

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ВОЛОДИМИРА ДАЛЯ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання курсового проекту з дисципліни
“УПРАВЛІННЯ ЛАНЦЮГОМ ПОСТАЧАНЬ”

(для студентів спеціальності 275.03 «Транспортні технології (на автомобільному транспорті)», 275.02 «Транспортні технології (на залізничному транспорті)»)

ЗАТВЕРДЖЕНО
на засіданні кафедри
«Логістичне управління та безпека
руху на транспорті»
Протокол № 2 від 10.09.2020р.

Севєродонецьк 2020

УДК 658.86

Методичні вказівки до виконання курсового проекту з дисципліни "Управління ланцюгом постачань" (для студентів спеціальності 275.03 «Транспортні технології (на автомобільному транспорті)», 275.02 «Транспортні технології (на залізничному транспорті) /Укл.: Баранов І.О. - Сєверодонецьк: Вид-во СНУ ім. В.Даля, 2020. - 71 с.

В методичних вказівках приведені матеріали, необхідні для виконання курсового проекту студентами денної та заочної форми навчання з дисципліни "Управління ланцюгом постачань", для організації самостійних наукових досліджень студентів, у результаті яких вони можуть здобути навички роботи з науково-технічною літературою, навчитись ставити експеримент, проводити докладні досліді, аналізувати отримані результати, формулювати висновки.

Укладачі:

І.О. Баранов, к.т.н., доц.

Відп. за випуск:

Н.Б. Чернецька-Білецька, д.т.н., проф.

1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Дані методичні вказівки розроблені для надання допомоги студентам у виконанні курсового і дипломного проектування при розгляді управління ланцюгом постачань і при моделюванні транспортних потоків на залізниці.

До задач курсового проектування входить закріплення і поглиблення теоретичних знань, а також одержання додаткових практичних навичок і умінь самостійно вирішувати питання, пов'язані з моделюванням і проектуванням транспортних потоків на залізниці.

Мета, склад і зміст проекту.

В умовах сучасної технічної революції, концентрації, централізації і комбінованого виробництва, між окремими галузями виробництва й усередині кожної з них, величезне значення набувають проблеми удосконалювання організації і управління певними системами, а в зв'язку з цим і комплексним підходом до того або іншого об'єкта дослідження. Переходячи від вибору методу дослідження до формування системотехнічної моделі транспортних комплексів, необхідно прийняти таке з можливих будівель системи, відповідно до якого однією з взаємодіючих підсистем є транспортні потоки на залізниці, іншою підсистемою – постійні пристрої.

Мета курсового проектування – поглибити і закріпити теоретичні знання, а також придбати практичні навички в моделюванні транспортних потоків на залізниці.

Курсовий проект складається з графічної частини і пояснювальної записки (40-50 стор.).

У пояснювальній записці повинні бути розглянуті наступні питання:

Введення.

1. Загальні питання побудови моделі транспортної мережі.
2. Характеристика аналізованої транспортної мережі.
3. Теоретичні положення по організації моделювання транспортних потоків.
4. Рішення задач моделювання транспортних потоків.
5. Пропозиції по удосконалюванню роботи досліджуваної транспортної мережі.

Висновки.

Література.

Структура розрахунково-пояснювальної записки:

1. Титульний лист (див. додаток А).
2. Завдання на курсовий проект (див. додаток Б).
3. Календарний план (див. додаток В).
4. Реферат.
5. Зміст.
6. Вихідні дані.
7. Вступ.
8. Основна частина записки.
9. Висновки.
10. Література.
11. Додатки.

Графічна частина складається з чотирьох аркушів формату А1:

1. Транспортна мережа з методикою визначення найкоротшої відстані між джерелом і стоком (алгоритм Дейкстри) та визначенням мінімального остовного дерева.
2. Рішення задачі про найкоротші відстані (алгоритм Флойда).
3. Рішення задачі про максимальний потік (алгоритм Форда-Фалкерсона).
4. Функціональна оцінка схеми станції

Вихідними даними для курсового проектування є:

1. Карта залізниць України
2. Варіанти транспортної мережі і її вантажопотоки (Додаток Г).

Основна частина курсового проекту виконується на аркушах формату А4 з маленькою рамкою (рис.1.1.).

					ПК.РТ-244.336.ПЗ	Лист
Зм.	Лист	№ докум.	Підпис	Дата		5

Рис.1.1. Маленька рамка

ПАРАМЕТРИ ТЕКСТУ:

Шрифт - Times New Roman, 14 pt;

Міжрядковий інтервал – 1,5;

Параметри сторінки:

верхнє поле – 2 см;

нижнє поле – 2 см;

ліве поле – 3 см;

праве поле – 1,5 см;

абзац – 1,27 см.

Складові коду (ПК.РТ-XXX.XXX.ПЗ) проекту:

ПК – проект курсовий;

РТ-XXX – номер групи;

XXX – останні три цифри залікової книжки;

ПЗ – пояснювальна записка.

Графічна частина курсового проекту повинна містити 3 графічні листи формату А1.

Приклад заповнення рамки на графічних листах (рис. 1.2):

						<i>ПК.РТ-241.XXX.01.СХ</i>			
						<i>Відділення залізниці</i>			
<i>Изм.</i>	<i>Коліч.</i>	<i>Лист</i>	<i>№док.</i>	<i>Подп.</i>	<i>Дата</i>				
<i>Разр</i>		<i>Іванов</i>				<i>Моделювання транспортних мереж</i>	<i>Стадія</i>	<i>Лист</i>	<i>Листов</i>
<i>Перев</i>		<i>Петров</i>						<i>1</i>	<i>3</i>
						<i>Визначення найкоротшої відстані між джерелом і стоком</i>	<i>СНУ ім. В.Даля Кафедра ОПУЗТ</i>		
<i>Затв</i>									

Рис. 1.2. Велика рамка для графічних листів і заголовків розділів у пояснювальній записці

2. ЗАГАЛЬНІ ПИТАННЯ ПОБУДОВИ МОДЕЛІ ТРАНСПОРТНОЇ МЕРЕЖІ НА ЗАЛІЗНИЦІ

Побудова моделі транспортної мережі починається зі складання системи доріг.

Система доріг – це розмічений мультиграф (без петель), що відрізняється від графа тим, що в ньому одна і та ж пара (різних) вершин може бути зв'язана більш ніж одним ребром. При цьому вершини відповідають містам, а ребра – дорогам. Однобічним дорогам відповідають дуги, а двостороннім дорогам – ребра.

Складання системи доріг полягає в розташуванні вершин (міст), відповідно до реального положення їх за картою. Після цього вершини з'єднуємо таким чином, щоб вийшла максимальна кількість непересічних ребер. У реальності це дороги, що з'єднують відповідні міста.

Кожна дорога має деяку *довжину* – позитивне речовинне число. *Довжина шляху в системі доріг* – це сума довжин доріг цього шляху. *Відстань між двома містами* – це довжина мінімального шляху між цими містами.

Методи, застосовувані при дослідженні транспортних процесів у даний час, можна умовно підрозділити на описові і математичні. Описові методи не дають кількісної міри для оцінки досліджуваних процесів. Тому нижче розглядаються в основному математичні методи, що одержали широке поширення в останні роки. У них насамперед потрібно відзначити існування двох напрямків: детерміністського і імовірностного.

У першому випадку транспорт розглядається як своєрідний механізм (конвеєр), а його складові частини – станції, вузли, дільниці, напрямки – як його ланки, причому зв'язки між елементами представляються у виді жорстких аналітичних залежностей. В другому випадку виходять з передумови, що експлуатаційні процеси носять імовірностний, кореляційний, а не однозначно детермінований характер. Часто обидва ці підходи сполучаються у формі, наприклад, середньозважених величин (вага потягів, часи ходу, інтервали тощо) і експлуатаційних констант (коефіцієнти знімання, параметр нагромадження тощо), значення яких устанавлюється на підставі спостережень і імовірнісних представлень про процеси. Формули, розроблені на підставі тільки імовірностного або детерміністського підходу до транспортних процесів, часто не відповідають існуючій системі роботи транспорту.

Моделі і моделювання відомі ще з глибокої стародавності, але у своєму теперішньому виді метод моделювання оформився в результаті зближення галузей сучасного наукового пізнання. Моделі можуть виконувати самі різноманітні функції: описові, виміру, обробки емпіричних даних, пояснювальні, передбачувальні, пізнавальні. Основне призначення системотехнічних моделей транспортних комплексів – одержання якісних характеристик об'єкта кількісними методами і виявлення «вузьких» місць.

Відмінною рисою процедури моделювання є повна відсутність яких-небудь реальних взаємодій між моделлю й оригіналом. У процесі моделювання встановлюється тільки контрольований інформаційний зв'язок об'єкта з моделлю. Наявний запас знань про оригінал переноситься на модель, дослідження якої, у свою чергу, збагачує наші знання про сторони, що нас цікавлять, структури або функції об'єкта. Модель характеризує об'єкт у деякому пізнавальному відношенні, а не об'єкт у цілому сам по собі.

Обов'язковою складовою частиною процедури моделювання на стадії вибору моделі і переносу отриманих моделюванням нових знань на оригінал є абстракція ототожнення, від рівня проведення якої залежить цінність одержуваних при моделюванні результатів. Абстракція ототожнення має верхню і нижню границі, за межами яких застосування методу моделювання втрачає зміст, тому що модель і оригінал як два об'єкти виходять або абсолютно тотожними, або настільки різними, що вивчення поведінки випробовуваної моделі нічого не дає для пізнання реального об'єкта.

Конкретний, змістовний характер такого визначення відрізняється від формального розуміння моделі, що виражається в логіко-математичних термінах (знакові моделі, математичні формули незалежно від виду їх представлення: у виді функціональних зв'язків або систем різного роду рівнянь).

При виконанні даного курсового проекту студент виходить із вихідних даних, що видає керівник проекту, тобто із знаходження міст, що указані в завданні та побудови транспортної мережі залізничного транспорту.

3. ЗАДАЧА ПОШУКУ НАЙКОРОТШОГО ШЛЯХУ (НАЙМЕНШОЇ ДОВЖИНИ).

Задачу пошуку найкоротшого шляху між джерелом і стоком (початковий і кінцевий пункти мережі) можна вирішити за допомогою алгоритму *Дейкстри*. Алгоритм *Дейкстри* розроблений для знаходження найкоротшого шляху між заданим вихідним вузлом і будь-яким іншим вузлом мережі.

У процесі виконання цього алгоритму при переході від вузла i до наступного вузла j використовується спеціальна процедура позначки ребер. Позначимо через u_i найкоротшу відстань від вихідного вузла 1 до вузла i , через d_{ij} – довжину ребра (i, j) . Тоді для вузла j визначимо мітку $[u_j, i]$ в такий спосіб:

$$[u_j, i] = [u_i + d_{ij}, i], \quad d_{ij} \geq 0$$

Мітки вузлів в алгоритмі Дейкстри можуть бути двох типів: *тимчасові* і *постійні*. Тимчасова мітка згодом може бути замінена на іншу тимчасову, якщо буде знайдений *більш короткий шлях* до даного вузла. Коли ж стане очевидним, що не існує більш короткого шляху від вихідного вузла до даного, статус тимчасової мітки змінюється на постійний.

Розрахункова схема алгоритму складається з наступних кроків.

Крок 0. Вихідному вузлу (вузол 1) привласнюється мітка $[0, -]$. Думаємо $i = 1$.

Крок i . а) Обчислюються тимчасові мітки $[u_i + d_{ij}, i]$ для усіх вузлів j , які можна досягти безпосередньо з вузла i , і які не мають постійних міток. Якщо вузол j уже має мітку $[u_j, k]$, отриману від іншого вузла k , і якщо $u_i + d_{ij} < u_j$, тоді мітка $[u_j, k]$ замінюється на $[u_i + d_{ij}, i]$.

б) Якщо усі вузли мають постійні мітки, процес обчислень закінчується. У противному випадку вибирається мітка $[u_r, s]$ з найменшим значенням відстані u_r серед усіх тимчасових міток (якщо таких міток декілька, то вибір довільний). Думаємо $i = r$ і повторюємо крок i .

Задачу пошуку шляху найменшої довжини можна алгоритмізувати та вирішити у будь-якому програмному продукті (це дозволяється при виконанні курсового проекту).

Приклад рішення задачі пошуку шляху найменшої довжини курсового проекту. Задача вирішується за допомогою алгоритму Дейкстри. Наприклад, задані такі вузли: Львів (8), Рівне (7), Хмельницький (4), Київ пас (14), Кіровоград (27), Херсон (24), Джанкой (25), Запоріжжя (19), Дніпропетровськ (29), Мерефа (32), Харків (33), Луганськ (18). Довжина між вузлами

знайдено за допомогою карти залізничних доріг України. Вузли нанесені на схему та з'єднані ребрами (рис. 3.1). Необхідно знайти найкоротші відстані від міста Львів (вузол 8) до всіх інших одинадцяти міст.

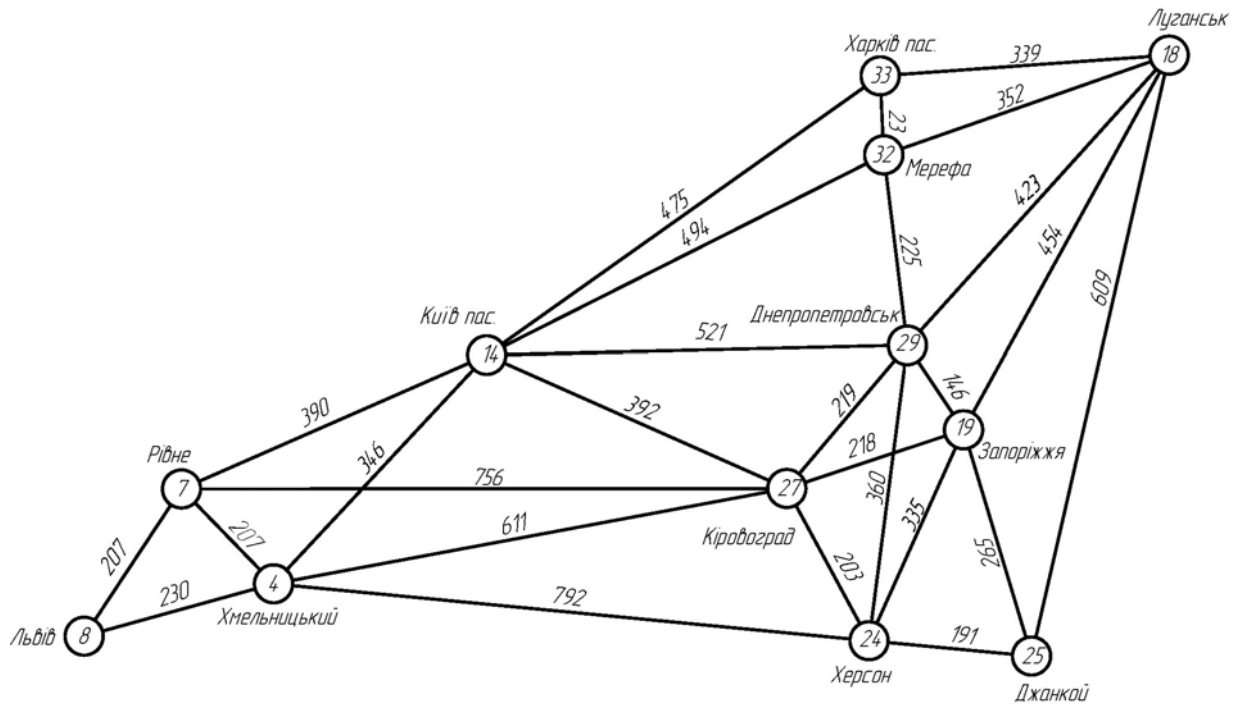


Рис. 3.1. Транспортна мережа

Крок 0. Призначаємо вузлові 8 постійну мітку $[0, -]$.

Крок 1. З вузла 8 можна досягти вузлів 7 і 4. Обчислюємо мітки для цих вузлів, у результаті одержуємо наступну таблицю міток:

Вузол	Мітка	Статус мітки
8	$[0, -]$	Постійна
7	$[0 + 207, 8] = [207, 8]$	Тимчасова
4	$[0 + 230, 8] = [230, 8]$	Тимчасова

Серед вузлів 4 і 7, вузол 7 має найменше значення відстані ($u_7 = 207$). Тому статус мітки цього вузла змінюється на «постійна».

Крок 2. З вузла 7 (останнього вузла з постійною міткою) можна потрапити у вузли 4, 14 і 27. Одержуємо наступний список вузлів:

Вузол	Мітка	Статус мітки
8	$[0, -]$	Постійна
7	$[207, 8]$	Постійна
4	$[230, 8]$	<-Тимчасова
4	$[207 + 207, 7] = [414, 7]$	Тимчасова
14	$[207 + 390, 7] = [597, 7]$	Тимчасова
27	$[207 + 756, 7] = [963, 7]$	Тимчасова

Тимчасовий статус мітки [230, 8] вузла 4 замінюється на постійний ($u_4 = 230$).

Крок 3. З вузла 4 можна досягти вузлів 14, 24 і 27. Після обчислення міток одержимо наступний їх список:

Вузол	Мітка	Статус мітки
8	[0, -]	Постійна
7	[207, 8]	Постійна
4	[230, 8]	Постійна
14	[597, 7]	Тимчасова
14	[230 + 346, 4] = [576, 4]	<-Тимчасова
27	[963, 7]	Тимчасова
27	[230 + 611, 4] = [841, 4]	Тимчасова
24	[230 + 792, 4] = [1022, 4]	Тимчасова

Тимчасова мітка [597, 7], отримана вузлом 7 на другому кроці, змінена на [576, 4]. Це вказує на те, що знайдено більш короткий шлях до цього вузла (що проходить через вузол 4).

Крок 4. З вузла 14 можна досягти вузлів 27, 29, 32 і 33. Після обчислення міток одержимо наступний їх список:

Вузол	Мітка	Статус мітки
8	[0, -]	Постійна
7	[207, 8]	Постійна
4	[230, 8]	Постійна
14	[576, 4]	Постійна
27	[841, 4]	<-Тимчасова
27	[576 + 392, 14] = [968, 14]	Тимчасова
24	[1022, 4]	Тимчасова
29	[576 + 521, 14] = [1097, 14]	Тимчасова
32	[576 + 494, 14] = [1070, 14]	Тимчасова
33	[576 + 475, 14] = [1051, 14]	Тимчасова

Тимчасовий статус мітки [841, 4] вузла 27 замінюється на постійний ($u_{27} = 841$).

Крок 5. З вузла 27 можна досягти вузлів 24, 19 і 29. Після обчислення міток одержимо наступний їх список:

Вузол	Мітка	Статус мітки
8	[0,-]	Постійна
7	[207, 8]	Постійна
4	[230,8]	Постійна
14	[576, 4]	Постійна
27	[841, 4]	Постійна
24	[1022, 4]	<-Тимчасова
24	[841+ 203, 27] = [1044, 27]	Тимчасова
29	[1097, 14]	Тимчасова
29	[841+ 219, 27] = [1060, 27]	Тимчасова
32	[1070, 14]	Тимчасова
33	[1051, 14]	Тимчасова
19	[841+ 218, 27] = [1059, 27]	Тимчасова

Тимчасовий статус мітки [1022, 4] вузла 24 замінюється на постійний ($u_{24} = 1022$).

Тимчасова мітка [1097, 14], отримана вузлом 29 на четвертому кроці, змінена на [1060, 27]. Це вказує на те, що знайдено більш короткий шлях до цього вузла (що проходить через вузол 27).

Крок 6. З вузла 24 можна досягти вузлів 19, 25 і 29. Після обчислення міток одержимо наступний їх список:

Вузол	Мітка	Статус мітки
8	[0,-]	Постійна
7	[207, 8]	Постійна
4	[230,8]	Постійна
14	[576, 4]	Постійна
27	[841, 4]	Постійна
24	[1022, 4]	Постійна
29	[1060, 27]	<-Тимчасова
29	[1022 + 360, 24] = [1382, 24]	Тимчасова
19	[1059, 27]	Тимчасова
19	[1022 + 335, 24] = [1357, 24]	Тимчасова
32	[1070, 14]	Тимчасова
33	[1051, 14]	Тимчасова
25	[1022 + 191, 24] = [1213, 24]	Тимчасова

Тимчасовий статус мітки [1060, 27] вузла 29 замінюється на постійний ($u_{29} = 1060$).

Крок 7. З вузла 29 можна досягти вузлів 18, 19 і 32. Після обчислення міток одержимо наступний їх список:

Вузол	Мітка	Статус мітки
8	[0,-]	Постійна
7	[207, 8]	Постійна
4	[230,8]	Постійна
14	[576, 4]	Постійна
27	[841, 4]	Постійна
24	[1022, 4]	Постійна
29	[1060, 27]	Постійна
19	[1060 + 146, 29] = [1206, 29]	Тимчасова
19	[1059, 27]	<-Тимчасова
32	[1070, 14]	Тимчасова
32	[1060 + 225, 29] = [1285, 29]	Тимчасова
33	[1051, 14]	Тимчасова
25	[1213, 24]	Тимчасова
18	[1060 + 423, 29] = [1483, 29]	Тимчасова

Тимчасовий статус мітки [1059, 27] вузла 19 замінюється на постійний ($u_{29} = 1059$).

Крок 8. З вузла 19 можна досягти вузлів 25 і 18. Після обчислення міток одержимо наступний їх список:

Вузол	Мітка	Статус мітки
8	[0,-]	Постійна
7	[207, 8]	Постійна
4	[230,8]	Постійна
14	[576, 4]	Постійна
27	[841, 4]	Постійна
24	[1022, 4]	Постійна
29	[1060, 27]	Постійна
19	[1059, 27]	Постійна
32	[1070, 14]	Тимчасова
33	[1051, 14]	Тимчасова
25	[1213, 24]	<-Тимчасова
25	[1059 + 265, 19] = [1324, 19]	Тимчасова
18	[1483, 29]	Тимчасова
18	[1059 + 454, 19] = [1513, 19]	Тимчасова

Тимчасовий статус мітки [1213, 24] вузла 25 замінюється на постійний ($u_{24} = 1213$).

Крок 9. З вузла 25 можна досягти вузла 18. Після обчислення міток одержимо наступний їх список:

Вузол	Мітка	Статус мітки
8	[0,-]	Постійна
7	[207, 8]	Постійна
4	[230,8]	Постійна
14	[576, 4]	Постійна
27	[841, 4]	Постійна
24	[1022, 4]	Постійна
29	[1060, 27]	Постійна
19	[1059, 27]	Постійна
25	[1213, 24]	Постійна
32	[1070, 14]	<-Тимчасова
33	[1051, 14]	Тимчасова
18	[1483, 29]	Тимчасова
18	[1213 + 609, 25] = [1822, 25]	Тимчасова

Тимчасовий статус мітки [1070, 14] вузла 32 замінюється на постійний ($u_{14} = 1070$).

Крок 10. З вузла 32 можна досягти вузлів 33 і 18. Після обчислення міток одержимо наступний їх список:

Вузол	Мітка	Статус мітки
8	[0,-]	Постійна
7	[207, 8]	Постійна
4	[230,8]	Постійна
14	[576, 4]	Постійна
27	[841, 4]	Постійна
24	[1022, 4]	Постійна
29	[1060, 27]	Постійна
19	[1059, 27]	Постійна
25	[1213, 24]	Постійна
32	[1070, 14]	Постійна
33	[1051, 14]	<-Тимчасова
33	[1070 + 23, 32] = [1093, 32]	Тимчасова
18	[1483, 29]	Тимчасова
18	[1070 + 352, 32] = [1422, 32]	Тимчасова

Тимчасовий статус мітки [1051, 14] вузла 33 замінюється на постійний ($u_{33} = 1051$).

