

УДК 656.212.5:681.3

ПАХОМОВА В. М., ЛЕПЕХА Р. О.

PAKHOMOVA V. N., LEPEKHA R. O.

## **Аналіз методів з природними механізмами визначення оптимального маршруту в комп'ютерній мережі Придніпровської залізниці**

### **ANALYSIS METHODS WITH THE NATURAL MECHANISMS OF DETERMINING THE OPTIMAL ROUTE NETWORK DNIEPER RAILWAY**

Для визначення оптимального маршруту розглянутого фрагмента комп'ютерної мережі Придніпровської залізниці розроблено на Java програмну модель «Route», що працює в двох режимах: за мурашиним алгоритмом елітної стратегії та канонічним генетичним алгоритмом. Оцінено, що залежність часу обчислення на моделі «Route» за мурашиним та генетичним алгоритмами від кількості маршрутизаторів в комп'ютерній мережі має поліноміальний характер, тоді як за алгоритмом Крускала – експоненціальний. Дослідження часу роботи моделі «Route» при різній кількості маршрутизаторів в комп'ютерній мережі показало, що в середньому визначення оптимального маршруту за мурашиним алгоритмом швидше приблизно в сто разів за генетичний алгоритм.

*Ключові слова:* комп'ютерна мережа, мурашиний алгоритм, феромон, елітні мурахи, генетичний алгоритм, популяція, хромосома, кроссинговер, мутація.

Для определения оптимального маршрута рассматриваемого фрагмента компьютерной сети Приднепровской железной дороги разработана на Java программная модель «Route», работающая в двух режимах: по муравьиному алгоритму элитной стратегии и каноническому генетическому алгоритму. Оценено, что зависимость времени вычисления на модели «Route» по муравьиному и генетическому алгоритмам от количества маршрутизаторов в компьютерной сети имеет полиномиальный характер, тогда как по алгоритму Крускала - экспоненциальный. Исследование времени работы модели «Route» при разном количестве маршрутизаторов в компьютерной сети показало, что в среднем определение оптимального маршрута по муравьиному алгоритму быстрее примерно в сто раз по сравнению с генетическим алгоритмом.

*Ключевые слова:* компьютерная сеть, муравьиный алгоритм, феромон, элитные муравьи, генетический алгоритм, популяция, хромосома, кроссинговер, мутация.

To determine the optimal route under consideration fragment network Dnieper railway developed in Java programming model «Route», working in two modes: on ant algorithm elite strategy and the canonical genetic algorithm. It is estimated that the dependence of the run-time model «Route» on genetic algorithm and ant on the number of routers in the network has a polynomial in nature, while the algorithm Kruskal - exponential. Study Time model «Route» with various numbers of routers in the network showed that the average determination of the optimal route ant algorithm is faster by about a hundred times in comparison with the genetic algorithm.

*Keywords:* computer network, the ant algorithm, pheromone elite ants, genetic algorithm, population, chromosome crossover, mutation.

### **Вступ і постановка проблеми**

Одною з основних вимог, що висуваються до алгоритмів маршрутизації, є їх швидка збіжність до оптимального рішення, яке продиктовано необхідністю їх протокольної реалізації в реальному масштабі часу в умовах характеристик трафіку, топології та завантаженості комп'ютерних мереж, що постійно змінюються [1]. Класичні алгоритми пошуку найкоротшого шляху на графі, такі як алгоритми Прима та Дійкстра, Беллмана-Форда та Крускала [2], не забезпечують прийняттого рівня рішень в умовах підтримки декількох метрик. Одним з підходів вирішення завдань маршрутизації в комп'ютерних мережах є науковий напрям «Природні обчислення» (*Natural Computing*), який поєднує методи з природними механізмами прийняття рішень, а саме: мурашині алгоритми (*Ant Colony Algorithms*), генетичні алгоритми (*Genetic Algorithms*), нейромережні обчислення (*Neural Network Computing*).

**Аналіз публікацій по темі дослідження**

Відомо, що актуальною є задача визначення оптимального маршруту в комп'ютерній мережі за мінімальний час. У [3] показано, що за допомогою нейронної мережі можна забезпечити знаходження близького до оптимального вирішення задачі комівояжера та пошуку найкоротшого шляху на графі. Використання нейронних мереж дозволяє знаходити маршрути для комп'ютерних мереж з великим ( $>100$ ) числом вузлів, високою динамікою зміни її топології та характеристик каналів зв'язку. В [1] проведена оцінка можливості використання для вирішення завдань маршрутизації наступних нейронних мереж: багатошарового персептрону, мережі RBF і мережі Хопфілда. Встановлено, що найбільш перспективним засобом вирішення завдання маршрутизації є нейронна мережа Хопфілда, яка здатна працювати в умовах динамічної зміни топології мережі та характеристик каналів передачі даних. Однак, при використанні мережі Хопфілда потрібне проведення додаткових досліджень передавальних функцій нейронів і функцій енергії нейронної мережі, які забезпечували б швидку збіжність і високу якість виконання завдання маршрутизації [4]. Але поки недостатня увага приділяється знаходженню мінімального остовного дерева (МОД) в комп'ютерних мережах з використанням мурашиних алгоритмів [5-8] та генетичних алгоритмів [9-12].

### Формулювання цілей статті

Визначити МОД в комп'ютерній мережі Придніпровської залізниці, фрагмент якої представлений на рис. 1, на основі розробленій програмній моделі за мурашиним алгоритмом елітної стратегії та канонічним генетичним алгоритмом. Виконати дослідження часу роботи моделі за різними алгоритмами.

