

**Леснікова І. Ю.**, кандидат технічних наук, доцент, доцент кафедри транспортних технологій та міжнародної логістики Університету митної справи та фінансів  
**Халіпова Н. В.**, кандидат технічних наук, доцент, доцент кафедри транспортних технологій та міжнародної логістики Університету митної справи та фінансів  
**Сазонець О. М.**, доктор економічних наук, професор, професор кафедри транспортних технологій та міжнародної логістики Університету митної справи та фінансів  
**Рудянова Т. М.**, кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук та інженерії програмного забезпечення Університету митної справи та фінансів

## **ОПТИМІЗАЦІЯ МАРШРУТІВ ВАНТАЖНИХ АВТОМОБІЛЬНИХ ПЕРЕВЕЗЕНЬ НА ОСНОВІ МОДИФІКОВАНОГО МУРАШИНОГО АЛГОРИТМУ**

*Присвячено виконанню складних комплексних завдань оптимізації вантажних перевезень, тобто пошуку і визначенню найбільш відповідного рішення для оптимізації (знаходження мінімуму або максимуму) цільової функції (ціни, точності, часу, відстані тощо) з дискретної множини можливих рішень. У статті обґрунтовано напрями модифікації мурашиного алгоритму, сформовано алгоритм маршрутизації виконання перевізного процесу на основі дослідження стану та визначення теоретичної пропускної спроможності окремих ділянок на проєктованих маршрутах. Проаналізовано ефективність класичного та модифікованого алгоритмів на прикладі проєктування оптимальних маршрутів від пункту відправлення до пункту призначення і для віднаходження максимального маршруту по фрахту.*

Ключові слова: *оптимізації вантажних перевезень; модифікація мурашиного алгоритму.*

© І. Ю. Леснікова, Н. В. Халіпова, О. М. Сазонець, Т. М. Рудянова, 2021

---

*The article is devoted to solving complex problems of optimization of freight transportation, namely search and determination of the most suitable solution for optimizing (finding a minimum or maximum) the target function (price, accuracy, time, distance, etc.) from a discrete set of possible solutions. The article substantiates the direction of modification of the ant algorithm, an algorithm for routing the implementation of the transport process is formed on the basis of the study of the state and determination of the theoretical throughput of individual sites on the projected routes. As well as an analysis of the effectiveness of classical and modified algorithms on the example of designing optimal routes from the point of departure to the destination and to find the maximum freight route.*

*Among the modern methods for solving the problem of a salesman, an ant algorithm is chosen, since its physical basis – the behavior of ants when searching for food – is the closest to formulating this problem, which determines the optimal route for transportation and its profitability for the company.*

*A mathematical model was constructed, an algorithm was proposed and a flowchart was developed to determine the optimal transportation route based on a modified ant algorithm.*

*A program was developed in the Delphi environment to determine the optimal route by the minimum distance criterion based on the proposed algorithm of the salesman problem. The task of determining the shortest route, which includes regional centers of Ukraine, is connected and a graph of this route is constructed. The optimal route was developed: Kyiv – Zhytomyr – Lutsk – Chernivtsi – Uzhgorod – Lviv – Ivano-Frankivsk – Khmelnytsky – Cherkasy – Zaporizhzhia – Kharkiv – Chernihiv – Nikolaev – Kirovograd – Dnipro – Poltava – Sumy – Vinnytsia – Ternopil – Rivne – Kherson – Odessa – Kyiv, with a total distance of 7,930 km and a freight of 190,710 UAH.*

*A program has been developed in the Delphi environment to determine the optimal route according to the maximum freight criterion of the carrier based on the proposed algorithm of the salesman problem. This task is connected for a network of 22 cities – regional centers of Ukraine, and a graph of the received route was constructed. The optimal route was developed: Kyiv – Mykolaiv – Chernivtsi – Zhytomyr – Khmelnytsky – Dnipro – Lutsk – Kirovohrad – Rivne – Kherson – Sumy – Ternopil – Chernihiv – Vinnytsia – Poltava – Odesa – Lviv – Kharkiv – Uzhgorod – Zaporizhzhia – Ivano-Frankivsk – Cherkasy – Kyiv, with a total freight for the carrier of UAH 357,800 and a distance of UAH 16,170 km.*

*Justification of optimal routes is quite an important task for the effective operation of logistics companies, since freight transportation is extremely important for the economy of the country and the company and is relevant in the present.*

*According to the results of calculations, technical solutions have been developed to improve the efficiency of the logistics company on the basis of a modified ant algorithm.*

*Key words: freight optimization; modification of the ant algorithm.*

---

**Постановка проблеми.** Найпоширенішими проблемами сьогодення виступають економічна криза та військова агресія сусідньої держави, що суттєво позначилося на обсягах перевезень вантажів Україною, а знецінення національної валюти і стрімке зростання цін на паливо поставило під сумнів можливість отримання прибутків підприємств-перевізників та їх доцільність перебування на ринку транспортних послуг. За таких умов порушено питання ретроспективного аналізу ситуації на ринку транспортних послуг з окресленням наявних та прогнозу майбутніх тенденцій. Утім передбачається збереження тенденції останнього часу щодо збільшення частки надання логістичних послуг, незважаючи на дані перешкоди.

**Аналіз останніх досліджень і публікацій.** Актуальність потокових завдань полягає в тому, що економічний розвиток будь-якої країни обумовлений наявністю мережі доріг. Транспортна структура, що має тенденцію до розширення, призводить до перевантаженості мережі доріг, аварійних ситуацій у результаті заторів.

Нині задачі маршрутизації є ключовими у сфері логістики. Існує велика кількість методів розв'язування задач дискретного типу з метою оптимізації логістичних процесів, проте вони не дають чіткого розв'язання наявних проблем.

Проблемам транспортної логістики присвячено праці багатьох як українських, так і зарубіжних учених [1–4]. Автори детально аналізують проблеми організації та здійснення процесу перевезень вантажів з погляду сучасної логістики. Значна увага приділяється перевезенню у межах міста як вантажів, так і пасажирів. Так, у праці [5] досліджено питання щодо організації та функціонування мережі міста, зокрема розглянуто проблему заторів та шляхів її подолання.

Задача проектування маршрутів транспортних засобів під час перевезення вантажів належить до класу задач просторово-часового синтезу та є однією з найскладніших для розв'язання.

Ураховуючи це, транспортні задачі є досить актуальним питанням у формуванні оптимальної логістичної системи доставки вантажів в умовах міст.

Різноманітність критеріїв та обмежень у застосуванні транспортної задачі до різних практичних сфер використання призвели до появи великої кількості задач, які відображали процес розвезення вантажу. Проте науковою проблемою під час транспортування вантажів є відсутність діючих моделей чи методик оптимізації перевезень, які б ураховували обмеження пропускної спроможності доріг [7]. Під час розгляду транспортних задач пропонується оцінювати перевезення як цілісну систему, а не проводити ранжування за критеріями [6].

Удосконалення методів на основі класичного мурашиного алгоритму,

---

модифікація самого алгоритму дасть змогу отримувати точніші розв'язки багатокomпонентних задач у різних сферах виробничої діяльності. Але ж у розробці раціональних автомобільних маршрутів вантажних перевезень недостатньо уваги приділено врахуванню якісної характеристики доріг. Урахування стану ділянок доріг та їх пропускної спроможності необхідне для визначення логістичного маршруту, щоб уникнути руху ділянками з ускладненим або частково неможливим проїздом [8].

**Мета статті** – виконання складних комплексних завдань оптимізації вантажних перевезень: пошук і визначення найбільш відповідного рішення для оптимізації (знаходження мінімуму або максимуму) цільової функції (ціни, точності, часу, відстані тощо) з дискретної множини можливих рішень.

Алгоритм оптимізації мурашиної колонії може бути успішно застосовано для виконання складних комплексних завдань оптимізації. Тому в даній статті розглядатиметься задача про призначення на основі модифікованого мурашиного алгоритму (МА). Отже, результатом пошуку оптимального розв'язку задачі про призначення стало використання природних поведінкових явищ, а саме: поведінки мурашиної колонії.

Для цього в статті виконано такі завдання:

- обґрунтування напрямів модифікації мурашиного алгоритму;
- формування алгоритму маршрутизації виконання перевізного процесу на основі дослідження стану та визначення теоретичної пропускної спроможності окремих ділянок на проєктованих маршрутах;
- порівняння ефективності класичного та модифікованого алгоритмів на прикладі проєктування оптимальних маршрутів від пункту відправлення до пункту призначення і для віднаходження максимального маршруту по фрахту перевізника.

**Виклад основного матеріалу.** Сучасні методи розв'язування задач, що ґрунтуються на класичному алгоритмі мурашиної колонії, досить конкурентоздатні порівняно з іншими евристичними методами, а для деяких задач демонструють навіть кращі результати розв'язку [10–14]. Розв'язання задачі транспортування на основі алгоритму мурашиних колоній для мультимодального перевезення та розширеної задачі комівояжера наведено відповідно в [10, 11].