Крок 11. З вузла 33 можна досягти вузла 18. Після обчислення міток одержимо наступний їх список:

Вузол	Мітка	Статус мітки
8	[0,-]	Постійна
7	[207, 8]	Постійна
4	[230,8]	Постійна
14	[576, 4]	Постійна
27	[841, 4]	Постійна
24	[1022, 4]	Постійна
29	[1060, 27]	Постійна
19	[1059, 27]	Постійна
25	[1213, 24]	Постійна
32	[1070, 14]	Постійна
33	[1051, 14]	Постійна
18	[1422, 32]	Тимчасова
18	[1051+ 339, 33] = [1390, 33]	Тимчасова

Тимчасова мітка [1422, 32], отримана вузлом 18 на десятому кроці, змінена на [1390, 33]. Це вказує на те, що знайдено більш короткий шлях до цього вузла (що проходить через вузол 33).

Кінцевий результат міток має такий вигляд:

Вузол	Мітка	Статус мітки
8	[0,-]	Постійна
7	[207, 8]	Постійна
4	[230,8]	Постійна
14	[576, 4]	Постійна
27	[841, 4]	Постійна
24	[1022, 4]	Постійна
29	[1060, 27]	Постійна
19	[1059, 27]	Постійна
25	[1213, 24]	Постійна
32	[1070, 14]	Постійна
33	[1051, 14]	Постійна
18	[1390, 33]	Постійна

Найкоротший шлях між вузлом 8 і будь-яким іншим вузлом визначається починаючи з вузла призначення шляхом проходження їх у зворотному напрямку за допомогою інформації, представленої в постійних мітках. Найкоротший маршрут між вузлами 8 і 18 має таку послідовність вузлів:

$$(18) \Rightarrow [1390, 33] \Rightarrow (33) \Rightarrow [1051, 14] \Rightarrow (14) \Rightarrow [576, 4] \Rightarrow (4) \Rightarrow [230, 8] \Rightarrow (8)$$

Таким чином, одержуємо шлях загальною довжиною 1390 км.

4. ЗАДАЧА ПОБУДОВИ ОСТОВНОГО ДЕРЕВА.

Граф без циклів називається ациклічним або *лісом*. Зв'язний ациклічний граф називається *деревом*. Якщо G – ліс, то кожен його компонент є деревом. *Остов графа* – це дерево, яке є остовним підграфом. *Остовний підграф* – це підграф, множина вершин якого збігається із множиною вершин графа.

Кожен зв'язний граф має остов. Зворотно, якщо граф має остов, то він - зв'язний.

Якщо граф має цикли, то, послідовно видаляючи ребра із циклів до одержання ациклічного підграфа, одержимо остов. Причому, видалення ребра із циклу графа не порушує його зв'язності, тому що такі ребра не є мостами.

Ребро x та вершина v називаються *покриваючими один одного*, якщо вони інцидентні. Множина вершин, що покривають всі ребра, називається *верховим покриттям графа*. Множина ребер, що покривають всі вершини, називається *реберним покриттям графа*. Становить інтерес найменші по числу вершин і ребер покриття. Якщо ребрам графа приписані ваги, то можна становити питання про реберне покриття, що має найменшу сумарну вагу.

Постановка задачі.

Нехай $G = (V, X)$ – зв'язний зважений граф. Позначимо $c_{ij} = c(v_i, v_j)$ вагу ребра $\{v_i, v_j\}$. Потрібно знайти остов, що має найменшу суму ваг ребер.

Алгоритм рішення.

1. Включити в остов одну довільну вершину w .
2. Розглянути всі вершини v_j , що ще *не належать* остову. Якщо вершина v_j *не суміжна* вершині w , що належить остову, то вона одержує мітку $[0; M]$, де M – нескінченно велике число. Якщо вершина v_j має *суміжну* вершину w , що належить остову, то вершина одержує мітку $[\alpha; \beta]$, де α – номер вершини w ; $\beta = c(w, v_j)$.
3. Із всіх вершин, що *не належать* остову, вибираємо вершину V^* із *мінімальним* значенням β і включаємо її в остов разом з ребром (w, V^*) , де w – вершина, номер якої збігається зі значенням мітки α вершини V^* .
4. Якщо число вершин, що належать кістяку, дорівнює числу вершин графа, то найкоротший остов знайдений, задача вирішена. Інакше – перехід до кроку 5.
5. Коректуємо мітки вершин v_j , які ще не належать остову й суміжні вершині V^* :

$$\text{якщо } \beta_j > c(V^*, v_j),$$

$$\text{то } \beta_j = c(V^*, v_j)$$

$$\alpha_j = V^*,$$

якщо $\beta_j \leq c(V^*, v_j)$, то мітка вершини не змінюється.

Після закінчення позначення – перехід до пункту 3.

Приклад. рішення задачі знаходження мінімального остовного дерева курсового проекту.

Крок 1. Включити в остов одну довільну вершину. Виберемо вершину 8.

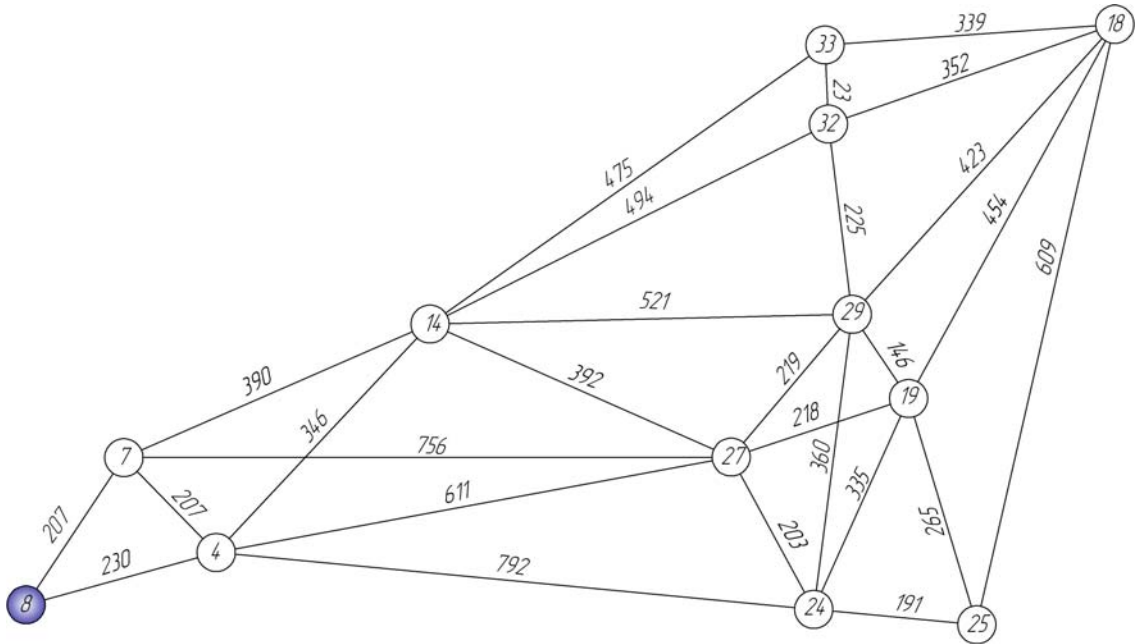


Рис. 4.1. Крок 1.

Крок 2. Розглянемо всі інші вершини v_j , які не належать остову. Якщо вершина v_j не суміжна з вершиною, що належить остову, то вона одержує мітку $[0;M]$, де M – нескінченно велике число.

Якщо вершина v_j – суміжна з вершиною, що належить остову, то v_j одержує мітку $[\alpha; \beta]$, де α – № вершини, що належить остову, а β – вага ребра, що з'єднує v_j та цю вершину.

Таким чином, вершина 7 одержує мітку $[8;207]$, вершина 4 одержує мітку $[8;230]$. Всі інші вершини одержують мітку $[0;M]$.

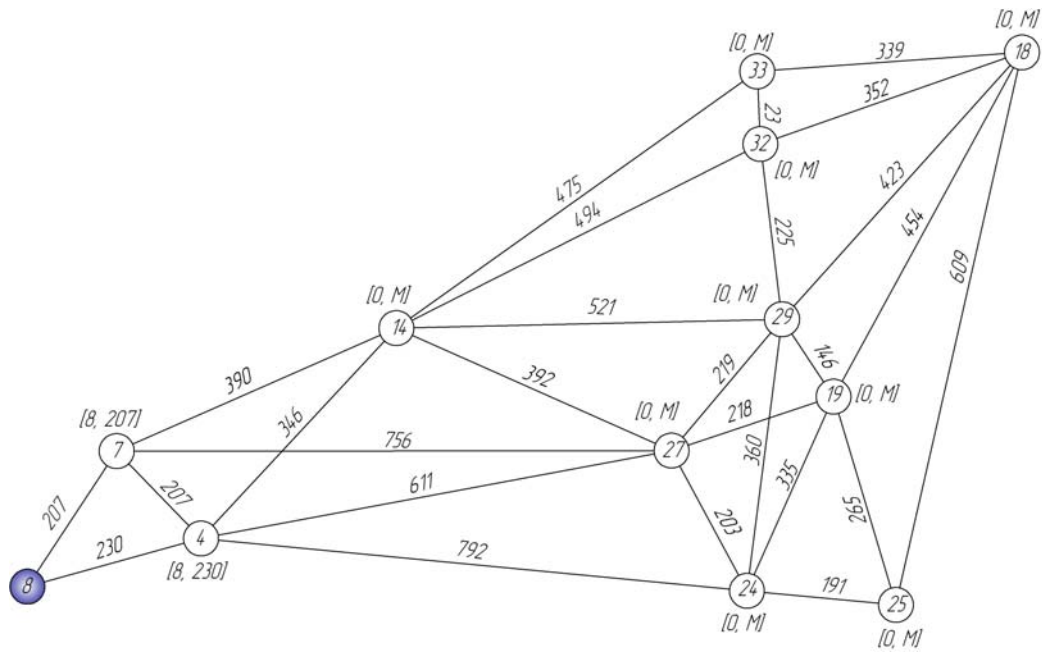


Рис. 4.2. Крок 2.

Крок 3. Серед всіх вершин, які ще не належать остову, вибираємо вершину V^* з мінімальним значенням мітки β та включаємо її в остов разом з ребром, що з'єднує цю вершину з вершиною, що має номер, який дорівнює мітці α вершини V^* .

У нашому випадку в остов повинні бути включені вершина 7 і ребро 7-8.

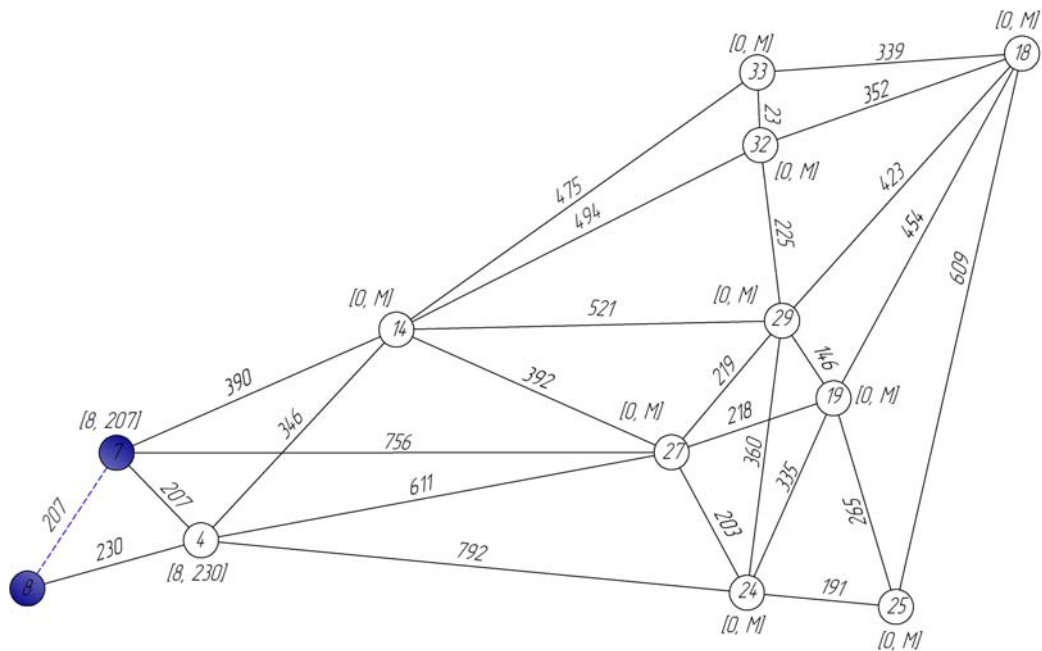


Рис. 4.3. Крок 3.

Крок 4. Якщо число вершин, що належать остову, дорівнює числу вершин графа, то найкоротший остов знайдений і задача вирішена. Інакше – продовжуємо рішення.

Тому що у кістяк входять поки тільки 2 вершини, продовжуємо рішення.

Крок 5. Коректуємо мітки вершин v_j , які ще не належать остову та суміжні вершині V^* :

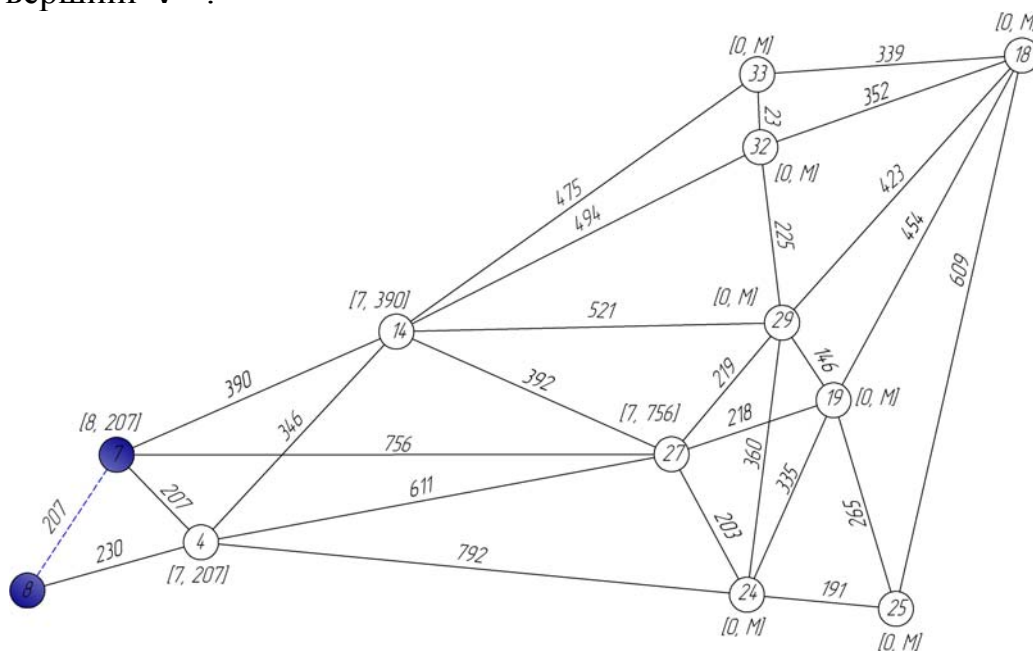


Рис. 4.4. Крок 5.

Якщо $\beta_j > c(V^*, v_j)$, то $\beta_j = c(V^*, v_j)$, $\alpha_j = V^*$. Якщо $\beta_j \leq c(V^*, v_j)$, то мітки не змінюються.

Вершини 4, 14 та 27 є суміжними для вершини 7. Коректуємо мітки, як показано на рис. 4.4.

Крок 3.2. В остов повинні бути включені вершина 4 і ребро 7-4. Коректуємо мітки, як показано на рис. 4.5

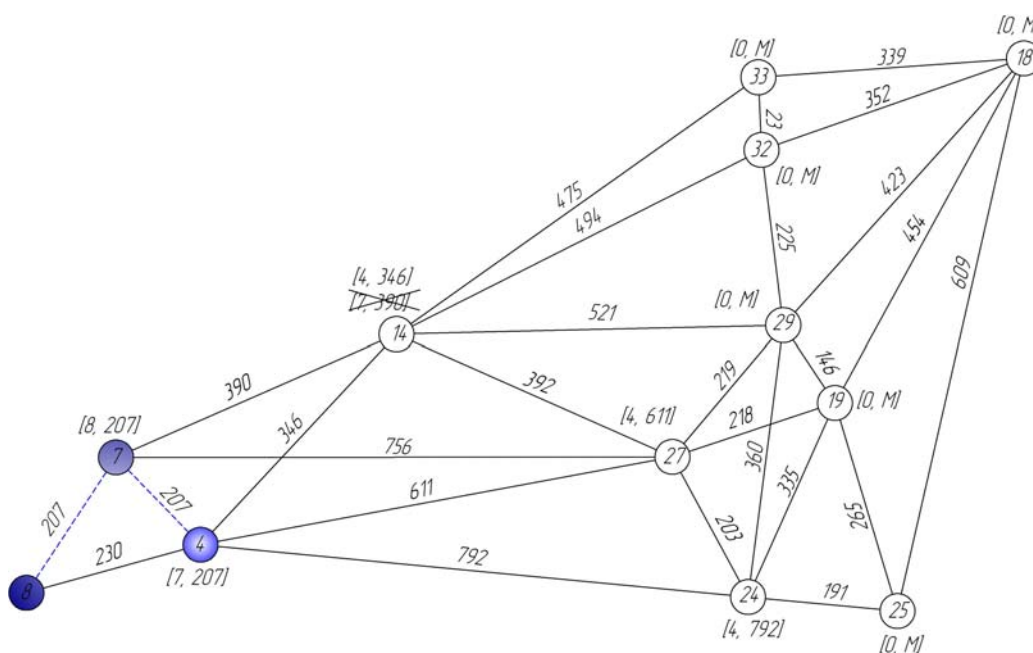


Рис. 4.5. Крок 3.2.

Крок 3.3. В остов повинні бути включені вершина 14 і ребро 4-14. Коректуємо мітки, як показано на рис. 4.6

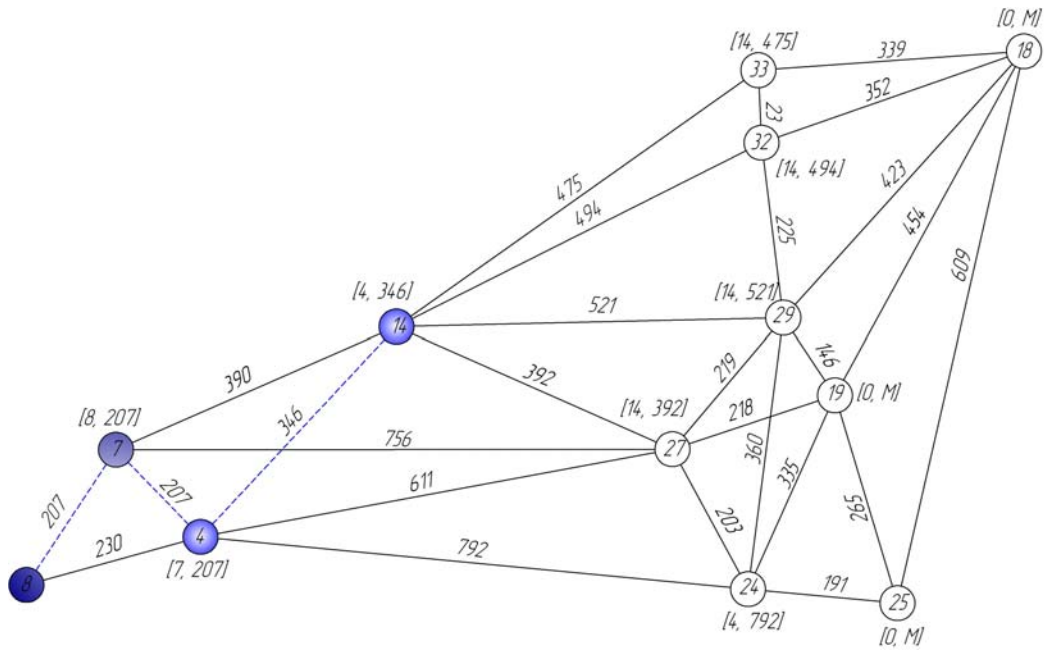


Рис. 4.6. Крок 3.3.

Крок 3.4. В остов повинні бути включені вершина 27 і ребро 27-14. Коректуємо мітки, як показано на рис. 4.7

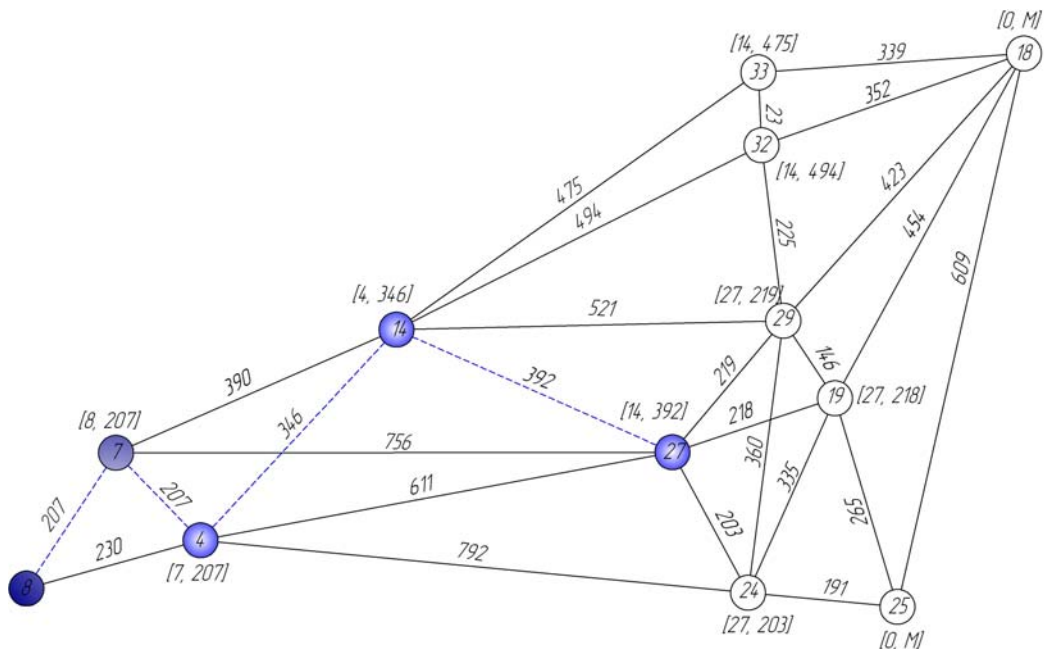


Рис. 4.7. Крок 3.4.

Крок 3.5. В остов повинні бути включені вершина 24 і ребро 24-27. Коректуємо мітки, як показано на рис. 4.8. Але мітка коректується лише для вершини 25, тому що для вершин 19 та 29 не виконується умова $\beta_2 > c(V^*, v_2)$.

