{(1)}, \dots, b_n^{(1)}) \\
 B^{(2)} &= (b_1^{(2)}, b_2^{(2)}, \dots, b_1^{(2)}, \dots, b_n^{(2)}) \\
 &\dots\dots\dots \\
 B^{(j)} &= (b_1^{(j)}, b_2^{(j)}, \dots, b_1^{(j)}, \dots, b_n^{(j)}) \\
 &\dots\dots\dots \\
 B^{(S_n)} &= (b_1^{(S_n)}, b_2^{(S_n)}, \dots, b_1^{(S_n)}, \dots, b_n^{(S_n)})
 \end{aligned} \tag{5}$$



елементи $b^{(j)}$, $j = 1, 2, \dots, S_n$ яких визначаються за формулою [29]:

$$b_1^{(j)} \equiv j + \sum_{i=1}^l k_i \pmod{S_n}; \quad l = 1, 2, \dots, n \tag{6}$$

Оскільки кожна послідовність $B^{(j)}$ є окремим блоком зрівноваженої неповної блок-схеми (5), легко знайти $b=S_n$ і $k=n$. Крім того:

- елементи, які визначаються за формулою (6) виражаються $v=S_n$ різними числами (у даному випадку $1, 2, \dots, S_n$);
- всі елементи j -ї послідовності - різні, тобто $l \neq r \Rightarrow B_l^{(j)} \neq B_r^{(j)}$;
- кожне з чисел, якими визначаються елементи $B_l^{(j)}$, повторюється в $r=k=n$ різних послідовностях;
- будь-яка з $v(v-1)/2 = S_n(S_n-1)/2$ різних пар вказаних чисел трапляється в $l=R$ різних послідовностях.

Отже, сукупність послідовностей (5) утворює циклічну блок-схему, яка відповідає ідеальній кільцевій в'язанці з параметрами n та R .

Проілюструємо це на прикладі послідовності чисел (1,1,1,1,1,2,2,1,3), що утворює ІКВ з параметрами $n=9$, $R=6$. На основі формули (6), враховуючи, що $n=9$, $R=6$, знаходимо:

$$\begin{aligned}
 B^{(1)} &= (2, 3, 4, 5, 6, 8, 10, 11, 1), & B^{(7)} &= (8, 9, 10, 11, 12, 1, 3, 4, 7), \\
 B^{(2)} &= (3, 4, 5, 6, 7, 9, 11, 12, 2), & B^{(8)} &= (9, 10, 11, 12, 13, 2, 4, 5, 8), \\
 B^{(3)} &= (4, 5, 6, 7, 8, 10, 12, 13, 3), & B^{(9)} &= (10, 11, 12, 13, 1, 3, 5, 6, 9), \\
 B^{(4)} &= (5, 6, 7, 8, 9, 11, 13, 1, 4), & B^{(10)} &= (11, 12, 13, 1, 2, 4, 6, 7, 10), \\
 B^{(5)} &= (6, 7, 8, 9, 10, 12, 1, 2, 5), & B^{(11)} &= (12, 13, 1, 2, 3, 5, 7, 8, 11), \\
 B^{(6)} &= (7, 8, 9, 10, 11, 13, 2, 3, 6), & B^{(12)} &= (13, 1, 2, 3, 4, 6, 8, 9, 12), \\
 & & B^{(13)} &= (1, 2, 3, 4, 5, 7, 9, 10, 13).
 \end{aligned}$$

Легко перевірити, що в даному випадку побудована симетрична блок-схема з множиною елементів $\{b_i\} = \{1, 2, \dots, 13\}$ і параметрами $v=b=S_n=13$, $k=r=n=9$, $\lambda=R=6$; причому вона однозначно є циклічною блок-схемою, бо кожний її блок можна отримати з попереднього блоку при збільшенні його елементів на 1 в полі чисел за

модулем S_n . Отже, будь-якій IKB з параметрами n і R завжди відповідає блок-схема з параметрами $v=S_n=n(n-1)/R+1$, $k=n$, $\lambda=R$, яка є симетричною і циклічною. З цього випливає, що IKB можна розглядати як модель симетричної блок-схеми з циклічним автоморфізмом, який змінює числові значення елементів у кожному наступному блоці на одиницю за модулем S_n , де S_n – сума усіх чисел IKB.

З вищезгаданого випливає, що на базі симетричної блок-схеми з параметрами v, k, λ та циклічним автоморфізмом α , яка відповідає множині елементів $\{b_i\}$, $i=1,2,\dots,S_n$, завжди можна побудувати її модель у вигляді IKB, для якої $S_n=v$, $n=k$, $R=1$. Слід зазначити, що побудова такої моделі дещо ускладнюється тоді, коли порядок розміщення елементів у кожному окремому блоці циклічної блок-схеми, або й самих блоків. Таке порушення допускається, що впливає з визначення поняття «симетрична блок-схема» та «циклічна блок-схема» [6]. Однак ускладнення можна усунути, коли всі $k=n$ елементи будь-якого з цих блоків замінити числовими значеннями $1,2,\dots,v=S_n$ їх порядкових номерів (індексів) на множині $\{b_i\}$, $(i=1,2,\dots,v)$ від 1 до $v=S_n$ включно.

Після заміни елементів блоку числами, які збігаються з індексами елементів, та розміщення цих чисел у порядку зростання, елементи IKB $(k_1, k_2, \dots, k_i, \dots, k_n)$ можна визначити за формулою [2]:

$$k_i \equiv \begin{cases} b_{i+1} - b_i, & i = 1, 2, \dots, n - 1 \\ b_1 - b_n, & i = n \end{cases} \pmod{v}.$$

Отже, будь-якій симетричній блок-схемі з параметрами v, k, l та циклічним автоморфізмом α відповідає IKB, для якої $S_n=v$, $n=k$, $R=1$. Така відповідність дає змогу переносити результати, що стосуються дослідження блок-схем, на числові в'язки з відповідними параметрами. Це відкриває шлях для розширення дослідження властивостей блок-схем за допомогою числових в'язанок, які, взагалі кажучи, можуть бути й неідеальними.

Висновки

1. Для математичного моделювання комбінаторних блок-схем доцільно використовувати апарат теорії числових в'язанок, оскільки за своєю будовою та властивостями в'язанки можуть відображати властивості блок-схем з різноманітними параметрами.

2. Встановлена відповідність між характеристиками блок-схем та властивостями в'язанок як моделей цих блок-схем відкриває можливості для розширення дослідження блок-схем, спрощення методів їх синтезу, упорядкування за основними ознаками та пошуку нових результатів.

3. Для моделювання блок-схем можуть бути використані не лише ідеальні кільцеві в'язанки, але й неідеальні числові послідовності, у тому числі й в'язанки з розгалуженою структурою, що дає змогу досліджувати нові типи блок-схем з різними наборами параметрів.

1. Дельсарт Ф. Алгебраический подход к схемам отношений теории кодирования. М. : Мир, 1976. 355 с. 2. Різник В. В. Синтез оптимальних комбінаторних систем, Львів, 1989. 168 с. 3. Рыбников К. А. Введение в комбинаторный анализ. М. : Мир,

1985. 297 с. 4. Тараканов В. Е. Комбинаторные задачи и (0,1)-матрицы. М., 1985. 153 с. 5. Уилкс. Математическая статистика. М. : «Наука», 1967. 367 с. 6. Холл М. Блок-схемы, прикладная комбинаторная математика. Изд-во «Мир», 1968. 348 с. 7. Холл М. Комбинаторика. М., 1970. 356 с.

Solomko M. T., Candidate of Engineering (Ph.D.), Associate Professor; Horobets V. Y. Senior Student (National University of Water and Environmental Engineering, Rivne)



Національний університет
водного господарства
та природокористування

MATHEMATICAL MODELING OF BLOCK-DESIGN'S WITH THE PERFECT OF NUMERICAL BUNDLES

The article considers the method of constructing combinatorial incomplete balanced block diagrams with the help of ideal ring bundles. The efficiency of the developed method and algorithm is investigated, its advantages and possibilities of practical application are substantiated.

Keywords: combinatorial configuration, modeling, block diagram, ring bundle, synthesis.

УДК 378:004

Рощенюк¹ А., к.п.н., доцент, керівник студентського наукового гуртка «Інформаційні технології в середній та вищих школах»; **Дідушок² А., студентка** (Національний університет водного господарства та природокористування, м. Рівне,
¹a.m.roshcheniuk@nuwm.edu.ua, ²didushok_ak18@nuwm.edu.ua)

ВАЖЛИВІСТЬ СПРИЯТЛИВИХ ВЗАЄМОДІЙ ВИКЛАДАЧА ТА СТУДЕНТА

У статті висвітлено основні поняття комунікації, особливості спілкування студента та викладача на міжособистісному рівні, проаналізовані деякі типові стилі керівництва викладача та висвітлено важливість побудови сприятливих відносин у вищій школі.

Ключові слова: комунікація, освітній процес, стилі керівництва, взаємодія, конфлікти.

Мета статті – висвітлення основних моделей керівництва викладача, дослідження комунікативних аспектів, важливість правильної позиції викладача у колективі та причини виникнення конфліктів.

Виклад основного матеріалу. Комунікація це надважливий компонент сучасних відносин між людьми, який є чинником існування суспільства в цілому. Комунікативний процес – це взаємодія між різними суб'єктами комунікації, при якому здійснюється обмін інформацією. Цей процес включає в себе динамічну зміну етапів формування, передавання, приймання, розшифрування і використання інформації в обох напрямках при взаємодії комунікантів. Основною метою

комунікативного процесу є прийняття та зрозумілість інформації, що є предметом обміну. Проблема спілкування є однією із найактуальніших проблем у суспільстві. Це породжує наступні каталізатори, які, в свою чергу, спричиняють проблеми як, наприклад, вплив на продукт спільної діяльності деякої групи людей, що не можуть порозумітись та донести у правильному руслі інформацію. Це безпосередньо стосується і навчального процесу, адже саме там формуються наші навички комунікації, а також виявляються психічні якості. Та насамперед, у цьому місці, людина розвиває себе як особистість зі стійкими цінностями та моральними нормами.

Так основою освітнього оточення у вищій школі є міжособистісні взаємодії між двома суб'єктами: викладачем та студентом. Важливою характеристикою подальшої ефективної співпраці є формування співдружньої атмосфери, яка забезпечить та вплине на формування у майбутнього фахівця загальної системи цінностей. Не виключено, що це повинна бути основа партнерської взаємодії. Вони мають будуватись на тій основі, в першу чергу, де учасники процесу спілкування рівноправні та виявляють взаємоповагу. Це призведе до встановлення не такого знайомого «викладач-студент», а міжособистісного. Такий вид спілкування у вищій школі дозволить створити більш задовільні умови розвитку особистості студента та якнайкращого формування його кваліфікаційних навичок. Однією з основних функцій педагога є створенням умов для студентів, які підштовхують на новий рівень самоорганізації і знаходження нових індивідуальних якостей. Тобто, по суті, завдання викладача – це дати студентам певні знання, проте тільки сам студент вже вирішує, чи брати їх, чи ні.

Бутенко Н. виділила три найосновніші складові оптимального педагогічного поведіння та спілкування.

По-перше, це достатньо високий авторитет педагога, який відразу забезпечить довіру аудиторії й інтерес до тієї інформації, що він планує запропонувати. Авторитет може бути результатом тривалої роботи з аудиторією і залучення її на свій бік за допомогою різних педагогічних прийомів [1, с. 58]. Прояв критерій авторитетності педагога – їхня повага і любов до нього. Деякі педагоги міркують таким чином: люблять мене – добре, не люблять – теж нічого: пізніше зрозуміють, що це я роблю для них. Така думка помилкова. Любов вихованців до свого вихователя – не одне з наївних побажань, а могутній позитивний фактор педагогічного процесу. Усіх відомих педагогів, дуже різних людей і фахівців, об'єднує та обставина, що всіх їх дуже любили студенти, які, як правило, проносили цю любов через усе своє життя. Ця любов – могутній засіб і найважливіша умова успіху всього педагогічного процесу.

Друга умова успішності педагогічного спілкування – опанування психологією і прийомами спілкування, тобто педагог повинен бути добре підготовлений як практичний психолог.

І, нарешті, третя складова успіху – це накопичений досвід, це те, що в повсякденній практиці називається «спочатку вміння, а потім і майстерність» [4, с. 272].

Задля того, аби успішно взаємодіяти зі студентом, педагогу потрібно навчатися разом з ним. Дана методика призводить до професійного та особистого

вдосконалення, розвитком гнучкості, внутрішньої видозміни себе. Викладач має чималий вплив на студентів, коли вони відкриваються йому як особистості. Після цього він готовий до взаємодії та допомоги. Часто буває, що студент проектує на предмет вивчення власне ставлення до викладача даної дисципліни. Саме тому важливим є те, щоб викладач володів комунікативними та організаторськими вміннями, щоб привернути увагу студента до творчої діяльності та заохотити до навчання. Студент також повинен вміти йти на компроміс та проявляти ініціативність і зацікавленість зі свого боку.

Максименко С. зазначає, що в організаційно-змістовому аспекті навчально-пізнавальна діяльність – це особлива розгорнута форма спілкування, яка дає можливість впливати на особистість того, хто навчається, і на формування, зокрема, його потреб та мотивів [3, с. 2–4].

Значна та основна взаємодія учасників процесу відбувається на лекціях, практичних заняттях, лабораторних роботах, захистах курсових/кваліфікаційних робіт, екзаменах та консультаціях. Не варто забувати, що педагогічне спілкування спрямоване не тільки на саму взаємодію з метою особистісного розвитку, але й на основу педагогічної системи, а саме організацію засвоєння поданих навчальних знань і формування, на цій основі, характерних вмінь майбутнього фахівця. Викладач вищої школи, при цьому, залишається основною ланкою в системі формування особистості спеціаліста. Він здійснює вплив на студентів не лише своїм навчальним матеріалом лекцій та практичних чи лабораторних занять, але й своєю особистістю, поведінкою, манерами, ставленням до власної дисципліни, інших людей та, в першу чергу, і до самих студентів.

Дослідження, які проводились у сфері педагогічної психології, надають відповідний результат: одна з частин педагогічних труднощів зумовлена не стільки недоліками наукової і методичної підготовки викладачів, скільки вибором викладачем того чи іншого стилю керівництва.

Одними з основних стилів керівництва є:

- владний стиль керівництва – надання можливості для студентів брати участь в обговоренні питань навчального або колективного життя, але кінцеве рішення завжди приймає сам викладач;

- самовладний стиль – здійснення викладачем одноосібного управління колективом студентів, при якій не дозволяється висловлювати свої погляди і зауваження. Педагог лише послідовно ставить до студентів вимоги та здійснює жорсткий контроль за їх виконанням;

- демократичний стиль – передбачання й врахування викладачем думок студентів, він намагається та хоче зрозуміти їхню позицію та, в разі помилкового припущення, переконати;

- ігнорування – прагнення викладача якомога менше втручатися у колективне життя студентів, практично усувається від керівництва над ними, лише виконує формальні обов'язки у вигляді передачі навчальної інформації;

- непослідовний стиль – викладач, залежно від зовнішніх обставин і власного емоційного стану, здійснює будь-який з названих стилів керівництва, який часто приводить до дезорганізації та появи конфліктних ситуацій [5, с. 152–156].

Для побудови взаємовідносин зі студентами та вирішення педагогічних завдань є найбільш ефективним демократичний стиль, за допомогою якого викладач оцінює та приймає до уваги індивідуальні особливості студентів, їх досвід, їхні потреби та можливості. Викладач, який має досвід з даним стилем та завжди використовує його принципи, ставить мотивуючі завдання перед студентами, не виявляє негативу, об'єктивний в оцінках, різнобічний і ініціативний у взаємодії з колективом. Даний стиль спілкування можна охарактеризувати як особистісний. Для його вироблення людина повинна мати високий рівень професійного усвідомлення, здатна до постійного самоаналізу своєї поведінки та адекватної самооцінки.

Однією із найосновніших причин виникнення конфліктів між двома суб'єктами навчальної платформи – оцінка знань студента. У даній ситуації суб'єктом однієї сторони може виступати необ'єктивна претензія студента на більш високий бал і суб'єктивність самого викладача, який занижує оцінки колективу. Існують викладачі, які досить рідко ставлять оцінку «відмінно», або ж взагалі не ставлять, бо переконані в тому, що досконало володіють знаннями своєї дисципліни тільки вони самі. В таких випадках конфлікт невичерпний та довготривалий, тому викладач постійно знаходиться в напруженій ситуації зі студентами. Вплив на бали, може також спричинити різну поведінку студента на лекціях та практичних заняттях даного предмету, часто виникають суперечки та тотальна неповага.

Найбільш характерними також є:

- ситуації, що виникають на підставі невиконання студентом завдань;
- ситуації, що виникають в разі, коли студент не виконує та не дотримується правил поведінки, норм, які прийняті вищою школою;
- ситуації, що виникають по іншим, особистісним причинам [2, с. 127].

Учасниками більшої частини зафіксованих студентами конфліктів є викладачі. Характер таких конфліктів цілком стосується особистості педагога, а саме: його поведінки стосовно студентів (упередженість, зарозумілість, зневажливе ставлення до студентів, некомпетентність, халатне ставлення до занять, неефективне викладання). У даних випадках необхідно шукати компроміс, йти на контакт, ставати лояльнішим, підходити з розуміння до ситуації. Це є необхідними вимогами до налагодження взаємодії, чіткого обрисів можливих проблем та їхніх поетапних розв'язань.

Роблячи коротке підведення до завершення роздумів, можна обумовити, що викладач вищого навчального закладу, в першу чергу, постає перед студентами як фахівець, керівник процесу навчання, який спрямує у потрібне русло особистісні якості студента. Також його авторитет, репутація значно залежать від уміння правильно будувати діалог, аби правильно підступити та надати інформацію, тонко відчувати аудиторію, уміння самопрезентувати себе та мати навик шукати підхід до окремого студента. Він повинен володіти не тільки професійними якостями, а також вміти розбиратися в основних психологічних процесах, щоб оволодіти високим рівнем професійно-педагогічного спілкування зі студентами.

Висновок. Отже, у процесі навчальної діяльності студент виступає як носій предметно-практичної активності і пізнання. Правильна організація міжособистісної взаємодії викладача і студентів, з використанням усіх описаних та приведених методів, підвищує ефективність навчального процесу у вищому навчальному

закладі, створюючи умови для довіри студентів, їх самовизначення, розкриваючи потенціал особи. Правильна модель поведінки викладача зменшує ймовірність виникнення конфліктних ситуацій між викладачем і студентом, формує кваліфікаційні навички й етичні якості з подальшим їх застосуванням у професійній діяльності. Проте не варто відкидати те, що й не тільки викладачі, а й студенти надають вплив на процес навчання, від них залежить, який буде підхід, побудова відносин педагога до навчальної групи та до кожного студента окремо. Основне завдання викладача та студента – віднаходження оптимальної, тобто «золотої» середини, партнерських стосунків, при якій взаємодія цих двох суб'єктів буде приносити професійний розвиток для обох сторін, а також вчитися самодисципліні та взаємоповазі.

1. Бутенко Н. Ю. Комунікативні процеси у навчанні. Київ : КНЕУ, 2004. 380 с.
2. Кузьмінський А. І. Педагогіка вищої школи. Київ : Знання, 2005. 486 с.
3. Максименко С. Д. Навчальна діяльність і розвиток особистості. *Психолог*. 2004. № 21–22. С. 13–22.
4. Мальковская Т. Н. Учитель – ученик. Педагогіка и психология. *Новое в жизни, науке, технике*. Москва : Знание, 1977. № 2. С. 64–66.
5. Подоляк Л. Г. Психологія вищої школи : навч. посіб. для магістрантів і аспірантів. Київ : ТОВ «Філд студія», 2006. 320 с.

Roshcheniuk A., Candidate of Pedagogic Sciences (Ph.D.), Associate Professor, head of student scientific circle "Information technologies in secondary and high schools";
Didushok A., Senior Student (National University of Water and Environmental Engineering, Rivne)

THE IMPORTANCE OF FAVORABLE INTERACTIONS BETWEEN TEACHER AND STUDENT

The article highlights the basic concepts of communication, features of communication between student and teacher at the interpersonal level, analyzed some typical styles of teacher leadership and highlighted the importance of building favorable relationships in higher education.

Keywords: communication, educational process, leadership styles, interaction, conflicts.

УДК 519.718

Соломко¹ М. Т., к.т.н, доцент; Шрамко Т. Ю., студент (Національний університет водного господарства та природокористування, м. Рівне,

¹m.t.solomko@nuwm.edu.ua)

МІНІМІЗАЦІЯ БУЛЕВИХ ФУНКЦІЙ У ПОЛІНОМНОМУ ТА ЗМІШАНОМУ БАЗИСАХ МЕТОДОМ ОБРАЗНИХ ПЕРЕТВОРЕНЬ

У статті розглянуто мінімізацію булевих функцій у поліномному та змішаному

базисах методом образних перетворень. Проведений аналіз результатів спрощення функції у базисі Рід-Мюллера методом образних перетворень та приклаів мінімізації функції у поліномному базисі з метою порівняння складності реалізації мінімальної функції та кількості процедурних кроків.

Ключові слова: мінімізація булевих функцій, образні перетворення, функція, поліноми, процесуальні кроки, диз'юнкція, кон'юнктерми.

Вступ. Довільну булеву функцію $f(x_1, x_2, \dots, x_n)$ можна подати у поліномній нормальній формі (ПНФ) (Exclusive-OR Sum-Of-Product form – ESOP). При цьому, залежно від того, які змінні ПНФ f (усі чи деякі з них) мають або не мають знак інверсії, що визначає так звану полярність змінних, розрізняють класи AND/EXOR виразів ПНФ функцій. У загальному випадку їх називають поліномами Ріда–Маллера (Reed–Muller expressions – RM-поліноми) [1].

Розвиток мікроелектронної технології забезпечив створення елементів, які утворюють багатомісні диз'юнкції з виключенням (EXOR-елементи). Це, у свою чергу, забезпечило синтез аналогічних дворівневих AND/EXOR-схем, які містять такі ж елементи у другому каскаді. Структура цих схем описується формулами, подібними до диз'юнктивної нормальної форми (ДНФ), в яких замість операторів диз'юнкції використовуються оператори диз'юнкції з виключенням.

Переваги цих формул виправдовується тим, що кількість логічних елементів у відповідних їм схемах зазвичай менша. Наприклад, після мінімізації ДНФ довільних булевих функцій чотирьох змінних міститься в середньому 4.13 кон'юнктермів, а у ESOP – тільки 3.66 [2; 3]. При розгляді булевих функцій, типових для схем, що реалізують арифметичні операції, вигреш ще більший. Крім того, AND/EXOR-схеми легше діагностуються [3; 4].

Перспективним застосуванням при мінімізації булевих функцій є також і змішані базис.

1. Особливості застосування логічної операції супер-склеювання змінних для ПНФ булевих функцій. Для методу образних перетворень алгоритм спрощення функції з процедурою вставки двох однакових кон'юнктермів поліномної функції з наступною операцією супер-склеювання змінних може мати такий, наприклад, вигляд:

$$\begin{aligned}
 & \left| \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{array} \right| = \left| \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{array} \right| = \left| \begin{array}{ccc} 1 & & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{array} \right| = \\
 & = \overline{x_1} \overline{x_3} \oplus \overline{x_1} x_2 \overline{x_3} x_4 \oplus x_1 x_2 x_3 \overline{x_4}.
 \end{aligned} \tag{1}$$

$$\begin{aligned}
& \overline{x_1 x_2 x_3 x_4} \oplus \overline{x_1 x_2 x_3 x_4} \oplus \overline{x_1 x_2 x_3 x_4} \oplus x_1 x_2 x_3 x_4 = \\
& = \overline{x_1 x_2 x_3 x_4} \oplus \overline{x_1 x_2 x_3 x_4} \oplus \overline{x_1 x_2 x_3 x_4} \oplus \\
& \oplus \overline{x_1 x_2 x_3 x_4} \oplus x_1 x_2 x_3 x_4 \oplus x_1 x_2 x_3 x_4 = \\
& = x_1 x_3 \oplus x_1 x_2 x_3 x_4 \oplus x_1 x_2 x_3 x_4.
\end{aligned} \tag{2}$$

Результати матричного (1) та алгебричного (2) способу мінімізації логічного виразу співпадають.

Порівняно складні алгоритми спрощення логічних виразів з процедурою вставки двох однакових кон'юнктернів функцій поліномного базису з наступною операцією супер-склеювання змінних (1) розширюють варіанти їхнього застосування. Це забезпечує збільшення ефективності процедури мінімізації булевих функцій у ПНФ методом образних перетворень.

2. Сингулярні функції. Для розв'язання оптимізаційних задач логічного синтезу потрібно мати ДНФ функції та RM-поліноми з мінімальною кількістю кон'юнктернів заданої функції $f(x_1, x_2, \dots, x_n)$. При цьому, якщо існує можливість вибору RM-полінома (за винятком PPRM- і NPRM-поліномів), то у випадку однакової кількості кон'юнктернів перевага надається RM-поліному з мінімальною сумарною кількістю літералів. А коли кількість останніх однакова мінімальним RM-поліномом вважається той, що має меншу кількість інвертованих літералів. У зв'язку з цим, кошт реалізації RM-полінома заданої функції $f(x_1, x_2, \dots, x_n)$ можна оцінювати числовим співвідношенням $k_0 / k_l / k_{in}$, де k_0 , k_l , k_{in} – кількість кон'юнктернів, літералів та інверторів відповідно [1]. Аналогічну оцінку кошту реалізації мінімальної функції можна застосувати і до ДНФ функції та, у певній мірі, до логічних функцій змішаного базису.

Переходи між базисами Буля і Ріда–Маллера здійснюються за допомогою сингулярної (особливої) функції, кон'юнктерни якої попарно ортогональні.

Для перетворення ДНФ функції у поліномну нормальну форму необхідно провести ортогоналізацію заданої функції. Наприклад, ортогоналізація функції $F(x_1, x_2, x_3, x_4) = \overline{x_1 x_3} + \overline{x_1 x_2 x_4} + \overline{x_1 x_2 x_3} + \overline{x_1 x_3 x_4}$, що задана у ДНФ методом образних перетворень має такий вигляд:

$$\begin{aligned}
& F(x_1, x_2, x_3, x_4) = \\
& = \begin{vmatrix} 0 & 1 & & \\ \mathbf{0} & \mathbf{1} & & \mathbf{1} \\ 1 & 0 & 0 & \\ 1 & 0 & 0 & \end{vmatrix} = \begin{vmatrix} 0 & 1 & & \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 1 & 0 & 0 & \\ 1 & 0 & 0 & \end{vmatrix} = \begin{vmatrix} 0 & 1 & & \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} \\ 1 & 0 & 0 & \\ \mathbf{1} & & \mathbf{0} & \mathbf{0} \end{vmatrix} = \begin{vmatrix} 0 & 1 & & \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \end{vmatrix} = \begin{vmatrix} 0 & 1 & & \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & \\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \end{vmatrix}. \tag{3}
\end{aligned}$$

Кон'юнктерни функції (3) попарно ортогональні. В алгебричній формі функція (3) має вигляд (4):

$$F(x_1, x_2, x_3, x_4) = \overline{x_1 x_3} + \overline{x_1 x_2 x_3 x_4} + \overline{x_1 x_2 x_3} + \overline{x_1 x_2 x_3 x_4}. \tag{4}$$

Функції (3), (4) є сингулярними, оскільки рівносильні перетворення для них можна здійснювати обравши алгебру одного з базисів – Буля $\{\wedge, \vee, \neg\}$, або Ріда-Маллера $\{\wedge, \oplus, 1\}$.

3. Результати мінімізації булевих функцій у базисі Ріда-Маллера методом образних перетворень. Рівносильні образні перетворення при мінімізації функцій у базисі Ріда-Маллера дають наступний результат:

- визначають герменевтику логічних операцій для класу еквівалентних бінарних матриць функцій базису Ріда-Маллера;
- протокол з відносно складними алгоритмами спрощення логічних виразів, який складається з процедури вставки двох однакових кон'юнктернів функцій поліномного базису з наступною операцією супер-склеювання змінних;
- забезпечують метод ортогоналізації заданих логічних функцій для встановлення сингулярних функцій;
- створюють алгебру у частині правил рівносильного перетворення булевих функцій у базисі Ріда-Маллера.

Спростити булеву функцію $f(x_1, x_2, x_3, x_4)$ у поліномному форматі (ПНФ), що задана Картою Карно (рис. 1) [5].

	ab	00	01	11	10
cd	00	1	1	1	1
01	0	0	1	1	
11	1	1	1	1	
10	1	0	0	1	

Рисунок. Мінімізація булевої функції за допомогою Карти Карно

Рішення. Оскільки кон'юнктерни вихідної функції попарно ортогональні (функція сингулярна), для спрощення заданої функції обираємо алгебру Ріда-Маллера. Мінімізація $f(x_1, x_2, x_3, x_4)$ (рис. 1) у поліномному форматі здійснюється наступними образними перетвореннями:



$$f = \begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 3 & 0 & 0 & 1 & 1 \\ 4 & 0 & 1 & 0 & 0 \\ 7 & 0 & 1 & 1 & 1 \\ 8 & 1 & 0 & 0 & 0 \\ 9 & 1 & 0 & 0 & 1 \\ 10 & 1 & 0 & 1 & 0 \\ 11 & 1 & 0 & 1 & 1 \\ 12 & 1 & 1 & 0 & 0 \\ 13 & 1 & 1 & 0 & 1 \\ 15 & 1 & 1 & 1 & 1 \end{array} = \begin{array}{c|cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{array} = \begin{array}{c|cccc} 0 & & & \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & & & \\ 1 & 1 & 1 & 0 \end{array} = 1 \oplus \begin{array}{c|cccc} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{array} = 1 \oplus \begin{array}{c|ccc} 0 & 0 & 1 \\ 1 & 1 & 0 \end{array} = 1 \oplus \overline{x_1 x_3 x_4} \oplus x_2 x_3 \overline{x_4} = \overline{\overline{x_1 x_3 x_4}} \oplus x_2 x_3 \overline{x_4} = (x_1 + x_3 + \overline{x_4}) \oplus x_2 x_3 \overline{x_4}.$$

МПНФ функції $f(x_1, x_2, x_3, x_4)$ (рисунок):

$$f_{\text{МПНФ}} = 1 \oplus \overline{x_1 x_3 x_4} \oplus x_2 x_3 \overline{x_4}. \quad (5)$$

Результатом спрощення функції $f(x_1, x_2, x_3, x_4)$ (рис. 1) методом образних перетворень є мінімальна функція (5), що містить шість літералів. Це на чотири літерали менше, порівняно з [5]. Ціна реалізації (5) $k_0 / k_l / k_{in} = 3 / 6 / 3$.

Для мінімізації функції (рис. 1) було застосовано алгоритм спрощення функції з процедурою вставки двох однакових кон'юнктерів ПНФ з наступною операцією супер-склеювання змінних (п. 2).

Спрощена функція $f(x_1, x_2, x_3, x_4)$ (рис. 1) у змішаному базисі:

$$f_{\text{МЗБ}} = (x_1 + x_3 + \overline{x_4}) \oplus x_2 x_3 \overline{x_4}. \quad (6)$$

Ціна реалізації (6) $k_0 / k_l / k_{in} = 2 / 6 / 2$.

Обидві функції (5) і (6) представляють 3-рівневу логіку, однак у змішаному базисі мінімальна функція (6) має кращі показники реалізації.

Висновки

1. Досконалу нормальну форму функцій поліномного базису можна подати

еквівалентними бінарними наборами або еквівалентною бінарною матрицею, яка у цьому випадку буде подавати кон'юнктерми поліномних функцій та операцію додавання за модулем два для них. Таку герменевтику доцільно застосовувати при спрощенні логічних функцій та при виведенні результату логічних операцій у класі бінарних матриць функцій у базисі Ріда–Маллера.

2. Порівняно складні алгоритми спрощення логічних виразів з процедурою вставки двох однакових кон'юнктермів поліномних функцій з наступною операцією супер-склеювання змінних розширюють варіанти їхнього застосування, що дає збільшення ефективності процедури мінімізації булевих функцій у ПНФ методом образних перетворень.

3. Для належного спрощення функцій у базисі Ріда–Маллера методом образних перетворень була доопрацьована алгебра Ріда–Маллера у частині класичних правил рівносильного перетворення ПНФ та ДПНФ булевих функцій поліномного базису. Створення алгебри поліномного базису у частині зазначених правил у значній мірі вирішує проблему мінімізації функцій у базисі Ріда–Маллера.

4. Ефективність методу образних перетворень для мінімізації булевих функцій у базисі Ріда–Маллера демонструється приклад [20] – мінімізація 4-розрядної булевої функції;

За результатами порівняння встановлено, що ефективність методу образних перетворень для мінімізації булевих функцій у базисі Ріда–Маллера дає підставу для доцільності застосування його у процедурах мінімізації логічних функцій, оскільки метод образних перетворень спроможний:

– забезпечити оперативний вибір стеку логічних операцій у першій бінарній матриці, що у підсумку дає оптимальний сценарій мінімізації логічних функцій у базисі Ріда–Маллера;

– збільшити ефективність процедури мінімізації логічних функцій у базисі Ріда–Маллера за рахунок реалізації відносно складних алгоритмів спрощення логічних виразів, які складаються з процедури вставки двох однакових кон'юнктермів функцій ПНФ з наступною операцією супер-склеювання змінних.

1. Рицар Б. Є. Числова теоретико-множинна інтерпретація полінома Жеґалкіна. *Управління системами і машинами*. 2013. № 1. С. 11–19.
2. Sasao T. Easily testable realizations for generalized Reed-Muller expressions. *IEEE Trans. On Comp.*, June 1997. 46. N 6. P. 709–716.
3. Fujiwara H. Logic testing and design for testability. *Comp. Syst. Series*. Cambridge, MA : Mass. Inst. Tech., 1986. DOI: 10.7551/mitpress/4317.001.0001.
4. Faraj Khalid. Design Error Detection and Correction System based on Reed-Muller Matrix for Memory Protection. *Inter. J. of Computer Applications (0975-8887)* (November 2011). Vol. 34, no. 8. P. 42–55. DOI: 10.5120/4123-5929.
5. Акинина Ю. С. Подвальный С. Л. Тюрин С. В. Применение Карт Карно для полиномиального преобразования булевых функций. *Информатика, вычислительная техника и управление. Вестник Воронежского государственного технического университета*. 2016.

Solomko M. T., Candidate of Engineering (Ph.D.), Associate Professor; Shramko T. Y., Senior Student (National University of Water and Environmental Engineering, Rivne)

MINIMIZATION OF BOOLEAN FUNCTIONS IN POLYNOMIAL AND MIXED BASES BY THE METHOD OF FIGURATIVE TRANSFORMATIONS

The minimization of Boolean functions in polynomial and mixed bases by the method of figurative transformations is considered. The analysis of the result of simplification of the function in the Reed – Muller basis by the method of figurative transformations and the example of minimization of the function in the polynomial basis in order to compare the cost of realization of the minimum function and the number of procedural steps.

Keywords: minimization of Boolean functions, figurative transformations, function, polynomials, procedural steps, disjunction, conjuncturms.

УДК 004.04

Іванчук Н. В., к.т.н., доцент; Шайнюк К. С., студент (Національний університет водного господарства та природокористування, м. Рівне, n.v.medvid@nuwm.edu.ua)

ПОРІВНЯННЯ ПРОТОКОЛІВ TCP ТА UDP

У статті розглянуто одні з найпопулярніших протоколів передачі даних – TCP та UDP. Було описано характеристики, розглянуто плюси й мінуси даних протоколів. Розглянуто порядок роботи хедерів, питання безпеки. Частково описано транспортний рівень моделі OSI та особливості моделі, що орієнтована на зв'язок в теле- та мережевих комунікаціях.

Ключові слова: TCP, UDP, Transmission Control Protocol, User Datagram Protocol, web-протоколи, заголовки протоколів.

Сучасний світ немислимий без засобів зв'язку. Десятки мільйонів пристроїв по всьому світу зв'язуються за допомогою комп'ютерних мереж. Кожна комп'ютерна мережа організована за певними стандартами. Будь-які пристрої взаємодіють за загальноприйнятою моделлю OSI, або Базовою Еталонною Моделлю Взаємодії Відкритих Систем. Дана модель визначає взаємодію різних мережевих пристроїв на семи рівнях – Media (до них відносяться фізичний, каналний і мережевий) і Host – (транспортний, сеансовий, подання та прикладної). У даній статті ми розглянемо два найбільш поширених мережевих протоколи транспортного рівня – TCP і UDP, приклади їх застосування, а також порівняємо їх характеристики.

Transmission Control Protocol, TCP (Протокол керування передачею) разом із протоколом IP утворюють кістяк мережі Інтернет, який дав назву моделі TCP/IP. Протокол призначений для управління передачею даних у комп'ютерних мережах, працює на транспортному рівні моделі OSI [1].

Транспортний рівень (4 рівень) – це концептуальний розділ методів у багат шаровій архітектурі протоколів мережевого стеку в наборі протоколів Інтернету та моделі OSI. Протоколи цього рівня надають послуги зв'язку від хоста до хоста для додатків. Орієнтовані на модель “Connection-oriented communication”, надійність, контроль потоку та мультиплексування.

Connection-oriented communication (Орієнтована на зв'язок комунікація) – режим мережевого зв'язку в телекомунікаціях та комп'ютерних мережах, коли сеанс зв'язку або напів-постійне з'єднання встановлюється перед передачею будь-яких корисних даних, що забезпечує можливість доставки даних у правильному порядку у верхній комунікаційний шар мережі. Альтернативою орієнтованій на зв'язок передачі є беззв'язкова комунікація, наприклад, зв'язок у режимі дейтаграми, що використовується протоколами IP та UDP, де дані можуть бути доставлені не в порядку відправлення, оскільки різні мережеві пакети маршрутизуються незалежно та можуть доставлятися за різними шляхами [1].

Деталі реалізації та семантика транспортного рівня набору протоколів Інтернету, який є основою Інтернету, та модель OSI загальної мережі є різними. Всі протоколи, що використовуються сьогодні на цьому рівні для Інтернету, виникли в процесі розробки TCP/IP. У моделі OSI транспортний рівень часто називають рівнем 4 або L4, тоді як нумеровані шари не використовуються в TCP/IP.

TCP-з'єднання встановлюється за допомогою тристороннього рукостискання. Це процес встановлення та підтвердження зв'язку. Після встановлення з'єднання розпочинається передача даних, а коли процес передачі закінчується, з'єднання розривається закриттям встановленої віртуальної схеми.

На відміну від UDP, TCP забезпечує надійний обмін даними між хостом-відправником та хостом-отримувачем. Для цього встановлюється логічний зв'язок між двома хостами.

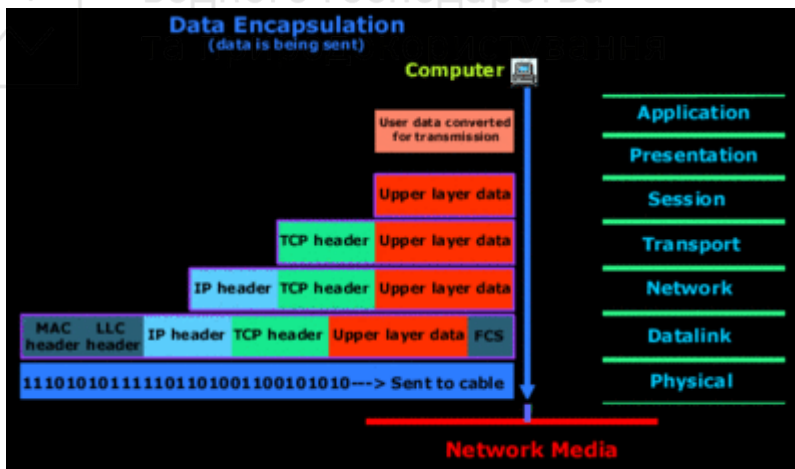


Рис. 1. Інкапсуляція TCP в моделі OSI

TCP отримує дані від протоколів верхніх рівнів OSI-моделі, початковим джерелом яких є протоколи прикладного рівня, такі як HTTP, DNS, IMAP, VoIP, FTP та інші. Кожний протокол верхнього рівня має свій чітко визначений TCP-порт.

TCP розбиває конкретний потік на частини, додає до кожної з них заголовок з номером послідовності, що є числом, яке відображає номер першого байту в сегменті надісланих даних від хоста-відправника до хоста-отримувача. Поточний номер послідовності являє собою суму номера послідовності попереднього сегменту і кількості даних, що в ньому були відправлені. Використовується для контролю кількості та правильності послідовно-отриманих сегментів даних. Отримані порції даних називають TCP-сегментами. Кожен сегмент інкапсулюється в пакет та передається через IP-протокол до хоста-отримувача.

Після надходження пакету до хоста-отримувача перевіряється коректність отриманих даних у TCP-сегменті, методом перерахування контрольної суми та переконується, що попередні сегменти даних також були успішно отримані. Після чого хост-отримувач надсилає запит до хоста-відправника про нову або повторну передачу порції даних, що одночасно є підтвердженням того, що всі сегменти з номерами послідовності були успішно отримані.

У свою чергу TCP-сегменти деінкапсулюються з IP-пакетів, розміщуються в правильному порядку та з них вилучаються TCP-заголовки. Отриманий таким чином потік передається до того протоколу верхнього рівня, з якого було відправлено дані на стороні хоста-відправника.

Згідно [1] важливі особливості TCP:

- підтвердження отримання даних
- повторне встановлення з'єднання в разі його розриву
- розподілення передач відповідно до завантаження мережі
- легке виявлення помилок

User Datagram Protocol, UDP (*Протокол датаграм користувача*) є одним з основних членів набору протоколів Інтернету. За допомогою UDP комп'ютерні програми можуть надсилати повідомлення, в даному випадку іменовані дейтаграмами, іншим хостам мережі Інтернет-протоколу (IP). Попередні зв'язки не потрібні для встановлення каналів зв'язку або шляхів передачі даних [2].

Від TCP відрізняється тим, що працює без встановлення з'єднання. UDP – це один з найпростіших протоколів транспортного рівня (4 рівень) моделі OSI, який виконує обмін повідомленнями без підтвердження та гарантії доставки. При використанні протоколу UDP відповідальність за обробку помилок, повторну передачу даних та спроби відновлення з'єднання покладена на протокол рівнем вище. Але попри всі недоліки, протокол UDP є ефективним для серверів, що надсилають невеликі відповіді великій кількості клієнтів.

UDP використовує просту модель зв'язку без підключення з мінімумом механізмів. UDP забезпечує контрольні суми для цілісності даних та номери портів для адресації різних функцій у джерела та місця призначення дейтаграми. Він не має діалогів про рукошукання, і таким чином не забезпечує безпеку потоку даних; немає гарантії доставки, замовлення або копіювання захисту. Якщо на рівні мережевого інтерфейсу потрібні засоби виправлення помилок, програма може використовувати Протокол управління передачею (TCP) або Протокол передачі потокового управління (SCTP), які призначені для цієї мети.

Порівняння хедерів TCP та UDP

		UDP datagram header																															
Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

Рис. 2. Таблиця заголовка UDP

Дейтаграма UDP [1] складається із заголовка дейтаграми та розділу даних. Заголовок дейтаграми UDP складається з 4 полів, кожне з яких становить 2 байти (16 бітів). Розділ даних слідує за заголовком і являє собою дані корисного навантаження, що передаються для програми.

Використання контрольної суми та порту вихідного порту необов'язково в IPv4 (рожевий фон у таблиці). У IPv6 необов'язково лише поле вихідного порту.

Номер вихідного порту:

Це поле визначає порт відправника, коли він використовується, і вважається портом, на який потрібно відповісти, якщо це необхідно. Якщо він не використовується, він повинен бути нульовим. Якщо вихідним хостом є клієнт, номер порту, швидше за все, буде ефемерним номером порту. Якщо вихідним хостом є сервер, номер порту, швидше за все, буде добре відомим номером порту.

Номер порту призначення:

Це поле визначає порт приймача та є обов'язковим. Подібно до номера вихідного порту, якщо клієнт є хостом призначення, тоді номер порту, швидше за все, буде ефемерним номером порту, а якщо кінцевим хостом є сервер, то номер порту, швидше за все, буде добре відомим номером порту.

Довжина:

Це поле визначає довжину в байтах заголовка UDP та даних UDP. Мінімальна довжина – 8 байт, довжина заголовка. Розмір поля встановлює теоретичну межу 65 535 байт (8 байтний заголовок + 65 527 байт даних) для датаграми UDP. Однак фактичне обмеження довжини даних, яке накладається базовим протоколом IPv4, становить 65 508 байт ($2^{16} - 8$ – 8-байтний заголовок UDP – 20-байтний IP-заголовок).

Використовуючи джамбограми IPv6, можна мати дейтаграми UDP розміром більше 65 535 байт. RFC 2675 визначає, що поле довжини встановлюється рівним нулю, якщо довжина заголовка UDP та даних UDP перевищує 65 535.

Контрольна сума:

Поле контрольної суми може використовуватися для перевірки помилок заголовка та даних. Це поле є необов'язковим для IPv4 та обов'язковим для IPv6. Поле містить усі нулі, якщо воно не використовується.

Offsets		TCP segment header																															
Octet	Bit	0								1								2								3							
0	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	Source port								Destination port																							
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset	Reserved 000	NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size																				
16	128	Checksum								Urgent pointer (if URG set)																							
20	160	Options (if data offset > 5. Padded at the end with "0" bytes if necessary.)																															
60	480																																

Рис. 3. Таблиця заголовків TCP

Порт джерела (16 біт):

Ідентифікує порт відправлення.

Порт призначення (16 біт):

Ідентифікує порт прийому.

Порядковий номер (32 біти):

Має подвійну роль:

Якщо встановлений прапор SYN (1), це початковий порядковий номер. Тоді порядковий номер фактичного першого байта даних та підтверджений номер у відповідному ACK є цим порядковим номером плюс 1.

Якщо прапор SYN чіткий (0), то це накопичений порядковий номер першого байта даних цього сегмента для поточного сеансу.

Номер підтвердження (32 біти):

Якщо встановлено прапор ACK, значення цього поля є наступним порядковим номером, який очікує відправник ACK. Це підтверджує отримання всіх попередніх байтів (якщо вони є). Перший ACK, надісланий кожним кінцем, підтверджує сам початковий порядковий номер іншого кінця, але не має даних.

Зміщення даних (4 біти):

Вказує розмір заголовка TCP у 32-розрядних словах. Мінімальний розмір заголовка становить 5 слів, а максимальний – 15 слів, таким чином, отримуючи мінімальний розмір 20 байт і максимум 60 байт, що дозволяє до 40 байт опцій у заголовку. Це поле отримало свою назву завдяки тому, що воно також є зміщенням від початку сегмента TCP до фактичних даних.

Зарезервовано (3 біти)

Для подальшого використання і повинен бути встановлений на нуль [2].

Прапори (9 біт):

Містить 9 1-бітових прапорів (контрольних бітів) наступним чином:

NS (1 біт): ECN-nonce – захист від приховування

CWR (1 біт): Прапор зменшеного вікна перевантажень (CWR) встановлюється відправляючим хостом, щоб вказати, що він отримав сегмент TCP із встановленим прапором ECE і відповів у механізмі контролю заторів.

ECE (1 біт): ECN-Echo виконує подвійну роль, залежно від значення прапора SYN. Це вказує:

Якщо встановлено прапорець SYN, то рівноправний TCP підтримує ECN.

Якщо прапор SYN чіткий (0), пакет із встановленим прапором перевантаження (ECN = 11) у заголовку IP отримано під час звичайної передачі. Це служить вказівкою на перевантаження мережі (або наближення перевантаження) до відправника TCP.

URG (1 біт): вказує на те, що поле термінового покажчика є значущим

ACK (1 біт): вказує на те, що поле підтвердження є значущим. Для всіх пакетів після початкового пакету SYN, надісланого клієнтом, слід встановити цей прапор.

PSH (1 біт): функція натискання. Просить надіслати буферизовані дані до програми, що отримує.

RST (1 біт): скидання з'єднання

SYN (1 біт): Синхронізуйте порядкові номери. Тільки перший пакет, надісланий з кожного кінця, повинен мати цей прапор. Деякі інші прапори та поля змінюють значення на основі цього прапора, а деякі діють лише тоді, коли він встановлений, а інші, коли він чіткий.

FIN (1 біт): Останній пакет від відправника

Розмір вікна (16 біт):

Розмір вікна прийому, який визначає кількість одиниць розміру вікна, які відправник цього сегменту бажає отримати в даний момент.

Контрольна сума (16 біт):

16-бітове поле контрольної суми використовується для перевірки помилок заголовка TCP, корисного навантаження та псевдо заголовка IP. Псевдозаголовок складається з вихідної IP-адреси, IP-адреси призначення, номера протоколу протоколу TCP та довжини заголовків і корисного навантаження TCP (у байтах) [2].

Порівняння TCP та UDP

TCP	UDP
Орієнтована на зв'язок комунікація	Протокол без конектів
TCP зчитує дані як потоки байтів і повідомлення передається на межі сегментів	Повідомлення UDP містять пакети, які були надіслані по одному. Він також перевіряє цілісність під час отримання потоку
Повідомлення TCP прокладаються через Інтернет з одного комп'ютера на інший	Не базується на підключенні, тому одна програма може надсилати безліч пакетів іншій
TCP переставляє пакети даних у визначеному порядку	Протокол UDP не має фіксованого порядку, оскільки всі пакети не залежать один від одного
Швидкість TCP є повільнішою.	UDP працює швидше, оскільки не намагається відновити помилки.

Розмір заголовку – 20 байт	Розмір заголовку – 8 байт
TCP потребує трьох пакетів для встановлення підключення до сокета перед тим, як можна надіслати будь-які дані користувача	Немає з'єднань для відстеження, упорядкування повідомлень
TCP виконує перевірку помилок, а також здійснює відновлення помилок	UDP виконує перевірку помилок, але відкидає помилкові пакети
Використання протоколів рукописання SYN, SYN-ACK, ACK	Відсутні протоколи рукописання
TCP надійний, оскільки гарантує доставку даних до маршрутизатора призначення	Доставка даних до пункту призначення не може бути гарантована
TCP пропонує широкі механізми перевірки помилок, оскільки забезпечує контроль потоку та підтвердження даних	UDP має лише один механізм перевірки помилок, який використовується для контрольних сум

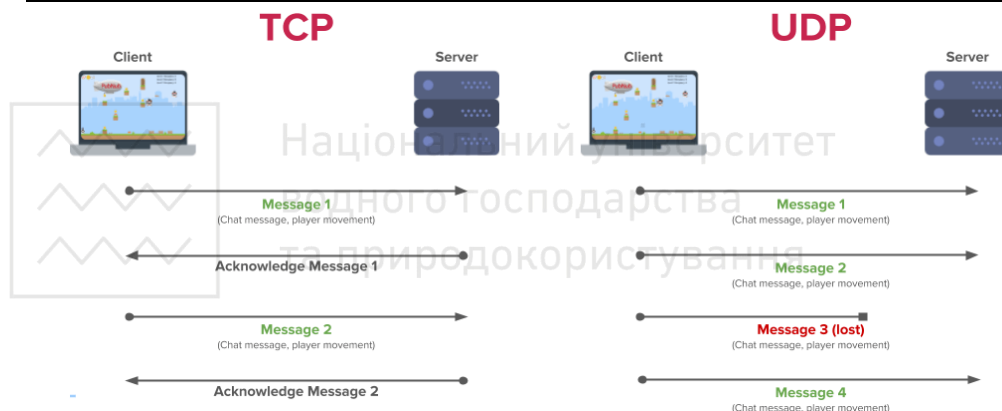


Рис. 4. Порівняння зв'язків TCP та UDP

1. Сідні Фейт. TCP/IP. Архітектура. Протоколи. Реалізація. Лори, 2016. 424 с.
2. Оліфер В., Оліфер Н. Комп'ютерні мережі. Принципи, технології, протоколи. Пітер, 2016. 993 с.

Ivanchuk N. V., Candidate of Engineering (Ph.D.), Associate Professor; Shainiuk K. S., Senior Student (National University of Water and Environmental Engineering, Rivne)

COMPARISON OF TCP AND UDP PROTOCOLS

The article considers some of the most popular data transfer protocols – TCP and UDP.

The characteristics were described, the pros and cons of these protocols were considered. The order of work of headers, questions of safety are considered. The transport level of OSI models and features of models that are focused on TV and network communications are partially described.

Keywords: TCP, UDP, Transmission Control Protocol, User Datagram Protocol, web protocols, protocol headers.

УДК 004.8

Національний університет
водного господарства
та природокористування

Сидор А. І., к.т.н., старший викладач; Опанасюк Е. М., студент (Національний університет водного господарства та природокористування, м. Рівне)

ПІДМІНА ОБЛИЧЧЯ У ВІДЕОРОЛИКАХ “DEEPFAKE” НА ОСНОВІ ВИКОРИСТАННЯ НЕЙРОННИХ МЕРЕЖ

Робота присвячена дослідженню нейронних мереж, а саме створенню Deepfake відеороликів на основі нейронних мереж. В роботі буде продемонстрований алгоритм нейрованої мережі та кінцевий результат у вигляді фейкового відеоролика.

Ключові слова: Deepfake, нейронна мережа (НМ), штучна нейронна мережа (ШНМ), електронно обчислювальна машина (ЕОМ), центральний процесор (ЦП), велика інтегральна схема (ВІС).

