УДК 519.173

Порівняльний аналіз методів задачі комівояжера для вибору оптимального туристичного маршруту

Рак Людмила Олександрівна

Титаренко Наталія Євгенівна

Мелітопольський державний педагогічний університет ім. Б. Хмельницького

Ключові слова: задача комівояжера, гамільтонів цикл, оптимальний маршрут, міра вигідності, жадібний алгоритм, дерев'яний алгоритм, алгоритм Дейкстри, метод гілок і границь.

Анотація. В статті розглядається розв'язання різними методами задачі про оптимальний туристичний маршрут по містах України. В статті висвітлюється постановка задачі комівояжера та відповідні елементи теорії графів такі, як гамільтонів цикл, вага ребра, міра вигідності тощо. Наведено розв'язання поставленої задачі наступними методами: жадібний алгоритм, дерев'яний алгоритм, алгоритм Дейкстри та метод гілок і границь. Авторами аналізуються методи розв'язання задачі за наступними критеріями: час на розв'язання, кількість ітерацій, зручність і достовірність. Кожний алгоритм має як переваги, так і недоліки, наприклад, час розв'язання залежить від кількості ітерацій алгоритму, що в свою чергу прямо пропорційне кількості вершин графа (міст на маршруті). Автори обгрунтовують послідовність середньої та вищої освіти на прикладі розв'язання однієї задачі все більш складними методами.

Ключевые слова: задача коммивояжера, гамильтонов цикл, оптимальный маршрут, мера выгодности, жадный алгоритм, деревянный алгоритм, алгоритм Дейкстры, метод ветвей и границ.

Аннотация. В статье рассматривается решение различными методами задачи об оптимальном туристическом маршруте по городам Украины. В статье освещается постановка задачи коммивояжера и соответствующие элементы теории графов, такие как гамильтонов цикл, вес ребра, мера выгодности и т.п. Приведены решения поставленной задачи следующими методами: жадный алгоритм, деревянный алгоритм, алгоритм Дейкстры и метод ветвей и границ. Авторами анализируются методы решения задачи по следующим критериям: время на решение, количество итераций, удобство и достоверность. Каждый алгоритм имеет как преимущества, так и недостатки, например, время решения зависит от количества итераций алгоритма, что в свою очередь, прямо пропорционально количеству вершин графа (городов на маршруте). Авторы обосновывают последовательность среднего и высшего образования на примере решения одной задачи все более сложными методами.

Keywords: traveling salesman problem, Hamiltonian cycle, the best route, a measure of profitability, greedy algorithm, wooden algorithm, Dijkstra's algorithm, a method of branches and borders.

Summary. There are different methods of solving the problem of optimal travel route in the cities of Ukraine in the article. In the article statement the traveling salesman problem and relevant elements of graph theory such as Hamiltonian cycle, weight of rib, measure of profitability and others. Also given solutions this problem the following methods: greedy algorithm, wooden algorithm Dijkstra's algorithm and method of branches and borders. The authors analyzed the methods of solving the problem on the following criteria: time to solve, the number of iterations, convenience and reliability. Each algorithm has both advantages and disadvantages, for example, a time of solution depends on the number of iterations of the algorithm, which is directly proportional to the number of vertices (cities on the route). The authors justify the sequence of secondary and higher education on the example of solving a problem more complex methods.

Вступ. Актуальність теми зумовлена тим фактом, що відомі науковці різних країн вже багато років намагаються відшукати найефективніший спосіб розв’язання класичної задачі комівояжера різними методами, такими як жадібний алгоритм, дерев'яний алгоритм, метод гілок і границь, а також методами, пов'язаними з використанням новітніх комп'ютерних технологій. Така активна наукова зацікавленість пояснюється тим, що задача комівояжера має широке практичне застосування і в галузі економіки, і в галузі планування та оптимізації процесів.

Науково-методичною проблемою є пошук алгоритму, який міг би відповісти, чи існує оптимальний маршрут (тобто гамільтонів цикл в довільному графі).

У наукових роботах вітчизняних і зарубіжних вчених накопичений значний досвід розв'язання прикладних задач, таких як задача оптимального розкрою тканини, складання маршруту доставки продукції споживачам.

Дослідження проблематики є актуальним, праці Стенлі Р., Дрозд Ю. А., Оре О., Базилевич Л. Г., Аніщенко Л. М. та інших науковців досить широко розкривають суть проблеми.