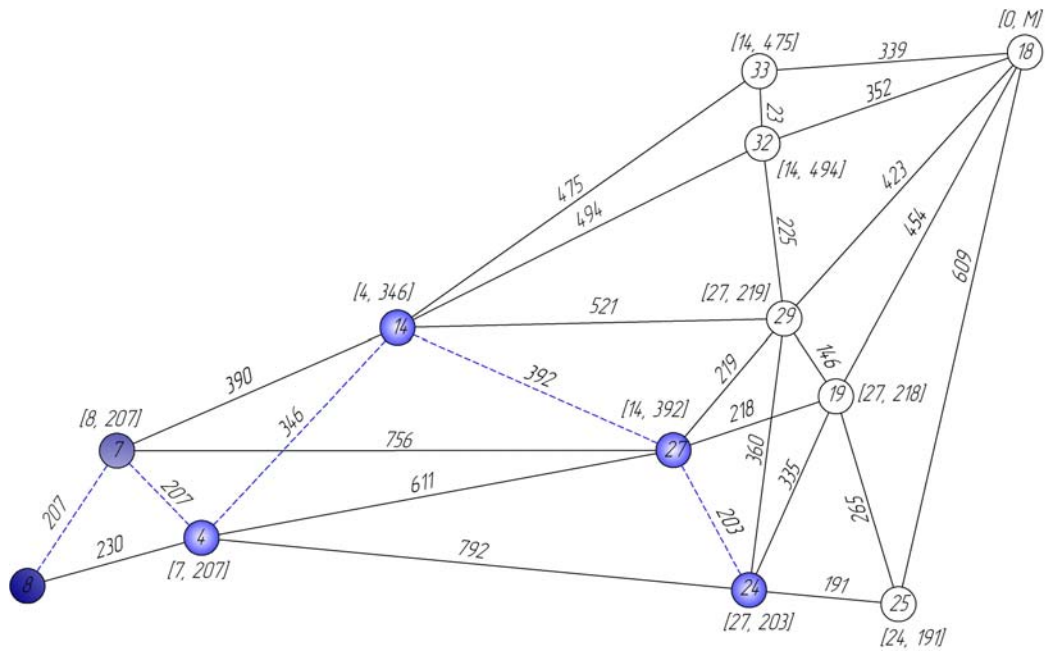


Рис. 4.8. Крок 3.5.

Крок 3.6. В остов повинні бути включені вершина 25 і ребро 24-25. Коректуємо мітки, як показано на рис. 4.9. Але мітка коректується лише для вершини 18, тому що для вершини 19 не виконується умова $\beta_2 > c(V^*, v_2)$.

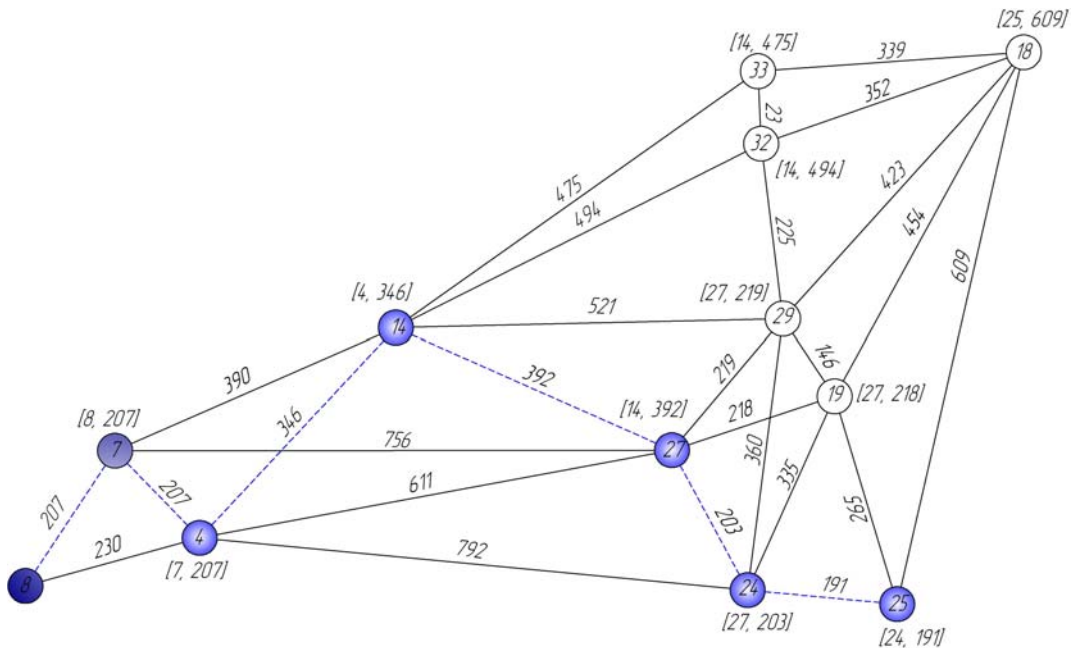


Рис. 4.9. Крок 3.6.

Крок 3.7. В остов повинні бути включені вершина 19 і ребро 19-27. Коректуємо мітки, як показано на рис. 4.10.

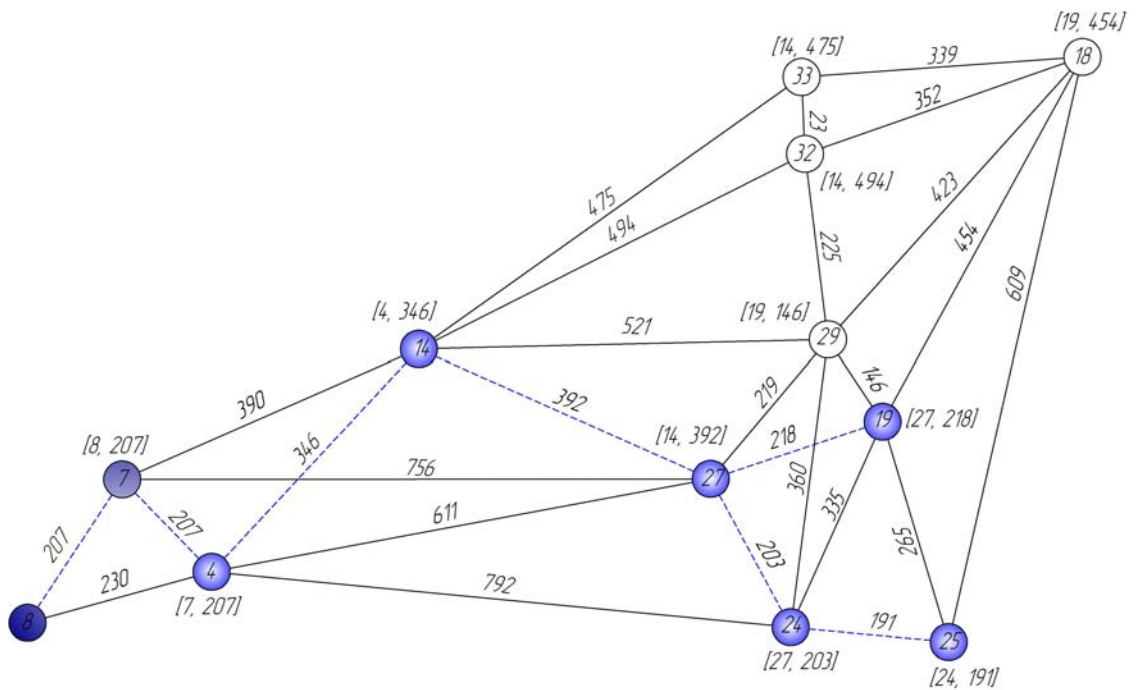


Рис. 4.10. Крок 3.7.

Крок 3.8. В остов повинні бути включені вершина 29 і ребро 29-19. Коректуємо мітки, як показано на рис. 4.11.

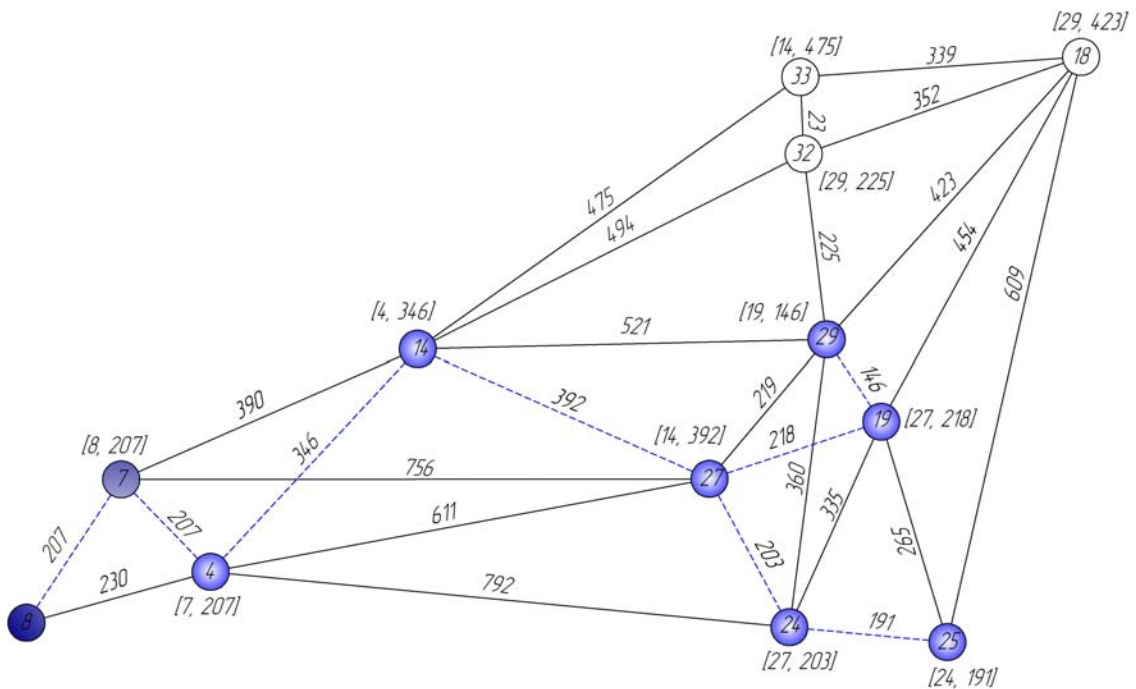


Рис. 4.11. Крок 3.8.

Крок 3.9. В остов повинні бути включені вершина 32 і ребро 29-32. Коректуємо мітки, як показано на рис. 4.12.

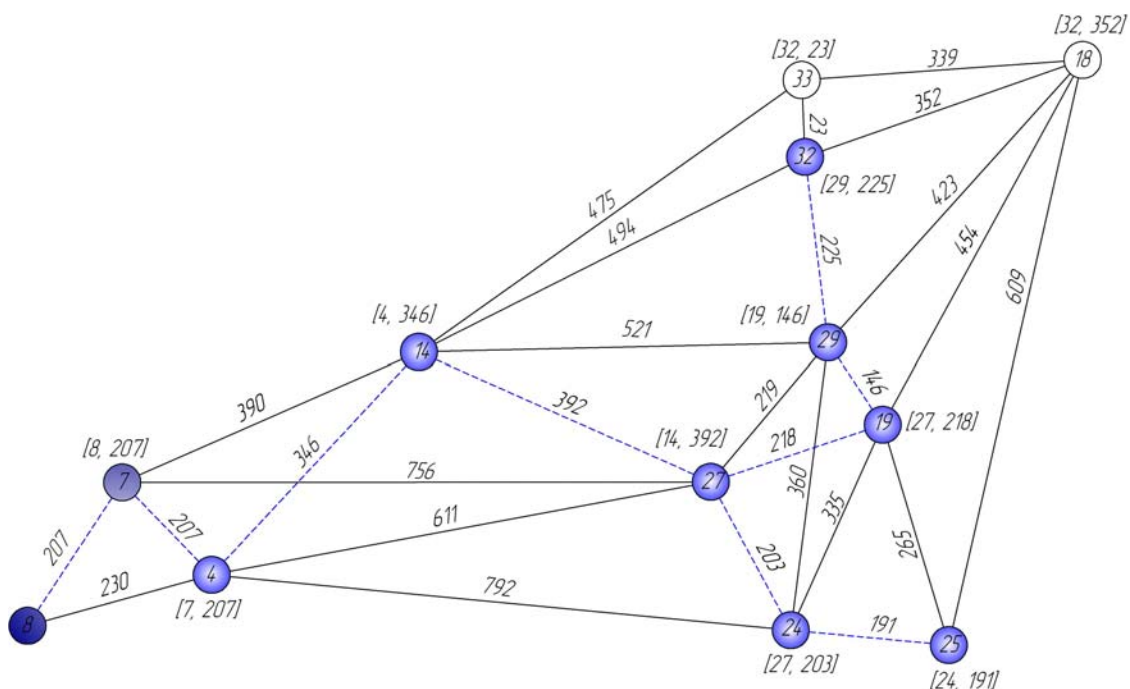


Рис. 4.12. Крок 3.9.

Крок 3.10. В остов повинні бути включені вершина 33 і ребро 32-33. Коректуємо мітки, як показано на рис. 4.13.

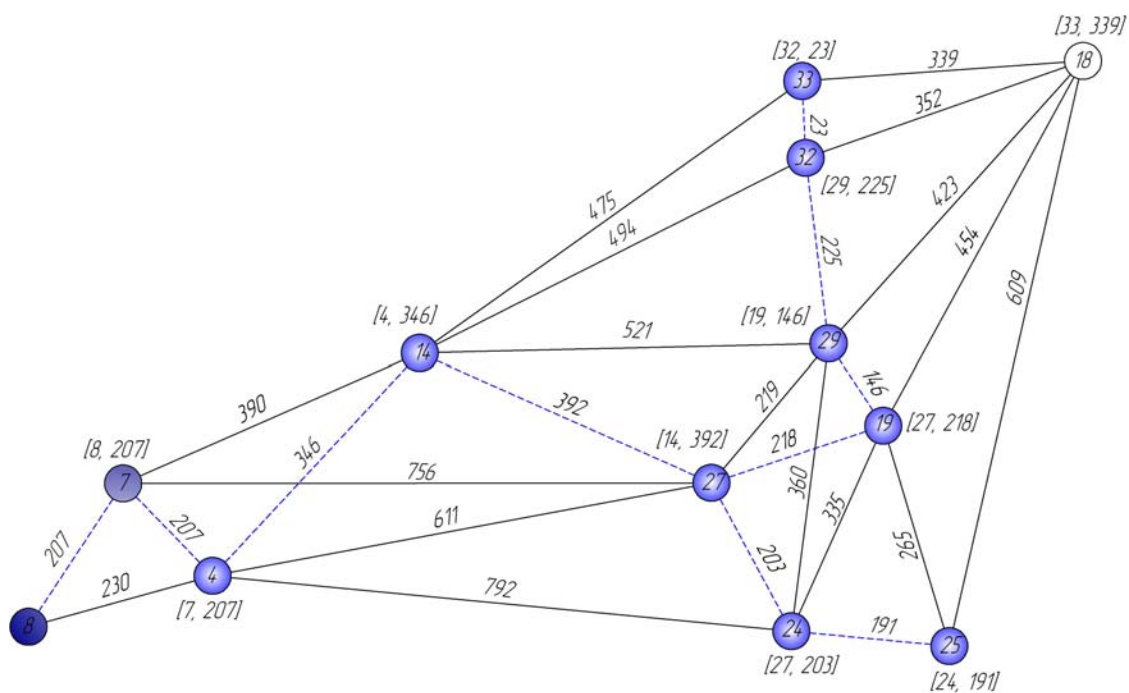


Рис. 4.13. Крок 3.10.

Результат. Продовжуючи рішення відповідно до алгоритму, одержуємо наступний найкоротший остов (рис. 4.14).

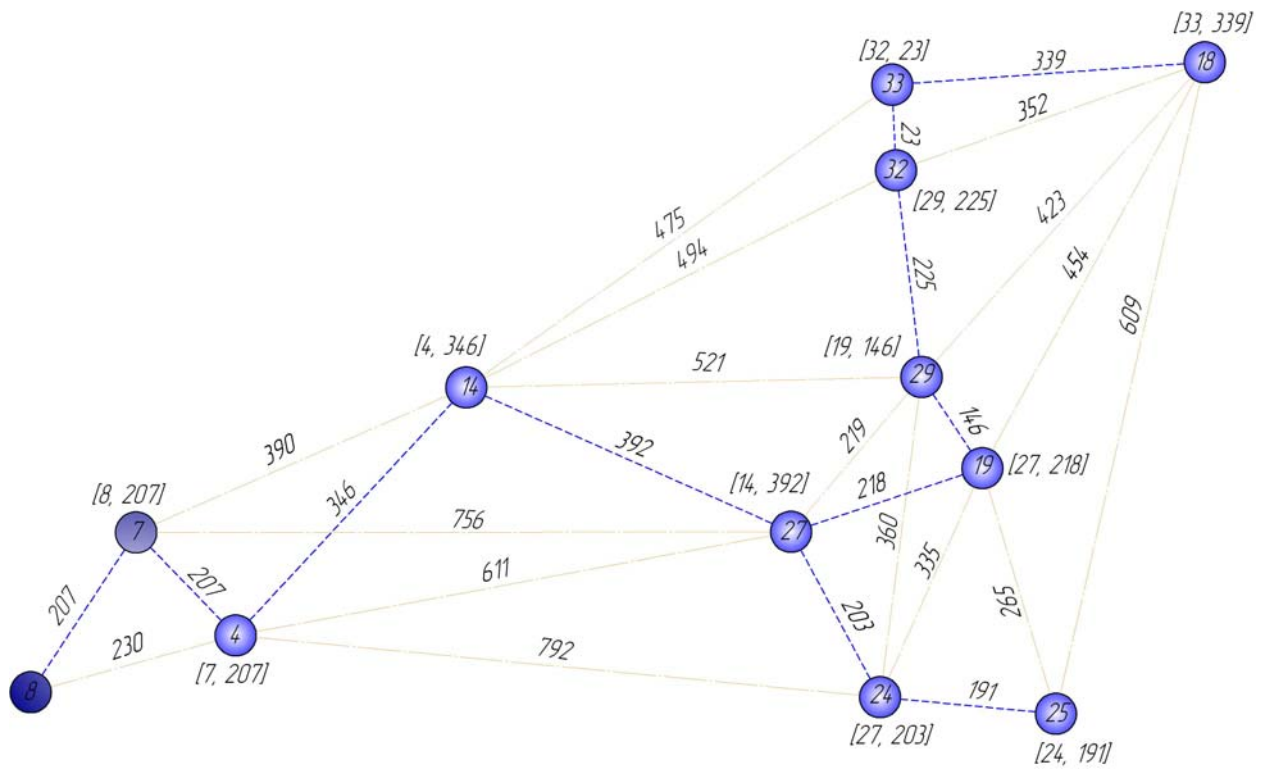


Рис. 4.14. Мінімальне остовне дерево

5. ЗАДАЧА ВИЗНАЧЕННЯ НАЙКОРОТШИХ ВІДСТАНЕЙ МІЖ ЕЛЕМЕНТАМИ ТРАНСПОРТНОЇ МЕРЕЖІ.

Дана задача вирішується за допомогою *алгоритму Флойда*.

Цей алгоритм більш загальний у порівнянні з алгоритмом Дейкстри, тому що він знаходить найкоротші шляхи між будь-якими двома вузлами мережі. У цьому алгоритмі мережа представлена у виді квадратної матриці з n рядками і n стовпцями. Елемент (i, j) дорівнює відстані d_{ij} від вузла i до вузла j , що має кінцеве значення, якщо існує дуга (i, j) , і дорівнює нескінченності в протилежному випадку.

Покажемо спочатку основну ідею методу Флойда. Нехай є три вузли i, j, k і задані відстані між ними (рис. 5.1). Якщо виконується нерівність $d_{ij} + d_{jk} < d_{ik}$, то доцільно замінити шлях $i \Rightarrow k$ шляхом $i \Rightarrow j \Rightarrow k$. Така заміна (далі її будемо умовно називати *трикутним оператором*) виконується систематично в процесі виконання алгоритму Флойда.

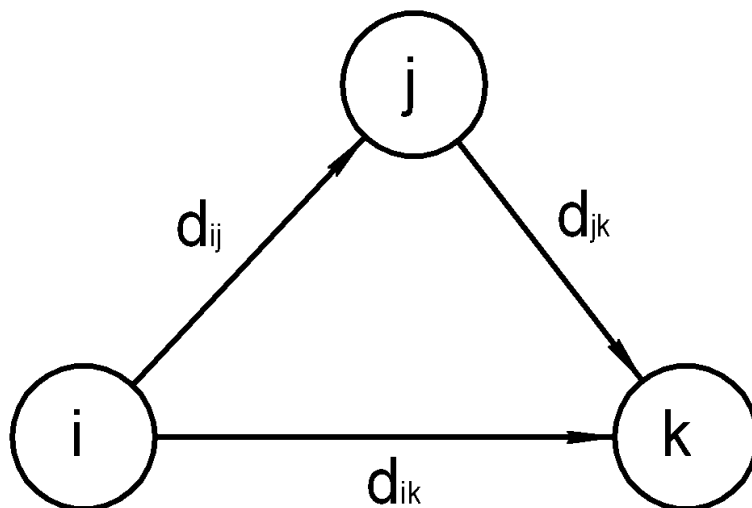


Рис.5.1. Трикутний оператор

Алгоритм Флойда вимагає виконання наступних дій.

Крок 0. Визначаємо початкову матрицю відстаней D_0 і матрицю послідовності вузлів S_0 . Діагональні елементи обох матриць позначаються знаком "-", що показує, що ці елементи в обчисленнях не беруть участь. Думаємо $k = 1$:

Початкова ситуація

$$D_o =$$

	1	2	...	j	...	n
1	–	d_{12}	...	d_{1j}	...	d_{1n}
2	d_{21}	–	...	d_{2j}	...	d_{2n}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	d_{i1}	d_{i2}	...	d_{ij}	...	d_{in}
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	d_{n1}	d_{n2}	...	d_{nj}	...	–

$$S_o =$$

	1	2	...	j	...	n
1	–	2	...	j	...	n
2	1	–	...	j	...	n
⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	1	2	...	j	...	n
⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	1	2	...	j	...	–

Основний крок k . Задаємо рядок k і стовпець k як *ведучий рядок* і *ведучий стовпець*. Розглядаємо можливість застосування трикутного оператора до всіх елементів d_{ij} матриці D_{k-1} . Якщо виконується нерівність $d_{ik} + d_{kj} < d_{ij}$ ($i \Rightarrow k, k \Rightarrow j, i \Rightarrow j$), тоді виконуємо наступні дії:

- створюємо матрицю D_k шляхом заміни в матриці D_{k-1} елемента d_{ij} на суму $d_{ik} + d_{kj}$,

- створюємо матрицю S_k шляхом заміни в матриці S_{k-1} елемента s_{ij} на k . Думаємо $k = k + 1$ і повторюємо крок k .