### Основна частина

**1. Постановка задачі.** Комп'ютерну мережу Придніпровської залізниці представлено у вигляді неорієнтованого графа  $G(V, W)$ , де  $V$  – множина вершин графа, кількість яких  $N$ , причому кожна вершина моделює собою вузол (маршрутизатор) комп'ютерної мережі;  $W$  – множина ребер графа, кількість яких  $M$ . Кожному ребру графа присвоюється певна вага, яка відповідає пропускної спроможності (максимальний обсяг даних, переданий мережею в одиницю часу):

$$C = \{c_{ij}\}, \quad (1)$$

де  $C_{ij}$  – пропускна спроможність каналу між  $i$  та  $j$  вузлами мережі,  $Mbit/c$ .

Необхідно знайти МОД в комп'ютерній мережі, тобто частку існуючих ребер, які забезпечують, з одного боку, шляхи з'єднання кожної вершини графа з кожною та, з іншого боку, надають мінімальну вагу графу.

Введемо масив

$$X = \{x_{ij}\}, \quad (2)$$

де  $x_{ij}$  – наявність трафіку, що передається в мережу між  $i$ -ою та  $j$ -ою вершинами.

В якості критерію оптимальності (ваги графу МОД) виступає наступний вираз

$$\min\left(\sum_{i=1}^N \sum_{j=1}^N f_{ij} x_{ij}\right), \quad f_{ij} = \frac{10^7}{c_{ij}} \quad (3)$$

В якості обмеження – забезпечення умови наявності шляхів  $x_{ij} \in \{0,1\}$ , тобто змінна приймає значення 1, якщо трафік протікає по каналу  $(i,j)$ ; у противному випадку - 0.

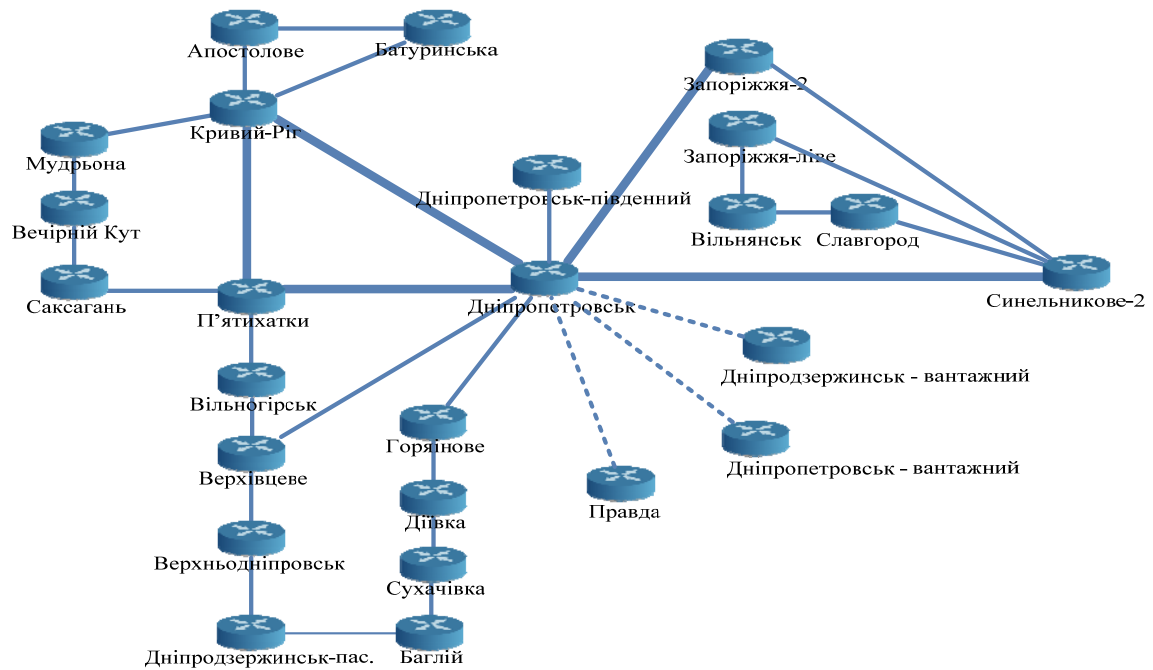


Рисунок 1 – Розглянутий фрагмент комп'ютерної мережі Придніпровської залізниці

Примітка: ————— – 10 Мбіт/с; ————— – 2 Мбіт/с; - - - - - – 384 Кбіт/с

**2. Рішення задачі за мурашиним алгоритмом.** Збільшена блок-схема мурашиного алгоритму представлена на рис. 2.

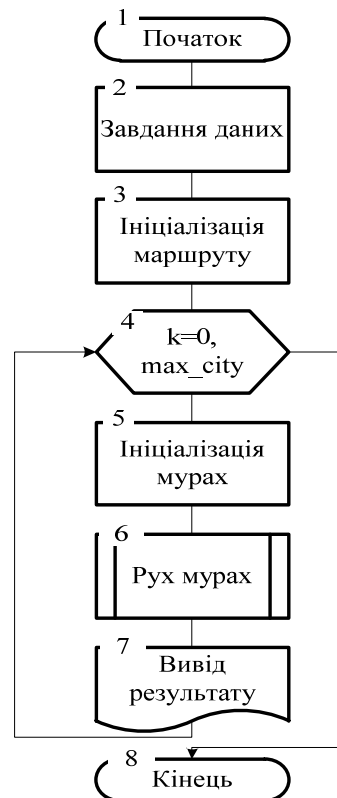


Рисунок 2 – Збільшена схема мурашиного алгоритму

Вхідні данні алгоритму: кількість мурах, кількість маршрутизаторів, важливість ваги ребра, важливість рівня феромона на ребрі, порядок довжини оптимального маршруту, коефіцієнт випаровування феромонів,

кількість елітних мурах, координати маршрутизаторів, метрика каналів зв'язку, завдання початкового кращого результату для мурахи та початкова кількість феромонів. Основною частиною мурашиного алгоритму є підпрограма «Рух мурах», деталізована схема якої представлена на рис. 3.

