УДК 519.852.33 doi: 10.15421/321813

О.С. Магас, О.С. Сергєєв
Дніпровський національний університет імені Олеся Гончара

ДОСЛІДЖЕННЯ РОБАСТНОСТІ РОЗВ'ЯЗКІВ ЗАДАЧІ МАРШРУТИЗАЦІЇ ТРАНСПОРТУ

Запропоновано алгоритм розв'язання задачі маршрутизації транспорту та отримано квазіоптимальний розв'язок задачі. Перевірена робастність розв'язків задачі на реальному прикладі.

Ключові слова: маршрутизація транспорту, евристичні алгоритми, робастність розв'язків, табу-пошук, агломеративна кластеризація.

Предложен алгоритм решения задачи маршрутизации транспорта и получено квазиоптимальное решение задачи. Проверена робастность решений задачи на реальном примере.

Ключевые слова: маршрутизация транспорта, эвристические алгоритмы, робастность решений, табу-поиск, агломеративная кластеризация.

The article deals with the problem which is of relevance nowadays: vehicle routing problem. This problem is considered in order to reduce delivery costs of the distributor. The solution of this problem allows to optimize and reorganize structure of the company and decrease using of recourses. The model of the vehicle routing problem, i. e. the use of different approaches, tools and algorithms to obtain a better result is investigated. Dividing this problem on sub-problems gives opportunity for using different approaches in solving vehicle routing problem. The algorithm for solving the vehicle routing problem is proposed. The main idea of this algorithm is to use agglomerative clustering, tabu search, and union of the clusters sequentially. The program module for every method is developed. There is an opportunity to make use of parallel computing, as a result of clustering-based approach. The proposed algorithm of the vehicle routing problem is checked with solving a real problem. Problem location is defined in Dnipro city. The visualization of every step of solving is presented as combining Google Maps API and JavaScript in web implementation. The quasi-optimal solution of the considered problem is obtained and its robustness is checked. The critical values which lead to rapid increase in the transportation cost is found. Randomly data perturbation is also tested. Graphical implementation for every test is considered to create applied interpretation of results. Research results can be applied to obtain a solution to the vehicle routing problem and reduce delivery costs of the distributor. The idea of creating an interface for software, the use of other clustering algorithms, introduction of another algorithm of union of the clusters, and the use of different approaches for checking the robustness of vehicle routing problem solution is considered as a further research.

Keywords: transport routing, heuristic algorithms, robustness of solutions, taboo-search, agglomeration clustering.

Вступ. З метою зменшення витрат компанії-дистриб'ютора, у роботі буде розглянута задача маршрутизації транспорту. Розв'язок цієї задачі дозволяє

оптимізувати роботу підприємства та заощадити ресурси. В цій роботі досліджується модель задачі маршрутизації транспорту, а саме використання різних підходів, засобів та алгоритмів з метою отримання кращого результату. Ефективність дослідження цієї теми зумовлена практичною цінністю при великій складності алгоритму розв'язання. Насамперед, спроби комбінування різних етапів розв'язання та застосування для цього різних методів є актуальним і цікавим з метою знаходження нових квазіоптимальних розв'язків за мінімальний час виконання алгоритмів.

Попереднім результатом роботи є публікація [3], у якій було розглянуто логістику українського дистриб'ютора продуктів харчування. Головною метою публікації було зниження витрат компанії у використанні транспорту. Також ефективний алгоритм розв'язання запропоновано у [1]. У роботі була розглянута задача маршрутизації транспорту з обмеженнями, досліджені деякі підходи розв'язання та отримано ефективний алгоритм. Альтернативою запропонованих алгоритмів у роботі [1] стали підходи, наведені у [2]. Був описаний алгоритм розв'язання задачі маршрутизації транспорту та проведена оцінка ефективності.

Результатом застосування запропонованого алгоритму для основної задачі, буде отримання її квазіоптимального розв'язку. Метою дослідження є перевірка робастності розв'язків та вдалості поєднання алгоритмів, що застосовані безпосередньо у роботі.

