DOI: https://doi.org/10.36910/6775-2524-0560-2020-41-23
УДК: 517.9, 519.85, 004.9
Димова Ганна Олегівна, к.т.н., доцент,
https://orcid.org/0000-0002-5294-1756
Ларченко Оксана Валеріївна, к.с.-г.н., доцент,
https://orcid.org/0000-0001-7857-0802
Херсонський державний аграрний університет, м. Херсон, Україна

РОЗРОБКА КОМП'ЮТЕРНОЇ ПРОГРАМИ РОЗВ'ЯЗАННЯ ЗАДАЧ МЕРЕЖЕВОЇ ОПТИМІЗАЦІЇ

Димова Г.О., Ларченко О.В. Розробка комп'ютерної програми розв'язання задач мережевої оптимізації. Статтю присвячено розробці комп'ютерної програми розв'язання мережевих оптимізаційних задач для використання в навчальному процесі. Багато задач оптимізації можна сформулювати у формі тієї чи іншої задачі оптимізації на графах. У зв'язку з цим вивчення загальних властивостей задач оптимізації на графах набуває самостійного значення, а вивчення методів їх розв'язання традиційно відносять до необхідних елементів сучасної освіти, які формують алгоритмічний спосіб мислення.

Хоч загальна математична постановка задачі оптимізації на графах не дає будь-якої інформації відносно можливих методів її розв'язання, всі методи розв'язання таких задач можна умовно поділити на два класи:

– більшість відомих задач оптимізації на графах можуть бути сформульовані у формі математичної моделі цілочисельного або булева програмування. В цьому випадку вибір способу їх розв'язання повністю визначається математичними властивостями відповідної постановки задачі;

– задачі оптимізації на графах можуть бути розв'язані із застосуванням спеціальних алгоритмів, які враховують специфічні особливості тих чи інших об'єктів графів і скінченну потужність множини можливих альтернатив (задачі комбінаторної оптимізації) [9].

У статті зроблено аналіз пакетів прикладних програм для розрахунку задач мережевої оптимізації та показана необхідність розробки комп'ютерної програми для розв'язання цих задач. Приведені алгоритми розв'язання мережевих оптимізаційних задач (алгоритм Прима, алгоритм Флойда-Уоршелла та алгоритм Форда-Фалкерсона). Для реалізації програми обрано інтегроване середовище розробки програмного забезпечення Delphi 7.0. Розроблена та протестована комп'ютерна програма «Розрахунок характеристик комп'ютерних мереж».

Ключові слова: мережева оптимізація, мінімальне кістякове дерево, найкоротший шлях, максимальний потік, комп'ютерна програма.

Дымова А.О., Ларченко О.В. Разработка компьютерной программы решения задач сетевой оптимизации. Статья посвящена разработке компьютерной программы решения сетевых оптимизационных задач для использования в учебном процессе. Многие задачи оптимизации можно сформулировать в форме той или иной задачи оптимизации на графах. В связи с этим изучение общих свойств задач оптимизации на графах приобретает самостоятельное значение, а изучение методов их решения традиционно относят к необходимым элементам современного образования, которые формируют алгоритмический способ мышления.

Хотя общая математическая постановка задачи оптимизации на графах не дает какой-либо информации относительно возможных методов ее решения, все методы решения таких задач можно условно разделить на два класса:

– большинство известных задач оптимизации на графах могут быть сформулированы в форме математической модели целочисленного или булева программирования. В этом случае выбор способа их решения полностью определяется математическими свойствами соответствующей постановки задачи;

– задачи оптимизации на графах могут быть решены с применением специальных алгоритмов, учитывающих специфические особенности тех или иных объектов графов и конечную мощность множества возможных альтернатив (задачи комбинаторной оптимизации) [9].

В статье сделан анализ пакетов прикладных программ для расчета задач сетевой оптимизации и показана необходимость разработки компьютерной программы для решения этих задач. Приведены алгоритмы решения сетевых оптимизационных задач (алгоритм Прима, алгоритм Флойда-Уоршелла и алгоритм Форда-Фалкерсона). Для реализации программы выбрана интегрированная среда разработки программного обеспечения Delphi 7.0. Разработана и протестирована компьютерная программа «Расчет характеристик компьютерных сетей».

Ключевые слова: сетевая оптимизация, минимальное остовное дерево, кратчайший путь, максимальный поток, компьютерная программа.

Dymova H., Larchenko O. Development of a computer program for solving network optimization problems. The article is devoted to the development of a computer program for solving network optimization problems for use in the educational process. Many optimization problems can be formulated in the form of one or another optimization problem on graphs. In this regard, the study of the general properties of optimization problems on graphs acquires an independent significance, and the study of methods for their solution is traditionally attributed to the necessary elements of modern education that form an algorithmic way of thinking.