Однак переважна більшість евристичних методів дозволяє знайти не найефективніший маршрут, а наближений розв'язок. Тому було поставлено завдання – дослідження відомих методів розв'язання класичної задачі комівояжера та відшукання найефективнішого з них при розв'язанні прикладних задач.

Постановка задачі. Задача комівояжера є однією із найвідоміших і важливих задач транспортної логістики. Комівояжер (бродячий торговець) повинен вийти із заданого (умовно першого) міста, відвідати по одному разу в довільному порядку запропоновані міста і повернутися назад у місто. Суть задачі зводиться до пошуку оптимального, тобто найкоротшого шляху проходження маршруту по запропонованих містах без повторних візитів в будь-яке місто [1]. Мірою вигідності може служити або мінімальний час, витрачений на подорож і продаж, або мінімальні витрачені кошти чи ресурси на шлях, або довжина шляху. Відстані між містами відомі. Порядок відвідування міст довільний, основна умова – оптимальність маршруту.

Для зручності завдання моделі задачі слід скористатися теорією графів, тобто потрібно відшукати гамільтонів цикл або гамільтонів маршрут в графі (маршрут, який містить усі вершини графа по одному разу) [3].

Існує багато алгоритмів для розв'язання задачі комівояжера [2]. Але за результатами застосування алгоритму можна судити, чи є гамільтонів цикл в довільному графі. Однак, однозначного алгоритму, який міг би відповісти, чи є гамільтонів цикл в довільному графі, досі нікому не відомо. Або алгоритм визначає, чи існує в довільному графі гамільтонів цикл, або похибка алгоритму при розвязанні задачі комівояжера може бути нескінченно велика [4].

Щоб встановити точність алгоритмів та всі переваги їх використання при розв'язанні задачі комівояжера розглянемо наступну задачу.

Група туристів з міста Мелітополь вирішила провести екскурсію в п'яти найближчих містах України. Міра вигідності – час, який потрібний для подолання шляху на автомобілі. Необхідно відшукати найвигідніший маршрут, який дозволить туристам об'їхати всі міста і повернутися назад.

Для зручності умову задачі представимо у вигляді графа, в якому вершина 1 – місто Мелітополь, вершини 2,3,4,5,6 – міста екскурсій, вага ребер – час переміщення між містами.

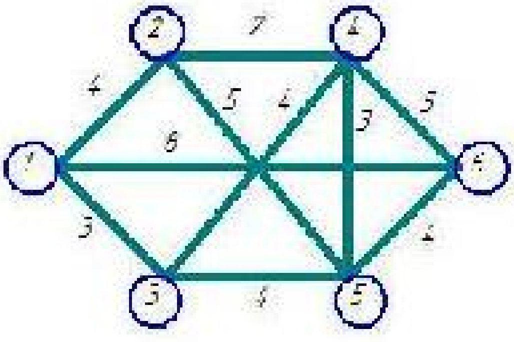

Рис. 2.1. Граф переміщення туристів

Використаємо *жадібний алгоритм*:

1) відсортуємо усі ребра множини E;
2) виберемо найкоротше ребро *e* та вилучимо його з множини E;
3) потрібно, щоб кожне вилучене ребро не утворювало циклу, тоді будемо додавати це ребро до множини T;
4) маршрут, який утворюється за наведеними умовами, представлений на рисунку 2.2:

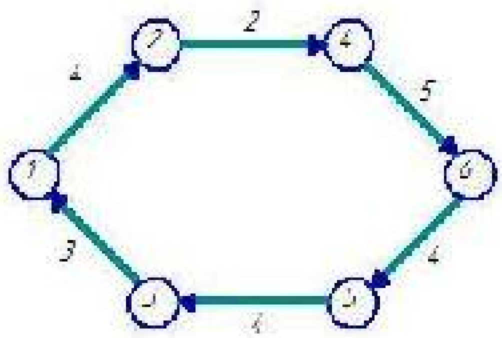

Рис. 2.2. Найкоротший шлях, отриманий «жадібним алгоритмом»

Довжина маршруту L=4+2+5+4+4+3=22 години.

Тобто найефективнішою подорожжю можна вважати запропоновану схему маршруту. В цьому випадку кожне місто туристи відвідають і витратять мінімальний час на переміщення між містами.