Постановка задачі. Задача маршрутизації транспорту може бути сформульована наступним чином. Нехай $G=(V,A)$ – граф з ребрами A який визначається $A=\{(v_i,v_j): v_i,v_j\in V, i\neq j\}$, а $V=\{0,1,...,N\}$ – множина n+1 вершин, де вершина 0 – це депо, а інші вершини – це міста чи клієнти, які будуть обслуговуватися. Для кожного ребра (i,j), $i\neq j$ задана невід'ємна матриця відстаней $D=(C_{ij})$, де C_{ij} може бути інтерпретовано як справжня відстань, час або вартість подорожі. Маємо N клієнтів, які повинні бути відвідані K транспортними засобами. Максимально дозволена довжина маршруту для k-го транспортного засобу $-D_k$. Математичне формулювання задачі:

$$\sum_{k=1}^{K}\sum_{i=0}^{N}\sum_{j=0}^{N} C_{ij}X_{ij}^{k}\to \min \tag{1}$$

$$X_{ij}^{k}=\{1,0\}$$
$$X_{ij}^{k}=1, \text{ якщо агент } k \text{ прямує від клієнта } i \text{ до } j;\ X_{ij}^{k}=0, \text{ інакше} \tag{2}$$

Виконуються наступні умови $\forall i,j=\overline{0,N}; \forall k=\overline{1,K}$:

$$\sum_{k=1}^{K}\sum_{i=0}^{N} X_{ij}^{k}=1,\ \sum_{k=1}^{K}\sum_{j=0}^{N} X_{ij}^{k}=1 \tag{3}$$

$$\sum_{i=0}^{N} X_{ik} - \sum_{j=0}^{N} X_{kj} = 0 \tag{4}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} C_{ij} X_{ij}^{k} = D_k \tag{5}$$

$$\sum_{i=0}^{N} X_{i0}^{k} \le 1,\ \sum_{j=0}^{N} X_{0j}^{k} \le 1 \tag{6}$$

Цільовий функціонал (1) визначає загальну вартість розв'язку. Обмеження (3) означають, що всі транспортні засоби, які виїхали з депо повернуться назад. Обмеження (4) гарантує, що кожний клієнт буде точкою зупинки на маршруті. Обмеження (6) відображає наявність хоча б одного транспортного засобу. Розв'язок такої задачі дозволяє мінімізувати транспортні витрати підприємства. Вважатимемо, що на підприємстві на кожному транспортному засобі працює один агент, тобто кількість агентів та кількість транспортних засобів співпадає і дорівнює K. Нехай δ_i відповідає заробітній платні i-го кур'єра за один робочий день та γ_i – витрати на утримання i-го транспортного засобу у автопарку. Зазначимо, що це можуть бути витрати на ремонт транспортних засобів або їх прокат. Тоді можна задати витрати пов'язані з заробітною платою та утриманням транспортних засобів наступним чином:

$$R = \sum_{k=1}^{K} (\gamma_k + \delta_k) = K(\gamma + \delta) \tag{7}$$

$$F = \sum_{k=1}^{K}\sum_{i=0}^{N}\sum_{j=0}^{N} C_{ij} X_{ij}^{k} \tag{8}$$

$$Z = F + R \tag{9}$$

Функціонал R у формулі (7) позначає загальну вартість утримання автопарку. Функціонал F у формулі (8) відповідає загальній довжині маршруту всієї задачі. У формулі (7) розглядаються ідентичні транспортні засоби та витрати на них. Те ж саме стосується заробітної платні агентів. Тоді в рамках підприємства задачу маршрутизації транспорту з обмеженнями можна представити у вигляді задачі мінімізації функціоналу Z:

$$Z \to \min \tag{10}$$

Важливим обмеженням роботи агентів є тривалість робочого дня. Тобто необхідно додати до обмежень наступну умову: час, затрачений на виконання перевезень у рамках одного маршруту повинен не перевищувати тривалість робочого для T. Для цього розглянемо наступне обмеження:

$$\sum_{i=0}^{N}\sum_{j=0}^{N} t_{ij} X_{ij}^{k} \le T, \quad k = \overline{1, K} \tag{11}$$

