

## УДК 004.7:[519.876.5:656.212.5]

В. М. ПАХОМОВА<sup>1\*</sup>, А. О. ОПРЯТНИЙ<sup>2\*</sup>

<sup>1\*</sup>Каф. «Електронні обчислювальні машини», Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел. +38 (056) 373 15 89, ел. пошта viknikpakh@gmail.com, ORCID 0000-0002-0022-099X

<sup>2\*</sup>Каф. «Електронні обчислювальні машини», Дніпровський національний університет залізничного транспорту імені академіка В. Лазаряна, вул. Лазаряна, 2, Дніпро, Україна, 49010, тел. +38 (056) 373 15 89, ел. пошта olkdi@outlook.com, ORCID 0000-0001-7145-9677

## Програмна модель для визначення оптимальних маршрутів у комп'ютерній мережі за двоколоніальним мурашиним алгоритмом

**Мета.** На сучасному етапі в комп'ютерних мережах інформаційно-телекомунікаційної системи (ІТС) залізничного транспорту використовують протокол OSPF, який не дозволяє враховувати декілька метрик під час визначення оптимального маршруту. Тому ми ставимо за мету дослідити можливість організації маршрутизації в комп'ютерних мережах ІТС залізничного транспорту за допомогою двоколоніального мурашиного алгоритму. **Методика.** За програмною моделлю «Two-ACO», створеною мовою Python за двоколоніальним мурашиним алгоритмом, визначено оптимальний маршрут у комп'ютерній мережі. Вхідні дані моделі «Two-ACO»: параметри комп'ютерної мережі (матриця суміжності мережі, кількість маршрутизаторів); параметри мурашиного алгоритму (кількість ітерацій; кількість мурах у колонії; кількість елітних мурах; початковий рівень феромону; коефіцієнт випаровування; параметр для регулювання величини відкладеного феромону). **Результати.** Визначення за програмною моделлю «Two-ACO» оптимального маршруту подано у вигляді графів із зображенням на них оптимальних шляхів: за критерієм сумарної затримки на маршрутизаторах (для першої колонії мурах) та кількістю хопів (для другої колонії мурах). **Наукова новизна.** За створеною програмною моделлю «Two-ACO» для комп'ютерної мережі із 7 маршрутизаторів та 17 каналів проведено дослідження часу визначення оптимального шляху за кількістю звичайних та елітних мурах, значеннями коефіцієнта випаровування та відкладеного феромону. Установлено, що достатньо використовувати кількість мурах, що дорівнює кількості маршрутизаторів, мати в колонії 2 елітних мурах, при цьому вистачає 1 000 ітерацій, коефіцієнта випаровування від 0,2 до 0,7, а відкладеного феромону мурахами близько до одиниці. **Практична значимість.** Створена програмна модель «Two-ACO» з використанням двох колоній мурах за загальною затримкою на маршрутизаторах та кількістю хопів, що складають маршрут, дозволяє паралельно визначити оптимальні маршрути в комп'ютерній мережі залізничного транспорту. З'ясовано, що для комп'ютерної мережі з 15 маршрутизаторів та 17 каналів достатньо мати 30 агентів (двох мурах на вершину), значення феромону, відкладеного агентами, близьке до одиниці, при цьому коефіцієнт випаровування складає 0,4.

**Ключові слова:** залізничний транспорт; комп'ютерна мережа; маршрутизатор; затримка; двоколоніальний мурашиний алгоритм; феромон; відкладання; випаровування; час пошуку

### Вступ

Основним питанням під час функціонування комп'ютерних мереж залізничного транспорту є організація маршрутизації, що досягається на сучасному етапі використання відомого протоколу OSPF за принципом пошуку найкоротшого шляху. Але такий протокол маршрутизації не може працювати в умовах різкої зміни інтенсивності потоків трафіка, а також зміни конфігурації мережі та врахування декількох метрик визначення оптимального шляху. У зв'язку з цим виникає необхідність в інших

підходах до маршрутизації в комп'ютерних мережах залізничного транспорту, зокрема з використанням мультиагентних методів інтелектуальної оптимізації: мурашиний алгоритм, бджолиний алгоритм, метод часток, метод переміщення бактерій та інші [1–11, 14, 15].

Одним із найбільш розповсюджених є мурашиний алгоритм, першу версію якого запропонував Марко Доріго [9].

У подальшому мурашиний алгоритм для отримання розв'язку різних задач використовували такі вчені: Б. Бульхаймер, К. Гамбарделла, С. Глушко, А. Павленко, Р. Хартл, Х. Хоос,

## ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

С. Штовба, К. Штраусс, Т. Штютцле та інші. Так, наприклад, у роботі [6] С. Д. Штовба подав результати розв'язку задачі комівояжера для різних модифікацій мурашиного алгоритму. У роботі [2] А. І. Павленко та Ю. П. Титов дослідили стратегії мурашиного алгоритму. С. І. Глушко й Т. В. Какатунова в роботі [1] використовували нечітку модифікацію алгоритму мурашиних колоній.

Автори цієї статті вже надавали розв'язок задачі маршрутизації в інформаційно-телекомунікаційній системі (ІТС) залізничного транспорту з використанням мурашиного методу під час передачі керівних повідомлень, що потребує побудови мінімального кістякового дерева. Однак найчастіше необхідно визначити оптимальний маршрут між заданими вузлами комп'ютерної мережі ІТС, така задача може бути розв'язана як на основі класичного алгоритму, зокрема алгоритму Дейкстри, що лежить в основі протоколу OSPF, так і з використанням нейромережної технології, наприклад, нейронної моделі [13] та нейронечіткої мережі [12]. Але під час використання алгоритму Дейкстри як критерій можна розглянути лише одну метрику, а використання нейромережної технології потребує проведення додаткових досліджень визначення оптимальних параметрів нейронної (нейронечіткої) мережі. Застосування мурашиного алгоритму дозволить визначити оптимальний маршрут у комп'ютерній мережі за різними метриками паралельно, використовуючи одночасно різні колонії мурах.

Ідея мурашиного алгоритму – моделювання поведінки мурах, що пов'язано з їх здатністю швидко знаходити найкоротший шлях від мурашника до джерела їжі та адаптуватися до змінних умов. Під час руху мураха мітить шлях феромоном, і цю інформацію використовують інші мурахи для вибору шляху. Поведінка мурах може мати різний математичний опис, тому існують такі модифікації: Elitist Ant System, Ant–Q, Ant Colony System, Max–min Ant System, ASrank. Відмінними особливостями наявних модифікацій алгоритму є введення елітних му-

рах, механізм виділення та випаровування. На сучасному етапі програмна реалізація мурашиного алгоритму здійснена різними науковцями в Delphi, C++, Java, Maple, MatLAB.

### Мета

У цій роботі ми передбачаємо визначити оптимальний маршрут у комп'ютерній мережі залізничного транспорту за двоколоніальним мурашиним алгоритмом, для програмної реалізації якого обрано мову Python, а також з'ясувати параметри мурашиного алгоритму на основі створеної моделі.

