

Національний технічний університет  
«Дніпровська політехніка»

# **INFORMATION TECHNOLOGY: COMPUTER SCIENCE, SOFTWARE ENGINEERING AND CYBER SECURITY**

Випуск 4



Видавничий дім  
«Гельветика»  
2023

## РЕДАКЦІЙНА КОЛЕГІЯ:

### ГОЛОВНИЙ РЕДАКТОР:

**Удовик Ірина Михайлівна**, кандидат технічних наук, доцент, декан факультету інформаційних технологій, Національний технічний університет «Дніпровська політехніка», Україна.

### ЧЛЕНИ РЕДАКЦІЙНОЇ КОЛЕГІЇ:

**Алексєв Михайло Олександрович**, доктор технічних наук, професор, завідувач кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», Україна;

**Бердник Михайло Геннадійович**, доктор технічних наук, доцент, професор кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», Україна;

**Кабак Леонід Віталійович**, кандидат технічних наук, доцент, доцент кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», Україна;

**Корнієнко Валерій Іванович**, доктор технічних наук, професор, завідувач кафедри безпеки інформації та телекомунікацій, Національний технічний університет «Дніпровська політехніка», Україна;

**Корченко Анна Олександрівна**, доктор технічних наук, доцент, професор кафедри безпеки інформаційних технологій, Національний технічний університет «Дніпровська політехніка», Україна;

**Лактіонов Іван Сергійович**, доктор технічних наук, доцент, професор кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», Україна;

**Литвин Василь Володимирович**, доктор технічних наук, професор, завідувач кафедри інформаційних систем та мереж, Національний університет «Львівська політехніка», Україна;

**Любченко Віра Вікторівна**, доктор технічних наук, професор, професор кафедри системного програмного забезпечення, Одеський національний політехнічний університет, Україна;

**Маттіас Реч**, Ph.D, професор, кафедра мехатроніки, Університет Ройтлінгену, Німеччина;

**Молоканова Валентина Михайлівна**, доктор технічних наук, професор, професор кафедри системного аналізу та управління, Національний технічний університет «Дніпровська політехніка», Україна;

**Мороз Борис Іванович**, доктор технічних наук, професор, професор кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», Україна;

**Мулеса Оксана Юріївна**, доктор технічних наук, професор, професор кафедри інформаційних систем та мереж, Ужгородський національний університет, Україна;

**Рак Тарас Євгенович**, доктор технічних наук, доцент, професор кафедри інформаційних технологій, ПЗВО «ІТ СТЕП Університет», Україна;

**Савенко Олег Станіславович**, доктор технічних наук, професор, професор кафедри комп'ютерної інженерії та інформаційних систем, декан факультету інформаційних технологій, Хмельницький національний університет, Україна;

**Семенов Сергій Геннадійович**, доктор технічних наук, професор, професор кафедри кібербезпеки та інформаційних технологій, Харківський національний економічний університет імені Семена Кузнеця, Україна;

**Сироткіна Олена Ігорівна**, кандидат технічних наук, доцент, доцент школи комп'ютерних наук, Віндзорський університет, Канада;

**Швачич Геннадій Григорович**, доктор технічних наук, професор, професор кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», Україна.

Журнал включений до Переліку наукових фахових видань України (категорія «Б») за спеціальностями 121 – Інженерія програмного забезпечення, 122 – Комп'ютерні науки, 123 – Комп'ютерна інженерія, 124 – Системний аналіз, 125 – Кібербезпека відповідно до наказів МОН України № 735 (додаток № 4) від 29.06.2021 р. та № 491 (додаток № 3) від 27.04.2023 р.

Науковий журнал «Information Technology: Computer Science, Software Engineering and Cyber Security» зареєстрований Міністерством юстиції України (свідоцтво про державну реєстрацію друкованого засобу масової інформації серія KB № 24879-14819P від 17.06.2021 р.)

Офіційний сайт видання: [www.journals.politehnica.dp.ua/index.php/it](http://www.journals.politehnica.dp.ua/index.php/it)

Статті у виданні перевірені на наявність плагіату за допомогою програмного забезпечення StrikePlagiarism.com від польської компанії Plagiat.pl.

ISSN 2786-507X (Print)  
ISSN 2786-5088 (Online)

© Національний технічний університет «Дніпровська політехніка», 2023

УДК 004.894:004.023

DOI <https://doi.org/10.32782/IT/2023-4-1>

### **Ілля ЗІБОРОВ**

аспірант кафедри системного аналізу та управління Національного технічного університету «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, [ziborov.il.k@ntmu.one](mailto:ziborov.il.k@ntmu.one)  
ORCID: 0000-0002-3118-387X

### **Тімур ЖЕЛДАК**

Кандидат технічних наук, доцент, завідувач кафедри системного аналізу та управління Національного технічного університету «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, [zheldak.t.a@ntmu.one](mailto:zheldak.t.a@ntmu.one)  
ORCID: 0000-0002-4728-5889  
Scopus Author ID: 55602208300

**Бібліографічний опис статті:** Зіборов, І., Желдак, Т. (2023). Еволюційний метод пошукової оптимізації на основі рою часток та моделювання штучних імунних систем. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 3–12, doi: <https://doi.org/10.32782/IT/2023-4-1>

## **ЕВОЛЮЦІЙНИЙ МЕТОД ПОШУКОВОЇ ОПТИМІЗАЦІЇ НА ОСНОВІ РОЮ ЧАСТОК ТА МОДЕЛЮВАННЯ ШТУЧНИХ ІМУННИХ СИСТЕМ**

Запропоновано метод оптимізації неперервних функцій багатьох змінних, який втілює підходи методів рою часток та моделювання штучної імунної системи. Проведено дослідження з метою визначення оптимальних налаштувань алгоритму, що реалізує запропонований метод. Показано, що запропонований метод може бути застосований як складова інформаційної технології супроводу підтримки управлінських рішень та розв'язання задач оптимізації технологічних процесів у металургійному виробництві.

**Метою роботи** є розробка оригінального гібридного методу глобальної оптимізації на основі методу штучної імунної системи та методу рою частинок, а також формулювання рекомендацій щодо налаштування його параметрів. Запропонований метод має прискорити прийняття рішень при керуванні технологічними процесами у металургійному виробництві та підвищити їх точність.

**Методологія** забезпечення рішення полягає у поєднанні основних пошукових операторів еволюційних методів оптимізації на основі рою часток і моделювання штучної імунної системи людини. Використовуються традиційні для роювого пошуку кроки з пошуку рішення та обміну інформацією про знайдені локальні оптимуми. Такий пошук доповнюється принципом змагальності, запозиченим у методі штучної імунної системи, для чого популяція поділяється на менші рої або пошукові команди. Також запропоновані кроки контролю різноманіття в рої та радіусу розсіювання часток. Останнє дозволяє вирішувати задачі як безумовної, так і умовної оптимізації у неперервному просторі високої розмірності.

**Наукова новизна** отриманих у роботі результатів полягає в формулюванні нового еволюційного методу оптимізації в неперервному просторі на основі роювого інтелекту, який на відміну від відомих раніше роювих методів використовує принцип змагання підгруп рою та оператор стиснення популяції, властивих методу моделювання штучних імунних систем. Емпірично встановленні параметри методу, що визначають його ефективність.

**Висновки.** Застосування запропонованого еволюційного методу пошукової оптимізації до мінімізації тестових функцій у неперервному просторі розмірністю до 20 показало його ефективність у порівнянні з відомими раніше. В подальшому актуальним бачиться застосування викладеного алгоритму для розв'язання задач оптимізації технологічних процесів у металургійному виробництві. Зокрема, даний алгоритм вбачається ефективним для задач шихтування, оптимізації використання феросплавів у ливарному виробництві, а також при прогнозуванні механічних властивостей готової продукції.

**Ключові слова:** еволюційний метод, оптимізація, штучні імунні системи, роювий інтелект, популяція, стиснення, технологія.

### **Illia ZIBOROV**

Postgraduate student of the System Analysis and Control Department, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave, Dnipro, Ukraine, 49005, [ziborov.il.k@ntmu.one](mailto:ziborov.il.k@ntmu.one)  
ORCID: 0000-0002-3118-387X

**Timur ZHELDAK**

Candidate of Technical Sciences, Associate Professor, Head of the System Analysis and Control Department, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave, Dnipro, Ukraine, 49005, zheldak.t.a@nmu.one

ORCID: 0000-0002-4728-5889

Scopus Author ID: 55602208300

**To cite this article:** Ziborov, I., Zheldak, T. (2023). Evoliutsiynyi metod poshukovoi optymizatsii na osnovi roiu chastok ta modeliuvannia shtuchnykh imunnykh system. [The evolutionary method based on particle swarm optimization and artificial immune systems modelling]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 3–12, doi: <https://doi.org/10.32782/IT/2023-4-1>

## THE EVOLUTIONARY METHOD BASED ON PARTICLE SWARM OPTIMIZATION AND ARTIFICIAL IMMUNE SYSTEMS MODELLING

*An optimization method, which embodies the approaches of particle swarm methods and artificial immune system modeling, for a random multidimensional continuous function is proposed. Research was conducted to determine the optimal settings of the algorithm that implements the proposed method. The proposed method can be applied as a component of information technology to support management decisions and solve technological processes optimizing problems in metallurgical production.*

**The purpose** of the work is to develop an original hybrid method of global optimization based on the artificial immune system method and the particle swarm method, as well as to formulate recommendations for setting its parameters. The proposed algorithm should speed up decision-making in the management of technological processes of metallurgical production and increase their accuracy.

**The methodology** of providing a solution consists in the combination of basic search operators of evolutionary optimization methods based on the particle swarms and the human artificial immune system modeling. Traditional swarm search steps of finding a solution and exchanging information about found local optima are used. The search method is supplemented by the principle of competition, borrowed from the method of the artificial immune system, for which the population is divided into smaller swarms or search teams. Steps to control the diversity in the swarm and the particle scattering radius are also proposed. This hybridization allows solving both unconditional and conditional optimization problems in a continuous high-dimensional space.

**The scientific novelty** of the results obtained in the work is that a new evolutionary optimization method in a continuous space based on swarm intelligence is proposed. This method, in contrast to the previously known swarm optimization methods, uses the principle of swarm subgroup competition and the population compression operator, which are characteristic of the method of modeling artificial immune systems. Also, the parameters of the method that maximize its effectiveness are empirically determined.

**Conclusions.** The application of the proposed evolutionary search optimization method to the minimization of test functions in a continuous space with up to 20 dimensions showed its effectiveness in comparison with previously known ones. The application of the described algorithm for solving the optimization problems of technological processes in metallurgical production is considered relevant in further. This method is considered effective for smelting charging task, for using of ferroalloys in foundry production optimization, as well as for predicting the mechanical properties of finished melting products.

**Key words:** evolutionary method, optimization, artificial immune systems, swarm intelligence, population, compression, technology.

**Актуальність проблеми.** Інформаційна технологія супроводу інтелектуальної підтримки процесів прийняття рішень в багатоетапному прокатному виробництві передбачає вирішення цілого ряду задач відносно окремих технологічних процесів, в тому числі задач прогнозування, оптимізації, формування рекомендацій та відновлення математичних залежностей. Значна частина цих задач, зокрема оптимізація шихтування, оптимізація використання феросплавів та побудова оптимальних моделей прогнозування механічних властивостей готової продукції мають багатокритеріальний та багатоекстремальний характер.

Для розв'язання таких типів задач в складі комплексного програмного забезпечення супро-

воду підтримки прийняття рішень на виробництві сортового прокату необхідно використувати надійний метод, який би забезпечував повторюваний пошук екстремуму за гарантований час.

**Аналіз останніх досліджень і публікацій.** Подібна задача вирішується в дійсному просторі має велику кількість обмежень (Богушевський, 2010) та може розглядатися як багатокритеріальна, де разом з критерієм мінімуму собівартості готової сталі можуть вживатися одразу кілька технологічних (Желдак, 2010). Для вирішення такого класу задач також існують як точні методи, так і наближені чисельні методи (Levitin, 2012). Серед точних методів

вирішення даного класу задач найбільш вживані методи множників Лагранжа та методи, засновані на похідних (Гончаренко, 2010)

Застосування точних методів в межах інформаційної технології, що розглядається, обмежене ресурсами часу і природою самої задачі. Адже при цілочисельності ряду змінних розв'язання задачі оптимізації без її врахування може привести до результатів, далеких від оптимуму (Жалдак, 2005). Водночас для застосування точних методів необхідно приведення задачі до однокритеріальної, що в свою чергу передбачає попереднє вирішення задач на пошук мінімуму і максимуму за кожним критерієм і виконання згортки, що неприпустимо збільшує складність задачі (Сергієнко, 2001). Відтак, для розв'язання задачі нелінійної багатовимірної умовної оптимізації надалі пропонується використовувати еволюційні багатоагентні методи, багато з яких розглянуті в (Субботін, 2009), зокрема генетичний алгоритм та еволюційна стратегія.

Серед великого класу еволюційних числових методів оптимізації до розв'язання розглянутих задач може бути застосований метод моделювання відпалу (англ. Simulated annealing, SA) (Das, 2005), який однаково добре показує себе при розв'язанні і безперервних задач нелінійної оптимізації та водночас при розв'язанні комбінаторних задач – як двійкових, так і задач на перестановках. Іншим ефективним методом оптимізації для дійсного простору є метод рою часток (англ. Particle Swarm Optimization, PSO) (Bratton, 2007), ефективний при вирішенні багатоцільових задач. Однак, ці методи передбачають глобальну безумовну оптимізацію і доволі складний механізм врахування обмежень, а також мають значну кількість параметрів, що мають встановлюватись емпірично і дуже суттєво впливають на ефективність пошуку рішення (Karafotis, 2015).

Зокрема в методі PSO у кожного пошукового агента є додаткова характеристика: крім координати  $X(t)$  як поточного розв'язку в момент часу  $t$ , пошуковий агент має вектор швидкості  $V(t)$  як напрямок майбутнього переміщення на наступній ітерації. Нова точка (рішення задачі) знаходяться як попередня з додаванням вектора швидкості:

$$X(t+1) = X(t) + V(t). \quad (1)$$

При цьому вважається, що кожен пошуковий агент знає:

– свої поточне положення та значення цільової функції в цьому положенні:  $X(t)$ ,  $F(X(t))$ ;

– найкраще з рішень, знайдених ним за час пошуку:  $X_b$ ,  $F(X_b)$ ;

– найкраще рішення, знайдене всім роєм за час пошуку:  $X_g$ ,  $F(X_g)$ .

Суть алгоритму полягає в тому, щоб на кожному кроці  $t$  для кожного елемента рою  $i = 1 \dots n$  отримати  $V_i(t)$ :

$$V_i(t) = a \cdot V_i(t-1) + b \cdot (X_b(t) - X_i(t)) + c \cdot (X_g(t) - X_i(t)), \quad (2)$$

де  $a, b, c$  – так звані соціально-когнітивні довірчі коефіцієнти (можуть змінюватися в процесі роботи алгоритму, при цьому  $a$  відповідає за збереження руху і називається спеціалізацією,  $b$  відповідає за довіру собі й зветься ностальгією, а  $c$  називається ройовою мораллю і відповідає за схильність індивідів довіряти один одному.

Отримане за (2) значення швидкості дозволяє кожному пошуковому агенту здійснювати перехід за (1).

Метод PSO часто застосовується в складі автоматичних та автоматизованих виробничих систем в якості методу пошуку екстремуму певних функцій через його ефективність у пошуку глобального оптимуму в багатовимірному дійсному просторі та відносну простоту. Також метод часто застосовується у якості базового при побудові багатьох гібридизацій.

Інший великий клас ітераційних методів чисельної оптимізації, що мають назву штучні імунні системи (ШИС) використовує імітацію властивості природної імунної системи і засновані на принципах соматичної теорії (Lucinska, 2009) та мережевої гіпотези (Rowan, 1990). Соматична теорія стверджує, що збільшення різноманітності антитіл відбувається за рахунок соматичної рекомбінації і мутації генів. В рамках мережевої гіпотези обґрунтовується припущення, згідно з яким контроль розмноження клонів здійснюється в результаті взаємного розпізнавання антитіл, що функціонують як єдина мережа (Lucinska, 2009; Bernardino, 2009).

Класичний алгоритм оптимізації, оснований на моделюванні імунної системи людини, спирається на традиційні еволюційні оператори:

– Генерація випадкових рішень в області пошуку;

– Оцінка пристосованості за цільовою функцією, яка передбачає урахування обмежень;

– Клонування кращих представників покоління в кількості пропорційній пристосованості;

– Випадкова мутація клонів зворотно пропорційна пристосованості;

– Оновлення популяції на основі пристосованості з доповненням випадковими особинами;

– Стискання популяції з урахуванням необхідного рівня відмінності між особинами.

Саме останній крок є особливістю алгоритму, що запобігає його передчасній збіжності.

В численних сучасних дослідженнях наголошується ефективність підходу на основі ШІС до розв'язання багатокритеріальних задач оптимізації у дійсному та двійковому просторах (Navarro-Caceres, 2018; Fernandez-Marquez, 2013; Li, 2022). Також обчислювальний метод, заснований на ШІС, добре сполучається з іншими обчислювальними методами локальної та глобальної оптимізації (Lin, 2019; Qi, 2018; Yildiz, 2009).

Авторами раніше був запропонований алгоритм розв'язання задач комбінаторної та безперервної умовної оптимізації, названий HISF, оснований на моделюванні штучних імунних систем (Zheldak, 2021). Втім, застосування даного алгоритму до широкого кола задач показало ряд проблем, пов'язаних з неможливістю однозначного вибору керуючих параметрів алгоритму та його велику обчислювальну ресурсоемність для просторів великої розмірності. Інша гібридизація пошукових алгоритмів (Avramenko, 2022) показала ефективність у глобальній оптимізації багатовимірних нелінійних функцій, також потребує значних обчислювальних витрат, пов'язаних із локальним пошуком. Відтак, актуальною є задача гібридизації розглянутого методу з іншими відомими ефективними методами оптимізації, особливо для дійсного простору.

**Мета дослідження:** розробка оригінального гібридного методу глобальної оптимізації на основі методу штучної імунної системи та методу рою частинок, а також формулювання рекомендацій щодо налаштування його параметрів. Запропонований алгоритм має прискорити прийняття управлінських рішень при керуванні технологічними процесами у металургійному виробництві та підвищити їх точність.

**Виклад основного матеріалу.** В основу запропонованого гібридного методу умовної оптимізації покладаються класичні кроки методу PSO, доповнені двома кроками з методу моделювання штучних імунних систем. На додачу до вільного пошуку в просторі рішень за (1)-(2) ми поділяємо популяцію на групи зі змаганням між ними та застосовуємо стиснення популяції з метою протидії збіжності.

В підсумку алгоритм, що реалізує пропонований метод для пошуку мінімуму в  $r$ -вимірному неперервному просторі, виглядає наступним чином:

**Крок 1.** Генерація початкового рішення. Ініціалізувати рій частинок, що містить  $M$  груп

з  $N_m$  пошукових агентів (векторів довжиною  $r$ ) у кожній, таким чином, що всі  $x_{im}, i = 1 \dots N_m$  представники групи  $m = 1 \dots M$  знаходяться в околі випадкової точки  $x_{im}^0$ , тоді як самі ці точки для кожної з груп якомога більше віддалені одна від одної з метою максимального охоплення області припустимих значень. Для кожної з часток згенерувати випадковий напрямок руху  $V_{i,m}(0) = U^r(-1,1)$  як  $r$ -вимірний вектор з рівномірним розподілом довжини від -1 до 1 по кожній з координат. Лічильник ітерацій  $t=0$ . Лічильники програшів покласти  $C_j = 0, j = 1 \dots M$ . Кращими відомими значеннями для кожного агента, для груп  $i$  для рою загалом покласти число, значно більше за будь-який припустимий розв'язок задачі.

$$\left\{ \begin{array}{l} F_{i,m}^b = \text{realmax}, \forall i = \overline{1, N}, \forall m = \overline{1, M}; \\ F_m^b = \min_j (F_{i,m}^b), \forall m = \overline{1, M}; \\ F^g = \min_m (F_m^b). \end{array} \right. \quad (3)$$

**Крок 2.** Оцінка пристосованості агентів і груп в цілому. Для кожного представника рою обчислити значення цільової функції. Після чого оновити краще значення, відоме даному агенту

$$\left\{ \begin{array}{l} X_{i,m}^b = X_{i,m}(t) \\ F_{i,m}^b = F(X_{i,m}(t)), \text{ if } F(X_{i,m}(t)) < F_{i,m}^b; \end{array} \right. \quad (4)$$

краще значення в групі

$$\left\{ \begin{array}{l} X_m^b = X_{i,m}(t) \\ F_m^b = F(X_{i,m}(t)), \text{ if } F(X_{i,m}(t)) < F_m^b; \end{array} \right. \quad (5)$$

та краще рішення для рою загалом

$$\left\{ \begin{array}{l} X^b = X_{i,m}(t) \\ F^b = F(X_{i,m}(t)), \text{ if } F(X_{i,m}(t)) < F^b. \end{array} \right. \quad (6)$$

**Крок 3.** Ройовий пошук. Лічильник ітерацій  $t=t+1$ . Всі пошукові агенти здійснюють рух за рівняннями

$$V_{i,m}(t) = aV_{i,m}(t-1) + b(X_{i,m}^b - X_{i,m}(t)) + c(X_m^b - X_{i,m}(t)), \quad (7)$$

$$X_{i,m}(t) = X_{i,m}(t-1) + V_{i,m}(t). \quad (8)$$

Останні рівняння отримані модифікацією (1)-(2) з урахуванням (4)-(6).

**Крок 4.** Контроль умов виконання задачі. Обчислити центри мас груп

$$X_m^c = \sum_{i=1}^{N_m} X_{i,m}(t) / N_m, \quad \forall m = \overline{1, M} \quad (9)$$

і перевірити виконання умов:

**Крок 4.1.** Якщо відстань від когось з агентів певної групи  $\forall i = 1, N_m$  до центру мас власної групи більша, ніж до центру мас іншої групи,

тобто  $|X_{i,m}(t) - X_m^c| < |X_{i,m}(t) - X_k^c|$ ,  $k = \overline{1, M}; k \neq m$ , то вектор його швидкості змінюють на спрямований до центру мас власної групи.

$$V_{i,m} = a(X_m^c - X_{i,m}(t)), \quad (10)$$

**Крок 4.2** Якщо пошуковий агент порушує якесь з обмежень (виходить за край області припустимих значень), його вектор швидкості обертається і зменшується вдвічі.

$$V_{i,m} = \frac{-V_{i,m}}{2} \quad (11)$$

**Крок 5.** Імунне стиснення всередині груп. Якщо номер ітерації кратний  $z$  (параметр алгоритму), здійснити в кожній групі стиснення популяції, замінивши надто схожі частинки на випадкові в околі лідера групи, інакше крок 6.

$$X_{i,m}(t) = X_{i,m}^b + N^r(0;1), \text{ if } |X_{i,m}(t) - X_{j,m}(t)| < R, \\ i, j = \overline{1, N_m}; i \neq j, \quad (12)$$

де  $N^r(0;1)$  – нормальний розподіл з математичним очікуванням 0 і дисперсією 1 (може варіюватися в процесі роботи алгоритму);  $R$  – так званий радіус стиснення, параметр алгоритму, який відповідає за запобігання дочасній збіжності алгоритму до одного з рішень.

**Крок 6.** Змагання між групами. Якщо номер ітерації кратний  $f$  (параметр алгоритму) ( $f \neq z$ ), виконати змагання груп рою між собою, спираючись на значення кращого представника групи  $X_m^b$ . Групі-переможниці  $X_w^b = \min(X_m^b)$  лічильник поразок скинути в 0:  $C_w = 0$ , групі, кращий представник якої виявився найгіршим  $X_i^b = \max(X_m^b)$ , додати лічильник програвів  $C_i = C_i + 1$ . Решті груп значення лічильника не змінювати.

**Крок 7.** Стиснення популяції за рахунок найгіршої групи. Якщо якась група набрала  $C_m > C_{max}$ , де  $C_{max}$  – параметр алгоритму, групу розформувати, її представників розподілити по інших групах. Відповідно  $M = M - 1$ ;  $N_m = N_m^{new}$ ;  $C_m = 1, \forall m = \overline{1, M}$  – змінюється кількість груп, в кожній групі змінюється чисельність за умови постійного розміру рою.

**Крок 8.** Перевірка умови завершення алгоритму. Якщо поточне число груп більше ніж одна  $M > 1$ , переходимо до кроку 2, інакше – крок 9.

**Крок 9.** Завершення алгоритму, виведення в якості рішення кращого значення  $F^g$ , знайденого в процесі пошуку за (6).

До викладеного алгоритму методу, названого авторами HPSO (Hybrid Immune Particle Swarm Optimization), слід додати декілька уточнюючих зауважень, які дозволять краще розуміти сутність методу.

По-перше, кожна група з тих, на які поділяється рій, буде мати власну, відмінну від інших груп кількість представників, які не є постійними

$$N_m \neq const, N_m \neq N_k, k, m = \overline{1, M}, k \neq m. \quad (13)$$

Це пов'язано з тим, що на початковому етапі рій має певну кількість пошукових агентів  $nPop = \sum_{m=1}^M N_m$ , яка зберігається протягом всього пошуку. І якщо навіть спочатку можна розподілити агентів по групах рівномірно, вже після першого стиснення популяції (Крок 7), цей рівномірний розподіл може бути порушений. Наприклад, рій з 100 особин ділиться на 5 груп по 20, як змагаються, після чого 20 особин рою, що виявився найгіршим, приєднуються до інших роїв. При цьому розподіл на 4 групи по 25 осіб припустимий, але не гарантований, адже кожна особина пристає до рою, центр мас якого найближчий.

По-друге, гальмування агента на межі області припустимих значень, що виконується на кроці 4.2, має на меті обстеження саме границі області припустимих значень, адже теоретично [4] в задачах умовної оптимізації оптимальні рішення лежать саме в крайніх точках, де проходять одне або декілька обмежень. Водночас соціально-когнітивні довірчі коефіцієнти, які визначають швидкість руху в (7) мають забезпечувати збереження активності пошукового агента навіть після кількарізового порушення обмежень.

Нарешті, слід зауважити, що параметри алгоритму  $z$  та  $f$ , що визначають довжину змагальних циклів відповідно всередині групи та між групами, мають бути по можливості не великими простими числами, аби забезпечувати якомога рідше настання випадку, коли стиснення груп і стиснення всередині групи відбуваються на одній ітерації. Такий випадок небажаний, адже означає фактичний перезапуск рою (спочатку кожна команда суттєво оновлюється, потім ці нові агенти розподіляються по випадкових групах рою).

Наприклад, якщо обрати  $z = 10$  та  $f = 12$ , перезапуск рою буде можливий кожні 60 ітерацій, тоді як при близьких значеннях  $z = 11$  та  $f = 13$  така ситуація можлива лише кожні 143 ітерації.

Вільними параметрами викладеного алгоритму є число груп  $M$ , загальний розмір рою  $nPop$ , частота застосування операції міжгрупових змагань  $f$ , частота стискань з оновленням груп рою  $z$  та максимальна допустима кількість програвів  $C_{max}$ . Разом з тим, у (2) та (7) важливими параметрами є коефіцієнти  $a$ ,  $b$  та  $c$  при складових швидкості, що відповідають за інер-

цію, рух до власного кращого рішення та рух до групового кращого рішення відповідно.

Для отримання оптимальних значень параметрів запропонованого алгоритму був проведений обчислювальний експеримент, який полягав у наступному. Згадані параметри ( $a, b, c, nPop, M, z, f, C_{max}$ ) кодувалися як невідомі координати в задачі оптимізації, яка вирішувалася класичною еволюційною стратегією з повторним запуском з випадковими початковими значеннями і обмеженнями на координати, які зазначені в таблиці 1.

Всі координати розглядалися як дискретні, оскільки для більшості з них (кількість ітерацій, розмір покоління) це природньо. Коефіцієнти ж, що визначають вплив складових швидкості, оцінювалися з інтервалом варіювання 0,1. Така дискретність дозволяє оцінити тенденцію – яким має бути співвідношення коефіцієнтів, не надто переймаючись точним значенням кожного. Пошуку найкращих значень  $a, b$  та  $c$  для розв'язання кожної окремої кожної прикладної задачі має бути присвячене окреме дослідження.

В якості цільової функції було обрано добуток відхилення отриманого розв'язку від наперед відомого глобального оптимуму на тестовій

функції на кількість ітерацій, витрачену для знаходження цього розв'язку. В якості тестових функцій було розглянуто функції Еклі, Растрігіна та Гриванка (Karafotias, 2015) при кількості вимірів  $r = 2, 8$  та  $20$ . Оскільки всі обрані тестові функції незалежно від мірності простору мають глобальний мінімум, що дорівнює нулю, оптимізаційна задача з пошуку найкращих налаштувань запропонованого алгоритму набуває вигляду

$$F_{syn} (a, b, c, M, nPop, f, z, C_{max}) = F_{test} \cdot lter \rightarrow min. \quad (14)$$

Аби зменшити вплив випадкового характеру пошуку рішення, властивого як запропонованому алгоритму, так і еволюційної стратегії, що відповідає за підбір найкращих параметрів, для отримання усередненого результату було виконано 20 запусків при кожній кількості вимірів (загалом 180 запусків).

Результати, усереднені по 20 запусках при відповідній мірності простору та однакових налаштуваннях еволюційної стратегії, ілюструються рисунками 1 і 2, а також наведені в таблиці 2, де крім середніх значень по 20 запусках (позначено як математичне очікування  $M\{x\}$ ), розраховані також оцінки вибіркової дисперсії величин (позначено як  $\sigma_x$ ).

Таблиця 1

**Параметри алгоритму методу HPSO і межі їх дослідження**

Параметр алгоритму	Сенс в алгоритмі	Значення, які приймає
$a$	Коефіцієнт спеціалізації частки	Від 0,5 до 2 з кроком 0,1
$b$	Коефіцієнт ностальгії частки	Від 0,3 до 2 з кроком 0,1
$c$	Коефіцієнт ройової моралі частки	Від 0,3 до 2 з кроком 0,1
$M$	Кількість груп в рої	3, 4, 5, 6, 8, 10, 15
$nPop$	Кількість часток в рої	30, 40, 50, 60, 70, 80, 90, 100, 120, 150
$f$	Періодичність міжгрупового стиснення	2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43
$z$	Періодичність внутрішньогрупового стиснення	2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43
$C_{max}$	Максимальна кількість поразок	2, 3, 4, 5, 6, 7, 8, 10, 12, 15

Таблиця 2

**Отримані оптимальні налаштування алгоритму HPSO**

Параметр	$a$	$b$	$c$	$M$	$nPop$	$f$	$z$	$C_{max}$	
Для 2-мірної задачі	$M\{x\}$	0,8308	1,4618	1,4819	3,88	33	6,27	10,2	3,67
	$\sigma_x$	0,0992	0,1348	0,0988	1,206	17,49	2,93	3,22	1,337
Для 8-мірної задачі	$M\{x\}$	0,773	1,4142	1,5523	5,11	49	12,3	11,1	5,09
	$\sigma_x$	0,0942	0,1887	0,1667	0,889	31,2	8,44	9,45	1,442
Для 20-вимірної задачі	$M\{x\}$	0,7519	1,3421	1,6729	6,22	74,2	18,3	11,9	5,33
	$\sigma_x$	0,0865	0,2506	0,1862	0,927	32,7	15,25	16,67	1,783



На рис. 1 показано контурний графік залежності цільової функції за (14) в просторі координат «кількість груп» та «максимальна кількість програвів» при мінімізації двовимірних функцій. Можна побачити, що при збільшенні кожної з величин понад 5, цільова функція значно зростає за рахунок збільшення кількості ітерацій.

На рис. 2 показано графік та контурний графік залежності цільової функції за (14) в просторі координат «періодичність міжгрупового стиснення» та «періодичність внутрішньогрупового стиснення» при розв'язанні тестових задач у 8-вимірному просторі. З графіка можна помітити, що залежність має один екстремум при сполученні значень трохи більше 10. При цьому помітно, що даний екстремум може бути виділений на кожній з координат.

За результатами у таблиці 2 можна зробити наступні спостереження:

- Коефіцієнти  $a$ ,  $b$  та  $c$  при складових швидкості, що відповідають за інерцію, рух до власного кращого рішення та рух до групового кращого рішення відповідно, з урахуванням їх дисперсій, по-перше, не залежать від розмірності простору, в якому вирішується задача, а по-друге, відповідають теоретичному розподілу  $b \approx c \approx 2a$  (Lin, 2019);

- Початкова кількість груп, на які розділяється рій, має зростати зі зростанням розмірності задачі, але при кількості змінних до 20

не має перевищувати 7, бо це лише затягує процес пошуку рішення без підвищення його точності;

- Загальний розмір рою має зростати зі збільшенням мірності простору, приблизно пропорційно кількості груп розбиття. Для розглянутих задач раціональна кількість пошукових агентів у групі не залежала від задачі й розмірності і складала 10-12 особин;

- Періодичність міжгрупового стиснення має збільшуватись приблизно пропорційно розмірності задачі й може обиратися простим числом у значних межах (велика дисперсія оптимальних рішень);

- Періодичність внутрішньогрупового стиснення для розглянутих задач майже не залежить від розмірності й може обиратися на рівні 7, 11 або 13 ітерацій;

- Кількість поразок певної групи рою, після якої дана група стає неефективною і розпускається, майже не залежить від розмірності задачі й може прийматися в діапазоні від 3 до 6, при цьому менші значення відповідають задачам меншої розмірності.

**Висновки і перспективи подальших досліджень.** В даній статті запропоновано алгоритм оптимізації довільної функції у багатовимірному дійсному просторі, який втілює підходи рою часток та штучної імунної системи. В основу запропонованого гібридного методу умовної оптимі-

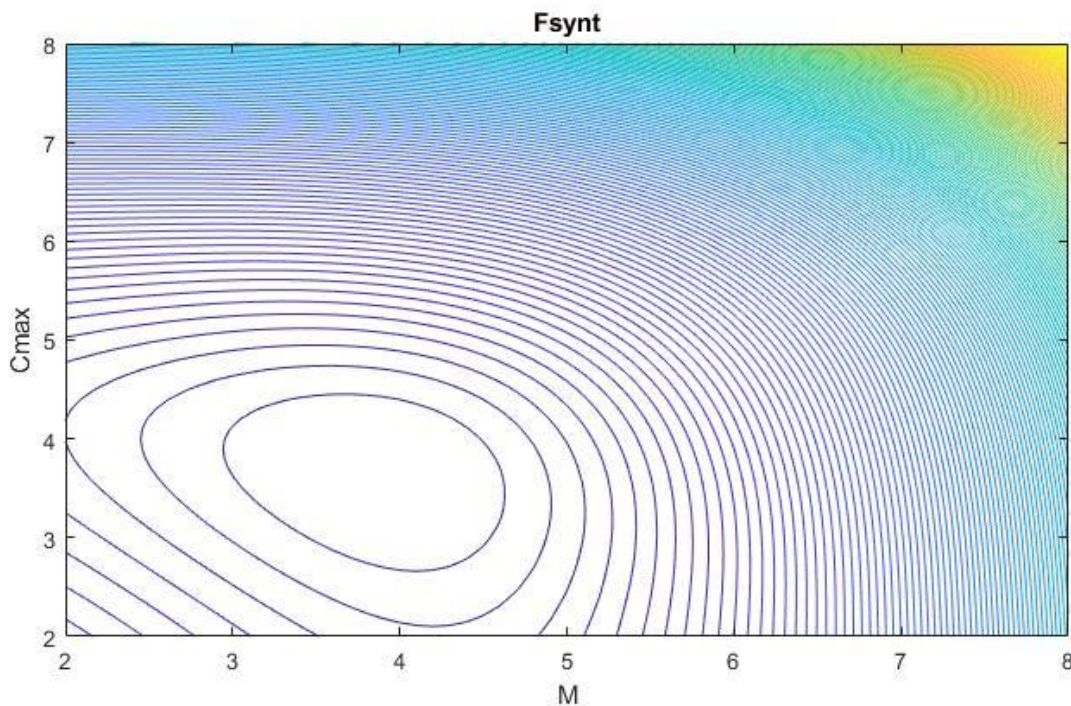


Рис. 1. Цільова функція за (14) в просторі координат  $M$  та  $C_{max}$  для  $r = 2$

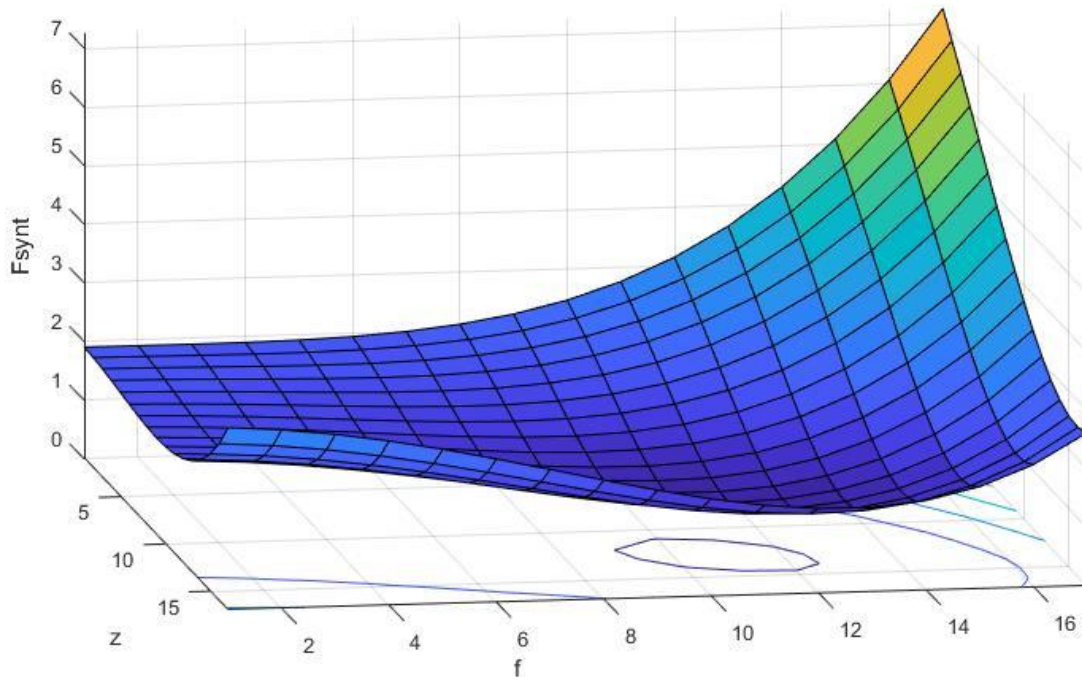


Рис. 2. Цільова функція за (14) в просторі координат  $z$  та  $f$  при  $r = 8$

зації покладаються класичні кроки методу PSO, доповнені двома кроками з методу моделювання штучних імунних систем, а саме поділ популяції на групи зі змаганням між ними та стиснення популяції з метою протидії збіжності. Кожна група з тих, на які поділяється рій, може мати власну, відмінну від інших груп кількість представників, яка не є постійною. З метою докладнішого обстеження області, прилеглої до межі в задачах умовної оптимізації, пошукові агенти гальмуються кожного разу після порушення обмежень.

Проведено дослідження з метою визначення оптимальних налаштувань алгоритму. З'ясовано, що коефіцієнти  $a$ ,  $b$  та  $c$  при складових швидкості, що відповідають за інерцію, рух до власного кращого рішення та рух до групового кращого рішення відповідно не залежать від розмірності простору задачі й відповідають теоретичному розподілу для методу PSO. Також в ході дослідження встановлено, що кількість груп, на які розділяється рій, має зростати зі зростанням розмірності задачі, але для розглянутих задач не має перевищувати 7, тоді як кількість поразок певної групи рою,

після якої дана група стає неефективною і розпускається, майже не залежить від розмірності задачі й може прийматися в діапазоні від 3 до 6. Збільшення обох цих параметрів алгоритму понад рекомендовані межі не є виправданим, це лише затягує процес пошуку рішення без підвищення його точності. Загальний розмір рою має зростати зі збільшенням мірності простору, приблизно пропорційно кількості груп розбиття.