Використаємо *дерев'яний алгоритм*:

Для заданого графа побудуємо мінімальне остове дерево, використовуючи запропонований алгоритм:

1) фіксуємо першу вершину – початковий полюс – вершина 1;
2) найближчий мінімальний сусід – вершина 3 (вага ребра дорівнює 3), а також – вершина 2 (вага ребра – 4);
3) обираємо наступні вершини, інцидентні ребра яких з мінімальною вагою і не утворюють цикли;
4) остове дерево буде мати вигляд, зображений на рисунку 2.3.

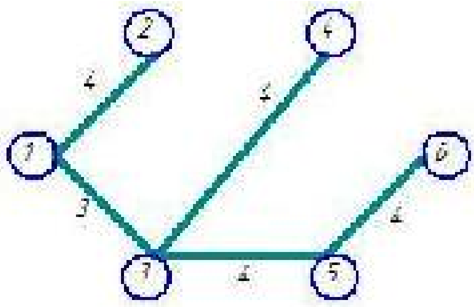

Рис. 2.3. Шлях, отриманий при використанні «дерев'яного алгоритму»

5) обчислюємо сумарну вагу ребер дерева L=3+4+4=11 годин.

Отримане дерево дає змогу обчислити відстань між сусідніми містами, а також найвигідніший маршрут від міста 1 до найвіддаленішого міста 6, транзитом через міста 3 і 5, тому що остове дерево за означенням не може містити циклів.

Використаємо *метод гілок і границь:*

Побудуємо таблицю (матрицю) даних графа. Буквами М позначимо час пересування по одному місту.

	1	**2**	**3**	**4**	**5**	**6**
1	M	4	3	6	7	8
2	4	M	7	2	5	7
3	3	7	M	4	4	8
4	6	2	4	M	3	5
5	7	5	4	3	M	4
6	8	7	8	5	4	M

Ітерація 1.

1) знайти мінімальне значення по рядках і записати в стовпчик d_i.

	1	**2**	**3**	**4**	**5**	**6**	$\mathbf{d_i}$
1	M	4	3	6	7	8	3
2	4	M	7	2	5	7	2
3	3	7	M	4	4	8	3
4	6	2	4	M	3	5	2
5	7	5	4	3	M	4	3
6	8	7	8	5	4	M	4

2) редукція рядків – в кожному рядку відняти його мінімальне значення.

	1	**2**	**3**	**4**	**5**	**6**	**d_i**
1	M	1	0	3	4	5	3
2	2	M	5	0	3	5	2
3	0	4	M	1	1	5	3
4	4	0	2	M	1	3	2
5	4	2	1	0	M	1	3
6	4	3	4	1	0	M	4

3) знайти мінімальне значення по стовпцях і записати в рядок d_j.

	1	**2**	**3**	**4**	**5**	**6**
1	M	1	0	3	4	5
2	2	M	5	0	3	5
3	0	4	M	1	1	5
4	4	0	2	M	1	3
5	4	2	1	0	M	1
6	4	3	4	1	0	M
d_j	0	0	0	0	0	1

4) редукція стовпців

	1	**2**	**3**	**4**	**5**	**6**
1	M	1	0	3	4	4
2	2	M	5	0	3	4
3	0	4	M	1	1	4
4	4	0	2	M	1	2
5	4	2	1	0	M	0
6	4	3	4	1	0	M

5) обчислення оцінок нульових клітин – це сума мінімального значення за рядком і стовпчиком, в яких розміщено нуль.

$O_{13}=1+1=2$, $O_{24}=2+0=2$, $O_{31}=2+1=3$, $O_{42}=1+1=2$, $O_{54}=0+0=0$,

$O_{56}=2+0=2$, $O_{65}=1+1=2$. Всі оцінки занесемо в таблицю

	1	**2**	**3**	**4**	**5**	**6**
1	M	1	0(2)	3	4	4
2	2	M	5	0(2)	3	4
3	0(3)	4	M	1	1	4
4	4	0(2)	2	M	1	2
5	4	2	1	0(0)	M	0(2)
6	4	3	4	1	0(2)	M

6) Редукція матриці: обрати нуль з найбільшою оцінкою , це один з відрізків шляху. Виписати його, викреслити відповідні рядок і стовпець.

	1	**2**	**3**	**4**	**5**	**6**
1	**M**	1	0(2)	3	4	4
2	**2**	M	5	0(2)	3	4
3	**0(3)**	**4**	**M**	**1**	**1**	**4**
4	**4**	0(2)	2	M	1	2
5	**4**	2	1	0(0)	M	0(2)
6	**4**	3	4	1	0(2)	M

Важливо, що при виключенні рядок і стовпець дають один з відрізків оптимального шляху L(3→1)=3. На цьому етапі для того, щоб виключити повернення в місто, яке вже відвідали, потрібно поставити символ M на обернений шлях (1→3).