### Методика

*Постановка задачі.* Комп'ютерну мережу, що лежить в основі ІТС залізничного транспорту, можна подати як зважений граф  $G(V, W)$ , де  $V$  – множина вершин графа, кількість яких дорівнює  $B$ , причому кожна вершина моделює собою вузол (маршрутизатор) мережі;  $W$  – множина ребер графа, кожне ребро моделює зв'язок між вузлами; кількість ребер графа дорівнює  $M$ . Розглянемо фрагмент такої мережі, зображений на рис. 1 ( $B = 15$ ,  $M = 17$ ). Кожному ребру графа присвоєна певна вага  $t_{ij}$  (час передачі пакета по лінії зв'язку від  $i$  до  $j$ -го маршрутизатора мережі з урахуванням середньої затримки пакета на  $i$ -му маршрутизаторі, мкс). Необхідно визначити оптимальний маршрут  $L_1$  між заданими маршрутизаторами комп'ютерної мережі за критерієм:

$$\sum_{(i,j) \in L_1} t_{ij} \rightarrow \min,$$

а оптимальний маршрут  $L_2$  за критерієм:

$$K \rightarrow \min,$$

де  $K$  – кількість хопів, що складають маршрут.

Пропонована методика визначає маршрути паралельно (одночасно) з використанням різних колоній мурах.

## ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

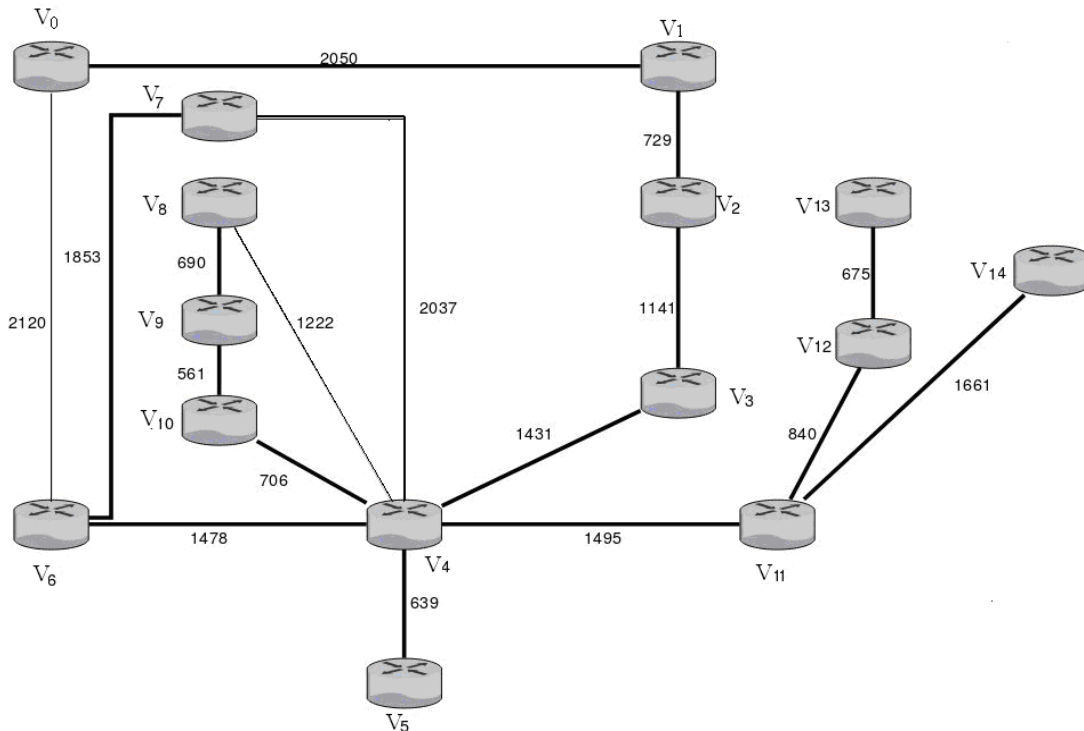


Рис. 1. Граф з'єднань маршрутизаторів комп'ютерної мережі

Fig. 1. Computer network router connection graph

*Біологічний аналог.* Мурахи використовують два способи передачі інформації: прямий – обмін їжею, візуальні й хімічні контакти, і непрямий – стигмержі [9]. *Стигмержі (stigmergy)* – це відокремлений у часі тип взаємодії, коли один суб'єкт змінює деяку частину навколишнього середовища, а інші використовують інформацію про його стан пізніше, коли перебувають у його межах. Біологічно стигмержі здійснюються через феромон – спеціальний секрет, який мурахи відкладають як слід під час переміщення [6]. *Феромон (pheromone)* – досить стійка речовина, його мурахи можуть сприймати кілька діб. Чим вища концентрація феромону на шляху, тим більше мурах буде по ньому рухатися. Із часом феромон випаровується, що дозволяє мурахам адаптувати свою поведінку під зміну зовнішнього середовища. Розподіл феромону по простору пересування мурах є свого роду динамічно-змінною глобальною пам'яттю мурашника. Будь-яка мураха у фіксований момент часу може сприймати й змінювати лише одну локальну комірку цієї глобальної пам'яті [6]. Самоорганізована система базується як на позитивному (інші мурахи

укріплюють феромонну тропу) так і на негативному (випаровування феромонної тропи) зворотному зв'язку, рис. 2.

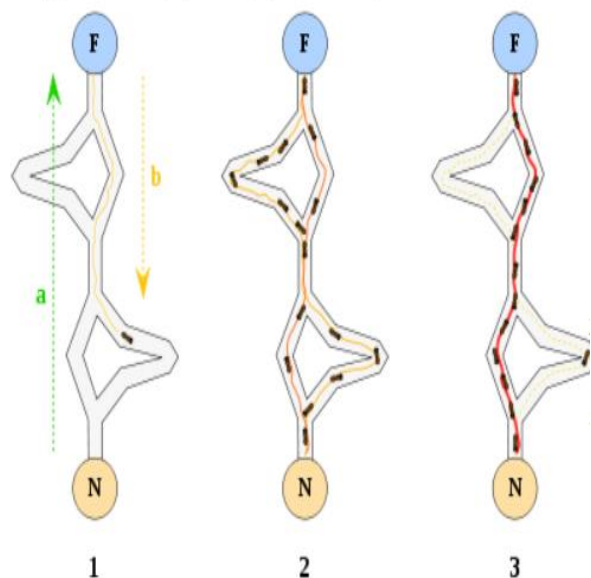


Рис. 2. Прокладання маршруту на основі феромону

Fig. 2. Routing based on a pheromone

## ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

*Математичний апарат.* Робота починається з розміщення мурах у вершинах графа, потім відбувається рух мурах – напрям визначають імовірнісним методом на підставі відомої формули:

$$P_i = \frac{l_i^q \cdot f_i^p}{\sum_{k=0}^N l_k^q \cdot f_k^p},$$

де  $P_i$  – імовірність переходу шляхом  $i$ ;  $l_i$  – довжина  $i$ -го переходу;  $f_i$  – кількість феромонів на  $i$ -му переході,  $q$  – величина, яка визначає «жадібність» алгоритму;  $p$  – величина, яка визначає «стадність» алгоритму, причому  $q + p = 1$ .