де t_{ij} розраховується за наступною формулою:

$$t_{ij} = \frac{C_{ij}}{h}, \ \ i, j = \overline{0, N}; \tag{12}$$

h - середня швидкість руху у місті

Метод розв'язання. Для того, щоб розв'язати задачу маршрутизації транспорту, спробуємо розбити її на частини і запропонувати алгоритми для розв'язання кожної з них. Так як ми маємо декілька транспортних засобів, то було б логічним розділити множину вершин V на декілька підмножин V_i, $i = \overline{1, K}$ таких, що $\bigcup_{i=1}^{K} V_i = V$, $V_i \cap V_j = \varnothing$, $i \neq j$, елементи яких знаходились географічно поряд та їх кількість була обмежена. Тоді б для кожної підмножини $V_i \subset V$, $i = \overline{1, K}$ можна було б поставити у відповідність транспортний засіб та агента. Тоді при кількості транспортних засобів K з множини вершин V можна отримати наступну множину:

$$W = \{V_1, ..., V_K\} \tag{13}$$

Тоді для кожного агента постає задача відвідати певну невелику кількість клієнтів, стартуючи з депо і повертаючись назад. При цьому необхідно враховувати обмеження (11) на тривалість робочого дня. Такі міркування відповідають задачі комівояжера: нехай $G = (V_k, A)$, $V_k \in W$ – граф з ребрами A який визначається $A = \{(v_i, v_j) : v_i, v_j \in V_k, i \neq j\}$. Для кожного ребра (i, j), $i \neq j$ задана невід'ємна матриця відстаней $D = (C_{ij})$, де C_{ij} може бути інтерпретовано як справжня відстань, час або вартість подорожі. Тоді має місце наступне формулювання задачі:

$$F = \sum_{i \neq j} C_{ij} X_{ij} \tag{14}$$

за такими обмеженнями:

$$\sum_{i=1}^{|V_k|} X_{ij} = 1, \ \sum_{j=1}^{|V_k|} X_{ij} = 1, \ i, j \in V_k, \ i \neq j \tag{15}$$

$$\sum_{i, j \in S} X_{ij} \leq |S| - 1, \ (S \subset V_k, 2 \leq |S| \leq |V_k| - 2) \tag{16}$$

$$X_{ij} \in \{1, 0\}, \ (i, j) \in A \tag{17}$$

Для розв'язання цієї задачі застосуємо евристичний алгоритм табу-пошуку. Крок 1. Нехай задана множина $V_k = \{x_1, ..., x_{N_1}\}$, де $x_1, ..., x_{N_1} (N_1 \leq N)$ - точки. Будуємо початковий розв'язок рішення $s = \{x_1, ..., x_{N_1}\}$. Знаходимо довжину цільової функції (14) та записуємо її у $F_{\min}$. Позначимо через $N(s)$ - множину всіх можливих сусідніх вершин. Генеруємо пустий список заборон ϕ. Задаємо число $K_{\max}$ – кількість можливих ітерацій. Покладаємо $k = 1$.

Крок 2. Знаходимо новий розв'язок x такий, що $x \in N(s)$, $F(x) = \min\limits_{y \in N(s)} F(y)$, та перехід з точки s до точки x не заборонений або $F_{\min} > F(x)$. Переходимо до кроку 3.

Крок 3. Переходимо з точки s до точки x. Додаємо точку s до списку заборон. Змінюємо $F_{\min} = F(x)$. Переходимо до кроку 4.

Крок 4. Перевіряємо обмеження на кількість ітерацій: $k < K_{\max}$. Якщо ТАК, то покладаємо $k = k + 1$ та переходимо до кроку 2. Якщо НІ – переходимо до кроку 5.

Крок 5. Отримали мінімальний результат $f = F_{\min}$. Зупиняємось.

Алгоритм описаний.

