

Кропивницька В.Б.

Івано-Франківський національний технічний університет нафти і газу

Кропивницький Д.Р.

Івано-Франківський національний технічний університет нафти і газу

МОДЕЛЮВАННЯ РОБОТИ МУРАШИНИХ АЛГОРИТМІВ У ЗАДАЧАХ МАРШРУТИЗАЦІЇ

У статті представлено моделювання та порівняльний аналіз трьох варіантів алгоритму мурашиної колонії (АСО) для задач маршрутизації: класичного АСО, Elitist АСО (елітарного) та Min-Max АСО. Розглянуто принципи функціонування АСО-алгоритмів, що базуються на поведінці мурах у природі, зокрема на механізмах феромонного маркування маршрутів і позитивного зворотного зв'язку. Увагу приділено механізму глобального оновлення феромонів, правилам вибору маршруту, ролі елітних агентів, а також обмеженням на значення феромонної концентрації, які суттєво впливають на поведінку алгоритмів у процесі оптимізації. Для трьох розглянутих варіантів АСО сформовано порівняльну таблицю за критеріями: спосіб оновлення феромонів, наявність посилення найкращого рішення, обмеження феромону, швидкість збіжності та ризик передчасної конвергенції. Проведено серію експериментів із використанням змодельованих мереж різної складності, у межах яких змінювались розмір колонії, кількість ітерацій та інші параметри. Результати дослідження наведено у вигляді таблиць і графіків, які ілюструють вплив параметрів на ефективність побудови маршрутів та поведінку алгоритмів у різних умовах. Встановлено, що класичний алгоритм АСО забезпечує стабільність при малих масштабах задачі, алгоритм Elitist АСО забезпечує високу точність і збіжність при помірній кількості агентів, тоді як Min-Max АСО демонструє найвищу стабільність і стійкість до змін параметрів завдяки введенню обмежень на феромон. Аналіз показав, що вибір конкретної модифікації алгоритму повинен ґрунтуватися на характеристиках мережі, ресурсних обмеженнях системи та необхідному рівні точності, що підкреслює практичну значущість дослідження для задач маршрутизації в динамічних середовищах сучасних обчислювальних систем, зокрема в телекомунікаційних, мобільних, безпілотних та сенсорних мережах із високим ступенем мобільності та змінною топологією.

Ключові слова: алгоритм мурашиної колонії, маршрутизація, динамічні мережі, самоорганізація, моделювання, адаптивні алгоритми, роевий інтелект, параметри алгоритму, програмна реалізація.

Постановка проблеми. У сучасних телекомунікаційних, сенсорних, мобільних та безпілотних мережах спостерігається стале зростання потреби у гнучких і адаптивних методах маршрутизації, здатних ефективно працювати за умов динамічної топології та обмежених обчислювальних ресурсів. Традиційні алгоритми виявляються недостатньо ефективними або занадто інертними для задоволення вимог таких середовищ. За таких умов алгоритми мурашиної колонії (АСО), що належать до класу методів роевого інтелекту, демонструють високу здатність до адаптації, масштабованості та знаходження якісних рішень у задачах складної оптимізації.

Їхня ефективність особливо проявляється у сфері маршрутизації в мережах WSN, MANET, FANET, VANET і IoT, де потрібна швидка реак-

ція на зміни й оптимальне використання ресурсів. Це визначає актуальність розробки та порівняльного аналізу різних модифікацій алгоритмів АСО для задач комбінаторної оптимізації на графах, зокрема задачі маршрутизації.

Аналіз останніх досліджень і публікацій.

У рамках дослідження ефективності мурашиних алгоритмів у задачах маршрутизації варто зазначити, що останні роки характеризуються зростанням інтересу до їх застосування в умовах динамічних мереж. Зокрема, у роботі [1] запропоновано модифікований протокол маршрутизації для FANET, у якому алгоритм мурашиної колонії використовується для вибору маршрутів з урахуванням параметрів якості зв'язку та енергоспоживання, що забезпечує низькі накладні витрати та високу ефективність у динамічних мережах. У дослідженні Шпорта і Лаврінця [2]