Кількість феромону, який був залишений на кожному ребрі шляху для мурахи  $k$ , розраховують за допомогою формули:

$$\Delta\tau_{ij}^k(t) = \frac{q}{L^k(t)},$$

де  $L^k(t)$  – довжина шляху мурахи  $k$  на ітерації  $t$ .

Для збільшення кількості феромону вздовж кожного ребра пройденого мурахою шляхом, отриманий результат обчислюється за формулою:

$$\tau_{ij}(t+1) = \Delta\tau_{ij}(t) + \tau_{ij}(t).$$

Випаровування феромону визначають за формулою:

$$\tau_{ij}(t) = \tau_{ij}(t) \cdot (1 - \rho),$$

де  $\rho$  – коефіцієнт випаровування, що може набувати значення від 0 до 1.

Програмна реалізація мурашиного алгоритму виконана мовою Python з використанням фреймворку для віконних інтерфейсів Qt, а також бібліотек NetworkX та Matplotlib для відображення графів на екран та визначення оптимального маршруту в мережі [3]. У програмі створені класи: NetworkX, що дозволяє представляти мережу як граф, та ACO (Ant Colony Optimization) для роботи з алгоритмом мурашиної оптимізації.

Вхідні параметри моделі:  $n$  – кількість маршрутизаторів;  $graph$  – матриця суміжності ме-

режі;  $iteration$  – кількість ітерацій;  $ants$  – кількість мурах у колонії;  $elite\_ants$  – кількість елітних мурах;  $phero\_lvl$  – початковий рівень феромону;  $ro$  – коефіцієнт випаровування;  $q$  – параметр для регулювання величини відкладення феромону.

Результати роботи моделі подають у вигляді графів (Figure1, Figure2) із зображенням на них оптимальних шляхів: за критерієм сумарної затримки на маршрутизаторах (для першої колонії мурах) та кількістю хопів (для другої колонії мурах).

Обмеження, які накладають на програмну модель: коефіцієнт випаровування набуває значення між 0 та 1 (включно); кількість елітних та звичайних мурах не обмежена. Визначено, що за  $n > 20$  (кількість маршрутизаторів у мережі) та  $iterations > 10\,000$  (кількість ітерацій) програмна модель починає працювати помітно повільніше. Загальна структура програмної моделі «Two-ACO» подана на рис. 3 [3].

У програмі створений клас Network, який дозволяє представляти комп'ютерну мережу як граф. Клас має такі властивості:  $routers\_num$ , що показує кількість маршрутизаторів у графі;  $graph$  – матриця суміжності для графа мережі;  $cost$  – функція, яка приймає два номери маршрутизаторів як аргументів та повертає затримку.

Клас ACO (Ant Colony Optimization) – клас для роботи з алгоритмом мурашиної оптимізації, що має такі властивості:  $iterations$  – кількість ітерацій алгоритму;  $ants$  – кількість звичайних мурах у колонії;  $elite\_ants$  – кількість елітних мурах у колонії;  $phero\_lvl$  – початковий рівень феромону;  $a$  – величина, що визначає «жадібність» алгоритму;  $b$  – величина, що визначає «стадність» алгоритму;  $ro$  – величина, що визначає «швидкість» випаровування феромону;  $q$  – параметр для регулювання величини відкладення феромону, а також такі функції:  $init\_phero$  – устанавлює початкову кількість феромону в мережі;  $calculate\_probs$  – розраховує можливості передачі даних між сусідніми маршрутизаторами;  $recalculate\_phero$  – функція для коригування рівня феромону, залишеного звичайними мурахами;  $recalculate\_elite\_phero$  – функція для коригування рівня феромону, залишеного елітними мурахами;  $path$  – функція, що визначає найкращий шлях між двома маршрутизаторами.

ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

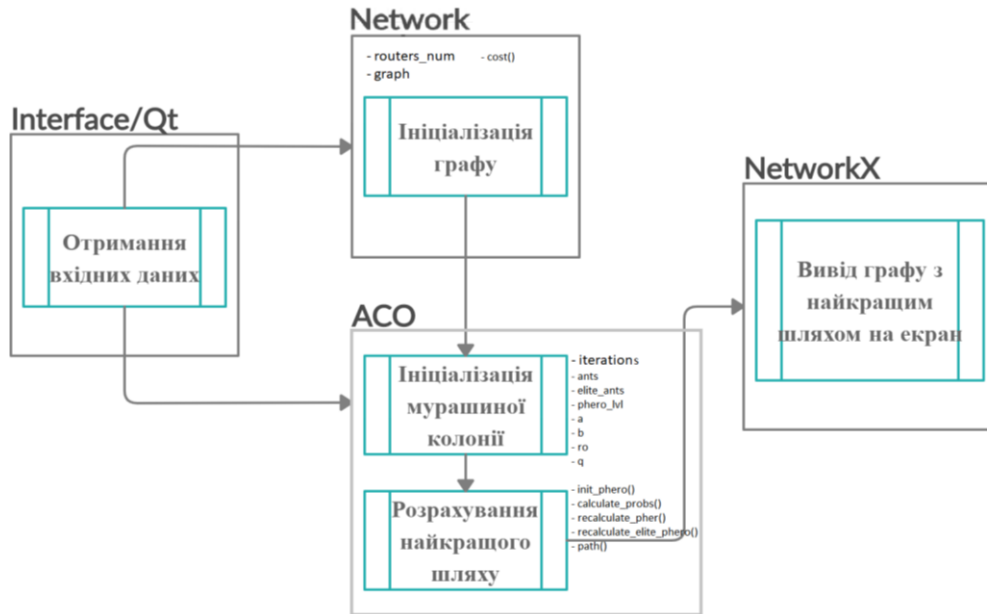


Рис. 3. Загальна структура створеної програмної моделі «Two-ACO»

Fig. 3. General structure of the created Two-ACO software model

*Перший можливий випадок.* За значень затримок, які вказані на рис. 4, програмна модель «Two-ACO» між вершинами  $V_0$  та  $V_4$  надала оптимальні маршрути (рис. 2):  $V_0 \rightarrow V_6 \rightarrow V_4 =$

$2\ 120 + 1\ 478 = 3\ 598$  мкс (Figure1 за першим критерієм);  $V_0 \rightarrow V_6 \rightarrow V_4 = 2$  (Figure2 за другим критерієм), при цьому сумарне значення затримок збігається та складає 3 598 мкс.