Вступ. В останнє десятиліття у світі бурхливо розвивається нова прикладна область математики, що спеціалізується на штучних нейронних мережах, які являють собою системи, що здатні до самонавчання та імітують роботу головного мозку людини.

Актуальність досліджень в цьому напрямку зумовлюється великою кількістю різноманітних можливостей застосування НМ. Це автоматизація процесів розпізнавання образів, адаптивне керування, прогнозування, створення експертних систем та ін.

За своєю природою нейронні мережі – це системи, які за своєю природою являються паралельними. Проте, на жаль, на сьогоднішній день для досліджень використовуються послідовні алгоритми їх навчання та роботи. А це вже зводить нанівець всі переваги НМ.

Введення в теорію штучних нейронних мереж. Сьогодні є безперечним значний науковий та практичний інтерес до обчислювальних структур нового типу - штучних нейронних мереж (ШНМ). Він зумовлений насамперед тим, їх що технологія дозволила розробити ефективні підходи до вирішення проблем, що вважалися складними для реалізації в традиційних комп'ютерах.

Штучні нейронні мережі – математичні моделі, а також їхня програмна та апаратна реалізація, побудовані за принципом функціонування біологічних нейронних мереж - мереж нервових клітин живого організму. Системи, архітектура і принцип дії базується на аналогії з мозком живих істот. Ключовим елементом цих

систем виступає штучний нейрон як імітаційна модель нервової клітини мозку – біологічного нейрона.

Аналогія мозку та комп'ютерного процесору. Мозок і Центральний Процесор (ЦП) ЕОМ, є найбільш складними системами переробки інформації. ЦП інтегрує мільйони напівпровідникових транзисторів і подібний мозку, в якому міститься близько 100 мільярдів нейронів, кожен з яких має в середньому 10 000 зв'язків. За аналогією зі схмотехнікою великих інтегральних схем (ВІС) ЕОМ, де відбувається прийом, передача і переробка цифрової інформації, біологічна мережа нейронів відповідає за всі явища, які ми називаємо думками, емоціями, пам'яттю, сенсомоторними та автономними функціями. Мозок, в якому, на відміну від ЦП, щодня гине велика кількість нейронів, надзвичайно надійний і продовжує функціонувати. Обробка величезних обсягів інформації мозком здійснюється дуже швидко, за доли секунди, незважаючи на те, що сам нейрон є повільно діючим елементом з часом реакції у декілька мілісекунд. На відміну від біологічного нейрона, електричні напівпровідники ЦП є більш швидкодіючими і безпомилковими, проте кількість зв'язків між нейронами мозку у багато разів перевищує зв'язки між напівпровідниками ЦП, що забезпечує паралелізм і багатопоточність обробки інформації.

Штучний нейрон. Основними компонентами нейронної мережі є нейрони /neurons/ (елементи, вузли), які з'єднані зв'язками. Сигнали передаються по зваженим зв'язкам (connection), з кожним з яких пов'язаний ваговий коефіцієнт (weighting coefficient) або вага.

Штучний нейрон (Математичний нейрон Маккалока – Піттса, Формальний нейрон – вузол штучної нейронної мережі, що є спрощеною моделлю природного нейрона. Математично, штучний нейрон зазвичай представляють як деяку нелінійну функцію від єдиного аргументу - лінійної комбінації всіх вхідних сигналів. Цю функцію називають функцією активації або функцією спрацьовування, передавальною функцією.

Отриманий результат посилається на єдиний вихід.

Штучний нейрон імітує в першому наближенні властивості біологічного нейрона. На вхід штучного нейрона поступає множина сигналів, які є виходами інших нейронів. Кожен вхід множиться на відповідну вагу, аналогічну його синаптичній силі, і всі виходи підсумовуються, визначаючи рівень активації нейрона.

Кожен нейрон (рис. 1) характеризується своїм поточним станом за аналогією з нервовими клітинами головного мозку, які можуть бути порушені або загальмовані. Він володіє групою синапсів - односпрямованих вхідних зв'язків, з'єднаних з виходами інших нейронів, а також має аксон – вихідну зв'язок даного нейрона, з якої сигнал (збудження або гальмування) надходить на синапси наступних нейронів. Кожен синапс характеризується величиною синаптичного зв'язку або її вагою w_i , який з фізичного змістом еквівалентний електричній провідності.

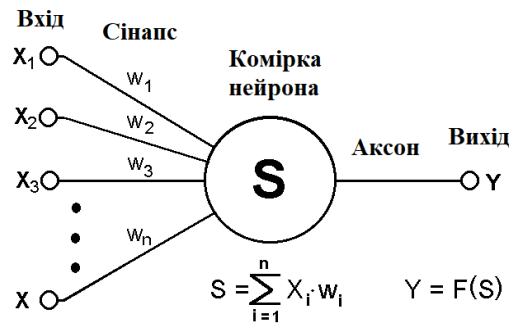


Рис. 1. Штучний нейрон

Поточний стан нейрона визначається, як зважена сума його входів:

$$S = \sum_{i=1}^n x_i \cdot w_i \quad (1)$$

Вихід нейрона є функція його стану:

$$y = f(s),$$

де w_i – вага (weight) синапса, $i=1\dots n$; b - значення зсуву (bias); s – результат сумування (sum); x_i – компонент вхідного вектора (вхідний сигнал), $i=1\dots n$; y – вихідний сигнал нейрона; n – число входів нейрона; f – нелінійне перетворення (функція активації).

В загальному випадку вхідний сигнал, вагові коефіцієнти і зсув можуть приймати дійсні значення, а в багатьох практичних задачах - лише деякі фіксовані значення. Вихід y визначається видом функції активації і може бути як дійсним, так і цілим.

Описаний обчислювальний елемент можна вважати спрощеною математичною моделлю біологічних нейронів, щоб підкреслити відмінність нейронів біологічних і штучних, другі іноді називають нейроноподібними елементами чи формальними нейронами.

Штучні нейронні мережі та їх архітектура. У загальному випадку поняття «штучна нейронна мережа» охоплює ансамблі нейронів будь-якої структури, проте практичне застосування знайшли тільки деякі з них. Це пояснюється тим, що архітектура ШНМ безпосередньо пов'язана з методом її навчання. Навіть різні етапи розвитку ШНМ визначалися появою нових архітектур мереж і спеціально розроблених для них методів навчання.

Сьогодні відома велика кількість нейронних структур та їх модифікацій, що орієнтовані на вирішення конкретного типу задач.

Нейронні структури з повним з'єднанням можуть бути як одношаровими, так і багатшаровими. У одношаровій структурі з повним з'єднанням всі вхідні сигнали можуть поступати на всі нейрони. Класичною структурою даного типу є перцептрон Розенблатта.

Багатшарові мережі з повним з'єднанням забезпечують можливість передачі інформації з кожного нейрона попереднього шару на будь-який нейрон наступного. Найчастіше це — багатшарові перцептрони.

Пряме поширення в межах одного шару використовують у випадку, коли група нейронів даного шару з'єднана з попереднім шаром опосередковано через виділені нейрони.

Двонаправлене поширення допускає також зворотну передачу, що дозволяє створювати алгоритми, за якими враховувався б взаємний міжнейронний обмін.

Таким чином, однонаправленість зв'язків призводить до побудови виключно ієрархічних структур, у яких обробка інформації розподіляється по рівнях. За кожний рівень ієрархічної обробки інформації відповідає свій шар нейронів.

На сьогоднішній день можна виділити чотири основні різновиди архітектури



Національного господарства
та природокористування

ШНМ:

- одношарові прямонаправлені мережі;
- багатшарові прямонаправлені мережі;
- рекурентні мережі;
- повністю зв'язані мережі.

Deepfake. Deepfake (діпфейк) - конкатенація слів «глибинне навчання» (англ. Deep learning) і «підробка» (англ. Fake), методика синтезу зображення, заснована на штучному інтелекті. Вона використовується для з'єднання і накладення існуючих зображень і відео на вихідні зображення або відеоролики. У переважній більшості випадків, для створення таких відео використовують генеративно-змагальні нейромережі (GAN). Одна частина алгоритму вчиться на реальних фотографіях певного об'єкта і створює зображення, буквально «змагаючись» з другою частиною алгоритму, поки та не почне плутати копію з оригіналом.

Deepfake може бути використаний для заміни певних елементів зображення на бажані образи, зокрема для створення фальшивих рекламних відео зі знаменитостями. Deepfake може бути використаний для створення підроблених новин і шкідливих обманів та не тільки ця технологія має також багато плюсів зокрема для використання в кіноіндустрії та при створенні рекламних роликів. Deepfake-ролики можна легко знайти на популярних сайтах потокового відео, таких як YouTube або Vimeo. Популярною програмою є FakeApp, що використовує TensorFlow.

Методи виявлення руху і перетворення в цільове відео, яке схоже на цільовий образ, були представлені в 2016 році і дозволяють створювати підроблені мімічні зображення в існуючому 2D-відео в режимі реального часу. Технологія deepfake використовувалася в музейному просторі, наприклад, в Музеї Сальвадора Далі у Флориді в честь 115-річчя відомого художника була організована спеціальна виставка Dalí Lives («Дали живий»). За словами кураторів проекту, на цьому заході «мистецтво зустрічає штучний інтелект». Вони використовували згенерований штучним інтелектом прототип художника, який спілкувався з відвідувачами музею, розповідав їм історії про своїх картинах і життя, а також робив Селфі і надсилав фотографії на пошту відвідувачів.

Створення Deepfake відео. Створимо власний Deepfake для синтезу діпфейка ми будемо використовувати популярну бібліотеку DeepFaceLab. Бібліотека стрімко розвивається, зараз існує кілька релізів. Цю бібліотеку можна вільно використовувати на ппформам: Windows, Google Colab – можна використовувати віддалені обчислювальні потужності та Linux.

Нижче описаний базовий процес створення дїпфейка на прикладї Windows. Важливо розумїти, що на якість результату впливає безлїч властивостей вихїдних файлїв (роздїльної здатностї та тривалостї вїдеофайлїв, рїзноманїтнїсть мїміки персонажїв, освїтлення і т. д.).

Архїв збїрки потрїбно розпакувати якомога ближче до кореня системного диска. Пїсля розпакування в каталозї DeerFaceLab ви знайдете безлїч bat-файлїв.

Мїсцем зберїгання моделї служить директорїя workspace. У нїй будуть мїститися вїдео, фотографїї і файли самої програми. Ви можете перейменувати каталог для збереження резервних копїй.

Як ви могли помітити, bat-файли мають на початку іменї номер. Кожен номер вїдповїдає певному кроцї виконання алгоритму.

1. Очищення робочого каталогу

На першому кроцї запуском 1) clear workspace.bat і натисканням пробїлу очищаємо зайве вміст папки workspace. Одночасно створюються необхіднї директорїї.

Вїдразу пїсля розпакування в workspace вже мїстяться приклади вїдеороликїв для тесту. Вїдповїдно до описаної термїнологїєю ви можете замїнити їх вїдеофайлами з тими ж назвами data_src і data_dst. Максимально пїдтримуваний дозвїл – 1080 p. Наведенї в документацїї приклади розширень файлїв: mp4, avi, mkv.

2. Витяг кадрїв з вїдеофайлу джерела

На другому кроцї витягаємо зображення з src-файлу (2) extract images from video data_src.bat). Для цього запускаємо bat-файл, отримуємо запрошення для вказївки кадрової частоти.

У формат png файли витягуються без втрат якостї, але на порядок повїльнїше і з великим об'ємом, нїж в jpg. Пїсля завдання параметрїв кадри витягуються в каталог data_src.

3. Витяг кадрїв сцени для перенесення особи

При необхідностї обрїзаємо вїдео за допомогою 3.1) cut video (drop video on me) .bat. Перетягуємо файл data_dst поверх bat-файлу. Вказуємо тимчасовї мїтки, номер дорїжки (якщо їх декїлька), бїтрейт вихїдного файлу. З'являється додатковий файл з суфїксом _cut.

Запускаємо 3.2) extract images from video data_dst FULL FPS.bat для вилучення кадрїв dst-сцени.

4. Складання вбїрки осїб джерела

Тепер необхідно детектувати особи на src-кадрах (рис. 2). Отримана вбїрка буде зберїгатися за адресою workspace \ data_src \ aligned. Цьому пункту вїдповїдає безлїч bat-файлїв, що починаються з 4) data_src extract faces і мають рїзнї доповнення пїсля:

Тип детектора особи: MT – трохи швидшїй, але виробляє бїльше помилкових осїб або S3FD - рекомендований, бїльш точний, менше помилкових осїб.

Варїант використання GPU: ALL (здїяти всї вїдеокарти), Best (використати кращу). Вибирайте другий варїант, якщо у вас є і зовнїшня, і вбудована вїдеокарти, і вам потрїбно паралельно працювати в офїсних додатках.

Запис роботи детекторїв (DEBUG). Кожен кадр з вїдїленими контурами осїб записується за адресою workspace \ data_src \ aligned_debug.



```
Performing 1st pass...
Running on GeForce 940MX. Recommended to close all programs using this device.
Using TensorFlow backend.
100%|#####|
Performing 2nd pass...
Running on GeForce 940MX. Recommended to close all programs using this device.
Using TensorFlow backend.
100%|#####|
Performing 3rd pass...
Running on CPU0.
Running on CPU1.
Running on CPU2.
Running on CPU3.
Running on CPU4.
Running on CPU5.
Running on CPU6.
Running on CPU7.
100%|#####|
-----
Images found:      655
Faces detected:   654
-----
Done.
```

Рис. 2. Запис детекторів

Bat-файл з параметром MANUAL застосовується для ручного переізвличення вже витягнутих осіб в разі помилок на етапі 4.2.other) data_src util add landmarks debug images.bat.

4.1. Видаляємо великі групи некоректних кадрів

Запускаємо 4.1) data_src check result.bat, переглядаємо результати в браузері XnView MP (при закритті запускайте цей bat-файл).

На цьому етапі необхідно видалити великі групи некоректних кадрів, щоб далі не витратити на них обчислювальний ресурс. До некоректним кадрам відносяться всі ті, що не містять чітко помітного особи. Особа також не повинно бути закрито предметом, волоссям тощо. Не витрачайте час на дрібні групи. Вони будуть видалені на наступному кроці.

4.2. Сортування і видалення інших некоректних кадрів

Файли з іменами, які починаються часткою з 4.2, служать для сортування та виявлення груп некоректних кадрів. Не закриваючи оглядач, послідовно запускайте bat-файли і видаляйте групи некоректних кадрів (зазвичай знаходяться в кінці).

4.2.1) data_src sort by blur.bat сортує кадри по різкості, видаляємо кадри з нечіткими особами.

4.2.2) data_src sort by similar histogram.bat групує кадри за змістом, дозволяє видаляти непотрібні особи групами.

4.2.4) data_src sort by dissimilar histogram.bat залишає ближче до кінця списку ті зображення, у яких найбільше схожих (зазвичай це особи анфас). На розсуд можна видалити частину кінця списку, щоб не проводити навчання на ідентичних осіб.

Опціонально: 4.2.5) data_src sort by face pitch.bat сортує особи так, щоб на початку списку обличчя дивилося вниз, а в кінці – вгору.

Опціонально: 4.2.5) data_src sort by face yaw.bat сортує особи по погляді зліва направо.

Додаткові сортувальні bat-файли, назви яких починаються з 4.2.other, сортують зображення за кількістю чорних пікселів, числу осіб в кадрі (потрібні кадри тільки з однією особою).

5. Складання вибірки осіб приймаючої сцени

Наступні операції з деякими відмінностями ідентичні вибірці осіб джерела. Головною відмінністю є те, що для приймаючої сцени важливо визначити dst-особи в усіх кадрах, що містять обличчя, навіть каламутні. Інакше в цих кадрах не буде здійснене заміни на джерело (рис. 3).

Опція + manual fix дозволяє вручну вказати контури особи на кадрах, де особу не було визначено. При цьому в кінці вилучення файлів відкрито вікно ручного виправлення контурів. Елементи управління описані у верхній частині вікна (викликаються клавішею H).



Рис. 3. Відображення контурів обличчя

Запуск 5.1) data_dst check results debug.bat дозволяє подивитися всі dst-кадри с накладеними поверх них передбаченими контурами обличчя. Видалить особи інших, неосновних персонажів.

6. Тренування

Навчання нейромережі (рис. 4) – сама часо-витратна частина, що триває години і цілі дні. Для тренування необхідно вибрати одну з моделей. Вибір і якість результату визначаються обсягом пам'яті відеокарти:

≥ 512 Мб → SAE. Найбільш гнучка модель з можливістю переносити стиль особи і освітлення.

≥ 2 Гб → H64. Найменш вимоглива модель.

≥ 3 Гб → H128. Аналогічна моделі H64, але з кращим дозволом.

≥ 5 Гб → DF – розумна тренування осіб, що виключає фон навколо особи, або LIAEF128 – модель аналогічна DF, але намагається морфіровать вихідне особа в цільове, зберігаючи риси вихідного особи.

≥ 6 Гб → AVATAR – модель для управління чужим обличчям, потрібні квадратні відеоролики або SAE HD - для самих останніх відеокарт.

При коректних умовах паралельно з консоллю відкривається вікно Training preview, в якому буде відображатися процес навчання і крива помилки. Зниження кривої відображає прогрес тренування. Кнопка p (англійська розкладка) оновлює попередній перегляд.



Рис. 4 . Тренування НР

Процес тренування можна переривати, натискаючи Enter у вікні Training preview, і запускати в будь-який час, модель буде продовжувати навчатися з тієї ж точки. Чим довше триває тренування, тим кращий результат ми отримаємо.

7. Накладення осіб

Тепер у нас є результат навчання. Необхідно поєднати src-особи і кадри dst-сцени. Зі списку bat-файлів вибираємо ту модель, на якій відбувалася тренування. Можливо кілька режимів накладення, за замовчуванням використовується метод Пуассона. Як інших параметрів для першої проби можна використовувати параметри за замовчуванням (після натискання Enter) і варіювати їх, якщо вас не влаштує результат накладення.

8. Склеювання відео

Наступні bat-файли склеюють картинки в відео з тієї ж частотою кадрів і звуком, що і data_dst (тому файл повинен залишатися в папці workspace). Підсумковий файл буде збережений під ім'ям result. Готово! Нижче представлений отриманий приклад (рис. 5).



Рис. 5. Скріншот готового відео

Висновок

В даній роботі було проведено аналіз існуючих архітектур та методів навчання нейронних мереж. Як результат для програмної реалізації обрано модель навчання нейронної мережі для створення Deepfake відео.

З першу було приведено поняття нейрону та штучної нейронної мережі. Також зроблено аналіз основних властивостей нейронних мереж, приведено варіанти побудови їх архітектури.

В результаті проведеного аналізу для подальших досліджень обрано багатопоточну архітектуру нейронної мережі. Вибір зумовлений характеристиками даної архітектури, які будуть необхідні для реалізації подальшої мети проекту, а саме вирішенню задачі класифікації.

Було проведено аналіз існуючих моделей та засобів організації багатопоточних обчислень. В результаті аналізу було обрано DeepFaceLab.

Другий етап присвячений саме багатопоточним системам. Були розглянуті різноманітні механізми взаємодії процесів, основні моделі організації багатопоточності. Також було створено Deep fake відео за допомогою алгоритмів бібліотеки DeepFaceLab.

В результаті проектування розроблено за допомогою штучних нейронних мереж відеоролик, в якому було змінено обличчя однієї людини на іншу.

При внесенні модифікацій розроблений продукт може використовуватися, наприклад, в телебаченні та кіноіндустрії.

1. Горбань А. Н. Обучение нейронных сетей. ПараГраф, 1990.
2. Эндрюс Г. Р. Основы многопоточного, параллельного и распределённого программирования.
3. ДБНВ.1.1-7 – 2002. Пожарная безопасность объектов строительства Госстрой Украины. Киев, 2003. 42 с.
4. Хинтон Дж. Е. Как обучаются нейронные сети. *В мире науки*. 1992. N 11. N 12. С. 103–107.
5. ДНАОП 0.00 - 1.31 - 99. Правила охорони праці під час експлуатації ЕОМ. К. : Основа, 1999. 111 с.
6. Методичні вказівки до виконання організаційно-економічного розділу дипломних проектів / за ред. А. Т. Чернявського. К. : НТУУ «КПІ», 1999.
7. Минский М., Пеперт С. Перцептроны. М. : Мир, 1971. 262 с.
8. URL: <https://uk.wikipedia.org/wiki/Deepfake> (дата звернення: 12.05.2021).
9. URL: <https://cedem.org.ua/articles/deep-fake/> (дата звернення: 12.05.2021).

Sydor A. I., Candidate of Engineering (Ph.D.), Senior Lecturer; Opanasiuk E. M., Senior Student (National University of Water and Environmental Engineering, Rivne)

FACE CHANGE IN “DEEFAKE” VIDEOS BASED ON THE USE OF NEURAL NETWORKS

The work is devoted to the study of neural networks, namely the creation of Deepfake videos based on neural networks. The paper will demonstrate the neural network algorithm and the final result in the form of a fake video.

Keywords: Deepfake, neural network (NM), artificial neural network (ARN).



водного господарства
та природокористування

УДК 004.658

Сидор А. І., к.т.н., старший викладач; Щепковський В. В., студент (Національний університет водного господарства та природокористування, м. Рівне)

ДОСЛІДЖЕННЯ ТА ВИКОРИСТАННЯ SAAS-ПЛАТФОРМ ДЛЯ WEB-РЕСУРСІВ

Дана дипломна робота присвячена аналізу використання SaaS-платформ для створення web-сайтів та web-додатків. Проведено аналіз таких SaaS-платформ: Weebly, Wix, uCoz, InSales, Shopify, Site123. Розроблено методичні рекомендації на рівні сайту по роботі з SaaS-платформами.

Ключові слова: SaaS, Web-сайт, Web-додаток, SEO, кросбраузерність, система керування вмістом, шаблон, модуль.

Вступ. В епоху Інтернету сайт необхідний кожному хто займається власним бізнесом, бо сайт – це обличчя вашої компанії. Але зазвичай, створення сайту це досить незрозумілий для бізнесмена процес.

SaaS (software as a service програмне забезпечення як послуга) це спосіб поширення програмного забезпечення на орендній основі. Тобто не треба купувати CMS і замовляти індивідуальну розробку, досить тільки конфігурувати виданий постачальником сайт під свої завдання і щомісяця (або щорічно) вносити абонентську плату.

Ціллю дипломної роботи є аналіз засобів та можливостей створення сайтів на SaaS-платформах Cloud, розробка інструкції по використанню платформ Cloud для створення Web-сайтів та Web-додатків.

У роботі проаналізовані можливі проблеми при використанні SaaS-платформ, надані області використання, що підвищують ефективність розробки, розроблені методичні рекомендації у вигляді сайту по роботі з SaaS-платформами, а також порівняні найвідоміші платформи.

Аналіз існуючих методів створення web-сайтів. У 1989 році Тім Бернерс-Лі створив проект «Всесвітня Павутина» (англ. World Wide Web). «Всесвітня Павутина» спочатку була задумана для працівників CERN, як альтернатива сховищу даних, щоб полегшити пошук інформації та систематизувати її.

Перший у світі сайт з'явився 6 серпня 1991 року та й досі зберігається за адресою <http://info.cern.ch>.

З цього часу почався стрімкий розвиток веб-технологій і майже через 20 років Тім Бернес-Лі сказав: «Я стільки часу витратив, щоб дати людям можливість викладати в мережу абсолютно будь-яку інформацію, але я й уявити собі не міг, що люди захочуть викласти туди буквально все».

- Ручний метод створення веб-сайтів – реалізується за допомогою HTML, CSS та одної або декількох мов веб-програмування.



- Системи керування контентом – CMS (англ. Content Management System – система керування контентом) – програмне забезпечення для організації веб-сайтів чи інших інформаційних ресурсів в Інтернеті чи окремих комп'ютерних мережах.

Найбільшу популярність мають такі CMS:

- Drupal
- Joomla!
- MODX
- UMI.CMS
- WordPress
- NetCat
- HostCMS
- CS-Cart
- AMIRO.CMS
- NespiCMS

- SaaS-платформи - програма як послуга (англ. Software as a service, SaaS) — модель поширення програм споживачам, при якій постачальник розробляє веб-програму, розміщує її й управляє нею (самостійно або через третіх осіб) з метою використання її замовниками через інтернет. Замовники платять не за володіння програмами як такими, а за їх використання.

Відомі конструктори сайтів на базі SaaS-систем:

- uKit
- Wix
- Sellbe
- uCoz
- InSales
- Nethouse
- Jimdo
- uWeb
- Weebly
- UMI
- Prom.ua
- LPGenerator
- WebsiteX5
- Skynell

Аналіз та порівняння saas-платформ. Існує велика кількість SaaS-платформ. На цих платформах можна створювати різні типи сайтів: візитки, блоги, інтернет-

магазини, landing page тощо. Вони мають велику кількість шаблонів, готових модулів, багато налаштувань та можливість додавати власний код.

Щоб розібратись у чому різниця між різними SaaS-платформи проаналізуємо та порівняємо такі з них: WIX, InSales, Shopify, Site123, uCoz. Порівнювати будемо по 4 параметрам: дизайн шаблонів, можливості панелі управління, SEO та тарифи.

1.1. WIX. WIX (рис. 1) має простий та інтуїтивно зрозумілий редактор – це дозволяє без певних навиків та знань настроїти та редагувати елементи на веб-сторінці.

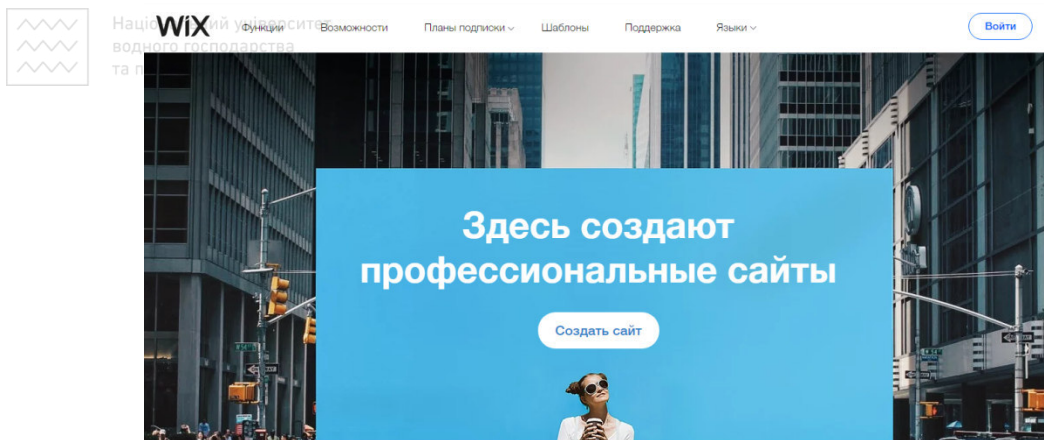


Рис. 1. Головна сторінка WIX

1.2. uCoz. uCoz (рис. 2) – SaaS-платформа, яка має зручний інтерфейс, на почakovій сторінці є пояснення, як користуватися платформою.

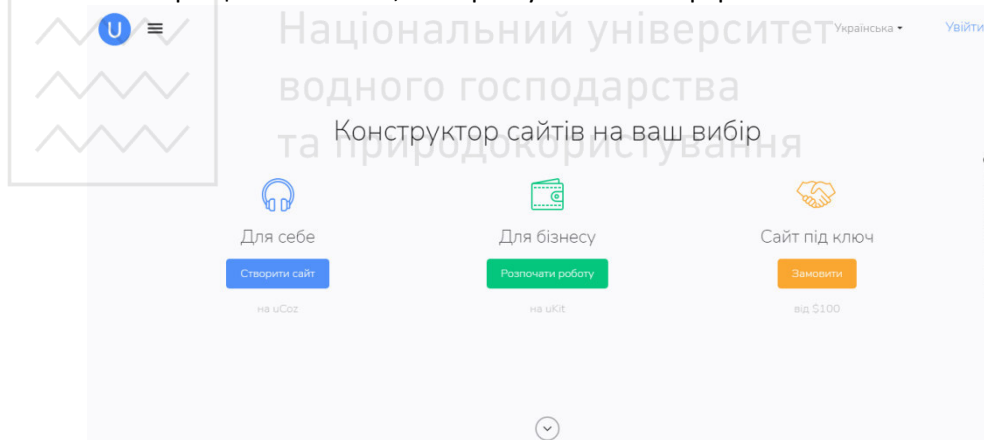


Рис. 2. Головна сторінка uCoz

1.3. InSales. InSales (рис. 3) – конструктор для створення лише інтернет-магазинів. Одразу після реєстрації вам запропонують ввести основні дані про магазин, додати перший товар, налаштувати дизайн сайту, способи доставки, способи сплати, додати власний домен, розширити функціонал сайту.

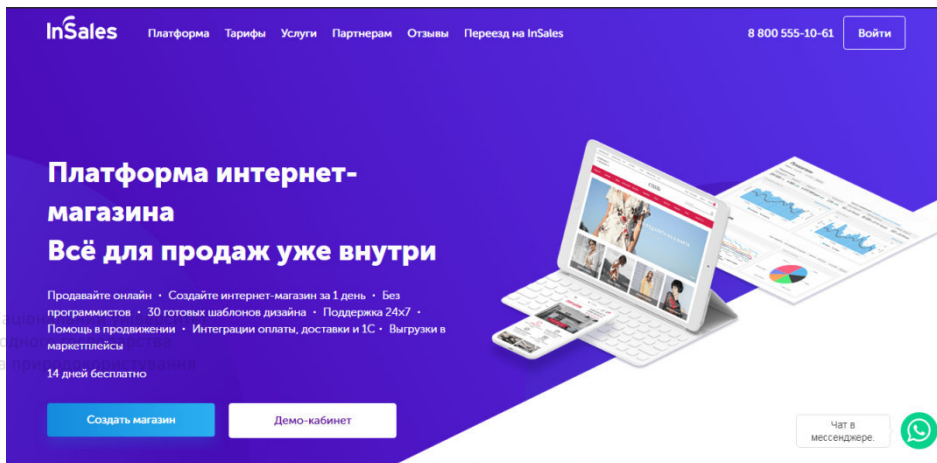
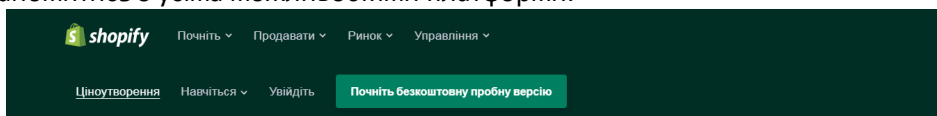


Рис. 3. Головна сторінка сайту InSales

1.4. Shopify. Shopify (рис. 4) – платформа для створення інтернет-магазину, пропонує 2-х тижневий безкоштовний тиждень для того, щоб ви могли познайомитись з усіма можливостями платформи.



Створіть свій магазин, виберіть план пізніше

Спробуйте Shopify безкоштовно протягом 14 днів, кредитна картка не потрібна

Почніть безкоштовну пробну версію

Рис. 4. Головна сторінка Shopify

1.5. Site123. Site123 (рис. 5) – це проста, зручна та швидка SaaS-платформа. Щоб створити сайт не потрібно мати навичок або знань з програмування. На одному акаунті можна створити скільки завгодно сайтів.

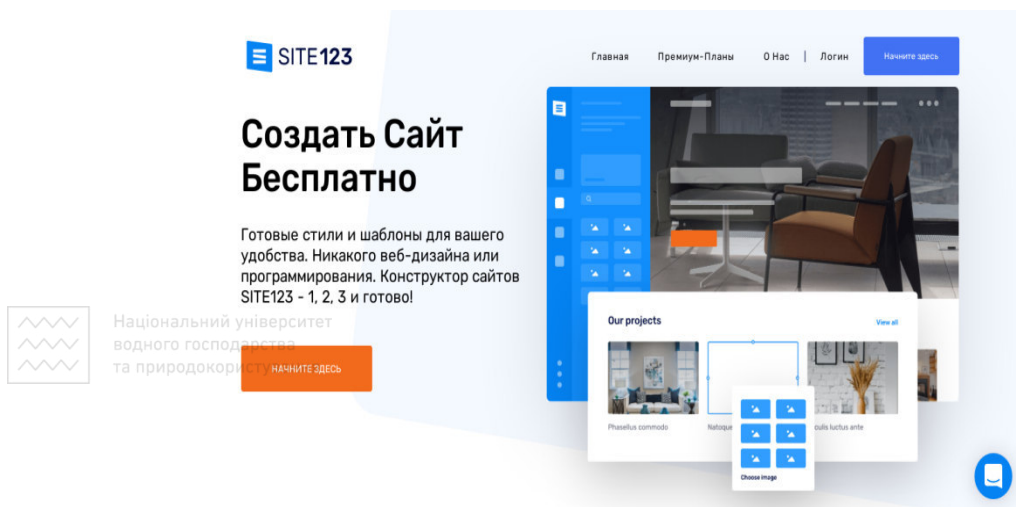


Рис. 5. Головна сторінка платформи Site123

Було проаналізовано п'ять SaaS-платформ. Такі платформи як InSales, Shopify направлені на створення інтернет-магазинів і тому вони мають багато додаткових модулів які пов'язані з продажами. Skynell, Wix, uCoz – платформи на яких можна створювати сайти різного напрямлення у тому числі й інтернет-магазини. Багато функціональних можливостей доступні тільки на платних тарифах, тому у Таблиці 1 наведу ціни мінімальних та максимальних тарифів, а у Таблиці 2 – переваги та недоліки проаналізованих SaaS-платформ.

Таблиця 1

Ціни SaaS-платформ

SaaS-платформа	Тарифи	
	Мінім. ціна(\$)	Максим. ціна(\$)
Wix	4.50	24.50
uCoz	2.99	15.99
InSales	17.59	57.83
Shopify	29	299
Site123	0	7,80

Аналіз SaaS-платформ

SaaS-платформа	Параметри		
	Дизайн	Панель управління	SEO
Wix	Створення шаблону за допомогою штучного інтелекту.	Зручна управління, велика кількість модулів.	Є SEO-мастер, який покроково допоможе налаштувати параметри SEO
uCoz	Версія сайту для людей з проблемами зору, змінювати HTML, велика кількість однотипних шаблонів.	Складна панель управління	Обмежений функціонал SEO модуля
InSales	Багато однотипних шаблонів, можливість змінити HTML/CSS/Java Script.	Доводиться довше чекати переключення між шаблоном та налаштуваннями	Гарний інструмент для збору статистики та SEO-оптимізації.
Shopify	Шаблони однотипні, змінювати код можуть тільки розробники з команди Shopify, всього 26 безкоштовних шаблонів.	Зручна панель управління для інтернет-магазинів.	Більшість додатків для SEO-оптимізації платні.
Site123	Усього 10 шаблонів, можна змінювати структуру(12 варіантів)	Проста панель управління, мало модулів.	Мало налаштувань для SEO-оптимізації.

Мета-мова XML. Метамова – це мова, що призначена для опису мов. XML – це розширювана мова розмітки, яка призначена для структурування, зберігання та передачі інформації. Вона не має певних тегів, як HTML, ви можете створювати їх самі. Наприклад, коли вам потрібно додати дані про книгу та автора на сторінку у вигляді XML, то це можливо зробити так:

```
<information>
```

```
<author></author >
```


<book></ book >
<year></ year >
<prise></ prise >
</ information >

У програмній реалізації для створення покрокової інструкції створення сайтів на SaaS-платформах були створені такі теги: TITLE, Tn, де n – це кількість кроків.

TITLE – додається заголовок сторінки

Tn – відповідає за покрокову інструкцію, відображається на сторінці у вигляді

клавiши.

Використовувати XML дуже зручно:

- 1) легко зрозуміти, бо використовуються теги, які вами написані і вони можуть чітко характеризувати дані, які там будуть вказуватись;
- 2) є поділ даних, якщо вам потрібно буде з часом зробити зміни, то ви просто зможете змінити XML файл. Це економить ваш час.
- 3) є можливість створити одну XML сторінку з даними та використовувати її на різних сайтах.

Отже, XML тримає ваші роботи організована і це допомагає економити час.

Висновки. Перший сайт був створений 6 серпня 1991 року з того часу минуло вже 26 років та відбулося дуже багато змін і зараз у 2017 кожному хто має свій власний бізнес і не важливо він малий чи середній, потрібен свій сайт.

У наш час сайт – це візитна картка підприємства, фірми. Велика кількість компаній витрачає багато грошей на створення брошур, але маючи свій персональний сайт цього роботи не потрібно. Потенційні клієнти можуть дізнатися про вашу фірму і вашу продукцію через мережу. Сьогодні майже всі користуються інтернетом. Тому свій власний сайт – це крок до розвитку власного бізнесу.

Модель SaaS (Software as a Service) – модель поширення програм споживачам, при якій постачальник розробляє веб- програму, розміщує її й управляє нею (самостійно або через третіх осіб) з метою використання її замовниками через інтернет.

Користувач SaaS системи, орендує його, оплачуючи певну суму за період часу. Завдяки цьому таке рішення є економічно вигідним. Перевага в тому, що у користувача немає необхідності займатися технічною стороною питання: установкою, підтримкою, відновленням, сумісністю та іншими питаннями - тобто йому залишається лише використовувати необхідний функціонал в своїх бізнес цілях.

Немає необхідності придбання ліцензії на користування продуктом: замість цього оплачується його оренда на певний час. Це може бути щомісячна оплата або оплата за обсяг даних. При цьому сервісне обслуговування (підтримка і оновлення системи) вже включені у вартість

Якщо сервіс чимось не влаштував або необхідність його використання пропала, можна просто не продовжувати оплату послуги.

Основна частина користувачів SaaS – це малий і середній бізнес, які не можуть витрачати кошти на розробку повноцінного власного сайту. SaaS дозволяє скоротити фінансові витрати на придбання програмного продукту і подальшу його підтримку, вони кросплатформені та кросбраузерні.

З боку розробника така модель дозволяє боротися з проблемою піратств – поширення неліцензійних копій програмного продукту, так як кінцева програма не влучає у руки користувача в готовому вигляді.

У SaaS-платформ дуже багато переваг:

- 1) відсутність витрат на обладнання;
- 2) сайт готовий одразу, як ви пройшли реєстрацію;
- 3) Додаткове місце для зберігання або послуги можуть бути доступні на вимогу без необхідності установки нового програмного забезпечення і апаратних засобів;
- 4) Оновлення платформ безкоштовне та автоматичне;
- 5) Додатки можуть бути доступні через будь-який інтернет-сумісний пристрій, наприклад, настільні комп'ютери, смартфони та планшети;
- 6) Додатки часто можуть бути змінені відповідно до потреб і брендингу окремих клієнтів.

Приклади, де SaaS можуть бути використані підприємствами:

- облік і виставлення рахунків
- відстеження продаж;
- планування
- моніторинг продуктивності
- комунікації

Один з прикладів SaaS-платформ – корпоративна пошта на Gmail, Amazon Web Services, Slack тощо.

1. Методи розробки сайтів. URL: <http://webstudio2u.net/ua/webdesign/354-site-develop-methods.html>. (дата звернення: 12.05.2021). 2. Б. Лоусон Изучаем HTML5. Библиотека специалиста / Б. Лоусон, Р. Шарп. Питер : Питер, 2011. 235 с. 3. Шмитт К. CSS. Рецепты программирования. URL: <http://static.ozone.ru/multimedia/book-file/1005871779.pdf> (дата звернення: 12.05.2021). 4. Офіційний сайт Wix. URL: <https://www.wix.com>. (дата звернення: 12.05.2021). 5. Офіційний сайт uCoz. URL: <https://www.ucoz.ua>. (дата звернення: 12.05.2021). 6. Офіційний сайт InSales. URL: <https://www.insales.ru> (дата звернення: 12.05.2021). 7. Офіційний сайт Shopify. URL: <https://ru.shopify.com>. (дата звернення: 12.05.2021). 8. Офіційний сайт Site123. URL: <https://ua.site123.com>. (дата звернення: 12.05.2021). 9. Горнаков С. Г. Осваиваем популярные системы управления сайтом. Москва : ДМК Пресс, 2009. 336 с. 10. Лучшие конструкторы сайтов. URL: <https://uguide.ru/rejting-luchshij-konstruktor-sajtov-runeta>. (дата звернення: 12.05.2021). 11. SaaS. URL: <https://ru.wikipedia.org/wiki/SaaS> (дата звернення: 12.05.2021).

Sidor A. I., Candidate of Engineering (Ph.D.), Senior Lecturer; Shchepkovsky V. V., Senior Student (National University of Water and Environmental Engineering, Rivne)

RESEARCH AND USE OF SAAS-PLATFORMS FOR WEB-RESOURCES

This thesis is devoted to the analysis of the use of SaaS-platforms for creating web-sites

and web-applications. The following SaaS platforms were analyzed: Weebly, Wix, uCoz, InSales, Shopify, Site123. Developed guidelines at the site level for working with SaaS-platforms.

Keywords: SaaS, Web-site, Web-application, SEO, cross-browser, content management system, template, module.

УДК 004.658



Національний університет

Сидор А. І., к.т.н., старший викладач; Кватерчук М. О., студент; Зищук М. В., студент (Національний університет водного господарства та природокористування, м. Рівне)

ТЕСТУВАННЯ ТА АНАЛІЗ ОСНОВНИХ ХАРАКТЕРИСТИК СИСТЕМ УПРАВЛІННЯ БАЗАМИ ДАНИХ

Робота присвячена дослідженню основних характеристик СУБД. Проведено огляд найбільш популярних систем управління базами даних, досліджено особливості налаштування СУБД, проведено порівняльний аналіз розмірності баз даних та швидкодії їх обробки в залежності від використовуваного середовища.

Ключові слова: бази даних (БД), системи управління базами даних (СУБД), SQL, sql-запити, тестування.

Вступ. Сучасне суспільство використовує у своєму повсякденному житті все більше різноманітної інформації. Основу сучасної інформаційної технології виконує концепція, згідно якої дані мають бути організовані в бази даних для покращення актуальності відображення їх вмісту та задоволення інформаційних потреб користувачів. База даних (БД) - це сукупність взаємозв'язаних даних, що зберігаються разом. Ці бази даних створюються і функціонують під управлінням спеціальних програмних комплексів, званих системами управління базами даних (СУБД).

Збільшення об'єму і структурної складності даних, що зберігаються, розширення кола користувачів інформаційних систем привели до широкого поширення найбільш зручних і простих для розуміння різних СУБД. Тому, у користувачів може виникнути проблема при виборі середовища, адже кожна програма має різний інтерфейс, розмір, різну складність установки та швидкість виконання запитів. Для того, щоб підібрати найбільш оптимальне середовище СУБД, доцільно провести тестування основних характеристик систем управління базами даних, і на основі проведеного аналізу та отриманих результатів дослідження виявити, які середовища являються найзручнішими у використанні, а які доцільніше використовувати для спеціалізованих задач.

Огляд найбільш популярних систем управління базами даних (СУБД). Для роботи з базами даних використовують спеціальні програми – системи управління базами даних (СУБД). Системи управління базами даних – це прикладні комп'ютерні програми, призначені для створення, збереження і використання баз даних (рис. 1). Залежно від моделі даних, яка використовується в СУБД, їх поділяють на ієрархічні,

мережеві, реляційні, об'єктно-реляційні тощо. Залежно від розміщення баз даних і компонентів, які забезпечують їхнє опрацювання, СУБД поділяють на:

- локальні – забезпечують зберігання та опрацювання даних на локальному комп'ютері;
- розподілені – дані зберігаються та опрацьовуються на різних комп'ютерах у локальній або глобальній мережі.

Залежно від способу доступу до бази даних СУБД розподіляють на:

- файл-серверні – файли з даними розміщуються на сервері, а на кожному клієнтському комп'ютері встановлено повну версію СУБД. Доступ з клієнтського сервера здійснюється з використанням мережі;
- клієнт-серверні – на сервері встановлюється серверна версія СУБД, а також база даних. На клієнтських комп'ютерах встановлюються тільки невеликі за обсягом і функціональністю клієнтські версії СУБД. Усі операції з даними здійснюються на сервері;
- інтегровані (вбудовані) – використовуються як складові інших програмних продуктів, наприклад електронних енциклопедій, словників, пошукових систем тощо. Доступ до даних здійснюється з використанням засобів прикладної програми, у яку інтегровано СУБД.



Національний університет водного господарства та природокористування



Рис. 1. Види СУБД

Розглянемо характеристики, переваги та недоліки найпопулярніших СУБД.

SQLite. SQLite – це дуже популярна база даних, яка з успіхом використовується у форматі файлу на диску для таких настільних додатків, як системи контролю версій, засоби фінансового аналізу, засоби каталогізації та редагування носіїв, CAD-пакети, програми ведення записів тощо.

Найголовнішими перевагами SQLite є: легкість, висока продуктивність, не потребує установки, надійність, портативність, доступність. Недоліками SQLite являються неможливість додаткового налаштування та використання для обробки http-запитів із низьким та середнім трафіком.

MS Access. Microsoft Office Access являється СУБД реляційного типу.

Найголовнішими перевагами MS Access є: простота інтеграції, наявність NET-friendly, зручний обсяг пам'яті, підтримка декількох користувачів, наявність імпорту

даних. Недоліками MS Access являються низька якість sql-запитів, обмеження для одночасного доступу користувачів, збереження всієї інформації у одному файлі.

MySQL. Denwer – набір дистрибутивів і програмна оболонка, яка дає змогу працювати з базою даних MySQL.

Найголовнішими перевагами Денвера є: висока швидкість, простота та зрозумілість, малий розмір програми, висока оптимізація, підтримка російської мови, наявність можливості інтеграції додаткових бібліотек та перенесення на інші пристрої. Найбільшим недоліком Денвера являється невеликий функціонал базового набору програми.

MONGODB. MongoDB – документо-орієнтована СУБД з відкритим програмним кодом, що не вимагає опису схеми таблиць, класифікована як NoSQL, використовує JSON - подібні документи та схеми бази даних.

Найголовнішими перевагами MongoDB є: наявність гнучкої БД, висока швидкість та доступність, горизонтальна масштабованість та просте налаштування навколишнього середовища. Найбільшими недоліками MongoDB являються велике використання пам'яті, обмеження обсягу даних.

PostgreSQL. PostgreSQL – вільна об'єктно-реляційна СУБД.

Найголовнішими перевагами PostgreSQL є: простота використання, можливість визначення типу даних користувачем та використання збережених процедур. Найбільшими недоліками PostgreSQL являються: низька продуктивність, складна реплікація, непросто встановлення програми.

Дослідження та тестування основних характеристик СУБД

Для тестування СУБД використовувалися дві робочі системи, а саме:

- ноутбук з наступними параметрами: ОС Windows 10, процесор Intel Core i5-8300N CPU 2.30 GHz, 8 Гб ОЗУ, 64-розрядний тип системи, процесор x64;
- ПК з такими параметрами: ОС Windows 10, процесор Intel Core i7-4790K CPU 4.00GHz, 16 Гб ОЗУ, 64-розрядний тип системи, процесор x64.


Для дослідження СУБД проводився підбір доступних в Інтернеті СУБД. Основними критеріями підбору були: популярність використання, наявність безплатної версії, зручність установки та налаштування. Для дослідження були обрані наступні СУБД: SQLite, MS Access, MySQL, MongoDB та PostgreSQL. Для порівняльного аналізу запропонованих СУБД були проведені тести швидкості виконання sql-запитів, досліджено розміри бази даних з однаковими параметрами, створеної в різних СУБД, а також розміри установщика і необхідної кількості пам'яті, що є необхідним для середовища кожної із розглянутих СУБД.

Спочатку досліджувався процес установки та початкового налаштування середовища. Найменше складностей виникло при установці SQLite та MySQL, що входить в пакет Denwer, найактуальніші версії були наявні в Інтернеті, є безліч статей та відеоуроків по установці та налаштуванню, сам процес установки зайняв найменше часу, а найбільші труднощі виникли з MongoDB через невелику кількість матеріалів по установці та потребу додаткових налаштувань.

Для того, щоб порівняти як розроблена база даних залежить від середовища розробки була створена однакова база даних з таблицею "student" для всіх

середовищ із наступними полями: “id_user” типу integer, “data” типу data, “nomer” типу integer, “priz”, “name”, “pobat” типу text.

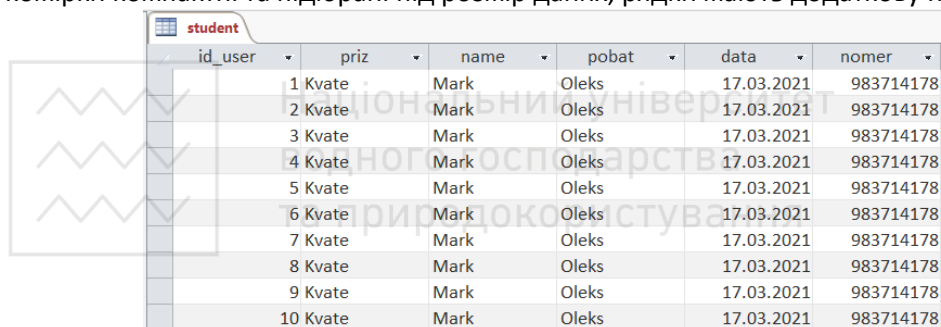
На рис. 2–6 наведені скріншоти створеної таблиці бази даних у середовищах СУБД, які були обрані для тестування.



	id_user	priz	Name	pobat	data	nomer
	Фильтр	Фильтр	Фил...	Фильтр	Фильтр	Фильтр
1	1	Kvate...	Mark	Oleks...	2021-03-17	983714178
2	2	Kvate	Mark	Oleks	2021-03-17	983714178
3	3	Kvate	Mark	Oleks	2021-03-17	983714178
4	4	Kvate	Mark	Oleks	2021-03-17	983714178
5	5	Kvate	Mark	Oleks	2021-03-17	983714178
6	6	Kvate	Mark	Oleks	2021-03-17	983714178
7	7	Kvate	Mark	Oleks	2021-03-17	983714178
8	8	Kvate	Mark	Oleks	2021-03-17	983714178
9	9	Kvate	Mark	Oleks	2021-03-17	983714178
10	10	Kvate	Mark	Oleks	2021-03-17	983714178

Рис. 2. База даних у середовищі SQLite

Таблиця в SQLite виглядає зручно, є наявні додаткові поля для фільтрування, комірки компактні та підібрані під розмір даних, рядки мають додаткову нумерацію.



	id_user	priz	name	pobat	data	nomer
1	1	Kvate	Mark	Oleks	17.03.2021	983714178
2	2	Kvate	Mark	Oleks	17.03.2021	983714178
3	3	Kvate	Mark	Oleks	17.03.2021	983714178
4	4	Kvate	Mark	Oleks	17.03.2021	983714178
5	5	Kvate	Mark	Oleks	17.03.2021	983714178
6	6	Kvate	Mark	Oleks	17.03.2021	983714178
7	7	Kvate	Mark	Oleks	17.03.2021	983714178
8	8	Kvate	Mark	Oleks	17.03.2021	983714178
9	9	Kvate	Mark	Oleks	17.03.2021	983714178
10	10	Kvate	Mark	Oleks	17.03.2021	983714178

Рис. 3. База даних у середовищі MS Access

Таблиця в MS Access містить дані середнього розміру, наявне додаткове меню для швидкого сортування та застосування фільтрів, поля розраховані на довгі дані.

	id_user	priz	name	Pobat	data	nomer
<input type="checkbox"/> Edit Copy Delete	2	Kvate	Mark	Oleks	2021-03-17	983714178
<input type="checkbox"/> Edit Copy Delete	3	Kvate	Mark	Oleks	2021-03-17	983714178
<input type="checkbox"/> Edit Copy Delete	4	Kvate	Mark	Oleks	2021-03-17	983714178
<input type="checkbox"/> Edit Copy Delete	5	Kvate	Mark	Oleks	2021-03-17	983714178
<input type="checkbox"/> Edit Copy Delete	6	Kvate	Mark	Oleks	2021-03-17	983714178
<input type="checkbox"/> Edit Copy Delete	7	Kvate	Mark	Oleks	2021-03-17	983714178
<input type="checkbox"/> Edit Copy Delete	8	Kvate	Mark	Oleks	2021-03-17	983714178
<input type="checkbox"/> Edit Copy Delete	9	Kvate	Mark	Oleks	2021-03-17	983714178
<input type="checkbox"/> Edit Copy Delete	10	Kvate	Mark	Oleks	2021-03-17	983714178

Рис. 4. База даних у середовищі MySQL

Таблиця MySQL містить дані середнього розміру, які зменшуються при необхідності відображення більшої кількості даних на сторінці, розмір комірки підганяється під розмір даних, є додаткові поля для швидкого редагування, копіювання та видалення даних, передбачена одночасна робота з декількома рядками.