Використовуючи цей алгоритм, ми можемо розв'язати задачу комівояжера для кожного агента. Слід зазначити, що є декілька варіантів побудови початкового розв'язку s. Перш за все, було б доцільно побудувати його починаючи з 0 вершини – депо. Однак для подальших міркувань спробуємо спочатку розв'язувати задачу для кожної множини $V_j (j = \overline{1, K})$, починаючи з деякої точки. Розглянемо точку $x_0 = 0$ – депо. Для реалізації цієї ідеї необхідно, згідно з алгоритмом, знати початкову точку маршруту для кожного кластеру $V_j (j = \overline{1, K})$. Природньо було б обрати для кожного V_j таку точку $p_j \in V_j$, для якої $\rho(x_0, y) \to \min$, $\forall y \in V_j (j = \overline{1, K})$, де функцію відстані ρ можна легко замінити значеннями з матриці відстаней $D = (C_{ij})$. Для того, щоб отримати розбиття множини W по формулі (13), застосуємо алгоритм кластеризації. В рамках роботи був обраний один з алгоритмів ієрархічної кластеризації – агломеративна кластеризація. Агломеративна кластеризація не вимагає конкретизації кількості кластерів, тобто існує можливість вибору кількості кластерів, оскільки будується дерево. Крім того, алгоритм не чутливий до вибору метрики відстані: всі вони, як правило, працюють однаково добре, тоді як з іншими алгоритмами кластеризації вибір метрики відстані є критичним. Нехай задана множина точок $V = \{1, 2, ..., N\}$. Зазначимо, що на цьому етапі депо відокремлюється від загальної множини V. Опишемо алгоритм агломеративної кластеризації:

Крок 1. Нехай кожний елемент множини V складає свій власний кластер $\{V_1, ..., V_N\}$. Переходимо до кроку 2.

Крок 2. Об'єднуємо найближчі кластери, наприклад V_i та V_j. Переходимо до кроку 3.

Зауважимо, що об'єднання кластерів, застосоване на кроці 2, може бути реалізоване наступним чином: для того, щоб об'єднати кластери V_i та $V_j, (i, j = \overline{1, K}; i \neq j)$ необхідно визначити новий кластер V_n, який буде містити клієнтів обох кластерів V_i та V_j; відстань до депо $p_n = \min\{p_i, p_j\}, p_i \in V_i$ та $p_j \in V_j$; загальна довжина маршруту складатиме

$f_n = f_i + f_j + \psi_{ij} + \psi_{ji} (i, j = \overline{1, K})$. При об'єднанні кластерів доречно знати мінімальну відстань між кожним з них. Для цього визначимо квадратну матрицю $B = (\psi_{ij})$, де ψ_{ij} - довжина найкоротшої відстані від кластеру V_i до кластеру V_j. Розглянемо алгоритм об'єднання кластерів.

Крок 3. Перевіряємо кількість кластерів N. Якщо $N = 1$ або виконаний інший критерій зупинки, то зупиняємось, інакше переходимо до кроку 2.

Для подальшого опису алгоритму введемо відстань $d(V_i, V_j)$, яка визначається наступною формулою:

$$d(V_i, V_j) = \frac{\sum\limits_{x \in V_i,\ x' \in V_j} C(x, x')}{|V_i| \cdot |V_j|},\ (i, j = \overline{1, K}) \tag{18}$$

У формулі (18) $C(x, x') = C_{x\,x'}$ позначає відстань між точками x та x' з матриці відстаней $D = (C_{ij})$. Критерієм зупинки алгоритму є обмеження на мінімальну відстань між кластерами. Результатом застосування алгоритму агломеративної кластеризації буде нова множина $W = \{V_1, ..., V_K\}$, де $1 \le K \le N$; $V_j (j = \overline{1, K})$ – кластер, в якому містяться деякі точки початкової множини $V = \{1, ..., N\}$.

Алгоритм описаний.

Формально K визначає кількість отриманих кластерів при застосуванні алгоритму агломеративної кластеризації. Отримавши поділ множини W за (13), постає питання про зменшення кількості агентів та транспортних засобів, що безпосередньо приведе до зменшення витрат (7). Застосуємо обмеження на тривалість робочого дня для кожного кластеру (11). Для кожного елементу $V_j \in W (j = \overline{1, K})$ знайдемо довжину оптимального шляху $f_j (j = \overline{1, K})$, застосувавши алгоритм табу-пошуку.

Нехай задано функціонал загальної вартості $Z(W)$ за формулою (9), який дорівнює загальній вартості усіх кластерів, які містяться у множині W.