Періодичність міжгрупового стиснення має збільшуватись зі збільшенням розмірності задачі й може обиратися простим числом у широких межах. Водночас періодичність внутрішньогрупового стиснення для розглянутих задач майже не залежить від розмірності й може обиратися рівним 7, 11 або 13 ітерацій.

В подальшому актуальним бачиться застосування викладеного алгоритму для розв'язання задач оптимізації технологічних процесів у металургійному виробництві. Зокрема, даний алгоритм вбачається ефективним для задач шихтування, оптимізації використання феросплавів у ливарному виробництві, а також при прогнозуванні механічних властивостей готової продукції.

#### ЛІТЕРАТУРА:

1. Богушевський, В. С. Розрахунок металевої частини шихти киснево-конвертерної плавки / В.С. Богушевський, В.Ю. Сухенко, К.О. Сергеева, С.В. Жук // *Металлургическая и горнорудная промышленность*. 2010. № 7. С. 266 – 269.

2. Желдак Т.А. Математична модель матеріально-теплого балансу плавки в кисневому конвертері та критерій її оптимізації / Т.А. Желдак, Д.О. Воловенко // Інформаційні технології в освіті, науці й техніці (ІТОНТ-2012): матеріали міжнар. наук.-практ. конф.: Черкаси, 25-27 квітня 2012 р. Черкаси: ЧДТУ, 2012. т.1. с. 23-24.
3. Levitin, Anany. Introduction to the design & analysis of algorithms / Anany Levitin. 3rd ed. 2012. 589p.
4. Гончаренко Я.В. Математичне програмування. К.: НПУ імені М.П. Драгоманова, 2010. 184 с.
5. Жалдак М.І., Триус Ю.В. Основи теорії і методів оптимізації.: Навчальний посібник. Черкаси: Брама-Україна, 2005. 608 с.
6. Комп'ютерна математика. Оптимізація обчислень [Текст] : зб. наук. пр. / відп. ред. І. В. Сергієнко ; НАН України, Ін-т кібернетики ім. В. М. Глушкова. К., 2001.
7. Субботін С. О., Олійник А. О., Олійник О. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей: Монографія / Під заг. ред. С. О. Субботіна. Запоріжжя: ЗНТУ, 2009. 375 с.
8. Das A. and Chakrabarti B. K. (Eds.), Quantum Annealing and Related Optimization Methods, Lecture Note in Physics, Vol. 679, Springer, Heidelberg. 2005
9. Bratton, Daniel; Kennedy, James. Defining a Standard for Particle Swarm Optimization. Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007). pp. 120–127. doi:10.1109/SIS.2007.368035
10. G. Karafotias, M. Hoogendoorn and A. E. Eiben, "Parameter Control in Evolutionary Algorithms: Trends and Challenges," in IEEE Transactions on Evolutionary Computation, vol. 19, no. 2, pp. 167-187, April 2015, doi: 10.1109/TEVC.2014.2308294.
11. Lucinska M., Wierzchon S.T. Hybrid Immune Algorithm for Multimodal Function Optimization // Recent Advances in Intelligent Information Systems, 2009, pp. 301-313. [Електронний документ]. URL: <http://iis.ipipan.waw.pl/2009/proceedings/iis09-30.pdf>.
12. Rowan T.H. Functional Stability Analysis of Numerical Algorithms, Ph.D. Thesis, Department of Computer Sciences, University of Texas at Austin, 1990. [Електронний документ]. URL: [http://reference.kfupm.edu.sa/content/f/u/functional\\_stability\\_analysis\\_of\\_numeric\\_1308737.pdf](http://reference.kfupm.edu.sa/content/f/u/functional_stability_analysis_of_numeric_1308737.pdf).
13. Bernardino, H.S.; Barbosa, H.J.C. Artificial Immune Systems for Optimization. In: Chiong, R. (eds) Nature-Inspired Algorithms for Optimisation. Studies in Computational Intelligence, vol 193. Springer, Berlin, Heidelberg. 2009. [https://doi.org/10.1007/978-3-642-00267-0\\_14](https://doi.org/10.1007/978-3-642-00267-0_14)
14. Navarro-Caceres, M.; Herath, P.; Villarrubia, G.; Prieto-Castrillo, F.; Venyagamoorthy, G.K. "An Evaluation of a Metaheuristic Artificial Immune System for Household Energy Optimization", Complexity, vol. 2018, Article ID 9597158, 11 pages, 2018. <https://doi.org/10.1155/2018/9597158>
15. Fernandez-Marquez, J.L., Di Marzo Serugendo, G., Montagna, S. et al. Description and composition of bio-inspired design patterns: a complete overview. Nat Comput 12, 43–67 (2013). <https://doi.org/10.1007/s11047-012-9324-y>
16. Lin, Q.; Zhu, Q.; Wang, N. and al. A multi-objective immune algorithm with dynamic population strategy, Swarm and Evolutionary Computation, Volume 50, 2019, 100477, <https://doi.org/10.1016/j.swevo.2018.12.003>
17. Li, L.; Lin, Q.; Ming, Zh. A survey of artificial immune algorithms for multi-objective optimization, Neurocomputing, Volume 489, 2022, Pages 211-229, <https://doi.org/10.1016/j.neucom.2021.08.154>
18. Qi, Y.; Hou, Zh.; Yin, M.; Sun, H. & Huang, J. An immune multi-objective optimization algorithm with differential evolution inspired recombination, Applied Soft Computing, Volume 29, 2015, Pages 395-410, <https://doi.org/10.1016/j.asoc.2015.01.012>
19. Yıldız, A. R. An effective hybrid immune-hill climbing optimization approach for solving design and manufacturing optimization problems in industry, Journal of Materials Processing Technology, Volume 209, Issue 6, 2009, Pages 2773-2780, <https://doi.org/10.1016/j.jmatprotec.2008.06.028>
20. Zheldak T. Efficiency Improvement of the Algorithm Based on an Artificial Immune System Modeling Applied to Continuous and Combinatorial Problems / Zheldak, T., Ziborov, I., Lyman, V., Zhuk, A. // CEUR Workshop Proceedings, 3106, 2021. pp. 82 – 95.
21. Avramenko S.E. Guided hybrid genetic algorithm for solving global optimization problems / S.E. Avramenko, T.A. Zheldak, L.S. Koriashkina // Radio Electronics, Computer Science, Control. 2021. № 2.: 174-188. <https://doi.org/10.15588/1607-3274-2021-2-18>

#### REFERENCES:

1. Bohushevskiy V.S., Sukhenko V.Yu., Serheieva K.O., Zhuk S.V. (2010). Rozrakhunok metalevoi chastyny shykhty kysnevo-konverternoi plavky [*Calculation of the metal part of the oxygen-converter melting charge*]. Metallurhycheskaia y hornorudnaia promyshlennost. – v.7. pp. 266 – 269. [in Ukrainian].

2. Zheldak T.A., Volovenko D.O. (2012). Matematychna model materialno-teplovoho balansu plavky v kysnevomu konverteri ta kryterii yii optymizatsii [*Mathematical model of material-heat balance of melting in an oxygen converter and criterion for its optimization*]. Informatsiini tekhnolohii v osviti, nauksi i tekhnitsi (ITONT-2012): materialy mizhnar. nauk. – prakt. konf.: Cherkasy, 25-27 kvitnia 2012 r. Cherkasy: ChDTU. V1. pp. 23-24. [in Ukrainian].
3. Levitin, Anany (2012). Introduction to the design & analysis of algorithms / Anany Levitin. 3rd ed. 589p.
4. Honcharenko Ya.V. (2010). Matematychni prohramuvannia [*Mathematical programming*]. K.: NPU imeni M.P. Drahomanova. 184 p. [in Ukrainian].
5. Zhaldak M.I., Tryus Yu.V. (2005). Osnovy teorii i metodiv optymizatsii.: Navchalnyi posibnyk [*Fundamentals of optimization theory and methods.: Training manual*]. Cherkasy: Brama-Ukraina. 608 p. [in Ukrainian].
6. Serhienko I.V. (at ed.) (2001). Kompiuterna matematika. Optymizatsiia obchyslen [*Computer mathematics. Optimization of calculations*]. Kyiv: NAN Ukrainy, In-t kibernetiky im. V. M. Hlushkova. [in Ukrainian].
7. Subbotin S.O., Oliinyk A.O., Oliinyk O.O. (2009). Neiteratyvni, evoliutsiini ta multyahentni metody syntezy nechitkolohichnykh i neiromerezhnykh modelei [*Non-iterative, evolutionary and multi-agent methods of synthesis of fuzzy and neural network models*]. Zaporizhzhia: ZNTU. 375 p. [in Ukrainian].
8. Das A. and Chakrabarti B. K. (Eds.), (2005). Annealing and Related Optimization Methods, Lecture Note in Physics, Vol. 679, Springer, Heidelberg
9. Bratton, Daniel; Kennedy, James (2007). Defining a Standard for Particle Swarm Optimization. Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007). pp. 120–127. <https://doi.org/10.1109/SIS.2007.368035>
10. G. Karafotias, M. Hoogendoorn and A. E. Eiben (2015). Parameter Control in Evolutionary Algorithms: Trends and Challenges," in IEEE Transactions on Evolutionary Computation, vol. 19, no. 2, pp. 167-187, <https://doi.org/10.1109/TEVC.2014.2308294>.
11. Lucinska M., Wierzchon S.T. (2009). Hybrid Immune Algorithm for Multimodal Function Optimization // Recent Advances in Intelligent Information Systems, pp. 301-313. URL: <http://iis.ipipan.waw.pl/2009/proceedings/iis09-30.pdf>.
12. Rowan T.H. (1990). Functional Stability Analysis of Numerical Algorithms, Ph.D. Thesis, Department of Computer Sciences, University of Texas at Austin. Retrieved from [http://reference.kfupm.edu.sa/content/f/u/functional\\_stability\\_analysis\\_of\\_numeric\\_1308737.pdf](http://reference.kfupm.edu.sa/content/f/u/functional_stability_analysis_of_numeric_1308737.pdf).
13. Bernardino, H.S.; Barbosa, H.J.C. (2009). Artificial Immune Systems for Optimization. In: Chiong, R. (eds) Nature-Inspired Algorithms for Optimisation. Studies in Computational Intelligence, vol 193. Springer, Berlin, Heidelberg. [https://doi.org/10.1007/978-3-642-00267-0\\_14](https://doi.org/10.1007/978-3-642-00267-0_14)
14. Navarro-Caceres, M.; Herath, P.; Villarrubia, G.; Prieto-Castrillo, F.; Venyagamoorthy, G.K. (2018) An Evaluation of a Metaheuristic Artificial Immune System for Household Energy Optimization", Complexity, vol. 2018, Article ID 9597158, 11 pages. <https://doi.org/10.1155/2018/9597158>
15. Fernandez-Marquez, J.L., Di Marzo Serugendo, G., Montagna, S. et al. (2013). Description and composition of bio-inspired design patterns: a complete overview. Nat Comput 12, 43–67. <https://doi.org/10.1007/s11047-012-9324-y>
16. Lin, Q.; Zhu, Q.; Wang, N. and al. (2019). A multi-objective immune algorithm with dynamic population strategy, Swarm and Evolutionary Computation, Volume 50, 100477, <https://doi.org/10.1016/j.swevo.2018.12.003>
17. Li, L.; Lin, Q.; Ming, Zh. (2022). A survey of artificial immune algorithms for multi-objective optimization, Neurocomputing, Volume 489, Pages 211-229, <https://doi.org/10.1016/j.neucom.2021.08.154>
18. Qi, Y.; Hou, Zh.; Yin, M.; Sun, H. & Huang, J. (2015). An immune multi-objective optimization algorithm with differential evolution inspired recombination, Applied Soft Computing, Volume 29, Pages 395-410, <https://doi.org/10.1016/j.asoc.2015.01.012>
19. Yildiz, A.R. (2009). An effective hybrid immune-hill climbing optimization approach for solving design and manufacturing optimization problems in industry, Journal of Materials Processing Technology, Volume 209, Issue 6, Pages 2773-2780, <https://doi.org/10.1016/j.jmatprotec.2008.06.028>
20. Zheldak, T., Ziborov, I., Lyman, V., Zhuk, A. (2021). Efficiency Improvement of the Algorithm Based on an Artificial Immune System Modeling Applied to Continuous and Combinatorial Problems. CEUR Workshop Proceedings, vol. 3106. pp. 82 – 95.
21. Avramenko, S.E.; Zheldak, T.A.; Koriashkina, L.S. (2021). Guided hybrid genetic algorithm for solving global optimization problems / Radio Electronics, Computer Science, Control. Vol. 2. Pp. 174-188. <https://doi.org/10.15588/1607-3274-2021-2-18>

**UDC 004.89**

**DOI <https://doi.org/10.32782/IT/2023-4-2>**

**Dmytro KORNIENKO**

*Student of the Faculty of Computer Sciences, Department of Software Engineering, Kharkiv National University of Radio Electronics, 14, Nauky Ave, Kharkiv, Ukraine, 61166*

**ORCID: 0009-0007-7631-0063**

**Nataliia GOLIAN**

*Associate Professor at the Department of Software Engineering, Kharkiv National University of Radio Electronics, 14, Nauky Ave, Kharkiv, Ukraine, 61166*

**ORCID: 0000-0002-1390-3116**

**To cite this article:** Kornienko, D., Golian, N. (2023). Metody prohnozuvannia dlia vyvchennia ta poperedzhennia nadzvychainykh pryrodnykh yavyshch [Forecasting methods for studying and detecting natural phenomena]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 13–19, doi: <https://doi.org/10.32782/IT/2023-4-2>

**FORECASTING METHODS FOR STUDYING  
AND DETECTING NATURAL PHENOMENA**

*This article describes ways to research and detect natural phenomena based on forecasting methods and techniques. The purpose of this work is creation of improved algorithm that will allow predicting the occurrence of any kind of natural phenomena based on existing statistics. To create proposed algorithm and software, we used existing forecasting methods and techniques, mathematical and causal methods as well as monitoring current affairs will be considered. Proposed algorithm improvements give us possibility to get general prediction or get prediction for some specific kinds of disaster. Also in that article we propose to combine mathematical methods together with artificial intelligence. AI allow us improve accuracy of prediction and provide possibility to increase number of parameters or characteristics to analysis. As AI is modern and fast-growing technology it provides unlimited ways to improve our algorithm and software not only for forecasting of natural phenomena but also for simulate them, analyze consequences, ways to minimize damage and most important – casualties. One of the main advantages of using proposed combining artificial intelligence with old mathematical and statistics methods over using only mathematical or statistical methods is a flexibility of artificial intelligence in their result as mathematical result stabilize with growing statistics data and each new occurrence will not take so big impact on result. But for artificial intelligence each new data can have a critical effect and can correct all forecasts together with expected consequences. As result of that article new software complex will be implemented and integrated to scientific complex for further improvements, learnings, researches and analysis.*

**Key words:** *natural phenomena, disaster prediction, perception architecture, neural network, prediction method, disaster monitoring.*

**Дмитро КОРНІЄНКО**

*студент факультету комп'ютерних наук кафедри програмної інженерії, Харківський національний університет радіоелектроніки, просп. Науки, 14, м. Харків, Україна, 61166*

**ORCID: 0009-0007-7631-0063**

**Наталія ГОЛЯН**

*доцент кафедри програмної інженерії, Харківський національний університет радіоелектроніки, просп. Науки, 14, м. Харків, Україна, 61166*

**ORCID: 0000-0002-1390-3116**

**Бібліографічний опис статті:** Корнієнко, Д., Голян, Н. (2023). Методи прогнозування для вивчення та попередження надзвичайних природних явищ. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 13–19, doi: <https://doi.org/10.32782/IT/2023-4-2>



## МЕТОДИ ПРОГНОЗУВАННЯ ДЛЯ ВИВЧЕННЯ ТА ПОПЕРЕДЖЕННЯ НАДЗВИЧАЙНИХ ПРИРОДНИХ ЯВИЩ

*У статті описано способи дослідження та виявлення природних явищ на основі методів і технік прогнозування. Метою даної роботи є створення вдосконаленого алгоритму, який дозволить прогнозувати виникнення будь-яких природних явищ на основі наявної статистики. Для створення запропонованого алгоритму та програмного забезпечення ми використали наявні методи та техніки прогнозування, математичні та причинно-наслідкові методи, а також моніторинг поточних подій. Запропоновані вдосконалення алгоритму дають нам можливість отримати загальний прогноз або отримати прогноз для деяких конкретних видів надзвичайних природних явищ. Також у цій статті ми пропонуємо поєднати математичні методи разом із штучним інтелектом. ШІ дозволяє підвищити точність прогнозу та надає можливість збільшити кількість параметрів або характеристик для аналізу. Оскільки штучний інтелект є сучасною технологією, яка швидко розвивається, вона надає необмежені способи вдосконалення нашого алгоритму та програмного забезпечення не лише для прогнозування природних явищ, але й для їх моделювання, аналізу наслідків, способів мінімізації збитків і найголовніше – жертв. Однією з головних переваг використання запропонованого поєднання штучного інтелекту зі старими математичними та статистичними методами перед використанням лише математичних або статистичних методів є гнучкість штучного інтелекту в їхніх результатах, оскільки математичні результати стабілізуються зі зростанням статистичних даних, і кожна нова подія не буде мати такого великого впливу на результат. Але для штучного інтелекту кожна нова інформація може мати критичний ефект і може скорегувати всі прогнози разом з очікуваними наслідками. Як результат цієї статті новий програмний продукт буде впроваджено та інтегровано в науковий комплекс для подальшого вдосконалення, навчання, досліджень та аналізу.*

**Ключові слова:** природне явище, попередження надзвичайних явищ, архітектура перцептронів, нейронна мережа, метод прогнозування, моніторинг надзвичайних явищ.

**Introduction.** Natural disasters are dangerous natural phenomena of geophysical, geological, atmospheric origin which are characterized by sudden destruction, which often leads to numerous casualties among people, destruction of residential infrastructure and also causes numerous secondary disasters. Hazardous natural phenomena occur at different times and on different scales and each of them is unique in its own way. Tornadoes and flash floods are short-term destructive events affecting a relatively small area. Other hazards, such as droughts, develop slowly but can affect almost an entire continent and entire countries within months or even years. An extreme weather event can cause several hazards to occur simultaneously or sequentially. In addition to strong winds and rain, a tropical storm can cause flooding and mudflows. In temperate latitudes, severe thunderstorms can be accompanied by large destructive hailstones, tornadoes, strong winds and rain, leading to flash floods. Winter thunderstorms with strong winds and heavy snowfall or freezing rain can also contribute to avalanches on some mountain slopes and heavy runoff or flooding in the next melt season. There are also human-made threats such as dam failure, chemical spills, radioactive threats, etc. But some of them cannot be predicted when they will appear or where they will fall after they appear.

Some National Meteorological and Hydrological Services and Specialized Centers are responsible for studying geophysical hazards, including volcanic

eruptions (airborne ash) and tsunamis, as well as airborne pollutants (radioactive nuclides, biological and chemical agents) and heavily polluted cities [7]. Thanks to them, we are finding new ways to combat natural and human-made disasters.

**Disaster monitoring and forecasting.** The general goal of monitoring hazardous phenomena and processes in nature is to increase the accuracy and reliability of the forecast of natural phenomena based on a combination of intellectual, informational and technological capabilities of various departments and organizations that monitor certain types of hazards. The monitoring data serve as the basis for forecasting. Also, the more accurate and faster the forecasting data is obtained, the faster and better people will have time to prepare for the danger that awaits them.

All of the above forecasting natural disasters is a leading reflection of the probability of occurrence and development of an emergency based on an analysis of the causes of its occurrence, its source in the past and present. Forecasting consists of different elements. One of them is information about the object of forecasting, which reveals its behavior in the past and present, as well as the patterns of this behavior. But there are also disasters that can behave unpredictably, which makes it difficult to predict the occurrence of this disaster, as well as predict its further behavior after its occurrence.

**Disaster predictions methods.** Forecasting natural disasters [3] is understood as a scientifically based prediction of their development, nature and scale. The method of forecasting natural

disasters should be understood as a method for solving the problem of predicting a specific natural disaster with a certain lead time and using certain initial observational materials. Obviously, as we approach the realization of the predicted phenomenon, i.e., as the lead time of the forecast decreases, its accuracy should increase.

Depending on the waiting time for a natural disaster, forecasts are divided into short-term (less than 12-15 days) and long-term (with greater lead time).

Method – a complex technique, an ordered set of simple techniques aimed at developing a forecast as a whole, a way to achieve the goal, based on knowledge of the most general laws.

Forecasting methods (methods) – a certain set of techniques (methods) for performing forecasting operations, obtaining and processing information about the future based on homogeneous forecasting methods.

Forecasting methodology is a field of knowledge about methods, methods, forecasting systems. Forecasting methodologies were divided into the following categories: foresight, goal setting, planning, programming, design, process development prospects in order to identify problems to be solved.

A forecast development methodology is a selected specific combination of forecasting techniques and methods [10]. A forecasting system (“forecasting system”) is an ordered set of techniques and technical means designed to predict complex phenomena or processes.

Forecasting technique – a specific form of theoretical or practical approach to the development of a forecast; one or more mathematical or logical operations aimed at obtaining a specific result in the process of developing a forecast. At the heart of all methods, methods and techniques of forecasting is a heuristic or mathematical approach. The essence of the heuristic approach is to use the opinions of experts. It finds applications for predicting processes that cannot be formalized.

The mathematical approach consists in using the available data on some characteristics of the predicted object, processing them by mathematical methods, obtaining a dependence that connects these characteristics with time, and calculating the dependence of the object’s characteristics at a given point in time according to the data.

This approach involves the use of modeling or extrapolation. Forecasting in most cases is the basis for the prevention of natural and man-made emergencies.

In the mode of daily activities, the possibility of natural disasters is predicted – the occurrence

of an emergency, its place, time and intensity, the possible scale and other characteristics of the upcoming event.

In the event of a natural disaster, the course of the development of the situation, the effectiveness of certain planned measures to eliminate the emergency, the required composition of forces and means are predicted [9]. The most important of all these forecasts is the forecast of the likelihood of a natural disaster. Its results can be most effectively used to prevent accidents and reduce possible losses and damage in advance.

**Comparison of analogues.** The first analog is the most interesting program is Natural Disaster Monitor. It shows user in real time where and when emergency events happened, for example earthquakes, tropical cyclones, floods, volcanoes, drought, forest fires. But the program does not show the probability, but only collects data from the site and shows. It just takes data from [sate https://www.gdacs.org](https://www.gdacs.org) which is created by United Nations and the European Commission, based on data from international organizations GDACS [4] using JRC tools and algorithms for real-time information exchange. It is interesting idea to use this site which would be help us to predict natural phenomena.

The second analog is Disaster Alert. As a previous analog it also shows to user natural phenomena. But unlike the previous analogue, it shows more information about natural phenomena. It based on PDC which take date from the world and sent user when and where natural phenomena.

The third analog Tropical Storm Risk (TSR). TSR offers a leading resource for predicting and mapping tropical storm activity worldwide. It provides forecasts and information to benefit basic risk awareness and decision making from tropical storms. It helps user to see for the detailed mapping and prediction of tropical storm impacts worldwide.

The four analog is Hurricane Hound. It provides hurricane tracking on a scrollable map and hurricane category, 36-hour forecast, location, distance from current location, max winds, gusts, and time. It helps user to know where hurricane goes and when it would be.

All analogues which were considered have their odds and cons. But they don’t predict the natural phenomena they only take information from the international independent open sources about natural phenomena and compare the user’s location with a possible disaster.

**Implementation.** In order to solve the problems of the programs that were given above, we conducted a study and found that the best result is to use a neural network that will help us calculate

the percentage of a possible natural phenomenon. For the method of determining natural phenomena will be given below.

A statistical method for calculating the probability of an emergency situation is proposed for implementation. We will consider the following as the main parameters for describing an emergency situation:

- type of emergency situation with the index  $j$  assigned to it;
- time interval  $t$  for assessing the probability of an emergency occurring;
- estimated number of events  $N_j$  for period  $\Delta t$ ;
- event intensity  $I_j$ ;
- admissible minimum event intensity threshold  $L_j$ .

Therefore any abnormal situation can be represented as a two-dimensional state space with dimensions  $I_j$  and  $N_j$ . The axis  $N_j$  is discrete while  $I_j$  is continuous. The measurements of  $I_j$  and  $N_j$  are not independent and we use the probability density function  $d_j(I_j, N_j)$  to characterize their relationship.

As result the probability of an emergency situation can be represented as a formula:

$$P(t) = \{P_1(I_1, N_1, t), \dots, P_j(I_j, N_j, t)\}. \quad (1)$$

According to the above formula, the probability of occurrence of an emergency situation of a certain type can be calculated by cutting the corresponding function  $P_j(I_j, N_j, t)$  on some interval  $t$ . Since the cross section is the distribution density of a two-dimensional random variable  $d_{j,t}(I_j, N_j)$ , the probability of an event with the expected intensity  $L_j \leq I_j \leq i_{max}$  equals  $n$  times in the interval  $\Delta t = t - t_0$  can be calculated using the next formula:

$$p_j(L_j, i_{max}, n, \Delta t) = \int_{L_j}^{i_{max}} d_{j,t}(i, n) di \quad (2)$$

The main problem of this approach is the large number of parameters in the functions  $P_j(I_j, N_j, t)$ . With an increase in the analyzed interval  $N_{j,max} \rightarrow \infty$  the problem of quantifying the parameters of the function  $P_j$  becomes intractable. Also  $I_j$  is a time-dependent parameter because  $I_{j,N}(t) \uparrow$ , since with increasing  $\Delta t$  the probability of an emergency situation of high intensity increases.

It is possible to optimize solutions to this problem by reducing the time interval  $\Delta t$  to the point where  $N_j = 1$ . As a result we get rid of the time component in the function  $P_j$ .

Therefore, when passing to periods of arbitrary duration  $y\Delta t$  and number of events  $n$  the task of calculating the probability of occurrence of an emergency situation of a certain type can be expressed by the following formula:

$$d_{n,y} = \frac{y!}{n!(y-n)!} p_{j,1}^n (1-p_{j,1})^{y-n} \quad (3)$$

To solve these problems that are described above was suggested to implement software complex for prediction of different natural phenomena. The program will consist of frontend for simplify interaction with program complex, back-end for grep and execute statistical methods for prediction and database which store statistics data (see Figure 1).

Prediction can be improved by adding neural network [1,8] to our program. It provide possibility to make prediction more flexible and count different points which skipped by pure mathematical methods. For implement software complex with neural network following step should be performed:

- selection of types and architects;
- choose data for training;
- training neural network.

To solve the problem we select perceptron architecture (see Figure 2).

Perceptron architecture – is a linear classifier (binary). And it is used in supervised learning. We

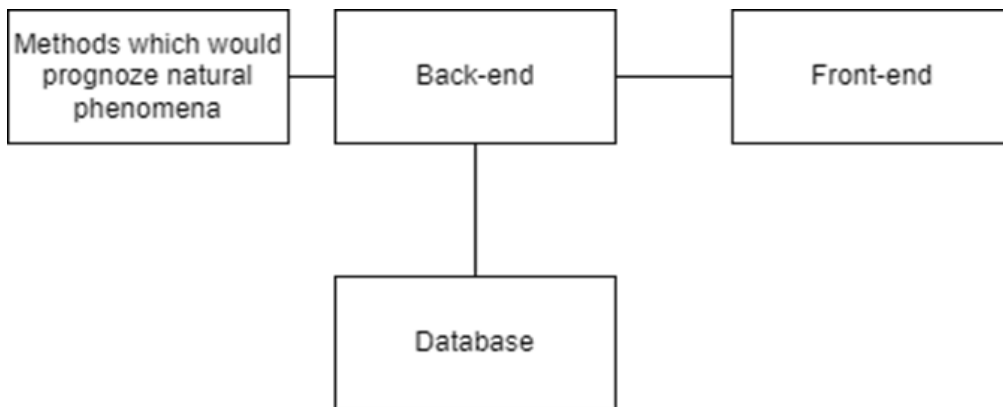
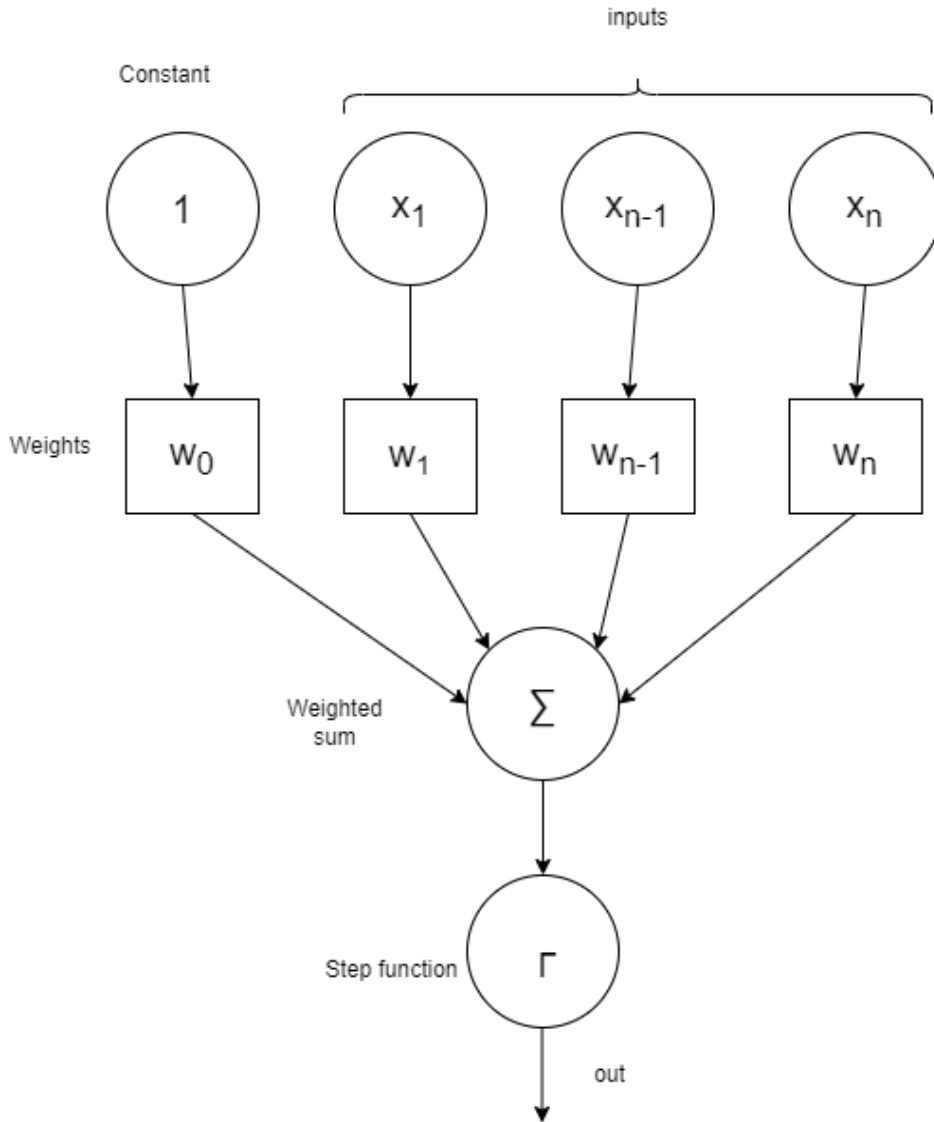


Fig. 1. Program structure





**Fig. 2. Perceptron architecture diagram**

would use – a multilayer perceptron [2]. Which would help us to see will there be a natural phenomenon. But we also need to teach our neural network. To teach it we would use standard algorithm for reverse error propagation. Propagation the way to move from the Input layer to the Output layer in the neural network. This algorithm is universal and it solving many problems also it has low computational complexity.

In the back end, methods will be implemented that will determine the accuracy of the forecast of the prediction of a natural phenomenon. The database will store users, as well as data needed to determine the accuracy of the forecast of a natural phenomenon. It would be contains user data (Example username, email, password) and forecast data (Example air temperature, wind speed and direction, humidity and precipitation, country and city).

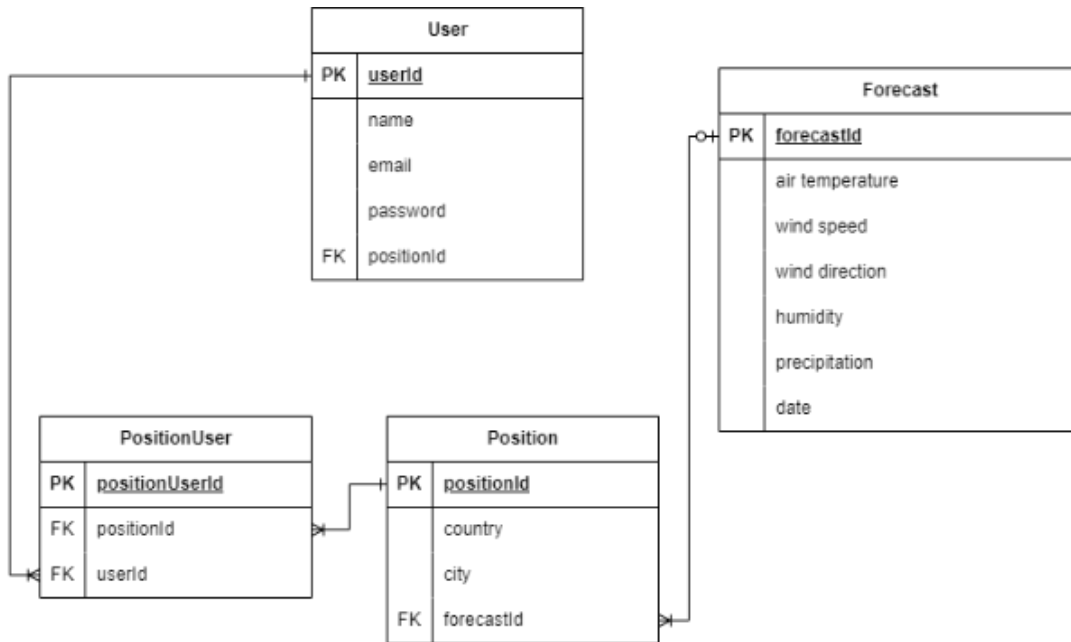
The front-end will represent a user interface where a person can enter data to determine the forecast of a natural phenomenon.

**Result.** The solution which present in the work would be use neural network and algorithm which help us to predict natural phenomena. Our system would help our user:

- when and where would be natural phenomena;
- how powerful it would be;
- where he can find shelter;
- user would have warning if near would be natural phenomena.

Thus, the system of detecting natural phenomena gives the chance to save live when near starts natural disaster.

**Conclusions.** This paper represents the general methods and approaches to predicting the occurrence of emergency situations. In particular, the method of statistical forecasting is con-



**Fig. 3. Database diagram**

sidered and a software implementation of the solution of the problem based on this method is proposed. The accuracy and quality of the forecast depend on the detail of the description of emergency situations that have already occurred, but the computational complexity of the solution

also increases. Therefore, as one of the optimization methods, it was proposed to introduce a lower event intensity threshold, which makes it possible to discard low-intensity events from consideration as insignificant, thereby reducing computational complexity.

**BIBLIOGRAPHY:**

1. Бурков А. Машинне навчання без зайвих слів. Print2print. 2019.
2. Горник К., Стінчкомб М., Уайт Г. Багаторівневі мережі прямого зв'язку є універсальними апроксиматорами. Neural Networks. 1999.
3. Тейлор С. Д. Наука про стихійні лиха: Коли природа і люди стикаються. Nomad Press. 2020.
4. Global Disaster Alert and Coordination System : веб-сайт. URL: <https://www.gdacs.org> (дата звернення: 10.09.2023).
5. Кіпроно Е.К. Як працює зворотне поширення в нейронних мережах? 2022. Towards Data Science : веб-сайт. URL: <https://towardsdatascience.com/how-does-back-propagation-work-in-neural-networks-with-worked-example-bc59dfb97f48> (дата звернення: 12.09.2023).
6. Джозеф Р. Зміна клімату: що потрібно знати кожному. Oxford University Press. 2015. 328 с.
7. Чару К. А. Нейронні мережі та глибоке навчання: Підручник. Springer. 2018. 520 с.
8. Льюїс Н.Д. Нейронні мережі для прогнозування часових рядів за допомогою R. Apress. 2018. 700 с.
9. ECMWF : веб-сайт. URL: <https://www.ecmwf.int/> (дата звернення: 12.09.2023).
10. МакГілліврей П. Передбачення непередбачуваного : веб-сайт. URL: <https://www.uvic.ca/knowledge/archives/2010/v10n04-april10.php> (дата звернення: 12.09.2023).

**REFERENCES:**

1. Burkov, A. (2019). Mashynne navchannia bez zaivvykh sliv [The Hundred-Page Machine Learning Book]. Print2print.
2. Hornik K., Stinchcombe M., White H. (1999). Bahatorivnevi merezhi priamoho zviazku ye universalnymy aproksymatoramy [Feedforward Networks are Universal Approximators]. Neural Networks.
3. Diane C. Taylor. (2020). Nauka pro stykhiini lykha: Koly pryroda i liudy stykaiutsia [The Science of Natural Disasters: When Nature and Humans Collide]. Nomad Press.
4. Global Disaster Alert and Coordination System. September 10, 2023, Retrieved from <https://www.gdacs.org>

5. Kiprono E. K. (2022). Yak pratsiuie zvorotne poshyrennia v neironnykh merezhakh? [How Does Back-Propagation Work in Neural Networks?]. September 12, 2023, Retrieved from <https://towardsdatascience.com/how-does-back-propagation-work-in-neural-networks-with-worked-example-bc59dfb97f48>
6. Joseph R. (2015). Zmina klimatu: shcho potribno znaty kozhnomu [Climate Change: What Everyone Needs to Know]. Oxford University Press.
7. Charu C. (2018). Neironni merezhi ta hlyboke navchannia: Pidruchnyk [A Neural Networks and Deep Learning: A Textbook]. Springer.
8. Lewis N. D. (2018). Neironni merezhi dlia prohnozuvannia chasovykh riadiv za dopomohoiu R [Neural Networks for Time Series Forecasting with R]. Apress.
9. ECMWF. September 12, 2023, Retrieved from <https://www.ecmwf.int/>
10. McGillivray P. (2010). Peredbachennia neperedbachuvanoho [Predicting the unpredictable]. September 12, 2023, Retrieved from <https://www.uvic>

УДК 519.8

DOI <https://doi.org/10.32782/IT/2023-4-3>

### **Ларуса КОРЯШКИНА**

кандидат фізико-математичних наук, доцент, доцент кафедри системного аналізу і управління, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, [koriashkina.l.s@nmu.one](mailto:koriashkina.l.s@nmu.one)

ORCID: 0000-0001-6423-092X

Scopus Author ID: 55844269100

### **Данило ЛУБЕНЕЦЬ**

аспірант кафедри системного аналізу і управління, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, [lubenets.d.y@nmu.one](mailto:lubenets.d.y@nmu.one)

ORCID: 0009-0000-8563-3760

**Бібліографічний опис статті:** Коряшкіна, Л., Лубенець, Д. (2023). Математичні моделі та методи мультиплексного розбиття і багатократного покриття множин для задач розміщення-розподілу. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 20–31, doi: <https://doi.org/10.32782/IT/2023-4-3>

## **МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДИ МУЛЬТИПЛЕКСНОГО РОЗБИТТЯ І БАГАТОКРАТНОГО ПОКРИТТЯ МНОЖИН ДЛЯ ЗАДАЧ РОЗМІЩЕННЯ-РОЗПОДІЛУ**

Розроблено математичні моделі для задач оптимального розміщення сервісних центрів й розподілу транспортних потоків, а також зонування територій, задля оцінювання місткості розміщуваних центрів та необхідної кількості транспортних засобів у логістичних системах. Представлені моделі та методи оптимального розбиття області на зони обслуговування центрів за критеріями мінімальної відстані або найшвидшого надання сервісу, враховуючи можливість забезпечення послугою будь-яким з декількох найближчих до споживачів сервісних центрів. Моделі неперервних задач оптимального багатократного кульового покриття множин удосконалено на випадок врахування особливості множини, на якій можуть бути розміщені центри, що забезпечує уникнення розташування центрів занадто близько один від одного. Описано методи і наближені алгоритми розв'язання вказаних задач. Наведено аналіз результатів обчислювальних експериментів.