#	_id ObjectId	name String	priz String	pobat string	data Double	nomer Double
1	60608090df4e2e9b09c45	"Mark"	No field	No field	No field	No field
2	606080c0df4e2e9b09c46	"Mark"	"kvater"	"oleks"	2021	9839529
3	606080c0df4e2e9b09c47	"Mark"	"kvater"	"oleks"	2021	9839529
4	606080c0df4e2e9b09c48	"Mark"	"kvater"	"oleks"	2021	9839529
5	606080c0df4e2e9b09c49	"Mark"	"kvater"	"oleks"	2021	9839529
6	606080c0df4e2e9b09c4a	"Mark"	"kvater"	"oleks"	2021	9839529
7	606080c0df4e2e9b09c4b	"Mark"	"kvater"	"oleks"	2021	9839529
8	606080c0df4e2e9b09c4c	"Mark"	"kvater"	"oleks"	2021	9839529
9	606080c0df4e2e9b09c4d	"Mark"	"kvater"	"oleks"	2021	9839529
10	606080c0df4e2e9b09c4e	"Mark"	"kvater"	"oleks"	2021	9839529

Рис. 5. База даних у середовищі MongoDB

Таблиця MongoDB містить дані, що відображаються дрібним шрифтом, додатково відображає тип поля, є наявна нумерація рядків таблиці, дані в комірках зручні для копіювання в SQL-запити.

	priz	name	Pobat	data	nomer	id
1	k	m	o	2021	3809739	1
2	k	m	o	2021	3809739	2
3	k	m	o	2021	3809739	3
4	k	m	o	2021	3809739	4
5	k	m	o	2021	3809739	5
6	k	m	o	2021	3809739	6
7	k	m	o	2021	3809739	7
8	k	m	o	2021	3809739	8
9	k	m	o	2021	3809739	9
10	k	m	o	2021	3809739	10

Рис. 6. База даних у середовищі PostgreSQL

Таблиця PostgreSQL містить дані середніх розмірів, тип полів вказаний відразу, присутня нумерація рядків таблиці.

Для перевірки швидкодії виконання sql-запитів був створений наступний sql-запит:

```
INSERT INTO `student` (`id_user`,`priz`,`name`,`Pobat`,`data`,`nomer`)
VALUES ('2','Kvate','Mark','Oleks','2021-03-17','983714178');
```

Даний запит вставляє в наявну таблицю відповідні дані. Такий запит був продубльований 1000 разів (для середовища MC Access - 450 у зв'язку з тим, що середовище не може виконувати більше запитів в одній базі даних) та запущений на виконання.

Результати проведеного тестування СУБД. Результати тестування наведені в таблицях 1–5 та рис. 7–8.

Таблиця 1

Порівняння розмірів файлу установки

Розміри файлів установки	
SQLite	17,2 Мб
MC Access	13 Мб
MySQL	8,7 Мб
MongoDB	270,5 Мб
PostgreSQL	208,3 Мб

У таблиці 1 наведені розміри установників СУБД, найменший розмір 8,7 Мб має пакет Denwer, що містить БД MySQL, а найбільший - MongoDB, що становить 270,5 Мб

Таблиця 2

Порівняння розмірів програм середовищ

Розміри програм середовищ	
SQLite	43,9 Мб
MC Access	15,4 Мб
MySQL	72 Мб
MongoDB	1,7 Гб
PostgreSQL	604 Мб

У таблиці 2 наводяться розміри СУБД після установки, як видно з таблиці найменше пам'яті потребує MC Access (15,4 Мб), у той же час, щоб використовувати СУБД MongoDB необхідно 1,7 Гб вільної пам'яті.

Таблиця 3

Порівняння розмірів баз даних

Розміри БД	
SQLite	52 кб
MC Access	716 кб
MySQL	64 кб
MongoDB	390 кб
PostgreSQL	64 кб

Досить важливою характеристикою є розмір створеної бази даних, у таблиці 3 наведені результати створеної БД з однаковими параметрами в різних середовищах СУБД, як видно з отриманих результатів найбільшою компактністю характеризуються БД, створені в середовищі SQLite 52 кб, у той же час найбільш масивними є БД, створені в середовищі MS Access, що мають розмір цілих 716 кб.

Таблиця 4

Порівняння швидкості виконання запитів у різних середовищах



Національний університет
водного господарства
та природокористування

Швидкість виконання запитів	
SQLite	9,33 с
MC Access	1,23 с
MySQL	0,0046 с
MongoDB	0,054 с
PostgreSQL	0,062 с

Основним показником ефективності БД є швидкість виконання запитів, з таблиці 4 видно, що по цьому параметру найкращі результати у MySQL, де 1000 SQL-запитів виконалися за 0,0046 с, найменшу швидкість виконання запитів продемонструвало середовище SQLite, де 1000 SQL-запитів виконалися за 9,33 с

Таблиця 5

Порівняння зручності інтерфейсу середовищ СУБД

Зручність інтерфейсу середовищ СУБД	
SQLite	простий, зручний, компактний
MC Access	простий, зручний, зрозумілий
MySQL	простий, зручний, багатофункціональний
MongoDB	простий, зручний, професійний
PostgreSQL	складний, незручний

У таблиці 5 наведені особисті враження авторів від початкової роботи з розглянутими СУБД.

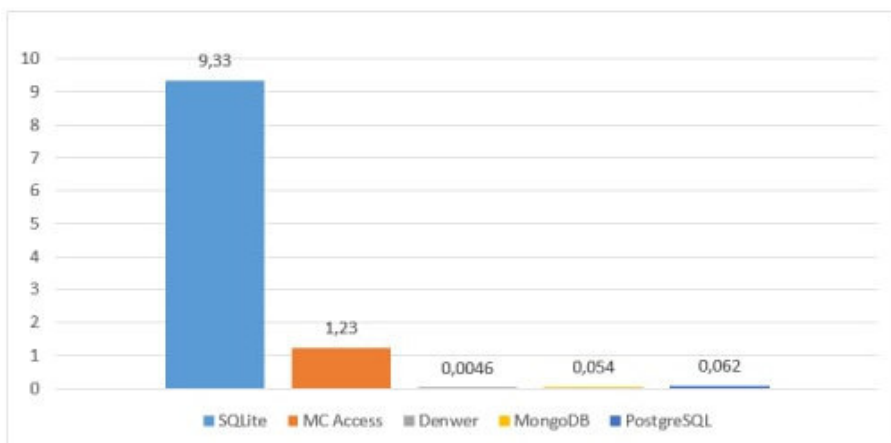


Рис. 7. Діаграма порівняння швидкості виконання sql-запитів

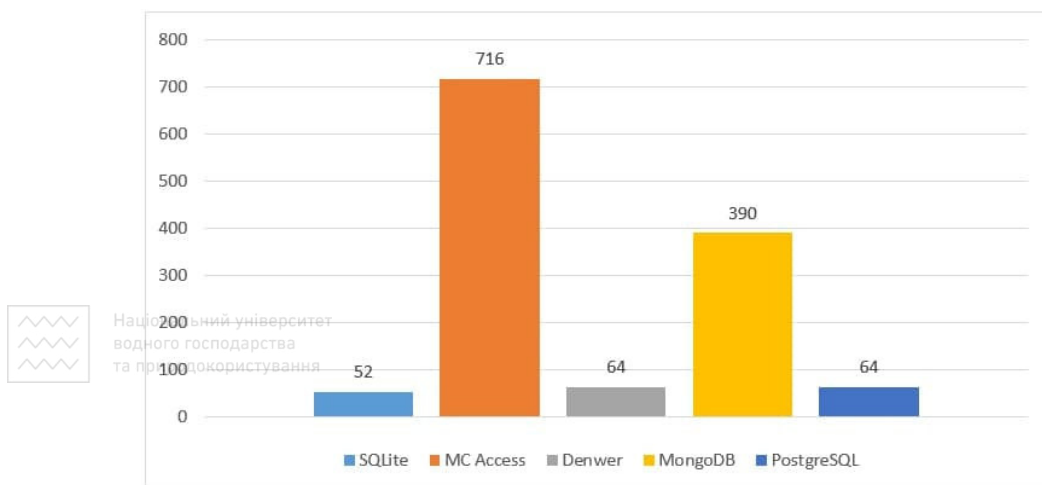


Рис. 8. Діаграма порівняння розмірів БД

На рис. 7–8 наведені порівняльні діаграми СУБД за найважливішими параметрами, а саме: розмірами створеної БД та швидкості виконання SQL-запитів.

Отже, у загальному, зважаючи на результат тестування, можна зробити висновки, що найменше місця займає установщик Denwer`а (8,7 Мб), а найбільше – MongoDB (270,5 Мб). З таблиці 2 можна зрозуміти, що найменший розмір програми має MC Access (15,4 Мб), а найбільший – MongoDB (1,7 Гб). Найменший розмір бази даних має SQLite (52 кб), а найбільший – MC Access (716 кб). Найбільшу швидкість виконання sql-запитів має Denwer (0,0046 с), найменшу – SQLite (9,33 с). Простий та зручний інтерфейс мають усі середовища, окрім PostgreSQL. Враховуючи всі вищенаведені показники, можна зробити висновок, що найкращим середовищем для роботи з базами даних є програмний пакет Denwer, що містить СУБД MySQL, а найгіршими параметрами характеризуються MC Access та MongoDB.

Висновки. Була здійснена перевірка найпопулярніших СУБД за такими критеріями, як швидкість виконання sql-запитів, розміри бази даних, установщика та середовища у кожній СУБД. Кожне середовище має свої плюси та мінуси, проте, зважаючи на наші тестування, можна зазначити, що найкраще проявив себе Denwer. Попри те, що SQLite є найповільнішим у виконанні sql-запитів, він має непогані показники розмірів програми середовища та бази даних. PostgreSQL має незручний інтерфейс та посередній розмір середовища, однак є доволі швидким, стосовно виконання sql-запитів. Щодо найгірших середовищ у результаті досліджень, MC Access має деякі обмеження при створенні sql-запитів, а також низьку швидкість, тому не рекомендується його використовувати при роботі з SQL. Стосовно MongoDB, програма має найбільший розмір установщика та, власне, самого середовища СУБД, тому її краще не використовувати на ПК, що мають обмежену кількість пам'яті. Отримані результати можуть використовуватися для підбору оптимального СУБД, а також для подальших досліджень інших наявних СУБД та більш детального ознайомлення з їх характеристиками.

1. Новиков Б. А., Левин М. Ю. Сравнительный анализ производительности SQL и NoSQL СУБД : монография. Санкт-Петербург, Эспоо. 2. Мелешко О. О., Шмир М. А. Аналіз методів захисту інформації в нереляційній базі даних MongoDB : монографія. Київ. 3. Левченко О. Ю. Тестування баз даних корпоративних інформаційних систем : монографія. Донецьк.

Sydor A. I., Candidate of Engineering (Ph.D.), Senior Lecturer; Kwaterchuk M. O., Senior Student; Zyschuk M. V., Senior Student (National University of Water and Environmental Engineering, Rivne)

TESTING AND ANALYSIS OF THE BASIC CHARACTERISTICS OF DATABASE MANAGEMENT SYSTEMS

The work is devoted to the study of the main characteristics of the database. The review of the most popular database management systems is carried out, the peculiarities of DBMS setup are investigated, the comparative analysis of dimension of databases and speed of their processing depending on the used environment is carried out.

Keywords: databases (DB), database management systems (DBMS), SQL, sql-queries, testing.

УДК 004.658

Козлюк С. М., ст. викладач; Подобайло А. Л., студент (Національний університет водного господарства та природокористування, м. Рівне)

СИСТЕМА GPS-МОНІТОРИНГУ ОБ'ЄКТІВ З ФУНКЦІЄЮ ВІДДАЛЕНОГО КЕРУВАННЯ

Робота присвячена створенню системи GPS-моніторингу об'єктів з функціями віддаленого керування. Проведено огляд актуальності даної системи для сфери оренди автомобілів. Розглянуто і обґрунтовано вибір технологій які використовуються при розробці системи. Наведений приклад апаратних засобів за допомогою яких проводиться моніторинг об'єктів. Також проведений огляд систем які мають схоже призначення.

Ключові слова: Carsharing, GPS-моніторинг, GPS трекер, OpenStreetMaps, Django, React, PostgreSQL, Clickhouse.

Актуальність роботи. В сучасному світі все більшої популярності набувають послуги спільного користування рухомим та нерухомим майном. Зараз люди мають вибір — купувати майно чи просто орендувати об'єкти спільного користування. Шаленою популярності користуються сервіси здачі житла такі як AirBNB, Booking і т.п.

Також активно набирають популярність сервіси похвилинної здачі автомобілів — так званий Каршеринг. Даний сервіс дозволяє заощаджувати час та гроші за рахунок спільного користування транспортними засобами і відсутності необхідності утримувати та обслуговувати авто.

Актуальність такого сервісу виросла в зв'язку з обмеженнями на доступ до громадських транспортних засобів. Тому метою даної роботи є розробка системи моніторингу та контролю транспортним засобом для надання можливості дистанційної оренди без залучення третіх осіб.

Ідея каршерингу не є новітньою у світі. В класичному варіанті каршеринг представляє володіння автомобілем разом з іншою людиною або людьми, тобто володіння часткою авто з правом користуватися. Модель каршерингу покликана швидко та зручно задовольнити традиційну потребу людини пересуватися з одного місця в інше, а також дозволяє економити власні кошти за рахунок зменшення витрат на одноосібне володіння авто [1].

Вагомою перевагою використання каршерингу є заощадження до 70% вартості транспорту, через те що вони платять тільки за час використання автомобіля. Люди вважають що ця послуга може замінити таксі чи навіть власний автомобіль. Це скоріше через те, що люди не використовують автомобіль щоденно і тому дешевшим буде використання каршерингу, який повністю задовольняє потреби автомобілю на час, коли він дійсно необхідний [1].

Аналіз вимог. Першим кроком для успішної реалізації проекту необхідно сформулювати вимоги, які повинні бути реалізовані в системі. Проаналізувавши систему і те що вона повинна включати, було сформовано наступний функціонал:

- Авторизація користувачів системи. Здійснювати аутентифікацію користувачів через Google та за допомогою власної авторизації по основних полях, а саме: номер телефону або e-mail та пароль.
- Виконувати прив'язку об'єктів до певних груп або відділів.
- Збереження всіх даних з пристроїв, встановлених на об'єктах у стовпчикову базу даних.
- Збереження всіх дій користувачів в реляційну базу даних.
- Оновлювати дані кожні 5 секунд, для отримання та відображення актуальної інформації про об'єкти.
- Надсилання смс/email сповіщень;
- Можливість відсилання дистанційних команд для керування на об'єкти.

Для успішної реалізації проекту, необхідно на початку узгодити вибір технологій, що будуть застосовуватись при розробці. Найкращим варіантом є стек Python/Django для серверної частини, оскільки цей фреймворк дозволяє з швидко побудувати структуру проекту та є досить надійним та розширюваним.

Ця технологія з «коробки», тобто після створення пустого проекту, отримує адміністративну панель, через яку можна керувати, з легкістю додавати та редагувати дані, які будуть отримуватись клієнтом.

Окрім цього одразу ж присутній захист від SQL ін'єкцій. Впровадження SQL, залежно від типу СКБД та умов впровадження, може дати можливість атакуючому виконати довільний запит до бази даних (наприклад, прочитати вміст будь-яких таблиць, видалити, змінити або додати дані), отримати можливість читання та/або запису локальних файлів та виконання довільних команд на сервері [2].

Django підтримує і інші технології захисту від міжсайтового скриптингу (XSS), який може завдати шкоди, якщо сайт має можливість виконувати скрипти клієнта, міжсайтова підробка запиту (CSRF), який завдає шкоди методом відправлення

запиту від імені користувача на сайті де останній аутентифікований, та клікджекінгу (Clickjacking), який розуміється як викрадення кліку, що дозволяє виконати клік на сайті від імені користувача сторінки.

Також окрім безпеки, фреймворк містить вбудовану аутентифікацію, яка одразу ж підтримує керування акаунтами користувачів, контроль доступу до об'єктів, авторизацію в системі.

Цей вибір аргументований великою кількістю додатків розроблених за допомогою цієї технології, оскільки вона дозволяє швидко реалізувати прототип.

Для роботи з асинхронними задачами, тобто запуск певних дій не в реальному часі, було використано служби Redis/Celery.

Celery - це програмний пакет для організації черг завдань на основі Python, який дозволяє виконувати асинхронні завдання, засновані на інформації, що міститься в повідомленнях, які генеруються в коді програми (в нашому випадку Django), призначеному для черги завдань Celery. Celery також може бути використаний для виконання повторюваних, періодичних (тобто запланованих) завдань [10].

Celery найкраще використовувати в поєднанні з рішенням для зберігання повідомлень, яке часто називають брокером повідомлень. Поширеним посередником повідомлень, який використовується з celery, є Redis, який є швидкодіючим сховищем даних по типу ключ-значення. Redis також служить сховищем результатів, що надходять з черг celery, які потім витягуються споживачами черги [10].

Також в додатку використовуються публічні кінцеві точки (endpoint), які відкриті для всіх і через які будуть отримуватись дані з трекерів, встановлених на об'єктах. Для взаємодії серверної і клієнтської частин застосовується мережа GPRS мобільного оператора, а на сервері встановлено сервіс, який приймає ці запити.

Як інтерфейс взаємодії між сервером і клієнтом, використовується архітектурний підхід REST. Особливість REST в тому, що сервер не запам'ятовує стан користувача між запитами – в кожному запиті передається інформація, яка ідентифікує користувача (наприклад, token, отриманий через OAuth-авторизацію) і все параметри необхідні для виконання операції.

Аутентифікацію і контроль користувачів було розроблено за допомогою протоколу OAuth2, який вимагає підтвердження зі сторони Google. Тобто реалізовано вхід за допомогою google акаунта та за допомогою власної аутентифікації на основі email, номеру телефону та пароллю.

OAuth2 являє собою фреймворк для авторизації, що дозволяє додаткам здійснювати обмежений доступ до призначених для користувача акаунтів на HTTP сервісах. Він працює за принципом делегування аутентифікації користувача сервісу, на якому знаходиться акаунт користувача, дозволяючи сторонньому додатку отримувати доступ до акаунту користувача. OAuth2 працює в інтернеті, на десктопних і мобільних додатках.

Систему інтегровано з сервісом сповіщення користувачів TurboSMS, який надає послуги СМС розсилок. За допомогою цього сервісу буде здійснюватися сповіщення клієнтів і користувачів системи. Для імейл інформування було створено власну систему розсилання електронних листів.

Для реалізації клієнтської частини слушним буде обрати бібліотеку React, яка оптимізує оновлення веб сторінки, за допомогою віртуального DOM дерева. Працює бібліотека за наступним алгоритмом, якщо є зміни тільки в одному місці сторінки, то вона оновлюється тільки змінена частина, не оновлюючи весь контент, що є вагомим показником для користувачів, так як це напряму впливає на враження від перебування у системі.

Для організації приємного зовнішнього вигляду додатку, було використано бібліотеку Material-UI, що включає набір з поширених React компонентів, для швидкої розробки інтерфейсів. А для організації збереження «стану» додатку на клієнтській частині, було застосовано бібліотеку Mobx, котра з легкістю інтегрується із основною технологією.

Також важливим пунктом реалізації системи являється маршрутизації додатку. Для впровадження цієї функції використовувалась бібліотека react-router, яка дозволяє використовувати лише один HTML-документ і організовує взаємодію з користувачем за допомогою динамічного завантаження HTML, CSS, JavaScript.

В якості бази даних вибрана комбінація реляційної бази даних PostgreSQL та колоночної Clickhouse. Для збереження інформаційних моделей системи використана надійна та функціональна об'єктно-орієнтована база даних з великим діапазоном типів даних PostgreSQL. Для збереження великої кількості однотипної інформації та швидкого доступу до цих даних та їхньої оптимізованої вибірки, доречним буде використовувати стовпчикову систему управління базою даних. В цьому проекті використано ClickHouse.

PostgreSQL — СУБД PostgreSQL використовує для своїх баз даних реляційну модель, підтримує стандартний мовний запит SQL. При цьому PostgreSQL пропонує широкий спектр можливостей. Можна сказати, що Postgres володіє майже всіма можливостями, що існують в інших базах даних (як комерційних, так і з відкритим кодом), а також поряд із додатковими. [3]

ClickHouse – стовпчикова система управління базами даних для онлайн обробки аналітичних запитів (OLAP). Ключові особливості OLAP сценарію роботи:

- Переважна більшість запитів на зчитування;
- Дані оновлюються великими об'ємами, а не по одному рядку;
- Дані додаються в базу даних, але не змінюються;
- Запити відбуваються не часто;
- Результат виконання запиту суттєво нижчий за вихідні дані, тобто дані фільтруються [4].

Для роботи з картою було обрано сервіс OpenStreetMap. Оскільки існують компоненти для взаємодії між сервісом відображення карти і технологією React це хороший вибір, оскільки інші API не можуть надати такого функціоналу або ж є платними.

OpenStreetMap надає картографічні дані для тисяч веб-сайтів, мобільних застосунків та різних пристроїв. Створений спільнотою маперів, які вносять і підтримують дані про дороги, стежки, кав'ярні, вокзали і багато чого іншого по всьому світу. Також OpenStreetMap надає виразності місцевим даним. Мапери використовують аерофотознімки, GPS пристрої та прості створені на місцевості схеми

для перевірки точності та актуальності даних OSM [5]. А приклад роботи представлено на рис. 1.

Для роботи із картами за допомогою мови програмування JavaScript необхідно скористатись бібліотекою Leaflet, що дозволяє виставляти маркери, будувати лінії, за допомогою яких і будуються маршрути, виділяти певні області, тощо.

Для того щоб поєднати Leaflet із картами OpenStreetMap, було застосовано бібліотеку React Leaflet, що служить містком між нативними JavaScript викликами та реалізацією додатків за допомогою React.

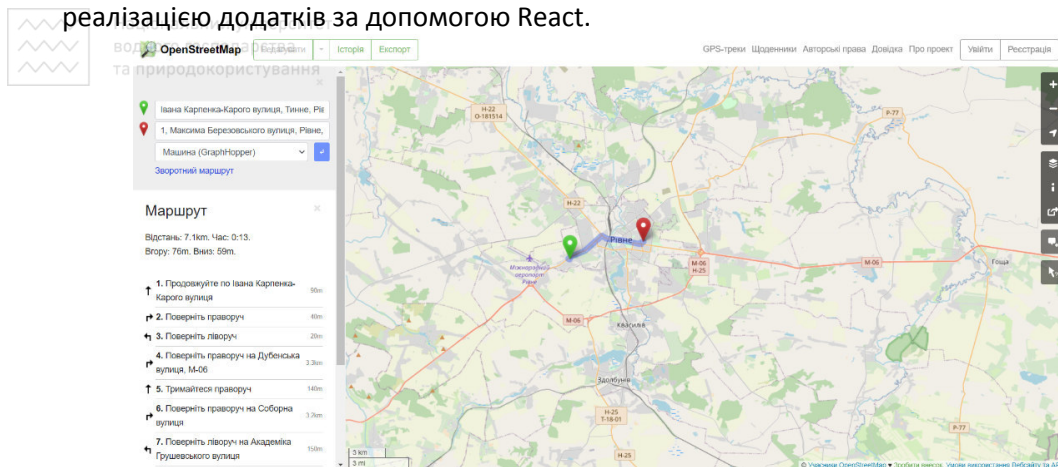


Рис. 1. Приклад роботи побудови маршруту на сайті openstreetmap.org

Для відслідковування об'єкта необхідно обладнання спроможне виконати цю функцію. Для цього було використано прилад FMB130 (рис. 2) – GPRS / GNSS трекер з гнучкою настроюванням входів, внутрішніми GNSS / GSM антенами і резервною батареєю, що дозволить здійснювати передачу місцезросташування об'єкту на сервер.

Teltonika FMB130 – СПЕЦІАЛЬНИЙ трекер з внутрішніми GNSS і GSM антенами, налаштованими цифровим / аналоговим входами, цифровими виходами, імпульсними входами, а також з налаштованим негативним / позитивним входом, плюс підтримка Bluetooth і резервний акумулятор. FMB130 призначений для стеження за легковими автомобілями, а також для більш складних завдань, таких як логістика, служби доставки, комунальні служби і багато іншого. Дуже добре підходить для роботи з холодильними установками, так як має збільшений набір входів / виходів і 1-wire інтерфейс для контролю температури. До FMB130 також можна підключати CAN адаптери. За допомогою CAN адаптерів можна зчитувати дані не тільки з легкових машин, а також сільгосптехніки, будівельної техніки, вантажної техніки. Також є підтримка рішення для каршеринга [6].



Національний університет
водного господарства
та природокористування



Рис. 2. Зображення FMB130 із сайту teltonika-gps.com

Також необхідною складовою системи є відображення даних об'єкту, що є не менш важливим як розташування. Цєю функцією буде займатись прилад CAN-CONTROL (рис. 3) – адаптер для каршерингу.

Найбільш широко використовувана функція CAN-CONTROL – можливість на відстані замикати / відкривати автомобіль і управляти вікнами за допомогою спеціальних команд SMS / GPRS. Це робить аксесуар CAN-CONTROL ідеальним рішенням для каршерингу і багатьох інших випадків використання. На додаток до цього CAN-CONTROL може зчитувати важливі параметри CAN, такі як: рівень палива, одометр, VIN номер, витрата палива, обороти двигуна, температуру двигуна і положення ручного гальма. Завдяки цьому, Ви зможете ефективно визначити галузі поліпшення, щоб мінімізувати накладні витрати на технічне обслуговування автомобіля. CAN-CONTROL підключається безпосередньо до CAN-шини автомобіля, що означає, що немає необхідності використовувати додаткові реле або дорогі автомобільні системи безпеки для дистанційного керування дверима. Пристрій підтримує понад 1300 автомобілів, і це число постійно зростає. Teltonika надає схеми підключення для кожного підтримуваного автомобіля, що допомагає зробити монтаж найбільш простим і швидким [7].



Рис. 3. Зображення Can-control із сайту teltonika-gps.com

Огляд аналогів. OpenGTS («Відкрита система відстеження GPS») – це перший доступний проект з відкритим кодом, розроблений спеціально для надання послуг

веб-відстеження GPS для «парку» транспортних засобів. Демо огляд системи зображено на рис. 4.

Переваги цієї системи:

- Веб-автентифікація: кожен обліковий запис може підтримувати декількох користувачів, і кожен користувач має власний пароль для входу та контрольований доступ до розділів у своєму обліковому записі.
- Підтримує OpenLayers/OpenStreetMap, не обмежуючись Google Maps, Microsoft Virtual Earth, і Mapstraction [8].

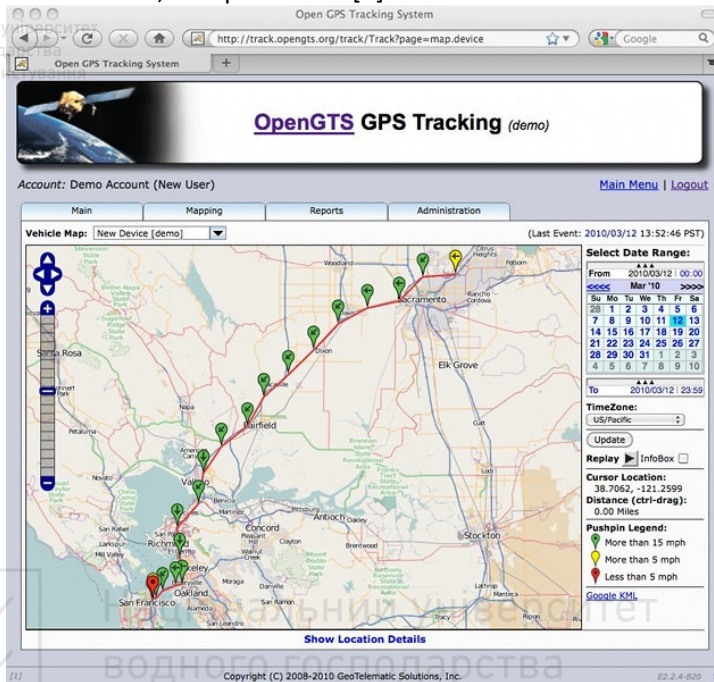


Рис. 4. Демо огляд GPS відслідкування в системі OpenGTS

Ruhavik сервіс для контролю власних поїздок, демонстрація якого відображена на рисунку 5, який підтримує:

- Перегляд подорожей на карті;
- Побудова діаграми зміни швидкості;
- Отримання найбільшої та середньої швидкості;
- Перегляну перелік подій;

Також сервіс реалізував налаштування та отримання сповіщень коли:

- Автомобіль починає та закінчує свій маршрут;
- Автомобіль входить чи покидає певну зону;
- Автомобіль перевищує вказаний ліміт швидкості;
- Автомобіль надсилає сигнал тривоги;
- Автомобіль отримує помилки від бортового комп'ютера [9].

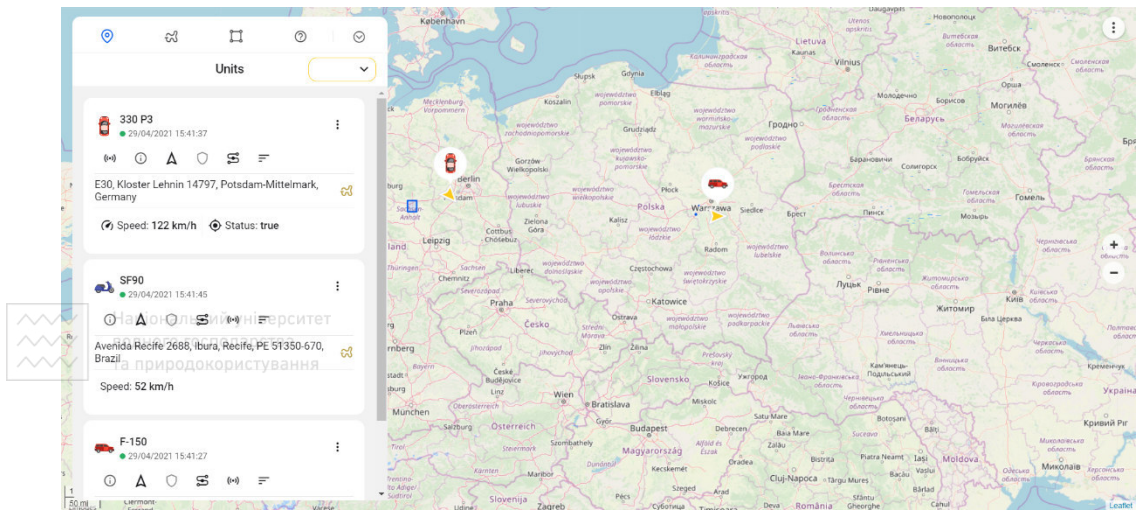


Рис. 5. Демонстрація системи Ruhavik

Висновки. Система підтримує відслідкування об'єктів, відображення їхніх маршрутів та інформацію про об'єкт на кожне оновлення даних, що зберігається в стовпчиковій системі керування базою даних. Система підтримує передачу команд на обладнання встановлене на об'єктах, що дозволяє здійснювати віддалене керування. Також здійснюється оновлення всіх даних кожні 5 секунд та зберігаються дії користувачів. Реалізовано зручний інтерфейс для отримання інформації об'єктів відслідковування та система акаунтів з різними доступами до системи.

1. Каршеринг – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Каршеринг> (дата звернення: 26.04.2021).
2. SQL-ін'єкція – Вікіпедія. URL: <https://uk.wikipedia.org/wiki/SQL-ін'єкція> (дата звернення: 27.04.2021).
3. PostgreSQL Documentation. URL: <https://www.postgresql.org/docs/13/index.html> (дата звернення: 27.04.2021).
4. ClickHouse Documention. URL: <https://clickhouse.tech/docs/en/> (дата звернення: 27.04.2021).
5. OpenStreetMap. URL: <https://www.openstreetmap.org/about> (дата звернення: 29.04.2021).
6. 2G Bluetooth GPS tracker FMB130. URL: <https://teltonika-gps.com/product/fmb130/> (дата звернення: 29.04.2021).
7. CAN-CONTROL – CAN Adapter For Car Sharing. URL: <https://teltonika-gps.com/product/can-control/> (дата звернення: 29.04.2021).
8. GPS Tracking: Open-Source GPS Tracking System – OpenGTS. URL: <http://www.opengts.org/> (дата звернення: 29.04.2021).
9. Ruhavik. URL: <https://gurtam.space/ruhavik> (дата звернення: 30.04.2021).
10. Асинхронні задачі в Django з Redis і Celery. URL: <https://webdevblog.ru/asinhronnye-zadachi-v-django-s-redis-i-celery/> (дата звернення: 03.05.2021).

Kozliuk S. M., Senior Lecturer; Podobailo A. L., Senior Student (National University of Water and Environmental Engineering, Rivne)

GPS-MONITORING SYSTEM OF OBJECTS WITH REMOTE CONTROL FUNCTION

The work is devoted to the creation of a GPS-monitoring system of objects with remote control functions. The article reviews the relevance of this system for the field of car rental. The analysis of the choice of technologies used in system development is carried out. Here is an example of hardware that is used to monitor objects. A review of systems with a similar purpose was also conducted.

Keywords: Carsharing, GPS-monitoring, GPS tracker, OpenStreetMaps, Django, React, PostgreSQL, Clickhouse.

УДК 681.5:631.5

Сидор А. І., к.т.н, старший викладач; Алтунін Є. А., студент (Національний університет водного господарства та природокористування, м. Рівне)

АНАЛІЗ ТЕХНІЧНИХ СПОСОБІВ КЕРУВАННЯ ПРИСТРОЯМИ ПІДТРИМКИ МІКРОКЛІМАТУ В ТЕПЛИЦІ

Робота присвячена дослідженню основних складових оптимального мікроклімату в теплиці та способів економного керування пристроями що задають та підтримують мікроклімат.

Ключові слова: система автоматичного керування, теплиця, тепличні господарства, енергоефективність, реле, ШІМ, Arduino.

Вступ. Теплиці є однією із важливих частин сільського господарства будь-якої країни. Україна не є виключенням. У зимово-весняний період року потреба людей в свіжих овочах, більшою мірою, задовольняється за рахунок вирощування останніх в спорудах закритого ґрунту – теплицях. Спеціально обладнане укриття дає змогу збирати свіжі овочі навіть в холодну пору року, коли людським організмам особливо потрібні вітаміни. Для підприємців такі методи дозволяють заробляти на вирощуванні овочів. Тому, вдосконалення технологій вирощування сільськогосподарських культур в теплицях є важливим завданням для отримання врожайності високої якості та забезпечення продуктами харчування населення країни.

Згідно з інформацією статей [3–4], комерційні скляні теплиці є не лише високотехнологічними виробничими приміщеннями для вирощування овочів або зелені, але, і досить дорогими в реалізації. Скляні теплиці заповнені обладнанням, включаючи екрануючими установками, системами опалення і охолодження, а також освітлення. В промислових масштабах основними джерелами затрат при експлуатації таких теплиць є забезпечення достатнього рівня освітлення рослин та підтримка оптимальної температури повітря та ґрунту.

Також, в наші дні, у зв'язку із збільшенням кількості людей що починають замислюватись над безпечністю магазинних продуктів, досить популярними стають невеликі індивідуальні тепличні господарства, в яких люди власноруч можуть вирощувати, для особистого споживання, абсолютно екологічно чисті овочі та зелень. В таких теплицях найважливішим параметром є енергоефективність, оскільки, немає сенсу вирощувати, хоча і абсолютно екологічно чисті, але значно дорожчі ніж магазинні, овочі та зелень. Для економічного використання досить дорогих ресурсів, необхідно запроваджувати системи автоматичного керування мікрокліматом теплиць, які будуть використовувати електроенергію та воду лише в необхідних для рослин кількостях.

Огляд основних складових оптимального мікроклімату теплиці. Теплиця це споруда закритого ґрунту зі своєю внутрішньою агроєкосистемою яка не піддається негативному впливу зовнішнього середовища. Схематичне зображення теплиці показано на рис. 1:

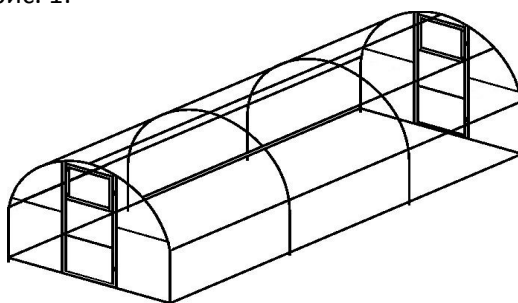


Рис. 1. Схематичне зображення теплиці

На процес росту та розвитку рослин впливають кілька чинників: кількість світла, міра вологості ґрунту та повітря, а також температура всередині теплиці. Побудова надійної системи автоматичного підтримання мікроклімату в теплиці є комплексним і досить складним завданням, адже від саме від якості його підтримання залежить урожайність. В основі такої «розумної» теплиці повинен бути мікроконтролер, який отримує покази датчиків і, на основі отриманих даних, здійснює керування параметрами мікроклімату. До таких параметрів належать:

- температура для кожної рослини є сприятливий діапазон температур при яких вона повноцінно може зростати та плодоносити;
- освітлення за недостатньої інтенсивності природного сонячного світла, необхідно компенсувати його нестачу за допомогою штучних джерел світла або ж навпаки, за надлишкової сонячної освітленості створення штучної тіні;
- вологість (як повітря, так і ґрунту) оскільки теплиця, зазвичай, унеможливорює потрапляння природних опадів, необхідно не просто лити воду, а контролювати вологість ґрунту і, за потреби, здійснювати зрошення;
- повітря в залежності від періоду доби, рослини потребують в складі повітря збільшення вуглекислого газу або кисню; Також необхідно забезпечувати циркуляцію повітря всередині самої теплиці [3].

Полив в теплицях. Системи поливу теплиць поділяються на дві групи за принципом подачі води: дощування і крапельний полив. В першому випадку вода проходить через спеціальні краплеутворювальні пристрої – дощовики, що

розбризкують воду на рослини зверху. Автоматичний полив у теплиці, влаштований за принципом крапельної подачі води, більш раціональний, ніж дощування. За такого способу волога надходить на пряму до коріння рослини. Це дозволяє дозовано зрошувати ґрунт, істотно знижуючи витрату води і електроенергії (порівняно з іншими способами поливу). Кількість точок зволоження не потрібно визначати заздалегідь, їх установлюють залежно від типу ґрунту і потреби тієї чи іншої культури у волозі. Крапельна автоматична система поливу використовується також для внесення кореневого підживлення. Розчинені у воді мінеральні добрива заливають у спеціальні резервуари, що приєднуються до труби [2; 7].

Головними силовими елементами систем поливу можуть бути представлені електронасосами та електроклапани різної потужності. Рівень вологості ґрунту та повітря визначають датчики вологості і забезпечують економне використання води та електроенергії.

Опалення теплиць. Однією із актуальних тепличних проблем у холодну пору є обігрів. Найпоширеніші види обігріву – повітряне (тепловентилятори або газові обігрівачі), водяне, за допомогою інфрачервоних обігрівачів. В обігріві полікарбонатних теплиць добре зарекомендували себе інфрачервоні обігрівачі та «тепла підлога». Це екологічний і нешкідливий спосіб опалення. Він безпечний і для людей, і для рослин. Основне завдання системи опалення – це підтримка заданої температури. Найвигіднішим вважають водяне опалення. Як правило, опалення теплиці складається з декількох роздільних контурів (систем). Регулювання температури повітря для водного опалення здійснюється шляхом зміни температури води в контурах за допомогою клапана, який змішує в необхідній пропорції воду від джерела тепла (прямий теплоносій) з водою, яка повернулася з теплиці (зворотним теплоносієм) [2; 7].

В процесі росту і дозрівання температура внутрішнього повітря в теплиці має змінюватися залежно від виду овочів та від часу доби. Для огірків, наприклад, температура повітря в нічний час має бути близько 18° С, а в денний час близько 22° С [2].

Зазвичай, керування потужністю таких пристроїв як тепловентиляторів, інфрачервоних та газових обігрівачів, може здійснювати вбудованими в них регуляторами. В системах водяного обігріву керувати температурою теплоносія (води) можна за допомогою електроклапанів в трубах теплопроводу. Замір температури здійснюється датчиками температури і при досяганні необхідної температури обігрівачі вимикаються та не використовують зайву електроенергію.

Вентиляція в теплицях. Вентиляція є обов'язковою для створення оптимального мікроклімату в теплиці. За допомогою вентиляції можна вирішити проблему появи плісняви і грибкових захворювань рослин та створити унікальний мікроклімат, необхідний для активного росту рослин. Для вентилявання повітря всередині теплиці в покрівлі передбачаються фрамуги, площа яких має становити до 20% від загальної площі. Відкриття та закриття фрамуг проводиться за допомогою моторизованих редукторів або термопоршнів. Примусова вентиляція в теплиці також може здійснюватися за допомогою вентиляторів, що створюють переміщення повітря в закритому приміщенні. Вентилятори працюють за двопозиційним законом керування (рис. 2), згідно показників датчиків вологості та температури.

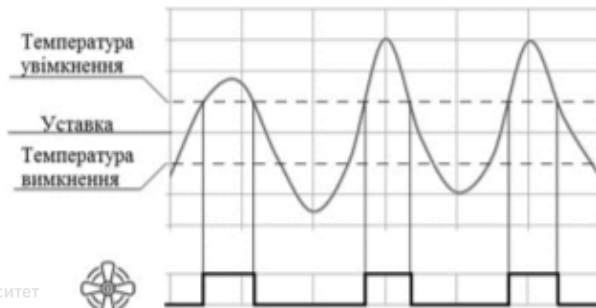


Рис. 2. Закон керування вентиляцією в теплиці

Вентилятори встановлюються як усередині теплиці, так і зовні. Зовнішні вентилятори потрібні в зимовий період часу, коли не можна відкривати фрамуги через низьку зовнішню температуру, щоб не заморозити рослини. Їх встановлюють у верхній частині теплиць. За допомогою цих вентиляторів холодне повітря потрапляє у верхню частину теплиці, перемішується з гарячим повітрям, таким чином знижуючи температуру в теплиці [2; 7].

Освітлення теплиць. Найбільш гостро постає проблема забезпечення достатньої освітленості рослин саме в зимовий час, коли світловий день значно коротший. Якщо рослини будуть отримувати світло менше 10 годин на добу, їхній ріст уповільниться, а це є неприйнятним, особливо для вирощування розсади. З цієї причини для забезпечення достатньої освітленості в теплицях застосовують лампи різного типу (LED, люмінесцентні, ртутні, натрієві), що включаються вранці і ввечері, продовжуючи для рослин світловий день. Регулювання освітленістю може здійснюватися згідно показів датчика зовнішнього освітлення, коли поточне значення датчика менше за заданий рівень, або за розкладом [2; 7]. Завдяки увімкненню джерел світла лише в потрібний момент та на необхідний час може суттєво скоротити споживання електроенергії.

Способи керування головними пристроями що задають та підтримують мікроклімат теплиць. Як відомо, щоб замкнути електричне коло і змусити текти струм від джерела до споживача використовуються механічні вимикачі. В нормальному стані вимикач вимкнений, а щоб перевести його в увімкнений стан, необхідно щоб хтось натиснув на кнопку, яка в свою чергу замкне електричне коло. В даному випадку, так звану комутацію кола (процес замикання та розмикання електричного кола), здійснює людина.

Струм також може здійснювати комутацію електричних кіл. Для таких цілей використовують електричні та електромеханічні вимикачі. В них, за допомогою слабких керуючих електричних сигналів відбувається керування великими струмами. Керуючий сигнал має два значення «0» і «1». В електроніці найчастіше за одиницю вважають струм з напругою 5В, а за нуль – відповідно 0В. Такі сигнали також називають цифровими. Саме за допомогою таких цифрових сигналів можна керувати електромагнітними реле, твердотільними реле, які в свою чергу, будуть виступати в ролі вимикачів.

Електромагнітні реле. Найпростішим способом керування, наприклад, температурою в приміщенні, є вмикання нагрівального приладу на певний час та вимикання його, при досягненні необхідної температури, за допомогою електромагнітного реле. Реле, зазвичай, складається з трьох основних складових: сприймального, проміжного та виконавчого елементів. Сприймальний (первинний) елемент, визначає контрольовану величину та перетворює її на іншу фізичну величину. Проміжний елемент, порівнює значення цієї величини із заданим значенням і, у разі його перевищення, передає первинний вплив, на виконавчий елемент. Виконавчий елемент передає вплив від реле у керовані електричні кола. Ці елементи може бути виконано самостійно, або об'єднано між собою.

Електромагнітні реле (англ. electromagnetic relay) – це електричні реле, у яких призначена реакція створюється через вплив електромагнітних сил. В цих реле (рис. 3) елементом керування є котушка реле (relay coil) 1, яка створює магнітний потік у магнітопроводі. Рухома частина 2 (проміжний елемент реле) при певному значенні напруги U , прикладеної до обмотки, починає рухатися і сприяє перемиканню контактів 3 (комутаційний елемент реле).

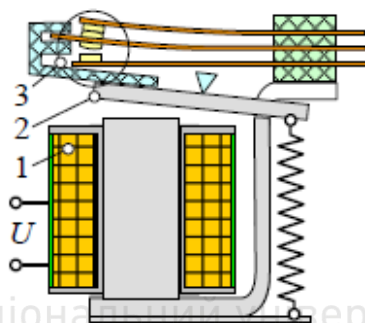


Рис. 3. Будова електромагнітного реле [10]

Головною перевагою використання реле є те, що ми можемо подавши всього 5В на котушку, замкнути дуже потужного навантаження і при цьому коло навантаження буде повністю гальванічно розв'язаним з керуючим колом.

Електромагнітні реле розділяють:

- за принципом дії:
 - нормально відкритий (Normally Open або NO) – при надходженні керуючого сигналу реле замикає коло навантаження;
 - нормально розімкнені (Normally Closed або NC) – при надходженні керуючого сигналу 5В реле розмикає коло навантаження.
- за способом керування:
 - сигналом високого рівня (+5В);
 - сигналом низького рівня (-5В).

В електричних пристроях «голі» реле, без спеціальної електронної обв'язки, не використовуються. Головними недоліками електромагнітних реле є достатньо мала максимальна швидкість спрацювання виконавчого елемента (не перевищує кількох десятків Гц) та значне іскріння контактів. Реле можуть керувати практично будь-чим, але проблеми можуть виникати саме з індуктивним навантаженням

(електродвигуни, соленоїди), причому як постійного, так і змінного струму. При різкому вмиканні і відмиканні індуктивного навантаження створюється викид, напруга якого може в кілька разів перевищувати напругу живлення кола. Для захисту решти електронних компонентів, використовують оптрони, діоди та, так звані, снаббери [6].

Твердотільні реле. Твердотільне реле (англ. solid-state relay, SSR) — електричний комутаційний апарат типу реле на базі електронних, магнітних, оптичних або інших компонентів без механічно рухомих частин, що слугує для вмикання/вимикання потужних кіл за допомогою низьких напруг (струмів), які подаються на контакти керування. Цей тип реле може використовуватись як в колах постійного, так і в колах змінного струму.

Серійні твердотільні реле, для керування змінними струмами, використовують напівпровідникові пристрої, такі як тиристори та семістори, а для постійних струмів – польові транзистори, Дані напівпровідникові компоненти забезпечують можливість перемикання струмів величиною до сотень ампер. Твердотільні реле мають суттєво вищу швидкість перемикання, ніж електромеханічні реле (до кількох сотень Гц) та також є повністю гальванічно розв'язаними.

Даний вид реле у меншій мірі, ніж їхні електро-механічні аналоги, пристосовані до витримування короточасних перевантажень та мають дещо більший опір у замкнутому стані. Також твердотільні реле не можуть використовуватись для перемикання між різними контактами [10].

Твердотільні реле для керування постійним струмом (SSR VDC) та змінним струмом (SSR VAC) мають абсолютно однаковий корпус (рис. 4), тому потрібно уважно читати що написано на корпусі щоб їх не переплутати. За основу в них використовується польові транзистори. Також в маркуванні після слова “SSR” звичайно зазначений ток в Амперах, тобто SSR-25 це реле на 25 Ампер. Максимальна напруга вказана поряд з вихідними клеммами. Також на реле постійного струму на вихідних клеммах важливо не переплутати полярність.

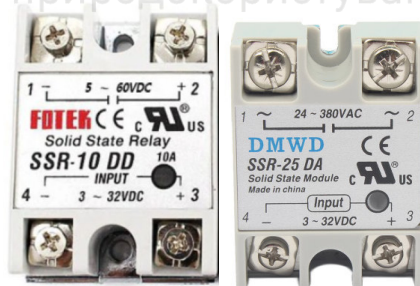


Рис. 4. Твердотільні реле постійного та змінного струму

Важливо враховувати те, що при комутації такими реле кіл з потужністю споживачів більшою за 600–1000 Вт, їхній корпус може сильно нагріватись. Частина корпусу в твердотільних реле представляє собою товсту алюмінієву пластину, що виконує функцію радіатора. Тому, навіть при використанні великих за розмірами твердотільних реле, необхідно використовувати їх разом із додатковими зовнішніми радіаторами.

За допомогою твердотільних реле можна плавно керувати сильно інерційним навантаженням, наприклад, лампою розжарювання (рис. 5) або великим обігрівачем. Для цього потрібно використовувати наднизькочастотний ШІМ (широтно-імпульсно модульований) сигнал в межах кількох сотень Гц. Змінюючи так зване «заповнення» ШІМ сигналу можна змінювати «сумарну» напругу (інтегровану) за деякий період (в даному прикладі – змінюватиметься яскравість лампи розжарювання).

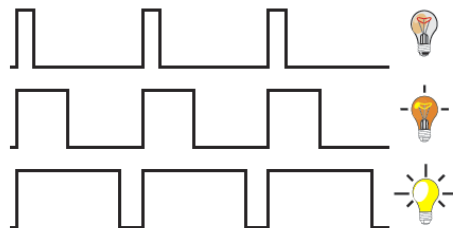


Рис. 5. Модуляція напруги при підключенні лампи розжарювання

Чим більше заповнення ШІМ сигналу, тим вища напруга, але не більше напруги, яку ми модулюємо (рис. 6).

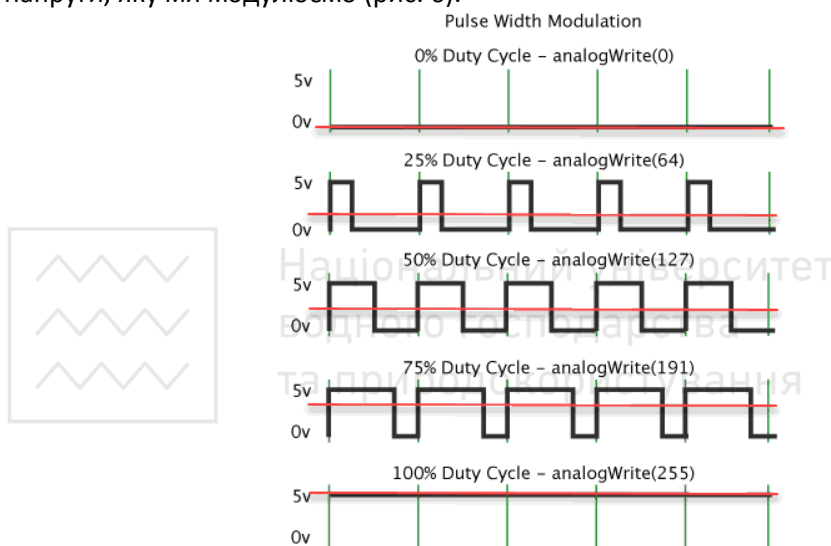


Рис. 6. Залежність зміни «сумарної» напруги від зміни міри заповнення сигналу

Платформа Arduino. Джерелом керуючого цифрового ШІМ сигналу певної частоти, як і у випадку з електромеханічними реле, також можуть виступати будь-які мікроконтролери. На сьогодні, найдоступнішим та найбільш гнучко налаштовуваним мікроконтролером є ATmega 328p, який є «серцем» плат серії Arduino.

Плати Arduino складається з мікроконтролера Atmel AVR, а також елементів обв'язки для програмування та інтеграції з іншими пристроями. На багатьох платах наявний лінійний стабілізатор напруги +5В або +3,3В. Тактування здійснюється на частоті 16 або 8 МГц кварцовим резонатором. У мікроконтролер записаний завантажувач (bootloader), тому зовнішній програматор не потрібен.

На концептуальному рівні усі плати програмуються через RS-232 (послідовне з'єднання), але реалізація даного способу різниться від версії до версії. Новіші плати програмуються через USB, що можливо завдяки мікросхемі конвертера USB-to-Serial FTDI FT232R.

Плати Arduino дозволяють використовувати значну кількість виводів мікроконтролера як вхідні/вихідні контакти у зовнішніх схемах. Наприклад, в одній із найкомпактніших за розміром плат Nano (рис. 7) доступно 14 цифрових входів/виходів, 6 із яких можуть генерувати ШІМ сигнал, і 8 аналогових входів. Ці сигнали доступні на платі через контактні площадки або штирові роз'єми. Також існує багато різних зовнішніх плат розширення, які називаються «shields» («щити»), які приєднуються до плати Arduino через штирові роз'єми [8].

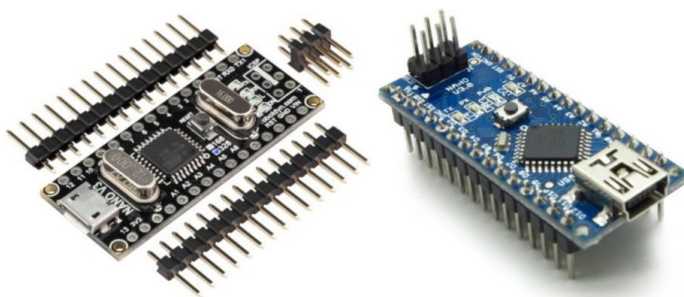


Рис. 7. Плата Arduino Nano

Основні технічні характеристики Arduino Nano:

- Напруга живлення 5 В;
- Вхідне живлення 7–12 В (рекомендоване);
- Кількість цифрових пінів – 14, з них 6 можуть використовуватися в якості виходів ШІМ;
- 8 аналогових входів;
- Максимальний струм цифрового виходу 40 мА;
- Флеш пам'ять 16 або 32 Кб, в залежності від чіпа;
- ОЗУ 1 або 2 Кб, в залежності від чіпа;
- EEPROM 512 байт або 1 Кб;
- Частота 16 МГц;
- Розміри 19 x 42 мм;
- Вага 7 г.

