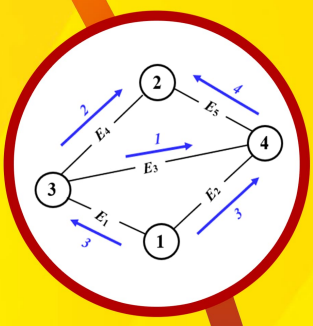
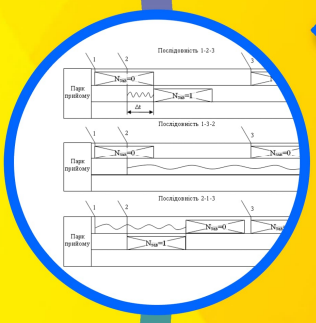
І. Я. СКОВРОН, А. С. ДОРОШ, Є. Б. ДЕМЧЕНКО
О. О. МАЗУРЕНКО, В. В. МАЛАШКІН, Т. В. БОЛВАНОВСЬКА

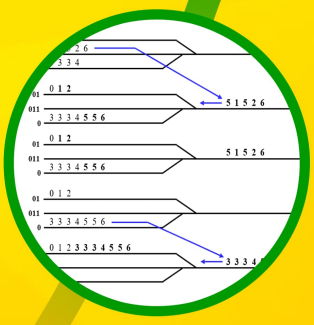




МЕТОДИ ОПТИМІЗАЦІЇ ПАРАМЕТРІВ ТРАНСПОРТНИХ СИСТЕМ

БАЗОВІ МЕТОДИ: ПРИКЛАДИ ТА ЗАДАЧІ

НАВЧАЛЬНИЙ ПОСІБНИК



ДНІПРО
2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

І. Я. Сковрон, А. С. Дорош, Є. Б. Демченко,
О. О. Мазуренко, В. В. Малашкін, Т. В. Болвановська

Методи оптимізації параметрів транспортних систем. Базові методи : приклади та задачі

НАВЧАЛЬНИЙ ПОСІБНИК

ДНІПРО
2026

УДК 656.1/.5:519.863(075.8)
М 54

Авторський колектив:

Сковрон І. Я., Дорош А. С., Демченко Є. Б.,
Мазуренко О. О., Малашкін В. В., Болвановська Т. В.

Рекомендовано до видання Радою якості
освітньої діяльності УДУНТ
Протокол № 1 від 16.09.2025

М 54 Методи оптимізації параметрів транспортних систем. Базові методи: приклади та задачі : навч. посіб. / І. Я. Сковрон, А. С. Дорош, Є. Б. Демченко, О. О. Мазуренко, В. В. Малашкін, Т. В. Болвановська ; Укр. держ. ун-т науки і технологій. – Електрон. вид. – Дніпро : УДУНТ, 2026. – 211 с.

ISBN 978-617-8314-42-2 (PDF)

У навчальному посібнику викладено теоретичні відомості про методи оптимізації транспортних систем та наведено приклади їх застосування для вирішення практичних задач удосконалення техніко-технологічних параметрів залізничного і автомобільного транспорту загального користування та транспорту промислових підприємств.

Навчальний посібник призначений для виконання практичних і самостійних робіт, курсового та дипломного проектування студентами, що навчаються у вищих навчальних закладах за спеціальностями J7 Залізничний транспорт та J8 Автомобільний транспорт і вивчають дисципліну «Методи оптимізації параметрів транспортних систем».

Іл. 97. Табл. 120. Бібліогр.: 27 назв.

УДК 656.1/.5:519.863(075.8)

Цей твір ліцензовано на умовах Ліцензії Creative Commons «Attribution-NonCommercial-ShareAlike» 4.0 International (CC BY-NC-SA 4.0) («Із зазначенням авторства – Некомерційна – Поширення на тих самих умовах» 4.0 Міжнародна)

ISBN 978-617-8314-42-2 (PDF)
DOI 10.15802/978-617-8314-42-2

© Сковрон І. Я., та ін., 2026
© Укр. держ. ун-т науки і технологій, 2026

UDC 656.1/.5:519.863 (075.8)
M 61

Writing Team:
Skovron I. Ya., Dorosh A. S., Demchenko Ye. B.,
Mazurenko O. O., Malashkin V. V., Bolvanovska T. V.