**Ключові слова:** логістика, зонування території, багатократне покриття множини, математичне моделювання, мультиплексне розбиття множин, потужність центру.

### **Larysa KORASHKINA**

PhD, Associate Professor, Associate Professor at Department of System Analysis and Control, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave, Dnipro, Ukraine, 49005, [koriashkina.l.s@nmu.one](mailto:koriashkina.l.s@nmu.one)

ORCID: 0000-0001-6423-092X

Scopus Author ID: 55844269100

### **Danylo LUBENETS**

Postgraduate Student of Department of System Analysis and Control, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave, Dnipro, Ukraine, 49005, [lubenets.d.y@nmu.one](mailto:lubenets.d.y@nmu.one)

ORCID: 0009-0000-8563-3760

**To cite this article:** Koriashkina, L., Lubenets, D. (2023). Matematychni modeli ta metody multiplypleksnoho rozbytta i bahatokratnoho pokryttia mnozhyn dlia zadach rozmishchennia-rozpodilu [Mathematical models of multiplex partitioning and multiple coverage of sets for the location-allocation problems]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 20–31, doi: <https://doi.org/10.32782/IT/2023-4-3>

## MATHEMATICAL MODELS OF MULTIPLEX PARTITIONING AND MULTIPLE COVERAGE OF SETS FOR THE LOCATION-ALLOCATION PROBLEMS

*We developed the mathematical models for the problems of optimal service centers' placement and traffic flow distribution, as well as area zoning to estimate the capacity of the located centers and the necessary number of vehicles in logistics systems. Presented are models and methods of optimal division of the region into service center zones based on the criteria of the minimum distance or the fastest service providing the service by any of several service centers closest to consumers. The mathematical formulation of the continuous problem of optimal multiple coverage of sets is improved by considering the peculiarity of the set on which the centers can be placed, which ensures the avoidance of the location of centers too close to each other. Methods and approximate algorithms for solving these problems are described. An analysis of the results of computational experiments is presented.*

**Key words:** *logistics, area zoning, multiple coverage of set, mathematical modeling, multiplex partitioning of sets, center capacity.*

**Вступ.** Існує велика кількість робіт вчених і практиків, у яких піднімаються різні аспекти математичного моделювання і методів розв'язання оптимізаційних задач розміщення-розподілу. Такі задачі виникають, зокрема, під час розробки комплексу запобіжних заходів щодо організації процесів евакуації населення або надання первинної гуманітарної допомоги у разі надзвичайних ситуацій. На сьогодні в науковій літературі запропоновано чимало моделей для задач раціонального територіального розподілу елементів систем екстреної логістики. Широко представлені дослідження евакуаційних процесів з використанням моделей і методів математичного програмування [1, 2]. Серед них є моделі розміщення-розподілу для складання планів евакуації під час ураганів або землетрусів, пошуку оптимальних маршрутів евакуації і місць укриття у випадках міських надзвичайних ситуацій. В даній роботі запропоновано для математичного опису оптимізаційних задач розміщення підрозділів систем екстреної логістики використовувати моделі і методи теорії оптимального мультиплексного розбиття і багатократного покриття множин [3, 4].

**Мета даної роботи** є розроблення моделей і методів розв'язання для логістичних задач, зокрема, в системах екстреної логістики, що дозволить завчасно визначати зони екстреної допомоги, проводити організацію логістичних процесів, раціонально розподіляючи транспортні та матеріальні ресурси.

Об'єктом дослідження є територіальний розподіл елементів транспортно-логістичних систем.

Предметом дослідження є моделі та методи оптимального розміщення структурних підрозділів транспортно-логістичних систем з визначенням зон їх обслуговування.

**Постановка проблеми.** Будемо розглядати наступну задачу. Нехай в деякому регіоні потрібно розмістити центри рятуваль-

них служб або пунктів первинної медичної допомоги на випадок надзвичайних ситуацій у такий спосіб, аби надання відповідної послуги можна було здійснити в найкоротший термін навіть найвіддаленішому мешканцю регіону. До того ж, для кожного центру потрібно визначити зону відповідальності. Це означає, що потрібно розбити територію регіону на підрегіони, мешканці кожного з яких обслуговуватимуться певним центром або центрами, якщо припустити можливість перекриття зон підпорядкування.

**Літературний огляд.** Наведемо тут короткий огляд останніх публікацій, які розглядають питання математичного моделювання практичних задач, що зводяться до задач кульового покриття множини. В роботі [5] на основі формалізації критеріїв повноти покриття побудовано узагальнену математичну модель задачі покриття довільної області ідентичними колами. При цьому отримано задачу негладкої оптимізації з областю допустимих розв'язків, що задається системою нерівностей. Останні слугують для врахування технологічних обмежень, що записується за допомогою  $\rho$ -функцій. Авторами розроблено засоби генерації множини реалізацій узагальненої математичної моделі покриття для широкого класу прикладних задач. Запропоновано стратегію розв'язку виникаючих задач нелінійного програмування.

Проблема пошуку найкращих місць розміщення центрів з покриттям області зонами обслуговування так, аби мінімізувати загальну кількість об'єктів, необхідних для задоволення всіх вимог, розглянуто в роботі [6]. Вивчається новий варіант проблеми, який називається проблемою розташування покриття з контролем перекриття. Ця задача моделює реальні контексти, пов'язані з перевантаженими системами відвідуваності, які потребують зон покриття з перекриттям. Таким чином, кожна вимога має бути покрита певною кількістю додаткових об'єктів, щоб гарантувати, що вимоги будуть

задоволені, навіть якщо призначений об'єкт не може це зробити через певну проблему об'єкта. Ця функція важлива в державних і екстрених службах. Практика розв'язання задач багатократного покриття показує, що кількість додаткових об'єктів, що покривають, є надмірною в деяких точках попиту. В зазначеній роботі автори намагаються здійснювати контроль збігів, щоб визначити пріоритетність регіонів із високою щільністю населення або мінімізувати кількість зон покриття для кожної точки попиту. Ними запропоновано нову математичну модель, яка контролює перекриття між зонами покриття.

В [6] також розглянуто клас задач покриття області множиною геометричних об'єктів заданої форми. Тут для формалізації задачі використовується поняття конфігураційного простору геометричних об'єктів, узагальненими змінними якого є метричні та параметри розміщення. Побудовано математичну модель задачі як задачі нелінійної оптимізації та досліджено її властивості. Практичне застосування запропонованих результатів описано на прикладі вирішення задачі визначення місця розміщення пожежних постів з метою забезпечення пожежної безпеки населених пунктів.

Задачі оптимізації розподільчих процесів в ієрархічних транспортно-логістичних системах вивчаються в [7]. Авторами запропоновано математичні моделі двоетапних і частково-двоетапних задач оптимального розподілу матеріальних потоків в системах екстреної логістики.

Роботи [9, 10] пропонують дискретно-неперервні моделі для задач оптимального розміщення сервісних центрів з обмеженнями, що враховують вартість об'єктів, включаючи постійні витрати на відкриття та змінні витрати, пов'язані з потужністю центрів. Проблема розташування об'єктів передбачає оптимальне розміщення набору об'єктів, які повинні задовольняти запити групи споживачів, розподілених на площинній території. В роботі [9] розташування існуючих об'єктів відомі та класифікуються як окремі. З іншого боку, розташування нових об'єктів, які необхідно знайти, класифікуються як безперервні та дискретні. Крім того, оптимальна кількість нових об'єктів визначається за формулою судження. Авторами розроблено жадібний багатозадачний метод локації для запропонованої задачі частково цілочисельного лінійного програмування, який ефективно визначає точні місцезнаходження нових об'єктів поетапно. Для задач великої розмірності розробили динамічну ітераційну часткову оптимізацію на основі двох змінних, щоб отри-

мати майже оптимальні рішення протягом заданого ліміту часу. В роботі [10] запропонована математична модель для задачі раціонального вибору місць розташування сервісних центрів на базі існуючих об'єктів із одночасним розбиттям території на зони обслуговування центрів з можливим перекриттям цих зон задля збільшення ймовірності отримання клієнтом послуги.

В даній роботі розглядаються математичні моделі для задач оптимального розміщення-розподілу у вигляді неперервних задач оптимального мультиплексного розбиття та багатократного покриття множин. За своїми інтерпретаціями ці задачі є спорідненими. Далі продемонстровано зв'язок між вказаними задачами у різних їх постановках, можливість визначити радіус кульового покриття, розв'язуючи неперервні лінійні задачі мультиплексного розбиття множин. Показано, яким має бути критерій якості мультиплексного розбиття множин, аби у відповідній задачі з розміщенням центрів можна було б отримати таке їх розташування, яке збігатиметься з центрами куль мінімального радіусу, котрі  $k$ -кратно покривають задану множину.

**Матеріали та методи.** Введемо наступні позначення:  $\Omega \subset E_2$  – заданий регіон;  $\rho(x), x \in \Omega$  – невід'ємна функція, що описує щільність розподілу мешканців на території регіону  $\Omega$ ;  $N$  – кількість рятувальних об'єктів, що потрібно розмістити у регіоні  $\Omega$ ;  $\tau_i = (\tau_i^{(1)}, \tau_i^{(2)}) \in \Omega$ , для усіх  $i = 1, \dots, N$ , – координати цих об'єктів (далі центрів);  $N = \{1, 2, \dots, N\}$  – множина всіх індексів центрів;  $k$  – кратність зонування, тобто на допомогу якої максимальної кількості рятувальних об'єктів може розраховувати клієнт;  $M(N, k)$  – множина всіх  $k$ -елементних підмножин множини  $N$ ,  $|M(N, k)| = L$ ;  $\sigma_l = \{j_1^l, j_2^l, \dots, j_k^l\}$ ,  $l = \overline{1, L}$  – елементи множини  $M(N, k)$ ;  $\Omega_{\sigma_l}$  – підмножина точок із  $\Omega$ ,  $l = \overline{1, L}$ , яка відповідає елементу  $\sigma_l$  множини  $M(N, k)$  або (що є тим самим) набору центрів  $\{\tau_{j_1^l}, \tau_{j_2^l}, \dots, \tau_{j_k^l}\}$ ;  $\Sigma_{\Omega}^{N, k}$  – клас всіх можливих розбиттів  $k$ -го порядку множини  $\Omega$ ;  $c(x, \tau_i) / w_i$  – час, за який можна дістатися від центру  $\tau_i$  до точки  $x \in \Omega$ ,  $c(x, \tau_i)$  – метрика, що визначає відстань між двома точками,  $1 / w_i > 0$  – коефіцієнт пропорційності;  $a_i$  – умовна вартість надання послуги центром  $\tau_i$ , розрахована на одного потенційного клієнта (постраждалого, користувача);  $b_1, b_2, \dots, b_N$  – так звані «потужності» рятувальних центрів, що визначають певним чином максимально можливу кількість користувачів, на яку розрахований відповідний центр.

Математичну формалізацію задачі зробимо за допомогою моделей неперервних задач

мультиплексного розбиття множин з певним критерієм якості.

**Задача 1.** Потрібно

$$F_R(\bar{\omega}, \tau^N) = \max_{l=1, L} \sup_{x \in \Omega_{\sigma_l}} \max_{i \in \sigma_l} (c(x, \tau_i) / w_i + a_i) \rho(x) \rightarrow \min_{\substack{\bar{\omega} \in \Sigma_{\Omega}^{N,k} \\ \tau^N \in \Omega^N}}, \quad (1)$$

за умов

$$\text{Capacity}(\tau_i) = b_i, \quad i = \overline{1, p},$$

$$\text{Capacity}(\tau_i) \leq b_i, \quad i = \overline{p+1, N}, \quad (2)$$

де  $\text{Capacity}(\tau_i) = \sum_{l: i \in \sigma_l} \int_{\Omega_{\sigma_l}} \gamma'_l \rho(x) dx \quad i = \overline{1, N},$

а коефіцієнти  $\gamma'_l$  у лівих частинах обмежень залежать певним чином від потужностей центрів і для всіх  $i = \overline{1, N}$ ,  $l = \overline{1, L}$ :

$$0 \leq \gamma'_l \leq 1, \quad \gamma'_{l_1} + \gamma'_{l_2} + \dots + \gamma'_{l_k} = 1. \quad (3)$$

Наприклад, якщо вважати, що виклики на допомогу можуть бути прийняті і оброблені об'єктами на всій області  $\Omega$  пропорційними їхнім потужностям, то для всіх  $l = \overline{1, L}$  і для всіх  $j = \overline{1, N}$ , таких, що  $j \in \sigma_l$ , величина  $\gamma'_l$  може бути задана наступним виразом:  $\gamma'_l = b_j / \sum_{q: q \in \sigma_l} b_q$ .

За таких міркувань обмеження (3) вказують на те, що всі можливості центрів  $\tau_i$ ,  $i = \overline{1, p}$ , мають бути повністю використані, а для решти центрів  $\tau_i$ ,  $i = p+1, \dots, N$ , вони є обмеженими.

Стисло задачу 1 називатимемо *задачею оптимального багатократного покриття множини при обмеженнях* з оглядом на те, що за певних вихідних даних цільовий функціонал визначає величину радіуса оптимального покриття  $R(\tau^N) = \inf_{\tau^N} \sup_{x \in \Omega} \min_{i=1, \dots, N} c(x, \tau_i)$ .

Наведемо математичну постановку задач багатократного покриття обмеженої в  $E_2$  області колами мінімального радіуса, використовуючи математичний апарат теорії неперервних задач оптимального покриття множин [12].

Нехай  $\Omega$  – обмежена, вимірна за Лебегом замкнена множина у просторі  $E_2$ ,  $\tau_i = (\tau_i^{(1)}, \tau_i^{(2)}) \in \Omega$ , для усіх  $i = \overline{1, \dots, N}$ , – деякі точки, що звуться «центрами» (вони можуть бути фіксованими або підлягати визначенню).  $B(\tau_i, R) = \{x \in E_n : c(x, \tau_i) \leq R\}$  – с-куля радіуса  $R$  з центром у точці  $\tau_i$  з  $\Omega$ , де  $c(x, \tau_i)$  – метрика.

Сукупність центрів  $\tau_1, \dots, \tau_N$  задає  $k$ -кратне кульове покриття множини  $\Omega$  з радіусом  $\hat{R}$ , якщо

$$\Omega \subseteq \bigcup_{i=1}^N B(\tau_i, \hat{R}),$$

і для кожної точки  $x \in \Omega$  виконується умова

$$x \in \bigcap_{j=1}^l B(\tau_{i_j}, \hat{R}), \quad i_j \in \{1, 2, \dots, N\}, \quad k \leq l \leq N.$$

Радіус  $\hat{R}$   $k$ -кратного покриття множини  $\Omega$ , котре генерується центрами  $\tau_1, \dots, \tau_N$  (вектором  $\tau^N$ ), визначається за наступною формулою:

$$\hat{R}(\tau^N) = \sup_{x \in \Omega} \min_{i=1, \dots, N} c(x, \tau_i), \quad (4)$$

при цьому для кожної точки  $x \in \Omega$  виконується умова

$$x \in \bigcap_{j=1}^l B(\tau_{i_j}, \hat{R}), \quad k \leq l \leq N, \quad i_j \in \{1, 2, \dots, N\}, \quad (5)$$

де  $\tau^N = (\tau_1, \dots, \tau_N) \in E_n^N$  (або, в окремому випадку,  $\tau^N = (\tau_1, \dots, \tau_N) \in \Omega^N$ ).

$k$ -кратне покриття множини  $\Omega$ , що задається вектором  $\tau^N = (\tau_1, \dots, \tau_N)$ , з радіусом  $\hat{R}(\tau^N)$ , який визначається за формулою (4) й задовольняє умови (5), є **мінімальним  $k$ -кратним с-кульовим покриттям**, яке генерується вектором  $\tau^N$ . Це означає, що жоден набір с-куль меншого радіусу з центрами в точках  $\tau_1, \dots, \tau_N$  не покриває  $k$ -кратно множини  $\Omega$ . А  $k$ -кратне покриття мінімального радіусу називається **оптимальним  $k$ -кратним покриттям**.

Отже, для пошуку оптимального  $k$ -кратного покриття необхідно визначити величину радіусу оптимального покриття

$$\hat{R}(\tau^N) = \inf_{\tau^N} \sup_{x \in \Omega} \min_{i=1, \dots, N} c(x, \tau_i)$$

і вектор  $\tau^N = (\tau_1^*, \dots, \tau_N^*)$ , на яких досягається значення  $\hat{R}(\tau^N)$  за умови виконання для кожної точки  $x \in \Omega$  включення (5).

Задача про пошук радіуса  $N$  куль, що створюють  $k$ -кратне с-кульове покриття множини, може бути формалізована математично іншим способом.

Нехай  $\Omega$  – обмежена замкнена множина з простору  $E_2$ ,  $\tau^N = (\tau_1, \dots, \tau_N)$  – заданий на множині  $\Omega$  (або в просторі  $E_2$ ) набір центрів. Точки  $\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}$  є  $k$ -найближчими сусідами точки  $x \in \Omega$  із заданих  $N$  точок, якщо

$$\forall j = \overline{1, k} \quad c(x, \tau_{i_j}) < c(x, \tau_m), \quad m \in \{1, 2, \dots, N\} \setminus \{i_1, i_2, \dots, i_k\}.$$

В окремих випадках для деяких точок  $x \in \Omega$  знак вищезазначеної нерівності може бути нестрогим. Тоді вважається, що точка  $x \in \Omega$  має декілька різних наборів з  $k$  найближчих сусідів, а під час чисельної реалізації алгоритмів їх пошуку для однозначності припускається, що набір  $k$  найближчих сусідів утворюють точки  $\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}$ , які задовольняють наступній системі нерівностей

$$\forall j = \overline{1, k} \quad c(x, \tau_{i_j}) \leq c(x, \tau_m), \quad m \in \{1, 2, \dots, N\} \setminus \{i_1, i_2, \dots, i_k\}$$

і мають найменші можливі індекси.

Введемо до розгляду множину  $\Lambda_N^k$   $N$ -вимірних векторів, координати яких можуть

приймати значення 0 або 1, причому у кожному такому векторі число одиниць дорівнює  $k$ :

$$\Lambda_N^k = \left\{ \lambda = (\lambda_1, \dots, \lambda_N) : \lambda_i = 0 \vee 1, i = \overline{1, N}; \sum_{i=1}^N \lambda_i = k \right\}.$$

Вочевидь,  $|\Lambda_N^k| = C_N^k$ .

Тоді для кожної точки  $x \in \Omega$   $k$  найближчих сусідів з фіксованого набору точок  $(\tau_1, \dots, \tau_N)$  можна знайти, розв'язуючи задачу пошуку такого вектору  $\lambda(x) = (\lambda_1(x), \dots, \lambda_N(x)) \in \Lambda$ , за якого досягається мінімальне значення наступної величини:

$$C(x) = \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, \dots, N} c(x, \tau_i) \lambda_i(x). \quad (6)$$

Для кожної точки  $x \in \Omega$  компонента  $\lambda_i(x)$  цієї вектор-функції дорівнює 1 тоді й тільки тоді, коли центр  $\tau_i$  є одним з  $k$  можливих її «сусідів». Вектор  $\lambda(x)$ , на якому досягається величина  $C(x)$ , визначає  $k$  найближчих до точки  $x$  центрів, а  $C(x)$  є радіусом куль з центрами  $\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}$ , які  $k$ -кратно покривають точку  $x$  (індекси цих центрів і одиничних компонент вектору  $\lambda(x)$  збігаються).

Задача про пошук радіуса  $N$  кругів, які створюють  $k$ -кратне  $c$ -кульове покриття множини полягає у пошуку величини

$$\bar{R} = \sup_{x \in \Omega} \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, \dots, N} c(x, \tau_i) \lambda_i(x).$$

Задача про мінімальне  $k$ -кратне  $c$ -кульове покриття записується у такий спосіб: знайти величину

$$\bar{R}(\lambda^*(\cdot), \tau^N) = \inf_{(\tau_1, \dots, \tau_N) \in \Omega^N} \sup_{x \in \Omega} \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, \dots, N} c(x, \tau_i) \lambda_i(x), \quad (7)$$

$$\text{де } \Lambda_N^k = \left\{ \lambda = (\lambda_1, \dots, \lambda_N) : \lambda_i = 0 \vee 1, i = \overline{1, N}; \sum_{i=1}^N \lambda_i = k \right\},$$

а також вектор-функцію  $\lambda^*(\cdot) : \forall x \in \Omega \lambda^*(x) \in \Lambda_N^k$ , та вектор  $\tau^N = (\tau_1^*, \dots, \tau_N^*) \in \Omega^N \subset E_2^N$ , за яких у виразі (7) досягається нижня грань.

Відтак, вектор-функція  $\lambda(\cdot)$  використовується для конструктивного запису математичної моделі задачі та є проміжним результатом, містить інформацію про те, які саме  $k$  центрів є найближчими до кожної точки з  $\Omega$ .

Наведемо опис методу розв'язання задачі про багатократне кульове покриття обмеженої множини і формули, які визначають субградієнт негладкої цільової функції (7).

Позаяк для компактної множини  $\Omega$  з  $E_2$  і неперервної функції  $c$ -оптимальне  $k$ -кратне покриття множини  $\Omega$  заданим  $N$  числом  $c$ -куль однакового радіуса існує, задачу (7) можна записати у наступному вигляді:

$$\hat{R}(\tau_1, \dots, \tau_N) \rightarrow \min_{(\tau_1, \dots, \tau_N) \in \Omega^N}, \quad (8)$$

де

$$\hat{R}(\tau_1, \dots, \tau_N) = \max_{x \in \Omega} \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, \dots, N} c(x, \tau_i) \lambda_i(x).$$

В роботі [11] наведені деякі властивості функції  $\hat{R}(\tau_1, \dots, \tau_N)$  з евклідовою метрикою  $c(x, \tau_i) = \|x - \tau_i\|_2$ , зокрема, неперервність, ліпшицевість, неопуклість, майже всюди диференційовність, багатоекстремальність. Крім того, для не пустої компактної множини  $\Omega$  з  $E_2$  і  $\tau^N = (\tau_1, \dots, \tau_N)$  з  $\Omega^N$  справедливою є наступна рівність:

$$\begin{aligned} \hat{R}(\tau_1, \dots, \tau_N) &= \max_{x \in \Omega} \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, \dots, N} c(x, \tau_i) \lambda_i(x) = \\ &= \max_{m=1, \dots, M} \max_{x \in \Omega_m} \max_{i \in T_m} c(x, \tau_i), \end{aligned}$$

де множини  $\Omega_m, m = \overline{1, M}$ ,  $M \leq C_N^k$ , складають  $k$ -кратну діаграму Вороного для множини  $\Omega$ , тобто таке розбиття множини  $\Omega$  на підмножини  $\Omega_1, \dots, \Omega_M$ , що:

$$\bigcup_{i=1}^M \Omega_i = \Omega; \text{mes}(\Omega_i \cap \Omega_j) = 0, \forall i \neq j, i, j = \overline{1, M};$$

$$\Omega_m = \{x \in \Omega : \forall j \in T_m \ c(x, \tau_j) < c(x, \tau_i), i \in \{1, 2, \dots, N\} \setminus T_m\},$$

де  $T_m = \{i_1^m, i_2^m, \dots, i_k^m\}$ ,  $m = \overline{1, M}$  – всі можливі  $k$ -елементні підмножини множини індексів  $\{1, 2, \dots, N\}$ .

Враховуючи відповідність набору  $k$  центрів з індексами  $i_1^m, i_2^m, \dots, i_k^m$  множині  $\Omega_m$ , для останньої надалі будемо використовувати таке

позначення:  $\Omega \left( \begin{smallmatrix} T_m \\ \tau \end{smallmatrix} \right)$ , де  $\tau \begin{smallmatrix} T_m \\ \tau \end{smallmatrix} = \{\tau_{i_1^m}, \tau_{i_2^m}, \dots, \tau_{i_k^m}\}$ . Від-

так,  $\Omega_m$  містить точки множини  $\Omega$ , для яких центри з індексами із набору  $T_m = \{i_1^m, i_2^m, \dots, i_k^m\}$  є  $k$ -найближчими сусідами.

Чисельні алгоритми розв'язання задачі (8), наведені нижче, реалізують метод проєкції узагальненого градієнтного спуску з розтягненням простору в напрямку різниці двох послідовних узагальнених градієнтів. При цьому  $j$ -ю компоненту  $N$ -вимірному вектору узагальненого градієнта

$$g_{\hat{R}}(\tau^N) = (g^{\tau_1}(\tau^N), \dots, g^{\tau_j}(\tau^N), \dots, g^{\tau_N}(\tau^N)) \quad (9)$$

функції  $\hat{R}(\tau_1, \dots, \tau_N)$  в точці  $\tau^N = (\tau_1, \dots, \tau_N)$  можна обчислювати двома способами:

1) з використанням чисельного диференціювання:

$$g^{\tau_j}(\tau^N) = (g_1^{\tau_j}(\tau^N), g_2^{\tau_j}(\tau^N), \dots, g_n^{\tau_j}(\tau^N)), \quad (10)$$

де  $s$ -та компонента обчислюється наближено за формулою:

$$g_s^{\tau_j}(\tau^N) = \frac{\hat{R}(\tau_1, \dots, (\tau_j^{(1)}, \dots, \tau_j^{(s)} + \Delta, \dots, \tau_j^{(n)}), \dots, \tau_N) - \hat{R}(\tau_1, \dots, (\tau_j^{(1)}, \dots, \tau_j^{(s)}, \dots, \tau_j^{(n)}), \dots, \tau_N)}{\Delta},$$

$$s = \overline{1, n}, \quad j = \overline{1, N};$$



$\Delta$  – деякий приріст компоненти вектора  $\tau$ , параметр методу;

2) за допомогою  $k$ -кратної діаграми Вороного. Позначимо через  $V$  фіксоване розбиття не пустої компактної множини  $\Omega$  на  $\overline{M} \leq C_N^k$  підмножин  $V_1, \dots, V_m, \dots, V_{\overline{M}}$ , котрі складають  $k$ -кратну діаграму Вороного для  $\Omega$ , причому всі  $V_i$  – неперожені, компактні. Введемо також для кожного  $i = 1, \dots, N$  функції

$$R_i(\tau^N) = \max_{T_m: i \in T_m} \max_{x \in V\left(\frac{T_m}{\tau}\right)} c(x, \tau_i). \quad (11)$$

Субградієнтну множину  $G_{R_i^V}(\hat{\tau}^N)$  функції (11) в точці  $\hat{\tau}^N$  обчислюватимемо за формулою

$$G_{R_i^V}(\hat{\tau}^N) = \overline{\text{co}} \bigcup_{x \in I(\hat{\tau}^N)} G_{c(x, \tau_i)}(\hat{\tau}^N), \quad (12)$$

де  $I(\hat{\tau}^N) = \left\{ x : x \in \bigcup_{m: i \in T_m} V_m; c(x, \hat{\tau}_i) = R_i(\hat{\tau}^N) \right\}$ ;

$G_{c(x, \tau_i)}(\hat{\tau}^N)$  – субдиференціал функції  $c(x, \tau_i)$  за  $\tau_i$  на  $E_2$  при фіксованому  $x \in V_i$ .

Узагальнений градієнт цільової функції задачі (8) має наступний вигляд:

$$g_{\hat{R}}(\tau^N) = (g_{R_1}^{\tau_1}(\tau^N), \dots, g_{R_i}^{\tau_i}(\tau^N), \dots, g_{R_N}^{\tau_N}(\tau^N)), \quad (13)$$

$i$ -та компонента якого є елементом субградієнтної множини функції  $R_i(\tau^N)$ ,  $i = \overline{1, N}$ , з (11) в точці  $\tau_i$ , що записується так:

$$G_{R_i^{\tau_i}}(\tau^N) = \overline{\text{co}} \left\{ g_c^{\tau_i}(\bar{x}, \tau^N), \bar{x} : c(\bar{x}, \tau_i) = \max_{T_m: i \in T_m} \max_{x \in V\left(\frac{T_m}{\tau}\right)} c(x, \tau_i) \right\},$$

$$i = \overline{1, N}.$$

Тут  $g_c^{\tau_i}(\bar{x}, \tau^N)$  – узагальнений градієнт функції  $c(x, \tau_i)$  в точці  $\tau^N = (\tau_1, \dots, \tau_i, \dots, \tau_N)$ ,  $\tau_i = (\tau_i^{(1)}, \tau_i^{(2)})$ ;  $T_m$  –  $m$ -та підмножина фіксованого  $k$ -кратного розбиття множини  $\Omega$ . Однак, якщо розбиття множини  $\Omega$  на підмножини не є фіксованим, і не є відомим розташування точок  $\tau_1, \dots, \tau_N$ , які генериують оптимальне покриття множини  $\Omega$ , тобто відповідне  $k$ -кратне розбиття Вороного, то функція  $\hat{R}(\tau_1, \dots, \tau_N)$  задачі (8) не є опуклою на  $\Omega^N$  внаслідок неопуклості за  $\tau^N$  функції  $\hat{r}(x, \tau^N) = \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, N} c(x, \tau_i) \lambda_i(x)$  на  $\Omega^N$ .

Хоча функція  $\hat{r}(x, \tau^N)$  не є опуклою за  $\tau^N$  за будь-якого фіксованого  $x \in \Omega$ , вона складається з опуклих частин, які відповідають мінімумам функції  $\hat{R}(\tau_1, \dots, \tau_N)$ , тому її **квазіградієнт** збігається із субградієнтом на одній з опуклих ділянок, котра примикає до даної точки.

В роботі [11] наведено умови, за яких існує монотонно спадна послідовність значень функції  $\hat{R}(\tau_1, \dots, \tau_N)$ .

Представимо наближений алгоритм розв'язання задачі про пошук радіуса  $N$  куль, що утворюють  $k$ -кратне  $c$ -кульове покриття множини  $\Omega = [0, 1] \times [0, 1]$ , а також алгоритм розв'язання задачі про мінімальне  $k$ -кратне  $c$ -кульове покриття з розміщенням центрів куль. В алгоритмі вектор  $\tau^N$  позначатимемо  $\tau$ .

Запропоновано два підходи.

**Перший підхід** реалізує математичну модель задачі, сформульовану наступним чином: нехай  $\bar{I}$  – множина всіх комбінацій з  $N$  елементів по  $k$  елементів, тобто множина всіх можливих індексів  $k$  центрів, зони спостереження яких можуть водночас покривати будь-яку точку множини  $\Omega$ . Позначимо  $I_m$  – елемент множини  $\bar{I}$ . Очевидно,  $|\bar{I}| = C_N^k$ . Потрібно знайти величину

$$R = \max_{x \in \Omega} \min_{I_m \in \bar{I}} \max_{i \in I_m} c(x, \tau_i). \quad (14)$$

Для реалізації (14)  $\Omega$  покривається прямокутною сіткою. Якщо  $\tilde{\Omega}$  – множина всіх вузлів сітки, то наближене значення шуканого радіусу обчислюється за наступною формулою:  $\tilde{R} = \max_{x \in \tilde{\Omega}} \min_{I_m \in \bar{I}} \max_{i \in I_m} c(x, \tau_i)$ .

**Другий підхід** розроблений з використанням алгоритмів сортування масиву відстаней від фіксованою точки  $x \in \Omega$  до заданих центрів  $(\tau_1, \dots, \tau_N)$  і реалізується за допомогою наступного алгоритму.

#### Алгоритм 1

**Ініціалізація.**  $\Omega$  покриваємо прямокутною сіткою з кроками  $\Delta_j, j=1, 2$ ; нехай  $\tilde{\Omega}$  – множина вузлів сітки. Задаємо розташування центрів  $(\tau_1, \dots, \tau_N)$ .

**Крок 1.** Для кожної точки  $x$  сітки  $\tilde{\Omega}$  будемо масив відстаней до заданих центрів:  $D(x) = (c(x, \tau_1), c(x, \tau_2), \dots, c(x, \tau_N))$ .

**Крок 2.** Сортуємо  $D(x)$  за зростанням елементів для  $x \in \tilde{\Omega}$ .

**Крок 3.** У відсортованому масиві вибираємо елемент, який стоїть на  $k$ -му місці (з порядковим номером  $k$ ), позначимо цей елемент  $c^k(x, \tau_{i_k})$ .

**Крок 4.** Серед усіх відібраних елементів знаходимо найбільший:

$$\tilde{R} = \max_{x \in \tilde{\Omega}} c^k(x, \tau_{i_k}).$$

Отримане максимальне значення є наближеним значенням радіусу кіл з центрами в точках  $(\tau_1, \dots, \tau_N)$ , які  $k$ -кратно покривають множину  $\Omega$ .

На однакових просторових сітках чисельні алгоритми, засновані на цих підходах, пока-

зують однакові результати. Але час роботи алгоритму 1 значно менший, про що свідчить інформація, наведена в табл. 1. Тут зазначений час розв'язання задачі пошуку радіусу  $N$  куль, які складають  $k$ -кратне  $c$ -кульове покриття одиничного квадрата  $\Omega=[0,1] \times [0,1]$ , коли  $\Delta_j = 0.005, j = 1, 2$ .

Опишемо можливі варіанти чисельних алгоритмів розв'язання задачі (8) у припущенні, що множина, яка покривається, має просту структуру. За основу обрано  $g$ -алгоритм Шора розв'язування задачі недиференційованої оптимізації.

У першому варіанті для обчислення компонент субградієнта використовуються формули чисельного диференціювання (10). Другий варіант включає етап побудови  $k$ -кратної діаграми Вороного та обчислення вектору узагальненого градієнта цільової функції задачі (8) за формулами (13).

**Алгоритм 2**

*Ініціалізація.*  $\Omega$  покриваємо прямокутною сіткою з кроками  $\Delta_j, j=1, 2$ ; нехай  $\tilde{\Omega}$  – множина вузлів прямокутної сітки на множині  $\Omega$ . Задаємо крок чисельного диференціювання  $\Delta$ , початкове наближення  $\tau^{(0)} = (\tau_1^{(0)}, \dots, \tau_N^{(0)})$ .

Обчислюємо величину

$$\hat{R}(\tau^{(0)}) = \max_{x \in \Omega} \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, N} c(x, \tau_i^{(0)}) \lambda_i(x),$$

застосовуючи алгоритм 1.

За формулами (10) обчислюємо вектор-градієнт  $g_{\hat{R}}(\tau)$  в точці  $\tau^{(0)}$ , вибираємо початковий крок  $g$ -алгоритму  $h_0 > 0$ .

*Перший крок* алгоритму проводимо за формулою

$$\tau^{(1)} = P_{\Omega} \left( \tau^{(0)} - h_0 g_{\hat{R}}(\tau^{(0)}) \right),$$

$P_{\Omega}$  – оператор проєктування на множині  $\Omega$ .

Переходимо до другого кроку.

Нехай в результаті обчислень після  $m, m = 1, 2, \dots$ , кроків алгоритму отримано певний вектор  $\tau^{(m)} = (\tau_1^{(m)}, \dots, \tau_N^{(m)})$ .

*(m+1)-й крок* алгоритму.

1. Обчислюємо за відомими центрами  $\tau_1^{(m)}, \dots, \tau_N^{(m)}$  за допомогою алгоритму 1 величину

$$\hat{R}(\tau^{(m)}) = \max_{x \in \Omega} \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, N} c(x, \tau_i^{(m)}) \lambda_i(x).$$

2. Обчислюємо  $g_{\hat{R}}(\tau)$  за формулами (10) при  $\tau = \tau^{(m)}$ .

3. Проводимо  $(m+1)$ -й крок  $g$ -алгоритму, ітераційна формула якого має вигляд

$$\tau^{(m)} = P_{\Omega} \left( \tau^{(m)} - h_m \frac{H_{m+1} g_{\hat{R}}(\tau^{(m)})}{\sqrt{(H_{m+1} g_{\hat{R}}(\tau^{(m)}), g_{\hat{R}}(\tau^{(m)})}} \right), \quad (15)$$

де  $H_{m+1}$  – матриця розтягнення простору з коефіцієнтом  $\alpha$  в напрямку різниці двох послідовних узагальнених градієнтів, яка перераховується наступним чином:

$$H_{m+1} = H_m + (1/\alpha_2 - 1) \frac{H_m \xi_m \xi_m^T H_m}{(H_m \xi_m, \xi_m)},$$

$$\xi_m = g_{\hat{R}}(\tau^{(m)}) - g_{\hat{R}}(\tau^{(m-1)}). \quad (16)$$

Якщо внаслідок округлень розрахунків  $H_{m+1}$  втрачає властивість додатної визначеності, замінюємо її одиничною матрицею.

Крок  $h_m$  вибираємо з умови:

$$\min_{h>0} \hat{R} \left( \tau^{(m)} - h \frac{H_{m+1} g_{\hat{R}}(\tau^{(m)})}{\sqrt{(H_{m+1} g_{\hat{R}}(\tau^{(m)}), g_{\hat{R}}(\tau^{(m)})}} \right). \quad (17)$$

4. Якщо умова

$$\|\tau^{(m+1)} - \tau^{(m)}\| \leq \varepsilon, \quad \varepsilon > 0, \quad (18)$$

не виконується, то переходимо до  $(m+2)$ -го кроку, інакше – перехід до п. 5.

5. Вважаємо  $\tau_* = \tau^{(l)}$ , де  $l$  – номер ітерації, на якій виконується умова (18) завершення роботи алгоритму.

6. Обчислюємо значення мінімального радіусу покриття за формулою

$$\hat{R}(\tau_*) = \max_{x \in \Omega} \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, N} c(x, \tau_{*i}) \lambda_i(x)$$

за допомогою алгоритму 1.

**Алгоритм 2** описаний.

**Алгоритм 3**

*Ініціалізація.*  $\Omega$  покриваємо прямокутною сіткою з кроком  $\Delta_j, j=1, 2$ . Нехай  $\tilde{\Omega}$  – множина вузлів прямокутної сітки на множині  $\Omega$ . Задаємо початкове розташування центрів покриття  $\tau^{(0)} = (\tau_1^{(0)}, \dots, \tau_N^{(0)})$ .

Таблиця 1

Час роботи алгоритмів при заданих кількості центрів  $N$  і кратності покриття  $k$

Вхідні дані		Час роботи алгоритму (мс), який реалізує	
$N$	$k$	Перший підхід	Другий підхід
4	2	2,2784	0,1822
9	2	6,3777	0,3108
25	3	276,2283	0,7917

Обчислюємо величину  $R(\tau^{(0)}) = \max_{x \in \Omega} \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, N} c(x, \tau_i^{(0)}) \lambda_i(x)$  за допомогою алгоритму 1. При цьому для кожного  $i = 1, \dots, N$  обчислюємо

$$R_i(\tau^{(0)}) = \max_{T_m, i \in T_m} \max_{x \in V\left(\frac{T_m}{\tau}\right)} c(x, \tau_i^{(0)}).$$

За формулою (12) будуємо субградієнтну множину  $G_{R_i}^{\tau_i}(\tau^{(0)}) = \overline{co} \left\{ g_c^{\tau_i} \left( \bar{x}, \tau^{(0)} \right), \bar{x} : c \left( \bar{x}, \tau_i \right) = R_i(\tau_i^{(0)}) \right\}$ , де  $g_c^{\tau_i} \left( \bar{x}, \tau^{(0)} \right)$  – узагальнений градієнт функції  $c(x, \tau_i)$  в точці  $\tau^{(0)}$ . Вибираємо вектор  $g_{\bar{R}}(\tau^{(0)}) \in G_{R_i}^{\tau_i}(\tau^{(0)})$ .