Пояснимо дії, виконувани на k -м кроці алгоритму, представивши матрицю D_{k-1} так, як вона показана на рис 5.2. На цьому рисунку рядок $k - 1$ і стовпець k є ведучими. Рядок i – будь-який рядок з номером від 1 до $k - 1$, а рядок p – довільний рядок з номером від $k + 1$ до n . Аналогічно стовпець j

представляє будь-як стовпець з номером від 1 до $k-1$, стовпець q – довільний стовпець з номером від $k+1$ до n . Трикутний оператор виконується в такий спосіб. Якщо сума елементів ведучих рядка i стовпця (показаних у квадратах) менше елементів, що знаходяться в перетинанні стовпця i рядка (показаних у кружках), що відповідають розглянутим ведучим елементам, то відстань (елемент у кружку) замінюється на суму відстаней, представлених ведучими елементами:

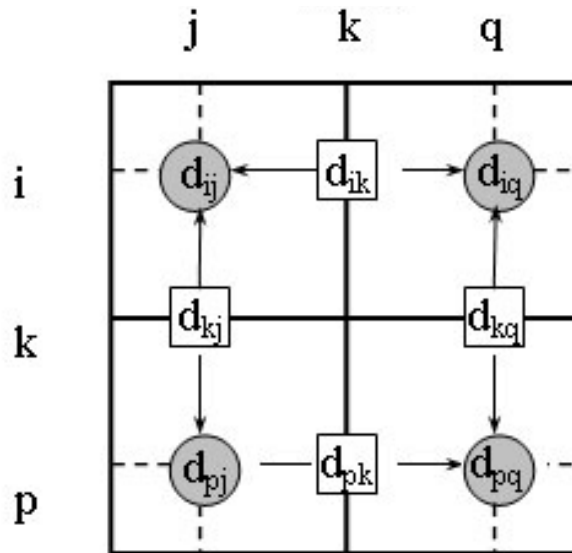


Рис.5.2. Ілюстрація алгоритму Флойда

Після реалізації n кроків алгоритму визначення по матрицях D_n і S_n найкоротшому шляху між вузлами i і j виконується за наступними правилами.

1. Відстань між вузлами i і j дорівнює елементові d_{ij} в матриці D_n .

2. Проміжні вузли шляху від вузла i до вузла j визначаємо по матриці S_n . Нехай $s_{ij} = k$, тоді маємо шлях $i \Rightarrow k \Rightarrow j$. Якщо далі $s_{ik} = k$ і $s_{kj} = j$, тоді вважаємо, що весь шлях визначений, тому що знайдені всі проміжні вузли. У протилежному випадку повторюємо описану процедуру для шляхів від вузла i до вузла k і від вузла k до вузла j .

Приклад рішення задачі визначення найкоротших відстаней між елементами транспортної мережі курсового проекту. Задача вирішується за допомогою алгоритму Флойда. Знайти для мережі, показаної на рис. 3.1, попередньої задачі, найкоротші шляхи між будь-якими вузлами. Відстані між вузлами цієї мережі проставлені на рисунку біля відповідних ребер.

Крок 0. Початкові матриці D_0 і S_0 будуються безпосередньо за заданою схемою мережі відповідно до умов завдання.

D_0

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	∞	∞	∞	∞	∞	∞	∞	∞	∞
7	207	-	207	390	756	∞	∞	∞	∞	∞	∞	∞
4	230	207	-	346	611	792	∞	∞	∞	∞	∞	∞
14	∞	390	346	-	392	∞	475	494	521	∞	∞	∞
27	∞	756	611	392	-	203	∞	∞	219	218	∞	∞
24	∞	∞	792	∞	203	-	∞	∞	360	335	191	∞
33	∞	∞	∞	475	∞	∞	-	23	∞	∞	∞	339
32	∞	∞	∞	494	∞	∞	23	-	225	∞	∞	352
29	∞	∞	∞	521	219	360	∞	225	-	146	∞	423
19	∞	∞	∞	∞	218	335	∞	∞	146	-	265	454
25	∞	∞	∞	∞	∞	191	∞	∞	∞	265	-	609
18	∞	∞	∞	∞	∞	∞	339	352	423	454	609	-

 S_0

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	14	27	24	33	32	29	19	25	18
7	8	-	4	14	27	24	33	32	29	19	25	18
4	8	7	-	14	27	24	33	32	29	19	25	18
14	8	7	4	-	27	24	33	32	29	19	25	18
27	8	7	4	14	-	24	33	32	29	19	25	18
24	8	7	4	14	27	-	33	32	29	19	25	18
33	8	7	4	14	27	24	-	32	29	19	25	18
32	8	7	4	14	27	24	33	-	29	19	25	18
29	8	7	4	14	27	24	33	32	-	19	25	18
19	8	7	4	14	27	24	33	32	29	-	25	18
25	8	7	4	14	27	24	33	32	29	19	-	18
18	8	7	4	14	27	24	33	32	29	19	25	-

Крок 1. У матриці D_0 виділяємо ведучий рядок і стовпець ($k=1$). Аналізуючи вузол 8, бачимо, що через нього здійснюються зв'язки: 7-4. Однак, застосувати трикутний оператор не можна.

 D_1

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	∞	∞	∞	∞	∞	∞	∞	∞	∞
7	207	-	207	390	756	∞	∞	∞	∞	∞	∞	∞
4	230	207	-	346	611	792	∞	∞	∞	∞	∞	∞
14	∞	390	346	-	392	∞	475	494	521	∞	∞	∞
27	∞	756	611	392	-	203	∞	∞	219	218	∞	∞
24	∞	∞	792	∞	203	-	∞	∞	360	335	191	∞
33	∞	∞	∞	475	∞	∞	-	23	∞	∞	∞	339
32	∞	∞	∞	494	∞	∞	23	-	225	∞	∞	352
29	∞	∞	∞	521	219	360	∞	225	-	146	∞	423
19	∞	∞	∞	∞	218	335	∞	∞	146	-	265	454
25	∞	∞	∞	∞	∞	191	∞	∞	∞	265	-	609
18	∞	∞	∞	∞	∞	∞	339	352	423	454	609	-

S_1

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	14	27	24	33	32	29	19	25	18
7	8	-	4	14	27	24	33	32	29	19	25	18
4	8	7	-	14	27	24	33	32	29	19	25	18
14	8	7	4	-	27	24	33	32	29	19	25	18
27	8	7	4	14	-	24	33	32	29	19	25	18
24	8	7	4	14	27	-	33	32	29	19	25	18
33	8	7	4	14	27	24	-	32	29	19	25	18
32	8	7	4	14	27	24	33	-	29	19	25	18
29	8	7	4	14	27	24	33	32	-	19	25	18
19	8	7	4	14	27	24	33	32	29	-	25	18
25	8	7	4	14	27	24	33	32	29	19	-	18
18	8	7	4	14	27	24	33	32	29	19	25	-

Крок 2. У матриці D_1 виділяємо ведучий рядок і стовпець ($k = 2$). Аналізуючи вузол 7, бачимо, що через нього здійснюються зв'язки: 8-14, 8-27, 4-14, 4-27, 14-27. Визначаємо значення шляху, що проходить через вузол 14.

$$d_{8-14}^7 = d_{8-7} + d_{7-14} = 207 + 390 = 597; \quad d_{8-27}^7 = d_{8-7} + d_{7-27} = 207 + 756 = 963;$$

$$d_{4-14}^7 = d_{4-7} + d_{7-14} = 207 + 390 = 597; \quad d_{4-27}^7 = d_{4-7} + d_{7-27} = 207 + 756 = 963;$$

$$d_{4-27}^7 = d_{4-7} + d_{7-27} = 207 + 756 = 963;$$

Порівнюючи отримані значення з первісними, встановлюємо мінімальні величини в кожній парі і заносимо в таблицю менше значення.

D_2

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	597	963	∞	∞	∞	∞	∞	∞	∞
7	207	-	207	390	756	∞	∞	∞	∞	∞	∞	∞
4	230	207	-	346	611	792	∞	∞	∞	∞	∞	∞
14	597	390	346	-	392	∞	475	494	521	∞	∞	∞
27	963	756	611	392	-	203	∞	∞	219	218	∞	∞
24	∞	∞	792	∞	203	-	∞	∞	360	335	191	∞
33	∞	∞	∞	475	∞	∞	-	23	∞	∞	∞	339
32	∞	∞	∞	494	∞	∞	23	-	225	∞	∞	352
29	∞	∞	∞	521	219	360	∞	225	-	146	∞	423
19	∞	∞	∞	∞	218	335	∞	∞	146	-	265	454
25	∞	∞	∞	∞	∞	191	∞	∞	∞	265	-	609
18	∞	∞	∞	∞	∞	∞	339	352	423	454	609	-

S_2

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	7	7	24	33	32	29	19	25	18
7	8	-	4	14	27	24	33	32	29	19	25	18
4	8	7	-	14	27	24	33	32	29	19	25	18
14	7	7	4	-	27	24	33	32	29	19	25	18
27	7	7	4	14	-	24	33	32	29	19	25	18
24	8	7	4	14	27	-	33	32	29	19	25	18
33	8	7	4	14	27	24	-	32	29	19	25	18
32	8	7	4	14	27	24	33	-	29	19	25	18
29	8	7	4	14	27	24	33	32	-	19	25	18
19	8	7	4	14	27	24	33	32	29	-	25	18
25	8	7	4	14	27	24	33	32	29	19	-	18
18	8	7	4	14	27	24	33	32	29	19	25	-

Крок 3. У матриці D_2 виділяємо ведучий рядок і стовпець ($k = 3$). Трикутний оператор застосовується до елементів матриць D_3 і S_3 , виділених заливанням.

$$d_{8-14}^7 = d_{8-4} + d_{4-14} = 230 + 346 = 576;$$

$$d_{8-27}^7 = d_{8-4} + d_{4-27} = 230 + 611 = 841;$$

$$d_{8-24}^4 = d_{8-4} + d_{4-24} = 230 + 792 = 1022;$$

$$d_{7-24}^4 = d_{7-4} + d_{4-24} = 207 + 792 = 999;$$

$$d_{14-24}^4 = d_{14-4} + d_{4-24} = 346 + 792 = 1138;$$

D_3

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	576	841	1022	∞	∞	∞	∞	∞	∞
7	207	-	207	390	756	999	∞	∞	∞	∞	∞	∞
4	230	207	-	346	611	792	∞	∞	∞	∞	∞	∞
14	576	390	346	-	392	1138	475	494	521	∞	∞	∞
27	841	756	611	392	-	203	∞	∞	219	218	∞	∞
24	1022	999	792	1138	203	-	∞	∞	360	335	191	∞
33	∞	∞	∞	475	∞	∞	-	23	∞	∞	∞	339
32	∞	∞	∞	494	∞	∞	23	-	225	∞	∞	352
29	∞	∞	∞	521	219	360	∞	225	-	146	∞	423
19	∞	∞	∞	∞	218	335	∞	∞	146	-	265	454
25	∞	∞	∞	∞	∞	191	∞	∞	∞	265	-	609
18	∞	∞	∞	∞	∞	∞	339	352	423	454	609	-

S_3

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	4	4	4	33	32	29	19	25	18
7	8	-	4	14	27	4	33	32	29	19	25	18
4	8	7	-	14	27	24	33	32	29	19	25	18
14	4	7	4	-	27	4	33	32	29	19	25	18
27	4	7	4	14	-	24	33	32	29	19	25	18
24	4	4	4	4	27	-	33	32	29	19	25	18
33	8	7	4	14	27	24	-	32	29	19	25	18
32	8	7	4	14	27	24	33	-	29	19	25	18
29	8	7	4	14	27	24	33	32	-	19	25	18
19	8	7	4	14	27	24	33	32	29	-	25	18
25	8	7	4	14	27	24	33	32	29	19	-	18
18	8	7	4	14	27	24	33	32	29	19	25	-

Крок 4. У матриці D_3 виділяємо ведучий рядок і стовпець ($k = 4$). Трикутний оператор застосовується до елементів матриць D_4 і S_4 , виділених заливанням.

$$\begin{aligned}
 d_{8-33}^{14} &= d_{8-14} + d_{14-33} = 576 + 475 = 1051; & d_{8-32}^{14} &= d_{8-14} + d_{14-32} = 576 + 494 = 1070; \\
 d_{8-29}^{14} &= d_{8-14} + d_{14-29} = 576 + 521 = 1097; & d_{7-33}^{14} &= d_{7-14} + d_{14-33} = 390 + 475 = 865; \\
 d_{7-32}^{14} &= d_{7-14} + d_{14-32} = 390 + 494 = 884; & d_{7-29}^{14} &= d_{7-14} + d_{14-29} = 390 + 521 = 911; \\
 d_{4-33}^{14} &= d_{4-14} + d_{14-33} = 346 + 475 = 821; & d_{4-32}^{14} &= d_{4-14} + d_{14-32} = 346 + 494 = 840; \\
 d_{4-29}^{14} &= d_{4-14} + d_{14-29} = 346 + 521 = 867; & d_{27-33}^{14} &= d_{27-14} + d_{14-33} = 392 + 475 = 867; \\
 d_{27-32}^{14} &= d_{27-14} + d_{14-32} = 392 + 494 = 886; & d_{24-33}^{14} &= d_{24-14} + d_{14-33} = 1138 + 475 = 1613; \\
 d_{24-32}^{14} &= d_{24-14} + d_{14-32} = 1138 + 494 = 1632; & d_{33-29}^{14} &= d_{33-14} + d_{14-29} = 475 + 521 = 996
 \end{aligned}$$

D_4

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	576	841	1022	1051	1070	1097	∞	∞	∞
7	207	-	207	390	756	∞	865	884	911	∞	∞	∞
4	230	207	-	346	611	792	821	840	867	∞	∞	∞
14	576	390	346	-	392	∞	475	494	521	∞	∞	∞
27	841	756	611	392	-	203	867	886	219	218	∞	∞
24	1022	∞	792	∞	203	-	1613	1632	360	335	191	∞
33	1051	865	821	475	867	1613	-	23	996	∞	∞	339
32	1070	884	840	494	886	1632	23	-	225	∞	∞	352
29	1097	911	867	521	219	360	996	225	-	146	∞	423
19	∞	∞	∞	∞	218	335	∞	∞	146	-	265	454
25	∞	∞	∞	∞	∞	191	∞	∞	∞	265	-	609
18	∞	∞	∞	∞	∞	∞	339	352	423	454	609	-

S_4

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	4	4	4	14	14	14	19	25	18
7	8	-	4	14	27	24	14	14	14	19	25	18
4	8	7	-	14	27	24	14	14	14	19	25	18
14	4	7	4	-	27	24	33	32	29	19	25	18
27	4	7	4	14	-	24	14	14	29	19	25	18
24	4	7	4	14	27	-	14	14	29	19	25	18
33	14	14	14	14	14	14	-	32	14	19	25	18
32	14	14	14	14	14	14	33	-	29	19	25	18
29	14	14	14	14	27	24	14	32	-	19	25	18
19	8	7	4	14	27	24	33	32	29	-	25	18
25	8	7	4	14	27	24	33	32	29	19	-	18
18	8	7	4	14	27	24	33	32	29	19	25	-

Крок 5. У матриці D_4 виділяємо ведучий рядок і стовпець ($k = 5$). Трикутний оператор застосовується до елементів матриць D_5 і S_5 , виділених заливанням.

$$\begin{aligned}
 d_{7-24}^{27} &= d_{7-27} + d_{27-24} = 756 + 203 = 959; & d_{14-24}^{27} &= d_{14-27} + d_{27-24} = 392 + 203 = 595; \\
 d_{8-29}^{27} &= d_{8-27} + d_{27-29} = 841 + 219 = 1060; & d_{4-29}^{27} &= d_{4-27} + d_{27-29} = 611 + 219 = 830; \\
 d_{8-19}^{27} &= d_{8-27} + d_{27-19} = 841 + 218 = 1059; & d_{7-19}^{27} &= d_{7-27} + d_{27-19} = 756 + 218 = 974; \\
 d_{4-19}^{27} &= d_{4-27} + d_{27-19} = 611 + 218 = 829; & d_{14-19}^{27} &= d_{14-27} + d_{27-19} = 392 + 218 = 610; \\
 d_{24-33}^{27} &= d_{24-27} + d_{27-33} = 203 + 867 = 1070; & d_{24-32}^{27} &= d_{24-27} + d_{27-32} = 203 + 886 = 1089; \\
 d_{19-33}^{27} &= d_{19-27} + d_{27-33} = 218 + 867 = 1085; & d_{19-32}^{27} &= d_{19-27} + d_{27-32} = 218 + 886 = 1104;
 \end{aligned}$$

D_5

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	576	841	1022	1051	1070	1060	1059	∞	∞
7	207	-	207	390	756	959	865	884	911	974	∞	∞
4	230	207	-	346	611	792	821	840	830	829	∞	∞
14	576	390	346	-	392	595	475	494	521	610	∞	∞
27	841	756	611	392	-	203	867	886	219	218	∞	∞
24	1022	959	792	595	203	-	1070	1089	360	335	191	∞
33	1051	865	821	475	867	1070	-	23	996	1085	∞	339
32	1070	884	840	494	886	1089	23	-	225	1104	∞	352
29	1060	911	830	521	219	360	996	225	-	146	∞	423
19	1059	974	829	610	218	335	1085	1104	146	-	265	454
25	∞	∞	∞	∞	∞	191	∞	∞	∞	265	-	609
18	∞	∞	∞	∞	∞	∞	339	352	423	454	609	-

S_5

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	4	4	4	14	14	27	27	25	18
7	8	-	4	14	27	27	14	14	14	27	25	18
4	8	7	-	14	27	24	14	14	27	27	25	18
14	4	7	4	-	27	27	33	32	29	27	25	18
27	4	7	4	14	-	24	33	32	29	19	25	18
24	4	27	4	27	27	-	27	27	29	19	25	18
33	14	14	14	14	27	27	-	32	14	27	25	18
32	14	14	14	14	27	27	33	-	29	27	25	18
29	27	14	27	14	27	24	14	32	-	19	25	18
19	27	27	27	27	27	24	27	27	29	-	25	18
25	8	7	4	14	27	24	33	32	29	19	-	18
18	8	7	4	14	27	24	33	32	29	19	25	-

Крок 6. У матриці D_5 виділяємо ведучий рядок і стовпець ($k = 6$). Трикутний оператор застосовується до елементів матриць D_6 і S_6 , виділених заливанням.

$$d_{8-25}^{24} = d_{8-24} + d_{24-25} = 1022 + 191 = 1213;$$

$$d_{7-25}^{24} = d_{7-24} + d_{24-25} = 959 + 191 = 1150;$$

$$d_{4-25}^{24} = d_{4-24} + d_{24-25} = 792 + 191 = 983;$$

$$d_{14-25}^{24} = d_{14-24} + d_{24-25} = 595 + 191 = 786;$$

$$d_{27-25}^{24} = d_{27-24} + d_{24-25} = 203 + 191 = 394;$$

$$d_{33-25}^{24} = d_{33-24} + d_{24-25} = 1070 + 191 = 1261;$$

$$d_{32-25}^{24} = d_{32-24} + d_{24-25} = 1089 + 191 = 1280;$$

$$d_{29-25}^{24} = d_{29-24} + d_{24-25} = 360 + 191 = 551;$$

$$d_{33-29}^{24} = d_{33-24} + d_{24-29} = 1070 + 360 = 1430.$$

D_6

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	576	841	1022	1051	1070	1060	1059	1213	∞
7	207	-	207	390	756	959	865	884	911	974	1150	∞
4	230	207	-	346	611	792	821	840	830	829	983	∞
14	576	390	346	-	392	595	475	494	521	610	786	∞
27	841	756	611	392	-	203	867	886	219	218	394	∞
24	1022	959	792	595	203	-	1070	1089	360	335	191	∞
33	1051	865	821	475	867	1070	-	23	1430	1085	1261	339
32	1070	884	840	494	886	1089	23	-	225	1104	1280	352
29	1060	911	830	521	219	360	1430	225	-	146	551	423
19	1059	974	829	610	218	335	1085	1104	146	-	265	454
25	1213	1150	983	786	394	191	1261	1280	551	265	-	609
18	∞	∞	∞	∞	∞	∞	339	352	423	454	609	-

S_6

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	4	4	4	14	14	27	27	24	18
7	8	-	4	14	27	27	14	14	14	27	24	18
4	8	7	-	14	27	24	14	14	27	27	24	18
14	4	7	4	-	27	27	33	32	29	27	24	18
27	4	7	4	14	-	24	14	14	29	19	24	18
24	4	27	4	27	27	-	27	27	29	19	25	18
33	14	14	14	14	14	27	-	32	24	27	24	18
32	14	14	14	14	14	27	33	-	29	27	24	18
29	27	14	27	14	27	24	24	32	-	19	24	18
19	27	27	27	27	27	24	27	27	29	-	25	18
25	24	24	24	24	24	24	24	24	24	19	-	18
18	8	7	4	14	27	24	33	32	29	19	25	-

Крок 7. У матриці D_6 виділяємо ведучий рядок і стовпець ($k = 7$). Трикутний оператор застосовується до елементів матриць D_7 і S_7 , виділених заливанням.

$$d_{8-18}^{33} = d_{8-33} + d_{33-18} = 1051 + 339 = 1390 ;$$

$$d_{7-18}^{33} = d_{7-33} + d_{33-18} = 865 + 339 = 1204 ;$$

$$d_{4-18}^{33} = d_{4-33} + d_{33-18} = 821 + 339 = 1160 ;$$

$$d_{14-18}^{33} = d_{14-33} + d_{33-18} = 475 + 339 = 814 ;$$

$$d_{27-18}^{33} = d_{27-33} + d_{33-18} = 867 + 339 = 1206 ;$$

$$d_{24-18}^{33} = d_{24-33} + d_{33-18} = 1070 + 339 = 1409 .$$

D_7

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	576	841	1022	1051	1070	1060	1059	1213	1390
7	207	-	207	390	756	959	865	884	911	974	1150	1204
4	230	207	-	346	611	792	821	840	830	829	983	1160
14	576	390	346	-	392	595	475	494	521	610	786	814
27	841	756	611	392	-	203	867	886	219	218	394	1206
24	1022	959	792	595	203	-	1070	1089	360	335	191	1409
33	1051	865	821	475	867	1070	-	23	1430	1085	1261	339
32	1070	884	840	494	886	1089	23	-	225	1104	1280	352
29	1060	911	830	521	219	360	1430	225	-	146	551	423
19	1059	974	829	610	218	335	1085	1104	146	-	265	454
25	1213	1150	983	786	394	191	1261	1280	551	265	-	609
18	1390	1204	1160	814	1206	1409	339	352	423	454	609	-

S_7

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	4	4	4	14	14	27	27	24	33
7	8	-	4	14	27	27	14	14	14	27	24	33
4	8	7	-	14	27	24	14	14	27	27	24	33
14	4	7	4	-	27	27	33	32	29	27	24	33
27	4	7	4	14	-	24	14	14	29	19	24	33
24	4	27	4	27	27	-	27	27	29	19	25	33
33	14	14	14	14	14	27	-	32	24	27	24	18
32	14	14	14	14	14	27	33	-	29	27	24	18
29	27	14	27	14	27	24	24	32	-	19	24	18
19	27	27	27	27	27	24	27	27	29	-	25	18
25	24	24	24	24	24	24	24	24	24	19	-	18
18	33	33	33	33	33	33	33	32	29	19	25	-

Крок 8. У матриці D_7 виділяємо ведучий рядок і стовпець ($k = 8$). Трикутний оператор застосовується до елементів матриць D_8 і S_8 , виділених заливанням.

$$d_{29-33}^{32} = d_{29-32} + d_{32-33} = 225 + 23 = 248.$$

D_8

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	576	841	1022	1051	1070	1060	1059	1213	1390
7	207	-	207	390	756	959	865	884	911	974	1150	1204
4	230	207	-	346	611	792	821	840	830	829	983	1160
14	576	390	346	-	392	595	475	494	521	610	786	814
27	841	756	611	392	-	203	867	886	219	218	394	1206
24	1022	959	792	595	203	-	1070	1089	360	335	191	1409
33	1051	865	821	475	867	1070	-	23	248	1085	1261	339
32	1070	884	840	494	886	1089	23	-	225	1104	1280	352
29	1060	911	830	521	219	360	248	225	-	146	551	423
19	1059	974	829	610	218	335	1085	1104	146	-	265	454
25	1213	1150	983	786	394	191	1261	1280	551	265	-	609
18	1390	1204	1160	814	1206	1409	339	352	423	454	609	-