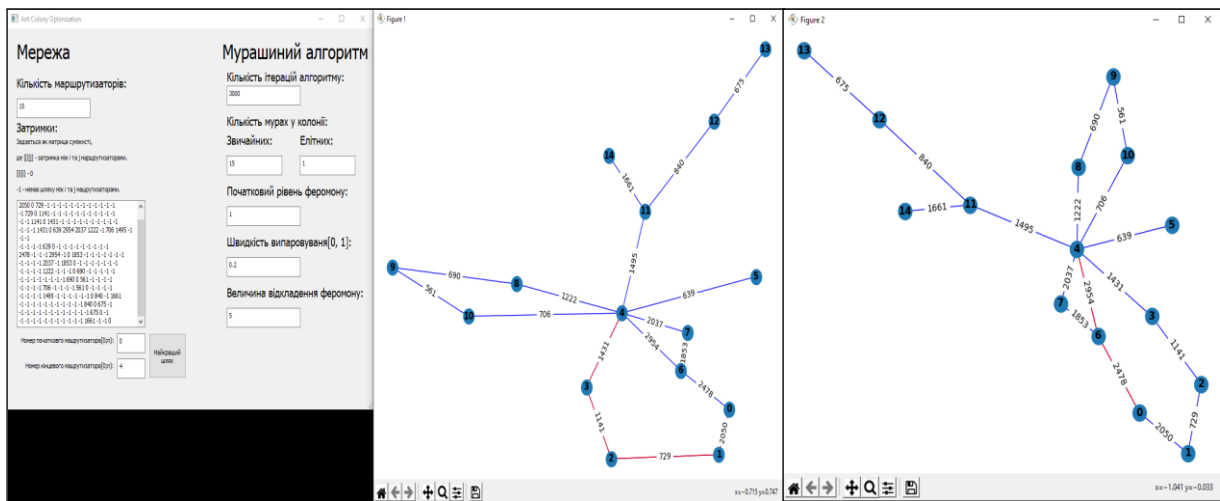


Рис. 4. Визначення за програмною моделлю оптимального маршруту за різними критеріями (за заданих значень затримок на маршрутизаторах)

Fig. 4. Determining the optimal route according to the program model based on various criteria (for set delay values on routers).

*Другий можливий випадок.* Проведемо запуск програмної моделі зі зміною значень затримок ( $V_0 \rightarrow V_6 = 2\ 478$  мкс;  $V_6 \rightarrow V_4 = 2\ 954$

мкс), рис. 5. У цьому випадку програмна модель «Two-ACO» між вершинами  $V_0$  та  $V_4$  надає вже інший оптимальний маршрут (див.

## ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

рис. 5):  $V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 = 2\,050 + 729 + 1\,141 + 1\,431 = 5\,351$  мкс (Figure 1 за першим критерієм); але  $V_0 \rightarrow V_6 \rightarrow V_4 = 2$  (Figure 2 за другим критерієм), при цьому сумарне значення затримок дорівнює  $2\,478 + 2\,954 = 5\,432$  мкс, що перевищує попереднє. Інакше

кажучи, використання шляху  $V_0 \rightarrow V_6 \rightarrow V_4$  як оптимального не є доцільним, оскільки існує інший шлях  $V_0 \rightarrow V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4$  з меншим значенням сумарної затримки на маршрутизаторах.

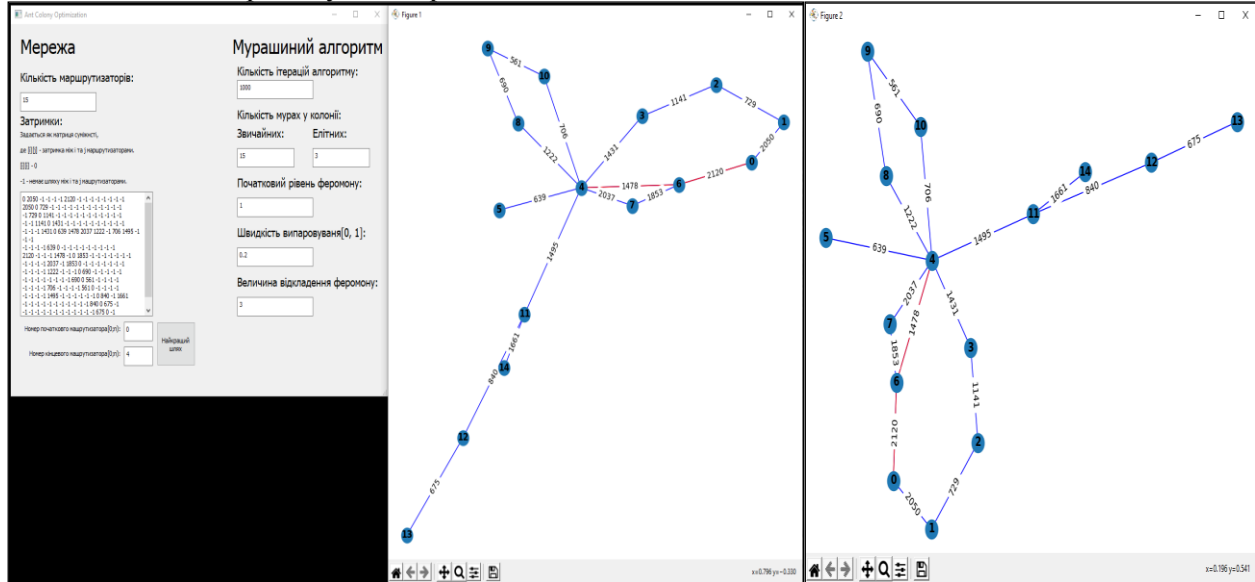


Рис. 5. Визначення за програмною моделлю оптимального маршруту за різними критеріями (у разі зміни значень затримок на маршрутизаторах)

Fig. 5. Determining the optimal route according to the program model based on various criteria (in case of change of delay values on the routers)

## Результати

За програмною моделлю «Two-ACO» проведено дослідження кількості ітерацій, необхідних для пошуку оптимального шляху, за параметрами алгоритму. Чим більше агентів, тим менша кількість ітерацій для знаходження оптимального шляху (рис. 6), для досліджуваної мережі двох мурах на вершину достатньо.

У разі 30 агентів оптимальне значення феромону, який вони відкладають, близьке до одиниці (рис. 7).

Проведемо дослідження кількості ітерацій за коефіцієнтом випаровування феромону в разі 30 агентів та визначеної кількості феромону, відкладеного на ребрах графа. Чим вище його значення, тим менше ітерацій пошуку оптимального шляху в комп'ютерній мережі із 15 маршрутизаторів та 17 каналів (рис. 8).

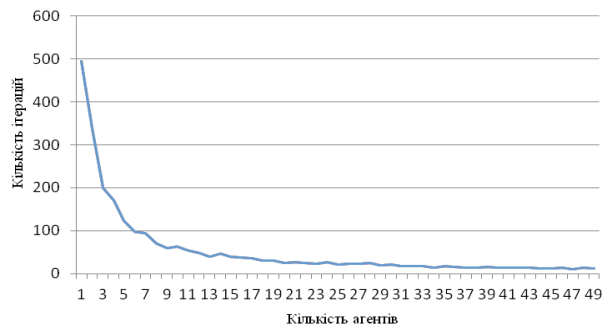


Рис. 6. Кількість ітерацій пошуку оптимального шляху за кількістю агентів

Fig. 6. Number of iterations to find the optimal path by the number of agents

## ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

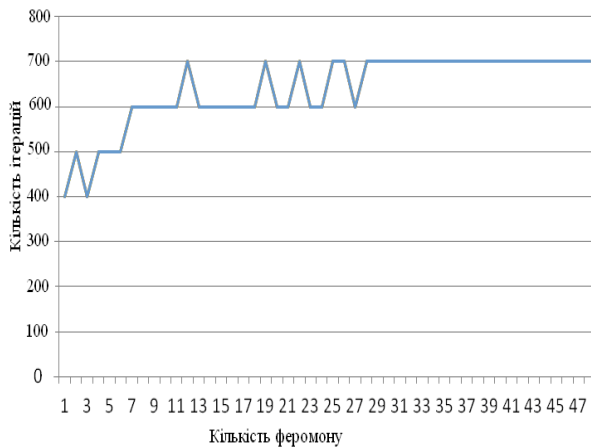


Рис. 7. Кількість ітерацій пошуку оптимального шляху за кількістю феромону

Fig. 7. Number of iterations to find the optimal path by the number of pheromones