Ітерація 2.

	2	**3**	**4**	**5**	**6**
1	1	M	3	4	4
2	M	5	0	3	4
4	0	2	M	1	2
5	2	1	0	M	0
6	3	4	1	0	M

1)знайти мінімальне значення по рядках і записати в стовпчик d_i.

редукція рядків – в кожному рядку відняти його мінімальне значення.

	2	**3**	**4**	**5**	**6**	d_i
1	1	M	3	4	4	1
2	M	5	0	3	4	0
4	0	2	M	1	2	0
5	2	1	0	M	0	0
6	3	4	1	0	M	0

	2	**3**	**4**	**5**	**6**	d_i
1	0	M	2	3	3	1
2	M	5	0	3	4	0
4	0	2	M	1	2	0
5	2	1	0	M	0	0
6	3	4	1	0	M	0

2) знайти мінімальне значення по стовпцях і записати в рядок d_j, редукція стовпців

	2	**3**	**4**	**5**	**6**
1	1	M	3	4	4
2	M	5	0	3	4
4	0	2	M	1	2
5	2	1	0	M	0
6	3	4	1	0	M
d_j	0	1	0	0	0

	2	**3**	**4**	**5**	**6**
1	0	M	2	3	3
2	M	4	0	3	4
4	0	1	M	1	2
5	2	0	0	M	0
6	3	3	1	0	M
d_j	0	1	0	0	0

3)обчислення оцінок нульових клітин

$O_{12}=2+0=2$, $O_{24}=3+0=3$, $O_{42}=0+1=1$, $O_{53}=1+0=1$, $O_{54}=0+0=0$, $O_{56}=0+2=2$, $O_{65}=1+1=2$. Всі оцінки занесемо в таблицю.

Редукція матриці: обрати нуль з найбільшою оцінкою , це один з відрізків шляху. Виписати його, викреслити відповідні рядок і стовпець.

	2	**3**	**4**	**5**	**6**
1	0(2)	M	2	3	3
2	M	4	0(3)	3	4
4	0(1)	1	M	1	2
5	2	0(1)	0(0)	M	0(2)
6	3	3	1	0(2)	M

	2	**3**	**4**	**5**	**6**
1	0(2)	M	2	3	3
2	M	4	0(3)	3	4
4	0(1)	1	M	1	2
5	2	0(1)	0(0)	M	0(2)
6	3	3	1	0(2)	M

Довжина відрізку маршруту L(2→4)=3. На цьому етапі для того, щоб виключити повернення в місто, яке вже відвідали, потрібно поставити символ M на обернений шлях (4→2).

Ітерація 3.

	2	**3**	**5**	**6**
1	0	M	3	3
4	M	1	1	2
5	2	0	M	0
6	3	3	0	M

1)знайти мінімальне значення по рядках і записати в стовпчик d_i.

редукція рядків – в кожному рядку відняти його мінімальне значення.

	2	**3**	**5**	**6**	d_i
1	0	M	3	3	0
4	M	1	1	2	1
5	2	0	M	0	0
6	3	3	0	M	0

	2	**3**	**5**	**6**	d_i
1	0	M	3	3	0
4	M	0	0	1	1
5	2	0	M	0	0
6	3	3	0	M	0

2) мінімальне значення по усіх стовпцях дорівнює нулю, редукція стовпців не потрібна.

3)обчислимо оцінки нульових клітин. Всі оцінки занесемо в таблицю. Редукція матриці. Виписати його, викреслити відповідні рядок і стовпець.

	2	**3**	**5**	**6**
1	0(5)	M	3	3
4	M	0(0)	0(0)	1
5	2	0(0)	M	0(1)
6	3	3	0(3)	M

	2	**3**	**5**	**6**
1	0(5)	M	3	3
4	M	0(0)	0(0)	1
5	2	0(0)	M	0(1)
6	3	3	0(3)	M

Довжина відрізку маршруту L(1→2)=5. На цьому етапі вже виключено повернення в місто, яке вже відвідали.

Ітерація 4. Мінімальне значення по рядках і стовпцях дорівнює нулю, редукція рядків і стовпів не потрібна. Обчислимо оцінки нульових клітин. Всі оцінки занесемо в таблицю. Редукція матриці.

