УДК 519.17

Цегольник В. В., здобувач 3 курсу ОС «Бакалавр», спеціальність 122 Комп'ютерні науки,

Ніколюк П. К., д-р фіз.-мат. наук, професор, професор кафедри інформаційних технологій

ПОБУДОВА МОДЕЛІ НА ОСНОВІ ТЕОРІЇ ГРАФІВ ДЛЯ ОПТИМІЗАЦІЇ ВІЙСЬКОВОЇ ЛОГІСТИКИ

Донецький національний університет імені Василя Стуса, м. Вінниця

Теорія графів – це розділ математики, який вивчає властивості графів – математичних структур, що складаються з вершин і ребер, які з'єднують ці вершини.

Моделі на основі теорії графів – це моделі, які використовують графи для представлення досліджуваної системи. Ці моделі можуть бути використані для різних цілей, як-от аналіз системи, прогнозування її поведінки або оптимізація її роботи. Вони мають низку переваг перед іншими типами моделей, оскільки є потужним інструментом для аналізу складних систем, і вони можуть бути використані для вирішення широкого спектру завдань.

В умовах повномасштабної війни рівень логістики має вирішальне значення для успіху Збройних Сил України. Від того, наскільки ефективно будуть доставлені зброя, боєприпаси, продовольство та інше необхідне майно, залежить, чи зможе Україна дати відсіч російському вторгненню. Логістика включає в себе планування, організацію та управління рухом матеріалів, людей і інформації. В умовах війни це завдання значно ускладнюється. Російські війська активно обстрілюють транспортні шляхи, що ускладнює доставку вантажів. До того ж необхідно враховувати безпеку людей, які займаються логістикою. Маючи безліч понівечених доріг різного рівня складності та небезпеки, набагато краще було б мати заздалегідь обраховані шляхи із можливістю зміни стану змінних за короткі проміжки часу [1].

Розглянемо реальну ситуацію. Волонтерам необхідно привезти визначену кількість їжі та медикаментів усім найближчим підрозділам в найкоротшим шляхом у найкоротший час. Кожна вершина має певну кількість з'єднань, кожна з яких має уже обраховану вагу, що включає в себе значення складності та небезпеки. Цей підхід допоможе нам створити універсальну модель, яка може застосовуватись у будь-якій сфері, де необхідно щось оптимізувати, необов'язково військовій [2].

Постановка задачі така: представимо зважений ненапрямлений граф із заданими параметрами та створимо програму для знаходження мінімального кістяка за алгоритмом Пріма, водночас у програмі виводитимемо зображення

графу у графічному вікні перед кожною зупинкою. Таке рішення буде раціональним, якщо в майбутньому на основі цієї моделі дані будуть оновлювати в режимі реального часу. Зрозуміло, що кількість вершин буде обмеженою для зручності сприйняття в невеликому графічному вікні. Під час обходу графу також побудуємо дерево його кістяка.

Мінімальний кістяк графу – це підмножина ребер графу, яка з'єднує всі вершини графу і має мінімальну вагу. Алгоритм Пріма – це жадібний алгоритм, який працює шляхом побудови дерева з фіксованою вершиною. На кожному кроці алгоритму вибирається ребро з найменшою вагою, яке з'єднує дерево з вершиною, яка не входить у нього. Це ребро додається до дерева, і вершина, до якої воно приєднується, включається в дерево. Алгоритм продовжує працювати, поки всі вершини графу не будуть включені в дерево. На практиці саме обраний алгоритм гарантуватиме, що побудований ним граф є мінімальним деревом, оскільки на кожному кроці він вибирає ребро з найменшою вагою, яке з'єднує дерево з вершиною, яка не входить у нього [3]. Це означає, що сума ваг ребер у цьому дереві є мінімальною. Алгоритм є лінійним відносно до кількості ребер у графі, що робить його ефективним для великих графів.

Для автоматичної побудови складного графу, використаємо правила побудови, запропоновані штучним інтелектом [4]:

$$A = mulmr((1.0 – 0.01 –7 \cdot 0{,}005 – 0.05) \cdot T).$$

Матриця ваг W формується так:

1) Wt = roundm((randm(n, n) · 100) ** A), де roundm – це функція, що округлює кожен елемент матриці до найближчого цілого числа, символ «**» – поелементне множення;

2) одержується матриця B, у якій bij = 0, якщо wij = 0, bij = 1, якщо wij > 0, bij ∈ B, wij ∈ Wt;

3) одержується матриця C, у якій cij = 1, якщо bij != bji, та cij = 0 в іншому випадку;

4) одержується матриця D, у якій dij = 1, якщо bij = bji = 1, та dij = 0 в інших випадках;

5) Wt = (C + (D ** Tr)) ** Wt, де Tr – верхній трикутник матриці одиниць (без головної діагоналі), + це поелементна сума матриць;

6) одержується матриця ваг W шляхом симетризування матриці Wt.

Використовуючи процедурну мову С, напишемо програму (рис. 1).