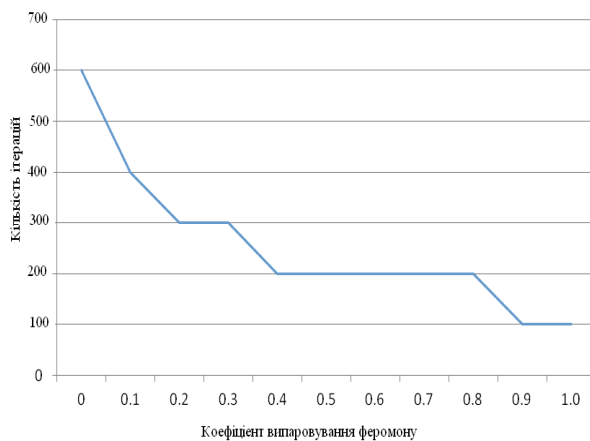


Рис. 8. Кількість ітерацій пошуку оптимального шляху за коефіцієнтом випаровування феромону

Fig. 8. Number of iterations to find for the optimal path by the pheromone evaporation coefficient

### Наукова новизна та практична значимість

Усі дослідження проведено для гіпотетичної комп'ютерної мережі із 7 маршрутизаторами та 17 каналами, при цьому: початковий рівень феромону дорівнює – 1; кількість звичайних мурах – 7; кількість елітний мурах 1. Граф такої мережі представлений на рис. 9 (значення затримок на маршрутизаторах подано в мкс).

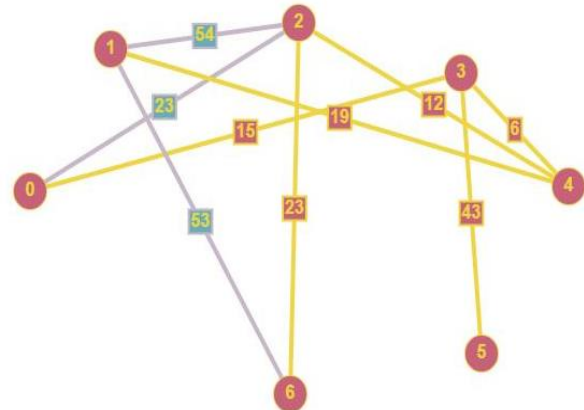


Рис. 9. Граф мережі, на якому проведено дослідження

Fig. 9. Graph of the network on which the study was conducted

Розглянемо залежність часу роботи програмної моделі в разі зміни таких параметрів мурашиного алгоритму: коефіцієнта випаровування (рис. 10); кількості елітних мурах (рис. 12); кількості звичайних мурах (рис. 14); кількості ітерацій (рис. 16); відкладення феромону (рис. 18).

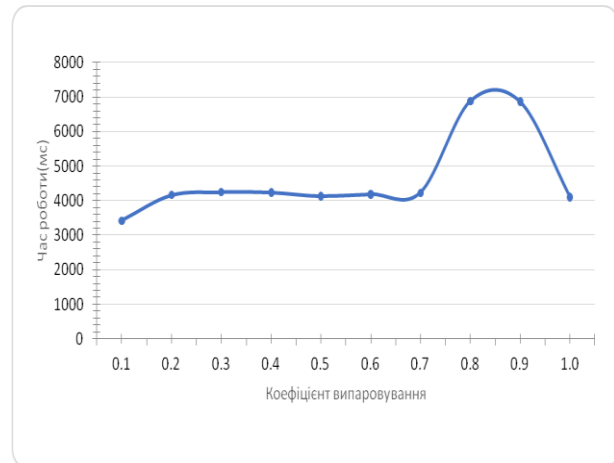


Рис. 10. Час роботи програмної моделі за коефіцієнтом випаровування

Fig. 10. Operating time of the software model by the evaporation coefficient

Із рис. 10 видно, що для комп'ютерної мережі із 7 маршрутизаторів доцільно використовувати коефіцієнт випаровування від 0,2 до 0,7. Відповідний скриншот програмної моделі представлений на рис. 11.



ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

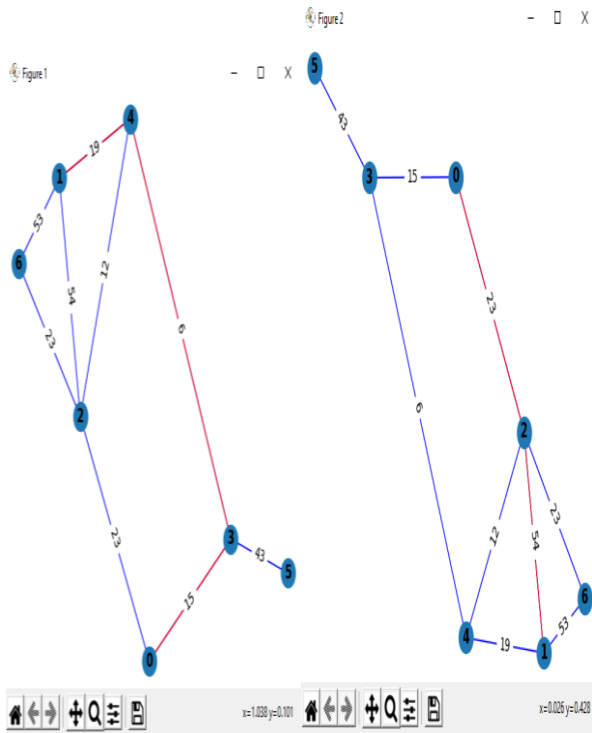


Рис. 11. Маршрут за програмною моделлю у першому дослідженні

Fig. 11. Route according to the software model in the first study

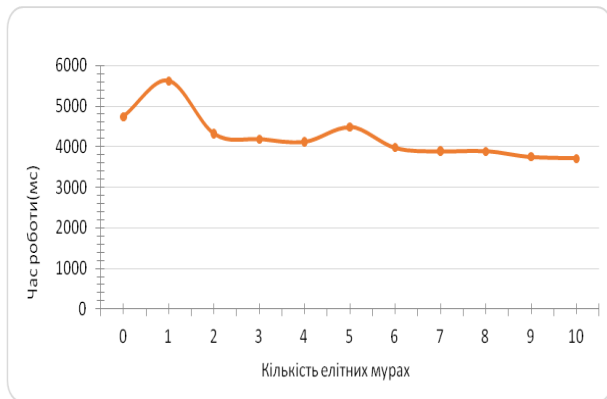


Рис. 12. Час роботи програмної моделі за кількістю елітних мурах

Fig. 12. Operating time of the software model by the number of elite ants

Рис. 12 демонструє: чим більше елітних мурах у колонії, тим швидше надходить оптимальне рішення. Однак зі збільшенням кількості елітних мурах зростає ймовірність того, що програмна модель буде застрягати на локально

оптимальних рішеннях; у нашому випадку достатньо мати двох елітних мурах на колонію. Відповідний скриншот моделі представлений на рис. 13.