розглянуто алгоритм AntHocNet, який у ряді метрик (PDR, затримка, продуктивність) перевершує класичні рішення. Значну практичну цінність має також дослідження АСО у сфері інтелектуального регулювання дорожнього руху [3], де алгоритм застосовано для адаптивного керування світлофорами – що підтверджує універсальність АСО підходу у задачах реального часу. У роботі [4] автори запропонували модифікацію АСО (ACORA-WR) для балансування навантаження в супутникових LEO-мережах, де обмеження вікна пошуку дозволяє підвищити ефективність при зниженні обчислювальних витрат. Водночас, у публікації [5] представлено алгоритм AntSynNet, що застосовується для синтезу мережових структур, орієнтованих на оптимізацію маршрутизації з урахуванням вартості та трафіку. Окремо заслуговує на увагу новітній підхід FAOA – алгоритм фермерських мурах [6], який демонструє конкурентні результати порівняно з іншими метаевристичними методами (GA, PSO), розширюючи рамки біологічно натхнених алгоритмів оптимізації. Високу ефективність мурашині алгоритми демонструють при оптимізації розподілених нестационарних систем. Авторами роботи [7] запропоновано адаптацію АСО-алгоритму для задачі комівояжера, де час переміщення між містами задано у вигляді нечітких трапецієподібних чисел. Запропонований підхід дозволяє отримувати близькі до оптимальних маршрути навіть за невеликої кількості ітерацій, що демонструє ефективність АСО у задачах з невизначеними параметрами та нечіткістю входних даних. Яскравим прикладом може служити знаходження мурашиними алгоритмами оптимальних трафіків в телекомунікаційних мережах [8]. У роботі [9] на прикладі завдання маршрутизації показано, як в алгоритми вирішення дискретних завдань оптимізації упровадити складові самоорганізації мурашок: випадковість, багатократність взаємодії, негативні і позитивні зворотні зв'язки. Враховуючи активний розвиток АСО-алгоритмів і розмаїття сфер їх застосування, доцільним є узагальнене дослідження їх ефективності на прикладі задач маршрутизації.

Постановка завдання. Метою роботи є дослідження та порівняльний аналіз ефективності трьох варіантів алгоритмів мурашиної колонії – класичного АСО, елітарного (Elitist ACO) та Min-Max АСО – у задачах маршрутизації на графах з різною складністю, а також моделювання їхньої роботи з візуалізацією процесу пошуку оптимального рішення.

Виклад основного матеріалу. Алгоритм мурашиної колонії (АСО) базується на спостереженнях за колективною поведінкою мурах, які, залишаючи феромонні сліди на маршрутах до джерела їжі, з часом зосереджуються на найкоротших шляхах завдяки позитивному зворотному зв'язку. Цей біологічний принцип покладено в основу низки алгоритмічних модифікацій, серед яких у даній роботі розглянуто три основні варіанти: класичний АСО, Elitist АСО (елітарний) та Min-Max АСО.

Ключовим етапом у роботі АСО-алгоритмів є механізм глобального оновлення феромонів, що реалізується після кожної ітерації та визначається за такою формулою (класичний алгоритм АСО):

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

де τ_{ij} – є феромоном на компоненті між станом i і j і p , це параметр, який використовується для зміни рівня випаровування.

В елітарних системах АСО математична процедура оновлення визначається за такими рівняннями:

$$\Delta\tau_{ij}^{bs} = \begin{cases} \frac{Q}{C^{bs}}, & \text{якщо компонент належить елітній мкрасі} \\ 0, & \text{інакше} \end{cases}$$

інакше

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^M \Delta\tau_{ij}^k + e\Delta\tau_{ij}^{bs}$$

де C^{bs} – загальна вартість рішення, Q – параметр для регулювання кількості феромонів (як правило дорівнює 1), e – параметр, який використовується для регулювання кількості додаткового феромону, що надається найкращому (або «елітному») розчину.

Алгоритм Min-Max подібний до елітарного АСО тим, що зосереджується на посиленні найбільш ефективних маршрутів, знайдених під час пошуку. Однак у Min-Max замість того, щоб просто надавати додаткову вагу елітним рішенням, лише найкраще поточне або найкраще глобальне рішення дозволяє відкладати феромонний слід. Крім того, Min-Max вимагає, щоб сліди феромонів знаходились в межах між максимальним і мінімальним значенням.

Таблиця 1 узагальнює ключові відмінності між трьома варіантами алгоритму мурашиної колонії.