Загальний алгоритм розв'язку таких задач досить простий: по-перше, звести задачу до задачі пошуку найкоротшого шляху на деякому графі; по-друге, визначити механізми оновлення феромону. Вплив на оновлення феромону можна задати багатьма факторами, наприклад: собівартість перевезення, довжина маршруту чи ціна пройденого маршруту тощо. І третій крок – визначити евристичні правила вибору маршруту. В статті особливістю вибору шляху транспортування є те, що враховано коефіцієнт стану дороги,

---

вплив якого покращує пошук оптимального маршруту. Якість отриманих результатів насамперед залежить від вибору і налаштування параметрів під час визначення ймовірності переходу в той чи інший пункт за допомогою ймовірнісно-пропорційного правила на основі відкладеної кількості феромонів. При цьому не слід забувати про вагомий вплив параметра випаровування феромону з кожною ітерацією.

Перехід мурахи з міста  $i$  в місто  $j$  на ітерації  $t$  алгоритму залежить від трьох складових: табу-списка, видимості та віртуального сліду феромону.

Табу-список – це перелік міст, які вже відвідані мурахою і заходить в які ще раз заборонено. Табу-список збільшується зі здійсненням маршруту та заповнюється нулями на початку кожної ітерації алгоритму. Позначимо через  $J^k$  список міст, які ще потрібно відвідати мурасі  $k$ , що перебуває у місті  $i$ . Зрозуміло, що об'єднання цих списків дає множину всіх міст з маршруту комівояжера.

Видимість – це величина обернена до відстані:

$$n_{ij} = 1/D_{ij},$$

де  $D_{ij}$  – відстань між містами  $i$  та  $j$ .

Видимість базується тільки на локальній інформації та являє собою “евристичну бажаність” вибору міста  $j$  під час перебування у місті  $i$ . Чим ближче міста  $i$  та  $j$ , тим більше бажання відвідати їх.

Віртуальний слід феромону на ребрі  $(i, j)$  являє собою “бажаність, підкріплену досвідом” переходу в місто  $i$  з міста  $j$ . Інформація, яку несе слід феромону, змінюється під час оптимізації та відображає набутий мурахами досвід. Кількість віртуального феромону на ребрі  $(i-j)$  на  $t$ -й ітерації алгоритму позначимо як  $\tau_{ij}(t)$ .

Показник імовірності того, що мураха обере певний перший пункт на своєму маршруті, розраховується за допомогою ймовірнісно-пропорційного правила, що виражається такою формулою:

$$\begin{cases} P_{ij,k}(t) = \frac{[\tau_{ij}(t)]^\alpha [\mu_{ij}(t)]^\beta}{\sum_{l \in J_{i,k}} [\tau_{il}(t)]^\alpha [\mu_{il}(t)]^\beta} \cdot 100\%, J \in J_{i,k}, \\ P_{ij,k}(t) = 0, j \notin J_{i,k} \end{cases} \quad (1)$$

де  $\alpha, \beta$  – параметри, що задають ваги сліду феромону та видимості; при  $\alpha = 0$  спрацьовує жадібний алгоритм, тобто мураха рухається щоразу до найближчого пункту, ігноруючи досвід попередників;

при  $\beta = 0$  мураха керується лише досвідом попередників, повністю ігноруючи відстань між пунктами на маршруті. Обидві умови є досить “суворими” і виключають коректну роботу алгоритму та повноцінний пошук оптимального маршруту. Виходячи з цього, за умови прийняття одного з параметрів  $\alpha$  чи  $\beta$  рівними нулю, сенсу в алгоритмі мурашиних колоній немає.

Формула (1) є класичним варіантом формули МА для визначення ймовірності переходу. Для врахування стану доріг на кожній ділянці дороги між пунктами пропонуємо вдосконалити формулу (1) шляхом введення нового коефіцієнта  $\varepsilon_{ij}$ :

$$\varepsilon_{ij} = \omega_{ij} \cdot Pp_{ij}, \quad (2)$$

де  $\omega_{ij}$  – стан ділянки дороги, який оцінюється від 0,5 до 1 з кроком 0,25. Значення показника 0,5 показує, що стан незадовільний; 0,75 – частково задовільний, є вибоїни/бугри; 1 – дорожнє покриття у задовільному стані;

$Pp_{ij}$  – пропускна спроможність ділянки маршруту, авто/год.

У табл. 1 подано оцінки пропускної здатності ділянки доріг, отримані за результатами аналізу даних про частки транспортних засобів у потоці [15].

Таблиця 1

#### Пропускна спроможність доріг

№ з/п	$Pp_{ij}$ , авто/год	Коефіцієнт
1	0–300	0,15
2	300–600	0,3
3	600–900	0,45
4	900–1200	0,6
5	1200–1500	0,75
6	1500 і більше	1

Тоді формула (1) набуває вигляду:

$$\begin{cases} P_{ij,k}(t) = \frac{\varepsilon_{ij}(t) [\tau_{ij}(t)]^\alpha [\mu_{ij}(t)]^\beta}{\sum_{l \in J_{i,k}} \varepsilon_{ij}(t) [\tau_{ij}(t)]^\alpha [\mu_{ij}(t)]^\beta} \cdot 100\%, J \in J_{i,k} \\ P_{ij,k}(t) = 0, j \notin J_{i,k} \end{cases} \quad (3)$$

Під час роботи алгоритму, керуючись основними показниками, ймовірно-пропорційне правило не змінюється, проте у кожної мурахи ймовірність прямування кожним пунктом буде різна.

---

Далі необхідно визначити, яка кількість феромону відкладатиметься мурахою після проходження кожного ребра на маршруті. Класична формула, що визначає кількість феромону після кожної ітерації має вигляд:

$$\Delta \tau_{ij,k}(t) = \begin{cases} \frac{Q}{L_k(t)}, & \text{якщо } (i, j) \in T_k(t) \\ 0, & \text{якщо } (i, j) \notin T_k(t) \end{cases}, \quad (4)$$

де  $T_k(t)$  – маршрут, що пройдений мурахою  $k$  на ітерації  $t$ ;

$L_k(t)$  – довжина цього маршруту;

$Q$  – регульований параметр, значення якого обирається одного порядку з довжиною оптимального маршруту.

Проте, як зазначено вище, якщо феромон не буде випаровуватись, то ймовірне існування хибного маршруту. Водночас необхідно забезпечити, щоб у разі “незадоволення” маршрутом мурахи обирали новий, а значення феромону на маршруті мало б випаровуватись, щоб колонії мурах “забували” даний шлях. Параметр випаровування феромону позначається через  $\rho$ . При цьому діапазон значень  $\rho$  перебуває у межах  $(0;1)$ . Тоді правило оновлення феромону набуває вигляду:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta \tau_{ij}(t), \quad (5)$$

де  $\Delta \tau_{ij}(t) = \sum_{k=1}^m \Delta \tau_{ij,k}(t)$ ;

$m$  – кількість мурах у колонії.

Щодо параметра  $\rho$  слід зазначити:

1. Якщо  $\rho = 0$ , тоді колонія враховуватиме всі маршрути, в тому числі зовсім неефективні. За такого варіанта досягти оптимального маршруту буде дуже складно.

2. Якщо  $\rho = 1$ , то в такому випадку мураха пам’ятає досвід тільки свого попередника, а досвід інших мурах повністю ігнорується. Знову ж таки існує велика ймовірність того, що мурахи йтимуть заздалегідь хибним шляхом.

Таким чином, урахувавши всі критерії, економіко-математична модель виглядатиме так:

$$\sum_{i=1}^n \sum_{j=1}^n L_{ij} \rightarrow \min$$

$$\begin{cases} 0,5 \leq \omega_{ij} \leq 1 \\ 0,5 \leq \alpha \leq 1 \\ 0,5 \leq \beta \leq 1 \\ 0 \leq \rho \leq 1 \\ 400 \leq Q \leq 500 \\ 0,15 \leq Pp_{ij} \leq 1 \end{cases} \quad (6)$$

На початку розрахунку кількість феромону приймається рівною невеликому додатному числу  $\tau_0$ . Загальна кількість мурах залишається незмінною протягом виконання всього алгоритму. Необхідно також зазначити, що треба обирати достатню (багаточисленну) кількість мурах. Адже коли мурах досить мало, наприклад до 10 штук, то виникає небезпека, що буде втрачена сукупність пошуку через швидке випаровування феромону.

Після завершення маршруту кожна мураха  $k$  відкладає на ребро  $(i-j)$  таку кількість феромону:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)}, & \text{якщо } (i,j) \in T^k(t) \\ 0, & \text{якщо } (i,j) \notin T^k(t) \end{cases} \quad (7)$$

де  $T^k(t)$  – маршрут, зроблений мурахою  $k$  на ітерації  $t$ ;

$L^k(t)$  – довжина цього маршруту;

$Q$  – регульований параметр, значення якого обирають одного порядку з довжиною оптимального маршруту.