S_8

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	4	4	4	14	14	27	27	24	33
7	8	-	4	14	27	27	14	14	14	27	24	33
4	8	7	-	14	27	24	14	14	27	27	24	33
14	4	7	4	-	27	27	33	32	29	27	24	33
27	4	7	4	14	-	24	14	14	29	19	24	33
24	4	27	4	27	27	-	27	27	29	19	25	33
33	14	14	14	14	14	27	-	32	32	27	24	18
32	14	14	14	14	14	27	33	-	29	27	24	18
29	27	14	27	14	27	24	32	32	-	19	24	18
19	27	27	27	27	27	24	27	27	29	-	25	18
25	24	24	24	24	24	24	24	24	24	19	-	18
18	33	33	33	33	33	33	33	32	29	19	25	-

Крок 9. У матриці D_8 виділяємо ведучий рядок і стовпець ($k = 9$). Трикутний оператор застосовується до елементів матриць D_9 і S_9 , виділених заливанням.

$$\begin{aligned}
 d_{27-18}^{29} &= d_{27-29} + d_{29-18} = 219 + 423 = 642; & d_{24-18}^{29} &= d_{24-29} + d_{29-18} = 360 + 423 = 783; \\
 d_{33-19}^{29} &= d_{33-29} + d_{29-19} = 248 + 146 = 394; & d_{33-25}^{29} &= d_{33-29} + d_{29-25} = 248 + 551 = 799; \\
 d_{32-19}^{29} &= d_{32-29} + d_{29-19} = 225 + 146 = 371; & d_{32-25}^{29} &= d_{32-29} + d_{29-25} = 225 + 551 = 776; \\
 d_{27-33}^{29} &= d_{27-29} + d_{29-33} = 219 + 248 = 467; & d_{27-32}^{29} &= d_{27-29} + d_{29-32} = 219 + 225 = 444; \\
 d_{24-33}^{29} &= d_{24-29} + d_{29-33} = 360 + 248 = 608; & d_{24-32}^{29} &= d_{24-29} + d_{29-32} = 360 + 225 = 585.
 \end{aligned}$$

D_9

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	576	841	1022	1051	1070	1060	1059	1213	1390
7	207	-	207	390	756	959	865	884	911	974	1150	1204
4	230	207	-	346	611	792	821	840	830	829	983	1160
14	576	390	346	-	392	595	475	494	521	610	786	814
27	841	756	611	392	-	203	467	444	219	218	394	642
24	1022	959	792	595	203	-	608	585	360	335	191	783
33	1051	865	821	475	467	608	-	23	248	394	799	339
32	1070	884	840	494	444	585	23	-	225	371	776	352
29	1060	911	830	521	219	360	248	225	-	146	551	423
19	1059	974	829	610	218	335	394	371	146	-	265	454
25	1213	1150	983	786	394	191	799	776	551	265	-	609
18	1390	1204	1160	814	642	783	339	352	423	454	609	-

S_9

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	4	4	4	14	14	27	27	24	33
7	8	-	4	14	27	27	14	14	14	27	24	33
4	8	7	-	14	27	24	14	14	27	27	24	33
14	4	7	4	-	27	27	33	32	29	27	24	33
27	4	7	4	14	-	24	29	29	29	19	24	29
24	4	27	4	27	27	-	29	29	29	19	25	29
33	14	14	14	14	29	29	-	32	32	29	29	18
32	14	14	14	14	29	29	33	-	29	29	29	18
29	27	14	27	14	27	24	32	32	-	19	24	18
19	27	27	27	27	27	24	29	29	29	-	25	18
25	24	24	24	24	24	24	29	29	24	19	-	18
18	33	33	33	33	29	29	33	32	29	19	25	-

Крок 10. У матриці D_9 виділяємо ведучий рядок і стовпець ($k = 10$). Трикутний оператор застосовується до елементів матриць D_{10} і S_{10} , виділених заливанням.

$$\begin{aligned}
 d_{33-25}^{19} &= d_{33-19} + d_{19-25} = 394 + 265 = 669; & d_{32-25}^{19} &= d_{32-19} + d_{19-25} = 371 + 265 = 636; \\
 d_{29-25}^{19} &= d_{29-19} + d_{19-25} = 146 + 265 = 411.
 \end{aligned}$$

D_{10}

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	576	841	1022	1051	1070	1060	1059	1213	1390
7	207	-	207	390	756	959	865	884	911	974	1150	1204
4	230	207	-	346	611	792	821	840	830	829	983	1160
14	576	390	346	-	392	595	475	494	521	610	786	814
27	841	756	611	392	-	203	467	444	219	218	394	642
24	1022	959	792	595	203	-	608	585	360	335	191	783
33	1051	865	821	475	467	608	-	23	248	394	669	339
32	1070	884	840	494	444	585	23	-	225	371	636	352
29	1060	911	830	521	219	360	248	225	-	146	411	423
19	1059	974	829	610	218	335	394	371	146	-	265	454
25	1213	1150	983	786	394	191	669	636	411	265	-	609
18	1390	1204	1160	814	642	783	339	352	423	454	609	-

 S_{10}

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	4	4	4	14	14	27	27	24	33
7	8	-	4	14	27	27	14	14	14	27	24	33
4	8	7	-	14	27	24	14	14	27	27	24	33
14	4	7	4	-	27	27	33	32	29	27	24	33
27	4	7	4	14	-	24	29	29	29	19	24	29
24	4	27	4	27	27	-	29	29	29	19	25	29
33	14	14	14	14	29	29	-	32	32	29	19	18
32	14	14	14	14	29	29	33	-	29	29	19	18
29	27	14	27	14	27	24	32	32	-	19	19	18
19	27	27	27	27	27	24	29	29	29	-	25	18
25	24	24	24	24	24	24	19	19	19	19	-	18
18	33	33	33	33	29	29	33	32	29	19	25	-

Крок 11. У матриці D_{10} виділяємо ведучий рядок і стовпець ($k = 11$). Однак, застосувати трикутний оператор не можна.

 D_{11}

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	207	230	576	841	1022	1051	1070	1060	1059	1213	1390
7	207	-	207	390	756	959	865	884	911	974	1150	1204
4	230	207	-	346	611	792	821	840	830	829	983	1160
14	576	390	346	-	392	595	475	494	521	610	786	814
27	841	756	611	392	-	203	467	444	219	218	394	642
24	1022	959	792	595	203	-	608	585	360	335	191	783
33	1051	865	821	475	467	608	-	23	248	394	669	339
32	1070	884	840	494	444	585	23	-	225	371	636	352
29	1060	911	830	521	219	360	248	225	-	146	411	423
19	1059	974	829	610	218	335	394	371	146	-	265	454
25	1213	1150	983	786	394	191	669	636	411	265	-	609
18	1390	1204	1160	814	642	783	339	352	423	454	609	-

S_{11}

	8	7	4	14	27	24	33	32	29	19	25	18
8	-	7	4	4	4	4	14	14	27	27	24	33
7	8	-	4	14	27	27	14	14	14	27	24	33
4	8	7	-	14	27	24	14	14	27	27	24	33
14	4	7	4	-	27	27	33	32	29	27	24	33
27	4	7	4	14	-	24	29	29	29	19	24	29
24	4	27	4	27	27	-	29	29	29	19	25	29
33	14	14	14	14	29	29	-	32	32	29	19	18
32	14	14	14	14	29	29	33	-	29	29	19	18
29	27	14	27	14	27	24	32	32	-	19	19	18
19	27	27	27	27	27	24	29	29	29	-	25	18
25	24	24	24	24	24	24	19	19	19	19	-	18
18	33	33	33	33	29	29	33	32	29	19	25	-

Крок 12. У матриці D_{11} виділяємо ведучий рядок і стовпець ($k = 12$). Однак, застосувати трикутний оператор не можна.

Кінцеві матриці D_{11} і S_{11} містять всю інформацію, необхідну для визначення найкоротших шляхів між будь-якими двома вузлами мережі. Наприклад, найкоротша відстань між вузлами 8 і 18 дорівнює $d_{8-18} = 1390$.

Для знаходження відповідних маршрутів нагадаємо, що сегмент маршруту (i, j) складається з ребра (i, j) тільки в тому випадку, коли $s_{ij} = j$. У протилежному випадку вузли i і j зв'язані, принаймні, через один проміжний вузол. Наприклад, оскільки $s_{8-18} = 33$ і $s_{8-33} = 14$ спочатку найкоротший маршрут між вузлами 8 і 18 буде мати вигляд $8 \Rightarrow 33 \Rightarrow 18$. Але тому що s_{8-33} не дорівнює 33, вузли 8 і 33 у визначеному шляху не зв'язані одним ребром (але у вихідній мережі вони можуть бути зв'язані безпосередньо). Далі варто визначити проміжний вузол (вузли) між восьмим і тридцять третім вузлами. Маємо $s_{8-33} = 14$ і $s_{8-14} = 4$, тому маршрут $8 \Rightarrow 33$ заміняємо $8 \Rightarrow 14 \Rightarrow 33$. Але тому що s_{8-14} не дорівнює 14, вузли 8 і 14 у визначеному шляху не зв'язані одним ребром. Далі варто визначити проміжний вузол (вузли) між восьмим і чотирнадцятим вузлами. Маємо $s_{8-14} = 4$ і $s_{8-4} = 4$, тому маршрут $8 \Rightarrow 14$ заміняємо $8 \Rightarrow 4 \Rightarrow 14$. Оскільки $s_{8-14} = 4$ і $s_{8-4} = 4$, інших проміжних вузлів немає. Комбінуючи визначені сегменти маршруту, остаточно одержуємо наступний найкоротший шлях від вузла 8 до вузла 18: $8 \Rightarrow 4 \Rightarrow 14 \Rightarrow 33 \Rightarrow 18$. Довжина цього шляху дорівнює 1390 кілометрам.

6. ЗАДАЧА ПРО МАКСИМАЛЬНИЙ ПОТІК (АЛГОРИТМ ФОРДА-ФАЛКЕРСОНА)

При аналізі транспортних мереж часто виникає задача визначення максимального потоку, що може пропустити дана мережа, а також задача розподілу цього потоку по дугах мережі.

З математичної точки зору задача про максимальний потік формулюється в такий спосіб: при заданій конфігурації мережі і відомої пропускної здатності C_{ij} знайти ненегативні значення φ_{ij} , що задовольняють умовам і, що максимізують функцію φ , тобто

$$\varphi = \sum_i \varphi_{oi} = \sum_j \varphi_{jn} \rightarrow \max.$$

Алгоритм для знаходження максимального потоку був запропонований Фордом і Фалкерсоном і полягає в поступовому збільшенні потоку, що пропускається по мережі, доти, поки він не стане найбільшим. Алгоритм заснований на теоремі Форда-Фалкерсона: у будь-якій транспортній мережі максимальний потік із джерела x_o в стік x_n , дорівнює мінімальній пропускній здатності розрізу, що відокремлює x_o від x_n .

Приклад рішення задачі про максимальний потік. Схема мережі показана на рис. 3.1. Дані щодо пропускної здатності елементів мережі наведені у табл. 6.1. При аналізі цієї транспортної мережі визначити максимальний потік, що може пропустити дана мережа, а також знайти розподіл цього потоку по дугах мережі.

Т а б л и ц я 6.1.

Вихідні дані щодо пропускної здатності мережі

№ вузла	Станція	Пропускна здатність
8	Львів	33
7	Рівне	10
4	Хмельницький	22
14	Київ	40
27	Кіровоград	9
24	Херсон	14
33	Харків	16
32	Мерефа	9
29	Дніпропетровськ	24
19	Запоріжжя	31
25	Джанкой	31
18	Луганськ	23

Т а б л и ц я 6.2

Матриця пропускних здатностей дуг мережі

	8	7	4	14	27	24	33	32	29	19	25	18
8	X	33	33 ⁻									
7	10	X	10	10	10							
4	22 ⁺	22	X	22	22	22 ⁻						
14		40	40	X	40		40	40	40			
27		9	9	9	X	9			9	9		
24			14 ⁺		14	X			14	14	14 ⁻	
33				16			X	16				16
32				9			9	X	9			9
29				24	24	24		24	X	24		24
19					31	31			31	X	31	31
25						31 ⁺				31	X	31 ⁻
18							23	23	23	23	23 ⁺	X

По табл. 6.2. знаходимо шлях $l_1 = (8, 4, 24, 25, 18)$ зі станції 8 у 18 з позитивною пропускною здатністю. Елементи цього шляху C_{ij} позначаємо знаком «мінус», а симетричні C_{ji} – знаком «плюс». Визначаємо пропускну здатність знайденого шляху, що дорівнює найменшій з пропускних здатностей дуг:

$$C_1 = \min\{33, 22, 14, 31\} = 14.$$

Визначаються залишкові пропускні здатності дуг знайденого шляху і симетричних йому дуг. Для цього з елементів C_{ij}^- табл. 6.2. віднімаємо C_1 , а до елементів C_{ji}^+ додаємо C_1 . У результаті одержимо нову табл. 5.3 зі зміненими пропускними здатностями.

Т а б л и ц я 6.3

Матриця пропускних здатностей дуг мережі

	8	7	4	14	27	24	33	32	29	19	25	18
8	X	33	19 ⁻									
7	10	X	10	10	10							
4	36 ⁺	22	X	22	22 ⁻	8						
14		40	40	X	40		40	40	40			
27		9	9 ⁺	9	X	9			9	9 ⁻		
24			28		14	X			14	14	0	
33				16			X	16				16
32				9			9	X	9			9
29				24	24	24		24	X	24		24
19					31 ⁺	31			31	X	31	31 ⁻
25						45				31	X	17
18							23	23	23	23 ⁺	37	X

Позначаємо стовпці табл. 6.3, знаходимо другий шлях $l_2 = (8, 4, 27, 19, 18)$ зі станції 8 у 18, і розставляємо знаки. Визначаємо пропускну здатність знайденого шляху, що дорівнює найменшій з пропускових здатностей дуг:

$$C_2 = \min\{19, 22, 9, 31\} = 9.$$

Змінимо пропускові здатності позначених дуг на C_2 (табл. 6.4).

Т а б л и ц я 6.4

Матриця пропускових здатностей дуг мережі

	8	7	4	14	27	24	33	32	29	19	25	18
8	X	33	10 ⁻									
7	10	X	10	10	10							
4	45 ⁺	22	X	22 ⁻	13	8						
14		40	40 ⁺	X	40		40 ⁻	40	40			
27		9	18	9	X	9			9	0		
24			28		14	X			14	14	0	
33				16 ⁺			X	16				16 ⁻
32				9			9	X	9			9
29				24	24	24		24	X	24		24
19					31	40			31	X	31	22
25						45				31	X	17
18							23 ⁺	23	23	32	37	X

Позначивши стовпці знаходимо $l_3 = (8, 4, 14, 33, 18)$.

Величина потоку по шляху l_3 : $C_3 = \min\{10, 22, 40, 16\} = 10$.

Розрахувавши нові пропускові здатності дуг, приходимо до табл. 6.5.

Т а б л и ц я 6.5

Матриця пропускових здатностей дуг мережі

	8	7	4	14	27	24	33	32	29	19	25	18
8	X	33 ⁻	0									
7	10 ⁺	X	10	10	10 ⁻							
4	55	22	X	12	13	8						
14		40	50	X	40		30	40	40			
27		9 ⁺	18	9	X	9			9 ⁻	0		
24			28		14	X			14	14	0	
33				26			X	16				6
32				9			9	X	9			9
29				24	24 ⁺	24		24	X	24		24 ⁻
19					31	40			31	X	31	22
25						45				31	X	17
18							33	23	23 ⁺	32	37	X

Позначивши стовпці знаходимо $l_4 = (8, 7, 27, 29, 18)$.

Величина потоку по шляху l_4 : $C_4 = \min\{33, 10, 9, 24\} = 9$.

Розрахувавши нові пропускні здатності дуг, приходимо до табл. 6.6.

Т а б л и ц я 6.6

Матриця пропускних здатностей дуг мережі

	8	7	4	14	27	24	33	32	29	19	25	18
8	X	24 ⁻	0									
7	19 ⁺	X	10	10 ⁻	1							
4	55	22	X	12	13	8						
14		40 ⁺	50	X	40		30	40	40 ⁻			
27		18	18	9	X	9			0	0		
24			28		14	X			14	14	0	
33				26			X	16				6
32				9			9	X	9			9
29				24 ⁺	33	24		24	X	24		15 ⁻
19					31	40			31	X	31	22
25						45				31	X	17
18							33	23	32 ⁺	32	37	X

Позначивши стовпці в табл. 6.6 знаходимо $l_5 = (8, 7, 14, 29, 18)$.

Величина потоку по шляху l_5 : $C_5 = \min\{24, 10, 40, 15\} = 10$.

Розрахувавши нові пропускні здатності дуг, приходимо до табл. 6.7.

Т а б л и ц я 6.7

Матриця пропускних здатностей дуг мережі

	8	7	4	14	27	24	33	32	29	19	25	18
8	X	14 ⁻	0									
7	29 ⁺	X	10 ⁻	0	1							
4	55	22 ⁺	X	12	13 ⁻	8						
14		50	50	X	40		30	40	30			
27		18	18 ⁺	9	X	9 ⁻			0	0		
24			28		14 ⁺	X			14	14 ⁻	0	
33				26			X	16				6
32				9			9	X	9			9
29				34	33	24		24	X	24		5
19					31	40 ⁺			31	X	31	22 ⁻
25						45				31	X	17
18							33	23	42	32 ⁺	37	X

Позначивши стовпці в табл. 6.7 знаходимо $l_6 = (8, 7, 4, 27, 24, 19, 18)$.

Величина потоку по шляху l_6 : $C_6 = \min\{14, 10, 13, 9, 14, 22\} = 9$.

Розрахувавши нові пропускні здатності дуг, приходимо до табл. 6.8.

Т а б л и ц я 6.8

Матриця пропускних здатностей дуг мережі

	8	7	4	14	27	24	33	32	29	19	25	18
8	X	5 ⁻	0									
7	38 ⁺	X	1 ⁻	0	1							
4	55	31 ⁺	X	12	4	8 ⁻						
14		50	50	X	40		30	40	30			
27		18	27	9	X	0			0	0		
24			28 ⁺		23	X			14	5 ⁻	0	
33				26			X	16				6
32				9			9	X	9			9
29				34	33	24		24	X	24		5
19					31	49 ⁺			31	X	31	13 ⁻
25						45				31	X	17
18							33	23	42	41 ⁺	37	X

Позначивши стовпці в табл. 6.8 знаходимо $l_7 = (8, 7, 4, 24, 19, 18)$.

Величина потоку по шляху l_6 : $C_7 = \min\{5, 1, 8, 5, 13\} = 1$.

Обчисливши нові пропускні здатності дуг, приходимо до табл. 6.9.

Т а б л и ц я 6.9

Матриця пропускних здатностей дуг мережі

	8	7	4	14	27	24	33	32	29	19	25	18
8	X	4	0									
7	39	X	0	0	1							
4	55	32	X	12	4	7						
14		50	50	X	40		30	40	30			
27		18	27	9	X	0			0	0		
24			29		23	X			14	4	0	
33				26			X	16				6
32				9			9	X	9			9
29				34	33	24		24	X	24		5
19					31	50			31	X	31	12
25						45				31	X	17
18							33	23	42	42	37	X

Переглядаючи рядки і позначаючи стовпці переконуємося в тім, що стовпець 27 позначити неможливо. Отже, більше не існує жодного шляху з позитивною пропускною здатністю з вершини 8 у вершину 18.

Заключний крок. Віднімаючи з елементів табл. 6.2 відповідні елементи табл. 6.9, одержимо табл. 6.10.

Т а б л и ц я 6.10

Матриця пропускних здатностей дуг мережі

	8	7	4	14	27	24	33	32	29	19	25	18
8	X	29	33									
7	-29	X	10	10	9							
4	-33	-10	X	10	18	15						
14		-10	-10	X			10		10			
27		-9	-18		X	9			9	9		
24			-15		-9	X				10	14	
33				-10			X					10
32								X				
29				-10	-9				X			19
19					-9	-10				X		19
25						-14					X	14
18							-10		-19	-19	-14	X

Позитивні елементи цієї таблиці характеризують величини дугових потоків. Величина максимального потоку дорівнює сумі елементів 8-го рядка табл. 6.10 або сумі елементів 18-го стовпця. Усі дуги розрізу (R^*, \bar{R}^*) є насиченими.

7. ЗАДАЧА КОМІВОЯЖЕРА

Характерними прикладами використання задач такого типу на залізничному транспорті є знаходження оптимальних варіантів маневрової роботи на сортувальних станціях, знаходження маршрутів контролю дефектних рейок та стрілочних переводів, які на даний час складаються спеціалістами дистанції колії, та не завжди складений ними маршрут є оптимальним.

Задачі такого типу відомі під загальною назвою задача комівояжера, в «класичному» формулюванні якої комівояжер намагається визначити найкоротший маршрут для одноразового відвідування n міст. Власне кажучи, ця задача є задачею про призначення з додатковими обмеженнями, які гарантують виключення з оптимального рішення неповних замкнутих маршрутів. У задачі комівояжера замкнутий маршрут, що проходить через кожен пункт тільки один раз, називається циклом; цикл, що проходить через всі пункти, називається повним, у протилежному випадку – частковим або підциклом. У задачі комівояжера з n містами можна визначити такі змінні:

$$x_{ji} = \begin{cases} 1, & \text{якщо місто } j \text{ досягне з міста } i \\ 0, & \text{якщо місто } j \text{ не досягне з міста } i. \end{cases}$$

Маючи значення відстані d_{ij} від міста i до міста j (вважається, що $d_{ij} = \infty$ при $i = j$), задачу комівояжера можна формалізувати в такий спосіб.

$$\text{Мінімізувати } z = \sum_{i=1}^n \sum_{i=1}^n d_{ij} x_{ij}$$

при обмеженнях

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= 1, i = 1, 2, \dots, n, \\ \sum_{i=1}^n x_{ij} &= 1, j = 1, 2, \dots, n, \\ x_{ij} &= 0 \text{ або } 1, \end{aligned}$$

Рішення повинне бути повним циклом.

Обмеження задачі, крім останнього, визначають звичайну задачу про призначення. У загальному випадку немає гарантії, що оптимальне рішення задачі комівояжера може бути отримане шляхом рішення відповідної задачі про призначення, тобто, що оптимальне рішення задачі про призначення утворить повний цикл. Більш імовірно, що воно буде містити часткові цикли, що з'єднують разом деякі підмножини вузлів мережі. На рис. 7.1 показана задача комівояжера, що складається з 5-ти міст. Ребра, що з'єднують вузли (міста) мережі, відповідають дорогам із двостороннім рухом. На цьому рисунку показане рішення задачі комівояжера й часткові цикли, що відповідають рішенню задачі про призначення. Якщо задача про призначення формує по-

вний цикл, то це буде оптимальне рішення задачі комівояжера. Якщо оптимальне рішення задачі про призначення складається з декількох часткових циклів, то в цю задачу додаються спеціальні обмеження, що видаляють рішення із частковими циклами.

