

В. В. Токарев<sup>1</sup>, І. В. Ільїна<sup>1</sup>, І. І. Шевченко<sup>2</sup>, І. К. Гриценко<sup>1</sup>

<sup>1</sup> Харківський національний університет радіоелектроніки, Харків, Україна

<sup>2</sup> Науково-дослідний, проектно-конструкторський та технологічний інститут мікрографії, Харків

## ПРО ОДИН ПІДХІД ДО РІШЕННЯ АСИМЕТРИЧНОЇ TSP - ЗАДАЧІ ПРИ В2С ДОСТАВКАХ ЗА ДОПОМОГОЮ ПЛАТФОРМИ "SWARM-BOT" - SYSTEM У ФІЗИЧНОМУ НЕОРГАНІЗОВАНОМУ СЕРЕДОВИЩІ

**Анотація.** На сьогоднішній день у сфері надання послуг, після проведеного аналізу ринку попиту та пропозицій, проводяться дослідження з метою визначення можливості застосування платформи "Swarm-bot" - system для доставки замовлень кінцевим користувачам. У ролі "s-bots", що входять до складу однієї "Swarm-bot" - system, активно застосовуються дроне. Доставка замовлень та кореспонденції з використанням платформи "Swarm-bot" - system передбачає наявність спеціалізованого контейнера, розміщеного на корпусі дроне. Дослідження проводяться для того, щоб з'ясувати чи зможуть дроне замінити поштове повідомлення до якого ми всі звикли. При впровадженні платформи "Swarm-bot" - system в реальну експлуатацію, людина бере участь у процесі доставки як оператор, а дроне переміщуються заздалегідь запрограмованим маршрутом. На сьогоднішній день найбільш оптимальним для вирішення проблеми зменшення відстані доставки замовлення є рішення асиметричної TSP - задачі (travelling salesman problem). Ця стаття присвячена дослідженню можливості застосувати алгоритм, заснований на методі "гілок та меж", для вирішення асиметричної TSP - задачі при В2С доставках за допомогою платформи "Swarm-bot" - system у physical unorganized environment.

**Ключові слова:** "Swarm-bot" - system, "s-bot", embedded systems, business-to-business, business-to-consume, drones, комунікація, метод "гілок та меж", асиметрична TSP - задача.

### Вступ

У сфері надання послуг, після проведеного аналізу ринку попиту та пропозицій, американська компанія Amazon стала застосовувати платформу "Swarm-bot" - system для транспортування замовлень клієнтам [1,2]. У ролі "s-bots", що входять до складу однієї "Swarm-bot" - system, Amazon застосувала дроне. Доставка замовлень та кореспонденції з використанням платформи "Swarm-bot" - system передбачає наявність спеціалізованого контейнера, розміщеного на корпусі дроне [3-6].

Після Amazon, всесвітньо відома міжнародна компанія доставки вантажів і документів, "DHL - express", стала активно проводити наукові дослідження з метою визначення можливості застосування платформи "Swarm-bot" - system для доставки замовлень кінцевим користувачам. Дослідження ведуться, щоб з'ясувати, чи зможуть дроне замінити поштове повідомлення, до якого ми всі звикли. Про проведення подібних досліджень, з метою визначення можливості застосування платформи "Swarm-bot" - system, після проведення ряду успішних тестових доставок покупцям, опублікували свої звіти пошти Швейцарії, Сингапура та Австралії. Також були відмічені експерименти "DHL - express" і в Україні. При впровадженні платформи "Swarm-bot" - system у реальну експлуатацію, людина бере участь у процесі доставки як оператор, а дроне переміщуються заздалегідь запрограмованим маршрутом. У загальному вигляді приклад алгоритму транспортування замовлень кінцевим споживачам може мати такий вигляд: крок №1 - замовлення покупця або кореспонденція розміщуються у спеціалізованому контейнері; крок №2 - у "Swarm-bot" - system програмується маршрут доставки та розрахунковий час у дорозі; крок №3 - дроне, що входять до складу "Swarm-bot" - system, готуються до руху за заданим маршрутом; крок №4 -

дроне здійснюють доставку замовлення клієнтам або кореспонденції в задану точку, орієнтуючись на сигнал GPS, і в процесі руху можуть уточнювати своє місце розташування в просторі через оператора "Swarm-bot" - system за допомогою проміжних наземних станцій.