Живлення плати може здійснюватися двома способами:

- Через mini-USB або micro-USB при підключенні до комп'ютера;
- Через зовнішнє джерело живлення, що має напругу 6–20 В з низьким рівнем пульсацій [8].

Написання програм керування платами Arduino здійснюється в Arduino IDE (включає в себе редактор коду, компілятор і модуль передачі прошивки в плату), що засноване на мові Processing. Програми для плат Arduino називаються скетчами і пишуться на мові програмування С або С++. Середовище розробки Arduino поставляється разом із бібліотекою програм «Wiring» (бере початок від проекту

Wiring, який дозволяє робити багато стандартних операцій вводу/виводу набагато простіше).

Висновки. В ході виконання дослідження було виявлено, що вирощування продуктів в теплицях, на сьогодні, є достатньо ефективним способом задовільнення потреб населення України в свіжих овочах, особливо в осінньо-зимово-весняний період. Проведено огляд теплиці як об'єкта автоматизації, описано основні складові оптимального мікроклімату теплиці та способи економного керування пристроями що задають та підтримують мікроклімат за допомогою реле та платформи Arduino. Отримані результати можуть використовуватись для подальших досліджень способів автоматизації вирощування рослин.

1. Бакало О. О., Пилипенко Ю. М. Дослідження автоматизованого контролю вологості ґрунту при вирощуванні сільськогосподарських культур в теплиці. *Мехатроніка, комп'ютерна інженерія та метрологія*. 2018. № 3. С. 1–9.
2. Системы управления микроклиматом теплицы, связь с котельной. URL: <http://www.fito-system.ru/climate-systems>. (дата звернення: 05.05.2021).
3. Теплиця. URL: <https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BF%D0%BB%D0%B8%D1%86%D1%8F>. (дата звернення: 05.05.2021).
4. Катков А. Ю., Ползунов И. В., Петрянин Д. Л. Проблемы построения систем управления микроклиматом теплиц. *Новые информационные технологии в автоматизированных системах*. 2014. № 17. С. 460–463
5. Ружицька В. В. Дослідження теплиці як об'єкта автоматизації та удосконалення системи автоматичного керування мікрокліматом : магістерська дисертація. Київ, 2018. 103 с.
6. Контролер теплиці власними руками. URL: <https://alexgyver.ru/gyvercontrol/> (дата звернення: 06.05.2021).
7. Євсеєнко О. М. Розробка апаратно-програмної системи керування мікрокліматом теплиці *Технічна інженерія*. 2020. № 1. С. 104–109
8. Плата Arduino Nano. URL: <https://arduino-master.ru/platy-arduino/plata-arduino-nano/> (дата звернення: 06.05.2021).
9. Апаратна обчислювальна платформа Arduino. URL: <https://uk.wikipedia.org/wiki/Arduino/> (дата звернення: 06.05.2021).
10. Клименко Б. В. Электричні апарати. Електромеханічна апаратура комутації, керування та захисту. Загальний курс : навч. посіб. Харків : Точка, 2012. 340 с.

Sydor A. I., Candidate of Engineering (Ph.D.), Senior Lecturer; Altunin Y. A., Senior Student (National University of Water and Environmental Engineering, Rivne)

ANALYSIS OF TECHNICAL METHODS OF CONTROL OF MICROCLIMATE SUPPORT DEVICES IN THE GREENHOUSE

The work is devoted to the study of the main components of the optimal microclimate in the greenhouse and ways of economical control of devices that set and maintain the microclimate.

Keywords: automatic control system, greenhouse, greenhouses, energy efficiency, relay, PWM, Arduino.

Козлюк С. М., старший викладач; Ярута А. Г., студент (Національний університет водного господарства та природокористування, м. Рівне)

АНАЛІЗ РОЗДРОБЛЕНОТІ МЕСЕНДЖЕРІВ

Робота присвячена дослідженню причин роздробленості та фрагментованості месенджерів, передумови та причини цього явища.

Ключові слова: месенджер, SMS, Whatsapp, Signal, iMessage, Telegram, Facebook Messenger, Viber.

Вступ. Slack, Signal, Hangouts, Wire, iMessage, Telegram, Facebook Messenger і так далі... Навіщо нам так багато додатків для виконання однієї і тієї ж задачі?

Десятиліття тому наукові фантасти уявляли собі літаючі машини, кухні, які автоматично будуть готувати їжу і можливість зателефонувати кому завгодно на всій планеті та багато чого іншого. Але вони не здогадувалися, що ми опинимося на роздоріжжі месенджерів, маючи на руках нескінченну кількість додатків, призначених для простої відправки тексту.

Спроба відправки тексту перетворилася в розумову гімнастику: цей користувач не використовує iMessage, але він відповість, якщо йому відправити повідомлення по WhatsApp. У іншого є WhatsApp, але він там не відповідає, доведеться використовувати Telegram. Інших можна знайти через Signal, SMS і Facebook Messenger.

У телефоні користувача виникає наступна ситуація: є цілий каталог месенджерів та немає можливості спростити все до одного. Великий вибір в цій області позитивно діє на конкуренцію, проте кожного разу, коли виникає ситуація зв'язатися з іншою людиною, виникає потреба проводити в розумі обчислення: яку програму потрібно вибрати для відправки повідомлення іншому користувачеві? Як сучасному користувачеві вийшло потрапити в цю плутанину месенджерів? Навіщо нам цілий каталог додатків для відправки повідомлень, потрібних тільки для зв'язку з друзями?

SMS: перший додаток для спілкування. Вернемось трохи назад в часі, тоді набували популярності прості кнопочні телефони і для відправки повідомлень на телефон був тільки один спосіб: SMS.

Наприклад візьмемо Нову Зеландію: там оператори пропонували тариф в \$10 за необмежену кількість повідомлень, проте незабаром обмежили їх кількість 10 000, виявивши, що тинейджери будуть відправляти стільки повідомлень, скільки їм дозволиш. Ввівши дане обмеження підлітки підраховували залишок повідомлень, відправляли тисячі повідомлень в день і намагалися не витратити їх всіх. Дійшовши до нуля, підліток опинявся відрізанним від світу, або повинен був платити по \$ 0,2 за повідомлення до початку наступного місяця. І всі завжди вичерпували це обмеження, накопичуючи борги за відправку крихітних фрагментів тексту.

Тоді все було простіше. Маючи лише номер телефону людини можна було з легкістю відправити йому повідомлення. Не потрібно було перевіряти безліч

додатків і перемикалися між сервісами. Всі повідомлення розташовувалися в одному місці та все було чудово. Якщо потрібно було відправити повідомлення з комп'ютеру, тоді використовували MSN Messenger або AIM (Не будемо незаслужено забувати про ICQ), але цим користувалися рідко, по тій причині що треба було бути фізично присутнім біля комп'ютера на відмін SMS.

А потім в телефони проник інтернет і з'явився новий різновид додатків для обміну повідомленнями: завжди в онлайні, на телефоні, з фотографіями, посиланнями та іншими видами матеріалів. Тоді вже відпадала вимога платити оператору за SMS.

Стартапи і техногіганти почали боротьбу за новий світ, що не відключається від глобальної мережі та в результаті в наступні роки з'явилися сотні додатків для обміну повідомленнями. iMessage здобув популярність у користувачів iPhone в США, зокрема тому, що міг відправляти повідомлення через SMS. WhatsApp, тоді ще незалежний (не куплений Facebook), завоював Європу, так як концентрувався на приватності. У справу втрутився Китай і поширив WeChat, де користувачі в результаті змогли робити все, від покупки музики до пошуку таксі.

Дивно, що імена майже всіх цих нових месенджерів будуть знайомі: Viber, Signal, Telegram, Messenger, Kik, QQ, Snapchat, Skype і так далі. Ще більш дивно, що на телефоні знайдеться кілька цих додатків - безумовно не тільки один з них. Більше немає тільки одного месенджера чи сервіса.

Розберемо ситуацію сьогодні в переважній більшості молоді користується Telegram або Instagram, дорослі використовують Viber або Facebook messenger якщо брати ситуацію в загальному. Але присутні більш уніфіковані ситуації, наприклад: коли є рідня з Росії, а вони використовують WhatsApp і Skype для спілкування, окрема частина молоді користується Discord або Signal.

Тисячі причин привели нас до цієї роздробленості, однак месенджери стали таким собі «зоопарком»: ніхто один з одним не дружить, а між месенджером повідомлення передавати не є можливим, тому що кожен з них використовує запатентовану технологію. Старі програми для обміну повідомленнями дбали про функціональну сумісність – наприклад, Google Talk використовував протокол Jabber, щоб користувачі могли відправляти повідомлення іншим людям, які використовують той же протокол.

Ніщо не може спонукати Apple відкрити протокол iMessage для інших додатків – або навіть для користувачів Android - оскільки тоді користувачам буде дуже легко піти з екосистеми iPhone. Месенджери стали символами закритого ПЗ, ідеальним інструментом для керування користувачами: від них важко відмовитися, коли всі твої друзі їх використовують.

Сервіс коротких повідомлень, SMS, незважаючи на всі недоліки, був відкритою платформою. Як email сьогодні, SMS працював всюди, незалежно від пристрою або провайдера. Можливо, провайдери вбили цей сервіс, вимагаючи непропорційно високу ціну, але SMS був зручний через те, що він «просто працював», та був єдиним, надійним способом відправити повідомлення кому завгодно.

Невелика надія ще є. Якщо Facebook впорається з об'єднанням месенджерів, то ця ситуація може змінитися та можливо буде заданий новий технологічний тренд:

у січні 2019 The New York Times повідомила, що компанія працює над об'єднанням Messenger, Instagram і WhatsApp в одну платформу(back-end), щоб користувачі могли листуватися один з одним, не перемикаючись між месенджерами. На перший погляд це виглядає привабливо, проте знаючи компанію Facebook є деякі сумніви: Instagram хороший тим, що він відокремлений, як і WhatsApp і їх об'єднання дозволить Facebook отримати конференційну інформацію про користувача яку вона перепродасть третім лицам щоб краще продавати рекламу.

Також така система буде великою ціллю для зловмисників, якщо всі месенджери зберуться в одному місці, то атакуючим досить буде зламати один з них, щоб дізнатися про користувачів все. Деякі користувачі, що піклуються про безпеку, спеціально перемикаються між різними додатками, вважаючи, що їхні розмови складніше відстежити, якщо розбити їх на кілька каналів.



Рисунок. Список техногігантів які підтримали протокол RCS

Є й інші проекти щодо поживлення відкритих систем обміну повідомлень. Протокол Rich Communication Services (RCS) продовжує справу SMS та недавно отримав підтримку операторів і виробників пристроїв по всьому світу. RCS переносить всі улюблені властивості iMessage на відкриту платформу – покажчики набору повідомлень співрозмовником, зображення, онлайн-статуси, тому його зможе реалізувати будь-який виробник і оператор.

Незважаючи на те, що Google активно просуває цей стандарт і інтегрує його в Android, RCS повільно набирає обертів, і відчуває проблеми, що відкладають його широку популяризацію. Наприклад, Apple відмовилася додавати його в iPhone. Стандарт отримав підтримку від таких великих гравців (рисунок), як Google, Microsoft, Samsung, Huawei, HTC, ASUS і так далі, проте Apple зберігає мовчання - можливо, побоюючись втрат привабливості iMessage. Також RCS залежить від підтримки його операторами, проте ті гальмують, оскільки він потребує значних інвестицій в інфраструктуру.



Але незручна реальність полягає в тому, що цей безлад навряд чи буде виправлений в найближчому майбутньому.

Монополісти та новий ринок. На відміну від більшої частини технологічного сектора, де близькі до монополіям гравці взяли контроль в свої руки - наприклад, Google в області пошуку і Facebook в соцмережах – обмін повідомленнями ще потрібно підім'яти під себе. Історично було дуже складно захопити монополію в обміні повідомленнями, оскільки ця область сильно фрагментована, а перемикатися між сервісами дуже неприємно. Однак Facebook, контролюючи так багато великих месенджерів, явно намагається захопити цей простір, щоб користувачі взагалі нікуди від нього не відходили.

Дана технологічна область попала на розквіт техногігантів та стартапів. На ринку не були присутні монополії, які б розділили між собою цей сегмент ринку, та не давали б його розділити іншим молодим компаніям, як це сталося з послугами мережі інтернет в США. Така свобода була дуже хорошим ґрунтом для того щоб компанії та стартапи випускати месенджери, а постійна конкуренція спонукала все до нових та нових ідей та цікавих реалізацій в цих додатках.

Причина по якій компанії продовжують надалі створювати все нові та нові месенджери полягає в одній речі – гроші. Цим фактором можна пояснити мотивацію 99% компаній їхні дії, причини через, які вони розробляють чи впроваджують ті чи інші речі. Сегмент з месенджерами не став виключенням з цього переліку. Велика частина месенджерів заробляють гроші на рекламі і продажі стікерів в цей перелік можна віднести Viber, Messenger, Instagram і WhatsApp, Snapchat та інші.

У азійських месенджерів інша стратегія. Стратегія Line, WeChat, і Kakao Talk полягає у вибудовуванні навколо месенджера екосистеми додаткових сервісів – ігри, мобільна комерція, продаж музики та інші. Наприклад, LINE в 2013 році заробила \$ 336 млн. На продажі ігор довелося близько 60%, на стікери – ще 20%.

Присутні також виключення з цієї грошової ситуації, цьому є приклад Telegram(був до березня 2021), FireChat, Signal, CryptoCat. Головна причина чому цими месенджерами користуються це приватність.

Бажання заробити гроші не покидає компанії і не покине до її банкрутства. Для прикладу візьмемо Google: не зважаючи на те що Google належить OS Android, створити популярний месенджер і монополізувати ринок їй не вдалось, хоча спроб було багато. На сьогоднішній день у компанії є 6 сервісів для спілкування, якщо не рахувати Gmail, але вони користуються малим попитом і компанія не покидає спроб створити новий месенджер, щоб заробити гроші і таке прагнення до прибутку присутнє не тільки у Google.

Парадоксальна людська інерція до нового. У різних країнах по різному

відбувався цей процес переходу від SMS до месенджерів, дуже впливав чинник часу цього переходу, по тій причині що в різні часи були популярні різні месенджери.

Не зважаючи на природню людську цікавість, яка вивела людину до вершини еволюційної ланки, людина не завжди готова до чогось нового. Ще одна причина роздробленості можна вважати «Синдром каченяти». Тобто людина прив'язується до добре знайомої речі або інформації, а сприймати щось нове їй важко так як нове дається їй дивним та незрозумілим. Отже, необов'язковим до вивчення. По цій причині багато користувачів залишаються в старих месенджерах навіть якщо з'явився новий месенджер який на декілька кроків випереджає у всьому старий месенджер. Та враховуючи користувацьку базу тих чи інших месенджерів, новий користувач буде користуватися тим месенджером де його друзі, колеги ніж новим з кращим функціоналом.

Але деякій частині користувачі, які прагнуть знайти і спробувати щось нове переходять з одного месенджера в надії знати це, але більшість оточення користувача все одно залишається на незручній користувачеві платформі, а частина знаходиться разом з ним на новій платформі і ось користувачеві для того щоб спілкуватися з іншими користувачами змушений користуватись декількома додатками.

Висновки. В даному дослідженні була проведена спроба пояснення причини роздробленості месенджерів. Були проаналізовані передумови, з чого все починалось і до чого прийшли месенджери сьогодення, як на це вплинули компанії та стартапи, людська інерція до нового та простий плин часу.

1. Зачем нам столько мессенджеров? URL: <https://habr.com/ru/post/450322/> (дата звернення: 06.05.2021).
2. Why Do We Need So Many Different Messaging Apps? URL: <https://www.vice.com/en/article/zma735/why-do-we-need-so-many-different-messaging-apps> (дата звернення: 06.05.2021).
3. Best chat apps for 2021. URL: <https://www.tomsguide.com/best-picks/best-chat-apps> (дата звернення: 06.05.2021).
4. «Пользователям придется смириться с беспорядком на смартфоне»: зачем Google выпускает так много мессенджеров. URL: <https://ain.ua/2017/03/07/zachem-google-mnogo-messendzherov/> (дата звернення: 06.05.2021).
5. Миллиарды авансом: на чем зарабатывают мессенджеры. URL: <https://www.forbes.ru/tekhnologii/internet-i-svyaz/270461-milliardy-avansom-na-chem-zarabatyvayut-messendzhery> (дата звернення: 06.05.2021).

Kozliuk S. M., Senior Lecturer; Yaruta A. H., Senior Student (National University of Water and Environmental Engineering, Rivne)

ANALYSIS OF MESSAGE FRAGMENTATION

The work is devoted to the study of the causes of fragmentation and fragmentation of messengers, the preconditions and causes of this phenomenon.

Keywords: messenger, SMS, Whatsapp, Signal, iMessage, Telegram, Facebook Messenger, Viber.

Соломко¹ М. Т., к.т.н, доцент; Глінчук М. В., студент (Національний університет водного господарства та природокористування, м. Рівне, ¹m.t.solomko@nuwm.edu.ua)

СУМАТОР БІНАРНИХ КОДІВ ДЛЯ РЕАЛІЗАЦІЇ АЛГЕБРИЧНИХ ОПЕРАЦІЙ

У статті розглянута побудова схеми суматора бінарних кодів для отримання властивостей операції алгебричного додавання. Для забезпечення логіки встановлення знаку коду суми при різних знаках вхідних аргументах та фіксації сигналу переповнення у двох старших розрядах схеми суматора використовується по одному вхідному елементу OR. Діапазон чисел схеми суматора з логічними елементами OR в останньому розряді при обробці модифікованого коду залишається у межах від 0 до $2^n - 1$.

Ключові слова: модифіковані коди, алгебричне додавання, суматор бінарних кодів.

Вступ. Фундаментальною ідеєю щодо відображення реального світу за допомогою обчислювальної техніки є методи дискретизації об'єктів. Для ефективних обчислень виникає потреба синтезу моделей реальних об'єктів та процесів їх перетворення. Досить часто такими моделями можуть бути конструкції дискретної математики, зокрема, алгебричні структури [1].

Прикладом практичного використання алгебричних структур – множин з алгебричними операціями може бути кодування інформації при передачі даних через канал зв'язку. При цьому ставиться вимога забезпечити виправлення помилок, що виникають внаслідок фізичних завад у каналах зв'язку та збереження інформації. Найкращих результатів можна досягти, якщо символи, що передаються, розглядаються як елементи певних алгебраїчних структур.

Сучасні алгоритми шифрування з таємним ключем використовують алгебраїчні структури скінчених полів, наприклад, стандарт AES симетричного шифрування. Широко вживаний алгоритм RSA шифрування з відкритим ключем ґрунтується на алгебричному підході фактор-кільця кільця цілих чисел за модулем великого натурального числа.

Простота виконання арифметичних операцій є важливою особливістю двійкової системи числення. Однак при виконанні арифметичних операцій у конкретній обчислювальній системі необхідно додатково враховувати не тільки знаки чисел, що обробляються, але і діапазон та форму подання чисел. Відомо, що одним з можливих шляхів виконання операції віднімання є заміна знаку числа, яке віднімається, на протилежний і додавання його до зменшуваного: $A - B = A + (-B)$.

Таким чином операцію арифметичного віднімання можна замінити операцією алгебричного додавання, яку можна виконати за допомогою двійкових суматорів. Від'ємні числа в обчислювальній системі подаються у прямому, доповняльному і оберненому кодах.

Проектування цифрових пристроїв і систем вимагає вирішення питання представлення додатних і від'ємних чисел, виконання основних алгебричних операцій, оцінювати можливості отримання заданої точності обчислень та визначати ознаки переповнення розрядної сітки. Такі задачі вирішуються за допомогою використання спеціальних кодів, при застосуванні яких можна спростити апаратну реалізацію АЛП, розширити їх функціональні можливості та збільшити швидкодію.

1. Агоритми виконання арифметичних операцій у цілочисельних операційних пристроях. Для більшості сучасної обчислювальної техніки загальноприйнятим є такий формат з фіксованою комою (ФК), коли кома фіксується праворуч від молодшого розряду коду числа. З цієї причини відповідні операційні пристрої називають цілочисельними ОПр. У формі з ФК можуть бути представлені як числа без знаку, коли всі n позицій числа відводяться під значущі цифри, так і зі знаком. В останньому випадку старший ($n - 1$)-й розряд числа займає знак числа (0 – плюс, 1 – мінус), а під значущі цифри відведені розряди з ($n - 2$)-го до 0-го. Під час запису від'ємних чисел використовується доповняльний код.

Цілочисельний ОПр повинний забезпечувати виконання таких арифметичних операцій над числами без знака і зі знаком: *додавання/віднімання; множення; ділення.*

При складанні таблиці додавання у будь-якій системі числення застосовують таке правило: якщо при додаванні двох цифр a_i і a_j

$$a_i + a_j < P, \text{ то } a_i + a_j = a_k,$$

де a_k – цифра даної системи числення; P – основа системи числення, якщо $a_i + a_j \geq P$, то $a_k = a_i + a_j - P$ і з'являється одиниця перенесення в наступний старший розряд.

Правило складання таблиці додавання можна записати у наступному вигляді:

$$a_i + a_j = \begin{cases} a_k, & \text{якщо } a_i + a_j < P; \\ 1a_k, & \text{якщо } a_i + a_j \geq P, \end{cases}$$

де $a_k = a_i + a_j - P$.

Потрібно враховувати, що у двійковій системі числення кожен старший розряд містить дві одиниці сусіднього молодшого розряду.

У цифрових обчислювальних системах використовуються дві форми подання чисел: *природна* (подання чисел з фіксованою комою) і *нормальна* (подання чисел з плаваючою комою).

2. Модифіковані коди для алгебричних операцій у цифрових пристроях. Для задання знаку числа у модифікованих кодах використовують *два розряди* (табл. 1) [2].

Кодування знаку числа модифікованим кодом

Біти знакових розрядів	Відповідність
00	Знак «+»
11	Знак «-»
01	Ознака позитивного переповнення
10	Ознака від'ємного переповнення

Модифіковані коди виявилися зручними (з точки зору побудови АЛП) для виявлення переповнення розрядної сітки. Якщо знакові розряди результату приймають значення 00 або 11, то переповнення розрядної сітки не було, а якщо 01 або 10 – то переповнення мало місце.

При алгебричному додаванні знакові розряди розглядаються як старші розряди числа. Якщо виникає перенесення із старшого розряду при операції додавання, то біт перенесення додається до молодшого розряду суми, у випадку застосування модифікованого оберненого коду або відкидається при застосуванні модифікованого доповняльного коду.

Недолік, властивий оберненому коду переходить у модифікований обернений код. Модифікований прямий код зберігає всі недоліки прямого коду. У зв'язку з чим будемо розглядати тільки модифіковані доповняльні коди (МДК).

МДК отримується з доповняльного простим дублюванням знакового розряду числа. Причому, у пам'яті комп'ютера числа можуть зберігатися у доповняльному коді та перетворюватися у модифікований код безпосередньо при пересиланні їх на виконавчий пристрій.

Виконання арифметичних операцій у МДК виглядає так:

Приклад 1. Додати у МДК числа $A = 0010101 (21_{10})$ і $B = 0010010 (18_{10})$. $A > 0$, $B > 0 [2]$.

```

0010101
0010010
0100111

```

Має місце позитивне переповнення.

Приклад 2. Додати у МДК числа $A = 0010101 (21_{10})$ і $B = 0001010 (10_{10})$. $A > 0$, $B > 0$.

```

0010101
0001010
0011111

```

Отримана позитивна сума у МДК.

Приклад 4. Додати у МДК числа $A = 0010101 (21_{10})$ і $B = 1100110 (-26_{10})$. $A > 0$, $B < 0$. $A < |B|$.

0010101
1100110
1111011

Отримана від'ємна сума у МДК.

Приклад 5. Додати у МДК числа $A = 1101011 (-21_{10})$ і $B = 1101110 (-18_{10})$. $A < 0$, $B < 0$.

1101011
1101110
1011001



Національний університет
водного господарства
та природокористування

Має місце від'ємне переповнення.

Модифіковані коди застосовуються для отримання алгебричного додавання, що, у підсумку, суттєво спрощує апаратну частину цифрових компонентів. Алгебричне додавання у схемі суматора забезпечується співставленням знакових розрядів модифікованого коду зі старшими розрядами схеми пристрою. Оскільки знакові розряди модифікованих кодів двох вхідних аргументів з різними знаками завжди мають структуру 00 і 11 або 11 і 00, перенесення у двох старших розрядах схеми суматора може бути тільки транзитивним. По-іншому структура 00 і 11 або 11 і 00 двох знакових розрядів створює умову для транзитивного перенесення та не генерує власний сигнал перенесення.

Умова транзитивного перенесення визначається логічною функцією (1) [3].

$$p_i = a_i \vee b_i \text{ або } p_i = a_i + b_i. \quad (1)$$

Якщо $p_i = 1$, то транзитивне перенесення до наступних розрядів буде можливим, у випадку коли $p_i = 0$ транзитивне перенесення до наступних розрядів неможливе. Аргументи $a_i = 0$ і $b_i = 1$ або $a_i = 1$ і $b_i = 0$ логічної функції (1) завжди дають значення $p_i = 1$. Тому, для забезпечення логіки встановлення знаку коду суми при різних знаках вхідних аргументах та фіксації сигналу переповнення, достатньо у двох старших розрядах схеми суматора використати по одному вхідному елементу OR (п. 4). Це спрощує логічну структуру суматора та оптимізує глибину схеми.

3. Суматор бінарних кодів з елементами OR у знакових розрядах вхідних аргументів для реалізації алгебричних операцій. 4-bit суматор бінарних кодів з елементами OR на входах двох останніх розрядах для додавання 2-bit чисел з різними знаками у модифікованому коді представлений на рисунку.

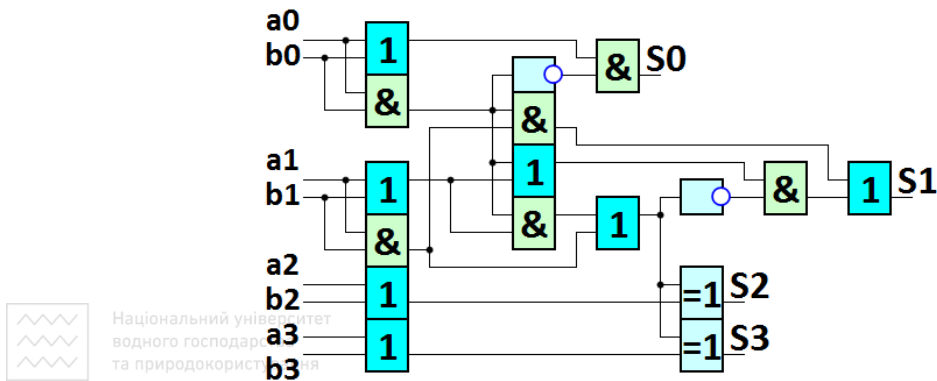


Рисунок. 4-bit ациклічний суматор з елементами OR на входах двох останніх розрядів для додавання 2-bit чисел з різними знаками у модифікованому коді

Логічні рівняння 4-bit суматора бінарних кодів на рисунку є такі:

$$\begin{aligned}
 S_0 &= a_0 \bar{b}_0 + \bar{a}_0 b_0; \\
 S_1 &= \bar{a}_0 a_1 \bar{b}_1 + a_0 b_0 a_1 b_1 + a_0 b_0 \bar{a}_1 \bar{b}_1 + \bar{a}_0 a_1 b_1 + \bar{b}_0 a_1 \bar{b}_1 + \bar{b}_0 a_1 b_1; \\
 S_2 &= a_0 b_0 a_1 \bar{a}_2 \bar{b}_2 + a_0 b_0 b_1 a_2 \bar{b}_2 + a_1 b_1 a_2 \bar{b}_2 + \bar{a}_1 b_1 \bar{b}_2 + \\
 &\quad + \bar{a}_0 a_1 b_2 + \bar{a}_1 b_1 a_2 + \bar{a}_0 a_1 a_2 + \bar{b}_0 a_1 b_2 + \\
 &\quad + \bar{b}_0 a_1 a_2 + \bar{a}_0 b_1 b_2 + \bar{a}_0 b_1 a_2 + \\
 &\quad + \bar{b}_0 b_1 b_2 + \bar{b}_0 b_1 a_2; \\
 S_3 &= +a_0 b_0 a_1 a_3 \bar{b}_3 + a_0 b_0 b_1 a_3 \bar{b}_3 + a_1 b_1 a_3 \bar{b}_3 + \bar{a}_1 b_1 b_3 + \bar{a}_0 a_1 b_3 + \\
 &\quad + \bar{a}_1 b_1 a_3 + \bar{a}_0 a_1 a_3 + \bar{b}_0 a_1 b_3 + \bar{b}_0 a_1 a_3 + \bar{a}_0 b_1 b_3 + \bar{a}_0 b_1 a_3 + \\
 &\quad + \bar{b}_0 b_1 b_3 + \bar{b}_0 b_1 a_3.
 \end{aligned}$$

Діапазон чисел схеми суматора з логічними елементами OR в останньому розряді при обробці модифікованого коду залишається у межах від 0 до $2^n - 1$, а для 2-розрядної сітки суматора на рисунку діапазон чисел у модифікованому коді становить від 0 до 3 (табл. 2).

Таблиця 2

Коди, що забезпечують
представлення знака числа для діапазону від 0 до 3

Коди для додатніх чисел				
Десятковий	Прямий	Обернений	Доповняльний	МДК
0	0,00	0,00	0,00	00,00
+1	0,01	0,01	0,01	00,01

продовження табл. 2

+2	0,10	0,10	0,10	00,10
+3	0,11	0,11	0,11	00,11
Коди для від'ємних чисел				
Десятковий	Прямий	Обернений	Доповняльний	МДК
-1	1,01	1,10	1,11	11,11
-2	1,10	1,01	1,10	11,10
-3	1,11	1,00	1,01	11,01

Висновки. Для отримання властивостей операції алгебричного додавання, знакові розряди модифікованих кодів співставляються зі старшими розрядами схеми суматора. Оскільки знакові розряди модифікованих кодів двох вхідних аргументів з різними знаками завжди мають структуру 00 і 11 або 11 і 00, перенесення у двох старших розрядах схеми суматора може бути тільки транзитивним. А відтак, для забезпечення логіки встановлення знаку коду суми при різних знаках вхідних аргументах та фіксації сигналу переповнення, достатньо у двох старших розрядах схеми суматора використати по одному вхідному елементу OR. Це спрощує логічну структуру суматора та оптимізує глибину схеми. Діапазон чисел схеми суматора з логічними елементами OR в останньому розряді при обробці модифікованого коду залишається у межах від 0 до $2^n - 1$.

Тому структура суматора з логічними елементами OR на двох старших розрядах схеми суматора бінарних кодів дає підставу для доцільності її застосування у процесах синтезу арифметичних пристроїв обробки цифрових даних зі знаком, оскільки зазначена схема суматора спроможна:

- збільшити швидкодію, порівняно з аналогами;
- зменшити енергоспоживання та тепловиділення цифрового пристрою, інтегральної схеми.

1. Завало С. Т. Курс алгебри. Київ : Вища школа, 1985. 503 с.
2. Solomko M., Tadeyev P., Zubyk Y., Hladka O. Reduction and is optimal of speed of acyclic adders of binary codes. *Eastern-European Journal of Enterprise Technologies*. 2019. Vol. 1. No 4(97). Pp 40–53.
3. Solomki M. T. Optimization of the acyclic adders of binary codes. *Technology audit and production reserves*. 2018. Vol 3/2 (41). Pp. 55–65.

Solomko M. T., Candidate of Engineering, Associate Professor; Glinchuk M. V. Senior Student (National University of Water and Environmental Engineering, Rivne)

BINARY CODE ADDER FOR REALIZATION OF ALGEBRIC OPERATIONS

In the article the construction of the scheme of the adder of binary codes for receiving

properties of operation of algebraic addition is considered. To ensure the logic of setting the sign of the sum code at different signs of the input arguments and fixing the overflow signal in the two highest bits of the adder circuit, one input element OR is used. The range of numbers of the adder circuit with logical elements OR in the last digit when processing the modified code remains in the range from 0 to.

Keywords: modified codes, algebraic addition, binary code adder.

УДК 004.4.272



Національний університет

Шевченко¹ І. М., ст. викладач; Опанасюк М. Ю., студент (Національний університет водного господарства та природокористування, м. Рівне,
¹i.m.shevchenko@nuwm.edu.ua)

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ ОРГАНІЗАЦІЇ КОНТРОЛЮ ЗНАНЬ ЗА ДОПОМОГОЮ ТЕСТІВ З ДИСЦИПЛІНИ

Розглянуто сучасні навчальні інформаційні системи, що призначені для автоматизації організації контролю знань з використанням тестів, визначено їх функціональність. Описано методологію створення власного тестового редактора для автоматизації процесу створення тестових завдань та організації проведення контролю знань, умінь та навичок студентів.

Ключові слова: тестовий редактор, тест, комп'ютерне тестування.

Контроль та оцінювання навчальних досягнень студентів є важливою складовою навчально-виховного процесу у закладі вищої освіти.

Традиційними формами контролю знань, умінь та навичок у навчальному процесі ЗВО є співбесіда, контрольні роботи, колоквиуми, написання рефератів, семінари, самоконтроль, взаємоконтроль, лабораторні, практичні роботи, курсові роботи, курсові та дипломні проекти, різні види навчальної та виробничої практик, заліки, іспити.

Світова практика широко застосовує для визначення рівня знань тестовий контроль і рейтинг, які в повному обсязі враховують усі функції педагогічного контролю.

Тестовий контроль знань, умінь і навичок студентів дає змогу ефективно й об'єктивно оцінювати успішність студентів. Рейтингова система та її складові — тестовий контроль оцінювання знань, умінь і навичок (ЗУН) — дають можливість підвищити об'єктивність, ритмічність та всебічність контролю з урахуванням індивідуальних рис студентів. Вона передбачає зміни у психології студента, який мусить працювати самостійно, ініціативно, систематично впродовж усього семестру. [1].

Тестовий контроль знань, умінь і навичок студентів дає змогу організувати такі види контролю: вхідний (стартовий або попередній), поточний, проміжний і підсумковий.

Використання сучасних комп'ютерних інформаційних технологій дозволяє автоматизувати процеси створення тестів та оцінювання навчальних здобутків студентів.

Відмітимо переваги комп'ютерного тестування:

- певний психологічний комфорт студентів під час тестування;
- підвищення об'єктивності оцінювання знань, і, як наслідок, позитивний стимулюючий вплив на пізнавальну діяльність студентів;
- індивідуалізація процесу навчання (автономність);
- конфіденційність при анонімному тестуванні;
- швидке одержання результатів і звільнення викладача від трудомісткої роботи з обробки результатів тестування;
- виключення негативного впливу на результати тестування таких факторів як настроїв, рівень кваліфікації й інші характеристики конкретного викладача;
- можливість застосування технічних засобів;
- універсальність, охоплення всіх стадій процесу навчання;
- контроль великого обсягу матеріалу.

Виходячи з вищезазначеного, появилася ідея створення тестового редактора для автоматизації процесу створення тестових завдань для організації проведення контролю знань, умінь та навичок студентів.

З метою визначення функціональності тестового редактора проаналізовано подібні програми: пакет MyTest, безкоштовний освітній сервіс Майстер-Тест, безкоштовна платформа Google для створення віртуальних класів, розробки та коригування практичних завдань і тестів, самостійних та контрольних робіт й активного обміну завданнями з учителем [2].

Основною задачею пропонованої розробки є створення тестової системи для контролю знань, умінь і навичок, з якою можуть працювати студент та викладач. Викладач повинен мати змогу реєструвати особистий обліковий запис, за допомогою якого заходити у систему, переглядати статистику успішності складених студентами тестів, видаляти, додавати та редагувати тести. Студент може реєструватись, проходити тести та переглядати свої результати.

Для реалізації інформаційної системи було обрано сучасну мову об'єктно-орієнтованого програмування C# із застосуванням Windows Forms – інтерфейсу програмування додатків, відповідального за графічний інтерфейс користувача. Середовище розробки – Microsoft Visual Studio.

На етапі проектування додатку розроблено користувацький інтерфейс, який представлений на рис.1 у вигляді UML діаграми класів.

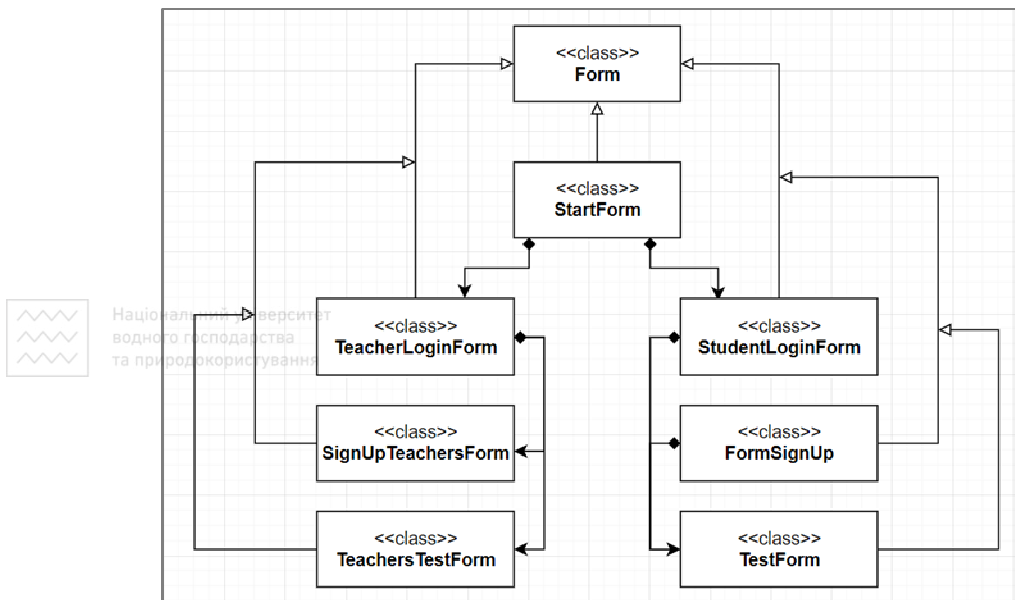


Рис. 1. Зв'язки між класами форм

StartForm – клас, у якому описуються усі дії початкової форми. Форма цього класу запускається першою у програмі.

StudentLoginForm – клас для опису подій форми входження в систему студента.

TeacherLoginForm – клас для опису подій форми входження в систему викладача.

FormSignUp – клас, де описуються події реєстрації студента.

SignUpTeachersForm – клас, де описуються події реєстрації викладача.

TestForm – клас для опису подій проходження тесту студентом.

TeacherTestForm – клас, у якому описуються події взаємодії викладача з тестовими питаннями та переглядом статистики успішності студентів.

Функціональність додатку, а саме: реєстрація студента та викладача, можливість створення, редагування та перегляду результатів тесту викладачем, проходження тесту та перегляду результату тестування студентом, реалізована в класах **Question**, **Teacher**, **Student**.

Алгоритм роботи програми представлений UML діаграмою послідовності, яка представлена на рис. 2.

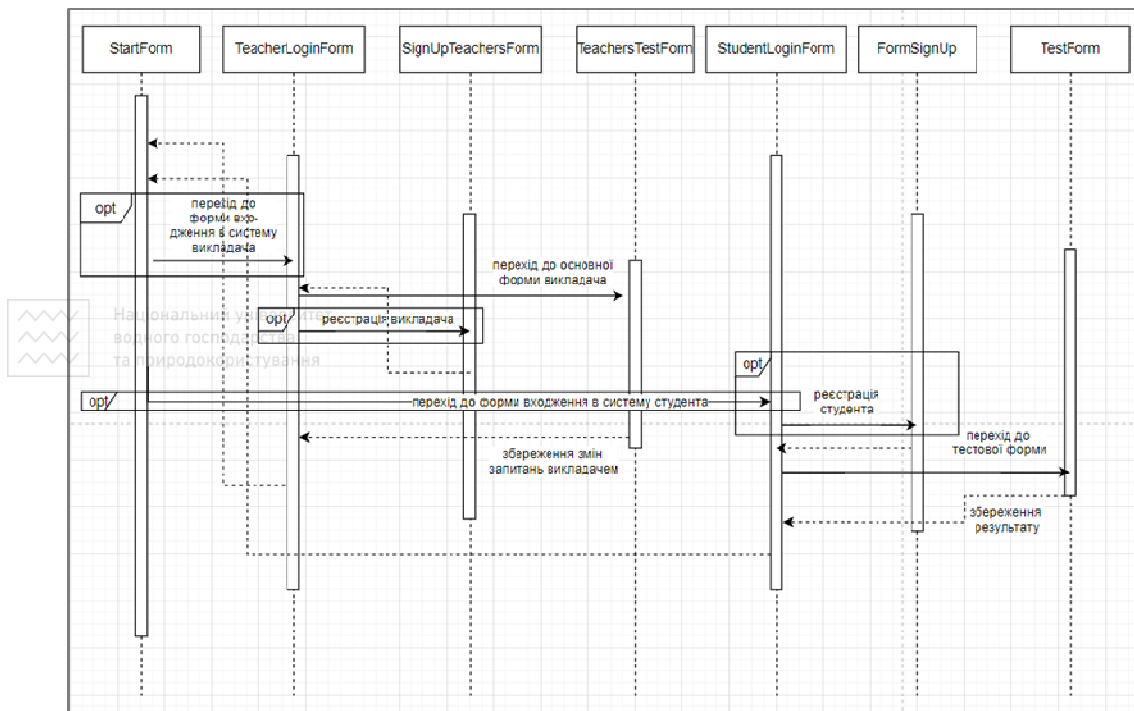


Рис. 2. Алгоритм програми

Після запуску програми користувач бачить стартову сторінку програми (рис. 3)

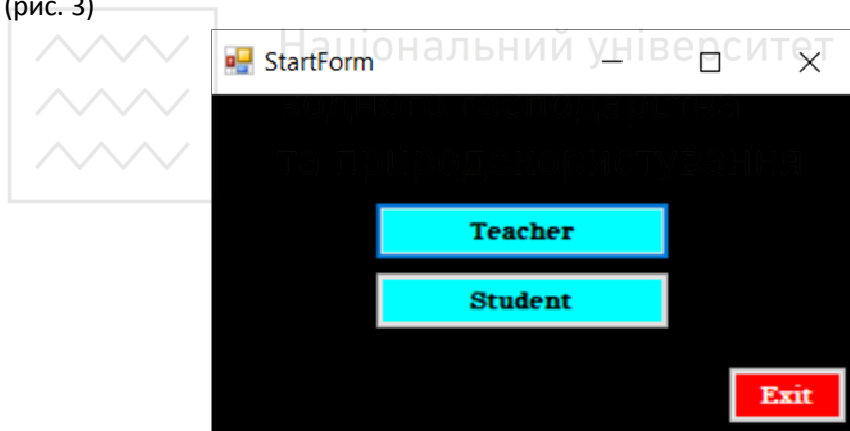


Рис. 3. Стартова сторінка

Обираючи кнопку Teacher, користувач може реєструватися як викладач та має можливість створювати, редагувати, видаляти та додавати нові питання в тест. Ці можливості наведені на рисунках 4, 5.

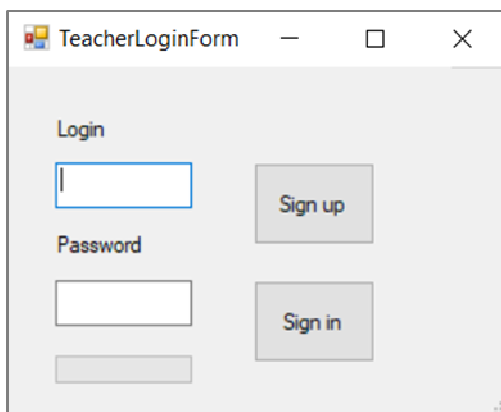


Рис. 4. Форма входу для викладача

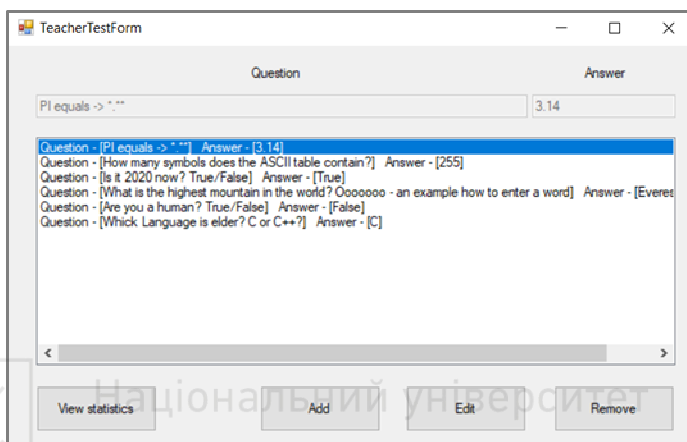


Рис. 5. Форма для редагування питань і перегляду інформації про успішність студентів

Обираючи кнопку Student, користувач може реєструватися як студент та має можливість проходити тест. Ці можливості наведені на рисунках 6, 7.

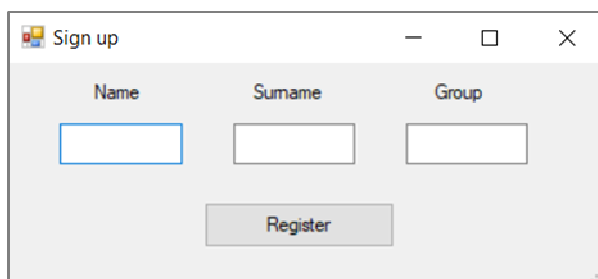


Рис. 6. Форма реєстрації студента

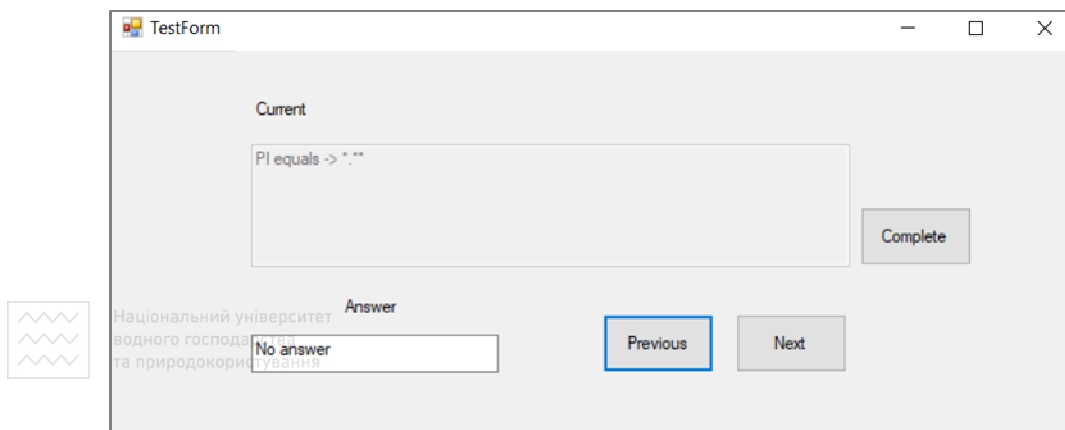


Рис. 7. Форма для відповіді на запитання тесту

Висновки. Інтерфейс програми є інтуїтивно зрозумілим, оскільки усі елементи керування мають інформативні назви. Перевагами додатку є зручне користування, оскільки не потрібно встановлювати окремих застосунків для викладача і студента. Встановивши дану програму викладач має змогу завантажити запитання, і, не виходячи з програми, студент може відразу пройти тест. Таку систему можна доповнювати новими тестами та зручно використовувати для організації контролю знань за допомогою комп'ютера.

1. Кизима Р. А., Яковчук В. В. Тестова система оцінювання знань студентів, як один із методів сучасної технології навчання. *Актуальні проблеми навчання та виховання людей з особливими потребами*: збірник. № 4(6). С. 19–29. URL: <http://ap.uu.edu.ua/article/279> (дата звернення: 06.05.2021). 2. 7 сервісів для створення навчальних тестів та завдань онлайн. URL: <https://buki.com.ua/news/7-servisiv-dlya-stvorenniya-navchalnykh-testiv-ta-zavdan-onlayn/> (дата звернення: 06.05.2021).

Shevchenko I. M., Senior Lecturer, Opanasiuk M. Yu., Senior Student (National University of Water and Environmental Engineering, Rivne)

INFORMATION SYSTEM FOR THE ORGANIZATION OF KNOWLEDGE CONTROL USING TESTS ON A SUBJECT

The modern educational information systems designed to automate the organization of knowledge control using tests have been considered, their functionality has been determined. The methodology of creating a custom test editor for automating the process of creating test tasks and organizing control of knowledge, skills and abilities of students has been described.

Keywords: test editor, test, computer testing.

Соломко¹ М. Т., к.т.н, доцент; Патійчук Н. Б., студент (Національний університет водного господарства та природокористування, м. Рівне,
¹m.t.solomko@nuwm.edu.ua)

СПРОЩЕННЯ БУЛЕВИХ ФУНКЦІЙ В ІМПЛІКАТИВНОМУ БАЗИСІ ЗА ДОПОМОГОЮ ОБРАЗНИХ ПЕРЕТВОРЕНЬ



Національний університет

водного господарства
та природокористування

У статті розглянуто мінімізацію булевих функцій в імплікативному базисі методом образних перетворень. Встановлена можливість зменшення обчислювальної складності, збільшення продуктивності спрощення булевих функцій у класі досконалих імплікативних нормальних форм (ДІНФ-1 та ДІНФ-2) методом образних перетворень.

Ключові слова: імплікативний базис, мінімізація булевих функцій, образні перетворення, функціональна повнота.

Вступ. Найчастіше функція імплікації виражається через диз'юнкцію і заперечення наступним чином:

$$x_1 \rightarrow x_2 = \bar{x}_1 \vee x_2. \quad (1)$$

Для функції (1) справедливі такі закони:

$$\begin{aligned} x \rightarrow x &= 1; & x \rightarrow 1 &= 1; & 0 \rightarrow x &= 1; \\ x \rightarrow \bar{x} &= \bar{x}; & x \rightarrow 0 &= \bar{x}; & 1 \rightarrow \bar{x} &= x. \end{aligned}$$

З представлених законів випливає, що функція імплікації володіє тільки властивостями комутативності (переміщувальний закон) у зміненому вигляді:

$$x_1 \rightarrow x_2 = \bar{x}_2 \rightarrow \bar{x}_1.$$

Властивість асоціативності (сполучний закон) для цієї функції не справедливий, оскільки $x_1 \rightarrow (x_2 \rightarrow x_3) \neq (x_1 \rightarrow x_2) \rightarrow x_3$. Дійсно:

$$x_1 \rightarrow (x_2 \rightarrow x_3) = \bar{x}_1 \vee (x_2 \rightarrow x_3) = \bar{x}_1 \vee \bar{x}_2 \vee x_3 = \overline{x_1 \cdot x_2} \vee x_3;$$

$$(x_1 \rightarrow x_2) \rightarrow x_3 = \overline{(x_1 \rightarrow x_2)} \vee x_3 = \overline{(\bar{x}_1 \vee x_2)} \vee x_3 = x_1 \cdot \bar{x}_2 \vee x_3.$$

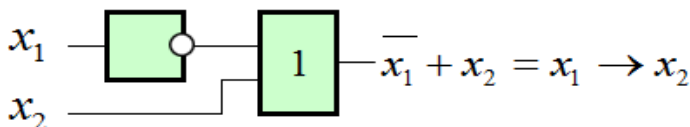
Функції НЕ, АБО, І виражаються через імплікацію наступним чином:

$$\bar{x} = x \rightarrow 0;$$

$$x_1 \vee x_2 = \bar{x}_1 \rightarrow x_2 = (x_1 \rightarrow 0) \rightarrow x_2; \quad (2)$$

$$x_1 \cdot x_2 = x_1 \rightarrow \overline{x_2} = x_1 \rightarrow (x_2 \rightarrow 0) = [x_1 \rightarrow (x_2 \rightarrow 0)]. \quad (3)$$

Логічна схема, що реалізує функцію (1), представлена на рис. 1.



Національний університет
водного господарства
та природокористування

Рис. 1. Логічна схема, що реалізує функцію прямої імплікації $f=x_1 \rightarrow x_2$

Процес мінімізації логічних функцій займає важливе положення всередині технології проектування цифрових компонентів. У зв'язку з цим актуальною проблемою залишається забезпечення адекватної відповідності розробленого продукту заданим специфікаціям собівартості, спрощення та гарантії отримання оптимального результату мінімізації різних представлень логічних функцій.

Оскільки імплікативний базис належить області оптимізації логічних функцій [1], актуальними є дослідження направлені, зокрема, на вдосконалення таких чинників, як:

- методи спрощення функцій імплікативного базису;
- мінімізація логічних схем на основі функцій імплікації;
- достовірність оптимального результату мінімізації імплікативного базису.

Характеристичні функції . Будь-яка функція алгебри логіки, крім константи 1 може бути представлена у вигляді [2].

$$f(x_1, x_2, \dots, x_n) = \bigwedge_{\langle \alpha_1, \alpha_2, \dots, \alpha_n \rangle} \left(\overline{x_1^{\alpha_1} \rightarrow \overline{x_2^{\alpha_2} \rightarrow \dots \rightarrow x_{n-1}^{\alpha_{n-1}} \rightarrow x_n^{\alpha_n}} \right)_{f=0}$$

Таке представлення функції називається ДІНФ-1 (досконала імплікативна нормальна форма 1 роду).