Для забезпечення можливості експлуатації простору рішень потрібно забезпечити випаровування феромону – зменшення кількості відкладеного на попередніх ітераціях феромону. Інтенсивність випаровування феромону задається за допомогою коефіцієнта випаровування  $\rho \in [0, 1]$ . Кінцеве правило оновлення феромону, яке стосується всіх ребер, набуває такого вигляду:

$$\tau_{ij}(t+1) \leftarrow (1-p)\tau_{ij}(t) + \Delta\tau_{ij}(t), \quad (8)$$

де  $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$ ;

$m$  – кількість мурах у колонії.

На початку оптимізації кількість феромону на ребрах приймається за малу додатну константу  $\tau_0$ . Загальна кількість мурах у колонії приймається



постійною на весь час розв'язання задачі. Забагато мурах призводить до швидкого підсилення субоптимальних маршрутів. Коли ж мурах замало, виникає небезпека втрати кооперативної поведінки через швидке випаровування феромону. Зазвичай кількість мурах приймають рівною кількості міст – кожна мураха починає маршрут з окремого міста [16].

Модифікація даного алгоритму буде виражена у його мінливості впродовж усього маршруту. Тобто врахуємо рельєфність місцевості, виїзні та невиїзні регіони, сезонність, тарифи замовників та подачу транспорту. Для цього розробимо граф на основі карти України з регіонами та вартостями поїздок.

Керуючись алгоритмом розв'язування задачі, можна застосувати алгоритм мурашиних колоній у реальних умовах.

Довжину ділянок доріг України між обласними центрами з урахуванням якості доріг та об'їзних маршрутів зображено на рис. 1.

	Київ	Чернігів	Суми	Полтава	Харків	Дніпро	Запоріжжя	Херсон	Миколаїв	Одеса	Кіровоград	Черкаси	Житомир	Хмельницький	Вінниця	Рівне	Луцьк	Чернівці	Івано-Франківськ	Тернопіль	Львів	Ужгород
Київ	-	150	340	345	500	480	560	480	480	320	200	140	330	270	330	400	530	670	490	540	820	
Чернігів	150	-	320	410	510	540	620	690	620	610	430	300	280	470	410	470	550	680	820	630	700	950
Суми	340	320	-	220	200	360	450	800	870	1000	430	330	480	660	610	660	740	870	1010	820	880	1150
Полтава	345	410	220	-	150	200	270	620	700	830	250	240	480	660	610	660	740	870	1010	830	880	1150
Харків	500	510	200	150	-	220	300	640	720	850	390	380	620	800	750	805	880	1010	1150	960	1020	1300
Дніпро	480	540	360	200	220	-	90	450	520	650	250	300	620	700	570	805	880	890	1150	970	1020	1300
Запоріжжя	560	620	450	270	300	90	-	360	430	560	310	350	700	760	640	890	960	850	1230	1050	1100	1370
Херсон	560	690	800	620	640	450	360	-	80	210	250	380	580	630	500	780	850	810	870	740	870	1140
Миколаїв	480	620	870	700	720	520	430	80	-	130	180	320	510	560	430	710	780	750	800	670	800	1070
Одеса	480	610	1000	830	850	650	560	210	130	-	320	450	500	550	430	700	770	740	800	660	790	1060
Кіровоград	320	430	430	250	390	250	310	250	180	320	-	130	400	450	320	600	670	635	700	560	700	920
Черкаси	200	300	330	240	380	300	350	380	320	450	130	-	330	510	360	520	590	640	700	560	730	1000
Житомир	140	280	480	480	620	620	700	580	510	500	400	330	-	230	130	190	280	370	430	350	400	670
Хмельницький	330	470	660	660	800	760	760	630	560	550	450	510	230	-	120	210	260	200	230	110	240	500
Вінниця	270	410	610	610	750	570	640	500	430	430	320	360	130	120	-	320	390	340	370	240	360	590
Рівне	330	470	660	660	805	805	890	780	710	700	600	520	190	210	320	-	80	330	300	160	210	480
Луцьк	400	550	740	740	880	880	960	850	780	770	670	590	280	260	390	80	-	340	270	170	150	420
Чернівці	820	960	870	870	890	890	850	810	750	740	635	640	370	200	340	330	340	-	140	170	270	430
Івано-Франківськ	670	820	1010	1010	1150	1150	1230	870	800	800	700	700	430	230	370	300	270	140	-	140	140	300
Тернопіль	490	630	820	830	970	970	1050	740	670	660	560	560	350	110	240	160	170	170	140	-	130	390
Львів	540	700	880	880	1020	1020	1100	870	800	790	700	730	400	240	360	210	150	270	140	130	-	270
Ужгород	820	950	1150	1150	1300	1300	1370	1070	1070	1060	920	1000	670	500	590	480	420	430	300	390	270	-

Рис. 1. Відстані між обласними центрами України, крім Донецька, Луганська та АР Крим (км)

Алгоритм математичної моделі побудовано з урахуванням стану дорожнього покриття, переїзду транспортного засобу на завантаження, про-

---

стоїв, відстані перевезення, фрахту перевізника. Типом вантажу (окрім сипучих, їх виключено), режимом перевезення (приймаємо, що наразі літній період), кількістю вантажу (вважатимемо, що вантаж займає увесь кузов, 20 т, 86 м<sup>3</sup>) нехтуємо.

Алгоритм розв'язування практичної задачі на основі МА:

1. Визначаємо вид продукції, яка перевозитиметься відповідно від пункту відправлення до пункту призначення й у зворотному напрямку, якщо він є.

2. У транспортному засобі (ТЗ), яким буде транспортуватись продукція, перевіряємо за критерієм забрудненості кузова, чи можна після перевезення у прямому напрямі у той самий автомобіль завантажити інший товар.

3. Складаємо за допомогою карти всі можливі маршрути доставки як у прямому, так і у зворотному напрямку, якщо він є.

4. Фіксуємо відстань  $L_{ij}$  між кожною парою пунктів. Визначаємо точку відправлення у прямому та зворотному напрямках. Визначаємо можливий переїзд у точку відправлення чи подачу ТЗ.

5. Визначаємо пропускну спроможність кожної ділянки дороги.

6. За формулою (2) визначаємо коефіцієнт, значення якого залежить від безпосереднього стану доріг та пропускну здатності.

7. Задаємо рівень феромонів  $\tau_{ij}$  на кожному ребрі. Даний показник є малим додатним значенням для того, щоб імовірність переходу на наступний пункт не була нульовою. Дана умова обумовлена формулою (3).

8. За формулою (3) визначаємо ймовірність  $P_{ij}$  переходу  $k$ -ї мурахи в наступний пункт на маршруті, задаємо показники  $\alpha$  та  $\beta$ .

9. Визначаємо вартість найкращого шляху, вартість поточного шляху і вартість найгіршого шляху. Задаємо параметр випарювання феромону  $\rho$ .

10. Оновлюємо феромони за формулою (5) після кожної ітерації.

11. Переходимо до наступного транспортного засобу і знову переходимо до пункту 8.

12. Після прогону всіх транспортних засобів визначаємо найкращий маршрут, його довжину та вартість.

На основі алгоритму побудуємо блок-схему (рис. 2).

Код програми для розрахунку задачі комівояжера мовою програмування Delphi наведено у додатку. Використовуємо дані про відстані між пунктами транспортної мережі (рис. 1), щоб сформувати матрицю для розрахунків у середовищі Delphi (рис. 3). Пункти мережі (з 1 до 22) відповідають обласним центрам, а саме: 1 – Київ; 2 – Чернігів; 3 – Суми; 4 – Полтава; 5 – Харків; 6 – Дніпро; 7 – Запоріжжя; 8 – Херсон; 9 – Миколаїв; 10 – Одеса; 11 – Кіровоград; 12 – Черкаси; 13 – Житомир; 14 – Хмельницький; 15 – Вінниця; 16 – Рівне; 17 – Луцьк; 18 – Чернівці; 19 – Івано-Франківськ; 20 – Тернопіль; 21 – Львів; 22 – Ужгород.