На сьогоднішній день основні зусилля розробників та проектувальників платформи "Swarm-bot" - system спрямовані на вирішення наступних завдань: як масштабувати платформу, якщо мова йтиме не тільки про В2В (business to business), але й про В2С (business-to-consume) доставки; як і яким способом повідомити клієнта про майбутню доставку; як і в який спосіб клієнт повинен підтвердити отримання замовлення; як вирішити проблему гарантії доставки замовлення; як вирішити проблему зменшення часу доставки замовлення; як вирішити проблему зменшення відстані доставки замовлення.

На сьогоднішній день найоптимальнішим для вирішення проблеми зменшення відстані доставки замовлення є рішення асиметричної TSP - задачі (travelling salesman problem).

**Мета статті** - дослідити алгоритм, заснований на методі "гілок і меж" для вирішення асиметричної TSP - задачі при В2С доставках за допомогою "Swarm-bot" - system у physical unorganized environment.

### Виклад основного матеріалу

В даний час науково обґрунтовано, що рішення NP (non-deterministic polynomial – недетермінований поліном) - складних задач більш ефективно тоді, коли застосовується платформа "Swarm-bot" - system у цілому, а не окремі елементи, що входять до її складу, наприклад, окремі дроне [7-10]. Тоді, при застосуванні платформи "Swarm-bot" - system, значно скорочується час В2С - доставки замовлень за рахунок кількості дроне, що входять до складу цієї "Swarm-bot" - system. TSP - задача відноситься до класу NP - складних. У загальній постановці ця

задача полягає у пошуку гамільтонова циклу, з умовою, що у повному орієнтованому графі сума ваг дуг буде мінімальна. Серед метаевристичних алгоритмів найбільшою популярністю рішення TSP - задачі користується алгоритм "Lin-Kernighan", проте на сьогоднішній день більш велику популярність отримав алгоритм заснований на методі "гілок і меж".

**Постановка TSP - задачі.** Перша згадка про TSP - задачу в наукових публікаціях зафіксована на Віденській конференції в доповіді Karl Menger. У термінах теорії графів постановка TSP - задачі будується шляхом зіставлення: міст, які має відвідати комівояжер (у нашому випадку drone), з вершинами графа; маршрути, що з'єднують ці міста та вартість проїзду - з навантаженими дугами.

На виході виходить повноорієнтований асиметричний граф (G) на (n) вершинах без власних петель, заданий матрицею вартостей:

$$A = (a_{ij}), \quad (1)$$

де  $(a_{ij})$  формалізується як:

$$a_{ij} = \infty \forall i = \overline{1, n}, \quad (2)$$

Тоді TSP - задача формулюється як задача пошуку гамільтонового циклу найменшої вартості на орієнтованому графі, який є несиметричною матрицею вартості A. При дослідженні TSP - задачі, були виділені три основні типи задач, кожна з яких має свою особливість.

*Перший тип - асиметрична TSP - задача.* TSP - задача називається асиметричною задачею, якщо дуги розглянутого орієнтованого графа є ребрами. Тоді для кожної пари міст вартість проїзду в один бік та назад не збігаються. *Другий тип – симетрична TSP - задача.* Коли дуги орієнтованого графа представлені ребрами, а для кожної пари міст вартість проїзду в один бік та назад збігаються, то така TSP - задача називається симетричною. Провідні зарубіжні дослідники A. Jonker і T. Volgenan при розробці алгоритмів дійшли висновку, що будь-яку асиметричну TSP - задачу можна перетворити на симетричну TSP - задачу. Таке перетворення дозволяє застосувати, отриманий в результаті експерименту заділ, для вирішення асиметричної TSP - задачі. *Третій тип – метрична TSP - задача.* Якщо уявити симетричну TSP - задачу у вигляді нерівності:

$$a_{ij} \leq a_{ik} + a_{kj} \forall i, j, k = \overline{1, n} (i \neq j, k \neq i, k \neq j), \quad (3)$$

то можна припустити, що в задачах такого класу непрямі шляхи завжди більші за прямі шляхи. При такому затвердженні TSP – задача називається метричною.

