

Міністерство освіти і науки України
Харківський національний автомобільно-дорожній університет
Харківський національний університет радіоелектроніки

Кваліфікаційна наукова
робота на правах рукопису

МАЦІЙ ОЛЬГА БОРИСІВНА

УДК 519.161

ДИСЕРТАЦІЯ

**МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА МЕТОДИ ОПТИМІЗАЦІЇ
ЗАМКНЕНИХ МАРШРУТІВ В ЗАДАЧАХ ТРАНСПОРТНОГО ТИПУ**

01.05.02 – математичне моделювання та обчислювальні методи
технічні науки

Подається на здобуття вченого ступеня кандидата наук

Дисертація містить результати власних досліджень. Використання ідей,
результатів і текстів інших авторів мають посилання на відповідне джерело

Підпис Мацій О. Б.

Науковий керівник

Панішев Анатолій Васильович, доктор технічних наук, професор

Цей примірник дисертаційної роботи ідентичний за змістом з іншими, поданими
до спеціалізованої вченої ради Д.64.052.02

Вчений секретар спецради Д64.052.02

Підпис
Печатка

Л.В. Колесник

Харків – 2018

АНОТАЦІЯ

Мацуй О. Б. Математичне моделювання та методи оптимізації замкнених маршрутів в задачах транспортного типу. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня кандидата технічних наук за фахом 01.05.02 «Математичне моделювання і обчислювальні методи». Харківський національний автомобільно-дорожній університет, Харківський національний університет радіоелектроніки, Міністерство освіти і науки, Україна, Харків, 2018.

Дисертаційна робота присвячена дослідженню математичних моделей оптимізації транспортних перевезень, розробленню нових і вдосконаленню відомих методів та алгоритмів комбінаторної оптимізації, які застосовуються в транспортній логістиці для побудови замкнених маршрутів.

Розв'язання таких задач є важливим і набуває особливого значення в екстремальних умовах, коли виникає необхідність швидкої перебудови маршрутів руху при зміні зовнішніх умов.

Серед оптимізаційних задач транспортної логістики задача маршрутизації є однією з найбільш важливих. На сьогодні відома ціла низка її узагальнень, у яких враховуються різні обмеження та критерії. Зазвичай, для більшості таких узагальнень пошук розв'язку, який задовольняв би за точністю та швидкодією, виявляється занадто трудомістким.

Запропоновано модифікацію методу Літтла для знаходження оптимального маршруту в транспортній мережі. Для класу задач про паросполучення, наведених задачею про призначення, задачею знаходження 2-фактора мінімальної ваги і задачею про зважене паросполучення, запропоновано методи розв'язання, розроблені за єдиною схемою, яка знижує трудомісткість відомих методів.

Особливість модифікації полягає в тому, що вперше оцінка знизу шуканого значення цільового функціонала визначається в результаті розв'язання

задачі знаходження 2-фактора мінімальної ваги. Застосування вибраної оцінки обмежує згори дерево перебору та істотно скорочує час пошуку розв'язку задач транспортно-го типу.

У вступі обгрунтовано актуальність теми дослідження, зокрема теоретичну і практичну важливість математичного моделювання та методів оптимізації замкнених маршрутів в задачах транспортно-го типу; сформульовано мету, об'єкт, предмет дослідження, визначено наукову новизну та наведено відомості про апробацію отриманих результатів, включаючи зв'язок з науковими програмами, планами.

У першому розділі наведено аналіз сучасного стану проблем транспортної логістики, пов'язаних із побудовою і дослідженням моделей маршрутизації.

Задачі маршрутизації на автомобільному транспорті та методи їх розв'язання вивчаються в рамках наукового напрямку – транспортної логістики, математичний апарат якої представлений теорією графів і дослідженням операцій.

Ключовою задачею маршрутизації є задача VRP (VRP – Vehicle Routing Problem), яка полягає в тому, що споживачеві i , $i = \overline{1, n}$, треба доставити однорідний вантаж у кількості d_i з бази $n + 1$, використовуючи транспортні засоби місткості S . Кожному споживачеві вантаж доставляється одним із транспортних засобів, який повертається на базу. Задана вартість перевезення $d_{ij} = d_{ji}$. Розв'язком VRP є k послідовностей σ_k розвезення вантажів, які мінімізують $\sum_{k=1}^k \sum_{i,j \in \sigma_k} d_{ij}$ при обмеженні на місткість транспортних засобів $\sum_{i \in \sigma_k} d_i \leq S$.

Виділені окремі випадки VRP, подані базовими, ефективно вирішуваними задачами про паросполучення: задачею про призначення, задачею знаходження 2-фактора мінімальної ваги, задачею про зважене паросполучення. До задачі про зважене паросполучення зводяться 2-VRP і VRP, в якій $S/3 \leq d_i \leq S, i = \overline{1, n}$, а задача про призначення і задача знаходження 2-фактора мінімальної ваги, окрім

логістичних формулювань, можуть служити релаксаціями, що прискорюють пошук розв'язку задач транспортно-го типу в методі Літтла. Усі відомі алгоритми задачі про паросполучення характеризуються непротими і різними способами побудови. Серед них алгоритми задачі про призначення і задачі знаходження 2-фактора мінімальної ваги, які можуть збільшити точність нижніх меж у методі Літтла не зменшують його трудомісткості.

Із наведеного аналізу виходить необхідність розв'язання базових задач про паросполучення єдиним методом, що прискорює побудову і знижує вартість маршрутів в задачах транспортно-го типу.

Другий розділ містить опис методу, розробленого для розв'язання задачі про призначення і задачі про зважене паросполучення.

Розв'язання задачі про призначення наведеним методом знаходиться на основі її формулювання в такому вигляді: для матриці вартостей $C = [c_{ij}]_n$, де $c_{ij} \in R_0^+$ або $c_{ij} = \infty$, R_0^+ – множина невід'ємних дійсних чисел, знайти

$$C(\sigma) = \min_{\pi} \sum_{i=1}^n c_{\pi[i]}.$$

Тут $\pi = (\pi[1], \pi[2], \dots, \pi[n])$ – (перестановка множини $\{1, 2, \dots, n\}$ номерів стовпців матриці C).

Рекурентна побудова розв'язання задачі про призначення $\sigma = \pi_n$ виконується засобами теорії паросполучень для дводольних графів.

Метод розв'язання задачі про призначення поданий покроковим знаходженням $\pi_1, \pi_2, \dots, \pi_n = \sigma$ з трудомісткістю $O(n^3)$, не більшою, ніж у найбільш поширеного угорського методу. Він складається з однотипних дій, що спрощують програмну реалізацію і забезпечують економію пам'яті на великій розмірності вхідних даних.

Викладена схема побудови σ застосовується для розв'язання задачі про зважене паросполучення. У задачі про зважене паросполучення заданий довільний граф $H = (V, U)$, $|V| = n$, в якому ребро $\{v_i, v_j\} \in U$ має вагу $c_{ij} \in R_0^+$, $i, j = \overline{1, n}$. Потрібно знайти в H максимальне паросполучення M_{opt} з мінімальною сумою вагів ребер $C(M_{opt})$.

З аналізу покрокового подання методу розв'язання задачі про зважене паросполучення випливає, що після впорядкування значень елементів над головною діагоналлю матриці $C = [c_{ij}]_n$ і графа H M_{opt} коректно знаходиться за час $O(n^3)$, менший, ніж у відомого алгоритму Едмондса.

У *третьому розділі* викладено модифікований метод розв'язання гамільтонової задачі комівояжера, яка є задачею комівояжера для довільного зваженого графа $H = (V, U)$. До гамільтонової задачі комівояжера зводиться ряд задач класу VRP.

У гамільтоновій задачі комівояжера задано граф $H = (V, U)$ з вагами ребер $\{v_i, v_j\} \in U$, які дорівнюють $c_{ij} \in R_0^+$, $v_i, v_j \in V$. Маршрутом комівояжера, або обходом називається простий основний цикл у H . Потрібно знайти обхід τ^* з мінімальною сумою ваг ребер $C(\tau^*)$ або встановити, що множина обходів графа H є порожньою.

Метод поданий вдосконаленою версією класичного методу Літтла в якій для підвищення точності нижніх оцінок вартості обходів застосовують розв'язок задачі 2-фактор.

2-фактором довільного графа називається об'єднання непересічних за вершинами циклів. Задача знаходження 2-фактора мінімальної ваги полягає в тому, що в зваженому графі $H = (V, U)$ потрібно знайти 2-фактор, сума ваг ребер якого була б найменшою.

У запропонованій модифікації методу Літтла висота дерева перебору

обмежена згори величиною $\lfloor n/3 \rfloor - 1$. Оскільки число всіх вершин в дереві не більше $2^u - 1$, то трудомісткість розв'язання гамільтонової задачі комівояжера оцінюється величиною $O(n^3 2^{\lfloor n/3 \rfloor})$.

У четвертому розділі описано процес проектування і розроблення програмного продукту, що дозволяє за вхідними даними будувати маршрути комівояжера на основі ідей, викладених у даній дисертаційній роботі.

Обчислювальний експеримент проведено на випадково згенерованих наборах даних, які подавалися на вхід алгоритмам розв'язання задачі про призначення і методу розв'язання задачі знаходження 2-фактора мінімальної ваги.

Досліджено час роботи відомих методів розв'язання задачі про призначення (методу потенціалів і угорського методу) і рекурентного методу, запропонованого в дисертаційній роботі.

Генерувалося по 100 випадкових матриць вартостей розмірності від 10 до 490 з кроком 20. Значення, що заповнюють матрицю, вибирались з діапазону від 10 до 100. Кожну серію задач вирішували трьома методами.

Досліджували залежність часу розв'язання задачі кожним із методів від розміру вхідних даних. Для зменшення похибки підрахунку часу засікався загальний час розв'язання серії з 100 завдань кожним із методів і отриманий час ділився на 100.

Досліджено час роботи методу методу розв'язання задачі знаходження 2-фактора мінімальної ваги для отримання 2-фактора мінімальної ваги. Використовувалися ті ж параметри експерименту і вхідні дані, на яких досліджувався час роботи методів розв'язання задачі про призначення.

Досліджувався час розв'язання задачі комівояжера модифікованим методом Літтла, в якому використані різні релаксації для отримання нижніх оцінок вартості маршрутів комівояжера. Генерувалося по 100 випадкових матриць вартостей розмірністю від 10 до 170 з кроком 5. Діапазон значень, що заповнюють матриці, становив від 10 до 100. Для кожної розмірності

вирішувалося по 100 задач трьома модифікаціями методу Літтла, в яких використано для отримання нижньої межі методи – розв’язання задачі знаходження 2-фактора мінімальної ваги, рекурентний метод розв’язання задачі про призначення і угорський алгоритм.

У результаті порівняння методів розв’язання задачі про призначення виявлено, що кращі характеристики показав рекурентний метод розв’язання задачі про призначення, наведений в другій главі дисертаційної роботи.

У ході обчислювального експерименту вирішувалися серії по 100 задач. Кожна серія розв’язана модифікованим методом Літтла, в якому використовувалися різні способи обчислення нижніх оцінок.

Результати обчислювального експерименту показують, що найменший час розв’язання задачі комівояжера демонструє модифікація методу Літтла, що використовує як релаксацію, метод розв’язання задачі знаходження 2-фактора мінімальної ваги. Другий результат має модифікація з розв’язанням задачі про призначення рекурентним методом.

Практична значущість отриманих результатів дисертаційної роботи визначається тим, що розроблені моделі і методи оптимізації маршрутів орієнтовані на вдосконалення організації перевезень у реальному масштабі часу і в реальних умовах руху транспортних засобів.

Зв’язок роботи з науковими програмами, планами, темами

Дисертаційні дослідження здійснювали відповідно до плану науково-технічних робіт Харківського національного автомобільно-дорожнього університету. Наукові результати, на яких базується дисертаційна робота, отримані при виконанні планових бюджетних програм, замовлень:

– проект на замовлення Державного агентства з питань науки, інновацій та інформатизації України №ДЗ/464-2011 що до теми «Розроблення та впровадження інформаційно-комунікаційної технології руху наземного транспорту великих міст»;

– НДР за договором № Ф62/106-2015 «Розроблення та впровадження новітніх інформаційно-комунікаційних технологій для мехатронних і

навігаційних систем броньованих колісних і гусеничних машин»;

– НДР за договором № 01-53-16 «Забезпечення конкурентоспроможності підприємств транспортної галузі України за рахунок підвищення ефективності віртуального управління процесами транспортного обслуговування»;

– НДР за галузевим замовленням МОН України за договором № 0117U002405 «Розроблення інформаційно-комунікаційної технології інтелектуального керування наземними безпілотними багатоцільовими транспортними засобами».

Результати дослідження впроваджено в навчальний процес кафедри комп'ютерних технологій і мехатроніки ХНАДУ в дисциплінах «Системний аналіз», «Математичні методи дослідження операцій».

За результатами дисертаційного дослідження опубліковано 18 наукових праць, у тому числі 6 статей у наукових фахових виданнях України з технічних наук, 3 статті – в міжнародних фахових виданнях (включені до наукометричної бази SCOPUS), і 9 публікації у тезах і матеріалах міжнародних конференцій і семінарів.

Ключові слова: гамільтонова задача комівояжера, задача про призначення, 2-фактор, задача про зважене паросполучення.

Список публікацій здобувача

1. Геращенко И. В., Маций О. Б., Панишев А. В. Полиномиальное преобразование в приближенных алгоритмах решения задач типа коммивояжера // Радиоэлектроника и информатика. 2007. №1. С. 45-49.

2. Левченко А. Ю., Маций О. Б., Панишев А. В. Оптимізація замкнених маршрутів на транспортній мережі // Штучний інтелект. 2010. № 1. С 43-49.

3. Маций О. Б. Повышение точности симметричной задачи коммивояжера большой размерности // Вестник Харьковского национального автомобильно-дорожного университета. 2011. № 55. С 100-102.

4. Маций О. Б. Удосконалення угорського методу рішення задачі про призначення // Открытые информационные и компьютерные технологии. 2014. № 66. С. 166-171.

5. Matsiy O. B., Morozov A. V., Panishev A. V. The Recurrent Method to Solve the Assignment Problem // Cybernetics and Systems Analysis. 2015. Vol. 51. No 6. Pp. 939-946.

6. Кушнір Н. А., Маций О. Б., Скачков В. А. Метод найкоротших збільшуючих шляхів у задачах про паросполучення // Вісник Житомирського державного технологічного університету. 2016. № 3(74). С. 101-111.

7. Matsiy O. B., Morozov A. V., Panishev A. V. Fast Algorithm to Find 2-Factor of Minimum Weight // Cybernetics and Systems Analysis. 2016. Vol. 32. No 3. Pp 464-474.

8. Matsiy O. B., Morozov A. V., Panishev A. V. A Recurrent Algorithm to Solve the Weighted Matching Problem // Cybernetics and Systems Analysis. 2016. Vol. 52. No 5. Pp 748-753.

9. Маций О. Б. Класс задач маршрутизации, сводимых к задаче коммивояжера // Автомобіль і Електроніка. Сучасні Технології. 2017. № 12. С. 167-170.

10. Никонов О. Я., Маций О. Б. Подход к решению симметричной задачи коммивояжера // Сучасні проблеми математичного моделювання та обчислювальних методів: матеріали Всеукраїнської наукової конференції. Рівне: РДГУ-НУВГ, 2013. С. 192.

11. Маций О. Б. Про один підхід до рішення задачі комівояжера з симетричною матрицею відстаней // Інформаційні технології і мехатроніка: матеріали Всеукраїнської науково-практичної конференції. Харків: ХНАДУ, 2014. С. 83.

12. Маций О. Б., Шаригін Г. М. Сучасні аспекти моделювання маршрутів перевезення // Інформаційні технології і мехатроніка: матеріали Всеукраїнської науково-практичної конференції. Харків: ХНАДУ, 2014. С. 160-161.

13. Маций О. Б. Рекуррентный метод решения задачи о назначении // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я (MicroCAD-2014): матеріали XXII міжнародної науково-практичної конференції. Харків: НТУ ХПІ, 2014. С. 249.

14. Маций О. Б. Метод ускоренного поиска кратчайших гамильтоновых маршрутов // Обчислювальний інтелект (результати, проблеми, перспективи): матеріали III міжнародної науково-практичної конференції. Київ-Черкаси: М-во освіти і науки України, Київ. нац. ун-т імені Тараса Шевченка та [ін.]; наук. ред. В.Є. Снитюк. – Черкаси: видавець Чабаненко Ю, 2015. С. 312.

15. Маций О. Б. Алгоритм решения задачи о взвешенном паросочетании // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я (MicroCAD-2015): матеріали XXIII міжнародної науково-практичної конференції. Харків: НТУ ХПІ, 2015. С. 231.

16. Маций О. Б. Схема поиска гамильтонова маршрута минимальной стоимости на транспортной сети // Інформаційні технології і мехатроніка: освіта, наука та працевлаштування: матеріали міжнародної науково-практичної конференції. Харків: ХНАДУ, 2016. С. 76-78.

17. Маций О. Б. Поліноміальне перетворення наближених алгоритмів в рішенні задач типу комівояжера // Інформаційні технології і мехатроніка: освіта, наука та працевлаштування: матеріали міжнародної науково-практичної конференції Харків: ХНАДУ, 2017. С. 54-57.

18. Панишев А. В., Маций О. Б. Базовые задачи оптимизации замкнутых маршрутов и способы усовершенствования методов их решения // Інноваційні технології в науці та освіті. Європейський досвід: матеріали міжнародної конференції. Відень: Издательство, 2017. С. 315-319.

ABSTRACT

Matsiy O. B. Mathematical modeling and optimization methods of closed routes in transport type problems. – Qualifying scientific work as a manuscript.

Thesis for the degree of technical sciences candidate in specialty 01.05.02 – «Mathematical Modeling and Computational Methods». Kharkov National Automobile and Highway University, Kharkov National University Of Radio Electronics, Ministry of Education and Science, Ukraine, Kharkov, 2018.

The target of this thesis work is a research of algorithmic properties for routing problems and based on findings further development of the known Little's method, used for solving a Traveling Salesman Problem. The speeding of method is achieved as a result of the more accurate definition of the lower bounds for optimal solution cost, defined by effective matching algorithms in bipartite and arbitrary graphs. For class of matching tasks, represented by the assignment problem, the 2-factor problem and the weighted matching problem, it was suggested to use a developed solution methods, where the same procedures are used, reducing, therefore, a complexity of the known methods. In question in this thesis work is an upgraded version of Little's algorithm for finding route of a traveling salesman in a transport network, described by an arbitrary graph.

The peculiarity of the upgraded version is that for the first time a lower bound estimate of target value for functional objective is determined because of solving the 2-factor problem. The use of chosen evaluation places an upper limit onto search tree and significantly shortens time for finding solutions to problems of the traveling salesman type.

Section "*Introduction*" gives the reasoning for topic of research, in particular for theoretical and the practical importance of improving the existing methods of the matching theory in solving a wide class of problems; for optimizing routes on the transport network; formulates goal, target and subject of research; defines its scientific novelty and provides information about approbation of the obtained results, including its association with scientific research programs and curriculums.

The *First section* gives an analysis of the current state of transport logistics problems, associated with creation and study of routing models.

The key task of routing is the task of VRP (VRP – Vehicle Routing Problem), which is to deliver to a consumer i , $i = \overline{1, n}$ similar types of merchandize in quantity of

d_i , from depot $n+1$, using vehicles of S capacity. Each consumer gets the cargo, delivered by one vehicle, which is then returning to the depot. The cost of the transportation is defined by the formula $d_{ij} = d_{ji}, i, j \in \{1, 2, \dots, n, n+1\}$. The VRP solution is represented by k sequences of σ_k cargo deliveries, delivering a minimum of $\sum_{k=1}^k \sum_{i, j \in \sigma_k} d_{ij}$, with a restriction for vehicles capacity, expressed by formula $\sum_{i \in \sigma_k} d_i \leq S$.

The emphasis is made on VRP particular cases, represented by basic and effectively solvable tasks on matchings: the assignment problem, the 2-factor problem, the weighted matching problem.

The problem of weighted matching comes to 2-VRP and VRP, where $S/3 \leq d_i \leq S, i = \overline{1, n}$; both assignment and 2-factor problems, apart from their logistic definitions, can serve as relaxations, accelerating search for finding solution of the Traveling Salesman Problem, using the Little's method.

All known algorithms for matching problems described by complex and different methods of building up. Among them – are algorithms for assignment problems and 2-factor problems that can increase the accuracy of the lower bounds in the Little's method, but cannot reduce its complexity.

The adduced analysis proves the need to solve the basic problems of matching by one single method that will accelerate the buildup and reduce the cost of routes in generic tasks of a Traveling Salesman Problem.

The Second section provides a description of the method, targeting the solution for both - the assignment problem and the problem of weighted matching.

The solution of the assignment problem using the suggested method can be found, based on problem definition as follows.

To find

$$C(\sigma) = \min_{\pi} \sum_{i=1}^n c_{\pi[i]}.$$

For cost matrix $C = [c_{ij}]_n$, where $c_{ij} \in R_0^+$ or $c_{ij} = \infty$, R_0^+ – a set of nonnegative real numbers.

In this formula $\pi = (\pi[1], \pi[2], \dots, \pi[n])$ – is set permutation $\{1, 2, \dots, n\}$ of matrix C column numbers.

The recurrent form for the assignment problem solution $\sigma = \pi_n$ is satisfied by means of matching theory for bipartite graphs.

The method of solving the assignment problem is represented by a step-by-step finding of $\pi_1, \pi_2, \dots, \pi_n = \sigma$, with labor intensity $O(n^3)$, which is no greater than that of the most common Hungarian method. It consists of the same type of actions, simplifying the software implementation and providing memory savings for input data of large dimensions.

The outlined architecture σ is applicable for solving problem of weighted matching. An arbitrary graph $H = (V, U)$, $|V| = n$, is given for weighted matching task; graph edge $\{v_i, v_j\} \in U$ has weight of $c_{ij} \in R_0^+$, $i, j = \overline{1, n}$. It is required to find in graph H the maximal matching M_{opt} with the minimal total amount of the edges weight $C(M_{opt})$.

Analyzing the step-by-step representation of the method for solving the weighted matching task, it follows that after ordering values of elements above the matrix main diagonal $C = [c_{ij}]_n$ and of graph H, it is possible to get a correct value of M_{opt} over a period of time $O(n^3)$, which is shorter than when using the well-known Edmonds algorithm.

The third section describes a modified method for solving the Hamiltonian Traveling Salesman Problem, which is a Traveling Salesman Problem for an arbitrary weighted graph $H = (V, U)$. A number of VRP-class problems can be narrowed down to Hamiltonian problem of the Traveling Salesman.

There is a graph $H = (V, U)$, given in the Hamiltonian Traveling Salesman Problem, with weights of edge $s \{v_i, v_j\} \in U$, equal to $c_{ij} \in R_0^+, v_i, v_j \in V$. The route of the traveling salesman or the cycle is a simple basic cycle in the graph H. It is required to find a cycle with the minimal total amount of the edges weight $C(\tau^*)$ or to prove that a set of cycles (loops) of the graph H is empty.

The method is represented by an upgraded version of the classical Little's algorithm, where solution of the 2-factor problem is used to get increasing accuracy of the lower bound estimates of the cost for cycles. A 2-factor of an arbitrary graph is called the union of not overlapping nodes of cycles. The 2-factor problem lies in the fact that, it is required to find in a weighted graph a 2-factor, which would have the smallest sum of its edges weights.

In the suggested updated version of the Little's algorithm, the height of the search tree is bounded from above by the value of $\lfloor n/3 \rfloor - 1$. Since the number of all nodes is not greater than $2^u - 1$, the complexity of the solution of the Hamiltonian Traveling Salesman Problem is estimated by the figure $O(n^3 2^{\lfloor n/3 \rfloor})$.

The Fourth section describes the process of designing and developing a software product, allowing building routes of traveling salesman, based on the ideas, outlined in this thesis work.

The computational experiment was carried out on randomly generated data sets, submitted to the input of algorithms for solving the assignment problem and of the method for solving the 2-factor problem.

Subject of the study was working time of known methods for solving assignment problem (the method of potentials and the Hungarian method) and the recurrent method, proposed in this thesis work.

The working time of the 2-factor method for solving problem of obtaining minimum weight of the 2-factor was under study. The same parameters and input data were used, as it was in experiment for studying working time of the methods for solving the assignment problem.

A time of solving a Travel Salesman Problem by updated version of the Little's method was under study, where various relaxations were used to obtain estimates of lower bound for the cost of traveling salesman routes. There were generated 100 random 10 by 170 cost-matrices in increments of 5. Range of values in matrix was from 10 to 100. 100 problems were solved using three updated versions of the Little's method for each matrix dimension, where in order to obtain the lower bound the 2-factor method, the recursive method for solving the assignment problem and the Hungarian algorithm were used.

A comparison of the methods for solving the assignment problem proved that the recursive method for solving the assignment problem, represented in the Second section of this thesis work, showed the best characteristics.

The results of computational experiment show that the shortest time for solving a Traveling Salesman Problem demonstrates the updated version of the Little's method, using the 2-factor relaxation. The second result shows usage of updated version for solution of the assignment problem by recursive method.

The practical significance of the results, received in this thesis work, is that the developed models and methods of routing optimization are focused on improvement of logistics in real time and in the real-life environment of traffic flow.

Association of the thesis work with scientific research programs, curriculums, topics

The research of this thesis work was carried out within the frame of scientific and technological work of the Kharkov National Automobile and Highway University. The research findings, on which this thesis work is based, have been obtained during realization of scheduled budget programs and projects, namely:

- Project, commissioned by the State Agency of Ukraine for Science, Innovation and Informatization No. DZ/464-2011 on the topic of «Development and implementation of information and communication technology for ground transportation system in large cities».

- R & D under contract No. F62/106-2015 on the topic of "Development and implementation of up-to-date information and communication technologies for

mechatronic and navigation systems of armored wheeled and tracked vehicles".

– R & D under contract No. 01-53-16 on the topic of "Provision of competitive strength for enterprises of the transport industry of Ukraine by increasing the efficiency of virtual management of transport services".

– R & D under contract No. 0117U002405 on an industry-specific request from the Ministry of Education and Science of Ukraine on the topic of "Development of information and communication technology for the intelligent management of ground unmanned multipurpose vehicles."

The results of the research have been implemented in the educational process of the Department of Computer Technologies and Mechatronics of Kharkov National Automobile and Highway University in curriculum modules "System Analysis" and "Operations research".

Based on the findings of this thesis paper, there were 18 academic papers published, including 6 articles on engineering and natural sciences in scientific publications of Ukraine; 3 articles in international professional publications (which are enlisted in scientometrical database SCOPUS), and 9 publications in abstracts and materials of international conferences and seminars.

Keywords: Hamiltonian Traveling Salesman Problem, Assignment Problem, 2-factor, Weighted Matching Problem.

List of publications of the applicant

1. Геращенко И. В., Маций О. Б., Панишев А. В. Полиномиальное преобразование в приближенных алгоритмах решения задач типа коммивояжера // Радиоэлектроника и информатика. 2007. №1. С. 45-49.

2. Левченко А. Ю., Маций О. Б., Панишев А. В. Оптимізація замкнених маршрутів на транспортній мережі // Штучний інтелект. 2010. № 1. С 43-49.

3. Маций О. Б. Повышение точности симметричной задачи коммивояжера большой размерности // Вестник Харьковского национального автомобильно-дорожного университета. 2011. № 55. С 100-102.

4. Маций О. Б. Удосконалення угорського методу рішення задачі про призначення // Открытые информационные и компьютерные технологии. 2014. № 66. С. 166-171.
5. Matsiy O. B., Morozov A. V., Panishev A. V. The Recurrent Method to Solve the Assignment Problem // Cybernetics and Systems Analysis. 2015. Vol. 51. No 6. Pp. 939-946.
6. Кушнір Н. А., Маций О. Б., Скачков В. А. Метод найкоротших збільшуючих шляхів у задачах про паросполучення // Вісник Житомирського державного технологічного університету. 2016. № 3(74). С. 101-111.
7. Matsiy O. B., Morozov A. V., Panishev A. V. Fast Algorithm to Find 2-Factor of Minimum Weight // Cybernetics and Systems Analysis. 2016. Vol. 32. No 3. Pp 464-474.
8. Matsiy O. B., Morozov A. V., Panishev A. V. A Recurrent Algorithm to Solve the Weighted Matching Problem // Cybernetics and Systems Analysis. 2016. Vol. 52. No 5. Pp 748-753.
9. Маций О. Б. Класс задач маршрутизации, сводимых к задаче коммивояжера // Автомобіль і Електроніка. Сучасні Технології. 2017. № 12. С. 167-170.
10. Никонов О. Я., Маций О. Б. Подход к решению симметричной задачи коммивояжера // Сучасні проблеми математичного моделювання та обчислювальних методів: матеріали Всеукраїнської наукової конференції. Рівне: РДГУ-НУВГ, 2013. С. 192.
11. Маций О. Б. Про один підхід до рішення задачі комівояжера з симетричною матрицею відстаней // Інформаційні технології і мехатроніка: матеріали Всеукраїнської науково-практичної конференції. Харків: ХНАДУ, 2014. С. 83.
12. Маций О. Б., Шаригін Г. М. Сучасні аспекти моделювання маршрутів перевезення // Інформаційні технології і мехатроніка: матеріали Всеукраїнської науково-практичної конференції. Харків: ХНАДУ, 2014. С. 160-161.

13. Маций О. Б. Рекуррентный метод решения задачи о назначении // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я (MicroCAD-2014): матеріали XXII міжнародної науково-практичної конференції. Харків: НТУ ХПІ, 2014. С. 249.

14. Маций О. Б. Метод ускоренного поиска кратчайших гамильтоновых маршрутов // Обчислювальний інтелект (результати, проблеми, перспективи): матеріали III міжнародної науково-практичної конференції. Київ-Черкаси: М-во освіти і науки України, Київ. нац. ун-т імені Тараса Шевченка та [ін.]; наук. ред. В.Є. Снитюк. – Черкаси: видавець Чабаненко Ю, 2015. С. 312.

15. Маций О. Б. Алгоритм решения задачи о взвешенном паросочетании // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я (MicroCAD-2015): матеріали XXIII міжнародної науково-практичної конференції. Харків: НТУ ХПІ, 2015. С. 231.

16. Маций О. Б. Схема поиска гамильтонова маршрута минимальной стоимости на транспортной сети // Інформаційні технології і мехатроніка: освіта, наука та працевлаштування: матеріали міжнародної науково-практичної конференції. Харків: ХНАДУ, 2016. С. 76-78.

17. Маций О. Б. Поліноміальне перетворення наближених алгоритмів в рішення задач типу комівояжера // Інформаційні технології і мехатроніка: освіта, наука та працевлаштування: матеріали міжнародної науково-практичної конференції Харків: ХНАДУ, 2017. С. 54-57.

18. Панишев А. В., Маций О. Б. Базовые задачи оптимизации замкнутых маршрутов и способы усовершенствования методов их решения // Інноваційні технології в науці та освіті. Європейський досвід: матеріали міжнародної конференції. Відень: Издательство, 2017. С. 315-319.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	21
ВСТУП.....	22
РОЗДІЛ 1 АНАЛІЗ ОПТИМІЗАЦІЙНИХ ЗАДАЧ МАШРУТИЗАЦІЇ І МЕТОДІВ ЇХ РОЗВ’ЯЗАННЯ	28
1.1 Задача маршрутизації та її різновиди	28
1.2. Паросполучення в моделях транспортної логістики.....	34
1.3 Підходи до розв’язання задач маршрутизації.....	38
1.4. Базові задачі оптимізації замкнених маршрутів і способи удосконалення методів їх розв’язання.....	45
Висновки до розділу 1	55
РОЗДІЛ 2 РЕКУРЕНТНИЙ МЕТОД РОЗВ’ЯЗАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ ПРО ПАРОСПОЛУЧЕННЯ В ДВОДОЛЬНИХ І ДОВІЛЬНИХ ГРАФАХ.....	56
2.1. Перестановно-матричний підхід до побудови оптимального значення	56
2.2. Обґрунтування і опис рекурентного методу розв’язання задачі про призначення.....	61
2.3 Обґрунтування рекурентного методу для розв’язання задачі про зважене паросполучення.....	72
2.4. Рекурсивний метод побудови максимального паросполучення мінімальної ваги в довільному графі.....	84
Висновки до розділу 2	95
РОЗДІЛ 3 РЕКУРЕНТНИЙ МЕТОД ЗНАХОДЖЕННЯ 2-ФАКТОРА МІНІМАЛЬНОЇ ВАГИ В РОЗВ’ЯЗАННІ ГАМІЛЬТОНОВОЇ ЗАДАЧІ КОМІВОЯЖЕРА.....	97
3.1. Постановка задачі 2- <i>f</i> термінах паросполучень для дводольних графів	97
3.2. Обґрунтування і опис алгоритму розв’язання задачі 2- <i>f</i>	102

3.3. Особливості і опис алгоритму розв’язання гамільтонової задачі комівояжера на основі модифікованого методу Літтла	118
Висновки до розділу 3	131
Розділ 4 ПРОГРАМНИЙ ПРОДУКТ І РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ	132
4.1. Основні вимоги до програмного продукту	132
4.3. Опис інтерфейсу програмного продукту.....	132
4.4. Структура програмного продукту.....	136
4.5. Результати обчислювального експерименту	138
4.5.1. Етапи експериментального дослідження методів розв’язання ЗП і задачі 2- <i>f</i>	138
4.5.2. Результати обчислювального експерименту щодо дослідження методів розв’язання ЗП і задачі 2- <i>f</i>	139
4.5.3. Результати обчислювального експерименту щодо дослідження модифікації методу Літтла з різними способами отримання нижніх оцінок	141
Висновки до розділу 4	145
Висновки	146
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	148
Додаток А СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЙНОЇ РОБОТИ	160
Додаток Б Ієрархія основних класів програмного продукту	163
Додаток В АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ.....	164
Додаток Г АКТ ВПРОВАДЖЕННЯ РЕЗУЛЬТАТІВ ДИСЕРТАЦІЙНОЇ РОБОТИ	1645

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ГЗК – гамільтонова задача комівояжера.

ГЗСЛ – гамільтонова задача про сільського листоношу.

ЗЗП – задача про зважене паросполучення.

ЗЗК – загальна задача комівояжера.

ЗК – задача комівояжера.

ЗКЛ – задача китайського листоноші.

ЗП – задача про призначення.

КЗСЛ – кільцева задача про сільського листоношу.

МЗК – метрична задача комівояжера.

СЗК – симетрична задача комівояжера.

ТЗ – транспортні засоби.

ЦП – циклове покриття.

BF – best-fit.

FF – first-fit.

HRPP – модифікація класичного алгоритму Літтла.

SBRT – School Bus Routing Problem.

VRP – Vehicle Routing Problem.

2- f – задача знаходження 2-фактора мінімальної ваги.

ВСТУП

Актуальність теми. Серед важливих проблем розвитку економічного потенціалу країни особливе місце займає проблема підвищення ефективності управління транспортним процесом. Її розв'язання залежить не лише від рівня модернізації транспортних засобів (ТЗ) і використання сучасних інформаційних технологій, але і від вибору маршрутів.

Формування раціональних маршрутів при строгому дотриманні термінів поставок допомагає не тільки мінімізувати витрати на експлуатацію, на перевезення людей, а й скоротити товарно-виробничі запаси на складах. За різними оцінками від 30% до 50% всіх витрат на логістику пов'язано з транспортними витратами.

Задачам в області формування оптимальних транспортних маршрутів присвячені численні дослідження в різних країнах світу. Особливої актуальності набувають роботи, що дозволяють точно обчислювати обсяги вантажоперевезень, розраховувати кількість одиниць транспорту, необхідних для забезпечення вантажопотоків, визначати раціональні маршрути руху, а також скоротити сумарні витрати на транспортування.

Реальні умови роботи транспортних засобів у мережах автомобільних доріг висувають ряд задач оптимізації замкнених маршрутів. В основі задач оптимізації замкнених маршрутів лежить класична задача маршрутизації (VRP). Встановлено, що VRP і її прикладні версії є узагальненнями NP-повної задачі комбінаторної оптимізації.

Дисертаційна робота присвячена дослідженню математичних моделей оптимізації транспортних перевезень, розробленню нових та вдосконаленню відомих методів і алгоритмів комбінаторної оптимізації, які застосовуються в транспортній логістиці для побудови замкнених маршрутів.

Розв'язання таких задач є важливим і набуває особливого значення в екстремальних умовах, коли виникає необхідність швидкої перебудови маршрутів руху при зміні зовнішніх умов.

Реальні умови роботи транспортних засобів у мережах автомобільних доріг висувають ряд задач оптимізації замкнених маршрутів. В основі задач оптимізації замкнених маршрутів лежить класична задача маршрутизації (VRP). На сьогодні відома ціла низка її узагальнень, у яких враховуються різні обмеження та критерії. Зазвичай, для більшості таких узагальнень пошук розв'язку, який задовольняв би за точністю та швидкодією, виявляється занадто трудомістким.

Запропоновано модифікацію методу Літгла для розв'язання багатомаршрутних задач транспортного типу. Особливість модифікації полягає в тому, що вперше оцінка знизу шуканого значення цільового функціонала визначається в результаті розв'язання задачі знаходження 2-фактора мінімальної ваги. Застосування вибраної оцінки обмежує згори дерево перебору та істотно скорочує час пошуку розв'язку задач транспортного типу.

Для класу задач про паросполучення, наведених задачею про призначення, задачею знаходження 2-фактора мінімальної ваги і задачею про зважене паросполучення, запропоновано методи розв'язання, розроблені за єдиною схемою, яка знижує трудомісткість відомих методів.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційне дослідження здійснювалося відповідно до плану науково-технічних робіт Харківського національного автомобільно-дорожнього університету. Наукові результати, на яких базується дисертаційна робота, отримані при виконанні планових бюджетних програм, замовлень:

– проект на замовлення Державного агентства з питань науки, інновацій та інформатизації України №ДЗ/464-2011 що до теми «Розроблення та впровадження інформаційно-комунікаційної технології руху наземного транспорту великих міст»;

– НДР за договором № Ф62/106-2015 «Розроблення та впровадження новітніх інформаційно-комунікаційних технологій для мехатронних і навігаційних систем броньованих колісних і гусеничних машин»;

– НДР за договором № 01-53-16 «Забезпечення конкурентоспроможності

підприємств транспортної галузі України за рахунок підвищення ефективності віртуального управління процесами транспортного обслуговування»;

– НДР за галузевим замовленням МОН України за договором № 0117U002405 «Розроблення інформаційно-комунікаційної технології інтелектуального керування наземними безпілотними багатоцільовими транспортними засобами».

Мета і завдання дослідження. Метою роботи є розроблення та модифікація математичних методів знаходження замкнених маршрутів у задачах транспортного типу.

Для досягнення поставленої мети в дисертації вирішуються такі завдання:

- огляд і аналіз сучасних досягнень комбінаторної оптимізації, що формує математичний апарат транспортної логістики;
- визначення класу різновидів та окремих випадків VRP, який містить основні задачі про паросполучення в дводольних та довільних графах: задачу знаходження 2-фактора мінімальної ваги (2-f) і задачу про зважене паросполучення (ЗЗП);
- розроблення методу для розв’язування задачі про призначення (ЗП), 2-f і ЗЗП зі швидкодією, що перевищує швидкодію відомих методів;
- розроблення наближеного розв’язання класичної задачі маршрутизації модифікацію алгоритму Літтла, що скорочує обсяг обчислень шляхом підвищення точності нижніх оцінок, які встановлюються швидкими алгоритмами побудови зважених паросполучень;
- програмна реалізація методу розв’язання задач про паросполучення і модифікації методу Літтла;
- експериментальне дослідження і аналіз досягнутих результатів.

Об’єкт дослідження – замкнені маршрути в задачах транспортної логістики.

Предмет дослідження – оптимальність замкнених маршрутів в задачах

транспортної логістики.

Методи досліджень. У дисертаційній роботі використано елементи теорії графів для побудови моделі маршрутизації, удосконалення методів розв'язання ЗП, $2-f$ і окремих випадків VRP, методи комбінаторної оптимізації для розроблення модифікації алгоритму Літтла, основи теорії складності для оцінювання трудомісткості розроблених методів.

Наукова новизна одержаних результатів. У результаті виконання дисертаційного дослідження розроблені нові та вдосконалені існуючі моделі та методи оптимізації замкнених маршрутів, розроблено програмне забезпечення для оптимізації замкнених маршрутів в задачах транспортного типу.

При цьому отримано такі нові наукові результати:

- набула подальшого розвитку схема пошуку в ширину, яка в комбінації із способами побудови в графах збільшуючих шляхів, утворює метод розв'язання базових задач теорії паросполучень;

- вперше на основі запропонованої схеми розроблено метод знаходження паросполучення мінімальної ваги в довільному графі з використанням більш простої структури дводольного графа і меншою часовою складністю, чим у відомих методах;

- вперше запропоновано рекурентний метод розв'язання задачі про призначення, що містить швидку процедуру побудови найкоротшого збільшуючого шляху в зваженому графі, чим досягається перевага в швидкості обчислень перед іншими методами;

- вдосконалено рекурентний спосіб побудови зважених паросполучень, на основі якої будується 2 -фактор мінімальної ваги з найменшою на сьогодні часовою складністю, в результаті зведення 2 -фактора до обмеженої задачі про призначення;

- вдосконалено модифікацію алгоритму Літтла з істотно меншим часом розв'язання задач класу комівояжера, що містить для швидкого обчислення точніших оцінок меж шуканого оптимуму один з окремих випадків 2 -фактора.

Практичне значення отриманих результатів. Розроблені моделі та методи орієнтовані на вдосконалення організації перевезень у реальному масштабі часу і в реальних умовах руху транспортних засобів. Їх практичне застосування дозволяє зменшити витрати часу і пального на виконання транспортних робіт, патрулювання населених пунктів, доставку пошти, прискорюють передачу даних із безпілотних літальних апаратів, що виконують замкнені маршрути.

Усі запропоновані методи програмно реалізовані, апробовані і показали свою працездатність і ефективність на серіях розв’язання тестових задач у широкому діапазоні розмірності вхідних даних. На їх основі створено програмний продукт, впроваджений в навчальний процес Харківського національного автомобільно-дорожнього університету та використаний у виробничому процесі науково-виробничого підприємства «Карсис» (м. Харків).

Особистий внесок здобувача. Основні результати дисертації отримані здобувачем і опубліковані в роботах [14, 45, 47, 54 – 56, 124 – 126]. У роботах, опублікованих у співавторстві, здобувачеві належать такі результати: в [14] – спосіб перетворення i -дерева в гамільтонів цикл для побудови наближеного розв’язання задачі комівояжера; в [45] – алгоритм розв’язання задачі про зважене паросполучення; в [47] – декомпозиція транспортної мережі на блоки для підвищення ефективності точних і наближених методів розв’язання загальної задачі комівояжера; в [124] – перестановочно-матрична постановка задачі про призначення і рекурентний метод її розв’язання; в [125] – зведення задачі $2-f$ до обмеженої задачі про призначення і швидка процедура побудови найкоротшого збільшуючого шляху відносно поточного паросполучення в схемі покрокового пошуку її розв’язання; в [126] – опис і доказ коректності методу знаходження зважених паросполучень, що не містить операцій з непарними циклами.

Апробація результатів дисертації. Основні результати дисертаційної роботи докладалися і обговорювалися на:

– Всеукраїнській науковій конференції «Сучасні проблеми математичного моделювання та обчислювальних методів» (Україна, Рівно, 2013 р.);

– Всеукраїнській науково-практичній конференції «Інформаційні технології і мехатроніка» (Україна, Харків, 2014 р.);

– XXII міжнародній науково-практичній конференції «Інформаційні технології: наука, техніка, технологія, освіта, здоров'я» (MicroCAD-2014) (Україна, Харків, 2014 р.);

– III міжнародній науково-практичній конференції «Обчислювальний інтелект (результати, проблеми, перспективи)» (Україна. Київ-Черкаси, 2015 р.);

– XXIII міжнародній науково-практичній конференції «Інформаційні технології: наука, техніка, технологія, освіта, здоров'я» (MicroCAD-2015) (Україна, Харків, 2015 р.);

– Міжнародній науково-практичній конференції «Інформаційні технології і мехатроніка: освіта, наука та працевлаштування» (Україна, Харків, 2016 р.);

– Міжнародній науково-практичній конференції «Синергетика, мехатроніка, телематика дорожніх машин і систем у навчальному процесі та науці» (Україна, Харків, 2017 р.);

– Міжнародній конференції «Інноваційні технології в науці та освіті. Європейський досвід» (Відень, 2017 р.).