Порівняльна характеристика алгоритмів АСО

Параметр	АСО	Elitist ACO	Min-Max ACO
Хто оновлює феромони?	Усі мурашки	Усі + краща мурашка	Тільки кращі мурашки
Додаткове посилення	Немає	Є	Немає
Обмеження феромонів	Немає	Немає	Так ($\tau_{\min} \leq \tau \leq \tau_{\max}$)
Швидкість збіжності	Помірна	Висока	Помірна стабільна
Ризик передчасної збіжності	Середній	Високий	Низький

На рис. 1 представлена узагальнена блок-схема мурашиного алгоритму.

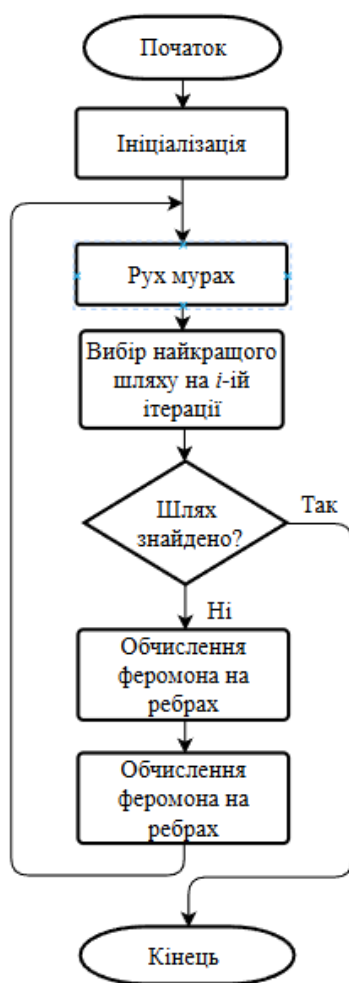


Рис. 1. Узагальнена блок-схема мурашиного алгоритму

Для того, щоб реалізувати відповідний мурашковий алгоритм для вирішення задачі маршрутизації, потрібно:

1. Зобразити задачу як граф, де вершини – це окремі компоненти (наприклад, вузли мережі), а ребра – можливі шляхи з відповідними

вагами (наприклад, відстань, енерговитрати), якими можуть рухатися мурахи при побудові маршрутів.

2. Визначити значення сліду феромону.

3. Сформулювати евристичне правило, яке керує вибором мурашки на кожному етапі маршруту, враховуючи параметри мережі (наприклад, відстань, якість з'єднання тощо).

4. Якщо дозволяють умови задачі, застосувати локальний пошук для уточнення маршруту, тобто спробувати покращити вже побудований шлях, змінюючи його невеликі частини.

5. Визначити, яку саме модифікацію мурашиного алгоритму застосувати, і реалізувати її для розв'язання задачі.

6. Встановити оптимальні параметри для обраного алгоритму мурашиної колонії. Необхідно також визначити: кількість мурашок, що беруть участь у пошуку; співвідношення між дослідженням нових маршрутів і використанням вже знайдених; час та правилом оновлення значення феромону.

Програма, яка моделює роботу алгоритмів мурашиної колонії (АСО): має структуру показану на рисунку 2 [10]. Схема відображає взаємозв'язки між основними компонентами програми:

- Canvas – відповідає за графічне відображення карти, міст, ребер та мурашок.

- Graph – структура, яка зберігає всі міста та з'єднання між ними (ребра з феромонами).

- AntColony – координує роботу всієї колонії мурашок, викликає створення мурашок та оновлення феромонів.

- Ant – окрема мурашка, яка проходить шлях між містами та формує маршрут.

- Tour – клас, що описує маршрут, який пройшла мурашка, з обчисленням загальної довжини.

- ACArtist – модуль, що відповідає за візуалізацію руху мурашок та процесу оптимізації.

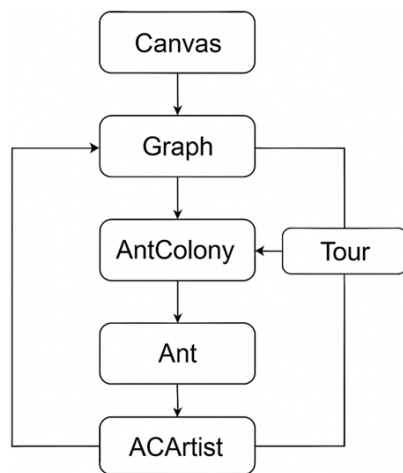


Рис. 2. Структура взаємодії модулів програми

Для дослідження і порівняння трьох мурашиних алгоритмів за заданою топологією мережі було вибрані такі параметри:

- розмір колонії мурах змінювався від 10 до 200;
- кількість ітерацій змінювалась від 50 до 500;
- значення коефіцієнтів початкової концентрації феромона: $\alpha = 2$, $\beta = 2$ (α – коефіцієнт значимості концентрації феромона; β – коефіцієнт значимості емпіричної інформації);
- значення початкової ймовірності переходу мурах від одного вузла до іншого $p_0 = 0,1$;
- початковий феромон – 2;
- вага феромонного депозиту – 1.