	3	**5**	**6**
4	0	0	1
5	0	M	0
6	3	0	M

	3	**5**	**6**
4	0(0)	0(0)	1
5	0(0)	M	0(1)
6	3	0(3)	M

Довжина відрізку маршруту L(6→5)=3. На цьому етапі вже виключено повернення в місто, яке вже відвідали. Потрібно поставити символ M на обернений шлях (5→6).

Ітерація 5.

	3	**6**
4	0	1
5	0	M

	3	**6**
4	0(0)	0(∞)
5	0(∞)	M

Довжина відрізку маршруту L(5→3)=4 години.

Останній оптимальний відрізок шляху 4→6, довжина якого L=5 годин

Таким чином, оптимальний маршрут знайдено 1→2→4→6→5→3→1.

Загальна довжина отриманого маршруту L=5+3+3+4+4+3=22 години.

Використаємо *алгоритм Дейкстри:*

1) присвоїти початкове значення вихідному місту(1) та вважати цю мітку постійною. Для решти вершин – мітки тимчасові.

2) оновити мітки. Довжина ребра до вершини 2 L(2)=0+4=4, вершини 3 L(3)=0+3=3, вершини 6 L(6)=0+8=8. Тепер міткам 2 і 3 присвоюємо постійне значення 4 і 3 відповідно як мінімальним, решта міток – тимчасові.

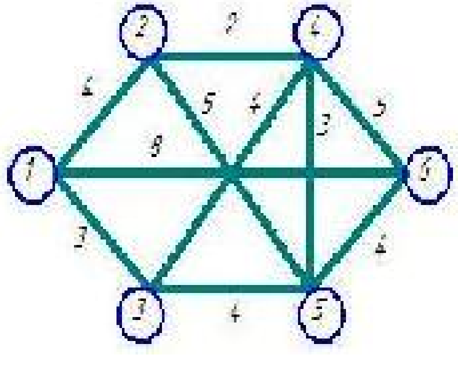

Рис. 2.4. Граф переміщення туристів

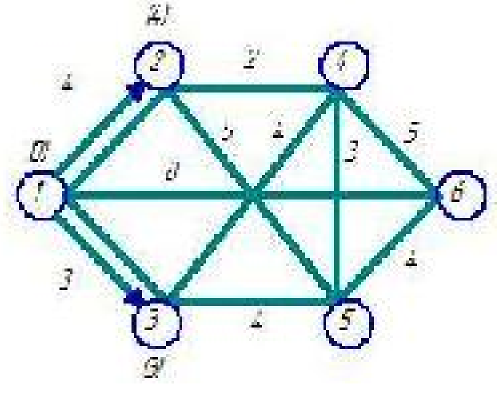

Рис. 2.5. Оновлення міток графа №1

3) оновити мітки. Довжина ребра до вершини 4 L(4)=2+4=6, L(4)=3+4=7 , вершини 6 L(6)=0+8=8, вершини 5 L(5)=4+5=9, L(5)=4+3=7. Тепер мітці 4 присвоюємо постійне значення 6 як мінімальній, решта міток – тимчасові.

4) оновити мітки. Довжина ребра до вершини 6 L(6)=0+8=8, L(6)=6+5=11, вершини 5 L(5)=4+5=9, L(5)=4+3=7, L(5)=6+3=9. Тепер мітці 5 присвоюємо постійне значення 7 як мінімальній, решта міток – тимчасові.

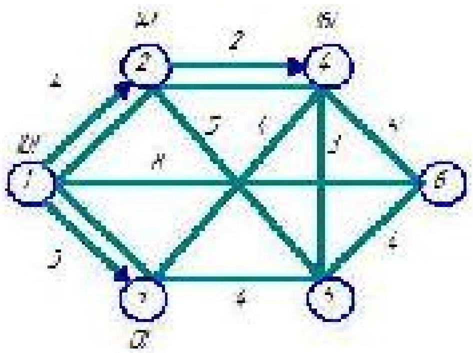

Рис. 2.6. Оновлення міток графа №2

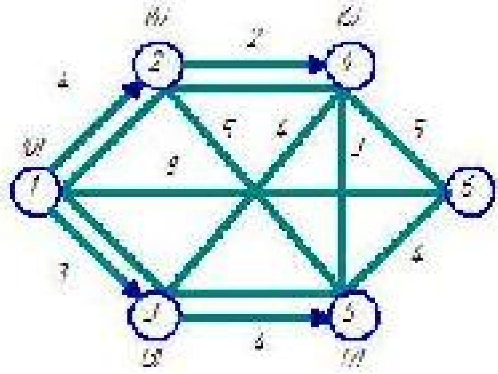

Рис. 2.7. Оновлення міток графа №3

4) оновити мітки. Довжина ребра до вершини 6 L(6)=6+5=11, L(6)=4+7=11. Тепер мітці 6 присвоюємо постійне значення 11 .