Публікації. Основні результати дисертації опубліковані в 18 наукових публікаціях, в їх числі 6 статей у наукових фахових виданнях України з технічних наук, 3 статті – у міжнародних фахових виданнях (наукометрична база SCOPUS), і 9 тез доповідей, опублікованих у матеріалах наукових конференцій, 6 з яких є міжнародними.

Структура і обсяг роботи. Дисертаційна робота складається зі вступу, 5 розділів, висновків, списку використаних джерел із 134 найменувань (12 с.), 4 додатків (6 с.), 63 рисунків, 3 таблиці. Загальний обсяг роботи складає 165 сторінок, з них 123 сторінок основного тексту.

РОЗДІЛ 1

АНАЛІЗ ОПТИМІЗАЦІЙНИХ ЗАДАЧ МАШРУТИЗАЦІЇ І МЕТОДІВ ЇХ РОЗВ'ЯЗАННЯ

1.1 Задача маршрутизації та її різновиди

Задачі маршрутизації, які займають важливе місце в дослідженні проблем комбінаторної оптимізації є задачами поточного планування, в процесі якого визначаються траєкторії і розклади руху об'єктів, що переміщуються. В умовах кожної задачі маршрутизації міститься опис мережі комунікацій, що визначає безліч можливих шляхів проходження до мети одного або декількох рухомих об'єктів. Як правило, структурні параметри мережі залишаються незмінними від початку і до закінчення процесу рішення задачі.

Кожна задача маршрутизації характеризується трьома основними факторами: транспортною мережею, вартістю перевезення і визначеністю. Опис транспортної мережі містить один або кілька параметрів (відстань, пропускну здатність, обмеження на швидкість та ін.). Вартість перевезення може залежати як від довжини, так і від завантаження транспортного засобу (ТЗ). Визначеність означає, що сформульовані у задачі умови переміщення замовлень не змінюються в процесі її розв'язання. Крім трьох перерахованих ознак класифікації, задачі маршрутизації мають певний набір інших властивостей, які доповнюють умови перевезення: характеристику пунктів виробництва і споживання, баз ТЗ, вид вантажу, обмеження на місткість ТЗ та інші обмеження, властиві реальному транспортному процесу.

Задачі маршрутизації займають ключове місце в економічно обгрунтованому прийнятті рішень, які прискорюють виконання транспортних робіт. Умови цих задач перетинаються з класичною задачею маршрутизації (VRP – Vehicle Routing Problem), сформульованої Данцигом і Рамсером [12].

У VRP розглядається n споживачів i , кожному з яких треба доставити однорідний вантаж у заданій кількості d_i з єдиної бази 0 , використовуючи K

ТЗ однакової місткості S . Кожен споживач обслуговується тільки одним ТЗ, що виконує замкнутий маршрут із початком у базі. Задана вартість d_{ij} перевезення з пункту i в пункт j , $i \in N \cup \{n+1\}$, $|N| = n$, не залежна від об'єму (ваги) вантажу, причому $d_{ij} = d_{ji}$. Допустимий розв'язок VRP полягає в розбитті множини N на K підмножин, наведених перестановками σ_k . Перестановка σ_k визначає послідовність доставки вантажів кожному споживачеві і задовольняє умові $\sum_{i \in \sigma_k} d_i \leq S$. Оптимальне розв'язання VRP мінімізує $\sum_{k=1}^K \sum_{i,j \in \sigma_k} d_{ij}$. Воно знаходиться в повному графі з $n+1$ вершинами, що відповідають n пунктам споживання і базі, і ребрами з вагами, які дорівнюють d_{ij} [12, 31]

Існує достатньо велика кількість задач VRP, які відрізняються одна від одної врахуванням тих або інших характеристик вузлів, транспортних засобів, а також введенням додаткових обмежень. Наявність різних обмежень визначається типом задач, а часто також тим, що реальні задачі є дуже складними і в таких випадках розглядаються спрощені задачі.

Найпростішим варіантом VRP є задача з одним складом та однотипними транспортними засобами (Capacitated VRP). До обмежень задач VRP відносяться: наявність часових вікон вузлів (VRP with Time Windows); використання декількох типів транспортних засобів (VRP with Heterogeneous Fleet); врахування різних типів вузлів, наприклад, коли в одному вузлі необхідно забрати певні товари і доставити їх згодом в інший вузол (VRP with Pickup and Delivery); попереднє обмеження спрощується, коли у деяких вузлах необхідно забрати товари та доставити їх на склад (VRP with Backhauls); кожний вузол-клієнт може обслуговуватись тільки транспортними засобами певного типу (Site Dependent VRP).

Задачі маршрутизації на автомобільному транспорті та методи їх розв'язання вивчаються в рамках наукового напрямку – транспортної логістики, математичний апарат якої представлений теорією графів і дослідженням операцій.

Якщо в VRP усі споживачі обслуговуються одним ТЗ, то $K = 1$ і $\sum_{i=1}^n d_i \leq S$.

У цьому випадку VRP формулюється як задача комівояжера (ЗК): потрібно знайти простий цикл, який містить усі $n + 1$ вершин повного графа, що становить мінімум транспортних витрат.

Одна з поширених постановок ЗК полягає в тому, що відомі відстані d_{ij} між кожною парою міст i та j , $i, j \in \{1, 2, \dots, n\}$. Потрібно знайти таку послідовність міст $(\pi[1], \pi[2], \dots, \pi[i], \dots, \pi[n])$, для є мінімальною величина

$$\sum_{i=1}^{n-1} d_{\pi[i]\pi[i+1]} + d_{\pi[n]\pi[1]} \cdot \quad (1.1)$$

Ця величина дорівнює довжині найкоротшого маршруту (обходу), що починається в місті відвідування $\pi[1]$, що по черзі проходить по усіх містах і закінчується в $\pi[1]$ після відвідування $\pi[n]$. ЗК, в якій $d_{ij} = d_{ji}$ для кожної пари міст $\{i, j\}$, називається симетричною ЗК (СЗК) [3, 25, 52]. ЗК і СЗК є NP-повними в сильному сенсі проблемами. Вони відносяться до основних задач комбінаторної оптимізації і, утворюючи безперервно поповнювану множину застосувань і узагальнень, залишаються актуальною темою досліджень [25, 67, 70].

Нехай в повному графі з $n + 1$ вершинами кожному ребру приписано нульову вагу (усі маршрути доставки в VRP мають однакову вартість), але визначено плату за використання кожної одиниці ТЗ. Ця плата фіксована для усіх ТЗ однакової місткості. Тут потрібно знайти мінімальне число ТЗ, що забезпечує перевезення n вантажів d_i . Така задача, відома як задача про упакування в контейнери, NP-повна в сильному сенсі. Оскільки VRP містить умови СЗК і задачі про упакування, то мало надії на побудову точного розв'язання VRP ефективними алгоритмами [25]. Крім того, обмеження $\sum_{i=1}^n d_i \leq KS$ при заданому $K > 2$ не є достатньою умовою існування допустимого розв'язку VRP.

До ЗК з обмеженнями зводиться задача маршрутизації, в умовах якої є один ТЗ, спочатку розташований у базі. ТЗ має перевезти однорідний вантаж з пунктів виробництва в пункти споживання і повернутися на базу.

Задані загальне число пунктів виробництва і споживання, яке дорівнює n , $N = \{1, 2, \dots, n\}$ (базі присвоєно номер 0), вартості d_{ij} транспортування вантажу з пункту i в пункт j , $i, j \in \{0\} \cup N$, місткість ТЗ, дорівнює S , вага вантажу q_i , який потрібно вивезти з пункту виробництва (при цьому $q_i < 0$), або доставити в пункт споживання (при цьому $q_i > 0$). Природно, що виконується умова балансу $\sum_{i=1}^n q_i = 0$. Потрібно знайти перестановку $(\pi[1], \pi[2], \dots, \pi[i], \dots, \pi[n])$ множини N , таку, що

$$0 \leq \sum_{i=1}^u q_{\pi[i]} \leq S, \quad u \in N, \quad (1.2)$$

$$d_{0,\pi[1]} + \sum_{i=1}^{n-1} d_{\pi[i],\pi[i+1]} + d_{\pi[n],0} \rightarrow \min \quad (1.3)$$

З виразів (1.1) і (1.3) виходить, що сформульована задача є ЗК, в якій множина допустимих розв'язків за умови (1.2) може виявитися порожньою.

Наприклад, вона не має розв'язання $(0, \pi[1], 0, \pi[2], \dots, \pi[i], \dots, \pi[n], 0)$ при $S = 1$, $n = 5$, $q_i = 2/3$ для постачальників $i = 1, 2, 3$ та $q_i = -1$ для споживачів $i = 4, 5$ [12].

Клас задач маршрутизації, що зводяться до ЗК, містить задачу про кран [12, 79]. У ній заданий змішаний граф (V, U, A) , в якому множина вершин V , множина дуг A і множина ребр U утворюють модель транспортної мережі. Відомою є матриця відстаней d_{ij} , визначена на її ребрах і дугах. Дуга мережі відповідає перенесенню вантажу з одного місця на інше, а ребро – переміщенню крана без

вантажу. Потрібно знайти такий маршрут крана, який би проходив по усіх дугах мережі і мав мінімальну довжину. Задача про кран зводиться до ЗК, якщо покласти довжини усіх дуг такими, що дорівнюють 0, а кожну пару вершин, пов'язаних дугою, об'єднати в одну вершину. Нехай i^1 є об'єднанням вершин i_1 і i_2 , а вершина i_2 – об'єднанням вершин j_1 та j_2 . Відстань між вершинами i^1 та j^1 визначається таким чином: $d_{i^1 j^1} = \min \{d_{i_1 j_1}, d_{i_1 j_2}, d_{i_2 j_1}, d_{i_2 j_2}\}$.

Після розв'язання ЗК з матрицею $\|d_{ij}\|$ треба повернутися до початкової транспортної мережі і додати усі дуги в отриманий обхід. На сьогодні відомі декілька прикладних версій VRP. До них відноситься, наприклад, задача про шкільний автобус (SBRP – School Bus Routing Problem), яка має таке формулювання. Школа має в розпорядженні k однакових ТЗ місткості S , призначених для доставки кожного учня i до місця його проживання по закінченні занять, $i = \overline{1, n}$, $n = |N|$. Школі присвоєний номер 0. Відомі час проїзду t_{ij} від пункту i в пункт j , $i, j \in \{0\} \cup N$ і вартість проїзду d_{ij} . Кожен ТЗ має повернутися в пункт 0 за час, що не перевершує T [12].

У SBRP потрібно знайти булівські змінні x_{ij} , $i, j \in \{0\} \cup N$, і таку кількість K ТЗ, що пункт 0 є початком (кінцем) маршруту кожного ТЗ:

$$\sum_{i=1}^n x_{i0} = \sum_{i=1}^n x_{0i} = k, \quad (1.4)$$

будь-який пункт доставки i входить в єдиний маршрут $i \in N$:

$$\sum_{i=1}^n x_{ij} = \sum_{i=1}^n x_{ji} = 1; \quad j \in N, \quad (1.5)$$

не існує маршрутів, що містять тільки пункти доставки:

$$\sum_{\substack{i,j \in U \\ U \subset N}} x_{ij} < |U|, \quad (1.6)$$

маршрут ТЗ $(0, i[1], i[2], \dots, i[j], \dots, i[r], 0)$, $i[j] \in N$ задовольняє умові місткості:

$$\sum_{j=1}^{\tau-1} x_{i[j], i[j+1]} = \tau - 1 \leq S - 1 \quad (1.7)$$

і обмеженню за часом його виконання:

$$t_{0i[1]} + \sum_{j=1}^{r-1} t_{i[j], i[j+1]} + t_{i[\tau]0} \leq T. \quad (1.8)$$

Цільова функція SBRP має вигляд:

$$\sum_{i=0}^n \sum_{j=0}^n d_{ij} x_{ij} \rightarrow \min. \quad (1.9)$$

Очевидно, в SBRP $d_i = 1$, $i = \overline{1, n}$, замість $d_i \in Z^+$ в VRP, а кількість ТЗ $= K = \lceil n/S \rceil$, Z^+ – множина цілих додатних чисел.

Якщо в SBRP вартість d_{ij} і час t_{ij} переміщення ТЗ з i -го пункту в j -й лінійно залежні і $d_{ij} = 0$, коли $t_{ij} = 0$, то (1.9) можна замінити на цільову функцію

$$\sum_{i=0}^n \sum_{j=0}^n t_{ij} x_{ij} \rightarrow \min, \quad (1.10)$$

використовуючи початкові дані d_{ij} , $i, j \in \{0\} \cup N$ як допоміжні, для економічної

оцінки побудованого розв'язку.

Вимога $d_i = 1, i = \overline{1, n}$, значною мірою спрощує пошук (1.9). Обмеження щодо завантаження ТЗ набуває вигляду нерівності $k \leq \lceil n/S \rceil$, яка є як необхідною, так і достатньою умовою існування допустимого розв'язку задачі.

Якщо покласти в (1.7) і (1.8) $r = n$, то SBRP виявляється ЗК на множині вершин $\{0\} \cup N, |N|=n$, транспортної мережі, що задається повним графом.

Близькою за змістом до VRP є задача k -VRP. На відміну від VRP у k -VRP не задані кількість вантажу d_i , що доставляється i -му споживачеві, $i = \overline{1, n}$, і місткість S кожного ТЗ, але необхідно, щоб воно обслуговувало не більше k клієнтів. Потрібно мінімізувати сумарну вартість маршрутів усіх ТЗ, кількість яких дорівнює $m \leq \lceil n/k \rceil$. Тому k -VRP вирішується для $n, k \in \mathbb{Z}^+$ і $n \geq k$. При $n=k$ вона являє собою ЗК, визначену на множині обходів $(0, i[1], i[2], \dots, i[j], \dots, i[n], 0)$, при $k=2$ поліноміально вирішувана, але вже при $k \geq 3$ відноситься до класу NP-повних проблем [16].

Очевидна особливість, що об'єднує розглянутий перелік задач маршрутизації, полягає в тому, що вони формулюються як узагальнення або варіанти NP-повної VRP з обмеженнями, які звужують область допустимих розв'язків. Найсильніші з обмежень стають недостатніми умовами розв'язності, стимулюючи інтерес до дослідження задач комбінаторної оптимізації.

1.2. Паросполучення в моделях транспортної логістики

Разом із задачами маршрутизації до оптимізаційних задач транспортної логістики слід віднести ті, які формулюються в термінах теорії паросполучень [49]. Множина ребер графа G називається незалежною, або паросполученням M , якщо ніякі два ребра з M не мають загальних вершин. Кількість ребер $|M|$ називається потужністю паросполучення M . Паросполучення M_{\max} є максимальним, якщо в графі G не існує паросполучення M потужності, більш

$|M_{\max}|$. Очевидно, $|M_{\max}|$ не перевершує $\lfloor v/2 \rfloor$, V – множина вершин графа G . Якщо $|M_{\max}| = \lfloor v/2 \rfloor$, то паросполучення M_{\max} називається досконалим, або 1-фактором. Задача про паросполучення полягає в побудові в графі G максимального паросполучення. Задача про паросполучення є окремим випадком задачі про зважене паросполучення (ЗЗП), в якому заданий граф $G = (V, W)$ і кожному ребру $\{v_i, v_j\} \in W$ приписано вагу $d_{ij} \geq 0$. Передбачається знайти в G паросполучення з мінімальною (максимальною) сумою ваг ребер. Залежно від змістовної інтерпретації пошук розв'язання задачі про паросполучення або ЗЗП виконується в довільному графі $H = (V, U)$ або в дводольному графі. Дводольним графом $D = (X, Y, E)$ називається граф, множину вершин якого можна навести розбиттям на дві підмножини X і Y так, що кінці кожного ребра належать різним підмножинам [99].

Однією з перших ЗЗП є класична задача про призначення (ЗП), що отримала таке логістичне формулювання. Число пунктів виробництва n дорівнює кількості ТЗ і числу пунктів призначення. У кожному пункті споживання i знаходиться один ТЗ, який після виконання маршруту вартістю d_{ij} в якому-небудь пункті споживання $j, i, j = \overline{1, n}$, залишається в цьому пункті. Треба знайти усі n маршрутів (i, j) , тих, що мінімізують їх сумарну вартість [12, 21, 29].

Розв'язання ЗП шукається в дводольному зваженому графі $D = (X, Y, E)$, де $|X| = |Y|$, $E = \{(i, j) | i \in X, j \in Y\}$ і відповідає досконалиму паросполученню з мінімальною сумою ваг його ребер. ЗП завжди вирішувана, якщо дводольний граф D повний, тобто такий, що кожна пара його вершин (i, j) , $i \in X, j \in Y$, пов'язана ребром [99].

Зараз відомі декілька задач транспортного типу, розв'язання яких надані паросполученнями в дводольних графах.

Наприклад, нехай в задачі про «упакування в контейнери»

вантажомісткість контейнера не перевищує B , а вага вантажу d_i , $i = \overline{1, n}$, задовольняє нерівність $B/3 < d_i \leq B$. Тоді мінімальне число контейнерів для розміщення в них усіх вантажів можна визначити таким чином. Будується граф $H = (V, U)$ з множини вершин V і множини U ребер $\{i, j\}$. Вершини i та j утворюють ребро $\{i, j\}$, якщо $d_i + d_j \leq B$. Очевидно, шукана множина контейнерів є максимальним паросполученням графа H .

Треба зауважити, що для розв'язання ЗП, задачі про паросполучення і ЗЗП відомі ефективні алгоритми, що базуються на нетривіальних ідеях, реалізація яких потребує чималих тимчасових витрат.

Неважко показати, що 2-VRP поліноміально розв'язувана шляхом зведення її до задачі побудови в повному зваженому графові з $n + 1$ вершинами досконалого паросполучення M_{\max} з мінімальною сумарною вагою ребер, що входять у нього. Якщо n непарне, то M_{\max} має вигляд, як на рис. 1.1, а, а розв'язання 2-VRP – як на рис. 1.1, б. При парному n M_{\max} (рис. 1.2, а) перетвориться в розв'язання 2-VRP, показане на рис. 1.2, б. Тут, як і надалі, ребра паросполучення зображуються потовщеними лініями.

Мінімальна вартість m маршрутів, $m = \lceil n/2 \rceil$, має постійну складову $\sum_{i=1}^n d_{0i}$ і варійовану величину, яку потрібно визначити в результаті розв'язання ЗЗП.

Паросполучення моделюють безліч додатків, що мають відношення до задач типу комівояжера. Маршрут у ЗК є зв'язним остовним підграфом, в якому усі вершини мають ступінь 2. Якщо в такому підграфі виключити умову зв'язності, то він називається 2-фактором [49].

Очевидно, що 2-фактором є узагальненням поняття досконалого паросполучення. Тому задачу 2- f – задачу знаходження в довільному графі $H = (V, U)$ з невід'ємними вагами ребер, що доставляє їх мінімальну сумарну вагу, можна розглядати як ослаблену ЗК. Відомо, що задача 2- f поліноміально зводиться до побудови досконалого паросполучення [49] і, отже, вирішувана.

Цей факт вказує на можливість застосування ЗП, ЗЗП і задачі 2- f як релаксації для обчислення оцінок, що обмежують знизу значення цільового функціонала в точних і наближених методах розв'язання задач маршрутизації, що зводяться до ЗК.

Релаксацією задачі комбінаторної оптимізації називається деяка інша задача, у множині допустимих розв'язків якої взаємно однозначно відображається (вкладається) допустимий розв'язок початкової [84, 85]. Відомою релаксацією ЗК, яка не відноситься до задач побудови паросполучень, є задача знаходження в повному n -вершинному зваженому графі i -дерева з мінімальною сумарною вагою ребер; i -дерево – це підграф повного графа, що містить дерево з множиною вершин $V \setminus \{i\}$, $|V|=n$, і два ребра, інцидентні вершині i [13, 14, 75]. Природно, розв'язання цієї задачі дає грубішу нижню оцінку мінімуму функціонала ЗК, чим розв'язання розглянутих вище задач, але знаходиться за значно менший час.

1.3 Підходи до розв'язання задач маршрутизації

За рідкісним виключенням оптимізаційні задачі транспортної логістики не піддаються ефективному розв'язанню. У ключовій задачі маршрутизації VRP шукана система маршрутів є комбінацією допустимих розв'язків задачі «про упакування в контейнери» і ЗК, що належать класу NP-повних в сильному сенсі проблем. Щоб знайти оптимальне розв'язання VRP, треба для кожного розподілу вантажу по K ТЗ вирішити ЗК. В той же час при фіксованому K значення S , d_i , $i = \overline{1, n}$, можуть бути такими, що заданої кількості ТЗ виявиться недостатньо для виконання плану перевезень. Інакше кажучи, VRP не завжди вирішувана. Її точне розв'язання досягається шляхом перебору, що викликає зі збільшенням розміру вхідних даних експоненціальне зростання часових витрат.

Сучасний рівень розвитку комп'ютерної математики характеризується різноманітністю наближених та евристичних методів комбінаторної оптимізації для припустимого за часом розв'язання логістичних задач реальної розмірності

[23, 24, 107]. Алгоритми, побудовані на природному подані VRP у вигляді задачі упакування і наступних за нею K ЗК, відносяться до евристичних. Вони виконуються в два етапи.

У таких алгоритмах на першому етапі всі пункти споживання розбиваються на k груп, що не порушує обмеження за місткістю ТЗ, а на другому знаходиться розв'язання ЗК.

Перший етап полягає в знаходженні числа K ТЗ, для якого виконується достатня умова існування допустимого розв'язку VRP. У задачі про упакування величина K при заданих значеннях $d_i, i = \overline{1, n}$, і S є верхньою оцінкою мінімального числа K^* ТЗ. У схемі розв'язання VRP, що додається тут, ця оцінка знаходиться в результаті виконання одного з наближених алгоритмів розподілу вантажів по контейнерах, розглянутих в [25].

Можна вважати, що $S=1, 0 < d_i \leq 1, L=(d_1, d_2, \dots, d_n)$ – послідовність, в якій вага d_1 завантажується першою, а вага d_k завантажується раніше, ніж d_{k+1} , для усіх $k = \overline{2, n-1}$.

Нехай контейнери завантажуються одним із двох алгоритмів FF (first-fit) або BF (best-fit).

Алгоритм FF (завантаження в перший відповідний контейнер) визначає кожну вагу d_k в контейнер з мінімальним номером з числа тих, які підходять для розміщення d_k .

Алгоритм BF (завантаження в кращий з відповідних контейнерів) розміщує кожну вагу d_k у такий контейнер, який після розміщення в ньому d_k матиме найменшу невикористану вантажопідйомність.

Доведено [25], що для будь-якої послідовності L і алгоритмів FF і BF справедлива оцінка

$$K \leq (17/10)K^* + 2. \quad (1.11)$$

Величина K ніколи не відрізняється від оптимуму K^* більш ніж на 70%. Невелике зменшення оцінки (1.11) до значення $(17/10)K^*$ отримано в [98]. Можна спробувати зменшити K , застосувавши алгоритм FF або BF до послідовності L , яка впорядкована за неспаданням ваг d_k . В цьому випадку

$$K \leq (11/9)K^* + 4. \quad (1.12)$$

Така ж оцінка справедлива для алгоритму FF або BF із вхідною послідовністю вантажів, упорядкованою за неспаданням ваг. Неважко помітити, що з ростом n оцінка (1.11) росте швидше, ніж оцінка (1.12).

Очевидно, VRP ефективно вирішувана, коли $d_k > 1/3$, $k = \overline{1, n}$. У цьому випадку вона набуває формулювання 2-VRP і зводиться до ЗЗП для довільного графа з $n + 1$ вершинами. Наближені алгоритми упакування виділяють допустиму множину вершин $M \subset N$, тобто такі, що $\sum_{j \in M} d_j \leq S$. Побудова маршруту комівояжера для кожної множини $M \cup \{0\}$ завершує пошук допустимого розв'язку VRP. Щоб знайти точне розв'язання VRP, недостатньо упакувати усі вантажі, використовуючи мінімальне число контейнерів, а потім знайти стільки ж точних розв'язків ЗК. На практиці цілком задовільними можуть виявитися розв'язки VRP, одержані в результаті упакування з оцінкою (1.11) або (1.12) і K звернень до якого-небудь наближеного алгоритму розв'язання ЗК.

Широке визнання отримали конструктивні методи розв'язання VRP, розроблені на основі ідеї ітераційної побудови і об'єднання замкнених маршрутів, не порушуючи обмежень за місткістю ТЗ. До них відноситься найвідоміший сьогодні алгоритм Кларка-Райта, який використовується у більшості комерційних пакетів. Він складається з таких трьох кроків [79].

Крок 1. Для кожного споживача i , $i = \overline{1, n}$, формується маятниковий маршрут $(0, i, 0)$.

Крок 2. Для всіх пар i та j обчислюються економії $S_{ij} = d_{0i} + d_{j0} - d_{ij}$, які упорядковуються за незростанням в список S .

Крок 3. Пари $(0, N_i, i, 0)$ і $(0, j, N_j, 0)$ зі списку S об'єднуються в маршрут $(0, N_i, i, j, N_j, 0)$, якщо не порушується умова завантаження ТЗ, $N_j, N_i, N_j \subseteq N = \{1, 2, \dots, n\}$, інакше вибирається наступна пара списку.

Одним із способів скорочення сумарної вартості маршруту, побудованого алгоритмом Кларка-Райта, є обчислення економії за формулою $S_{ij} = d_{0i} + d_{j0} - \gamma d_{ij}$, $\gamma > 1$. Варіювання параметром γ дозволяє знайти декілька досить хороших розв'язань VRP і вибрати з них найкраще. Ідеї алгоритму Кларка-Райта привели до появи його модифікацій, орієнтованих як на скорочення невикористаної вантажопідйомності кожного ТЗ, так і на мінімізацію сумарного пробігу. До таких модифікацій відноситься широко відомий «вимітаючий алгоритм», що працює на площині із заданою полярною системою координат і центром у базовому пункті. Пункти споживачів i , $i = \overline{1, n}$, включаються в маршрут одного ТЗ у міру збільшення кута в полярних координатах при виконанні обмежень на вантажомісткість. Далі визначається маршрут комівояжера для кожної отриманої підмножини пунктів [79].

Недавні результати дослідження VRP містять різноманітні механізми пошуку і поліпшення її допустимого розв'язку. Для такого розв'язання невідомі аналітично виражені оцінки похибки. Більше того, далеко не просто оцінити його точність на основі експериментально отриманих даних. Можливість застосування алгоритмів розв'язання VRP у додатках зазвичай встановлюється таким чином. Формується серія задач із реальними значеннями параметрів. Після програмної реалізації кожного алгоритму для сформованої серії задач і аналізу отриманих розв'язань вибирається алгоритм, що визначає найменші транспортні витрати в межах заданого інтервалу часу.

Важливе значення в розвитку методів керування логістичними системами має досвід розв'язання задач маршрутизації шляхом занурення їх в область

чисельного лінійного програмування [12] чи в область метаевристик [23, 24]. Проте точне розв'язання VRP і її узагальнень досягається за обмежений час тільки для невеликої розмірності при використанні методу гілок і меж у комбінації з алгоритмами відсікань, методу динамічного програмування і методу Лагранжа [12, 75, 79]. Тому на практиці мають широке поширення евристичні алгоритми, що не гарантують знаходження оптимальних маршрутів задач класу VRP, але використовують розумні міркування для побудови допустимих розв'язків і подальшого їх поліпшення. Прикладом такого алгоритму є розглянутий вище двоетапний алгоритм, за допомогою якого знаходять спочатку наближене розв'язання задачі упаковки, а потім звертається до точної або наближеної процедури розв'язання ЗК. Усі відомі евристики VRP є складними для формального вивчення, аналітично встановлюючого, наскільки у гіршому разі відрізняється від оптимального евристичне розв'язання. До поліноміальних алгоритмів класичної VRP не вдається застосувати підхід, який би за рахунок відмови від пошуку оптимального розв'язання визначав допустиме з гарантованою оцінкою точності. Вагомішими є результати, що присвячені аналізу поведінки поліноміальних алгоритмів NP-повної задачі k -VRP [16]. Для 3-VRP відомий наближений ефективний алгоритм, який знаходить розв'язання з сумою довжин маршрутів, обмеженою чотирма мінімальними сумами. Побудований поліноміальний $(5/2 - 1/k)$ – наближений алгоритм задачі Metric k -VRP, коли відстані між базою і споживачами задовольняють евклідовій метриці. За останнє десятиліття набув розвитку підхід до розв'язання задач класу VRP на випадкових вхідних даних. Він допускає певну частку числових прикладів задачі, які або порушують очікувану оцінку точності алгоритму, або потребують для його виконання експоненціальних часових витрат. Імовірнісному аналізу швидких алгоритмів розв'язання k -VRP присвячена робота [16].

Маршрут в k -VRP – це простий цикл, що містить не більше ніж $n+1$ вершин, включаючи базу 0, наприклад, $(0, i_1, i_2, \dots, i_k, 0)$, де $i_j \in \{1, 2, \dots, n\}$, $j \in \overline{1, k}$, або $(0, i, 0)$, $i \in V \setminus \{0\}$, V – множина вершин повного графа $|V| = n+1$.

У матриці вартостей цієї задачі елементи головної діагоналі дорівнюють ∞ , а інші елементи є незалежними випадковими величинами з однаковою щільністю розподілу.

Алгоритми задач на випадкових вхідних даних характеризуються відносною похибкою і вірогідністю неспрацьовування.

Для задачі на мінімум $F_A(I)$, $F^*(I)$ позначають значення цільового функціонала на розв'язанні, отриманому алгоритмом A для входу I , і на оптимальному розв'язанні для цього ж входу відповідно.

У класі K_n задач розмірності n на мінімум алгоритм A має оцінки $(\varepsilon_n, \sigma_n)$, якщо для кожного n виконується нерівність

$$P_r \{ F_A(I) > (1 + \varepsilon_n) F^* \} \leq \sigma_n,$$

де $P_r(\omega)$ – вірогідність події ω ,

I – індивідуальне завдання з K_n ,

ε_n – відносна похибка,

σ_n – вірогідність неспрацьовування алгоритму A .

Алгоритм A називається асимптотично точним у класі $K = \bigcup (K_n \mid n = 1, 2, \dots)$, якщо для нього існують такі оцінки $(\varepsilon_n, \sigma_n)$, що $\varepsilon_n \rightarrow 0$ і $\sigma_n \rightarrow 0$ при $n \rightarrow \infty$.

Для розв'язання k -VRP в [16] запропонований наступний алгоритм з часом роботи $O(n^2)$:

Етап 1. Вибрати випадковим чином $2m$ вершин $\{1, 2, \dots, n\}$, $m = \lceil n/k \rceil$.

Нехай вибрані вершини утворюють множину $V' = \{1, 2, \dots, 2m\}$. Позначимо $V'' = \{1, 2, \dots, n\} \setminus V'$.

Етап 2. Покрити вершини графа m простими циклами, що перетинаються

тільки у базі.

2.1 Загальний крок $i = 1, 2, \dots, m - 1$. Послідовно для кожного ТЗ i побудувати маршрут $(0, i_1, \dots, i_k)$.

2.1.1 В якості i_1 береться довільна вершина з V' (наприклад, з найменшим номером).

2.1.2 Вершини i_2, \dots, i_{k-1} , беруться з V'' і знаходяться за правилом «найближчого непройденого міста», починаючи з вершини $i_1 \in V'$.

2.1.3 Вершина $i_k \in V'$, відмінна від i_1 , вибирається як найближча до вершини i_{k-1} .

2.1.4 З V' видаляються вершини i_1 і i_k , а з V'' – вершини i_2, \dots, i_{k-1} .

2.1.5 Останній крок $i = m$. Будується маршрут $(0, m_1, m_2, \dots, m_l, 0)$, при цьому вершини $m_1, m_l \in V'$ замикають вершини, що залишилися, з V'' у результаті виконання вже застосованої евристики.

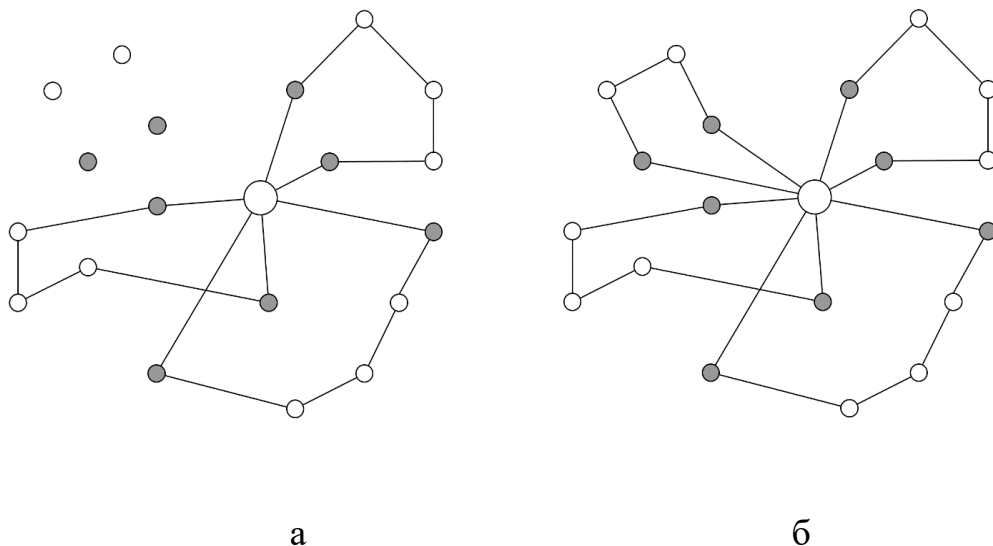


Рисунок 1.3 – Маршрути, побудовані алгоритмом перед виконанням останнього кроку для 5-VRP і результат розв’язання 5-VRP: а – маршрути, побудовані алгоритмом перед виконанням останнього кроку для 5-VRP з 19 споживачами; б – результат розв’язання задачі 5-VRP з 19 споживачами

На рис. 1.3, а зображено маршрути, побудовані алгоритмом перед виконанням останнього кроку для 5-VRP з 19 споживачами, а на рис. 1.3, б показано результат розв'язання цієї задачі.

1.4. Базові задачі оптимізації замкнених маршрутів і способи удосконалення методів їх розв'язання

Розв'язання більшості задач маршрутизації полягає в побудові одного або декількох циклів, що проходять по усіх вершинах графа $H = (V, U)$, який служить базовою моделлю транспортної мережі. Вершина $i \in V$ графа H може бути подана споживачем вантажу, населеним пунктом або перехрестям доріг. У графі H вершини i та j з'єднані ребром $\{i, j\} \in U$, якщо вони відповідають сусіднім на трасі населеним пунктам або сусіднім перехрестям вулиць міста. Граф не містить петель, тобто ребер $\{i, i\}$. Кожне ребро $\{i, j\}$ має вагу $d_{ij} \in R_0^+$, яке дорівнює відстані або вартості переміщення з i в j , $d_{ij} = d_{ji}$, $i, j = \overline{1, n}$, $d_{ii} = \infty$, $i = \overline{1, n}$, R_0^+ – множина дійсних невід'ємних чисел.

Для опису маршрутів ТЗ, доріг, що проходять по ділянках, з одностороннім рухом, використовують змішаний граф (V, U, A) з множиною вершин, множиною ребер U і множиною дуг.

Однією з базових задач маршрутизації на графі $H = (V, U)$ є задача китайського листоноші (ЗКЛ), в якій потрібно знайти найкоротший замкнений маршрут, що містить принаймні один раз кожне ребро [51].

Якщо в графі H усі вершини мають парну ступінь, тобто якщо граф H парний, то ЗКЛ за час $O(|U|)$ вирішується побудовою ейлерова циклу – циклу, що містить у точності один раз кожне ребро.

Якщо граф H непарний, то ЗКЛ також ефективно вирішується. Проте в цьому випадку побудова шуканого маршруту виконується за допомогою алгоритму знаходження найкоротших ланцюгів між усіма парами вершин із

непарними степенями і досить трудомісткого алгоритму знаходження паросполучення з максимальною вагою [51].

До базових задач маршрутизації на графі $H = (V, U)$ слід віднести загальну задачу комівояжера (ЗЗК). У ЗЗК потрібно знайти найкоротший замкнутий маршрут, що пов'язує всі вершини V .

ЗЗК NP-повна в сильному сенсі. У [47, 73] викладено точний метод її розв'язання. Метод містить алгоритм побудови найкоротших ланцюгів між усіма парами вершин графа H для його перетворення в повний граф ЗК, алгоритм розв'язання ЗК і процедуру побудови за отриманим розв'язанням шуканого маршруту. При цьому матриця відстаней повного графа задовольняє нерівності трикутника

$$d_{ij} \leq d_{ik} + d_{kj}, i \neq j, j \neq k, i \neq k, i, j, k \in \{1, 2, \dots, n\}, |V| = n. \quad (1.13)$$

СЗК називається метричною (МЗК), якщо вона обмежена на симетричні матриці, для яких виконується умова (1.13).

Для МЗК відомі ефективні алгоритми з оцінкою точності, що прямує до константи з ростом порядку матриці відстаней $[d_{ij}]_n$. Тому для ЗЗК існує наближене розв'язання з гарантованою похибкою, що досягається за поліноміальний час [73]. Час побудови маршруту ЗЗК істотно зменшується в мережі, поданій у вигляді розрідженого графа $H = (V, U)$. У цьому випадку граф H розглядається як сукупність блоків – максимальних зв'язних підграфів, що не мають точок з'єднання, а кожен блок – як вхід підзадачі ЗЗК. Шукане розв'язання знаходиться в результаті швидкої процедури об'єднання розв'язків підзадач в один маршрут [73, 123].

Умови ряду задач маршрутизації перетинаються з умовами NP-повної гамільтонової задачі комівояжера (ГЗК).

У ГЗК задано зважений граф $H = (V, U)$ транспортної мережі, і потрібно

знайти гамільтоновий цикл (маршрут комівояжера) мінімальної вартості.

На відміну від ЗК розв'язання ГЗК шукається в неповному графі, в якому множина гамільтонових циклів може виявитися порожньою (негамільтоновому графі). Щоб з'ясувати, чи є граф H гамільтоновим, треба побудувати в ньому гамільтонів цикл. Але така побудова виконується тільки методом перебору. Тому для розв'язання ГЗК втрачає сенс розроблення ефективних і наближених алгоритмів.

До базових задач маршрутизації відноситься задача про сільського листоношу (ЗСЛ). У ЗСЛ задано граф $H = (V, U)$, вага $d(u) \in R_0^+$ кожного ребра $u \in U$ і підмножина $R \subset U$. Потрібно знайти в графі H цикл, який містить кожне ребро з R і має мінімальну суму ваг ребер [51].

Звуження множини допустимих розв'язків СЗК до множини гамільтонових циклів $z(R)$ графа H визначає гамільтонову задачу про сільського листоношу (ГЗСЛ) [62, 63]. ГЗСЛ полягає в побудові в графі H гамільтонового циклу $z^*(R)$, такого, що мінімізує функціонал

$$C(z(R)) = \sum_{\{k, l\} \in z(R)} d_{kl}. \quad (1.14)$$

Нехай в ЗСЛ вимагається, щоб шуканий цикл був простим. Позначимо $y(R)$ простий цикл графа $H = (V, U)$, що містить усі ребра заданої множини R . Задача знаходження простого циклу $y^*(R)$, такого, що доставляє мінімум функціоналу

$$C(y(R)) = \sum_{\{k, l\} \in y(R)} d_{kl}, \quad (1.15)$$

називається кільцевою задачею про сільського листоношу (КЗСЛ) [65].

Очевидно, ГЗСЛ є окремим випадком КЗСЛ.

Крім того, ГЗСЛ і КЗСЛ характеризуються тим, що до них, як і до ГЗК, незастосовні ефективні наближені алгоритми або евристики, що не містять необхідних і достатніх умов існування розв'язків цих задач. За заданим графом $H = (V, U)$ і підмножиною його ребер $R \subset U$ не можна встановити, коли ГЗСЛ і КЗСЛ мають розв'язання. На рис. 1.5 показано граф $H = (V, U)$, що не містить простих циклів, які містять усі ребра множини $R = \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$. Тому безальтернативною схемою розв'язання ГЗСЛ і КЗСЛ є будь-яка схема обмеженого перебору, яка коректно або будує шуканий цикл, або встановлює, що в заданому графі $H = (V, U)$ з підмножиною ребер $R \subset U$ його немає.

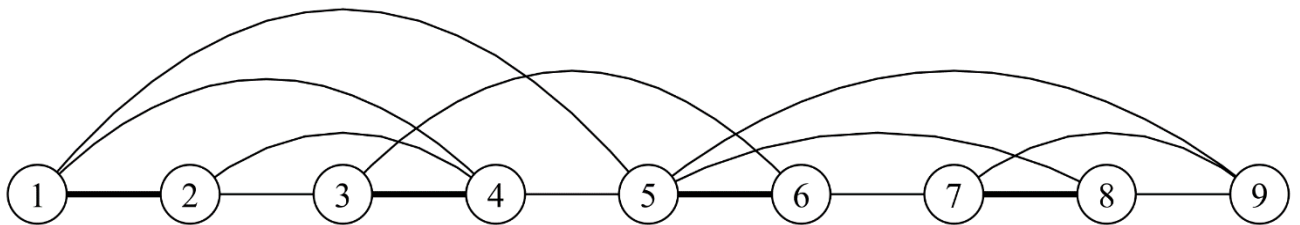


Рисунок 1.5 – Граф $H = (V, U)$, що не містить простих циклів, які включають усі ребра множини $R = \{\{1, 2\}, \{3, 4\}, \{5, 6\}, \{7, 8\}\}$

У ГЗК відносно ГЗСЛ $R \neq \emptyset$. Умови КЗСЛ на відміну від ГЗСЛ викликають невизначеність підмножини вершин графа, по яких пройде цикл $y^*(R)$, і істотно ускладнюють схему розв'язання задачі.

У [61, 73] викладено підхід до розв'язання ГЗК, ГЗСЛ і КЗСЛ, що містить модифікації класичного алгоритму Літтла зі вбудованими в нього засобами зменшення розмірності вхідної матриці і звуження області усіх простих циклів графа транспортної мережі. Цей підхід містить два етапи.

На першому етапі розв'язання ГЗК або задач мінімізації (1.14) і (1.15) перевіряються усі відомі достатні умови негамільтоновості графа H . Якщо хоч би одне з них виконується, то множина гамільтонових циклів у H є порожньою.

Ясно, що граф H негамільтонів, коли він містить вершини одиничної степені і точки з'єднання [99]. Якщо в H немає висячих вершин і точок з'єднання, але існують вершини степені 2, то кожна із задач перетвориться на задачу меншої розмірності в результаті розбиття графа H на певні частини і переміщення кожної частини в одну вершину. Процес перетворення зупиняється, коли по-перше, степені усіх вершин отриманого графа H^0 більше 2, по-друге, побудований шуканий цикл. У результаті знаходження H^0 в ньому утворюється множина ребер R^0 . У ГЗСЛ і КЗСЛ R^0 міститься підмножина $R \subset U$, а в ГЗК $R^0 \not\subset U$. Якщо R^0 не є паросполученням, то ГЗК, ГЗСЛ і КЗСЛ не мають розв'язків. Трудомісткість першого етапу розв'язання даних задач оцінюється величиною $O(|V|^2)$.

Другий етап розв'язання ГЗК, ГЗСЛ і КЗСЛ виконується, коли початковий граф H перетворюється на граф H^0 з підмножиною ребер R^0 , що утворює паросполучення R^0 . На цьому етапі застосовують модифікацію класичного алгоритму Літтла (HRPP) для розв'язання ГЗК і ГЗСЛ і її більше трудомістка версія CRPP, яка коректно шукає розв'язок КЗСЛ, ГЗСЛ і ГЗК [65, 73].