Для мережі, яка складається з 7 вузлів (рис. 3) було проведено серію дослідів при фіксованій кількості ітерацій, яка дорівнює 100 і при різних значеннях розміру колоній мурах: 10, 20, 30, 50, 100, 200.

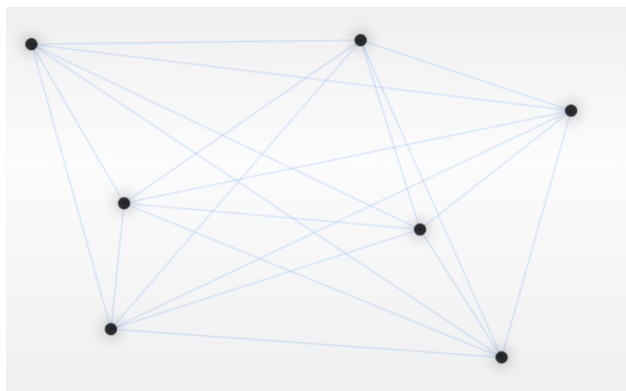


Рис. 3. Топологія мережі з 7 вузлів

У результаті моделювання було отримано дані, наведені в таблиці 2.

Таблиця 2
Результати моделювання для мережі з 7 вузлів

Розмір колонії	Найкоротший шлях		
	ACO	Elit	Min-Max
10	1583,67	1583,67	1583,67
20	1583,67	1583,67	1583,67
30	1583,67	1583,67	1583,67
50	1583,67	1583,67	1583,67
100	1583,67	1583,67	1583,67
200	1583,67	1594,16	1583,67

Класичний алгоритм мурашиної колонії АСО та алгоритм Min-Max дають повністю стабільний результат при будь-якому розмірі колонії – 1583.67. Це свідчить про їхню високу стабільність при малих розмірах колонії. Елітарний алгоритм показує ідентичні результати до АСО до розміру колонії 100. Проте при збільшенні кількості мурах вартість маршруту зростає до 1594.16, що може вказувати на перенаванчання, накопичення надлишкового феромону або деградацію ефективності при занадто великій популяції.

Друга серія експериментів була проведена для мережі, яка складається з 30 вузлів. При цьому було кількість мурах в колонії дорівнювала: 10, 20, 30, 50, 100, 200; а кількість ітерацій дорівнювала: 50, 100, 200, 300, 500.

Результати моделювання занесено в таблицю 3. На рисунку 4 відображено графічне представлення результатів моделювання роботи трьох алгоритмів при кількості ітерацій $N = 200$ та величині колонії мурах 30.

Таблиця 3
Результати моделювання для 30 вузлів

Розмір колонії	Кількість ітерацій	Найкоротший шлях		
		ACO	Elit	Min-Max
50	50	3410,39	3224,65	3013,57
50	100	3009,44	3008,24	3089,68
50	200	3021,59	2995,59	2988,07
50	300	3039,81	2954,54	2994,07
50	500	2953,34	2947,38	2968,65
10	200	3009,44	3006,32	3115,61
20	200	3062,42	2654,54	2975,07
30	200	2958,48	2995,59	2947,38
50	200	3021,59	2995,59	2988,07
100	200	3039,37	3104,93	3028,84
200	200	3086,09	3046,84	3100,89

Результати таблиці 3 узагальнено у вигляді графіків (рис. 5 та 6), що ілюструють вплив кількості ітерацій і розміру колонії на ефективність кожного з трьох алгоритмів. На графіку (рис. 5) зображено залежність якості знайденого маршруту від кількості ітерацій для трьох алгоритмів: АСО, Elit та Min-Max, при фіксованому розмірі колонії, що дорівнює 50.