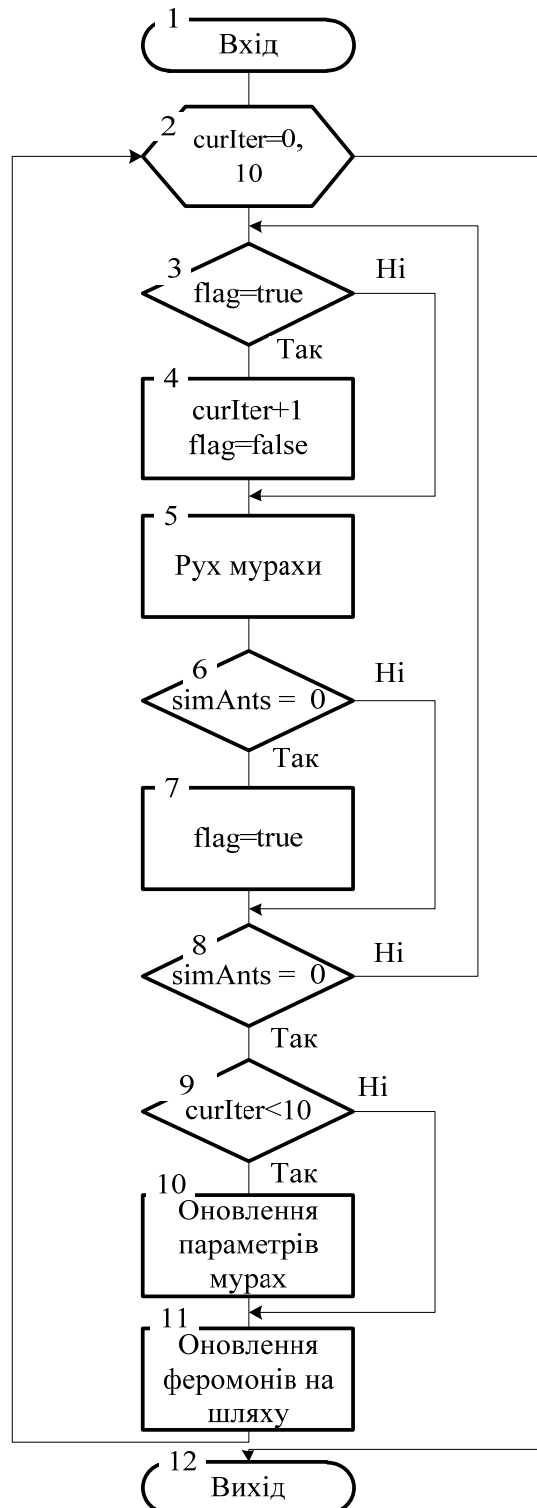


Рисунок 3 – Деталізована схема «Рух мурах»

Ймовірність переходу  $k$ -ї мурахи із маршрутизатора  $i$  на маршрутизатор  $j$  на  $t$ -й ітерації (вибір напрямку мурашки) розраховується за випадково-пропорційним правилом [7]

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \times [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \times [\eta_{il}]^\beta}, & \text{якщо } j \in J_i^k \\ 0, & \text{якщо } j \notin J_i^k \end{cases}, \quad (3)$$

де  $\tau_{ij}(t)$  – кількість феромону між маршрутизаторами залишеного за час  $t$ ,  $\eta_{ij}(t)$  – видимість, величина обернена до відстані:  $\eta_{ij} = 1/D_{ij}$ , де  $D_{ij}$  – відстань між маршрутизаторами  $i$  та  $j$ ,  $\alpha$  і  $\beta$  – параметри, які є вагами інтенсивності сліду феромону та видимості.

Після завершення маршруту кожна мураха  $k$  відкладає на ребро  $(i, j)$  таку кількість феромону [7]

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

де  $T^k(t)$  – маршрут, зроблений мурахою  $k$  на ітерації  $t$ ;  $L^k(t)$  – його довжина;  $Q$  – регульований параметр, значення якого обирають одного порядку з довжиною оптимального маршруту.

Крім того, потрібно забезпечити випаровування феромону – це зменшення кількості відкладеного на попередніх ітераціях феромону. Інтенсивність випаровування феромону задається за допомогою коефіцієнта випаровування  $\rho \in [0,1]$ . Кінцеве правило оновлення феромону приймає вигляд [7]

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

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

Для покращення характеристик мурашиного алгоритму використовують елітних мурах, які підсилюють ребра найкращого маршруту  $T^*$ , знайденого від початку пошуку. Кількість феромону, що відкладається на ребрах маршруту  $T^*$ , приймаються рівною  $Q/L^*$ , де  $L^*$  – довжина маршруту  $T^*$ . Підсилений слід феромону вздовж маршруту  $T^*$  буде спрямовувати інших мурах до пошуку розв'язків, що містять декілька ребер найкращого на даний момент маршруту  $T^*$ . Якщо в мурашнику є елітних мурах, то ребра найкращого маршруту  $T^*$  отримають загальне підсилення [7]

$$\Delta \tau_e = e \cdot \frac{Q}{L^*}, \quad (6)$$

де  $e$  – кількість елітних мурах;  $Q$  – параметр, що має значення порядку довжини оптимального маршруту;  $L^*$  – довжина маршруту найкращого маршруту  $T^*$ .