Формальні постановки задач оптимізації циклічних маршрутів нерідко описують технологічні процеси механообробки. У [4] розглядається робота автоматичного пристрою для нанесення розмітки на плоску металеву поверхню ударним голчастим механізмом. На прямокутному полі заданих розмірів цей механізм наносить N написів, кожний з яких можна умовно уявити як однозв'язну область з координатами двох крайніх точок (x_{i1}, y_{i1}) і (x_{i2}, y_{i2}) , $1 \leq i \leq N$. Час нанесення напису залежить від його конфігурації і кількості ударів голки і не залежить від того, в прямому напрямку – від точки $i1$ до точки $i2$ або в зворотному – від точки $i2$ до точки $i1$ виконано цей напис. Час холостого ходу голки від останньої точки одного напису до першої точки наступного напису є пропорційним відстані між цими точками. Для N написів необхідно визначити

послідовність їх нанесення з мінімальним холостим пробігом.

Змістовна постановка задачі має таке графове формулювання. У повному графі задано паросполучення – набір пар точок, пов'язаних «сильними» ребрами (написами). Слід побудувати ланцюг, що містить усі «сильні» ребра і «слабкі» ребра переходів, які доставляють мінімум суми їх довжин. Відмінність поставленої задачі від КЗСЛ полягає в тому, що її розв'язання є не циклом, а ланцюгом і шукається в повному, а не в довільному графі. Тому за заданим набором написів можна завжди отримати найкоротшу послідовність їх нанесення. З представлення плоскої поверхні, на яку наносять написи, у вигляді повного графа, що задовольняє умові (1.13), випливає, що, задача розмітки відкрита для розв'язання ефективними наближеними алгоритмами з гарантованими оцінками похибки.

Для розглянутого списку задач маршрутизації можна виділити два підходи до їх розв'язання. Перший підхід викликає практично непереборні складнощі побудови оптимальних маршрутів, але забезпечує деякі до них наближення. Його застосовують, наприклад, для розв'язання VPR, SBRP, k -VRP, за умови, що транспортна мережа наведена повним графом. Другий підхід використовують тоді, коли заздалегідь невідомо, чи має дана задача допустимі розв'язки. Він містить різноманітні переборні алгоритми, побудовані переважно за методом гілок і меж, його застосовують для розв'язання всіх задач маршрутизації на транспортних мережах, що моделюються довільними графами.

Головна особливість більшості задач маршрутизації полягає в тому, що вони з необхідністю зводяться до ЗК, для якої застосовують найбільш досконалу на сьогодні схему пошуку точного розв'язання, що лежить в основі класичного алгоритму Літгла [3, 73]. Вона містить: процедуру побудови дерева перебору – бінарного дерева маршрутів комівояжера (обходів), що відсікаються в процесі розгалуження; процедуру обчислення в кожній вершині дерева оцінки знизу вартості частково побудованого маршруту; блок вибору в ньому такого ребра, що всі можливі варіанти маршрутів розбиваються на дві групи: одна з них містить вибране ребро, а інша його не містить, причому для кожної групи формується

своя матриця вартостей; спосіб зберігання даних у дереві перебору. Вибір ребра, що ініціює розгалуження, виконують так, щоб воно було найбільш відповідним для включення в шуканий обхід [3, 42, 43, 73].

Робота алгоритму Літтла може бути істотно прискорена в результаті підвищення швидкодії усіх включених в нього процедур і гранично економної організації оперативної пам'яті.

Час точного розв'язання ЗК при заданому порядку n вхідної матриці вартостей передусім залежить від вибору нижньої оцінки вартості шуканого обходу, яка має забезпечувати досягнення оптимуму швидше, ніж оцінка в алгоритмі Літтла. Тому необхідно, щоб вибрана оцінка перевершувала за точністю оцінку Літтла і мала таку ж трудомісткість. У [73] викладено модифікацію алгоритму Літтла з оцінкою, наведеною в корені дерева перебору розв'язком ЗП, а в інших його вузлах розв'язаннями різновидів цього завдання, які знаходяться за час $O(n^2)$. За такий же час обчислюється оцінка в алгоритмі Літтла. Швидкодію цієї модифікації можна підвищити, якщо в ній, як релаксацію замість ЗП вибрати задачу 2- f [84, 85].

Нехай неорієнтований граф $H = (V, U)$ не містить кратних ребер і петель, тобто якщо $\{v, u\} \in U$, то $\{u, v\} \notin U$ і $\{v, v\} \notin U$. Підмножина $U' \subset U$ ребер графа H називається 2- f , якщо кожна вершина $v \in V$ графа інцидентна рівно двом ребрам в U' . Очевидно, 2- f складається з простих циклів, що покривають усі вершини графа H , і таких, що кожен цикл містить не менше трьох ребер.

Підмножина $U'' \subset U$ ребер графа $H = (V, U)$ називається паросполученням (досконалим паросполученням), якщо кожна вершина $v \in V$ інцидентна не більше (рівно) одного ребра.

Початковими даними для побудови маршруту комівояжера в графі $H = (V, U)$ є симетрична матриця вартостей $[d_{ij}]_n$ з числами із R_0^+ і елементами, які дорівнюють ∞ . З цієї ж матриці знаходяться розв'язання ЗП, задачі 2- f , і ЗЗП. У граф H розв'язання задачі 2- f є сукупністю незв'язних простих циклів із трьох

або більш ребер. Кількість таких циклів не більша, ніж $\lfloor n/3 \rfloor$. Розв'язання ЗП у графі H також можна розглядати як множину компонент зв'язності, в якому кожна компонента є ребром.

Для зображення графа розв'язання ЗП замінимо в H кожне ребро $\{i, j\}$ на пару дуг (i, j) і (j, i) з вагами, які дорівнюють вазі цього ребра, отримавши в результаті оргграф $G = (V, E)$, $|E| = 2|U|$. Підграф $G' = (V, E')$ оргграфа $G = (V, E)$ називається контурним покриттям, якщо кожна його вершина $v \in V$ має напівступінь входу та виходу, які дорівнюють 1 [75]. Неважко побачити, що розв'язання ЗП взаємно однозначно відповідає такому контурному покриттю, в якому сума ваг дуг мінімальна на множині усіх контурних покриттів графа $G = (V, E)$.

В загальному випадку графове представлення розв'язання ЗП для матриці вартостей ЗК є сукупністю простих контурів, що покривають множину вершин V оргграфа G . Кількість контурів m_j довжини j у цій сукупності задовольняє рівності

$$\sum_{j=2}^n jm_j = n. \quad (1.16)$$

З (1.16) випливає, що максимальне число компонент у контурному покритті оргграфа дорівнює $\lfloor n/2 \rfloor$ і воно досягається коли $\lfloor n/2 \rfloor$ контурів містять по дві дуги (i, j) і (j, i) . При $m_n = 1$ контурне покриття в G є гамільтоновим маршрутом СЗК.

Нехай σ, η, τ^* – відповідно розв'язання ЗП, задачі 2- f , і СЗК, $C(\sigma)$, $C(\eta)$, $C(\tau^*)$ – вартості цих розв'язків. Якщо графове уявлення σ не містить контурів довжини 2, то $\sigma = \eta$. Звідси випливає, що задача 2- f є обмеженою ЗП.

Тому $C(\sigma) \leq C(\eta) \leq C(\tau^*)$, тобто $C(\eta)$ служить точнішою оцінкою знизу вартості шуканого обходу τ^* , чим $C(\sigma)$. Маючи в розпорядженні алгоритм розв'язання задачі 2- f , швидкодія якого не поступається будь-якому відомому алгоритму розв'язання ЗП, можна значно скоротити число вузлів дерева перебору, породжуваних у методі гілок і меж, викладеному в [98].

Побудову в графі H максимального паросполучення з мінімальною (максимальною) сумою ваг ребер застосовують у наближених алгоритмах розв'язання ЗК, включаючи найвідоміший алгоритм Кристофидеса [75]. За алгоритмом спочатку знаходять з матриці вартостей МЗК остовне дерево T з мінімальною сумою ваг ребер, в якому позначаються усі вершини з непарними степенями. Потім у повному графі, що містить тільки помічені вершини, будується досконале паросполучення M мінімальної вартості. Алгоритм завершує роботу після побудови ейлерова циклу і його обходу в мультиграфі, що містить усі ребра з T і M .

Трудомісткість алгоритму в порівнянні з трудомісткістю алгоритму розв'язання ЗЗП, яке зараз знаходиться за час $O(n^4)$ алгоритмом Едмондса. Довжина побудованого маршруту комівояжера не перевершує $3/2$ його мінімальної довжини. Велика точність алгоритму Кристофидеса досягається ціною виконання трудомістких операцій алгоритму Едмондса.

У теорії паросполучень ЗП, задача 2- f і ЗЗП відносяться до ключових. Їх розв'язок знаходять відомими алгоритмами, які характеризуються невисокою швидкодією і відрізняються способами побудови. Найменш трудомістким з них є угорський метод розв'язання ЗП з часовою складністю $O(n^3)$. [75].

З такою ж часовою складністю вирішують задачу 2- f для n -вершинного повного графа [119]. Єдиний відомий метод розв'язання задачі 2- f для остовного підграфа H повного графа H_n містить її зведення до ЗЗП [49].

Викладені зауваження вказують на необхідність розв'язання базових задач теорії паросполучень за єдиною схемою, яка прискорює і спрощує обчислення.

В основу схеми покладено ідею розв'язання задач про паросполучення в довільному графі $H = (V, U)$ на основі дводольного графа $D = (X, Y, E)$, де X і Y – підмножини вершин, $|V| = |X| = |Y|$, E – множина ребер. В D $(i, j) \in E$ і $(j, i) \in E$, якщо в H $\{v_i, v_j\} \in U$, $i \neq j$, $i, j \in \{1, 2, \dots, n\}$. Зі збігу матриць вартостей графа H і D виходить, що якщо побудовано розв'язання ЗЗП в D , то воно побудовано і в H .

Таким чином, ЗП, задача 2- f і ЗЗП утворюють клас комбінаторних задач оптимізації, які визначають, як (F, c) , де $F = \{$ множина допустимих розв'язків Q у вигляді паросполучень дводольного графа $D\}$, c – функція вартості, що виконує відображення

$$c: F \rightarrow \sum_{(i,j) \in Q} c_{ij}, c_{ij} \in R_0^+, i, j \in \{1, 2, \dots, n\}, i \neq j.$$

Тут Q – досконале паросполучення в ЗП і задачі 2- f і максимальне паросполучення в ЗЗП.

Мета цієї роботи полягає в розробці рекурентного методу знаходження мінімуму в F і його застосуванні для прискорення процесу розв'язання тих задач маршрутизації, які зводяться до ЗК. Метод є результатом розвитку способу пошуку в ширину максимального паросполучення в незваженому дводольному графі [75]. Починаючи з вибору ребра з найменшою вагою, на кожній ітерації методу знаходять паросполучення з мінімальною сумою ваг ребер на множині усіх паросполучень певної потужності. Після виконання процедури побудови найкоротшого збільшуючого шляху відносно паросполучення, отриманого на попередній ітерації, визначається чергове паросполучення мінімальної вартості зі збільшеною на одиницю потужністю.

Рекурентний метод має перевершувати за швидкістю відомі алгоритми розв'язання ЗП, задачі 2- f і ЗЗП.

Висновки до розділу 1

1. Показано, що задачі оптимізації замкнених маршрутів (задача маршрутизації), займають ключові позиції в управлінні процесами переміщення вантажів і пасажирів.

2. Основу задач ефективної організації транспортних робіт утворює класична задача маршрутизації – VRP, пов'язана з NP-складною задачею комівояжера і доповнена реальними умовами процесу перевезень.

3. Визначені окремі випадки VRP, що зводяться до ефективно вирішуваних задач теорії паросполучень: ЗП, задачі 2-*f*, ЗЗП. Відомі алгоритми перерахованих задач характеризуються невисокою швидкістю і складними способами програмної реалізації. Тому їх використання як допоміжних для обчислення нижніх оцінок у методах типу гілок і меж при розв'язанні VRP і ЗК не прискорює і не спрощує процес розгалужень.

4. Мета цього дослідження полягає в розробці методу розв'язання ЗП, 2-*f* і ЗЗП, побудованого за єдиною схемою, такого що підвищує швидкість відомих методів.

Основні результати першого розділу опубліковано у роботах [45, 54].

Список джерел, які використано у даному розділі, наведено у повному списку використаних джерел [3, 4, 12, 13,14, 16, 21, 23, 24, 25, 29, 31, 42, 43, 47, 49, 51, 61, 63, 67, 70, 73, 75, 79, 84, 85, 98, 99, 107, 119, 123].

РОЗДІЛ 2

РЕКУРЕНТНИЙ МЕТОД РОЗВ'ЯЗАННЯ ОПТИМІЗАЦІЙНИХ ЗАДАЧ ПРО ПАРОСПОЛУЧЕННЯ В ДВОДОЛЬНИХ І ДОВІЛЬНИХ ГРАФАХ

Розділ містить результати досліджень щодо вдосконалення відомих алгоритмів знаходження максимальних паросполучень з мінімальною сумарною вагою ребер. Вони представлені методами розв'язання ЗП і ЗЗП, які основані на застосуванні схеми пошуку в ширину як в дводольному, так і в довільному графах.

2.1. Перестановно-матричний підхід до побудови оптимального значення

Широко відомі методи розв'язання ЗП, такі, як угорський метод, метод Кана-Мункреса і метод потенціалів, що побудовані з використанням різних підходів і характеризуються різною часовою складністю, не меншою, ніж $O(n^3)$, де n – порядок матриці вартостей [75, 82]. В [48] викладено алгоритм розв'язання одного з варіантів ЗП, складність якого знижена до $O(n^2)$. В [46] показано, що він виконує функції процедури, вбудованої в метод гілок і меж для швидкого обчислення точніших нижніх оцінок вартості замкнених маршрутів у задачі комівояжера (ЗК). Алгоритм полягає в знаходженні зваженого паросполучення мінімальної сумарної ваги в дводольному графі з $2n$ вершинами із використанням введених в [75] понять найкоротшого збільшуючого шляху.

У даному і наступному підрозділах описано рекурентний метод розв'язання ЗП, що розвиває результати робіт [46, 48, 75] і технічно спрощує найбільш поширений угорський метод.

Опишемо метод розв'язання ЗП, що розглядається тут, виходячи з її формулювання в такому вигляді.

Для матриці вартостей (ваг) $C = [c_{ij}]_n$, де $c_{ij} \in R_0^+$ або $c_{ij} = \infty$, R_0^+ – множина невід'ємних дійсних чисел, знайти

$$C(\sigma) = \min_{\pi} \sum_{i=1}^n c_{\pi[i]}. \quad (2.1)$$

Тут $\pi = (\pi[1], \pi[2], \dots, \pi[n])$ – перестановка множини $\{1, 2, \dots, n\}$ номерів стовпців матриці C , $\sigma = (\sigma[1], \sigma[2], \dots, \sigma[n])$ – оптимальна перестановка вартістю $C(\sigma) = \sum_{i=1}^n c_{\sigma[i]}$, $c_{\sigma[i]} \neq \infty$, $i = \overline{1, n}$, $i = \overline{1, n}$, або розв'язку ЗП. Перестановку $\pi = (\pi[1], \pi[2], \dots, \pi[n])$, в якій $c_{\pi[i]} \neq \infty$, назвемо допустимим розв'язком ЗП. Замітимо, що ЗП із матрицею вартостей, що містить елементи $c_{ij} = \infty$, може не мати розв'язку. В цьому випадку необхідно встановити, що множина допустимих розв'язків задачі є порожньою. Будемо шукати σ , покроково збільшуючи на одиницю число елементів k , $k = \overline{1, n}$, послідовності, що утворює певну частину допустимого розв'язку ЗП. Розглянемо властивості цієї послідовності і спосіб її побудови.

Будь-яка частина допустимого розв'язку ЗП, що містить k елементів, однозначно визначає підматрицю $[c_{i_s j_t}]_k$ матриці C , $i_1 < i_2 < \dots < i_s < \dots < i_k$, $j_1 < j_2 < \dots < j_t < \dots < j_k$. Нехай послідовність $\pi_k = (\pi_k[i_1], \pi_k[i_2], \dots, \pi_k[i_s], \dots, \pi_k[i_k])$, $\pi_k[i_s] \in \{j_1, j_2, \dots, j_t, \dots, j_k\}$, $k = \overline{1, n-1}$: а) є розв'язанням ЗП для підматриці $[c_{i_s j_t}]_k$, б) вартість π_k не більше вартості розв'язання ЗП для будь-якої підматриці порядку k матриці C . Якщо існує ефективна процедура перетворення послідовності π_k в послідовність π_{k+1} , $k = \overline{0, n-1}$, і задача (2.1) має розв'язок, то на знаходження $\sigma = \pi_n$ потрібно n кроків.

Покажемо, як будуються послідовності π_k , $k = \overline{1, n}$.

Початкова послідовність $\pi_1 = (\pi_1[i_1])$ визначається тривіально: в матриці C знаходиться $c_{lr} = \min\{c_{ij} \mid 1 \leq i, j \leq n\}$, і, отже, $i_1 = l$, $\pi_1[i_1] = r$.

Щоб отримати $\pi_2 = (\pi_2[i_1], \pi_2[i_2])$, визначимо $c_{ms} = \min\{c_{ij} \mid i \neq l, j \neq r\}$, $c_{lp} = \min\{c_{lj} \mid j \neq r\}$, $c_{vr} = \min\{c_{ir} \mid i \neq l\}$ (рис. 2.1).

	<i>p</i>	<i>q</i>	<i>r</i>	<i>s</i>	
<i>m</i>	c_{mp}				c_{ms}
<i>l</i>	c_{lp}		c_{lr}		
<i>v</i>			c_{vr}	c_{vs}	
<i>w</i>		c_{wq}			

Рисунок 2.1 – Визначення $c_{ms} = \min\{c_{ij} \mid i \neq l, j \neq r\}$, $c_{lp} = \min\{c_{lj} \mid j \neq r\}$,

$$c_{vr} = \min\{c_{ir} \mid i \neq l\}$$

Неважко побачити, що якщо $c_{lr} + c_{ms} \leq c_{lp} + c_{vr}$, то умови а) і б) виконуються для послідовності π_2 , в якій $i_1 = l$, $\pi_2[i_1] = r$, $i_2 = m$, $\pi_2[i_2] = s$. Їй відповідає підматриця, зображена на рис. 2.2.

	<i>r</i>	<i>s</i>
<i>l</i>	c_{lr}	
<i>m</i>		c_{ms}

Рисунок 2.2 – Підматриця з елементами $i_1 = l$, $\pi_2[i_1] = r$, $i_2 = m$, $\pi_2[i_2] = s$

Інакше цим умовам задовольняє послідовність π_2 з елементами $i_1 = l, \pi_2[i_1] = p, i_2 = v, \pi_2[i_2] = r$ і підматрицею, зображеною на рис. 2.3.

	p	r	
l	c_{lp}	c_{lr}	
v		c_{vr}	.

Рисунок 2.3 – Підматриця π_2 з елементами $i_1 = l, \pi_2[i_1] = p, i_2 = v, \pi_2[i_2] = r$

Перетворимо послідовність π_2 в послідовність $\pi_3 = (\pi_3[i_1], \pi_3[i_2], \pi_3[i_3])$.

Припустимо, що $\pi_2 = (\pi_2[l] = r, \pi_2[m] = s)$.

Знайдемо $c_{wq} = \min \{c_{ij} \mid i \neq l, m; j \neq s, r\}$ і $MIN1 = c_{lr} + c_{ms} + c_{wq}$.

Зауважимо, що $c_{wq} = c_{ms}$, якщо $\pi_2 = (\pi_2[l] = p, \pi_2[v] = r)$.

Перетворення π_2 в π_3 є результатом розв'язання наступної допоміжної задачі. Для рядків l, m і стовпців r, s , визначуваних числами c_{lr} і c_{ms} матриці C , потрібно знайти трійку елементів з мінімальною сумою їх значень за умови, що будь-які два елементи з трійки мають знаходитися в трьох різних рядках, які містять l, m , і трьох різних стовпцях, які містять r, s . Якщо шукана трійка не містить c_{lr} , але містить c_{ms} , то сума значень її елементів обмежена знизу величиною $S_1 = c_{vr} + c_{lp} + c_{ms}$, де

$$c_{vr} = \min \{c_{ir} \mid i \neq l, m\}, c_{lp} = \min \{c_{lj} \mid j \neq r, s\}.$$

Нехай розв'язання допоміжної задачі є трійкою, в яку входить c_{lr} і не входить c_{ms} . Тоді $S_2 = c_{lr} + c_{mp} + c_{vs}$, де

$$c_{mp} = \min \{c_{mj} \mid j \neq s, r\}, c_{vs} = \min \{c_{is} \mid i \neq l, m\}.$$

Елементи розв'язання допоміжної задачі, що не містить c_{lr} і c_{ms} , визначають величину суми, $S_3 = \min \{c_{lp} + c_{mr} + c_{ws}, c_{mq} + c_{ls} + c_{vr}\}$. Тут $c_{ws} = \min \{c_{is} \mid i \neq l, m\}$, $c_{mr} = \min \{c_{mj} \mid j \neq p, s\}$ (рис. 2.4, а), $c_{ls} = \min \{c_{is} \mid i \neq m, v\}$, $c_{mq} = \min \{c_{mj} \mid j \neq r, s\}$ (рис. 2.4, б).

	<i>p</i>	<i>q</i>	<i>r</i>	<i>s</i>
<i>m</i>			c_{mr}	c_{ms}
<i>l</i>	c_{lp}		c_{lr}	
<i>v</i>				
<i>w</i>				c_{ws}

а

	<i>p</i>	<i>q</i>	<i>r</i>	<i>s</i>
<i>m</i>		c_{mq}		c_{ms}
<i>l</i>			c_{lr}	c_{ls}
<i>v</i>			c_{vr}	
<i>w</i>				

б

Рисунок 2.4 – Елементи розв'язання допоміжної задачі, що не містить c_{lr} і

$$c_{ms} : \text{а} - c_{ws} = \min \{c_{is} \mid i \neq l, m\}, c_{mr} = \min \{c_{mj} \mid j \neq p, s\};$$

$$\text{б} - c_{ls} = \min \{c_{is} \mid i \neq m, v\}, c_{mq} = \min \{c_{mj} \mid j \neq r, s\}$$

Визначимо величину, яка дорівнює $MIN2 = \min \{S_1, S_2, S_3\}$. Ясно, що вона відповідає шуканій послідовності π_3 , якщо $MIN2 \leq MIN1$, інакше $\pi_3 = (\pi_3[l] = r, \pi_3[m] = s, \pi_3[w] = q)$.

У загальному випадку послідовність $\pi_k = (\pi_k[i_1], \pi_k[i_2], \dots, \pi_k[i_s], \dots, \pi_k[i_k])$

з властивостями а і б перетвориться в послідовність $\pi_{k+1} = (\pi_{k+1}[i_1], \pi_{k+1}[i_2], \dots, \pi_{k+1}[i_r], \dots, \pi_{k+1}[i_{k+1}])$ з цими ж властивостями таким чином.

У матриці C знаходиться

$$c_{\pi_k[i_{k+1}]} = \min \{c_{ij} \mid i \neq i_1, i_2, \dots, i_s, \dots, i_k, j \neq \pi_k[i_1], \pi_k[i_2], \dots, \pi_k[i_s], \dots, \pi_k[i_k]\},$$

формується послідовність $\pi_{k+1}^1 = (\pi_k, \pi_k[i_{k+1}])$ і обчислюється

$$MIN1 = \sum_{s=1}^k c_{\pi_k[i_s]} + c_{\pi_k[i_{k+1}]}.$$

Далі вирішується задача знаходження $k+1$ елементів, які в матриці C доставляють мінімальну суму своїх значень і розташовуються в різних рядках і стовпцях, включаючи усі рядки і стовпці з номерами, що задаються величинами $c_{\pi_k[i_1]}, c_{\pi_k[i_2]}, \dots, c_{\pi_k[i_s]}, \dots, c_{\pi_k[i_k]}$. Позначимо цю суму $MIN2$. Знайдені елементи утворюють шукану послідовність $\pi_{k+1} = \pi_{k+1}^2$, якщо $MIN2 \leq MIN1$. Інакше $\pi_{k+1} = \pi_{k+1}^1$.

2.2. Обґрунтування і опис рекурентного методу розв'язання задачі про призначення

Викладена схема пошуку оптимального призначення є основою методу, в якому розв'язання задачі (2.1) знаходиться виключно засобами теорії паросполучень для дводольних графів [51, 75, 99].

Матриці вартостей ЗП C взаємно-однозначно відповідає дводольний граф (X, Y, U) , в якому $|X|=|Y|=n$ і вершина $i \in X$ сполучена з вершиною $j \in Y$ ребром $(i, j) \in U$ вагою $c_{ij} \neq \infty$.

Паросполучення графа визначається як така множина ребер, що ніякі два з них не мають загальної вершини. Ребра, що не входять у паросполучення, називаються вільними. Максимальне паросполучення – паросполучення з найбільшим числом ребер. Вершина, що належить ребру паросполучення, називається насиченою, інші вершини графа – вільними. Ребро (i, j) , включене в паросполучення, позначимо $[i, j]$. Вершина j ребра $[i, j]$ визначається як напарник i . Досконале паросполучення – це паросполучення, що насичує усі вершини графа. У дводольному графі (X, Y, U) , де $|X| = |Y| = n$, потужність досконалого паросполучення, якщо воно існує, дорівнює n . Розв’язання ЗП полягає в побудові в дводольному зваженому графі (X, Y, U) , $|X| = |Y| = n$, досконалого паросполучення з мінімальною сумарною вагою ребер [67, 75, 106].

Нехай в графі зафіксоване паросполучення M . Простий шлях називається таким, що чергується відносно паросполучення M , якщо ребра шляху через одне є присутніми в M [49]. Шлях, що чергується, який починається і закінчується ребрами, не належними до паросполучення M , називається збільшуючим відносно паросполучення M . Отже, якщо $(i_0, j_1, i_1, j_2, i_2, j_3, \dots, i_{k-1}, j_k, i_k, j_{k+1})$ – збільшуючий шлях у дводольному графі (X, Y, U) , то в ньому вільні вершини i_0, j_{k+1} і $k+1$ ребер $(i_0, j_1), (i_1, j_2), (i_2, j_3), \dots, (i_{k-1}, j_k), (i_k, j_{k+1})$. Інші k ребер шляху утворюють паросполучення $\pi_k = ([i_1, j_1], [i_2, j_2], [i_3, j_3], \dots, [i_k, j_k])$. Довжиною шляху називається число ребер, що зустрічаються в ньому.

Припустимо, що в зваженому графі (X, Y, U) , яке відповідає матриці вартостей ЗП C , не зафіксовано паросполучення. Тоді кожне ребро в (X, Y, U) є збільшуючим шляхом довжини 1, а ребро з мінімальною вагою утворює паросполучення або початкову послідовність π_1 .

Нехай в графі (X, Y, U) побудовано паросполучення $\pi_k = ([i_1, j_1], [i_2, j_2], \dots, [i_l, j_l], \dots, [i_k, j_k])$ з мінімальною сумою $C(\pi_k)$ ваг ребер серед усіх паросполучень потужності k . Перетворимо π_k в паросполучення

$\pi_{k+1} = ([i'_1, j'_1], [i'_2, j'_2], \dots, [i'_l, j'_l], \dots, [i'_k, j'_k], [i'_{k+1}, j'_{k+1}])$, величина $C(\pi_{k+1})$ якого досягає мінімуму на множині Π_{k+1} усіх паросполучень потужності $k+1$.

Паросполучення π_k розбиває множину X, Y відповідно на підмножини насичених вершин $I_k = \{i_1, i_2, \dots, i_l, \dots, i_k\}, J_k = \{j_1, j_2, \dots, j_l, \dots, j_k\}$ і на підмножину вільних вершин $X - I_k = \{i_{k+1}, i_{k+2}, \dots, i_s, \dots, i_n\}, Y - J_k = \{j_{k+1}, j_{k+2}, \dots, j_q, \dots, j_n\}$. Знайдемо вільне ребро вагою

$$c_{i_r j_p} = \min \{c_{i_s j_q} \mid i_s \in X - I_k, j_q \in Y - J_k\}, \quad (2.2)$$

і, приєднавши його до паросполучення π_k , отримаємо паросполучення $\pi'_{k+1} = \pi_k \cup [i_r, j_p]$ вартістю $MIN1 = C(\pi_k) + c_{i_r j_p}$. Якщо π'_{k+1} не є паросполученням π_{k+1} з мінімальною сумою ваг на множині Π_{k+1} , то $\pi_{k+1} \in \Pi_{k+1} - \{\pi'_{k+1}\}$.

Лема 2.1. Нехай $\pi_{k+1} \in \Pi_{k+1} - \{\pi'_{k+1}\}$. Тоді $\pi_{k+1} = (P_{k+1} - \pi_k) \cup (\pi_k - P_{k+1})$, де P_{k+1} – найкоротший збільшуючий шлях відносно паросполучення π_k .

Доведення. Побудуємо найкоротший збільшуючий шлях P_{k+1} відносно паросполучення π_k , і, отже, паросполучення π_{k+1}^2 . Покажемо, що $\pi_{k+1} = \pi_{k+1}^2$. Ясно, що $C(\pi_k) \leq C(\pi_{k+1}^2)$. У величинах $C(\pi_k)$ і $C(\pi_{k+1}^2)$ виділимо загальний доданок

$$C_0 = \sum_{\pi_k[i_s] \in \pi_k \cap \pi_{k+1}^2} c_{\pi_k[i_s]}.$$

Тому $C(\pi_k) - C_0 \leq C(\pi_{k+1}^2) - C_0$. Але $\pi_k \cap \pi_{k+1}^2 = \pi_k - P_{k+1}$, а $C(\pi_{k+1}^2) - C_0$ – сума ваг ребер великої множини $P_{k+1} - \pi_k$. Ця сума мінімальна, коли P_{k+1} –

найкоротший збільшуючий шлях відносно паросполучення π_k . Отже, π_{k+1}^2 – паросполучення з мінімальною сумою ваг ребер на множині $\Pi_{k+1} - \{\pi'_{k+1}\}$, тобто $\pi_{k+1} = \pi_{k+1}^2$. Доведення завершено.

Нехай вершина $j_l \in J_k, l \in \{1, 2, \dots, k\}$ є кінцем вільного ребра (i_r, j_l) вагою

$$c_{i_r j_l} = \min \{c_{i_s j_l} \mid i_s \in X - I_k\}, \quad (2.3)$$

вершина $i_f \in I_k, f \in \{1, 2, \dots, k\}$ – початком вільного ребра (i_f, j_p) вагою

$$c_{i_f j_p} = \min \{c_{i_f j_q} \mid j_q \in Y - J_k\}. \quad (2.4)$$

Позначимо через X_k множину вільних вершин i_r , кожна з яких інцидентна ребру (i_r, j_l) вагою, визначеною з (2.3), $|X_k| \leq k$. Відповідно Y_k – множина вільних вершин j_p , кожна з яких інцидентна ребру (i_f, j_p) вагою, визначеною з (2.4), $|Y_k| \leq k$. Ясно, що найкоротший збільшуючий шлях відносно паросполучення π_k починається у вершині множини X_k і закінчується у вершині множини Y_k .

Побудуємо допоміжний зважений оргграф (V, A) для пошуку найкоротшого збільшуючого шляху P_{k+1} відносно паросполучення π_k .

В оргграфі (V, A) множина вершин $V = \{i_0\} \cup X_k \cup I_k \cup Y_k$. Множину дуг A оргграфа (V, A) наведено розбиттям на підмножини A_0, A_1, A_2, A_3 . Підмножина A_0 містить $|X_k|$ дуг (i_0, i_r) нульової ваги, $i_r \in X_k$. У підмножину A_1 входить дуга (i_r, i_l) , $i_r \in X_k, i_l \in I_k$, якщо і тільки якщо, вершина j_l ребра (i_r, j_l) – напарник вершини i_l , $[i_l, j_l] \in \pi_k$. Дузі (i_r, i_l) присвоєна вага $c(i_r, i_l) = c_{i_r j_l} + c_{i_l j_l}$. Дуга

$(i_d, i_l) \in A_2, i_d, i_l \in I_k$, тоді і тільки тоді, коли вершина j_l ребра $(i_d, j_l), j_l \in J_k$, є напарником $i_l, [i_l, j_l] \in \pi_k$. Дугі (i_d, i_l) присвоєно вагу $c(i_d, i_l) = c_{i_d j_l} + c_{i_l j_l}$. Підмножина A_3 містить дугу $(i_f, j_p), i_f \in I_k, j_p \in Y_k$, якщо вершини i_f і j_p сполучені в графі (X, Y, U) ребром (i_f, j_p) . Дугі (i_f, j_p) присвоєно вагу $c(i_f, j_p) = c_{i_f j_p}$.

На рис. 2.5 показано граф (X, Y, U) , для якого отримано паросполучення $\pi_k, k = 4$. Ребра паросполучення зображені потовщеними лініями. Вершини i_5 і i_6 утворюють множину Y_4 , а вершини j_5 і j_6 – множину X_4 . Ребра $(i_5, j_1), (i_6, j_2), (i_6, j_3)$ мають вагу, отриману з (2.3), а ваги ребер $(i_2, j_5), (i_3, j_6), (i_4, j_6)$ визначені з (2.6).

Допоміжний оргграф (V, A) , побудований на графі (X, Y, U) , який відповідає паросполученню π_4 , зображено на рис. 2.6. Множина вершин V містить разом з вершиною i_0 підмножини $X_4 = \{i_5, i_6\}, I_4 = \{i_1, i_2, i_3, i_4\}, Y_4 = \{j_5, j_6\}$. Множина дуг A утворюють підмножини $A_0 = \{(i_0, i_5), (i_0, i_6)\}, A_1 = \{(i_5, i_1), (i_6, i_2), (i_6, i_3)\}, A_2 = \{(i_1, i_3), (i_1, i_4), (i_3, i_1)\}, A_3 = \{(i_2, j_5), (i_3, j_6), (i_4, j_6)\}$.

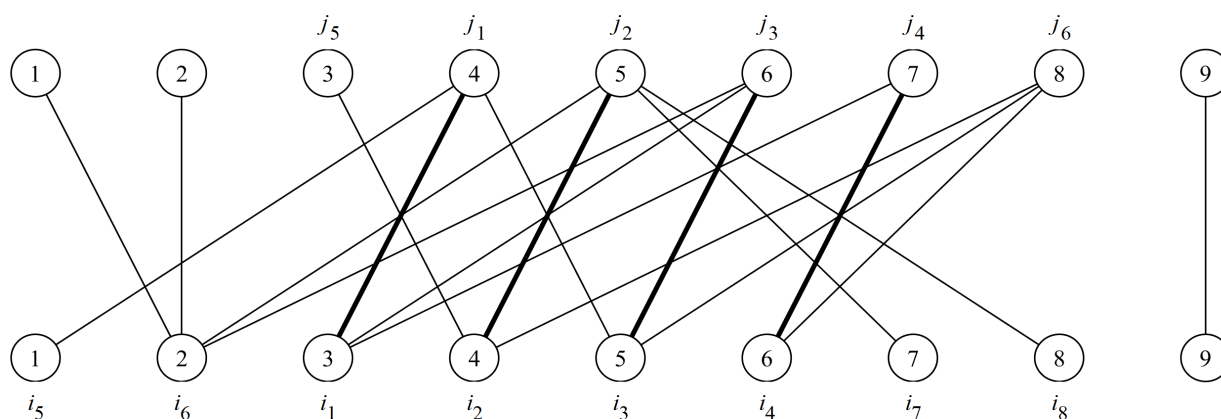


Рисунок 2.5 – Граф (X, Y, U) , для якого отримано паросполучення $\pi_k, k = 4$

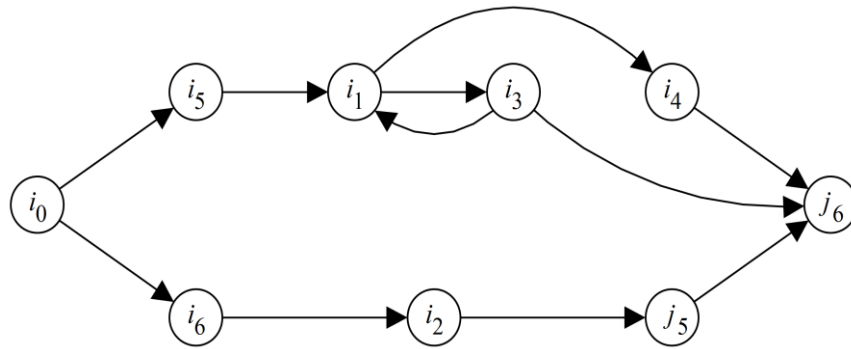


Рисунок 2.6 – Допоміжний оргграф (V, A) , побудований на графі (X, Y, U) , який відповідає паросполученню π_4

Із способу побудови оргграфа (V, A) виходить, що множина шляхів, досяжних з вершини i_0 в усі вершини $j_p \in Y_k$, збігається з множиною збільшуючих шляхів відносно паросполучення π_k , які з'єднують у графі (X, Y, U) кожну вершину $i_r \in X_k$ з кожною вершиною $j_p \in Y_k$. Якщо в графі (V, A) побудовано найкоротший шлях з вершини i_0 у вершину $j_p \in Y_k$, то він містить одну з дуг $(i_r, i_l) \in A_1$ і одну з дуг $(i_f, j_p) \in A_3$. Нехай $j_m \in Y_k$, – вершина шляху, найкоротшого серед усіх шляхів з i_0 в інші вершини множини Y_k . Очевидно, такий шлях у графі (X, Y, U) визначає найкоротший збільшуючий шлях P_{k+1} відносно π_k .

Отже, для знаходження в графі (X, Y, U) паросполучення π_{k+1} , вартість якого мінімальна на множині всіх паросполучень π_{k+1} потужності $k+1$, необхідно:

а) визначити паросполучення $\pi'_{k+1} = \pi_k \cup [i_r, j_p]$ і сумарну вагу його ребер $MIN1 = C(\pi_k) + c_{i_r, j_p}$, π_k – паросполучення мінімальної вартості на множині усіх паросполучень π_k потужності k , c_{i_r, j_p} – вага ребра $[i_r, j_p]$, що визначають із з (2.2);

б) знайти паросполучення π_{k+1}^2 і $MIN2 = C(\pi_{k+1}^2)$, побудувавши найкоротший збільшуючий шлях P_{k+1} відносно паросполучення π_k ;

в) покласти $\pi_{k+1} = \pi_{k+1}^1$, якщо $MIN1 \leq MIN2$, і $\pi_{k+1} = \pi_{k+1}^2$ інакше.

Наступний алгоритм виконує пошук розв'язання ЗП, рекурсивно визначаючи в дводольному графі (X, Y, U) , $|X| = |Y| = n$, паросполучення π_k , що містять k ребер мінімальної сумарної ваги, $k = \overline{1, n}$.

Запропонований метод складається з такого ж числа етапів і має таку ж тимчасову складність, що і найкращий з відомих алгоритмів оптимального призначення – угорський алгоритм [29, 57, 67, 74].

S0. Метод розв'язання ЗП для матриці вартостей $C = [c_{ij}]_n$, $n \geq 2$, елементи якої набувають значень з множини невід'ємних дійсних чисел або дорівнюють ∞ ; розв'язання подано досконалим паросполученням $\pi = \pi_n$ з мінімальною сумою $C(\pi)$ ваг ребер $[i, j]$ в дводольному графі (X, Y, U) , $|X| = |Y| = n$, $i \in X$, $j \in Y$, де $c_{ij} \in R_0^+$, де $c_{ij} \in R_0^+$, якщо $(i, j) \in U$, інакше $c_{ij} = \infty$; $k = 1$; знайти $c_{i_k j_k} = \min\{c_{ij} \mid i, j = \overline{1, n}\}$, $I_k = \{i_k\}$, $J_k = \{j_k\}$, $\pi_k = \{[i_k, j_k]\}$, $C(\pi_k) = c_{i_k j_k}$.

S1. $k = k + 1$; якщо $k > n$, то кінець: побудовано розв'язання ЗП π .

S2. Знайти $c_{i_k j_k} = \min\{c_{ij} \mid i \in X - I_{k-1}, j \in Y - J_{k-1}\}$; якщо $c_{i_k j_k} = \infty$, то покласти $MIN1 = \infty$, інакше $\pi_k^1 = \pi_{k-1} \cup [i_k, j_k]$, $MIN1 = C(\pi_k^1)$.

S3. Знайти усі i_l , такі, що для $j_l \in J_{k-1}$, $l \in \{1, 2, \dots, k-1\}$, $c_{i_l j_l} = \min\{c_{ij} \mid i \in X - I_{k-1}\} \neq \infty$, і сформувати з них список X_k ; якщо $X_k = \emptyset$, то покласти $MIN2 = \infty$ і перейти до кроку S6.

S4. Знайти всі j_p , такі, що для $i_l \in I_{k-1}$, $l \in \{1, 2, \dots, k-1\}$, $c_{i_l j_p} = \min\{c_{ij} \mid j \in J_{k-1}\} \neq \infty$, і сформувати з них список Y_k ; якщо $Y_k = \emptyset$, то покласти $MIN2 = \infty$ і перейти до кроку S6.

S5. Побудувати зважений оргграф (V, A) , $V = \{i_0\} \cup X_k \cup I_{k-1} \cup Y_k$ і виконати в ньому пошук шляху, найкоротшого на множині всіх шляхів у вершини Y_k , що досягаються з i_0 ; якщо побудовано такий шлях, то в графі (X, Y, U) знайти найкоротший збільшуючий шлях P_k , що відповідає йому, відносно паросполучення π_{k-1} і визначити $\pi_k^2 = (P_k - \pi_{k-1}) \cup (\pi_{k-1} - P_k)$, $MIN2 = C(\pi_k^2)$, інакше покласти $\pi_k^2 = \emptyset$, $MIN2 = \infty$.

S6. Якщо $MIN1 = MIN2 = \infty$, то кінець: не існує для матриці $[c_{ij}]_n$ розв'язку ЗН; якщо $MIN1 \neq \infty$ або $MIN2 \neq \infty$, то якщо $MIN1 \leq MIN2$, то $\pi_k = \pi_k^1$, $I_k = I_{k-1} \cup \{i_k\}$, $J_k = J_{k-1} \cup \{j_k\}$, інакше $\pi_k = \pi_k^2$, $I_k = \{i_l \mid l = \overline{1, k}; [i_l, j_l] \in \pi_k^2\}$, $J_k = \{j_l \mid l = \overline{1, k}; [i_l, j_l] \in \pi_k^2\}$ – перейти до кроку S1.

Теорема 2.1. Розв'язання ЗП σ коректно знаходиться побудовою в дводольному графі (X, Y, U) , $|X| = |Y| = n$, що відповідає її матриці вартостей $[c_{ij}]_n$, послідовностей паросполучень π_k , $k = \overline{1, n}$, де π_k – паросполучення, що містить k ребер мінімальної ваги, $\sigma = \pi_n$.

Доведення. Нехай побудовано паросполучення π_{k-1} , $k = \overline{2, n}$. Алгоритм зупиняється, коли:

- не знаходить ребра, об'єднання якого з π_{k-1} давало б π_k^1 ;
- в допоміжному графі немає шляху з вершини i_0 у будь-яку іншу вершину множини Y_k , отже, не існує збільшуючого шляху відносно поточного паросполучення π_{k-1} і π_{k-1} – максимальне [75]. *Доведення завершено.*

Щоб оцінити трудомісткість розв'язання ЗП, відмітимо, що воно будується в результаті виконання n етапів, кожен з яких збільшує паросполучення на одне ребро.

На першому етапі знаходиться π_1 за час, що оцінюється у гіршому разі

величиною $O(n^2)$. На кожному наступному етапі будуються паросполучення π_k^1 і π_k^2 , $k = \overline{2, n}$.

Час побудови π_k^1 оцінюється величиною $O((n-k+1)^2)$. Знаходження π_k^2 потребує $2(n-k+1)$ операцій на пошук вершин множин X_k, Y_k , $k+1$ операцій на побудову допоміжного орграфа, $O(k^2)$ дій на побудову в ньому алгоритмом Дейкстри найкоротшого шляху і $O(k)$ операцій з множинами для визначення π_k^2 . Тому час кожного k -го етапу обмежений величиною $O(n^2)$.

Приклад

Початкова матриця $[c_{ij}]_n$ має вигляд, зображений на рис. 2.7.