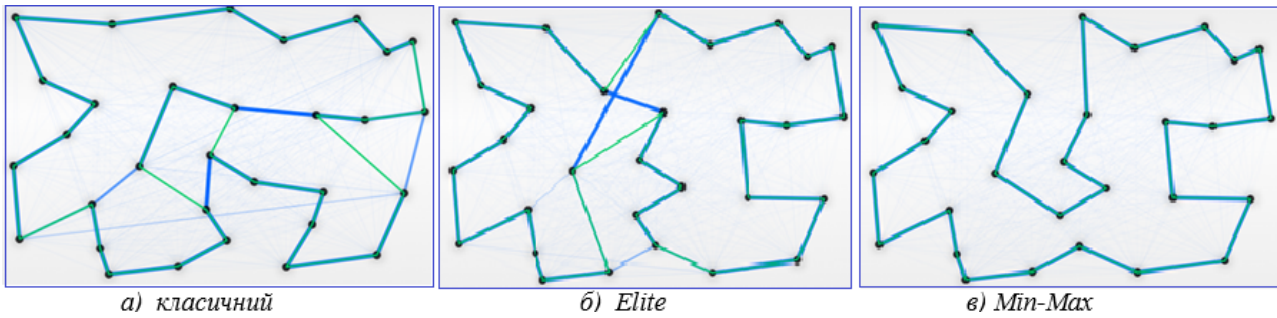


Рис. 4. Найкоротший шлях. Алгоритм ACO

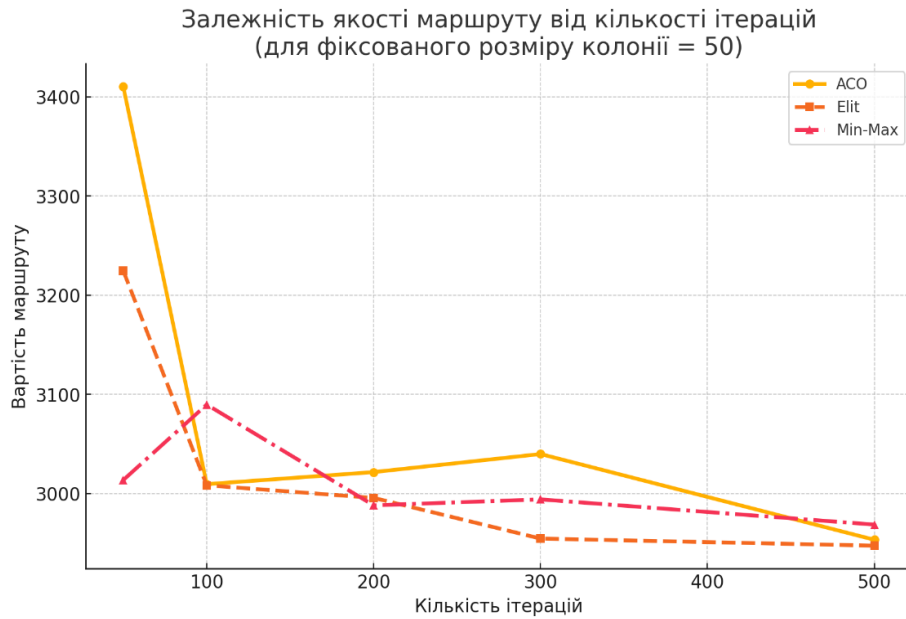


Рис. 5. Залежність якості знайденого маршруту від кількості ітерацій

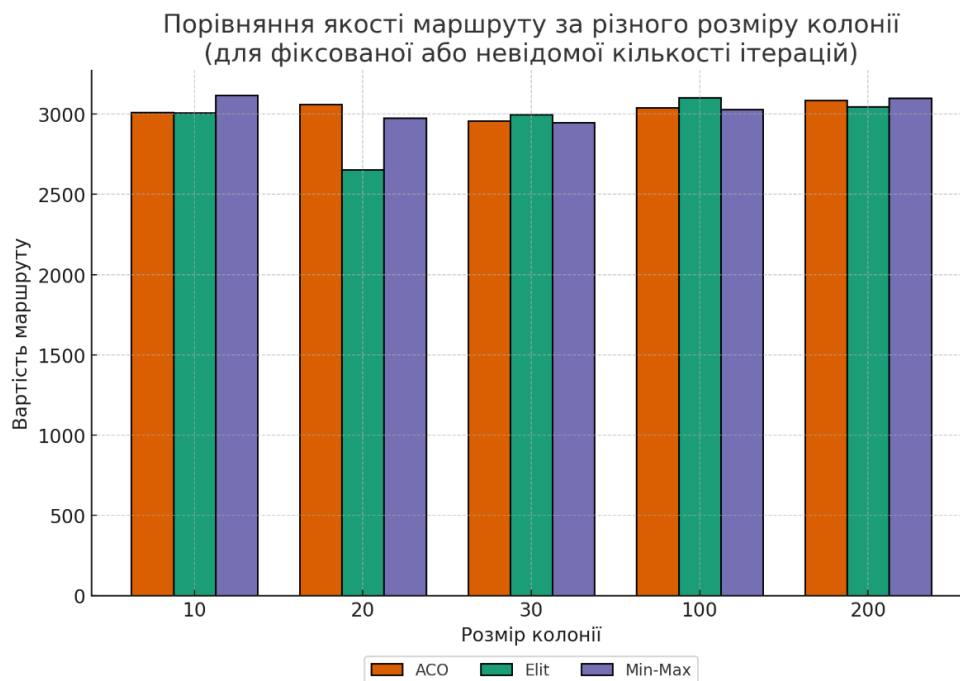


Рис. 6. Залежність якості маршруту від розміру колонії

На основі графічної інтерпретації можна зробити висновок, що алгоритм АСО при малій кількості ітерацій (50) демонструє найгірший результат, формуючи найдовший маршрут. Проте зі збільшенням кількості ітерацій до 500, якість його рішень поступово покращується, хоча й не досягає рівня інших алгоритмів. Алгоритм Elit виявляє стабільну тенденцію до зниження вартості маршруту при зростанні кількості ітерацій і вже з 200 ітерацій випереджає інші варіанти за ефективністю. Алгоритм Min-Max спочатку демонструє коливання, однак також поступово покращує результат і наближається до Elit при великій кількості ітерацій.

Як показує діаграма (рис. 6), збільшення кількості мурах у колонії призводить до незначного покращення результатів для алгоритму АСО. Алгоритм Elit, навпаки, вже при невеликому розмірі колонії досягає стабільно хороших результатів, і подальше збільшення кількості агентів майже не впливає на покращення. Найбільш виражену позитивну динаміку демонструє Min-Max: при збільшенні розміру колонії від 10 до 50 вартість маршруту значно знижується, що свідчить про здатність алгоритму ефективніше знаходити рішення в умовах більшої популяції. Таким чином, діаграма демонструє, що розмір колонії впливає на ефективність алгоритмів, але Elit виявляється найстабільнішим і найрезультативнішим у більшості випробувань. Це може свідчити про його здатність ефективно адаптуватися до задачі вже на ранніх ітераціях і з невеликим числом агентів.