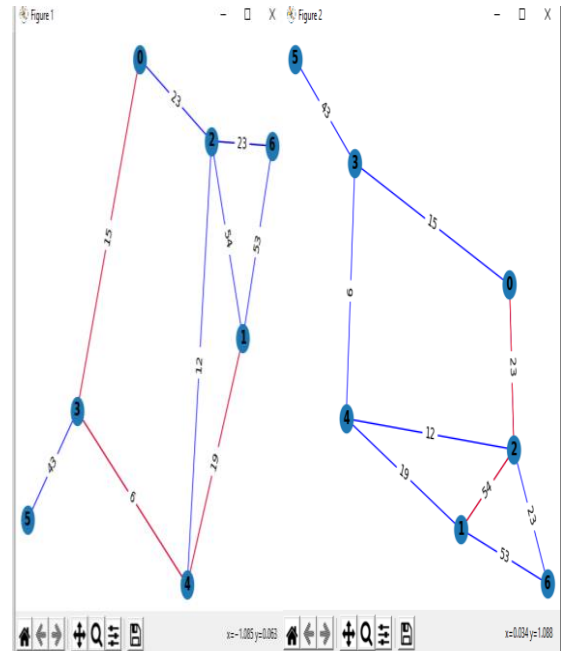


Рис. 13. Маршрут у програмній моделі в другому дослідженні

Fig. 13. Route in the software model in the second study

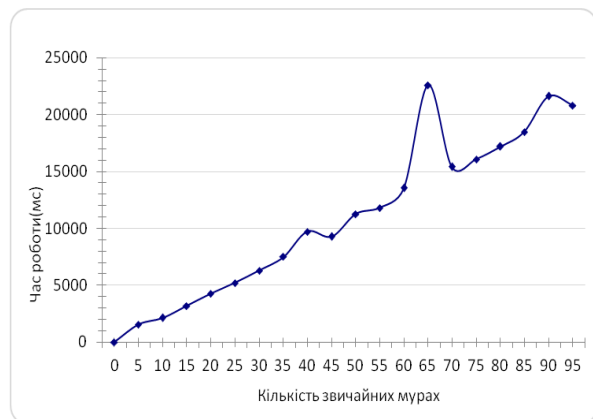


Рис. 14. Час роботи програмної моделі за кількістю звичайних мурах

Fig. 14. Operating time of the software model by the number of ordinary ants

Як бачимо з рис. 14, зі збільшенням кількості звичайних мурах у колонії зростає час виконання програмної моделі. Однак кількості мурах, що дорівнює кількості маршрутизаторів



ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

у комп’ютерній мережі, вистачає для знаходження оптимального рішення. Відповідний скріншот програмної моделі представлений на рис. 15.

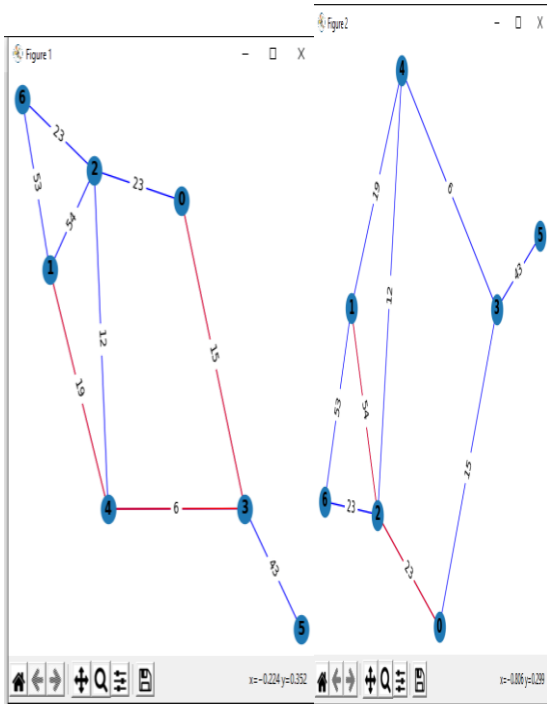


Рис. 15. Маршрут у програмній моделі в третьому дослідженні

Fig. 15. Route in the software model in the third study

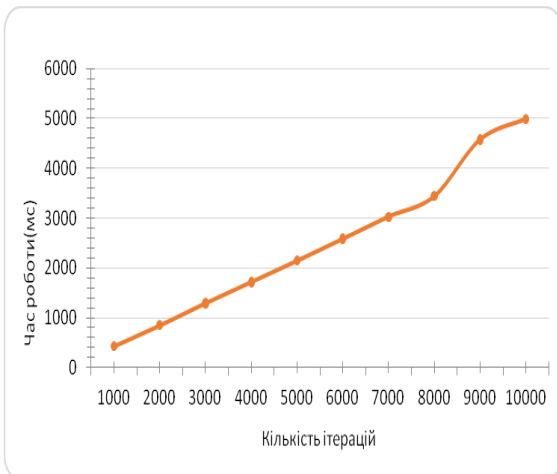


Рис. 16. Час роботи програмної моделі за кількістю ітерацій

Fig. 16. Operating time of the program model by the number of iterations

Із рис. 16 видно, що в разі збільшення кількості ітерацій мурашиного алгоритму збільшується час роботи програмної моделі. Проте для невеликих комп’ютерних мереж 1 000 ітерацій вистачає. Відповідний скріншот програмної моделі представлений на рис. 17.

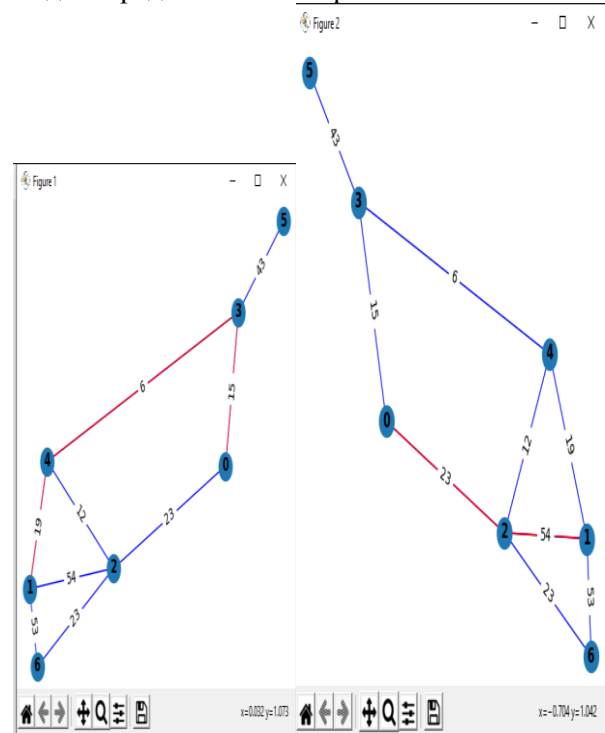


Рис. 17. Маршрут у програмній моделі в четвертому дослідженні

Fig. 17. Route in the software model in the fourth study



Рис. 18. Час роботи програмної моделі за кількістю відкладеного феромону

Fig. 18. Operating time of the software model by the amount of deposited pheromone

## ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ ТА МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

Як свідчить рис. 18 для комп'ютерної мережі із 7 маршрутизаторів значення відкладеного феромону може належати діапазону [5, 8], але достатньо задати його близьким до одиниці. Відповідний скриншот програмної моделі представлений на рис. 19.