	1	2	3	4	5	6	7
1	∞	∞	2	8	9	0	∞
2	15	∞	3	0	24	0	24
3	5	∞	∞	5	0	2	∞
4	∞	10	2	15	23	∞	0
5	∞	0	0	∞	∞	∞	∞
6	0	15	10	3	17	∞	∞
7	14	0	24	2	2	15	21

Рисунок 2.7 – Початкова матриця $[c_{ij}]_n$

$$S0. X=Y=\{1, 2, 3, 4, 5, 6, 7\}, k=1, c_{i_1 j_1} = \min\{c_{ij} \mid i, j = \overline{1, 7}\} = c_{16} = 0,$$

$$I_1 = \{1\}, J_1 = \{6\}, X_1 = Y_1 = \emptyset, \pi_1 = \{[1, 6]\}, C(\pi_1) = c_{16} = 0.$$

$$S1. k = 2.$$

$$S2. c_{i_2 j_2} = \min\{c_{ij} \mid i \neq 1, j \neq 6\} = c_{24} = 0, \pi_2^1 = \pi_1 \cup [2, 4] = \{[1, 6], [2, 4]\},$$

$$MIN1 = C(\pi_2^1) = c_{16} + c_{24} = 0 + 0 = 0.$$

$$S3. \text{ Для } 6 \in J_1 \quad c_{i,6} = \min\{c_{i6} \mid i \neq 1\} = c_{26} = 0, \quad X_2 = \{2\}.$$

$$S4. \text{ Для } 1 \in I_1 \quad c_{1j_p} = \min\{c_{1j} \mid j \neq 6\} = c_{13} = 2, \quad Y_2 = \{3\}.$$

S5. Зважений граф (V, A) є шляхом $(i_0, 2, 1, 3)$, де $2 \in X_2$, $1 \in I_1$, $3 \in Y_2$, $c_{i_0,2} = 0$, $c(2,1) = c_{26} + c_{16} = 0$, $c(1,3) = 2$. Цьому шляху в підграфі графа (X, Y, U) відповідає найкоротший збільшуючий шлях $P_2 = (2, 6, 1, 3)$ відносно паросполучення $\pi_1 = \{[1, 6]\}$.

Шлях P_2 утворюють ребра $(2,6), [1,6], (1,3)$. Тому $\pi_2^2 = \{[2,6], [1,3]\}$, $MIN2 = c_{26} + c_{13} = 0 + 2 = 2$.

$$S6. \text{ Оскільки } MIN1 < MIN2, \text{ то } \pi_2 = \pi_2^1 = \{[1, 6], [2, 4]\}, \quad I_2 = \{1, 2\},$$

$$J_2 = \{6, 4\}.$$

$$S1. k=3.$$

$$S2. c_{i_3j_3} = \min\{c_{ij} \mid i \neq 1, 2; j \neq 6, 4\} = c_{35} = 0, \quad \pi_3^1 = \pi_2 \cup [3, 5] = \{[1, 6], [2, 4], [3, 5]\}, \quad MIN1 = C(\pi_3^1) = 0.$$

$$S3. \text{ Так як } I_2 = \{1, 2\}, J_2 = \{6, 4\}, \text{ то } c_{i,6} = \min\{c_{i6} \mid i \neq 1, 2\} = c_{36} = 2, \quad c_{i,4} = \min\{c_{i4} \mid i \neq 1, 2\} = c_{74} = 2, \text{ и } X_3 = \{3, 7\}.$$

$$S4. c_{1j_p} = \min\{c_{1j} \mid j \neq 6, 4\} = c_{13} = 2, \quad c_{2j_p} = \min\{c_{2j} \mid j \neq 6, 4\} = c_{23} = 3, \quad Y_3 = \{3\}.$$

S5. Підграфу дводольного графа (X, Y, U) , визначеному на множині вершин $X_3 \cup I_2 \cup J_2 \cup Y_3$ (рис. 2.8), відповідає допоміжний граф (рис. 2.9) з вагами дуг $c_{i_0,3} = c_{i_0,7} = 0$, $c(3,1) = c_{36} + c_{16} = 2 + 0 = 2$, $c(1,2) = c_{14} + c_{24} = 8 + 0 = 8$, $c(2,1) = c_{26} + c_{16} = 0 + 0 = 0$, $c(7,2) = c_{74} + c_{24} = 0 + 2 = 2$, $c(1,3) = 2$, $c(2,3) = 3$, $3 \in Y_3$.

Найкоротший шлях $(3, 1, 3 \in Y)$, пов'язує в оргграфі (V, A) вершини із X_3 з

вершинами із Y_3 , складається з ребер $(3, 6)$, $[1, 6]$, $(1, 3)$, підграфа графа (X, Y, U) , які утворюють найкоротший збільшуючий шлях P_3 відносно паросполучення $\pi_2 = \{[1, 6], [2, 4]\}$.

Тому $\pi_3^2 = \{[2, 4], [3, 6], [1, 3]\}$, $MIN2 = c_{24} + c_{36} + c_{13} = 0 + 2 + 2 = 4$, а $MIN1 = c_{16} + c_{24} + c_{35} = 0$, отже $\pi_3 = \pi_3^1$.

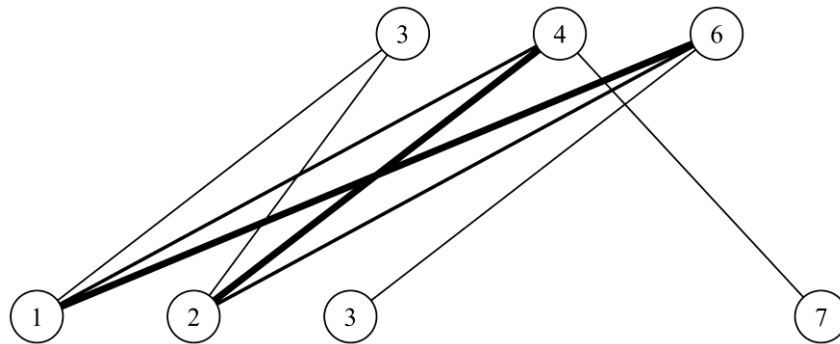


Рисунок 2.8 – Підграф дводольного графа (X, Y, U) , визначеного на множині вершин $X_3 \cup I_2 \cup J_2 \cup Y_3$

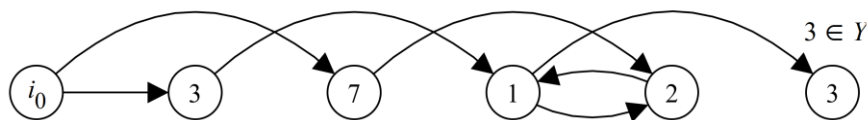


Рисунок 2.9 – Допоміжний граф

Після виконання дій алгоритму для $k=4, 5, 6, 7$ отримуємо розв'язання ЗП $\sigma = \{[1, 6], [2, 4], [3, 5], [4, 7], [5, 3], [6, 1], [7, 2]\}$, $C(\sigma) = 0$.

Часова складність розглянутого методу оцінюється величиною $O(n^3)$, тобто теоретично його трудомісткість не більше трудомісткості найбільш ефективного на сьогодні угорського алгоритму.

Метод знаходить розв'язання ЗП у результаті виконання кінцевого числа звернень до однотипних дій, що спрощують програмну реалізацію і забезпечують

економію обчислювальних ресурсів на великій розмірності даних.

Запропонований метод є прикладом загального підходу до розв'язання класу задач про зважені паросполучення. Підхід полягає в неодноразовому знаходженні збільшуючих шляхів відносно поточного паросполучення для послідовного нарощування його потужності. Метод базується на відомому алгоритмі знаходження максимальних паросполучень в дводольних незважених графах, який побудований за схемою, що розширює способи розв'язання більш трудомістких оптимізаційних задач [48, 75].

Виявляється, що відсутність дводольної структури графа лише трохи ускладнює пошук ЗЗП методом, що представляє модифікацію викладеного методу. Це являє собою модифікацію викладеного методу. Модифікація побудована на ідеї перетворення довільного зваженого графа в такий дводольний граф, в якому б розв'язання ЗЗП збігалося з початковим.

2.3 Обґрунтування рекурентного методу для розв'язання задачі про зважене паросполучення

Нехай $H = (V, U)$ – довільний граф, де V – множина вершин, U – множина ребер (неорієнтованих пар вершин). У H недопустимі петлі, тобто ребра виду $\{v, v\}$, $v \in V$, і кратні або «паралельні» ребра. У додатках теорії графів широкої популярності набула задача про паросполучення. Вона полягає в знаходженні в заданому графі $H = (V, U)$ паросполучення з найбільшим числом ребер – максимального паросполучення. В узагальненні цієї задачі задані ваги ребер – невід'ємні числа, і потрібно визначити максимальне паросполучення графа, що містить ребра з мінімальною (максимальною) сумарною вагою. Сформульоване узагальнення називається задачею про зважене паросполучення (ЗЗП). Окремим випадком ЗЗП є ЗП. Відомо, що ЗЗП поліноміально вирішувана [49, 51, 75]. Класичний алгоритм Едмондса для знаходження найбільшого зваженого паросполучення в недвродольному графі $H = (V, U)$, викладений в [75],

характеризується трудомісткістю $O(|V|^4)$. Головною причиною відносно невисокої швидкодії алгоритму Едмондса є існування в графі H квітів – циклів, вершин, що містять $2k + 1$, і k ребер деякого фіксованого паросполучення M . Виявлена квітка не дозволяє організувати швидкий пошук паросполучення потужності $|M| + 1$ способом, що застосовується для дводольних графів. Для роботи з довільними графами алгоритм Едмондса містить процедуру виявлення квітки і операцію її заміни однією вершиною, допустиму в процесі знаходження поточного паросполучення.

Найбільш ефективні алгоритми знаходження максимальних паросполучень у довільних графах побудовані на розвитку ідей Едмондса про стискання непарних циклів. У них включені способи зберігання даних і організації процесу обчислень, що знижують складність до $O(|V|^3)$ для графів з n вершинами [49].

В умові, що розглядається тут, де ЗЗП – заданий граф $H = (V, U)$, $|V| \in n$, в якому кожне ребро $\{i, j\} \in U$ має вагу $c_{ij} \in R_0^+$, $i, j = \overline{1, n}$, R_0^+ – множина невід’ємних дійсних чисел. Потрібно знайти в графі H максимальне паросполучення з мінімальною сумою ваг ребер. У цій роботі викладається метод розв’язання поставленої ЗЗП, що коректно виконує дії з побудови шуканого паросполучення в дводольному графі, який відповідає графу H . Тому він не містить складні операції знаходження і зрізування квіток, які використовуються в алгоритмі Едмондса і його модифікаціях.

Графу ЗЗП $H = (V, U)$ відповідає симетрична матриця вартостей (ваг) ребер $C = [c_{ij}]_n$, де $c_{ij} \in R_0^+$, якщо $\{i, j\} \in U$, і $c_{ij} = \infty$ в іншому випадку. Ця ж матриця визначає дводольний граф $D = (X, Y, E)$, де X, Y – множина вершин, $|X| = |Y| = n$, $E = \{(i, j) | i \in X, j \in Y\}$, – множина ребер з вагами $c_{ij} \in R_0^+$, $|E| = 2|V|$. Звідси випливає, що для розв’язання поставленої задачі застосовують ідеї пошуку в ширину в дводольних графах [75].

Для викладання методу розв'язання ЗЗП, що розглядається тут, досить скористатися тими визначеннями теорії паросполучень, які наведені в підрозділі 2.1. Слід зауважити, що потужність максимального паросполучення довільного графа H з n вершинами не може бути більше $\lfloor n/2 \rfloor$. Якщо вона дорівнює $\lfloor n/2 \rfloor$, то паросполучення вважається повним. При парному n паросполучення насичує всі вершини H і називається досконалим [75].

Ребро паросполучення M , що пов'язує вершини v і u , позначимо $[v, u]$. У ньому u є напарником v .

Ребра, що не входять у паросполучення M , називаються вільними. Вершина, що належить ребру паросполучення, визначається як насичена. Інші вершини графа називаються ненасиченими, або вільними.

Нехай M – паросполучення в H . Якщо $(v_1, v_2, v_3, \dots, v_{2k-2}, v_{2k-1}, v_{2k})$ – збільшуючий шлях відносно паросполучення M в графі H , то $P = (i_1, j_2, i_3, \dots, j_{2k-2}, i_{2k-1}, j_{2k})$ – збільшуючий шлях в дводольному графі $D = (X, Y, E)$ відносно паросполучення тієї ж потужності, що і M . Шлях P починається в ненасиченій вершині $i_1 \in X$, закінчується в ненасиченій вершині $j_{2k} \in Y$ і містить k вільних ребер $(i_1, j_2), (i_3, j_4), \dots, (i_{2k-1}, j_{2k})$. Інші k ребер шляху P утворюють паросполучення $\{[i_3, j_2], [i_5, j_4], \dots, [i_{2k-1}, j_{2k-2}]\}$.

На рис. 2.10, а зображено збільшуючий шлях у графі $H = (V, U)$ відносно паросполучення $\{[v_2, v_3], [v_4, v_5]\}$, а на рис. 2.10, б – збільшуючий шлях P , що відповідає йому в графі $D = (X, Y, U)$ відносно паросполучення $\{[i_3, j_2], [i_5, j_4]\}$. Тут $k = 3$.

Ребра паросполучень наведені потовщеними лініями. Тонкими лініями зображені ребра графа D , не належні P .

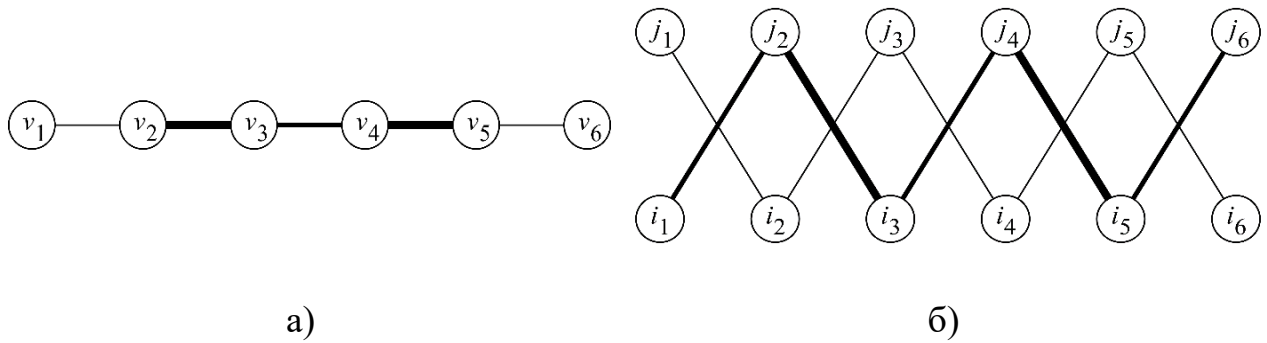


Рисунок 2.10 – Збільшуючі шляхи: а – збільшуючий шлях у графі $H = (V, U)$ відносно паросполучення $\{[v_2, v_3], [v_4, v_5]\}$; б – збільшуючий шлях P , що відповідає йому в графі $D = (X, Y, U)$ відносно паросполучення $\{[i_3, j_2], [i_5, j_4]\}$

Процедура знаходження збільшуючого шляху є варіантом методу пошуку в ширину, що базується на такому відомому факті: якщо P – множина ребер збільшуючого шляху щодо паросполучення M в графі H , то $M \oplus P$ – паросполучення потужності $|M| + 1$. Наприклад, зі збільшуючого шляху відносно паросполучення $\{[v_2, v_3], [v_4, v_5]\}$ (рис. 2.10, а) впливає паросполучення $\{[v_1, v_2], [v_3, v_4], [v_5, v_6]\}$. Множина ребер $\{[i_1, j_2], [i_3, j_2], [i_3, j_4], [i_5, j_4], [i_5, j_6]\}$ утворює збільшуючий шлях $P = (i_1, j_2, i_3, j_4, i_5, j_6)$ відносно паросполучення $M = \{[i_3, j_2], [i_5, j_4]\}$, визначаючи паросполучення $M \oplus P = \{[i_1, j_2], [i_3, j_4], [i_5, j_6]\}$.

Шлях $P = (i_1, j_2, \dots, i_{2k-1}, j_{2k})$ у графі D є простим ланцюгом, ізоморфним у графі H ланцюга $(v_1, v_2, \dots, v_{2k})$ при відповідності $(v_s, v_{s+1}) \Leftrightarrow (i_s, j_{s+1})$, $s = 1, 3, \dots, \dots, 2k-1$, і $[v_{2s}, v_{2s+1}] \Leftrightarrow [j_{2s+1}, i_{2s}]$, $s = 1, 2, \dots, k-1$ (рис. 2.10).

Паросполучення M і шлях P утворюють паросполучення $M \oplus P = \{[i_1, j_2], [i_3, j_4], \dots, [i_{2k-1}, j_{2k}]\}$.

Допустимий розв'язок ЗЗП – це максимальне паросполучення зваженого графа H . Пошук максимального паросполучення в H яким-небудь методом завершується при виконанні умови теореми, в такому формулюванні.

Паросполучення M у графі H максимально тоді і тільки тоді, коли в H не існує збільшуючого шляху відносно M [75]. Збільшуючий шлях відносно паросполучення M графа H називається найкоротшим, якщо його вартість не більше вартості будь-якого збільшуючого шляху відносно M . Розв'язанням ЗЗП є максимальне паросполучення M_{opt} мінімальної вартості в графі H .

Нехай $M_{k-1} = \{[i_1, j_2], [i_3, j_4], \dots, [i_{2k-3}, j_{2k-2}]\}$ – паросполучення з найменшою сумою $C(M_{k-1})$ ваг $k-1$ ребер на множині всіх паросполучень потужності $k-1$ в дводольному графі D $k \geq 2$. У графі H йому взаємно однозначно відповідає паросполучення $\{[v_1, v_2], [v_3, v_4], \dots, [v_{2k-3}, v_{2k-2}]\}$.

Нехай $M_{k-1} = \{[i_1, j[i_1]], [i_2, j[i_2]], \dots, [i_l, j[i_l]], \dots, [i_{k-1}, j[i_{k-1}]]\}$, $i_l \neq j[i_l]$, де i_l – номер вершини множини X , $j[i_l]$ – номер вершини множини Y . В M_{k-1} всі $2k-2$ вершини пронумеровані різними числами множини $\{1, 2, \dots, n\}$, $k \leq \lfloor n/2 \rfloor$. Позначимо $M_k^1 = M_{k-1} \cup \{[i_k, j[i_k]]\}$ – паросполучення, що містить ребро $[i_k, j[i_k]]$ з найменшою вагою серед усіх ребер, які можна приєднати до M_{k-1} , P_k – найкоротший збільшуючий шлях відносно паросполучення M_{k-1} , $M_k^2 = M_{k-1} \oplus P_k$, $C(M_k^1)$ и $C(M_k^2)$ – вартості паросполучень M_k^1 і M_k^2 . Справедливо таке твердження, доказ якого відрізняється тільки позначеннями від доказу твердження в [124].

Лема 2.2. $C(M_k^2) \leq C(M_k^1)$, то $C(M_k) = C(M_k^2)$, інакше $C(M_k) = C(M_k^1)$, M_k – паросполучення з мінімальною сумою ваг k ребер у графі D .

Очевидно, для деякого k паросполучення M_k максимальне. Тоді $M_{opt} = M_k$ у графі $H = (V, U)$. Наведений метод розв'язання ЗЗП полягає в покроковому знаходженні в графі H паросполучень M_k , $k = \overline{1, M_{opt}}$, в результаті побудови в дводольному графі D кожного найкоротшого збільшуючого шляху P_k відносно

M_k , знаходження паросполучень M_{k+1}^1 і $M_{k+1}^2 = M_k \oplus P_k$ і вибору з них M_{k+1} .

Паросполучення M_1 складається з одного ребра, вага якого дорівнює мінімальному значенню в матриці C .

Якщо матриця C містить декілька елементів мінімальної ваги, то $M_1 = \{[i_1, j[i_1]]\}$, $c_{i_1 j[i_1]} = \min\{c_{ij} \mid i, j = \overline{1, n}\}$, i_1 – номер першого по порядку рядка, якому належить $c_{i_1 j[i_1]}$.

У матриці C паросполучення M_2 з мінімальною сумою ваг двох ребер $C(M_2) \neq \infty$ визначається із співвідношення:

$$C(M_2) = \min\{C(M_2^1), C(M_2^2)\}. \quad (2.5)$$

Паросполучення M_2^1 містить ребро $[i_1, j[i_1]]$ вагою $c_{i_1 j[i_1]}$ і ребро $[i_1^1, j[i_1^1]]$ мінімальної ваги $c_{i_1^1 j[i_1^1]}$ в підматриці, отриманій видаленням з матриці C рядків і стовпців із номерами $i_1, j[i_1]$ (рис. 2.11, а), $c_{i_1^1 j[i_1^1]} = \min\{c_{ij} \mid i, j \notin \{i_1, j[i_1]\}\}$.

У паросполучення M_2^2 (рис. 2.11, б) входить ребро $[i_1, s]$ вагою $c_{i_1 s} = \min\{c_{i_1 j} \mid j \neq j[i_1]\}$ і ребро $[r, j[i_1]]$ вагою $c_{r j[i_1]} = \min\{c_{ij} \mid i \neq i_1\}$.

Коли матриця C містить не менше двох елементів мінімальної ваги, вибір серед них елемента з найменшим номером рядка позбавляє єдиний випадок втрати оптимального розв'язання M_2 .

Значення (2.5) може не досягти мінімуму, якщо:

$$c_{l, l+1} = c_{l+1, l+2} = \min\{c_{ij} \mid i, j = \overline{1, n}\}, \quad 1 \leq l \leq n-3,$$

$$c_{l+2, l+3} < c_{rs} = \min\{c_{ij} \mid i, j \neq l+1, l+2\}.$$

Дійсно, при $c_{i_1 j[i_1]} = c_{l+1, l+2}$ паросполучення $M'_2 = \{[l+1, l+2], [r, s]\}$ отримає вагу $C(M'_2) = c_{l+1, l+2} + c_{r, s}$. Ця вага більше ваги паросполучення $M_2^1 = \{[l, l+1], [l+2, l+3]\}$ (рис. 2.11, б).

На рис. 2.12 зображено паросполучення M_2^1 в дводольному графі D , на рис. 2.13 – підграф графа D , що містить паросполучення $M_2^2 = M_1 \oplus P_1$, де $P_1 = ((r, j[i_1]), [i_1, j[i_1]], (i_1, s))$ – множина ребер найкоротшого збільшуючого шляху відносно M_1 .

Таким чином, в дводольному графі D $M_2 = M_2^1$, якщо $C(M_2^1) = c_{i_1 j[i_1]} + c_{i_1 j[i_1^1]} < C(M_2^2) = c_{i_1 s} + c_{r j[i_1]}$ і $M_2 = M_2^2$ в інших випадках.

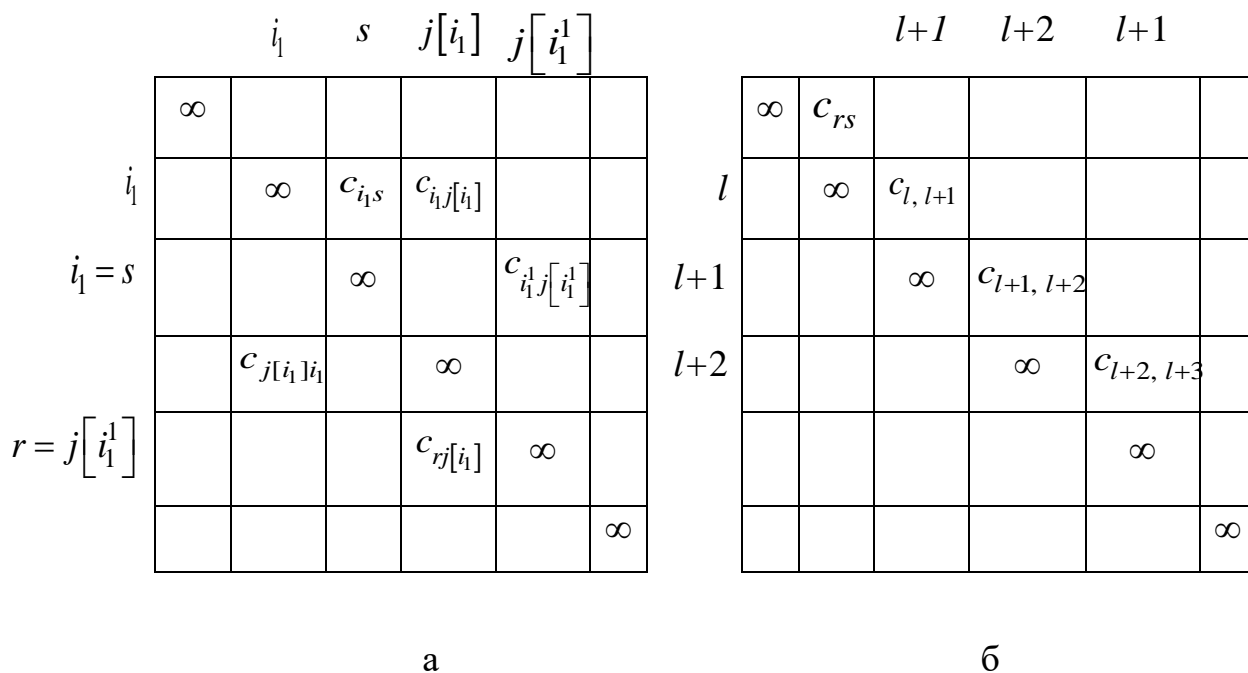


Рисунок. 2.11 – Паросполучення M_2^1 і M_2^2 : а – паросполучення M_2^1 ; б – паросполучення M_2^2

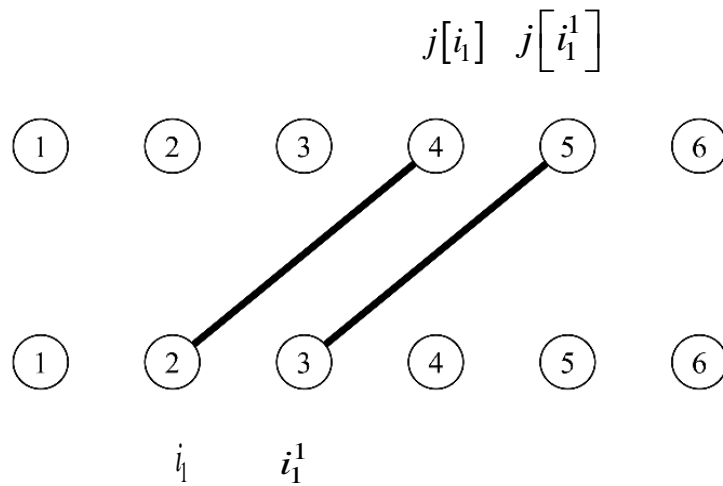


Рисунок 2.12 – Паросполучення M_2^1 в дводольному графі D

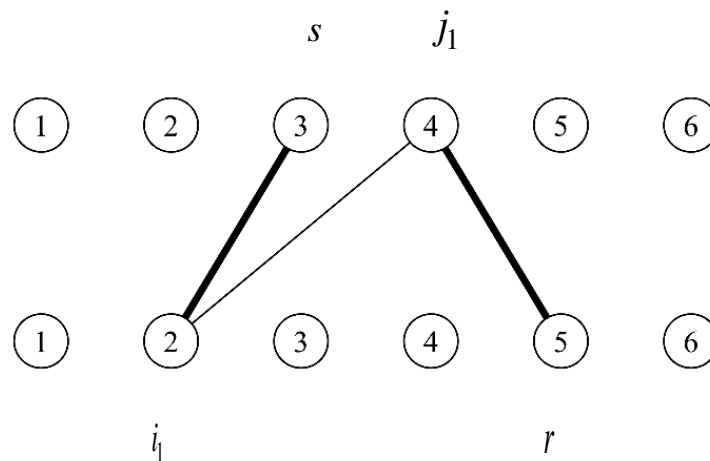


Рисунок 2.13 – Підграф графа D , що містить паросполучення $M_2^2 = M_1 \oplus P_1$, де $P_1 = ((r, j[i_1]), [i_1, j[i_1]], (i_1, s))$ – множина ребер найкоротшого збільшуючого шляху відносно M_1

За матрицею C і паросполученням M_2^1, M_2^2 знайдемо паросполучення M_3 з мінімальною сумою ваг трьох ребер $C(M_3)$.

Аналогічно (2.5) наведемо

$$C(M_3) = \min \{C(M_3^1), C(M_3^2)\}.$$

Щоб отримати паросполучення M_3 і $C(M_3)$, знайдемо $c_{i_2j[i_2]} =$
 $= \min \left\{ c_{ij} \mid i, j \notin \{i_1, j[i_1], i_1^1, j[i_1^1]\} \right\}$.

Нехай $C(M_2^1) = c_{i_1j[i_1]} + c_{i_1^1j[i_1^1]} < C(M_2^2) = c_{i_1s} + c_{rj[i_1]}$.

Тоді для паросполучень $M_3^1 = M_2^1 \cup \{[i_2, j[i_2]]\}$ і $M_3' = M_2^2 \cup \{[i_2, j[i_2]]\}$ справедливою є нерівність

$$C(M_3^1) = C(M_2^1) + c_{i_2j[i_2]} \leq C(M_3') = c_{i_1s} + c_{rj[i_1]} + c_{i_2j[i_2]}.$$

Для вибору точнішої верхньої межі $C(M_3^1)$ вартості $C(M_3)$ оптимального поточного розв'язання M_3 , чим $C(M_3')$, побудуємо найкоротший збільшуючий шлях P_2 відносно M_2^1 , знайдемо $M_3^2 = M_2^1 \oplus P_2$ і $C(M_3^2)$.

Очевидно, $M_3 = M_3^1$, якщо $C(M_3^1) < C(M_3^2)$, і $M_3 = M_3^2$ – інакше.

Припустимо, що

$$C(M_2^2) = c_{i_1s} + c_{rj[i_1]} \leq C(M_2^1) = c_{i_1j[i_1]} + c_{i_1^1j[i_1^1]}.$$

Оскільки $c_{i_1^1j[i_1^1]} \leq c_{i_2j[i_2]}$, то вартості паросполучень $M_3^1 = M_2^1 \cup \{[i_1^1, j[i_1^1]]\}$ і $M_3' = M_2^2 \cup \{[i_2, j[i_2]]\}$ задовольняють нерівності $C(M_3^1) \leq C(M_3')$. Тому величина $C(M_3^1)$ є точнішою оцінкою зверху $C(M_3)$, чим $C(M_3')$.

Для знаходження M_3 побудуємо найкоротший збільшуючий шлях P_2 відносно M_2^2 , визначимо паросполучення $M_3^2 = M_2^2 \oplus P_2$ і його вартість $C(M_3^2)$.

Отже, $C(M_3^1) < C(M_3^2)$, і $M_3 = M_3^2$ – інакше.

Викладений спосіб знаходження M_3 і $C(M_3)$ укладається у схему розв'язання ЗЗП за допомогою рівності:

$$C(M_k) = \min \{C(M_k^1), C(M_k^2)\}, 2 \leq k \leq |M_{opt}|. \quad (2.6)$$

При $k = |M_{opt}|$ хоч би одне зі значень $C(M_k^1)$ або $C(M_k^2)$ досягає $C(M_{opt})$; $M_k = M_{opt}$, якщо при деякому k $C(M_k) \neq \infty$, а при $k+1$ $C(M_{k+1}^1) = C(M_{k+1}^2) = \infty$.

Назвемо вершину $j_l \in Y$ відображенням у графі D початку i_l ребра $[i_l, [j[i_l]]]$ паросполучення $M_{k-1} = \{[i_1, j[i_1]], [i_2, j[i_2]], \dots, [i_l, j[i_l]], \dots, [i_{k-1}, j[i_{k-1}]]\}$ і позначимо її j'_l . Вершину $i_m \in X$ назвемо відображенням $j[i_l]$ цього ребра і позначимо її i'_m . Нехай $I_{k-1} = \{i_l \mid l = \overline{1, k-1}\}$, $J_{k-1} = \{j[i_l] \mid l = \overline{1, k-1}\}$ – множина вершин паросполучення M_{k-1} , а I'_{k-1} , J'_{k-1} – множина їх відображень I_{k-1} , $I'_{k-1} \subset X$, J_{k-1} , $J'_{k-1} \subset Y$. Побудова M_{opt} розпочинається зі знаходження $M_1 = \{[i_1, j[i_1]]\}$ і видалення ребер, інцидентних відображенням j'_1 і i'_1 вершин i_1 і $j[i_1]$ відповідно. У матриці C видалені ребра набувають ваги, яка дорівнює ∞ .

Щоб з (2) визначити M_k , спочатку знаходять $M_k^1 = M_{k-1} \cup \{[i_k, j[i_k]]\}$, де

$$c_{i_k j[i_k]} = \min \{c_{ij} \mid i \notin I_{k-1} \cup I'_{k-1}, j \notin J_{k-1} \cup J'_{k-1}\}. \quad (2.7)$$

У підграфі графа D видаляються ребра, інцидентні відображенням j'_k , i'_k вершин i_k і $j[i_k]$ відповідно. Кожне ребро, що видаляється, отримує в матриці C вагу, яка дорівнює ∞ . Щоб знайти M_k^2 , для кожної вершини $i_k \notin I_{k-1} \cup I'_{k-1}$

формується підграф $D_{i_k} = (X_{i_k}, Y_{i_k}, E_{i_k})$ графа D , який містить підмножину $E_{i_k}^1$ вільних ребер $(i_k, j[l_i])$ $l \in \{1, 2, \dots, k-1\}$, підмножину $E_{i_k}^2$ усіх ребер, що з'єднують вершини множини I_{k-1} з вершинами множини J_{k-1} , і підмножини $E_{i_k}^3$ вільних ребер (i_l, j_s) , $j_s \neq j_k$, $j_s \in \{Y - \{J_{k-1} \cup J_{k-1} - \{j_k\}\}\}$, $l \in \{1, 2, \dots, k-1\}$, з вагами

$$c_{ij_s} = \min \{c_{ij} \mid j \in Y - \{J_{k-1} \cup J'_{k-1}\} - \{j_k\}\}, \quad (2.8)$$

вершин $Y_{i_k}^1$, що утворюють множину. Отже, підграф $D_{i_k} = (X_{i_k}, Y_{i_k}, E_{i_k})$ складається з множини вершин $X_{i_k} = \{i_k\} \cup I_{k-1}$, $Y_{i_k} = Y_{i_k}^1 \cup J_{k-1}$ і множини ребер $E_{i_k}^1 \cup E_{i_k}^2 \cup E_{i_k}^3$. Підграф D_{i_k} показано на рис. 2.14. Вільні ребра $(i_k, j[l_1])$, $(i_k, j[l_l])$, $(i_k, j[l_{k-1}])$ утворюють підмножину $E_{i_k}^1$. Ребра $(i_1, j[l_l])$, $(i_1, j[l_{k-1}])$, $(i_l, j[l_l])$, $(i_l, j[l_{k-1}])$, $(i_{k-1}, j[l_l])$ та всі ребра паросполучення M_{k-1} входять у підмножину $E_{i_k}^3$. Підмножина $E_{i_k}^3$ включає наступні ребра (i_1, j_s) , (i_l, j_r) , (i_{k-1}, j_r) . В підграфі $D_{i_k} = (X_{i_k}, Y_{i_k}, E_{i_k})$ $X_{i_k} = \{i_k, i_1, i_l, i_{k-1}\}$, $Y_{i_k}^1 = \{j_s, j_r\}$, $Y_{i_k} = Y_{i_k}^1 \cup \{j[l_l], j[l_l], j[l_{k-1}]\}$. Підграф D_{i_k} будується для знаходження в ньому шляху P_{i_k} , найкоротшого серед усіх збільшуючих шляхів відносно паросполучення M_{k-1} . Цей шлях має починатися у вершині i_k і закінчуватися в деякій вершині $j_s \in Y_{i_k}^1$, $j_s \neq j_k$.

Якщо існує шлях P_{i_k} , то згідно з лемою $M_{i_k} = P_{i_k} \oplus M_{k-1}$ — паросполучення, що доставляє мінімальну суму ваг k ребер у підграфі D_{i_k} .

У початковому графі H йому відповідає паросполучення тієї ж потужності і з такими ж вагами ребер, що і в D_{i_k} .

Побудова підграфа D_{i_k} і пошук у ньому шляху P_{i_k} повторюється для кожної вершини $i_k \in X - \{I_{k-1} \cup I'_{k-1}\}$ і завершується вибором паросполучення M_k^2 вартістю

$$C(M_k^2) = \min \{C(M_{i_k}^2) | i_k \notin I_{k-1} \cup I'_{k-1}\}. \quad (2.9)$$

Пошук шляху P_{i_k} спрощується в допоміжному орграфі (Z_{i_k}, A_{i_k}) , отриманому в результаті перетворення D_{i_k} .

Орграф (Z_{i_k}, A_{i_k}) складається з множини вершин $Z_{i_k} = \{i_k\} \cup I_{k-1} \cup Y_{i_k}^1$ і множини дуг $A_{i_k} = A_{i_k}^1 \cup A_{i_k}^2 \cup A_{i_k}^3$. У підмножину $A_{i_k}^1$ входить дуга (i_k, i_l) , $i_k \in X - \{I_{k-1} \cup I'_{k-1}\}$, $i_l \in I_{k-1}$, тоді і тільки тоді, коли вершина $j[i_l]$ ребра $(i_d, j[i_l])$, є напарником вершини i_l . Дузі (i_k, i_l) привласнюється вага $c(i_k, i_l) = c_{i_k j[i_l]} + c_{i_l j[i_l]}$. Дуга (i_d, i_l) , $i_d, i_l \in I_{k-1}$, входить в $A_{i_k}^2$, якщо і тільки якщо вершина $j[i_l]$ ребра $(i_d, j[i_l])$, є напарником вершини i_l .

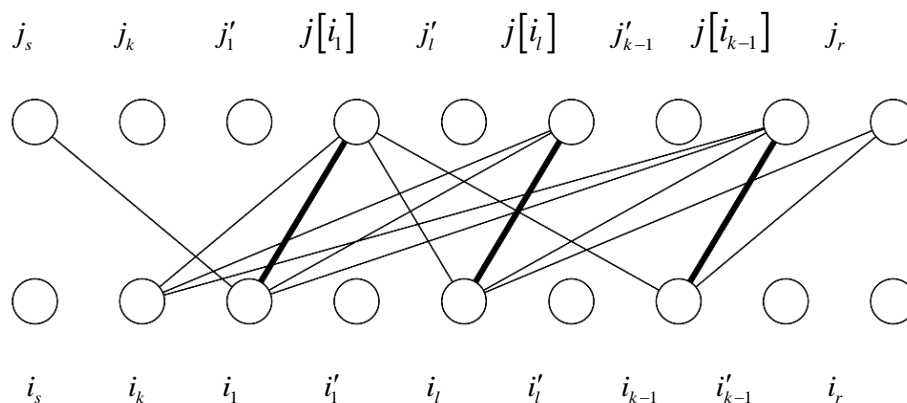


Рисунок 2.14 – Підграф D_{i_k}

Дуга (i_d, i_l) отримує вагу $c(i_d, i_l) = c_{i_d j[i_l]} + c_{i_l j[i_l]}$. Підмножина $A_{i_k}^3$ містить

усі дуги (i_l, j_s) , $i_l \in I_{k-1}$, $j_s \in Y_{i_k}^1$, якщо вершини i_l і j_s з'єднані ребром у D_{i_k} . Дуга (i_l, j_s) має вагу $c(i_l, j_s) = c_{i_l j_s}$ (рис. 2.15).

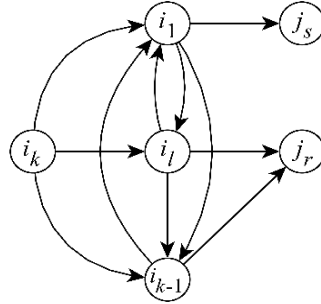


Рисунок 2.15 – Орграф (Z_{i_k}, A_{i_k})

Неважно переконатися, що при невід'ємних вагах ребер графа D шлях P_{i_k} – найкоротший шлях серед усіх простих шляхів з вершини i_k у вершини множини $Y_{i_k}^1$ орграфа (Z_{i_k}, A_{i_k}) . Пошук P_{i_k} виконується за алгоритмом Дейкстри.

Позначимо $\langle D_k \rangle$ підграф дводольного графа (X, Y, E) , породжений множинами вершин $I_k = \{i_l \mid l = \overline{1, k}\}$ і $J_k = \{j[i_l] \mid l = \overline{1, k}\}$ паросполучення $M_k = \{[i_l, j[i_l]] \mid l = \overline{1, k}\}$.

2.4. Рекурсивний метод побудови максимального паросполучення мінімальної ваги в довільному графі

Метод розв'язання ЗЗП складається з таких кроків.

S0. Метод знаходження в зваженому графі $H = (V, U)$ максимального паросполучення M_{opt} з мінімальною сумою ваг його ребер. $C = [c_{ij}]_n$ – симетрична матриця ваг ребер графа H , в якій $c_{ij} = c_{ji} \in R_0^+$, якщо $\{i, j\} \in U$, $i \neq j$,

а $c_{ij} = c_{ji} = \infty$ інакше, R_0^+ – множина невід’ємних дійсних чисел. Розв’язання M_{opt} взаємно однозначно відповідає максимальному паросполученню $M_k = \{[i_l, j[i_l]] \mid i_l \neq j[i_l], l = \overline{1, k}\}$ дводольного зваженого графа (X, Y, E) , $|X| = |Y| = |V|$, $|E| = 2|U|$, $i_l \in X, j[i_l] \in Y$, побудованого для матриці C .

У матриці C знайти $c_{i_l j[i_l]} = \min\{c_{ij} \mid i, j = \overline{1, n}\}$, i_l – номер першого по порядку рядка, що містить мінімальний елемент; $M_1 = \{[i_1, j[i_1]]\}$, $I_1 = \{i_1\}$, $J_1 = \{j[i_1]\}$, $I'_1 = \{j[i_1]\} \subset X$, $J'_1 = \{i_1\} \subset Y$, видалити всі ребра, інцидентні вершинам $j[i_1] \in I'_1$, $i_1 \in J'_1$; D_1 – підграф, що містить ребро $[i_1, j[i_1]]$, $\langle D_1 \rangle = D_1$, $k = 1$.

S1. $k = k + 1$; якщо $k > \lfloor n/2 \rfloor$, то $M_{opt} = M_{k-1}$.

S2. Знайти $c_{i_k j[i_k]}$ за формулою (3); якщо $c_{i_k j[i_k]} = \infty$, то кінець: $M_{opt} = M_{k-1}$; $M_k^1 = M_{k-1} \cup \{[i_k, j[i_k]]\}$, вчислити $C(M_k^1)$.

S3. Для кожної вершини $i_k \in X - \{I_{k-1} \cup I'_{k-1}\}$ побудувати підграф D_{i_k} і після перетворення його в допоміжний оргграф (Z_{i_k}, A_{i_k}) знайти шлях P_{i_k} , найкоротший із шляхів, що з’єднують вершину i_k з вершинами $j_s \in Y - \{J_{k-1} \cup J'_{k-1}\} - \{j_k\}$. Кожна вершина j_s є кінцем ребра (i_l, j_s) , $i_l \in I_{k-1}$, $l \in \{1, 2, \dots, k-1\}$ вагою $c_{i_l j_s}$, яка обчислюється (2.8). Визначити $M_{i_k}^2 = P_{i_k} \oplus M_{k-1}$ і $C(M_{i_k}^2)$. Якщо в графі D_{i_k} не існує шляху P_{i_k} , то покласти $M_{i_k}^2 = \infty$. З (2.9) знайти M_k^2 і $C(M_k^2)$; якщо для усіх i_k $C(M_{i_k}^2) = \infty$, то $C(M_k^2) = \infty$, $M_k^2 = \emptyset$.