Для рішення поставленої задачі розроблена на Java програмна модель «Route», яка працює в двох режимах: визначення оптимального маршруту за мурашиним та генетичним алгоритмами (режими 1-2 відповідно). Структура моделі «Route» в режимі 1 представлена на рис. 4, в результаті її виконання виводяться граф, що наведений на рис. 5, на якому зображені вузли та найкоротші маршрути їх з'єднань. Вага отриманого графу (МОД) розглянутого фрагмента в комп'ютерній мережі Придніпровській залізниці за мурашиним алгоритмом склала 76928 ум. од.; час затрачений на розрахунок досягає 0,032 с.

**3. Рішення задачі за генетичним алгоритмом.** Збільшена схема генетичного алгоритму представлена на рис. 6, що передбачає ініціалізацію наступних параметрів: коефіцієнта виконання мутації, розміру турніру, розміру популяції, кількості епох, увімкнення функції елітизму, координат маршрутизаторів, метрики каналів зв'язку. До основних генетичних операторів відносяться: створення популяції, селекція, кросинговер, мутація.

Створення початкової популяції виконується випадковим чином, наслідком першого кроку є популяція  $H$ , що налічує  $n$  осіб. У проміжну популяцію копіюються хромосоми із поточної популяції відповідно до їх імовірності відтворення. Є різноманітні схеми відбору, найпопулярніша – пропорційний відбір

$$P(i) = \frac{f(i)}{\sum_{i=1}^n f(i)}. \quad (7)$$

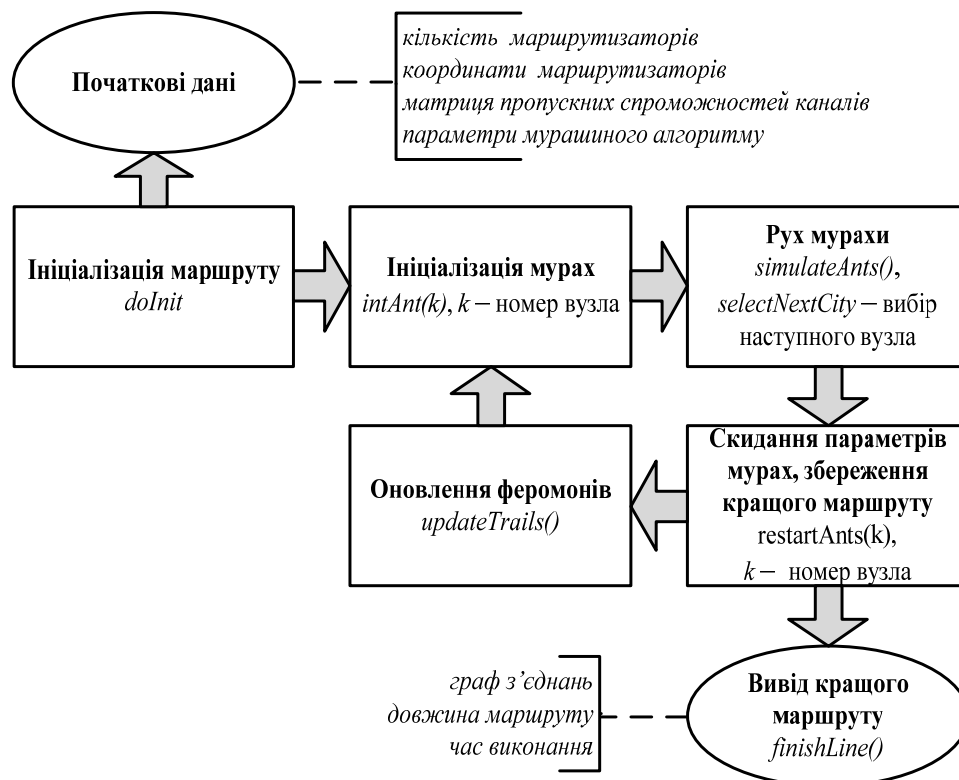


Рисунок 4 – Структура програмної моделі “Route” (режим 1)