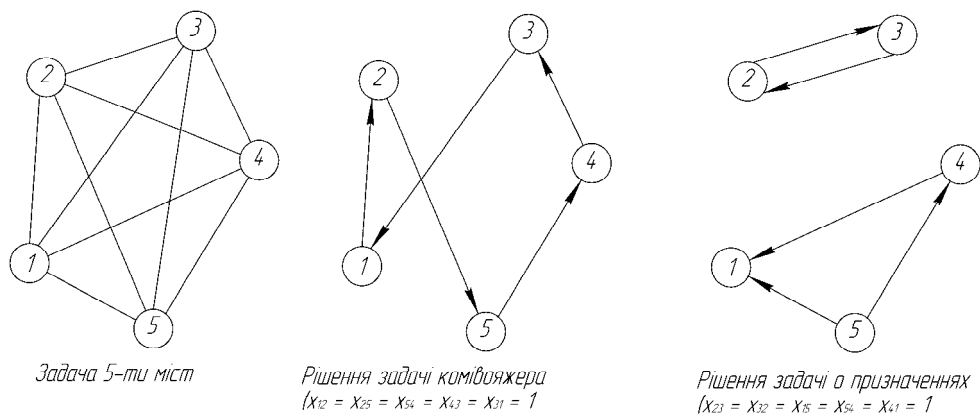


Рис. 7.1. Приклад задачі комівояжера для 5-ти міст

Основні методи рішення задачі комівояжера побудовані на тих же основах, що й методи цілочислового програмування.

Застосування методу відгалужень і меж для рішення задачі комівояжера. Спочатку алгоритм застосовується до задачі про призначення, що відповідає даній задачі комівояжера. Якщо отримане оптимальне рішення утворить повний цикл, то на цьому обчислення закінчуються. У протилежному випадку в задачу необхідно ввести обмеження, що видаляють часткові цикли. Це створює ряд підзадач, яких у загальному випадку стільки ж, скільки змінних утворять один частковий цикл. У кожній підзадачі одна з таких змінних встановлюється рівною нулю. Рішення цих підзадач можуть або сформувати, або ні повний цикл. Якщо такий цикл сформований у будь-якій підзадачі, то оптимальне значення цільової функції цієї підзадачі дає верхню границю довжини повного циклу. Якщо ж підзадача не формує повний цикл, то вона знову розбивається на ряд нових підзадач відповідно до змінної, яка утворить часткові цикли в рішенні даної підзадачі. Цей процес триває доти, поки не будуть розглянуті всі існуючі підзадачі, або поки не буде отримане краще (менше) значення верхньої межі довжини повного циклу, або поки не буде доведено, що підзадачі не можуть мати кращого рішення. Оптимальним рішенням задачі комівояжера буде повний цикл, на якому досягається найкраще значення верхньої межі.

Приклад вирішення задачі комівояжера. У наступній матриці записані відстані між 6 містами задачі комівояжера, схема якої наведена на рис. 7.2.

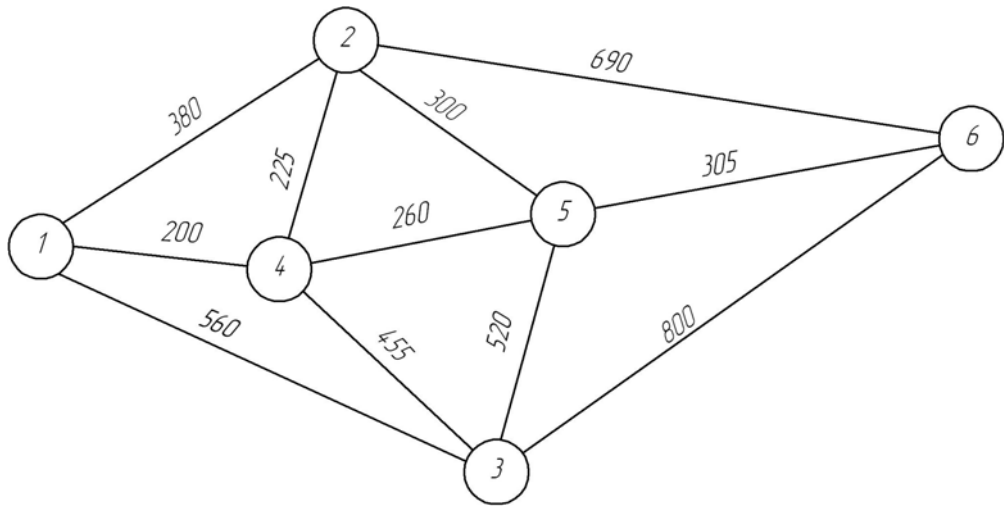


Рис. 7.2. Схема транспортної мережі

$$\|d_{ij}\| = \begin{bmatrix} \infty & 380 & 560 & 200 & \infty & \infty \\ 380 & \infty & \infty & 225 & 300 & 690 \\ 560 & \infty & \infty & 455 & 520 & 800 \\ 200 & 225 & 455 & \infty & 260 & \infty \\ \infty & 300 & 520 & 260 & \infty & 305 \\ \infty & 690 & 800 & \infty & 305 & \infty \end{bmatrix}$$

Розглянемо особливості вирішення задачі комівояжера у MS Excel. Робочий аркуш з введеними умовами задачі наведений на рис. 7.3.

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7			1	2	3	4	5	6		
8		1	10000	380	560	200	10000	10000		
9		2	380	10000	10000	225	300	690		
10		3	560	10000	10000	455	520	800		
11		4	200	225	455	10000	260	10000		
12		5	10000	300	520	260	10000	305		
13		6	10000	690	800	10000	305	10000	Сумарна відстань	
14									0	
15										
16										
17			1	2	3	4	5	6		
18		1							0	
19		2							0	
20		3							0	
21		4							0	
22		5							0	
23		6							0	
24			0	0	0	0	0	0		
25										

Рис. 7.3. Введення вихідних даних задачі комівояжера

У клітинки робочого аркушу вводимо за допомогою майстра функцій відповідні формули:

- У клітинку I14 = СУММПРОИЗВ(C8:H13;C18:H23) для розрахунку цільової функції задачі;
- У клітинку C24 = СУММ(C18:C23) для розрахунку суми змінних у клітинках C18–C23;
- Аналогічним чином виконуємо підсумовування в клітинках D24–H24.
- У клітинку I18 = СУММ(C18:H18) для розрахунку суми змінних у клітинках C18–H18;
- Аналогічним чином виконуємо підсумовування в клітинках I18–I23.

Вибираємо пункт головного меню «Сервіс» – «Пошук рішення». У діалоговому вікні «Пошук рішення», що з'явилося на екрані (рис. 7.4) встановлюємо: цільова клітинка \$I\$14; значення цільової клітинки мінімальне; змінювані клітинки \$C\$18:\$H\$23.

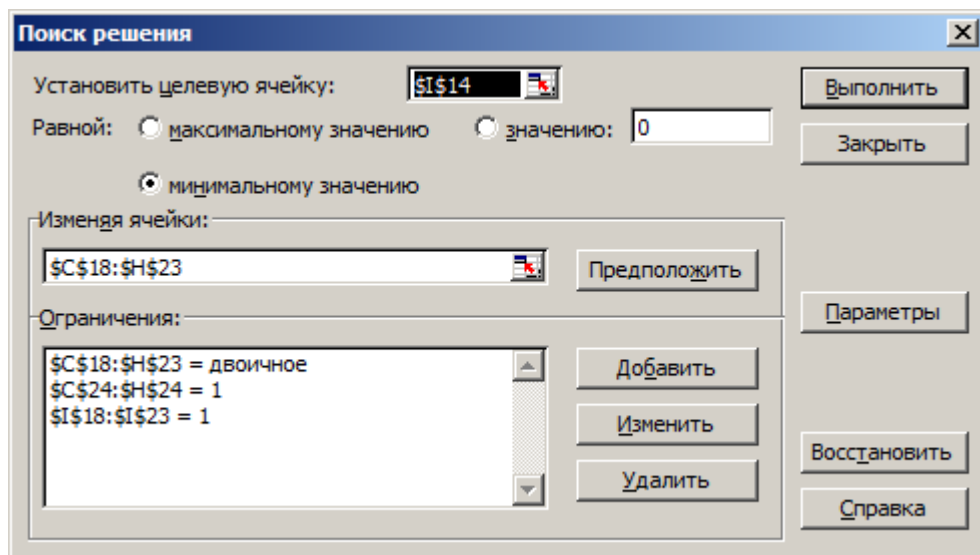


Рис. 7.4. Введення параметрів і обмежень задачі комівояжера

Натискаючи кнопку «Додати» діалогового вікна «Пошук рішення» послідовно задаємо обмеження задачі:

- Сума значень змінних у кожному стовпці повинна дорівнювати одиниці (кожне ребро транспортної мережі може вийти у маршрут тільки один раз: $\$C\$24:\$H\$24 = 1$).
- Сума значень змінних у кожному рядку повинна дорівнювати одиниці (кожне ребро транспортної мережі при зворотному русі може вийти у маршрут тільки один раз: $\$I\$18:\$I\$23 = 1$).
- Для введення обмеження на значення змінних задачі $x_{ij} = \{0, 1\}$ у діалоговому вікні «Додавання обмеження» (рис. 7.5) у полі посилання на клітинку зазначаємо діапазон клітинок $\$C\$18:\$H\23 , який відповідає змінним задачі, а у списку видів обмежень вибираємо значення «двоич». Значення у полі «Обмеження» автоматично встановлюється на «двоичне».

Натискаємо кнопку «Параметри» і у діалоговому вікні «Параметри пошуку» рішення (рис. 7.6) встановлюємо прапорці у пунктах «Лінійна модель» (використовуємо симплекс-метод). Натиснувши кнопку «Виконати» діалогового вікна «Пошук рішення» отримаємо оптимальний розв’язок задачі безпосередньо на робочому аркуші з вихідними даними (рис. 7.7).

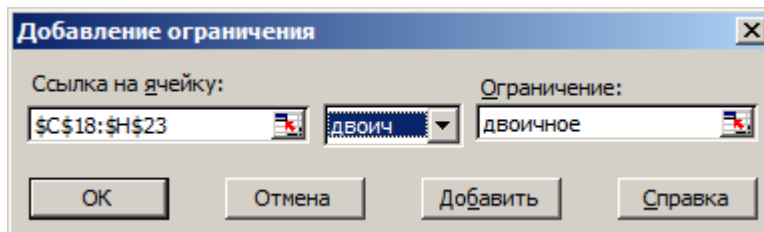


Рис. 7.5 Встановлення обмежень на значення змінних задачі

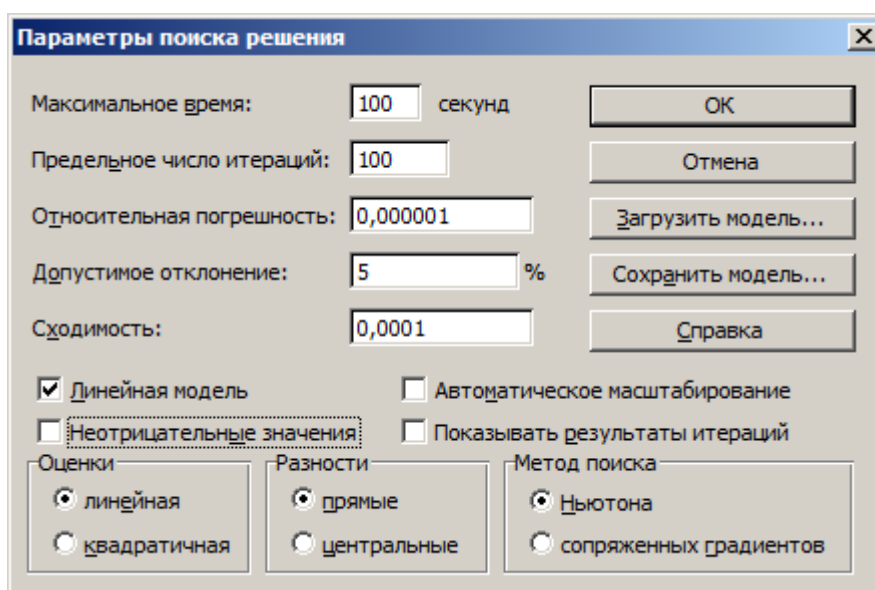


Рис. 7.6. Налаштування параметрів пошуку

З рис. 7.7 бачимо, що оптимальне рішення цільової функції дорівнює 2180 (клітинка I14). Оптимальному призначенню ребер транспортної мережі відповідають одиниці у діапазоні клітинок C18:H23. Знайдене рішення наступне: $z = 2180 (1-3-1)(2-4-2)(5-6-5)$. Це рішення породжує три часткових цикли (1-3-1), (2-4-2) і (5-6-5). Це ж рішення позначене як вузол 1 дерева підзадач на рис. 7.8. Отримане значення цільової функції $z = 2180$ приймаємо як нижню межу довжини оптимального циклу, що проходить через всі 6 міст.

Прямолінійний спосіб визначення верхньої межі оптимального циклу полягає в тому, щоб вибрати який-небудь повний цикл і підрахувати його довжину. Наприклад, повний цикл 1-2-6-5-3-4-1 (обраний довільно) дає загальну довжину $380+690+305+520+455+200=2550$. Обчислені нижня й верхня межі показують, що довжина оптимального повного циклу лежить в інтервалі від 2180 до 2550. Підзадачі, що дають рішення з довжиною циклу більше 2550, відкидаються.

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7			1	2	3	4	5	6		
8		1	10000	380	560	200	10000	10000		
9		2	380	10000	10000	225	300	690		
10		3	560	10000	10000	455	520	800		
11		4	200	225	455	10000	260	10000		
12		5	10000	300	520	260	10000	305		
13		6	10000	690	800	10000	305	10000	Сумарна відстань	
14									2180	
15										
16										
17			1	2	3	4	5	6		
18		1	0	0	0	4,4E-16	0	0		1
19		2	2,2E-16	0	0	0	1	2,2E-16		1
20		3	1	0	0	0	0	2,2E-16		1
21		4	2,2E-16	1	0	0	0	0		1
22		5	0	2,2E-16	0	0	0	1		1
23		6	0	0	2,2E-16	0	1	0		1
24			1	1	1	1	1	1		

Рис. 7.7. Результати розв'язку задачі

Аналітичне рішення задачі про призначення без використання MS Excel. Таблиця 7.1 відповідає матриці відстаней для транспортної мережі (дані представлені в км).

Таблиця 7.1

Вихідні дані задачі про призначення

∞	380	560	200	∞	∞	$p_1 = 200$
380	∞	∞	225	300	690	$p_1 = 225$
560	∞	∞	455	520	800	$p_1 = 455$
200	225	455	∞	260	∞	$p_1 = 200$
∞	300	520	260	∞	305	$p_1 = 260$
∞	690	800	∞	305	∞	$p_1 = 305$

Вирішимо цю задачу про призначення угорським методом.

Етап 1. У вихідній матриці вартостей визначимо в кожному рядку мінімальну вартість і віднімемо її від інших елементів рядка.

Таблиця 7.2

Виконання етапу 1

∞	180	360	0	∞	∞	$p_1 = 200$
155	∞	∞	0	75	465	$p_2 = 225$
105	∞	∞	0	65	345	$p_3 = 455$
0	25	255	∞	60	∞	$p_4 = 200$
∞	40	260	0	∞	45	$p_5 = 260$
∞	385	495	∞	0	∞	$p_6 = 305$
$q_1 = 0$	$q_2 = 25$	$q_3 = 255$	$q_4 = 0$	$q_5 = 0$	$q_6 = 45$	

Тому, що не можливо відмітити у кожному рядку та стовпці по одному нулю, то оптимальне призначення не знайдено, переходимо до етапу 2.

Етап 2. У матриці, отриманій на першому етапі, знайдемо в кожному стовпці мінімальну вартість і віднімемо її від інших елементів стовпця.

Таблиця 7.3

Виконання етапу 2

∞	155	105	0	∞	∞
155	∞	∞	0	75	420
105	∞	∞	0	65	300
0	0	0	∞	60	∞
∞	15	5	0	∞	0
∞	360	240	∞	0	∞
$q_1 = 0$	$q_2 = 25$	$q_3 = 255$	$q_4 = 0$	$q_5 = 0$	$q_6 = 45$

Тому, що не можливо відмітити у кожному рядку та стовпці по одному нулю, то оптимальне призначення не знайдено, переходимо до етапу 3.

Етап 3. Якщо після виконання першого й другого етапів описаного алгоритму не отримане припустиме рішення, виконуються наступні дії.

I) В останній матриці проведіть мінімальне число горизонтальних і вертикальних прямих по рядках і стовпцям, щоб викреслити в матриці всі нульові елементи

II) Знайдіть найменший не викреслений елемент і відніміть його з інших не викреслених елементів і додайте до елементів, що знаходяться на перетинанні проведених на попередньому етапі прямих.

III) Якщо новий розподіл нульових елементів не дозволяє побудувати припустиме рішення, повторите етап 3.

У задачі даного приклада виконання етапу 3 вимагає проведення чотирьох прямих і приводить до табл. 7.4.

Таблиця 7.4

Виконання етапу 3.

∞	155	105	0	∞	∞
155	∞	∞	0	75	420
105	∞	∞	0	65	300
0	0	0	∞	60	∞
∞	15	<u>5</u>	0	∞	0
∞	360	240	∞	0	∞

Найменший не викреслений елемент (він підкреслений) дорівнює 5. Цей елемент віднімаємо з інших не викреслених елементів і додаємо до елементів, що знаходяться на перетинанні прямих. У результаті одержимо матрицю, представлену у вигляді табл. 7.5.

Тому що новий розподіл нульових елементів не дозволяє побудувати припустиме рішення, повторимо етап 3.

Таблиця 7.5

Виконання етапу 3.

∞	150	<u>100</u>	0	∞	∞
150	∞	∞	0	70	420
100	∞	∞	0	60	300
0	0	0	∞	60	∞
∞	10	0	0	∞	0
∞	360	240	∞	0	∞

Найменший не викреслений елемент (він підкреслений) дорівнює 100. Цей елемент віднімаємо з інших не викреслених елементів і додаємо до елементів, що знаходяться на перетинанні прямих. У результаті одержимо матрицю, представлену у вигляді табл. 7.6.

Таблиця 7.6

Виконання етапу 3.

∞	50	0	0	∞	∞
50	∞	∞	0	70	320
0	∞	∞	0	60	200
0	0	0	∞	160	∞
∞	10	0	100	∞	0
∞	260	140	∞	0	∞

Оптимальне рішення, показане в таблиці підкресленими нулями, пропонує рішення: (1-3-1)(2-4-2)(5-6-5), що є однаковим з рішенням за допомогою MS Excel. Відповідне значення цільової функції можна одержати шляхом підсумовування значень відстаней: $(560+225+560+225+305+305) = 2180$ км.

Щоб виключити часткові цикли в задачі вузла 1, треба «розбити» ці часткові цикли шляхом примусового прирівнювання змінних x_{ij} . Частковий цикл 1-3-1 можна розбити, якщо покласти $x_{1-3} = 0$ або $x_{3-1} = 0$ (одночасно тільки одну змінну). Аналогічно довільний частковий цикл 3-6-5-3 можна розбити, якщо ввести обмеження $x_{3-6} = 0$, $x_{6-5} = 0$ або $x_{5-3} = 0$. Кожне з подібних обмежень породжує окрему галузь дерева підзадач. Немає необхідності розбивати всі наявні часткові цикли – досить виключити тільки *один* частковий цикл. Причина цього в тім, що введення в задачу нового обмеження автоматично впливає на значення *всіх* змінних цієї задачі, що створює "сприятливі" умови для формування повного циклу. На підставі цього аргументу звичайно розбивають один частковий цикл, найменший по кількості складових його дуг, оскільки це приводить до меншої кількості нових галузей у дереві підзадач.

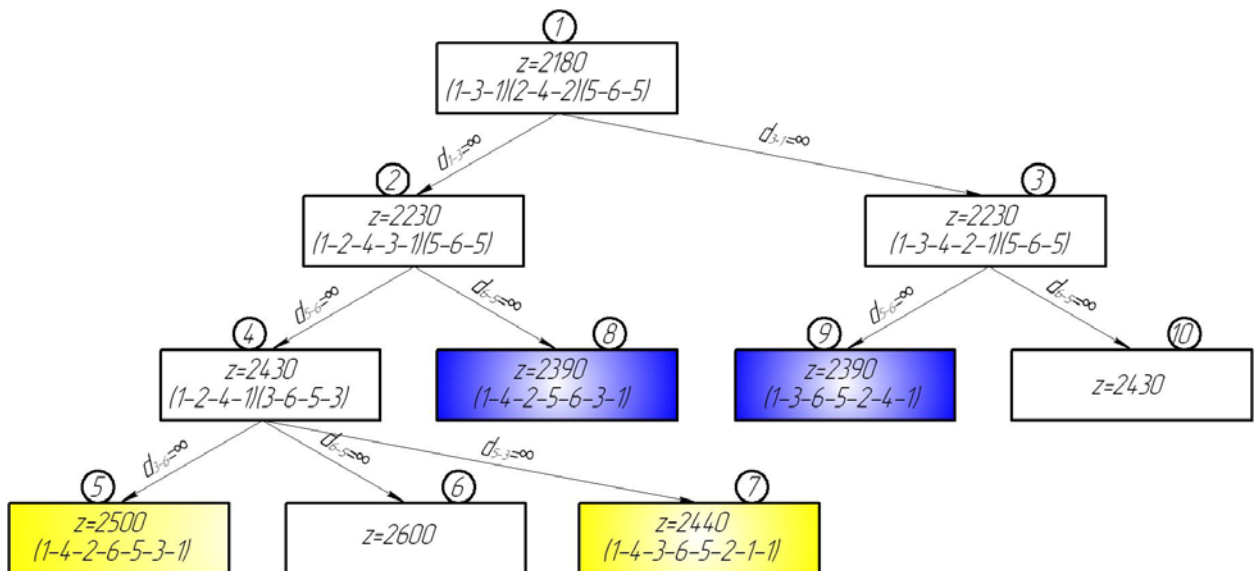


Рис. 7.8. Дерево підзадач рішення задачі комівояжера методом відгалужень і меж

Обираючи для видалення короткий цикл 1-3-1, одержуємо на дереві підзадач дві гілки, обумовлені умовами $x_{1-3} = 0$ ($d_{1-3} = \infty$) та $x_{3-1} = 0$ ($d_{3-1} = \infty$), що виходять із вузла 1.

Із двох наявних підзадач вирішимо спочатку задачу з умовою $x_{1-3} = 0$ ($d_{1-3} = \infty$) (вузол 2 на рис. 7.8) – порядок рішення підзадач довільний. Одержуємо рішення $z = 2230$ (1-2-4-3-1)(5-6-5), що формує два часткових цикли (1-2-4-3-1) і (5-6-5).

Обираючи для видалення короткого циклу 5-6-5, одержуємо на дереві підзадач дві гілки, обумовлені умовами $x_{5-6} = 0$ ($d_{5-6} = \infty$) та $x_{6-5} = 0$ ($d_{6-5} = \infty$), що виходять із вузла 2. Із двох наявних підзадач вирішимо спочатку задачу з умовою $x_{5-6} = 0$ ($d_{5-6} = \infty$) (вузол 4 на рис. 7.8) – порядок рішення підзадач довільний. Одержуємо рішення $z = 2430$ (1-2-4-1)(3-6-5-3), що формує два часткових цикли (1-2-4-1) і (3-6-5-3).