Recommended by the Council for the Quality of Educational Activities of the USUST
Protocol No. 1 of September 16, 2025

M 61 Methods of optimizing parameters of transport systems. Basics methods: examples and tasks : manual / I. Y. Skovron, A. S. Dorosh, Y. B. Demchenko, O. O. Mazurenko, V. V. Malashkin, T. V. Bolvanovska ; Ukrainian State University of Science and Technology. – Electronic edition – Dnipro : USUST, 2026. – 211 p.

ISBN 978-617-8314-42-2 (PDF)

The study guide presents theoretical information on methods for optimizing transport systems and provides examples of their application for solving practical problems related to improving the technical and technological parameters of mainline rail and road transport, as well as the transport of industrial enterprises.

The study guide is intended for practical work and self-study, as well as for coursework and diploma project development by students studying in higher education institutions under specialties J7 Railway Transport and J8 Road Transport and taking the course "Methods of Transport System Parameter Optimization".

Ill. 97. Tables 120. Bibliography: 27 titles.

UDC 656.1/.5:519.863 (075.8)

This work is licensed under Creative Commons License
«Attribution-NonCommercial-ShareAlike» 4.0 International (CC BY-NC-SA 4.0)

ISBN 978-617-8314-42-2 (PDF)
DOI 10.15802/978-617-8314-42-2

© Skovron I. Y. and other., 2026
© Ukrainian State University of Science and Technologies, 2026

ЗМІСТ

Вступ 5
Основні терміни та визначення понять 7
1. Загальні положення 11
1.1 Вступ 11
1.2 Основні поняття дисципліни 11
1.3 Математична модель операції 12
1.4 Загальна постановка задачі 14
2. Визначення найкоротшого маршруту на транспортній мережі 16
2.1 Теоретичні відомості 16
2.2 Методи пошуку найкоротшого маршруту 17
2.3. Приклад розв’язання задач. Задача 1 19
3. Оптимізація розподілу потоків на транспортній мережі 24
3.1 Теоретичні відомості 24
3.2 Приклад розв’язання задач. Задача 2 25
4. Розрахунок раціональної ширини пасажирських платформ 32
4.1 Теоретичні відомості 32
4.2 Приклад розв’язання задач. Задача 3 37
5. Сітьове планування та управління. Правила побудови сітьових графіків та розрахунок їх параметрів 41
5.1 Теоретичні відомості 41
5.2 Призначення і область застосування сітьових графіків 43
5.3 Основні поняття 45
5.4 Правила побудови сітьових графіків 46
5.5 Нумерація подій сітьового графіка методом викреслювання 48
5.6 Особливості сітьових графіків та їх параметрів 50
5.7 Розрахунок параметрів сітьових графіків 51
5.8 Приклад розв’язання задач. Задача 4 53
5.9 Приклад розв’язання задач. Задача 5 58
6. Оптимізація технології роботи методами сітьового планування та управління 63
6.1 Теоретичні відомості 63
6.2 Приклад розв’язання задач. Задача 6 64
6.2 Приклад розв’язання задач. Задача 7 73