Примітка. Функціонал $Z(W)$ може бути визначений як сума значень для всіх $V_j \in W (j = \overline{1, K})$ та розраховуватися за (9).

Крок 1. Нехай вхідним розв'язком буде множина $W = \{V_1, ..., V_K\}$, яка містить K кластерів. Нехай існують додаткові множини S_{old} та S_{new}. Додамо всі елементи з множини W до множини S_{old}.

Крок 2. Обчислимо значення $\text{cost} = Z(S_{old})$.

Крок 3. Перевіримо кількість елементів: $|S_{old}| > 1$. Якщо НІ, то переходимо до кроку 6. Інакше обираємо оптимальну комбінацію кластерів V_i та V_j, для яких $i \ne j$, об'єднуємо їх та додаємо до S_{new}. Переходимо до кроку 4.

Крок 4. Перевіримо чи покращився розв'язок: cost $> Z(S_{new})$. Якщо ТАК, то переходимо до кроку 5. Якщо НІ, то переходимо до кроку 6.

Крок 5. Перевіримо обмеження на тривалість робочого дня (11). Якщо за межі 8 годин не вийшли, то замінюємо всі елементи у множині S_{old} елементами множини S_{new} та переходимо до кроку 2. Інакше переходимо до кроку 6.

Крок 6. Зупиняємося. Множина S_{old} та її вартість $Z(S_{old})$ є новим покращеним розв'язком. Формуємо нову множину $W^* = \{V_j : V_j \in S_{old}, j = \overline{1,K}\}$.

Примітка. Під оптимальною комбінацією кластерів можна розуміти поєднання кластерів V_i та V_j до тих пір, поки не буде отримана мінімальна вартість Z.

Алгоритм описаний.

Наведемо загальний алгоритм розв'язання задачі маршрутизації транспорту:

Нехай задана множина $V = \{1,2,...,N\}$ та точка x_0 - депо. Нехай також відома матриця відстаней $D = (C_{ij})$.

Крок 1. Застосуємо до множини $V = \{1,2,...,N\}$ алгоритм агломеративної кластеризації. В результаті отримаємо множину $W = \{V_1,...,V_K\}$, де $V_j \in W(j = \overline{1,K})$ – кластер, який містить в собі деякі точки з початкової множини. Переходимо до кроку 2.

Крок 2. До кожного кластеру $V_j \in W(j = \overline{1,K})$ застосуємо табу-пошук і отримаємо значення $f_j(j = \overline{1,K})$ – квазіоптимальна довжина шляху на кластері. Переходимо до кроку 3.

Крок 3. Застосуємо до множини $W = \{V_1,...,V_K\}$ алгоритм об'єднання кластерів і отримаємо множину $W^* = \{V_j : V_j \in S_{old}, j = \overline{1,K}\}$ та значення $Z(W^*)$ – квазіоптимальну вартість доставки для задачі маршрутизації транспорту.

Алгоритм описаний.

Аналіз результатів. Перевіримо запропонований алгоритм на реальній задачі зменшення витрат компанії-дистриб'ютора. За географічну локацію виберемо м. Дніпро. Кожна географічна точка буде представлена двома координатами довжиною та широтою. За початкову точку x_0 – депо буде обрана точка з наступними координатами: $x_0 = (\text{lat}, \text{lng}) = (48.458297, 35.055854)$.

Для точок-клієнтів розглядалися наступні категорії місць: аптеки, кафе, поштові відділення, банки. В рамках роботи загальна кількість точок-клієнтів складає 100 точок. Для отримання даних відстаней між кожною точкою, використовується сервіс Google API Matrix Distance. Для візуалізації процесу розв'язання задачі буде використовуватися веб-інтерфейс інтегрований з

Google Maps API, написаний на мові програмування JavaScript. На рис. 1 зображені всі точки-клієнти, яких необхідно відвідати.