Обираємо далі для видалення короткого циклу 3-6-5-3, одержуємо на дереві підзадач три гілки, обумовлені умовами $d_{3-6} = \infty$, $d_{6-5} = \infty$ та $d_{5-3} = \infty$. Із трьох наявних підзадач вирішимо спочатку задачу з умовою $d_{3-6} = \infty$ (вузол 5 на рис. 7.8). Одержуємо рішення $z = 2500$ з повним циклом (1-4-2-6-5-3-1). Оскільки отримано рішення з повним циклом, то розгалуження у вузлі 5 не буде. Рішення підзадачі вузла 5 дозволяє поліпшити значення верхньої межі, $z = 2500$.

З нерозглянутих підзадач 6 й 7 вибираємо для рішення підзадачу 6. Одержуємо рішення цієї підзадачі з $z = 2600$, але це рішення більше ніж значення верхньої межі, таким чином розгалужень у вузлі 6 не буде.

Після вирішення підзадачі 7, одержуємо рішення $z = 2440$ і повним циклом 1-4-3-6-5-2-1. Це рішення пропонує нову верхню межу $z = 2440$.

Після вирішення підзадачі 8, одержуємо рішення $z = 2390$ і повним циклом 1-4-2-5-6-3-1. Це рішення пропонує нову верхню межу $z = 2390$.

Залишилася нерозглянутою тільки підзадача вузла 3. Для її рішення в даних вихідної задачі про призначення, покладемо $d_{3-1} = \infty$. Одержуємо рішення з $z = 2230$ (1-3-4-2-1)(5-6-5), що формує два часткових цикли (1-3-4-2-1) і (5-6-5).

Обираючи для видалення короткий цикл 5-6-5, одержуємо на дереві підзадач дві галузі, обумовлені умовами $x_{5-6} = 0$ ($d_{5-6} = \infty$) та $x_{6-5} = 0$ ($d_{6-5} = \infty$), що виходять із вузла 3. Із двох наявних підзадач вирішимо спочатку задачу з умовою $x_{5-6} = 0$ ($d_{5-6} = \infty$) (вузол 9 на рис. 7.8) – порядок рішення підзадач довільний. Одержуємо рішення $z = 2390$ з повним циклом 1-3-6-5-2-4-1, що відповідає верхній межі.

Залишилася нерозглянутою тільки під задача 10. Одержуємо рішення з $z = 2430$, але це рішення більше ніж значення верхньої межі, таким чином розгалужень у вузлі 10 не буде.

Нерозглянутих підзадач не залишилося. Оптимальним циклом буде 2 цикли: (1-4-2-5-6-3-1) та (1-3-6-5-2-4-1), які є однаковими за довжиною рівною 2390 км та однаковими самі по собі, які відрізняються напрямком слідування.

8. ФУНКЦІОНАЛЬНА ОЦІНКА СХЕМ СТАНЦІЙ

Питання побудови схем станційних пристроїв мають важливе теоретичне і практичне значення. Труднощі їх рішення пояснюються великою кількістю вимог до взаємного розміщення пристроїв у плані і профілі, необхідністю в кожному конкретному випадку більш повно враховувати розмаїтість місцевих умов; міркуваннями експлуатаційного характеру – забезпечення паралельності операцій, мінімуму пробігів вагонів і локомотивів, усунення складних кутових заїздів тощо.

У виборі найкращих схем велике значення має вже накопичений інженерами-проектувальниками практичний досвід, що допомагає відібрати схеми, найбільш ефективно і повно відповідному характеру станційних процесів.

Метод 1. Підрахунок кількості точок перетинання маршрутів. Кількість точок перетинання маршрутів у горловинах станції можна розглядати лише як досить наближений критерій якості схеми. Основними видами перетинань є: перетинання маршрутів прийому, маршрутів відправлення, маршрутів прийому і відправлення. Ці перетинання не рівнозначні: найбільш несприятливе перетинання маршрутів прийому потягів, тому що в цьому випадку виникають затримки потягів на перегонах.

Метод 2. Підрахунок сумарної тривалості заняття точок перетинання маршрутів. Для заданих розмірів руху по перетинанню N_1 й N_2 і тривалості заняття перетинання t_1 і t_2 визначається сума:

$$\sum_{i=1}^n N_i t_i = N_1 t_1 + N_2 t_2, \quad (8.1)$$

де $i = 1, 2, 3, \dots, n$ – число пересічних маршрутів.

Знайдена сума деякою мірою дає представлення про використання пропускну здатності й експлуатаційних достоїнств схеми.

Метод 3. Імовірнісний підрахунок тривалості затримок рухомого складу на перетинаннях. Перераховані вище способи не враховують тривалості затримок рухомого складу на перетинаннях.

У найпростішому випадку сумарна тривалість затримок по кожній точці перетинання дорівнює:

для рівноправних маршрутів

$$\sum^{N_1+N_2} T_{1,2} = \frac{N_1 N_2}{2T_p} (t_1^2 + t_2^2);$$

для нерівноправних маршрутів

$$\sum^{N_2} T_{1,3} = \frac{N_1 N_2}{2T_p} (t_1 + t_2)^2,$$

де T_p – тривалість розрахункового періоду.

Для більш складних випадків запропоновані відповідні формули, що дають можливість орієнтовно оцінити величину очікуваних затримок рухомого складу. Однак порівняння розрахункових значень затримок з результатами моделювання показує на значні погрішності розрахункових значень, звичайно в меншу сторону.

Схема шляхового розвитку станцій багато в чому визначається розташуванням на мережі і характером операцій, виконуваних на спеціалізованих групах шляхів.

Головні шляхи розташовуються в зоні, де здійснюється рух організованих потягів. Поточковий прийом потягів з перегонів на шляхи зон станційного обслуговування і зручне відправлення поїздів з цих шляхів на перегони є неодмінною передумовою раціонального проектування внутрішньостанційних з'єднань, що враховує характер виконуваних операцій у тій або іншій частині станції.

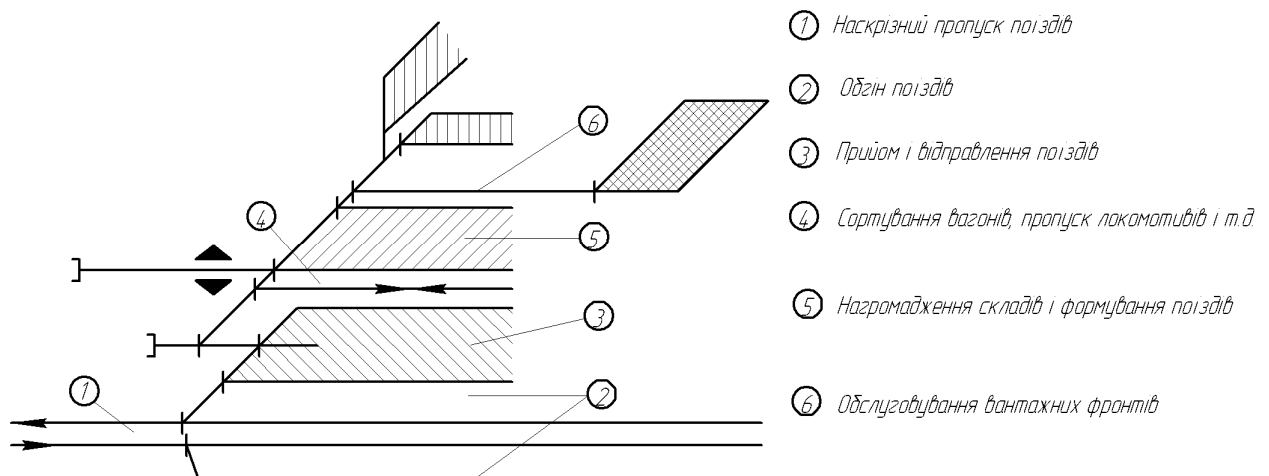


Рис. 8.1. Схема зональності шляхового розвитку станцій

Таким чином, при побудові схем шляхового розвитку великої уваги заслуговує формування найголовніших зон станцій, з перевіркою кожного елемента зони індивідуально на відповідність експлуатаційним вимогам, виходячи з характеру й обсягу виконуваних на станції операцій.

Проф. М. Поттгоффом запропонований спосіб оцінки схем шляхового розвитку, заснований на принципах теорії інформації. Суть способу полягає у визначенні ентропії шляхового розвитку

$$H = - \sum_{x=1}^n p_x \log_2 p_x, \quad (8.2)$$

де n – число шляхів у парку; p_x – імовірність проходження відчеплення на шлях x .

Якщо говорити про схему станції в цілому, то більш правильно було б оперувати не окремими шляхами і стрілочними переводами, а основними маршрутами руху – поїзними і маневровими. Коли всі маршрути розв'язані і перетинання відсутні, умовно можна вважати, що станція має «просту» схему й умови її роботи залежать не від схеми, а від «матеріального наповнення», тобто кількості станційних колій, пристроїв СЦБ, технологічних факторів. Кількісною мірою складності в цьому випадку буде показник

$$\mu_s = \frac{\sum_{x=1}^{x=s} \eta_x \log_2 \eta_x}{\sum_{x=1}^2 \eta_x \log_2 \eta_x}, \quad (8.3)$$

де s – фактична кількість конфліктних точок; n – кількість маршрутів (груп технологічних ланцюгів); η_x – імовірність надходження заявки на маршрут x .

Чисельник вираження (8.3) представляє ентропію ворожих пересувань, знаменник – критичну ентропію схеми.

В курсовому проекті пропонується розрахувати показник μ_s станції, яка має найменшу пропускну здатність (або станції, на яку вказує керівник проекту), тобто, що лімітує максимальний потік заданої мережі. Розглянемо вирішення даної задачі на прикладі схеми мережі, що була розглянута у попередніх задачах. В даній мережі найменшу пропускну здатність має станція 27 – Кіровоград (9). Розглянемо один з шляхів збільшення пропускну здатності цієї станції (див. рис. 8.2).

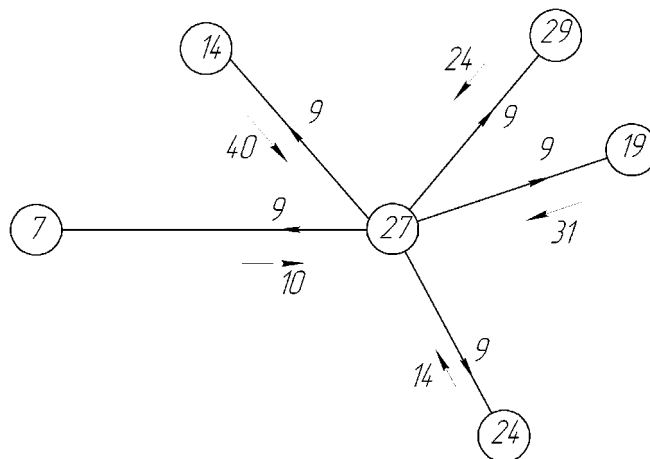


Рис. 8.2. Пропускна здатність станції 27

Спростимо схему станції до заданого вигляду (див. рис. 8.3). Схема станції, до якої необхідно привести дійсну схему видається керівником курсового проекту.

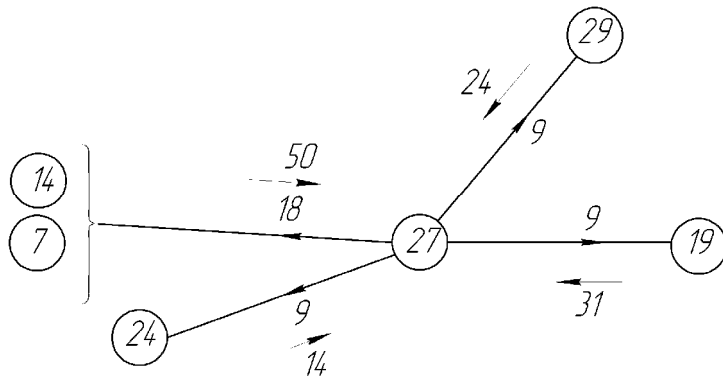


Рис. 8.3. Спрощена схема розподілу пропускної здатності

Приклад функціональної оцінки схеми станції. Визначимо показник μ_s на прикладі однієї зі схем при різному ступені ворожості маршрутів (рис. 8.4).

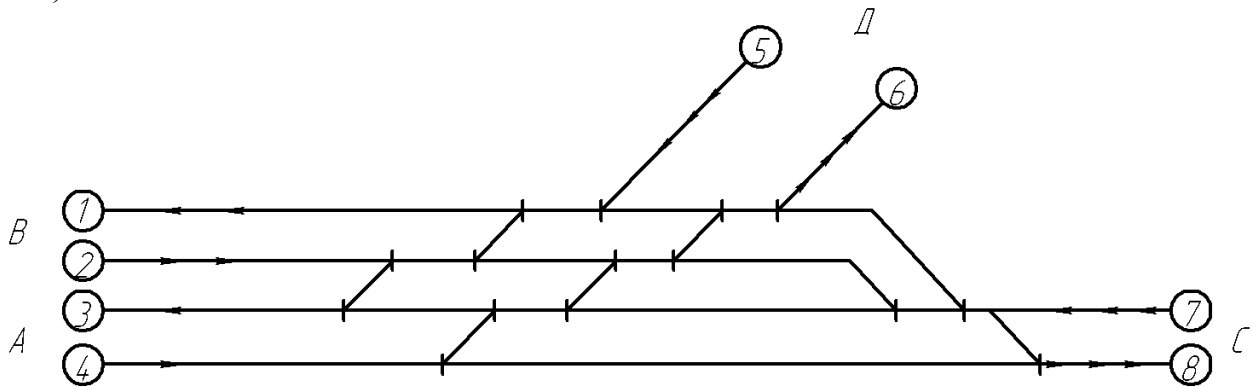


Рис. 8.4. Схема станції для функціональної оцінки

Спростимо та узагальнимо схему станції. У першому випадку (схема I) маємо шість точок ворожих перетинань, у другому чотири, у третьому – дві.

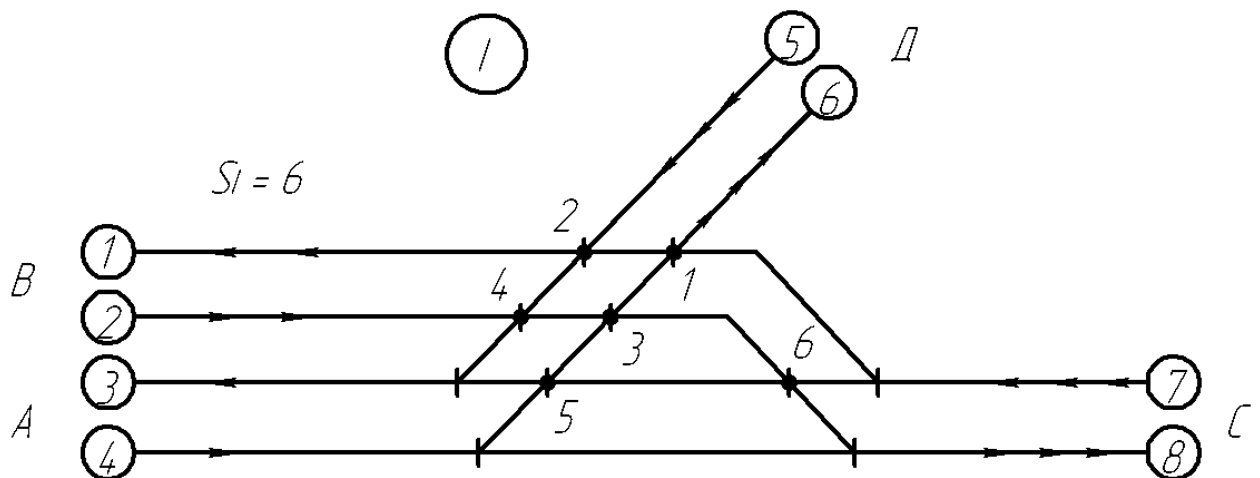


Рис. 8.5. Схема до визначення ступеня ворожості маршрутів при шістьох точках ворожих перетинань

У першому наближенні розраховуємо максимальну пропускну здатність заданої схеми. Для заданої схеми колії 1, 2 відносяться до потоку станцій 7 і 14, колії 3,4 – станції 24, колії 5,6 – станції 29, колії 7,8 – станції 19.

Найбільша кількість перетинань потоків буде в точці 1(потік з 7 колії на 1, потоки з 2 і 4 колій на 6), тобто завантаження точки 1 по потоку буде сума трьох потоків. Але, тому що потік на першу колію буде потім розподілено між 7 і 14 станціями, то його подвоюємо, а, також потік, що приходить на другу колію складається з потягів 7 і 14 станцій, то його також подвоюємо, отже в точці 1 будемо мати суму п'ятьох потоків. Виходячи з того, що час заняття перетинання одним потягом однаковий для всіх напрямків максимальна пропускну здатність перетинання дорівнює 60 потягам [7]. Вважаємо, що кількість потягів у кожному напрямку однакова, і дорівнює 12 потягам (60 потягів розподілено на п'ять потоків).

Але, варто не забувати про те, що при розрахунках реальної пропускну здатності, звичайно залишають деякий резерв пропускну здатності, що враховує ймовірність одночасного підходу поїздів і різних затримок поїздів на інших ділянках. У загальному випадку резервується приблизно 10-30% пропускну здатності. Таким чином, при розрахунку реальної пропускну здатності, з урахуванням поправок ми одержимо не 12 поїздів у кожному потоці, а $12 \cdot (0,7...0,9) = 8...11$, і в такому випадку можна вважати, що пропускну здатність у кожному потоці відповідає приблизно 9 поїздам, що є пропускну здатністю заданої станції (Кіровоград(9)). Але далі, до закінчення розрахунку функціональної оцінки схеми станції ми це враховувати не будемо, а зробимо це наприкінці рішення.

У табл. 8.1 приведена кількість потягів, що обслуговується.

Т а б л и ц я 8.1

Характеристика потоку

3/На	1	2	3	4	5	6	7	8
1	–
2	.	–	.	.	.	24	.	24
3	.	.	–
4	.	.	.	–	.	12	.	12
5	24	.	12	.	–	.	.	.
6	–	.	.
7	24	.	12	.	.	.	–	.
8	–

Загальна кількість потягів дорівнює $24 + 24 + 12 + 12 + 24 + 24 + 12 + 12 = 12 \cdot 12 = 144$.

У табл. 8.2 приведена характеристика структури поїздопотоків, що обслуговується, тобто розподіл сумарного потоку, прийнятого за одиницю, по маршрутах руху.

Т а б л и ц я 8.2

Характеристика структури потоку (у частках одиниці)

3/На	1	2	3	4	5	6	7	8
1	—
2	.	—	.	.	.	0,167	.	0,167
3	.	.	—
4	.	.	.	—	.	0,084	.	0,084
5	0,167	.	0,084	.	—	.	.	.
6	—	.	.
7	0,167	.	0,084	.	.	.	—	.
8	—

З даної таблиці для кожної з точок перетинання маршрутів визначені величини η_x (табл. 8.3). Порядок обчислення цих величин наступний. Відповідно до прийнятої схеми й розподілу руху по маршрутах установлюється завантаження кожної з розглянутих точок (див. рис. 8.6) шляхом підсумовування потоків, що накладаються.

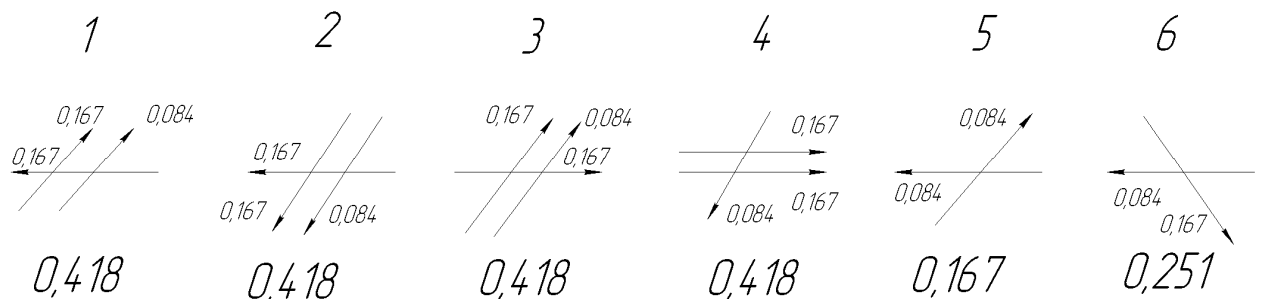


Рис. 8.6. Перетинання в точках

Т а б л и ц я 8.3.

Значення η_x		1	2	3	4	5	6	Σ
Точки								
η_x по потоці		0,418	0,418	0,418	0,418	0,167	0,251	2,09
по точках		0,2	0,2	0,2	0,2	0,08	0,12	1,00

Аналогічно підраховуються відповідні величини для інших точок схеми. Далі визначається сумарне завантаження потоками всіх точок перетинання, при цьому величини η_x по потоках виражаються також у частках одиниці. Таким чином, маються всі необхідні дані для підрахунку μ_s для схеми I

$$\begin{aligned} \sum_{x=1}^{s=6} \eta_x \log_2 \eta_x &= 4 \cdot 0,2 \log_2 0,2 + 0,08 \log_2 0,08 + 0,12 \log_2 0,12 = \\ &= -4 \cdot 0,2 \cdot 2,322 - 0,08 \cdot 3,64 - 0,12 \cdot 3,059 = \\ &= -(1,858 + 0,291 + 0,367) = -2,516 \end{aligned}$$

для якої показник $\mu_{s_1} = \frac{\sum_{x=1}^{s=6} \eta_x \log_2 \eta_x}{\sum_{x=1}^2 \eta_x \log_2 \eta_x} = \frac{-2,516}{-2,516} = 1,00$.

Розглянемо схему II із чотирма точками ворожих перетинань (див. рис. 8.7)

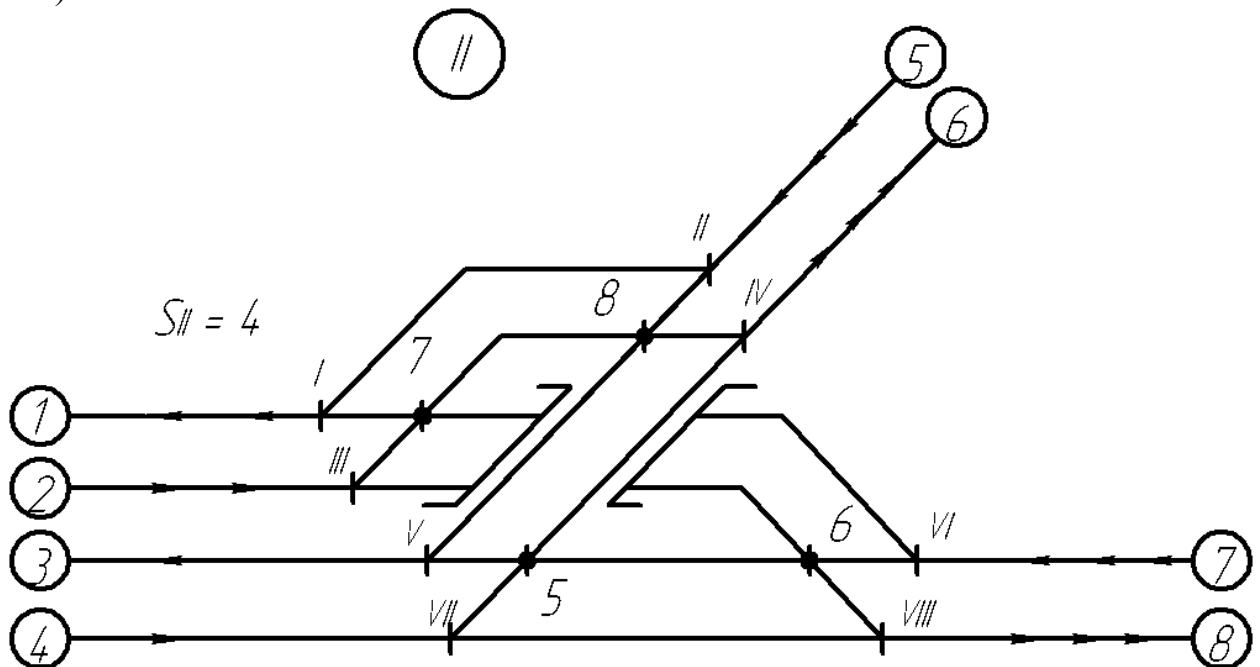


Рис. 8.7. Схема до визначення ступеня ворожості маршрутів при чотирьох точках ворожих перетинань

Користуючись таблицею 8.2 для кожної з точок перетинання маршрутів визначаємо величини η_x (табл. 8.4). Порядок обчислення цих величин аналогічний розрахунку для першої схеми.