Висновки. У результаті проведеного дослідження було встановлено, що всі три розглянуті варіанти алгоритму мурашиної колонії – класичний АСО, елітарний Elitist АСО та Min-Max АСО – здатні ефективно вирішувати задачі маршрутизації, проте кожен з них має свої особливості. Класичний АСО забезпечує стабільні результати при невеликій складності задачі, але втрачає ефективність при збільшенні розмірів мережі або кількості ітерацій. Elitist АСО демонструє високу точність і швидку збіжність при невеликих розмірах колонії, однак при надмірному збільшенні кількості агентів може спостерігатися перенасичення феромонного поля, що знижує якість рішень. Min-Max АСО виявився найбільш стійким до змін параметрів, зберігаючи баланс між дослідженням нових маршрутів і використанням вже знайдених рішень, завдяки введенню обмежень на значення феромону. Моделювання показало, що оптимальні результати досягаються при правильному налаштуванні параметрів алгоритму з урахуванням розміру мережі, кількості ітерацій та розміру колонії. Найкращі результати при великій кількості ітерацій показав алгоритм Elitist АСО, тоді як Min-Max АСО виявився найбільш стабільним у широкому діапазоні умов. Таким чином, доцільність використання того чи іншого варіанта мурашиного алгоритму залежить від конкретних вимог до задачі маршрутизації, ресурсних обмежень системи та необхідного рівня точності.