Алгоритм отримання ДІНФ-1:

1. Вибираємо ті набори змінних у таблиці істинності, на яких $f = 0$.
2. Випишемо імплікативні набори змінних, причому.

$$\begin{cases} x_i = 1 \Rightarrow \overline{x_i} \\ x_i = 0 \Rightarrow x_i \end{cases}$$

3. Об'єднуємо всі набори змінних кон'юнкціями.

Приклад 1

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	
0	0	0	0	✓
0	0	1	1	

0	1	0	1	
0	1	1	0	✓
1	0	0	0	✓
1	0	1	1	
1	1	0	0	✓
1	1	1	1	



$$f(x_1, x_2, x_3) = (\overline{\overline{\overline{x_1} \rightarrow x_2} \rightarrow x_3}) \cdot (\overline{\overline{\overline{x_1} \rightarrow x_2} \rightarrow x_3}) \cdot (\overline{\overline{\overline{x_1} \rightarrow x_2} \rightarrow x_3}) \cdot (\overline{\overline{\overline{x_1} \rightarrow x_2} \rightarrow x_3})$$

Будь-яка функція алгебри логіки, крім константи 0 може бути представлена у вигляді [2].

$$f(x_1, x_2, \dots, x_n) = \bigvee_{f=1}^{<\alpha_1, \alpha_2, \dots, \alpha_n>} \left(\overline{\overline{\overline{\dots} \rightarrow x_1^{\alpha_1} \rightarrow x_2^{\alpha_2} \rightarrow \dots \rightarrow x_n^{\alpha_n}}} \right)$$

Таке представлення функції називається ДІНФ-2 (досконала імплікативна нормальна форма 2 роду).

Алгоритм отримання ДІНФ-2:

1. Вибираємо ті набори змінних у таблиці істинності, на яких $f = 1$.
2. Випишуємо імплікативні набори змінних, причому

$$\begin{cases} x_i = 1 \Rightarrow \bar{x}_i \\ x_i = 0 \Rightarrow x_i \end{cases}$$

3. Об'єднуємо всі набори диз'юнкціями.

Приклад 2

x_1	x_2	x_3	$f(x_1, x_2, x_3)$	
0	0	0	0	
0	0	1	1	✓
0	1	0	1	✓
0	1	1	0	
1	0	0	0	
1	0	1	1	✓
1	1	0	0	
1	1	1	1	✓

$$f(x_1, x_2, x_3) = (\overline{\overline{\overline{\overline{x_1} \rightarrow x_2} \rightarrow x_3}}) \vee (\overline{\overline{\overline{\overline{x_1} \rightarrow x_2} \rightarrow x_3}}) \vee (\overline{\overline{\overline{\overline{x_1} \rightarrow x_2} \rightarrow x_3}}) \vee (\overline{\overline{\overline{\overline{x_1} \rightarrow x_2} \rightarrow x_3}})$$

Результати спрощення функцій імплікативного базису методом образних перетворень

Рівносильні образні перетворення при мінімізації функцій імплікативного базису дають наступний результат:

– визначають герменевтику логічних операцій на бінарних структурах функцій імплікативного базису;

– утворюють алгебру імплікативного базису у частині спрощення ДІНФ-1, ДІНФ-1.1, ДІНФ-2 та ДІНФ-2.1 булевих функцій.



Національний університет
водних ресурсів
та природокористування

Герменевтика логічних операцій в імплікативному базисі

В імплікативному базисі герменевтика логічних операцій подібна до герменевтики раніш розглянутих базисів [3].

Для подання досконалих імплікативних нормальних форм, наприклад ДІНФ-1 n -містких булевих функцій бінарним еквівалентом або матрицею потрібно змінні з інверсією \bar{x}_n замінити на 1_n , а змінні без інверсії x_n – на 0_n (п. 4.2), де n – числовий індекс, який визначає розрядність символу-змінної «1» або «0» у термах функції імплікативного базису. Для змінної x_1 , у термах функції імплікативного базису, умови входження до двійкового еквіваленту протилежні (п. 4. 2).

Досконалу імплікативну нормальну форму 3-місткої функції імплікативного базису

$$F = (\overline{x_1 \rightarrow x_2 \rightarrow x_3}) (\overline{x_1 \rightarrow x_2 \rightarrow x_3}), \quad (4)$$

можна подати бінарними наборами (кортежами)

$$F = (0,0,0_3)(0,0,1_3), \quad (5)$$

або матрицею

$$F = \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix}. \quad (6)$$

Назвемо матрицю (6) екземпляром класу бінарних матриць функцій імплікативного базису.

Герменевтика логічних операцій для матриці (6) полягає у тому, що матриця (6) подає терми ДІНФ-1 функції імплікативного базису та операцію кон'юнкції для них. Зазначену герменевтику доцільно застосовувати при виведенні результату логічних операцій у класі бінарних матриць функцій імплікативного базису.

Рівносильні перетвореннями у ДІНФ-1 булевих функцій імплікативного базису

У загальному випадку під час мінімізації булевих функцій імплікативного базису методом образних перетворень можливими є наступні правила алгебри логіки.

Склеювання змінних 2-містких термів ДІНФ-1 функції можна здійснювати за допомогою перетворення

$$(\overline{x_1} \rightarrow x_2)(\overline{x_1} \rightarrow \overline{x_2}) = x_1 = 1 \rightarrow x_1. \quad (7)$$

Рівносильні перетворення для правила склеювання змінних 2-містких термів ДІНФ-1 (7) мають ілюстрацію комбінаторного образу (8).



Національний університет
водного господарства
та природокористування

$$\begin{vmatrix} 1 & 1 \\ 1 & 0 \end{vmatrix} = 1 \quad | = x_1 = 1 \rightarrow x_1. \quad (8)$$

Отримана мінімальна булева функція у ДІНФ-1 має вигляд:

$$f_{\text{ДІНФ-1}} = 1 \rightarrow x_1.$$

Склеювання змінних 3-містких термів ДІНФ-1 можна здійснювати за допомогою перетворення

$$(\overline{\overline{x_1} \rightarrow x_2 \rightarrow x_3})(\overline{\overline{x_1} \rightarrow x_2 \rightarrow x_3}) = \overline{x_1} \rightarrow x_3. \quad (9)$$

Рівносильні перетворення для правила склеювання змінних 3-містких термів ДІНФ-1 (9) мають ілюстрацію образу (10).



$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix} = 1 \quad | = x_1 + x_3 = \overline{x_1} \rightarrow x_3.$$

(10)

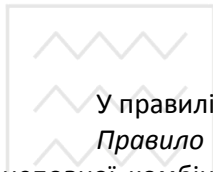
Правило супер-склеювання змінних. Для 4-містких термів ДІНФ-1 правило супер-склеювання змінних [4] може мати такий, наприклад, вигляд:



$$\begin{aligned}
 & \left(\overline{\overline{x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4}} \right) \left(\overline{\overline{x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow \bar{x}_4}} \right) \times \\
 & \times \left(\overline{\overline{x_1 \rightarrow x_2 \rightarrow \bar{x}_3 \rightarrow x_4}} \right) \left(\overline{\overline{x_1 \rightarrow x_2 \rightarrow \bar{x}_3 \rightarrow \bar{x}_4}} \right) = \\
 & = \left(\overline{\overline{x_1 \rightarrow x_2 \rightarrow x_3 + x_4}} \right) \left(\overline{\overline{x_1 \rightarrow x_2 \rightarrow x_3 + \bar{x}_4}} \right) \times \\
 & \times \left(\overline{\overline{x_1 \rightarrow x_2 \rightarrow \bar{x}_3 + x_4}} \right) \left(\overline{\overline{x_1 \rightarrow x_2 \rightarrow \bar{x}_3 + \bar{x}_4}} \right) = \\
 & = \left(\overline{\overline{x_1 \rightarrow x_2 + x_3 + x_4}} \right) \left(\overline{\overline{x_1 \rightarrow x_2 + x_3 + \bar{x}_4}} \right) \times \\
 & \times \left(\overline{\overline{x_1 \rightarrow x_2 + \bar{x}_3 + x_4}} \right) \left(\overline{\overline{x_1 \rightarrow x_2 + \bar{x}_3 + \bar{x}_4}} \right) = \\
 & = \left(\overline{\overline{x_1 + x_2 + x_3 + x_4}} \right) \left(\overline{\overline{x_1 + x_2 + x_3 + \bar{x}_4}} \right) \times \\
 & \times \left(\overline{\overline{x_1 + x_2 + \bar{x}_3 + x_4}} \right) \left(\overline{\overline{x_1 + x_2 + \bar{x}_3 + \bar{x}_4}} \right) = \\
 & = \left(\overline{\overline{x_1 + x_2 + x_3}} \right) \left(\overline{\overline{x_1 + x_2 + x_3}} \right) \times \\
 & \times \left(\overline{\overline{x_1 + x_2 + \bar{x}_3}} \right) \left(\overline{\overline{x_1 + x_2 + \bar{x}_3}} \right) = \\
 & = \left(\overline{\overline{x_1 + x_2}} \right) \left(\overline{\overline{x_1 + x_2}} \right) \left(\overline{\overline{x_1 + x_2}} \right) \left(\overline{\overline{x_1 + x_2}} \right) = \\
 & = \overline{\overline{x_1 + x_2}} = x_1 \rightarrow x_2.
 \end{aligned} \tag{11}$$

Рівносильні перетворення для правила супер-склеювання змінних 4-містких термів ДІНФ-1 (11) мають ілюстрацію образу (12):

$$\left| \begin{array}{cccc|c}
 0 & 1 & 1 & 1 & \\
 0 & 1 & 1 & 0 & \\
 0 & 1 & 0 & 1 & \\
 0 & 1 & 0 & 0 &
 \end{array} \right| = \overline{\overline{x_1 + x_2}} = x_1 \rightarrow x_2. \tag{12}$$



У правилі (28) використовується 2-(2, 4)-design [4].

Правило неповного супер-склеювання змінних. Комбінаторні властивості неповної комбінаторної системи з повторенням 2-(n, x/b)-design [4] забезпечують правило неповного супер-склеювання змінних в імплікативному базисі.

Для 2-містких термів ДІНФ-1 правило неповного супер-склеювання змінних може мати такий, наприклад, вигляд:

$$\begin{aligned}
 f(x_1, x_2) &= \left(\overline{\overline{x_1 \rightarrow x_2}} \right) \left(x_1 \rightarrow x_2 \right) \left(x_1 \rightarrow \overline{\overline{x_2}} \right) = \\
 &= \left(x_1 + x_2 \right) \left(\overline{\overline{x_1 + x_2}} \right) \left(\overline{\overline{x_1 + \bar{x}_2}} \right) = \\
 &= x_2 \left(\overline{\overline{x_1 + \bar{x}_2}} \right) = \overline{\overline{x_1 x_2}} = \overline{\overline{x_1 + x_2}} = \\
 &= \overline{\overline{x_1 \rightarrow x_2}}.
 \end{aligned} \tag{13}$$

Рівносильні перетворення для правила неповного супер-склеювання змінних 2-містких термів ДІНФ-1 (13) мають ілюстрацію образу (14)

$$\begin{aligned} \left| \begin{array}{cc|c} 1 & 1 & \\ 0 & 1 & \\ 0 & 0 & \end{array} \right| &= \left| \begin{array}{cc|c} & 1 & \\ 0 & 0 & \\ 0 & 0 & \end{array} \right| = \left| \begin{array}{cc|c} & & 1 \\ 0 & & \\ 0 & & \end{array} \right| = \\ &= \overline{x_1}x_2 = \overline{x_1 + x_2} = \overline{x_1 \rightarrow x_2}. \end{aligned} \quad (14)$$

У правилі (14) використовується 2-(2, 3/4)-design [4].

Узагальнене склеювання змінних в імплікативному базисі можна здійснювати за допомогою перетворення



Національний університет
водного господарства
та природокористування

$$(x_1 \rightarrow \overline{x_2})(x_1 \rightarrow x_3)(x_2 \rightarrow \overline{x_3}) = (x_1 \rightarrow x_3)(x_2 \rightarrow \overline{x_3}). \quad (15)$$

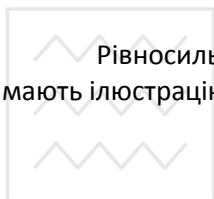
Рівносильні перетворення для правила узагальненого склеювання змінних (15) мають ілюстрацію образу (16).

$$\begin{aligned} \left| \begin{array}{cc|c} 0 & 0 & \\ 0 & 1 & \\ 0 & 0 & \end{array} \right| &= \left| \begin{array}{cc|c} 0 & & 1 \\ 0 & 0 & 0 \\ 0 & & \end{array} \right| = \\ &= (\overline{x_1 + x_3})(\overline{x_2 + x_3}) = (x_1 \rightarrow x_3)(x_2 \rightarrow \overline{x_3}). \end{aligned} \quad (16)$$

Правило поглинання змінних зводиться до перетворень:

$$1. (\overline{x_1} \rightarrow 0)(\overline{x_1} \rightarrow x_2) = \overline{x_1} \rightarrow 0. \quad (17)$$

Рівносильні перетворення для правила поглинання змінних ДІНФ-1 (17) мають ілюстрацію образу (18).



Національний університет
водного господарства
та природокористування

$$\begin{aligned} (\overline{x_1} \rightarrow 0)(\overline{x_1} \rightarrow x_2) &= x_1(x_1 + x_2) = \\ &= \left| \begin{array}{cc|c} 1 & & \\ 1 & 1 & \end{array} \right| = \left| \begin{array}{cc|c} & & 1 \\ 1 & 1 & \end{array} \right| = x_1 = \overline{x_1} \rightarrow 0. \end{aligned} \quad (18)$$

$$2. (x_1 \rightarrow 0)(x_1 \rightarrow x_2) = x_1 \rightarrow 0. \quad (19)$$

$$3. (x_1 \rightarrow x_2)(\overline{x_1 \rightarrow x_2 \rightarrow x_3}) = x_1 \rightarrow x_2. \quad (20)$$

$$\left| \begin{array}{ccc|c} 0 & 1 & & \\ 0 & 1 & 1 & \end{array} \right| = \left| \begin{array}{ccc|c} & 1 & & \\ 0 & 1 & & \end{array} \right| = \overline{x_1 + x_2} = x_1 \rightarrow x_2.$$

$$4. (x_1 \rightarrow x_2)(\overline{\overline{x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4}}) = x_1 \rightarrow x_2. \quad (21)$$

$$\left| \begin{array}{cccc|c} 0 & 1 & & & \\ 0 & 1 & 1 & 1 & \end{array} \right| = \left| \begin{array}{cccc|c} & 1 & & & \\ 0 & 1 & & & \end{array} \right| = \overline{x_1 + x_2} = x_1 \rightarrow x_2.$$

Правило напівсклеювання змінних в імплікативному базисі можна здійснювати за допомогою наступних перетворень:

$$(\overline{x_1} \rightarrow x_2)(\overline{x_1 \rightarrow x_2} \rightarrow x_3) = (\overline{x_1} \rightarrow x_2)(\overline{x_2} \rightarrow x_3). \quad (22)$$

Правило напівсклеювання змінних (22) має ілюстрацію образу (23):



Національний університет
водного господарства
та природокористування

$$\begin{aligned} \begin{vmatrix} 1 & 1 \\ 0 & 1 \end{vmatrix} &= \begin{vmatrix} 1 & 1 \\ 1 & 1 \end{vmatrix} = \\ &= (x_1 + x_2)(x_2 + x_3) = (\overline{x_1} \rightarrow x_2)(\overline{x_2} \rightarrow x_3). \end{aligned} \quad (23)$$

Приклад 3. Спростити логічну функцію $f(x_1, x_2, x_3, x_4)$ (табл. 4) у досконалій імплікативній нормальній формі –1 (ДІНФ-1).

Таблиця 4

Таблиця істинності функції $f(x_1, x_2, x_3, x_4)$

№	x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$	№	x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
0	0	0	0	0	1	8	1	0	0	0	1
1	0	0	0	1	1	9	1	0	0	1	0
2	0	0	1	0	0	10	1	0	1	0	0
3	0	0	1	1	0	11	1	0	1	1	1
4	0	1	0	0	0	12	1	1	0	0	0
5	0	1	0	1	0	13	1	1	0	1	0
6	0	1	1	0	1	14	1	1	1	0	1
7	0	1	1	1	0	15	1	1	1	1	1

Мінімізація ДІНФ-1 $f(x_1, x_2, x_3, x_4)$ проводиться наступними образними перетвореннями:



$$F_{\text{МНФ-1}} = \begin{array}{c|ccc|ccc} 2 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 3 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 4 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 5 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 7 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 9 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 10 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 12 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 13 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \end{array} = \begin{array}{c|ccc|ccc} & 1 & 1 & 0 & 1 & & & & \\ & 1 & 1 & 0 & 0 & & & & \\ & 1 & 0 & 1 & 1 & & & & \\ & 1 & 0 & 1 & 0 & & & & \\ & 1 & 0 & 0 & 0 & & & & \\ & 0 & 1 & 1 & 0 & & & & \\ & 0 & 1 & 0 & 1 & & & & \\ & 0 & 0 & 1 & 1 & & & & \\ & 0 & 0 & 1 & 0 & & & & \end{array} = \begin{array}{c|ccc|ccc} & 1 & 0 & 1 & & 1 & 0 & 1 & \\ & 1 & 0 & 0 & & 0 & 1 & & \\ & 0 & 1 & 0 & & 0 & 1 & 0 & \end{array} =$$

$$\begin{aligned} &= (\overline{x_2 + x_3 + x_4})(\overline{x_1 + x_3 + x_4}) \times \\ &\quad \times (\overline{x_2 + x_3})(\overline{x_1 + x_3 + x_4}) = \\ &= (\overline{x_2 + x_3 \rightarrow x_4})(\overline{x_1 + x_3 \rightarrow x_4}) \times \\ &\quad \times (x_2 \rightarrow x_3)(\overline{x_1 + x_3 \rightarrow x_4}) = \\ &= (\overline{x_2 \rightarrow x_3 \rightarrow x_4})(\overline{x_1 \rightarrow x_3 \rightarrow x_4}) \times \\ &\quad \times (x_2 \rightarrow x_3)(\overline{x_1 \rightarrow x_3 \rightarrow x_4}). \end{aligned}$$

МНФ-1 функції $f(x_1, x_2, x_3, x_4)$ (табл. 10):

$$\begin{aligned} F_{\text{МНФ-1}} &= (\overline{x_2 \rightarrow x_3 \rightarrow x_4})(\overline{x_1 \rightarrow x_3 \rightarrow x_4}) \times \\ &\quad \times (x_2 \rightarrow x_3)(\overline{x_1 \rightarrow x_3 \rightarrow x_4}). \end{aligned} \tag{24}$$

Висновки

1. Досконалу нормальну форму n -місткої функції імплікативного базису можна подати бінарними наборами (19) або матрицею (20), яка у цьому випадку буде подавати терми функції імплікації та операцію кон'юнкції для них. Таку герменевтику доцільно застосовувати при спрощенні логічних функцій та при виведенні результату логічних операцій у класі бінарних матриць функцій імплікативного базису.

2. Для належного спрощення функцій імплікативного базису методом образних перетворень була розроблена алгебра імплікативного базису у частині правил спрощення ДІНФ-1, ДІНФ-1.1, ДІНФ-2 та ДІНФ-2.1 булевих функцій імплікативного базису. Створення алгебри імплікативного базису вирішує проблему мінімізації функцій в імплікативному базисі.

3. Рівносильні перетворення комбінаторними образами, що за своїми властивостями мають більшу інформаційну ємність, спроможні з ефектом замінити вербальні процедури алгебричних перетворень. Алгебра, що створена, у складі правил спрощення функцій імплікативного базису забезпечує безпосереднє перетворення логічних виразів. А метод образних перетворень, використовуючи наглядні комбінаторні структури бінарних матриць та уніфікацію оригінальних процедур, забезпечує належне перетворення логічних виразів та функцій. У свою чергу, вербальні процедури мають меншу інформаційну ємність, вимагають

активного спостереження, що створює початок для не виявлення логічних операції (наприклад, узагальненого склеювання змінних, супер-склеювання змінних, неповного супер-склеювання змінних, напівсклеювання змінних), а відтак, зменшують можливості аналітичного методу.

1. Різник В. В., Соломко М. Т., Тадеєв П. О., Назарук В. Д., Зубик Л. В., Волошин В. С. Алгоритм мінімізації булевих функцій методом оптимального комбінування послідовності образних перетворень. *Східно-Європейський журнал передових технологій*. 2020. № 3/4 (105). С. 43–51. 2. Поспелов Д. А. Логические методы анализа и синтеза схем. Изд. 3, перераб. и доп. М. : «Энергия», 1974. 368 с. 3. 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. 4. Різник В. В., Соломко М. Т. Застосування алгебричної операції супер-склеювання змінних для мінімізації булевих функцій комбінаторним методом. *Технологічний аудит та резерви виробництва*. 2017. Вип. 6/2 (38). Р. 60–76.

Solomko M. T., Candidate of Engineering (Ph.D.), Associate Professor; Patiichuk N. B., Senior Student (National University of Water and Environmental Engineering, Rivne)

SIMPLIFICATION OF BOOL FUNCTIONS IN THE IMPLIATIVE BASIS WITH THE HELP OF IMAGE TRANSFORMATIONS

The article considers the minimization of Boolean functions in the implicative basis by the method of figurative transformations. The possibility of reducing the computational complexity, increasing the productivity of simplification of Boolean functions in the class of perfect implicative normal forms (DINF-1 and DINF-2) by the method of figurative transformations is established.

Keywords: implicative basis, minimization of Boolean functions, figurative transformations, functional completeness.

УДК 519.6

Бойчура¹ М. В., к.т.н; Абдулалі А. А., аспірант (Національний університет водного господарства та природокористування, м. Рівне,

¹m.v.boichura@nuwm.edu.ua)

ОДИН ПІДХІД ДО ІДЕНТИФІКАЦІЇ КОЕФІЦІЄНТА ФІЛЬТРАЦІЇ ШАРУВАТОГО СЕРЕДОВИЩА ЧИСЛОВИМИ МЕТОДАМИ КВАЗІКОНФОРМНИХ ВІДОБРАЖЕНЬ

Запропоновано підхід до реконструкції зображення шаруватого середовища числовими методами квазіконформних відображень у випадку, коли апріорно відомими є потенціали в окремих точках розриву значень коефіцієнта фільтрації

та повна витрата. Наближене розв'язання відповідної задачі здійснюється за умови почергової ітераційної побудови полів швидкостей та ідентифікації параметрів. Проведено числові експерименти та здійснено їх аналіз. Результати реконструкції зображення в розумних, з точки зору практики, межах відповідають апріорно відомим. Встановлено, що наявні несуттєві відхилення у розв'язку можливо зменшити, збільшивши кількість вузлів розбиття гідродинамічної сітки.

Ключові слова: нелінійні задачі, ідентифікація, реконструкція зображення, квазіконформні відображення, числові методи.



Національний університет
водних ресурсів та природокористування

Вступ. На сьогодні, існує низка праць стосовно моделювання фільтраційної течії у різного роду середовищах [1]. Проте, на практиці, часто виникають ситуації, коли не вистачає якихось відомостей для побудови відповідної картини руху рідини. Зазвичай невідомим є коефіцієнт фільтрації (КФ) [2] (чи, згідно принципу електрогідродинамічної аналогії, коефіцієнт електричної провідності [3]). У такому випадку, доцільно розв'язувати одночасно 2 задачі: моделювання руху рідини та ідентифікації (реконструкції зображення) невідомих (шуканих) параметрів. При цьому, доцільно, щоб відповідний алгоритм надавав можливість задання межі досліджуваного середовища у вигляді «довільної» наперед заданої кривої [4].

Практичний інтерес представляють задачі ідентифікації КФ, у яких відомими є потенціали в окремих точках його розриву та повна витрата. У роботі [2] проводились подібні дослідження. Там виходили з припущення, що лінії розриву значень КФ співпадають з еквіпотенціалами. Такий підхід дозволяє досить швидко розраховувати шукані параметри із високою точністю. У випадку ж наявності великої кількості неспівпадінь ліній розриву значень КФ з еквіпотенціалами, доводиться шукати прийнятний баланс між швидкістю розрахунків і їх точністю. Ми ж пропонуємо розвинути підхід до ідентифікації параметрів КФ, запропонований у [2], на випадки великої кількості ліній розриву, які не обов'язково співпадають з еквіпотенціалами; при цьому забезпечуючи прийнятну швидкість і високу точність розрахунків.

Постановка задачі. Сформулюємо задачу ідентифікації КФ кусково-однорідного пласта $L^* = \{z = x + iy : x = x^*(\tau), y = y^*(\tau), 0 \leq \tau < 1, x^*(0) = x^*(1-0) = x_0^*, y^*(0) = y^*(1-0) = y_0^*\}$ (рис. 1, а) при наявності у ньому свердловини типу експлуатаційної $L_* = \{z = x + iy : x = x_*(\tau), y = y_*(\tau), 0 \leq \tau < 1, x_*(0) = x_*(1-0) = x_{*0}, y_*(0) = y_*(1-0) = y_{*0}\}$, де $x^*(\tau), y^*(\tau), x_*(\tau), y_*(\tau), O^*(x_0^*, y_0^*), O_*(x_{*0}, y_{*0})$ – визначені неперервно диференційовані функції і задані точки відліку на відповідних кривих. Як відомо, рух рідини (наприклад, нафтогазової суміші) у пласті, в процесі викачування, відбувається у відповідності до закону Дарсі $\vec{v} = \kappa \text{grad} \varphi$ та, часто, рівняння нерозривності $\text{div} \vec{v} = 0$ [1]. Тут $\varphi = \varphi(x, y)$ – квазіпотенціал, $\vec{v} = (v_x, v_y)$ – вектор швидкості, κ – КФ. Як відомо [1], відповідна пряма модельна задача руху

рідини записується в однозв'язній області G_z (рис. 1, а), і полягає у знаходженні комплексного спряжених функцій $\varphi = \varphi(x, y)$ та $\psi = \psi(x, y)$ (течії) [1; 2] при умовах

$$\kappa \frac{\partial \varphi}{\partial x} = \frac{\partial \psi}{\partial y}, \quad \kappa \frac{\partial \varphi}{\partial y} = -\frac{\partial \psi}{\partial x}; \quad (1)$$

$$\varphi|_{AB} = \varphi^*, \quad \varphi|_{CD} = \varphi^*; \quad (2)$$

$$\psi|_{AD} = 0, \quad \psi|_{BC} = Q, \quad (3)$$

де $G_\omega = \{(\varphi, \psi) : \varphi^* \leq \varphi \leq \varphi^*, 0 \leq \psi \leq Q\}$ (рис. 1, б) – область комплексного квазіпотенціалу; Q – потік векторного поля (рідини) через контактні ділянки AB і CD (наприклад, для AB : $Q = \int_{AB} -\frac{\partial \varphi}{\partial y} dx + \frac{\partial \varphi}{\partial x} dy$); AD і BC – непроникні

граничні лінії течії; КФ $\kappa = \kappa(\varphi)$ задаємо у вигляді, аналогічному до [2]:

$$\kappa = \begin{cases} \tilde{\kappa}_1, & \text{при } \tilde{\varphi}_0 = \varphi^* \leq \varphi \leq \tilde{\varphi}_1, \\ \tilde{\kappa}_2, & \text{при } \tilde{\varphi}_1 < \varphi \leq \tilde{\varphi}_2, \\ \dots \\ \tilde{\kappa}_s, & \text{при } \tilde{\varphi}_{s-1} < \varphi \leq \varphi^* = \tilde{\varphi}_s; \end{cases} \quad (4)$$

$\tilde{\kappa}_k$ ($k = \overline{1, s}$), $\tilde{\varphi}_k$ ($k = \overline{0, s}$) та s – задані сталі.

Ідентифікацію КФ (4) здійснюватимемо за умови відшукання $\tilde{\kappa}_k$ ($k = \overline{1, s}$) при заданих апіорно відомих повній витраті \tilde{Q} і потенціалах $\tilde{\varphi}_k$ у деяких точках $(\tilde{x}_k, \tilde{y}_k) \in G_z$ ($k = \overline{1, s-1}$) та $(\tilde{x}_0, \tilde{y}_0) \in L_*$, $(\tilde{x}_s, \tilde{y}_s) \in L^*$. При цьому, не накладаються додаткові обмеження типу [2] на розміщення точок $(\tilde{x}_k, \tilde{y}_k) \in G_z \cup \partial G_z$.

Як слідує з [1], зручно перейти від задачі (1)–(4) до наступної:

$$\begin{cases} \frac{\partial}{\partial \varphi} \left(\frac{1}{\kappa} \frac{\partial x}{\partial \varphi} \right) + \kappa \frac{\partial^2 x}{\partial \psi^2} = 0, \\ \frac{\partial}{\partial \varphi} \left(\frac{1}{\kappa} \frac{\partial y}{\partial \varphi} \right) + \kappa \frac{\partial^2 y}{\partial \psi^2} = 0, \quad \forall (x, y) \in G_z; \end{cases} \quad (5)$$

$$\begin{aligned} x(\varphi^*, \psi) &= x_*(\tau_{AB}(\psi)), \quad y(\varphi^*, \psi) = y_*(\tau_{AB}(\psi)), \\ x(\varphi^*, \psi) &= x^*(\tau_{CD}(\psi)), \quad y(\varphi^*, \psi) = y^*(\tau_{CD}(\psi)), \quad \psi \in [0, Q], \\ x(\varphi, 0) &= x(\varphi, Q), \quad y(\varphi, 0) = y(\varphi, Q), \quad \varphi \in [\varphi^*, \varphi^*]; \end{aligned} \quad (6)$$

$$x'_\varphi(\varphi^*, \psi) x'_\psi(\varphi^*, \psi) + y'_\varphi(\varphi^*, \psi) y'_\psi(\varphi^*, \psi) = 0, \quad (7)$$

$$x'_\varphi(\varphi^*, \psi) x'_\psi(\varphi^*, \psi) + y'_\varphi(\varphi^*, \psi) y'_\psi(\varphi^*, \psi) = 0, \quad \psi \in [0, Q]$$

за умови відшукування комплексно спряжених функцій $x = x(\varphi, \psi)$ і $y = y(\varphi, \psi)$ та ідентифікації КФ (4) [2; 3]. Тут $\tau = \tau_{AB}(\psi)$, $\tau = \tau_{CD}(\psi)$ – розрахункові залежності параметрів задання граничних вузлів від значень функції течії на ділянках AB , CD , відповідно.

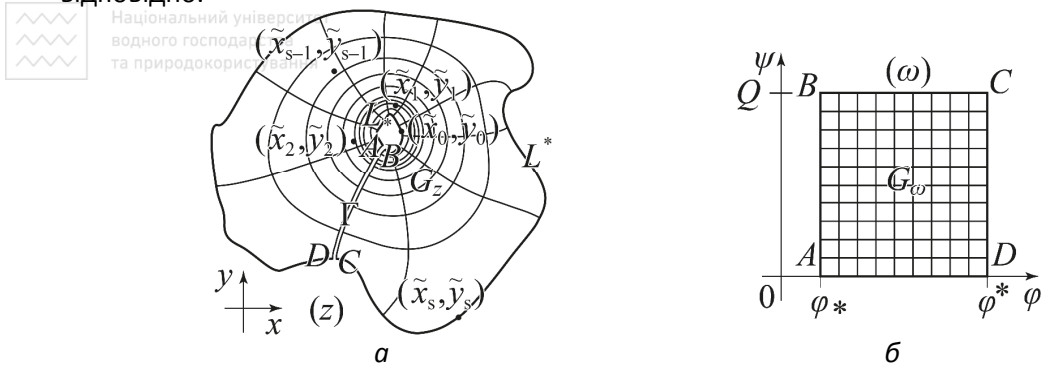


Рис. 1. Фізична область (а) і відповідна область комплексного квазіпотенціалу (б)

Ідентифікацію КФ (4) шаруватого середовища пропонуємо здійснювати, скориставшись виразом $\kappa \frac{\partial \varphi}{\partial l} = \frac{\partial \psi}{\partial s}$ [3], де ∂l та ∂s – елементи довжин вздовж ліній течії та еквіпотенціальних ліній, відповідно. А саме: за формулою

$$\tilde{\kappa}_k = \frac{1}{\tilde{\varphi}_k - \tilde{\varphi}_{k-1}} \int_{(\tilde{x}_{k-1}, \tilde{y}_{k-1})}^{(\tilde{x}_k, \tilde{y}_k)} \frac{\partial \psi}{\partial s} dl \quad (k = \overline{1, s}). \quad (8)$$

Різницеві аналоги. Наближений розв'язок задачі (4)–(8) побудуємо із застосуванням різницевих її аналогів в областях $G_\omega^\gamma = \{(\varphi_i, \psi_j) : \varphi_i = \varphi^* + i \Delta \varphi, i = \overline{0, m+1}; \psi_j = j \Delta \psi, j = \overline{0, n+1}; \Delta \varphi = (\varphi^* - \varphi_*) / (m+1), \Delta \psi = Q / (n+1); \gamma = \Delta \varphi / \Delta \psi; m, n \in \mathbb{N}\}$, де m і n – кількості вузлів розбиття, квазіконформний інваріант визначається наступним чином [1]:

$$\gamma = \frac{1}{(m+1)(n+1)} \sum_{i,j=0}^{m,n} \frac{1}{\kappa_{i+1/2}} \frac{a_{i,j} + a_{i,j+1}}{b_{i,j} + b_{i+1,j}}. \quad (9)$$

Тут, для скорочення запису, введені позначення $x_{i,j} = x(\varphi_i, \psi_j)$, $y_{i,j} = y(\varphi_i, \psi_j)$ ($i = \overline{0, m+1}$, $j = \overline{0, n+1}$), $x_{i,-1} = x(\varphi_i, Q)$, $y_{i,-1} = y(\varphi_i, Q)$, $x_{i, n+2} = x(\varphi_i, 0)$, $y_{i, n+2} = y(\varphi_i, 0)$ ($i = \overline{0, m+1}$), $\kappa_{i \pm 1/2} = \kappa(0.5(\varphi_{i \pm 1} + \varphi_i))$,

$$a_{i,j} = \sqrt{(x_{i+1,j} - x_{i,j})^2 + (y_{i+1,j} - y_{i,j})^2}, \quad \kappa_i = \kappa(\varphi_i), \quad z_{i,j} = x_{i,j} + i y_{i,j},$$

$$b_{i,j} = \sqrt{(x_{i,j+1} - x_{i,j})^2 + (y_{i,j+1} - y_{i,j})^2}. \text{ Зокрема, узагальнення рівнянь Лапласа}$$

(5) запишемо у вигляді, «приспосованому» до кусково-однорідних середовищ, використавши консервативні різницеві схеми [1] при $i = \overline{1, m}, j = \overline{0, n+1}$:

$$x_{i,j} = \left(x_{i-1,j} \kappa_{i-1/2,j}^{-1} + x_{i+1,j} \kappa_{i+1/2,j}^{-1} + (x_{i,j-1} + x_{i,j+1}) \gamma^2 \kappa_{i,j} \right) / \left(\kappa_{i-1/2,j}^{-1} + \kappa_{i+1/2,j}^{-1} + 2\gamma^2 \kappa_{i,j} \right), \quad (10)$$

$$y_{i,j} = \left(y_{i-1,j} \kappa_{i-1/2,j}^{-1} + y_{i+1,j} \kappa_{i+1/2,j}^{-1} + (y_{i,j-1} + y_{i,j+1}) \gamma^2 \kappa_{i,j} \right) / \left(\kappa_{i-1/2,j}^{-1} + \kappa_{i+1/2,j}^{-1} + 2\gamma^2 \kappa_{i,j} \right).$$

Апроксимацію крайових умов (6) та (7) здійснимо наступним чином [1]:

$$x_{i,0} = x_{i,n+1}, \quad y_{i,0} = y_{i,n+1}, \quad i = \overline{0, m+1}; \quad (11)$$

$$x_{0,j} = x^* (\tau_{AB_j}), \quad y_{0,j} = y^* (\tau_{AB_j}), \quad (12)$$

$$x_{m+1,j} = x^* (\tau_{CD_j}), \quad y_{m+1,j} = y^* (\tau_{CD_j}), \quad j = \overline{0, n+1};$$

$$\begin{aligned} & (3x_{m+1,j} + x_{m-1,j} - 4x_{m,j})(x_{m+1,j+1} - x_{m+1,j-1}) + (3y_{m+1,j} + y_{m-1,j} - \\ & - 4y_{m,j})(y_{m+1,j+1} - y_{m+1,j-1}) = 0, \quad (4x_{1,j} - 3x_{0,j} - x_{2,j})(x_{0,j+1} - \\ & - x_{0,j-1}) + (4y_{1,j} - 3y_{0,j} - y_{2,j})(y_{0,j+1} - y_{0,j-1}) = 0, \quad j = \overline{0, n+1}, \end{aligned} \quad (13)$$

де $\tau_{AB_j} = \tau_{AB}(\psi_j)$, $\tau_{CD_j} = \tau_{CD}(\psi_j)$.

Вираз (8) можемо наблизити, застосувавши формули лівих прямокутників до відповідних інтегралів [3]. Проте, з метою кращого врахування властивостей розрахункової сітки, (8) перепишемо у вигляді:

$$\begin{aligned} \tilde{\kappa}_k &= \frac{(n+1)^{-2} \tilde{Q}}{\tilde{\varphi}_k - \tilde{\varphi}_{k-1}} \sum_{j=0}^n \left(\frac{d(\tilde{z}_{k-1,j}, z_{m_{k-1}+1,j}) + d(\tilde{z}_{k-1,j+1}, z_{m_{k-1}+1,j+1})}{d(\tilde{z}_{k-1,j}, \tilde{z}_{k-1,j+1}) + b_{m_{k-1}+1,j}} + \right. \\ & \left. + \sum_{i=m_{k-1}+1}^{m_k-1} \frac{a_{i,j} + a_{i,j+1}}{b_{i,j} + b_{i+1,j}} + \frac{d(z_{m_k,j}, \tilde{z}_{k,j}) + d(z_{m_k,j+1}, \tilde{z}_{k,j+1})}{b_{m_k,j} + d(\tilde{z}_{k,j}, \tilde{z}_{k,j+1})} \right) \quad (k = \overline{1, s}), \end{aligned} \quad (14)$$

$$\text{де } d(z_{i_1,j_1}, z_{i_2,j_2}) = \sqrt{(x_{i_1,j_1} - x_{i_2,j_2})^2 + (y_{i_1,j_1} - y_{i_2,j_2})^2}, \quad \tilde{z}_{k,j} = \tilde{x}_{k,j} + i \tilde{y}_{k,j},$$

$$(\tilde{x}_{k,j}, \tilde{y}_{k,j}) = (x(\tilde{\varphi}_k, \psi_j), y(\tilde{\varphi}_k, \psi_j)), \quad m_k = [(\tilde{\varphi}_k - \varphi_*) / \Delta \varphi] \quad (k = \overline{0, s}).$$

Алгоритм розв'язання задачі. Опишемо алгоритм розв'язання різницевої задачі (9)–(14). Вважатимемо далі, що $\gamma^{(r)}$, $(x_{i,j}^{(r)}, y_{i,j}^{(r)})$, $(\tilde{x}_{k,j}^{(l)}, \tilde{y}_{k,j}^{(l)})$, $\tilde{\varphi}_k^{(l)}$, $\tilde{\kappa}_k^{(l)}$ є r -ми та l -ми ($r = 0, 1, \dots$, $l = 0, 1, \dots$) наближеннями відповідних величин γ , $(x_{i,j}, y_{i,j})$, $(\tilde{x}_{k,j}, \tilde{y}_{k,j})$, $\tilde{\varphi}_k$, $\tilde{\kappa}_k$. Спочатку задаємо границі L_* та L^* досліджуваної фізичної області функціями $x = x_*(\tau)$, $y = y_*(\tau)$, $x = x^*(\tau)$,

$y = y^*(\tau)$, кількості вузлів розбиття m , n , параметр точності δ , значення потенціалів $\varphi|_{L_*} = \varphi_*$, $\varphi|_{L^*} = \varphi^*$ та $\varphi = \tilde{\varphi}_k$ у відомих точках $(\tilde{x}_k, \tilde{y}_k)$ ($k = \overline{0, s}$), повну витрату \tilde{Q} ; обчислюємо $\Delta\varphi = (\varphi^* - \varphi_*) / (m + 1)$.

Задаємо початкові наближення $\tilde{\kappa}_k^{(0)}$ ($k = \overline{1, s}$) та $\tilde{\varphi}_k^{(0)}$ ($k = \overline{0, s}$), координат граничних і внутрішніх вузлів $(x_{i,j}^{(0)}, y_{i,j}^{(0)})$ ($i = \overline{0, m+1}$, $j = \overline{0, n+1}$); обчислюємо $\gamma^{(0)}$ за формулою (9).

Розпочинаємо ітераційний процес реконструкції зображення за умови почергової побудови полів швидкостей та ідентифікації параметрів КФ [2; 3]. Для цього спочатку здійснюємо уточнення граничних і внутрішніх вузлів $(x_{i,j}^{(r)}, y_{i,j}^{(r)})$ ($i = \overline{0, m+1}$, $j = \overline{0, n+1}$) та квазіконформного інваріанту $\gamma^{(r)}$ за формулами (11)–(13), (10) та (9), відповідно, до тих пір, доки не досягнемо виконання умов завершення ітераційного процесу (див., напр., [1]). Далі обчислюємо потенціали $\tilde{\varphi}_k^{(l)}$ у точках $(\tilde{x}_k, \tilde{y}_k)$, після чого знаходимо вузли $(\tilde{x}_{k,j}^{(l)}, \tilde{y}_{k,j}^{(l)})$, в яких $\varphi = \tilde{\varphi}_k^{(l)}$ ($k = \overline{0, s}$, $j = \overline{0, n+1}$). Шукаємо наступне наближення параметрів $\tilde{\kappa}_k^{(l)}$ КФ (4) за формулою (14) при $\tilde{\varphi}_k = \tilde{\varphi}_k^{(l)}$. Якщо $|\tilde{\kappa}_k^{(l)} - \tilde{\kappa}_k^{(l-1)}| < \delta$, то завершуємо ітераційний процес реконструкції зображення і виводимо на екран отримані результати, інакше – повертаємось до уточнення координат граничних і внутрішніх вузлів та квазіконформного інваріанту.

Значимо, що аналогічним є алгоритм у випадку іншого числа зв'язності області G_z .

Числові експерименти. Наведемо нижче результати розрахунків, отриманих у процесі розв'язання задачі (9)–(14). Криву L^* задаємо у вигляді кола $x_*(\tau) = 550 + 10 \cos(0.5\tau / \pi)$, $y_*(\tau) = 400 + 10 \sin(0.5\tau / \pi)$, а L^* –

бісплайновими функціями $x^*(\tau) = \sum_{\xi=0}^{m^*} x_{\xi}^* b_{\xi,3}^*(\tau)$, $y^*(\tau) = \sum_{\xi=0}^{m^*} y_{\xi}^* b_{\xi,3}^*(\tau)$ [4] за

допомогою $m^* = 16$ контрольних точок

$$\begin{aligned} (x_0^*, y_0^*) &= (x_{15}^*, y_{15}^*) = (219, 425), (x_1^*, y_1^*) = (140, 346), (x_2^*, y_2^*) = (179, 177), \\ (x_3^*, y_3^*) &= (307, 272), (x_4^*, y_4^*) = (479, 137), (x_5^*, y_5^*) = (647, 46), \\ (x_6^*, y_6^*) &= (872, 193), (x_7^*, y_7^*) = (752, 387), (x_8^*, y_8^*) = (876, 559), \\ (x_9^*, y_9^*) &= (850, 643), (x_{10}^*, y_{10}^*) = (626, 652), (x_{11}^*, y_{11}^*) = (599, 523), \\ (x_{12}^*, y_{12}^*) &= (418, 550), (x_{13}^*, y_{13}^*) = (252, 633), (x_{14}^*, y_{14}^*) = (330, 505). \end{aligned}$$

Значення параметрів $\tilde{Q} = 2.185735$, $(\tilde{x}_0, \tilde{y}_0) = (549.880938, 409.999291)$, $(\tilde{x}_1, \tilde{y}_1) = (601.875468, 416.714573)$, $(\tilde{x}_2, \tilde{y}_2) = (541.046342, 194.852104)$, $(\tilde{x}_3, \tilde{y}_3) = (210.313062, 415.85968)$ отримано за допомогою спеціально розробленої програми імітації фізичного експерименту при $s = 3$, $\tilde{\kappa}_1 = 1$, $\tilde{\kappa}_2 = 2$, $\tilde{\kappa}_3 = 0.5$, $\tilde{\varphi}_0 = \varphi^* = 0$, $\tilde{\varphi}_1 = 0.6$, $\tilde{\varphi}_2 = 0.8$, $\tilde{\varphi}_3 = \varphi^* = 1$, $m = 44$, $n = 97$.

«Ключовий» результат реконструкції зображення при $m = 44$, $n = 99$, $\delta = 10^{-6}$ наведено на рис. 2, що відповідає параметрам $r = 242400$, $l = 10$,

$$\begin{aligned} \tilde{\kappa}_1^{(l)} &= 0.991774, \tilde{\kappa}_2^{(l)} = 1.979118, \tilde{\kappa}_3^{(l)} = 0.492007, \tilde{\varphi}_0^{(l)} = 0, \tilde{\varphi}_1^{(l)} = 0.59197, \\ \tilde{\varphi}_2^{(l)} &= 0.79621, \tilde{\varphi}_3^{(l)} = 1. \end{aligned}$$

При цьому, обчислено нев'язки складових КФ

$$\begin{aligned} |\tilde{\kappa}_1 - \tilde{\kappa}_1^{(l)}| &= 0.008226, |\tilde{\kappa}_2 - \tilde{\kappa}_2^{(l)}| = 0.020882, |\tilde{\kappa}_3 - \tilde{\kappa}_3^{(l)}| = 0.007993, \\ |\tilde{\varphi}_0 - \tilde{\varphi}_0^{(l)}| &= 0, |\tilde{\varphi}_1 - \tilde{\varphi}_1^{(l)}| = 0.00803, |\tilde{\varphi}_2 - \tilde{\varphi}_2^{(l)}| = 0.00379, |\tilde{\varphi}_3 - \tilde{\varphi}_3^{(l)}| = 0, \end{aligned}$$

а також ортогональності $\varepsilon_{\perp}^* = 0.034526$ і відношення довжин суміжних відрізків

$$\varepsilon_{\pm}^* = 0.068475 \text{ за формулами } \varepsilon_{\perp}^* = \sum_{i,j=0}^{m,n} \varepsilon_{\perp,i,j}^* \text{ (рис. 3, а) і } \varepsilon_{\pm}^* = \sum_{i,j=0}^{m,n} \varepsilon_{\pm,i,j}^*$$

(рис. 3, б), відповідно, із врахуванням квазіконформності відображення, де

$$\varepsilon_{\perp,i,j}^* = \left| \frac{\pi}{2} - \arccos \frac{(x_{i+1,j} - x_{i,j})(x_{i,j+1} - x_{i,j}) + (y_{i+1,j} - y_{i,j})(y_{i,j+1} - y_{i,j})}{a_{i,j} b_{i,j}} \right|,$$

$$\varepsilon_{\pm,i,j}^* = \begin{cases} a_{i,j} / (\gamma \kappa_i b_{i,j}) - 1, & \text{при } a_{i,j} > \gamma \kappa_i b_{i,j}, \\ \gamma \kappa_i b_{i,j} / a_{i,j} - 1, & \text{при } a_{i,j} < \gamma \kappa_i b_{i,j}. \end{cases}$$

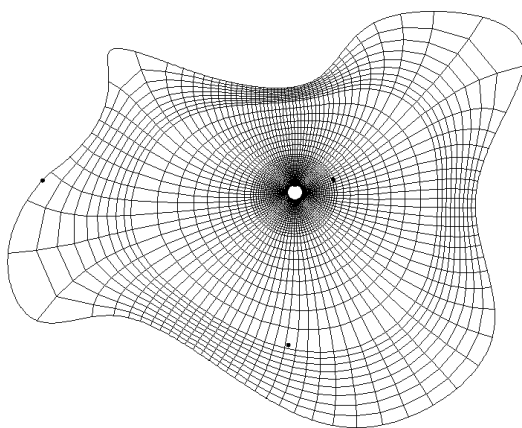


Рис. 2. Розрахована гідродинамічна сітка із наведеними апріорно відомими точками розриву значень коефіцієнта фільтрації

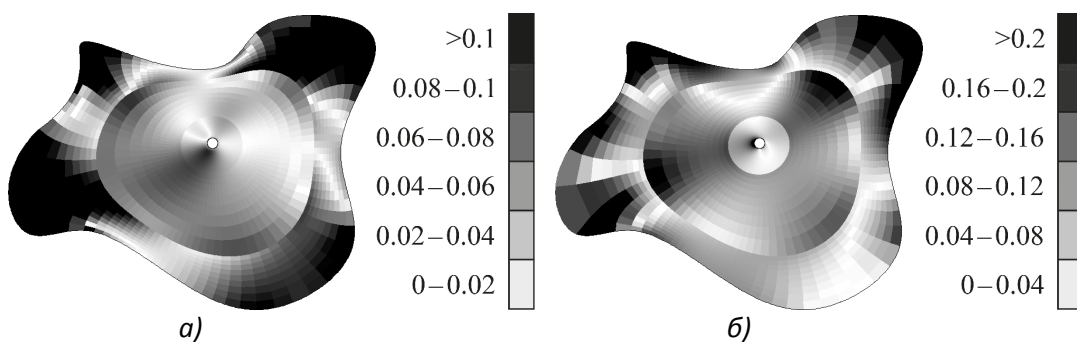


Рис. 3. Графічне зображення розподілів $\varepsilon_{\perp,i,j}^*$ (а) та $\varepsilon_{\dot{z},i,j}^*$ (б) в області G_z

Такі (хоч і незначні для практики) нев'язки пояснюються, в першу чергу, порушеннями умов квазіконформності (ε_{\perp}^* , $\varepsilon_{\dot{z}}^*$) з найбільшими їх значеннями у «застійних» зонах. Як наслідок, вихідні дані (отримані за допомогою програми імітації фізичного експерименту) задаються теж із певними похибками. Обидві перелічені проблеми у значній мірі можливо вирішити, збільшивши кількість вузлів розбиття [1] розрахункової області. З іншого боку, це негативно впливатиме на загальний час проведення розрахунків та на накопичення машинної похибки.

Висновки. Розроблено алгоритм реконструкції зображення шаруватого середовища при наявних відомостях щодо розподілу потенціалів в окремих точках розриву значень коефіцієнта фільтрації і заданої повної витрати. Ітераційний процес розв'язання відповідної задачі полягає у почерговому застосуванні числових методів квазіконформних відображень (для побудови гідродинамічних сіток) і уточненні коефіцієнта фільтрації. Останнє здійснюється за допомогою виразу, утвореного з умови рівності швидкостей за даними розрахунковими параметрами елементарних ліній течії та еквіпотенціальних ліній гідродинамічних сіток.

Наближені розв'язки представлено на прикладі двозв'язної області (хоча алгоритм неважко поширити на будь-яку зв'язність). Проведені розрахунки показують достатню для практики ефективність наведеного у роботі підходу. З

іншого боку, точність отримуваних розв'язків можливо підвищувати за умови збільшення кількості вузлів розбиття області (хоча швидкість розрахунків, при цьому, суттєво знижуватиметься).

У перспективі: перенесення алгоритму на тривимірні та неоднорідні середовища, зокрема, для ідентифікації параметрів у навколосвердловинному просторі [1].

1. Бомба А. Я., Каштан С. С., Пригорницький Д. О., Ярощак С. В. Методи комплексного аналізу : монографія. Рівне : Національний університет водного господарства та природокористування, 2013. 430 с.
2. Bomba A. Y., Hladka O. M. Problems of identification of the parameters of quasiideal filtration processes in nonlinear layered porous media. *Journal of Mathematical Sciences*. 2017. Vol. 220, No. 2. P. 213–225.
3. Bomba A., Boichura M. Identification of burst parameters using numerical quasiconformal mapping methods. *International Journal of Applied Mathematics*. 2020. Vol. 33, No. 5. P. 903–917.
4. Prautzsch H., Boehm W., Paluszny M. *Bézier and B-Spline Techniques*. Heidelberg : Springer, 2002. 304 p.

Boichura M. V., Candidate of Engineering (Ph.D.); Abdulali A. A., Post-graduate Student
(National University of Water and Environmental Engineering, Rivne)

ONE APPROACH TO IDENTIFICATION OF FILTRATION COEFFICIENT OF THE LAYERED MEDIA USING NUMERICAL QUASICONFORMAL MAPPING METHODS

An approach to the image reconstruction of the layered medium using numerical quasiconformal mapping methods is proposed for the case when the potentials at some points of separation of values of the filtration coefficient and the full discharges are a priori known. The approximate solving of the corresponding problem is carried out under the condition of alternate iterative velocity fields construction and parameters identification. Numerical experiments were performed and analyzed. The results of image reconstruction within reasonable, from the point of view of practice, limits correspond to a priori known. It was found that the existing insignificant deviations in the solution can be reduced by increasing the number of nodes of the hydrodynamic mesh.

Keywords: nonlinear problems, identification, image reconstruction, quasiconformal mappings, numerical methods.