Although the general mathematical formulation of the optimization problem on graphs does not provide any information on possible methods for its solution, all methods for solving such problems can be conditionally divided into two classes:

– most of the known optimization problems on graphs can be formulated in the form of a mathematical model of integer or Boolean programming. In this case, the choice of the way to solve them is completely determined by the mathematical properties of the corresponding problem statement;

– optimization problems on graphs can be solved using special algorithms that take into account the specific features of certain graph objects and the finite cardinality of the set of possible alternatives (combinatorial optimization problems) [9].

The article analyzes the packages of applied programs for calculating network optimization problems and shows the need to develop a computer program to solve these problems. Algorithms for solving network optimization problems (Prim's algorithm, Floyd-Warshall's algorithm and Ford-Fulkerson's algorithm) are presented.

Delphi 7.0 integrated software development environment was chosen to implement the program. The computer program "Calculation of the characteristics of computer networks" has been developed and tested.
Keywords: network optimization, minimum spanning tree, shortest path, maximum flow, computer program.

Постановка проблеми. Задачею оптимізації в математиці називається задача про знаходження екстремуму (мінімуму або максимуму) речової функції в деякій області.

Методи, за допомогою яких розв'язуються задачі оптимізації, підрозділяються на види, які відповідають завданням, до яких вони застосовуються [1-3]:

1) локальні методи;
2) глобальні методи.

Існуючі в даний час методи пошуку можна розбити на три великі групи: детерміновані; випадкові; комбіновані.

Оскільки галузь телекомунікаційних технологій є найбільш розвинутою в теперішній час, одним з найважливіших питань є оптимізація в галузі комп'ютерних мереж і телекомунікацій. До класичних задач в галузі телекомунікацій входять [5, 7, 9, 11]:

– побудова мінімального кістякового дерева;
– пошук найкоротшого шляху в мережевому графі;
– визначення максимального потоку в мережі.

На даний час є кілька комп'ютерних програм, які забезпечують розв'язання задач мережевої оптимізації, але, з різних причин, не являються зручними у використанні. Тому стає задача розробки зручної комп'ютерної програми розрахунку характеристик комп'ютерної мережі для використання здобувачами вищої освіти в навчальному процесі.

Аналіз досліджень. Для розв'язання задач оптимізації використовується спеціальне програмне забезпечення, до якого відносяться, наприклад, засоби пошуку рішень і аналізу даних в програмі Microsoft Excel, а також спеціальний пакет прикладних програм QSB + (Quantitative Systems for Business Plus).

Пакет прикладних програм QSB + володіє дуже великими можливостями. Він включає в себе 16 програм – майже вичерпний набір запрограмованих математичних методів, необхідних для обґрунтування рішень. Ось деякі з можливих областей їх застосування:

– прогнозування;
– планування виробництва;
– управління запасами;
– планування та розміщення об'єктів;
– календарне планування та впорядкування робіт;
– планування робіт над проектами;
– задачі з ризиками;
– задачі призначення;
– задачі лінійного та нелінійного програмування;
– мережеве моделювання та ін.

Ця програма дозволяє розв'язувати задачі лінійного та динамічного програмування, а також задачі оптимізації на мережевих графах, такі, як пошук мінімального кістякового дерева, знаходження найкоротшого шляху, розрахунок максимального потоку. Проте програма має наступні недоліки: незручний інтерфейс, який був розроблений ще для операційної системи MS-DOS, відсутність графічної інтерпретації розв'язуваних задач та досить складне введення вхідних даних.

Зараз існують доволі багато програм, які реалізують розрахунок задач з теорії графів. Ще одною з таких програм є Grin-50.

Програма Grin-50, на відміну від QSB+, розв'язує лише задачі, пов'язані з оптимізацією в мережевих графах: пошук мінімального кістякового дерева, розрахунок найкоротшого шляху, максимального потоку в мережевому графі та задачу комівояжера.

В цій програмі присутні такі недоліки:

– розрахунок найкоротших шляхів проходить за алгоритмом Дейкстри, який, на жаль, не є універсальним (він розраховує найкоротший шлях тільки від заданої вершини);
– досить складно знайти вікно налаштувань з'єднання (встановлення пропускної здатності або тривалості доступу до мережевого вузла).

Виклад основного матеріалу й обґрунтування отриманих результатів. Задачі оптимізації на графах у початковій постановці використовують поняття орієнтованого або неорієнтованого графа. В додаток до загальних властивостей графа стосовно типової задачі оптимізації розглядаються спеціальні об'єкти, такі як шляхи, дерева та потоки. При цьому кожному такому об'єкту ставиться у відповідність деяке кількісне значення цільової функції, яке розраховується для конкретного графа. Таким чином, окрема задача оптимізації на графі формулюється як знаходження такого спеціального об'єкта, якому відповідає максимальне або мінімальне значення цільової функції [10].

Деревом називається зв'язна множина неорієнтованих ребер (дуг), в якому не містяться цикли. Таким чином, якщо задана множина m вузлів, з'єднаних неорієнтованими ребрами, то для побудови дерева необхідно виділити підмножину, яка складається з $(m - 1)$ дуг [2]. Інакше кажучи, кожний вузол з'єднаний з іншим вузлом одним-єдиним шляхом.