S4. Якщо $C(M_k^1) = C(M_k^2) = \infty$, то кінець: $M_{opt} = M_{k-1}$, інакше якщо $C(M_k^1) < C(M_k^2)$, то $M_k = M_k^1$, $I_k = I_{k-1} \cup \{i_k\}$, $J_k = J_{k-1} \cup \{j[i_k]\}$,

$I'_k = I'_{k-1} \cup \{i'_k\}$, $J'_k = J'_{k-1} \cup \{j'_k\}$, де вершини $i'_k \in X$ і $j'_k \in Y$ – відображення відповідно кінця $j[i_k]$ і початку i_k ребра $[i_k, j[i_k]]$; видалити ребра, інцидентні вершинам i'_k і j'_k , сформувати підграф $\langle D_k \rangle$, породжений множиною вершин $I_k \cup J_k$, і перейти до кроку S1, інакше $M_k = M_k^2$, визначити I_k, J_k, I'_k, J'_k , видалити усі ребра, інцидентні вершинам множини $I'_k \cup J'_k$ і сформувати підграф $\langle D_k \rangle$, породжений множиною вершин $I_k \cup J_k$; перейти до кроку S1.

Метод наведений послідовністю ітерацій. Перша ітерація алгоритму завершується на кроці S0 знаходженням мінімального елемента в C , що утворює паросполучення M_1 . Кожна наступна ітерація містить кроки S2–S4 для побудови паросполучення M_k , $k = 2, \overline{M_{opt}}$ з мінімальною сумою (M_k) ваг k ребер.

Теорема 2.2. Якщо після впорядкування за неспаданням значень елементів, розташованих над головною діагоналлю матриці вартостей $C = [c_{ij}]_n$ графа H , тоЗЗП коректно вирішується за час $O(n^3)$.

Доведення. Згідно з лемою 2.2 треба спочатку показати, що $M_k = M_k^2$, якщо $C(M_k^2) \leq C(M_k^1)$. Дійсно, для кожної вільної вершини i_k метод буде підграф D_{i_k} , що містить множину всіх збільшуюючих з i_k шляхів відносно паросполучення M_{k-1} , $k = 2, \overline{M_{opt}}$, вибирає серед них найкоротший шлях P_{i_k} і визначає $M_{i_k}^2$ і $C(M_{i_k}^2)$. Коректність одноразового звернення до i_k виходить з наведеного в [75] доказу того, що якщо в графі не існує збільшуючого шляху з вільної вершини, то збільшуючого шляху з цієї вершини не існує на всіх подальших етапах побудови паросполучення. *Доведення завершено.*

Метод завершує роботу на k -ой ітерації. Якщо $k = \lfloor n/2 \rfloor$, то побудоване паросполучення M_k є максимальним. Інакше не існує шляху з кожної вільної вершини i_k в орграфі (Z_{i_k}, A_{i_k}) , що відповідає їй, а в D не існує збільшуючих

шляхів відносно поточного паросполучення M_{k-1} . Звідси виходить, що M_{k-1} є максимальним [75].

Розв'язання M_{opt} ЗЗП визначає із матриці C , що відповідає як дводольному, так і довільному графам $D=(X, Y, E)$ і $H=(V, U)$.

Оцінимо зверху час роботи методу. Воно максимальне, коли n -вершинний граф H повний і тому $|M_{opt}| = \lfloor n/2 \rfloor$.

Якщо впорядкувати за неспаданням значення елементів, розташованих над головною діагоналлю матриці C , то для виконання кроку S0, який визначає $c_{i_1 j[i_1]}$, знадобиться час $O\left(\left(n(n-1)/2\right)\log_2\left(n(n-1)/2\right)\right)$.

Найбільше число операцій порівняння на кроці S2 k -ї ітерації, $k = \overline{2, \lfloor n/2 \rfloor}$, яке дорівнює $2(n-2(k-1))$, досягається на матриці C , де $c_{12} = c_{i_1 j[i_1]}$, $c_{34} = c_{i_2 j[i_2]}$, ..., $c_{2k-1, 2k} = c_{i_k j[i_k]}$. Щоб визначити $\lfloor n/2 \rfloor - 1$ елементів матриці C , що утворюють разом з елементом $[i_1, j[i_1]]$ паросполучення M_k^1 , $k = \lfloor n/2 \rfloor$, необхідно виконати $t_1 = 2(n-2) + 2(n-4) + \dots + 2(n-2(\lfloor n/2 \rfloor - 1))$ порівнянь. Тому $t_1 = O(n^2)$.

Побудова графа D_{i_k} на кроці S3 при $k = 2$ потребує $n-4$ порівнянь для знаходження (2.8) і одну операцію складання для обчислення $C(M_{i_k}^2)$. На цьому кроці при $k = 2$ треба побудувати $n-2$ графів D_{i_k} , визначити $n-2$ значень $C(M_{i_k}^2)$, що відповідають їм, і знайти (2.9). Тому $C(M_2^2)$ визначають в результаті виконання $(n-3)(n-2) + n-3 = (n-2)^2 - 1$ операцій. Для знаходження (2.5) на кроці S4 потрібно одне порівняння. Таким чином, обчислення $C(M_2)$ і побудова M_2 виконуються за час $t_2 = O(n^2)$.

Для $k = \overline{3, \lfloor n/2 \rfloor}$ на кроці S3 треба побудувати $n-2(k-1)$ графів D_{i_k} .

Граф D_{i_k} будується в результаті виконання $n - 2(k - 1) - 2$ порівнянь за знаходженням (2.8) для кожної з $k - 1$ його вершин $i_l \in I_{k-1}$. Крім того, на кроці S3 він за час $c_1(k - 1)$ перетвориться в оргграф (Z_{i_k}, A_{i_k}) , в якому з трудомісткістю $c_2(k - 1)^2$ шукається шлях P_{i_k} , його довжина $C(P_{i_k})$ і паросполучення $M_{i_k}^2$ вартістю $C(M_{i_k}^2)$, $c_1, c_2 < k - 1$.

Щоб вибрати M_k , досить $n - 2(k - 1) - 1$ порівнянь для знаходження $C(M_k^2)$ і одну операцію порівняння $C(M_k^1)$ и $C(M_k^2)$, яку виконують на кроці S4. Крок S4 завершується формуванням графа $\langle D_k \rangle$ за час c_3k , $c_3 < k$.

Визначимо число операцій t_{k1} , виконуваних на кроці S3, і число операцій t_{k2} , виконуваних на кроці S4, для $k = 3, \overline{\lfloor n/2 \rfloor}$:

$$t_{k1} = \lceil n - 2(k - 1) \rceil \lceil (k - 1)(n - 2k) + c_1(k - 1) \rceil + n - 2(k - 1) + c_3k,$$

$$t_{k2} = \lceil n - 2(k - 1) \rceil c_2(k - 1)^2.$$

Оскільки для будь-якого n і $k = 3, \overline{\lfloor n/2 \rfloor}$

$$\lceil n - 2(k - 1) \rceil (k - 1)(n - 2k) \leq n^2,$$

$$\lceil n - 2(k - 1) \rceil c_1(k - 1) \leq \lceil n - 2(k - 1) \rceil (k - 1)^2 \leq n^2,$$

$$n - 2(k - 1) + c_3k \leq n - 2(k - 1) + k^2 = c_{k1}n, \quad c_{k1} < n,$$

$$\lceil n - 2(k - 1) \rceil c_2(k - 1)^2 \leq \lceil n - 2(k - 1) \rceil (k - 1)^3 = c_{k2}n^2, \quad c_{k2} < n,$$

$$t_{k1} \leq 2n^2 + c_{k1}n = O(n^2), \quad t_{k2} = O(n^2).$$

Трудомісткість виконання $\lfloor n/2 \rfloor - 1$ ітерацій, починаючи з другої, оцінюється числом елементарних дій, яке дорівнює сумі з $2\lfloor n/2 \rfloor$ доданків, $t = t_1 + t_2 + \sum_{k=3}^{\lfloor n/2 \rfloor} (t_{k1} + t_{k2})$, в якій кожен доданок обмежений поліномом другої степені. Тому $t = O(n^3)$.

Приклад

Для демонстрації роботи алгоритму вибрано приклад ЗЗП з [75] з матрицею вартостей повного графа, показаного на рис. 2.16.

$$S0. c_{23} = \min\{c_{ij} \mid i, j = \overline{1,8}\} = 0, M_1 = \{[2,3]\}, I_1 = \{2\}, J_1 = \{3\}, I'_1 = \{3\}, J'_1 = \{2\},$$

$$C(M_1) = c_{23} = 0.$$

У графі D видаляються ребра, інцидентні вершинам $3 \in I'_1$ і $2 \in J'_1$, утворюючи підграф D_1 .

$$S1. k = 2.$$

$$S2. c_{45} = \min\{c_{ij} \mid i, j \neq 2, 3\} = 2, M_2^1 = M_1 \cup \{[4,5]\} = \{[2,3], [4,5]\}, C(M_2^1) = 2.$$

	1	2	3	4	5	6	7	8
1	∞	19	8	8	18	18	25	29
2	19	∞	0	8	10	4	15	18
3	8	0	∞	4	8	2	15	18
4	8	8	4	∞	2	10	15	16
5	18	10	8	2	∞	10	22	25
6	18	4	2	10	10	∞	19	19
7	25	15	15	15	22	19	∞	37
8	29	18	18	16	25	19	37	∞

Рисунок 2.16 – Матриця вартостей повного графа

S3. Для вершин 1, 4, 5, 6, 7, 8, що утворюють множину $X - (I_1 \cup I'_1)$,

будуються підграфи $D_1, D_4, D_5, D_6, D_7, D_8$ і оргграфи, що відповідають їм, в яких виконується пошук найкоротших збільшуючих шляхів $P_1, P_4, P_5, P_6, P_7, P_8$ відносно паросполучення M_i^2 . Для кожного шляху $P_i, i \in X - (I_1 \cup I_1')$, якщо він існує, визначається паросполучення M_i^2 і його вартість $C(M_i^2)$.

На рис. 2.17 зображено підграф D_4 , на рис. 2.18 – допоміжний оргграф (Z_4, A_4) . Єдине ребро $(2, 6)$ в D_4 утворює підмножину вільних ребер E_4^3 . Його вага $c_{26} = 4$ визначається з (2.8) $Y_4^1 = \{6\}$.

Множина ребер $\{(4, 3), [2, 3], (2, 6)\}$ шляху $P_4 = (4, 2, 6) \in (Z_4, A_4)$ і паросполучення $M_1 = \{[2, 3]\}$ утворюють паросполучення $M_4^2 = \{[4, 3], [2, 6]\}$ вартістю $C(M_4^2)$, отриманої з (2.9).

Таким чином, $M_2^2 = M_4^2, C(M_2^2) = c_{43} + c_{16} = 4 + 4 = 8$.

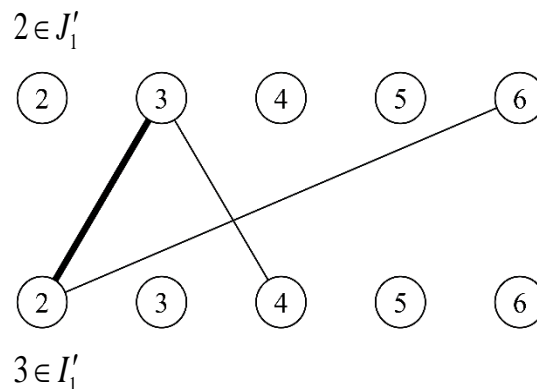


Рисунок 2.17 – Підграф D_4



Рисунок 2.18 – Допоміжний оргграф (Z_4, A_4)

S4. Оскільки $M_2^1 < M_2^2$, то до підграфа D_1 додається ребро $[4, 5], I_2 = \{2, 4\}$,

$J_2 = \{3,5\}$, $I'_2 = \{3,5\}$, $J'_2 = \{2,4\}$, видаляються ребра, інцидентні вершинам $5 \in I'_2$, $4 \in J'_2$. У результаті побудовано підграф $\langle D_2 \rangle$, породжений множиною вершин I_2 і J_2 (рис. 2.19).

$k=3$. За формулою (2) знаходиться $c_{16} = \min\{c_{ij} \mid i, j \neq 2,3,4,5\} = 18$, $M_3^1 = M_2 \cup \cup\{[1,6]\}$, $C(M_3^1) = c_{23} + c_{45} + c_{16} = 0 + 2 + 18 = 20$ (рис. 2.20). Щоб визначити M_3^2 , для вершин $i_k \in X - (I_3 \cup I'_3) = \{1,6,7,8\}$ формуються підграфи D_{i_k} , $\langle D_2 \rangle \subset D_{i_k}$, в яких за допомогою орграфів (Z_{i_k}, A_{i_k}) знаходяться найкоротші збільшуючі шляхи P_{i_k} відносно паросполучення M_2 , паросполучення $M_{i_k}^2$ та їх вартості $C(M_{i_k}^2)$.

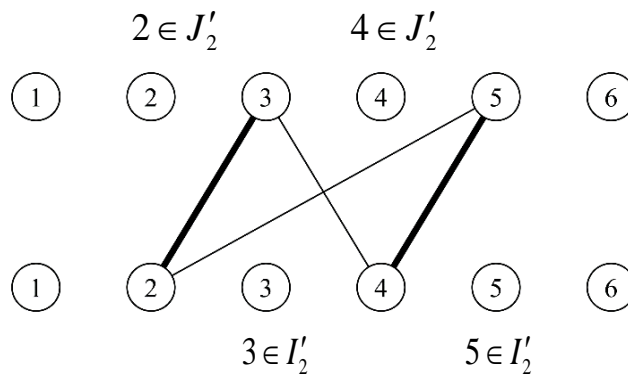


Рисунок 2.19 – Підграф $\langle D_2 \rangle$

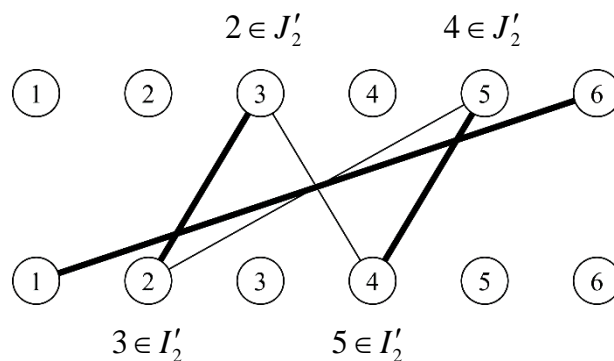


Рисунок 2.20 – Орграф (Z_1, A_1)

Підграф D_1 (рис. 2.21) містить найкоротший збільшуючий шлях відносно

M_2 , що відповідає шляху $P_1=(1, 2, 6)$ в орграфі (Z_1, A_1) (рис. 2.22) і доставляє вартість паросполучення M_1^2 , не більше від вартостей паросполучень M_6^2, M_7^2, M_8^2 .

Оскільки $P_1=\{(1, 3), [2, 3], (2, 6)\}$, $M_2=\{[2, 3], [4, 5]\}$, то $M_1^2=\{[1, 3], [4, 5], [2, 6]\}$, $C(M_1^2)=c_{13}+c_{45}+\dots+c_{26}=8+2+4=14$, $C(M_3^2)=C(M_1^3)$.

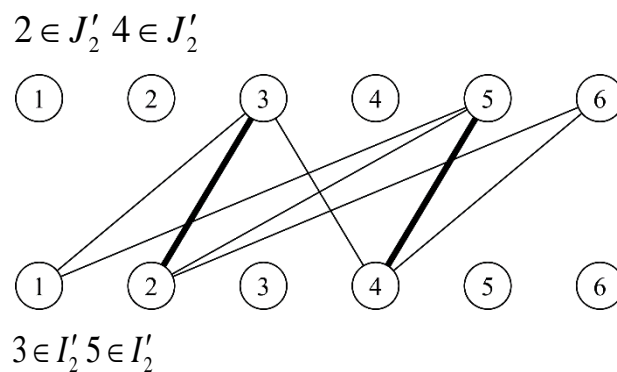


Рисунок 2.21 – Підграф D_1

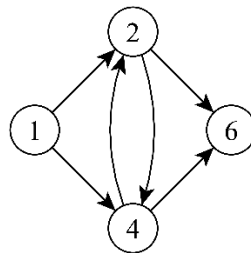


Рисунок 2.22 – Орграф (Z_1, A_1)

У цьому випадку $C(M_3^2) < C(M_3^1)$, $M_3 = M_3^2$, $I_3 = \{1, 4, 2\}$, $J_3 = \{3, 5, 6\}$, $I'_3 = \{3, 5, 6\}$, $J'_3 = \{1, 2, 4\}$. Видаляються ребра, інцидентні вершинам множини I'_3 и J'_3 . На рис. 2.23 зображено підграф $\langle D_3 \rangle$, породжений множиною вершин I_3

і J_3 .

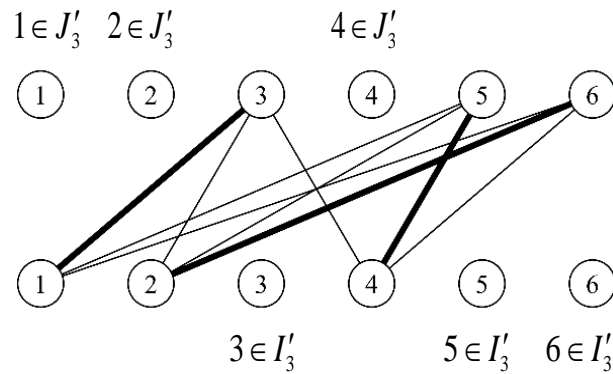


Рисунок 2.23 – Підграф $\langle D_3 \rangle$

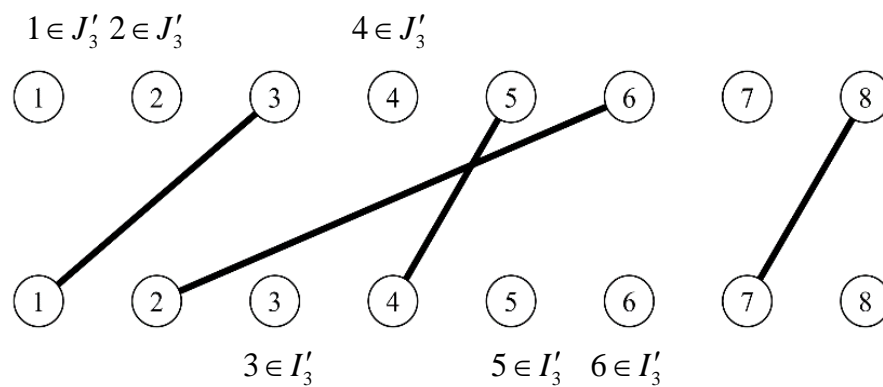


Рисунок 2.24 – Оргграф (Z_3, A_3)

$k=4$. Тут $c_{78} = \min\{c_{ij} \mid i, j \neq 1, 2, 3, 4, 5, 6\} = 37$, $M_4^1 = \{[1, 3], [2, 6], [4, 5], [7, 8]\}$, $C(M_4^1) = 3 + 4 + 2 + 37 = 51$ (рис. 2.24).

Знаходження паросполучення M_4^2 розпочинається з побудови для вершин множини $X - (I_3 \cup I'_3) = \{7, 8\}$ підграфів D_7, D_8 і оргграфів $(Z_7, A_7), (Z_8, A_8)$, що відповідають їм. На рис. 2.25 зображено підграф D_8 , а на рис. 2.26 – оргграф (Z_8, A_8) . Шлях $P_8 = (8, 2, 7)$ в оргграфі (Z_8, A_8) є найкоротшим.

У підграфі D_8 він показаний як найкоротший збільшуючий шлях

$P_8 = ((8, 6), [2, 6], (2, 7))$ відносно паросполучення M_3 .

Із $M_8^2 = P_8 \oplus M_3 = \{[1, 3], [2, 7], [4, 5], [8, 6]\}$ отримаємо, що $C(M_8^2) = c_{13} + c_{27} + c_{45} + \dots + c_{86} = 8 + 15 + 2 + 19 = 44$. Після знаходження в графі D_7 шляху P_7 , паросполучення M_7^2 і його вартості $C(M_7^2)$ виявляється, що $C(M_7^2) > C(M_8^2) = 44$. Звідси випливає, що $M_4^2 = M_8^2$, а оскільки $C(M_4^2) < C(M_4^1)$, то $M_4 = M_8^2$. Оскільки $k = \lfloor n/2 \rfloor = 4$, паросполучення $M_4 = \{[1, 3], [2, 7], [4, 5], [8, 6]\}$ є максимальним і, отже є розв'язанням ЗЗП.

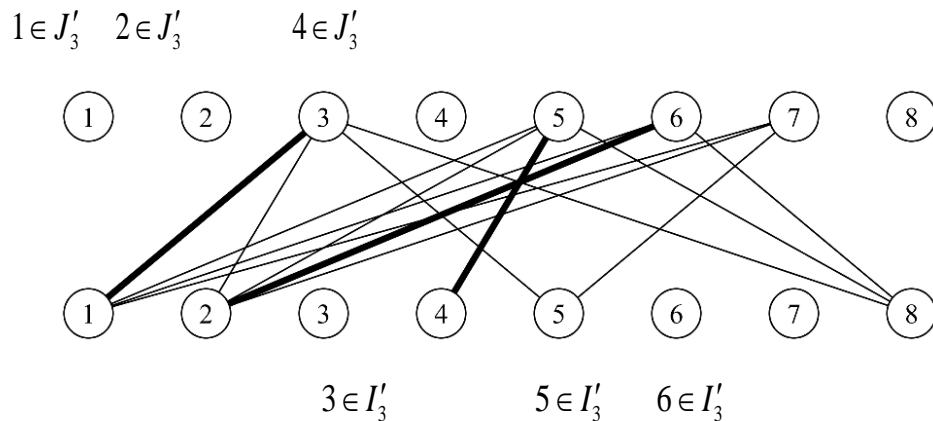


Рисунок 2.25 – Підграф D_8

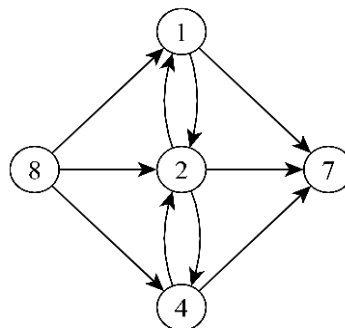


Рисунок 2.26 – Орграф (Z_8, A_8)

Шлях $P_8 = (8, 2, 7)$ в орграфі (Z_8, A_8) є найкоротшим. У підграфі D_8 він

показаний як найкоротший збільшуючий шлях $P_8 = ((8,6), [2,6], (2,7))$ відносно паросполучення M_3 . З $M_8^2 = P_8 \oplus M_3 = \{[1,3], [2,7], [4,5], [8,6]\}$ отримуємо, що $C(M_8^2) = c_{13} + c_{27} + c_{45} + c_{86} = 8 + 15 + 2 + 19 = 44$. Після знаходження в графі D_7 шляху P_7 , паросполучення M_7^2 і його вартості $C(M_7^2)$ виявляється, що $C(M_7^2) > C(M_8^2) = 44$. Звідси випливає, що $M_4^2 = M_8^2$, а оскільки $C(M_4^2) < C(M_4^1)$, то $M_4 = M_8^2$. Оскільки $k = \lfloor n/2 \rfloor = 4$, паросполучення $M_4 = \{[1,3], [2,7], [4,5], [8,6]\}$ максимально і, отже є розв'язанням ЗЗП.

Таким чином, ЗЗП коректно зводиться до однієї з оптимізаційних задач про паросполучення в дводольному графі з $2n$ вершинами для розв'язання якої застосовують модифікацію рекурентного методу, викладеного в підрозділах 2.1 і 2.2. Особливістю методу та його модифікації є загальний підхід до знаходження оптимального призначення і максимального паросполучення мінімальної ваги. Запропонована модифікація виконує перехід від довільного графа до дводольного, виключаючи технічні труднощі, пов'язані з перетворенням квіток, на якому побудовано алгоритм Едмонса. Тому часова складність модифікації оцінюється величиною $O(n^3)$, тобто такою же, як у найефективніших алгоритмів розв'язання менш складної ЗП.

Висновки до розділу 2

1. Запропоновано нову перестановочно-матричну модель оптимального призначення, яка, забезпечує можливість рекурсивного знаходження розв'язків на множині збільшуючих шляхів, побудованих відносно поточного паросполучення.

2. Розроблено рекурсивний метод розв'язання ЗП, який характеризується простотою реалізації і часом роботи, не більшим, ніж час роботи найефективнішого на цей час угорського алгоритму.

3. Уперше для розв'язання ЗЗП запропоновано модифікацію рекурентного

методу оптимального призначення, що знижує часову складність алгоритму Едмонса $O(n^4)$ до $O(n^3)$.

Основні результати першого розділу опубліковано у роботах [45, 124, 126].

Список джерел, які використано у даному розділі, наведено у повному списку використаних джерел [29, 45, 46, 48, 49, 51, 57, 67, 74, 75, 82, 106, 124].

РОЗДІЛ 3

РЕКУРЕНТНИЙ МЕТОД ЗНАХОДЖЕННЯ 2-ФАКТОРА МІНІМАЛЬНОЇ ВАГИ В РОЗВ'ЯЗАННІ ГАМІЛЬТОНОВОЇ ЗАДАЧІ КОМІВОЯЖЕРА

У розділі показано, що задача побудови в графі $H = (V, U)$ 2-фактора мінімальної ваги (задача 2- f) коректно зводиться до обмеженої версії задачі про призначення (ЗП), для розв'язання якої використовується рекурентний метод із часом роботи $O(n^3)$, $n = |V|$. Розділ містить опис удосконаленої модифікації класичного алгоритму Літгла.

3.1. Постановка задачі знаходження 2-фактора мінімальної ваги в термінах паросполучень для дводольних графів

У графі $H = (V, U)$, де V – множина вершин, помічених числами $\{1, 2, \dots, \dots, n\}$, U – множина ребер, що не містять петель $\{i, i\}$, $i = \overline{1, n}$, 2-фактор є об'єднанням циклів, що не перетинаються (по вершинах) [99].

Задача 2- f формулюється так.

Нехай задано граф $H = (V, U)$ з вагами $c_{ij} \in R_0^+$ ребер $\{i, j\} \in U$.

Потрібно знайти в графі $H = (V, U)$ 2-фактор, сума ваг ребер якого була б найменшою.

Поставлена задача для повного графа H_n завжди має розв'язок. В [119] викладено алгоритм, який визначає в n -вершинному повному графі 2-фактор мінімальної ваги за час $O(n^3)$. З [49] відомо, що для остовного підграфа H повного графа H_n задача 2- f вирішується шляхом її зведення до задачі знаходження паросполучення мінімальної ваги в графі з істотно більшим числом вершин і ребер, чим в H . Пошук паросполучення виконується за час $O(n^4)$ алгоритмом Едмондса або його модифікаціями, що містять процедуру виявлення

квітки – циклу з $2k+1$ вершинами, в якому k ребер утворюють паросполучення, і процедуру зрізування квітки – його заміну однією вершиною [75].

Оскільки зв'язний 2-фактор являє собою гамільтонов цикл графа H_n , розв'язання поставленої задачі використовують при розробленні ефективних алгоритмів із гарантованими оцінками для класу задач комівояжера, які мають широкий спектр застосувань [35, 75]. Більшість цих алгоритмів характеризуються такою ж оцінкою трудомісткості, як і в алгоритмі Едмондса.

Можливість знаходження 2-фактора мінімальної ваги за поліноміальний час є основою для вибору сформульованої задачі як релаксації, яка забезпечує зараз найбільш близькі до оптимуму нижні оцінки в точних алгоритмах розв'язання симетричної задачі комівояжера (СЗК), побудованих за методом гілок і меж [48, 84, 85].

Головна причина, що ускладнює обчислення нижніх меж алгоритмом Едмондса в точних методах розв'язання СЗК, полягає в складності процедур виявлення і зрізування квітки, неодноразове виконання яких призводить до відчутних часових витрат.

У цьому підрозділі пропонується метод знаходження в графі $H = (V, U)$ 2-фактора мінімальної ваги, що не містить дій з квітками. Його трудомісткість оцінюється величиною $O(n^3)$.

На основі цього методу виконується пошук у дводольному графі з $2n$ вершинами досконалого паросполучення з мінімальною сумарною вагою ребер, що відповідає в графі H 2-фактору мінімальної ваги, на основі використання поняття найкоротшого збільшуючого шляху і способу його побудови [49, 51, 75].

Зважений граф $H = (V, U)$ з n вершинами множини V , поміченими числами з множини $\{1, 2, \dots, n\}$, що не містить петель $\{i, i\}$, повністю визначається матрицею вартостей (ваг) ребер $C = [c_{ij}]_n$, в якій $c_{ij} \in R_0^+$, якщо $\{i, j\} \in U$ і $c_{ij} = \infty$ інакше. Заміна кожного ребра $\{i, j\}$ із H на пару дуг (i, j) і (j, i) з вагами c_{ij} і c_{ji} , $c_{ij} = c_{ji}$, дає зважений оргграф $G = (V, E)$. Підграф $G' = (V, E')$ оргграфа

$G = (V, E)$, $E' \subset E$ називається контурним покриттям, якщо кожна вершина підграфу G' має напівстепені входу і виходу, які дорівнюють 1.

Якщо в оргграфі G побудовано контурне покриття $K = \{K_1, K_2, \dots, K_\mu\}$, в якому кожна підмножина K_i , $i = \overline{1, \mu}$, містить не менше трьох вершин, то в графі H побудовано 2-фактор із тих же μ підмножин вершин. Але контурне покриття збігається з цикловим розкладанням перестановки π множини $\{1, 2, \dots, n\}$ номерів стовпців матриці вартостей C , для якої π задає допустимий розв'язок задачі про призначення (ЗП). Звідси впливає формулювання задачі 2-фактора в термінах ЗП.

Для симетричної матриці вартостей (ваг) $C = [c_{ij}]_n$, в якій $c_{ij} = \infty$ при $i = j$ і $c_{ij} \in R_0^+$ або $c_{ij} = \infty$ при $i \neq j$, $i, j = \overline{1, n}$, потрібно знайти.

$$C(\eta) = \min_{\xi} \sum_{i=1}^n c_{[\eta][i]}. \quad (3.1)$$

Тут $\eta = (\eta[1], \eta[2], \dots, \eta[n])$ – перестановка множини $\{1, 2, \dots, n\}$ номерів стовпців матриці C , визначена на множині перестановок $\xi = (\xi[1], \xi[2], \dots, \xi[n])$, у кожній з яких циклове розкладання наведено контурами, що містять не менше трьох вершин.

Слід зауважити, що задача 2-фактор з матрицею вартостей, що містить елементи $c_{ij} = \infty$, може не мати розв'язку. В цьому випадку необхідно показати, що множина перестановок ξ є порожньою.

Будемо шукати η , покроково збільшуючи на одиницю число k елементів послідовності, що утворює певну частину допустимого розв'язку задачі 2-f. Відмітимо особливості цієї послідовності.

Першим k елементам допустимого розв'язку ξ задачі 2-f поставимо у відповідність підматрицю порядку k матриці C і розв'язання ЗП для цієї

підматриці, яке задовольняє наступному обмеженню: його циклове розкладання не містить контурів із двома вершинами. Назвемо таке розв'язання ЗП обмеженим.

Нехай на множині всіх обмежених розв'язків ЗП з k елементів матриці C розв'язання $\xi_k = (\xi[i_1], \xi[i_2], \dots, \xi[i_k])$ має мінімальну вартість. Тоді (3.1) знаходиться за n ітерацій, кожна з яких перетворює послідовність ξ_k у послідовність ξ_{k+1} , $k = \overline{1, n-1}$. Викладені міркування відкривають можливість застосування для знаходження (3.1) основних результатів теорії паросполучень для дводольних графів [75].

Симетричній матриці вартостей C орграфа $G = (V, E)$, побудованого з $H = (V, U)$, взаємно однозначно відповідає дводольний граф $D = (X, Y, E)$, де X, Y – множина вершин, $|X| = |Y| = |V| = n$, $E = \{(i, j) | i \in X, j \in Y\}$ – множина ребер із вагами $c_{ij} \in R_0^+$, $i \neq j$, з матриці C , $|E| = 2|U|$.

Для формального опису задачі 2- f і методу її розв'язання скористаємося наведеними в розділі 2 визначеннями максимального і досконалого паросполучень, вільного ребра, вільної і насиченої вершин паросполучення, шляхів в графі відносно фіксованого паросполучення M , що чередуються і збільшуються, напарника j вершини i для ребра в M .

Позначимо $[i, j]$ ребро $\{i, j\}$, $i \neq j$, що належить M . Ребро $\{i, j\}$, не включене, в M позначимо (i, j) . Розв'язанням задачі 2- f в дводольному графі D є досконале паросполучення η з мінімальною сумою ваг ребер, в якому якщо $[i, j]$ належить η , то ребро з початковою вершиною $j \in X$ і кінцевою вершиною $i \in Y$ не належить η .

Нехай в дводольному графі D зафіксовано паросполучення $\xi_k'' = \{[i_3, j_2], [i_4, j_3], \dots, [i_{2k-2}, j_{2k-3}], [i_2, j_{2k-2}]\}$. Тоді його об'єднанню з вільними ребрами (i_1, j_2) і (i_{2k-2}, j_{2k}) (рис. 3.1) відповідає в графі H простий ланцюг $(v_1, v_2, v_3, \dots, v_{2k-1}, v_{2k})$, $k \geq 3$ (рис. 3.2).

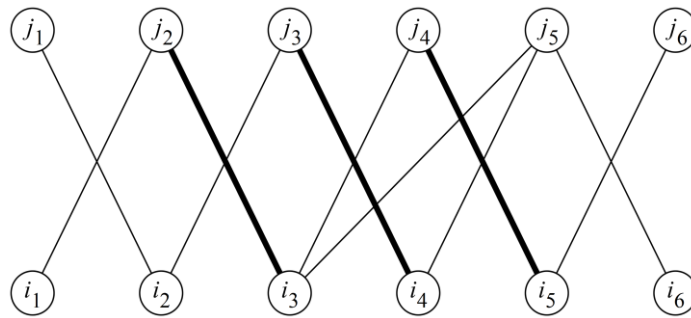


Рисунок 3.1 – Паросполучення ξ_k'' графа D , об'єднане з вільними ребрами (i_1, j_2) і (i_{2k-2}, j_{2k})

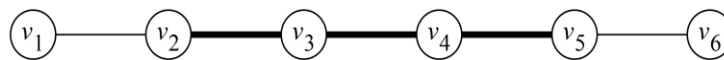


Рисунок 3.2 – Простий ланцюг $(v_1, v_2, v_3, \dots, v_{2k-1}, v_{2k})$, $k \geq 3$ в графі H

Паросполученню $\xi_k'' = \{[i_3, j_2], [i_4, j_3], \dots, [i_{2k-2}, j_{2k-3}], [i_2, j_{2k-2}]\}$ графа D доповненому вільними ребрами (i_1, j_2) і (i_{2k-2}, j_{2k-1}) (рис. 3.3), відповідає в графі H підграф, що містить простий цикл $(v_2, v_{2k-2}, v_{2k-3}, \dots, v_3, v_2)$ і два ребра $(v_1, v_2), (v_{2k-2}, v_{2k-1}), k \geq 3$ (рис. 3.4).

Паросполучення ξ_k' або ξ_k'' узгоджується з частиною деякого допустимого розв'язку задачі 2- f . Тому пошук 2- f мінімальної ваги в графі H полягає в знаходженні паросполучення в дводольному графі D . До моменту початку розв'язання задачі 2- f у графі D не зафіксоване паросполучення, що відповідає частині шуканої послідовності η .

Якщо будь-яке ребро в D визначити як збільшуючий шлях відносно паросполучення \emptyset , то початкову послідовність ξ_1 , що відповідає оптимальному обмеженому розв'язку ЗП з одного елемента, утворює ребро мінімальної ваги, взятої на множині значень c_{ij} , $i, j = \overline{1, n}$, $i < j$, матриці C .

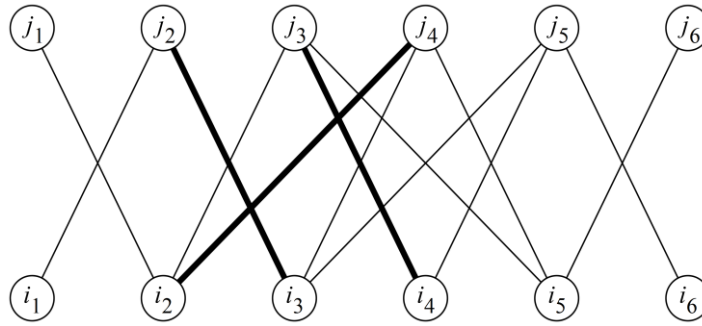


Рисунок 3.3 – Паросполучення ξ_k'' графа D , доповнене вільними ребрами

$$(i_1, j_2) \text{ і } (i_{2k-2}, j_{2k-1})$$

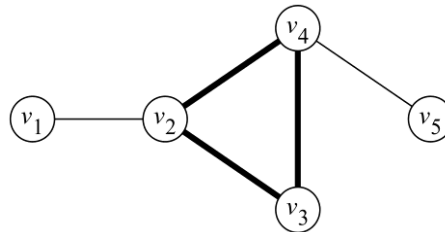


Рисунок 3.4 – Підграф графа H , що містить простий цикл $(v_2, v_{2k-2}, v_{2k-3}, \dots, v_3, v_2)$ і два ребра $(v_1, v_2), (v_{2k-2}, v_{2k-1}), k \geq 3$

3.2. Обґрунтування і опис алгоритму розв'язання задачі 2-f

Нехай на множині Π_k усіх паросполучень $\pi_k = \{[l_1, \pi[l_1]], [l_2, \pi[l_2]], \dots, [l_m, \pi[l_m]], \dots, [l_k, \pi[l_k]]\}$ графа $D = (X, Y, E)$, в яких немає ребер, що з'єднують вершини $\pi[l_m] \in X$ і $l_m \in Y$, $l_m \neq \overline{\pi[l_m]}, m = \overline{1, k}$, побудоване паросполучення $\xi_k = \{[i_1, j_1], [i_2, j_2], \dots, [i_l, j_l], \dots, [i_k, j_k]\}$ мінімальної вартості. Видаливши в графі D ребра $(j_1, i_1), (j_2, i_2), \dots, (j_l, i_l), \dots, (j_k, i_k)$, $j_l \in X$, $i_l \in Y$, і поклавши їх ваги в матриці C , отримаємо остовний підграф D_k графа D .

У графі D_k перетворимо ξ_k на паросполучення $\xi_{k+1} = \{[i'_1, j'_1], [i'_1, j'_1], [i'_2, j'_2], \dots, [i'_k, j'_k], [i'_{k+1}, j'_{k+1}]\}$, що доставляє мінімальну сумарну вагу ребер $C(\xi_{k+1})$ на множині Π_{k+1} усіх паросполучень π_{k+1} .

Паросполучення ξ_k розбиває множини X і Y відповідно на підмножини насичених вершин $I_k = \{i_1, i_2, \dots, i_l, \dots, i_k\}$, $J_k = \{j_1, j_2, \dots, j_l, \dots, j_k\}$ і на множини вільних вершин $X - I_k = \{i_{k+1}, i_{k+2}, \dots, i_p, \dots, i_n\}$, $Y - J_k = \{j_{k+1}, j_{k+2}, \dots, j_g, \dots, j_n\}$. Знайдемо вільне ребро вагою

$$c_{ms} = \min \{c_{ij} \mid i \in X - I_k, j \in Y - J_k\}, (m, s) \in D, \quad (3.2)$$

і, приєднавши його до паросполучення ξ_k , отримаємо паросполучення $\xi_{k+1}^1 = \xi_k \cup [m, s]$ вартістю $MIN1 = C(\xi_k) + c_{ms}$. Якщо ξ_{k+1}^1 не доставляє мінімальну суму ваг ребер на множині Π_{k+1} , то її доставляє $\xi_{k+1}^2 \in \Pi_{k+1} - \{\xi_{k+1}^1\}$ вартістю $MIN2 = C(\xi_{k+1}^2)$. Таким чином, $C(\xi_{k+1}) = \min \{MIN1, MIN2\}$.

Лема 3.1. Нехай $\xi_{k+1} \in \Pi_{k+1} - \{\xi_{k+1}^1\}$. Тоді $\xi_{k+1} = P_{k+1} \oplus \xi_k = (P_{k+1} - \xi_k) \cup \cup (\xi_k - P_{k+1})$, де P_{k+1} – найкоротший збільшуючий шлях відносно паросполучення ξ_k в підграфі D_k .

Доказ леми повторює доказ леми з [124], що визначає спосіб перетворення π_k в паросполучення π_{k+1} у рекурентному методі розв'язання ЗП.

Нехай в D_k вершина $j_r \in J_k$, $r \in \{1, 2, \dots, k\}$, є кінцем вільного ребра (i_l, j_r) з мінімальною вагою серед усіх ребер (i_s, j_r) , $i_s \notin I_k$, а вершина $i_f \in I_k$, $f \in \{1, 2, \dots, k\}$ – початком вільного ребра (i_f, j_p) з мінімальною вагою серед усіх ребер (i_f, j_g) , $j_g \notin J_k$.

Позначимо X_k множину вільних вершин i_l , інцидентних знайденим ребрам (i_l, j_r) , $|X_k| \leq k$. Відповідно Y_k – множина вільних вершин j_p , інцидентних ребрам (i_f, j_p) , $|Y_k| \leq k$.

Неважко побачити, що найкоротший збільшуючий шлях P_{k+1} відносно паросполучення ξ_k починається в деякій вершині множини X_k і закінчується в деякій вершині множини Y_k підграфа $\langle D_k \rangle$, породженого множиною вершин $X_k \cup I_k \cup J_k \cup Y_k$ підграфа D_k [99].

Перетворимо підграф $\langle D_k \rangle$ у зважений оргграф (Z, A) , за допомогою якого виконується пошук найкоротшого збільшуючого шляху P_{k+1} відносно паросполучення ξ_k [49, 51, 75].

Граф (Z, A) складається з множини вершин $Z = \{i_0\} \cup X_k \cup J_k \cup Y_k$ і множини дуг $A = A_0 \cup A_1 \cup A_2 \cup A_3$. Підмножина A_0 містить $|X_k|$ дуг (i_0, i_l) нульової ваги, $i_l \in X_k$. У підмножину A_1 входить дуга (i_l, i_r) , $i_l \in X_k$, $i_r \in I_k$, тоді і тільки тоді вершина j_r ребра (i_l, j_r) – напарник вершини i_r в паросполученні ξ_k .

Дузі (i_l, i_r) привласнюється вага $c(i_l, i_r) = c_{i_l j_r} + c_{i_r j_r}$. Дуга (i_d, i_l) , $i_d, i_l \in X_k$, входить в A_2 тоді і тільки тоді, коли вершина j_l ребра (i_d, j_l) , $j_l \in J_k$, є напарником вершини i_l в паросполученні ξ_k . Дуга (i_d, i_l) отримує вагу $c(i_d, i_l) = c_{i_d j_l} + c_{i_l j_l}$. Підмножина A_3 містить дуги (i_f, j_p) , $i_f \in I_k$, $j_p \in Y_k$, якщо вершини i_f і j_p сполучені в підграфі $\langle D_k \rangle$ ребром (i_f, j_p) . Дузі присвоюється вага $c(i_f, j_p) = c_{i_f j_p}$.

На рис. 3.5 показано підграф $\langle D_4 \rangle$ з паросполученням $\xi_4 = \{[i_2, j_3], [i_3, j_4], [i_4, j_5], [i_5, j_6]\}$. Підграф $\langle D_4 \rangle$ не містить ребер $(i_3, j_2), (i_4, j_3), (i_5, j_4), (i_6, j_5)$, $I_4 = \{i_2, i_3, i_4, i_5\}$, $J_4 = \{j_3, j_4, j_5, j_6\}$. Початки ребер $(i_1, j_4), (i_6, j_3), (i_7, j_6)$,

утворюють множину $X_4 = \{i_1, i_6, i_7\}$, а кінці ребер (i_2, j_1) , (i_3, j_1) , (i_4, j_7) , (i_5, j_7) – множина $Y_4 = \{i_1, j_7\}$. У підграф $\langle D_k \rangle$ входять усі вільні ребра (i_2, j_4) , (i_3, j_6) , (i_5, j_3) , початки і кінці яких є насиченими вершинами.

Допоміжний оргграф (Z, A) , побудований для підграфа $\langle D_4 \rangle$, зображений на рис. 3.6.