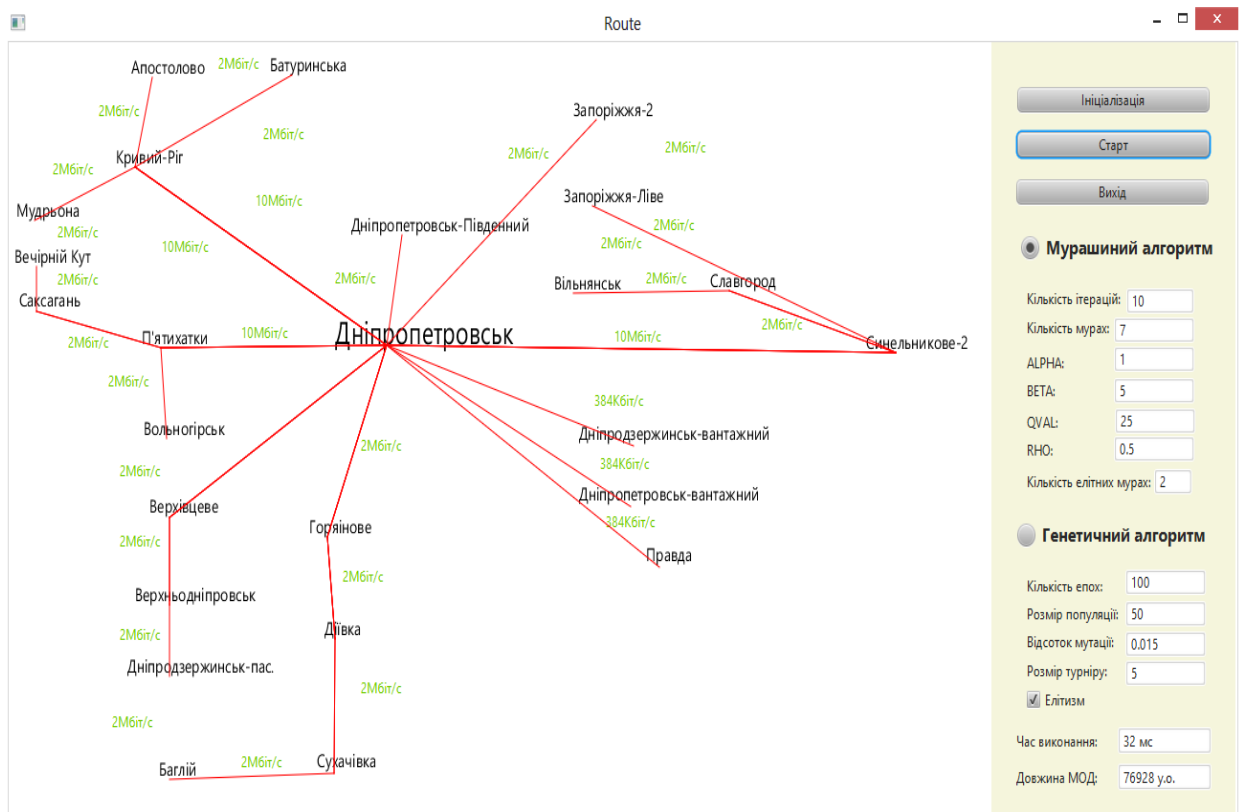


Рисунок 5 – Результат виконання моделі «Route» (режим 1)

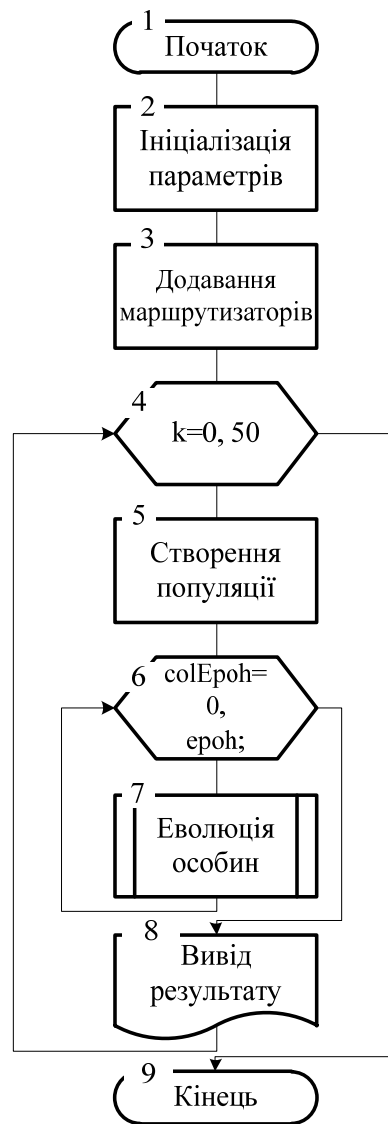


Рисунок 6 – Збільшена схема генетичного алгоритму

Основною частиною генетичного алгоритму є підпрограма «Еволюція особин», деталізована схема якої представлена на рис. 7.

При турнірному відборі з популяції, яка містить  $n$  особин, вибираються випадковим чином  $t$  особин, і найкраща з них особина записується в проміжний масив. Ця операція повторюється  $n$  разів. Особини в отриманому проміжному масиві потім використовуються для схрещування (також випадковим чином). Розмір групи рядків, що відбираються для турніру, часто дорівнює 2; взагалі  $t$  називають чисельністю турніру. Випадковим чином визначається точка всередині хромосоми (точка розриву), в якій обидві хромосоми діляться на дві частини й обмінюються ними. Такий тип кросингверу називається однокрапковим, що представлено на рис. 8, при ньому батьківські хромосоми розділяються тільки в одній випадковій точці; можливе розділення в декількох точках.

Використовуючи генотипи двох батьків у вигляді перестановок, створюються генотипи пари нащадків у вигляді нових допустимих перестановок. Генотипи нащадків зберігають порядок і позицію імен міст в послідовності одного з батьків, в той же час вони зберігають від порушення відносного порядку імен міст від іншого батька. Основна ідея кросингверу порядку полягає в тому, що важливий порядок міст, а не їх позиції.

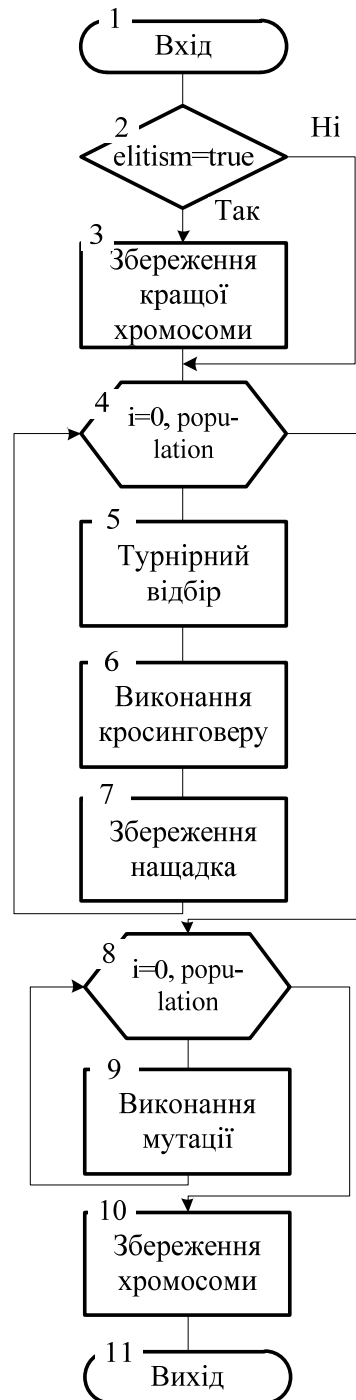


Рисунок 7 – Деталізована схема «Еволюція особин»