Розглянемо граф, що містить n вузлів, сукупність яких утворює множину S. Кістяковим деревом називається зв'язна множина, яка складається з $(n - 1)$ дуг (ребер) і n вузлів [2]. Із любої власної підмножини множини S може бути утворене дерево, яке, однак, може не бути кістяковим деревом вихідної мережі (якщо воно включає не всі вузли початкового графа). Припускаємо, що кожній дузі, що з'єднує вузли i та j з множини S, приписано число C_{ij}, яке називається відстанню, або вагою, дуги. Найкоротшим кістяком називається такий кістяк графу, у якого сума ваг C_{ij} всіх його дуг мінімальна [1, 2].

Таким чином, *задача про найкоротший кістяк* складається у виборі таких дуг заданого графу, що їх сумарна вартість мінімальна і для любої пари вузлів знайдеться шлях (або маршрут), що з'єднує їх. Цього можна досягнути, вибираючи дуги так, що утворене ними дерево з'єднає всі вузли заданого графу. Алгоритм починає роботу з вибору довільного вузла графу і найкоротшої дуги з множини дуг, що з'єднує цей вузол з іншими вузлами. З'єднаймо два вузла вибраною дугою. Виберемо найближчу до цих вузлів третій вузол. Додаємо цей вузол і відповідну дугу до мережі. Продовжуємо даний процес до тих пір, доки всі вузли не будуть з'єднані між собою [9].

Опишемо покроково цей алгоритм:

1) використовуючи вузли початкової мережі, визначимо наступні дві множини: S – множина з'єднаних вузлів; Ω – множина нез'єднаних вузлів. Спочатку всі вузли належать множині Ω;
2) обираємо довільний вузол з множини Ω і з'єднуємо його з найближчим сусіднім вузлом; ці два вузла переносимо з множини Ω в множину S;
3) серед всіх дуг, що з'єднують вузли з множини S з вузлами з множини Ω, вибираємо найкоротшу дугу. Кінцевий вузол цієї дуги, що лежить в множині Ω, переносимо в множину S;
4) виконуємо крок 3 доти, доки всі вузли не будуть належати множині S.

Безліч задач оптимізації пов'язані саме з *пошуком найкоротших шляхів*. Як результат – пошук шляху в сучасному світі використовується практично скрізь: від систем глобального позиціонування «GPS» і «ГЛОНАСС» для знаходження найкоротшого маршруту серед міських вулиць і шляхів між містами, у військових і цивільних системах автопілотування, в транспортно-експедиційному обслуговуванні – і аж до маршрутизації пакетів мережі в Internet. Для пошуку шляхів в графі існує досить багато алгоритмів різного ступеня швидкодії, ресурсоємності, точності і складності – починаючи з примітивних алгоритмів пошуку в ширину і в глибину, закінчуючи продуктивними і потужними алгоритмами Дейкстри, Беллмана-Форда і Флойда-Уоршелла.

Алгоритм Флойда-Уоршелла є динамічним алгоритмом для знаходження найкоротших відстаней між усіма вершинами зваженого орієнтованого графа. Більш суворе формулювання цієї задачі наступне: є орієнтований граф $G = (V, E)$ кожній дузі $v \to w$ цього графа співставлена невід'ємна вартість $C[v, w]$. Загальна задача знаходження найкоротших шляхів полягає в знаходженні для кожної впорядкованої пари вершин v, w будь-якого шляху від вершини v в вершини w, довжина якого мінімальна серед усіх можливих шляхів від v до w [5, 6].

Для визначеності припустимо, що вершини графа послідовно пронумеровані від 1 до n. Алгоритм Флойда-Уоршелла використовує матрицю **D** розміру $n \times n$, в якій обчислюються довжини найкоротших шляхів. Спочатку $D[i, j] = C[i, j]$ для всіх $i \neq j$. Якщо дуга $i \to j$ відсутня, то $C[i, j] = \infty$. $D[i, i] = 0$.

Над матрицею **D** виконується n ітерацій. Після k-ї ітерації $D[i, j]$ містить значення найменшої довжини шляхів з вершини i в вершину j, які не проходять через вершини з номером, більшим k. Іншими словами, між кінцевими вершинами шляху i та j можуть знаходитися тільки вершини, номери яких менше або дорівнюють k. На k-й ітерації для обчислення матриці **D** застосовується наступна формула:

$$D_k[i, j] = \min(D_{k-1}[i, j], D_{k-1}[i, k] + D_{k-1}[k, j]) \tag{1}$$

Нижній індекс k позначає значення матриці **D** після k-ї ітерації, але це не означає, що існує n різних матриць, цей індекс використовується для скорочення запису. Для обчислення $D_k[i, j]$ проводиться порівняння величини $D_{k-1}[i, j]$ (тобто вартість шляху від вершини i до вершини j без участі вершини k чи іншої вершини з більш високим номером) з величиною $D_{k-1}[i, k] + D_{k-1}[k, j]$ (вартість шляху від вершини i до вершини k плюс вартість шляху від вершини k до вершини j). Якщо шлях через вершину k коротше, ніж $D_{k-1}[i, j]$, то величина $D_k[i, j]$ змінюється.