### Огляд алгоритмів розв'язання асиметричної TSP - задачі

**Точні алгоритми розв'язання асиметричної TSP - задачі.** Група дослідників Dantzig G., Fulkerson R. та Johnson S. запропонували універсальний "алгоритм - Гоморі" для вирішення TSP - задачі. Трохи пізніше цей алгоритм стали застосовувати для вирішення завдань

цілочисленного програмування, але було відзначено, що час роботи "алгоритму - Гоморі" є довгим. Для усунення цієї проблеми вчені M. Held та R. Karр і, незалежно від них R. Bellman, запропонували ряд кроків щодо усунення цього недоліку. Результатом цієї роботи з'явилася пропозиція – застосувати метод динамічного програмування для вирішення асиметричної TSP – задачі, який надалі отримав назву "алгоритм - Held - Karр". Під час проведення багаторазових експериментальних досліджень in a physical unorganized environment, було встановлено збільшення швидкості цього алгоритму порівняно з "алгоритмом - Гоморі", але виявився недолік, який вимагав більшого обсягу оперативної пам'яті обчислювальних пристроїв.

Після проведення ряду вдалих експериментів, відомі дослідники Little J., Murty K., Sweeney D. та Karel C. запропонували використовувати метод "гілок і меж" для вирішення асиметричної TSP - задачі. Перша вдала модифікація цього методу була продемонстрована науковій громадськості вченими Padberg M. і Rinaldi G. Ця модифікація була схвалена і отримала назву – метод "гілок та відсічень". Далі було проведено ряд експериментів, після яких цей метод став успішно застосовуватися для вирішення TSP - задачі великої розмірності [11-13].

### Наближені алгоритми розв'язання асиметричної TSP - задачі.

Оскільки асиметрична TSP - задача відноситься до класу NP - складних, то це означає, що якщо гіпотеза про нерівність класів P і NP вірна, то на сьогоднішній день не існує точного поліноміального алгоритму для розв'язання TSP - задачі. У зв'язку з цим після проведення низки досліджень стала актуальною пропозиція застосування наближених алгоритмів (у науковій літературі вони відомі як евристичні алгоритми або метаевристичні алгоритми) для вирішення TSP - задачі. Хоча, як показали кінцеві результати, отримані під час проведення низки експериментів, наближені алгоритми не гарантують точного рішення, але вони здатні із задовільною похибкою знайти рішення, досить близьке до точного за прийнятний час (на думку дослідників).

При проведенні огляду та аналізу наукових публікацій у цій галузі було встановлено, що наближені алгоритми вирішення TSP - задачі поділяються на три базові сімейства алгоритмів.:

- перше сімейство – "жадібні алгоритми";
- друге сімейство – "swarm алгоритми";
- третє сімейство – алгоритми, спрямовані на покращення рішення.

У першому сімействі широку популярність серед представників "жадібних алгоритмів" отримав алгоритм "найближчих сусідів". Але при проведенні досліджень було встановлено, що похибка кінцевого результату більшості "жадібних алгоритмів" дуже висока. У розвиток другого сімейства "swarm алгоритмів" великий внесок зробили роботи таких відомих вчених, як M. Dorigo - "мурашиний алгоритм", Hamed Shah-Hosseini - алгоритм "краплі дощу" та T. Sato - "бджолиний алгоритм". У розвиток третього сімейства наближених алгоритмів свій внесок внесли такі відомі вчені, як S. Lin - алгоритм " $\lambda$ -opt", C. Rego - алгоритм "Tabu search" і

В. Kernighan - алгоритм "Lin - Kernighan".

**Алгоритм, заснований на методі "гілок і меж", для вирішення асиметричної TSP - задачі.**

Головна ідея методу "гілок і меж" полягає у поділі безлічі допустимих рішень на підмножини. Потім виконуються подальші скорочення перебором. Ця процедура називається – розгалуженням. З кожною такою підмножиною існує пов'язана оцінка, яка називається – нижня межа при пошуку мінімуму. Ця оцінка забезпечує відсікання підмножин, що не містять оптимального рішення. Ця процедура називається – побудовою меж. Таким чином, за допомогою методу "гілок і меж" дослідник приходять до деревоподібної моделі простору рішень. У нашому випадку такою вихідною множиною є безліч "всіх можливих рухів по маршруту", на якій мінімізується цільовий функціонал, що задає вартість "всіх можливих рухів по маршруту". При проектуванні алгоритмів розробники програмного забезпечення мають передбачити такі кроки: крок № 1 - операція переходу при виконанні чи невиконанні певної умови; крок № 2 - вибудовування меж.