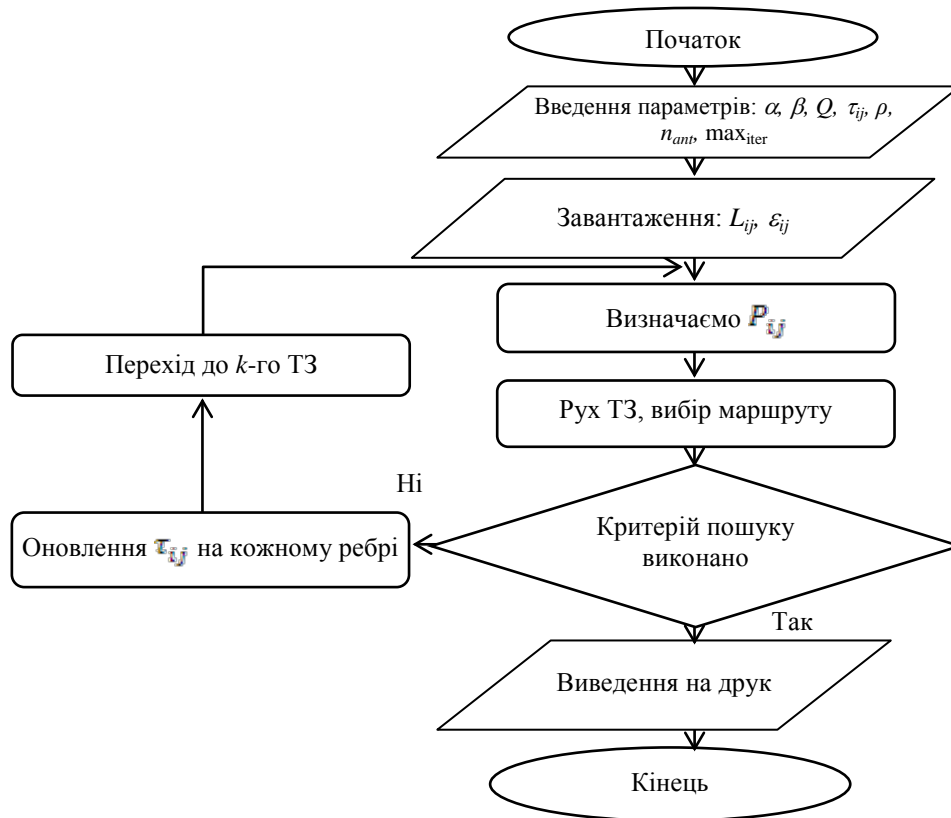


Рис. 2. Блок-схема розв'язку задачі на основі алгоритму мурашиних колоній

Після ітерацій програма показує результат (рис. 4), що прокладає мінімальний за відстанню маршрут.

Оптимальний маршрут за критерієм мінімальної відстані послідовно включає пункти  $1 \rightarrow 13 \rightarrow 17 \rightarrow 18 \rightarrow 22 \rightarrow 21 \rightarrow 19 \rightarrow 14 \rightarrow 12 \rightarrow 7 \rightarrow 5 \rightarrow 2 \rightarrow 9 \rightarrow 11 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 15 \rightarrow 20 \rightarrow 16 \rightarrow 8 \rightarrow 10 \rightarrow 1$ .

Додаємо останнім пунктом точку 1, оскільки авто має повернутись у початок. Також до загальної мінімальної відстані треба додати відстань повернення з пункту 10 у пункт 1.

Отже, ТЗ треба проїхати маршрутом “Київ – Житомир – Луцьк – Чернівці – Ужгород – Львів – Івано-Франківськ – Хмельницький – Черкаси – Запоріжжя – Харків – Чернігів – Миколаїв – Кіровоград – Дніпро – Полтава – Суми – Вінниця – Тернопіль – Рівне – Херсон – Одеса – Київ” із загальною відстанню 7930 км (7450 + 480), що є найменшою відстанню. Маршрут відображено на карті (рис. 5).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	.	150	340	345	500	480	560	480	480	320	200	140	330	270	330	400	530	670	490	540	820	
2	150	.	320	410	510	540	620	690	620	610	430	300	280	470	410	550	680	820	630	700	950	
3	340	320	.	220	200	360	450	800	870	1000	430	330	480	660	610	740	870	1010	820	880	1050	
4	345	410	220	.	150	200	270	620	700	830	250	240	480	660	610	740	870	1010	830	880	1050	
5	500	510	200	150	.	220	300	640	720	850	390	380	620	800	750	805	880	1010	1150	960	1020	1300
6	480	540	360	200	220	.	90	450	520	650	250	300	620	700	570	805	880	1150	970	1020	1300	
7	560	620	450	270	300	90	.	360	430	560	310	350	700	760	640	890	960	1230	1050	1100	1370	
8	560	690	800	700	640	450	360	.	80	210	250	380	580	630	500	780	850	810	870	740	870	1140
9	480	620	870	830	720	520	430	80	.	130	180	320	510	560	430	710	780	750	800	670	800	1070
10	480	610	1000	250	850	650	560	210	130	.	320	450	500	550	430	700	770	740	800	660	790	1060
11	320	430	430	240	390	250	310	250	180	320	.	130	400	450	320	600	670	635	700	560	700	920
12	200	300	330	480	380	300	350	380	320	450	130	.	330	510	360	520	590	640	700	560	730	1000
13	140	280	480	480	620	620	700	580	510	500	400	330	.	230	130	190	280	370	430	350	400	670
14	330	470	660	660	800	700	760	630	560	550	450	510	230	.	120	210	260	200	230	110	240	500
15	270	410	610	610	750	570	640	500	430	430	320	360	130	120	.	320	390	340	370	240	360	590
16	330	470	660	660	805	805	890	780	710	700	600	520	190	210	320	.	80	330	300	160	210	480
17	400	560	740	740	880	880	960	850	780	770	670	590	280	260	390	80	.	340	270	170	150	420
18	530	680	870	870	1010	890	850	810	750	740	635	640	370	200	340	330	340	.	140	170	270	430
19	670	820	1010	1010	1150	1150	1230	870	800	800	700	700	430	230	370	300	270	140	.	140	140	300
20	490	630	820	830	960	970	1050	740	670	660	560	560	350	110	240	160	170	170	140	.	130	390
21	540	700	880	880	1020	1020	1100	870	800	790	700	730	400	240	360	210	150	270	140	130	.	270
22	820	950	1050	1050	1300	1300	1370	1140	1070	1060	920	1000	670	500	590	480	420	430	300	390	270	.

Рис. 3. Загальний вигляд програми у середовищі Delphi з відстанями

**Задача комівозера методом Монте-Карло**

[Знаходження розв'язку задачі комівозера методом Монте-Карло](#)

Знайти оптимальний маршрут

Очистити таблицю

Сохранить таблицу

Загрузить таблицу

Кількість населених пунктів: 22

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	.	150	340	345	500	480	560	560	480	480	320	200	140	330	270	330	400	530	670	490	540	820
2	150	.	320	410	510	540	620	630	620	610	430	300	280	470	410	470	550	660	820	630	700	960
3	340	320	.	220	200	360	450	800	870	1000	430	330	480	660	610	660	740	870	1010	820	880	1150
4	345	410	220	.	150	200	270	620	700	830	250	240	480	660	610	660	740	870	1010	830	880	1150
5	500	510	200	150	.	220	300	640	720	850	390	380	620	800	750	805	880	1010	1150	960	1020	1300
6	480	540	360	200	220	.	90	450	520	650	250	300	620	700	570	805	880	890	1150	970	1020	1300
7	560	620	450	270	300	90	.	360	430	580	310	360	700	760	640	890	960	850	1230	1050	1100	1370
8	560	690	800	620	640	450	360	.	80	210	250	380	580	630	500	780	850	810	870	740	870	1140
9	480	620	870	700	720	520	430	80	.	130	180	320	510	560	430	710	780	750	800	670	800	1070
10	480	610	1000	830	850	650	560	210	130	.	320	450	500	550	430	700	770	740	800	660	790	1060
11	320	430	430	250	390	250	310	250	180	320	.	130	400	450	320	600	670	630	700	560	700	920
12	200	300	330	240	380	300	350	380	320	450	130	.	330	510	360	520	590	640	700	560	730	1000
13	140	280	480	480	620	620	700	580	510	500	400	330	.	230	130	190	280	370	430	350	400	670
14	330	470	660	660	800	700	760	630	560	550	450	510	230	.	120	210	260	200	230	110	240	500
15	270	410	610	610	750	570	640	500	430	430	320	360	130	120	.	320	390	340	370	240	360	590
16	330	470	660	660	805	805	890	780	710	700	600	520	190	210	320	.	80	330	300	160	210	480
17	400	550	740	740	880	880	960	850	780	770	670	590	280	260	390	80	.	340	270	170	150	420
18	530	680	870	870	1010	890	850	810	750	740	630	640	370	200	340	330	340	.	140	170	270	430
19	670	820	1010	1010	1150	1150	1230	870	800	800	700	430	230	230	370	300	270	140	.	140	140	300
20	490	630	820	830	960	970	1050	740	670	660	560	560	350	110	240	160	170	170	140	.	130	390
21	540	700	880	880	1020	1020	1100	870	800	790	700	730	400	240	360	210	150	270	140	130	.	270
22	820	950	1150	1150	1300	1300	1370	1140	1070	1060	920	1000	670	500	590	480	420	430	300	390	270	.