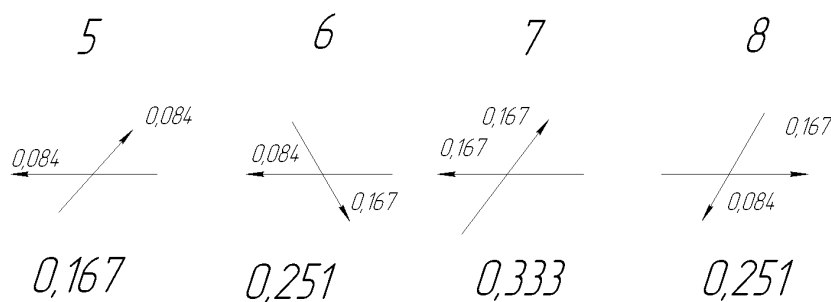


Рис. 8.8. Перетинання в точках для схеми II

Т а б л и ц а 8.4.

Значення η_x

Точки	5	6	7	8	Σ
η_x по потоці	0,167	0,251	0,333	0,251	1,00
по точках	0,167	0,251	0,333	0,251	1,00

Таким чином, маються всі необхідні дані для підрахунку μ_s для схеми II:

$$\begin{aligned} \sum_{x=1}^{s=4} \eta_x \log_2 \eta_x &= 0,167 \log_2 0,167 + 2 \cdot 0,251 \log_2 0,251 + 0,333 \log_2 0,333 = \\ &= -0,167 \cdot 2,582 - 2 \cdot 0,251 \cdot 1,994 - 0,333 \cdot 1,586 = \\ &= -(0,431 + 1,00 + 0,528) = -1,959 \end{aligned}$$

$$\text{для якої показник } \mu_{s_{II}} = \frac{\sum_{x=1}^{s=4} \eta_x \log_2 \eta_x}{\sum_{x=1}^2 \eta_x \log_2 \eta_x} = \frac{-1,959}{-2,516} = 0,779.$$

Таким чином, показник $\mu_{s_{II}} = 0,779$ знизився на 12,1 %. Точка 7 при схемі II має найбільшу завантаженість – 0,333, що відповідає $0,333 \cdot 144 = 48$ потягам. Але максимальна пропускна здатність ворожого перетинання – 60 потягів, тобто ми можемо збільшити наявну пропускна здатність на $60 - 48 = 12$ потягів. Загальна кількість потягів буде дорівнювати $144 \cdot \frac{60}{48} = 180$ потягів. Тепер треба перевірити точки I, II, III, IV, V, VI, VII, VIII чи забезпечують вони пропуск збільшеної кількості потягів. Згідно завдання

стрілковий перевід має пропускну здатність 160 потягів. Розраховуємо кількість потягів з обліком підвищення кожного потоку в $\frac{60}{48} = 1,25$ рази.

Точка $I = 30 + 30 = 60$ потягів < 160
 $II = 30 + 15 = 45$ потягів < 160
 $III = 30 + 30 = 60$ потягів < 160
 $IV = 30 + 15 = 45$ потягів < 160
 $V = 15 + 15 = 30$ потягів < 160
 $VI = 30 + 15 = 45$ потягів < 160
 $VII = 15 + 15 = 30$ потягів < 160
 $VIII = 30 + 15 = 45$ потягів < 160

Таким чином, умову забезпечення збільшеної кількості потягів через точки $I, II, III, IV, V, VI, VII, VIII$ виконано, і побудовою шляхопроводу можна досягти збільшення пропускну здатності вузла на 25 % при зменшенні показника $\mu_{сп}$ на 12%, що призводить до зменшення витрат на обслуговування і експлуатацію постійних пристроїв розв'язки.

Розглянемо схему III із двома точками ворожих перетинань (див. рис. 8.9)

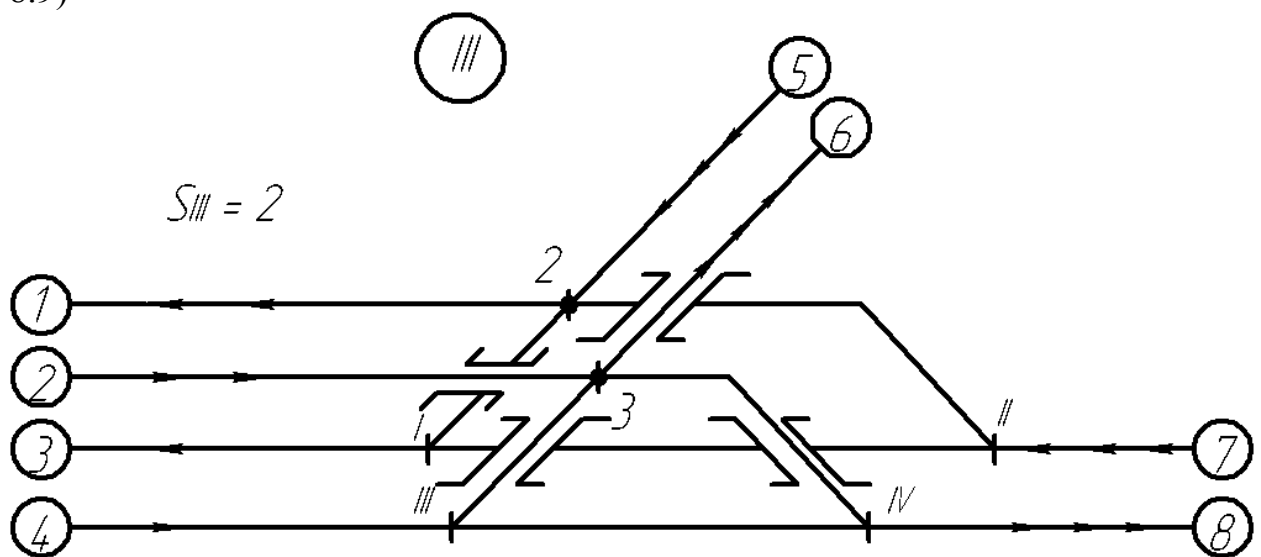


Рис. 8.9. Схема до визначення ступеня ворожості маршрутів при двох точках ворожих перетинань

Користуючись таблицею 8.2 для кожної з точок перетинань маршрутів визначаємо величини η_x (табл. 8.5). Порядок обчислення цих величин аналогічний розрахунку для першої схеми.

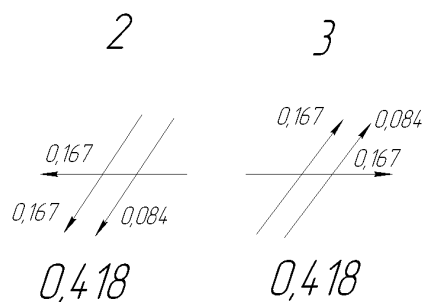


Рис. 8.9. Перетинання в точках для схеми III

Т а б л и ц я 8.5.

Значення η_x

Точки	2	3	Σ
η_x по потоці	0,418	0,418	0,836
по точках	0,5	0,5	1,00

Таким чином, маються всі необхідні дані для підрахунку μ_s для схеми III:

$$\sum_{x=1}^{s=2} \eta_x \log_2 \eta_x = 2 \cdot 0,5 \log_2 0,5 = -2 \cdot 0,5 \cdot 1 = -1$$

$$\text{для якої показник } \mu_{s_{III}} = \frac{\sum_{x=1}^{s=2} \eta_x \log_2 \eta_x}{\sum_{x=1}^2 \eta_x \log_2 \eta_x} = \frac{-1}{-2,516} = 0,397.$$

Таким чином, показник $\mu_{s_{III}} = 0,397$ знизився на 60 % у порівнянні з вихідною схемою. Точки 2 і 3 при схемі III мають найбільшу завантаженість – 0,418, що відповідає $0,418 \cdot 144 = 60$ потягам. Тобто максимальну пропускну здатність вичерпано для шляхів з 5 на 3, з 5 на 1, з 7 на 1, з 4 на 6, з 2 на 6 і з 2 на 8.

Таким чином, для схеми III побудовою чотирьох шляхопроводів можна досягти лише зменшення показника $\mu_{s_{III}}$ на 60%, що призводить до зменшення витрат на обслуговування і експлуатацію постійних пристроїв розв'язки, але ці витрати навряд сягають витрат на побудову шляхопроводів.

Розглянемо економічну доцільність побудови шляхопроводів для схеми II.

Приведені будівельні витрати (виходячи зі строку окупності 10 років) на спорудження одноколійного шляхопроводу стандартної довжини становлять 0,17 млн. умов. одиниць на рік.

Приведені будівельні витрати (виходячи зі строку окупності 10 років) на спорудження двоколійного шляхопроводу стандартної довжини становлять 0,22 млн. умов. одиниць у рік.

Річна економія від ліквідації затримок на перетинаннях залежить від типу локомотива й ваги поїзда, але умовно її можна розрахувати по формулі:

$$E_z = An^2,$$

де A – коефіцієнт пропорційності, що враховує тип локомотива, вагу поїзду, вартість години затримки локомотива й години затримки поїзда (зазначений у завданні до курсового проекту).

n – кількість поїздів, на яку збільшено пропускну здатність вузла.

Щорічні експлуатаційні витрати на обслуговування колійного розвитку з коефіцієнтом $\mu = 1$ можна розрахувати по формулі:

$$E_e = B + Cn^2,$$

де B – витрати, що не залежать від кількості поїздів, що обслуговуються, C – коефіцієнт пропорційності. Коефіцієнт C пов'язаний з коефіцієнтами μ й D (зазначений у завданні до курсового проекту) залежністю:

$$C = D(\mu^2 - e^{\mu-1} + 1)$$

Розглянемо доцільність будівлі шляхопровідних розв'язок на прикладі розрахованої схеми II:

Для схеми II експлуатаційні витрати до побудови шляхопроводів ($\mu_I = 1,0$) і економія від ліквідації затримок складуть:

$$\begin{aligned} E &= A(n - n_{uzla})^2 + B + Cn^2 = B + (A + C)n^2 + An_{uzla}^2 - 2Ann_{uzla} = \\ &= B + An_{uzla}^2 - 2Ann_{uzla} + (A + D(\mu_I^2 - e^{\mu_I-1} + 1))n^2 \end{aligned}$$

Витрати на спорудження двоколійного шляхопроводу й експлуатаційні витрати на обслуговування колійного розвитку після його будівлі ($\mu_{II} = 0,779$), у тис. умов. од.

$$E_p = 220 + B + Cn^2 = 220 + B + D(\mu_{II}^2 - e^{\mu_{II}-1} + 1)n^2.$$

Для спрощення рішення завдання побудуємо графіки залежності $E = f(n)$ й $E_p = f(n)$, при $A = 0,2$; $B = 10$; $D = 0,05$; $n_{uzla} = 144$.

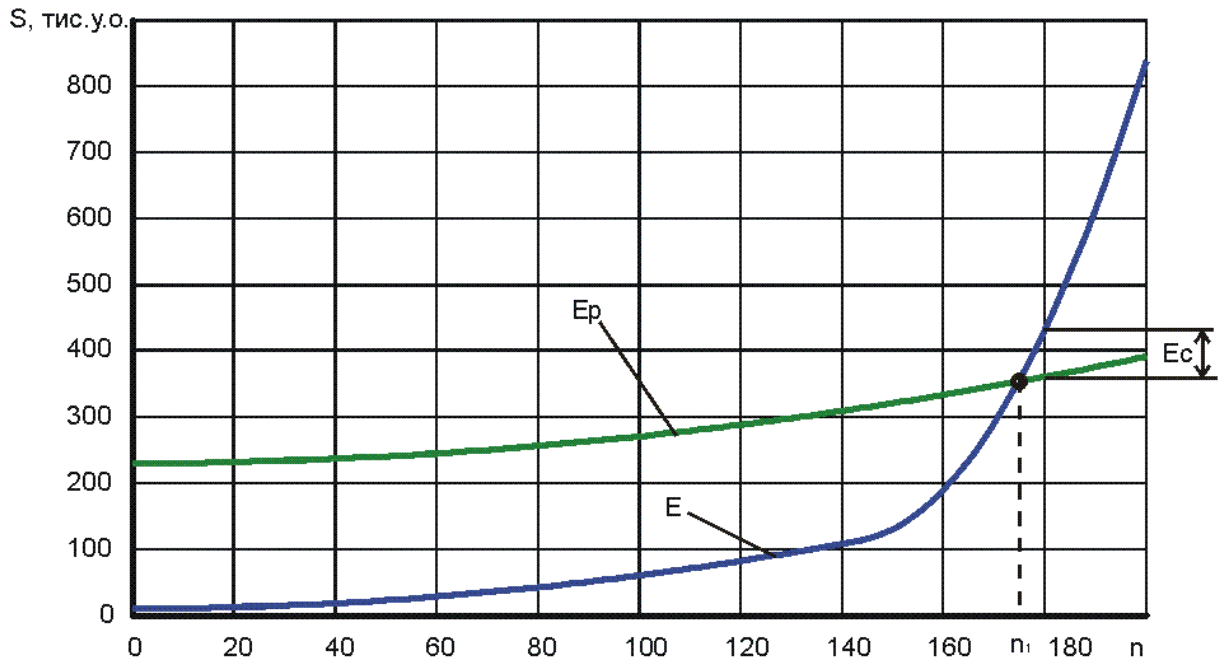


Рис. 8.10. Економічна доцільність спорудження шляхопроводу при зміні схеми вузла за схемою II.

Як видно з рис. 8.10 вигідність шляхопровідної розв'язки настає при $n > n_1 = 175$ поїздів на добу, що проходять через вузол. Економія від використання схеми II, при збільшенні пропускної здатності до 180 поїздів складе $E_c = 70$ тис. у.о. на рік при зменшенні ентропії ворожих перетинань.

Тому що при використанні схеми III пропускна здатність станції не збільшується, то спорудження чотирьох шляхопроводів є економічно необґрунтованим. Але, внаслідок того, що зменшується показник μ , зменшується ступінь ворожості маршрутів, що приводить до поліпшення безпеки руху.

ЛІТЕРАТУРА

1. Иванов Д.А. Управление цепями поставок / Д.А. Иванов. – Спб.: Изд-во Политехн. ун-та, 2009. – 660 с.
2. Уотерс Д. Логистика. Управление цепями поставок: Пер. с англ. – М.: ЮНИТИ-ДАНА, 2003. – 503 с.
3. Моделирование транспортных систем. Персианов В.А., Скалов К.Ю., Усков Н.С., М-изд-во «Транспорт» 1992г., – 209 с.
4. Нелинейные сетевые транспортные задачи. Левит Б.Ю., Лившиц В.Н. Институт комплексных транспортных проблем. Изд-во «Транспорт», 2002 – 257с.
5. Левин Д.Ю. Оптимизация потоков поездов. – М.: Транспорт, 1988. – 175 с.
6. Хорафас Д.Н. Системы и моделирование. Перевод с англ. Изд-во «Мир», М., 1967.
7. Хейт Ф. Математическая теория транспортных потоков. Перевод с англ. Изд-во «Мир», М., 1966.
8. Нестеров Е.П. Транспортные задачи линейного программирования. М.: Изд-во «Транспорт», – 1971.
9. Корнаков А.М. Развязки железнодорожных линий в узлах. М.:Изд-во «Изд.-полигр. объединение МПС», – 1962. – 155 с.
10. Таха Хемди А. Введение в исследование операций, 7-е изд.: Пер. с англ. / М.: Издательский дом «Вильямс», 2005. – 912 с.

ДОДАТОК А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені Володимира Даля
ННІ ТРАНСПОРТУ ТА БУДІВНИЦТВА

Кафедра Логістичне управління та безпека руху на транспорті

КУРСОВИЙ ПРОЕКТ

з дисципліни **Управління ланцюгом постачань**

Тема: **Моделювання транспортної мережі**

Здобувач
(прізвище, ініціали) (підпис)

Група **ОПЗТ-19дм** Варіант **01**

Керівник роботи **доц. Баранов І.О.**

2019 р.

Захищений з оцінкою

Комісія:

.....
(посада, прізвище, ініціали) (підпис)

.....
(посада, прізвище, ініціали) (підпис)

.....
(посада, прізвище, ініціали) (підпис)

Дата **25.12.19.**

ДОДАТОК Б

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені Володимира Даля
ННІ ТРАНСПОРТУ ТА БУДІВНИЦТВА
Кафедра “Логістичне управління та безпека руху на транспорті”

"ЗАТВЕРДЖУЮ"

*Зав. кафедрою проф. Чернецька-Білецька Н.Б.*_____

Курс 1 Група ОПЗТ-19дм Семестр 1
Спеціальність – 275 «Транспортні технології (за видами)»
Дисципліна " Управління ланцюгом постачань "

З А В Д А Н Н Я
на курсовий проект

1. Здобувач

2. Тема проекту: Моделювання транспортної мережі

3. Строк здачі проекту 25.12

4. Початкові дані: Карта України,
варіанти транспортної мережі і її вантажопотоки

5. Зміст розрахунково-пояснювальної записки:

1. Загальні питання побудови моделі транспортної мережі.
2. Характеристика аналізованої транспортної мережі.
3. Теоретичні положення по організації моделювання транспортних потоків.
4. Рішення задач моделювання транспортних потоків.
5. Пропозиції по удосконаленню роботи досліджуваної транспортної мережі.

6. Перелік графічних матеріалів (з визначенням обов'язкових креслень):

1. Транспортна мережа з методикою визначення найкоротшої відстані між джерелом і стоком (алгоритм Дейкстри).
2. Рішення задачі про найкоротші відстані (алгоритм Флойда).
3. Рішення задачі про максимальний потік (алгоритм Форда-Фалкерсона)

7. Дата видачі завдання

01.10

КАЛЕНДАРНИЙ ПЛАН

№, п/п	Етап роботи	Термін виконання	Примітка
1	Побудова моделі транспортної мережі	01.10-15.10	
2	Рішення задачі визначення найкоротшої відстані між джерелом і стоком	15.10-29.10	
3	Рішення задачі про найкоротші відстані	29.10-10.11	
4	Рішення задачі про максимальний потік	10.11-25.11	
5	Креслення графічного матеріалу	25.11-05.12	
6	Оформлення розрахунково-пояснювальної записки	05.12-21.12	
7	Захист курсового проекту	25.12	

Здобувач _____
(підпис) П.І.Б.

Керівник доц. Баранов І.О.
_____ (підпис) П.І.Б.

" 01 " 10 " 2019 р.

Додаток Г

Вихідні дані до курсового проекту

№ вузла	Станція	Проп. здатність	№ вузла	Станція	Проп. здатність
1	Вінниця	12	21	Ромодан	18
2	Житомир	13	22	Нікополь	21
3	Довжанська	7	23	Одеса	22
4	Хмельницький	20	24	Херсон	14
5	Івано-Франківськ	4	25	Джанкой	31
6	Луцьк	27	26	Ніжин	16
7	Рівне	5	27	Кіровоград	9
8	Львів	23	28	Тернопіль	15
9	П'ятихатки	11	29	Дніпропетровськ	24
10	Ковель	8	30	Полтава	16
11	Чоп	24	31	Лозова	22
12	Шепетівка	7	32	Мерефа	9
13	Коростень	11	33	Харків-Пас.	16
14	Київ-Пас.	40	34	Белгород	27
15	Сарни	10	35	Красноармійськ	5
16	Новгород-Волинський	13	36	Верхній Токмак	9
17	Чернівці	15	37	Ужгород	19
18	Луганськ	23	38	Дебальцеве	28
19	Запоріжжя	31	39	Чернігів	12
20	Гомель	27	40	Миколаїв	19

Варіанти для виконання курсового проекту.

№ вар	№ станції								
1	2	11	3	17	14	18	23	33	15
2	7	8	3	18	9	23	36	22	34
3	35	15	30	39	24	37	26	32	16
4	11	12	19	7	28	5	2	20	26
5	40	25	36	37	13	8	32	5	41
6	10	22	32	6	25	21	15	11	17
7	15	31	2	22	3	21	23	12	38
8	18	1	25	10	14	24	30	5	23
9	32	27	25	19	29	18	33	8	14
10	5	32	40	18	16	23	39	20	13
11	35	8	26	10	21	3	25	9	32
12	19	1	3	21	24	10	36	5	18
13	10	33	20	12	18	17	19	37	38
14	5	7	9	29	8	33	12	25	27
15	38	31	35	28	13	7	30	16	27
16	6	25	24	22	23	33	17	10	38
17	14	36	28	31	2	15	1	32	16
18	15	16	28	30	7	10	9	31	37
19	22	15	12	9	8	17	5	11	18
20	1	23	9	34	25	17	7	31	3
21	20	1	12	28	29	36	39	37	7
22	29	34	35	17	7	19	3	30	27
23	11	27	14	12	17	23	13	9	21
24	23	15	4	40	19	27	22	9	32
25	27	11	34	29	5	6	12	28	1
26	34	7	2	3	18	17	22	30	5
27	25	1	15	17	20	2	10	12	29
28	29	6	15	21	10	27	30	34	28
29	14	32	3	7	9	27	18	8	26
30	34	36	3	24	6	16	20	30	19
31	37	13	4	24	33	39	12	19	25
32	17	13	20	14	23	24	15	9	5
33	23	12	24	36	18	25	21	5	39
34	15	24	22	10	23	31	30	26	4
35	12	26	23	13	16	11	27	1	34
36	22	40	34	12	27	14	23	30	36
37	32	7	11	14	40	35	10	26	18
38	12	30	32	5	31	2	38	18	3
39	11	25	20	12	29	10	8	6	4
40	26	9	11	16	8	24	12	25	22
41	16	35	6	15	23	13	5	34	24
42	36	2	28	10	14	22	6	4	34
43	39	1	17	7	30	27	25	34	37
44	23	29	15	13	6	23	3	16	27
45	20	10	7	15	27	14	16	24	30
46	28	14	22	4	8	20	2	34	23
47	14	36	7	11	37	20	27	15	26
48	38	15	31	23	36	25	16	21	11
49	3	9	39	1	8	20	2	29	12
50	5	34	14	38	11	23	6	35	17