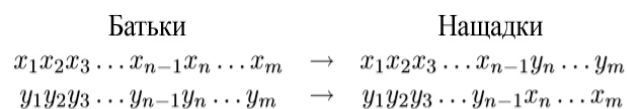


Рисунок 8 – Однокрапковий кросингвер

Так само, як і кросингвер, мутації можуть проводитися не тільки по одній випадковій точці. Можна вибирати для зміни кілька точок у хромосомі, причому їх число також може бути випадковим. Точкова мутація –

кодування мутанта виходить шляхом одноразової перестановки алелей у випадково обраному локусі та сусідньому з ним. Імовірність мутації може бути фіксованим або випадковим числом на відріжку  $[0;1]$ .

Структура моделі «Route» в режимі 2 представлена на рис. 9, в результаті її виконання також виводиться граф, на якому зображені вузли та найкоротші маршрути їх з'єднань. Вага отриманого графу (МОД) розглянутого фрагмента в комп'ютерній мережі Придніпровській залізниці за генетичним алгоритмом склала 76928 ум. од.; час затрачений на розрахунок досягає 2,957 с.

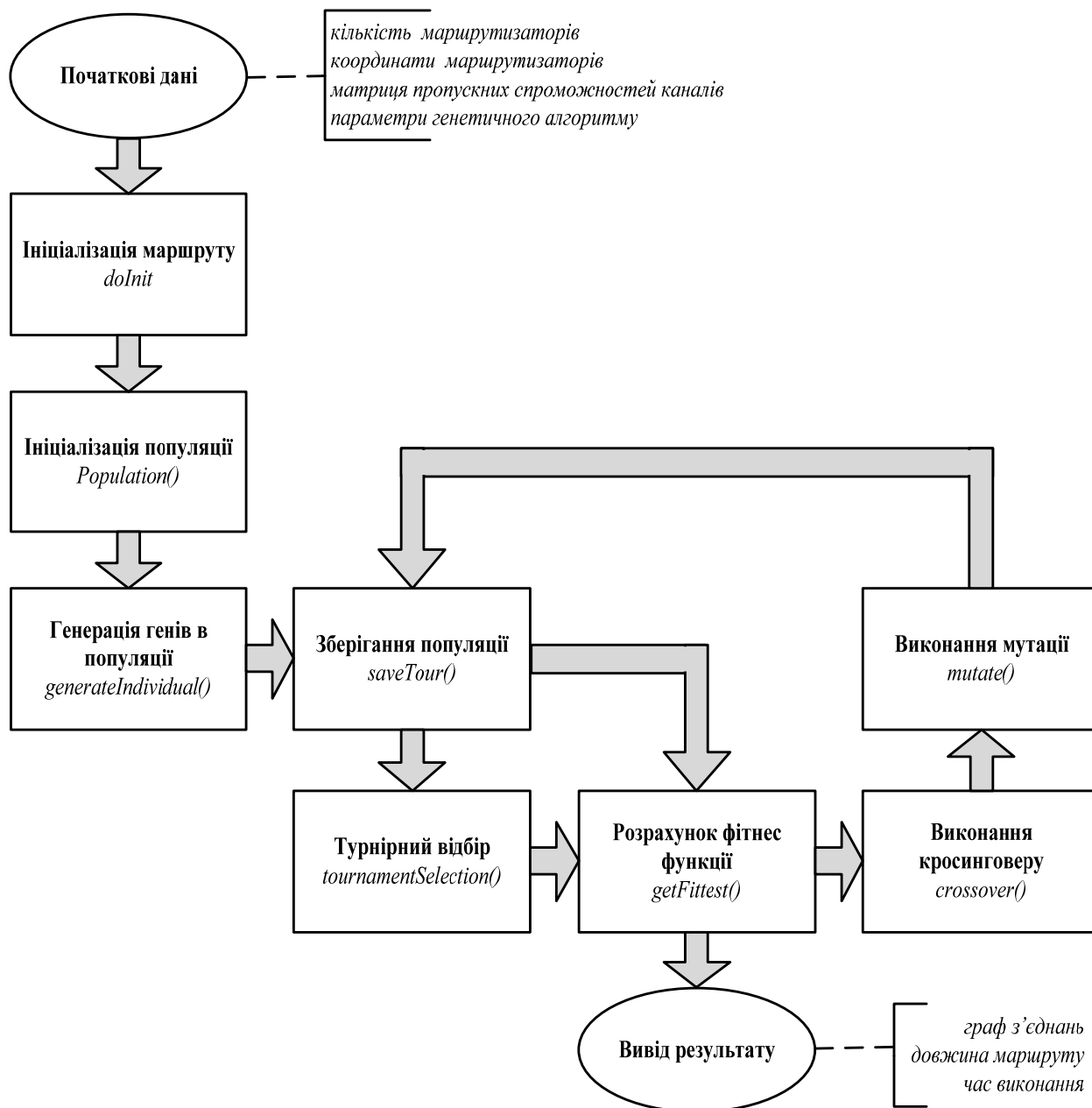


Рисунок 9 – Структура програмної моделі «Route» (режим 2)