7. Використання методики вирішення розподільчої задачі для оптимізації технологічних процесів 82
7.1 Теоретичні відомості 82
7.2 Математична постановка розподільчої задачі 83
7.3 Алгоритм розв’язання розподільчої задачі 84
7.4 Приклад розв’язання задач. Задача 8 88
8. Оптимізація техніко-технологічних параметрів транспортних систем із застосуванням методів динамічного програмування 98
8.1 Теоретичні відомості 98
8.2. Приклад розв’язання задач. Задача 9 100
9. Методи формування багатогрупних составів на залізничних станціях 116
9.1 Теоретичні відомості 116
9.2 Комбінаторний метод формування составів 117
9.3 Розподільчий метод формування составів 120
9.4 Приклад розв’язання задач. Задача 10 122
9.5 Приклад розв’язання задач. Задача 11 125
9.6 Умовна нумерація груп вагонів 128
10. Задача комівояжера 133
10.1 Теоретичні відомості 133
10.2 Короткий опис точних методів вирішення задачі комівояжера 136
10.3 Короткий опис наближених методів вирішення задачі комівояжера . 140
10.4 Приклад розв’язання задач. Задача 12 147
10.5 Приклад розв’язання задач. Задача 13 160
11. Багатокритеріальні задачі оптимізації та методи їх розв'язку 180
11.1 Вступ 180
11.2 Теоретичні відомості 180
11.3 Приклад розв’язання задач. Задача 14 185
12. Удосконалення техніко-технологічних параметрів транспортних систем з використанням векторної оптимізації 195
12.1 Теоретичні відомості 195
12.2 Приклад розв’язання задач. Задача 15 197
Література 208

Вступ

У сучасному світі, де швидкість і ефективність перевезень відіграють ключову роль у розвитку економіки та забезпеченні комфорту громадян, саме оптимізація параметрів транспортних систем стає одним із пріоритетних напрямків досліджень і практичної діяльності.

Транспортні системи складаються з різноманітних компонентів, таких як автомобільні дороги, залізниці та інші елементи інфраструктури, які спільно працюють для забезпечення руху товарів і пасажирів. Вони є складними багатокомпонентними системами, які включають не тільки фізичні об'єкти, але й процеси управління, планування, моніторингу та контролю.

Оптимізація транспортних систем — це не лише технічне завдання, але й стратегічне рішення, яке вимагає комплексного підходу та врахування численних факторів. Оптимізація параметрів таких систем дозволяє підвищити їх ефективність, знизити витрати, мінімізувати час перевезень і зменшити потенційний негативний вплив на навколишнє середовище [1, 2].

Навчальний посібник містить теоретичні відомості про ряд важливих методів оптимізації та їх практичне застосування до вирішення різного роду прикладних задач. При підготовці посібника автори керувалися вимогами законів України «Про вищу освіту» та «Про наукову і науково-технічну діяльність», а також офіційних нормативних документів Кабінету Міністрів України та Міністерства освіти і науки України.

Навчальний посібник підготовлено відповідно до робочої програми дисципліни «Методи оптимізації параметрів транспортних систем» для студентів освітнього рівня «магістр» за спеціальностями J7 Залізничний транспорт та J8 Автомобільний транспорт (за видами) та забезпечує такі програмні результати навчання (РН): РН-03 «*Приймати ефективні рішення у сфері транспортних систем і технологій з урахуванням технічних, соціальних, економічних та правових аспектів, генерувати і порівнювати альтернативи, оцінювати потрібні ресурси і обмеження, аналізувати ризики*»; РН-06 «*Розробляти нові та удосконалювати існуючі транспортні системи та технології, визначати цілі розробки, наявні обмеження,*

критерії ефективності та сфери використання при організації залізничних та автомобільних перевезень»; РН-07 «*Розробляти та аналізувати графічні, математичні та комп'ютерні моделі транспортних систем та технологій*».

Також даний навчальний посібник може бути використаний для виконання самостійних робіт, курсового та дипломного проектування.

Робота між співавторами даного посібника розподілена в рівних частках, тобто, наступним чином: І. Я. Сковрон, А. С. Дорош, Є. Б. Демченко, О. О. Мазуренко, В. В. Малашкін, Т. В. Болвановська – по 1,5 авторських аркуші (або по 16,667 %).

Основні терміни та визначення понять

Алгоритм A* (A-star) – це евристичний метод, що використовується в умовах, коли необхідно знайти найкоротший шлях швидко, навіть у дуже великих графах. *Він поєднує в собі переваги алгоритмів Дейкстри та пошуку в ширину, додаючи евристику, яка оцінює "вартість" проходження до кінцевої точки*.