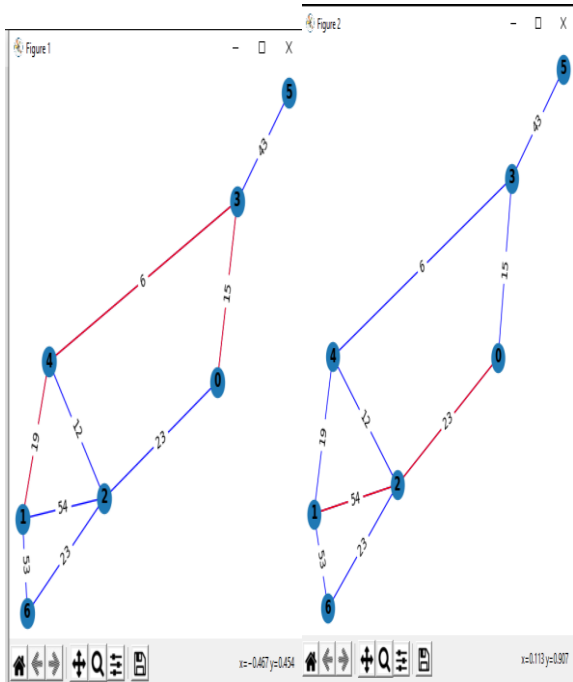


Рис. 19. Маршрут у програмній моделі в п'ятому дослідженні

Fig. 19. Route in the software model in the fifth study

### Висновки

Для організації маршрутизації в комп'ютерних мережах можна використовувати мурашиний алгоритм, до основних положень якого належать: стигмержі (stigmergy) та феромон (pheromone). Поведінка мурах може мати різний математичний опис, тому існують такі модифікації: метод мурашиних систем, який використовує елітну стратегію; метод мурашиних систем, заснований на ранжуванні; метод системи мурашиних колоній; максимінний метод мурашиних систем; метод мурашиних колоній

зі стратегією королеви. Для подальшої роботи обрано метод мурашиних колоній з елітною стратегією, а для його реалізації – мову Python із широким спектром сучасних стандартних засобів.

1. Для визначення оптимального маршруту в комп'ютерній мережі за мурашиним алгоритмом елітної стратегії створена мовою Python програмна модель «Two-ACO», вхідні параметри якої: кількість маршрутизаторів у мережі; матриця суміжності мережі та параметри мурашиного алгоритму (кількість ітерацій алгоритму; кількість звичайних та елітних мурах у колонії; початковий рівень феромону, коефіцієнт випаровування; параметр для регулювання величини відкладеного феромону мурахами на ребрах графа мережі).

2. За створеною програмною моделлю «Two-ACO» для гіпотетичної комп'ютерної мережі (із 7 маршрутизаторів та 17 каналів) проведено дослідження часу визначення оптимального шляху за кількістю звичайних та елітних мурах, значень коефіцієнта випаровування та відкладеного феромону. З'ясовано, що достатньо використовувати кількість мурах, що дорівнює кількості маршрутизаторів, мати в колонії 2 елітних мурах, при цьому вистачає 1 000 ітерацій, коефіцієнта випаровування від 0,2 до 0,7, а значення відкладеного феромону мурахами близького до одиниці.

3. Визначено оптимальний маршрут у комп'ютерній мережі залізничного транспорту (із 15 маршрутизаторів та 17 каналів) на основі моделі «Two-ACO» з використанням двох колоній мурах за такими критеріями: загальною затримкою на маршрутизаторах (для першої колонії мурах) та кількістю хопів, що складають маршрут (для другої колонії мурах). Установлено, що достатньо мати 30 агентів (двох мурах на вершину), значення феромону, відкладеного агентами, близьке до одиниці, при цьому коефіцієнт випаровування складає 0,4.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Глушко С. И., Какатунова Т. В. Нечеткая модификация алгоритма муравьиных колоний. *Научное обозрение*. 2013. № 1. С. 377–381.
2. Павленко А. И., Титов Ю. П. Сравнительный анализ модифицированных методов муравьиных колоний. *Прикладная информатика*. 2012. № 4 (40). С. 100–112.

3. Пахомова В. М., Опрятний А. О. Визначення оптимального маршруту в захищеній комп'ютерній мережі з використанням двохколоніального мурашиного алгоритму. *International scientific-practical conference «International scientific integration '2020»* (13–14 nov 2020). USA, 2020.
4. Скаков Е. С., Малыш В. Н. Пчелиный алгоритм оптимизации для решения задачи планирования беспроводной сети. *Программные продукты и системы*. 2016. № 4 (67). С. 67–73. DOI: <https://doi.org/10.15827/0236-235X.115.067-073>
5. Смирнова О. С., Богорадникова А. В., Блинов М. Ю. Описание роевых алгоритмов, инспирированных неживой природой и бактериями, для использования в онтологической модели. *International Journal of Open Information Technologies*. 2015. Vol. 3, № 12. С. 28–37.
6. Штовба С. Д. Муравьиные алгоритмы. *Exponenta Pro. Математика в приложениях*. 2003. № 4. С. 70–75.
7. Davidovic T., Teodorovic D., Selmic M. Bee Colony Optimization – Part I : The Algorithm Overview : Invited survey. *YJOR*. 2015. Vol. 25. Iss. 1. P. 33–56. DOI: <https://doi.org/10.2298/YJOR131011017D>
8. Dli M. I., Glushko S. I., Ivanova I. V. Ant algorithms as a tool of infrastructure project management. *European Science and Technology : materials of the III research and practice conference (Munich, 2012)*. Munich, 2012. P. 127–129.
9. Dorigo M., Gambardella L. M. Ant Colony System : A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Trans. on Evolutionary Computation*. 1997. Vol. 1. Iss. 1. P. 53–66. DOI: <https://doi.org/10.1109/4235.585892>
10. Hussein W. A., Sahrani S., Sheikh Abdullah S. N. H. The variants of the Bees Algorithm (BA) : A survey. *Artificial Intelligence Review*. 2017. Vol. 47. Iss. 1. P. 67–121. DOI: <https://doi.org/10.1007/s10462-016-9476-8>
11. Kumar K., Zindani D., Davim J. P. *Bees Algorithm*. Optimizing Engineering Problems through Heuristic Techniques. 2019. pp. 43–50. DOI: <https://doi.org/10.1201/9781351049580-5>
12. Pakhomova V. M., Mandybura Y. S. Optimal route definition in the railway information network using neural-fuzzy models. *Наука та прогрес транспорту*. 2019. № 5 (83). P. 81–98. DOI: <https://doi.org/10.15802/stp2019/184385>
13. Pakhomova V. M., Tsykalo I. D. Optimal route definition in the network based on the multilayer neural model. *Наука та прогрес транспорту*. 2018. № 6 (78). P. 126–142. DOI: <https://doi.org/10.15802/stp2018/154443>
14. Vaishali S. *Nature – All Mathematics*. URL: <http://thebridge.psgtech.ac.in/index.php/2014/11/12/nature-all-mathematics>.
15. Ying Tan. Survey of swarm intelligence. *Swarm Intelligence Vol. 1 : Principles, current algorithms and methods*. 2018. P. 1–28. DOI: [https://doi.org/10.1049/pbce119f\\_ch1](https://doi.org/10.1049/pbce119f_ch1)