**4. Дослідження часу роботи алгоритмів.** Зрівняння результатів отриманих різними методами для розглянутого фрагмента комп'ютерної мережі Придніпровської залізниці з різною кількістю маршрутизаторів приведені у табл. 1. Усі методи (мурашиний алгоритм елітної стратегії та канонічний генетичний алгоритм, а також алгоритм Крускала) дають правильний результат. Запуск відповідних програмних моделей проводився 10 разів для 15, 20 та 25 маршрутизаторів. З табл. 1 видно, що зі зміною кількості маршрутизаторів в комп'ютерній

мережі (15, 20, 25) мурашиний алгоритм дає найкращий результат (0,012 с, 0,02 с і 0,032 с відповідно) в порівнянні з генетичним алгоритмом (1,56 с, 2,18 с і 2,96 с відповідно).

Оцінено за допомогою пакета Excel характер залежності часу роботи розробленої моделі «Route» (в режимах 1-2) та існуючої моделі «KruskalMST» [14-15] від кількості маршрутизаторів в комп'ютерній мережі Придніпровської залізниці: поліноміальна залежність за мурашиним і генетичним алгоритмами та експоненціальна залежність за алгоритмом Крускала.

Таблиця 1–Середні значення часу роботи моделей, с

| Кількість маршрутизаторів в мережі | Модель «Route»     |                     | Модель «KruskalMST» |
|------------------------------------|--------------------|---------------------|---------------------|
|                                    | Мурашиний алгоритм | Генетичний алгоритм | Алгоритм Крускала   |
| 15                                 | 0,012              | 1,560               | 0,029               |
| 20                                 | 0,020              | 2,180               | 0,043               |
| 25                                 | 0,032              | 2,960               | 0,068               |

Крім того, виконано дослідження процесу визначення оптимального маршруту від параметрів мурашиного алгоритму елітної стратегії на моделі «Route» при змінюванні наступних параметрів: кількості мурах (від 2 до 9) та кількості ітерацій (від 3 до 11). Експерименти проводилися для фрагмента комп'ютерній мережі Придніпровської залізниці, структура якого представлена на рис. 1. На програмній моделі «Route» (режим 1) знайдений оптимальний маршрут, довжина якого 76928 ум. од. та 24 задіяних каналів зв'язку. Динаміку зміни довжини маршруту та кількості задіяних каналів від кількості мурах відображено на рис. 10 (а); штрихованою лінією позначено оптимальну кількість мурах: 7. Динаміку зміни довжини маршруту та кількості задіяних каналів зв'язку в залежності від кількості ітерацій показано на рис. 10 (б); штрихованою лінією позначено оптимальну кількість ітерацій: 10.

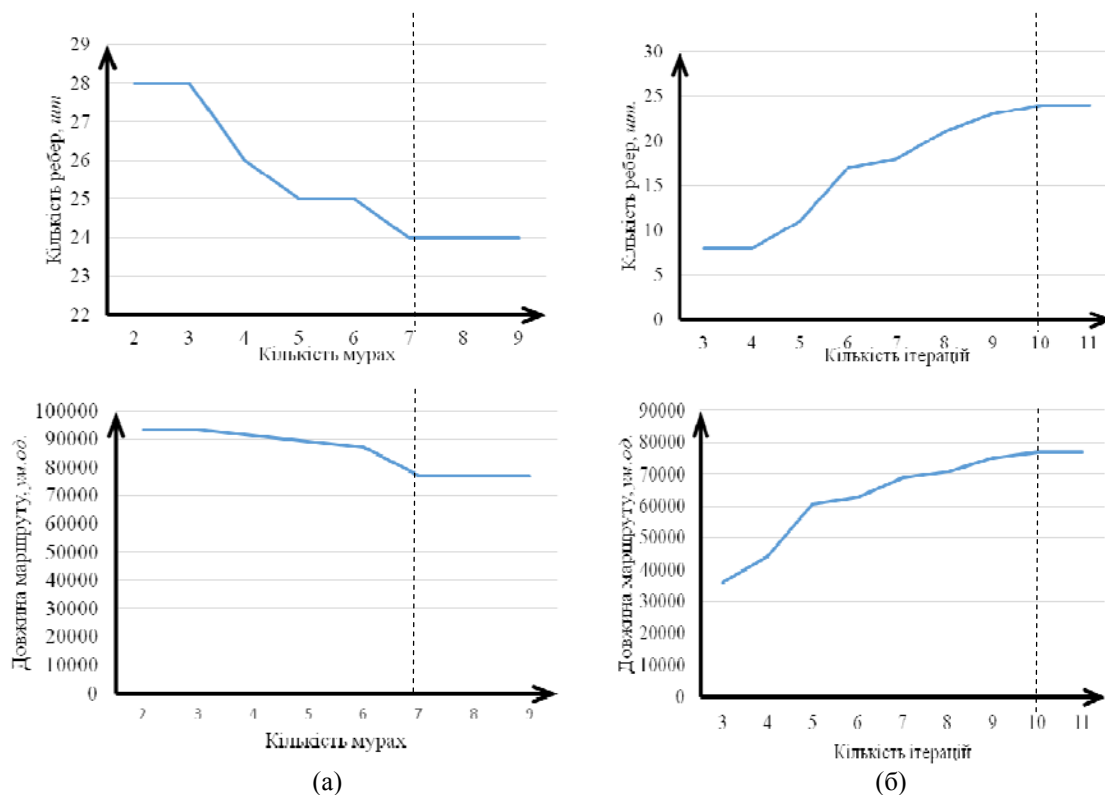


Рисунок 10 – Дослідження процесу за мурашиним алгоритмом (при  $\alpha=1$ ,  $\beta=5$ ,  $q=25$ ,  $\rho=0,5$ ) в залежності від: (а) – кількості мурах; (б) – кількості ітерацій