Алгоритм Беллмана-Форда – метод пошуку найкоротшого шляху для графів з від'ємними вагами ребер, що здатний обробляти графи з негативними циклами.

Алгоритм Дейкстри – це один із найпоширеніших методів для знаходження найкоротшого шляху в графах з невід'ємними вагами ребер.

Алгоритм Флойда-Воршелла – це алгоритм, який дозволяє знайти найкоротші шляхи між всіма парами вершин у графі і підходить для випадків, коли потрібно провести багато запитів щодо найкоротших шляхів.

Бічна платформа – пасажирська платформа, яка розташована з одного боку від колії. Доступ до такої платформи може бути як в одному рівні (в тому числі і без перетину колій), так і в різних рівнях.

Вільний резерв – величина, яка показує допустиме збільшення тривалості роботи (або запізнення її початку), при якому її кінцева подія настає у свій ранній строк.

Вузли (вершини) графа – для транспортної мережі представляють собою місця або транспортні зупинки (міста, станції, порти, магазини, склади).

Динамічне програмування – це математичний метод оптимізації рішень, який пристосований до багатоетапних (багатокрокових) операцій.

Дійсна робота – це робота, яка вимагає витрат праці, матеріальних ресурсів і часу (наприклад – технічне обслуговування вагонів).

Елемент системи – найпростіша складова частини системи, яку умовно розглядають як неподільну.

Завершальна подія – це кінцева подія, яка означає досягнення кінцевої мети всього комплексу робіт**.**

Критичний шлях – це такий шлях, який має найбільшу тривалість від початкової події до завершальної, тобто це повний шлях найбільшої тривалості.

Маневрова робота – будь-які переміщення тягового залізничного рухомого складу з вагонами або без вагонів, що здійснюються під'їзними коліями, станційними коліями, та іншими коліями за виключенням руху поїздів, що прибувають з перегону і відправляються на перегін.

Маневровий состав – група вагонів, зчеплених між собою та з локомотивом, що проводить маневри.

Мета дослідження – кінцевий результат, на досягнення якого спрямоване дослідження.

Метод дослідження – спосіб застосування старого знання для здобуття нового знання. Він є засобом отримання наукових фактів.

Ребра (дуги) графа – для транспортної мережі представляють собою шляхи (автодороги, колії тощо), які з'єднують вузли графа між собою.

Наближені методи – це методи розв'язання задач, які не гарантують точного (оптимального) розв'язку, але дають достатньо хороший або прийнятний результат за менший час. Вони особливо корисні для складних або великих задач, де використання точних методів є надто дорогим з точки зору обчислювальних ресурсів.

Незалежний резерв – показує максимальний час, на який можна змінити тривалість виконання роботи, незалежно від строків її початкової та кінцевої події.

Об'єкт дослідження – процес або явище, що породжує проблемну ситуацію чи потребує більш детального знання.

Операція – будь-який захід (або система заходів), що об'єднаний з іншими єдиним задумом і спрямований на досягнення певної мети.

Оптимізація розподілу потоків на мережі – це комплексний процес визначення найбільш ефективних маршрутів для руху транспортних засобів, що спрямований на мінімізацію заторів, скорочення часу перебування в дорозі, зниження експлуатаційних витрат та покращення екологічної ситуації.

Острівна платформа – платформа, що розташована між двома коліями та обслуговує поїзди з обох боків. Доступ до такої платформи може бути в одному рівні (з перетином колій), так і в різних рівнях.

Очікування – це робота, яка не вимагає витрат праці і матеріальних ресурсів, але займає деякий час.

Під'їзні колії – залізничні колії, які належать підприємствам, підприємцям, організаціям та установам незалежно від форм власності, а також громадянам – суб'єктам підприємницької діяльності (далі – підприємство), з'єднані із загальною мережею залізниць безперервною рейковою колією і призначені для транспортного обслуговування одного або кількох підприємств у взаємодії із залізничним транспортом загального користування.