Соломко¹ М. Т., к.т.н, доцент; Кирикович В. О., студент 4 курсу (Національний університет водного господарства та природокористування, м. Рівне,
¹m.t.solomko@nuwm.edu.ua)

ОПТИМАЛЬНЕ ЧЕРГУВАННЯ ЛОГІЧНИХ ОПЕРАЦІЙ ПРИ МІНІМІЗАЦІЇ БУЛЕВИХ ФУНКЦІЙ

В роботі розглянуто оптимальне чергування образних перетворень для процесу мінімізації булевих функцій. Ефективність обраної послідовності рівносильних перетворень демонструється на прикладах мінімізації булевих функцій, запозичених у роботах інших авторів для порівняння.

Ключові слова: мінімізація булевих функцій, образні перетворення, логічна операція супер-склеювання змінних.

Вступ. Алгебра логіки, як і будь-який апарат обчислення, є сукупність аксіом, тотожностей, законів, правил, які дозволяють здійснювати перетворення логічних виразів. Але нема тут приписів про те, як необхідно використовувати цей апарат для синтезу оптимальних логічних схем. І тільки вибудовування у певній послідовності цих перетворень (це – алгоритми) може забезпечити оптимальне рішення. Відомі методи такого вибудовування: Рота, Квайна – МакКласки, Петрика, Закревського Д.А., Рыцаря Б.Е. та ін. [1–5]. У свою чергу визначення алгоритмів спрощення є передумовою для створення автоматизованих методів мінімізації булевих функцій.

1. Особливості застосування комбінаторних структур 2-(n, b)-design та 2-(n, x/b)-design для мінімізації булевих функцій. Алгебричні перетворення, що необхідні для процесу мінімізації булевих функцій замінюються рівносильними перетвореннями за допомогою підматриць (комбінаторними образами) таблиці істинності, яка є власне комбінаторною системою. Оскільки комбінаторні образи дають більше інформації стосовно ортогональності, суміжності, однозначності блоків комбінаторної системи, порівняно з алгебричними перетвореннями, що є вербальною процедурою, застосування їх під час пошуку об'єктів для рівносильного перетворення, у процесі мінімізації логічної функції, є ефективним [6–9].

Для вибору оптимального чергування протоколів мінімізації образними перетвореннями необхідно визначити початкову операцію алгебричного перетворення булевих функцій. У зв'язку з цим потрібно встановити особливості процесу мінімізації логічних функцій при використанні комбінаторних структур 2-(n, b)-design та 2-(n, x/b)-design.

Приклад 1. Мінімізувати функцію $F(x_1, x_2, x_3, x_4)$ образними перетвореннями, що задана наступною таблицею істинності:

$$F = \Sigma(6, 8, 9, 10, 11, 12, 13, 14)$$

Примітка: значення в Σ є мінтермами для рядків, коли функція

$F(x_1, x_2, x_3, x_4)$ повертає «1» на виході.

Для мінімізації заданої функції $F(x_1, x_2, x_3, x_4)$ використаємо комбінаторну структуру неповної бінарної системи з повторенням 2-(n, x/b)-design (рис. 1).



Національний університет
водного господарства
та природокористування

6	0	1	1	0
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0

Рис. 1. Комбінаторна система (таблиця істинності) функції $F(x_1, x_2, x_3, x_4)$ зі структурою неповної бінарної системи з повторенням 2-(3, 7/8)-design (виділена червоним кольором)

Проведемо мінімізацію функції $F(x_1, x_2, x_3, x_4)$ образними перетвореннями, використовуючи 2-(3, 7/8)-design.



$$F = \begin{vmatrix} 6 & 0 & 1 & 1 & 0 \\ 8 & 1 & 0 & 0 & 0 \\ 9 & 1 & 0 & 0 & 1 \\ 10 & 1 & 0 & 1 & 0 \\ 11 & 1 & 0 & 1 & 1 \\ 12 & 1 & 1 & 0 & 0 \\ 13 & 1 & 1 & 0 & 1 \\ 14 & 1 & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & & \\ 1 & & 0 & \\ 1 & & & 0 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 0 \\ 1 & & 0 \\ 1 & & 0 \end{vmatrix}$$

Мінімізована функція:

$$F = x_2 x_3 \bar{x}_4 + x_1 \bar{x}_2 + x_1 \bar{x}_3 + x_1 \bar{x}_4 \tag{1}$$

Протокол мінімізації для 2-(3, 7/8)-design:

$$\begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & & \\ 1 & 0 & \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & & \\ & 0 & \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & & \\ & 0 & \\ & & 0 \end{vmatrix}$$

або

$$\begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 0 & 0 \\ 0 & 0 \end{vmatrix}$$



Національний університет
водного господарства
та природокористування

Приклад 2. Мінімізуємо логічну функцію $F(x_1, x_2, x_3, x_4)$ з прикладу 1 образними перетвореннями, використовуючи комбінаторну структуру повної бінарної системи з повторенням 2-(n, b)-design (рис. 2).

6	0	1	1	0
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0

Рис. 2. Комбінаторна система (таблиця істинності) функції $F(x_1, x_2, x_3, x_4)$ зі структурою повної бінарної системи з повторенням 2-(2, 4)-design (виділена червоним кольором)

Проведемо мінімізацію функції $F(x_1, x_2, x_3, x_4)$ образними перетвореннями, використовуючи 2-(2, 4)-design.

$$F = \begin{vmatrix} 6 & 0 & 1 & 1 & 0 \\ 8 & 1 & 0 & 0 & 0 \\ 9 & 1 & 0 & 0 & 1 \\ 10 & 1 & 0 & 1 & 0 \\ 11 & 1 & 0 & 1 & 1 \\ 12 & 1 & 1 & 0 & 0 \\ 13 & 1 & 1 & 0 & 1 \\ 14 & 1 & 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 \\ 1 & 1 & 0 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{vmatrix}$$

Мінімізована функція:

$$F = x_2 x_3 \bar{x}_4 + x_1 \bar{x}_2 + x_1 \bar{x}_3 \quad (2)$$

Операція супер-склеювання змінних у першій матриці проведена для блоків 8–11, які виділені червоним кольором. Просте склеювання змінних проведено для блоків 12-13, які виділені синім кольором та 6, 14 які виділені чорним кольором. Споглядаючи мінімальні функції (1) і (2) бачимо, що мінімальна функція (4) простіша на один терм.

Порівнюючи особливості процесу мінімізації логічних функцій з використанням комбінаторних структур повної бінарної системи з повторенням 2-(n, b)-design та неповної бінарної системи з повторенням 2-(n, x/b)-design, можна зробити висновок, що комбінаторна система 2-(n, b)-design і послідовне чергування алгебричних операцій супер-склеювання змінних (якщо така операція можлива) та простого склеювання змінних, у першій матриці (таблиці істинності), забезпечує оптимальну ефективність процесу та достовірність результату мінімізації булевих функцій.

2. Результати вибору чергування образних перетворень при мінімізації булевих функцій

Застосування протоколу оптимального чергування образних перетворень при мінімізації булевих функцій дає наступні результати:

- розширює можливості застосування векторних інтервалів булевого простору \mathbb{Z}^n при мінімізації логічних функцій;
- визначає частковий алгоритм розпізнавання комбінаторних систем 2-(n, b)-design та пошуку її меж;
- забезпечує автоматизований пошук комбінаторних систем 2-(n, b)-design у структурі таблиці істинності заданої логічної функції.

Ефективність застосування протоколу оптимального чергування образних перетворень при мінімізації булевих функцій демонструється прикладами мінімізації логічних функцій, запозичених з робіт інших авторів з метою порівняння.

Приклад 3. Знайти мінімальні ДНФ і КНФ логічної функції, використовуючи карту Махоні зі заданої таблиці істинності на рис. 3 [10].

<i>F</i>	1	0	1	0	1	1	1	1	1	0	1	0	0	1	0	1
<i>D</i>	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
<i>C</i>	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
<i>B</i>	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
<i>A</i>	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Рис. 3. Таблиця істинності логічної функції для прикладу 3

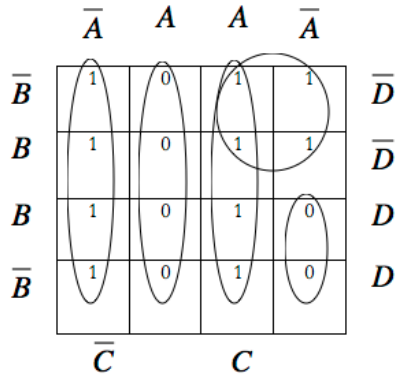


Рис. 4. 4-bit карта Махоні для прикладу 3

Рис. 3 являє собою підсумкову карту Махоні із таблиці істинності логічної функції на рис. 3. Збір одиниць позначається суцільними еліпсами, а збір нулів – пунктирними еліпсами.

Карти Махоні у багатьох відношеннях є набагато ефективнішими, порівняно з картами Карно, оскільки вони легко розширюються до необхідної кількості вхідних даних, що суттєво розширює загальний контент застосування карт Махоні [10].

Використовуючи збір одиниць на рис. 4, мінімальна ДНФ заданої логічної функції (рис. 3) буде мати вигляд:

$$OUT = C \cdot A + \bar{D} \cdot C + \bar{C} \cdot \bar{A}. \quad (3)$$

Мінімізація функції $F(x_1, x_2, x_3, x_4)$ (рис. 3) методом образних перетворень зводиться до наступної процедури:



$$F = \begin{array}{c|cccc} \text{№} & x_4 & x_3 & x_2 & x_1 \\ \hline 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 \\ 4 & 0 & 1 & 0 & 0 \\ 5 & 0 & 1 & 0 & 1 \\ 6 & 0 & 1 & 1 & 0 \\ 7 & 0 & 1 & 1 & 1 \\ 8 & 1 & 0 & 0 & 0 \\ 10 & 1 & 0 & 1 & 0 \\ 13 & 1 & 1 & 0 & 1 \\ 15 & 1 & 1 & 1 & 1 \end{array} = \begin{array}{c|ccc} & 0 & 0 & \\ \hline 0 & 1 & & \\ 1 & 1 & 1 & \end{array} = \begin{array}{c|cc} & 0 & 0 \\ \hline 0 & 1 & \\ 1 & 1 & 1 \end{array}$$

Блоки 4–7 (виділені червоним кольором) та блоки 0, 2, 8, 10 (виділені синім кольором) мінімізовані за протоколом супер-склеювання змінних. Інші блоки мінімізовані за протоколами простого склеювання та напівсклеювання змінних [6; 7].

Мінімізована ДНФ функції:

$$F = \overline{x_1 x_3} + x_3 \overline{x_4} + x_1 x_3. \quad (4)$$

Результат мінімізації (4) збігається з результатом мінімізації (3), проведеної за допомогою карти Махоні [10], однак отримання мінімальної ДНФ логічної функції методом образних перетворень є простішою процедурою.

Використовуючи збір нулів на рис. 4, мінімальна КНФ заданої логічної функції (рис. 3) буде мати вигляд:



Національний університет
водного господарства
та природокористування

$$\overline{OUT} = \overline{C \cdot A} + D \cdot C \cdot \overline{A}$$

Застосовуючи закон де Моргана, отримуємо:


$$\overline{\overline{OUT}} = \overline{(\overline{C \cdot A}) + (D \cdot C \cdot \overline{A})},$$

$$OUT = \overline{(\overline{C \cdot A})(D \cdot C \cdot \overline{A})},$$

і, нарешті, мінімальна КНФ

$$OUT = (C + \overline{A})(\overline{D} + \overline{C} + A). \quad (5)$$

Мінімізація КНФ функції $F(x_1, x_2, x_3, x_4)$ (рис. 3) методом образних перетворень зводиться до такої процедури [9]:



$$F = \begin{array}{c|cccc} \mathcal{N} & x_4 & x_3 & x_2 & x_1 \\ \hline 1 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 1 & 1 \\ 9 & 1 & 0 & 0 & 1 \\ 11 & 1 & 0 & 1 & 1 \\ 12 & 1 & 1 & 0 & 0 \\ 14 & 1 & 1 & 1 & 0 \end{array} = \begin{array}{c|cccc} & 1 & 1 & 1 & 0 \\ & 1 & 1 & 0 & 0 \\ & 0 & 1 & 1 & 0 \\ & 0 & 1 & 0 & 0 \\ & 0 & 0 & 1 & 1 \\ & 0 & 0 & 0 & 1 \end{array} = \begin{array}{cc|c} & & 1 & 0 \\ & 0 & 0 & 1 \end{array}$$

Блоки 1, 3, 9, 11 (виділені червоним кольором) мінімізовані за протоколом супер-склеювання змінних. Інші блоки мінімізовані за протоколами простого склеювання та напівсклеювання змінних [6; 7].

Мінімізована КНФ функції:

$$F = (\overline{x_1} + x_3)(x_1 + \overline{x_3} + \overline{x_4}). \quad (6)$$

Результат мінімізації (6) збігається з результатом мінімізації (5), проведеної за допомогою карти Махоні [10], однак отримання мінімальної КНФ логічної функції методом образних перетворень є простішою процедурою.

Висновки

1. З порівняння особливостей процесу мінімізації логічних функцій з використанням комбінаторних структур повної бінарної системи з повторенням 2-(n,

b)-design та неповної бінарної системи з повторенням 2-(n, x/b)-design, можна зробити висновок, що комбінаторна система 2-(n, b)-design і послідовне чергування алгебричних операцій супер-склеювання змінних (якщо така операція можлива) та простого склеювання змінних, у першій матриці (таблиці істинності), забезпечує оптимальну ефективність процесу та достовірність результату мінімізації булевих функцій.

2. Ефективність чергування протоколів мінімізації булевих функцій демонструється прикладом 3 [10], – мінімізація 4-розрядної булевої функції. З огляду на представлений приклад ефективність послідовного застосування логічних операцій супер-склеювання змінних та простого склеювання змінних для мінімізації булевих функцій дає підставу для доцільності застосування зазначеної послідовності логічних операцій для мінімізації логічних функцій, оскільки зазначена послідовність логічних операцій спроможна:

– підтримати автоматизований процес пошуку комбінаторних систем 2-(n, b)-design у структурі таблиці істинності заданої логічної функції з наступною мінімізацією логічних функцій методом образних перетворень;

– збільшити продуктивність процесу мінімізації логічних функцій.

1. Curtis H. A. A new approach to the design of switching circuits. N.J. : Princeton, Toronto. 1962. 635 p.
2. Проектирование цифровых вычислительных машин / под ред. С. А. Майорова : учеб. пособие для вузов. М. : Высшая школа, 1972. 344 с.
3. Поспелов Д. А. Логические методы анализа и синтеза схем. М. : Энергия, 1974. 368 с.
4. Закревский А. Д. Логический синтез каскадных схем. М. : Наука, 1981. 416 с.
5. Рыцар Б. Е. Метод минимизации булевых функций. *Проблемы управления и информатики*. 1997. № 2. С. 100–113.
6. Riznyk V., Solomko M. Minimization of boolean functions by combinatorial method. *Technology audit and production reserves*. 2017. Vol 4/2 (36). P. 49–64.
7. Riznyk V., Solomko M. Application of super-sticking algebraic operation of variables for boolean functions minimization by combinatorial method. *Technology audit and production reserves*. 2017. Vol 6/2 (38). P. 60–76.
8. Riznyk V., Solomko M. Research of 5-bit boolean functions minimization protocols by combinatorial method. *Technology audit and production reserves*. 2018. Vol 4/2 (42). С. 41–52.
9. Riznyk V., Solomko M. Minimize of conjutable normal forms of boolean functions combinatorial method. *Technology audit and production reserves*. 2018. Vol 5/2 (43). С. 42–55.
10. David Bonal. Karnaugh and Mahoney – Map Methods for Minimizing Boolean Expressions. October 19th, 2013.

Solomko M. T., Candidate of Engineering (Ph.D.), Associate Professor; Kyrykovych V. O., Senior Student (National University of Water and Environmental Engineering, Rivne)

OPTIMAL SEQUENCE OF FIGURATIVE TRANSFORMATIONS AT MINIMIZING BOOLEAN FUNCTIONS

The optimal alternation of image transformations for the process of minimization of

Boolean functions is considered in the work. The efficiency of the chosen sequence of reinforcement transformations is demonstrated by examples of minimization of Boolean functions borrowed from the works of other authors in order to compare

Keywords: minimization of Boolean functions, figurative transformations, logical operation of super-gluing of variables.

УДК 004.432.4

Національний університет

водного господарства

Хмельничий Т. В., студент (Національний університет водного господарства та природокористування, м. Рівне)

ПРОЕКТУВАННЯ ТА СТВОРЕННЯ WEB-РЕСУРСУ ДЛЯ МОНІТОРИНГУ ЯКОСТІ ОСВІТИ У ЗАКЛАДАХ ВИЩОЇ ОСВІТИ

У статті розглянуто процес проектування web-ресурсу моніторингу якості освіти з підтримкою динамічного створення анкет, перегляду статистики, та керування поточними опитуваннями. Обґрунтовано вибір ресурсів на базі яких створюється проект.

Ключові слова: web-ресурс, опитування, бази даних, система побудови анкети, візуалізація даних.

Моніторинг якості на сьогоднішній день є дуже важливою складовою багатьох галузей, особливо в сфері освіти де він використовується при анкетуванні студентів для оцінки якості викладання, зацікавленості в предметах, поведінки тощо. Тому в зв'язку з цим проект який задовольняє ці потреби та знаходиться під повним контролем навчального закладу є дуже актуальним. Оскільки такий ресурс має бути доступним з будь якого пристрою через їх він має бути у вигляді web-сайту.

Основна особливість цього проекту в тому, що він включає в собі всі необхідні функції при чому знаходиться під цілковитим контролем навчального закладу. Наприклад він дозволяє мати прямий доступ до бази даних з результатами проведених опитувань, надає можливість провести модифікацію системи, додати нові функції та на відміну від майже всіх інших сервісів він є безкоштовним.

На стадії проектування такого ресурсу одним з найважливіших чинників було забезпечення гнучкості для створених анкет. Це враховується при створення бази даних та API для нашого проекту.

Успішне проведення опитування студентів не закінчується після збору відповідей. Насправді саме тоді починається одна з найважливіших частин процесу - аналіз та інтерпретація результатів. Точний та ефективний аналіз та інтерпретація даних та результатів опитувань учнів є критично важливою для досягнення цілей опитування від навчального закладу.

Тому під час реалізації проекту основними завданнями, які потрібно вирішити були: динамічна система побудови опитування, можливість додання нових

типів питань в подальшому, адаптивність web-ресурсу. Також дуже важливою частиною був вибір архітектури серверу, бази даних та клієнтської частини.

The screenshot shows the 'Nuwm Survey' interface. At the top, there is a header with the text 'Nuwm Survey' and 'Анкету Результати'. Below this is a form titled 'Нова анкета' (New survey) with a date range from 2021-05-17 to 2021-05-24. The main form area is divided into two sections for adding questions. The first section is titled 'Нове питання' (New question) and features a dropdown menu set to 'Шкала' (Scale). Below the dropdown, there is a toggle switch for 'Обов'язкова відповідь' (Required answer) which is turned on. The question text is 'Національний університет водного господарства та природокористування'. The form includes fields for 'Мінімальне' (Minimum) set to 1, 'Максимальне' (Maximum) set to 5, and 'Крок' (Step) set to 1. There are three buttons at the bottom of this section: 'Вверх' (Up), 'Вниз' (Down), and 'Видалити' (Delete). The second section is also titled 'Нове питання' and has a dropdown menu set to 'Коротка текстова відпс' (Short text answer). It also has the 'Обов'язкова відповідь' toggle turned on. It has the same 'Вверх', 'Вниз', and 'Видалити' buttons. Below these two sections are two large buttons: a blue one with a plus sign and the text '+ Додати питання' (Add question), and a green one with a checkmark and the text 'Зберегти опитування' (Save survey).

Nuwm Survey

Рис. 1. Форма для побудови опитування

За допомогою такої форми даний сервіс надає можливість створювати опитування з потрібною кількістю питань. Також важливим було забезпечити можливість подальшого розширення наявних типів питань в цьому сервісі завдяки новим компонентам які будуть інтегровані в наш web-проект. Для цього структура проекту було створена таким чином щоб в майбутньому вона не була перешкодою для розробки та реалізації нових функцій.

Важливою частиною в цьому проекті є візуальна складова. Вона має забезпечувати зрозумілий інтерфейс та включає в собі всі можливі аспекти взаємодії між сервісом та користувачем. Тому особливу увагу потрібно звернути на поведінку сторінки при різних діях на сторінці таких як: наведення курсору на елементи сторінки, наведення курсору на форми та поля для введення, правильне відображення контенту при зміні розширення пристрою виводу.

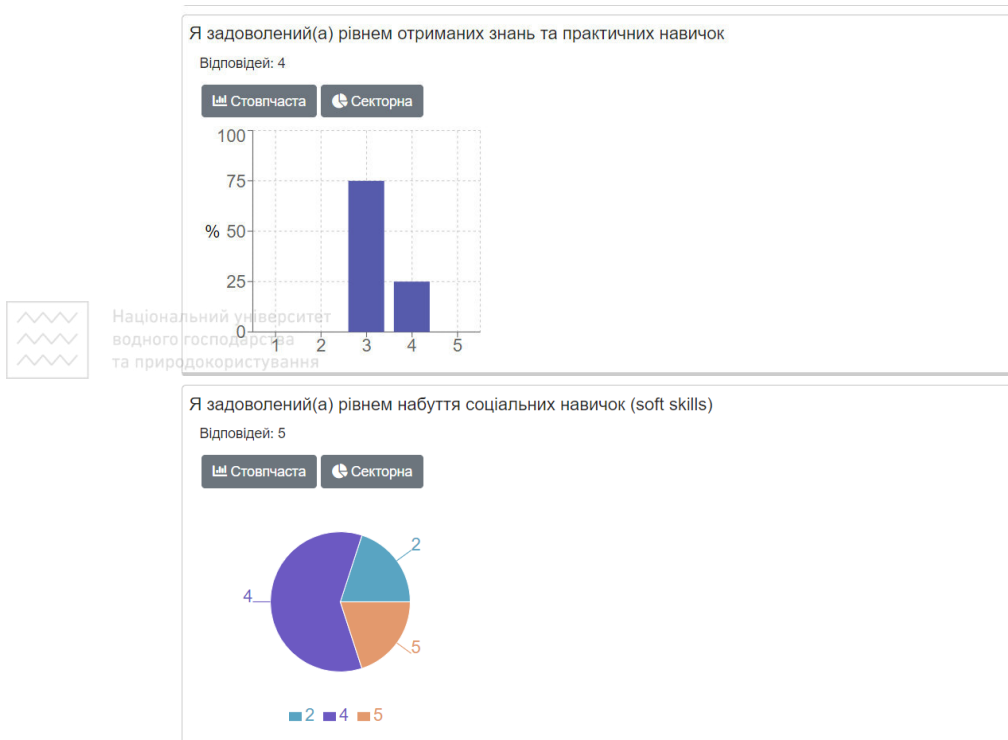


Рис. 2. Сторінка з результатами опитувань

Для розробки візуальної частини web-сервісу була вибрана JavaScript-бібліотека – ReactJS. Основними її перевагами є популярність, структура проекту у вигляді компонентів, використання шаблону Virtual-DOM – який дозволяє оновлювати контент сторінки значно швидше ніж звичайний DOM. Код ReactJS складається з багатьох компонентів, кожен з яких має власні елементи керування та логіку, які є специфічними для цього компонента. Це допомагає підтримувати структуру проекту впорядкованою під час розробки великих проектів або при впровадженні нових функцій чи логіки в проект.

Також разом з React для стилізації сторінок було вирішено використати Bootstrap та React-Bootstrap. Цей фреймворк забезпечує одні з найкращих заздалегідь визначених компонентів, які потім можна використовувати в будь-якій частині додатка. Основними перевагами використання їх в парі є:

- менша кількість помилок і багів в різних браузерах;
- система сіток для розміщення елементів на сторінках web-сайту;
- легке розміщення компонентів в коді проекту;
- зменшення кількості «boilerplate» коду в проекті;

Базою даних в проекті було вибрано об'єктно-реляційну – PostgreSQL. Вона є доступною безкоштовно для використання незалежно від масштабу проекту. Також перевагою є відкритість коду, що забезпечує незалежність від будь яких компаній. Підтримує велику кількість типів даних: стандартні (цілі числа, рядки, а т.д.), структуровані (дата/час, масив, UUID), документи (JSON, XML) та спеціальні типи які можливо налаштувати в залежності від потреб.

Іншою важливою особливістю PostgreSQL є те, що він дозволяє розширювати або змінювати базу даних. Це означає, що додати нові функції користувач може самостійно вже після створення бази даних. Разом з PostgreSQL при встановленні у комплекті також є інтерфейс користувача для роботи з базами даних pgAdmin 4 з оновленим дизайном та різними інформаційними панелями.

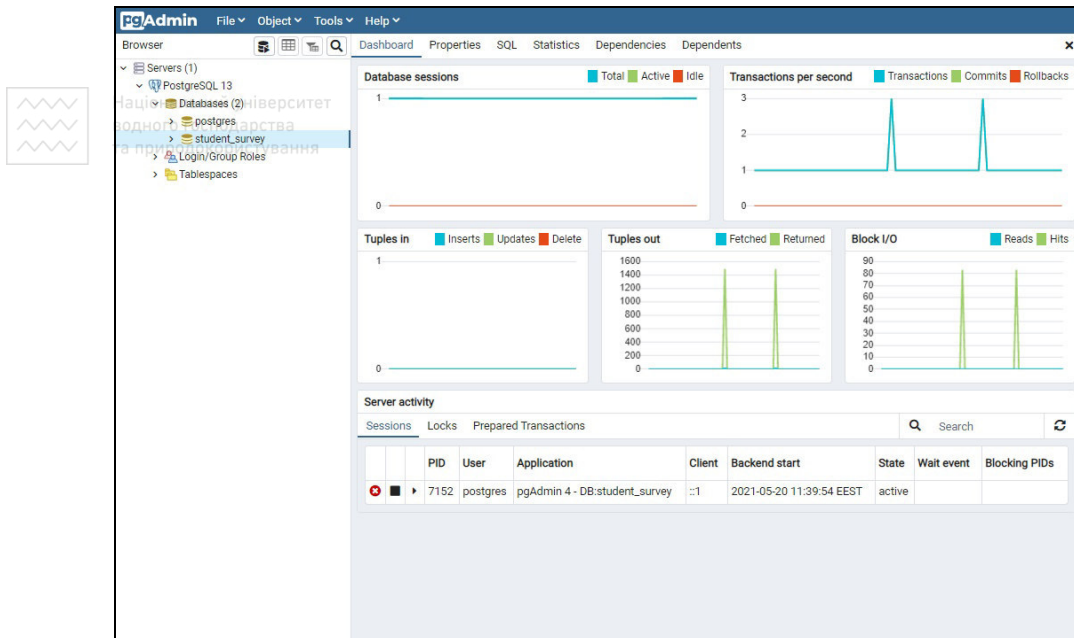


Рис. 3. Інтерфейс pgAdmin 4

Особливості pgAdmin 4 включають в себе: інструменти для SQL запитів, допоміжні повідомлення про помилки, редактор SQL із виділенням синтаксису.

Для виконання запитів до бази даних та реалізації API було вибрано фреймворк Django. Написаний на одній з найпопулярніших мов програмування Python він є безкоштовним з відкритим кодом, має активну спільноту, чудову документацію та безліч варіантів безкоштовної та платної підтримки. До всесвітньо відомих сайтів, які використовують Django, належать: Instagram, Knight Foundation, Mozilla, National Geographic, Pinterest та багато інших.

Для побудови веб API на основі Django використовувався Django REST Framework (DRF). Основними його перевагами є:

- веб API з можливістю перегляду для зручності розроблення та перевірки працездатності;
- дозволяє проводити серіалізацію джерел даних ORM та не ORM;
- має детальну та зрозумілу документацію;
- готові компоненти для серіалізації об'єктів бази даних;
- забезпечує декілька API views, що використовуються в загальнозживаних шаблонах;

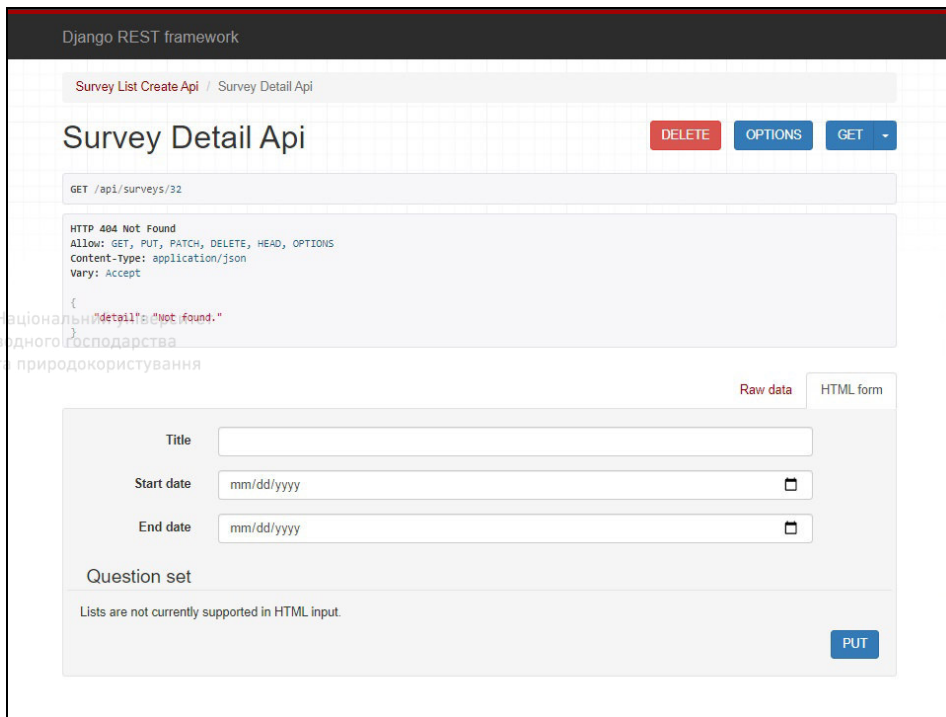


Рис. 4. Веб API DRF фреймворку

При побудові проекту було використано стандартну структуру Django REST Framework при якій є чотири основні компоненти: Models – джерело інформації, яке містить поля об'єкту та поведінку даних які зберігаються, як правило для кожної таблиці бази даних існує одна модель, API Serializers – дозволяють перетворювати складні дані, такі як запитів до бази даних та екземпляри моделей, у власні типи даних Python, які потім можна легко перетворити в JSON, XML або інші типи, API Views – це функція або клас який відповідальний за отримання запиту та повернення відповіді, у нашому випадку повертає дані у JSON форматі.

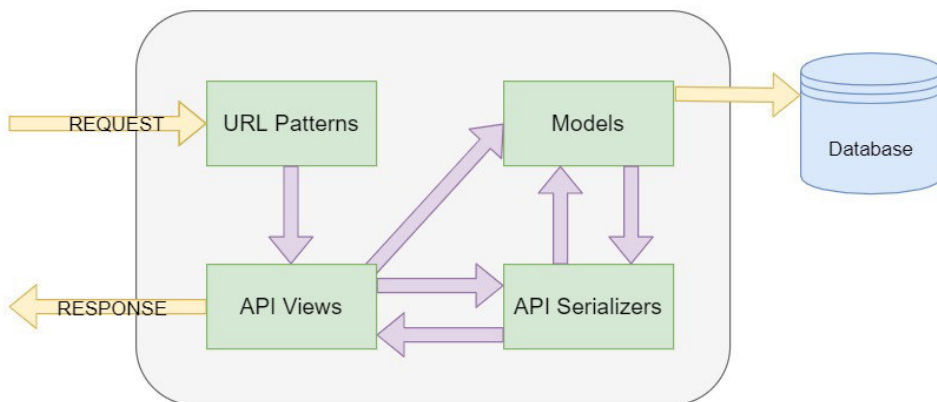


Рис. 5. Структура Django Rest Framework

Після створення перелічених частин проекту веб-ресурс набуває потрібного нам функціоналу який забезпечує:

- динамічне створення анкет для опитування;
- редагування анкет при потребі;
- перегляд статистики;
- проведення анонімних та не анонімних опитувань;

1. Cody Lindley, DOM Enlightenment: O'Reilly Media, 2015. 162 с. 2. Документація Django версія 3.2. URL: <https://docs.djangoproject.com/en/3.2/> (дата звернення: 21.05.2021). 3. Robin Wieruch, The Road to React: Your journey to master plain yet pragmatic React.js (2020 Edition): Опублікована незалежно, 2020. 229 с.

Khmelnychi T. V., Senior Student (National University of Water and Environmental Engineering, Rivne)

DESIGN AND CREATION OF WEB-RESOURCE FOR MONITORING THE QUALITY OF EDUCATION IN HIGHER EDUCATION INSTITUTIONS

The article considers the process of designing a web-resource for monitoring the quality of education with support for the dynamic creation of surveys, revision of statistics, and management of current surveys. The choice of resources on the basis of which the project is created is substantiated.

Keywords: web-resource, surveys, databases, survey construction system, data visualization.

УДК 004.9

Жакун¹ А. В., студент, Мічута² О. Р., к.т.н., доцент (Національний університет водного господарства та природокористування, м. Рівне, ¹zhakun_ak19@nuwm.edu.ua, ²o.r.michuta@nuwm.edu.ua)

РОЗРОБКА ЧАТ-БОТА З ВИКОРИСТАННЯМ TELEGRAM VOT API ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ НАВЧАННЯ

У статті проведений аналіз підходів до створення чат-ботів, розглянутий поточний стан і передумови їх затребуваності в мережах. Обґрунтована актуальність використання чат-ботів для взаємодії людини і віртуального середовища. Виконана класифікація чат-ботів за типами користувачів, видами інтерфейсів, методами доступу до бота, призначенням та принципами роботи.

Ключові слова: чат-бот, діалог, інструментальні засоби, класифікація чат-ботів.

Вступ. Оскільки сучасний світ не стоїть на місці, технології стрімко розвиваються та кожного дня з'являються нові тренди та ідеї, багато з них мають можливість вплинути на ІТ галузь, яка в свою чергу впливає на інші галузі, що в

подальшому впливатиме на наше повсякденне життя. Якщо до недавнього часу популярними були додатки або комп'ютерні програми, то на даний момент лідерство займають чат-боти, які мають великі перспективи в різних сферах нашого життя [1].

Комунікаційні функції Інтернет набувають нових форм, які пов'язані з сучасними технічними можливостями. Особливої актуальності останнім часом набули технології комунікацій, засновані на месенджерах і чат-ботах. Еволюція месенджерів привела до того, що в даний час вони можуть відправляти як текстову інформацію, так аудіо і відео повідомлення. Сучасне покоління вважає за краще мобільні пристрої замість настільних комп'ютерів. Його представники не звикли взаємодіяти в Інтернет середовищі за допомогою браузерів і спливаючих меню, а використовують смартфони для невеликої кількості завдань. Дослідження показали, що телефони використовуються для обміну повідомленнями частіше, ніж для інших цілей. Тому компанії прагнуть завоювати увагу онлайн-користувачів і створюють чат-боти, щоб вбудуватися в месенджери [2].

Актуальність. Під чат-ботами в загальному сенсі зазвичай розуміють спеціальні програми, які здійснюють інтернет-спілкування найчастіше з одним або декількома користувачем, використовуючи штучний інтелект. Сучасні види інтернет-комунікацій здатні виступати в якості віртуального співрозмовника і повторювати і відтворювати письмовий набір знаків людини, надаючи запрограмовану відповідь на задані питання. В силу того, що контакт з чат-ботами є не тільки цікавим заняттям, але і корисним для людини з точки зору отримання нової інформації і рішення споживчих завдань, варто відзначити, що інтернет-аудиторія прийняла їх появу доброзичливо.

Класифікація чат-ботів

Таблиця 1

Ознака класифікації	Вид чат-бота
Користувач	Персональний Бізнесовий
Інтерфейс	Кнопковий Текстовий
Доступ	Доданий в групу За підпискою Вбудований в діалог
Призначення	Комунікаційний Функціональний
Принцип роботи	Шаблонний Який навчається

Виходячи з поточної оцінки стану ринку, чат-боти, особливо для месенджерів дуже перспективний напрям, який в даний час переживає бурхливе зростання [3].

Відсоток користувачів, які очікують отримати миттєвий відгук по різних каналах зв'язку. Від чат-ботів швидкої відповіді очікують 75% опитаних.

Response Time by Communication Channel

How soon would you expect to get a response on each of these channels?

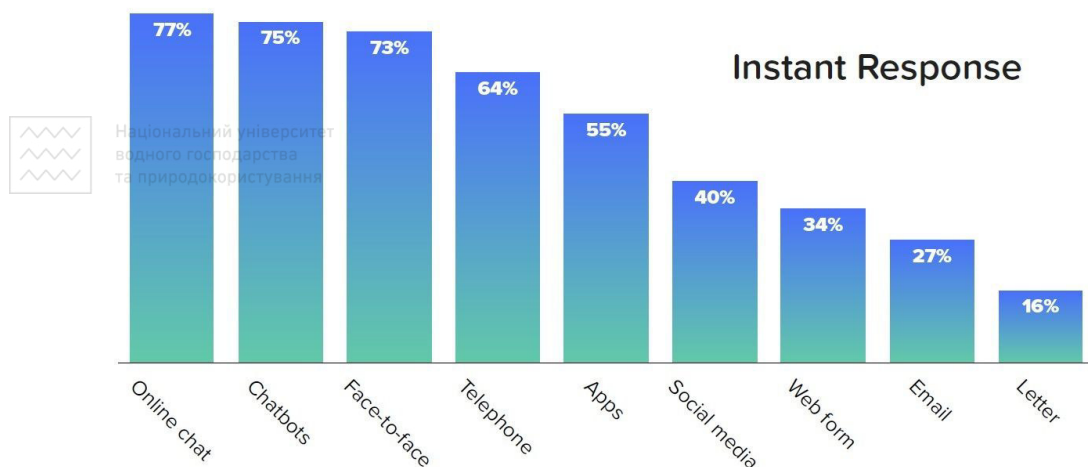


Рис. 1. Час відгуку з центром зв'язку

Розробники колл-центрів оцінюють середній час очікування відповіді на дзвінок в 11 хвилин. У той час як 75% клієнтів вішають трубку протягом 5 хвилин очікування. Навіть найпримітивніші чат-боти працюють набагато швидше. Середній час очікування відповіді від бота – 45 секунд. Крім того, користувачі цінують ботів за можливість отримати миттєві відповіді на питання в будь-який час доби [5].

Технологія розробки. Розглянемо узагальнений приклад серверної архітектури бота на рис. 2.

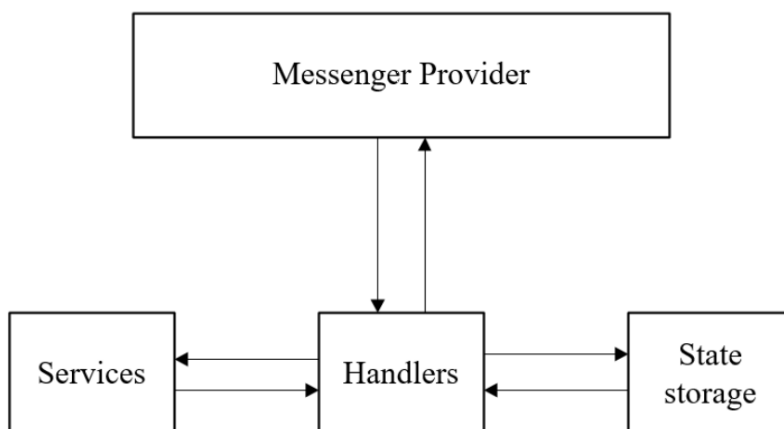


Рис. 2. Узагальнений приклад серверної архітектури бота

У даному випадку бот складається із чотирьох основних частин:

- а) Провайдер месенджера (Messenger Provider);
- б) Обробники команд (Handlers);
- в) Сервіси (Services);
- г) Контейнер стану (State storage).

Провайдер месенджера обгортає Telegram Bot API і забезпечує універсальний інтерфейс для прийому, обробки та відправки повідомлень. Провайдер месенджера визначає, що хоче користувач, і спрямовує його до правильного обробника. Мета цієї частини забезпечити шар абстракції над API месенджерів [4].

Для створення бота у системі Телеграм існує бот – BotFather. Необхідно розпочати з ним діалог та виконати декілька простих кроків. Основною ціллю цієї процедури є отримання токена авторизації бота, який дозволяє серверам Телеграм унікально ідентифікувати бота.

Власне Bot API – це те, що дозволяє легко та зручно створювати чат-ботів. Цей API дозволяє підключати ботів до системи Телеграм. З точки зору системи Телеграм, боти – це спеціальні облікові записи, для яких не потрібно встановлювати додатковий номер телефону. Ці облікові записи служать інтерфейсом для коду, що працюватиме на сервері. Для зв'язку з цим сервером використовується простий HTTPS-інтерфейс, який пропонує спрощену версію API Телеграм [4].

Модуль TeleBot – є оболонкою над запитами до TelegramBotAPI, використовується для спрощення і мінімізації написаного коду. Всі типи повністю відповідають визначенням типів API Telegram, за винятком from поля Message, яке перейменовано в from_user (оскільки from це зарезервований токен Python). До таких атрибутів, як message_id можна звертатися безпосередньо, наприклад: message.message_id. При написанні програм не дуже вимогливих до швидкості виконання варто звернути увагу, що атрибут message.chat може належати як певному користувачеві, так і до групового чату, це з повністю окупається перевагами мови.

В класі TeleBot розташовані всі методи API. Щоб слідувати загальним угодам про імена Python, вони перейменовані. Наприклад: sendMessage send_message, editMessageText, edit_message_text. Функція, прикрашена декоратором примірника TeleBot – обробник повідомлень. Обробники повідомлень складаються з одного або декількох фільтрів. Кожен фільтр повертає True або False для певного повідомлення і обробник отримує дозвіл на обробку повідомлення, якщо повертається True [4].

Розглянемо основні типи які підтримує Bot API, та які можуть бути передані як у запиті бота, так і при відповіді на цей запит:

- User – цей об'єкт являє собою користувача або бота;
- Chat – цей об'єкт являє собою чат чи групу;
- Message – цей об'єкт являє собою повідомлення. Використовується як для надсилання інформації користувачу, так і для отримання від нього відповідей;
- Audio – аудіо файл, який можна програвати за допомогою месенджера;
- Document – файл будь-якого не описаного типу;
- Video – описує відео файл;

- Voice – голосове повідомлення надіслане користувачем;
- VideoNote – відео повідомлення надіслане користувачем;
- Location – відповідає певній точці на карті;

Створення чат-ботів

1. Визначення вимог:

- ознайомитися з предметною областю для чат-бота і визначити завдання, які він буде вирішувати;
- сформулювати вимоги до розробки чат-бота: вибрати платформи, через які відбуватиметься спілкування, вибрати сервіс для розробки та сервіс для запуску бота [1].

2. Розроблення чат-бота:

- створити комунікаційну архітектуру, яка дозволить вибудувати ієрархію контенту для спілкування;
 - зібрати інформацію, необхідну для повноцінного діалогу, щоб навчити бота розуміти різні варіанти одного і того ж поняття і давати коректні відповіді. Це можна зробити, провівши опитування серед групи людей цільової аудиторії; задати схему ведення діалогу на основі власного сценарію або на основі умов, при яких бот буде відповідати певним чином [4];
 - структурувати контент і змодельювати схему діалогу, детально опрацюючи відповіді ботів, щоб спілкування виглядало природним;
 - навчити бота, розробивши чіткі структуровані сценарії відповіді на основі великої бази даних для навчання або використовуючи готові навчальні алгоритми для інтерпретації відповідей в разі відсутності такої бази.

3. Підключення:

- інтегрувати платформу з іншими сервісами, якщо платформа вже інтегрована з іншими системами, то подібну схему інтеграції можна застосувати і для чат-бота;
- виконати тестування роботи чат-бота; активувати бота.

4. Аналіз роботи та доопрацювання:

- збирати та аналізувати інформацію про роботу бота;
- допрацьовувати і оптимізувати поведінку чат-бота: включити нові питання, поліпшити логіку і класифікацію понять, додати більше відповідей тощо.

Таким чином, розроблення чат-ботів має певну специфіку і вимагає багато людських зусиль і широкого кола професійних навичок [3].

Висновки. У роботі проаналізовані особливості існуючих підходів до створення чат-ботів, а саме: обґрунтовано, що чат-боти є одним з найбільш перспективних напрямків в найближчому майбутньому і є актуальною темою для багатьох наукових досліджень. Це обумовлено двома факторами: активне використання месенджерів і розвиток технологій штучного інтелекту. Розроблена класифікація чат-ботів (за типами користувачів, видами інтерфейсів, способам доступу до боту, призначенням, принципом роботи). Провели аналіз Telegram Bot API. Визначили зміст основних етапів створення чат-ботів.

1. Розвиток чат-ботів. URL: <https://stecpoint.ru/Practices-ChatBotsEvolution/> (дата

звернення: 06.05.2021). **2.** A curated directory of chat bot resources & tools. URL: <http://www.botsfloor.com/botstash/products/?category=Development%20Platforms>. (дата звернення: 06.05.2021). **3.** Telegram Bot API. URL: <https://core.telegram.org/bots/api>. (дата звернення: 06.05.2021). **4.** Telegram Bot. URL: <https://core.telegram.org/bots>. (дата звернення: 06.05.2021). **5.** Глобальна статистика актуальності месенджерів. URL: <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/> (дата звернення: 06.05.2021).

Національний університет

Zhakun A. V, Senior Student, Michuta O. R., Candidate of Engineering (Ph.D.), Associate Professor (National University of Water and Environmental Engineering, Rivne)

DEVELOPMENT OF A CHAT BOT USING TELEGRAM BOT API TO ASSESS THE QUALITY OF LEARNING.

The analysis of the approaches to creating chatbots has been completed; the current state and the preconditions for their demand in networks are considered. The relevance of the use of chatbots for human interaction and the virtual environment was justified. Classification of chatbots according to user types, types of interfaces, methods of access to the bot, purpose and principles of operation has been done.

Keywords: chatbot, dialogue, tools, classification of chatbots.

УДК 519.62; 519.85; 51-78

Остапчук¹ О. П., к.т.н, доцент; Стрілець² П. М., студент; Бойко³ В. В., студент (Національний університет водного господарства та природокористування, м. Рівне, ¹o.p.ostapchuk@nuwm.edu.ua, ²strilets_ak18@nuwm.edu.ua, ³boiko_ak18@nuwm.edu.ua)

ВИКОРИСТАННЯ ТРІАНГУЛЯЦІЇ ДЕЛОНЕ ПРИ РОЗРОБЦІ ІГРОВИХ ПРОГРАМНИХ ПРОДУКТІВ

Стаття присвячується використанню тріангуляції Делоне у написанні ігрових продуктів. Зроблено огляд алгоритмів побудови тріангуляції Делоне. Розглянуто можливість використання тріангуляції Делоне для генерації ігрових карт, розробки штучного інтелекту. Зроблено висновок про широку можливість використання тріангуляції Делоне.

Ключові слова: тріангуляція, тріангуляція Делоне, діаграма Вороного, граф, тесселяція, генерація.

Однією з базових задач в обчислювальній геометрії є побудова тріангуляції Делоне, тому до неї зводиться багато інших завдань. Вона широко використовується в геоінформаційних системах для моделювання поверхонь, в машинній графіці, у вирішенні просторових задач.

Триангуляцією називається планарний граф, у якого всі внутрішні області є трикутниками (рис. 1). Опуклою триангуляцією називається така триангуляція, для якої мінімальний багатокутник, що охоплює всі трикутники, буде опуклим. Триангуляція, яка не є опуклою, називається неопуклою. Кажуть, що триангуляція задовольняє умові Делоне, якщо всередину кола, описаного навколо будь-якого побудованого трикутника, не потрапляє жодна з заданих точок триангуляції. Триангуляція називається триангуляцією Делоне, якщо вона є опуклою і задовольняє умові Делоне [1].

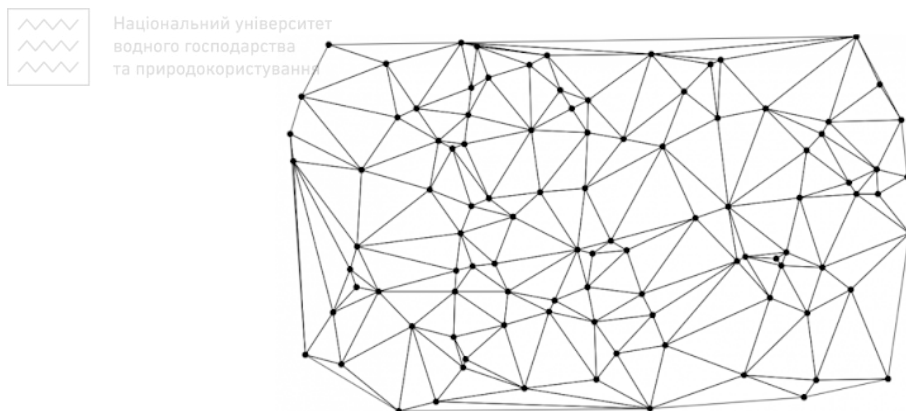


Рис. 1. Приклад триангуляції

Розглянемо жадібний алгоритм побудови триангуляції.

Крок 1. Генерується список всіх можливих відрізків, що з'єднують пари вихідних точок, і він сортується за довжинами відрізків.

Крок 2. Починаючи з самого короткого, послідовно виконується вставка відрізків в триангуляцію. Якщо відрізок не перетинається з іншими раніше вставленими відрізками, то він вставляється, інакше він відкидається.

Трудомісткість роботи жадібного алгоритму при деяких його поліпшеннях становить $O(N^2 \log N)$. У зв'язку з настільки великою трудомісткістю на практиці він майже не застосовується [2].

У наведеній нижче таблиці зібрані основні характеристики всіх алгоритмів. В колонці А представлена оцінка трудомісткості в гіршому випадку, в колонці Б – трудомісткість в середньому випадку, в колонці В – час роботи на 10 000 точок у відносних одиницях і в колонці Г – експертна оцінка простоти реалізації по 5-бальній системі (чим більше зірочок, тим алгоритм краще). Всі наведені оцінки часу отримані реалізацією алгоритмів в одному стилі (на структурі «Вузли та трикутники») і на одній програмно-апаратній платформі. Деякі з алгоритмів не реалізовані, тому в таблиці там стоять прочерки. Зауважимо, що оцінки часу досить умовні і вони, мабуть, будуть суттєво відрізнятися в різних програмних реалізаціях і на різних наборах вихідних точок [1].

Таблиця

Загальні характеристики алгоритмів триангуляції (А, Б – трудомісткість у найгіршому та в середньому випадках, В – час роботи, Г – простота)

Назва алгоритму	А	Б	В	Г
Ітеративні алгоритми				
Простий ітеративний алгоритм	$O(N^2)$	$O(N^{3/2})$	5.8	*****
Ітеративний алгоритм “Видаляй і будуй”	$O(N^2)$	$O(N^{3/2})$	8.42	**
Алгоритм з індексуванням пошуку R-деревом	$O(N^2)$	$O(N \log N)$	9.23	***
Алгоритм з індексуванням пошуку k-D-деревом	$O(N^2)$	$O(N \log N)$	7.61	***
Алгоритм з індексуванням пошуку квадродеревом	$O(N^2)$	$O(N \log N)$	7.14	***
Алгоритм статичного кешування	$O(N^2)$	$O(N^{9/8})$	1.68	*****
Алгоритм динамічного кешування	$O(N^2)$	$O(N)$	1.49	*****
Алгоритм із смуговим розбиттям точок	$O(N^2)$	$O(N)$	3.6	*****
Алгоритм із квадратним розбиттям точок	$O(N^2)$	$O(N)$	2.61	*****
Алгоритм пошарового згущення	$O(N^2)$	$O(N)$	1.93	****
Алгоритм з сортуванням точок вздовж фрактальної кривої	$O(N^2)$	$O(N)$	5.01	****
Алгоритм з сортуванням точок по Z-коду	$O(N^2)$	$O(N)$	5.31	*****
Алгоритми злиття				
Алгоритм “Розділяй і володій”	$O(N \log N)$	$O(N \log N)$	3.14	***
Рекурсивний алгоритм з розрізанням по діаметру	$O(N \log N)$	$O(N \log N)$	4.57	**
Алгоритм випуклого смугового злиття	$O(N^2)$	$O(N)$	2.79	***
Алгоритм невивуклого смугового злиття	$O(N^2)$	$O(N)$	2.54	***
Алгоритми прямої побудови				
Покроковий алгоритм	$O(N^2)$	$O(N^2)$	-	**
Покроковий алгоритм з k-D- деревом пошуку	$O(N^2)$	$O(N \log N)$	-	**
Покроковий клітинний алгоритм	$O(N^2)$	$O(N)$	-	**

Двопрохідні алгоритми				
Двопрохідний алгоритм "Розділай і володій"	$O(N \log N)$	$O(N \log N)$	2.79	****
Двопрохідний алгоритм з розрізанням по діаметру	$O(N \log N)$	$O(N \log N)$	4.13	***
Двопрохідний алгоритм випуклого смугового злиття	$O(N^2)$	$O(N)$	2.56	****
Двопрохідний алгоритм невивпуклого смугового злиття	$O(N^2)$	$O(N)$	2.24	****
Модифікований ієрархічний алгоритм	$O(N^2)$	$O(N^2)$	15.42	*****
Алгоритм лінійного замітання	$O(N^2)$	$O(N)$	4.36	*****
Віяловий алгоритм	$O(N^2)$	$O(N)$	4.18	*****
Алгоритм рекурсивного розщеплення	$O(N \log N)$	$O(N \log N)$	-	*
Стрічковий алгоритм	$O(N^2)$	$O(N)$	2.6	*****

Оскільки триангуляцією є планарний граф, це дає можливість широко використовувати її в програмуванні, наприклад при розробці ігор.