Операція переходу при виконанні або невиконанні певної умови, яка починається з кореня, включає в себе дію пошуку дерева рішень. Тоді можна стверджувати, що корінь дерева рішень складається з множини  $K$  всіх  $(n-1)!$  "можливих рухів по маршруту". Завдання передбачає наявність якоїсь кількості міст  $n$ . Операція переходу виконується за алгоритмом вибору однієї дуги, наприклад  $(m, l)$ , з безлічі дуг, що виходять із кореня. Проектувальники алгоритму на основі методу "гілок і меж" запропонували ідею - розбити вихідну множину "всіх можливих рухів по маршруту" на дві множини:

- перша множина - включає в себе рух по оптимальному маршруту;
- друга множина - не включає в себе рух по оптимальному маршруту.

Для того щоб ця ідея спрацювала, проектувальники розробили особливий алгоритм визначення дуги  $(m, l)$ , яка входить до "оптимального руху по маршруту". Множина  $K$  поділяється на дві множини: множина  $\{m, l\}$ , до якої входять всі "рухи по маршруту" із  $K$ , які містять цю дугу, тобто проходять через неї; множина  $\overline{\{m, l\}}$  - "рухи по маршруту", які не містять цю дугу.

Коли операція переходу при виконанні або невиконанні певної умови організована так, що здійснюється вибір "правильної" дуги, то перехід закінчується після  $n$  ітерацій, рис. 1.

Вибудовування меж. Число 11, яке знаходиться в корені дерева рішень, позначає нижню межу вартості "всіх можливих рухів по маршруту". На наступному етапі алгоритму проектувальники запрограмували вибір дуги для операції переходу. Тоді, множина "всіх можливих рухів по маршруту" поділяється на множину "рухів по маршруту", які містять дугу  $(1, 4)$ , на рис. 1 позначається як  $1 \rightarrow 4$ ; множину "рухів по маршруту", які не містять дугу  $(1, 4)$ , на рис. 1 позначається як  $1 \not\rightarrow 4$ .

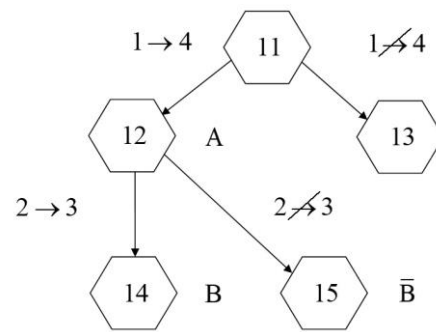


Рис. 1. Фрагмент дерева вибору "правильної" дуги

Далі, згідно з логікою роботи алгоритму, відбувається вибір множини "рухів по маршруту". Вона відноситься до вершини з нижньою межею 12. Тоді цю множину можна розділити на наступні підмножини:

- підмножина №1 - містить у собі дугу  $(2, 3)$ ;
- підмножина №2 - не містить у собі дугу  $(2, 3)$ .

На рис.2 прийняті наступні позначення:

- $B$  – множина, яка є підмножиною "рухів по маршруту" із  $A$ , що проходять через дугу  $(m, l)$ ;
- $\overline{B}$  – множина, яка є підмножиною "рухів по маршруту" із  $A$ , що не проходять через дугу  $(m, l)$ .

З кожною вершиною дерева зв'язується нижня межа вартості будь-якого "руху по маршруту" з цієї множини. Тоді можна стверджувати, що завдання полягає в отриманні якомога точніших нижніх меж.