Оптимальний маршрут: 1 -> 13 -> 17 -> 18 -> 22 -> 21 -> 19 -> 14 -> 12 -> 7 -> 5 -> 2 -> 9 -> 11 -> 6 -> 4 -> 3 -> 15 -> 20 -> 16 -> 8 -> 10

Загальна відстань: 7450

Рис. 4. Програмний розв'язок задачі визначення оптимального маршруту за критерієм мінімальної відстані

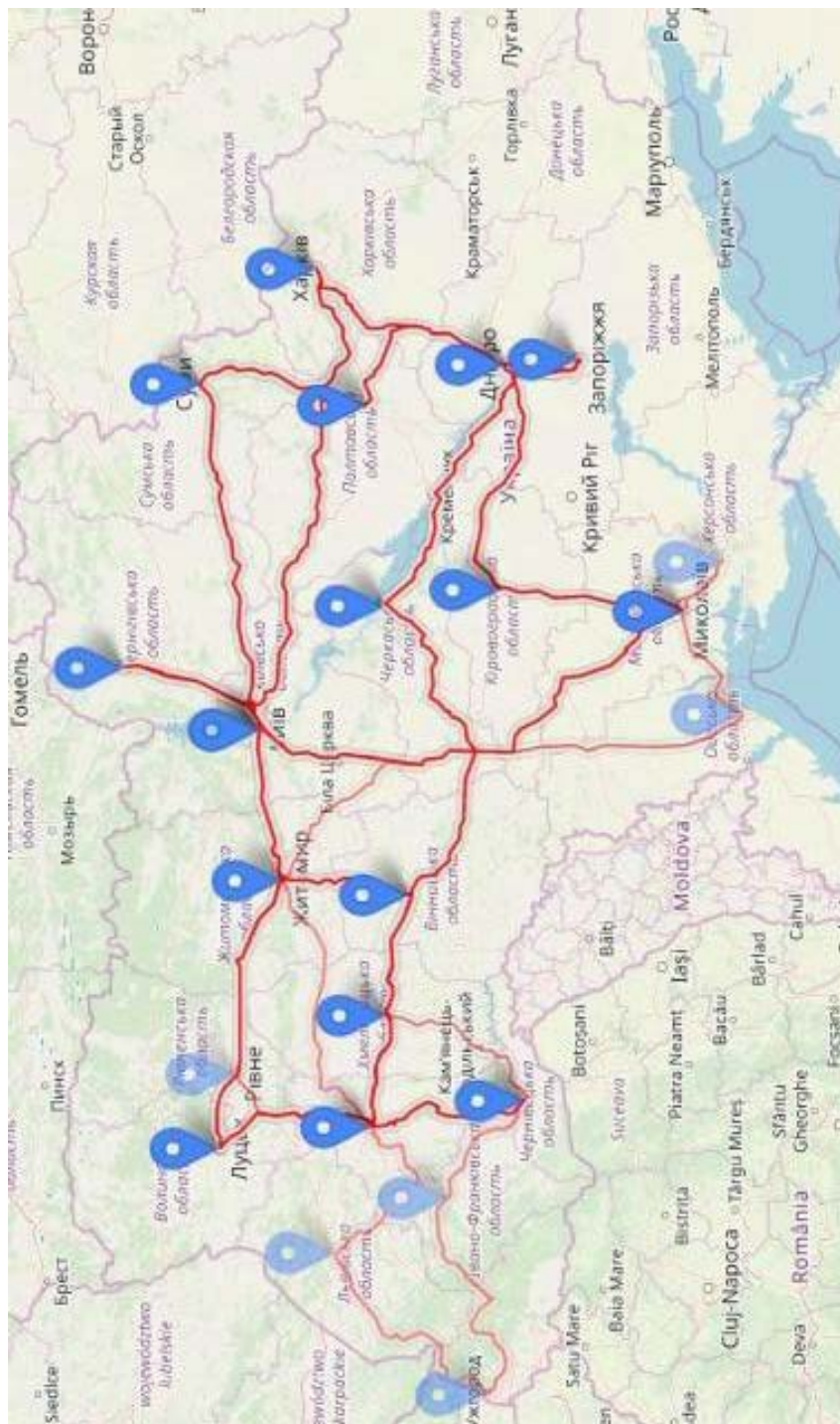


Рис. 5. Відображення пунктів найкоротшого маршруту на карті

---

Граф отриманого оптимального маршруту, що є напрямленим та одностороннім, зображено на рис. 6.

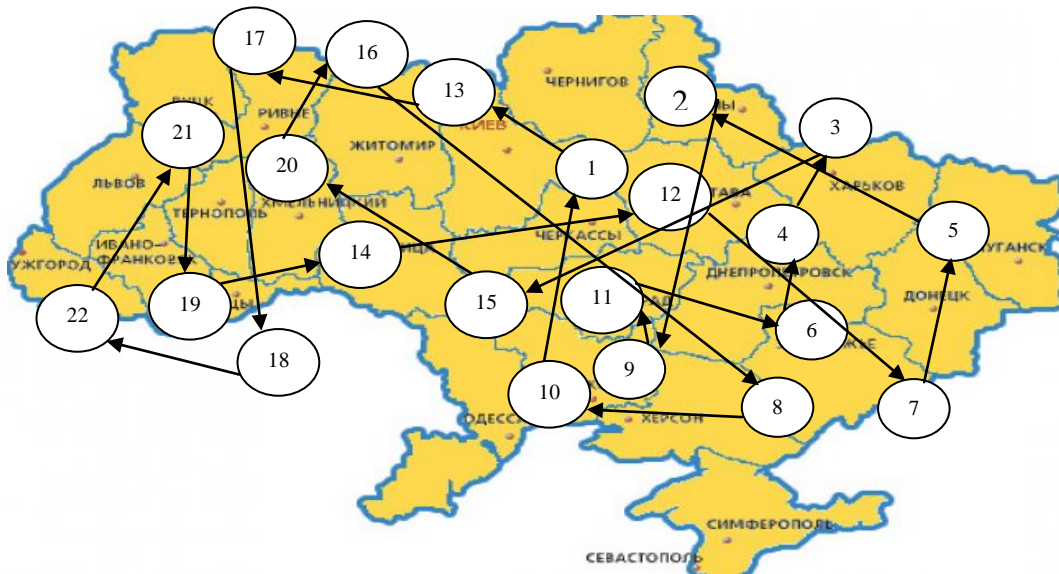


Рис. 6. Відображення на карті графу, розробленого за критерієм мінімальної відстані маршруту

Для перевізника не менш важливим критерієм ефективності перевезень є вартість, а саме отриманий чистий прибуток. Розглянемо завдання визначення оптимального маршруту перевезення Україною на основі даних тарифів перевізника.

На основі даних про відстані між обласними центрами України (крім Донецька, Луганська та АР Крим) (рис. 1) можна розрахувати тарифи на всіх напрямках перевезень у межах України. Враховуючи подачу ТЗ до місця завантаження, тобто можливий переїзд по області, та з урахуванням сезонності (дані розрахунки подано в осінній період) можна сформуванати таблицю тарифів перевізника. Коли перевізник прораховує тариф на перевезення, то він ураховує рельєф місцевості. Для цього до загального кілометражу додається 5–8 %, що показує рельєф, протяжність міста чи транспортну мережу та амортизаційні витрати. Тоді ставка перевізника дорівнює добутку кілометражу (+ 5 – 8 %) й тарифу перевізника. Тариф перевізника залежить від вартості палива та особистих потреб перевізника. Ціна дизельного пального визначається згідно з даними джерела [17].

Упорядкуємо отримані тарифи перевізника в таблицю у вигляді матриці (рис. 7). Комірки з однаковим ім'ям позначені «–», оскільки перевезення по місту не враховуємо. Інші комірки містять розраховані значення залежно від відстані між пунктами мережі (рис. 1).

	Київ	Чернігів	Суми	Полтава	Харків	Дніпро	Запоріжжя	Херсон	Миколаїв	Одеса	Кіровоград	Черкаси	Житомир	Хмельницький	Вінниця	Рівне	Луцьк	Чернівецький	Тернопіль	Львів	Ужгород	
Київ	3500	9000	7800	12000	11000	12500	12000	10500	13500	8000	5500	4000	10000	6500	8500	12000	10500	13000	14500	13000	10500	16500
Чернігів	6000	5500	7500	15000	15500	21000	18500	15500	12000	12000	9500	8000	8500	10500	10500	12000	13500	17000	18500	18500	18500	18500
Суми	5800	5800	3500	6000	7000	8000	15000	16000	18500	18500	10500	7800	9500	13000	11300	12600	14000	20000	19000	14500	21000	21000
Полтава	5500	9500	6500	4500	4500	6000	12000	12300	14500	8000	7000	7500	11500	10000	12500	12700	17000	16500	17000	14000	18000	18000
Харків	8000	11500	7000	5300	5300	7500	13000	15000	17500	11500	13000	10500	15000	14500	15000	16800	19000	19000	18000	20000	22000	22000
Дніпро	8000	10500	10000	6500	7500	3500	10000	10500	14000	7000	6500	8500	13000	13000	13000	13000	14500	18000	16500	15000	21000	21000
Запоріжжя	9000	13500	13500	7000	8000	3000	8500	9500	11500	8500	8500	13000	15000	15000	15000	17500	17000	21000	19000	17000	17500	23000
Херсон	12000	14000	16000	14000	14700	9000	7300	2500	7500	7500	10000	13000	13000	11500	17000	18000	18000	15000	19500	21500	23000	23000
Миколаїв	12000	13500	20000	14500	18000	12000	9000	4000	6000	8000	12000	13500	16000	14000	18000	17000	17000	18000	19000	20000	23500	23500
Одеса	8640	10980	20000	15770	16150	12350	10640	3900	7680	10800	9000	10450	8170	12600	13860	17020	17600	17600	14520	14220	19080	19080
Кіровоград	6080	8170	10320	7000	9360	4750	6200	4500	7680	6000	8000	8000	7040	12000	13400	18415	18200	14000	14700	14700	19320	19320
Черкаси	5500	6500	8500	8000	10500	7000	8500	10000	10800	6000	6000	6800	11200	8500	10400	11800	18200	16000	16000	16000	21000	21000
Житомир	3920	6720	13440	10560	13640	15400	13920	12240	12000	10800	8580	6440	4550	8400	10730	11180	8750	8800	8750	8800	14070	14070
Хмельницький	8580	13000	16500	14500	18000	15000	16000	12880	12500	12600	14280	6440	4800	7350	8320	8000	6900	4400	4400	7200	14000	14000
Вінниця	7000	10000	12000	11000	15000	12500	13000	10500	9000	8500	8000	7500	6000	5500	8000	10000	10000	6000	6000	9500	14500	14500
Рівне	7260	10340	16500	14520	18515	20470	17940	16330	16800	14560	5320	6300	9600	3600	3600	9900	9000	6400	5880	13440	13440	13440
Луцьк	8800	12100	18500	16280	20240	20240	22080	19550	17940	18480	16750	14750	7840	7280	9750	3600	9520	7560	5950	4200	10500	10500
Чернівецький	9000	11700	21750	19140	19580	19580	18700	17820	16500	16280	13970	14080	6660	6000	8250	8500	5600	6800	7560	10750	10750	10750
Івано-Франківськ	12060	14760	25250	22220	25300	27060	19140	17600	15400	15400	8170	6900	10360	8400	7560	6300	3500	6300	3500	7500	7500	7500
Тернопіль	8820	11340	20500	18260	21340	21340	23100	16280	14740	14520	12320	12320	6650	3300	6720	4480	4760	5600	4550	9750	9750	9750
Львів	10000	12000	16500	15500	19000	21000	25000	21500	21000	21000	14500	13500	9500	7000	8000	6600	6000	9000	5000	5000	7900	7900
Ужгород	13940	16150	19550	21850	24700	24700	26080	20330	20140	17480	19000	12730	10500	12390	10080	8820	8400	10920	6750	6750	-	-