Задаємо початковий пробний крок  $g$ -алгоритму  $h_0 > 0$ .

*Перший крок* алгоритму проводимо за формулою

$$\tau^{(1)} = P_{\Omega} \left( \tau^{(0)} - h_0 g_{\bar{R}}(\tau^{(0)}) \right),$$

$P_{\Omega}$  – оператор проектування на множину  $\Omega$ .

Переходимо до другого кроку.

Нехай у результаті обчислень після  $m$ ,  $m = 1, 2, \dots$ , кроків алгоритму отримано певний вектор  $\tau^{(m)} = (\tau_1^{(m)}, \dots, \tau_N^{(m)})$ .

*(m+1)-й крок* алгоритму.

1. За центрами  $\tau_1^{(m)}, \dots, \tau_N^{(m)}$  за допомогою алгоритму 1 обчислюємо

$$R(\tau^{(m)}) = \max_{x \in \Omega} \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, N} c(x, \tau_i^{(m)}) \lambda_i(x),$$

при цьому для кожного  $i = 1, \dots, N$  обчислюємо значення функції (11):

$$R_i(\tau^{(m)}) = \max_{T_m, i \in T_m} \max_{x \in V\left(\frac{T_m}{\tau}\right)} c(x, \tau_i^{(m)}).$$

2. За формулою (12) будуємо субградієнтну множину в точці  $\tau^{(m)}$ .

3. Вибираємо вектор  $g_{\bar{R}}(\tau^{(m)}) \in G_{R_i}^{\tau_i}(\tau^{(m)})$ .

4. Проводимо  $(m+1)$ -й крок  $g$ -алгоритму в  $H$ -формі за ітераційними формулами (15), (16). Крок  $h_m$  вибираємо з умови (17).

5. Якщо умова (18) не виконується, то переходимо до  $(m+2)$ -му кроку алгоритму, інакше – до п. 6.

6. Вважаємо  $\tau_* = \tau^{(l)}$ , де  $l$  – номер ітерації, на якій виконалася умова (18) завершення роботи алгоритму.

7. За допомогою алгоритму 1 обчислюємо значення мінімального радіусу покриття  $\hat{R}(\tau_*) = \max_{x \in \Omega} \min_{\lambda(x) \in \Lambda_N^k} \max_{i=1, N} c(x, \tau_{*i}) \lambda_i(x)$ .

Алгоритм 3 описаний.

Наведемо результати деяких обчислювальних експериментів з багатократного покриття одиничного квадрата з  $E_2$  у випадку, коли функція  $c(x, \tau_i)$  є евклідовою метрикою. На рис. 1 представлені результати двократного покриття одиничного квадрата 11-ма і 15-ма колами, отримані за допомогою алгоритмів 2 та 3. На рис. 2 показані трикратні оптимальні покриття за різних  $N$ .

Результати наведених нижче та інших обчислювальних експериментів з чисельного розв'язання задачі (8) свідчать про багатоекстремальність її цільової функції. За різних початкових наближень набору центрів  $\tau^{(0)} = (\tau_1^{(0)}, \dots, \tau_N^{(0)})$  можна отримати різні локальні розв'язки.

Практика розрахунків також свідчить про те, що найкращі результати багатократного покриття можна отримати, якщо спочатку на достатньо густій просторовій сітці знайти оптимальні розташування центрів за допомогою алгоритму 2, а потім уточнити координати цих центрів за допомогою алгоритму 3.

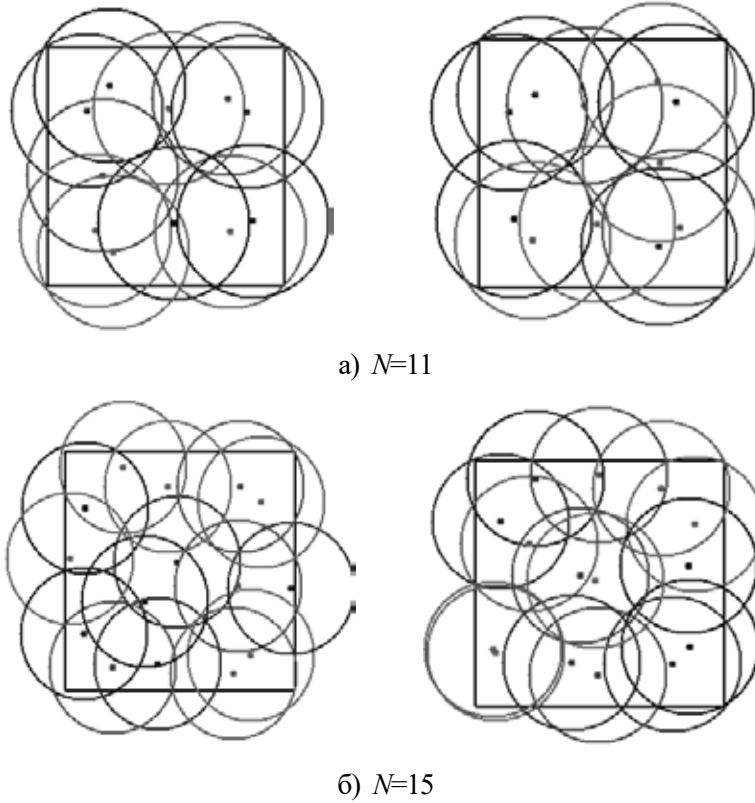
В дослідженні [12] наведено порівняльний аналіз результатів розв'язання багатьох тестових задач оптимального двократного покриття одиничного квадрата відомих в науковій літературі і отриманих за допомогою описаного вище алгоритму із застосуванням формул чисельного диференціювання, а також з використанням багатократних діаграм Вороного.

На рис. 3 наведено оптимальні триплексне і дуплексне розбиття множини  $\Omega = [0, 1] \times [0, 1]$  з розміщенням  $N=17$  і 15 центрів відповідно. Обчислений при цьому радіус покриття дорівнює:  $R=0.33232$  і  $R=0.2961$ .

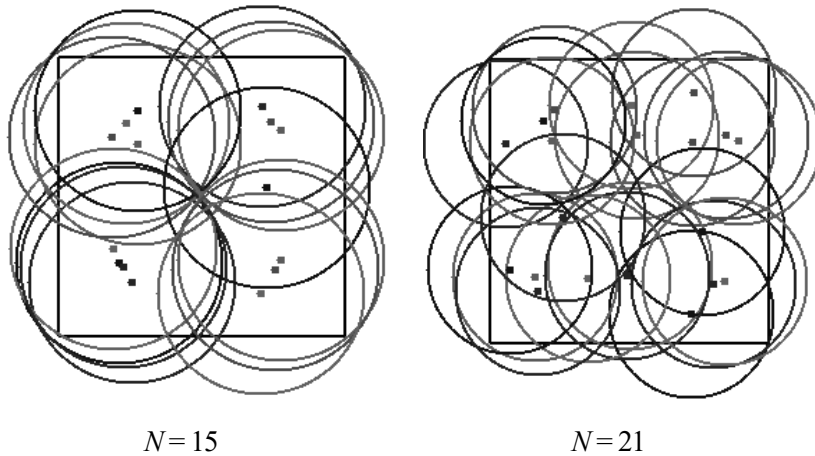
На рис. 4 ліворуч подано дуплексне розбиття квадрата з оптимально розташованими тринадцятьма центрами із зазначенням радіусу відповідного двократного покриття; праворуч представлено мінімальне двократне кульове покриття квадрату для 13 центрів [13]. Обчислений радіус двократного покриття для наведених прикладів дорівнює  $R=0.2961$ ,  $R=0.2911$  відповідно.

Якісний аналіз результатів обчислювальних експериментів дозволяє виявити переваги та недоліки алгоритмів. По-перше, наближені значення радіусу покриття і центрів, що його генерує, залежать від вихідних даних та параметрів алгоритму – початкового наближення координат центрів, величини кроку просторової сітки, кроку чисельного диференціювання при обчисленні компонент узагальненого градієнта.

В роботі [13] наведені постановки і методи розв'язання неперервних задач багатократного



**Рис. 1. Двократні оптимальні покриття одиничного квадрата, отримані із застосуванням: ліворуч – чисельного диференціювання; праворуч – двократної діаграми Вороного**

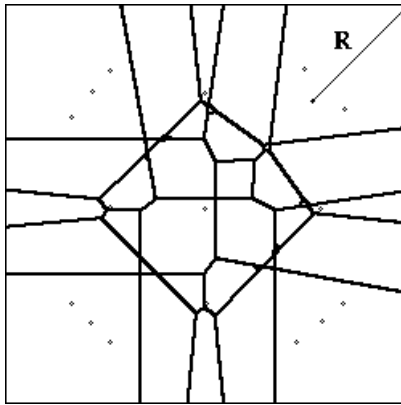


**Рис. 2. Трикратне покриття одиничного квадрата,  $\|x\| = \|x\|_2$**

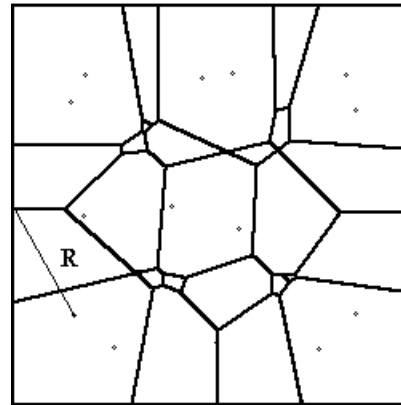
покриття з обмеженнями, які відповідають за таке розміщення центрів, при якому кожні два з них перебували б на відстані, не меншій за  $\sigma$ . Крім того, розглянуто випадки вимог, аби центри були розміщені в області якомога рівномірно, або жоден з них не потрапив в деяку заборонену зону. При цьому введення доданку

у функціонал, котрий відповідає за неможливість «злипання» центрів куль, що утворюють покриття, називається регуляризацією задачі (функціоналу) про мінімальне  $k$ -кратне кульове покриття.

**Висновки.** Представлені математичні моделі задач оптимального розміщення ава-

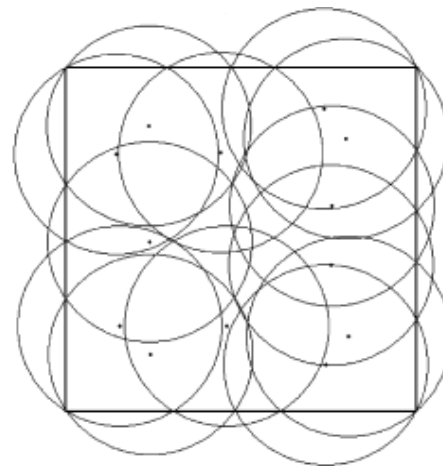
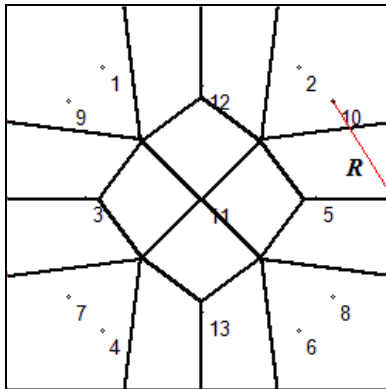


$N = 17, k = 3$



$N = 15, k = 2$

**Рис. 3. Оптимальне  $k$ -кратне розбиття квадрату з розміщенням  $N$  центрів і радіус відповідного мінімального покриття**



**Рис. 4. Дуплексне розбиття  $\Omega$  з оптимальним розміщенням 13 центрів і відповідне мінімальне покриття**

рійно-рятувальних об'єктів з метою мінімізації часу надання допомоги навіть найвіддаленішому користувачу заданого регіону, які є неперервними задачами багатократного покриття обмеженої множини з простору  $E_2$   $s$ -кулями найменшого радіусу за певних умов розміщення центрів, що утворюють покриття. Описано алгоритми розв'язання задачі багатократного кульового покриття, розроблені на основі субградієнтних методів негладкої оптимізації із залученням апарату штрафних функцій та елементів теорії неперервних задач оптимального розбиття множин.

Досліджено властивості розглянутих задач про багатократне оптимальне  $s$ -кульове покриття множини  $\Omega$ , встановлено їх зв'язок із задачами

оптимального мультиплексного розбиття, що дозволяє в тих випадках, коли це доцільно, зводити розв'язання задачі про покриття до задачі розбиття. На прикладах розв'язання неперервних задач багатократного кульового покриття множин, а також задач пошуку розбиття відповідної кратності цих множин, продемонстрована можливість визначення радіусу покриття у процесі розв'язання неперервних лінійних задач мультиплексного розбиття множин. А для задачі мультиплексного розбиття множини з розміщенням центрів запропоновано критерій якості розбиття, який дозволяє отримувати таке розташування центрів, яке збігається з набором центрів куль мінімального радіусу, що багатократно покривають задану множину.

**ЛІТЕРАТУРА:**

1. Dhamala T.N. A survey on models and algorithms for discrete evacuation planning network problems / T.N. Dhamala // *Journal of industrial and management optimization*, 2015. Vol. 11(1). P. 265–289. [Online]. URL: <https://doi.org/10.3934/jimo.2015.11.265>
2. Alghanmi N. A Survey of Location-Allocation of Points of Dispensing During Public Health Emergencies / N. Alghanmi, R. Alotaibi, S. Alshammari, A. Alhothali, O. Bamasag, K. Faisal // *Front. Public Health*, 2022. 10:811858. doi: 10.3389/fpubh.2022.811858
3. Koriashkina L.S. Continuous problems of optimal multiplex-partitioning of sets without constraints and solving methods / L.S. Koriashkina, A.P. Cherevatenko // *Journal of Computational & Applied Mathematics*. 2015. № 2 (119). P. 15 – 32.
4. Koriashkina L. The continuous problems of the optimal multiplex partitioning an application of sets / L. Koriashkina, A. Cherevatenko, O. Mykhalova // *Power Engineering and Information Technologies in Technical Objects Control – Pivnyak, Beshta & Alekseyev (eds)*. – Taylor & Francis Group, London. 2016. P. 233 – 239.
5. Антошкін О.А. Узагальнена математична модель задачі покриття області ідентичними колами та її основні реалізації / О.А. Антошкін, О.В. Панкратов // *Системи обробки інформації*. 2019. № 1(156). С. 44-49. DOI: 10.30748/soi.2019.156.06
6. Eliseu J. Araújo. A mathematical model for the coverage location problem with overlap control / Eliseu J. Araújo, Antônio A. Chaves, Luiz A.N. Lorena // *Computers & Industrial Engineering*. August 2020. Vol. 146. <https://doi.org/10.1016/j.cie.2020.106548>
7. Yakovlev S. Modeling and Simulation of Coverage Problem in Geometric Design Systems / S. Yakovlev, O. Kartashov, V. Komyak, S. Shekhovtsov, O. Sobol, I. Yakovleva // *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, Polyana, Ukraine, 2019. P. 20-23. DOI: 10.1109/CADSM.2019.8779303
8. Лубенець Д.Є. Системний аналіз і оптимізація розподілу матеріальних ресурсів в ієрархічних транспортно-логістичних системах / Д.Є. Лубенець, Л.С. Коряшкіна // *Молодь: наука та інновації: матеріали XI Міжнародної науково-технічної конференції студентів, аспірантів та молодих вчених, Дніпро, 22–24 листопада 2023 року: у 2-х т. / Національний технічний університет «Дніпровська політехніка». Дніпро : НТУ «ДП», 2023. Том 2. С. 21 – 22.*
9. Zhao R. Discrete-continuous model for facility location problem with capacity-cost relation constraints / R. Zhao, Y. Xiao, R. Luo, R. Yang, S. Zhou, S. Zhang // *Computers & Industrial Engineering*, November 2023. Vol. 185. <https://doi.org/10.1016/j.cie.2023.109661>
10. Dziuba, S., & Bulat, A., & Koriashkina, L., & Blyuss, B. Discrete-Continuous Model of the Optimal Location Problem for the Emergency Logistics System. URL: <https://ssrn.com/abstract=4401341>
11. Kiseleva E.M. Constructive algorithms for solution problems of multiple continuous coverage / E.M. Kiseleva, L.S. Koriashkina, A.A. Mikhalova // *System technologies*. N4(93). Dnipropetrovsk, 2014. P. 3 – 16.
12. Kiseleva E.M. Continuous problem of multiple ball covering with restrictions and the method of its solving / E.M. Kiseleva, L.S. Koriashkina, A.A. Mikhalova // *System technologies*. N 1(96). Dnipropetrovsk, 2015. P. 165 – 179.
13. Коряшкіна Л.С. Застосування методів оптимального розбиття множин до неперервних задач багатократного покриття / Л.С. Коряшкіна, О.О. Михальова, В.І. Навоєнко // *Питання прикладної математики і математичного моделювання. Збірник наук. праць. Дніпропетровськ, 2014. С. 141 – 154.*

**REFERENCES:**

1. Dhamala T.N. (2015). A survey on models and algorithms for discrete evacuation planning network problems / T.N. Dhamala // *Journal of industrial and management optimization*. Vol. 11(1). P. 265–289. [Online]. Retrieved from <https://doi.org/10.3934/jimo.2015.11.265>
2. Alghanmi N. (2022). A Survey of Location-Allocation of Points of Dispensing During Public Health Emergencies / N. Alghanmi, R. Alotaibi, S. Alshammari, A. Alhothali, O. Bamasag, K. Faisal // *Front. Public Health*. 10:811858. doi: 10.3389/fpubh.2022.811858
3. Koriashkina L.S. (2015). Continuous problems of optimal multiplex-partitioning of sets without constraints and solving methods / L.S. Koriashkina, A.P. Cherevatenko // *Journal of Computational & Applied Mathematics*. № 2 (119). P. 15 – 32.
4. Koriashkina L. (2016). The continuous problems of the optimal multiplex partitioning an application of sets / L. Koriashkina, A. Cherevatenko, O. Mykhalova // *Power Engineering and Information Technologies in Technical Objects Control – Pivnyak, Beshta & Alekseyev (eds)*. – Taylor & Francis Group, London. P. 233 – 239.

5. Antoshkin O.A. (2019). Uzahalnena matematychna model zadachi pokryttia oblasti identychnymy kolamy ta yii osnovni realizatsii [A generalized mathematical model of the problem of covering an area with identical circles and its main implementations] / O.A. Antoshkin, O.V. Pankratov // *Systemy obrobky informatsii*. № 1(156). S. 44-49. DOI: 10.30748/soi.2019.156.06
6. Eliseu J. Araújo. (2020). A mathematical model for the coverage location problem with overlap control / Eliseu J. Araújo, Antônio A. Chaves, Luiz A.N. Lorena // *Computers & Industrial Engineering*. August 2020. Vol. 146. <https://doi.org/10.1016/j.cie.2020.106548>
7. Yakovlev S. (2019). Modeling and Simulation of Coverage Problem in Geometric Design Systems / S. Yakovlev, O. Kartashov, V. Komyak, S. Shekhovtsov, O. Sobol, I. Yakovleva // 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Polyana, Ukraine. P. 20-23. DOI: 10.1109/CADSM.2019.8779303
8. Lubenets D.Ie. (2023). Systemnyi analiz i optymizatsiia rozpodilu materialnykh resursiv v iierarkhichnykh transportno-lohistychnykh systemakh [System analysis and optimization of the distribution of material resources in hierarchical transport and logistics systems] / D.Ie. Lubenets, L.S. Koriashkina // *Molod: nauka ta innovatsii: materialy Khl Mizhnarodnoi naukovo-tekhnichnoi konferentsii studentiv, aspirantiv ta molodykh vchenykh, Dnipro, 22–24 lystopada 2023 roku: u 2-kh t. / Natsionalnyi tekhnichnyi universytet «Dniprovska politekhnika»*. Dnipro : NTU «DP». Tom 2. S. 21 – 22.
9. Zhao R. (2023). Discrete-continuous model for facility location problem with capacity-cost relation constraints / R. Zhao, Y. Xiao, R. Luo, R. Yang, S. Zhou, S. Zhang // *Computers & Industrial Engineering*, November 2023. Vol. 185. <https://doi.org/10.1016/j.cie.2023.109661>
10. Dziuba, S., & Bulat, A., & Koriashkina, L., & Blyuss, B. Discrete-Continuous Model of the Optimal Location Problem for the Emergency Logistics System. Retrieved from <https://ssrn.com/abstract=4401341>
11. Kiseleva E.M. (2014). Constructive algorithms for solution problems of multiple continuous coverage / E.M. Kiseleva, L.S. Koriashkina, A.A. Mikhalova // *System technologies*. N 4(93). Dnipropetrovsk. P. 3 – 16.
12. Kiseleva E.M. (2015). Continuous problem of multiple ball covering with restrictions and the method of its solving / E.M. Kiseleva, L.S. Koriashkina, A.A. Mikhalova // *System technologies*. N 1(96). Dnipropetrovsk. P. 165 – 179.
13. Koriashkina L.S. (2014). Zastosuvannia metodiv optymальноho rozbyttia mnozhyn do neperervnykh zadach bahatokratnoho pokryttia [Application of methods of optimal partitioning of sets to continuous problems of multiple coverage] / L.S. Koriashkina, O.O. Mykhalova, V.I. Navoienko // *Pytannia prykladnoi matematyky i matematychnoho modeliuвання*. Zbirnyk nauk. prats. Dnipropetrovsk. S. 141 – 154.

**UDC 004.94 (072)**

**DOI <https://doi.org/10.32782/IT/2023-4-4>**

**Ivan LAKTIONOV**

*Doctor of Technical Sciences, Associate Professor, Professor at the Department of Computer Systems Software, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave, Dnipro, Ukraine, 49005, [Laktionov.I.S@nmu.one](mailto:Laktionov.I.S@nmu.one)*

**ORCID:** 0000-0001-7857-6382

**Scopus Author ID:** 57194557735

**Oleksandr ZHABKO**

*Postgraduate student in the speciality 123 Computer Engineering, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave, Dnipro, Ukraine, 49005, [Zhabko.O.S@nmu.one](mailto:Zhabko.O.S@nmu.one)*

**ORCID:** 0009-0002-7996-9115

**Grygorii DIACHENKO**

*Candidate of Technical Sciences, Associate Professor at the Department of Electric Drive, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave, Dnipro, Ukraine, 49005, [Diachenko.G@nmu.one](mailto:Diachenko.G@nmu.one)*

**ORCID:** 0000-0001-9105-1951

**Scopus Author ID:** 57201252081

**Mykola PROKOPENKO**

*Senior DevOps engineer at DataArt, 7, Barykadna St, Dnipro, Ukraine, 49000, [nprokopenko@dataart.com](mailto:nprokopenko@dataart.com)*

**ORCID:** 0009-0000-9478-2311

**To cite this article:** Laktionov, I., Zhabko, O., Diachenko, G., Prokopenko, M. (2023). Obgruntuvannia vymoh do strukturno-alhorytmichnoho zabezpechennia IoT systemy monitorynhu gruntoklimatychnykh parametriv silskohospodarskykh pidpriemstv roslynnytstva [Substantiation of requirements for the structural and algorithmic organization of an IoT monitoring system for soil and climate parameters in agricultural crop enterprises]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 32–39, doi: <https://doi.org/10.32782/IT/2023-4-4>

## **SUBSTANTIATION OF REQUIREMENTS FOR THE STRUCTURAL AND ALGORITHMIC ORGANIZATION OF AN IOT MONITORING SYSTEM FOR SOIL AND CLIMATE PARAMETERS IN AGRICULTURAL CROP ENTERPRISES**

**Relevance.** As one of the key sectors of the global economy, agriculture plays a decisive role in shaping sustainable economic development and ensuring food security worldwide. Achieving satisfactory indicators of food security and the export potential of agricultural products is a domestic priority for economic recovery and sustainable development. Accordingly, the development and implementation of modern digital technologies in agriculture becomes a strategically important task. The utilization of cutting-edge technologies for the agricultural sector, coupled with the improvement of cultivation and processing methods for agricultural crops, is a key element in ensuring the resilience and efficiency of the industry. **The main aim** is to conduct a critical analysis of existing solutions in the field of information monitoring for agricultural purposes. This analysis will allow the development of an optimized structural and algorithmic organization for the computerized system. This system is designed to implement effective monitoring and decision support for managing agricultural processes. **The research object** is existing approaches and methods for organizing monitoring systems in the agricultural sector to collect, transmit, and process measurement data. **The research subject** is information and communication software and hardware solutions in the field of computerized monitoring for agricultural purposes. **Conclusions.** It has been substantiated that the development and implementation of conceptual software and hardware solutions in information technology for monitoring soil and climatic parameters when cultivating agricultural crops in open-field conditions are currently a relevant scientific and applied task. The set of requirements for the structural and algorithmic organization of the IoT technology for soil and climatic monitoring has been substantiated. An analysis and selection of technologies on which the investigated hardware and software solution will be based have been carried out. The overall and detailed structural and algorithmic organization of the investigated IoT system for agrotechnical monitoring has been developed, and priority directions for further research have been established.

**Key words:** Internet of Things, monitoring, architecture, wireless sensor network, agriculture.



**Іван ЛАКТИОНОВ**

д-р техн. наук, доц., проф. кафедри програмного забезпечення комп'ютерних систем, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, Laktionov.I.S@ntu.one

ORCID: 0000-0001-7857-6382

Scopus Author ID: 57194557735

**Олександр ЖАБКО**

здобувач вищої освіти за освітньо-науковим рівнем «Доктор філософії» за спеціальністю 123 Комп'ютерна інженерія, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, Zhabko.O.S@ntu.one

ORCID: 0009-0002-7996-9115

**Григорій ДЯЧЕНКО**

канд. техн. наук, доцент кафедри електропривода, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, Diachenko.G@ntu.one

ORCID: 0000-0001-9105-1951

Scopus Author ID: 57201252081

**Микола ПРОКОПЕНКО**

Senior DevOps інженер у компанії DataArt, вул. Барикадна, 7, м. Дніпро, Україна, 49000, prokopenko@dataart.com

ORCID: 0009-0000-9478-2311

**Бібліографічний опис статті:** Лактіонов, І., Жабко, О., Дяченко, Г., Прокопенко, М. (2023). Обґрунтування вимог до структурно-алгоритмічного забезпечення IoT системи моніторингу ґрунтокліматичних параметрів сільськогосподарських підприємств рослинництва. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 32–39, doi: <https://doi.org/10.32782/IT/2023-4-4>

**ОБґРУНТУВАННЯ ВИМОГ ДО СТРУКТУРНО-АЛГОРИТМІЧНОГО  
ЗАБЕЗПЕЧЕННЯ ІОТ СИСТЕМИ МОНІТОРИНГУ ҐРУНТОКЛІМАТИЧНИХ  
ПАРАМЕТРІВ СІЛЬСЬКОГОСПОДАРСЬКИХ ПІДПРИЄМСТВ РОСЛИНИЦТВА**

**Актуальність.** Сільське господарство, як одна з ключових галузей світової економіки, має визначальне значення у формуванні сталого економічного розвитку та забезпеченні продовольчої безпеки у світі. Досягнення задовільних показників продовольчої безпеки та експортного потенціалу сільськогосподарської продукції є одним із вітчизняних пріоритетів економічного відновлення та сталого розвитку. Відповідно до цього розробка та впровадження сучасних цифрових технологій у сільське господарство стає стратегічно важливим завданням. Адже, використання новітніх технологій для цілей аграрної галузі, водночас з удосконаленням методів вирощування та обробки сільськогосподарських культур, є ключовим елементом забезпечення стійкості та ефективності галузі. **Метою роботи** є критичний аналіз існуючих рішень у сфері інформаційного моніторингу агротехнічного призначення, що дозволить розробити оптимізовану структурно-алгоритмічну організацію комп'ютеризованої системи, яка спрямована на реалізацію ефективного моніторингу та підтримки прийняття рішень щодо керування аграрними процесами. **Об'єктом дослідження** є існуючі підходи та методи організації систем моніторингу в агросекторі з метою збору, передачі та обробки вимірjувальних даних. **Предметом дослідження** є інфокомунікаційні програмно-апаратні рішення в галузі комп'ютеризованого моніторингу сільськогосподарського призначення. **Висновки.** Доведено, що розробка і впровадження концептуальних програмно-апаратних рішень інформаційних технологій для моніторингу ґрунтокліматичних параметрів під час вирощування сільськогосподарських культур в умовах відкритого ґрунту є актуальною науково-прикладною задачею у теперішній час. Обґрунтовано сукупність вимог до структурно-алгоритмічної організації IoT технології ґрунтокліматичного моніторингу, здійснено аналіз і вибір технологій, на яких базуватиметься досліджуване апаратно-програмних рішення. Обґрунтовано загальну та деталізовану структурно-алгоритмічну організацію досліджуваної IoT системи агротехнічного моніторингу та встановлено пріоритетні напрямки подальших досліджень.

**Ключові слова:** IoT-технологія, моніторинг, архітектура, бездротова сенсорна мережа, сільське господарство.

**The relevance of the scientific and applied research task.** As one of the key sectors of the global economy, agriculture plays a crucial role in shaping sustainable economic development and ensuring food security worldwide. The relevance of agriculture is directly linked to its impact on the social and economic aspects of society. Agricultural activities serve as the primary source of food production, a vital factor in providing the population with essential food products.

In contemporary times, as the world faces challenges such as global climate change and an increasing population, the significance of agriculture becomes even more pronounced. Achieving food security becomes a priority, and the integration of modern technologies into agriculture becomes a strategically important task. The use of cutting-edge technologies in agronomic practices, coupled with advancements in cultivation and processing methods of agricultural crops, is a key element in ensuring the resilience and efficiency of the industry.

The necessity to implement modern information and computer technologies in agriculture is also driven by the need to optimize production processes. Digitization and innovations contribute to increased productivity and quality of agricultural products, ensuring high competitiveness in both domestic and global markets.

The integration of modern technologies into agriculture is not only a national necessity but also a global trend that determines the effectiveness and competitiveness of Ukraine in the world market. It becomes a strategic direction of development, combining economic, technical, ecological, and social aspects.

The contemporary development of agriculture is accompanied by several important trends that define not only technological shifts but also strategic directions in the industry.

Firstly, the significant impact on agriculture is exerted by global digitization and the intellectualization of production processes. The use of modern information technologies, sensors, artificial intelligence systems, and big data allows the optimization of production processes, resource management, and the enhancement of farming efficiency.

The second important trend is sustainability and ecological resilience. Agriculture is becoming increasingly oriented towards environmental conservation. The use of environmentally friendly approaches to cultivation, the development of organic farming, and the implementation of resource-saving technologies become essential in meeting consumer demands and addressing global environmental challenges.

The third trend is the globalization of markets and increased competition. Agriculture is part of a global supply chain that requires a high level of competitiveness. Improving product quality, standardization, and the development of new markets are strategic objectives for the agricultural sector.

Additionally, a significant trend is the development and practical implementation of the conceptual principles of 'Agriculture 4.0'. The integration of the Internet of Things, robotics, as well as hardware and software solutions into the digitization of production processes, transforms agriculture, making it more efficient and high-tech.

**Aim and objectives of the article.** The main aim of the article is to analyse and synthesize approaches to the development of the structural and algorithmic organization of IoT technologies for implementing comprehensive digital monitoring of soil and climatic parameters of open-field crop farms. To achieve the set aim, the following objectives need to be met:

- critical analysis and logical generalization of existing approaches to the construction of information technologies and computerized systems for agrotechnical purposes;
- identification and study of the most effective architectural solutions in the construction of information technologies for agrotechnical monitoring;
- development of a structural diagram and algorithms for the functioning of the investigated IoT technology;
- substantiation of further directions for priority research in the declared subject area.

**Critical analysis and logical generalization of recent research, publications, and scientific and technical developments.** Based on the analysis of sources [1–3], the following aspects requiring development in the domestic and global agricultural sectors can be highlighted:

1. Strategies for agricultural digitization:
  - investigation of specific strategies used in different countries for the development, improvement, and implementation of digital technologies in agriculture;
  - overview of programs and initiatives aimed at supporting the digital transformation of the agricultural sector.
2. Regulations and standards:
  - research the regulatory environment regulating the use of digital technologies in agriculture;
  - review standards for data exchange between various agricultural systems.
3. Legislation:
  - reviewing legislative initiatives aimed at supporting the integration of digital technologies into agriculture;

- studying legal aspects related to data use in agriculture and agrotechnologies;
- international research and collaboration;
- analysis of projects and research conducted internationally for the development of digital agriculture;
- considerations of opportunities for international cooperation in the digitization of the agricultural sector.

4. Innovations and start-ups:

- investigation of the role of innovations and start-ups in the implementation of digital technologies in agriculture;
- analysis of research works and innovative solutions being developed in this field.

5. Efficiency and challenges:

- study the results of implementing digital technologies in agriculture and their impact on productivity and the sustainability of agricultural production;
- analysis of challenges such as cybersecurity, data confidentiality, and technology accessibility for agricultural productions.

All these trends define a new dimension of agricultural development, requiring a combination of traditional methods with innovative technologies to ensure sustainable and efficient farming [1].

The main advantages of applying digital and information technologies include the following [2]:

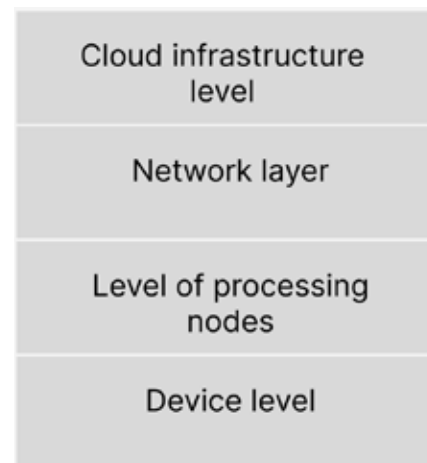
- cost savings through efficient resource utilization;
- optimized crop monitoring and minimization of crop losses due to decrease of the destabilizing factors influence such as diseases or aggressive climatic conditions;
- predictive planning of agricultural activities.

The aforementioned features are part of the global trend in agricultural development. Similar stages correspond to the domestic strategy for the development of the agricultural sector. It is important to emphasize that the lack of uniform standards and precise solutions complicates the implementation of modern digitization strategies in agriculture. This problem hinders the integration of innovations and reduces their effectiveness, necessitating the search for scientifically grounded solutions. For the successful development of the agricultural sector and to ensure its sustainable functioning, active efforts are needed to create unified standards and develop precise strategies for implementing technological innovations, which will respond to modern challenges and contribute to the creation of a competitive and sustainable domestic agricultural sector [3].

In terms of the possible architecture of IoT systems, various approaches to monitoring crops

using innovative Internet of Things (IoT) technologies can be implemented for the optimization of agriculture. Through the use of high-precision sensors, data collection, and analysis, comprehensive monitoring of plant growth can be realized, supporting decision-making regarding the rational use of resources and the control of soil quality and fertilizer levels. The diversity of approaches lies in the use of various types of sensors, graphical images obtained from drones or satellites, and automated data collection systems that are integrated to create comprehensive solutions aimed at optimizing processes in agriculture [2].

The Internet of Things (IoT) is based on a hierarchical architecture that includes multiple levels for effective data management and processing. A typical IoT architecture includes the following levels, as shown in Figure 1 [2].



**Fig. 1. Generalized Architecture of IoT Systems for Agrotechnical Purposes**

The main advantages of the implementation of software and hardware solutions of IoT in agriculture are described in Table 1. The results of the comparison, which are given in Table 1, are the result of a logical generalization at the qualitative level of current global developments and used technologies for agrotechnical purposes [4].

Based on the conducted analysis, it can be concluded that ground monitoring, using wireless sensor networks, proves to be the most reasonable approach for optimizing processes in crop farming through the implementation of the conceptual IoT principles based on hierarchical structural and functional organization (see Fig. 1).

This method not only ensures high measurement accuracy but also allows obtaining real-time information about the condition of plants, soil, and other aspects of agriculture. This approach also considers the advantages of designing computerized systems, taking into account their mobil-

Table 1

**The results of comparing IoT monitoring technologies**

Parameter	Computerized Systems	IoT technologies
Support for collective farming	Almost absent	High level of support
Logistic and quality tracing of food production	Average level	High level
Direct interaction with consumers	N/A	Available
Harvest monitoring	Available	Available
Automated systems	Available	Available

Table 2

**The results of the comparison of agrotechnical monitoring technologies**

Parameter	Classical Computerized	Satellite	Ground Stationery	Ground Wireless
Coverage	Low	High	High	High
Integration	Low	High	Medium	High
Ease of Use	Medium	High	High	High
Mobility	Low	High	High	High
Resource Conservation	Medium	Medium	Medium	Medium
Energy Efficiency	Low	Medium	Medium	Medium
Maintenance Complexity	Medium	High	Medium	Low
Economic Efficiency	Low	Low	Medium	High

ity, scalability, and benefits for small and medium-sized businesses in the digital agriculture sector by realizing the following trends: precision, responsiveness, mobility, scalability, and investment attractiveness for small and medium-sized enterprises.

Therefore, this approach enables the creation of flexible, mobile, and scalable solutions that meet current conditions for projects in the field of digital agriculture, especially for small and medium-sized enterprises [4].

The next step in the research is to carry out a comparative analysis of four types of

agrotechnical monitoring systems: classical computerized, satellite-based, ground-based stationary, and ground-based wireless. During the comparative analysis, the following criteria were taken into account: accuracy of data collection, reliability of data transmission, coverage area, implementation and operating costs, the ability to simultaneously detect various physical and chemical parameters (humidity, temperature, chemical composition of the soil, etc.), energy consumption, protection against external influences, system response time, weather resistance, scalability, and ease of integration with other technological systems [5]. The qualitative results of the comparative analysis are presented in Table 2.

Comparing the aforementioned agrotechnical monitoring technologies, it can be concluded that ground-based wireless monitoring is the most optimal in terms of utilization based on integral characteristics. Specifically, such systems offer

high accuracy, mobility, a scalable coverage area, the ability to detect multiple parameters simultaneously, resistance to weather conditions, low energy consumption, and attractive investment in implementation and ongoing maintenance.

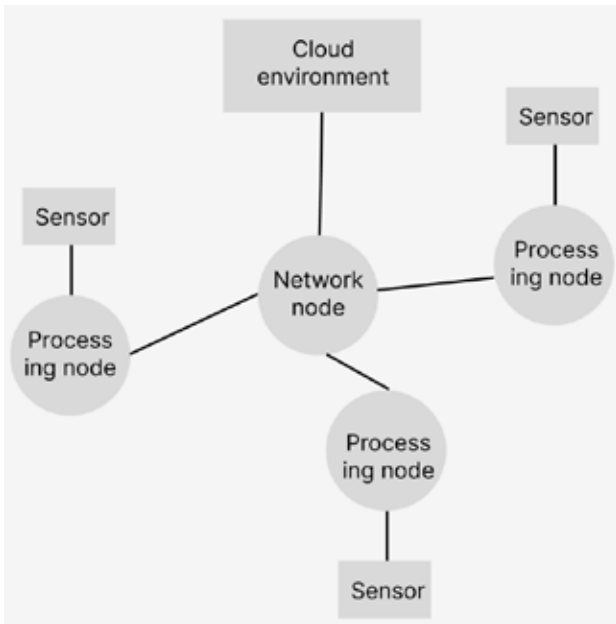
It is also worth noting that the ground-based wireless sensor networks demonstrate high efficiency in tracking environmental parameters and respond to changes in real-time, making them an effective tool for resource management to enhance crop yields. Such an approach can also be a beneficial investment solution, ensuring optimal resource utilization and improved productivity in agriculture [6].