Підсистема – складова частина системи, яка сама є системою.

Повний резерв – величина, яка показує допустиме збільшення тривалості роботи (або запізнення її початку), при якому тривалість критичного шляху залишається незмінною.

Повний шлях – це такий шлях, початок якого збігається з початковою подією мережі, а кінець – із завершальною, тобто це будь-яка послідовність робіт, що поєднує початкову та кінцеву подію;

Поїзд – сформований і зчеплений состав вагонів з одним або декількома діючими локомотивами чи моторними вагонами, що мають встановлені сигнали. Локомотиви без вагонів, моторні вагони та спеціальний самохідний рухомий склад, що відправляються на перегін, вважаються поїздом.

Початкова подія – це момент початку виконання всього комплексу робіт.

Проміжна подія – це результат однієї або декількох робіт, що представляють можливість почати одну або кілька безпосередньо наступних робіт. Тривалість проміжної події в часі завжди дорівнює нулю (адже подія визначає стан, а не процес).

Система – множина взаємопов'язаних елементів, що працюють злагоджено та направлені на досягнення спільної мети.

Сітьовий графік – комплекс взаємозв'язаних робіт, що має вигляд мережі односторонніх дуг, які відповідають деяким операціям (роботам), а також вершин (подій), які відповідають початку і закінченню відповідних робіт.

Стан системи – сукупність значень її параметрів (властивостей) у певний момент часу.

Станція – роздільний пункт з колійним розвитком, що дозволяє проводити операції з приймання, відправлення, схрещення й обгону поїздів, операції з приймання, видачі вантажів та обслуговування пасажирів, а за умови розвинених колійних пристроїв – маневрову роботу з розформування та формування поїздів і технічні операції з поїздами.

Стрілочний перевід – пристрій, що служить для переведення рухомого складу з однієї колії на іншу. Стрілочні переводи складаються із стрілок, хрестовин і сполучних колій між ними. Хрестовини можуть бути з нерухомим або рухомим осердям.

Точні методи — це методи розв’язання задач, які дозволяють отримати точний (оптимальний) розв’язок у межах математичної моделі без використання наближень або спрощень. Такі методи зазвичай базуються на строгих математичних підходах і алгоритмах.

Фіктивна робота (залежність) – це робота, яка показує зв'язок між двома або більше подіями та яка не вимагає витрат праці, матеріальних ресурсів і часу, але вказує, що можливість початку однієї операції безпосередньо залежить від виконання іншої. Тривалість такої роботи дорівнює нулю.

Шлях між подіями *i* та *j* – шлях, що з'єднує будь-які дві події *i* та *j*, з яких жодна не є вихідною або завершальною подією сітьового графіка.

Шлях, наступний за подією – шлях, що з'єднує подію з завершальною подією.

Шлях, що передує події – шлях від вихідної події мережі до даної події.

РОЗДІЛ 1

Загальні положення

1.1 Вступ

Розробка технологічних процесів роботи об’єктів транспортної інфраструктури, графіка руху та плану формування поїздів, ефективна організація вагонопотоків, планування роботи підприємств транспорту неможливі без застосування математичного апарату. Математичні методи дозволяють врахувати складність технології, елементи випадковості в роботі і завдяки системному аналізу встановити ефективні режими функціонування транспортних систем [2 – 4].

Потреби виробництва в найкращому (оптимальному) управлінні складними цілеспрямованими процесами створили передумови для виникнення спеціальних наукових методів, які покликані забезпечувати кількісне обґрунтування рішень в усіх сферах цілеспрямованої людської діяльності.

1.2 Основні поняття дисципліни

Операція – будь-який захід (система заходів), що об’єднаний єдиним задумом і спрямований на досягнення певної мети. Операція завжди є керованим заходом, тобто існує можливість тим чи іншим способом обрати певні параметри, що характеризують спосіб її (операції) організації.

Рішення – певний обмежений набір керованих параметрів деякої заданої операції.