Методи розв'язання *задачі про максимальний потік* у графі застосовуються на транспортних, комунікаційних, електричних мережах, при моделюванні різних процесів фізики і хімії, в деяких операціях над матрицями, для розв’язання родинних задач теорії графів, і навіть для пошуку Web-груп в WWW. Найбільш реалізованим в безлічі застосувань є алгоритм Форда-Фалкерсона, розрахунок за яким сучасні комп’ютери проводять за долі секунди.

Дано граф $G = \{A, B\}$, де A − множина вузлів, B − множина дуг [8, 9]. Граф G має одне джерело s і один стік t. Дуги b_{ij} мають обмежену пропускну здібність.

$$f_{ij} \le U_{ij}\,, \tag{2}$$

де f_{ij} – можливий потік через гілку b_{ij};

U_{ij} – задане значення пропускної здібності для гілки b_{ij}.

Величина потоку з джерела не обмежена і в кожному проміжному вузлі виконується умова збереження потоку.

$$\sum_i f_{ij} - \sum_i f_{ji} = 0, \quad i \ne s, \quad i \ne t\,, \tag{3}$$

Задача складається у визначенні дугових потоків f_{ij} таких, щоб загальний потік з джерела s в стік t був максимальним:

$$\sum_{i \ne j} f_{ij} \to \max\,, \tag{4}$$

Задача розв’язується за допомогою ітераційної процедури розстановки поміток вузлів. Кожна помітка вказує величину потоку та його джерело, і може бути як позитивною, так і негативною.

Рис. 1 – Помітки вузлів: q_i , q_j – кількість одиниць потоку

Позитивна помітка збільшує потік по гілці b_{ij}, але так, щоб сумарний потік не перевищував U_{ij}; негативна помітка зменшує потік по гілці b_{ij}, але так, щоб він не став від’ємним.

Вузли помічаються послідовно від 1 до n.

Якщо вузол a_i помічений, то a_j можна помітити з нього по прямій гілці на величину не більшу, за $q_i = \min\left[q_i; U_{ij} - f_{ij}\right]$, а по оберненій гілці на величину $q_j = \min\left[q_i; f_{ij}\right]$.

На кожній ітерації джерело помічається міткою $[\infty; -]$ і розставляються мітки інших вузлів по одному із шляхів з джерела до стоку. Коли доходить черга до стоку, йому приписується мітка $[+U_n; k]$. Тобто потік в мережі можна збільшити на U_n; після чого всім дугам шляху, що розглядається, приписується $f_{ij} = U_n$.

Потім всі мітки стираються і переходять до наступної ітерації. Алгоритм закінчує роботу, якщо жоден вузол мережі вже не може бути поміченим.

Програмний додаток розв’язування задач мережевої оптимізації реалізований в середовищі візуального програмування Delphi 7.0 фірми Borland [4].

Для розглянутих задач мережевої оптимізації розроблена блок-схема алгоритму (рис. 2), на підставі якої реалізована комп’ютерна програма для розв’язання задач.