Рис. 7. Матричний вигляд таблиці тарифів перевізника по Україні



Кількість пасажирів	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	3500	9000	7800	12000	11000	12500	12000	10500	13500	8000	9500	4000	10000	9500	8500	14200	13000	13000	12000	10500	18000
2	5000	5000	7900	15000	15500	21000	18500	19500	12000	9500	8000	9500	8500	10500	12000	12500	17000	18500	15300	13000	16500
3	6500	5800	3500	6800	7000	8000	18000	16000	18500	16500	7800	9500	13000	11300	12800	14000	20000	19000	19500	14500	21000
4	5500	9500	6500	4500	4900	6800	12000	12300	14500	8600	7000	7900	11500	10500	12500	12700	18500	17000	15000	14000	18000
5	8000	11500	7000	5300	5300	7500	11000	15000	17500	11500	13000	10500	15000	14500	15000	16800	19000	19000	18000	20000	22000
6	9000	16500	10000	6500	7500	5000	10000	16500	14000	7000	9500	8500	13000	13000	13000	14500	18000	16500	14000	15000	21000
7	9000	12000	12500	7000	8000	3000	8500	9500	11500	8500	9500	13000	15000	15000	15000	17500	27000	19000	17000	17500	23000
8	12000	14000	16000	14000	14200	9000	7100	2500	7500	7500	10000	13000	13000	11500	12000	18000	15000	19500	19500	21500	22000
9	12000	18000	20000	14500	18000	12000	9600	4000	8000	8000	12000	13500	16000	14000	18000	17000	18000	19000	17000	30000	22500
10	8700	11000	20000	19000	15200	12400	10700	5300	3800	7700	10800	9000	10200	8200	12800	13800	17000	17600	14200	14200	19000
11	6000	8200	10300	7600	9400	4800	6200	6600	6500	7700	8000	9000	9000	7100	12000	13400	18500	16200	14000	14700	15300
12	5500	6500	8500	16500	7000	8500	16000	9500	10000	5000	8000	8000	11200	8500	10400	11800	16600	16200	16000	16000	21000
13	4000	6700	12000	18600	13200	13200	15400	14000	12300	12000	10800	8600	6500	4600	6700	9400	10700	11200	8800	8800	14000
14	8600	13000	18500	14500	18000	15000	16000	14500	12900	12500	12800	14200	8500	4800	7400	8000	9000	5800	4400	7200	14000
15	7800	18000	12000	11000	15000	12500	13000	16500	9000	8500	8000	7500	8000	5500	9000	9000	10000	10000	9000	9000	14500
16	7300	10000	16500	14500	18500	20500	18000	16300	16800	16800	14600	5300	5300	5600	3600	3900	5900	9000	5400	5900	12500
17	8800	12000	8500	15200	20000	20000	22000	19600	18000	18500	16600	14800	7900	3600	3600	9000	7600	9000	9000	4200	16500
18	9000	11700	21800	19000	19600	19600	17800	16500	16300	14000	14100	6700	6000	8500	8000	8500	9000	9000	8000	7600	16800
19	12100	14800	25300	22300	25300	27100	19300	17600	17600	15400	15400	8200	8500	10400	8400	7600	6200	6200	6200	3900	7900
20	9000	11400	20600	18200	21400	23100	16300	14800	14500	12300	12300	6700	3000	6700	4500	4800	7700	5800	4800	9000	9000
21	10000	12000	16500	19000	21000	25000	21500	21000	21000	14500	13500	9500	7000	9000	8000	8000	9000	9000	5000	5000	7900
22	14000	15200	18600	21500	24300	24700	26000	26300	26300	26300	17500	19000	12800	10500	10100	8000	9600	8400	10500	8000	8000

Рис. 8. Загальний вигляд даних у середовищі Delphi з урахуванням тарифів перевізника

Задача коммівояжера методом Монте-Карло  
 Згенерований маршрут з задачі коммівояжера методом Монте-Карло

Кількість населених пунктів: 22

Оптимальний маршрут: 1 → 9 → 18 → 13 → 14 → 6 → 17 → 31 → 16 → 8 → 3 → 20 → 2 → 15 → 4 → 10 → 21 → 5 → 22 → 7 → 19 → 12

Значимий тариф: 352300

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	-	3500	9800	7800	12100	11100	12500	12000	10500	17200	9000	8500	4000	10000	8500	8500	14500	13000	13000	12000	10500	18500
2	6100	-	8500	7500	15100	15500	21000	18500	19500	12000	9500	8000	8500	8500	10500	12000	13500	17000	16500	15500	13000	18500
3	6500	5800	-	3600	8000	7000	8000	15000	16000	18500	19500	7800	9000	13000	13500	14000	20000	19000	19000	14500	21000	
4	5500	9500	8500	-	4500	4500	8000	12000	12500	14500	8000	7500	7500	11500	10000	12700	16500	17000	15000	14000	18000	
5	8000	11500	7800	5100	-	5000	7500	13000	15000	17500	11500	13000	16500	15000	14500	19000	16000	19000	18000	20000	22000	
6	8000	16500	16000	6500	7500	-	3500	16000	10500	14000	7000	6500	8000	13000	13000	14500	18000	18000	16000	14000	19000	21000
7	9000	13500	13500	7000	9000	3000	-	8500	9000	11500	9500	8500	13000	15000	15000	17500	21000	21000	17000	17500	23000	
8	12000	14000	16000	14000	14200	9000	7300	-	2500	7500	7500	10000	13000	13000	15000	17000	18000	15000	19500	19500	23000	
9	12000	12000	20000	14000	18000	12000	9000	4000	-	6000	8000	12000	13500	16000	14000	18000	17000	18000	19000	17000	20000	25000
10	8700	11000	29000	16500	16200	12400	10700	5300	3900	-	7700	10800	9000	10500	8200	13500	13000	17000	17800	14500	14200	19000
11	6000	8200	16300	7000	5400	4600	6200	5000	4500	7700	-	6000	9000	9000	7100	12000	13400	16500	16200	14000	14700	19300
12	5500	9500	8500	8000	10500	7000	9500	10000	9500	10800	6000	-	6800	11200	8900	10400	11800	18500	18200	16000	16000	21000
13	4800	6700	13500	16000	12700	12700	15600	14000	12300	12000	18000	8400	-	8200	4900	6700	8400	10700	11200	8800	8800	14000
14	6600	13000	16500	14500	16000	15000	16000	14500	12500	12500	13500	14300	6500	-	4900	7400	8300	8000	6300	4400	7200	14000
15	7000	10000	12000	11000	15000	12500	13000	10500	9000	8500	9000	7500	5000	5600	-	8000	8000	10000	10000	4000	9500	14500
16	7200	10300	16500	14500	16500	19500	20500	18000	16300	16500	16800	14500	5300	5300	9600	-	3600	9000	9000	5400	5900	13500
17	8600	12000	8500	16300	20300	20300	22000	19600	18000	16500	16800	14600	7900	7900	9600	3600	-	9500	7600	6000	4200	10500
18	9000	11700	21000	19100	19500	19600	10700	17800	16500	16300	14000	14100	6700	6000	8000	8000	8500	-	5600	6800	7600	10800
19	12100	14000	25300	22300	25300	25300	27100	19200	17600	17600	15400	15400	8200	6800	10400	8400	7900	6500	-	6300	3500	7500
20	8900	11400	29500	18300	21400	21400	23100	16300	14900	14500	12300	12300	5700	3300	5700	4900	4900	7000	5000	-	4600	9800
21	10000	12000	16500	19500	19000	21000	26000	21500	21000	21000	14500	12500	9000	7000	9000	6000	6000	9000	5000	5000	-	7500
22	14000	16300	19600	21600	24700	24700	36000	30300	30300	30200	17500	19000	13800	10500	12400	10100	8800	9000	8400	10800	6800	-

Рис. 9. Програмний розв'язок задачі визначення оптимального маршруту за критерієм максимального фрахту

---

Сформовані дані для визначення маршруту за найбільшим фрахтом і результати розв’язку задачі комівояжера (код програми мовою програмування Delphi подано в додатку) зображено на рис. 8, 9.