Для того, щоб оцінити ефективність кожного рішення та їх порівнювати, необхідно мати певний чисельний критерій оцінки (показник ефективності). Інколи вибір критерію ефективності являє собою досить складне завдання.

Критерій ефективності – кількісний (інколи якісний) показник, що характеризує ступінь досягнення поставленої мети при певній організації операції. Це може бути (цільова) функція або якесь розподілення.

Приклади критеріїв ефективності:

– що буде критерієм ефективності, якщо необхідно відсортувати состав збірного поїзда?

– що буде критерієм ефективності, якщо необхідно розподіляти порожній вагонопотік?

– що буде критерієм ефективності при виборі оптимального маршруту руху автомобіля з вантажем?

Обраний критерій повинен відповідати таким вимогам:

– показовість, тобто здатність відображати реальну мету операції;

– чутливість, тобто здатність змінюватись при зміні значень параметрів операції;

– простота визначення і розуміння;

– інтегральність, тобто об’єднання по можливості всіх основних елементів операції.

Оптимальне рішення – набір параметрів операції, які забезпечують екстремум (максимум чи мінімум) критерію ефективності при заданій системі обмежень.

Можливі дві форми знаходження оптимального рішення:

– досягнення заданого ефекту при мінімальних витратах ресурсів [F(R)→min];

– досягнення максимального ефекту при фіксованих витратах ресурсів [F(E)→max].

Класична постановка задач передбачає максимізацію критерію ефективності при пошуку оптимального рішення. При необхідності забезпечити мінімум критерію задача легко зводиться до задачі максимізації, якщо змінити знак цільової функції.

1.3 Математична модель операції

При пошуку оптимального рішення з організації операції можна досліджувати як саму операцію (реальний процес, об’єкт), так і її модель. У більшості випадків виконання досліджень на реальних об’єктах не представляється можливим. Давайте представимо, як провести таке

дослідження: зміна кількості вагонів у складі поїздів, зміна напрямку слідування вагонопотоків, зміна кількісного складу працівників бригад ПТО або ДСП на станції (взяли, а потім через годину звільнили).

В цьому зв'язку виникає проблема розробки адекватної (достовірної) моделі реального процесу. В рамках даної дисципліни ми будемо займатися дослідженням математичних моделей процесів та об'єктів. При цьому застосовується операційний метод, суть якого полягає у таких формулюваннях:

1) постановка задачі – формулювання умов та вибір критерію оцінки;

2) побудова математичної моделі операції (формалізація задачі);

3) знаходження оптимального рішення за допомогою моделі;

4) перевірка моделі і оцінка отриманого рішення.

Формалізація передбачає представлення вербального (словесного) запису задачі у вигляді системи аналітичних (математичних) виразів. Така система і являє собою **математичну модель** [2, 3, 5].

Побудова моделі операції представляє собою з одного боку складну, а з іншого – відповідальну задачу. Загальних способів побудови моделей не існує. В кожному конкретному випадку модель будується, виходячи з фізичного змісту задачі з урахуванням цільової спрямованості дослідження та здорового глузду.

Вимоги до моделі досить суперечливі. З одного боку, модель повинна бути досить повною, тобто в ній необхідно врахувати всі важливі фактори, які суттєво впливають на результат операції. З іншого боку, модель повинна бути досить простою, щоб можна було встановити явні (бажано аналітичні) залежності між її параметрами.

Розрізняють аналітичні та статистичні моделі.

Для **аналітичних моделей** характерно встановлення формульних (аналітичних) залежностей між параметрами задач, що записані у вигляді алгебраїчних рівнянь чи нерівностей. Недоліком таких моделей є певні спрощення та припущення при їх побудові, а також їх досить значна складність при необхідності врахування великої кількості факторів.

Статистичні моделі, що базуються на методах математичної статистики, теорії ймовірностей та імітаційного моделювання, дозволяють врахувати значну кількість факторів, а також випадковість процесу, що досліджується. Такі моделі якби відтворюють реальний процес функціонування операції. В більшості випадків статистичне моделювання виконується на ЕОМ, так як при цьому необхідне багаторазове повторення процедури моделювання при різних значеннях

випадкових факторів. Недоліком статистичних моделей є те, що отримані результати значно гірше піддаються аналізу та осмисленню, що ускладнює встановлення взаємозв'язків між факторами операції.