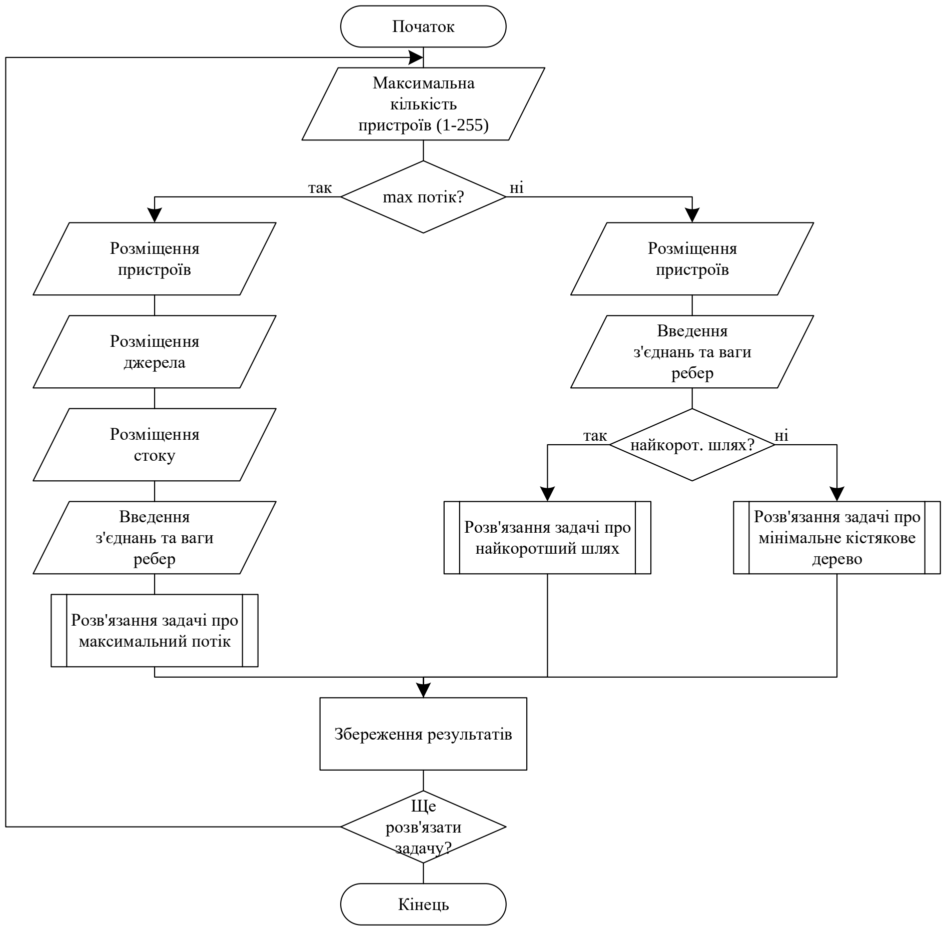

Рис. 2 – Блок-схема алгоритму комп'ютерної програми розв'язання задач мережевої оптимізації

При запуску програми на екран виводиться меню (рис. 3), в якому пропонується 2 варіанти для вибору задачі: побудова мінімального кістякового дерева і розрахунок найкоротших шляхів, розрахунок максимального потоку в мережі.

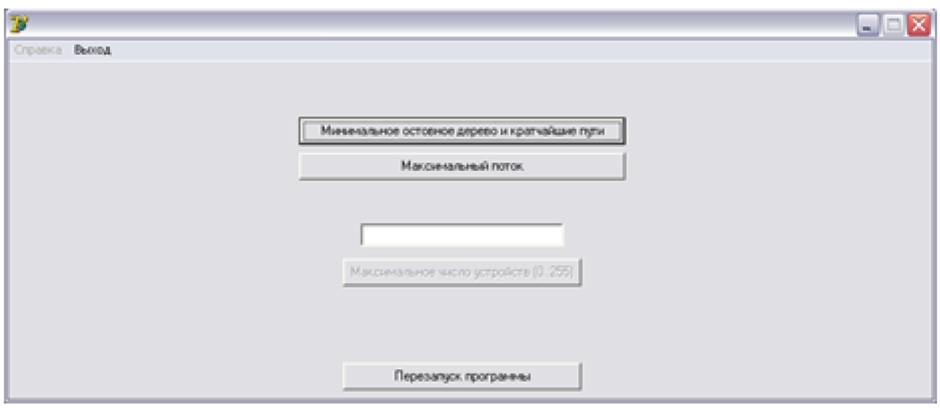

Рис. 3 – Головне меню програми

Необхідно обрати один з пунктів, далі слід ввести до поля вводу максимальне число пристроїв – вершин (для розрахунку максимального потоку – за виключенням джерела та стоку).

При виборі пункту «*Минимальное остовное дерево и кратчайшие пути*» (рис. 4), на екрані з'являється робоче поле та панель, на якій розташовані кнопки розміщення пристроїв та ребер. Для коректної побудови розв'язання задач необхідно, щоб побудований граф був зв'язним, тобто, щоб будь-які дві вершини були з'єднані послідовністю ребер.

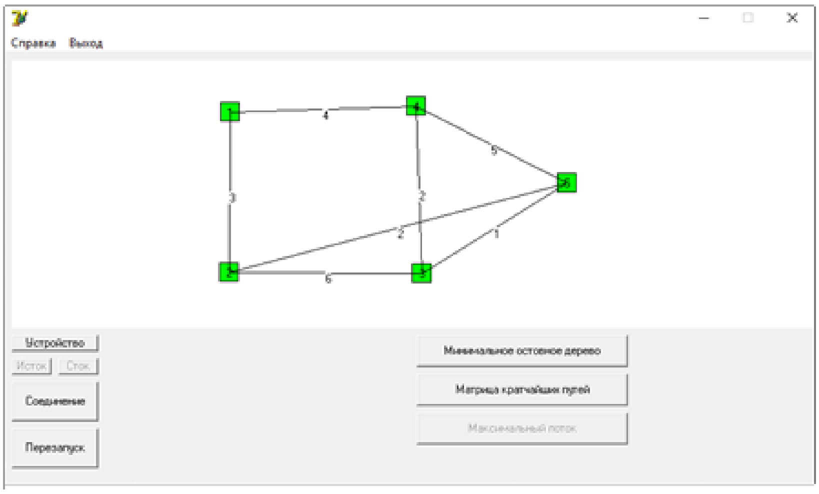

Рис. 4 – Інтерфейс вікна побудови мінімального кістякового дерева та розрахунку найкоротших шляхів з вхідними даними

Отже, встановивши вершини та сполучення між ними, натискаємо на кнопку «*Минимальное остовное дерево*» та отримуємо результат, виділений червоним (рис. 5).