## Висновки

1. Для визначення МОД розглянутого фрагменту комп'ютерної мережі Придніпровської залізниці розроблено на мові Java модель «Route», яка працює в двох режимах: за мурашиним алгоритмом елітної стратегії та канонічним генетичним алгоритмом відповідно. У моделі «Route» передбачено визначення ваги отриманого графу (МОД) та витраченого часу на моделювання.
2. Результати, що одержано на розробленій моделі «Route» в режимах 1-2, співпадають з мінімальним остовним деревом, який побудовано за алгоритмом Крускала на основі існуючої моделі «KruskalMST», що підтверджує правильність розробленої моделі.
3. Дослідження часу роботи програмних моделей при різних кількості маршрутизаторів комп'ютерної мережі Придніпровської залізниці показало, що в середньому визначення МОД за мурашиним алгоритмом елітної стратегії швидше в 2 рази за алгоритм Крускала та приблизно в 100 разів за канонічний генетичний алгоритм.
4. Оцінено, що залежність часу виконання розробленої програмної моделі «Route» за мурашиним та генетичним алгоритмами від кількості маршрутизаторів в комп'ютерній мережі Придніпровської залізниці має поліноміальний характер, тоді як за алгоритмом Крускала на основі існуючої моделі «KruskalMST» – експоненціальний.
5. Дослідження процесу визначення оптимального маршруту розглянутого фрагмента в комп'ютерній мережі Придніпровської залізниці з 25-ю маршрутизаторами на моделі «Route» за мурашиним алгоритмом елітної стратегії (при  $\alpha=1$ ,  $\beta=5$ ,  $q=25$ ,  $\rho=0,5$ ) показало, що кількість мурах та ітерацій для досягнення оптимального рішення складе 7 та 10 відповідно.

## Література

1. Павленко М. А. Анализ возможностей искусственных нейронных сетей для решения задач однопутевой маршрутизации в ТКС [Текст] / М. А. Павленко // Проблемы телекоммуникаций. – 2011. – № 2 (4). – С. 118–127.
2. Бертсекас Д. Сети передачи данных [Текст] / Д. Бертсекас, Р. Галлагер. – М.: Мир, 1989. – 544 с.
3. Hopfield J. J. Neural networks and physical systems with emergent collective computational abilities [Текст] / J. J. Hopfield // Proceedings of National Academy of Sciences. – 1982. – Vol. 79, No. 8. – P. 2554–2558.
4. Павленко М. А. Решение задачи маршрутизации на основе использования нейронной сети Хопфилда с разработкой функции Ляпунова с заданными свойствами [Текст] / М. А. Павленко, А. А. Романюк, В. Ю. Яковлев // Проблемы телекоммуникаций. – 2012. – № 2 (7). – С. 43–57.
5. Dorigo M. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem [Text] / M. Dorigo, L.M. Gambardella // IEEE Trans. on Evolutionary Computation. – 1997. – №1 (1). – P. 53–66.
6. Штовба С. Д. Муравьиные алгоритмы [Текст] / С. Д. Штовба // Exponenta Pro. Математика в приложении. – 2003. – № 4 (4). – С. 70–75.
7. Штовба С. Д. Мурашині алгоритми оптимізації [Текст] / С. Д. Штовба, О. М. Рудий // Вісник ВПІ. – 2004. – № 4. – С. 62–69.
8. Чураков М. Муравьиные алгоритмы [Электронный ресурс] / М. Чураков, А. Якушев // Режим доступа: <http://rain.ifmo.ru/cat/data/theory/unsorted/ant-algo-2006/article.pdf>
9. Aarts E. H. L. Simulated annealing. Local search in combinatorial optimization [Text] / E. H. L. Aarts, J. K. Lenstra // Chichester: Wiley, 1997. – P.91–120.
10. Гладков Л. А. Генетические алгоритмы [Текст] / Л. А. Гладков, В. В. Курейчик, под ред. В. М. Курейчика. – М.: ФИЗМАТЛИТ, 2006. – 320 с.
11. Панченко Т. Б. Генетические алгоритмы [Текст]: учебно-методическое пособие / Т. Б. Панченко, Ю. Ю. Тарасевич // Астрахань: Издательский дом «Астраханский университет», 2007. – 87 с.
12. Егоров К. Генетические алгоритмы [Электронный ресурс] / К. Егоров, М. Чураков // Режим доступа: <http://yury.name/internet/03ia-seminar-note.doc>
13. Захватов М. Якість обслуговування в операторських мережах [Електронний ресурс] / Режим доступу: [http://www.opennet.ru/docs/RUS/qos\\_oper](http://www.opennet.ru/docs/RUS/qos_oper)
14. Минимальное остовное дерево. Алгоритм Крускала [Електронний ресурс] / М А Х і м а л – 2008. – Режим доступа: [http://e-maxx.ru/algo/mst\\_kruskal](http://e-maxx.ru/algo/mst_kruskal)
15. Kumar K. Implementation of Kruskal's algorithm in Java [Електронний ресурс] / Режим доступа: <http://krishami.com/2011/12/implementation-of-kruskals-algorithm-in-java>