**Research results.** Based on the results of the analysis of monitoring technologies, a generalized structural diagram of the soil and climatic parameters monitoring system during the cultivation of agricultural crops has been developed. The proposed structural diagram is based on the technology of wireless sensor networks, as shown in Figure 2.

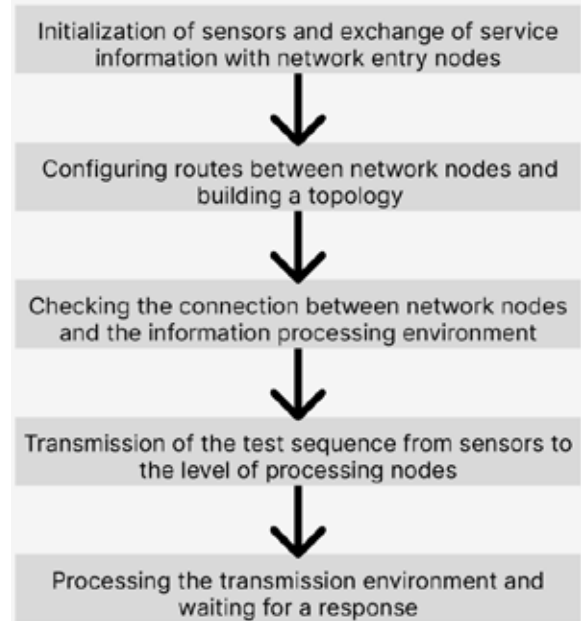
Infrastructure for information processing is a server environment designed for receiving, processing, and storing data, providing access to information for users with different rights, as well as updating network nodes and sensors as needed.

Sensors are end devices of the technology that, during interaction with the surrounding environment, transmit data to the network and exchange service information with network nodes.

Further specification of requirements for the investigated IoT technology necessitates the



**Fig. 2. Generalized structural diagram of the infocommunication network for wireless monitoring of soil and climatic parameters**



**Fig. 3. Algorithm for the initial setup of the investigated IoT technology**

establishment of a standard for further research and system development. To achieve this, two main types of interaction algorithms are proposed [7]:

1. Establishing interaction (processing and transmission) of data between all levels of the architecture.
2. Configuring the algorithm of actions in case of emergencies.

The initial setup algorithm is triggered in the event of the first installation of the technology or its complete reboot. In such a case, the sequence of actions after the connection and activation of all system components is shown in Figure 3.

Also, taking into account the modern regulations in the field of open-field crop production regarding the soil and climate parameters influencing the efficiency of growing agricultural crops [8], as well as taking into account the technical and functional characteristics of wireless information and communication technologies, the detailed structural diagram of the IoT technology of agrotechnical monitoring has been substantiated, as shown in Figure 4 [9].

In the specified diagram (see Fig. 4), information processing occurs at the processing nodes (ZigBee) to perform preliminary analysis of the information and unload the network at the next level of data collection. Subsequently, the data passes to processing centres, which are optimally located in cloud infrastructure in terms of providing a high level of support and servicing dedicated resources of the processing system [8, 10–12].

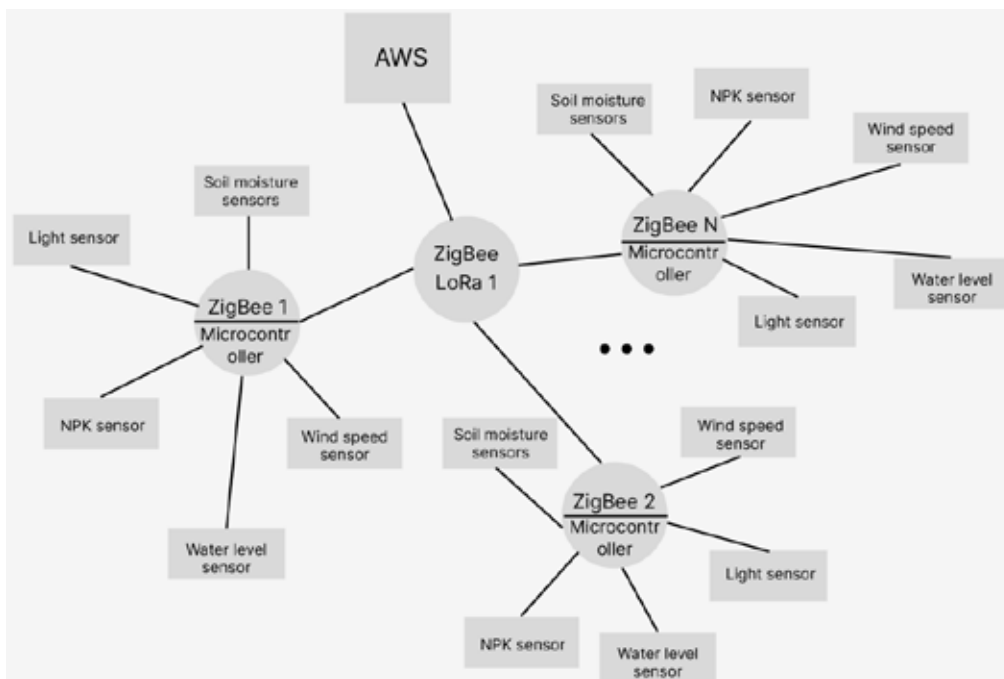
Therefore, based on the conducted research, it has been established that the indicated structural and algorithmic organization of the IoT technology for monitoring soil and climatic parameters during the cultivation of crops correlates with general conceptual trends in the field of information technology in terms of precision, speed, scalability, adaptability, and the complexity of aggregating and transforming measurement data.

**Priority directions for further research.**

Based on the analysis and formulation of key requirements, the next steps involve addressing three crucial tasks:

1. Development and modelling of a network capable of qualitatively and reliably performing the functions of collecting and processing measurement information regarding a set of soil and climatic parameters at multiple hierarchical levels of the system.
2. Development of an infocommunication standard that unifies the general concept of the architecture and demonstrates the functioning aspects of monitoring systems at all stages of system operation.
3. Experimental testing of the system and verification of algorithms under emergency conditions of system operation.

**Conclusions.** It has been proven that the development and implementation of conceptual hardware and software solutions of information technologies for monitoring soil and climatic parameters during the cultivation of crops in open-field conditions is a relevant scientific and applied



**Fig. 4. Detailed structural diagram of the infocommunication network for wireless monitoring of soil and climate parameters**

task at present. A set of requirements for the structural and algorithmic organization of the IoT soil and climatic monitoring technology has been substantiated, as well as an analysis and selection of technologies on which the investigated hardware and software solutions will be based have been carried out. The general and detailed structural

and algorithmic organization of the investigated IoT agricultural monitoring system has been substantiated. As a result of this scientific work, basic principles, and directions for the development of IoT technologies for effective monitoring of soil and climatic conditions on open-field crop farms have been identified.

#### BIBLIOGRAPHY:

1. Karunathilake E.M.B.M., Le A.T., Heo S., Chung Y.S., Mansoor S. The Path to Smart Farming: Innovations and Opportunities in Precision Agriculture. Agriculture. 2023. Vol. 13 (8). P. 1–26.
2. Farooq M.S., Riaz S., Abid A., Umer T., Zikria Y.B. Role of IoT Technology in Agriculture: A Systematic Literature Review. Electronics. 2020. Vol. 9 (2). P. 1–41.
3. National Economic Strategy 2030. URL: nes2030.org.ua/ (accessed: 12.11.2023).
4. Elzain H., Wu Y. Software Defined Wireless Mesh Network Flat Distribution Control Plane. Future Internet. 2019. Vol. 11 (166). P. 1–17.
5. FAO: E-agriculture. The possibilities of Internet of Things (IoT) for Agriculture. URL: fao.org/e-agriculture/news/possibilities-internet-things-iot-agriculture (accessed: 15.11.2023).
6. Boobalan J., Jacintha V., Nagarajan J., Thangayogesh K., Tamilarasu S. An OT Based Agriculture Monitoring System. In: 2018 Int. Conf. ICCSP. Chennai, India. 2018. P. 0594–0598.
7. Sita Kumari K., Abdul Haleem S.L., Shivaprakash G., Saravanan M., Arunsundar B., Thandava Krishna Sai Pandraju. Agriculture monitoring system based on internet of things by deep learning feature fusion with classification. Computers and Electrical Engineering. 2022. Vol. 102. P. 1–14.
8. Saleheen M.M.U., Islam M.S., Fahad R., Belal M.J.B., Khan R. IoT-Based Smart Agriculture Monitoring System. In: 2022 IEEE Int. Conf. IICAIET. Kota Kinabalu, Malaysia. 2022. P. 1–6.
9. IEEE: Standard for Information Technology. URL: standards.ieee.org/ieee/802.11s/4243/ (accessed: 16.11.2023).
10. CSA: ZigBee. URL: csa-iot.org/all-solutions/zigbee/ (accessed: 04.12.2023).
11. Laktionov I., Diachenko G., Rutkowska D., Kisiel-Dorohinicki M. An Explainable AI Approach to Agrotechnical Monitoring and Crop Diseases Prediction in Dnipro Region of Ukraine. JAISCR. 2023. Vol. 13 (4). P. 247–272.

12. Laktionov I., Diachenko G., Koval V., Yevstratiev M. Computer-Oriented Model for Network Aggregation of Measurement Data in IoT Monitoring of Soil and Climatic Parameters of Agricultural Crop Production Enterprises. *Baltic J. Modern Computing*. 2023. Vol. 11 (3). P. 500–522.

#### REFERENCES:

1. Karunathilake E.M.B.M., Le A.T., Heo S., Chung Y.S., Mansoor S. (2023). The Path to Smart Farming: Innovations and Opportunities in Precision Agriculture. *Agriculture*. Vol. 13 (8). P. 1–26.
2. Farooq M.S., Riaz S., Abid A., Umer T., Zikria Y.B. (2020). Role of IoT Technology in Agriculture: A Systematic Literature Review. *Electronics*. Vol. 9 (2). P. 1–41.
3. National Economic Strategy 2030. Retrieved from [nes2030.org.ua/](https://nes2030.org.ua/) (accessed: 12.11.2023).
4. Elzain H., Wu Y. (2019). Software Defined Wireless Mesh Network Flat Distribution Control Plane. *Future Internet*. Vol. 11 (166). P. 1–17.
5. FAO: E-agriculture. The possibilities of Internet of Things (IoT) for Agriculture. Retrieved from [fao.org/e-agriculture/news/possibilities-internet-things-iot-agriculture](https://fao.org/e-agriculture/news/possibilities-internet-things-iot-agriculture) (accessed: 15.11.2023).
6. Boobalan J., Jacintha V., Nagarajan J., Thangayogesh K., Tamilarasu S. (2018). An OT Based Agriculture Monitoring System. In: 2018 Int. Conf. ICCSP. Chennai, India. P. 0594–0598.
7. Sita Kumari K., Abdul Haleem S.L., Shivaprakash G., Saravanan M., Arunsundar B., Thandava Krishna Sai Pandraju. (2022). Agriculture monitoring system based on internet of things by deep learning feature fusion with classification. *Computers and Electrical Engineering*. Vol. 102. P. 1–14.
8. Saleheen M.M.U., Islam M.S., Fahad R., Belal M.J.B., Khan R. (2022). IoT-Based Smart Agriculture Monitoring System. In: 2022 IEEE Int. Conf. IICAIET. Kota Kinabalu, Malaysia. P. 1–6.
9. IEEE: Standard for Information Technology. Retrieved from [standards.ieee.org/ieee/802.11s/4243/](https://standards.ieee.org/ieee/802.11s/4243/) (accessed: 16.11.2023).
10. CSA: ZigBee. Retrieved from [csa-iot.org/all-solutions/zigbee/](https://csa-iot.org/all-solutions/zigbee/) (accessed: 04.12.2023).
11. Laktionov I., Diachenko G., Rutkowska D., Kisiel-Dorohinicki M. (2023). An Explainable AI Approach to Agrotechnical Monitoring and Crop Diseases Prediction in Dnipro Region of Ukraine. *JAISCR*. Vol. 13 (4). P. 247–272.
12. Laktionov I., Diachenko G., Koval V., Yevstratiev M. (2023). Computer-Oriented Model for Network Aggregation of Measurement Data in IoT Monitoring of Soil and Climatic Parameters of Agricultural Crop Production Enterprises. *Baltic J. Modern Computing*. Vol. 11 (3). P. 500–522.

**UDC 614.2+574/578+004.38**

**DOI <https://doi.org/10.32782/IT/2023-4-5>**

**Oleksandr LYTVYNOV**

*Candidate of Technical Sciences, Associate Professor, the Faculty of Physics, Electronics and Computer Systems, Oles Honchar Dnipro National University, 72, Haharina Ave, Dnipro, Ukraine, 49000, lytvynov@ffeks.dnu.edu.ua*

**ORCID: 0000-0001-7660-1353**

**Dmytro HRUZIN**

*Master, Postgraduate Student, the Faculty of Physics, Electronics and Computer Systems, Oles Honchar Dnipro National University, 72, Haharina Ave, Dnipro, Ukraine, 49000, hruzin\_dl@ffeks.dnu.edu.ua*

**ORCID: 0009-0004-8534-2559**

**To cite this article:** Lytvynov, O., Hruzin, D. (2023). Metody optymizatsii zavantazhennia ta onovlennia vebstorinok za dopomohoiu khmarnykh tekhnolohii [Methods for optimizing the loading and updating of web pages using cloud technologies]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 40–50, doi: <https://doi.org/10.32782/IT/2023-4-5>

## **METHODS FOR OPTIMIZING THE LOADING AND UPDATING OF WEB PAGES USING CLOUD TECHNOLOGIES**

*An essential component of modern systems built on cloud technologies, which determines the performance and efficiency of system operation, is caching methods and technologies. For systems that provide static information to end-users, technologies such as SSG (Static Site Generation) and SSR (Server-side Rendering) are applied in building the client side. SSG is typically used for pages with a very low frequency of content changes, as modifying the content on one page requires rebuilding the entire site. To cache SSR application pages, the use of a cache proxy server is recommended. An example of implementing such an approach is Next.js Serverless. However, because the Next.js Serverless technology employs a weak caching model, delivering up-to-date data to end users often takes a significant amount of time, which can negatively impact user interaction.*

*This work is dedicated to addressing this problem. The paper proposes a flexible and efficient solution for generating and caching static pages using a cache proxy server and invalidation for data updates by the cache proxy server upon each change in the system's state. The integration of this solution into complex information systems is explored, and a performance comparison and evaluation of data delivery to end-users are conducted.*

*The results of the conducted experiment demonstrate that the proposed approach solves the identified problem without compromising the performance compared to the Next.js Serverless approach.*

**Key words:** Cloud technologies, Server Side Rendering, Static Web Pages, Edge caching, Cache Proxy Server, Domain Driven Design.

**Олександр ЛИТВИНОВ**

*кандидат технічних наук, доцент, факультет фізики, електроніки та комп'ютерних систем, Дніпровський національний університет імені Олеся Гончара, просп. Гагаріна, 72, м. Дніпро, Україна, 49000, lytvynov@ffeks.dnu.edu.ua*

**ORCID: 0000-0001-7660-1353**

**Дмитро ГРУЗІН**

*магістр, аспірант, факультет фізики, електроніки та комп'ютерних систем, Дніпровський національний університет імені Олеся Гончара, просп. Гагаріна, 72, м. Дніпро, Україна, 49000, hruzin\_dl@ffeks.dnu.edu.ua*

**ORCID: 0009-0004-8534-2559**

**Бібліографічний опис статті:** Литвинов, О., Грузін, Д. (2023). Методи оптимізації завантаження та оновлення вебсторінок за допомогою хмарних технологій. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 40–50, doi: <https://doi.org/10.32782/IT/2023-4-5>



## МЕТОДИ ОПТИМІЗАЦІЇ ЗАВАНТАЖЕННЯ ТА ОНОВЛЕННЯ ВЕБСТОРИНОК ЗА ДОПОМОГОЮ ХМАРНИХ ТЕХНОЛОГІЙ

Важливою складовою сучасних систем, побудованих на базі хмарних технологій, від якої залежить продуктивність та ефективність функціонування системи, є методи та технології кешування. Для ряду систем, які надають статичну інформацію кінцевому користувачеві, для побудови клієнтської частини застосовуються такі технології, як SSG (Static Site Generation), SSR (Server-side Rendering). SSG, як правило, застосовується для сторінок, частота зміни вмісту яких дуже низька, оскільки для зміни вмісту на одній сторінці потрібне перезбирання всього сайту. Для кешування сторінок SSR-додатків пропонується використовувати кеш-проксі-сервер. Прикладом реалізації такого підходу може служити Next.js Serverless. Проте через те, що технологія Next.js Serverless використовує слабку модель кешування, доставка актуальних даних кінцевим користувачам часто займає значну кількість часу, що може негативно вплинути на взаємодію з користувачем.

Вирішенню цієї проблеми присвячена дана робота. У роботі пропонується гнучке та ефективне рішення для генерації та кешування статичних сторінок з використанням кеш-проксі-сервера та інвалідації для актуалізації даних кеш-проксі-сервера при кожній зміні стану системи. Досліджено інтеграцію цього рішення в складні інформаційні системи, проведено порівняння продуктивності та оцінку доставки даних кінцевому користувачеві.

Результати проведеного експерименту показують, що запропонований підхід вирішує поставлену проблему, не уступаючи в продуктивності підходу Next.js Serverless.

**Ключові слова:** хмарні технології, рендеринг на боці сервера, статичні вебсторінки, Едж Кешування, Кеш Проксі-сервер, Доменно-орієнтований Дизайн.

The trend of migrating businesses to the cloud emerged long ago and has been resilient for quite some time. Correspondingly, cloud technologies continue to evolve. Currently, individuals are not just allowed to rent space for their servers but can gradually purchase it, dynamically scaling computational power both vertically and horizontally as needed.

The primary advantage of **cloud technologies** (Thakur, 2022) lies in their flexibility. Users no longer need to worry about physical hardware, its maintenance, and updates. They can focus on their business, leaving all technical aspects in the hands of the cloud provider. This also reduces operational costs and enhances efficiency, allowing companies to concentrate on the development and implementation of new products and services.

With each passing year, cloud technologies become increasingly accessible. The cost of services offered makes them viable even for novice startups. Moreover, major conglomerates such as Amazon (2), Google (3), and Microsoft (4) provide various discounts and free services up to specific credit amounts to attract potential clients. In addition, user interfaces are evolving, enabling non-technical users to leverage cloud services. While this, on one hand, reduces the skill requirements for DevOps personnel, on the other hand, it necessitates their familiarity with the specialized knowledge of using a specific console. Given the multitude of options available, training and even user certification programs are conducted to guide users in effectively utilizing cloud systems.

Talking about cloud services and web performance optimization, special attention is given to

**data caching** (Chockler, 2011: pp. 1-11)(Choi, 2020: pp. 98-110)(Berger, 2014: pp. 2-23). Cloud platforms provide a range of caching mechanisms, including in-memory caches, distributed caches, and cache on the side of content delivery networks (CDNs). In-memory caches like Redis (8) and Memcached (Fukuda, 2014) prove particularly effective in accelerating data retrieval by storing frequently accessed data in RAM, enabling ultra-low-latency access. Distributed caching solutions, such as Amazon ElastiCache (10), extend this capability across multiple nodes, further enhancing the speed and reliability of data access. CDNs, on the other hand, facilitate the efficient distribution of content and data to multiple geographic locations, reducing latency for users and researchers worldwide. These caching techniques are invaluable in scenarios where real-time access to data and services is critical.

Caching in cloud services not only enhances the speed and efficiency of scientific computations but also contributes to cost savings. By reducing the amount of data transfer and computational overhead, researchers can optimize resource utilization, ultimately leading to lower cloud service bills. Moreover, caching enables the reuse of previously computed results, minimizing redundant calculations and improving the sustainability of research projects.

Another technology that significantly increases cloud performance is **Edge caching** (Wu, 2021). The critical component of content delivery and data distribution networks is designed to enhance the speed and efficiency of content delivery to end-users. Unlike traditional caching, which is typ-

ically centralized on a single server or data center, edge caching decentralizes the caching process, bringing content closer to the end-users by placing cached data on edge servers distributed across various geographic locations. This approach significantly reduces the latency and congestion associated with serving content from a centralized location, leading to faster and more reliable access.

The best performance of edge caching is achieved when a website is developed using static web page technology. Pre-rendered pages are cached at the edge, ensuring rapid and efficient functionality. The concept of **pre-generating static web content** (Large, 2022)(Yang, 2022), often referred to as "**static web pages**", predates the modern static site generators (Jiang, 2010: pp: 588-591) that we use today. It's challenging to attribute this concept to a single individual, as it evolved gradually as the web itself developed. Static web pages have been used since the early days of the World Wide Web.

While the concept of static web pages was established early in the history of the web, the specific tools and technologies that we now associate with modern static site generators were developed later. Tools like Jekyll (15), Eleventy (16), Gatsby (17), and others were created to automate and simplify the process of generating static websites. The idea is to generate all possible web pages in advance and cache them. This provides significant performance improvement for read requests. However, this approach provides a low level of flexibility. The main con of the solution is when content does change, the site must be rebuilt to have these changes reflected. In practice, some hybrid approaches work. For example, as described in (Vepsäläinen, 2022), static and dynamic parts of the page are kept separately. The JSON for site definition is leveraged on the client side for editing, bridging the continuum's ends.

Another approach is the **Single Page Application (SPA)** (Fink, 2014), a web application that loads a single HTML page and dynamically updates its content as users interact, eliminating the need for a full page reload. This development paradigm has gained popularity due to its enhanced user experience, faster navigation, and reduced server load. However, a notable drawback of this approach is that the page renders after the initial load, introducing a certain delay before displaying data to the user. Additionally, this can impact page indexing by search engine robots, as dynamically generated content is ignored by some bots, negatively affecting Search Engine Optimization (Gudivada, 2015: pp. 67-76).

Another option for building web clients for complex information systems is **Server Side Rendering (SSR)** (Sun, 2019: pp. 191-217). When a user requests a page from a server, the server determines which page to render and processes the request, including any data fetching or computations needed to render the page. This may involve querying a database, making API requests, or other server-side logic. The server uses a template engine (e.g., Handlebars (22), Pug (23), etc.) to render the HTML template with the data retrieved in the previous step. The resulting HTML is a fully formed web page, often including the content, layout, and initial data. The server sends the complete HTML page, along with any associated assets like CSS and JavaScript files, to the client's browser. Once the initial HTML is loaded in the browser, any client-side JavaScript can take over and enhance the page's interactivity. The process repeats when a user navigates to another page within the application. A new request is sent to the server, which generates the HTML for the new page, and the client-side JavaScript updates the page content.

In practice, a common approach involves employing both methods for different parts of the system. Static pages, which do not require frequent updates, are generated using a generator and cached, while web pages in need of frequent updates are based on SSR or SPA approaches. Although these solutions work sufficiently, concerns regarding performance and content updates persist. Additionally, it is often a challenge for developers to find the right balance, determining which pages to generate and which to render in real time.

Methods exist to optimize page load time when using the SSR approach, such as employing a multi-level cache (Vilas, 2006: pp. 713-720) and a cache proxy server (Wang, 1999). The system's operation using these methods is illustrated in Fig. 1. When a client sends a request for a page, the request is received by the CDN service, and if an up-to-date version of the page is present, it is delivered to the client. If the page is not present or its time-to-live has expired, the request is redirected to the cache proxy server. This cache has a larger size, and the time-to-live for pages can be longer. If the cache proxy server also lacks an up-to-date version of the page, the page is requested from the server, which renders it and sends it to the client, also saving the updated version on the cache proxy server.

An example of implementing such an approach can be found in the Next.js Serverless solution (26) (27)(28). This framework is designed to create a

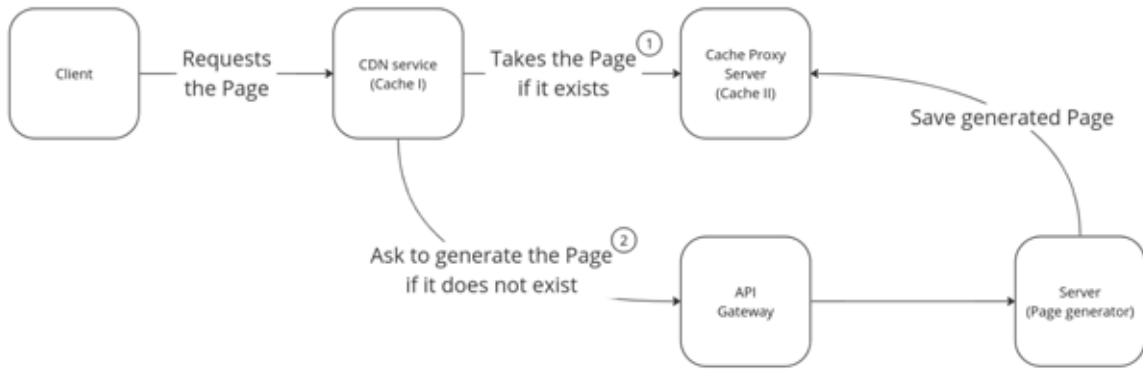


Fig. 1. Cache Proxy server approach

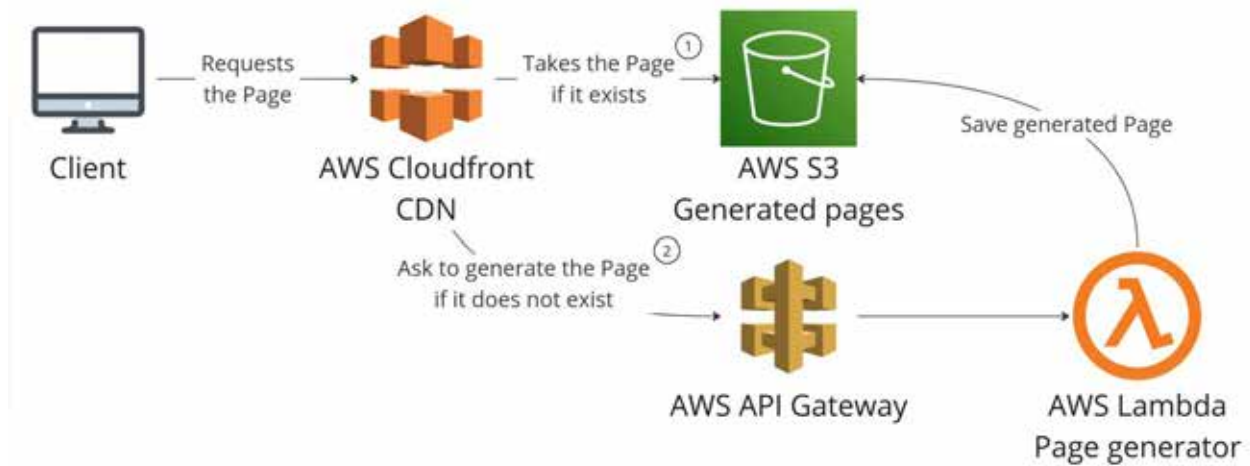


Fig. 2. The diagram of components for the Next.js Serverless approach

serverless architecture (Rajan, 2018), which offers certain advantages over the traditional server architecture for certain systems. The concept involves a resource distribution approach. In the case of the Next.js (30) SSR framework, the conventional server, which requires scaling and continuous operation with associated costs, is replaced with a storage solution for static files (AWS S3) and a stateless worker that can be invoked on-demand (AWS Lambda). The S3 server in this system essentially serves as a cache proxy server, with AWS CloudFront as the CDN service. Implementation of a system based on the Next.js Serverless framework is depicted in Fig. 2.

This approach ensures a high level of scalability, enabling the parallel rendering of a vast number of pages. However, the question of what cache model, weak or strong, to use remains. Cache invalidation operations on CDNs provided by cloud platforms can be relatively costly, especially with a large number of calls. Therefore, it is advisable to avoid cache invalidation unless necessary. Another consideration is data freshness in case a weak cache model is used.

Next.js Serverless approach utilizes a weak caching model, delivering up-to-date data to end-users often takes a considerable amount of time, which can negatively impact user experience.

**Task definition.** The solutions mentioned above are aimed at systems whose data state changes not too frequently but is not entirely static either.

The example can be illustrated by the health care system developed by a team from DBB Software company (31). The system aims to provide users with public profiles of doctors and their practices, along with reviews about these doctors. The system undergoes changes when content managers add or modify doctor profiles or their practices, and when patients add reviews. These are not real-time operations, and the update frequency of the page does not exceed tens of changes per hour.

One of the possible architectural solutions considered for building such a system was the generation of static pages. However, despite the high performance of this approach, especially when using edge caching, such a solution was rejected

at the planning stage. The challenge with the page generation approach lies in the need for rebuilding and redeploying all pages to make any changes. Using this approach in the current scenario would be inconvenient, as the rebuilding process takes time, and rebuilding the entire site multiple times a day would be resource-intensive.

The first solution implemented was the SSR Next.js server. SSR renders pages in real-time, and all system changes are reflected on the client almost immediately after the system data is updated. However, this approach comes at the cost of performance, as each page request involves one or more API requests, leading to increased page load time. Even with the use of CDN caching, the average page load speed for some pages reached up to 1700 ms.

Using the Cache Proxy Server with a weak caching model partially addresses this issue by introducing a second level of managed cache. However, on some pages, data may remain unchanged for months, while on others, it is constantly added.

For example, there is a doctor who obtained the last license several years ago and only treats his regular patients. Such a doctor's page will not be updated for months or even years. On the other hand, some doctors ask their patients to leave reviews and constantly receive new certificates. The pages of these doctors are updated multiple times per day. By default, the cache TTL is set the same for all pages. To ensure users receive up-to-date data, the TTL value should be set close to the update frequency of the most frequently updated pages.

One solution to this problem could be to implement logic on the cache proxy server that

is responsible for grouping pages based on their update frequency. For each group, set a TTL that is most suitable for that specific group. However, with this solution, the dynamics of data changes need to be taken into account. For example, a doctor may start asking patients to leave reviews, and a page that used to change once a year will suddenly start changing several times a day. Considering the high complexity and low level of reliability, this solution is not considered effective.

This article aims to find an effective solution for the flexible response of the page generation module to changing data. It also provides a performance comparison and evaluates the delivery of updated data to the end user.

**Main part.** Similar to the Cache Proxy Server approach described above, we also propose rendering and storing pages in a static repository but using a strong caching model and storing not just already requested pages, but all possible pages in advance. The freshness of data in the storage is ensured by regenerating pages with each change (or some amount of changes) in the system that concerns these pages. One or more handlers subscribed to events about changes in the system's data, determine which pages are affected by these changes, and invoke the page renderer service to update these pages in the cache proxy server. After the pages are successfully rendered and saved, a request for invalidating the CDN cache is sent (Fig. 3).

For example, in Domain-Driven Design (DDD) systems (Evans, 2004) (Fig. 4), one of the handlers subscribed to changes in the aggregate's state can perform this function.

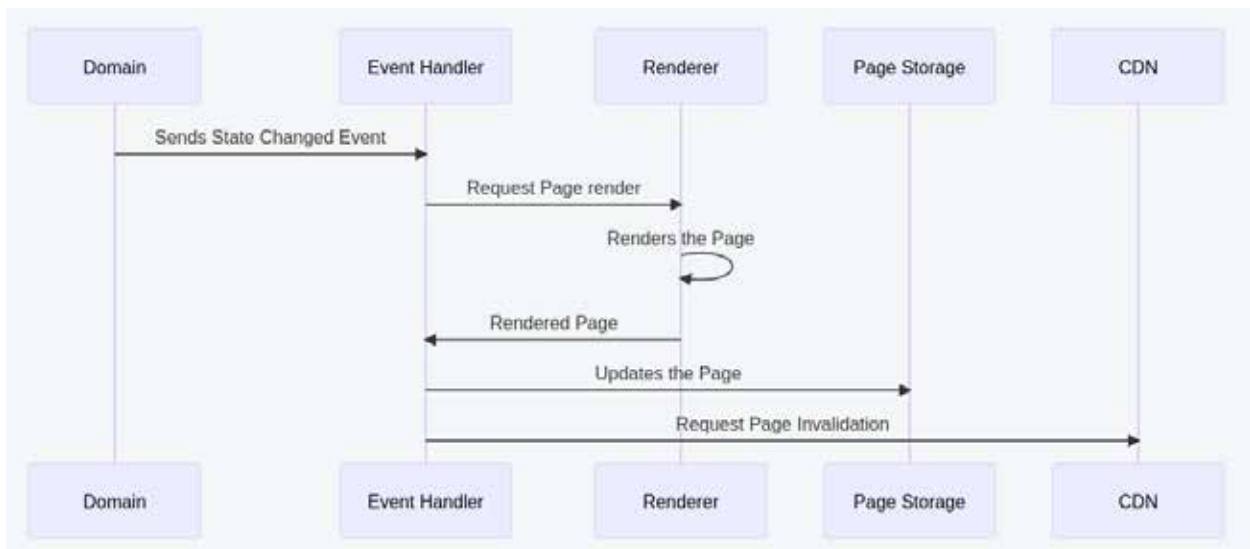


Fig. 3. Update the Page flow

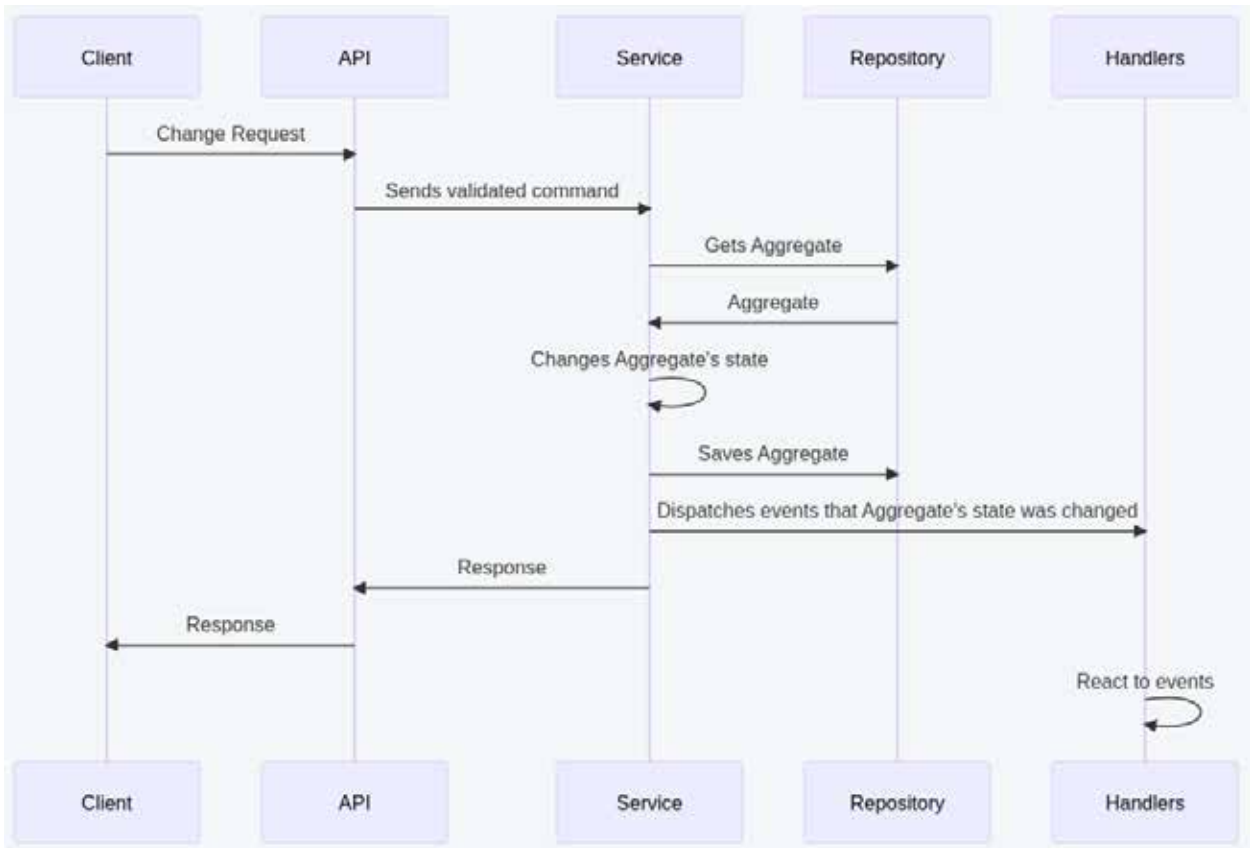


Fig. 4. Sequence diagram of successful DDD change request flow

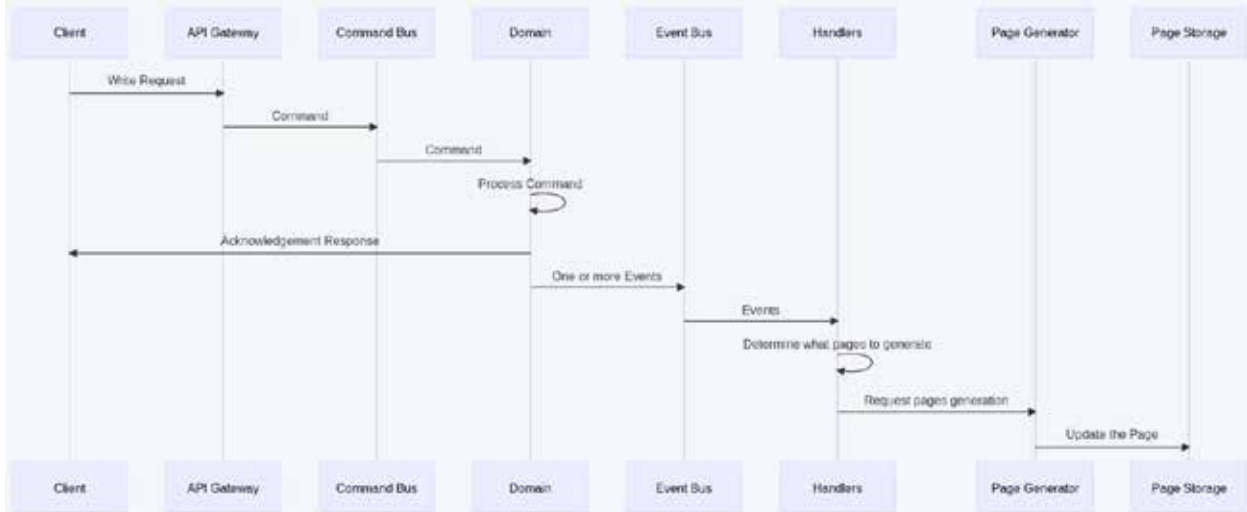
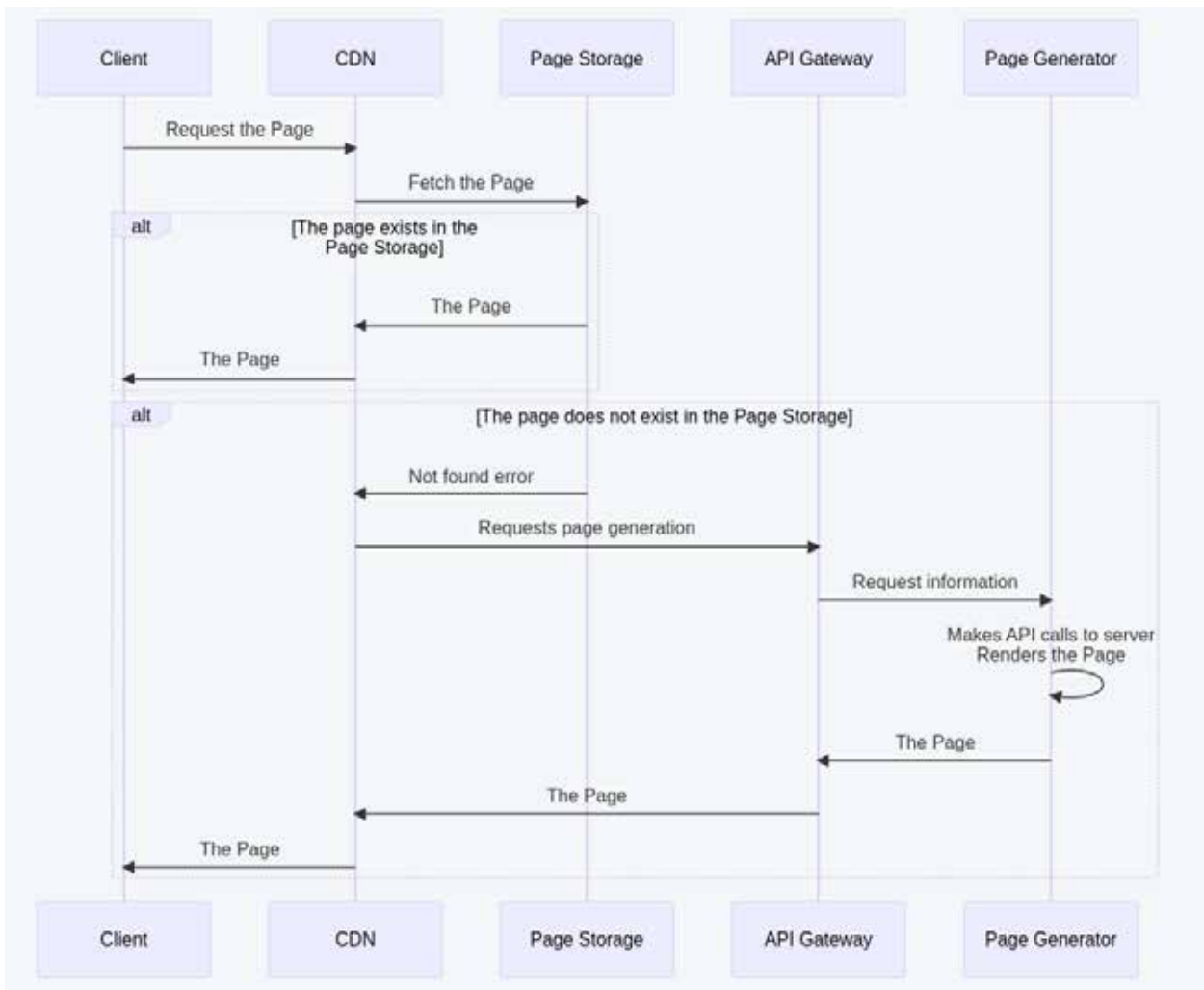


Fig. 5. Event source approach with Page Generator

This approach integrates particularly well with an Event Sourcing system (Chamberlain, 2017), as the implementation of sending events upon changing the state of an aggregate forms the foundation of such systems. The pre-rendered page essentially serves as a projection in this context.