## Висновки

При проведенні огляду та аналізу наукової літератури було встановлено, що на сьогодні не існує однозначного алгоритму розв'язання асиметричної TSP – задачі, який міг би надати кінцевий результат за прийнятний час з використанням допустимого обсягу оперативної пам'яті обчислювальних пристроїв у реальному часі. І, швидше за все, такого алгоритму немає. Навіть при тому недоліку, що час, який витрачається обчислювальними пристроями для точного розв'язання асиметричної TSP - задачі великий, а для великих розмірностей і зовсім недосяжний, бізнес-потреби вирішення цього завдання на сьогодні залишаються актуальними. Оптимізуючи свої дії, дослідники для досягнення кінцевого результату вирішили вдатися до модифікації рішення асиметричної TSP - задачі. При проведенні експериментальних досліджень у physical unorganized environment Хороші результати при розв'язанні асиметричної TSP - задачі показали наближені алгоритми. Хоча вони мають певну похибку при вирішенні асиметричної TSP - задачі, але, як показали кінцеві результати - вартість знайденого шляху руху близька до вартості оптимального шляху руху, а час, який витрачає обчислювальний пристрій на рішення асиметричної TSP - задачі, є досить прийнятним. Як було встановлено при проведенні серії експериментів, жертвуючи точністю при отриманні кінцевого результату, досить швидко виникає гамільтонів цикл, що в більшості випадків є необхідною і достатньою умовою та задовольняє потребам дослідників. Але,

незважаючи на позитивні кінцеві результати вирішення асиметричної TSP - задачі, отримані за допомогою наближених алгоритмів, при проведенні експериментальних досліджень в physical unorganized environment, дослідники вирішили не відмовлятися від застосування точних алгоритмів розв'язання асиметричної TSP - задачі, навіть якщо це пов'язано з такими недоліками, як час роботи обчислювального

пристрою, так і ресурси об'ємів пам'яті, що витрачаються. Тому, проведений огляд та аналіз наукової літератури та проведені наукові дослідження, підтвердили подальшу необхідність у удосконаленні алгоритму, заснованого на методи "гілок та меж", для вирішення асиметричної TSP - задачі при B2C доставках за допомогою платформи "Swarm-bot" - system у physical unorganized environment.