Оптимальний маршрут за критерієм максимального фрахту послідовно включає пункти 1→9→18→13→14→6→17→11→16→8→3→20→2→15→4→10→21→5→22→7→19→12→1.

Отже, ТЗ треба проїхати маршрутом “Київ – Миколаїв – Чернівці – Житомир – Хмельницький – Дніпро – Луцьк – Кіровоград – Рівне – Херсон – Суми – Тернопіль – Чернігів – Вінниця – Полтава – Одеса – Львів – Харків – Ужгород – Запоріжжя – Івано-Франківськ – Черкаси – Київ” із загальним фрахтом для перевізника 357 800 грн (352 300 + 5500), що є найбільшим загальним прибутком. Відображення маршруту на карті (рис. 10).

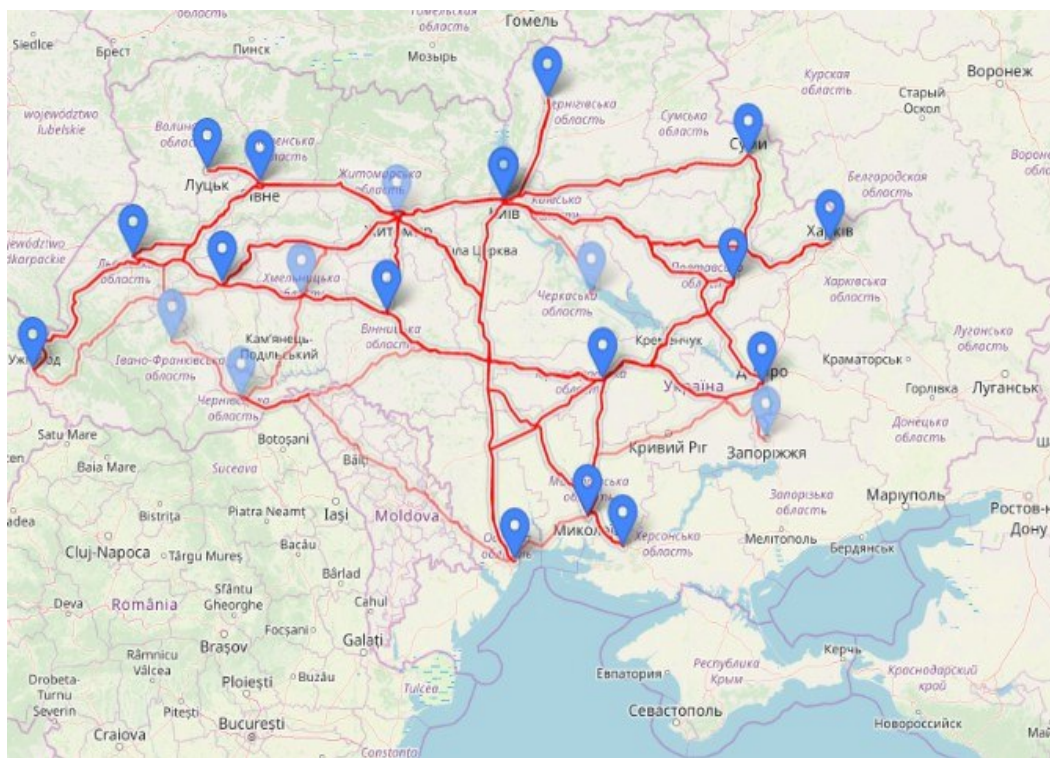


Рис. 10. Маршрут на карті за найбільшим загальним тарифом

Відображення оптимального маршруту за критерієм максимального фрахту у вигляді напрямленого одностороннього графу на карті (рис. 11).

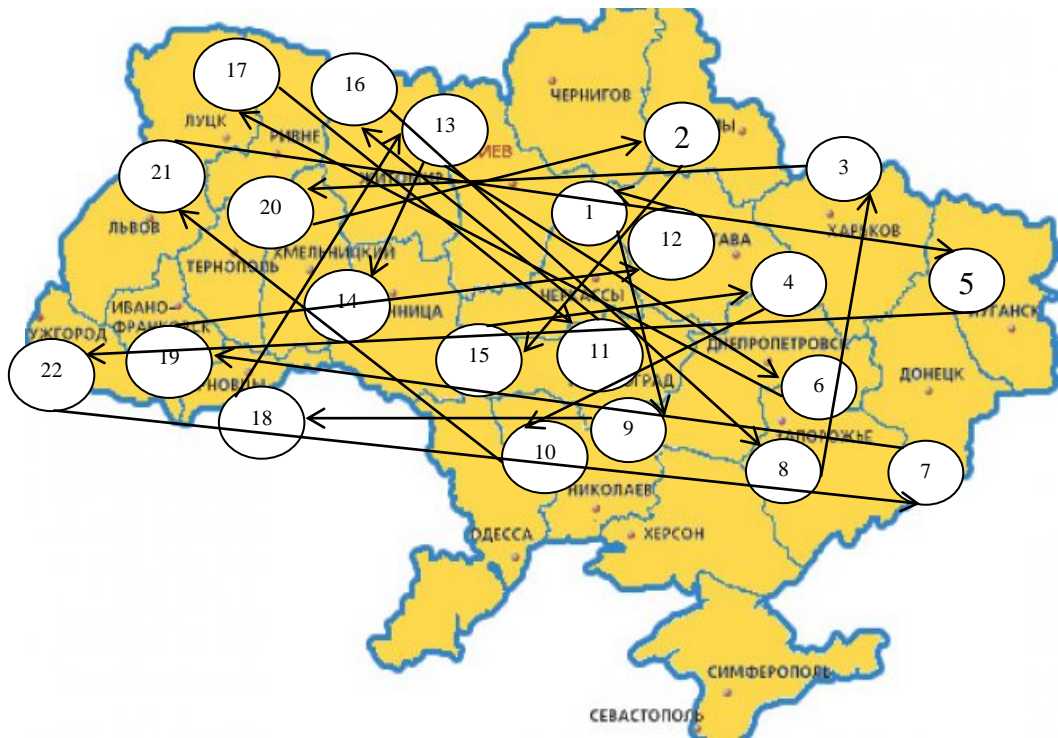


Рис. 11. Граф маршруту за максимальним загальним фрахтом на карті

Створено досить хаотичний, проте найвигідніший для перевізника маршрут. Порівнюючи дані маршрути, слід зазначити, що перший (на найкоротшу відстань) становитиме 7930 км, проте у фінансовому плані не принесе стільки коштів, як другий (на найбільший фрахт перевізника), що дорівнює 357 800 грн. Додатково прорахувавши, виявимо, що для першого маршруту прибуток становитиме 190 710 грн, а для другого відстань – 16 170 км.

**Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі.** У статті розглядалося питання покращання вантажних автомобільних перевезень по Україні на основі модифікованого мурашиного алгоритму. Серед сучасних методів для розв’язування задачі комівояжера обрано мурашиний алгоритм, оскільки його фізична основа – поведінка мурах під час пошуку їжі – є найближчою до формулювання цієї задачі, що визначає оптимальний маршрут для перевезення та його прибутковість для фірми.

1. Побудовано математичну модель, розроблено алгоритм перевезення на основі мурашиного алгоритму та розроблено блок-схему, поставлено завдання та описано процес перевезення з розрахунками тарифів перевізника.

2. Розроблено програму для прорахунку мінімального маршруту за

---

відстанню на основі задачі комівояжера в середовищі Delphi, розв'язана задача на мінімальну відстань, побудовано граф даного маршруту.

3. Розроблено програму для знаходження максимального маршруту за фрахтом перевізника, розв'язано дану задачу, а її результат у вигляді маршруту нанесено на карту у вигляді графа.

Перший (мінімальний за відстанню) маршрут – “Київ – Житомир – Луцьк – Чернівці – Ужгород – Львів – Івано-Франківськ – Хмельницький – Черкаси – Запоріжжя – Харків – Чернігів – Миколаїв – Кіровоград – Дніпро – Полтава – Суми – Вінниця – Тернопіль – Рівне – Херсон – Одеса – Київ” із загальною відстанню 7930 км та фрахтом 190 710 грн.