У ньому множину дуг A утворюють підмножини $A_0 = \{(i_0, i_1), (i_0, i_6), (i_0, i_7)\}$, $A_1 = \{(i_1, i_3), (i_6, i_2), (i_7, i_4), (i_7, i_5)\}$, $A_2 = \{(i_2, i_3), (i_3, i_5), (i_5, i_2)\}$, $A_3 = \{(i_2, j_1), (i_3, j_1), (i_4, j_7), (i_5, j_7)\}$.

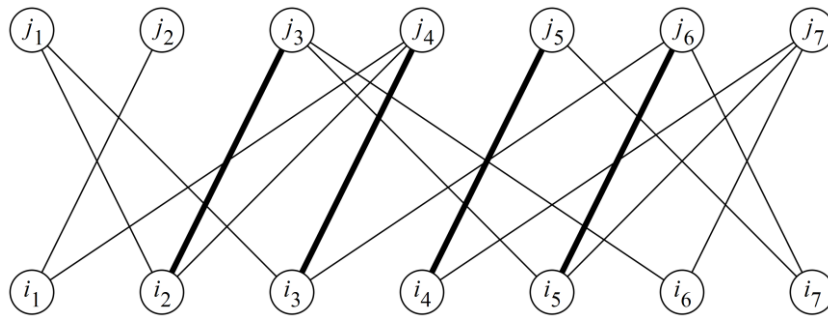


Рисунок 3.5 – Підграф $\langle D_k \rangle$

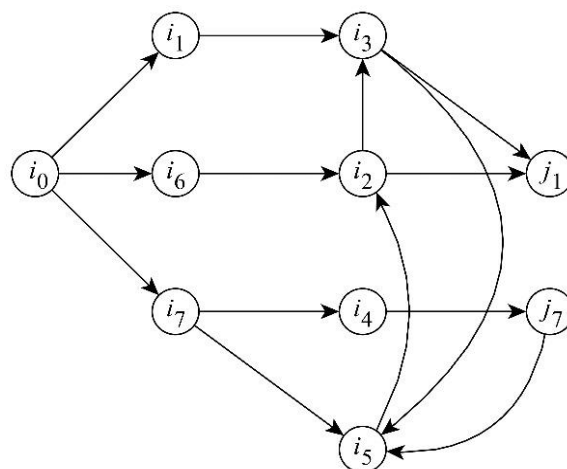


Рисунок 3.6 – Допоміжний оргграф (Z, A)

Із способу побудови орграфа (Z, A) виходить, що множина простих шляхів із вершини i_0 в усі вершини множини Y_k збігається з множиною збільшуючих шляхів відносно паросполучення ξ_k , які з'єднують у підграфі $\langle D_k \rangle$ вершини множини X_k з вершинами множини Y_k .

Крім того, кожному простому шляху орграфа (Z, A) з початком у вершині $i_l \in X_k$ і кінцем у вершині $j_p \in Y_k$ взаємно однозначно відповідає збільшуючий шлях з i_l в j_p відносно паросполучення ξ_k підграфі $\langle D_k \rangle$.

Наприклад, простому шляху (i_6, i_2, i_3, j_1) в орграфі (Z, A) , зображеному на рис. 3.6, відповідає в підграфі $\langle D_k \rangle$ (рис. 3.5) простий шлях $(i_6, j_3, i_2, j_4, i_3, j_1)$. Він містить три вільних ребра (i_6, j_3) , (i_2, j_4) , (i_3, j_1) і два ребра $[i_2, j_3]$, $[i_3, j_4]$ паросполучення ξ_4 .

Шлях $(i_6, i_2, i_3, i_5, i_2, j_1)$ (рис. 3.6) двічі проходить по вершинах $i_2 \in Y_4$. У підграфі $\langle D_4 \rangle$, зображеному на рис. 3.5, йому відповідає шлях $(i_6, j_3, i_2, j_4, i_3, j_6, i_5, j_3, i_2, j_1)$, який двічі відвідує ребро $[i_2, j_3]$ паросполучення ξ_4 . Ясно, що цей шлях не є збільшуючим відносно паросполучення ξ_4 .

Твердження 3.1. Найкоротшим шляхом з будь-якої вершини $i_l \in X_k$ у будь-яку вершину $j_p \in Y_k$ є простий шлях в орграфі (Z, A) і збільшуючий відносно паросполучення ξ_k в підграфі $\langle D_k \rangle$.

Доведення. Нехай в орграфі (Z, A) вершина $j_p \in Y_k$ є досяжною з вершини $i_l \in X_k$, наприклад, шляхом $(i_l, i_r, i_d, i_f, i_g, i_d, i_f, j_p)$, $i_r, i_d, i_f, i_g \in I_k$.

Оскільки кожна дуга шляху має невід'ємну вагу, то вершини i_l і j_p зв'язує коротший простий шлях $(i_l, i_r, i_d, i_f, j_p)$ (рис. 3.7). У підграфі $\langle D_k \rangle$ йому відповідає збільшуючий шлях $(i_l, j_r, i_r, j_d, i_d, j_f, i_f, j_p)$ відносно паросполучення

$\xi_k = \{[i_a, j_a], [i_r, j_r], [i_d, j_d], [i_f, j_f], [i_g, j_g], [i_k, j_k]\}$ (рис. 3.8). Доведення завершено.

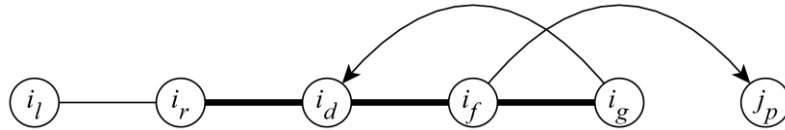


Рисунок 3.7 – Коротший шлях $(i_l, i_r, i_d, i_f, j_p)$, який зв'язує вершини i_l і j_p

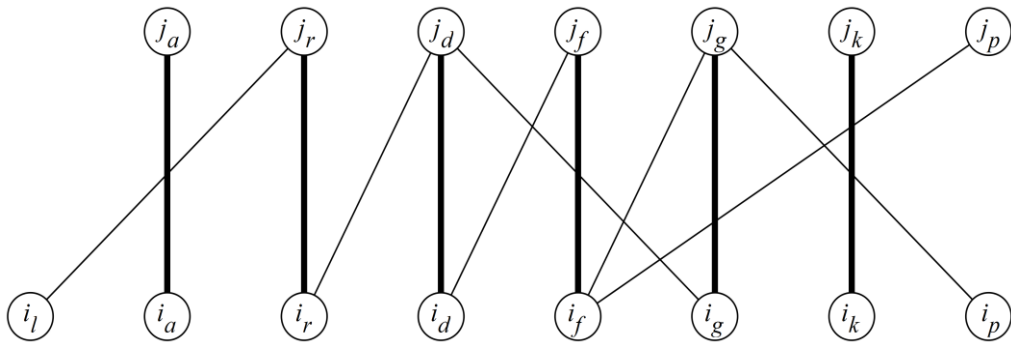


Рисунок 3.8 – Збільшуючий шлях $(i_l, j_r, i_r, j_d, i_d, j_f, i_f, j_p)$ відносно паросполучення ξ_k у підграфі $\langle D_k \rangle$

Твердження 3.2. Якщо оргграф (Z, A) не містить шляхів, що починаються в довільній вершині $i_l \in X_k$ і закінчуються в довільній вершині $j_p \in Y_k$, то ξ_k – максимальне паросполучення підграфа $\langle D_k \rangle$.

Доведення. З умови твердження 3.1 випливає, що в оргграфі (Z, A) немає простих шляхів із початковою вершиною з X_k і кінцевою вершиною з Y_k , а підграф $\langle D_k \rangle$ не містить збільшуючих шляхів відносно паросполучення ξ_k . Таким чином, для підграфа $\langle D_k \rangle$ виконується необхідна і достатня умова того, що паросполучення ξ_k максимальне в D_k [75].

Якщо паросполучення ξ_k максимальне в D_k , то згідно з лемою 3.1 у D_k не існує паросполучення ξ_{k+1}^2 , побудови 2-фактора мінімальної ваги, що розвиває процес. Припустимо, що в орграфі (Z, A) множина шляхів, що з'єднують пари вершин (i_l, j_p) , $i_l \in X_k$, $j_p \in Y_k$, непорожньо. Результатом побудови найкоротшого з них є найкоротший збільшувачий шлях P_{k+1} відносно паросполучення $\xi_k = \{[i_1, j_1], [i_2, j_2], \dots, [i_l, j_l], \dots, [i_k, j_k]\}$. Тоді $\xi_{k+1}^2 = P_{k+1} \oplus \xi_k$. Перетворення ξ_k в паросполучення ξ_{k+1}^2 містить такі дії:

1. У графі D видаляються ребра (j_l, i_l) , де $j_l \in X$, $i_l \in Y$, $l \in \{1, 2, \dots, k\}$, а в початковій матриці $C = [c_{ij}]$ відповідні їм вартості покладають такими, що дорівнюють ∞ .

2. В отриманому підграфі D_k графа D формується множини вершин X_k і Y_k , та визначається підграф $\langle D_k \rangle$, породжений множиною вершин $X_k \cup I_k \cup J_k \cup Y_k$.

3. Для підграфа $\langle D_k \rangle$ будується допоміжний зважений орграф (Z, A) , $Z = \{i_0\} \cup X_k \cup I_k \cup Y_k$, в якому алгоритмом Дейкстри шукається шлях, найкоротший серед усіх шляхів, що починаються у вершині з X_k і закінчуються у вершині з Y_k .

4. Для такого шляху, якщо він існує, в підграфі $\langle D_k \rangle$ визначається збільшувачий відносно ξ_k шлях P_{k+1} і паросполучення ξ_{k+1}^2 .

Щоб визначити ξ_{k+1} , коли ξ_k максимальне, слід знайти в підграфі D_k ребро вагою, яке визначається (3.2), і приєднати його до ξ_k . Якщо такого ребра немає, то задача 2-f не має розв'язку.

Метод розв'язання задачі 2-f:

S0. Алгоритм пошуку в графі $H = (Y, U)$ 2-фактора мінімальної ваги.

$C = [c_{ij}]_n$ – симетрична матриця вартостей ребер графа H , в якій $c_{ij} = c_{ji} \in R_0^+$, якщо $\{i, j\} \in U$, і $c_{ij} = c_{ji} = \infty$ інакше, R_0^+ – множина дійсних невід’ємних чисел. Розв’язання задачі 2-f наведено перестановкою $\eta = ([1], [2], \dots, [n])$ множини $\{1, 2, \dots, n\}$ номерів стовпців матриці C з мінімальною сумою ваг ребер $C(\eta) = \sum_{i=1}^n c_{[i]}$, $c_{[i]} \neq \infty$, $i = \overline{1, n}$, серед усіх таких перестановок степені n , в циклових розкладаннях яких кожен контур містить не менше трьох вершин. Шуканий 2-фактор визначається як досконале паросполучення ξ_n дводольного графа $D = (X, Y, E)$, де X, Y – множини вершин, $|X| = |Y| = n$, $E = \{(i, j) | i \in X, j \in Y\}$ – непорожня множина ребер (i, j) з вагами $c_{ij} \in R_0^+$ із матриці C , $|E| = 2|U|$, $c_{ij} = \infty$, якщо $(i, j) \notin E$. Покласти $k = 1$, знайти ребро (i_k, j_k) вагою $c_{i_k j_k} = \min\{c_{ij} | i, j = \overline{1, n}\}$, $I_k = \{i_k\}$, $J_k = \{j_k\}$, $\xi_k = \{(i_k, j_k)\}$, $C(\xi_k) = c_{i_k j_k}$, D_k – остовний підграф графа D , отриманий з D видаленням ребра (j_k, i_k) , $j_k \in X$, $i_k \in Y$, в матриці C покласти $c_{j_k i_k} = \infty$.

S1. $k = k + 1$, якщо $k > n$, то кінець: побудовано розв’язання задачі 2-f.

S2. У підграфі D_{k-1} знайти $c_{i_l j_l} = \min\{c_{ij} | i \in X - I_{k-1}, j \in Y - J_{k-1}\}$; якщо $c_{i_l j_l} = \infty$, то покласти $MIN1 = \infty$, інакше $\xi_k^1 = \xi_{k-1} \cup \{(i_l, j_l)\}$, $MIN1 = C(\xi_k^1) = C(\xi_{k-1}) + c_{i_l j_l}$, $c_{j_l i_l} = \infty$, $D_k = D_{k-1} - (j_l, i_l)$, $I_k = I_{k-1} \cup \{i_l\}$, $J_k = J_{k-1} \cup \{j_l\}$.

S3. Для кожної вершини $j_r \in J_{k-1}$ в D_{k-1} знайти ребра (i_m, j_r) з вагами $c_{i_m j_r} = \min\{c_{ij} | i \in X - I_{k-1}\} \in R_0^+$ і сформуванати з вершин i_m множину X_k ; якщо $X_k = \emptyset$, то покласти $MIN2 = \infty$ і перейти до кроку S6.

S4. Для кожної вершини $i_f \in I_{k-1}$ в D_{k-1} знайти ребра (i_m, j_p) з вагами $c_{i_f j_p} = \min\{c_{ij} | j \in Y - J_{k-1}\} \in R_0^+$ і сформуванати з вершин j_p множину Y_k ; якщо

$Y_k = \emptyset$, то покласти $MIN 2 = \infty$ і перейти до кроку S6.

S5. Визначити підграф $\langle D_k \rangle$, породжений множиною вершин $X_k \cup I_{k-1} \cup J_{k-1} \cup Y_k$ підграфа D_{k-1} ; для $\langle D_k \rangle$ побудувати допоміжний зважений оргграф (Z, A) , $Z = \{i_0\} \cup X_k \cup I_{k-1} \cup Y_k$, і виконати в ньому пошук шляху, найкоротшого на множині всіх шляхів у вершини Y_k , досяжні з i_0 ; якщо побудовано такий шлях, то в підграфі $\langle D_k \rangle$ знайти збільшуючий шлях P_k , що відповідає йому, відносно паросполучення ξ_{k-1} , а також $\xi_k^2 = (P_k - \xi_{k-1}) \cup (\xi_{k-1} - P_k)$, $MIN 2 = C(\xi_k^2)$, інакше покласти $\xi_k^2 = \emptyset$, $MIN 2 = \infty$.

S6. Якщо $MIN 1 = MIN 2 = \infty$, то кінець: для графа $H = (Y, U)$ з матрицею вагів ребер $C = [c_{ij}]_n$ задача 2-f не має розв'язання; якщо $MIN 1 \neq \infty$ або $MIN 2 \neq \infty$, то якщо $MIN 1 \leq MIN 2$, то $\xi_k = \xi_k^1$, $c_{j_i i_l} = \infty$, і перейти до кроку S1, інакше $\xi_k = \xi_k^2$, $\xi_k = \{[i_l, j_l] | l = \overline{1, k}\}$, $I_k = \{i_l | l = \overline{1, k}\}$, $J_k = \{j_l | l = \overline{1, k}\}$, утворити підграф D_k , видаливши в $\langle D_k \rangle$ ребра (j_l, i_l) , $j_l \in I_k$, $i_l \in J_k$, з вагами $c_{j_i i_l} \in R_0^+$; в матриці C покласти $c_{j_i i_l} = \infty$, $l \in \{1, 2, \dots, k\}$, і перейти до кроку S1.

Теорема 3.1. Метод коректно виконує за час $O(|V|^3)$ пошук розв'язання задачі 2-f у графі $H = (Y, U)$ з невід'ємними вагами ребер.

Доказ. Процес обчислень зупиняється, коли в дводольному графі $D = (X, Y, V)$, $|X| = |Y| = |V| = n$, побудовано максимальне паросполучення ξ_{k-1} , $k = \overline{2, n}$. Паросполучення ξ_{k-1} максимальне, якщо: а) підграф D_k графа D не містить жодного ребра, яке можна було б приєднати до ξ_{k-1} , щоб отримати ξ_k^1 ; б) підграф D_k графа D не містить збільшуючого шляху відносно паросполучення ξ_{k-1} для побудови ξ_k^2 . Вимога б) виконується, оскільки згідно з твердженням 3.2 для максимального паросполучення ξ_{k-1} досить, щоб допоміжний оргграф (Z, A) не

містив шляхів з вершин множини X_k у вершини множини Y_k .

Трудомісткість методу оцінюється, виходячи з того, що вона максимальна, тобто тоді, коли початковий граф є повним. Процес побудови оптимального розв'язання $\eta = \xi_n$ містить n етапів. Перший етап завершується на кроці S_0 побудовою ξ_1 за $n(n-1)/2$ операцій. На інших етапах знаходяться парсполучення ξ_k^1 і ξ_k^2 , $k = \overline{2, n}$. Побудова ξ_k^1 потребує $(n-k+1)(n-k)/2$ операцій. Для знаходження ξ_k^1 треба виконати $2(n-k-1)$ операцій на пошук вершин множин X_k і Y_k , $k = \overline{1, n-1}$, не більше ніж $O(n)$ операцій на побудову підграфа $\langle D_k \rangle$ і допоміжного орграфу (Z, A) , $O(n^2)$ дій на побудову в (Z, A) алгоритмом Дейкстри найкоротшого шляху P_k і не більше ніж $O(n)$ операцій з множиною ребер P_k і ребер поточного парсполучення ξ_{k-1} . Таким чином, кожен етап алгоритму характеризується квадратичною часовою складністю, а трудомісткість розв'язання задачі 2- f оцінюється величиною $O(n^3)$.

Роботу методу розв'язання задачі 2- f проілюструємо на такому прикладі.

Приклад

Граф $H = (Y, U)$, в якому потрібно визначити 2-фактор мінімальної ваги, показано на рис. 3.9, а його матриця вартостей C – на рис. 3.10.

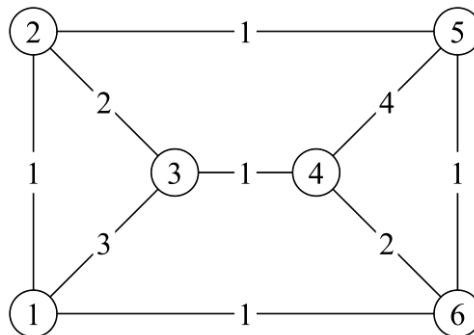


Рисунок 3.9 – Граф $H = (Y, U)$, в якому потрібно визначити 2-фактор мінімальної ваги

$j \backslash i$	1	2	3	4	5	6
1	∞	1	3	∞	∞	1
2	1	∞	2	∞	1	∞
3	3	2	∞	1	∞	∞
4	∞	∞	1	∞	4	2
5	∞	1	∞	4	∞	1
6	1	∞	∞	2	1	∞

Рисунок 3.10 – Матриця вартостей C графа $H = (Y, U)$

S0. $k=1$, $c_{12} = \min\{c_{ij} | i, j = \overline{1,6}\} = 1$, $I_1 = \{1\}$, $J_1 = \{2\}$, $\xi_1 = \{[1,2]\}$, $C(\xi_1) = c_{12} = 1$, D_1 – підграф, отриманий з графа $D = (X, Y, E)$, того, що відповідає матриці C , видаленням ребра $(2, 1)$, $c_{21} = \infty$.

S1. $k = 2$.

S2. В D_1 $c_{25} = \min\{c_{ij} | i \neq 1, j \neq 2\} = 1$, $\xi_2^1 = \xi_1 \cup [2,5] = \{[1,2], [2,5]\}$, $MIN1 = C(\xi_2^1) = c_{12} + c_{25} = 1 + 1 = 2$, $c_{52} = \infty$, $D_2 = D_1 - (5,2)$, $I_2 = \{1,2\}$, $J_2 = \{2,5\}$.

S3. Для вершини $2 \in J_1$ ребро $(5, 2)$ має найменшу вагу: $c_{25} = \min\{c_{i2} | i \neq 1\} = 1$, $X_2 = \{5\}$.

S4. Для вершини $1 \in I_1$ ребро $(1,6)$ має найменшу вагу: $c_{16} = \min\{c_{1j} | j \neq 2\} = 1$, $Y_2 = \{6\}$.

S5. Підграф $\langle D_2 \rangle$ і побудований для нього оргграф (Z, A) зображені на рис. 3.11 і 3.12. Пунктирною лінією позначено ребро, видалене з графа (X, Y, E) , потовщеною – ребро паросполучення ξ_1 . Ці ребра утворюють найкоротший збільшуючий шлях P_2 відносно паросполучення ξ_1 ; $\xi_2^2 = (\xi_1 - P_2) \cup$

$$\cup(P_2 - \xi_1) = \{[5,2],[1,6]\}, \quad MIN2 = C(\xi_2^2) = c_{52} + c_{16} = 1 + 1 = 2.$$

С6. Оскільки $MIN1 = MIN2$, то $\xi_2 = \xi_2^1$ або ξ_2^2 . Нехай $\xi_2 = \xi_2^1 = \{[1,2],[2,5]\}$, $I_2 = \{1, 2\}$, $J_2 = \{2, 5\}$. У підграфі D_1 віддаляється ребро (5, 2) ($c_{52} = \infty$), утворюючи підграф D_2 .

$$\text{При } k = 3 \quad \xi_3^1 = \{[1,2],[2,5],[3,4]\}, \quad MIN1 = C(\xi_3^1) = c_{12} + c_{25} + c_{34} = 1 + 1 + 1 = 3.$$

Підграф $\langle D_3 \rangle$ для знаходження ξ_3^2 зображений на рис. 3.13, а оргграф (Z, A) , що відповідає йому, – на рис. 3.14. В (Z, A) ваги дуг $(3, 1)$, $(6, 2)$, $(1, 6 \in Y_3)$ ($2, 3 \in Y_3$) визначають так: $c(3,1) = c_{32} + c_{12} = 2 + 1 = 3$, $c(6,2) = c_{65} + c_{25} = 1 + 1 = 2$, $c(1,6) = 1$, $c(2,3) = 2$. В оргграфі (Z, A) містяться два найкоротші шляхи з вершин множини $X_3 = \{3, 6\}$ у вершини множини $Y_3 = \{3, 6\}$: $(3, 1, 6 \in Y_3)$, $(6, 2, 3 \in Y_3)$. Виберемо будь-який з них, наприклад, $(3, 1, 6 \in Y_3)$. У підграфі $\langle D_3 \rangle$ йому відповідає найкоротший збільшувачий шлях P_3 , відносно паросполучення $\xi_2 = \{[1,2],[2,5]\}$, що містить ребра $(3,2)$, $[1,2]$, $(1,6)$. Отже, $\xi_3^2 = (P_3 - \xi_2) \cup (\xi_2 - P_3) = \{[1,6],[3,2]\} \cup \{[2,5]\} = \{[1,6], [2,5], [3,1]\}$, $MIN2 = C(\xi_3^2) = 1 + 1 + 2 = 4$. Оскільки $MIN1 \leq MIN2$, $\xi_3 = \xi_3^1 = \{[1,2],[2,5],[3,4]\}$, $I_3 = \{1, 2, 3\}$, $J_3 = \{2, 4, 5\}$, у підграфі D_2 видаляється ребро $(4, 3)$. В результаті отримаємо підграф ($c_{21} = c_{52} = c_{43} = \infty$).

$k = 4$. Паросполучення ξ_4^1 можна отримати приєднанням до ξ_3 ребра $(6, 1)$ або $(5, 6)$. Їх ваги, які дорівнюють 1, мінімальні на кроці S2. Нехай $\xi_4^1 = \{[1,2],[2,5],[3,4],[6,1]\}$. Отже, $MIN1 = C(\xi_4^1) = 1 + 1 + 1 + 1 = 4$. Для побудови ξ_4^2 формуються множини $X_4 = \{6\}$, $Y_4 = \{1, 3, 6\}$ і підграф $\langle D_4 \rangle$ (рис. 3.15). Допоміжний оргграф (Z, A) , побудований для $\langle D_4 \rangle$ показано на рис. 3.16. У ньому найкоротший шлях $(6, 2, 3 \in Y_4)$ містить ребра $(6, 5)$, $[2,5]$, $(2, 3)$ найкоротшого

збільшуючого шляху P_4 відносно паросполучення ξ_3 в $\langle D_4 \rangle$. Таким чином, $\xi_4^2 = \{[1,2],[2,3],[3,4],[6,5]\}$, $MIN2 = C(\xi_4^2) = 1 + 2 + 1 + 1 = 5$. Оскільки $MIN1 < MIN2$, то $\xi_4 = \xi_4^1$, $I_4 = \{1,2,3,6\}$, $J_4 = \{1,2,4,5\}$, $D_4 = D_3 - (1,6)$, $c_{16} = \infty$ (рис. 3.17).

$k = 5$. $\xi_5^1 = \{[1,2],[2,5],[3,4],[5,6],[6,1]\}$, $C(\xi_5^1) = 5$, $X_5 = \{4,5\}$, $Y_5 = \{3\}$. Найкоротший шлях в орграфі (рис. 3.19) містить ребра $(4, 5)$, $(2, 5)$, $(2, 3)$, що утворюють найкоротший збільшуючий шлях P_5 відносно паросполучення ξ_4 в підграфі $\langle D_5 \rangle$, який зображений на рис. 3.18; $\xi_5^2 = \{[1,2],[2,3],[3,4],[4,5],[6,1]\}$, $MIN2 = C(\xi_5^2) = 1 + 2 + 1 + 4 + 1 = 9$; $MIN1 < MIN2$, $\xi_5 = \xi_5^1$, $I_5 = \{1,2,3,5,6\}$, $J_5 = \{1,2,4,5,6\}$, $D_5 = D_4 - (1,6)$, $c_{16} = \infty$ (рис. 3.20).

$k = 6$. Тут $MIN1 = \infty$, оскільки $X - I_5 = \{4\}$, $Y - J_5 = \{3\}$, а $c_{43} = \infty$; $X_6 = \{4\}$, $Y_6 = \{3\}$. Тому вважаємо $\xi_5 = \xi_5^2 = \{[1,2],[2,3],[3,4],[4,5],[6,1]\}$, $I_5 = \{1,2,3,4,6\}$, $J_5 = \{2, 3, 4, 5, 1\}$, $c_{21} = c_{32} = c_{43} = c_{54} = c_{61} = \infty$. На кроці S3 побудова ξ_6 єдиним ребром, яке можна приєднати до паросполучення ξ_5 , є ребро $(5, 6)$, $c_{56} = 1$. Отже, $\xi_6 = \xi_6^1 = \{[1,2],[2,3],[3,4],[4,5], [6,1],[5,6]\}$, $MIN1 = 1 + 2 + 1 + 4 + 1 + 1 = 9$. Розв'язок задачі 2-f подано у вигляді гамільтонового циклу $(1, 2, 3, 4, 5, 6, 1)$. Пошук розв'язку задачі 2-f у довільному зваженому графі $H = (V, U)$ з матрицею вартостей C коректно виконується в дводольному графі $D = (X, Y, E)$, $|X| = |Y| = |V|$, $|E| = 2|U|$, що задається тією ж матрицею C . Задача 2-f формулюється як обмежена версія задачі про призначення. Її розв'язання подано перестановкою стовпців матриці C з цикловим розкладанням, в якому кожен контур містить не менше трьох дуг.

Оптимальний розв'язок є досконалим паросполученням, отриманим у результаті покрокового вибору ребер графа D , починаючи з ребра мінімальної ваги. На кожному кроці будується допустиме паросполучення ξ_k , $k = \overline{1, n-1}$, ξ_k ,

$k = \overline{1, n-1}$, з найменшою сумою ваг k ребер. Чергове паросполучення ξ_{k+1} знаходиться або побудовою найкоротшого збільшуючого шляху відносно ξ_k , або додаванням до ξ_k ребра, вагою не більшим ваги будь-якого ребра, яке утворює ξ_{k+1} .

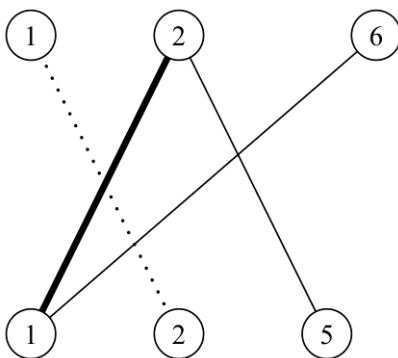


Рисунок 3.11 – Підграф $\langle D_2 \rangle$

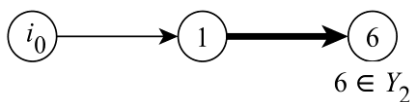


Рисунок 3.12 – Відповідний підграфу $\langle D_2 \rangle$ оргграф (Z, A)

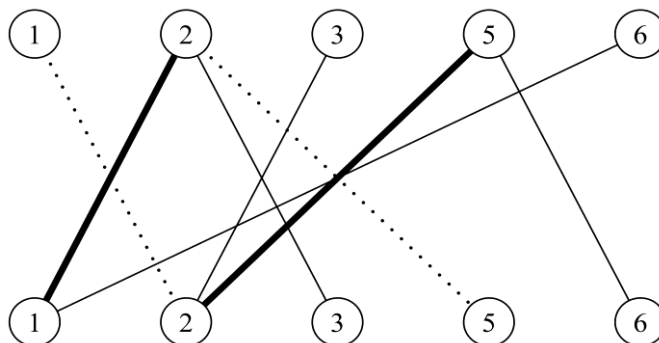


Рисунок 3.13 – Підграф $\langle D_3 \rangle$

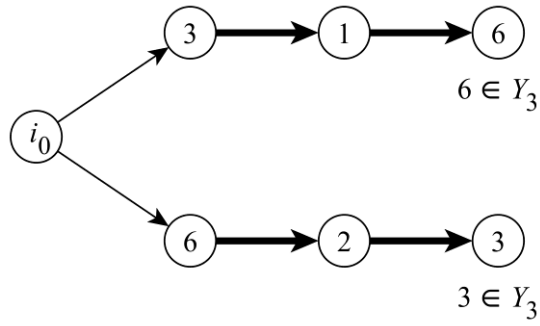


Рисунок 3.14 – Відповідний підграфу $\langle D_3 \rangle$ оргграф (Z, A)

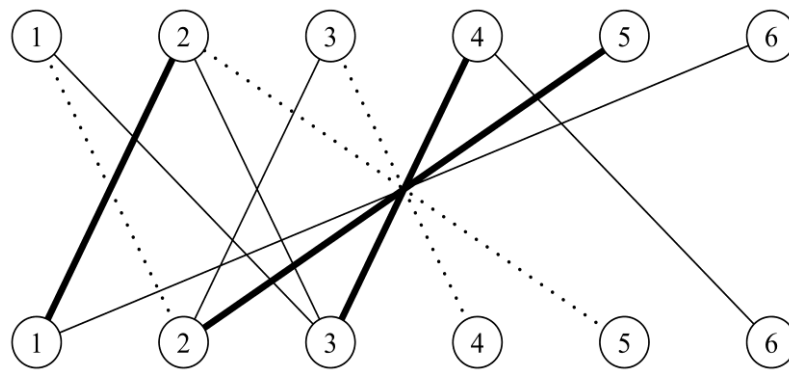


Рисунок 3.15 – Підграф $\langle D_4 \rangle$

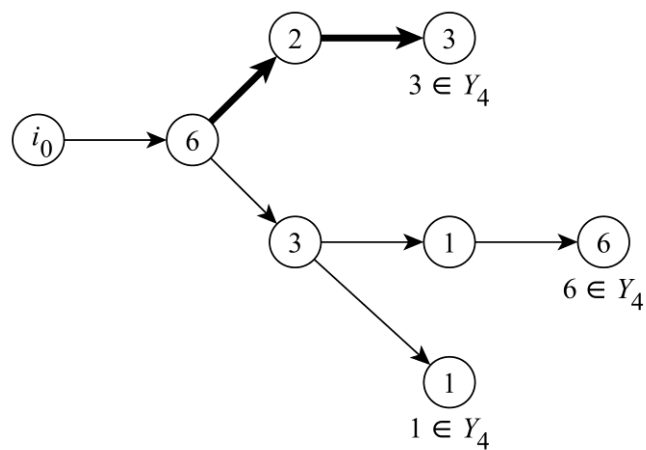


Рисунок 3.16 – Відповідний підграфу $\langle D_4 \rangle$ оргграф (Z, A)

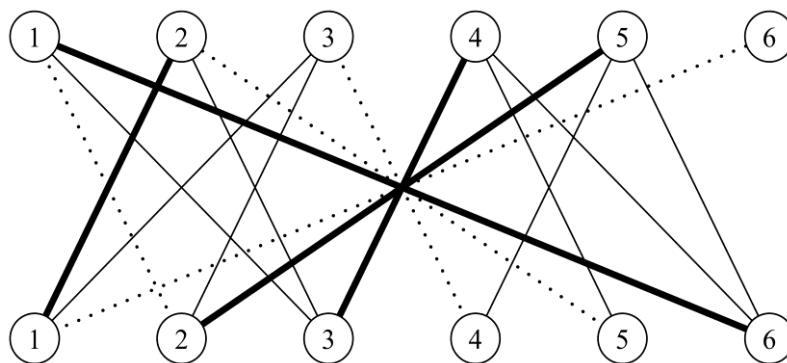


Рисунок 3.17 – Підграф $D_4 = D_3 - (1,6)$

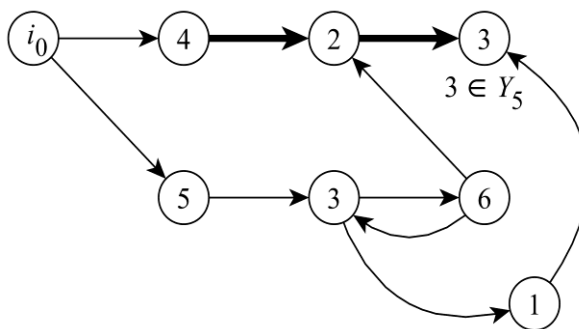


Рисунок 3.18 – Найкоротший шлях в орграфі, який містить ребра $(4, 5)$, $[2, 5]$, $(2, 3)$

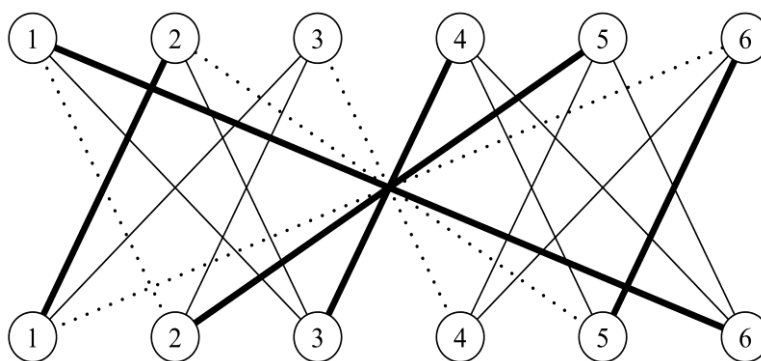


Рисунок 3.19 – Найкоротший збільшуючий шлях P_5 відносно паросполучення ξ_4 в підграфі $\langle D_5 \rangle$

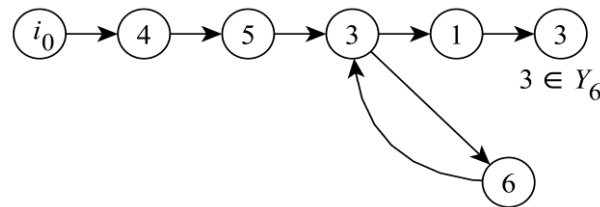


Рисунок 3.20 – 2-фактор мінімальної ваги графа $H = (Y, U)$

3.3. Особливості і опис алгоритму розв’язання гамільтонової задачі комівояжера на основі модифікованого методу Літгла

Нова вдосконалена версія класичного алгоритму Літгла для розв’язання гамільтонової задачі комівояжера (ГЗК) відрізняється від попередньої її версії, детально розглянутої в [46, 48, 73], застосуванням точнішої межі знизу шуканого значення цільового функціонала.

У ГЗК задано граф $H = (V, U)$, V – множина вершин, $V = \{v_i | i = \overline{1, n}\}$, U – множина ребер $\{v_i, v_j\}$, $v_i, v_j \in V$. В H відсутні петлі, тобто ребра $\{v_i, v_i\}$ і кратні ребра. Кожному ребру $\{v_i, v_j\}$ приписано вагу $c_{ij} \in R_0^+$, R_0^+ – множина невід’ємних дійсних чисел. Маршрутом комівояжера, або обходом називається простий остовний цикл графа H . Граф H , що містить обхід τ , визначається як гамільтоновий.

У ГЗК потрібно знайти обхід τ^* з мінімальною сумою $C(\tau^*)$ ваг ребер або встановити, що граф H не є гамільтоновим.

Нижня межа величини $C(\tau^*)$, яка застосовується в даній модифікації, визначається в результаті розв’язання задачі 2- f – задачі побудови в графі H 2-фактора з мінімальною сумою ваг ребер (2-фактора мінімальної ваги).

Будь-який 2-фактор задано множиною незв’язних простих циклів, що покривають усі вершини графа H , причому кожен цикл містить не менше трьох

вершин. Очевидно, зв'язний 2-фактор мінімальної ваги є шуканим обходом τ^* ГЗК.

Розв'язання задачі 2- f для графа $H = (V, U)$ подамо множиною циклів $\eta = \{\eta_m \mid m = \overline{1, \mu}\}$. Вагу (вартість) η позначимо $C(\eta) = \sum_{m=1}^{\mu} C(\eta_m)$, $C(\eta_m)$ – сума ваг ребер циклу η_m , $|\eta_m| \geq 3$, $|V| = \sum_{m=1}^{\mu} |\eta_m|$. Назвемо η цикловим покриттям (ЦП) вершин графа H .

У попередній модифікації алгоритму Літтла оцінкою знизу величини $C(\tau^*)$ у корені дерева перебору служить розв'язок задачі про призначення (ЗП), а в інших вершинах дерева – розв'язок її окремого випадку [46, 48].

Нехай $(\sigma, C(\sigma))$ – розв'язок ЗП, $(\eta, C(\eta))$ – розв'язок задачі 2- f для цієї матриці вартостей $C = [c_{ij}]_n$ графа $H = (V, U)$; $c_{ij} \in R_0^+$, якщо $i \in \{i, j\} \in U$, і $c_{ij} = \infty$ – інакше.

Із того, що розв'язок задачі 2- f є розв'язком ЗП, в цикловому розкладанні якого всі контури містять не менше трьох вершин, випливають нерівності $C(\sigma) \leq C(\eta) \leq C(\tau^*)$.

Ясно, що в корені дерева перебору, яке будується за схемою алгоритму Літтла, в якості нижньої межі вартості розв'язку ГЗК $C(\tau^*)$ слід вибрати вартість $C(\eta)$, у будь-якому випадку не меншу $C(\sigma)$.

Час пошуку τ^* істотно залежить від вибору в ЦП η першого ребра $\{x, y\}$ для включення його в шуканий обхід.

Нехай $\{v_k, v_l\} \in \eta_f$, $f \in \{1, 2, \dots, \mu\}$, таке ребро ЦП η , яке або утворює простий цикл $(v_k, v_r, v_d, v_l, v_k)$, $\{v_r, v_d\} \in \eta_g$, $f \neq g$, (рис. 3.21, а), або вершина v_k інцидентна вершині з η_a , а вершина v_l – вершині з η_b ; $f \neq a, b$; $a \neq b$. (рис. 3.21, б).

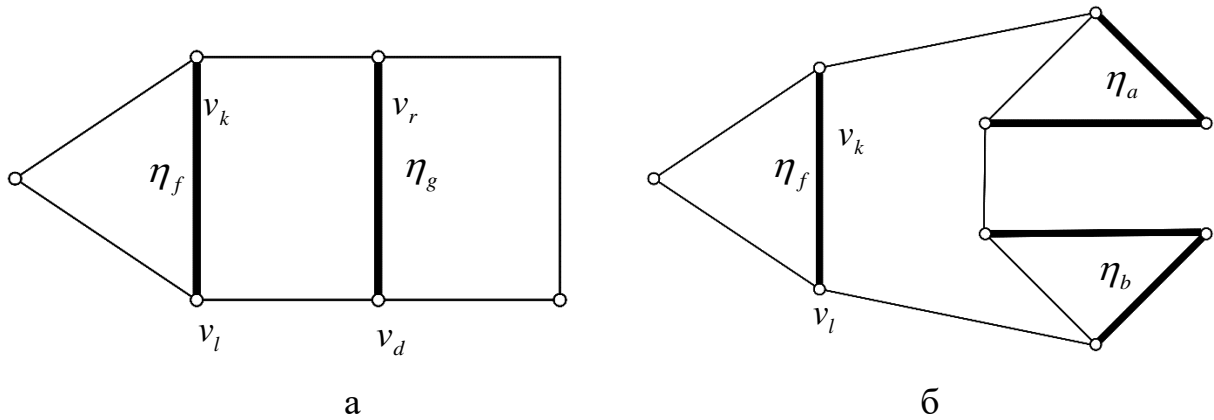


Рисунок 3.21 – Вибір в ЦП η першого ребра: а – ребро $\{v_k, v_l\} \in \eta_f$, $f \in \{1, 2, \dots, \mu\}$ ЦП η утворює простий цикл $(v_k, v_r, v_d, v_l, v_k)$, $\{v_r, v_d\} \in \eta_g$, $f \neq g$; б – вершина v_k інцидентна вершині з η_a , а вершина v_l – вершині з η_b ;
 $f \neq a, b$; $a \neq b$

Виберемо ребро $\{x, y\}$ вагою

$$c_{xy} = \max \{c_{kl} \mid \{v_k, v_l\} \in \eta\}. \quad (3.3)$$

У корені дерева ребро $\{x, y\}$ ініціює дві вершини X_{xy}^- і X_{xy} , що означають відповідно множину усіх обходів, які не містять ребро $\{x, y\}$ і множину усіх обходів, що включають його. Оцінку $C(\eta_{xy}^-)$ у вершині $X(\eta_{xy}^-)$, що обмежує знизу вартість усіх обходів, які не містять ребро $\{x, y\}$, знаходять побудовою ЦП η_{xy}^- в підграфі $H - \{x, y\}$.

Якщо не існує ЦП η_{xy}^- , то, очевидно, що не існує в графі H обходів, які б не містили ребро $\{x, y\}$. У разі нерозв'язності задачі 2- f для графа $H - \{x, y\}$ отримаємо $C(\eta_{xy}^-) = \infty$.

Визначимо цикл η_m , якому належить ребро $\{x, y\}$ $m \in \{1, 2, \dots, \mu\}$. Для обчислення у вершині $X(\eta_{xy})$ дерева перебору нижньої межі $C(\eta_{xy})$ будь-якого обходу, що містить ребро $\{x, y\}$, знайдемо в η_m ребро $\{p, q\}$ з другою за величиною після c_{xy} вагою

$$c_{pq} = \max \{c_{kl} \mid \{v_k, v_l\} \in \eta_m - \{x, y\}\}. \quad (3.4)$$

Щоб вичислити $C(\eta_{xy})$, треба спочатку з (3.3) визначити ребро $\{x, y\}$ з (3.4) – ребро $\{p, q\}$ а потім знайти в підграфі $H - \{p, q\}$ ЦП η_{pq}^- . Якщо підграф $H - \{p, q\}$ містить η_{pq}^- , то $C(\eta_{xy}) = C(\eta_{pq}^-)$. Інакше $C(\eta_{xy}) = \infty$.

Важливо відмітити, що кожне з побудованих ЦП η_{xy}^- або η_{xy} може містити менше або таке, що дорівнює μ , число компонент.

Таким чином, множина всіх обходів ГЗК наведена об'єднанням підмножин $X(\eta_{xy}^-) \cup X(\eta_{xy})$, що не перетинаються, яка є основою для пошуку τ^* методом гілок і меж.

Якщо обидва ЦП η_{xy}^- і η_{xy} побудовані відповідно у вершинах $X(\eta_{xy}^-)$ і $X(\eta_{xy})$, виявляються обходами, то $\tau^* = \eta_{xy}^-$ при $C(\eta_{xy}^-) \leq C(\eta_{xy})$ і $\tau^* = \eta_{xy}$ при $C(\eta_{xy}^-) \geq C(\eta_{xy})$. Очевидно, що ГЗК не має розв'язку, коли $C(\eta_{xy}^-) = C(\eta_{xy}) = \infty$. Якщо побудовано тільки одне ЦП η_{xy}^- або η_{xy} і воно виявляється обходом, то такий обхід є розв'язком τ^* ГЗК.