1.4 Загальна постановка задачі

Нехай є певна операція *O*, тобто керований процес, на результат якого можна впливати, вибираючи тим чи іншим способом керовані параметри. Ефективність операції визначається певним чисельним критерієм *W*, який необхідно максимізувати (мінімізувати). Розглянемо найбільш простий випадок, при якому всі фактори, від яких залежить результат операції, діляться на дві групи:

1) задані, тобто заздалегідь відомі фактори (умови виконання операції), на які впливати неможливо: $\alpha_1, \alpha_2, \ldots \alpha_n$;

2) залежні, тобто керовані, фактори (елементи рішення), значення яких можна вибирати в певних межах: $x_1, x_2, \ldots x_m$.

Цей випадок, в якому фактори, що впливають на хід операції, або заздалегідь відомі, або залежать від управління, називається <u>детермінованим</u>.

Слід зазначити, що під заданим умовами $\alpha_1, \alpha_2, \ldots \alpha_n$ та елементами рішення $x_1, x_2, \ldots x_m$ можна розуміти не тільки звичайні числа, але й функції, наприклад, – обмеження, що накладені на елементи рішення (кількість наявного рухомого складу, вантажних фронтів, і т. і.).

Задачу дослідження можна записати таким чином:

$$W=F(\alpha_1, \alpha_2, \ldots \alpha_n, x_1, x_2, \ldots x_m) \qquad (1.1)$$

Таким чином, задача математично формулюється так:

При заданих умовах $\alpha_1, \alpha_2, \ldots \alpha_n$ знайти такі елементи рішення $x_1, x_2, \ldots x_m$, при яких показник ефективності W досягає екстремуму (максимуму або мінімуму).

В більшості випадків дану задачу можна вирішити.

На жаль, детермінований випадок не так часто зустрічається в реальному житті. Більш типовий випадок, при якому не всі умови виконання операції заздалегідь відомі (<u>недетермінований випадок</u>). Таким чином, ефективність операції залежить вже від трьох категорій факторів:

– задані, тобто заздалегідь відомі фактори (умови виконання операції), на які впливати неможливо: $\alpha_1, \alpha_2, \ldots \alpha_n$;

– залежні, тобто керовані, фактори (елементи рішення), значення яких можна вибирати в певних межах: x_1, x_2, … x_m.

– невідомі умови або фактори Y_1, Y_2,… Y_k;

Тоді критерій ефективності:

$$W=F(\alpha_1, \alpha_2, ... \alpha_n, Y_1, Y_2,\dots Y_k, x_1, x_2, \dots x_m) \quad (1.2)$$

Задача в цьому випадку формулюється так:

При заданих умовах α_1, α_2, ... α_n, з врахуванням невідомих факторів Y_1, Y_2,… Y_k найти такі елементи рішення x_1, x_2, … x_m, при яких показник ефективності W досягає екстремуму (максимуму або мінімуму).

Дана задача може вирішуватись:

– з врахуванням законів розподілу випадкових величин Y_1, Y_2,… Y_k, коли випадкові значення замінюються їх математичними очікуваннями і задача зводиться до детермінованого випадку.

– за допомогою „оптимізації у середньому”, коли операція виконується декілька раз при різних значеннях факторів Y_1, Y_2,… Y_k. При цьому вибирається таке рішення (x_1, x_2, … x_m), при якому операція у середньому буде найбільш ефективною.