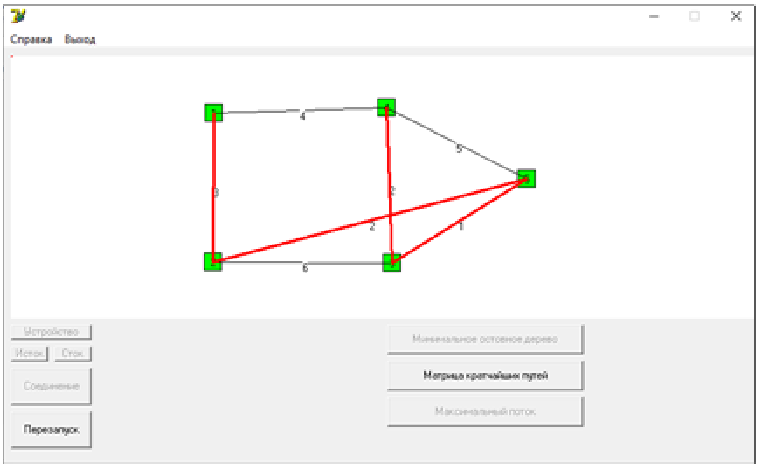

Рис. 5 – Результат побудови мінімального кістякового дерева

Також можна подивитись матрицю найкоротших шляхів за допомогою кнопки «*Матрица кратчайших путей*» (рис. 6).

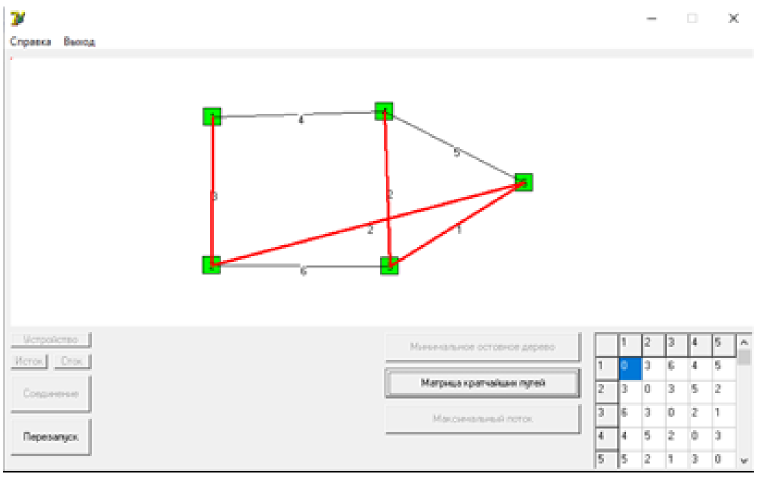

Рис. 4 – Результат розрахунку найкоротших відстаней

При виборі розрахунку максимального потоку, будується орієнтований граф (рис. 5). Обов’язково необхідно встановити джерело (може бути лише початком ребра) та стік (може бути лише кінцем ребра).

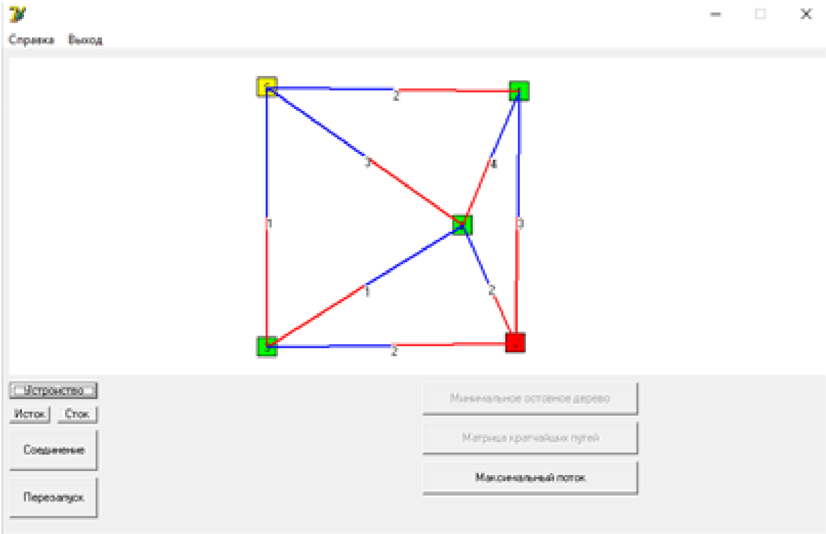

Рис. 5 – Інтерфейс вікна розрахунку максимального потоку з вхідними даними

За допомогою кнопки «*Максимальный поток*» отримаємо значення максимального потока від джерела до стоку. (рис. 6)