Кінцева вершина дерева перебору, якій відповідає ЦП, що містить декілька компонент зв'язності, відноситься до множини активних вершин.

Розгалуження виконується для активної вершини з найменшою вартістю ЦП, що відповідає їй. У ЦП при вершині галуження вибирається чергове ребро,

що ініціює розбиття на дві підмножини обходів. Обходи однієї з підмножин не містять вибране ребро, а обходи іншого його мають.

Далі для кожної підмножини розбиття вирішується задача 2-f, що встановлює нову оцінку знизу величини $C(\tau^*)$.

Вершина стає неактивною, коли їй відповідає зв'язний 2-фактор або оцінка, яка дорівнює ∞ . Побудова τ^* завершується знаходженням кінцевої вершини і обходу, що відповідає їй, вартістю, не більшою, ніж оцінка у будь-якій іншій кінцевій вершині.

Дана модифікація методу Літтла умовно містить два етапи: попередній і основний.

Попередній етап складається з одного кроку S0, на якому виконуються такі дії.

S0. Виконати алгоритм розв'язання задачі 2-f для початкового графа $H = (V, U)$ з матрицею вартостей $C = [c_{ij}]_n$;

– якщо для матриці C з елементами $c_{ij} = \infty$, $i \neq j$, задача 2-f не має розв'язання η , то ГЗК так само не має розв'язання;

– навести кожному компоненту η_j ЦП η , списком ребер, $j = \overline{1, \mu}$;

– якщо $\mu = 1$, то кінець: η – обхід τ^* мінімальної вартості в графі H ;

– X – множина кінцевих вершин дерева перебору, $X = \emptyset$; $\eta = \{\eta_j \mid j = \overline{1, \mu}\}$,

$C(\eta)$ – ЦП і його вартість у вершині галуження $X(\eta)$, поданій коренем дерева,

$X = X \cup \{X(\eta)\}$, $Q(\eta)$ – список ребер, виключених в процесі галуження;

$Q(\eta) = \emptyset$;

– $h = 1$; $\eta_h = \eta$, $C(\eta_h) = C(\eta)$, $Q(\eta_h) = Q(\eta)$, $X = X \cup \{X(\eta_h)\}$ (виконання основного етапу розпочинається з галуження кореня).

S1. $X = X - \{X(\eta_h)\}$; знайти в η_h ребро $\{x_h, y_h\}$ максимальної ваги,

галуження, що ініціює, у вершині $X(\eta_h)$;

(формування множини обходів $X(\eta_{x_h y_h})$, що не містять ребро $\{x_h, y_h\}$).

S2. $X = X \cup \{X(\eta_{x_h y_h})\}$, $Q(\eta_{x_h y_h}) = Q(\eta_h) \cup \{\{x_h, y_h\}\}$;

– сформувати з матриці C матрицю C' , в якій кожному ребру списку $Q(\eta_{x_h y_h})$ відповідає пара значень, які дорівнюють ∞ ;

– виконати пошук ЦП $\eta_{x_h y_h}$ алгоритмом розв'язання задачі 2- f для матриці C' ;

– якщо не існує $\eta_{x_h y_h}$, то вершина $X(\eta_{x_h y_h})$ отримує оцінку $C(\eta_{x_h y_h}) = \infty$,

перейти до кроку S3;

– якщо побудований обхід, то $X(\eta_{x_h y_h})$ – неактивна вершина з допустимим

розв'язком ГЗК вартістю $C(\eta_{x_h y_h})$; перейти до кроку S3;

– $\eta_{x_h y_h}$ – ЦП з двох і більше компонент зв'язності, $X(\eta_{x_h y_h})$ – активна вершина, $C(\eta_{x_h y_h})$ – оцінка знизу вартості обходів, що не містять ребро $\{x_h, y_h\}$.

S3. За матрицею C' , і списком $Q(\eta_{x_h y_h})$ відновити матрицю C

(формування множини обходів $X(\eta_{xy})$, тих, що містять ребро $\{x_h, y_h\}$).

S4. У компоненті ЦП η_h , що містить ребро $\{x_h, y_h\}$, знайти ребро $\{p_h, q_h\}$ з другою за величиною після $c_{x_h y_h}$ вагою.

S5. $X = X \cup \{X(\eta_{x_h y_h})\}$, $Q(\eta_{x_h y_h}) = Q(\eta_h) \cup \{\{p_h, q_h\}\}$;

– сформувати з матриці C матрицю C'' в якій кожному ребру списку $Q(\eta_{x_h y_h})$ відповідає пара значень, які дорівнюють ∞ ;

– виконати алгоритм розв'язання задачі 2- f для знаходження ЦП $\eta_{x_h y_h}$ за

матрицею C'' ;

– якщо не існує $\eta_{x_h y_h}$, то вершина $X(\eta_{x_h y_h})$ отримує оцінку $C(\eta_{x_h y_h}) = \infty$;

перейти до кроку S6;

– якщо побудований обхід, то $X(\eta_{x_h y_h})$ – вершина, що відповідає допустимому розв'язку ГЗК $(\eta_{x_h y_h}, C(\eta_{x_h y_h}))$; перейти до кроку S6;

– $\eta_{x_h y_h}$ – ЦП із двох і більше компонент зв'язності, $X(\eta_{x_h y_h})$ – активна вершина, $C(\eta_{x_h y_h})$ – оцінка знизу вартістю обходів, що містять ребро $\{x_h, y_h\}$.

S6. За матрицею C'' і списком $Q(\eta_{x_h y_h})$ відновити матрицю C .

S7. $h = h + 1$; у списку X кінцевих вершин знайти вершину $X(\eta_h)$ з найменшим значенням $C(\eta_h)$;

– якщо $C(\eta_h) = \infty$, то не існує розв'язання ГЗК : кінець;

– якщо η_h – обхід, то $\tau^* = \eta_h$: кінець; інакше перейти до кроку S1.

Слід зазначити, що звернення до алгоритму розв'язання задачі 2- f на кроках S2 і S5 для пошуку ЦП $\eta_{x_h y_h}$ і $\eta_{x_h y_h}$ не завжди приводить до зменшення числа циклів, що містяться в них.

Наприклад, для вхідної матриці C , яка містить велике число однакових значень елементів, число циклів в $\eta_{x_2 y_2}$ або $\eta_{x_2 y_2}$ може співпадати з числом циклів в η_1 .

Природно, що цей факт, відбиваючи певною мірою довільну конструкцію дерева перебору, погіршує його властивості і, отже, веде до збільшення трудомісткості побудови τ^* .

Властивості дерева перебору можна поліпшити заміною алгоритму розв'язання задачі 2- f процедурою розв'язання її обмеженої версії, яку для простоти позначень і без втрати спільності сформулюємо для $h = 1$.

На кроці S2 замість задачі 2- f вирішується така задача.

За відомим ЦП $\eta = \{\eta_1, \eta_2, \dots, \eta_\mu\}$ і ребром $\{x, y\}$ вагою (3.3), що видаляється з η_m , $m \in \{1, 2, \dots, \mu\}$, знайти ЦП η_{xy}° на множині всіх 2-факторів, що містять менше, ніж μ , число циклів.

Процедура знаходження ЦП η_{xy}° складається з таких кроків:

Ш0. V_m – послідовність (x, \dots, y) вершин графа $H = (V, U)$, утворює цикл η_m , якому належить ребро $\{x, y\}$.

Ш1. Визначається підграф $\langle H \rangle$, породжений множиною вершин $(V - V_m) \cup \{x, y\}$.

Ш2. У підграфі $\langle H \rangle$ знаходиться 2-фактор мінімальної ваги η_{xy}° , що містить цикл η° , який проходить по ребру $\{x, y\}$. Якщо не існує η_{xy}° то завдання знаходження η_{xy}° не має розв'язання.

Ш3. З циклу η° видаляється ребро $\{x, y\}$. В результаті його заміни на простий ланцюг, що проходить по усіх вершинах послідовності V_m визначається цикл η_{\max}° , який разом із циклами множини $\eta_{xy}^\circ - \{\eta^\circ\}$ утворює шуканий 2-фактор η_{xy}° . Цикл η_{\max}° є обходом, якщо $\eta_{xy}^\circ - \{\eta^\circ\} = \emptyset$.

На кроці Ш2 побудова ЦП η_{xy}° полягає у включенні в нього ребра $\{x, y\}$ і у виконанні алгоритму розв'язання задачі 2-f, який починається вибором в підграфі $\langle H \rangle - \{x, y\}$ ребра з мінімальною вагою.

На рис. 3.22, а зображено граф $H = (V, U)$ і його ЦП η , в якому $\{x, y\} \in \eta_m$. Цикл η° отримано в результаті побудови в підграфі $\langle H \rangle$ 2-фактора мінімальної ваги η_{xy}° того, що містить ребро $\{x, y\}$ (рис. 3.22, б).

Шукане ЦП η_{xy}° показано на рис. 3.22, в.

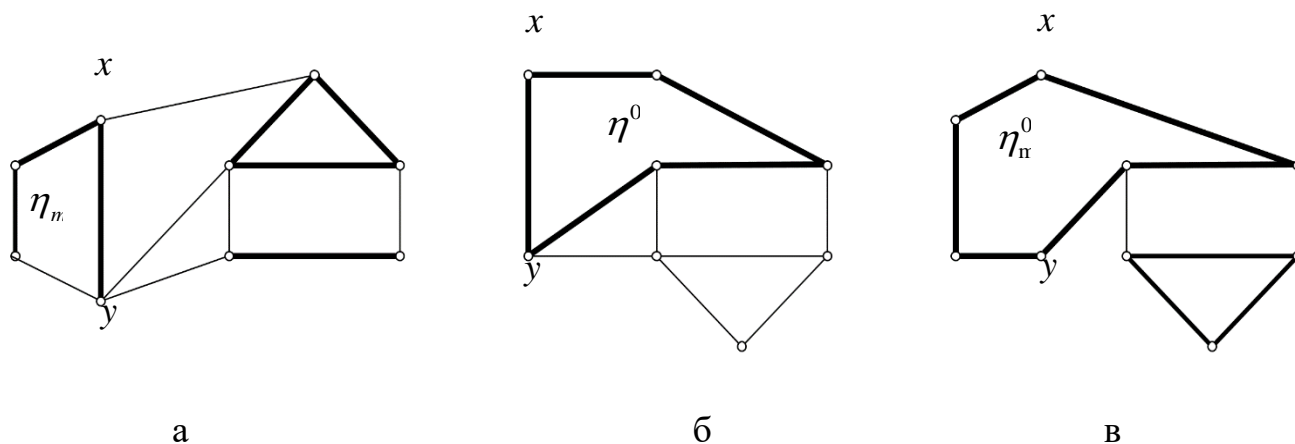


Рисунок 3.22 – Процедура знаходження ЦП η_{xy}° : а – граф $H = (V, U)$ і його ЦП η , в якому $\{x, y\} \in \eta_m$; б – цикл η° отриманий в результаті побудови в підграфі $\langle H \rangle$ 2-фактора мінімальної ваги η_{xy}° того, що включає ребро $\{x, y\}$; в – шукане ЦП η_{xy}°

Обмежена версія задачі 2- f , що вирішується при $h=1$ на кроці S5 модифікованого алгоритму Літтла, має таке формулювання.

Задано ЦП $\eta = \{\eta_1, \eta_2, \dots, \eta_{\mu}\}$ і ребро $\{p, q\}$, яке має вагу (3.4) видаляємо з η_m , $m \in \{1, 2, \dots, \mu\}$. Потрібно знайти ЦП η_{pq}° на множині усіх 2-факторів, що містять менше, ніж μ число циклів.

Очевидно, ЦП η_{pq}° знаходиться в результаті виконання кроків Ш0-Ш3 процедури після заміни ребра $\{x, y\}$ на ребро $\{p, q\}$.

Таким чином, число циклів у побудованих 2-факторах $\eta_{x_1 y_1}^{\circ}$ і $\eta_{x_1 y_1}^{\circ} = \eta_{pq}^{\circ}$ принаймні на одиницю менше, якщо в $\eta_1 = \eta$.

У загальному випадку вершина галуження $X(\eta_h)$ породжує два ЦП $\eta_{x_h y_h}^{\circ}$ і $\eta_{x_h y_h}^{\circ}$ з меншим, ніж в η_h , числом компонент, $h = 2, 3, \dots$.

Застосування процедури побудови $\eta_{x_h y_h}^{\circ}$ і $\eta_{x_h y_h}^{\circ}$ дозволяє оцінити структуру дерева перебору, визначувану його висотою.

Довжина шляху від кореня до довільної вершини x називається глибиною цієї вершини, а максимальна глибина вершин – висотою дерева [39].

Обмежимо згори висоту дерева перебору.

У граничному випадку воно є повним двійковим деревом, тобто таким, усі кінцеві вершини якого мають однакову глибину, і ступінь кожної внутрішньої вершини дорівнює 3 [39].

Якщо кореню дерева T відповідає ЦП, що містить μ компонент, то для кожної вершини глибини 1 процедура будує 2-фактор із числом циклів, меншим або таким, що дорівнює $\mu - 1$.

Очевидно, 2-фактор, що відповідає будь-якій вершині глибини 2, складається з циклів, число яких не більше $\mu - 2$.

Кожна кінцева вершина повного дерева T надана або оцінкою ∞ , або гамільтоновим циклом і його вартістю.

Тому дерево T характеризується висотою $H = \mu - 1 \leq \left\lfloor \frac{n}{3} \right\rfloor - 1$.

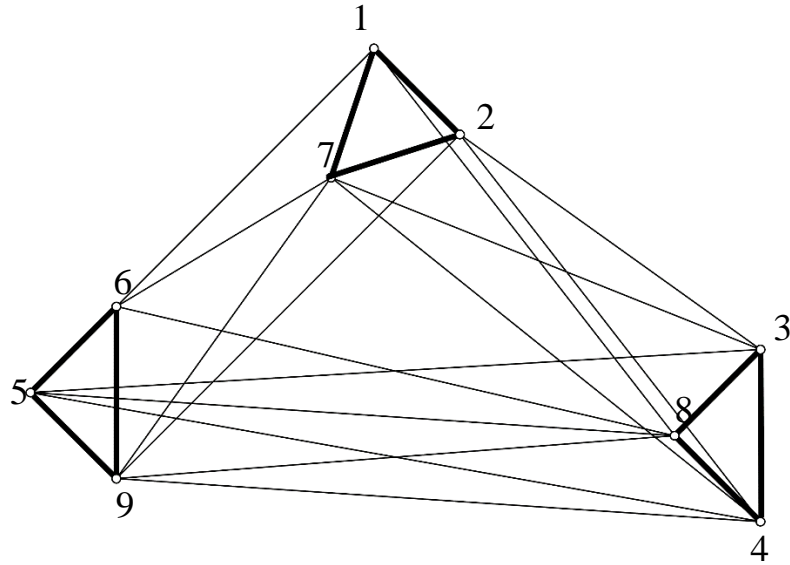
Число некінцевих вершин у повному дереві T заввишки H дорівнює $1 + 2 + 2^2 + \dots + 2^{H-1} = 2^H - 1 = 2^{\mu-1} - 1$.

Це число на одиницю менше числа його кінцевих вершин.

Кількість всіх вершин у T дорівнює $2^{\mu} - 1$.

Оскільки трудомісткість побудови $\eta_{x_h y_h}^{\circ}$ або $\eta_{x_h y_h}^{\circ}$ оцінюється величиною $O(n^3)$, то шукане розв'язання τ^* знаходиться за час $O\left(n^3 2^{\lfloor n/3 \rfloor}\right)$.

Приклад 3.2. Опишемо дії модифікації алгоритму Літгла для розв'язання ГЗК у графі $H = (V, U)$, зображеному на рис. 3.23, з матрицею вартостей C , показано на рис. 3.24.

Рисунок 3.23 – Граф $H = (V, U)$

$j \backslash i$	1	2	3	4	5	6	7	8	9
1	∞	9	∞	∞	∞	17	0	18	∞
2	9	∞	8	17	∞	∞	3	∞	23
3	∞	8	∞	1	10	∞	3	4	∞
4	∞	17	1	∞	10	∞	20	3	26
5	∞	∞	10	10	∞	1	∞	11	4
6	17	∞	∞	∞	1	∞	18	26	13
7	0	3	3	20	∞	18	∞	∞	29
8	18	∞	4	3	11	26	∞	∞	15
9	∞	23	∞	26	4	13	29	15	∞

Рисунок 3.24 – Матриця вартостей C

S0. У графі $H = (V, U)$ методом розв'язання задачі 2- f знаходиться ЦП

$\eta = (\eta_1, \eta_2, \eta_3)$: $\eta_1 = \{\{1,2\}, \{2,7\}, \{7,1\}\}$, $\eta_2 = \{\{3,8\}, \{8,4\}, \{4,3\}\}$, $\eta_3 = \{\{5,6\}, \{6,9\}, \{9,5\}\}$. На рис. 3.23 ребра ЦП η зображені потовщеними лініями $\mu = |\eta| = 3 > 1$.

- $X = \emptyset$ – множина кінцевих вершин дерева перебору;
- $X(\eta)$ – корінь дерева, кореню відповідає ЦП η і його вартість

$$C(\eta) = \sum_{i=1}^3 C(\eta_i) = 9 + 3 + 0 + 4 + 3 + 1 + 1 + 13 + 4 = 12 + 8 + 18 = 38;$$

– $X = X \cup \{X(\eta)\} = \{X(\eta)\}$; $Q(\eta) = \emptyset$ – список ребер, що виключаються в процесі галуження.

$$h = 1; \quad \eta^1 = \eta, \quad C(\eta^1) = C(\eta) = 38, \quad Q(\eta^1) = Q(\eta) = \emptyset, \quad X = X \cup \{X(\eta^1)\} = \{X(\eta), X(\eta^1)\}.$$

S1. $X = X - \{X(\eta^1)\} = \{X(\eta)\}$. Виберемо в η^1 ребро $\{x, y\}$, яке ініціює розгалуження у вершині $X(\eta)$, скориставшись вимогою (3.3).

Цим вимогам задовольняє ребро (6.9) вагою $c_{69} = 13$.

$$S2. \quad X = X \cup \left\{ X \left(\eta_{x_1 y_1}^- \right) \right\} = \{X(\eta), X(\eta_{69}^-)\}, \quad Q(\eta_{69}^-) = Q(\eta^1) \cup \{\{6,9\}\} = \{\{6,9\}\};$$

– сформуємо матрицю C' , поклавши в матриці C $c_{69} = \infty$ і $c_{96} = \infty$;

– після виконання процедури знаходження ЦП η_{69}° для матриці C' ,

отримаємо:

$$\eta_{69}^\circ = \{\{1,6\}, \{6,5\}, \{5,9\}, \{9,8\}, \{8,4\}, \{4,3\}, \{3,2\}, \{2,7\}, \{7,1\}\};$$

$$C(\eta_{69}^\circ) = 17 + 1 + 4 + 15 + 3 + 1 + 8 + 3 + 0 = 52;$$

– покладемо $\eta_{\overline{69}} = \eta_{69}^{\circ}$, $C(\eta_{\overline{69}}) = C(\eta_{69}^{\circ})$.

Побудований обхід, тому вершина $X(\eta_{\overline{69}})$ стає неактивною.

S3. За матрицею C' , і списком $Q(\eta_{\overline{69}})$ відновимо матрицю C .

S4. Оскільки в ЦП η ребро $\{6,9\} \in \eta_3$, то ребро $\{p_1, q_1\}$, що задовольняє умові, є $\{9,5\}$ вагою $c_{95} = 4$.

S5. $X = X \cup \{X(\eta_{69})\} = \{X(\eta), X(\eta_{\overline{69}}), X(\eta_{69})\}$; $Q(\eta_{69}) = Q(\eta^1) \cup \{\{9,5\}\} = \{\{9,5\}\}$;

– поклавши в матриці C $c_{95} = c_{59} = \infty$, отримаємо матрицю C'' ;

– знайдемо для матриці C'' ; ЦП η_{69} і його вартість $C(\eta_{69})$ в результаті звернення до процедури побудови η_{95}° :

$$\eta_{69} = \{\{1,7\}, \{7,2\}, \{2,1\}, \{3,5\}, \{5,6\}, \{6,9\}, \{9,8\}, \{8,4\}, \{4,3\}\};$$

$$C(\eta_{69}) = 0 + 3 + 9 + 10 + 1 + 13 + 15 + 3 + 1 = 55;$$

– ЦП η_{69} складається з двох компонент зв'язності і його вартість більша $C(\eta_{\overline{69}})$.

S6. За матрицею C'' і списком $Q(\eta_{69})$ відновимо матрицю C .

S7. $h = 2$; у списку X кінцевих вершин знайдемо вершину $X(\eta^2)$ з найменшим значенням $C(\eta^2)$; вершині $X(\eta_{\overline{69}})$ відповідає обхід $\eta_{\overline{69}}$ мінімальної вартості $C(\eta_{\overline{69}}) = 52$; таким чином, шуканий маршрут $\tau^* = \eta^2 = \eta_{\overline{69}}$.

Дерево перебору для побудови τ^* із вхідними даними прикладу, показано на рис. 3.25.

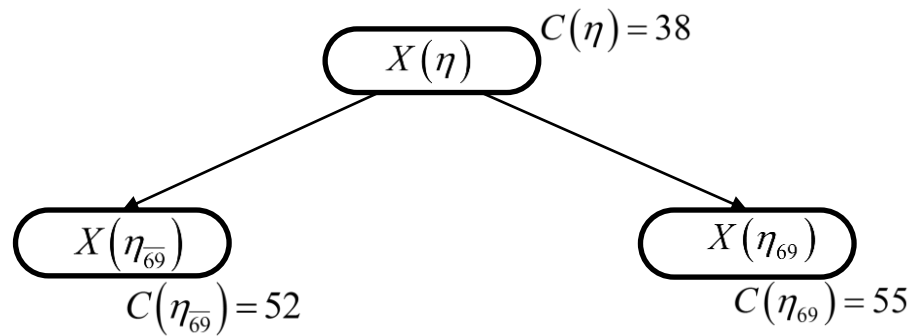


Рисунок 3.25 – Дерево перебору для побудови τ^* з вхідними даними прикладу

Висновки до розділу 3

1. Вперше показано, що задачу 2- f можна звести до ЗП, в цикловому розкладанні якої контур містить не менше трьох дуг.

2. Набув подальшого розвитку рекурентний метод знаходження паросполучення мінімальної ваги в дводольному і довільному графах, що знижує часову складність відомих алгоритмів розв’язання задачі 2- f .

3. Обґрунтовано вибір задачі 2- f для обчислення оцінок знизу вартості маршрутів у ГЗК.

4. Вперше при побудові оптимального маршруту в ГЗК отримано найбільш точний спосіб обчислення нижніх меж, що забезпечує істотне скорочення обчислення в схемах перебору за методом Літгла.

Основні результати першого розділу опубліковано у роботах [124, 125].

Список джерел, які використано у даному розділі, наведено у повному списку використаних джерел [35, 39, 46, 48, 49, 73, 84, 85, 99, 119, 124, 125].

РОЗДІЛ 4

ПРОГРАМНИЙ ПРОДУКТ І РЕЗУЛЬТАТИ ОБЧИСЛЮВАЛЬНОГО ЕКСПЕРИМЕНТУ

У розділі описано процес проектування і розроблення програмного продукту, що дозволяє за вхідними даними будувати маршрути комівояжера на основі ідей, викладених у цій дисертаційній роботі.

Окремою частиною програми винесено модуль для проведення обчислювального експерименту.

4.1. Основні вимоги до програмного продукту

Програмний продукт має реалізовувати такі можливості:

- забезпечувати введення вхідних даних у вигляді матриці відстаней;
- генерувати вихідні задачі за вказаними параметрами;
- виконувати перевірку введених даних;
- знаходити розв'язання задачі про призначення угорським методом, потенціалів і рекурентним методом, розв'язання задачі 2- f ;
- знаходити розв'язання ЗК класичним методом гілок і меж і модифікованим з використанням, як релаксація, методу розв'язання задачі 2- f ;
- проводити обчислювальний експеримент для порівняння часу пошуку розв'язків ЗП різними методами;
- порівнювати час пошуку розв'язання ЗК методом гілок і меж з різними релаксаціями.

4.3. Опис інтерфейсу програмного продукту

Як цільову операційну систему вибрано ОС Microsoft Windows, оскільки, на даний момент, у світі приблизно 80% комп'ютерів користувачів працюють під

управлінням різних версій ОС Windows.

Для реалізації запропонованих методів і розроблення програмного продукту вибрано мову програмування Microsoft Visual C#. Вибір обумовлений рядом переваг, які досягаються при використанні цієї мови. Перерахуємо головні з них:

- можливість за допомогою C# писати код, працюючий на мобільних пристроях під управлінням ОС Windows, Android і iOS;

- велика бібліотека класів і ефективних структур даних, що дозволяють зосередитися не на низькорівневому програмуванні, а саме на реалізації алгоритмів;

- повноцінна підтримка об'єктно-орієнтованої парадигми і сучасних трендів у програмуванні.

Програмний продукт має два можливих способи запуску – з командного рядка виконанням файлу `tspcl.exe` і виконанням файлу `tsp.exe`, що запускає графічний інтерфейс програми.

Програмний продукт має два можливих способи запуску – з командного рядка виконанням файлу `tspcl.exe` і виконанням файлу `tsp.exe`, що запускає графічний інтерфейс програми.

Основними елементами інтерфейсу (рис. 4.1) головного вікна програми є:

- меню «Файл», яке дозволяє зберігати або завантажувати файли з даними (матрицями відстаней);

- таблиця для перегляду і редагування матриці, що займає основну частину вікна програми;

- поля для введення або вибору розмірності завдання;

- кнопки для очищення і генерування матриці;

- кнопка «Вирішити» або «Зупинити»;

- випадний список для вибору методу розв'язання задачі комівояжера (методу гілок і меж із конкретною релаксацією);

- кнопки «Зберегти» і «Завантажити».

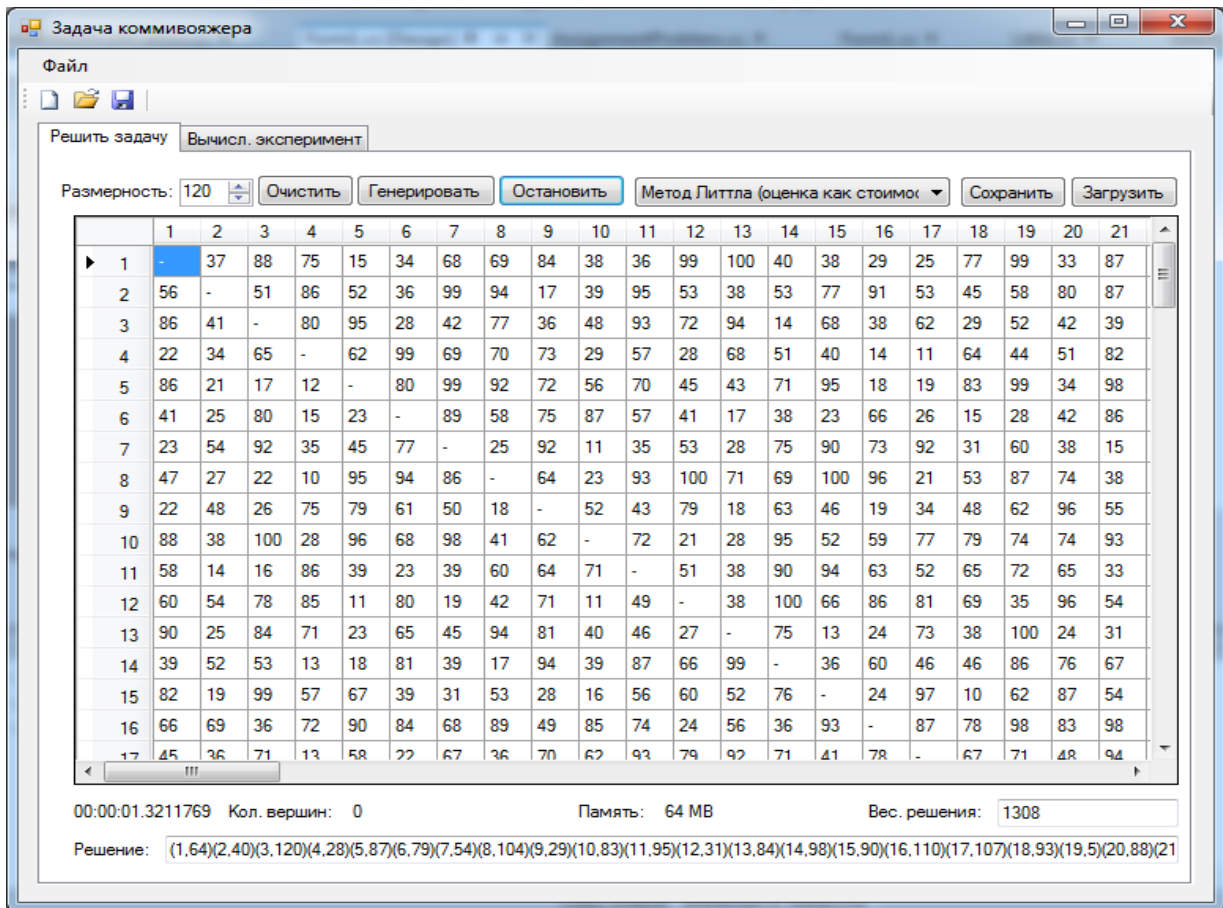


Рисунок 4.1 – Зовнішній вигляд головного вікна програми

Оскільки процес розв’язання задачі займає певний час, в програмі передбачено використання декількох паралельних потоків.

Перший потік управляє інтерфейсом програми, забезпечує оброблення дій користувача і подій операційної системи, а другий потік виконує розв’язання задачі.

Такий підхід дозволяє уникнути деяких проблем, що виникають в однопоточних застосуваннях, таких, як «підвисання» інтерфейсу програми в процесі розв’язання задачі, складність відстежування прогресу розв’язання, неможливість зупинки розв’язання, і т. п.

Завдяки використанню багатопоточності, в нижній частині головного вікна програми відображається інформація про поточний стан розв’язання задачі:

– час, що пройшов від початку розв’язання;

- кількість висячих вершин на даний момент;
- об'єм оперативної пам'яті, використовуваний для зберігання усіх даних програми;
- поточний проміжний оптимум і його вартість.

Для проведення експериментального дослідження реалізованих в програмі методів у головному вікні передбачено вкладку «Вычисл. эксперимент» (рис. 4.2).

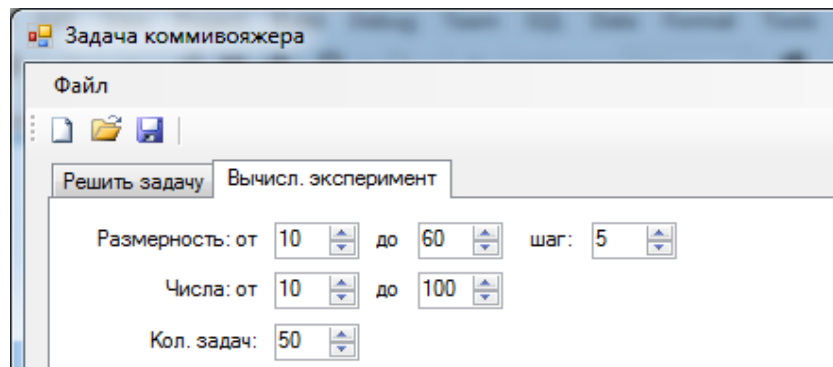


Рисунок 4.2. Параметри експерименту

У верхній частині знаходяться елементи для установлення:

- діапазону розмірності завдання і кроку;
- діапазону чисел, якими заповнюватиметься матриця відстаней;
- кількість завдань, які генеруватимуться;
- прапорці для вибору методів, час роботи яких досліджуватиметься;
- прапорець для генерування симетричних матриць.

Після натиснення на кнопку «Начать» починається обчислювальний експеримент.

Серія випадково згенерованих завдань вирішується вибраними методами.

Фіксується загальний час розв'язання, який потім ділиться на кількість завдань.

При порівнянні методів окрім часу розраховується виграш або програш у часі того або іншого методу.

4.4. Структура програмного продукту

До складу початкового коду програмного продукту входять 112 файлів, серед яких основними є такими, що наведені у табл. 4.1.

Таблиця 4.1. Файли початкового коду програмного продукту

Ім'я програмного файлу	Кількість рядків	Розмір, (байт)
Graph.cs	412	13793
SolutionAlgorithm.cs	80	2143
Program.cs	230	5243
Little.cs	374	14293
AssignmentProblem.cs	1225	19356
AssignmentLittleModification.cs	652	18462
TwoFactor.cs	1652	21423
Сумарний об'єм коду	4625	94713

Файл Graph.cs містить оголошення класів, які використовуються в інших модулях :

- *OrientedEdge* – зберігання дуги або ребра графа;
- *OrientedWeightEdge* – подання зваженої дуги графа;
- *Graph* – реалізація графа;
- *CMatrix* – зберігання матриці;
- *OrientedWay* – подання маршруту комівояжера.

Файл SolutionAlgorithm.cs містить опис абстрактного базового класу SolutionAlgorithm. У цьому класі містяться загальні поля і методи для реалізації методів розв'язання ЗК, ЗП і 2-f.

Файл Little.cs містить реалізацію класичного методу Літла.

У файлі AssignmentProblem.cs реалізовано клас для розв'язання завдання про призначення угорським методом і методом потенціалів.

Файл AssignmentLittleModifications.cs містить реалізацію запропонованих

у дисертаційній роботі модифікацій методу Літтла з використанням різних релаксацій.

У файлі TwoFactor.cs реалізовані класи для розв'язання завдання 2-f рекурентним методом.

Для написання програмного продукту спроектовані 36 класів. Основні з них наведені на діаграмі ієрархії на рис. 4.3.

Для зберігання даних у програмних реалізаціях алгоритмом, запропонованих у розділах 2 і 3 цієї дисертаційної роботи, були вибрані такі структури даних, подані класами-узагальненнями:

- List<T> – клас, що представляє список елементів у вигляді динамічного масиву;
- Queue<T> – клас, що реалізує чергу елементів;
- Stack<T> – стік;
- LinkedList<T> – двозв'язковий список;
- HashSet<T> – множина (неврегульований список унікальних елементів);
- Dictionary<T, V> – колекція пар «ключ-значення».

LinkedList<T> використовується у випадках, коли необхідно виконувати багато операцій додавання елементів у середину списку або видалення елементів. List<T> у даному випадку використати не ефективно, оскільки при видаленні елементу з List<T> виконується зміщення елементів, розташованих за ним, так само, як і при додаванні елемента виконується зміщення. LinkedList<T>, у разі додавання нового елемента дозволяє обійтися простим переприсвоюванням покажчиків на наступний і попередній елементи, що набагато ефективніше, ніж зрушення елементів.

HashSet<T> являє собою структуру даних «множина». Перевагою використання цього класу є автоматичне усунення дублікатів елементів, що містяться в ньому.

Крім того в класі є методи для виконання основних операцій над множинами: `UnionWith` (об'єднання), `IntersectWith` (перетин), `Except` (різниця), `SymmetricExceptWith` (симетрична різниця).

У програмному коді багато операцій виконується за допомогою LINQ (Language – Integrated Query), який дозволяє виконувати операції над звичайними масивами і списками у вигляді запитів, інтегрованих у програмний код. При цьому надається можливість автоматичного розпаралелювання простих операцій над масивами і списками.

4.5. Результати обчислювального експерименту

4.5.1. Етапи експериментального дослідження методів розв'язання ЗП і задачі 2-f

Обчислювальний експеримент проводився на випадково згенерованих наборах даних, які подавалися на вхід алгоритмам розв'язання ЗП і методу розв'язання задачі 2-f.

1. Досліджувався час роботи відомих методів розв'язання ЗП (методу потенціалів та угорського методу) і рекурентного методу, запропонованого в дисертаційній роботі. Генерувалося по 100 випадкових матриць вартостей розмірності від 10 до 490 з кроком 20. Значення, що заповнюють матрицю, вибирали з діапазону від 10 до 100. Кожну серію завдань вирішували трьома методами.

Досліджували залежність часу розв'язання задачі кожним із методів від розміру вхідних даних. Для зменшення похибки підрахунку часу визначали загальний час розв'язання серії з 100 завдань кожним з методів і отриманий час ділили на 100.

2. Досліджували час роботи методу 2-f для отримання 2-фактора мінімальної ваги. Використовували ті ж параметри експерименту і вхідні дані, на яких досліджували час роботи методів розв'язання ЗП.

3. Досліджували час розв'язання ЗК модифікованим методом Літтла, в якому використані різні релаксації для отримання нижніх оцінок вартості маршрутів комівояжера. Генерувалося по 100 випадкових матриць вартостей розмірністю від 10 до 170 з кроком 5.

Діапазон значень, що заповнюють матриці, становив від 10 до 100. Для кожної розмірності вирішували по 100 завдань трьома модифікаціями методу Літтла, в яких використані для отримання нижньої межі методи – $2-f$, рекурентний метод розв'язання ЗП і угорський алгоритм.

4.5.2. Результати обчислювального експерименту щодо дослідження методів розв'язання ЗП і задачі $2-f$

Обчислювальний експеримент проводили на комп'ютері з такими характеристиками устаткування:

- процесор: IntelPentium G4400, тактова частота: 3.3 GHz;
- материнська плата: ASUS H110M - CS/C/SI/Bulk;
- оперативна пам'ять: DDR4 8GB 2400 MHz;
- жорсткий диск: SSD Western Digital 2.5" 120GB, 6Gb/s.

Усереднений час розв'язку завдань для кожної з розмірностей вхідної матриці вартостей наведено в табл. 4.2 і проілюстровано у формі графіків на рис. 4.5.

Таблиця 4.2 – Час розв'язку задачі ЗП і $2-f$

n	Метод потенціалів, с	Угорський алгоритм, с	Рекурентний метод, с	Метод $2-f$, с
1	2	3	4	5
10	1,328	0,515	0,396	1,982
30	2,382	0,952	0,732	3,661

1	2	3	4	5
50	3,233	1,255	0,965	4,828
70	4,512	1,637	1,258	6,294
90	6,144	2,625	2,018	10,094
110	9,219	4,055	3,119	15,597
130	12,008	5,107	3,928	19,643
150	16,360	7,509	5,77	28,882
170	21,542	9,950	7,653	38,269
190	26,457	12,747	9,805	49,028
210	31,871	15,441	11,877	59,389
230	39,394	19,275	14,827	74,135
250	46,851	23,347	17,959	89,797
270	55,083	28,886	22,220	111,100
290	60,749	34,016	27,467	128,867
310	69,745	39,368	31,769	149,173
330	79,381	45,120	36,391	170,999
350	89,657	51,272	41,333	194,345
370	100,573	57,824	46,595	219,211
390	112,129	64,776	52,177	245,597
410	124,325	72,128	58,079	273,503
430	137,161	79,880	64,301	302,929
450	150,637	88,032	70,843	333,875
470	164,753	96,584	77,705	366,341
490	179,509	105,536	84,887	400,327

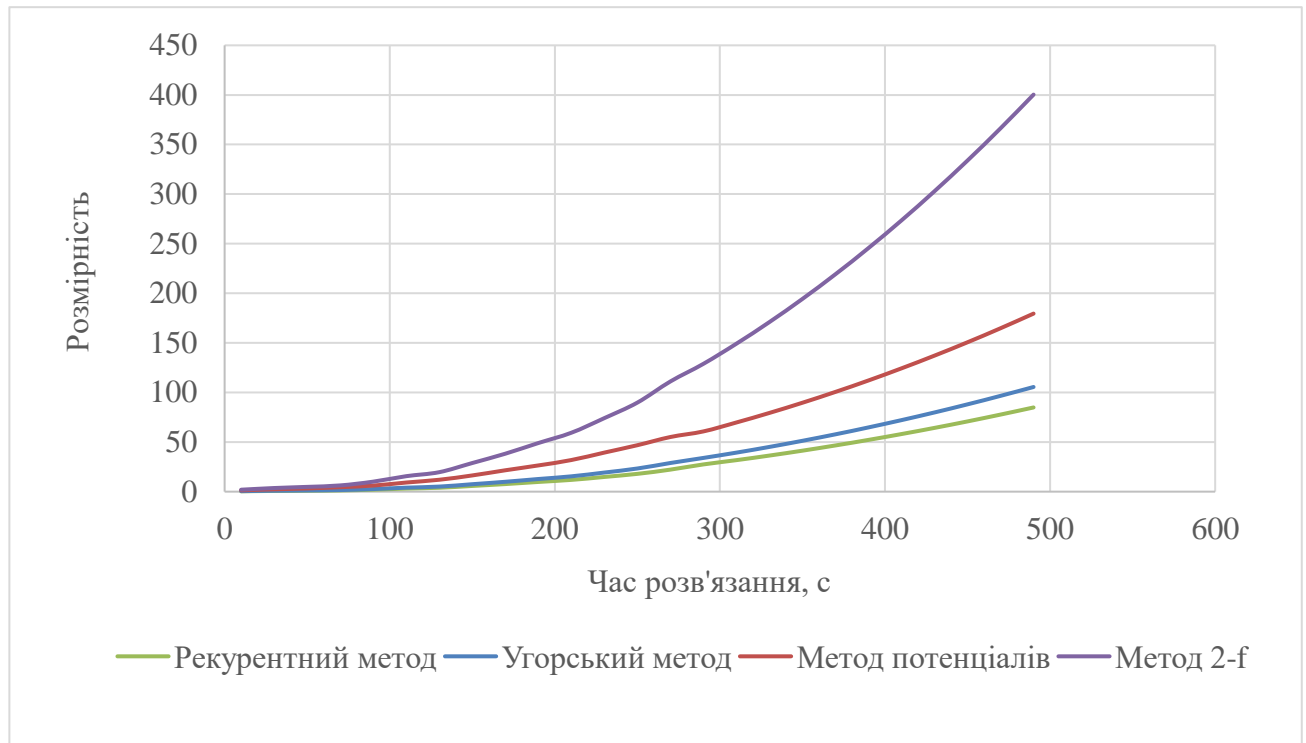


Рисунок 4.5 – Графіки залежності часу розв'язку ЗП і 2-f від розмірності вхідних матриць

Результати дослідження дають можливість зробити висновок, що отримання 2-фактора мінімальної ваги є більше трудомістким, ніж розв'язок ЗП. Порівняння методів розв'язку ЗП показало, що кращі характеристики показав рекурентний метод розв'язку ЗП, наведений в другій главі дисертаційної роботи. Відомі методи (угорський і потенціалів) за одних і тих же умов показують гірші результати.

4.5.3. Результати обчислювального експерименту щодо дослідження модифікації методу Літтла з різними способами отримання нижніх оцінок

У ході обчислювального експерименту вирішували серії по 100 завдань і подальшому усереднюванню загального часу розв'язання. Кожна серія розв'язана модифікованим методом Літтла, в якому використовували різні

способи обчислення нижніх оцінок. Результати експерименту наведені в табл. 4.2 і показані на рис. 4.3.

Таблиця 4.3 – Час розв'язання ЗК модифікованим методом Літтла з різними способами обчислення нижніх оцінок

Розмірність	Час розв'язання класичним методом Літтла, с	Час роботи модифікації методу Літтла			Виграш в порівнянні з класичним методом Літтла, с
		Угорський алгоритм, с	Рекурентний метод, с	$2 - f$, с	
1	2	3	4	5	6
10	00:00	00:00	00:00	00:00	–
15	00:00	00:00	00:00	00:00	2,93
20	00:00	00:00	00:00	00:00	3,12
25	00:00	00:00	00:00	00:00	3,50
30	00:01	00:01	00:00	00:00	3,44
35	00:01	00:01	00:00	00:00	3,28
40	00:02	00:01	00:01	00:01	2,92
45	00:03	00:02	00:01	00:01	2,74
50	00:06	00:03	00:03	00:02	3,31
55	00:08	00:05	00:03	00:03	3,60
60	00:17	00:10	00:07	00:06	2,76
65	00:16	00:10	00:07	00:06	3,22
70	00:15	00:09	00:05	00:04	3,33
75	00:28	00:16	00:11	00:09	3,45
80	00:52	00:31	00:19	00:17	2,95

85	00:55	00:34	00:20	00:16	3,23
90	01:05	00:36	00:25	00:21	3,50
95	02:00	01:19	00:47	00:38	3,37
100	01:44	01:07	00:42	00:31	2,94
105	01:56	01:12	00:52	00:43	2,97
110	02:02	01:14	00:44	00:39	3,20
115	03:11	01:53	01:09	01:00	2,70
120	04:12	02:19	01:37	01:12	3,19
125	03:08	01:57	01:21	00:57	2,90
130	05:24	03:09	02:11	01:35	2,95
135	09:22	05:31	03:27	02:45	2,83
140	10:44	06:18	04:19	03:31	3,18
150	12:42	07:39	05:03	03:37	3,25
155	15:13	08:48	05:24	04:35	3,08
160	20:12	11:48	08:43	06:58	3,40
165	20:47	11:43	08:54	06:34	3,10
170	21:05	11:59	08:06	06:07	3,21

Результати обчислювального експерименту показують, що найменший час розв'язку ЗК демонструє модифікація методу Літтла, що використовує релаксацію $2-f$. Другий результат показує модифікація з рекурентним методом.