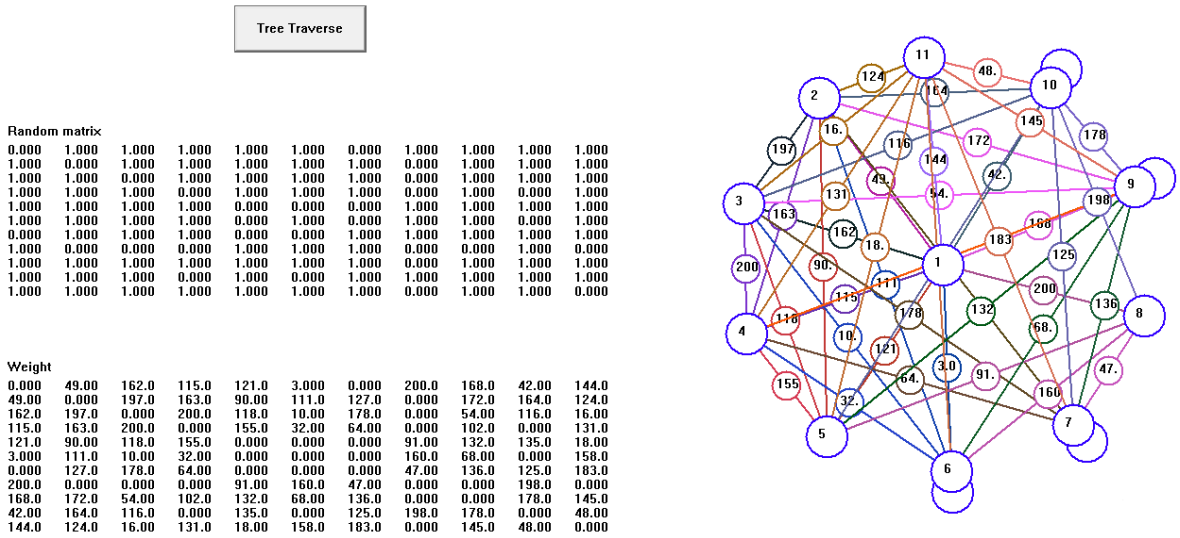

Рисунок 1. Стартовий вигляд моделі

Для зручності орієнтації в малюнку зв'язки позначаються різними кольорами, а зв'язки вершин самих до себе є лише побічним ефектом рандому і жодної цінності для досліджуваної задачі не несуть. Під час натискання кнопки «Обхід дерева», «Вершина», «Шлях, обраний алгоритмомом» та «Вага шляху» будуть підсвічуватись іншим кольором, що допоможе обрати найоптимальніший шлях (рис. 2).

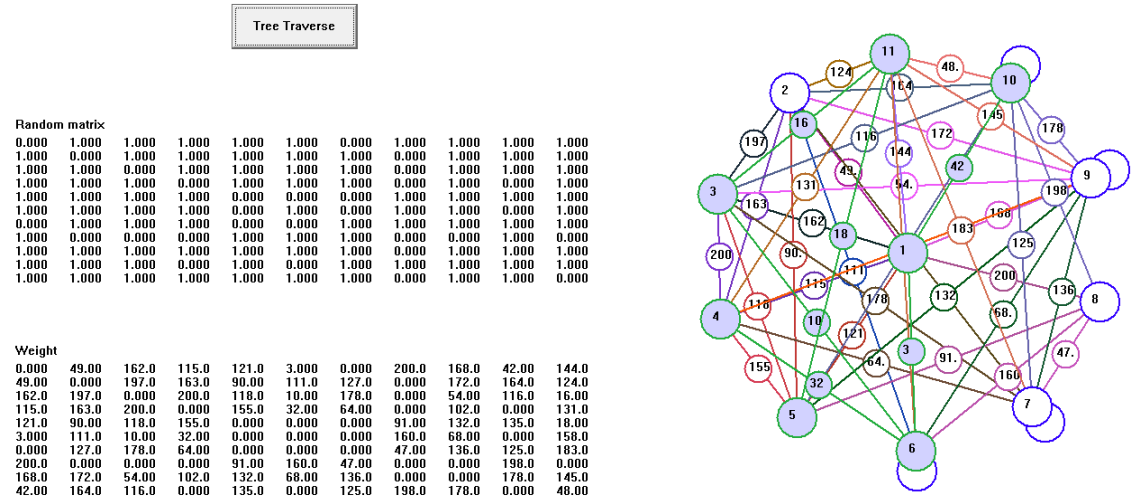

Рисунок 2. Приклад роботи моделі

Отже, розгляд витрат у логістичних системах є одним із найважливіших і найзатребуваніших питань. Становлення логістики насамперед обумовлюється прагненням до скорочення різної класифікації витрат, які пов'язані з товарорухом і утворенням результативних логістичних ланцюгів для підвищення ефективності військових операцій та зменшення ризику затримок у доставці

матеріальних ресурсів. Внаслідок виконання поставленої задачі була отримана модель, представлена графічним способом, що досліджує найкоротший шлях між усіма вершинами графу, базуючись на складності зв’язків, і може бути сміливо застосована у будь-якій сфері оптимізації шляху, насамперед військовій.

Список використаних джерел

1. Contemporary challenges in military logistics support. URL: https://securityanddefence.pl/Contemporary-challenges-in-military-logistics-support,103332,0,2.html

2. Data science in battle: Applying graph theory to the Ukraine war. URL: https://medium.com/intuitionmath/data-science-in-battle-applying-graph-theory-to-the-ukraine-war-6cda499c86fc

3. Prim’s Algorithm for Minimum Spanning Tree (MST) URL: https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/

4. C Program to Create a Random Graph Using Random Edge Generation. URL: https://www.sanfoundry.com/c-program-generate-random-undirected-graph-given-number-edges/