Застосуємо алгоритм кластеризації до вхідних даних: результат зображений на рис. 2. Застосуємо окремо для кожного кластеру процедуру табу-пошуку. За початкову точку на кожному кластері було обрано найближчу точку до x_0 - депо. Для того, щоб отримати вартість відвідання кластеру необхідно використати актуальну вартість пального 30 грн за 1 літр. За транспортний засіб був обраний автомобіль «ГАЗ-3302-XXX» придатний для перевозу товарів вагою до 1.5 тони. Витрата на пальне складає 16 літрів на 100 км. Заробітна плата кур'єрів складає 1200 грн за 1 робочий день. Витрати на утримання автопарку складає 1500 грн. До вартості відвідання кластеру слід додати витрати (7). Результати застосування табу-пошуку наведені у таблиці 1. Загальна вартість доставки товарів на даному етапі становить 30 440 грн. Зменшимо вартість доставки товарів, застосувавши алгоритм об'єднання кластерів. Для цього побудуємо матрицю найменших відстаней $B = (\psi_{ij})$ між кластерами.

Рис. 1. Зображення точок-клієнтів на карті.

Рис. 2. Застосування алгоритму кластеризації

Таблиця 1

Результат застосування табу-пошуку для кожного кластеру

Номер кластеру	Відстань на кластері, м	Відстань до депо, м	Витрати, грн
0	23307	23307	2930
1	22156	12764	2920
2	46537	4171	3160
3	28904	11696	2980
4	24547	13755	2940
5	58606	646	3280
6	41689	4116	3110
7	31230	8663	3010
8	43312	3702	3130
9	28335	13452	2980
		Загалом:	30440

Послідовне застосування алгоритму відображено на рис. 3 та на рис. 4. В результаті було отримано 5 кластерів з загальними витратами на доставку 16791 грн. Зменшення витрат досягається в силу зменшення кількості агентів та транспортних засобів. Таким чином, був отриманий квазіоптимальний розв'язок задачі маршрутизації транспорту з використанням реальних даних за наведеним алгоритмом.

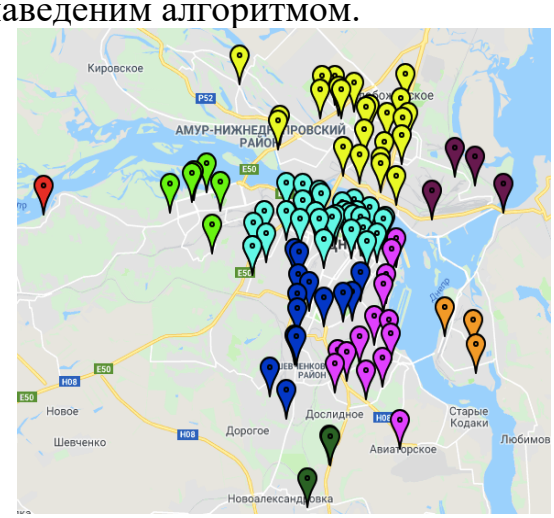

Рис. 3. Друга ітерація процесу об'єднання кластерів.

Рис. 4. Шоста(остання) ітерація процесу об'єднання кластерів.

Перейдемо до перевірки робастності розв'язків цієї задачі. Застосуємо модифікацію для матриці відстаней $D=(C_{ij})$, а саме збільшимо кожне значення $C_{ij}(i,j=\overline{0,N})$ на деякий відсоток. Результат застосування ітераційного процесу зображений на рис. 5. Виділяється різке збільшення вартості розв'язку задачі на проміжку від 100% до 105%, що зумовлюється додаванням ще одного клієнту. Такий результат інтерпретується, як проблеми трафіку у м. Дніпро, які призводять до росту вартості доставки товарів для компанії-дистриб'ютора на 14%.

Таблиця 2

Процес поєднання кластерів

№ ітерації	Кількість кластерів, шт.	Перелік кластерів	Затрати, грн.
1	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	30440
2	9	0, 1, (2,4), 3, 5, 6, 7, 8, 9	27709
3	8	(0,3), 1, (2,4), 5, 6, 7, 8, 9	24959
4	7	(0,3), (1,8), (2,4), 5, 6, 7, 9	22216
5	6	(0,3), (1,8), (2,4,7), 5, 6, 9	19493
6	5	(0,3), (1,8), (2,4,7), (5, 6), 9	16791

Примітка. Позначення типу (1,2,…) у таблиці 2 відповідає одному кластеру та несе інформацію про номери кластерів, що об'єднуються та їх порядок.