Уявімо наступну проблему: потрібно знайти найближчу аптечку (можна вважати будь-який ігровий об'єкт) на карті. Нижче наведено приклад розв'язання цієї проблеми.

Просторове відношення – це будь-яка інформація, яка описує, як пов'язаний один об'єкт в просторі з іншим. Приклади: відстань між об'єктами, площа, що покривається кожним з об'єктів простору і перетин цих площ, кількість таких об'єктів, розташованих в одній області. Такі відношення постійно використовуються у відеоіграх і можуть надавати дуже корисну інформацію для штучного інтелекту, а також самому гравцеві.

Діаграма Вороного (рис. 2) описує просторове відношення між близько розташованими точками або між їхніми найближчими сусідами. Це безліч з'єднаних багатокутників, отриманих з точок або локацій. Кожна лінія «області» Вороного знаходиться посередині між двома точками. Система, зворотна діаграмі Вороного, називається триангуляцією Делоне. Ця діаграма складається з ліній від кожної точки до її найближчих сусідів, і кожна лінія є перпендикулярною до ребер діаграми Вороного та перетинає нею саме ребро діаграми Вороного (рис. 3).

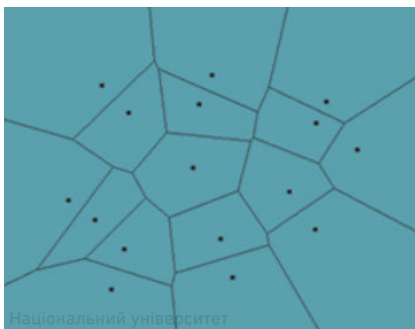


Рис. 2. Діаграма Вороного

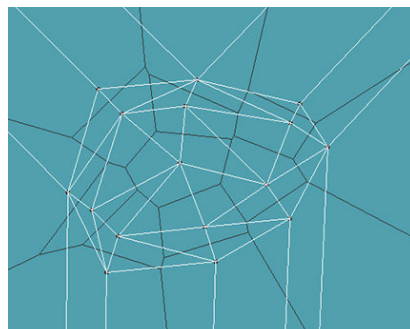


Рис. 3. Триангуляція Делоне, нанесена на діаграму Вороного

Завдяки триангуляції Делоне бачимо, що замість багатокутників тепер є безліч трикутників. Це неймовірно корисно, тому що ми розділили область на трикутники, які можна рендерити. Цю методику можна застосовувати для тесселяції чи триангуляції фігур. Крім того, це відмінний спосіб відображення точок графа на випадок, якщо необхідно буде переміститися від однієї точки до іншої. Наприклад, точки можуть позначати міста.

Розглянемо наступну Діаграму Вороного (рис. 4):

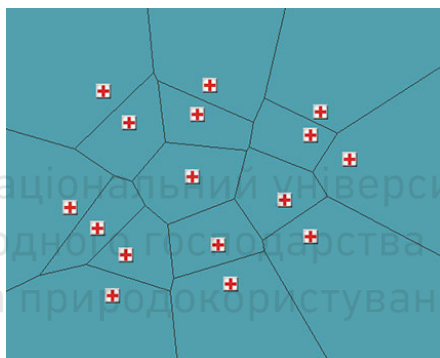


Рис. 4. Діаграма Вороного з нанесеними аптечками

Якщо кожна точка позначає аптечку, то можна досить швидко визначити, де знаходиться найближча до нас, але спочатку потрібно визначити область, в якій ми знаходимося. Діаграми Вороного не забезпечують ефективного способу визначення області, проте для прискорення пошуку ми можемо зберігати посилання на кожену область в дереві квадрантів або в R-дереві. А дізнавшись область, зможемо дізнатися її сусідів, та сусідів її сусідів.

Наприклад, якщо аптечки у вашій області більше немає, то нам потрібно знайти шлях до іншої найближчої. Зрозуміло, що знаючи область, можна дізнатися її ребра. А за допомогою цих ребер можна отримати сусідів. Будемо брати найближчого сусіда і дивитися, чи є в ньому аптечка.

Тут також можна застосувати триангуляцію Делоне. Вона складається з ліній між аптечками. Тоді її можна обійти за допомогою будь-якого алгоритму пошуку шляху, щоб знайти найближчу аптечку.

Уявімо ще одну проблему: потрібно знайти безпечний маршрут. Нижче розглянуто приклад розв'язання цієї проблеми.

Замінімо всі аптечки ворожими сторожовими вишками. Потрібно знайти найбезпечніший маршрут між ними, щоб не бути спійманим. Так як діаграма Вороного - це граф, реалізувати пошук дуже легко, потрібно лише використати алгоритм обходу графа, що підтримує спільні структури графів.

Підготувавши граф, потрібно призначити кожному ребру ціну. Для нас значенням ціни буде відстань до цих сторожових вишок, і отримати його можна безпосередньо зі структури даних. Звичайно чим менше значення на ребрі, тим краще, але в нашому випадку краще буде більше значення, тому що воно означає відстань до вежі.

Вигляд початкового графа при переміщенні з точки А в точку В зображений на рис. 5.

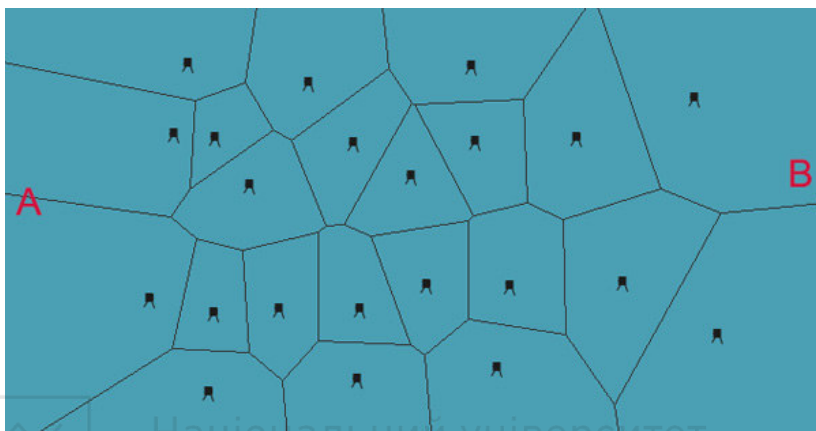


Рис. 5. Діаграма Вороного з нанесеними ворожими вишками

Задавши для кожного ребра ціну, легко побачити, який маршрут краще буде вибрати (рис. 6):

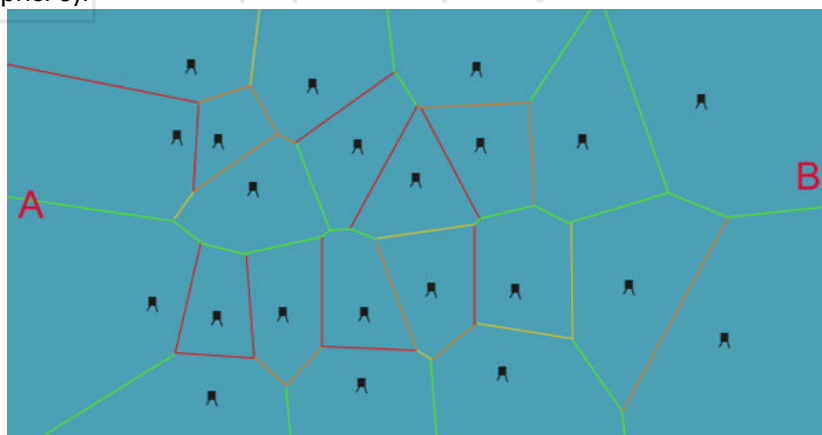


Рис. 6. Діаграма Вороного з нанесеними цінами на ребра

Червоні ребра позначають найближчі контакти з вежами. Помаранчевим позначені більш далекі; жовтим ще більш далекі, і, нарешті, зелені - найбезпечніші.

Після пошуку найкоротшого шляху, беручи до уваги ціни, нанесені на ребра, отримаємо наступний шлях (рис. 7). При такому використанні цін, буде обраний не найшвидший, а найбезпечніший шлях, що нам і потрібно. Штучному інтелекту слід дотримуватися цього шляху і не відхилятися від нього.

Можна зробити ще один крок, щоб гарантувати безпечний шлях: усунути всі ребра, що знаходяться ближче, ніж мінімальне безпечну відстань. Наприклад, якщо кожна сторожова вежа має радіус видимості 30 одиниць, то всі ребра, відстань до точок яких менше, можна буде видалити з графа і не обходити їх.

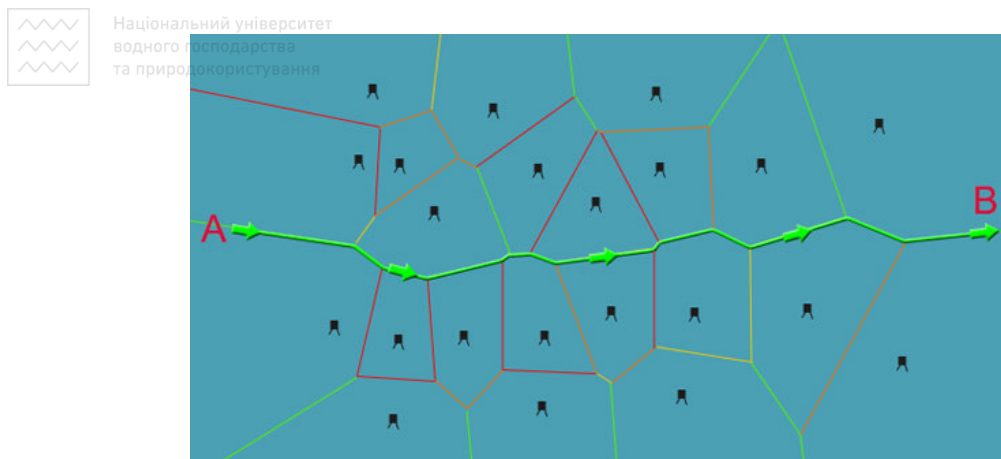


Рис. 7. Діаграма Вороного з нанесеними безпечним шляхом

Можна зробити і зворотну операцію – використовувати триангуляцію Делоне діаграми, і отримати лінії, що йдуть від кожної сторожової вежі. Штучний інтелект охоронців зможе швидко визначати, які вежі знаходяться поблизу і при необхідності попрямувати до них на допомогу [3].

Досить часто триангуляцію Делоне використовують для процедурної генерації карт (рис. 8). Розглянемо алгоритм генерації підземель на прикладі відеогри TinyKeeper:

1. Спочатку встановлюємо кількість клітинок, які хочемо сформувати, скажімо, 150. Це насправді довільна кількість, але чим більше число, тим більше підземелля і загалом більше складності.

2. Для кожної «комірки» породжуємо прямокутник випадкової ширини та довжини в межах певного радіуса. Знову ж таки радіус не має великого значення, але, ймовірно, він повинен бути пропорційний кількості клітинок.

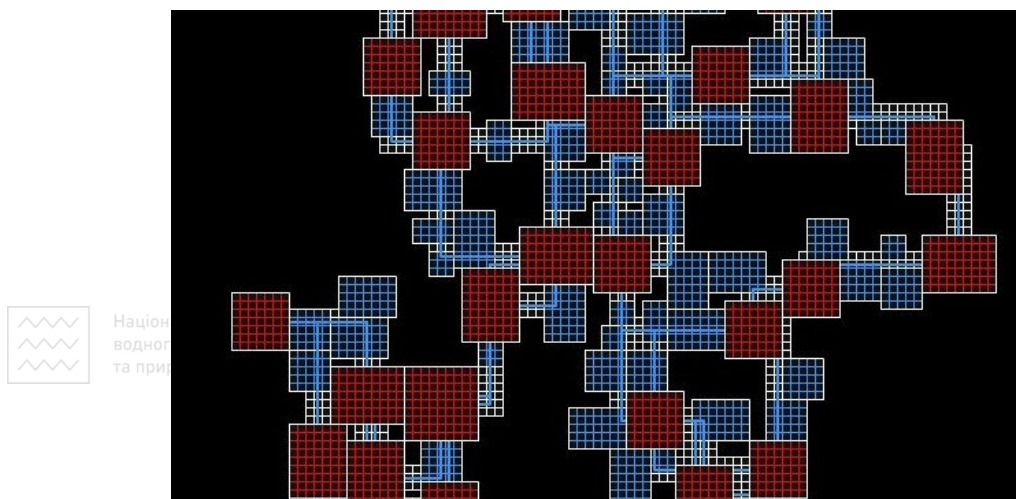


Рис. 8. Приклад згенерованої карти з використанням триангуляції Делоне

Замість використання рівномірно розподілених випадкових чисел (генератор випадкових чисел `Math.random` у більшості мов), використовуємо нормальний розподіл Парка-Міллера. Це призводить до перекосу розміру клітинок, так що вони, швидше за все, мають невеликий розмір (більше менших клітин, менше більших клітин).

На додаток до цього переконуємося, що співвідношення між шириною та довжиною кожної клітини не надто велике, ми не хочемо ідеально квадратних кімнат, але не хочемо і справді худих, але десь посередині.

3. На даний момент ми маємо 150 випадкових комірок на невеликій ділянці, більшість з них перекриваються. Далі ми використовуємо просту поведінку рульового управління, щоб відокремити всі прямокутники так, щоб жоден не перекривався. Ця методика гарантує, що клітини не перекриваються, але в цілому залишаються якомога щільнішими.

4. Будь-які прогалини заповнюємо клітинками розміром 1×1 . В результаті ми отримуємо квадратну сітку клітинок різного розміру, всі ідеально упаковані разом.

5. Далі визначаємо, які з комірок у сітці є кімнатами – будь-яка комірка шириною та висотою вище певного порогу перетворюється на кімнату. Через описаний раніше нормальний розподіл Парка-Міллера, кімнат буде лише невелика кількість у порівнянні з кількістю клітинок, між якими буде багато місця. Решта комірок все ще корисні.

6. На наступному етапі потрібно зв'язати кожну кімнату разом. Для початку побудуємо графік усіх центральних точок кімнат, використовуючи триангуляцію Делоне. Тепер усі кімнати з'єднані між собою без перетинання ліній.

7. Оскільки не має потреби, щоб кожна окрема кімната була пов'язана між собою коридором (що може призвести до дуже заплутаного плану), то будуємо мінімальне дерево розмаху, використовуючи попередній граф. Це створює граф, який гарантує, що всі кімнати з'єднані (і, отже, доступні в грі).

8. Мінімальне обшивне дерево виглядає приємно, але знову ж таки це нудне розташування підземелля, оскільки воно не містить петель. Це інша крайність

триангуляції Делоне. Отже, тепер ми повторно включаємо невелику кількість ребер з триангульованого графа (скажімо, 15% решти ребер після створення мінімального дерева, що охоплює). Таким чином, остаточний макет буде графіком усіх кімнат, кожна з яких гарантовано буде доступною і міститиме кілька петель для різноманітності.

9. Щоб перетворити графік у коридори, для кожного ребра будемо ряд прямих ліній (або L фігур), що йдуть від кожної кімнати графіка до його сусіда. Тут корисними стають клітини, якими ще не користувались (ті комірки в сітці, які не є приміщеннями). Будь-які комірки, які перетинаються з фігурами L, стають плитками коридору. Через різноманітність розмірів комірок стіни коридорів будуть звивистими і нерівними, ідеально підходять для підземелля [4].

Зважаючи на широке використання триангуляції Делоне, було розроблено відеогру WARHEX, використовуючи рушій Unity та мову програмування C#.

Суть гри полягає в захопленні бази супротивника. Перемагає той, хто першим це зробить. При цьому комп'ютер ходить по маршруту близькому до найкоротшого, гравцеві потрібно перехитрити його та перемогти. На початку гри гравець має пройти навчальний рівень (туторіал) (рис. 9), де він має почерпнути інформацію для проходження наступних рівнів (рис. 10).

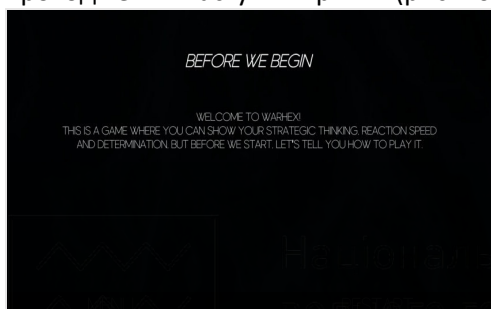


Рис. 9. Вигляд екрану з навчальним рівнем до гри

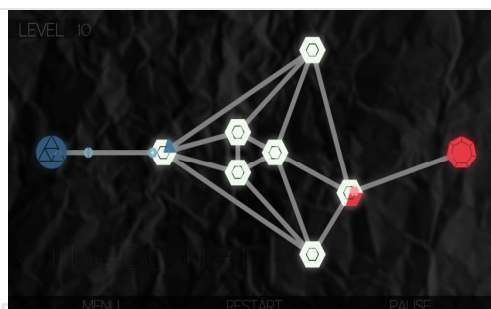


Рис. 10. Вигляд екрану з грою

Беручи до уваги наведену вище інформацію, стає зрозуміло наскільки розповсюдженим є використання триангуляції Делоне при розробці відео-ігор, завдяки можливості широкого спектру її застосування.

1. Скворцов А. В. Триангуляция Делоне и её применение. Томск : Издательство Томского университета, 2002. 128 с. 2. Математика и математики в Ивановском государственном университете. Триангуляция Делоне. URL: <http://math.ivanovo.ac.ru/dalgebra/Khashin/cutil/delaunay.html>. (дата звернення: 04.05.2021). 3. Хабр. Как использовать диаграммы Вороного для управления ИИ. URL: <https://habr.com/ru/post/460813/> (дата звернення: 03.05.2021). 4. Reddit. Procedural Dungeon Generation Algorithm Explained. URL: https://www.reddit.com/r/gamedev/comments/1dlwc4/procedural_dungeon_generation_algorithm_explained/ (дата звернення: 06.05.2021).

Ostapchuk O. P., Candidate of Engineering (Ph.D.), Associate Professor; Strilets P. N., Senior Student; Boiko V. V., Senior Student (National University of Water and Environmental Engineering, Rivne)

THE USE OF DELAUNAY TRIANGULATION IN THE DEVELOPMENT OF GAMES

The article is devoted to the use of Delaunay triangulation in writing of game products. An overview of algorithms for constructing Delaunay triangulation was made. The possibility of using Delaunay triangulation to generate playing cards and develop artificial intelligence WAS considered. The conclusion about the wide possibility of using Delaunay triangulation was made.

Keywords: triangulation, Delaunay triangulation, Voronoi diagram, graph, tessellation, generation.

УДК 004.7:004.032

Берник В. О., студент; Матус О. В., студентка (Національний університет водного господарства та природокористування, м. Рівне)

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНІТОРИНГУ ОСНОВНИХ ХАРАКТЕРИСТИК ЛОКАЛЬНОЇ КОМП'ЮТЕРНОЇ МЕРЕЖІ

Робота присвячена дослідженню пристроїв, що утворюють локальну комп'ютерну мережу, їхнім основним характеристикам та способам визначення характеристик мережі, що впливають на її роботу.

Ключові слова: Інтернет, комп'ютерна мережа, інтернет-протоколи, трафік, моніторинг, аналіз.

Вступ. Інтернет – всесвітня система сполучених між собою комп'ютерних мереж та мережевих пристроїв для передавання інформації. Загальна кількість користувачів мережі станом на 2019 рік – 4.1 млрд людей, що становило приблизно 53,6% від усього населення планети. Станом на кінець 2020 – початок 2021 кількість користувачів мережі зросла до 4.8 млрд (63% від населення планети) і продовжує зростати. В Україні станом на початок 2021 року налічувалось майже 30 мільйонів користувачів (приблизно 67% усього населення країни).

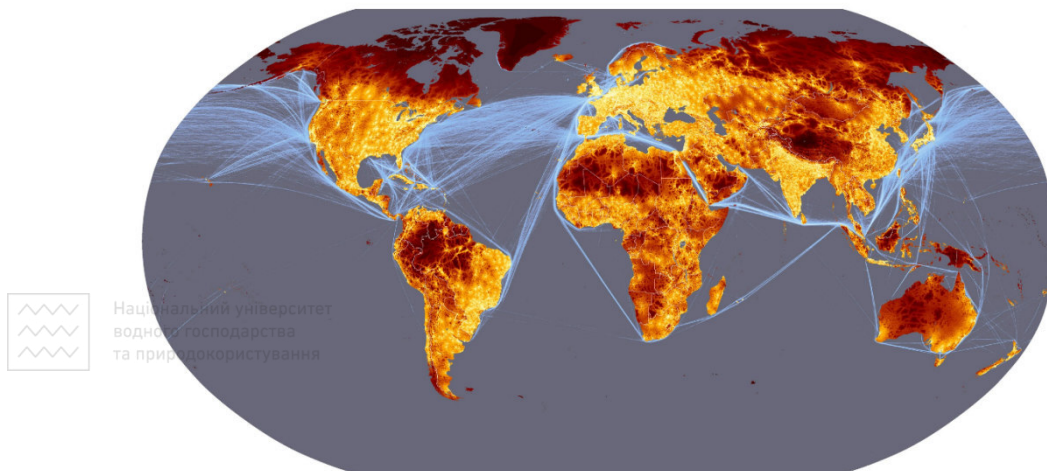


Рис. 1. Візуалізація користувачів мережі та трансконтинентальних підводних кабелів зв'язку

Мережа Інтернет стала невід'ємною частиною сучасного суспільства, широко використовується в банківській справі, безпеці, навчанні, наукових дослідженнях, зберіганні будь-яких даних, як повномасштабна заміна класичних інструментів отримання інформації, зв'язку і комунікації (газет, радіо, телебачення), тощо.

Останім часом, широко розвивається така концепція як «Інтернет речей» (IoT – Internet of Things) – система, що складається із взаємопов'язаних пристроїв, датчиків, а також програмного забезпечення, що дозволяє здійснювати передачу і обмін даними між фізичним світом і комп'ютерними системами в автоматичному режимі. Тобто, це означає, що практично будь-який побутовий пристрій можна зробити «розумним» і взаємодіяти з ним через мережу. Прикладом впровадження такої концепції є «розумні будинки», «розумні побутові пристрої», тощо.

З червня 2011 року було прийнято резолюцію ООН, яка визнає доступ в Інтернет базовим правом людини. Таким чином, відключення тих чи інших регіонів від Інтернету є порушенням прав людини [1].

Не зважаючи на стрімкий розвиток інтернет-технології, більшість протоколів, що використовуються для забезпечення роботи мережі не змінювали загальну концепцію близько 40-ка років і будуть актуальними ще довгий час. Наприклад, найпоширеніший протокол адресації IPv4 (Internet Protocol version 4) був стандартизований в 1981-му році, протокол керування передаванням даних TCP – у 1972-му році, UDP – у 1980-му році.

Більшість сучасних комп'ютерних систем об'єднані в локальну мережу з подальшим виходом до мережі Інтернет.

Огляд засобів моніторингу мережі. Моніторинг – процес збору даних, що описують поточний стан роботи комп'ютерної мережі (статистика по кількості кадрів та пакетів, що передаються мережею, тип трафіку, пропускна здатність окремих пристроїв та ділянок мережі, пристрої, які об'єднані даною мережею та їх доступність, тощо).

Існуючі програмні засоби моніторингу комп'ютерної мережі бувають наступних типів:

Системи управління мережею (Network Management Systems) – централізовані програмні системи, які збирають дані про стан вузлів і комунікаційних пристроїв в мережі, а також дані про трафік, що проходить мережею. Дані системи не лише здійснюють моніторинг, а й в автоматичному чи напівавтоматичному режимі виконують дії для керування мережею (наприклад ввімкнення чи вимкнення портів пристроїв, зміна мережевих адрес, регулювання, розподілення чи перенаправлення трафіку, тощо).

Засоби керування системою (System Management) – їх функції здебільшого повторюють функції систем управління мережею, проте, як правило засоби керування системою використовують для більших за масштабом мереж. Також, у порівнянні із системами управління мережею, додаються можливості управління безпекою (ідентифікація користувачів застосування політик безпеки) та фільтрація трафіку (в тому числі перевірка інформації, що циркулює в мережі на наявність шкідливого програмного забезпечення).

Вбудовані системи діагностики та управління (Embedded systems) – дані системи виконуються у вигляді програмно-апаратних модулів, що встановлюються в комунікаційне обладнання, а також у вигляді програмних модулів, вбудованих в операційні системи. Вони виконують функції керування та діагностики лише для одного пристрою і часто слугують SNMP-агентами.

Аналізатори протоколів (Protocol analyzers) – програмні або апаратно-програмні засоби, функціонал яких обмежений лише функціями моніторингу та аналізу трафіку в мережі. Як правило, дозволяють також виконувати декодування окремих захоплених пакетів в мережі, що дозволяє проаналізувати протоколи, які використовуються в мережі та їх ефективність, та, застосовуючи отриману інформацію, зменшити кількість надлишкового трафіку.

Експертні системи – акумулюють знання технічних спеціалістів про причини збоїв та неправильної роботи мережі. В найпростішому випадку представлені так звані «Майстром усунення неполадок» та дозволяють в багатьох випадках самостійно вирішити проблеми, що виникли під час роботи мережі, чи провести її первинне налаштування [2].

При моніторингу та аналізу характеристик локальної комп'ютерної мережі слід враховувати особливості мережевої адресації в таких мережах. Пристроєм, що об'єднані в LAN, як правило, назначають IP-адресу з діапазону приватних IP-адрес (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 – для IPv4 адрес та FC00::/7 – для IPv6 адрес).

До приватних IP-адрес немає доступу із мережі Інтернет, тому щоб пристрої, які мають приватні IP-адреси, отримали доступ до глобальної мережі і навпаки застосовують технологію **NAT (Network Address Translation**— «перетворення мережевих адрес») – механізм, що дозволяє змінювати IP-адресу в заголовку пакетів, при проходженні даного пакету через пристрій маршрутизації (рис. 2).

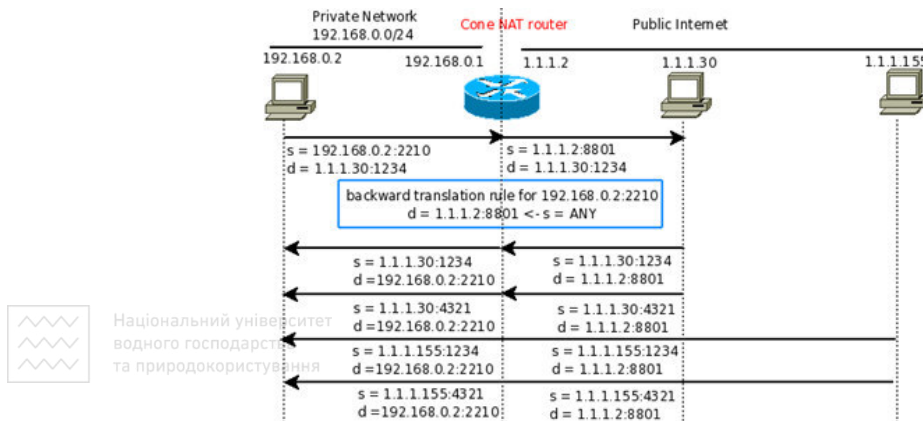


Рис. 2. Принцип роботи NAT

Розрізняють наступні типи NAT:

- 1) Статичний NAT – підміна приватної IP-адреси на публічну на основі один до одного. Корисно, коли пристрій повинен бути доступним із глобальної мережі;
- 2) Динамічний NAT – підміна приватної IP-адреси на публічну з групи зареєстрованих, Динамічний NAT також встановлює безпосередній зв'язок між приватними та публічними адресами, але взаємозв'язок може змінюватись під час комунікації в залежності від публічної адреси, доступної у пулі адрес;

3) Перевантажений NAT (NAT Overload, PAT, NAPT) – форма динамічного NAT, яка перетворює кілька приватних IP-адрес на одну публічну, використовуючи різні порти. При використанні даного типу NAT кожен комп'ютер використовує одну і ту ж саму публічну IP-адресу, проте з різним номером мережевого порту [5].

Програмна реалізація проекту. Розроблений програмний продукт забезпечує можливість моніторингу основних характеристик локальної комп'ютерної мережі для її підтримки в робочому стані.

Функціонал розробленого додатка:

- Визначення IP та MAC-адрес пристрою;
- Збір інформації про структуру мережевого трафіку;
- Збір інформації про мережеві адаптери пристрою (назва, IP адреси);
- Проведення тестування швидкості з'єднання (швидкість завантаження та вивантаження, затримка в мережі);
- Можливість захоплення та аналізу IP-пакетів, що передаються мережею;
- Пошук адрес локальних DNS серверів;
- Сканування відкритих портів;
- Отримання записів DNS сервера.

Для розміщення віджетів у вікнах програми використано менеджери геометрії grid та pack. Grid організовує віджети у структурі, подібній до таблиці, у батьківському вікні або фреймі. Pack організовує віджети у блоки перед їх розміщенням у батьківському вікні чи фреймі. Вигляд стартового вікна програми та розміщені на ньому віджети продемонстровано на рисунку 3.

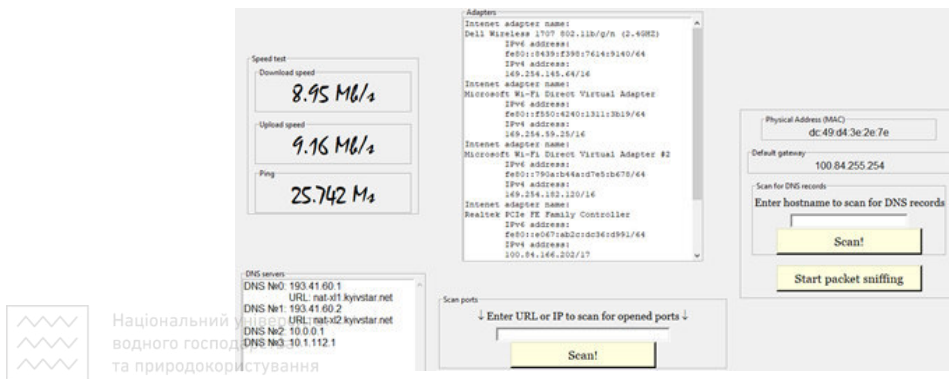


Рис. 3. Вигляд стартового вікна програми

Для підключення до хосту створюємо новий сокет за допомогою функції `socket.socket(socket.AF_INET, socket.SOCK_STREAM)`. Перший аргумент представляє сімейство адрес та протоколів, другий аргумент встановлює режим з'єднання, описує доставку, порядок доставки та цілісність пакетів, контроль потоку даних. Використовуємо функцію `s.connect()` для спроби підключення до IP-адреси по певному мережевому порту. В якості аргументу функції передають кортеж із IP-адреси та номеру мережевого порту. Після успішного встановлення з'єднання додаємо номер мережевого порту, по якому вдалося встановити з'єднання, у список і закриваємо сокет викликом функції `close()`. Після отримання номерів усіх відкритих портів, їх список виводиться на екран, що продемонстровано на рисунку 4.

Перехоплення IP-пакетів відбувається функцією `data = soc.recvfrom(65565)`, де аргумент – розмір буферу (65565 – максимально можливий розмір IP-пакету в байтах).

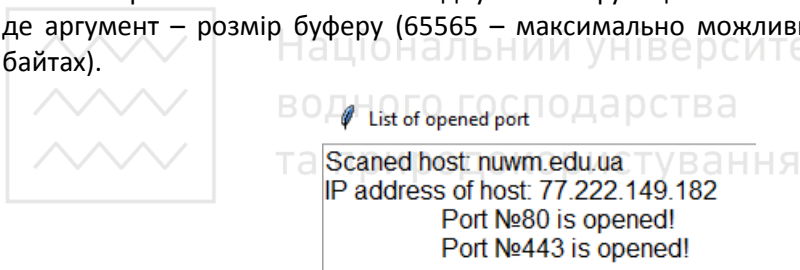


Рис. 4. Вигляд вікна із списком відкритих портів

Перехоплені дані представлені у двійковому вигляді. Для їх представлення у зручному вигляді проведемо розпакування заголовку IP-пакету: `unpackedData = struct.unpack('!BBNNHBBN4s4s', data[:20])`. Аргумент `!BBNNHBBN4s4s` означає в якому типі даних потрібно представити розпаковані дані (B - цілі беззнакові числа, H – беззнакові символи, 4s – рядок з чотирьох символів). Стандартна довжина заголовку IP-пакету 160 біт (20 байт), тому для отримання заголовку працюємо зі зрізом перших 20-ти байтів бінарних даних (другий аргумент функції). Структура заголовку IP-пакету представлена на рисунку 5.

Offsets	Octet	0				1				2				3																			
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP				ECN				Total Length															
4	32	Identification												Flags				Fragment Offset															
8	64	Time To Live				Protocol				Header Checksum																							
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160																																
!	!																																
60	480																	Options (if IHL > 5)															

Рис. 5. Структура заголовку IP-пакету



Національний університет
водного господарства
та природокористування

Висновки. В ході виконання роботи було розглянуто основні характеристики комп'ютерної мережі, що впливають на її роботу. Проведено огляд комп'ютерної мережі, як об'єкту для моніторингу та технологій, які застосовують при роботі в комп'ютерній мережі. Отримані результати можуть використовуватись для аналізу поточного стану комп'ютерної мережі та для покращення її роботи. Запропоновано

1. Моніторинг і аналіз мереж. АКАДЕМИК. URL: <https://dic.academic.ru/dic.nsf/ruwiki/1567353>, вільний. (дата звернення: 06.05.2021).
2. J. Case, M. Fedor, M. Schoffstall, J. Davin. A Simple Network Management Protocol (RFC 1067). Internet Engineering Task Force, 1988. 33 с.

Bernik V. O., Senior Student; Matus O. V., Senior Student (National University of Water and Environmental Engineering, Rivne)

DEVELOPMENT OF SOFTWARE FOR MONITORING THE BASIC CHARACTERISTICS OF THE LOCAL AREA NETWORK

The work is devoted to the research of devices that form a local area network, their basic characteristics and methods which allow to determine the characteristics of the network that have an influence on its functioning.

Keywords: Internet, computer network, internet protocols, traffic, monitoring, analysis.

УДК 519.6

Кондратюк К. М., студентка (Національний університет водного господарства та природокористування, м. Рівне)

ОГЛЯД ОСНОВНИХ ПІДХОДІВ ДО МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ІНФЕКЦІЙНИХ ЗАХВОРЮВАНЬ

У статті проведено огляд основних та найефективніших підходів до математичного моделювання епідемій. Також розглянуто моделі та методи мультиагентного моделювання, які широко застосовуються в прогнозуванні епідемій.

Ключові слова: інфекція, хвороба, математичне моделювання, епідемія, мультиагентне моделювання, прогнозування.

У розпал поширення в усьому світі гострої респіраторної хвороби COVID-19 надзвичайно актуальним є питання дослідження та прогнозування наслідків пандемії. Сучасна медицина в основному є доволі експериментальною наукою з величезним емпіричним досвідом впливу на ту чи іншу хворобу. Але для більш детального вивчення процесів біосередовищ недостатньо лише експериментальних досліджень, тому стає в пригоді математичне моделювання. Математичне моделювання – це технологія дослідження реальних процесів і явищ на їх математичних моделях [1]. У зв'язку з ускладненням об'єктів дослідження різко зростає роль математичного моделювання як методу пізнання реальності. В основі даної методології лежить заміна об'єкта дослідження його образом – математичною моделлю. За допомогою цього потужного інструменту розв'язують різні технологічні, інженерні та наукові проблеми.

Моделювання дозволяє варіювати параметрами системи й середовища в досить широких межах, дає економію засобів при проектуванні систем, усуває зайві витрати людських та матеріальних ресурсів [1]. Модель є необхідною для:

- розуміння, яка будова конкретного об'єкта, яка його структура, якими є основні властивості, принципи його функціонування та взаємодії з навколишнім світом;
- навчання щодо керування об'єктом або процесом, пізнання найкращих способів управління за заданими цілями та критеріями;
- прогнозування прямих та непрямих наслідків реалізації різних способів і форм управління об'єктами та системами [1].

Математичне моделювання – найвища форма моделювання, воно є найбільш універсальним, найбільш застосовним, найефективнішим, економічним та потужним засобом пізнання процесів та явищ [1]. Це можна пояснити тим, що цей шлях моделювання опирається на методи математики та можливості ЕОМ. Практична реалізація можливостей математичного моделювання підвищує ефективність науково-технічних розробок.

Специфіка розв'язуваних задач формує саме той перелік засобів математичного моделювання, що будуть використовуватися для опису біологічних та медичних об'єктів. У медичній сфері існує багато різних завдань, які для свого розв'язання вимагають застосування теорії математичного моделювання. Наприклад, математичні моделі прогнозу динаміки розвитку онкологічних захворювань; роботи системи кровообігу; інфаркту міокарда; динаміки поширення інфекційних захворювань та інші.

В математичному моделюванні широко застосовуються диференціальні рівняння. Це можна пояснити тим, що досить часто об'єктивні закони, яким підпорядковуються певні явища та процеси, записують у формі диференціальних рівнянь, а самі ці рівняння є засобом для кількісного вираження цих законів. Для дослідження складних процесів в об'єктах, що змінюються з плином часу, застосовуються дескриптивні (описові) математичні моделі у вигляді диференціальних рівнянь (або систем диференціальних рівнянь) [2]. Рівняння складаються на основі фізичних, хімічних або біологічних законів. Розв'язками таких систем є функції часу, тому вони можуть описувати зміни в часі процесів, що відбуваються всередині модельованих об'єктів [2].

Протягом багатьох десятиліть вчені з усього світу розробляли власні моделі для опису поширення та прогнозування інфекційних захворювань. Розглянемо найпопулярніші підходи, які використовуються для математичного моделювання поширення інфекційних захворювань.

Початок у вивченні динамічних процесів поширення інфекцій поклали Дж. Граунт та У. Петті, які в XVII ст. склали так звані «біллі про смертність в Лондоні» [3]. В даній роботі представлена статистика смертності в Лондоні протягом двадцятирічного періоду і на основі цієї статистики вони оцінювали ймовірність загинути від певного виду захворювання. Першою моделлю поширення захворюваності вважається робота Д. Бернуллі. В ній вчений за допомогою найпростішого математичного апарату, в основі якого лежать диференціальні рівняння, оцінив ефективність профілактичних заходів.

Дуже важливий внесок в математичне моделювання епідемій зробив В. Гамер, який у 1906 році сформулював та описав принцип, який і досі лежить в основі епідемічного моделювання [3]. Цей принцип базується на використанні диференціальних рівнянь. Нехай $x(t)$ – це кількість здорових та схильних до захворювання індивідів, а $y(t)$ – кількість інфікованих. Тоді зміну інфікованих індивідів з часом можна описати так

$$\frac{dy}{dt} = \beta xy,$$

де β – параметр, що описує інтенсивність передачі інфекції.

Більшість відомих на сьогоднішній день моделей поширення інфекційних захворювань є неперервними. В таких моделях динаміка процесу описується за допомогою системи диференціальних рівнянь. Розв'язком цієї системи є рівень процесу, що розглядається в кожен конкретний момент модельного часу [3].

Досить відомою математичною моделлю в епідеміології стала SIR модель Кермака и МакКендріка, яка була описана в 1927 році [3]. В даній моделі популяція ділиться на 3 групи: сприйнятливих, інфікованих та тих, що одужали. Функції $S(t)$, $I(t)$, та $R(t)$ позначають їх відповідні частки в популяціях в момент часу t . Якщо припустити що загальна кількість індивідів популяції залишається незмінною, то приріст в кожній з цих груп можна описати за допомогою наступної системи рівнянь [3]

$$\frac{dS}{dt} = -\beta SI, \quad \frac{dI}{dt} = \beta SI - \gamma I, \quad \frac{dR}{dt} = \gamma I,$$

де β – інтенсивність контактів між індивідами, γ – інтенсивність переходу індивідів в стан R . При застосуванні такого методу моделювання можна зробити висновки про епідемічний поріг, розмір епідемії, коли вона відбудеться. При використанні такої моделі не враховується багато чинників, які є важливими при поширенні інфекційного захворювання. Наприклад, дана модель не враховує що деякі люди можуть бути більш сприйнятливими до інфекції або більш заразними ніж інші, деякі люди більше контактують з іншими людьми, а деякі навпаки менше. Також дана модель не враховувала сезонність, а це одна з основних характеристик.

SIR модель отримала багато різних модифікацій, які призначені для різних процесів та враховують різні особливості моделювання конкретних випадків. Однією з таких модифікацій є SEIR модель, яка найбільше підходить для моделювання поширення грипу. Модифікація відбувається за рахунок введення ще однієї групи індивідів – люди під час інкубаційного періоду інфекції [3]. Тоді система рівнянь набуває наступного вигляду [3]

$$\frac{dS}{dt} = B - \beta SI - \mu S, \quad \frac{dE}{dt} = \beta SI - (\varepsilon + \mu)E, \quad \frac{dI}{dt} = \varepsilon E - (\gamma + \mu)I, \quad \frac{dR}{dt} = \gamma I - \mu R,$$

де B – середній рівень народжуваності індивідів, μ – середній рівень смертності індивідів, $\frac{1}{s}$ – середня тривалість інкубаційного періоду.

Моделі SIR та SEIR доволі прості, не вимагають великих обчислювальних витрат, що дозволяє реалізувати їх в будь-якій системі комп'ютерної математики або мовою програмування. Вчені навіть розробили середовище моделювання STEM, в якому дані моделі вже реалізовані та можуть використовуватися після мінімальних налаштувань [3].

Найвідомішою моделлю на сьогоднішній день є модель, що була запропонована О. В. Барояном та Л. А. Рвачевим. Дана модель вважається найкращою детермінованою моделлю епідемічної поведінки динамічних систем [3]. Модель Барояна-Рвачевича схожа до моделей SEIR, але має свої характерні відмінності. В ній так само використовується поділ населення на 4 групи, але для визначення приросту кількості індивідів в кожній з груп використовується система нелінійних інтегро-диференціальних рівнянь в частинних похідних, що має наступний вигляд:

$$\frac{dS(t)}{dt} = -\frac{\lambda}{p(t)} S(t) \int I(t, \tau) d\tau, \quad \frac{\partial E(t, \tau)}{\partial t} + \frac{\partial E(t, \tau)}{\partial \tau} = -\gamma(\tau) E(t, \tau),$$

$$\frac{dI(t, \tau)}{dt} + \frac{\partial I(t, \tau)}{\partial \tau} = \gamma(\tau) E(t, \tau) - \delta(\tau) I(t, \tau), \quad \frac{dR(t)}{dt} = \int \delta(\tau) I(t, \tau) d\tau,$$

де t – календарний час розвитку епідемії, λ – середня частота передачі збудника від інфікованих до чутливих індивідів, p – чисельність популяції, τ – час, що минув з моменту зараження індивіда, $\gamma(\tau)$ – функція розвитку періоду інкубації, $\delta(\tau)$ – функція розвитку інфекційного періоду. Для можливості використання диференціальних рівнянь у моделях SEIR типу припускається, що всі індивіди на території, що моделюється є однотипними, безперервно та рівномірно перемішуються. Реальні популяції мають значно складнішу структуру, що зумовлена різними чинниками. Зазвичай SEIR моделі використовують при дослідженні поведінки дуже великих популяцій, що зумовлено повною детермінованістю моделей.

Незважаючи на популярність SIR моделей, існують і інші підходи до побудови стохастичних моделей динамічних систем епідемічного характеру. Наприклад є моделі в яких кількість сприйнятливих до захворювання індивідів ($X(t)$) та кількість джерел захворювання $Y(t)$ представлені як випадкові процеси [3]. Тобто якщо

частота контактів між індивідами – β , то ймовірність появи в інтервалі Δt нового джерела захворювання буде дорівнювати $\beta XY \Delta t$. Якщо частота видалення з популяції заражених індивідів дорівнює γ , то ймовірність того, що в інтервалі Δt буде видалений один індивід дорівнює $\gamma Y \Delta t$ [3]. Якщо замінити часовий масштаб на $\tau = \beta t$, і позначити відносну частоту видалення $\frac{\gamma}{\beta} = \rho$, то отримаємо наступне рівняння для функцій ймовірностей [1]



Національний університет
водного господарства
та природокористування

$$\frac{\partial P}{\partial \tau} = (y^2 - xy) \frac{\partial^2 P}{\partial x \partial y} + \rho(1 - y) \frac{\partial P}{\partial y}.$$

Отримати розв'язки подібних рівнянь досить складно, тому моделі такого типу зазвичай не використовуються для розв'язання практичних задач моделювання поширення інфекційних захворювань.

Найбільш актуальними та перспективними на сьогоднішній день є дискретно-подійні моделі. Як найпростіший приклад дискретно-подійної моделі можна навести клітинний автомат. Клітинний автомат – дискретна математична модель, яка визначає сукупність та описується набором клітинок, що утворюють періодичну решітку, та заданими правилами переходу, що визначають стан клітини за теперішнім станом самої клітинки та тих її сусідів, що знаходяться від неї на певній відстані, яка не перевищує максимальну [4].

Першопрохідцем в даному напрямку став Н. Бейлі. У своїх дослідженнях він розглядав квадратну решітку, кожен вузол якої зайнятий одним індивідом. В один з вузлів він поміщав джерело захворювання і розглядав такий процес, в якому в зоні ризику знаходилися лише ті індивіди, які прилягають до джерела захворювання. На кожному часовому кроці такої моделі інфекція з певною ймовірністю може бути передана від зараженого індивіда до одного з сприйнятливих сусідів [3]. Зазвичай популяція, що моделюється ділиться на ті самі 4 групи, що і моделі типу SEIR: сприйнятливі індивіди при достатньому контакті з інфікованим переходять в стан інкубаційного періоду, після закінчення якого також стають хворими. Хворі індивіди з часом одужують та, або залишаються в цьому стані, або переходять в стан сприйнятливих.

Колосальний внесок в розвиток моделювання динамічних систем з епідемічною поведінкою за останні роки зробили популяційні моделі. Ці моделі створені на основі дискретно-подійних та відрізняються тим, в них популяція розділена на соціальні групи, які формуються за віком індивідів, а в деяких детальних моделях може враховуватися навіть рід занять. При застосуванні популяційної моделі припускається, що поширення захворювання можливе лише в межах однієї «контактної» групи [3]. Залежно від структури популяції, що моделюється утворюються контактні групи. В цій моделі індивіди щодня формують контактні групи, в яких можлива передача інфекції від хворого до здорового.

Одна з найперших популяційних моделей була запропонована І. Лонгіні, М. Халоррана, А. Нізама та Я. Янга. В даній моделі розглядається популяція чисельністю 10 000 осіб, яка ділиться на п'ять груп по 2000 осіб. Кожна така група включає в себе по 4 райони, одну середню та дві початкові школи. В кожному районі перебуває кілька сімей («С»), дитячих садків («ДС») та ін. Кожна з цих локацій утворює свою контактну групу. В залежності від віку індивід відвідував декілька локацій

(наприклад, школяр відвідував локацію «сім'я» та «школа»). Тип локації також визначав ймовірність передачі інфекції від хворого індивіда до сприйнятливою. Етапи перебігу захворювання в популяційних моделях зазвичай такі ж як і в моделях SEIR, час рухається з фіксованим кроком дискретизації [3]. На кожному кроці аналізується, які локації відвідував той чи інший індивід та аналітично розраховується ймовірність його зараження за минулий проміжок часу. Для розрахунку ймовірності використовуються складні формули, які враховують багато різних чинників [3]. Більшість популяційних моделей відрізняються одна від одної в більшості структурою популяції та формулою для розрахунку ймовірності передачі захворювання сприйнятливому індивіду.

Найбільш вдалі популяційні моделі лягли в основу систем підтримки прийняття рішень та середовищ моделювання. Популяційні моделі та середовища моделювання – найкращий спосіб на сьогоднішній день для моделювання поширення інфекцій у великих популяціях з урахуванням їх географічних та соціальних особливостей. Проте формула для розрахунку ймовірності достатнього контакту – основне джерело похибок в роботі популяційної моделі [3]. Великий крок дискретизації та необхідність калібрування коефіцієнтів, які використовуються, можуть привести до того, що результати роботи популяційної моделі допустимої точності для дуже великих популяцій будуть вже непридатними для менших популяцій. Під калібруванням мається на увазі процес настроювання наборів початкових даних таким чином, щоб забезпечити максимальне наближення результатів розрахунку до даних натурних експериментів [3]. Але це можна виправити шляхом розширення популяційних моделей до мультиагентних моделей поширення захворювань.

Мультиагентні моделі поширення епідемічних процесів схожі на популяційні. В якості агентів в цих моделях так як і в популяційних розглядаються індивіди, що утворюють популяцію, але в мультиагентних моделях кожен індивід має індивідуальні властивості, наприклад, вікову групу [3]. При застосуванні мультиагентної моделі, так само як і в популяційній формуються контактні групи, між якими агенти рухаються та в яких можлива передача інфекції. Основною відмінністю мультиагентних моделей є децентралізованість. В популяційних моделях чітко визначений період часу через який в моделі виникає подія обробки всіх переміщень індивідів, що відбулися з попередньої такої події. Вся обробка ведеться централізовано та за певними аналітичними формулами, що визначають ймовірність різних змін в моделі [3]. В мультиагентних моделях, кожен агент як активний об'єкт самостійно викликає всі події, пов'язані з його рухом між локаціями, поширенням захворювання, розвитком захворювання та інше. Кожна така подія містить в собі набір нескладних дій, але навіть при малих популяціях агенти викликають сотні подій, обробка яких вимагає чималих обчислювальних ресурсів. Мультиагентні моделі поширення захворювань є найбільш перспективними, адже вони не мають явних обмежень. Жоден інший підхід до моделювання поширення захворювань не дає таких широких можливостей, як мультиагентний. Такі моделі підходять для створення систем підтримки прийняття рішень. Також мультиагентна модель, що була побудована для одного захворювання, може бути доповнена та використана для іншого інфекційного захворювання. Цей підхід є доволі молодим, але вже отримав визнання та зацікавив багатьох вчених з усього світу. Однією з опублікованих робіт на цю тему є робота японців Ю. Окхусо і Т. Суговара. В своїй

роботі вони представили мультиагентну модель поширення захворювання з повітряно-крапельним шляхом передачі, яка містить в собі детальне пересування агентів громадським транспортом [3]. В моделі вчені використали реальні розклади поїздів, автобусів та врахували способи передачі інфекцій в цих місцях. Однак реалізація настільки детальної моделі потребує багато обчислювальних ресурсів, тому проектувальники змодельовали поширення хвороби лише для дуже малої популяції. Ця робота отримала лише одну реалізація випадкового процесу, що визначається моделлю, тому на її підставі неможливо зробити адекватні висновки.

На сьогоднішній день не існує моделі поширення епідемічного процесу, яка б повністю реалізовувала всі можливості мультиагентного підходу.



Національний університет
водного господарства
та природокористування

1. Маценко В. Г. Математичне моделювання. Чернівці : Чернівецький національний університет, 2014. 519 с.
2. Диференціальні рівняння як математична модель реальних процесів. URL: <https://naurok.com.ua/diferencialni-rivnyannya-yak-matematichna-model-realnih-procesiv-109415.html> (дата звернення: 06.05.2021).
3. Чумаченко Д. І., Чумаченко Т. О. Математичні моделі та методи прогнозування епідемічних процесів. Харків : ТОВ «Планета-Прінт», 2020. 180 с.
4. Клітинний автомат. URL: https://uk.wikipedia.org/wiki/%D0%9A%D0%BB%D1%96%D1%82%D0%B8%D0%BD%D0%BD%D0%B8%D0%B9_%D0%B0%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82 (дата звернення: 06.05.2021).

Kondratiuk K. M., Senior Student (National University of Water and Environmental Engineering, Rivne)

OVERVIEW OF MAIN APPROACHES TO MATHEMETICAL MODELING OF INFECTIOUS DISEASES

In this article was conducted review of most famous and effective approaches to epidemic processes modeling. In addition, there are analysis of models and methods of agent-based modeling, which are widely used in predicting the dynamic of epidemics.

Keywords: infection, disease, mathematical modeling, epidemic, agent-based modeling, predicting.

УДК 519.863:004.8;336.7

Бабич¹ Т. Ю., к.е.н., доцент; Ящук² Д. А., студент (Національний університет водного господарства та природокористування, м. Рівне,

¹t.iu.babych@nuwm.edu.ua, ²yashchuk_em19@nuwm.edu.ua)

НЕЙРО-НЕЧІТКІ МЕРЕЖІ В ПРИЙНЯТТІ РІШЕНЬ У ФІНАНСОВІЙ ДІЯЛЬНОСТІ

Запропоновано шляхи визначення кредитного рейтингу при зверненні потенційного клієнта з метою отримання кредиту. Розроблено нечітку модель

засобами нечіткої логіки згідно алгоритму Сугено. На базі експертних даних виконано аналіз адекватності побудованої моделі. Здійснено підвищення її адекватності через налаштування нечітких логічних правил та параметрів функцій належності. Показано, що в результаті застосування генетичного алгоритму до налаштування можна досягти значного підвищення адекватності моделі до навчальних даних. Рекомендовані шляхи вдосконалення моделі дають змогу уточнювати модель відповідно до актуальних вимог до позичальника. Налаштовану модель можна використовувати у фінансових організаціях для прийняття обґрунтованих рішень щодо можливості видачі кредиту.