In Fig. 5, the flow of state change in the event-sourced system is depicted. A change request transforms into an understandable message

for Domain (command), which is validated and enters the domain through the command bus. The domain comprises command handlers, a repository, and an aggregate description. The command handler requests the repository for the aggregate based on the ID specified in the command. The repository, relying on the aggregate description, creates an empty object of the required aggregate type, queries the event store for all events



**Fig. 6. Sequence diagram Request prerendered Page**

related to the aggregate, replays them on the new object, and passes the resulting aggregate to the command handler. The command handler, using the aggregate's methods, modifies its state. Upon the state change of the aggregate, multiple events are generated. These events are sent to the repository, saved in the event store, and an acknowledgment response is sent to the client, indicating that the source of truth (event store) of the system has been successfully changed, and all other parts of the system will be eventually updated. Subsequently, the events are sent to the event bus, notifying numerous handlers about the system change. One such handler determines which pages need to be updated based on the dispatched events, triggers the page generator, and invalidates the cache when pages are updated in the Cache Proxy Server.

When a user's browser sends a request to retrieve a page, this request goes to the CDN, which in turn fetches data from the repository and serves the pre-rendered page to the client.

In case the requested page does not exist in the repository, the request is redirected to the generator, which renders the missing page, serves it to the client, and subsequently stores it in the storage (Fig. 6). Since the proposed architecture uses a strong caching model, such cases may occur only under normal system operation when requesting a new page that has not been rendered by the system since the last update.

The described system above has been implemented and deployed in a live project. Fig. 7 shows the component diagram of the implementation of this system based on AWS services and CQRS architecture.

This approach allows generating the necessary pages and invalidating the cache only in situations where it is justified by changes in the system. In scenarios where the system undergoes frequent changes, this approach can be resource-intensive. To enhance scalability, it is advisable to utilize an AWS Lambda function as a hosting environment for the renderer. This solution enables the paral-





Fig. 7. Component diagram

l invocation of multiple renders when necessary, without incurring server maintenance costs during periods of frequent system changes.

The main concern with this solution could be the quantity and frequency of cache invalidation requests, as a large number of requests can become financially burdensome. This issue is addressed by an additional service that is invoked instead of direct invalidation. This service accumulates a predefined number of invalidation requests, aggregates them, removes duplicates, and invalidates all necessary pages with a single request. Such invalidation can be triggered either upon reaching a certain number of invalidation requests or when a specified timeout elapses if the request pool is not empty. This solution has also been implemented. A timeout of 1 hour and a pool size of 100 pages have been set. For the developed system, this results in approximately 200,000 CDN cache invalidations per month for a thousand pages under standard system load. Given the prices for CloudFront invalidation (34), this costs \$1000 per month.

**Methods.** To assess the performance of the solutions in this study, the average read request time for pages with different amounts of content is measured. The measurements were conducted on a real system using an analytical algorithm that, upon each page load not from the browser cache, sent load duration statistics to the statistics server.

**Experiment.** The experiment was conducted based on the DBB Software company's (31) proprietary platform, which provided the necessary infrastructure and tools for data collection and analysis. This platform offered essential capabilities for our research, ensuring the accuracy and reliability of our experimental results. For a typical test system, as described above, the problem was solved using three different methods by two middle-level developers.

This approach significantly accelerates the delivery of up-to-date data to end users by invalidating the cache immediately after a system change rather than waiting for its expiration. In other words, pages of doctors who regularly receive new certificates or have new reviews from their patients will be updated regularly, while pages, where new data does not appear for months, will not undergo a re-render.

Another question is how the proposed modification affects performance compared to other approaches. The average page load time for a real website with varying amounts of content was measured using three considered approaches:

- SSR application using Next JS React framework, hosted on AWS Elastic Beanstalk (35) M2 instance.
- Cache Proxy Server with a weak caching model approach as Next JS Serverless realization, hosted on AWS Lambda function.

Table 1

**Different update frequency issue solving**

	SSR	Cache Proxy Server with a weak caching model	Suggested approach
<b>Issue solved</b>	Yes	No	Yes

Table 2

**Average page load time**

Page size	time, ms		
	SSR	Cache Proxy Server with a weak caching model	Suggested approach
<b>9 Mb</b>	1782	827	799
<b>5 Mb</b>	496	93	89
<b>2 Mb</b>	301	61	55

– Custom implementation of suggested approach on AWS Lambda function with a subscription to system events.

From Table 2, it can be seen that the proposed approach, and the Cache Proxy Server with a weak caching model, significantly reduce page load time compared to the pure SSR approach. This is logically explained by the fact that an SSR application renders the page every time if it is not in the CDN cache, and the CDN cache cannot store pages that quickly become outdated. The proposed approach is not inferior in speed to Cache Proxy Server with a weak caching model, even showing a slight performance improvement. This is because, using the weak cache model, the system requests page rendering upon user demand when the page's time-to-live is considered outdated. The proposed approach updates the data in the cache when the system state changes and assumes the presence of the page in the page storage. Thus, rendering upon user request occurs only when the data is not in the cache, which, under normal system operation, should not happen.

A drawback of the proposed approach is its considerable implementation complexity. The Cache Proxy Server with a weak caching model

has a similarly high level of complexity, but Next.js Serverless provides a ready-made implementation that can be easily used for one's system. Similarly, the proposed approach offers a high degree of reusability. Once implemented for one system, it can be encapsulated into a separate module and reused for other systems with relatively small effort.

**Summary.** The page load time and timely delivery of up-to-date data significantly impact user experience and are critical indicators in assessing system performance from the end user's perspective. This article explored approaches to organizing the frontend part of the system and optimizing page load time for systems whose data state changes not too frequently but is not entirely static. Additionally, a modification to one of these methods was proposed, which does not lag in page load speed compared to existing approaches and speeds up the delivery of up-to-date data to end users. Examples of implementing the proposed approach for an event source and a standard DDD system were described. An experiment was also conducted, demonstrating that the proposed approach, for a specific type of system, performed no worse than existing ones.

**BIBLIOGRAPHY:**

1. Thakur N. Singh A. Sangal A.L. Cloud services selection: A systematic review and future research directions. *Computer Science Review*. Volume 46, 2022. Doi: 10.1016/j.cosrev.2022.100514.
2. Cloud computing services – Amazon Web Services. URL: <https://aws.amazon.com>.
3. Google Cloud console. URL: <https://console.cloud.google.com>.
4. Azure Cloud Services. URL: <https://azure.microsoft.com/en-us/products/cloud-services>.
5. Chockler G. Laden G. Vigfusson Y. Design and implementation of caching services in the cloud, 2011. Pages: 1 – 11. DOI:10.1147/JRD.2011.2171649.
6. Choi J. Gu Y. Kim J. Learning-based dynamic cache management in a cloud. *Journal of Parallel and Distributed Computing*. Volume 145, 2020. Pages: 98-110. DOI: 10.1016/j.jpdc.2020.06.013.
7. Berger D.S. Gland P. Singla S. Ciucu F. Exact analysis of TTL cache networks. *Performance Evaluation*. Volume 79, 2014. Pages 2-23. DOI: 10.1016/j.peva.2014.07.001.
8. Redis. URL: <https://redis.io>.



9. Fukuda E.S. Caching memcached at reconfigurable network interface. Conference: IEEE International Conference on Field Programmable Logic and Applications (FPL), 2014. DOI: 10.1109/FPL.2014.6927487.
10. Elasticache – AWS. URL: <https://aws.amazon.com/elasticache>.
11. Wu H. Fan Y. Wang Y. Ma H. Xing L. A Comprehensive Review on Edge Caching from the Perspective of Total Process: Placement, Policy and Delivery, 2021. Doi: 10.3390/s21155033.
12. Large D. What is a Static Site Generator? Cloud cannon, 2022. URL: <https://cloudcannon.com/blog/what-is-a-static-site-generator/>.
13. Yang K. An Introduction to Static Site Generators. Digital Ocean, 2022. URL: <https://www.digitalocean.com/community/conceptual-articles/introduction-to-static-site-generators>.
14. Jiang W.R. Yan J.H. Implementation of Static Web-Pages Generator Using JavaScript. Applied Mechanics and Materials, 2010. Pages. 588-591. DOI:10.4028/www.scientific.net/AMM.39.588.
15. Jekyll. Simple, blog-aware, static sites. URL: <https://jekyllrb.com>.
16. Eleventy. A simpler static site generator. URL: <https://www.11ty.dev>.
17. The Best React-Based Framework. Gatsby. URL: <https://www.gatsbyjs.com>.
18. Vepsäläinen J. Vuorimaa P. Bridging Static Site Generation with the Dynamic Web. In: Di Noia, T., Ko, IY., Schedl, M., Ardito, C. (eds) Web Engineering. ICWE 2022. Lecture Notes in Computer Science, vol 13362, 2022. DOI: 10.1007/978-3-031-09917-5\_32. ISBN: 978-3-031-09916-8.
19. Fink G. Flatow I. Introducing Single Page Applications, 2014. DOI: 10.1007/978-1-4302-6674-7\_1.
20. Gudivada V.N. Rao D. Paris J. Understanding Search Engine Optimization, 2015. Pages: 67-76. DOI: 10.1109/MC.2015.297.
21. Sun Y. Server-Side Rendering: Building Reliable, High-Performance Web Apps Using Elm-Inspired Architecture, Event Pub-Sub, and Components. Practical Application Development with AppRun, 2019. Pages: 191-217. DOI:10.1007/978-1-4842-4069-4\_9.
22. Handlebars. URL: <https://handlebarsjs.com>.
23. Pug. URL: <https://pugjs.org/api/getting-started.html>.
24. Vilas J.F. Pazos-Arias J.J. Vilas A.F. Optimizing Web Services Performance Using Cache. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2006. Pages: 713-720. DOI: 10.20965/jaciii.2006.p0713.
25. Wang J. A Scalable Efficient Robust Adaptive (SERA) Architecture for the Next Generation of Web Service, 1999.
26. Running Next.js applications with Serverless services on AWS. Serverlessland. URL: <https://serverlessland.com/repos/nextjs-serverless-architecture>.
27. Serverless Nextjs Plugin. URL: <https://www.serverless.com/plugins/serverless-nextjs-plugin>.
28. Pull-request with a description of how Next.js Serverless works inside. URL: <https://github.com/serverless-nextjs/serverless-next.js/pull/1028>.
29. Rajan A.P. Serverless Architecture – A Revolution in Cloud Computing, 2018. DOI: 10.1109/ICoAC44903.2018.8939081.
30. The React Framework for the Web. URL: <https://nextjs.org>.
31. DBB Software compare. Website URL: <https://dbbsoftware.com/en>.
32. Evans E. Domain-Driven Design: Tackling Complexity in the Heart of Software, 2004. ISBN: 978-0321125217.
33. Chamberlain N. Applying Domain Driven Design with CQRS and Event Sourcing, 2017.
34. CloudFront – How AWS Pricing Works. URL: <https://docs.aws.amazon.com/whitepapers/latest/how-aws-pricing-works/cloudfront.html>.
35. Elasticbeanstalk – AWS. URL: <https://aws.amazon.com/elasticbeanstalk>.

#### REFERENCES:

1. Thakur, N., Singh, A., Sangal, A.L. (2022). Cloud services selection: A systematic review and future research directions. Computer Science Review. Volume 46. Doi: 10.1016/j.cosrev.2022.100514.
2. Cloud computing services – Amazon Web Services. Retrieved from <https://aws.amazon.com>.
3. Google Cloud console. Retrieved from <https://console.cloud.google.com>.
4. Azure Cloud Services. Retrieved from <https://azure.microsoft.com/en-us/products/cloud-services>.
5. Chockler, G., Laden, G., Vigfusson, Y. (2011). Design and implementation of caching services in the cloud. DOI:10.1147/JRD.2011.2171649.
6. Choi, J., Gu, Y., Kim, J. (2020). Learning-based dynamic cache management in a cloud. Journal of Parallel and Distributed Computing. Volume 145. DOI: 10.1016/j.jpdc.2020.06.013.

7. Berger, D.S., Gland, P., Singla, S., Ciucu, F. (2014). Exact analysis of TTL cache networks. *Performance Evaluation*. Volume 79. DOI: 10.1016/j.peva.2014.07.001.
8. Redis. Retrieved from <https://redis.io>.
9. Fukuda, E.S. (2014). Caching memcached at reconfigurable network interface. Conference: IEEE International Conference on Field Programmable Logic and Applications (FPL). DOI: 10.1109/FPL.2014.6927487.
10. Elasticache – AWS. Retrieved from <https://aws.amazon.com/elasticache>.
11. Wu, H., Fan, Y., Wang, Y., Ma, H., Xing, L. (2021). A Comprehensive Review on Edge Caching from the Perspective of Total Process: Placement, Policy and Delivery. Doi: 10.3390/s21155033.
12. Large, D. (2022). What is a Static Site Generator? Cloud cannon. Retrieved from <https://cloudcannon.com/blog/what-is-a-static-site-generator/>.
13. Yang, K. (2022). An Introduction to Static Site Generators. Digital Ocean. Retrieved from <https://www.digitalocean.com/community/conceptual-articles/introduction-to-static-site-generators>.
14. Jiang, W.R., Yan, J.H. (2010). Implementation of Static Web-Pages Generator Using JavaScript. *Applied Mechanics and Materials*. DOI: 10.4028/www.scientific.net/AMM.39.588.
15. Jekyll. Simple, blog-aware, static sites. Retrieved from <https://jekyllrb.com>.
16. Eleventy. A simpler static site generator. Retrieved from <https://www.11ty.dev>.
17. The Best React-Based Framework. Gatsby. Retrieved from <https://www.gatsbyjs.com>.
18. Vepsäläinen, J., Vuorimaa, P. (2022). Bridging Static Site Generation with the Dynamic Web. In: Di Noia, T., Ko, IY., Schedl, M., Ardito, C. (eds) *Web Engineering*. ICWE 2022. Lecture Notes in Computer Science, vol 13362. DOI: 10.1007/978-3-031-09917-5\_32. ISBN: 978-3-031-09916-8.
19. Fink, G., Flatow, I. (2014). Introducing Single Page Applications. DOI: 10.1007/978-1-4302-6674-7\_1.
20. Gudivada, V.N., Rao, D., Paris, J. (2015). Understanding Search Engine Optimization. DOI: 10.1109/MC.2015.297.
21. Sun, Y. (2019). Server-Side Rendering: Building Reliable, High-Performance Web Apps Using Elm-Inspired Architecture, Event Pub-Sub, and Components. *Practical Application Development with AppRun*. DOI:10.1007/978-1-4842-4069-4\_9.
22. Handlebars. Retrieved from <https://handlebarsjs.com>.
23. Pug. Retrieved from <https://pugjs.org/api/getting-started.html>.
24. Vilas, J.F., Pazos-Arias, J.J., Vilas, A.F. (2006). Optimizing Web Services Performance Using Cache. *Journal of Advanced Computational Intelligence and Intelligent Informatics*. DOI: 10.20965/jaciii.2006.p0713.
25. Wang, J. (1999). A Scalable Efficient Robust Adaptive (SERA) Architecture for the Next Generation of Web Service.
26. Running Next.js applications with Serverless services on AWS. Serverlessland. Retrieved from <https://serverlessland.com/repos/nextjs-serverless-architecture>.
27. Serverless Nextjs Plugin. Retrieved from <https://www.serverless.com/plugins/serverless-nextjs-plugin>.
28. Pull-request with a description of how Next.js Serverless works inside. Retrieved from <https://github.com/serverless-nextjs/serverless-next.js/pull/1028>.
29. Rajan, A.P. (2018). Serverless Architecture – A Revolution in Cloud Computing. DOI: 10.1109/ICoAC44903.2018.8939081.
30. The React Framework for the Web. Retrieved from <https://nextjs.org>.
31. DBB Software compare. Website Retrieved from <https://dbbsoftware.com/en>.
32. Evans, E. (2004). *Domain-Driven Design: Tackling Complexity in the Heart of Software*. ISBN: 978-0321125217.
33. Chamberlain, N. (2017). *Applying Domain Driven Design with CQRS and Event Sourcing*.
34. CloudFront – How AWS Pricing Works. Retrieved from <https://docs.aws.amazon.com/whitepapers/latest/how-aws-pricing-works/cloudfront.html>.
35. Elasticbeanstalk – AWS. Retrieved from <https://aws.amazon.com/elasticbeanstalk>.

УДК 004.4:617

DOI <https://doi.org/10.32782/IT/2023-4-6>

### **Віра ЛЮБЧЕНКО**

доктор технічних наук, професор кафедри інженерії програмного забезпечення, Національний університет «Одеська політехніка», просп. Шевченка, 1, м. Одеса, Україна, 65044, [lvv@op.edu.ua](mailto:lvv@op.edu.ua)

ORCID: 0000-0002-4611-7832

Scopus Author ID: 56667638800

### **Данило ЧУМАЧЕНКО**

аспірант кафедри інженерії програмного забезпечення, Національний університет «Одеська політехніка», просп. Шевченка, 1, м. Одеса, Україна, 65044, [chumachdk@gmail.com](mailto:chumachdk@gmail.com)

ORCID: 0009-0000-1477-534X

**Бібліографічний опис статті:** Любченко, В., Чумаченко, Д. (2023). Модель якості програмного забезпечення для біонічних протезів. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 51–57, doi: <https://doi.org/10.32782/IT/2023-4-6>

## **МОДЕЛЬ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ БІОНІЧНИХ ПРОТЕЗІВ**

Розробка біонічних протезів та програмного забезпечення, що їх супроводжує, за останні роки значно просулася вперед, пропонуючи користувачам інноваційні способи контролю та взаємодії зі своїми протезами. Це дослідження заглиблюється в різні аспекти розробки програмного забезпечення для біонічних протезів, зосереджуючись на отриманні даних, передачі тактильних відчуттів і зворотного зв'язку, математичній обробці даних і заходах забезпечення якості. Основна увага приділяється методам отримання даних від біонічних протезів, які включають міоелектричне пряме управління, розпізнавання міоелектричних патернів і управління за допомогою механоміограми. Кожен метод пов'язаний з унікальними проблемами, такими як зовнішні шумові перешкоди та м'язова втома, які вимагають надійних програмних рішень для збору, інтерпретації та обробки даних. Передача тактильних відчуттів і зворотного зв'язку – ще один важливий аспект, який розглядається в цьому дослідженні, з акцентом на остеоперцепцію, простий тактильний зворотний зв'язок і електричну стимуляцію нервів. Програмне забезпечення відіграє ключову роль у точному відтворенні сигналів управління, щоб забезпечити користувачам реалістичні тактильні відчуття і зворотний зв'язок, покращуючи їх загальний досвід користування протезом і функціональність. Досліджуються математичні моделі та методи обробки даних, включаючи метод Кейна, моделювання аферентної активності, ієрархічну класифікацію та методи статистичного аналізу. Ці математичні інструменти допомагають зрозуміти рухи користувача, коактивацію м'язів та ефективність систем управління протезами. Нарешті, запропоновано комплексну модель якості програмного забезпечення для біонічних протезів, що охоплює вісім ключових характеристик: надійність, безпека, простота використання, швидкість реагування, адаптивність, довговічність, сумісність і конфіденційність. Хоча деякі характеристики відповідають встановленим моделям якості програмного забезпечення, таким як SQuaRE, додаткові атрибути, такі як безпека, адаптивність і довговічність, розроблені спеціально для програмного забезпечення для біонічних протезів.

**Ключові слова:** Біонічні протези, Розробка програмного забезпечення, Управління протезами, Тактильний зворотний зв'язок, Математичне моделювання, Забезпечення якості.

### **Vira LIUBCHENKO**

Doctor of Technical Sciences, Professor at the Software Engineering Department, Odesa Polytechnic National University, 1, Shevchenko Ave, Odesa, Ukraine, 65044, [lvv@op.edu.ua](mailto:lvv@op.edu.ua)

ORCID: 0000-0002-4611-7832

Scopus Author ID: 56667638800

### **Danylo CHUMACHENKO**

Postgraduate Student at the Software Engineering Department, Odesa Polytechnic National University, 1, Shevchenko Ave, Odesa, Ukraine, 65044, [chumachdk@gmail.com](mailto:chumachdk@gmail.com)

ORCID: 0009-0000-1477-534X

**To cite this article:** Liubchenko, V., Chumachenko, D. (2023). Model yakosti prohramnoho zabezpechennia dlia bionichnykh proteziv [Quality model for software for bionic prostheses]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 51–57, doi: <https://doi.org/10.32782/IT/2023-4-6>

## QUALITY MODEL FOR SOFTWARE FOR BIONIC PROSTHESES

*The development of bionic prostheses and their accompanying software has advanced significantly in recent years, offering innovative ways for users to control and interact with their prosthetic limbs. This research delves into various aspects of software development for bionic prostheses, focusing on obtaining data, transmitting tactile sensations and feedback, mathematical processing of data, and quality assurance measures. One primary focus is on the methods of obtaining data from bionic prostheses, which include myoelectric direct control, recognition of myoelectric patterns, and mechanomyogram control. Each method presents unique challenges, such as external noise interference and muscle fatigue, which necessitate robust software solutions for data collection, interpretation, and processing. Transmission of tactile sensations and feedback is another crucial aspect addressed in this research, with emphasis on osteoperception, simple tactile feedback, and electrical nerve stimulation. Software plays a pivotal role in accurately reproducing control signals to provide users with realistic tactile sensations and feedback, enhancing their overall prosthetic limb experience and functionality. Mathematical models and methods for data processing are explored, including the Kane method, afferent activity modeling, hierarchical clustering, and statistical analysis techniques. These mathematical tools aid in understanding user movements, muscle coactivation, and the effectiveness of prosthetic control systems. Finally, a comprehensive quality model for bionic prosthesis software is proposed, encompassing eight key characteristics: reliability, security, ease of use, responsiveness, adaptability, durability, interoperability, and privacy. While some characteristics align with established software quality models like SQuaRE, additional attributes such as safety, adaptability, and durability are tailored specifically for bionic prosthetic software.*

**Key words:** *Bionic prostheses, Software development, Prosthetic control, Tactile feedback, Mathematical modeling, Quality assurance.*

**Вступ.** Біонічні протези – це технологічні пристрої, які створені для відновлення або розширення функціональності втрачених або пошкоджених частин людського тіла. Вони є синтезом біологічних і штучних компонентів, і надають можливість користувачам відновити моторику, тактильні відчуття і навіть частково відновити відчуття дотику.

Історія біоніки розпочала свій розвиток з появою нових матеріалів та технологій, які дозволили створювати пристрої, натхненні біологічними системами. Одним з ключових моментів стало відкриття способів з'єднання скла та металу, що послужило відправною точкою для розробки нейропротезів та інших інноваційних пристроїв. Розвиток склокераміки та припоїв на основі скла дав можливість широкого застосування різноманітних матеріалів в медичних імплантатах, відкриваючи нові горизонти для біоніки та її потенціалу в поліпшенні якості життя людей (Donaldson, 2016).

У США налічується близько 2 мільйонів людей з ампутованими кінцівками, при цьому щорічно проводиться 185 000 ампутацій. Згідно зі статистичними даними Німеччини, Італії та Ірландії, в ЄС налічується близько 3,18 мільйона людей з ампутованими кінцівками (4,66 мільйона в усій Європі), і щорічно проводять близько 295 000 ампутацій (431 000 у всій Європі) (Bumbaširević, 2020).

Після початку війни в Україні проблема ампутованих кінцівок гостро постає перед нашим суспільством, адже величезна кількість людей зазнає травм своїх кінцівок і змушені замінювати їх протезами. Наразі достеменно не

відомо, як багато людей в Україні втратили свої біологічні кінцівки і скільки з них отримали або потребують протезів. Однак війна все ще триває і кількість людей, які потребують протезів, зростатиме.

Існує кілька типів біонічних протезів, що охоплюють різні галузі медичної практики. Приклади включають в себе біонічні протези для кінцівок (рук і ніг), а також протези для органів чуття, таких як очі або слухові апарати. Важливою складовою біонічних протезів є програмне забезпечення, яке керує їхньою функціональністю та взаємодією з людиною (Abrams, 2023). Програмне забезпечення для біонічних протезів відіграє вирішальну роль у забезпеченні синхронізації між штучними компонентами та біологічною системою організму. Воно дає змогу переводити нервові сигнали або інші форми команд від користувача в рухи або інші дії протеза. Ефективне програмне забезпечення забезпечує плавне й точне керування протезом, максимально наближаючи його функціональність до природних рухів.

Проблеми, з якими стикається розробка програмного забезпечення для біонічних протезів, охоплюють не тільки технічні аспекти, як-от сумісність і надійність, а й біологічні чинники (Hong-liu, 2009). Важливо враховувати індивідуальні особливості кожного користувача, оскільки адаптація до їхньої фізіології та психології відіграє вирішальну роль в успішній інтеграції протезів у їхнє повсякденне життя.

**Аналіз останніх досліджень.** Модель якості програмного забезпечення – це концептуальна структура, яка визначає ключові аспекти та

характеристики, за якими можна оцінити якість програмного продукту. Ці моделі надають відомості про те, як вимірювати, аналізувати та забезпечувати високий рівень якості у програмному забезпеченні протягом всього його життєвого циклу.

Зараз до основних характеристик, які враховуються в моделях якості програмного забезпечення, зазвичай включають функціональну придатність, надійність, ефективність, безпеку, спроможність утримуватися, переносимість, сумісність та зручність використання.

Існує кілька моделей якості програмного забезпечення, включаючи стандарти ISO/IEC 9126, ISO/IEC 25010, та інші, які надають керівництво для визначення, вимірювання та покращення якості програмного продукту. На їх основі Міжнародним консорціумом з якості програмного забезпечення (ISO/IEC JTC1/SC7) було розроблено модель SQuaRE (Software Quality Requirements and Evaluation), яка визначає підходи до оцінки та вимог до якості програмного забезпечення.

Проте різноманітність застосувань програмного забезпечення призвела з часом до необхідності врахування додаткових характеристик якості, які притаманні конкретному типу застосувань. Наприклад, для програмних систем з елементами штучного інтелекту додатково розглядається 11 специфічних атрибутів якості (Nascimento, 2020).

**Актуальність роботи.** Вплив на якість програмного забезпечення для біонічних протезів біологічних чинників суттєво впливає на процес оцінювання якості програмного забезпечення. Розповсюджена модель якості SQuaRE стає неадекватною через неможливість визначити важливі характеристики та атрибути якості. Тому існує потреба в визначенні моделі якості програмного забезпечення для біонічних протезів.

**Метою роботи** є розширення моделі якості програмного забезпечення SQuaRE для забезпечення можливості оцінювання та контролю якості програмного забезпечення для біонічних протезів.

**Виклад основного матеріалу дослідження.** Існує багато способів взаємодії протезів і програмного забезпечення, яке дає змогу здійснювати контроль і комунікацію з користувачем. Ці системи характеризуються природною нестабільністю, сильними взаємозв'язками та нелінійністю.

**Способи отримання даних з біонічних протезів.** Способи отримання даних і передавання даних до біонічного протеза дуже різно-

манітні. Опишемо найбільш розповсюджені на практиці реалізації (Fermin, 2019):

1. Міоелектричний прямий контроль. Всередину кінцівки вбудовуються один або два міоелектричні датчики так, щоб вони контактували зі шкірою на поверхні кінцівки. Датчики призначені для виявлення м'язових рухів. Назва «прямий контроль» дана тому, що користувач повинен активувати певні м'язи, щоб керувати вибором режиму захоплення, а також діями відкриття і закриття.

2. Розпізнавання міоелектричних образів. У нових версіях біонічних протезів впроваджено розпізнавання міоелектричних структур замість прямого управління. Завдяки великій кількості датчиків тепер можна не тільки виявляти окремі м'язові рухи, а й ідентифікувати закономірності в цих рухах. Кожен зразок може бути пов'язаний з конкретною дією. Як результат, управління стає більш інтуїтивним, оскільки користувачу більше не потрібно активно напружувати окремі м'язи. Він просто мислить про переміщення своєї фантомної кінцівки так, ніби вона все ще присутня, і біонічна версія автоматично відображає його наміри.

3. Контроль механоміограми. Датчики механоміограми призначені для реєстрації механічних сигналів, що виникають під час скорочення м'язів, на відміну від датчиків електроміографії (ЕМГ), які реєструють електричні сигнали, що створюються м'язами під час їхньої активації.

Отже, завданням програмного забезпечення є коректна взаємодія і обробка отриманих даних датчиків з урахуванням їх специфіки. У випадку розпізнавання міоелектричних образів варто завжди звертати увагу на те, що зовнішні джерела шуму, такі як вітер, шум міста, серцебиття та інші, можуть впливати на інтерпретацію отриманих даних. Тому програмне забезпечення повинно не лише збирати, передавати та сегментувати дані датчиків, але й перевіряти їх на правильність та відповідність. У випадку використання механоміограми причиною невірних даних можуть виступати інші чинники, такі як втомленість м'язів, непередбачуваність сигналу, індивідуальні відмінності та інше. Отже, програмне забезпечення повинно не лише вміти збирати дані та працювати з ними, але й правильно їх інтерпретувати, сегментувати, бути зручним у використанні, розширюваним і, звичайно ж, безпечним.

**Передача тактильних відчуттів і зворотного зв'язку.** Частиною системи взаємодії між користувачем і біонічним протезом є передача тактильних відчуттів. Основними способами передачі тактильних відчуттів є (Fermin, 2019):

1. Остеоперцепція. Вібрації, що передаються від біонічної кінцівки через остеоінтегрований імплантат у кістку кукси кінцівки, є потенційно багатим джерелом інформації.

2. Простий тактильний зворотний зв'язок. Датчики, вбудовані в біонічні протези, зворотно зв'язуються з вібраторами, розміщеними в гнізді протеза. Ці вібратори використовують різні рівні і частоти вібрації для передачі інформації. Приклади такого використання включають передачу контакту з об'єктом (одиначна вібрація), а також сили стиснення (повторювані вібрації, можливо, зі збільшеною інтенсивністю).

3. Електрична стимуляція нервів. Системи зворотного зв'язку прагнуть стимулювати нервові клітини через шкіру і проміжні тканини. Це може призвести до того, що мозок сприйме певні відчуття у відсутній кінцівці, які можна абстрактно порівняти із сенсорним зворотним зв'язком від біонічного протезу. Наприклад, відчуття поколювання у відсутній руці може виникнути, коли біонічна рука взаємодіє з об'єктом. Це може надати користувачу більш реалістичні відчуття, ніж простий тактильний зворотний зв'язок, але обмеження інформації, яку можна передати в такий спосіб, залишаються суттєвими.

Стосовно аспекту передачі тактильних відчуттів програмне забезпечення повинно акцентувати не на сегментації або фільтрації шумів, а на характеристиках (сили або обмеженнях) керівних сигналів, що впливають на тактичні відчуття. Важливо на етапі проектування програмного забезпечення забезпечити коректне відтворення керівних сигналів, наприклад, сили натиску для вібрації, або мінімальної та максимальної напруги для електронної стимуляції. Ця інформація має бути строго конфіденційною та передаватися безпечним способом, що безумовно також має враховуватися при розробці методів взаємодії та впливу.

#### **Математичний апарат обробки даних.**

Отримані дані потребують адекватної обробки з метою забезпечення керування біонічним протезом. З цією метою застосовують різні математичні моделі і методи. Перелічимо найбільш вживані.

Метод Кейна являє собою метод моделювання динамічних систем, який дозволяє автоматично визначати динамічні рівняння через концепцію узагальнених швидкостей, часткових швидкостей і сил інерції (Menga, 2018). Цей метод дає змогу спростити математичне підґрунтя та обробляти складніші проблеми. Він також забезпечує вирази для моделювання та розробки керування, а також генерує прототипи

вихідного коду для рівнянь у вигляді звичайних диференціальних рівнянь.

Моделювання активності аферентів (Hardesty, 2020) – це моделювання, яке здійснюється з використанням рівнянь

$$Ia(v,l) = Av^{0.5} + Bl + C, \quad (1)$$

де  $Ia$  – швидкість аферентної активації,  $l$  – довжина м'яза, що змінюється в часі,  $v$  – швидкість її зміни та  $A, B, C$  – визначені експериментом константи.

Формула (1) дозволяє приблизно визначити активність аферентів під час руху. Це допомагає дослідникам зрозуміти, які зміни у фузиторному приводі можуть впливати на коактивацію м'язів.

Ієрархічна кластеризація – це метод аналізу структури кластерів, який використовується для порівняння структури кластерів довжини м'язів у різних положеннях із кластерною структурою стимульованої активності первинних аферентів. Це дає змогу оцінити схожість між довжиною м'язів і профілями активності аферентів за формулою

$$HVE = \begin{cases} 1 - r^2, & |r| > 0, p < \alpha \\ 1 + r^2, & |r| < 0, p < \alpha \\ 1, & |p| \geq \alpha \end{cases} \quad (2)$$

де  $r$  – коефіцієнт кореляції Пірсона, який вимірює силу і напрямок лінійного зв'язку між двома змінними,  $p$  – рівень значущості, який вказує, наскільки ймовірно отримати такі самі або більш екстремальні результати за умови, що нульова гіпотеза вірна, а – використовується для позначення параметра або коефіцієнта в рівнянні.

Часто застосовують методи статистичного аналізу (Lotti, 2020) як методи аналізу даних, що використовується для інтерпретації результатів експериментів і досліджень. Вони дозволяють оцінити відмінності між групами, визначити зв'язки між змінними та перевірити гіпотези. Зокрема, метод Колмогорова-Смирнова використовується для перевірки того, наскільки вибірка даних відповідає відомому розподілу. Він дає змогу оцінити, наскільки добре дані відповідають теоретичному розподілу. Метод дисперсійного аналізу (ANOVA) використовується для порівняння середніх значень трьох або більше груп, щоб визначити, чи є статистично значущі відмінності між ними. ANOVA дає змогу визначити, чи впливають різні чинники на залежну змінну.

Ці та інші методи використовуються на етапі моделювання поведінки за допомогою різних програм, таких як OpenSim (Camargo, 2022), що дають змогу моделювати можливі дані або

ситуації з подальшим застосуванням до математичного опрацювання та аналізу.

Оскільки математичний апарат має забезпечувати високу точність розрахунків, важливо враховувати, що його застосування повинно бути строго регламентовано і піддано глибокому аналізу і тестуванню перед тим, як буде використане в програмному забезпеченні. Якість програмного забезпечення повинна враховувати можливі відхилення, аналіз експериментів і досвідного використання, а самі алгоритми та методи повинні бути спробовані спочатку в модельних середовищах.

**Модель якості програмного забезпечення для біонічних протезів.** Наразі в галузі біонічного протезування працює багато компаній, які застосовують різні програмні рішення. В біонічних протезах найчастіше використовують вбудоване програмне забезпечення, але у багатьох версіях є і віддалене програмне забезпечення, що дає змогу протезу оновлювати компоненти, передавати аналітичні дані, взаємодіяти з про-

тезом. Програмне забезпечення функціонує шляхом збирання сигналів від датчиків і передавання їх до компонентів, відповідальних за оброблення даних.

Після узагальнення способів взаємодії протезів з програмним забезпеченням, а також описів наявних промислових рішень, можемо стверджувати, що модель якості містить вісім ключових характеристик (таблиця 1).

Як можна побачити, частина ключових характеристик – надійність, зручність використання, захищеність – є характеристиками моделі SQuaRE. Але додаються і специфічні характеристики, такі як безпека, адаптивність, довговічність.

**Висновки.** Вивчення різних методів отримання даних, передачі тактильних відчуттів і зворотного зв'язку та методів математичної обробки показало, що програмне забезпечення відіграє ключову роль у забезпеченні функціональності, зручності використання та загальної якості біонічних кінцівок.

Таблиця 1

#### Характеристики якості програмного забезпечення для біонічних протезів

Характеристика якості	Опис характеристики
Надійність	Програмне забезпечення біонічних протезів повинно бути високонадійним, щоб забезпечити стабільну продуктивність і роботу в різних умовах. Надійність має вирішальне значення для того, щоб протези функціонували так, як очікується, без несподіваних збоїв.
Безпека	Програмне забезпечення повинно відповідати суворим стандартам і протоколам безпеки, щоб мінімізувати ризик травмування користувачів. Це включає в себе надійну обробку помилок, відмовистійкі механізми та ретельне тестування для виявлення та зменшення потенційних загроз безпеці.
Зручність використання	Програмне забезпечення повинно мати інтерфейс користувача, який дозволяє користувачам, часто з обмеженими можливостями, ефективно і комфортно контролювати свої протези. Крім того, налаштування та адаптивні функції можуть підвищити зручність використання, задовольняючи індивідуальні уподобання та потреби користувачів.
Швидкість реагування	Програмне забезпечення повинно швидко реагувати на дії користувача, забезпечуючи зворотний зв'язок у реальному часі та плавний контроль над протезом. Низька латентність і мінімальна затримка мають важливе значення, особливо під час діяльності, що вимагає точних рухів і координації.
Адаптивність	Програмне забезпечення для біонічних протезів має бути адаптивним, щоб пристосовуватися до змін у фізіології користувача, навколишньому середовищі та моделях використання з плином часу. Тобто має бути забезпечені постійна оптимізація та налаштування функціональності протеза на основі фідбеків і даних користувача.
Довговічність	Програмне забезпечення має бути розроблене таким чином, щоб витримувати суворі умови щоденного використання і потенційний знос, пов'язаний з протезами. Це має призвести до мінімізації потреби в обслуговуванні або заміні протезу.
Інтероперабельність	Програмне забезпечення біонічних протезів має бути сумісним з різними апаратними компонентами, датчиками та комунікаційними протоколами, щоб полегшити безперешкодну інтеграцію з іншими допоміжними технологіями та медичними пристроями.
Захищеність	Захист даних користувача і забезпечення конфіденційності чутливої інформації є критично важливими для програмного забезпечення для біонічних протезів. Надійне шифрування, механізми автентифікації та контролю доступу повинні бути впроваджені для захисту конфіденційності користувачів і запобігання несанкціонованому доступу або втручанню в роботу протеза.