Незважаючи на більший час розв'язку задачі $2-f$ порівняно з розв'язком ЗП, використання цієї нижньої оцінки в схемі методу Літтла дає кращий результат. Це пояснюється більш точною нижньою оцінкою, що стримує ріст кількості висячих вершин у дереві розв'язку. З ростом розмірності класичний метод Літтла значно програє іншим методам, з точнішими оцінками.

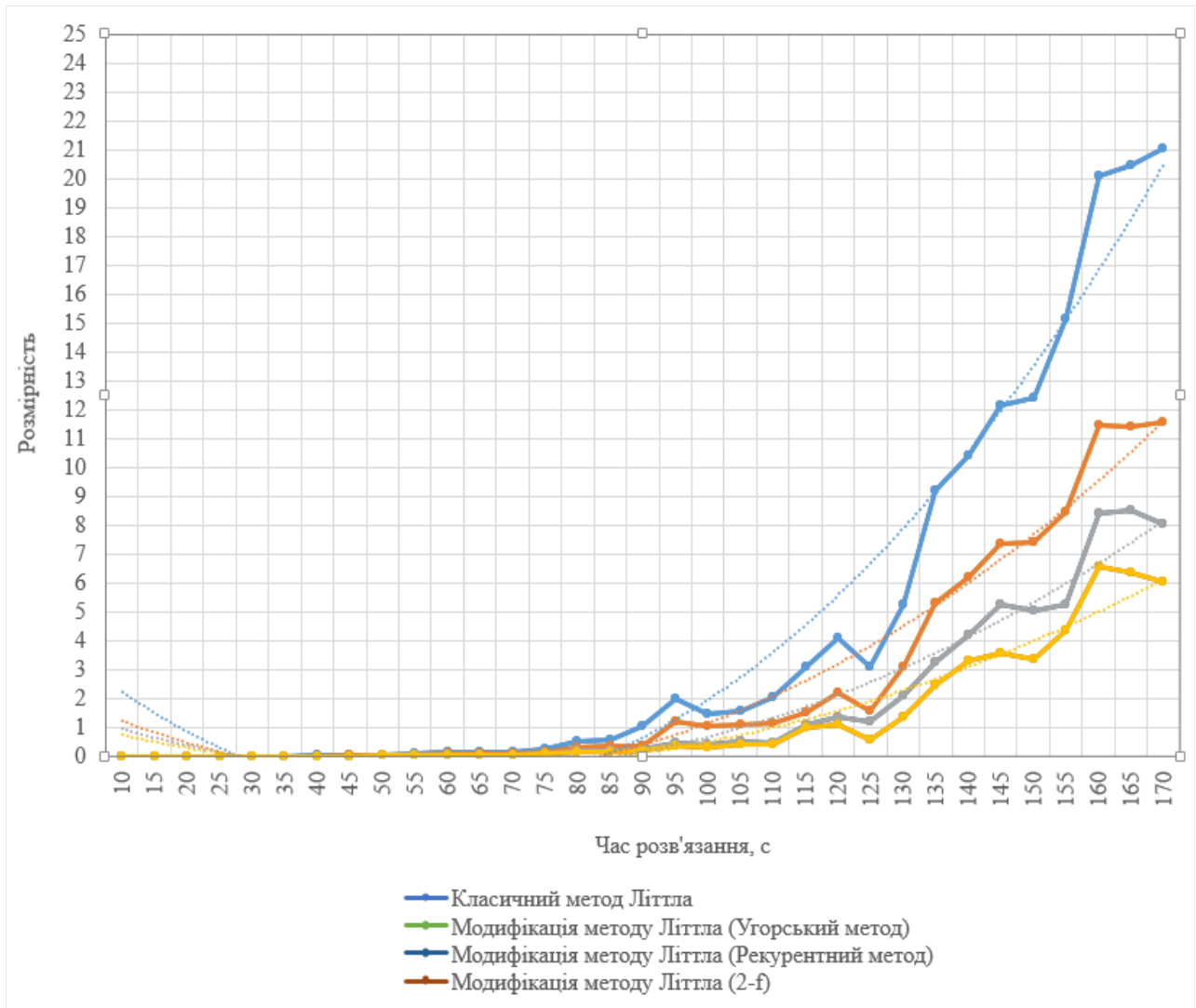


Рисунок 4.6 – Графік залежності часу роботи модифікації методу Літгла з різними способами отримання нижніх меж

Результати обчислювального експерименту показують, що найменший час розв'язку ЗК демонструє модифікація методу Літгла, що використовує релаксацію $2-f$. Другий результат показує модифікація з рекурентним методом.

Незважаючи на більший час розв'язку задачі $2-f$ порівняно з розв'язком ЗП, використання цієї нижньої оцінки в схемі методу Літгла дає кращий результат. Це пояснюється більш точною нижньою оцінкою, що стримує ріст кількості висячих вершин у дереві розв'язку. З ростом розмірності класичний метод Літгла значно програє іншим методам, з точнішими оцінками.

Висновки до розділу 4

1. Розроблено програмне забезпечення для швидкого пошуку розв'язків класу задач маршрутизації на транспортній мережі та експериментальної реалізації викладеного результату.

2. Порівняння методів розв'язання ЗП показало, що кращі характеристики показав рекурентний метод розв'язання ЗП. Відомі методи (угорський і потенціалів) за одних і тих же умов показують гірші результати.

3. Показано, що найменший час розв'язання ЗК демонструє модифікація методу Літтла, в якому використовано релаксацію 2-фактор. Другий результат показує модифікація з розв'язком ЗП рекурентним методом.

Основні результати першого розділу опубліковано у роботах [124, 125].

Список джерел, які використано у даному розділі, наведено у повному списку використаних джерел [52, 58, 64, 77, 124, 125].

ВИСНОВКИ

1. У роботі проведено аналіз сучасного стану проблем маршрутизації транспортних засобів, орієнтований на вдосконалення планування перевезень методами комбінаторної оптимізації та їх програмної реалізації.

2. Набув подальшого розвитку підхід до розв'язання VRP її різновидів. Встановлено, що ряд окремих випадків VRP представлений основними задачами паросполучень. Серед них задачі про призначення і задачі знаходження 2-фактора мінімальної ваги служать релаксаціями, що прискорюють розв'язання тих задач маршрутизації, які потребують побудови кращих обходів пунктів транспортної мережі.

3. Вперше запропоновано загальну схему розв'язання задачі про призначення, задачі знаходження 2-фактора мінімальної ваги і задачі про зважене паросполучення, що виконує рекурентну побудову паросполучень у змішаних графах. Вона містить кінцеве число простих однотипних дій, що знижують трудомісткість відомих алгоритмів.

4. Виходячи із запропонованої схеми отримано нову перестановочно-матричну модель оптимального призначення та розроблено методіку розв'язання задачі про призначення, що не поступається за швидкістю угорському алгоритму.

5. Вперше виконано перетворення довільного зваженого графа в дводольний граф, в якому шукається паросполучення, відповідне розв'язку задачі про зважене паросполучення. На основі перетворення для розв'язання задачі про зважене паросполучення розроблено модифікацію методу оптимального призначення. Тимчасова складність модифікації оцінюється такою ж величиною, як у самих ефективних алгоритмів розв'язання задачі про призначення, менш складною, ніж задача про зважене паросполучення.

6. У роботі встановлено, що задача побудови в будь-якому графі 2-фактора мінімальної ваги зводиться до задачі про паросполучення, в цикловому розкладі якого кожний контур містить не менше трьох дуг. На цій основі розроблено

рекурентний метод розв'язання задачі знаходження 2-фактора мінімальної ваги з меншою трудомісткістю, ніж у відомих методів.

7. Вперше розроблено модифікацію алгоритму Літтла, що містить для підвищення точності нижніх оцінок вартості маршрутів комівояжера та скорочення часу їх побудови розв'язання задачі знаходження 2-фактора мінімальної ваги.

8. Розроблено програмне забезпечення для швидкого пошуку розв'язків класу задач маршрутизації на транспортній мережі та експериментальної перевірки викладеного результату.

9. Порівняння методів розв'язання задачі про призначення показало, що кращі характеристики показав рекурентний метод розв'язання задачі про призначення. Відомі методи (угорський і потенціалів) за одних і тих же умов показують гірші результати.

10. Показано, що найменший час розв'язання задачі комівояжера демонструє модифікація методу Літтла, в якому використано релаксацію – розв'язок задачі знаходження 2-фактора мінімальної ваги. Другий результат показує модифікація з розв'язком задачі про призначення рекурентним методом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Агеев А. А., А. Е. Бабурин А. Е., Э. Х. Гимади Э. Х. Полиномиальный алгоритм с оценкой точности $3/4$ для отыскания двух непересекающихся гамильтоновых циклов максимального веса // Дискретный анализ и исследование операций. 2006. Серия 1. Т. 13. № 2. С. 11-20.
2. Агеев А. А., Пяткин А. В. Приближённый алгоритм решения метрической задачи о двух коммивояжёрах с оценкой точности 2 // Дискретный анализ и исследование операций. 2009. Т. 16. № 4. С. 3-20.
3. Литтл Д. Ж., Мурти К., Суини Д., Кэрел К. Алгоритм для решения задачи коммивояжера // Экономика и математические методы. 1965. Т. 1. вып. 1. С. 90-107.
4. Алексеев М. Н., Алексеев Ф. М. Об алгоритмах оптимальной разметки. Челябинский физико-математический журнал // Челябинский физико-математический журнал. 2016. Т.1. вып.1. С. 16-23.
5. Бабурин А. Е., Гимади Э. Х., Коркишко Н. М. Приближенные алгоритмы для нахождения двух реберно непересекающихся гамильтоновых циклов минимального // Дискретный анализ и исследование операций. 2007. Серия 2. Т. 11. № 1. С. 11-25
6. Базилевич Р. П., Кутельмах Р. К. Алгоритм оптимізації розв'язків задачі комівояжера у локальній області // Радіоелектрон. і комп'ют. системи. 2009. №7. С. 41-45.
7. Беллман Р. Применение динамического программирования к задаче о коммивояжере // Кибернетический сборник. 1964. Вып. 9. С. 219-222.
8. Береснев В. Л., Мельников А. А. Алгоритм ветвей и границ для задачи конкурентного размещения предприятий с предписанным выбором поставщиков // Дискретный анализ и исследование операций. 2014. Том 21. №2. С.3-22.
9. Борханов И. Ф., Фазылов В. Р. Метод Литтла с оптимальным приведением матрицы // Ученые записки Казанского государственного университета. Физико-математические науки. 2006. Т. 148. кн. 4. С. 13–21.

10. Борханов И. Ф., Фазылов В. Р. Метод Литтла со штрафами для решения задачи о развозке // Ученые записки Казанского государственного университета. Физико-математические науки. 2008. Т. 150. Кн. 4. С. 88–97.

11. Борханов И. Ф., Фазылов В. Р. Об оптимальном приведении матрицы стоимостей // Ученые записки Казанского государственного университета. Физико-математические науки. 2006. Т. 148. Кн. 2. С. 18–22.

12. Бронштейн Е. М., Зайко Т. А. Детерминированные оптимизационные задачи транспортной логистики // Автоматика и телемеханика. 2010. №10. С. 133–147.

13. Гаращенко И. В., Морозов А. В., Панишев А. В. Метод решения гамильтоновой задачи коммивояжера // Искусственный интеллект. 2008. №3. С. 630 - 637.

14. Гаращенко И. В., Панишев А. В., Маций О. Б. Полиномиальное преобразование в приближенных алгоритмах решения задач типа коммивояжера // Радиоэлектроника и информатика. 2007. № 1. С. 45–49.

15. Гимади Э. Х., Сердюков А. И. О некоторых результатах для задачи коммивояжера на максимум // Дискретный анализ и исследование операций. 2001. Серия 2. Т. 8. №1. С. 22–39.

16. Гимади Э. Х., Шахшнейдер А. В. Приближенные алгоритмы с оценками для задач маршрутизации на случайных входах с ограниченным числом клиентов в каждом маршруте // Автоматика и телемеханика. 2012. № 2. С. 126–139.

17. Гимади Э. Х., Истомина А. М., Рыков И. А. О задаче нескольких коммивояжеров с ограничениями на пропускные способности ребер графа // Дискретный анализ и исследование операций. 2013. Том 20. №5. С. 13–30.

18. Глебов А. Н. Замбалаева Д. Ж. Полиномиальный алгоритм с оценкой точности $7/9$ для задачи о двух коммивояжерах на максимум // Дискретный анализ и исследований операций. 2011. Т. 18. № 4. С. 17–48.

19. Глебов А. Н. Замбалаева Д. Ж. Приближенный алгоритм решения задачи о двух коммивояжерах на минимум с различными весовыми функциями //

Дискретный анализ и исследований операций. 2011. Т. 18. № 5. С. 17–37.

20. Глебов А. Н. Замбалаева Д. Ж., Скретнова А. А. 2/3 – приближенный алгоритм для несимметричной задачи о двух коммивояжерах на максимум // Дискретный анализ и исследование операций. 2014. Том 21. №6. С. 11–27.

21. Глебов Н. И. Об одном обобщении минимаксной задачи о назначениях. Дискретный анализ и исследование операций. 2004. Серия 1. Т. 11. №4. С. 36–43.

22. Гребенник И. В. Описание и генерация перестановок, содержащих циклы // Кибернетика и системный анализ. 2010. № 6. С. 97–105.

23. Гуляницкий Л. Ф., Самусь А. В. Решение N-методом задачи оптимизации маршрутов транспортных средств с временными окнами // Компьютерная математика. 2012. №2. С. 147–155.

24. Гуляницкий Л. Ф., Гобов Д. А. Застосування N-методу для розв'язання задач комбінаторної оптимізації на перестановках // Системні дослідження та інформаційні технології. 2007. № 2. С. 74–86.

25. Гуляницкий Л. Ф., Мулеса О. Ю. Прикладні методи комбінаторної оптимізації // Навч. посіб. Видавничо-поліграфічний центр "Київський університет", 2016. 142 с.

26. Демиденко В. М. Построение релаксации политопа симметричной задачи о коммивояжере на основе сильно разрешимого случая Кальмансона // Дискретный анализ и исследование операций. 2004. Серия 2. Т. 11. № 2. С. 3–24.

27. Демиденко В. М. Релаксационный политоп симметричной задачи о коммивояжере, порождаемый конусом матриц Супника // Весці Нацыянальнай акадэміі навук Беларусі. Серыя фізіка-матэматычных навук. 2007. № 2. С. 109–115.

28. Джонсон Д. С., Вуд Д. Ф. Современная логистика // Москва, 2005. 624 с.

29. Диниц Е. А., Кронрод М. А. Один алгоритм решения задачи о назначениях // Доклады АН СССР, 1969. Т. 189. № 1. С. 23–25.

30. Донец Г. А., Шор Н. З. Алгебраический подход к проблеме раскраски плоских графов // Наук. думка. 1982. 141с.

31. Емец О. А., Парфенова Т. А. Транспортные задачи на перестановках:

свойства оценок в методе ветвей и границ // Кибернетика и системный анализ. 2010. № 6. С. 106–112.

32. Еремеев А. В. О сложности оптимальной рекомбинации для задачи коммивояжера // Дискретный анализ и исследование операций. 2011. Т. 18. №1. С. 27–40.

33. Ефремов П.В. Паросочетания в геометрических двудольных графах. Дискретный анализ и исследование операций // 1997. Серия 2. Том 4. №2. С.10– 22.

34. Згуровский М. З., Павлов А. А. Принятие решений в сетевых системах с ограниченными ресурсами: Монография // Наук. думка. 2010. 573 с.

35. Иванко Е.Е. Метод масштабирования в приближенном решении задачи коммивояжера // Автоматика и телемеханика. 2011. №12. С. 115–129.

36. Иванко Е.Е. Усеченный метод динамического программирования в замкнутой задаче коммивояжера с симметричной функцией стоимости // Труды института математики и механики УрО РАН. 2013. Том 19. №1. С.122–128.

37. Касьянов В. Н., Евстигнеев В. А. Графы в программировании: обработка, визуализация и применение // БХВ-Петербург, 2003. 1104 с.

38. Компьютер и задачи выбора // Автор предисловия Ю. И. Журавлёв. Наука. 1989. 208 с.

39. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ // МЦНМО. 2000. 960 с.

40. Костюк Ю. Л., Жихарев С. А. Эффективный алгоритм приближенного решения метрической задачи коммивояжера // Дискретный анализ и исследование операций. 2000. Серия 2. Том 7. № 1. С. 65–74.

41. Костюк Ю. Л., Пожидаев М. С. Приближенные алгоритмы сбалансированной задачи k коммивояжеров // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2008. №1 (2). С.106–112.

42. Костюк Ю.Л. Задача коммивояжера: улучшенная нижняя оценка в методе ветвей и границ // Прикладная дискретная математика. 2013. №4 (22).

С.74–81.

43. Костюк Ю.Л. Эффективная реализация алгоритма решения задачи коммивояжера методом ветвей и границ // Прикладная дискретная математика. 2013. №2 (20). С. 78–90.

44. Кристофидес Н. Теория графов. Алгоритмический подход // Мир. 1978. 432 с.

45. Кушнір Н. А., Маций О. Б., Скачков В. А. Метод найкоротших збільчуючих шляхів у задачах про паросполучення // Вісник ЖДТУ. 3(74). 2016. С. 101-111.

46. Левченко А. Ю., Морозов А. В., Панишев А. В. Механизм ускорения вычислений в методе Литтла для решения задач класса коммивояжера // Искусственный интеллект. 2012. Вып. 2. С. 96–110.

47. Левченко А. Ю., Маций О. Б., Панишев А. В. Оптимізація замкнених маршрутів на транспортній мережі // Штучний інтелект. Національна академія наук України. Інститут проблем штучного інтелекту. 2010. № 1. С 43-49.

48. Левченко А. Ю., Морозов А. В., Панишев А. В. Быстрый алгоритм решения задачи о назначениях для нахождения нижней границы стоимости маршрута коммивояжера // Искусственный интеллект. – 2011. Вып. 4. С. 406–416.

49. Ловас Л., Пламмер М. Прикладные задачи теории графов. Теория паросочетаний в математике, физике, химии. Мир. 1998. 653 с.

50. Люггер Д. Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем // Москва – Санкт-Петербург – Киев, 2005. 863 с.

51. Майника Э. Алгоритмы оптимизации на сетях и графах // Мир. 1981. 323 с.

52. Макки А. Введение в .NET 4.0 и Visual Studio 2010 для профессионалов // Издательство «Вильямс». 2010. 416 с.

53. Максименко А.Н. Многогранники задачи о выполнимости являются гранями многогранника задачи коммивояжера // Дискретный анализ и исследование операций. 2011. Т. 18. №3. С.76–83.

54. Маций О. Б. Класс задач маршрутизации, сводимых к задаче коммивояжера // Автомобіль і Електроніка. Сучасні Технології. 2017. №12. С. 167–170.

55. Маций О. Б. Удосконалення угорського методу рішення задачі про призначення. Открытые информационные и компьютерные технологии // 2014. № 66. С. 166–171.

56. Маций О.Б. Повышение точности симметричной задачи коммивояжера большой размерности // Вестник Харьковского национального автомобильно-дорожного университета. 2011. Выпуск 55. С 100–103.

57. Медведева О.А. Задача о назначениях с возможностью обучения. Вестник Санкт-Петербургского Университета. 2013. Сер.10. Вып.1. С. 85–93.

58. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера // Вопросы теории. Автоматика и телемеханика. 1989. № 9. С. 3–33.

59. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Приближенные алгоритмы // Автоматика и телемеханика. 1989. № 11. С. 3–26.

60. Меламед И. И., Сергеев С. И., Сигал И. Х. Задача коммивояжера. Точные алгоритмы. Автоматика и телемеханика. 1989. № 10. С. 3–29.

61. Морозов А. В., Панишев А. В. Вершинно-рёберное преобразование в гамильтоновой задаче о сельском почтальоне // Искусственный интеллект. 2009. Вып. 4. С. 138–143.

62. Морозов А. В., Панишев А. В. Метод гілок та меж у гамільтоновій задачі про сільського листоношу // Системні дослідження та інформаційні технології. 2012. №2. С. 57–66.

63. Морозов А. В. Про одне узагальнення задачі комівояжера. Вісник Житомирського державного технологічного університету // Технічні науки. 2009. №3 (50). С.161–171.

64. Нечипуренко М. И., Попков В. К., Маймагалиев С. М. Алгоритмы и программы решения задач на графах и сетях // Наука. Сиб. отд-ние. 1990. 515 с.

65. Овезгельдыев А. О., Морозов А. В. Развитие метода ветвей и границ в задаче поиска оптимального кольцевого маршрута // Кибернетика и системный

анализ. 2013. №5. С. 112–123.

66. Оре О. Теория графов // Наука. 1968. 352 с.

67. Панишев А. В., Плечистый Д. Д. Модели и методы оптимизации в проблеме коммивояжера // Житомир: ЖГТУ. 2006. 300 с.

68. Панишев А. В., Плечистый Д. Д., Шевченко В. А. Процедуры локального поиска в комбинаторных алгоритмах решения общей задачи коммивояжера // Искусственный интеллект. 2005. С. 465–470.

69. Панишев А.В., Морозов А. В., Скачков В. А. Модификация метода Литтла для решения кольцевой задачи о сельском почтальоне // Искусственный интеллект. 2010. Вып. 3. С. 103–115.

70. Панішев А. В., Данильченко О. М., Скачков В. А. Вступ до теорії складності дискретних задач // Житомир: ЖДТУ, 2004. 326 с.

71. Панішев А. В., Плечистий Д. Д., Скачков В. А. Вузлові питання задачі комівояжера. 1 // Вісник Житомирського державного технологічного університету. 2004. № 29. С. 198–204.

72. Панішев А. В., Костикова М. В., Плечистий Д. Д. Вузлові питання задачі комівояжера. 2 // Вісник Житомирського державного технологічного університету. 2004. № 30. С. 99–108.

73. Панишев А. В., Морозов А. В. Модели и методы оптимизации замкнутых маршрутов на транспортных сетях // Житомир. ЖГТУ. 2014. 324 с.

74. Панов С. А. Модели маршрутизации на автомобильном транспорте. Транспорт. 1974. 152 с.

75. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность // Мир. 1985. 510 с.

76. Попов Л. Д. Опыт многоуровневого распараллеливания метода ветвей и границ в задачах дискретной оптимизации // Автоматика и телемеханика. 2007. № 5. С. 171–181.

77. Посыпкин М. А., Сигал И. Х. Оценки ускорения для некоторых вариантов параллельной реализации метода ветвей и границ // Журнал вычислительной математики и вычислительной физики. 2006. Т. 46. № 12.

С. 2289 - 2304.

78. Посыпкин М. А., Сигал И. Х. Применение параллельных эвристических алгоритмов для ускорения параллельного метода ветвей и границ // Журнал вычислительной математики и вычислительной физики. 2007. Т. 47. № 9. С. 1524–1537.

79. Резер С. М., Ловецкий С. Е., Меламед И. И. Математические методы оптимального планирования в транспортных системах // Итоги науки и техники. Серия «Организация управления транспортом». 1990. 171 с.

80. Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика // Мир. 1980. 476 с.

81. Робертс Ф. С. Дискретные математические модели с приложением к социальным, биологическим и экономическим системам // Наука. 1986. – 496 с.

82. Саати Т. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы // Мир. 1973. 330 с.

83. Свами М., Тхуласираман К. Графы, сети, алгоритмы. Мир. 1984. 454 с.

84. Сергеев С. И. Симметричная задача коммивояжера II // Новые нижние границы. Автоматика и телемеханика. 2010. №4. С. 150–167.

85. Сергеев С. И. Симметричная задача коммивояжера. Новые быстрые нижние границы для задачи оптимального 2-паросочетания // Автоматика и телемеханика. 2009. № 11. С. 148–160.

86. Сергеев С. И. Гибридные системы управления и динамическая задача коммивояжера // Автоматика и телемеханика. 2008. №1. С. 45–54.

87. Сергиенко И. В. Математические модели и методы решения на ЭВМ комбинаторных задач оптимизации // Наук. думка. 1988. 472 с.

88. Сергиенко И. В., Каспшицкая М. Ф. Модели и методы решения на ЭВМ комбинаторных задач оптимизации // Наук. думка. 1981. 288 с.

89. Сергиенко И. В., Шило В. П. Задачи дискретной оптимизации: проблемы, методы решения, исследования: Монография // Нац. акад. наук Украины, Ин-т кибернетики им. В. М. Глушкова. Наук. Думка. 2003. 258 с.

90. Сердюков А. И. Полиномиальные алгоритмы с оценками точности

решений для одного класса задач коммивояжера на максимум. Комбинаторно-алгебраические методы в дискретной оптимизации // Нижний Новгород: Нижегород. гос. ун-т. 1991. С. 107–144.

91. Сигал И. Х., Иванова А. П. Введение в прикладное дискретное программирование // Физматмет. 2002. 237 с.

92. Сигал И. Х. Дискретный подход к решению задачи коммивояжера большой размерности и некоторые его применения // Известия АН СССР. Технич. кибернетика. 1990. № 6. С. 143–155.

93. Стоян Ю. Г., Яковлев С. В. Математические модели и оптимизационные методы геометрического пространства // Наук. думка. 1986. 268 с.

94. Стоян Ю. Г., Ємець О. О. Теорія і методи евклідової комбінаторної оптимізації // Ю. Г. Стоян, О. О. Ємець. ІСДО. 1993. 188 с.

95. Танаев В. С., Сотсков Ю. Н., Струсевич В. А. Теория расписаний. Многостадийные системы // Наука, 1989. 328 с.

96. Тараканов В. Е. Комбинаторные задачи и $(0, 1)$ -матрицы // Наука, 1985. 189 с.

97. Тарьян Р. Э. Сложность комбинаторных алгоритмов // Кибернетический сборник. 1980. Вып. 17. С. 60–113.

98. Теория расписаний и вычислительные машины // под ред. Э. Г. Коффмана. Наука. 1984. 334 с.

99. Харари Ф. Теория графов // Мир. 1973. 300 с.

100. Хелд М., Карп Р. Применение динамического программирования к задачам упорядочения // Кибернетический сборник. 1964. Вып. 9. С. 208–218.

101. Ченцов А. А., Ченцов А. А. О реализации метода динамического программирования в обобщенной задаче курьера // Труды института математики и механики УрО РАН. 2007. Том 13. № 3. С. 137–160.

102. Ченцов А. А., Ченцов А. Г., Ченцов П. А. Экстремальная задача маршрутизации с внутренними потерями // Труды института математики и механики УрО РАН. 2008. Том 14. № 3. С. 183–201.

103. Ченцов А. А., Ченцов А. Г. Экстремальная задача маршрутизации «на

узкие места» с ограничениями в виде условий предшествования // Труды института математики и механики УрО РАН. 2008. Том 14. № 2. С. 129–142.

104. Ченцов А. Г. О структуре одной экстремальной задачи маршрутизации с ограничениями в виде условий предшествования // Вестник Удмуртского университета. 2006. № 1. С. 127–150.

105. Ченцов А.Г. К вопросу о маршрутизации комплекса работ // Вестник Удмуртского университета. 2013. С.60–82.

106. Шарифов Ф.А. Совершенные паросочетания и расширенный полиматроид // Кибернетика и системный анализ . 2008. №3. С.173–179.

107. Шаріфов Ф.А., Кандиба Г. Ю. Оптимізація маршрутів повітряних суден, що виконують агроавіаційні роботи // Наукові технології. 2014. №3 (23). С. 319–326.

108. Яблонский А. А. Минимизация кольцевых маршрутов доставки продукции потребителям // Экономика и математические методы. 2006. Т. 42. № 3. С. 124–131.

109. Яковлев С. В., Гребенник И. В. Локализация решений некоторых нелинейных целочисленных задач оптимизации // Кибернетика и системный анализ. 1993. № 5. С. 116–123.

110. Barvinok A. I. Two algorithmic results for the traveling salesman problem // Math. Oper. Res. 1996. V. 21, № 1. P. 65–84.

111. Bellman R. Dynamic Programming Treatment of the Traveling Salesman Problem // J. ACM. 1962. № 9. P. 61–63.

112. Berbeglia G., Cordeau J. F., Gribkovskaia I., Laporte G. Static pickup and delivery problems: A classification scheme and survey // TOP. 2007. V. 15. №1. P. 1–31.

113. Caprard A., Toht P., Fischetti M. Algorithms for set covering problem // Annals of Operations Research. 2000. Vol. 98. P. 353–371.

114. Chauhan S. S., Proth I. M. The concave cost supple problem // European J. Oper. Res. 2003. V. 148, № 2. P. 374–383.

115. De Brey M. Y. D., Volgenant A. Welt-solved cases of the 1-peripatetic

Salesman Problem // Optimization. 1997. Vol. 39. № 3. P.275–239.

116. De Kort J. B. J. M. A branch and bound algorithm for symmetric 2-peripatetic salesman problems // Europ. J. European Res. 1993. Vol. 70. P. 229–243.

117. De Kort J. B. J. M. Lower bounds for the symmetric 2-peripatetic salesman problem // Optimization. 1991. V. 22. №1. P. 113–122.

118. Edmons E. Paths, trees and flowers // Canada J. Math. 1965. V. 17. P. 449–467.

119. Gabow H. N. A good algorithm for smallest spanning trees with a degree constraint // Networks. 1978. 8. № 3. P. 201–208.

120. Garashchenko I, Panishev A. Method of Finding Hamilton Routes in Transport Network. Artificial Intelligence and Decision Making // Sofia: ITHEA, 2008. № 7. P. 43–48.

121. Hopcroft J. E., Tarjan R. E. Algorithm 447: Efficient Algorithms for Graph Manipulation // Comm. ACM. 1973. 16. P. 372–378.

122. Karp R. M. The probabilistic analysis of some combinatorial search algorithms. Algorithms and Complexity: New Directions and Recent Results // New York: Acad. Press, 1976. P. 1–19.

123. Levchenko A., Panishev A. B. Cycle routes optimization for not full graph // Information Models of Knowledge. Kiev-Sofia, 2010. P 435–441.

124. Matsiy O. B., Morozov A. V., Panishev A. V. The Recurrent Method to Solve the Assignment Problem // Cybernetics and Systems Analysis. 2015. Vol. 51. No 6. Pp. 939-946.

125. Matsiy O. B., Morozov A. V., Panishev A. V. Fast Algorithm to Find 2-Factor of Minimum Weight // Cybernetics and Systems Analysis. 2016. Vol. 32. No 3. Pp 464-474.

126. Matsiy O. B., Morozov A. V., Panishev A. V. A Recurrent Algorithm to Solve the Weighted Matching Problem // Cybernetics and Systems Analysis. 2016. Vol. 52. No 5. Pp 748-753.

127. Morozov A., Panishev A. Modified branch and bound algorithm for solving the Hamiltonian Rural Postman Problem. Information Models of Knowledge // Kiev-

Sofia, 2010. P. 442–450.

128. Ovezgeldyyev A.O., Morozov A. B. Developing the Branch and Bound Method in the Problem of Searching for the Optimal Cyclic Route (Cyclic Rural Postman Problem) // Cybernetics and Systems Analysis. September 2013. Volume 49. Issue 5. P. 739–748.

129. Panishev A. V., Plechystyy D. D. An effective exact algorithm for one particular case of the traveling-salesman problem // International journal «Information Theories & Applications». 2003. Vol 10. P. 355–359.

130. Papadimitriou C. H., Yannakakis M. The traveling salesman problem with distance one and two // Math. Oper. Res. 1993. V. 18. № 1. P. 1–11.

131. Plesnik J. Acta Math. Univ // Comenian. 1983. V. 42 – 43. P. 271–273.

132. Ramdane-Cherif W. Problèmes d'optimisation en tournées sur arcs: Thèse Pour l'obtention du Grade de Docteur de L'UTT // Wahiba Ramdane-Cherif. Troyes, 2002. 186 p.

133. Tarjan R. E. Depth – First Search and Linear Graph Algorithms // SIAM J. Comput. 1972. 1. P. 146–160.

134. Toth P., Vigo P. Branch-and-bound algorithms for the capacitated VRP. Vehicle Routing Problem // SIAM. 2001. P. 29–51.

ДОДАТОК А

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЙНОЇ РОБОТИ

1. Геращенко И. В., Маций О. Б., Панишев А. В. Полиномиальное преобразование в приближенных алгоритмах решения задач типа коммивояжера // Радиоэлектроника и информатика. 2007. №1. С. 45-49.

2. Левченко А. Ю., Маций О. Б., Панишев А. В. Оптимізація замкнених маршрутів на транспортній мережі // Штучний інтелект. 2010. № 1. С 43-49.

3. Маций О. Б. Повышение точности симметричной задачи коммивояжера большой размерности // Вестник Харьковского национального автомобильно-дорожного университета. 2011. № 55. С 100-102.

4. Маций О. Б. Удосконалення угорського методу рішення задачі про призначення // Открытые информационные и компьютерные технологии. 2014. № 66. С. 166-171.

5. Matsiy O. B., Morozov A. V., Panishev A. V. The Recurrent Method to Solve the Assignment Problem // Cybernetics and Systems Analysis. 2015. Vol. 51. No 6. Pp. 939-946.

6. Кушнір Н. А., Маций О. Б., Скачков В. А. Метод найкоротших збільшуючих шляхів у задачах про паросполучення // Вісник Житомирського державного технологічного університету. 2016. № 3(74). С. 101-111.

7. Matsiy O. B., Morozov A. V., Panishev A. V. Fast Algorithm to Find 2-Factor of Minimum Weight // Cybernetics and Systems Analysis. 2016. Vol. 32. No 3. Pp 464-474.

8. Matsiy O. B., Morozov A. V., Panishev A. V. A Recurrent Algorithm to Solve the Weighted Matching Problem // Cybernetics and Systems Analysis. 2016. Vol. 52. No 5. Pp 748-753.

9. Маций О. Б. Класс задач маршрутизации, сводимых к задаче коммивояжера // Автомобіль і Електроніка. Сучасні Технології. 2017. № 12.

С. 167-170.

10. Никонов О. Я., Маций О. Б. Подход к решению симметричной задачи коммивояжера // Сучасні проблеми математичного моделювання та обчислювальних методів: матеріали Всеукраїнської наукової конференції. Рівне: РДГУ-НУВГ, 2013. С. 192.

11. Маций О. Б. Про один підхід до рішення задачі комівояжера з симетричною матрицею відстаней // Інформаційні технології і мехатроніка: матеріали Всеукраїнської науково-практичної конференції. Харків: ХНАДУ, 2014. С. 83.

12. Маций О. Б., Шаригін Г. М. Сучасні аспекти моделювання маршрутів перевезення // Інформаційні технології і мехатроніка: матеріали Всеукраїнської науково-практичної конференції. Харків: ХНАДУ, 2014. С. 160-161.

13. Маций О. Б. Рекуррентный метод решения задачи о назначении // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я (MicroCAD-2014): матеріали XXII міжнародної науково-практичної конференції. Харків: НТУ ХПІ, 2014. С. 249.

14. Маций О. Б. Метод ускоренного поиска кратчайших гамильтоновых маршрутов // Обчислювальний інтелект (результати, проблеми, перспективи): матеріали III міжнародної науково-практичної конференції. Київ-Черкаси: М-во освіти і науки України, Київ. нац. ун-т імені Тараса Шевченка та [ін.]; наук. ред. В.Є. Снитюк. – Черкаси: видавець Чабаненко Ю, 2015. С. 312.

15. Маций О. Б. Алгоритм решения задачи о взвешенном паросочетании // Інформаційні технології: наука, техніка, технологія, освіта, здоров'я (MicroCAD-2015): матеріали XXIII міжнародної науково-практичної конференції. Харків: НТУ ХПІ, 2015. С. 231.

16. Маций О. Б. Схема поиска гамильтонова маршрута минимальной стоимости на транспортной сети // Інформаційні технології і мехатроніка: освіта, наука та працевлаштування: матеріали міжнародної науково-практичної конференції. Харків: ХНАДУ, 2016. С. 76-78.

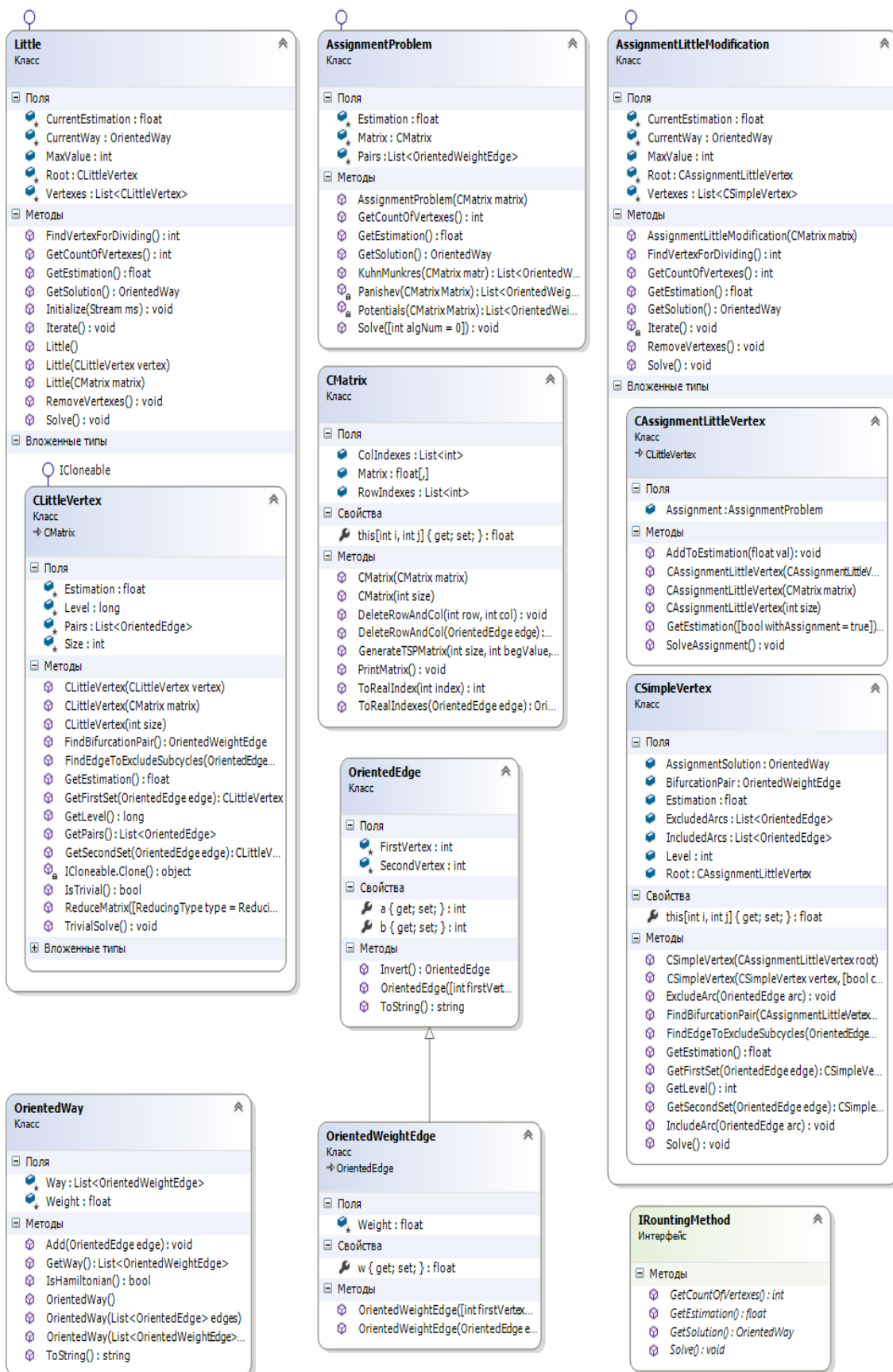
17. Маций О. Б. Поліноміальне перетворення наближених алгоритмів в

рішенні задач типу комівояжера // Інформаційні технології і мехатроніка: освіта, наука та працевлаштування: матеріали міжнародної науково-практичної конференції Харків: ХНАДУ, 2017. С. 54-57.

18. Панишев А. В., Маций О. Б. Базовые задачи оптимизации замкнутых маршрутов и способы усовершенствования методов их решения // Інноваційні технології в науці та освіті. Європейський досвід: матеріали міжнародної конференції. Відень: Издательство, 2017. С. 315-319.

ДОДАТОК Б

ІЄРАРХІЯ ОСНОВНИХ КЛАСІВ ПРОГРАМНОГО ПРОДУКТУ



ДОДАТОК В

ЗАТВЕРДЖУЮ

Проректор ХНАДУ

з наукової роботи

проф. Богомолов В. О.



«19» жовтня 2018 р.

АКТ

про впровадження в навчальний процес результатів дисертаційної роботи

Мацій О. Б.

Комісія в складі голови – завідувача кафедри «Комп'ютерних технологій і мехатроніки» професора Клеца Д. М., та членів комісії – професора Ніконова О. Я. і доцента Подоляки О. О. підтверджує, що в Харківському національному автомобільно-дорожньому університеті на кафедрі Комп'ютерних технологій і мехатроніки використані в навчальному процесі наукові і практичні результати, отримані в кандидатській дисертації Мацій О. Б.

1. Перестановочно-матрична постановка задачі про призначення і рекурентний метод її рішення.
2. Спосіб перетворення i -дерева в гамільтонів цикл для побудови наближеного рішення задачі комівояжера.

Наукові і практичні результати дисертації використовуються в навчальних курсах кафедри «Системний аналіз» та «Математичні методи дослідження операцій»

Голова комісії

Д. М. Клец

Члени комісії

О. Я. Ніконов

О. О. Подоляка

Згідно з оригіналом.

Вчений секретар спецради Д 64.052.02

Підпис
Печатка

Л.В. Колесник

ДОДАТОК Г

ЗАТВЕРДЖУЮ

Комерційний директор
науково-виробничого підприємства
«Карсис»-ТОВ

С.О. Погорелов



2018 р.

АКТ ВПРОВАДЖЕННЯ

результатів дисертаційної роботи Маций Ольги Борисівни за темою
«Математичне моделювання та методи оптимізації замкнених маршрутів в
задачах транспортного типу», представленої на здобуття наукового ступеня
кандидата технічних наук за спеціальністю
01.05.02 «Математичне моделювання і обчислювальні методи».

Комісія у складі:

- | | | |
|----------------|---|---|
| Погорелов С.О. | – | комерційний директор науково-виробничого підприємства «Карсис»-ТОВ |
| Волков В.П. | – | завідувач кафедри технічної експлуатації та сервісу автомобілів Харківського національного автомобільно-дорожнього університету |
| Ніконов О.Я. | – | завідувач кафедри комп'ютерних технологій і мехатроніки Харківського національного автомобільно-дорожнього університету |

підтверджує те, що алгоритми оптимізації і програмні засоби для пошуку розв'язків класу задач маршрутизації на транспортній мережі, які розроблені в результаті виконання дисертаційної роботи «Математичне моделювання та методи оптимізації замкнених маршрутів в задачах транспортного типу» асистента кафедри комп'ютерних технологій і мехатроніки Харківського національного автомобільно-дорожнього університету Ольги Борисівни Маций, впровадженні у розрахункову практику науково-виробничого підприємства «Карсис»-ТОВ і використані при створенні автомобільно-управляючих систем.

С.О. Погорелов

В.П. Волков

О.Я. Ніконов

Згідно з оригіналом.

Вчений секретар спецради Д 64.052.02

Підпис
Печатка

Л.В. Колесник