Для визначення загальної поведінки розв'язку задачі маршрутизації транспорту оцінимо ріст вартості при випадковому невеликому (не більше 10%) збуренні кожної відстані матриці $D=(C_{ij})$.Також залишимо деякий статичний крок збільшення на кожному етапі – 10%.

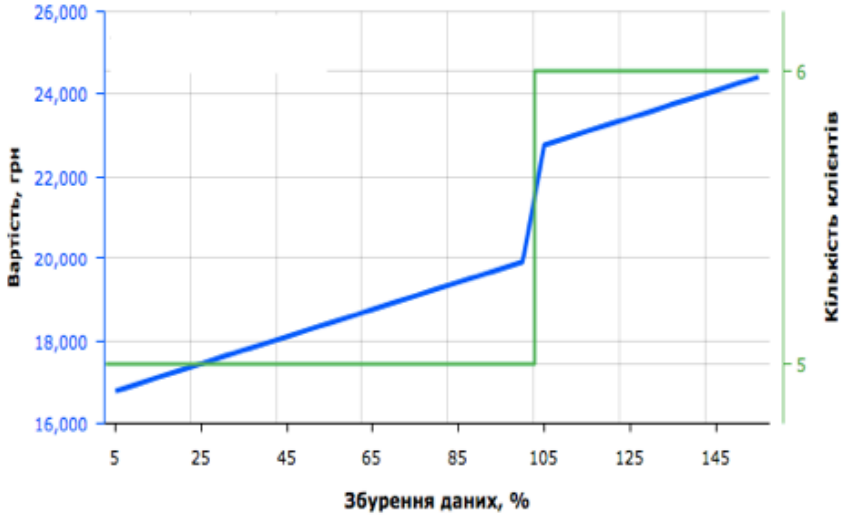

Рис. 5. Залежність росту вартості розв'язку задачі від відсотку збурення даних

Результати такого аналізу зазначені на рис. 7. На кожному етапі збільшення процес повторювався 10 разів та обиралися мінімальне, середнє та максимальне значення вартості розв'язку задачі. Отримані результати свідчать про те, що в разі довільних перебоїв трафіку вартість розв'язку зростає нелінійно. Для найгіршого випадку не помічено різкої зміни вартості. Кожен перегин для випадку максимальної вартості відповідає появі нового агенту та транспортного засобу. Для мінімального випадку існує значення при збуренні на 70-90% з'являється спад вартості розв'язку задачі маршрутизації транспорту, який відповідає зменшенню кількості кур'єрів і транспортного засобу. В середньому алгоритм показує рівномірне зростання при випадкових збуреннях випадкових ділянок доріг.

Спробуємо по черзі збільшувати мінімальні відстані до кожного кластеру і визначати яке збурення призведе до найгіршого результату. Графічний результат такого збурення даних зображено на рис. 6. Отриманий результат свідчить, що існує кластер, при віддаленні якого вартість отриманого розв'язку задачі зростає на 16%. На рис. 9 відтворений мінімальний маршрут між цими кластерами, який лежить через Південний міст.

Отримане різке збільшення вартості розв'язку задачі можна інтерпретувати як не тільки віддалення кластерів один від одного, а й ще як дорожні проблеми на маршруті A-B.

У випадку перекриття Південного мосту, розв'язок задачі збільшується на 16%. Слід зауважити, що в цьому випадку не враховується зміни загального трафіку при перекритті Південного мосту.