## СПИСОК ЛІТЕРАТУРИ

1. Koshovyi M. D., Pylypenko O. T., Ilyina I. V., Tokarev V. V. Growing tree method for optimisation of multifactorial experiments, *Radio Electronics, Computer Science, Control*, 2023, № 3, pp. 55–61. Doi: 10.15588/1607-3274-2023-3-6.
2. Koshevoy N., Ilina I., Tokariyev V., Malkova A., Muratov V. Implementation Of The Gravity Search Method For Optimization By Cost Expenses Of Plans For Multifactorial Experiments, *Radioelectronic and Computer Systems*, 2023, №. 1(105), pp. 23-32. Doi: 10.32620/reks.2023.1.02.
3. Кривуля Г.Ф., Токарев В.В., Іліна І.В., Кравець В.С. Взаємодія між «s-bots» однієї «Swarm-bot» system у фізичному неорганізованому середовищі, *Системи управління, навігації та зв'язку*, 2023, №1(71), с.108-111. Doi: 10.26906/SUNZ.
4. Krivoulya G., Koshevoy N., Tokariyev V., Ilina I., Dubinsky D. Solving the Task of Topological Formation Intelligent Mobile «S-bots» for One «Swarm-bot» System, *Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems: (COLINS 2023)*. CEUR Workshop Proceedings, 2023. Kharkiv. Ukraine, pp. 273-282.
5. Krivoulya G., Tokariyev V., Ilina I., Lebediev O., Shcherbak V. Algorithm of Iterations of Distribution of Subtasks Between «S-Bot» in One «Swarm-Bot» System, *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems: (COLINS 2022)*. CEUR Workshop Proceedings, 2022. Gliwice. Poland, pp. 1531-1541.
6. Krivoulya G., Ilina I., Tokariyev V., Shcherbak V. Mathematical Model for Finding Probability of Detecting Victims of Man-Made Disasters Using Distributed Computer System with Reconfigurable Structure and Programmable Logic, *IEEE Int. Scientific-Practical Conf. Problems of Inf., Science and Technology: (PIC S&T)*, 2020. Kharkiv. Ukraine, pp.573 - 576.
7. Serkov A., Kravets V., Yakovenko I., Churyumov G., Tokariyev V., Nannan W. Ultra Wideband Signals in Control Systems of Unmanned Aerial Vehicles, *The 10th IEEE International Conference on Dependable Systems, Services and Technologies: (DESSERT'2019)*, 2019. Leeds. England, 2019, pp.26 - 29.
8. Serkov A., Pustovoitov P., Yakovenko I., Lazurenko B., Churyumov G., Tokariyev V., Nannan W. Ultra wideband technologies in mobile object management systems, *Сучасні інформаційні системи*, 2019, т.3, №2, сс. 22-27.
9. Serkov A., Breslavets V., Tolkachov M., Kravets V. Method of coding information distributed by wireless communication lines under conditions of interference, *Advanced Information Systems*, 2018, vol.2, no.2, pp. 145-148. Doi:10.20998/2522-9052.2018.2.25.
10. Серков О.А., Князев В.В., Лазуренко Б.О., Яковенко І.В., Чурюмов Г.І., Токарев В.В. Надширокосмугові технології в задачах забезпечення електромагнітної сумісності рухомих об'єктів, *Збірник 4 міжн. НТК Проблеми електромагнітної сумісності перспективних бездротових мереж зв'язку: (EMC-2019)*, 2019. Харків. с.55 - 57.
11. T. Gao, X. Bai. Bayesian. Optimization-based Three-dimensional, Time-varying Environment Monitoring using an UAV, *IEEE Journal of Intelligent & Robotic Systems*, 2022, vol.105, no.4, pp.219 - 235. Doi:10.1007/s10846-022-01709-x.
12. Han, J. Jiang, C. Yu. Distributed Fault Estimation and Fixed-Time Fault-Tolerant Formation Control for Multi-UAVs subject to Sensor Faults, *IEEE J. of Intelligent & Ro. Syst.*, 2022, vol.104, no.4, pp.310 - 325. Doi:10.1007/s10846-022-01698-x.
13. K. Muhammad, A. Ullah, J. Lloret. Deep Learning for Safe Autonomous Driving: Current Challenges and Future Directions, *IEEE Trans. on Intelligent Transportation Systems*, 2021, vol.22, no.7, pp.4316 - 4336. Doi:10.1109/TITS.2020.3032227.
14. L. Zhao, Y. Song, C. Zhang. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction, *IEEE Transactions on Intelligent Transportation Systems*, 2020, vol.21, no.9, pp.3848 - 3858. Doi:10.1109/TITS.2019.2935152.
15. E. Seraj, A. Silva, M. Gombolay. Multi-UAV planning for cooperative wildfire coverage and tracking with quality-of-service guarantees, *Autonomous Agents and Multi-Agent Systems*, article number.39, 2022, Springer. Doi:10.1007/s10458-022-09566-6.

Received (Надійшла) 15.09.2023

Accepted for publication (Прийнята до друку) 29.11.2023

**Interaction between «s-bots» one «Swarm-bot» system in a physical unorganized environment**

V. Tokariyev, I. Ilina, I. Shevchenko, I. Hrytsenko

**Abstract.** Today, in the field of service provision, after an analysis of the supply and demand market, research is being conducted to determine the possibility of using the "Swarm-bot" - system platform for delivering orders to end users. In the role of "s-bots", which are part of one "Swarm-bot" - system, drones are actively used. Delivery of orders and correspondence using the "Swarm-bot" - system platform implies the presence of a specialized container placed on the drone body. Research is being conducted to see if drones can replace the email messages we're all used to. When implementing the "Swarm-bot" - system platform into real operation, a person participates in the delivery process as an operator, and drones move along a pre-programmed route. Today, the most optimal way to solve the problem of reducing the order delivery distance is to solve the asymmetric TSP problem (travelling salesman problem). This article is devoted to exploring the possibility of applying an algorithm based on the "branch-and-bound" method to solve an asymmetric TSP problem in B2C deliveries using the "Swarm-bot" - system platform in a physical unorganized environment.

**Keywords:** "Swarm-bot" - system, "s-bots", embedded systems, business-to-business, business-to-consume, drones, communication, branch-and-bound method, asymmetric TSP - problem.