Другий (максимальний за фрахтом) – “Київ – Миколаїв – Чернівці – Житомир – Хмельницький – Дніпро – Луцьк – Кіровоград – Рівне – Херсон – Суми – Тернопіль – Чернігів – Вінниця – Полтава – Одеса – Львів – Харків – Ужгород – Запоріжжя – Івано-Франківськ – Черкаси – Київ” із загальним фрахтом для перевізника 357 800 грн та відстанню 16 170 км.

Дані розрахунки маршрутів досить важливі для розвитку роботи логістичних компаній, оскільки вантажні перевезення надзвичайно значущі для економіки країни й компанії та є актуальними в сьогоденні. Клієнтська база невпинно зростає, нові види товарів з кожним днем розробляються та виробляються в надзвичайних кількостях, а доставка їх до місця призначення є невід'ємною частиною торговельного процесу.

За результатами розрахунків вироблено технічні рішення щодо підвищення ефективності роботи логістичної компанії на основі модифікованого мурашиного алгоритму.

#### **Список використаних джерел:**

1. *Миротин Л. Б.* Транспортная логистика: учебник для транспортных вузов / под общей редакцией Л. Б. Миротина. Москва: Экзамен, 2003. 512.

2. *Бауэрсокс Доналд Дж., Клосс Дейвин Дж.* Логистика: интегрированная цепь поставок: пер. с англ. Н. Н. Барышниковой, Б. С. Пинскера. 2-е изд. Москва: ЗАО “Олимп-Бизнес”, 2008. 608 с.

3. *Лукинский В. С., Лукинский В. В., Пластуняк И. А., Плетенева Н. Г.* Транспортировка в логистике: учеб. пособие. Санкт-Петербург: СПбГИУЭ, 2005. 139 с.

4. *Горев А. Э.* Грузовые автомобильные перевозки. Москва, 2008. 288 с.

5. *Леснікова І. Ю., Халіпова Н. В.* Оптимізація пасажирських перевезень у вузлах мегаполісів // Вісник Академії митної служби України. Серія: “Технічні науки”. 2015. № 2 (54). С. 55–69.

6. *Літвінова Я. В.* Удосконалення логістичного управління різними

---

видами транспорту, складуванням та переробкою вантажів у транспортних вузлах: дис. ... канд. тех. наук: спец. 05.22.01 "Транспортні системи". Харків, 2017. 232 с.

7. Шуліка О. О. Формування процесу доставки тарно-штучних вантажів автомобільним транспортом у міжміському сполученні: дис. ... канд. тех. наук: спец. 05.22.01 "Транспортні системи". Харків, 2017. 232 с.

8. Khalipova N., Pasichnyk A., Lesnikova I., Kuzmenko A., Kokina M., Kutirev V. & Kushchenko Ye.. Developing the method of rational trucking routing based on the modified ant algorithm // Eastern-European Journal of Enterprise Technologies: Control processes, 3 (91). Vol 1. 68–76. DOI: 10.15587/1729-4061.2018.123862.

9. Lei K., Zhu X., Hou J., Huang W. Decision of Multimodal Transportation Scheme Based on Swarm Intelligence // Mathematical Problems in Engineering. 2014. Article ID 932832, 10.

10. Ramadhani T., Hertono G. F., Handari B. D. An Ant Colony Optimization algorithm for solving the fixed destination multi-depot multiple traveling salesman problem with non-random parameters. AIP Conference Proceedings. 2017. 1862, 030123.

11. Krzysztof K., Damian K., Iwona P., Wojciech M. K., Cezary G. Ant colony optimisation for scheduling of flexible job shop with multi-resources requirements. MATEC Web of Conferences 112. 2017. Article number 06018, 6,

12. Wang Y., Lu J. Optimization of China Crude Oil Transportation Network with Genetic Ant Colony Algorithm // Information. 2015. 6 (3), P. 467–480.

13. Hassan Md. R., Islam Md. M., Murase K. A New Local Search Based Ant Colony Optimization Algorithm for Solving Combinatorial Optimization Problems // IEICE Transactions on Information and Systems. 2010. Vol. E93. D. № 5. P. 1127–1136.

14. Мурашиний алгоритм. 2019. URL: [https://uk.wikipedia.org/wiki/Мурашиний\\_алгоритм](https://uk.wikipedia.org/wiki/Мурашиний_алгоритм).

15. Штовба С. Д., Рудий О. М. Мурашині алгоритми оптимізації // Вісник Вінницького політехнічного інституту. 2004. № 4. С. 62–69.

16. Халіпова Н. В., Черненко А. О., Леснікова І. Ю. Щодо моделювання транспортних потоків для аналізу завантаженості доріг в містах // Транспортні системи та технології перевезень: зб. наук. праць Дніпропетровського національного університету залізничного транспорту імені академіка В. Лазаряна. 2016. Вип. 12. С. 90–98.

17. Стоимость дизельного топлива на АЗС Украины. Минфин. 2019. URL: <https://index.minfin.com.ua/markets/fuel/dt>

**Код програми розрахунку оптимального маршруту в середовищі Delphi**

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, ExtCtrls, Spin, Grids, ComCtrls, ShellAPI;
type
  MAS = array[1..100, 1..100] of integer;
  VECT = array[1..1000] of integer;
type
  TForm1 = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Panel2: TPanel;
    Label2: TLabel;
    SpinEdit1: TSpinEdit;
    StringGrid1: TStringGrid;
    Button1: TButton;
    StatusBar1: TStatusBar;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure CR_Change;
    procedure SpinEdit1Change(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Label1MouseLeave(Sender: TObject);
    procedure Label1MouseMove(Sender: TObject; Shift: TShiftState; X,
      Y: Integer);
    procedure StringGrid1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

---

```

var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
  A: MAS;
  b, b_min: VECT;
  i, j, k, n, m, l, ln, S, S_min: integer;
  bbb: boolean;
  path: String;
begin
  m := SpinEdit1.Value;
  for i := 1 to m do
    for j := 1 to m do
      if (StringGrid1.Cells[j, i] <> "") and (StringGrid1.Cells[j, i] <> '-') then
        A[i, j] := StrToInt(StringGrid1.Cells[i, j])
      else
        A[i, j] := 99999;
  l := 1;
  ln := 1000;
  S_min := 99999; {Для задачі на максимум S_min:=0;}
  repeat
    b[1] := 1;
    n := 2;
    while (n <= m) do
      begin
        bbb := true;
        k := random(m + 1);
        for i := 1 to n - 1 do
          if (k = b[i]) or (k = 0) then
            begin
              bbb := false;
            break;
            end;
        if (bbb = true) then
          begin
            b[n] := k;
            n := n + 1;
          end;
        end;
      end;
end;

```



---

```

S := 0;
for k := 1 to m do
  if (k <= (m - 1)) then
    begin
      i := b[k];
      j := b[k + 1];
      S := S + A[i, j];
    end
  else
    begin
      i := b[k];
      j := b[1];
      S := S + A[i, j];
    end;
  if (S < S_min) then {Для задачі на максимум (S > S_min)}
    begin
      S_min := S;
      b_min := b;
    end;
  l := l + 1;
  until (l >= ln);
  path := IntToStr(b_min[1]);
  for i := 2 to m do
    path := path + '->' + IntToStr(b_min[i]);
  StatusBar1.Panels[0].Text := 'Оптимальний маршрут: ' + path;
  StatusBar1.Panels[1].Text := 'Загальна відстань: ' + IntToStr(S_min);
end;
procedure TForm1.CR_Change;
var
  i: integer;
begin
  StringGrid1.ColCount := SpinEdit1.Value + 1;
  StringGrid1.RowCount := SpinEdit1.Value + 1;
  for i := 1 to SpinEdit1.Value do
    begin
      StringGrid1.Cells[0, i] := IntToStr(i);
      StringGrid1.Cells[i, 0] := IntToStr(i);
      StringGrid1.Cells[i, i] := '-';
    end;
end;
procedure TForm1.FormCreate(Sender: TObject);

```

---

```
begin
  CR_Change
end;
procedure TForm1.SpinEdit1Change(Sender: TObject);
begin
  CR_Change;
end;
procedure TForm1.StringGrid1Click(Sender: TObject);
begin
end;
procedure TForm1.Button2Click(Sender: TObject);
var
  i, j: integer;
begin
  for i := 1 to SpinEdit1.Value do
    for j := 1 to SpinEdit1.Value do
      if (i <> j) then
        StringGrid1.Cells[i, j] := ";
    end;
  end;
procedure TForm1.Label1MouseLeave(Sender: TObject);
begin
  Label1.Font.Style := [];
end;
procedure SaveGrid(Grid:TStringGrid;FileName:string);
var
  f: textfile;
  x, y: integer;
begin
  assignfile(f,Filename);
  rewrite(f);
  writeln(f, grid.colcount);
  writeln(f, grid.rowcount);
  for X := 0 to grid.colcount - 1 do
    for y := 1 to grid.rowcount - 1 do
      writeln(F, grid.cells[x, y]);
    closefile(f);
  end;
procedure TForm1.Button3Click(Sender: TObject);
begin
  SaveGrid(form1.StringGrid1,'Grid1.txt');
end;
```