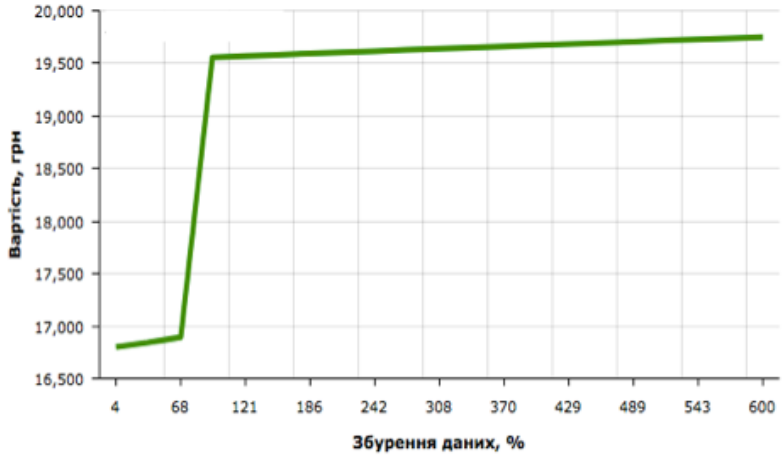

Рис. 6. Залежність росту вартості розв'язку задачі від збурення мінімальних відстаней між кластерами

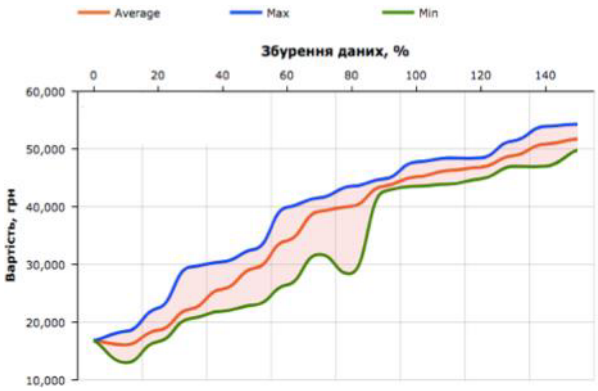

Рис. 7. Залежність росту вартості розв'язку задачі від відсотку випадкового збурення даних

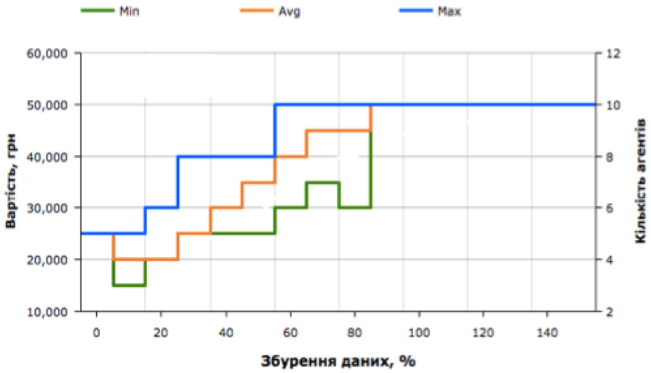

Рис. 8. Кількість клієнтів для мінімального, середнього та максимального випадку.

Рис. 9. Географічне зображення мінімального маршруту між кластерами 1 та 8.

Висновки. Запропоновано алгоритм розв'язання задачі маршрутизації транспорту, який поділяє процес розв'язання на декілька послідовних етапів. Проведена перевірка робастності розв'язків задачі на реальному прикладі. Знайденні значення збурення за яких наявний швидкий зріст вартості розв'язку задачі. Результати дослідження можуть бути застосовані для отримання розв'язку задачі маршрутизації транспорту та зменшення витрат компанії-дистриб'ютора. Для подальших досліджень розглядається ідея створення інтерфейсу для програмного забезпечення, використання інших алгоритмів кластеризації, використання нових підходів для оцінки робастності розв'язків задачі.

Бібліографічні посилання

1. **Abbas I.** Solving CVRP by Using Two-stage (DPSOTS) Algorithm / I. Abbas, A. Hassan. // Global Journal of Pure and Applied Mathematics. – 2017. – №13.
2. **Korablev V.** Approaches to Solve the Vehicle Routing Problem in the Valuables Delivery Domain / V. Korablev, I. Makeev, E. Kharitovon. // Procedia Computer Science. – 2016. – №88. – С. 487–492.
3. **Kuznietsov K. A.** Cluster-based supply chain logistics: a case study of a Ukrainian food distributor / K. A. Kuznietsov, V. A. Gromov, V. A. Skorohod. // IMA Journal of Management Mathematics. – 2017. – №28. – С. 553–578.

Надійшла до редколегії: 10.10.2018.