Методи отримання даних, такі як міоелектричний прямий контроль і контроль механоміограми, підкреслюють важливість надійних програмних рішень для точної інтерпретації намірів користувача і перетворення їх у точні рухи протеза. Аналогічно, передача тактильних відчуттів і зворотного зв'язку підкреслює важливість програмного забезпечення в наданні користувачам реалістичних сенсорних відчуттів, що покращує їхню взаємодію з протезом.

Методи математичного моделювання дають цінну інформацію про рухи користувача і коактивацію м'язів, допомагаючи в розробці більш інтуїтивно зрозумілих і чутливих систем управління протезами. Крім

того, запропонована модель якості окреслює ключові характеристики, необхідні для забезпечення надійності, безпеки, адаптивності та інтероперабельності програмного забезпечення біонічних протезів, що в кінцевому підсумку підвищує задоволеність і безпеку користувачів.

Це дослідження підкреслює невід'ємну роль розробки програмного забезпечення у формуванні майбутнього технологій біонічного протезування. Постійно впроваджуючи інновації та вдосконалюючи програмні рішення, ми можемо ще більше розширити можливості людей з втраченою кінцівкою вести більш незалежне та повноцінне життя.

#### ЛІТЕРАТУРА:

1. Bumbaširević M., Lesic A., Palibrk T., Milovanovic D., Zoka M., Kravić-Stevović T., Raspopovic S. The current state of bionic limbs from the surgeon's viewpoint. *EFORT Open Rev.* 2020. 5(2). P. 65–72. doi: 10.1302/2058-5241.5.180038. PMID: 32175092; PMCID: PMC7047902.
2. Abrams Z. A New Era for Bionic Limbs. *IEEE Pulse.* 2023. 14(1). P. 12–15. doi: 10.1109/MPULS.2023.3243313. PMID: 37028369.
3. Nascimento E., Nguyen-Duc A., Sundbø I., Conte T. Software engineering for artificial intelligence and machine learning software: A systematic literature review. arXiv preprint. 2020. doi: 10.48550/arXiv.2011.03751.
4. Hong-liu Y., Xing-san Q., Ling S. (2009) Analysis and comparison of intelligent control methods for computer-controlled artificial leg. *Rehabilitation Engineering & Assistive Technology (i-CREATE '09)* : Proceedings of the 3rd International Convention, New York, NY, USA, Article 22, 1–5. doi: 10.1145/1592700.1592725.
5. Menga G., Ghirardi M. Lower Limb Exoskeleton for Rehabilitation with Improved Postural Equilibrium. *Robotics.* 2018. 7(2). 28. doi: 10.3390/robotics7020028
6. Hardesty R. L., Boots M. T., Yakovenko S. et al. Computational evidence for nonlinear feedforward modulation of fusimotor drive to antagonistic co-contracting muscles. *Sci Rep.* 2020. 10. 10625. doi: 10.1038/s41598-020-67403-w.
7. Lotti N. et al. Adaptive Model-Based Myoelectric Control for a Soft Wearable Arm Exosuit: A New Generation of Wearable Robot Control. *IEEE Robotics & Automation Magazine.* 2020. Vol. 27(1). P. 43–53. doi: 10.1109/MRA.2019.2955669.
8. Camargo J., Bhakta K., Maldonado-Contreras J., Zhou S., Herrin K., Young A. OpenSim Model for Biomechanical Analysis with the Open-Source Bionic Leg. *Medical Robotics (ISMR)* : Proceedings of International Symposium, GA, USA, 2022, pp. 1–6. doi: 10.1109/ISMR48347.2022.9807551.
9. Aragon F. C., Huegel J. C., Vargas-Martinez A., Lozoya-Santos J. d. J., Ramirez-Mendoza R., Morales-Menendez R. Comparison between Classic Control Systems Techniques against Adaptive and Nonlinear Control Techniques in a Lower Limb Prosthesis. *Control and Robotics Engineering (ICCRE)* : Proceedings of 4th International Conference, Nanjing, China, 2019, pp. 75–78. doi:10.1109/ICCRE.2019.8724210
10. Donaldson N., Brindley G. S. The Historical Foundations of Bionics. *Engineering.* 2016. P. 1–37. doi: 10.1002/9781118816028.ch1

#### REFERENCES:

1. Bumbaširević, M., Lesic, A., Palibrk, T., Milovanovic, D., Zoka, M., Kravić-Stevović, T., Raspopovic, S. (2020). The current state of bionic limbs from the surgeon's viewpoint. *EFORT Open Rev.*, 5(2), pp. 65–72. doi: 10.1302/2058-5241.5.180038. PMID: 32175092; PMCID: PMC7047902.
2. Abrams, Z. (2023). A New Era for Bionic Limbs. *IEEE Pulse*, 14(1), pp. 12–15. doi: 10.1109/MPULS.2023.3243313. PMID: 37028369.
3. Nascimento, E., Nguyen-Duc, A., Sundbø, I., Conte, T. (2020). Software engineering for artificial intelligence and machine learning software: A systematic literature review. arXiv preprint. doi: 10.48550/arXiv.2011.03751.
4. Hong-liu, Y., Xing-san, Q., Ling, S. (2009). Analysis and comparison of intelligent control methods for computer-controlled artificial leg. *Proceedings of the 3rd International Convention on Rehabilitation Engineering*



& Assistive Technology (i-CREATe '09). Association for Computing Machinery, New York, NY, USA, Article 22, 1–5. doi: 10.1145/1592700.1592725.

5. Menga, G., Ghirardi, M. (2018). Lower Limb Exoskeleton for Rehabilitation with Improved Postural Equilibrium. *Robotics*, 7(2), 28. doi: 10.3390/robotics7020028

6. Hardesty, R. L., Boots, M. T., Yakovenko, S. et al. (2020). Computational evidence for nonlinear feedforward modulation of fusimotor drive to antagonistic co-contracting muscles. *Sci Rep*, 10, 10625. doi: 10.1038/s41598-020-67403-w.

7. Lotti, N. et al. (2020). Adaptive Model-Based Myoelectric Control for a Soft Wearable Arm Exosuit: A New Generation of Wearable Robot Control. *IEEE Robotics & Automation Magazine*, vol. 27(1), pp. 43–53. doi: 10.1109/MRA.2019.2955669.

8. Camargo, J., Bhakta, K., Maldonado-Contreras, J., Zhou, S., Herrin, K., Young, A. (2022). OpenSim Model for Biomechanical Analysis with the Open-Source Bionic Leg. 2022 International Symposium on Medical Robotics (ISMR), GA, USA, pp. 1–6. doi: 10.1109/ISMR48347.2022.9807551.

9. Aragon, F. C., Huegel, J. C., Vargas-Martinez, A., Lozoya-Santos, J. d. J., Ramirez-Mendoza, R., Morales-Menendez, R. (2019). Comparison between Classic Control Systems Techniques against Adaptive and Nonlinear Control Techniques in a Lower Limb Prostheses. 2019 4th International Conference on Control and Robotics Engineering (ICCRE), Nanjing, China, pp. 75–78. doi:10.1109/ICCRE.2019.8724210

10. Donaldson, N., Brindley, G. S. (2016). The Historical Foundations of Bionics. *Engineering*, pp. 1–37. doi: 10.1002/9781118816028.ch1

УДК 004.9

DOI <https://doi.org/10.32782/IT/2023-4-7>

### **Тетяна СЕЛІВЬОРСТОВА**

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій і систем, факультет прикладних комп'ютерних технологій, Український державний університет науки і технологій, просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, [tatyanamikhaylovskaya@gmail.com](mailto:tatyanamikhaylovskaya@gmail.com)

ORCID: 0000-0002-2470-6986

### **Никита КРАСНОШАПКА**

аспірант кафедри інформаційних технологій і систем, ННІ «Інститут промислових та бізнес технологій» Український державний університет науки і технологій, просп. Дмитра Яворницького, 19, м. Дніпро, Україна, 49005, [nikkiredhood@gmail.com](mailto:nikkiredhood@gmail.com)

ORCID: 0009-0002-8127-1410

**Бібліографічний опис статті:** Селівьорстова, Т., Красношапка, Н. (2023). Особливості проєктування масштабованої мікросервісної архітектури для вебсервісів. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 58–66, doi: <https://doi.org/10.32782/IT/2023-4-7>

## **ОСОБЛИВОСТІ ПРОЄКТУВАННЯ МАСШТАБОВАНОЇ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ ДЛЯ ВЕБСЕРВІСІВ**

**Мета роботи.** Метою цієї роботи є розробка та аналіз масштабованої мікросервісної архітектури, здатної забезпечити високу доступність та ефективність інтеграції з хмарними сервісами. Особлива увага приділяється створенню оптимізованих методів розгортання, моніторингу та обслуговування мікросервісів у динамічних умовах використання, а також оцінці впливу кількості сервісів на продуктивність системи. Дослідження спрямоване на заповнення прогалів у наявних методах оркестрації мікросервісів, забезпечуючи підвищення їх ефективності та масштабованості.

**Методологія.** У цьому дослідженні ми застосували комплексний підхід, що включає кілька ключових методів:

– **Теоретичний аналіз:** Виконано систематичний огляд літератури для ідентифікації існуючих мікросервісних архітектур і виявлення потенційних областей для поліпшення. Особлива увага була приділена дослідженню можливостей масштабування і відмовостійкості.

– **Програмування та розробка:** Розробка прототипів мікросервісів з використанням сучасних мов програмування та фреймворків. Реалізація включала в себе створення RESTful API, використання контейнеризації через Docker, та оркестрацію за допомогою Kubernetes.

– **Бенчмаркінг та тестування:** Проведено низку тестів продуктивності для оцінки масштабованості і швидкодії мікросервісів. Використання засобів навантажувального тестування, таких як JMeter, та моніторингу, як Prometheus, для збору метрик продуктивності.

– **Аналіз даних:** Збір та статистичний аналіз даних для визначення закономірностей і виявлення вузьких місць в архітектурі, для прогнозування поведінки системи при різних сценаріях навантаження.

Ці методи та підходи були інтегровані для розробки та аналізу масштабованої мікросервісної архітектури, що дозволило оцінити її ефективність і визначити оптимальні конфігурації для різних типів навантажень і бізнес-вимог.

**Наукова новизна.** У цій статті ми представляємо ряд важливих нововведень у сфері проєктування масштабованих мікросервісних архітектур:

– **Розробка інноваційної моделі масштабування:** Наша розроблена модель відрізняється від існуючих підходів здатністю ефективно масштабуватися у великих розподілених системах, що враховує динамічність навантаження та розподіл ресурсів.

– **Застосування нових методів контейнеризації:** Ми впровадили новий спосіб використання Docker та Kubernetes для оптимізації розгортання мікросервісів, що забезпечує значно краще використання ресурсів і знижує час відгуку системи.

– **Розширений статистичний аналіз продуктивності:** Використання передових методів статистичного аналізу для оцінки продуктивності мікросервісів дозволило отримати нові знання про фактори, які впливають на масштабованість і ефективність.

– **Практична цінність:** Наші висновки та розробки мають значний потенціал для покращення роботи реальних вебсервісів та додатків, забезпечуючи їм високу продуктивність і доступність, що є критично важливим для сучасних інформаційних технологій.

**Висновки.** У результаті проведеного нами дослідження ми досягли наступних ключових результатів:

– Підтвердження ефективності розробленої моделі масштабування: Наша інноваційна модель масштабування мікросервісів показала значне поліпшення в управлінні ресурсами і часом відгуку в порівнянні з традиційними підходами. Це було підтверджено за допомогою експериментів, що включали стрестування та аналіз продуктивності.

– Використання контейнеризації як ключового елементу для оптимізації розгортання: Застосування Docker та Kubernetes дозволило нам ефективно масштабувати сервіси з мінімальними затратами ресурсів, що демонструє великий потенціал для практичного впровадження в реальних системах.

– Розширення знань про масштабування мікросервісів: Наше дослідження внесло важливий вклад у розуміння факторів, що впливають на продуктивність і масштабування мікросервісних архітектур, забезпечуючи цінні інсайти для майбутніх досліджень та розробок.

– Практичне застосування та вплив на індустрію: Встановлено, що наші розробки можуть значно покращити продуктивність і доступність вебсервісів та додатків, що є особливо актуальним для сучасного цифрового світу, де швидкість реакції та ефективність обслуговування клієнтів мають вирішальне значення.

Ці висновки демонструють не лише теоретичну цінність нашого дослідження, але й його практичний потенціал для впровадження в реальних бізнес-сценаріях, відкриваючи нові можливості для розвитку інформаційних технологій.

**Ключові слова:** архітектура мікросервісів, масштабованість, вебсервіси, оркестровка, продуктивність.

### **Tetiana SELIVIORSTOVA**

Candidate of Technical Sciences, Associate Professor, Associate Professor at the Department of Information Technologies and Systems, Faculty of Applied Computer Technologies, Ukrainian State University of Science and Technology, 19, Dmytra Yavornytskogo Ave, Dnipro, Ukraine, 49005, tatyamikhaylovskaya@gmail.com

### **Nykyta KRASNOSHAPKA**

Postgraduate student of the Department of Information Technologies and Systems, Research Institute "Institute of Industrial and Business Technologies" Ukrainian State University of Science and Technology, 19, Dmytra Yavornytskogo Ave, Dnipro, Ukraine, 49005, nikkiredhood@gmail.com

**To cite this article:** Seliviorstova, T., Krasnoshapka, N. (2023). Osoblyvosti proiektuvannia masshtabovanoi mikroservisnoi arkhitektury dlia vebservisiv [Aspects of Designing Scalable Microservices Architecture for Web Services]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 58–66, doi: <https://doi.org/10.32782/IT/2023-4-7>

## **ASPECTS OF DESIGNING SCALABLE MICROSERVICES ARCHITECTURE FOR WEB SERVICES**

**Objective of the study.** The objective of this work is to develop and analyze a scalable microservice architecture capable of ensuring high availability and efficient integration with cloud services. Particular attention is paid to creating optimized methods for deployment, monitoring, and maintenance of microservices under dynamic usage conditions, as well as evaluating the impact of the number of services on system performance. The research is aimed at filling gaps in existing orchestration methods of microservices, enhancing their efficiency and scalability.

**Methodology.** In this study, we applied a comprehensive approach that includes several key methods:

– **Theoretical Analysis:** A systematic literature review was conducted to identify existing microservice architectures and to discover potential areas for improvement. Special attention was paid to studying scalability and fault tolerance capabilities.

– **Programming and Development:** Prototypes of microservices were developed using modern programming languages and frameworks. The implementation included the creation of RESTful APIs, the use of containerization through Docker, and orchestration using Kubernetes.

– **Benchmarking and Testing:** A series of performance tests were conducted to assess the scalability and speed of microservices. Load testing tools such as JMeter, as well as monitoring tools like Prometheus, were used to gather performance metrics.

– **Data Analysis:** Collection and statistical analysis of data to identify patterns and pinpoint bottlenecks in the architecture, and to predict system behavior under different load scenarios.

These methods and approaches were integrated to develop and analyze a scaled microservice architecture, which made it possible to evaluate its effectiveness and determine optimal configurations for different types of equipment. y and business-viable.

**Scientific novelty.** In this article, we present a series of important innovations in the field of designing scalable microservice architectures:

– *Development of an Innovative Scaling Model: Our developed model differs from existing approaches by its ability to scale effectively in large distributed systems, taking into account the dynamism of load and resource distribution.*

– *Application of New Containerization Methods: We have introduced a new way of using Docker and Kubernetes to optimize the deployment of microservices, which ensures significantly better resource utilization and reduces system response time.*

– *Advanced Statistical Analysis of Performance: The use of advanced statistical analysis methods to assess the performance of microservices has enabled us to gain new insights into the factors that affect scalability and efficiency.*

– *Practical Value: Our findings and developments have significant potential to improve the operation of real web services and applications, providing them with high performance and availability, which is critically important for modern information technologies.*

**Conclusions.** As a result of our research, we have achieved the following key outcomes:

– *Validation of the effectiveness of the developed scaling model: Our innovative microservices scaling model has shown significant improvement in resource management and response time compared to traditional approaches. This was confirmed through experiments that included stress testing and performance analysis.*

– *Using containerization as a key element for deployment optimization: The application of Docker and Kubernetes allowed us to scale services efficiently with minimal resource expenditure, demonstrating great potential for practical implementation in real systems.*

– *Expanding knowledge about scaling microservices: Our research has made an important contribution to understanding the factors affecting the performance and scalability of microservice architectures, providing valuable insights for future research and development.*

– *Practical application and impact on the industry: We have established that our developments can significantly improve the performance and availability of web services and applications, which is particularly relevant in the modern digital world where speed of response and efficiency of customer service are crucial.*

*These conclusions demonstrate not only the theoretical value of our research but also its practical potential for implementation in real business scenarios, opening new possibilities for the development of information technologies.*

**Key words:** microservices architecture, scalability, web services, orchestration, performance.

## Теоретичний аналіз Мікросервісів

Мікросервіси – це методологія розробки програмного забезпечення, яка полягає у використанні набору невеликих, автономних служб, що спілкуються через легкі механізми. Цей підхід відрізняється від традиційної монолітної архітектури, де всі функції програми втілені в одному проєкті.

### Переваги мікросервісів

**Незалежність Розгортання:** Кожен мікросервіс може розгортатися, оновлюватися та масштабуватися незалежно.

**Гнучкість у Виборі Технологій:** Різні мікросервіси можуть використовувати різні мови програмування та технологічні стеки.

**Поліпшення Масштабованості та Надійності:** Невеликі, автономні сервіси легше масштабувати та відновлювати після збоїв.

### Виклики мікросервісної архітектури

**Складність у Координації та Управлінні:** Управління численними сервісами вимагає ефективних механізмів координації та моніторингу.

**Забезпечення Безпеки та Логування:** Необхідно ретельно планувати стратегії безпеки та логування для розподілених систем.

**Методологія 12-факторних додатків:** Ця методологія надає керівництво для створення мікросервісів, орієнтованих на автоматизацію процесів та незалежність сервісів.

## Контейнеризація та Оркестрація

**Контейнеризація:** Використання контейнерів, таких як Docker, дозволяє упаковувати мікросервіси з усіма їх залежностями та конфігурацією.

**Оркестрація Контейнерів:** Інструменти, такі як Kubernetes, дозволяють управляти розгортанням, масштабуванням та забезпеченням високої доступності контейнерів.

### Приклади Застосування Мікросервісів

**E-commerce Платформи:** Мікросервіси дозволяють ізолювати функції, такі як обробка замовлень, управління запасами, обслуговування клієнтів.

**Медіа-Сервіси:** Швидке впровадження нових функцій та масштабування під велику кількість користувачів.

### Комунікація між Мікросервісами

**Синхронна комунікація:** Через HTTP REST або gRPC.

**Асинхронна комунікація:** Через черги повідомлень, такі як RabbitMQ або Apache Kafka.

**Виявлення та Реєстрація Сервісів:** Важливим аспектом є механізми виявлення та реєстрації сервісів, такі як Eureka або Consul, що дозволяють сервісам ефективно спілкуватися між собою.

### Висновок

Мікросервісна архітектура відкриває нові можливості для гнучкості, масштабування

та швидкого розвитку програмних продуктів. Однак, необхідно враховувати виклики, пов'язані з управлінням, безпекою та координацією між сервісами.

### Програмування та розробка

#### Створення мікросервісів

#### Розгортання мікросервісів за допомогою Docker та Docker Compose

Для розгортання мікросервісної архітектури з використанням Docker, спочатку було визначено кількість менеджерів та робочих вузлів, які повинні бути створені. Використовуючи docker-machine, було автоматизовано процес створення віртуальних машин, на яких будуть розміщені сервіси. Потім, за допомогою docker-swarm, було організовано ці машини у кластер, забезпечуючи високу доступність та масштабування.

Наведемо приклад Bash-скрипта для створення Swarm-кластера (рис. 1).

Після створення кластера, було сконфігуровано docker-compose.yml файл для опису сервісів, мережі та об'ємів. Docker Compose дозволяє визначити та запустити багатоконтейнерні Docker додатки, використовуючи простий YAML файл. Це особливо корисно для мікросервісних архітектур, де кожен сервіс може бути описаний як окремий контейнер.

Приклад файлу docker-compose.yml для мікросервісного додатку (рис. 2).

У цьому файлі описано три мікросервіси: gateway, user-service, та order-service, кожен з яких розгортається як контейнер. Вказано кількість реплік кожного сервісу, що забезпечує

необхідне навантаження та відмовостійкість. Також визначено мережу mynetnetwork, яка дозволяє контейнерам взаємодіяти між собою.

Запуск сервісів з docker-compose.yml виконується командою: **docker-compose up -d**

#### Моніторинг та логування

Моніторинг та логування в мікросервісній архітектурі відіграють важливу роль у забезпеченні стійкості та оперативності виявлення та усунення проблем. В рамках цієї практичної частини було впроваджено ELK Stack для збору, обробки та аналізу логів від окремих мікросервісів.

#### Конфігурація Logstash:

Logstash використовується для агрегування та нормалізації логів перед їх надсиланням до Elasticsearch. Нижче наведено приклад конфігурації Logstash, яка зчитує логи з файлу і передає їх до Elasticsearch (рис. 3).

Ця конфігурація визначає вхідний плагін file, який зчитує логи з певного шляху, filter з використанням grok для розбору та структурування повідомлень логів, і output для відправки оброблених даних у Elasticsearch.

Після збору та індексації логів у Elasticsearch, Kibana використовується для їх візуалізації. Це дозволяє створювати дашборди, які відображають важливі метрики системи, наприклад, кількість помилок HTTP за часовий період, час відповіді сервісів тощо.

#### Аналіз продуктивності та масштабованості

Дані були зібрані з усіх мікросервісів і збережені у форматі CSV для подальшого аналізу.

```
# Конфігурація кількості вузлів
MANAGER_COUNT=3
WORKER_COUNT=5

# Створення менеджерів
for i in $(seq 1 $MANAGER_COUNT); do
    docker machine create --driver virtualbox manager$i
done

# Створення робочих вузлів
for i in $(seq 1 $WORKER_COUNT); do
    docker machine create --driver virtualbox worker$i
done

# Ініціалізація Swarm кластера на першому менеджері
docker machine ssh manager1 "docker swarm init --advertise-addr $(docker machine ip manager1)"

# Отримання жокена для робочих вузлів
WORKER_JOIN_TOKEN=$(docker machine ssh manager1 "docker swarm join-token worker -q")

# Додавання робочих вузлів до кластера
for i in $(seq 1 $WORKER_COUNT); do
    docker machine ssh worker$i "docker swarm join --token $WORKER_JOIN_TOKEN $(docker machine ip manager1):2377"
done

echo "Swarm кластер успішно створено."
```

Рис. 1

```

version: '3.8'

services:
  gateway:
    image: my-gateway-service:latest
    ports:
      - "8080:8080"
    networks:
      - mynetwork
    deploy:
      replicas: 2
      update_config:
        parallelism: 2
      delay: 10s

  user-service:
    image: my-user-service:latest
    networks:
      - mynetwork
    deploy:
      replicas: 3
      update_config:
        parallelism: 1
      delay: 10s

  order-service:
    image: my-order-service:latest
    networks:
      - mynetwork
    deploy:
      replicas: 3
      update_config:
        parallelism: 1
      delay: 10s

networks:
  mynetwork:

```

Рис. 2

Використовуючи мову програмування R, були розроблені регресійні моделі для прогнозування поведінки системи при масштабуванні. Аналіз дозволив визначити вплив кількості контейнерів на загальну продуктивність системи. Для оцінки продуктивності мікросервісів були визначені ключові метрики, такі як час відгуку (латентність) та пропускна здатність (кількість оброблених запитів на одиницю часу). Дані метрики були зібрані в умовах різних рівнів навантаження, використовуючи інструменти як JMeter та Locust для симуляції трафіку користувачів (рис. 4).

Цей код аналізує взаємозв'язок між кількістю контейнерів та часом відгуку, дозволяючи візуально оцінити, як зміна кількості контейнерів впливає на продуктивність.

Застосування Закону Амдала дозволило прогнозувати максимально можливу продуктивність системи при збільшенні кількості контейнерів. Отримані значення  $\sigma$  вказують на мінімальний вплив конкуренції між контейне-

```

input {
  file {
    path => "/var/log/microservices/*.log"
    start_position => "beginning"
  }
}

filter {
  grok {
    match => { "message" => "%{TIMESTAMP_ISO8601:timestamp} %{LOGLEVEL:loglevel} %{GREEDYDATA:message}" }
  }
  date {
    match => [ "timestamp", "ISO8601" ]
  }
}

output {
  elasticsearch {
    hosts => ["elasticsearch:9200"]
    index => "microservices-%{+YYYY.MM.dd}"
  }
}

```

Рис. 3

```

library(ggplot2)
data <- read.csv('data.csv')

# Лінійна регресійна модель
model <- lm(response_time ~ number_of_containers, data = data)

# Візуалізація результатів
ggplot(data, aes(x=number_of_containers, y=response_time)) +
  geom_point() +
  geom_smooth(method='lm') +
  theme_minimal() +
  labs(title="Вплив кількості контейнерів на час відгуку")

```

Рис. 4

рами, що є позитивним аспектом для масштабованості. Результати тестування представлені в таблиці 1.

Таблиця 1

Кількість контейнерів	Середній час затримки (мс)
10	262.78
20	178.140
30	190.40
40	164.70
50	289.30
60	416.64
70	446.34

Для оцінки масштабованості архітектури мікросервісів був використаний загальний закон масштабованості. Час запуску контейнера був використаний як час відгуку, а продуктивність визначалась шляхом множення оберненого значення часу запуску на кількість контейнерів. Масштабованість була виміряна на рівні оркестрації та на рівні додатку. Для вимірювання масштабованості на рівні оркестрації були використані дані. Для перевірки масштабованості за допомогою загального закону масштабованості використовувалась програма R (табл. 2).

Таблиця 2

Кількість контейнерів	Середній час затримки (мс)	Пропускна здатність = кількість контейнерів/затримка час
10	262.78	38.16794
20	178.140	112.3596
30	190.40	157.8947
40	164.70	243.9024
50	289.30	173.0104
60	416.64	144.2308
70	446.34	156.9507

Для побудови графіків максимальної продуктивності в залежності від кількості контейнерів (функціональна масштабовність) були використані виміряні дані максимальної продуктивності. Аналіз регресії визначає, яка з моделей масштабованості найкраще описує дані. Деталі аналізу обговорюються в роботі (Williams і Smith 2004). Аналіз надає параметри моделі, які потім використовуються для екстраполяції поведінки до більшої кількості контейнерів.

Аналіз регресії показує, що найкраще відповідність виміряним даним забезпечує Закон Амдала. Функціональна масштабовність цього додатку описується рівнянням 1 (Williams і Smith 2004).

$$X_{\max}(p) = \frac{X_{\max}(1) \times p}{1 + \sigma(p-1)}$$

Де:

p – кількість контейнерів,

Xmax(1) – максимальна продуктивність з одним контейнером,

Xmax(p) – максимальна продуктивність з p контейнерами,

σ – частка роботи, яка виконується послідовно.

Значення σ, отримане з аналізу регресії, дорівнює 0.0000. Це означає, що вплив конкуренції на масштабованість є мінімальним.

$$X_{\max}(p) = X_{\max}(1) \times p$$

Застосування екстраполяції за допомогою Закону Амдала показує, що максимальна продуктивність з 100 контейнерами буде становити 380 запитів, а з 1000 контейнерами – 3800 запитів. Таким чином, необхідну продуктивність 2000 можна досягти з 526 контейнерами (2000/380 \* 100).

Аналіз показав, що при певній кількості контейнерів (близько 40 у нашому випадку) продуктивність починає знижуватися через обмеження системних ресурсів (ОЗУ, диск, ЦПУ). Це вказує на необхідність оптимізації ресурсів або розгляд альтернативних підходів до масштабування, наприклад, вертикального масштабування або оптимізації коду мікросервісів.

#### Безпека в мікросервісній архітектурі

Здійснено розробку політик безпеки для мікросервісів, включаючи автентифікацію та авторизацію, шифрування даних та безпечну комунікацію між сервісами за допомогою протоколів TLS/SSL. Виконано ревізію коду на предмет вразливостей та впроваджено заходи для їх усунення.

Автентифікація та Авторизація: важливим аспектом безпеки мікросервісів є забезпечення надійної системи автентифікації та авторизації. Для цього було впроваджено OAuth 2.0 та OpenID Connect. Приклад використання OAuth 2.0 для автентифікації (рис. 5).

Цей фрагмент коду Java з Spring Security налаштовує захист ресурсів, дозволяючи публічний доступ до певних ендпоінтів та вимагаючи автентифікації для інших.

Шифрування Даних: для захисту даних, що передаються, використовувалось шифрування TLS/SSL. Налаштування TLS для вебсервера може виглядати наступним чином (рис. 6).

Цей фрагмент конфігурації Nginx включає використання SSL сертифікатів та вказує на безпечні протоколи TLS.



```
@Configuration
@EnableResourceServer
public class ResourceServerConfig extends ResourceServerConfigurerAdapter {

    @Override
    public void configure(HttpSecurity http) throws Exception {
        http
            .authorizeRequests()
            .antMatchers("/public/**").permitAll()
            .antMatchers("/user/**").authenticated();
    }
}
```

Рис. 5

```
server {
    listen 443 ssl;
    server_name myservice.com;
    ssl_certificate /etc/ssl/certs/myservice.com.crt;
    ssl_certificate_key /etc/ssl/private/myservice.com.key;
    ssl_protocols TLSv1.2 TLSv1.3;
    ...
}
```

Рис. 6

Ревізія Коду та Заходи щодо Вразливостей. Було проведено ретельну ревізію коду з метою виявлення та усунення вразливостей. Це включало в себе автоматизовані інструменти аналізу коду, такі як SonarQube, та ручне код-рев'ю експертами з кібербезпеки.

### Балансування навантаження та відмовостійкість

Стратегії Балансування Навантаження

Реалізація балансування навантаження була здійснена для оптимального розподілу трафіку між мікросервісами. Використано балансувальники навантаження, такі як Nginx або HAProxy, які динамічно направляють запити користувачів до відповідних сервісів на основі їхньої поточної завантаженості та доступності.

Впровадження Патернів Відмовостійкості

Для забезпечення стійкості системи до помилок та збоїв були впроваджені патерни відмовостійкості:

Circuit Breaker

Патерн "Circuit Breaker" використовується для запобігання ланцюговій реакції збоїв між сервісами. Якщо мікросервіс стає недоступним, "Circuit Breaker" тимчасово перериває взаємодію з ним, дозволяючи сервісу відновити роботу, не перевантажуючи його зайвими запитами.

Bulkhead

Патерн "Bulkhead" ізолює частини системи таким чином, що збій в одному сервісі не веде до повного збою всієї системи. Це досягається за рахунок обмеження ресурсів (наприклад, пам'яті або потоків) для кожного мікросервісу (рис. 7).

```
http {
    upstream my_microservices {
        server microservice1.example.com;
        server microservice2.example.com;
        server microservice3.example.com;
    }

    server {
        location / {
            proxy_pass http://my_microservices;
        }
    }
}
```

Рис. 7

Ця конфігурація Nginx направляє трафік до групи мікросервісів, забезпечуючи їхнє балансування та високу доступність.

### Тестування та CI/CD

Встановлено процеси неперервної інтеграції та неперервної доставки (CI/CD), які забезпечують автоматизацію тестування та розгортання змін. Використання інструментів, таких як Jenkins або GitLab CI, сприяє підтримці високої якості коду та швидкому впровадженню нових функціональностей.

### Керування конфігурацією

Використання інструментів керування конфігурацією, наприклад, Ansible або Terraform, а також систем управління секретами, як Vault, забезпечує безпечне та централізоване управління налаштуваннями та доступом до важливих даних. Результати тестування для перевірки масштабованої архітектури

### Висновок

У майбутньому мікросервіси стануть впровадженню популярною архітектурою серед програ-



містів, оскільки вони мають значні переваги для розробки високомасштабованих додатків. Однак, як і з будь-якою новою технологією, є кілька перешкод, як-от неправильні бізнес-вимоги. Немає значення, наскільки швидко ми проектуємо, впроваджуємо або масштабуємо систему, якщо бізнес-вимоги некоректні. Для підтримки мікросервісів також потрібні високі навички DevOps, щоб забезпечити їх доступність та безперебійну роботу. Нарешті, розбиваючи монолітну систему на співпрацюючі сервіси, вводяться інтерфейси між сервісами. Кожен з цих сервісів потребує підтримки при випуску нових версій. Очікується, що у найближчому майбутньому з'являться стандарти для мікросервісів. В результаті, архітектурний стиль мікросервісів має перспективи, і додатки поступово перейдуть до архітектури мікросервісів.

Використання контейнеризованих мікросервісів дозволяє досягнути функціональної масштабованості. За допомогою Docker Compose нам вдалося продемонструвати, як можна масштабувати сервіс, вказавши кількість контейнерів через інтерфейс командного рядка. Змінюючи кількість контейнерів для певного мікросервісу, ми змогли досягти бажаної масштабованості.

В ході цієї роботи було реалізовано комплексне рішення для розробки, моніторингу та оптимізації мікросервісної архітектури. Важливість цієї роботи полягає в тому, що вона демонструє глибоке розуміння сучасних вимог до розробки

програмного забезпечення та забезпечення його безперервної роботи та безпеки.

– Розгортання та Конфігурація: Використання Docker та Docker Compose для розгортання мікросервісів забезпечило гнучкість та легкість управління, а також сприяло швидкому масштабуванню.

– Моніторинг та Логування: Впровадження ELK Stack (Elasticsearch, Logstash, Kibana) для моніторингу та аналізу логів дозволило отримати цінну інформацію про стан системи та своєчасно реагувати на потенційні проблеми.

– Аналіз Продуктивності: Використання статистичних методів та регресійного аналізу в R для оцінки продуктивності та масштабованості системи дало можливість зрозуміти вплив різних факторів на загальну продуктивність.

– Безпека: Розробка політик безпеки, включаючи автентифікацію, авторизацію та шифрування, гарантувала захист даних та сервісів від несанкціонованого доступу та інших загроз.

– Балансування Навантаження та Відмовостійкості: Впровадження стратегій балансування навантаження та патернів відмовостійкості забезпечило високу доступність та надійність системи.

Ця робота відображає сучасний підхід до розробки та підтримки високодоступних, безпечних та ефективних мікросервісних систем, підкреслюючи важливість глибокого розуміння як технічних, так і оперативних аспектів їх впровадження та управління.

#### ЛІТЕРАТУРА:

1. Mazlami, J., Cito, J., & Leitner, P. Extraction of Microservices from Monolithic Software Architectures, 2017.
2. P. D. Francesco, I., Malavolta, I., & Lago, P. Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption, 2017.
3. Song, M., Luo, G., & Haihong, E. A Service Discovery System based on Zookeeper with Priority Load Balance Strategy, 2016.
4. Bilgin, B., Unlu, H., & Demirsors, O. Analysis and Design of Microservices: Results from Turkey, 2020.
5. Waseem, M., & Liang, P. Microservices Architecture in DevOps, 2017.
6. Manciola Meloca, R., Re´, R., & Luis Schwerz, A. (2018). An Analysis of Frameworks for Microservices, 2018.
7. Nickoloff, J. Docker in Action, 2016
8. Williams, L. G., & Smith, C. U. Web Application Scalability: A Model-Based Approach, 2004.
9. Newman, S. Building Microservices: Designing Fine-Grained Systems. O'Reilly Media, 2015.
10. Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. Microservices: yesterday, today, and tomorrow, 2017.

#### REFERENCES:

1. Mazlami, J., Cito, J., & Leitner, P. (2017). Extraction of Microservices from Monolithic Software Architectures.
2. P. D. Francesco, I., Malavolta, I., & Lago, P. (2017). Research on Architecting Microservices: Trends, Focus, and Potential for Industrial Adoption.

3. Song, M., Luo, G., & Haihong, E. (2016). A Service Discovery System based on Zookeeper with Priority Load Balance Strategy.
4. Bilgin, B., Unlu, H., & Demirors, O. (2020). Analysis and Design of Microservices: Results from Turkey.
5. Waseem, M., & Liang, P. (2017). Microservices Architecture in DevOps.
6. Manciola Meloca, R., Re', R., & Luis Schwerz, A. (2018). An Analysis of Frameworks for Microservices.
7. Nickoloff, J. (2016). Docker in Action.
8. Williams, L. G., & Smith, C. U. (2004). Web Application Scalability: A Model-Based Approach.
9. Newman, S. (2015). Building Microservices: Designing Fine-Grained Systems. O'Reilly Media.
10. Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: yesterday, today, and tomorrow.

УДК 519.8:338.4

DOI <https://doi.org/10.32782/IT/2023-4-8>

### **Світлана УС**

кандидат фізико-математичних наук, доцент, професор кафедри системного аналізу та управління, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького 19, м. Дніпро, Україна, 49005, [us.s.a@ntu.one](mailto:us.s.a@ntu.one)

ORCID: 0000-0003-0311-9958

Scopus Author ID: 55603096000

### **Любов ТИМОШЕНКО**

кандидат економічних наук, доцент, доцент кафедри прикладної економіки, підприємництва та публічного управління, Національний технічний університет «Дніпровська політехніка», просп. Дмитра Яворницького 19, м. Дніпро, Україна, 49005, [tymoshenko.l.v@ntu.one](mailto:tymoshenko.l.v@ntu.one)

ORCID: 0000-0003-4869-8244

### **Анастасія ЮДИНА**

студентка Національного технічного університету «Дніпровська політехніка», просп. Дмитра Яворницького, 19, м. Дніпро, Україна, [yudina.a.o@ntu.one](mailto:yudina.a.o@ntu.one)

**Бібліографічний опис статті:** Ус, С., Тимошенко, Л., Юдина, А. (2023). Системний аналіз регіонів України для започаткування бізнесу у сфері надання ветеринарних послуг. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 67–78, doi: <https://doi.org/10.32782/IT/2023-4-8>

## **СИСТЕМНИЙ АНАЛІЗ РЕГІОНІВ УКРАЇНИ ДЛЯ ЗАПОЧАТКУВАННЯ БІЗНЕСУ У СФЕРІ НАДАННЯ ВЕТЕРИНАРНИХ ПОСЛУГ**

З формуванням в Україні ринкових умов лікування тварин та надання ветеринарно-санітарних послуг все більше стають прерогативою приватної ветеринарної медицини. Незважаючи на умови жорсткої конкуренції, в якій працюють комерційні заклади та лікарі-підприємці, ринок ветеринарних послуг весь час зростає. Проте ця сфера залишається менш дослідженою і поширеною порівняно з іншими галузями. Отже, започаткування бізнесу у сфері надання ветеринарних послуг, навіть при наявності конкуренції, є перспективним напрямом підприємницької діяльності, це зумовлює актуальність проведеного дослідження.