Ключові слова: визначення кредитоспроможності, нечітка модель, алгоритм Сугено, функція належності, налаштування нечіткої моделі

На сучасному етапі розвитку світової економіки спостерігається зростання динаміки фінансових операцій. Кредитні операції є найбільш дохідними для банку, але й дуже ризикованими водночас. У процесі кредитування необхідно виконати аналіз фінансових ризиків невиконання позичальником взятих на себе зобов'язань, тобто неповернення, невчасного повернення або повернення не в повному обсязі кредитних коштів. Для банків неповернення кредитів є важливою проблемою. Рівень кредитного ризику значно зростає. Саме тому одним із найбільш актуальних питань залишаються дослідження у напрямку розробки якісних моделей оцінки кредитоспроможності позичальників. При визначенні кредитного рейтингу позичальника кредиту, як правило, доводиться оперувати лінгвістичними поняттями («великий стаж роботи», «поганий матеріальний стан» тощо). Тому розробити систему прийняття рішень в умовах неповної та нечіткої інформації є можливим через залучення до управління кредитним ризиком банку засобів нечіткої логіки.

Різні методики оцінювання кредитного ризику розглядаються в роботах Зайченка Ю.П., Камінського А.Б., Матвійчука А.В., Шаповал Н.В., що пов'язані з розробкою теорії ризику в умовах невизначеності інформації. Основи нечіткої логіки та розробки нечітких нейронних мереж викладені в роботах Заде Л., Мамдані Е., Сугено М.

Банківська діяльність характеризується підвищеними ризиками порівняно з іншими видами комерційної діяльності. Тому у банківській практиці процес управління ризиками розглядається як ключовий напрям банківського менеджменту. Банківські ризики, як і ризики в інших видах комерційної діяльності, перш за все пов'язують з фінансовими втратами, що виникають у випадку їх реалізації [1]. Оскільки взяття ризиків є основою банківської діяльності, банки мають успіх тоді, коли прийняті ними ризики будуть розумними, контрольованими та перебувати у межах їх фінансових можливостей і компетенцій. Серед основних банківських ризиків Національний банк виокремлює: кредитний ризик, ризик ліквідності, ризик зміни відсоткової ставки, ринковий ризик, валютний ризик тощо.

За визначенням Базельського комітету з банківського нагляду, кредитний ризик означає можливість того, що позичальник або контрагент банку не виконає свої зобов'язання у погоджений термін. Мета управління кредитним ризиком полягає в максимізації норми прибутку банку з урахуванням ризику, шляхом підтримання кредитного ризику в межах допустимих параметрів. Банки повинні

управляти кредитним ризиком, властивим всьому портфелю, а також ризиком окремих кредитів або операцій [2].

Серед особливостей оцінювання кредитного ризику значною мірою наявний суб'єктивний підхід (людський фактор). Один і той самий позичальник може отримати в різних банках різну оцінку рівня кредитного ризику, відповідно, і різний висновок щодо можливості видачі (або відмови у видачі) кредиту. Внаслідок переважання індивідуальної складової кредитний ризик погано описується поняттям статистичної сукупності, тому в процесі оцінювання кредитного ризику індивідуальна робота з позичальником є дуже важливою ланкою прийняття рішень. Отже, постає необхідність створення певного методу, що враховував би особливості кожного конкретного позичальника.

Таким чином, проблема зниження кредитного ризику банку істотно залежить від досконалості застосовуваних банком методів його оцінювання.

Існують різні підходи до визначення кредитного ризику для фізичної особи, серед яких суб'єктивні оцінки фахівців банку, засновані на особистому досвіді чи на враженні про конкретного клієнта, автоматизовані системи оцінки ризику, створені з використанням математичних моделей. Найчастіше для оцінки кредитоспроможності позичальника використовують: статистичні методи, дерева рішень, генетичні алгоритми, нейронні мережі. Кожна кредитна організація сама визначає, якими методами користуватися.

Основними *перевагами* систем з нечітким логічним висновком є те що:

1. Вони дозволяють втілити природний перехід від експертного висновку до нечітких «Якщо-То» правил;
2. Можна обирати механізм нечіткого логічного висновку найбільш підходящий для поставленої задачі;
3. Вхідні та вихідні дані приймають реальні значення.

Недоліком же систем нечіткого логічного висновку є те, що вони не здатні автоматично навчатися і параметри та вид функцій належності, які описують фактори кредитоспроможності, задаються експертом та можуть бути неадекватними. Нечіткі нейронні мережі поєднують переваги систем нечіткого логічного висновку та нейронних мереж – здатність до адаптації та автоматичного навчання, а також можливість інтерпретації процесу отримання результату.

Для визначення кредитоспроможності позичальника побудова нейронечіткої моделі складається з двох етапів. Перший етап (структурна ідентифікація) – формування нечіткої бази знань, що грубо відтворює залежність виходу від входів за допомогою лінгвістичних правил «ЯКЩО–ТО», які генеруються з експериментальних даних [3]. Наприклад, ЯКЩО <позичальник працевлаштований> ТА <заробітна плата позичальника висока> ТА <позичальник є власником нерухомості>, ТО <кредитний ризик щодо позичальника низький>.

На другому етапі проводиться параметрична ідентифікація: пошук таких параметрів нечіткої бази знань, які мінімізують відхилення теоретичних (розрахованих за моделлю) від експериментальних результатів.

У випадку розв'язання задачі визначення кредитоспроможності отримане за такою процедурою число характеризує рішення, що прийняте банком стосовно видачі кредиту на основі кредитного рейтингу потенційного клієнта.

Розглянемо алгоритм нечіткого логічного висновку [4]. Кожну кредитну заявку можна представити вектором $X = \{X_1, X_2, \dots, X_i, \dots, X_N\}$, де X_i – певним чином формалізовані дані з анкети позичальника та параметри кредиту. Банку необхідно по заданому вектору прийняти фінансове рішення про надання кредиту, тобто віднести позичальника до групи «відмовити у кредитуванні», до групи «надати кредит» тощо. У нашому випадку нечіткий логічний висновок, загалом має такі етапи:

1. Визначення множини вхідних змінних про позичальника:

$$X = \{X_1, X_2, \dots, X_i, \dots, X_M\}.$$

2. Визначення множини вихідних змінних – можливі відповіді щодо кредитування: $D = \{D_1, D_2, \dots, D_i, \dots, D_M\}$.

3. Формування базової терм-множини з відповідними функціями належності кожної змінної: $A = \{a_1, a_2, \dots, a_i\}$.

4. Формування кінцевої множини нечітких правил, узгоджених щодо використовуваних в них змінних.

5. Реалізація логічного висновку, тобто визначення істинності для передумов кожного правила та визначення нечітких підмножин для змінних виходу для кожного правила.

6. Композиція нечітких підмножин кожної змінної виходу по всім правилам.

7. Знаходження чіткого значення для кожної з вихідних лінгвістичних змінних.

Першим кроком при зверненні за кредитом для потенційного позичальника є заповнення анкети. Інформація з анкети може бути неточною, неповною, а дані про позичальника можуть бути такими, що важко формалізуються. Для врахування цих факторів доцільно використовувати нечітку логіку, де процес прийняття рішень базується на зрозумілій базі правил [3].

До характеристик фізичної особи-позичальника кредиту відносять комплексну інформацію: наявність рахунку у банку, термін, на який береться кредит, кредитна історія клієнта, мета використання кредиту, сума кредиту, сума на рахунках у банках, тривалість роботи на одному місці, часткова сплата представленої суми, сімейний стан, можливість поручителя, тривалість проживання на одному місці, фінансовий стан, вік клієнта, борги клієнта, характеристики квартири/будинку, кількість попередніх кредитів, спеціальність, кількість осіб на утриманні, наявність телефону, належність до жителів чи гостей [5]. У даній роботі як найбільш вагому, вирішено розглядати таку інформацію про позичальника (лінгвістичні змінні): вік позичальника, рівень його доходу, майновий стан та кредитна історія. На етапі аналізу даних, що вказано в анкеті, банк оцінює кредитоспроможність позичальника та на базі отриманої оцінки формує рішення про видачу кредиту. Кількісні та якісні характеристики позичальника, що відображені в анкеті, представлено у вигляді

$$y = f(x_1, x_2, x_3, x_4), \quad (1)$$

де y – рішення банку щодо видачі кредиту; x_1 – вік; x_2 – дохід; x_3 – майновий стан; x_4 – кредитна історія. Кожна лінгвістична змінна описується відповідними

термами (табл. 1).

Таблиця 1

Ідентифікація вхідних та вихідної змінної для визначення кредитоспроможності позичальника

Змінна	Лінгвістична змінна	Терми		Діапазон
у	Decision Рішення про можливість кредитування	Відмова в кредитуванні	NO - 1	[1–3]
		Видача з жорсткими умовами	MB - 2	
		Надати	YES - 3	
x ₁	Age Вік	Молодий	Y	[18–70]
		Середній	AV	
		Старший	O	
		Пенсійний	P	
x ₂	Income Дохід	Низький	L	[5000–50000]
		Середній	A	
		Високий	H	
x ₃	Property Майновий стан	Немає майна	NP	[1–3]
		Цільовий кредит	G	
		Власник	OW	
x ₄	Credit History Кредитна історія	Негативна	N	[1–3]
		Відсутня	AB	
		Позитивна	POS	

Крім універсальних множин, слід задати також функції належності, за допомогою яких формалізуються терми лінгвістичних змінних. Параметри функцій належності визначаються в процесі побудови функцій.

У даній роботі було використано гаусові функції для всіх термів всіх змінних:

$$\mu^{a_j}(x_i) = \exp\left[-\frac{(x_i - g_{a_j})^2}{2h_{a_j}^2}\right], \quad (2)$$

де a_j – j -й терм змінної $x_i, i = 1, 2, j = 1, \dots, k_i, k_i$ – кількість термів змінної x_i ; g_{a_j} – координата максимуму функції $\mu^{a_j}(x_i)$; h_{a_j} – коефіцієнт стиснення-розтягу.

Генерування нечіткої системи прийняття рішень виконано за допомогою інструменту Fuzzy Logic ToolBox середовища Matlab згідно алгоритму Сугено [3]. У запропонованій моделі алгоритмом передбачається формування бази правил системи нечіткого виводу у вигляді впорядкованого погодженого списку нечітких продукційних правил у вигляді

«IF A AND B THEN w»,

де антецеденти ядр правил нечіткої продукції побудовані з двох простих нечітких висловлювань A, B за допомогою логічних зв'язок «I», w – чітке значення вихідної

змінної, визначене в нечіткому правилі, як дійсне число (1, 2 або 3). У випадку розв'язування задачі визначення кредитоспроможності отримане за такою процедурою число характеризує рішення, що прийняте банком стосовно видачі кредиту на основі кредитного рейтингу потенційного клієнта.

Максимальна можлива кількість нечітких правил відповідає кількості всіх можливих комбінацій термів лінгвістичних змінних – 108. При побудові нечіткої бази знань і відповідно нечітких правил використовувалися знання експертів.

Основні критерії були такими:

- «молодий» вік вважається ознакою ненадійності позичальника, особливо за наявності «низького» рівня доходу. Отримати позитивне рішення «з жорсткими умовами» передбачено лише для позичальників з рівнем доходу не нижче «середнього» та наявності «позитивної» або «відсутньої» кредитної історії;
- «середній» вік позичальника позитивно оцінюється банком, тут дозволено мати не «низький» рівень доходу, «позитивну» або «відсутню» кредитну історію тощо;
- рівень кредитного ризику дещо збільшується, коли у кредитній історії присутня невизначеність (раніше фізична особа не брала кредит) та значно збільшується у випадках «пенсійного» віку та «середнього» рівня доходів.

Після формування бази правил зайві (з нашого погляду) продукційні правила було виключено на базі таких принципів:

- 1) виключення суперечливих правил, які взаємно компенсуються;
- 2) виключення одного з двох співпадаючих правил, як таких, що не несуть нової інформації;
- 3) видалення тих продукційних правил, вплив яких на точність виявляється мінімальним.

Наприклад, для всіх позичальників, яких можна віднести до молодого віку, дохід яких не є низьким, які не відносяться до таких, що не мають майна, кредитна історія у яких не є негативною, рішення щодо кредитування буде прийнято позитивне. Після виключення надлишкових правил та їх узагальнення кількість правил було скорочено до 28 (нечітка модель 1- НМ-1).

Апробацію роботи НМ-1 було виконано на базі вибірки уявних звернень за кредитом та відповідних рішень щодо видачі кредитів експертами одного з вітчизняних банків засобами нейро-нечіткого моделювання (70 записів). Інформація про позичальників кредиту відповідає структурі побудованої НМ-1. У результаті обробки даних навчальної вибірки було виявлено розходження між рішеннями щодо видачі кредиту, які було прийнято експертами, та рішеннями НМ-1. Варто відзначити, що НМ-1 має на меті перш за все вигоду банку, тому в ситуаціях, коли наявні певні сумніви в надійності позичальника, у рішенні про видачу кредиту передбачено формулювання «на жорстких умовах».

Для налаштування моделі НМ-1 з метою зменшення кількості відхилень було використано можливості середовища Matlab-2019b та генетичний алгоритм.

Налаштування нечіткої системи виконаємо двома шляхами: перший варіант передбачає налаштування тільки нечітких правил, інший варіант включає в себе налаштування одразу параметрів функцій належності та нечітких правил.

У першому варіанті кожне налаштування параметрів правила включає ім'я нечіткої системи, індекс правила у нечіткій системі та опції параметрів для *antecedent* (попереднього) та *consequent* (наступного) пунктів безпосередньо правила. При налаштуванні використовується метод найменших квадратів та генетичний алгоритм (за замовчуванням). Апробовано 4 комбінації, що пов'язані з дозволом (заборонаю) використання логіки NOT і допуском (заборонаю) відсутності вхідних/вихідних змінних. Максимальну кількість генерацій генетичного алгоритму було встановлено послідовно 5, 7, 9 та 11 для кожної з чотирьох комбінацій дозволів. Після налаштування нечітких правил отримана нечітка модель НМ-2 (кращий варіант з тих, що було отримано) демонструє збільшення кількості співпадінь на 10% і становить 52 випадки (табл. 2). Серед 18 випадків, де рішення не співпадає з експертним, спостерігається 6 випадків погіршення рейтингу позичальника, тоді як 12 – більш поблажливого ставлення та позитивного рішення. Тобто, налаштована НМ-2 не стала більш вимогливою до позичальників кредиту.

Таблиця 2

Результати налаштування нечіткої моделі

	EP	HM-1	HM-2	HM-4
Кількість рішень «Відмовити» (NO)	16	23	18	18
Кількість рішень «Видача з жорсткими умовами» (MB)	34	12	24	35
Кількість рішень «Видати кредит» (YES)	20	35	28	17
Співпадає з експертним рішенням (EP):	вип.	-	45	52
	%	-	64	79

Для покращення НМ-1 через налаштування одразу параметрів функцій належності та нечітких правил було також використано генетичний алгоритм з аналогічними обмеженнями на максимальну кількість генерацій. Проведені комп'ютерні експерименти включали в себе надання дозволів використання заперечень та відсутніх змінних для першої частини нечіткого правила. Для другої частини нечіткого висновку пробували різні комбінації. Найбільш успішним результатом стала НМ-4 – 55 співпадінь з експертним рішенням. Серед 15 випадків, в яких рішення не співпадає з експертним, спостерігається 10 випадків погіршення рейтингу позичальника, і лише в 5 випадках спостерігається більш поблажливе ставлення. Можна констатувати більшу адекватність налаштованої НМ-4 до рішень експерта та достатню ступінь вимогливості до позичальників кредиту.

Разом із підвищенням рівня співпадінь із рішеннями експертів змінився і розподіл по кількості правил, що передбачають той чи інший висновок. Структура правил НМ-4 не змінилась в порівнянні з пропонованою НМ-1 (табл. 3).

Таблиця 3

Результати налаштування правил нечіткої моделі

Кількість правил з результатом	HM-1	HM-2	HM-4
«Відмовити» (NO)	7	9	7
«Видача з жорсткими умовами» (MB)	13	12	13
«Видати кредит» (YES)	8	7	8

Зміни параметрів функцій належності для моделей HM-1 та HM-4 продемонстровано в табл. 4, 5, 6. Параметри функцій належності змінної *Income* залишились незмінними.

Таблиця 4

Параметри функцій належності змінної *Age*

x_1	Y		AV		O		P	
	g	h	g	h	g	h	g	h
HM-1	7.363	18	7.363	35.34	7.363	52.66	7.363	70
HM-4	18	18	18	35.34	18	52.66	18	70

Таблиця 5

Параметри функцій належності змінної *Property*

x_3	NP		G		OW	
	g	h	g	h	g	h
HM-1	0.3398	1	0.3398	2	0.3398	3
HM-4	1	1	1	2	1	3

Таблиця 6

Параметри функцій належності змінної *Credit history*

x_4	N		AB		POS	
	g	h	g	h	g	h
HM-1	0.3398	1	0.3398	2	0.3398	3
HM-4	1	1	1	2	1	3

На рис. 1–4 представлено вигляд функцій належності моделі HM-4 після налаштування.

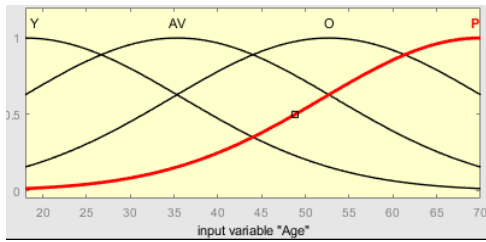


Рис. 1. Функції належності змінної Age (HM-4)

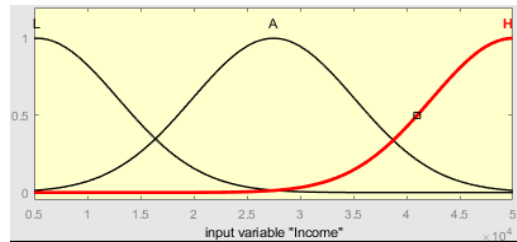


Рис. 2. Функції належності змінної Income (HM-4)

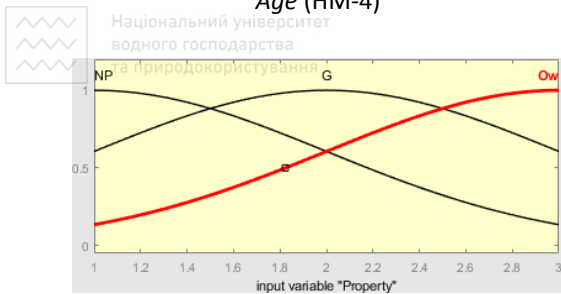


Рис. 3. Функції належності змінної Property (HM-4)

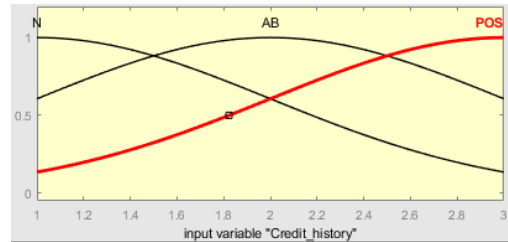


Рис. 4. Функції належності змінної Credit history (HM-4)

Візуально зміну поверхні відгуку функції y (Рішення) на комбінацію вхідних змінних x_1 (Вік) та x_2 (Дохід) до та після налаштування відображено на рис. 5–6.

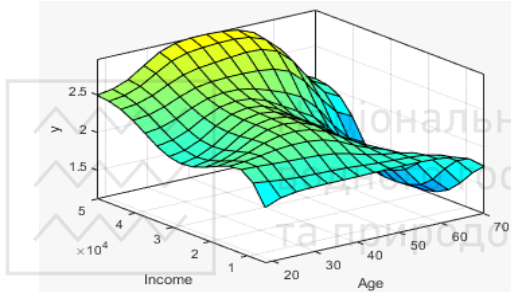


Рис. 5. Поверхня відгуку (HM-1)

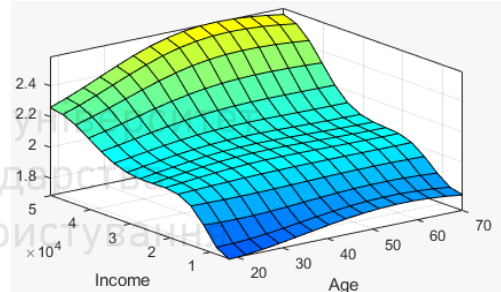


Рис. 6. Поверхня відгуку (HM-4)

Висновки: в результаті проведеного дослідження розроблено модель прийняття рішень у фінансовій сфері щодо визначення кредитоспроможності фізичних осіб-позичальників комерційних банків. Реалізація моделі передбачає максимальне усунення втручання людського фактору на прийняття рішень. Запропоновано шляхи вдосконалення моделі, що дає змогу налаштовувати модель відповідно до наявних вимог у банківському секторі. Робота носить практичний характер та представляє інтерес для використання в інформаційних системах підтримки прийняття рішень щодо оцінювання кредитоспроможності позичальників комерційних банків з метою швидкого та обґрунтованого рішення про видачу кредиту з урахуванням кількісних та якісних відомостей про потенційного позичальника.

1. Лук'янова В. В. Економічний ризик. Київ : Академвидав, 2007. 462 с. 2. Методичні вказівки з інспектування банків «Система оцінки ризиків». Постанова Правління НБУ від 15.03.2004 р. № 104. URL: <http://zakon2.rada.gov.ua/laws/show/v0104500-04>. (дата звернення: 06.05.2021). 3. Акіменко В. В., Загородній Ю. В. Проектування СППР на основі нечіткої логіки : навчально-методичний посіб. Київ : Вид-во КНУ, 2007. 94 с. 4. Матвійчук А. В. Штучний інтелект в економіці: нейронні мережі, нечітка логіка : монографія. Київ : КНЕУ, 2011. 439 с. 5. Оцінка кредитоспроможності та інвестиційної привабливості суб'єктів господарювання: монографія / Єпіфанов А. О., Дехтяр Н. А., Мельник Т. М., Школьник І. О. та ін. / за ред. доктора економічних наук А. О. Єпіфанова. Суми : УАБС НБУ, 2007. 286 с.

Babych T. Yu., Candidate of Economics (Ph.D.), Associate Professor; Yashchuk D. A., Senior Student (National University of Water and Environmental Engineering, Rivne)

NEURO-FUZZY NETWORKS IN FINANCIAL ACTIVITY DECISION-MAKING

Neuro-fuzzy models are one of the effective approaches to work with uncertain or linguistic information. In this study, it is proposed to use fuzzy logic to build a model for deciding on a loan to a potential borrower. For this purpose, the most important characteristics of the borrower are selected. A base of fuzzy rules according to the Sugeno algorithm is formed. The analysis of adequacy of the constructed model on the basis of expert data is executed. The fuzzy rules were configured with Matlab-2019b. The adequacy of the model was improved after setting fuzzy logic rules and membership function parameters. Implementation of the model involves the maximum elimination of human intervention in decision-making.

Keywords: determination of creditworthiness, fuzzy model, Sugeno algorithm, membership function, fuzzy model tuning.

УДК 004.9

Макарець¹ М. В., студентка, Мічута² О. Р, к.т.н., доцент (Національний університет водного господарства та природокористування, м. Рівне,

¹makarets_ak17@nuwm.edu.ua, ²o.r.michuta@nuwm.edu.ua)

ВИКОРИСТАННЯ СУЧАСНОГО ФРЕЙМВОРКУ FLUTTER ДЛЯ СТВОРЕННЯ МОБІЛЬНИХ ДОДАТКІВ ТА ІНТЕГРАЦІЯ З БАЗОЮ ДАНИХ FIREBASE В РЕАЛЬНОМУ ЧАСІ

Розглянуто переваги та недоліки використання Flutter. Наведено основні проблеми під час роботи з платформою. Описано покрокову інструкцію для інтеграції з базою даних Firebase в реальному часі.

Ключові слова: мобільний додаток, фреймворк, розробка, база даних, інтеграція, механізм візуалізації, плагін.

Вступ. Інформаційні технології стали незмінною та важливою складовою в сучасному світі. Під час пандемії люди вкотре в цьому переконались, адже не зважаючи на карантин, закриття шкіл, університетів, підприємств, навчання та робота продовжилась дистанційно, а все завдяки веб-ресурсам, онлайн-платформам та мобільним додаткам. Мільйони людей щодня активно користуються власними смартфонами, які стали не гіршою альтернативою тих самих комп'ютерів, телевізорів та радіо. За допомогою мобільних телефонів ми не тільки спілкуємося один з одним, але і замовляємо товари з магазинів, купуємо квитки, бронюємо житло, викликаємо таксі, використовуємо телефони як навігатори, фото- і відеокамери, соцмережі, онлайн-банки, і просто як спосіб розважитися. Саме тому актуальність розробки мобільних додатків зростає не те що з кожним роком, а й з кожним днем.

На даний час існує багато інструментів програмування, доступних розробникам для створення мобільних застосунків. Серед них і Flutter – безкоштовний фреймворк для мобільного інтерфейсу з відкритим кодом, створений та представлений у грудні 2018 року компанією Google. Він дозволяє створити власний мобільний додаток лише з однією кодовою базою, використовуючи одну мову програмування для двох різних платформ – iOS та Android [1].

Переваги використання Flutter закладені в мові програмування і комплекті засобів розробки, що дозволяють усувати проблеми і недоліки, з якими не можуть впоратися інші технології (рис. 1).

	Нагив	Конструктор	Flutter
Гнучкість та кастомізація	✓	✗	✓
Розвиток і підтримка	✓	✗	✓
Продуктивність	✓	✗	✓
Швидкість запуску	✗	✓	✓

Рис. 1. Переваги використання Flutter

Серед основних переваг Flutter – це високопродуктивний механізм візуалізації Skia та мова програмування Dart, яка компілюється у бінарний код, що значно впливає на швидкість. Через це додатки, створені на базі Flutter, містять менше змінних компонентів та мають швидку анімацію. У фреймворк вбудований також Hot-reload, який завантажує зміни коду у віртуальну машину та відновлює дерево віджетів, зберігаючи стан програми; він не повторює main () або initState () [2]. Для тих, хто хоче розібратися з новою платформою, компанія Google також розробила детальну документацію з прикладами коду та відео-уроками.

Звичайно, як і будь-яка інша платформа, Flutter має і свої недоліки. Це молода платформа, тому готових рішень для неї досить мало. На написання UI на Flutter потрібно більше часу, та й саме написання коду на Dart - більш тривале. Крім

цього платформа вимагає наявності більш глибоких навичок програмування. Також самих вакансій для розробників на Flutter поки досить мало.

Незважаючи на вищезгадані недоліки, все-таки переваг у платформи значно більше, цим пояснюється масовий перехід розробників на Flutter. Всього лише за рік дана платформа обійшла за популярністю на GitHub і Stack Overflow платформу React Native, яка була до цього найбільш використовуваною (рис. 2).

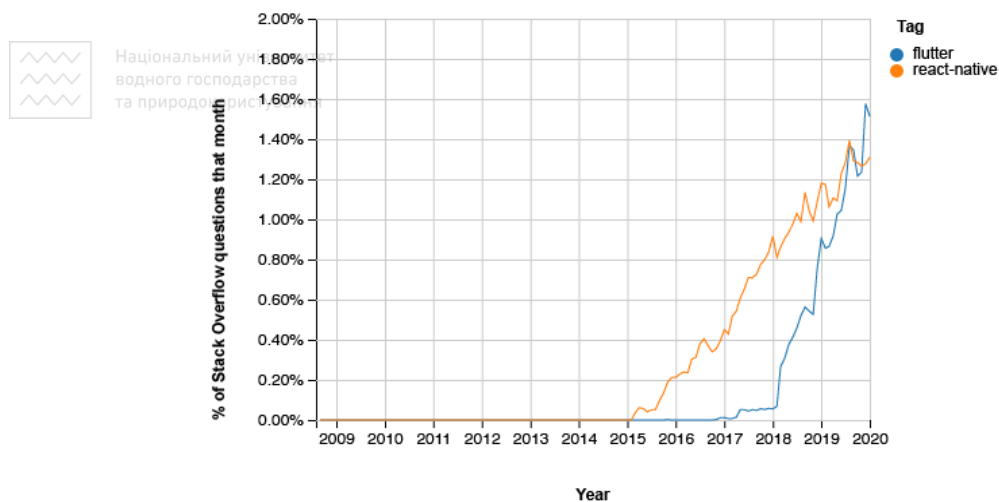


Рис. 2. Порівняння використання Flutter та React Native на Stack Overflow

Початок розробки додатку за допомогою Flutter розпочинається зі скачування SDK з офіційного сайту, додавши шлях до папки `flutter\bin` у “Змінні оточення” Path. Щоб перевірити правильність установки, у командному рядку пишемо `flutter -version`, якщо Windows розпізнав команду – Flutter налаштовано [3]. *Flutter doctor* буде наступною командою, яка перевірить всі потрібні залежності (рис.3). Після успішної установки, додатково потрібно встановити плагіни *Flutter i Dart* до редактора VSCode або Android Studio.

```
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.0.2, on Microsoft Windows [Version 10.0.19042.985], locale uk-UA)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.3)
[✓] Chrome - develop for the web
[!] Android Studio (not installed)
[✓] VS Code (version 1.56.2)
[✓] Connected device (2 available)

! Doctor found issues in 1 category.
```

Рис. 3. Виконання команди *flutter doctor*

Компанія Google дуже часто випускає нові версії flutter SDK, тому при оновленні платформи програма, яка раніше повноцінно працювала, може не компілюватись або зовсім «поламатись». Також варто звернути увагу на те, що при спробі відкрити проект у папці з назвою українськими літерами, компілятор буде видавати помилку, тому обов’язково назва папки проекту задається латинськими літерами.

За допомогою Google Firebase, однієї з найпотужніших і універсальних доступних на сьогоднішній день серверних платформ, можна легко додавати в свої мобільні додатки такі важливі функції, як управління користувачами, звіти про збій, аналітика, зберігання даних в реальному часі і навіть машинне навчання. Саме тому розглянемо, як інтегрувати Flutter з базою даних Firebase в реальному часі на прикладі сторінки авторизації [4]. Для початку потрібно створити новий проект та додати Firebase до нашого Android додатка, вказавши ім'я пакета (*applicationId* у файлі *build.gradle* на рівні програми), назву самого проекту та відбиток SHA-1 для авторизації за допомогою Google, який можна отримати ввівши в терміналі команду `keytool -list -v | -alias androiddebugkey -keystore %USERPROFILE%.android\debug.keystore`. Наступний крок – завантажити файл *google-services.json* та додати його у кореневий каталог модуля програми Android (рис. 4).

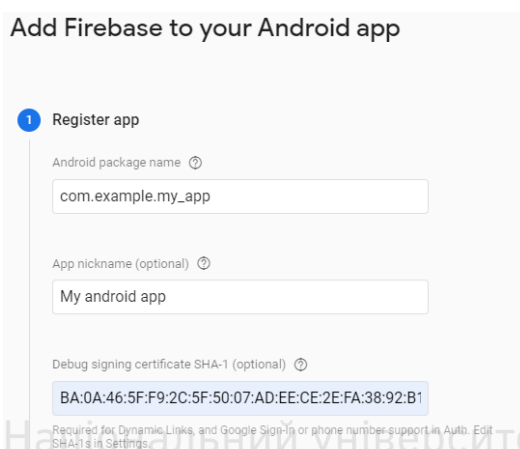
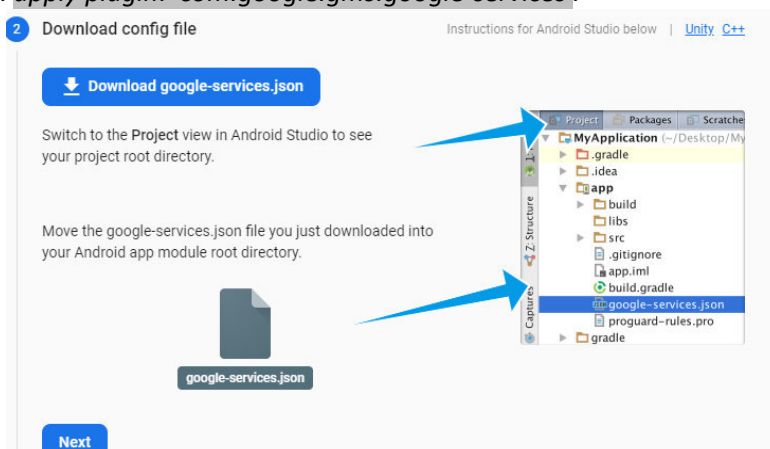


Рис. 4. Інтеграція Firebase з додатком на Android

Для того, щоб прочитати файл JSON, потрібен плагін Google Services. Тому потрібно відкрити файл *build.gradle* кореневого рівня і додати шлях до класу плагіна в розділі *dependencies*: `classpath 'com.google.gms:google-services:4.0.1'`. Після цього активувати плагін, додавши наступний рядок в кінець файлу на рівні додатку *build.gradle*: `apply plugin: 'com.google.gms.google-services'`.



FlutterFire має незалежні плагіни Google Flutter для більшості сервісів Firebase. Щоб додати плагіни в якості залежностей у свій проект Flutter, потрібно згадати їх в ключі *dependencies* файлу *pubspec.yaml*. Для сторінки авторизації потрібні наступні плагіни: *firebase_auth* (для використання API автентифікації), *google_sign_in* (для авторизації за допомогою Google), *firebase_core* (для використання Core API, що дозволяє підключатися до декількох додатків Firebase) (рис. 5). Знайти інші плагіни, дізнатись версію та переглянути приклади використання можна за посиланням: <https://pub.dev/>.

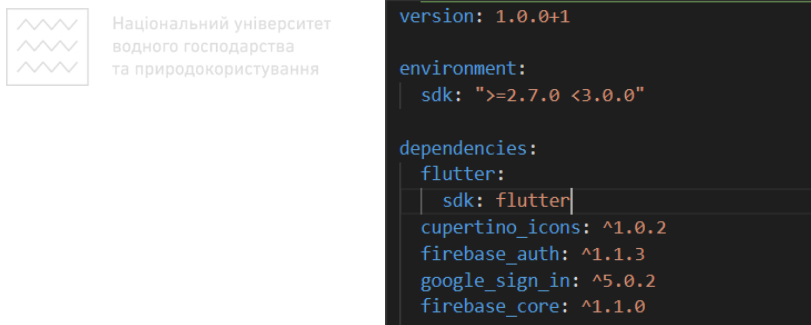


Рис. 5. Плагіни для сторінки авторизації

Для створення екземплярів *FirebaseAuth* та *GoogleSignIn* потрібно імпортувати пакети раніше завантажених плагінів (рис. 6).

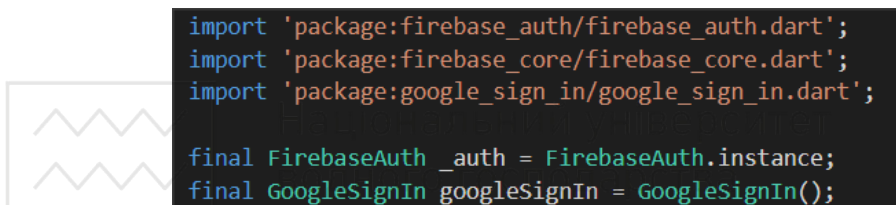


Рис. 6. Створення екземплярів *FirebaseAuth* та *GoogleSignIn*

Зареєстрованих користувачів можна переглянути в розділі *Authentication* на вкладці *Users* (рис. 7).

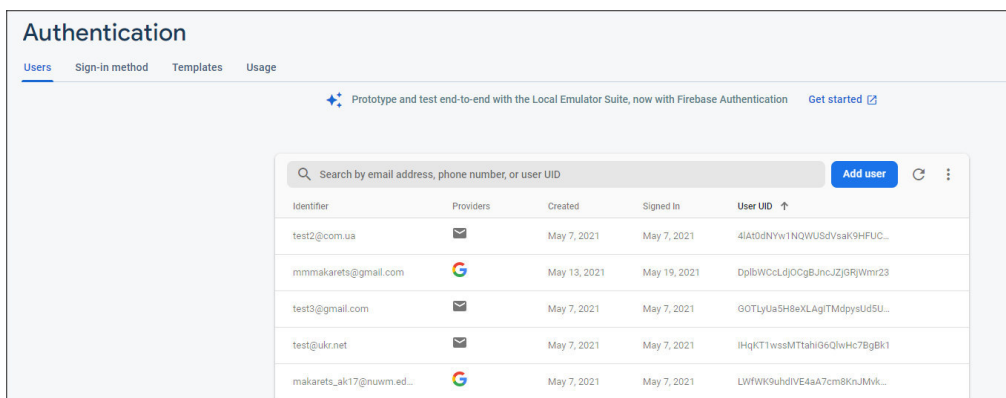


Рис. 7. Зареєстровані користувачі

Висновки. У статті наведено основні аспекти роботи з новою платформою для розробки мобільних додатків Flutter та розглянуто на прикладі сторінки авторизації особливості інтеграції з Firebase.

1. Flutter. URL: <https://flutter.dev/> (дата звернення: 23.05.2021). 2. Flutter. Docs. Hot reload. URL: <https://flutter.dev/docs/development/tools/hot-reload> (дата звернення: 23.05.2021). 3. Введение в Flutter. Что такое Flutter. Установка. URL: <https://metanit.com/dart/flutter/1.1.php> (дата звернення: 23.05.2021). 4. Firebase. URL: <https://firebase.google.com/docs/flutter/setup> (дата звернення: 23.05.2021).



водного господарства
та природокористування

Makarets M. V., Senior Student; Michuta O. R., Candidate of Engineering (Ph.D.), Associate Professor (National University of Water and Environmental Engineering, Rivne)

USING THE MODERN FLUTTER FRAMEWORK TO CREATE MOBILE APPLICATIONS AND INTEGRATION WITH THE FIREBASE DATABASE IN REAL TIME

The advantages and disadvantages of using Flutter are considered. The main problems when working with the platform are given. Describes step-by-step instructions for integrating with a real-time Firebase database.

Keywords: mobile application, framework, development, database, integration, visualization mechanism, plug-in.

УДК 378:004

Рощенюк¹ А. М., к.пед.н., доцент кафедри комп'ютерних наук та прикладної математики, керівник студентського наукового гуртка «Інформаційні технології в середній та вищих школах»; Юревич² І. О., студентка (Національний університет водного господарства та природокористування, м. Рівне,

¹e-mail: a.m.roshcheniuk@nuwm.edu.ua, ²e-mail: yurevych_ak18@nuwm.edu.ua)

ДИСТАНЦІЙНЕ НАВЧАННЯ У ВИЩІЙ ШКОЛІ

У статті розглянуто сутність поняття «дистанційне навчання» як сучасна освітня технологія. Визначено його сучасний стан, основні переваги, недоліки та перспективи розвитку.

Ключові слова: інформаційні технології, дистанційне навчання, освітня платформа, форма навчання.

В умовах карантину всім навчальним закладам не тільки України, а й світу, довелось перейти на такий вид навчання як дистанційне. Під цим терміном «дистанційне навчання» розуміється індивідуалізований процес набуття знань, умінь, навичок і способів пізнавальної діяльності людини, який відбувається в основному за опосередкованої взаємодії віддалених один від одного учасників

навчального процесу у спеціалізованому середовищі, яке функціонує на базі сучасних психолого-педагогічних та інформаційно-комунікаційних технологій [4].

Історично, дистанційне навчання, бере свій початок з «кореспондентського навчання», саме коли, Ісаак Пітман запропонував навчання через поштовий зв'язок, у 1840 році, для студентів Англії. З кожним роком популярність дистанційного навчання у багатьох країнах світу стрімко зростає.

Наприклад, у США та Канаді як альтернативу традиційному навчанню створено віртуальні університети, де кожен студент може отримати освіту за основними дистанційними курсами на базі будь якого університету. В Європі створено відкриті університети дистанційної освіти, група навчальних закладів, які реалізують дистанційні програми. Методики такого навчання передбачають застосування нових інформаційних технологій, які включають супутникове телебачення, комп'ютерні мережі, мультимедіа тощо.

Зараз, у всьому світі спостерігається збільшення числа студентів, що навчаються в режимі віддаленого доступу, зростає кількість вищих навчальних закладів, що використовують дистанційні технології в навчальному процесі; створюється велика кількість міжнародних освітніх структур і т.д. Наприкінці 1997 року в 107 країнах діяло 986 навчальних закладів дистанційного типу. Число тих, хто здобував вищу освіту в системі віддаленого доступу, тобто дистанційного навчання, становило 50 мільйонів чоловік; у 2000 році - кількості студентів збільшилась до 90 мільйонів, а 2023 році – до 120 мільйонів чоловік, причому ці дані стосуються лише навчальних закладів, що мають державну ліцензію на право освітньої діяльності [6].

Тож, більшість університетів поруч із класичними формами навчання, денною та очною, впроваджує й дистанційну, оскільки наявність цієї форми може залучити більшу кількість абітурієнтів не тільки з України, а й з закордону, а також підвищить рейтинг навчального закладу. Для студентів же, дистанційне навчання значно збільшує можливості отримання знань, мінімізуючи необхідність присутності в університеті, витрачений час і гроші, завдяки чому стати дипломованим фахівцем під силу будь-кому, хто має бажання вчитися.

Головним принципом дистанційної форми навчання є використання віртуального простору для взаємодії викладача зі студентом. При цьому в основі отримання освіти дистанційно лежить не стовідсотково самостійна робота, а досить тісний контакт і спілкування з викладачем в потрібному для розуміння матеріалу обсязі.

Технології дистанційного навчання складаються з педагогічних та інформаційних технологій дистанційного навчання [3].

Педагогічні технології дистанційного навчання – це технології опосередкованого активного спілкування викладачів зі студентами з використанням телекомунікаційного зв'язку та методології індивідуальної роботи студентів із структурованим навчальним матеріалом, представленим в електронному вигляді.

Інформаційні технології дистанційного навчання – це технології створення, передачі і збереження навчальних матеріалів, організації і супроводу навчального процесу дистанційного навчання за допомогою телекомунікаційного зв'язку.

Організаційна структура навчального процесу за дистанційною освітою ґрунтується на таких компонентах: організаційно-управлінському; нормативно-правовому; навчально-методичному; інформаційно-телекомунікаційному; економічно-фінансовому.

У дистанційному навчанні можна виділити три основні технології:

- кейс-технологія, за якої навчально-методичні матеріали чітко структуровані і відповідно комплектуються у спеціальний набір (кейс). Потім вони пересилаються студентові для самостійного навчання з періодичними консультаціями у викладачів-консультантів (т'юторів) у створених для цього віддалених (регіональних) навчальних центрах або пунктах;

- TV-технологія, що ґрунтується на використанні телевізійних лекцій з консультаціями викладачів;

- мережна технологія, яка базується на використанні мережі «Інтернет» для забезпечення студентів навчально-методичним матеріалом, і інтерактивної взаємодії між викладачем і студентом.

Якщо проаналізувати хронологію етапів запровадження названих технологій, то слід відзначити, що спочатку була створена кейс-технологія, де кожний навчальний курс забезпечувався спеціальним посібником, програмою та методичними вказівками для самостійного навчання. Модульний принцип формування навчальних планів дав можливість з успіхом об'єднувати потреби та можливості студентів з потребами освітніх стандартів. Застосування TV-технологій поширилося пізніше і нині використовується в багатьох країнах. Однак в Україні серйозних досліджень у цьому напрямі останніми роками практично не проводилося. *Найбільш сучасною і одночасно перспективною технологією дистанційного навчання є Інтернет-технологія.* Це комплекс освітніх послуг, які надаються студентам за допомогою спеціалізованого інформаційно-навчального середовища. В основі цієї технології лежить методологія, яка спрямована на індивідуальну (що не залежить від часу та місця) роботу студентів, із спеціально структурованим навчальним матеріалом за різноманітними формами спілкування із віддаленими експертами, викладачами чи студентами.

Для організації дистанційного навчання у навчальному закладі повинна бути створена сучасна комп'ютерна мережа з локальним сервером для забезпечення можливостей створювати мультимедійні фрагменти дистанційних навчальних курсів та організовувати інтерактивне спілкування викладачів і студентів за допомогою відео конференцій навчальний заклад може обладнати спеціальну аудиторію, в якій буде знаходитися потужний мультимедійний комп'ютер під'єднаний до мережі Інтернет або до локального сервера.

Основними перевагами дистанційного навчання є [3]:

- гнучкість (студенти можуть не відвідувати регулярні заняття, а навчаються у зручній для себе час та в зручному місці);

- модульність (система дидактичних модулів дає студентам можливість сформувати навчальну програму, що відповідає індивідуальним або груповим потребам);

- паралельність (навчання може здійснюватися одночасно з професійною діяльністю або з навчанням за іншим напрямом, тобто без відриву від виробництва

або іншого виду діяльності);

- можливість одночасного спілкування великої кількості студентів (за допомогою телекомунікаційного зв'язку та мережі Інтернет студенти і викладачі мають можливість постійного спілкування);

- економічність (ефективне використання навчальних площ та технічних засобів, концентроване представлення інформації, використання комп'ютерних мереж призводить в цілому до зменшення витрат на підготовку фахівців);

- технологічність (у процесі дистанційного навчання набувають розвитку сучасні навчальні та інформаційні технології, що сприяє входженню людини у світовий інформаційний простір);

- доступність освіти (студенти мають рівні можливості одержання освіти незалежно від місця проживання, стану здоров'я і соціального статусу);

- інтернаціональність (система дистанційної освіти надає можливість студентам одержати освіту у навчальних закладах іноземних держав, не виїжджаючи зі своєї країни; також надавати освітні послуги іноземним громадянам і співвітчизникам, що проживають за кордоном);

- нова роль викладача (дистанційне навчання розширює і оновлює роль викладача, робить його наставником-консультантом, який повинен керувати навчальним процесом, впливати на творчу активність студентів відповідно сучасних педагогічних інновацій);

- позитивний вплив на студента (за умов нових форм діяльності студентів таких, як самоуправління, самоорганізація, авто-дидактика, рефлексія, використання сучасних інформаційних технологій та ін., підвищується творчий та інтелектуальний потенціал людини);

- підвищення якості освітнього процесу (для організації дистанційного навчання використовуються найсучасніші навчально-методичні засоби навчання і контролю, які відповідають найвищим освітнім стандартам).

Якщо розглянути особливості дистанційної освіти з погляду комунікацій між викладачем і студентом, то можна визначити такі її характерні риси:

- самоосвіта як основа дистанційного навчання, що передбачає самомотивацію студента щодо власного навчання, а також певний рівень самоорганізації особистості;

- спілкування викладача і слухача за принципом «один до одного», що відповідає за формою і змістом індивідуальній консультації;

- спілкування і взаємодія «один до одного» не виключає взаємодії «одного до багатьох», оскільки викладач, відповідно до заздалегідь складеного графіка, працює відразу з безліччю студентів. Така форма взаємодії нагадує традиційне навчання;

- взаємодія «багатьох до багатьох» означає, що можливе одночасне спілкування безлічі студентів, які обмінюються між собою досвідом і враженнями.

Однак, незважаючи на переваги, сучасний стан розвитку дистанційного навчання в нашій країні бажає кращого, оскільки не відповідає вимогам суспільства у якому мільйони громадян задовольняють свої освітньо-інформаційні потреби через телекомунікаційні мережі, у тому числі через Інтернет. Аналіз стану дистанційного

навчання в Україні показує, що всі навчальні заклади, організації й установи, які впроваджують або використовують технології дистанційного навчання, стикаються з реальними труднощами [2, с. 56], серед яких нестача професійних кадрів, технічного та фінансового забезпечення.

Що стосується НУВГП, то вимушене дистанційне навчання організоване за допомогою навчальної платформи Moodle, надає широкий набір функціональності як для викладачів, так і для студентів. Дана платформа орієнтована насамперед на організацію взаємодії між викладачем та студентами, хоча підходить для організації традиційних, дистанційних курсів та підтримки очного навчання [5]. Однак, дистанційної форми навчання, як основної, в університеті немає.

На нашу думку, було б цікаво створювати відкриті для загальності онлайн-курси на інших навчальних платформах. Для прикладу, на освітній платформі Stepik можна опублікувати як безкоштовний, так і платний курс. Змістом та темою курсу може бути, наприклад, певна дисципліна, яка є вибірковою, але на яку, з тих чи інших причин не набирається необхідна кількість студентів. Також у студентів університету буде хороша можливість обирати кілька вибіркових дисциплін, які вони не мають змоги вибрати під час традиційної форми навчання. Щодо тих, хто не навчається в університеті, їм може бути запропонований платний курс, це позитивно вплине на бюджет навчального закладу і підвищить рейтинг освіти у ньому. А також, онлайн-курси можуть дати університету можливість залучити більшу кількість потенційних студентів.

Дистанційний курс розширює також можливості здійснення самоконтролю знань за рахунок проходження тестування після закінчення кожного модуля у тренувальному режимі, коли студент не тільки може дізнатись, яка відповідь є невірною, але й побачити відповідь правильну.

Ще однією цікавою альтернативою є створення університетського каналу на Youtube, куди будуть викладатись лекції з різних дисциплін. Це дасть змогу вчитися і дізнаватися про університет не тільки студентам з України, але й з інших країн Європи, хоча звісно лекції, в такому випадку, мають викладатися англійською мовою, а також зробити навчання в НУВГП бажанішим серед іноземців. Щоб не впасть рівень традиційної форми навчання можна на каналі викладати скорочений курс, а в разі зацікавленості переглядача у ньому, запрошувати на навчання в університет. Так не постраждає ні дистанційна, ні традиційні форми навчання.

Дистанційне навчання – це також хороше рішення для людей із обмеженими фізичними можливостями. Вони можуть не мати можливості вчитися традиційно, тому надають перевагу навчанню онлайн.

Висновок. Впровадження дистанційної форми навчання у вищому закладі з використанням інформаційно-комунікативних технологій є дійсно хорошою ідеєю, оскільки це є тією формою навчання, яка надалі буде розвиватись більше, ніж традиційні форми навчання, а отже вона є більш перспективною. Застосування сучасних комп'ютерних і телекомунікаційних технологій в навчальному процесі не тільки створює умови для більш ефективної самостійної роботи студентів, сприяє індивідуалізації процесу підготовки фахівців, а і суттєво змінює форми і зміст комунікацій між викладачем і студентом. Також, дистанційне навчання дозволяє більш ефективно організувати самостійну роботу студентів завдяки збільшенню

джерельної бази навчального матеріалу, кращого його унаочнення, можливості працювати у зручній для студента час, обирати власну траєкторію навчання, індивідуалізуючи й диференціюючи процес навчання та здійснюючи самоконтроль засвоєння знань.

1. Дистанційна освіта: забезпечення доступності та неперервної освіти впродовж життя (e-learning and university education-2017). *Матеріали XLII Міжнародної науково-методичної конференції*. Полтава : ПУЕТ, 2017. 365 с.
2. Дистанційне навчання як сучасна освітня технологія : матеріали міжвузівського вебінару (м. Вінниця, 31 березня 2017 р.) / відп. ред. Л. Б. Ліщинська. Вінниця : ВТЕІ КНТЕУ, 2017. 102 с.
3. Концепція розвитку дистанційної освіти в Україні від 20 грудня 2000 року. URL: <http://uiite.kpi.ua/2019/06/03/1598/>. (дата звернення: 06.05.2021).
4. Положення про дистанційне навчання. URL: <https://zakon.rada.gov.ua/laws/show/z0703-13>. (дата звернення: 06.05.2021).
5. Сторінка навчальної платформи Moodle. URL: <https://uk.wikipedia.org/wiki/Moodle>. (дата звернення: 06.05.2021).
6. Татарчук Г. М. Институціоналізація дистанційного об'учення: соціологічний аспект. URL: <https://www.dissercat.com/content/institutsionalizatsiya-distantcionnogo-obucheniya-sotsiologicheskii-aspekt>. (дата звернення: 06.05.2021).

Roshcheniuk A. M., Candidate of Pedagogic Sciences (Ph.D.), Associate Professor, head of student scientific circle "information technologies in secondary and high schools";
Yurevych I. O., Senior Student (National University of Water and Environmental Engineering, Rivne)

DISTANCE LEARNING IN HIGHER SCHOOL

This article considers distance learning as a modern educational technology. Its current state and main advantages, disadvantages and prospects of development are determined.

Keywords: information technologies, distance learning, educational platform, form of education.

Наукове видання



Національний університет
водного господарства
та природокористування

ВІСНИК
НАВЧАЛЬНО-НАУКОВОГО ІНСТИТУТУ АВТОМАТИКИ, КІБЕРНЕТИКИ ТА
ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ НУВГП

Випуск 8

Технічний редактор

*Галина Сімчук
Оксана Прищепя*

Друкується в авторській редакції



Підписано до друку 28.05.2021 р. Формат 70×100¹/₁₆.

Ум.-друк. арк. 9,9. Обл.-вид. арк. 10,9.

Тираж 100 прим. Зам. № 5570.

Видавець і виготовлювач
Національний університет
водного господарства та природокористування
вул. Соборна, 11, м. Рівне, 33028.

Свідоцтво про внесення суб'єкта видавничої справи до державного
реєстру видавців, виготівників і розповсюджувачів
видавничої продукції РВ № 31 від 26.04.2005 р.