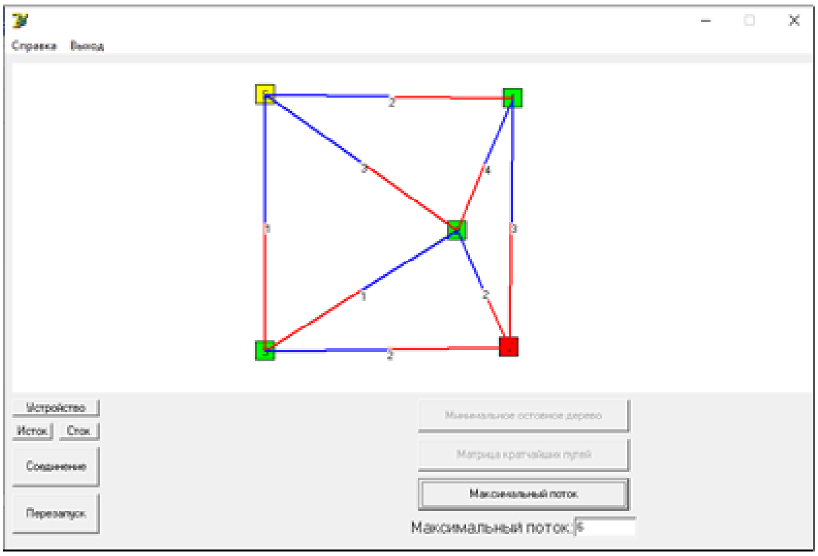

Рис. 6 – Результат розрахунку максимального потоку

Для користування програмним додатком для розв’язання задач мережевої оптимізації були розроблені інструкції (рис. 7, 8).

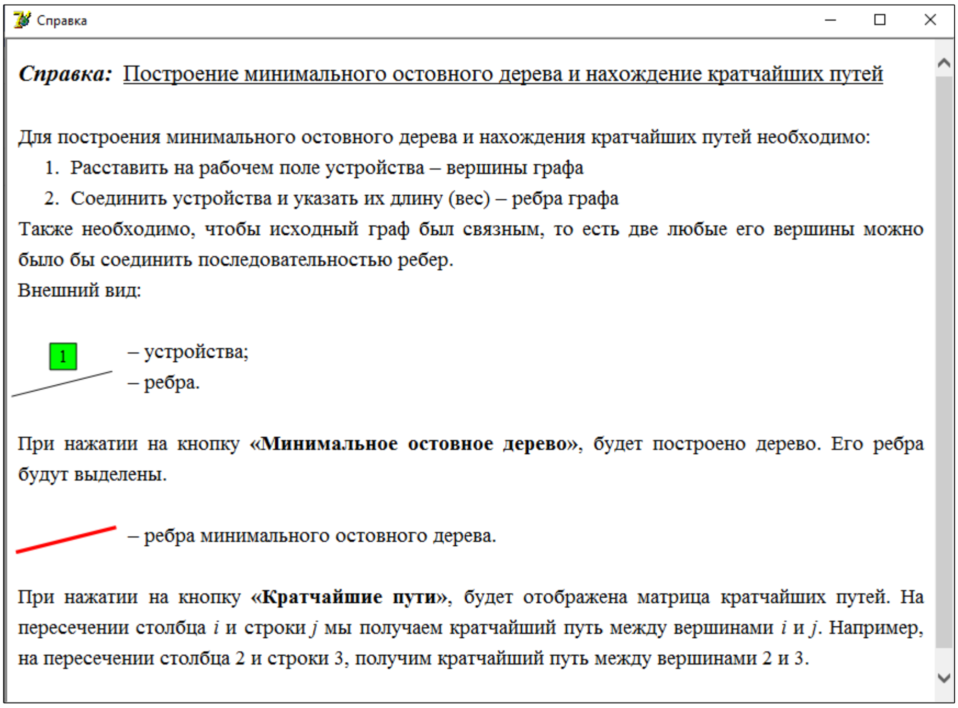

Рис. 7 – Інтерфейс вікна довідки для побудови мінімального кістякового дерева та розрахунку найкоротших шляхів