V. M. PAKHOMOVA<sup>1\*</sup>, A. O. OPRIATNYI<sup>2\*</sup>

<sup>1\*</sup>Dep. «Electronic Computing Machines», Dnipro National University of Railway Transport named after Academician V. Lazaryan, Lazaryana St., 2, Dnipro, Ukraine, 49010, tel. +38 (056) 373 15 89, e-mail viknikpakh@gmail.com, ORCID 0000-0002-0022-099X

<sup>2\*</sup>Dep. «Electronic Computing Machines», Dnipro National University of Railway Transport named after Academician V. Lazaryan, Lazaryana St., 2, Dnipro, Ukraine, 49010, tel. +38 (056) 373 15 89, e-mail olkdi@outlook.com, ORCID 0000-0001-7145-9677

## Software Model for Determining the Optimal Routes in a Computer Network Based on the Two-Colonial Ant Algorithm

**Purpose.** At present, the computer networks of the information and telecommunication system (ITS) of railway transport use the OSPF protocol, which does not allow taking into account several metrics when determining the optimal route. Therefore, there is a need to study the possibility of organizing routing in computer networks of rail transport ITS using a two-colonial ant algorithm. **Methodology.** According to the Two-ACO software model, created in the Python language based on the two-colonial ant algorithm, the optimal route in a computer network was determined. Two-ACO model inputs: computer network parameters (network adjacency matrix, number of routers); parameters of the ant algorithm (number of iterations; number of ants in the colony; number of elite ants; initial pheromone level; evaporation rate; parameter for adjusting the amount of pheromone deposition). **Findings.** The results of the Two-ACO model are presented in the form of graphs depicting the optimal paths: the criterion of the total delay on the routers (for the first colony of ants) and the number of hops (for the second colony of ants). **Originality.** According to the created Two-ACO software model for a computer network of 7 routers and 17 channels, a study of the time for determining the optimal path in a computer network by the number of ordinary and elite

ants, evaporation rate and deposited pheromone was conducted. It is determined that it is enough to use the number of ants equal to the number of routers and have 2 elite ants in the colony, with 1000 iterations, evaporation rate from 0.2 to 0.7, and pheromone deposition by ants close to one. **Practical value.** Created Two-ACO software model using two colonies of ants on the following criteria: the total delay on the routers (for the first colony of ants) and the number of hops that make up the route (for the second colony of ants) allows you to parallel determine the optimal routes in a computer network of railway transport. It is estimated that for a computer network of 15 routers and 17 channels, it is sufficient to have 30 agents (two ants on top), the value of the pheromone deposited by the agents is close to one, and the evaporation rate is 0.4.

**Keywords:** railway transport; computer network; router; delay; two-colonial ant algorithm; pheromone; deposition; evaporation; search time

## REFERENCES

1. Glushko, S. I., & Kakatunova, T. V. (2013). Fuzzy modification of the algorithm of ant colonies. *Scientific Review, 1*, 377-381. (in Russian)
2. Pavlenko, A. I., & Titov, Yu. P. (2012). Srovnitelnyy analiz modifitsirovannykh metodov muravinykh koloniy. *Prikladnaya informatika 4(40)*, 100-112. (in Russian)
3. Pakhomova, V. M., & Opriatny, A. A. (2020). Determination of the optimal route in a secure computer network using a two-colonial ant algorithm. *International scientific-practical conference «International scientific integration '2020»*. USA. (in Ukrainian)
4. Skakov, E., & Malyshev, V. (2016). Bee colony optimization algorithm for wireless network planning problem solving. *SOFTWARE & SYSTEMS, 4(67)*, 67-73. DOI: <https://doi.org/10.15827/0236-235X.115.067-073> (in Russian)
5. Smirnova, O. S., Bogoradnikova, A. V., & Blinov, M. Yu. (2015). Opisaniye roevykh algoritmov, inspirirovannykh nezhyvoy prirodoy i bakteriyami, dlya ispolzovaniya v ontologicheskoy modeli. *International Journal of Open Information Technologies, 3(12)*, 28-37. (in Russian)
6. Shtovba, S. D. (2003). Muravinye algoritmy. *Exponenta Pro. Matematika v prilozeniyakh, 4*, 70-75. (in Russian)
7. Davidovic, T., Teodorovic, D., & Selmic, M. (2015). Bee Colony Optimization-Part I: The Algorithm Overview: Invited survey. *Yugoslav Journal of Operations Research, 25(1)*, 33-56. DOI: <https://doi.org/10.2298/YJOR131011017D> (in English)
8. Dli, M. I., Glushko, S. I., & Ivanova, I. V. (2012). Ant algorithms as a tool of infrastructure project management. *European Science and Technology: materials of the III research and practice conference* (pp. 127-129). Munich, Germany. (in English)
9. Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1(1)*, 53-66. DOI: <https://doi.org/10.1109/4235.585892> (in English)
10. Hussein, W. A., Sahran, S., & Sheikh Abdullah, S. N. H. (2016). The variants of the Bees Algorithm (BA): a survey. *Artificial Intelligence Review, 47(1)*, 67-121. DOI: <https://doi.org/10.1007/s10462-016-9476-8> (in English)
11. Kumar, K., Zindani, D., & Davim, J. P. (2019). *Bees Algorithm*. Optimizing Engineering Problems through Heuristic Techniques (pp. 43-50). DOI: <https://doi.org/10.1201/9781351049580-5> (in English)
12. Pakhomova, V. M., & Mandybura, Y. S. (2019). Optimal route definition in the railway information network using neural-fuzzy models. *Science and Transport Progress, 5(83)*, 81-98. DOI: <https://doi.org/10.15802/stp2019/184385> (in English)
13. Pakhomova, V. M., & Tsykalo, I. D. (2018). Optimal route definition in the network based on the multilayer neural model. *Science and Transport Progress, 6(78)*, 126-142. DOI: <https://doi.org/10.15802/stp2018/154443> (in English)
14. Vaishali, S. *Nature-All Mathematics*. Retrieved from <http://thebridge.psgtech.ac.in/index.php/2014/11/12/nature-all-mathematics> (in English)
15. Ying, Tan. (2018). Survey of swarm intelligence. *Swarm Intelligence-Vol. 1: Principles, current algorithms and methods*, 1-28. DOI: [https://doi.org/10.1049/pbce119f\\_ch1](https://doi.org/10.1049/pbce119f_ch1) (in English)

Надійшла до редколегії: 12.02.2021

Прийнята до друку: 14.06.2021