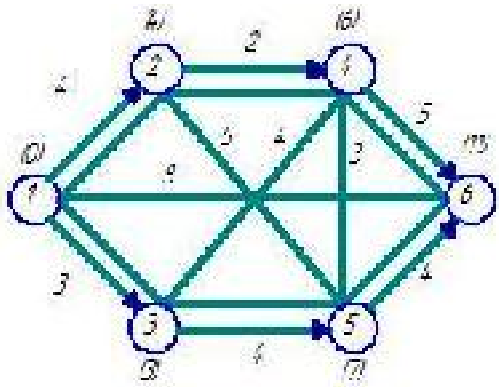

Рис. 2.8. Оновлення міток графа №4

Якщо отриманий маршрут зациклити 1→2→4→6→5→3→1, то довжина цього маршруту буде дорівнювати 22 години.

Аналіз отриманих результатів проведемо за кількома критеріями і запишемо в зведену таблицю.

Таблиця 2.1

Зведена таблиця використання алгоритмів для розв'язання задачі комівояжера

Алгоритм	Витрачений час на розв	Тип алгоритму	Зручність (1-10 балів)	Кількість ітерацій	Достовірність (%)
Жадібний алгоритм	15хв	точний	7	6	75
Дерев'яний алгоритм	20хв	точний	8	3	95
Метод гілок і границь	1год 30хв	точний	9	6	100
Алгоритм Дейкстра	45хв	точний	9	4	99

Висновок. Проаналізувавши результати, можна зробити такі висновки. Кожний алгоритм має як переваги, так і недоліки, наприклад, час розв'язання залежить від кількості ітерацій алгоритму, що в свою чергу прямо пропорційне кількості вершин графа (міст на маршруті), тому метод гілок і границь складно використовувати при збільшенні кількості вершин графа. Проте зручність використання досить висока і достовірність отриманих результатів 100%.

Використовуючи жадібний алгоритм, можна отримати декілька розв'язків задачі комівояжера, серед яких потрібно обрати такий, що найбільш відповідає умовам задачі.

Остове дерево також можна отримати декількох видів, серед яких потрібно обрати оптимально пристосоване до умови задачі. Остове дерево не можна зациклювати, тому циклічний маршрут за цим алгоритмом відшукати неможливо.

Алгоритм Дейкстри використовують для розв'язання задачі комівояжера в тому випадку, коли необхідно обчислити відстань від будь-якої вершини графа до будь-якої іншої вершини графа.

Отже, розвязання задачі комівояжера є чудовим прикладом послідовності сердньої та вищої освіти: для застосування жадібного алгоритму достатньо знань зі шкільної математики; інші методи можна засвоїти після вивчення у ВНЗ таких дисциплін, як лінійна алгебра (теорія матриць), дискретна математика (теорія графів), програмування.

Список використаних джерел

1. Базилевич Л. Г. Дискретна математика у прикладах і задачах / Л. Г. Базилевич. – Львів: Видавець І. Е. Чижиков, 2013. – 487 с.
2. Дрозд Ю. А. Дискретна математика / Ю. А. Дрозд – К.: Освіта, 2004. – 420 с.
3. Оре О. Теорія графів / О.Оре – К.: Освіта, 2004. – 420 с.
4. Стенли Р. Дискретна математика / Р.Стенли – К.: Техніка, 2005. – 768 с.

References

1. Bazylevyč L. H. Dyskretna matematyka u prykladax i zadačax / L. H. Bazylevyč. – L′viv: Vydavec′ I. E. Čyžykov, 2013. – 487 s.
2. Drozd Ju. A. Dyskretna matematyka / Ju. A. Drozd – K.: Osvita, 2004. – 420 s.
3. Ore O. Teorija hrafiv / O.Ore – K.: Osvita, 2004. – 420 s.
4. Stenly R. Dyskretna matematyka / R.Stenly – K.: Texnika, 2005. – 768 s.