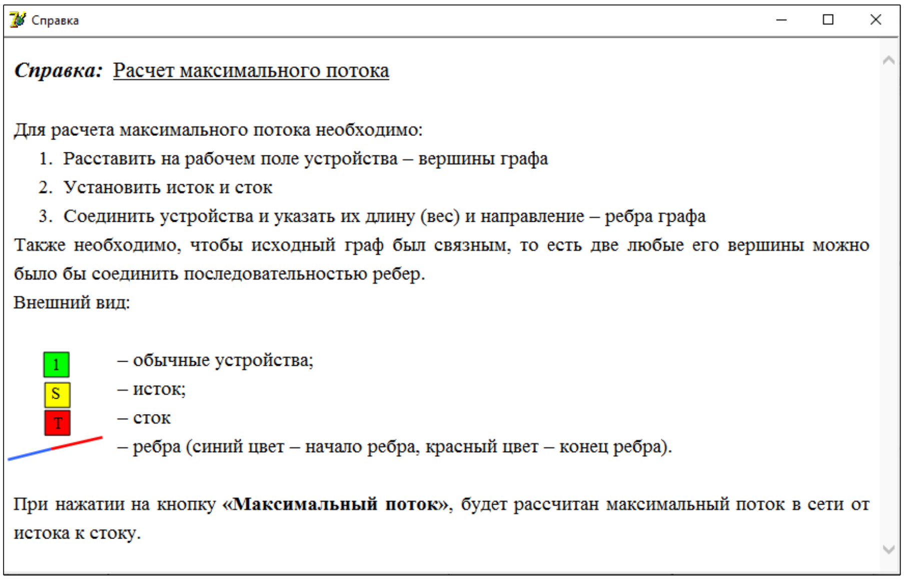

Рис. 8 – Інтерфейс вікна довідки для розрахунку максимального потоку

Висновки та перспективи подальшого дослідження. Стаття присвячена розробці комп’ютерної програми для розв’язування мережевих оптимізаційних задач з метою необхідності перевірки результатів розв’язання задач мережевої оптимізації для дисциплін «Оптимізаційні методи та моделі», «Комп’ютерні мережі» та «Мережеві технології». Проведений аналіз пакетів прикладних програм для розрахунку задач мережевої оптимізації. Доказана необхідність розробки програмного додатку для перевірки розрахунків характеристик комп’ютерної мережі. Розглянуті алгоритми розв’язання оптимізаційних задач з теорії графів таких як, задача про мінімальне кістякове дерево за допомогою алгоритма Прима, задача про найкоротший шлях за допомогою алгоритма Флойда-Уоршелла, задача про максимальний потік за допомогою алгоритма Форда-Фалкерсона. Розроблений алгоритм роботи комп’ютерної програми для розв’язання цих задач. Реалізований програмний додаток та зроблене його тестування. Комп’ютерна програма є зручною, простою у використанні та побудована на універсальних алгоритмах розрахунку характеристик комп’ютерних мереж.

Список бібліографічного опису

1. Акимов О.Е. Дискретная математика. Логика, группы, графы. М.: Лаборатория базовых знаний, 2001. 376с.
2. Андерсон Дж. Дискретная математика и комбінаторика. М.: Издательский дом «Вильямс», 2003. 960 с.
3. Горбатов В.А. Фундаментальные основы дискретной математики. М.: Наука. Физматлит, 2000. 544с.
4. Осипов Д. Delphi. Профессиональное программирование М.:Символ-Плюс, 2006 1056 с.
5. Дискретная математика и математические вопросы кибернетики. Ред. С.В. Яблонского, А.А. Лупанова. М.: Наука, 1974. 278 с.
6. Капітонова Ю.В. и др. Основи дискретної математики. К.: Наукова думка, 2002. 579с.
7. Математичні основи теорій телекомунікаційних систем. В.В. Поповський, С.О. Сабурова, В.Ф. Олійник, Ю.І. Лосєв, Д.В. Агеєв та ін.: Ред. В.В. Поповського. Харків: ТОВ «Компанія СМІТ», 2006. 564 с.
8. Мельников В.Н. Логические задачи. К.; Одесса: Выща шк., 1989. 344 с.
9. Новиков Ф.А. Дискретная математика для программистов. СПб.: Питер, 2002. 304 с.
10. Оре О. Теория графов. М. Мир, 1968. 426 с.
11. Стеклов В.К., Кільчицький Є.В. Основи управління мережами і послугами телекомунікацій: Підруч. для вищ. навч. закл. за напрямком «Телекомунікації». Ред. В.К. Стеклова. К.: Техніка, 2002. 438 с.

References

1. Akimov O.E. Discrete mathematics. Logic, groups, graphs. M .: Laboratory of basic knowledge, 2001.376 p.
2. Anderson J. Discrete mathematics and combinatorics. M .: Publishing house "Williams", 2003. 960 p.
3. Gorbatov V.A. Fundamental principles of discrete mathematics. M .: Science. Fizmatlit, 2000.544 p.
4. Osipov D. Delphi. Professional programming. M.: Symbol-Plus, 2006 1056 p.
5. Discrete mathematics and mathematical problems of cybernetics. Ed. S.V. Yablonsky, A.A. Lupanov. M.: Science, 1974. 278 p.
6. Kapitonova Yu.V. etc. Fundamentals of discrete mathematics. K .: Naukova dumka, 2002. 579p.
7. Mathematical bases of theories of telecommunication systems. V.V. Popovsky, S.O. Saburova, V.F. Oliynyk, Yu.I. Losev, DV Ageev et al . Ed. V.V. Popovsky. Kharkiv: SMIT Company LLC, 2006. 564 p.
8. Melnikov V.N. Logical problems. K.; Odessa: Vyshcha school, 1989. 344 p.
9. Novikov F.A. Discrete mathematics for programmers. SPb.: Peter, 2002. 304 p.
10. Ore O. Graph Theory. M .: Mir, 1968.426 p.
11. Steklov V.K., Kilchytsky E.V. Fundamentals of management of telecommunications networks and services: Textbook. for higher textbook lock in the direction of "Telecommunications". Ed. V.K. Steklova. K .: Tehnika, 2002. 438 p.