Список літератури:

1. Wang, Z., Yu, X., Wang, B., Cheng, S. A Low-Overhead Routing Protocol for FANET Based on Ant Colony Algorithm. In: Gao, F., Wu, J., Li, Y., Gao, H. (eds) *Communications and Networking*. ChinaCom 2022. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol 500. Springer, Cham. 2023. P. 321–333. https://doi.org/10.1007/978-3-031-34790-0_25
2. Шпорт Д. В., Лаврінець К. Г. Експериментальна оцінка можливості використання алгоритму мурашиної колонії – AntHocNet для вирішення задачі маршрутизації у FANET. Зв'язок. 2024. № 6 (172). С. 53–63. DOI: <https://doi.org/10.31673/2412-9070.2024.060501>
3. Danyliuk A., Muliarevych O. Ant Colony Algorithm in Traffic Flow Control. *Advances in Cyber-Physical Systems*. 2024. Vol. 9, № 2. P. 158–163. <https://doi.org/10.23939/acps2024.02.158>
4. Deng, X., Zeng, S., Chang, L., Wang, Y., Wu, X., Liang, J., Ou, J., & Fan, C. (2021). An Ant Colony Optimization-Based Routing Algorithm for Load Balancing in LEO Satellite Networks. *Wireless Communications and Mobile Computing*. 2022 (1), 3032997. <https://doi.org/10.1155/2022/3032997>
5. Wilkinson, Simon & White, Tony. The AntSynNet Algorithm: Network Syntheses Using Ant Colony Optimization. *Proceedings of the International Conference on Artificial Intelligence, IC-AI '04*, Volume 2. Las Vegas, Nevada, USA. June 21–24, 2004. P. 648–663.
6. Asghari A, Zeinalabedinmalekmian M, Azgomi H, Alimoradi M, Ghaziantafrishi S. Farmer Ants Optimization Algorithm: A Novel Metaheuristic for Solving Discrete Optimization Problems. *Information*. 2025. 16 (3) : 207. <https://doi.org/10.3390/info16030207>
7. E. Ivohin, K. Yushtin. Use of Ant Colony Optimization Algorithm for Solving Fuzzy Problem of Traveling Salesman. *Advanced Information Technology*, 2024. Vol.1 (3). P. 23–31. <https://doi.org/10.17721/AIT.2024.1.03>

8. Berliński M., Rasmus M., Kopertowski Z. and Kozdrowski S. Ant Colony Algorithms Application for Telco Networks Performance with Multi-criteria Optimization. *2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. Split, Croatia. 2023. P. 1–3. DOI: 10.23919/SoftCOM58365.2023.10271586.
9. Dorigo M., Gambardella L. M. Ant Algorithms for Discrete Optimization. *Artificial Life*. № 5 (2). 1999. P. 137–172. *Artificial Life*. 5. 137–172. 10.1162/106454699568728.
10. Recent Tutorials. Веб-сайт: <https://www.theprojectspot.com/tutorials> (дата звернення: 10.04.2025)

Kropyvnytska V.B., Kropyvnytskyi D.R. MODELING THE PERFORMANCE OF ANT COLONY ALGORITHMS FOR NETWORK ROUTING TASKS

This article presents the modeling and comparative analysis of three variants of the Ant Colony Optimization (ACO) algorithm applied to routing tasks: the classical ACO, Elitist ACO, and Min-Max ACO. The paper explores the functioning principles of ACO algorithms, which are inspired by the natural behavior of ants—particularly mechanisms of pheromone trail marking and positive feedback. Special attention is paid to the process of global pheromone update, path selection rules, the role of elite agents, and the implementation of pheromone concentration limits, all of which significantly influence algorithm behavior during optimization. A comparative table is constructed for the three considered ACO variants based on the following criteria: pheromone update method, reinforcement of the best solution, pheromone bounds, convergence speed, and risk of premature convergence. A series of experiments was conducted using simulated networks of varying complexity, where colony size, iteration count, and other parameters were adjusted. The results are presented in the form of tables and graphs illustrating how parameter changes affect route-building efficiency and algorithm behavior under different conditions. The classical ACO algorithm showed stability on small-scale tasks, Elitist ACO achieved high accuracy and convergence with a moderate number of agents, while Min-Max ACO demonstrated the greatest stability and robustness to parameter changes due to the introduction of pheromone value limits. The analysis indicates that the selection of a specific algorithm modification should be based on network characteristics, system resource constraints, and required solution accuracy, highlighting the practical relevance of this research for routing tasks in dynamic environments of modern computing systems, particularly in telecommunications, mobile, unmanned, and sensor networks with high mobility and variable topology.

Key words: ant colony algorithm, routing, dynamic networks, self-organization, modeling, adaptive algorithms, swarm intelligence, algorithm parameters, software implementation.