Метою роботи є здійснення аналізу тенденцій розвитку ветеринарних послуг в різних регіонах України та визначення областей, які можуть бути потенційно привабливими для започаткування ветеринарного бізнесу.

Для цього проведено аналіз кількості закладів ветеринарної медицини у регіонах України, виділено п'ять регіонів, де спостерігається стрімкий розвиток сфери ветеринарних послуг. Ці регіони розглядаються як найбільш перспективні для підприємництва. Крім того, проведено прогноз розвитку ветеринарного бізнесу в трьох регіонах за допомогою різних методів короткострокового прогнозування, зокрема із застосуванням моделей Хольта – Вінтерса, Брауна та методу ковзного середнього. Виявлено найбільш застосовні підходи для прогнозування на два роки. Отримані оцінки прогнозу підтвердили його адекватність.

Проведений аналіз підтвердив потенціал започаткування бізнесу у сфері надання ветеринарних послуг і дозволив визначити найбільш перспективні регіони для започаткування ветеринарного бізнесу. Майбутні наукові дослідження можуть бути спрямовані на моделювання розвитку ветеринарних послуг з урахуванням специфіки обраних регіонів та напрямків ветеринарної медицини. Поглиблене вивчення динаміки цієї галузі дозволить виявити та використати нові можливості для розвитку бізнесу в сфері ветеринарної медицини в Україні.

**Ключові слова:** підприємництво, системний аналіз, прогнозування ветеринарних послуг.

### **Svitlana US**

Candidate of Physics and Mathematics Science, Associate Professor, Professor at the Department of System Analysis and Control, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave, Dnipro, Ukraine, 49005, [us.s.a@ntu.one](mailto:us.s.a@ntu.one)

ORCID: 0000-0003-0311-9958

Scopus Author ID: 55603096000

### **Liubov TYMOSHENKO**

*Candidate of Economic Science, Associate Professor, Associate Professor at the Department of Applied Economics, Entrepreneurship and Public Administration, Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave, Dnipro, Ukraine, 49005, tymoshenko.l.v@nmu.one*

**ORCID:** 0000-0003-4869-8244

### **Anastasia YUDYNA**

*Student of Dnipro University of Technology, 19, Dmytra Yavornytskoho Ave, Dnipro, Ukraine, 49005, yudina.a.o@nmu.one*

**To cite this article:** Us, S., Tymoshenko, L., Yudyna, A. (2023). Systemnyi analiz rehioniv Ukrainy dlia zapochatkuvannia biznesu u sferi nadannia veterynarnykh posluh [System analysis of the regions of Ukraine for starting a business in the provision of veterinary services]. *Information Technology: Computer Science, Software Engineering and Cyber Security*, 4, 67–78, doi: <https://doi.org/10.32782/IT/2023-4-8>

## **SYSTEM ANALYSIS OF THE REGIONS OF UKRAINE FOR STARTING A BUSINESS IN THE PROVISION OF VETERINARY SERVICES**

*The treatment of animals and the provision of veterinary and sanitary services in Ukraine are increasingly becoming the prerogative of private veterinary medicine due to the formation of market conditions. Despite the fierce competition in the work of commercial institutions and entrepreneurial doctors, the market for veterinary services is constantly growing. However, this field remains less researched and widespread compared to other fields. Therefore, starting a business in the field of providing veterinary services, even in the presence of competition, is a promising direction of entrepreneurial activity, which determines the relevance of the conducted research.*

*The purpose of the work is to analyze trends in the development of veterinary services in different regions of Ukraine and to identify those that may be potentially attractive for starting a veterinary business.*

*For this purpose, an analysis of the number of veterinary medicine institutions in the regions of Ukraine was carried out and five regions where rapid development of the field of veterinary services is observed. These regions are considered the most promising for entrepreneurship. In addition, a forecast of the development of the veterinary business of three regions was made using various methods of short-term forecasting, in particular, according to the Holt-Winters, Brown and moving-average models. The most applicable approaches to forecasting for two years have been determined. The received evaluations of the forecast confirmed its adequacy.*

*The conducted analysis confirmed the potential of starting a business in the field of providing veterinary services and made it possible to determine the most promising regions for starting a veterinary business. Further scientific research can be aimed at modeling the development of the veterinary service, taking into account the specifics of the selected regions and areas of veterinary medicine. An in-depth study of the dynamics of this industry will allow us to identify and use new opportunities for business development in the field of veterinary medicine in Ukraine.*

**Key words:** entrepreneurship, system analysis, forecasting of veterinary services.

**Актуальність проблеми.** Економічна стабільність держави ґрунтується на забезпеченні максимальної зайнятості населення. Досвід розвинених країн свідчить, що ефективним інструментом формування економічної стабільності виступає розвиток сектору малого та середнього бізнесу. Заснування власного бізнесу та самозайнятість сприяють зменшенню рівня безробіття, насиченню вітчизняного ринку високоякісними товарами і послугами, зростанню добробуту громадян за рахунок підвищення рівня доходів населення. Все це, в свою чергу, призводить до зростання спроможності територіальних громад, розташованих в різних регіонах України.

Започаткування бізнесу у сфері надання ветеринарних послуг, навіть при наявності конкуренції на ринку, є перспективним напрямом підприємницької діяльності в Україні. За резуль-

татами опитування, проведеного Центром дослідження суспільства Національної суспільної телерадіокомпанії (1), 48,1% опитаних мали вдома kota, а 40,5% – собаку. Експерти дійшли спільного висновку, що загалом в Україні досить високий рівень піклування про своїх домашніх улюбленців, які періодично потребують кваліфікованої ветеринарної допомоги.

Ветеринарні послуги користуються постійним попитом, і в цьому сегменті бізнесу відсутня сезонність. Ця галузь визначається високою початковою лояльністю клієнтів, що забезпечує підприємцеві стійкі та вигідні взаємини зі споживачами.

Ветеринарна практика – це діяльність з надання послуг, пов'язаних з профілактикою, діагностикою та лікуванням хвороб тварин, консультуванням з питань ветеринарної медицини, яка провадиться закладами ветеринар-

ної медицини та спеціалістами ветеринарної медицини на підставі відповідної ліцензії (Закон України «Про ветеринарну медицину», 2021), яка видається фізичним та юридичним особам у порядку, визначеному Законом України «Про ліцензування видів господарської діяльності» (Закон України «Про ліцензування видів господарської діяльності», 2015)

Уряд в країні працює над поліпшенням умов для підприємницької діяльності та вирішенням проблем, які гальмують розвиток бізнесу. Поточна дерегуляція в умовах конфлікту створює оптимістичний прогноз для відновлення підприємницької активності при сприятливому розвитку воєнної ситуації.

Слід зазначити, що в сфері надання ветеринарних послуг чинники формування рівня конкуренції на ринку впливають на прибутковість підприємницької діяльності. Тому перед прийняттям рішення щодо започаткування бізнесу з надання послуг саме в цій сфері в певному населеному пункті необхідно провести аналіз ринку таких послуг з урахуванням регіональних особливостей.

**Аналіз останніх досліджень і публікацій.** В сучасних умовах ветеринарна діяльність ведеться як державною, так і приватними службами. Область діяльності державної ветеринарної служби обмежується здійсненням державного ветеринарного нагляду, лабораторними дослідженнями, ветеринарно-санітарною експертизою продуктів тваринного і рослинного походження. Основний обсяг ветеринарної роботи проводиться спеціалістами приватної ветеринарної медицини, яка об'єднує найманих ветеринарних лікарів та приватно практикуючих спеціалістів (Жуковський М., 2016).

Науковці на основі маркетингово-орієнтованого підходу до формування сфери ветеринарних послуг (Поперечний, 2006) зазначають, що ринок ветеринарних послуг формується під впливом значної кількості чинників: потреб споживачів у певному виді послуг та попиту на них, привабливості сфери ветеринарних послуг для нинішніх і потенційних суб'єктів господарювання, перешкод для входження у галузь, рівня монополізації ринку, впливу державних органів тощо.

Кручиненко О.В. в дослідженні (Кручиненко О.В., 2017) визначає ветеринарну послугу як особливий вид діяльності фахівців ветеринарної медицини, яка спрямована на забезпечення корисного ефекту у формі здорового та продуктивного розвитку тваринного світу і на цій основі – відповідних прибутків. Ринковий успіх у сфері ветеринарних послуг матимуть суб'єкти

господарювання, які зможуть забезпечити оперативність надання послуг, володітимуть позитивним іміджем професійного спілкування, будуть забезпечені необхідним обладнанням тощо.

Аранчій Я.С., Круковська О.В. (Аранчій Я.С., 2011) акцентують увагу на тому, що ветеринарні послуги не виступають аналогом інших видів обслуговування. Проведений авторами системно-структурний аналіз сфери послуг показує, що в умовах вільного вибору суб'єкти, що надають послуги, змушені вивчати поведінку споживачів, їх потреби та бажання. На основі цього послуги можуть бути наділені тими ознаками, яких найбільше потребують потенційні покупці. Набір цих ознак і визначає якість послуги на ринку, її конкурентоспроможність. Але споживачі ветеринарних послуг переважно не є фахівцями з ветеринарної медицини і не можуть належним чином оцінити якість послуги з професійної точки зору.

На думку Кручиненко О.В. (Кручиненко О.В., 2017), формуючи ринок ветеринарних послуг з метою належного задоволення потреб споживачів, необхідно оптимізувати систему ветеринарного обслуговування шляхом поєднання одноосібної ветеринарної практики в кожному населеному пункті з діяльністю підприємств ветеринарної медицини, які б спеціалізувалися на наданні навіть найскладніших ветеринарних послуг.

Для вирішення вище зазначених питань доцільно застосовувати відповідний математичний апарат, але слід звернути увагу на наступне. Автори роботи (Кременовська І.В., 2015) наголошують на тому, що одним лише неправильним вибором і неадекватним використанням математичних моделей у таких галузях як медицина, сільське господарство, екологія, хімія, біологія, тощо, може бути заподіяно шкоду, яка матиме катастрофічні наслідки: від створення загрози для життя і здоров'я людей – аж до повної руйнації цілих сфер діяльності.

Отже, виникає нагальна потреба проведення дослідження, яке буде ґрунтуватися на використанні доцільних математичних інструментів, для визначення забезпечення регіонів України ветеринарними послугами, що буде сприяти як інформуванню підприємницьких структур для започаткування бізнесу, так і в подальшому досягненню більш повного задоволення потреб населення у сфері ветеринарного обслуговування.

**Мета дослідження** здійснення аналізу сучасного стану та динаміки розвитку ветеринарних послуг в регіонах України і визначення

перспективних областей для започаткування ветеринарного бізнесу.

**Виклад основного матеріалу дослідження.** Вихідними даними для проведення дослідження слугували статистичні дані про кількість ветеринарних клінік в Україні (Державна служба статистики України). На рис. 1 показано динаміку кількості ветеринарних клінік в Україні за областями.

Легко побачити, що до п'ятірки найбільш розвинених областей за кількістю закладів ветеринарної медицини у 2014 році входили Київська, Харківська, Дніпропетровська, Вінницька області та місто Київ.

В період з 2014 по 2020 їх кількість у цих областях, крім Вінницької, стало збільшуватися. В середньому вона зросла на 55 закладів (47, 57, 61 у Дніпропетровській, Харківській та Київській областях відповідно). Проте кількість клінік у Вінницької області почала стрімко зменшуватись в період з 2015 по 2018 роки. Як видно з діаграми незмінним лідером є місто Київ. З 2014 по 2020 роки кількість ветеринарних клінік столиці зросла на 93 заклади. Отже станом на 2020 рік п'ятірку лідерів складала Дніпропетровська, Харківська, Київська, Одеська області та місто Київ. Порівняно із ними, всі інші регіони не мали потужного розвитку. Кількість ветери-

нарних закладів у них зросла в середньому на 6 у кожній області.

Проте абсолютна кількісна оцінка може створити помилкове бачення ситуації, оскільки чисельність населення у кожному із регіонів сильно відрізняється. Отже, для більш точного аналізу ситуації доречно провести нормалізацію даних за кількістю населення в ті самі роки (2014 – 2020). Аналіз нормалізованих даних показав, що в цілому тенденції збігаються, але розрив між областями із найбільшими значеннями та іншими вже не є таким значним. Найбільші значення відношення *кількість клінік/населення* відповідають м. Київ, Київській, Харківській, Кіровоградській, Одеській та Чернівецькій областям. Результати, отримані після нормалізації, для шести областей, які мають найбільшу кількість ветеринарних закладів, показано на рис. 2. При цьому найбільша концентрація закладів ветеринарної медицини знаходиться у Київській області. Столиця знаходиться на другому місці. Зберігаються тенденції змін у Вінницькій області, кількість досліджуваних закладів тут набуває свого піку в 2015 році, після чого відбувається різкий спад впродовж 2016 – 2020 років. Тенденцію стійкого зростання також показують Одеська та Дніпропетровська області.

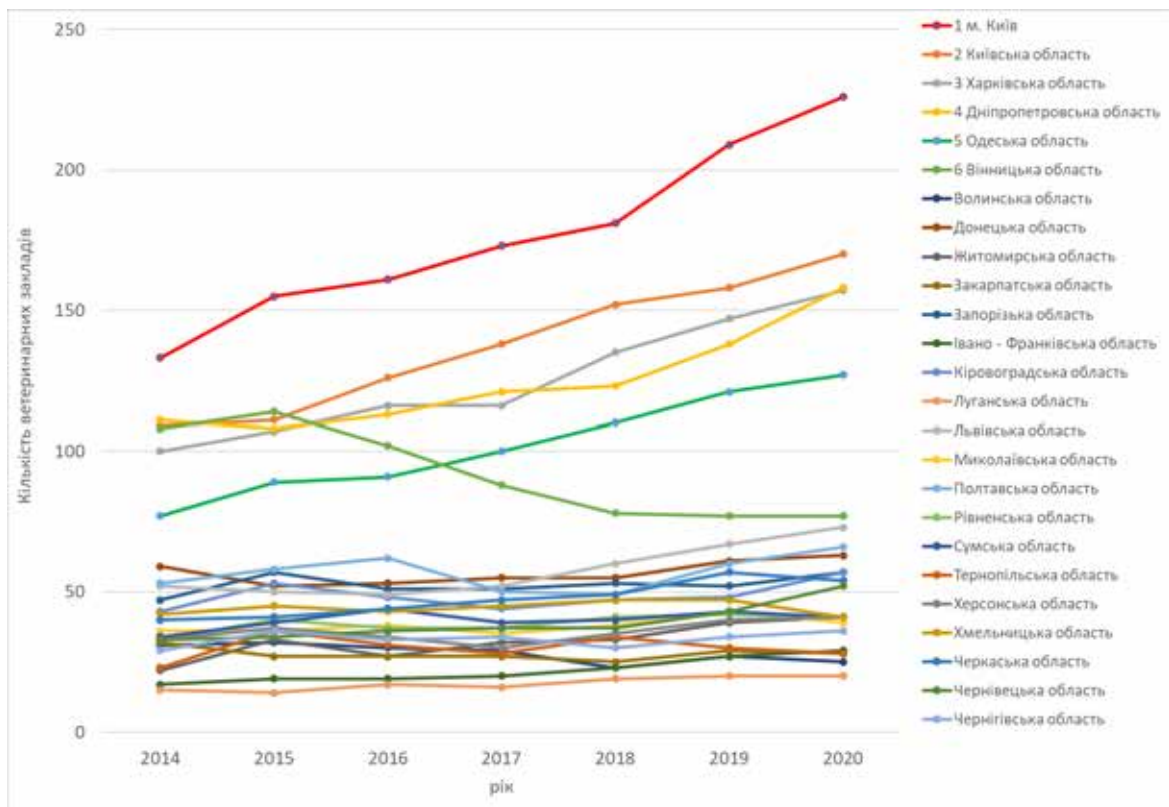


Рис. 1. Динаміка чисельності ветеринарних клінік в областях України у 2014 – 2020 роках





а)



б)



в)

Рис. 4. Реальні дані та результати прогнозування для Дніпропетровської області, які отримані: а) за моделлю Хольта – Вінтерса, б) за моделлю Брауна, в) методом середнього ковзного



У даній роботі для дослідження було використано моделі Хольта – Вінтерса, Брауна та метод ковзної середньої.

Модель Хольта – Вінтерса являє собою один із методів прогнозування часових рядів з використанням експоненційного згладжування. Для згладжування використовується зважене середнє, причому ваги обираються таким чином, щоб нова інформація мала більший вплив на прогноз. Дана модель включає три параметри прогнозу, які дозволяють врахувати згладжений експоненціальний ряд, тренд та сезонність. Аби зробити якісний прогноз за цією моделлю, потрібно мати дані не менш ніж за 3 – 5 повних років. Для оцінки точності прогнозів цієї моделі використовують середню абсолютну та середню квадратичну похибки. Якщо похибка буде менше 10 %, вважають що модель Хольта – Вінтерса має високу точність і може бути використана.

Модель Брауна дозволяє відобразити розвиток у вигляді лінійної тенденції, у вигляді випадкового процесу, який не має тенденції, а також у вигляді мінливої параболічної тенденції. Відповідно до цього розрізняють моделі Брауна нульового, першого та другого порядку.

Модель нульового порядку описує процеси, які не мають тенденцій розвитку. Вона має один параметр  $A_0$  (оцінка поточного рівня).

Модель першого порядку, включає два коефіцієнти:  $A_0$  – значення, близьке до останнього рівня, і яке являє собою закономірну складову цього рівня,  $A_1$  – визначає зростання, яке здійснилося в основному до кінця періоду спостережень,

та відображає швидкість збільшення на більш ранніх етапах.

Модель другого порядку, відображає розвиток у вигляді параболічних тенденції з мінливими «швидкістю» й «прискоренням». Вона має три параметри (крім названих вище ще параметр  $A_2$  – оцінка поточного зростання або «прискорення»).

Метод ковзної середньої є одним з широко відомих методів згладжування часових рядів. Згладжування за допомогою ковзної середньої засноване на тому, що в середніх величинах взаємно погашаються випадкові відхилення. Це відбувається внаслідок заміни первинних рівнів часового ряду середньою арифметичною величиною всередині обраного інтервалу часу. Отримане значення відноситься до середини обраного інтервалу часу (періоду). Потім період зсувається на одне спостереження і розрахунок середньої повторюється. При цьому періоди визначення середньої мають бути однаковими протягом всього періоду прогнозування. При згладжуванні часового ряду за допомогою ковзних середніх у розрахунках беруть участь всі рівні ряду. Чим ширше інтервал згладжування, тим більше плавним виходить тренд.

Всі розрахунки було виконано за допомогою засобів Microsoft Excel.

Розглянемо результати прогнозування для Дніпропетровської, Вінницької областей та м. Київ.

Спрогнозуємо кількість ветеринарних закладів для Дніпропетровської області із застосуванням отриманих моделей. Результати прогнозування показано на рис. 4, а – в і в табл. 1.

Таблиця 1

## Результати прогнозування кількості ветеринарних закладів у Дніпропетровській області

Рік	Реальні дані	Модель Хольта – Вінтерса		Модель Брауна		Метод ковзної середньої						
		Прогноз	Абсолютна похибка	Прогноз	Абсолютна похибка	За двома роками		За трьома роками		За чотирма роками		
						Прогноз	Абсолютна похибка	Прогноз	Абсолютна похибка	Прогноз	Абсолютна похибка	
2014	111	111		107,8	3,2							
2015	108	108		110,86	2,86							
2016	113	109,3186	3,68142	109,35	3,65	109,5	3,5					
2017	121	117,3227	3,677342	114,20	6,80	110,5	10,5	110,7	10,3			
2018	123	129,7193	6,719325	125,84	2,84	117	6	114	9	113,25	9,75	
2019	138	139,8086	1,808563	134,65	3,35	122	16	119	19	116,25	21,75	
2020	158	157,9982	0,001823	146,80	11,20	130,5	27,5	127,3	30,7	123,75	34,25	
2021		182,7787		170,16		148		139,7		135		
Середня абсолютна похибка			2,65		4,84		16,5		19,6		21,9	
Відносна похибка			0,0194		0,034		0,091		0,123		0,151	



а)



б)



в)

Рис. 5. Реальні дані та результати прогнозування для м. Київ, які отримані: а) за моделлю Хольта – Вінтерса, б) за моделлю Брауна, в) методом середнього ковзного

Як бачимо, для даних Дніпропетровської області найбільш точним виявився прогноз методом Хольта – Вінтерса, хороші показники також має прогнозування за моделлю Брауна, метод ковзного середнього був застосований для періоду згладжування 2, 3, та 4 роки, але жоден з них не дав прийняттого результату.

*Прогнозування кількості закладів ветеринарної медицини у м. Києві.*

Результати прогнозу, отриманого за визначеними моделями для м. Київ показано на рис. 5, а – в. Порівнюючи реальні та прогнозовані дані за 2015 – 2020 роки, можна побачити досить точну відповідність прогнозу реальним даним при застосуванні моделі Хольта – Вінтерса, що підтверджує також обчислений коефіцієнт прийнятності прогнозу, який дорівнює 4 % (див. табл. 2). Застосування моделі Брауна також дає прийнятну відносну помилку – біля 5 % (див. табл. 2). Метод ковзного середнього не є прийнятним для цих даних, оскільки дає велику помилку (див. рис. 5, в, табл. 2).

Розглянемо тепер застосування обраних моделей прогнозування для Вінницької області. Результати показано на рис. 6, а – в та в табл. 3. Як бачимо, модель Хольта – Вінтерса в цьому випадку не дала прийняттого результату. Коефіцієнт прийнятності даного прогнозу дорівнює 16 %, а прогноз вважають точним коли похибка становить менше 10 %. Відносна похибка моделі Брауна також є досить великою – біля 14 %. Зауважимо, що обидві ці моделі показу-

ють зміну тенденції, тобто відповідно до них кількість ветеринарних закладів у Вінницькій області буде зростати. Найкращим за результатами дослідження для даних Вінницької області виявився прогноз методом ковзного середнього із періодом згладжування два роки.

**Висновки і перспективи подальших досліджень.** Проведені дослідження показали, що розвиток ветеринарного бізнесу є перспективним. Причому найбільше його зростання спостерігається у великих містах. Відтак, перспективними виявилися м. Київ та Київська область, а також Харківська, Кіровоградська, Чернівецька та Дніпропетровська області. Зауважимо, що при аналізі потрібно враховувати суттєву залежність результату від обраного методу прогнозування і його прийнятності до наявного набору даних.

Подальші дослідження можуть бути зосереджені на моделюванні динаміки ветеринарних послуг з врахуванням наступного. В умовах сьогодення формування ринку ветеринарних послуг відбувається також під впливом факторів, спричинених воєнним станом в країні. Внаслідок міграційних процесів до «відносно спокійних міст» переїхати як суб'єкти підприємництва, що надавали ветеринарні послуги у своїх містах, так і переселенці із своїми домашніми тваринами. До цього слід додати, що на опікуванні волонтерів зросла чисельність домашніх тварин, які втратили своїх хазяїв та потребують ветеринарної допомоги.

Таблиця 2

## Результати прогнозування кількості ветеринарних закладів у м. Київ

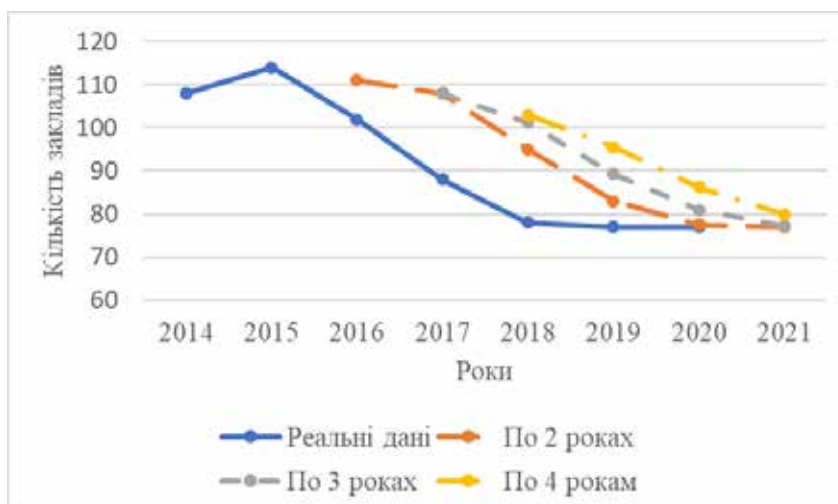
Рік	Реальні дані	Модель Хольта – Вінтерса		Модель Брауна		Метод ковзного середнього					
		Прогноз	Абсолютна похибка	Прогноз	Абсолютна похибка	За двома роками		За трьома роками		За чотирма роками	
						Прогноз	Абсолютна похибка	Прогноз	Абсолютна похибка	Прогноз	Абсолютна похибка
2014	133			137,8	4,8						
2015	155	155	0	144,5	10,5						
2016	161	177	16	158,6	2,4	144	17				
2017	173	183	10	182,4	9,4	158	15	149,7	23,3		
2018	181	195	14	196,7	15,7	167	14	163	18	155,5	25,5
2019	209	203	6	195,4	13,6	177	32	171,7	37,3	167,5	41,5
2020	226	231	5	207,6	18,4	195	31	187,7	38,3	181	45
2021		248		238,3		217,5		205,3		197,25	
Середня абсолютна похибка			8,5		10,7		21,8		29,25		37,3
Відносна похибка			0,041		0,05		0,122		0,145		0,179



а)



б)



в)

**Рис. 6. Реальні дані та результати прогнозування для Вінницької області, які отримані: а) за моделлю Хольта – Вінтерса, б) за моделлю Брауна, в) методом середнього ковзного**

Таблиця 3

## Результати прогнозування кількості ветеринарних закладів у Вінницькій області

Рік	Реальні дані	Модель Хольта – Вінтерса		Модель Брауна		Метод ковзного середнього					
		Прогноз	Абсолютна похибка	Прогноз	Абсолютна похибка	За двома роками		За трьома роками		За чотирма роками	
						Прогноз	Абсолютна похибка	Прогноз	Абсолютна похибка	Прогноз	Абсолютна похибка
2014	108			115,2	7,2						
2015	114	114	0	108,13	5,87						
2016	102	120	18	110,87	8,87	111	9				
2017	88	108	20	98,47	10,47	108	20	108	20		
2018	78	94	16	75,6	2,4	95	17	101,3	23,3	103	25
2019	77	84	7	55,13	21,87	83	6	89,3	12,3	95,5	18,5
2020	77	83	6	56,53	20,47	77,5	0,5	81	4	86,25	9,25
2021		83		78,4		77		77,3		80	
Середня абсолютна похибка			11,17		11,0		10,5		14,9		17,6
Відносна похибка			0,16		0,14		0,124		0,185		0,227

## ЛІТЕРАТУРА:

1. Коти проти собак: кого українці заводять частіше – опитування [Електронний ресурс]. URL: <https://susplne.media/179010-koti-proti-sobak-kogo-ukrainci-zavodat-castise-opituvanna/> (дата звернення: 22.12.2023). Назва з екрану.
2. Закон України «Про ветеринарну медицину»: офіц. текст прийнятий Верховною Радою України 04 лютого 2021 р. : із змінами та допов. станом на 22 грудня 2023р. [Електронний ресурс]. URL: <https://zakon.rada.gov.ua/laws/show/1206-20#Text> (дата звернення: 22.12.2023).
3. Закон України «Про ліцензування видів господарської діяльності»: офіц. текст прийнятий Верховною Радою України 02 березня 2015 р. : із змінами та допов. станом на 22 грудня 2023 р [Електронний ресурс]. URL: <https://zakon.rada.gov.ua/laws/show/222-19#Text> (дата звернення: 22.12.2023).
4. Жуковський М. О. Аналіз міжнародного досвіду фінансування протиепізоотичних заходів / М. О. Жуковський, В. В. Місніченко, В. В. Недосєков. // Наукові доповіді Національного університету біоресурсів і природокористування України. 2016. № 1. URL: [http://nbuv.gov.ua/UJRN/Nd\\_2016\\_1\\_24](http://nbuv.gov.ua/UJRN/Nd_2016_1_24)
5. Поперечний С. І. Маркетингово орієнтований підхід до формування сфери ветеринарних послуг / С. І. Поперечний, Л. В. Одинак // Формування ринкової економіки : зб. наук. пр. / ДВНЗ «Київ. нац. екон. ун-т ім. В. Гетьмана» ; редкол.: О. О. Беляєв (відп. ред.) [та ін.]. Київ : КНЕУ, 2006. Спец. вип.: Маркетинг: теорія і практика : в 2 т. Т. 2. С. 616 – 623.
6. Кручиненко О. В. Теоретичні аспекти формування ринку ветеринарних послуг у тваринництві / О. В. Кручиненко // Вісник Дніпропетровського державного аграрно-економічного університету. 2017. № 4. С. 98 –102. URL: [http://nbuv.gov.ua/UJRN/vddau\\_2017\\_4\\_19](http://nbuv.gov.ua/UJRN/vddau_2017_4_19)
7. Аранчій Я.С. Системно-структурний аналіз сутності ветеринарних послуг і особливостей їх надання // Я.С. Аранчій, О.В. Круковська // Наукові праці Полтавської державної аграрної академії. 2011. Вип. 2. С. 32 – 38.
8. Випадки використання методів математичного моделювання для фальсифікації наукового підґрунтя регулювання чисельності бродячих (безпритульних) тварин / І.В. Кременовська, О.А. Святогор // Математичне моделювання в економіці. 2015. № 1(2). С. 61 – 72.
9. Державна служба статистики України // Офіційний сайт. URL: [ukrstat.gov.ua](http://ukrstat.gov.ua) (дата звернення: 22.12.2023) Назва з екрану
10. Круковська О.В. Формування ринку ветеринарних послуг: стан та перспективи розвитку [Електронний ресурс]. URL: [https://www.rusnauka.com/30\\_NNM\\_2010/Economics/71756.doc.htm](https://www.rusnauka.com/30_NNM_2010/Economics/71756.doc.htm) (дата звернення: 22.12.2023)

#### REFERENCES:

1. Koty proty sobak: koho ukrainci zavodiak chastishe – opytuvannia [Cats versus dogs: which Ukrainians get more often – survey]. Retrieved from <https://suspilne.media/179010-koti-proti-sobak-kogo-ukrainci-zavodat-castise-opituvanna/> (accessed date: 22.12.2023). Name from the screen [in Ukrainian].
2. Zakon Ukrainy «Pro veterynarnu medytsynu»: ofits. tekst pryiniaty Verkh. Radoiu Ukrainy 04 liutoho 2021 r. : iz zminamy ta dopov. stanom na 22 hrudnia 2023r. [Law of Ukraine "On Veterinary Medicine": officer. text adopted Top. By the Council of Ukraine on February 4, 2021: with changes and supplements. as of December 22, 2023] Retrieved from <https://zakon.rada.gov.ua/laws/show/1206-20#Text> (accessed date: 22.12.2023).
3. Zakon Ukrainy «Pro litsenzuvannia vydiv hospodarskoi diialnosti»: ofits. tekst pryiniaty Verkh. Radoiu Ukrainy 02 bereznia 2015 r.: iz zminamy ta dopov. stanom na 22 hrudnia 2023r. [Law of Ukraine "On Licensing of Types of Economic Activity": Officer. text adopted Top. By the Council of Ukraine on March 2, 2015: with changes and additions. as of December 22, 2023]. Retrieved from <https://zakon.rada.gov.ua/laws/show/222-19#Text> (accessed date: 22.12.2023).
4. Zhukovskiy M., Misnichenko V. & Nedosiekov V. (2016). Analiz mizhnarodnoho dosvidu finansuvannia protyepizootychnykh zakhodiv [Analysis of the international experience of financing anti-epidemic measures] *Scientific reports of the National University of Bioresources and Nature Management of Ukraine*. № 1. Retrieved from [http://nbuv.gov.ua/UJRN/Nd\\_2016\\_1\\_24](http://nbuv.gov.ua/UJRN/Nd_2016_1_24) (accessed date: 22.12.2023) [in Ukrainian].
5. Poperechnyi S., Odynak L. (2006). Marketynhovo oriietovanyi pidkhid do formuvannia sfery veterynarnykh posluh [A marketing-oriented approach to the formation of the field of veterinary services] *The formation of a market economy: a collection of scientific works*. Spec. issue: Marketing: theory and practice: vol. 2. – pp. 616 – 623. [in Ukrainian].
6. Kruchynenko O. (2017). Teoretychni aspekty formuvannia rynku veterynarnykh posluh u tvarynnytstvi [Theoretical aspects of the formation of the veterinary services market in animal husbandry] *Bulletin of the Dnipropetrovsk State Agrarian and Economic University*. № 4. P. 98 – 102. Retrieved from [http://nbuv.gov.ua/UJRN/vddau\\_2017\\_4\\_19](http://nbuv.gov.ua/UJRN/vddau_2017_4_19) [in Ukrainian].
7. Aranchiy, Ya., Krukovska O. (2011). Systemno-strukturnyi analiz sutnosti veterynarnykh posluh i osoblyvosti yikh nadannia [System-structural analysis of the essence of veterinary services and the features of their provision] *Scientific works of the Poltava State Agrarian Academy*. № 2. P. 32 – 38. [in Ukrainian].
8. Kremenovska I., Sviatohor O. (2015). Vypadky vykorystannia metodiv matematychnoho modeliuвання dlia falsyfikatsii naukovoho pidhruntia rehuliuвання chyselnosti brodiachykh (bezprytulnykh) tvaryn [Cases of using mathematical modeling methods to falsify the scientific basis for regulating the number of stray (homeless) animals]. *Mathematical modeling in economics*. № 1(2). P. 61 – 72. [in Ukrainian].
9. Derzhavna sluzhba statystyky Ukrainy [State Statistics Service of Ukraine] // Official site. Retrieved from [ukrstat.gov.ua](http://ukrstat.gov.ua) (accessed date: 22.12.2023) Name from the screen [in Ukrainian].
10. Krukovska O. Formuvannia rynku veterynarnykh posluh: stan ta perspektyvy rozvytku [Formation of the veterinary services market: state and development prospects]. Retrieved from [https://www.rusnauka.com/30\\_NNM\\_2010/Economics/71756.doc.htm](https://www.rusnauka.com/30_NNM_2010/Economics/71756.doc.htm) (accessed date: 22.12.2023) [in Ukrainian].

## ЗМІСТ

**Ілля ЗІБОРОВ, Тімур ЖЕЛДАК**

ЕВОЛЮЦІЙНИЙ МЕТОД ПОШУКОВОЇ ОПТИМІЗАЦІЇ НА ОСНОВІ РОЮ ЧАСТОК  
ТА МОДЕЛЮВАННЯ ШТУЧНИХ ІМУННИХ СИСТЕМ .....3

**Dmytro KORNIENKO, Nataliia GOLIAN**

FORECASTING METHODS FOR STUDYING AND DETECTING NATURAL PHENOMENA.....13

**Ларуса КОРЯШКІНА, Данило ЛУБЕНЕЦЬ**

МАТЕМАТИЧНІ МОДЕЛІ ТА МЕТОДИ МУЛЬТИПЛЕКСНОГО РОЗБИТТЯ  
І БАГАТОКРАТНОГО ПОКРИТТЯ МНОЖИН ДЛЯ ЗАДАЧ РОЗМІЩЕННЯ-РОЗПОДІЛУ.....20

**Ivan LAKTIONOV, Oleksandr ZHABKO, Grygorii DIACHENKO, Mykola PROKOPENKO**

SUBSTANTIATION OF REQUIREMENTS FOR THE STRUCTURAL  
AND ALGORITHMIC ORGANIZATION OF AN IOT MONITORING SYSTEM FOR SOIL  
AND CLIMATE PARAMETERS IN AGRICULTURAL CROP ENTERPRISES.....32

**Oleksandr LYTVYNOV, Dmytro HRUZIN**

METHODS FOR OPTIMIZING THE LOADING AND UPDATING OF WEB PAGES  
USING CLOUD TECHNOLOGIES.....40

**Віра ЛЮБЧЕНКО, Данило ЧУМАЧЕНКО**

МОДЕЛЬ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ БІОНІЧНИХ ПРОТЕЗІВ.....51

**Тетяна СЕЛІВЬОРСТОВА, Никита КРАСНОШАПКА**

ОСОБЛИВОСТІ ПРОЄКТУВАННЯ МАСШТАБОВАНОЇ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ  
ДЛЯ ВЕБСЕРВІСІВ.....58

**Світлана УС, Любов ТИМОШЕНКО, Анастасія ЮДИНА**

СИСТЕМНИЙ АНАЛІЗ РЕГІОНІВ УКРАЇНИ ДЛЯ ЗАПОЧАТКУВАННЯ БІЗНЕСУ  
У СФЕРІ НАДАННЯ ВЕТЕРИНАРНИХ ПОСЛУГ.....67

## CONTENTS

<b>Illia ZIBOROV, Timur ZHELDAK</b> THE EVOLUTIONARY METHOD BASED ON PARTICLE SWARM OPTIMIZATION AND ARTIFICIAL IMMUNE SYSTEMS MODELLING.....	3
<b>Dmytro KORNIENKO, Nataliia GOLIAN</b> FORECASTING METHODS FOR STUDYING AND DETECTING NATURAL PHENOMENA.....	13
<b>Larysa KORIASHKINA, Danylo LUBENETS</b> MATHEMATICAL MODELS OF MULTIPLEX PARTITIONING AND MULTIPLE COVERAGE OF SETS FOR THE LOCATION-ALLOCATION PROBLEMS.....	20
<b>Ivan LAKTIONOV, Oleksandr ZHABKO, Grygorii DIACHENKO, Mykola PROKOPENKO</b> SUBSTANTIATION OF REQUIREMENTS FOR THE STRUCTURAL AND ALGORITHMIC ORGANIZATION OF AN IOT MONITORING SYSTEM FOR SOIL AND CLIMATE PARAMETERS IN AGRICULTURAL CROP ENTERPRISES.....	32
<b>Oleksandr LYTVYNOV, Dmytro HRUZIN</b> METHODS FOR OPTIMIZING THE LOADING AND UPDATING OF WEB PAGES USING CLOUD TECHNOLOGIES.....	40
<b>Vira LIUBCHENKO, Danylo CHUMACHENKO</b> QUALITY MODEL FOR SOFTWARE FOR BIONIC PROSTHESES.....	51
<b>Tetiana SELIVIORSTOVA, Nykyta KRASNOSHAPKA</b> ASPECTS OF DESIGNING SCALABLE MICROSERVICES ARCHITECTURE FOR WEB SERVICES.....	58
<b>Svitlana US, Liubov TYMOSHENKO, Anastasia YUDYNA</b> SYSTEM ANALYSIS OF THE REGIONS OF UKRAINE FOR STARTING A BUSINESS IN THE PROVISION OF VETERINARY SERVICES.....	67



## НОТАТКИ

# **INFORMATION TECHNOLOGY: COMPUTER SCIENCE, SOFTWARE ENGINEERING AND CYBER SECURITY**

Випуск 4

Коректура • Ірина Миколаївна Чудеснова

Комп'ютерна верстка • Андрій Олександрович Філатов

Підписано до друку: 28.12.2023. Формат 60x84/8. Гарнітура Arial.  
Папір офсет. Цифровий друк. Ум. друк. арк. 9,53. Замов. № 0224/173. Наклад 300 прим.

Видавництво і друкарня – Видавничий дім «Гельветика»  
65101, Україна, м. Одеса, вул. Інглєзі, 6/1  
Телефон +38 (095) 934 48 28, +38 (097) 723 06 08  
E-mail: mailbox@helvetica.ua  
Свідоцтво суб'єкта видавничої справи  
ДК № 7623 від 22.06.2022 р.