Контрольні запитання

1. Без чого неможливі розробка технологічних процесів роботи об’єктів транспортної інфраструктури, графіка руху та плану формування, ефективна організація потоків, планування роботи підприємств транспорту?
2. Що дозволяють врахувати математичні методи?
3. Що таке критерій ефективності та для чого він потрібний?
4. Які основні етапи операційного методу?
5. Яка відмінність між аналітичними та статистичними моделями?
6. Описати особливості детермінованованого випадку визначення критерію ефективності.
7. Описати особливості недетермінованованого випадку визначення критерію ефективності.
8. Який із випадків частіше зустрічається в реальному житті: детермінований чи недетермінований?
9. Навести математичний вираз для пошуку критерію ефективності для детермінованованого випадку.
10. Навести математичний вираз для пошуку критерію ефективності для недетермінованованого випадку.

РОЗДІЛ 2

Визначення найкоротшого маршруту на транспортній мережі

2.1 Теоретичні відомості

2.1.1 Характеристика проблеми визначення найкоротшого маршруту в транспортній мережі

Визначення найкоротшого маршруту в транспортній мережі є однією з ключових задач оптимізації транспортних систем, в результаті вирішення якої можна знайти оптимальний шлях між двома точками в мережі, мінімізуючи витрати часу, відстані або інших ресурсів [2, 3, 7 – 10].

Існує велика кількість методів визначення найкоротшого маршруту, які активно використовуються з метою оптимізації логістики, планування маршрутів громадського транспорту, управління дорожнім рухом та інших транспортних задач.

Вони дозволяють:

- знижувати витрати на перевезення;
- підвищувати ефективність доставки товарів;
- зменшувати час в дорозі;
- знижувати навантаження на транспортну інфраструктуру.

2.1.2 Постановка задачі

Транспортну мережу можна уявити як граф (див. рисунок 2.1)

Граф у контексті транспортної мережі є математичною структурою, яка використовується для моделювання та аналізу транспортних систем, яка складається з вузлів та ребер.

При цьому, **вузли** (вершини) графа представляють собою місця або транспортні зупинки (міста, станції, порти, магазини, склади), а **ребра** (дуги) – це шляхи, які з'єднують ці вузли (автодороги, колії тощо).

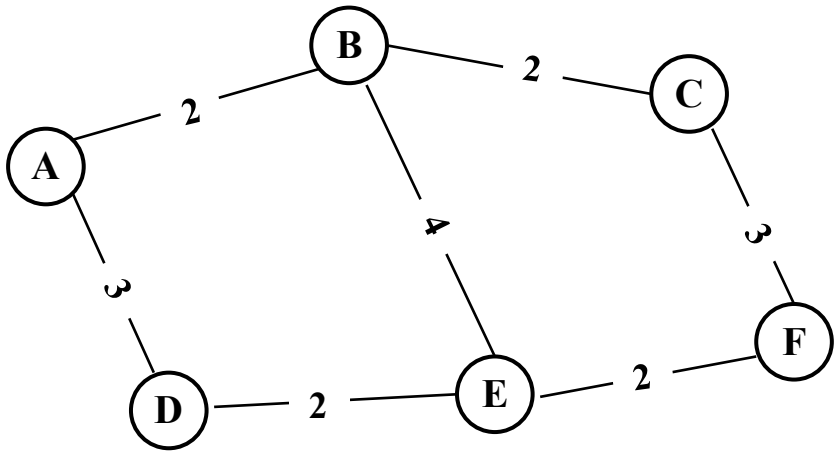


Рисунок 2.1 – Приклад транспортної мережі у вигляді графа

Кожне ребро має певну вагу, яка відображає важливу характеристику, наприклад, відстань, час у дорозі або вартість перевезення.

Задача визначення найкоротшого маршруту полягає в пошуку шляху між двома заданими вузлами з мінімальною сумою довжин ребер.

2.1.3 Доцільність вирішення даної проблеми

Визначення найкоротшого маршруту є фундаментальною задачею в управлінні транспортними системами. Використання різних алгоритмів і методів дозволяє знаходити оптимальні рішення для широкого спектру задач, від планування міських маршрутів до міжнародних транспортних мереж.

Завдяки постійному розвитку комп'ютерних технологій і нових методів обробки даних, ці процеси стають все більш ефективними, забезпечуючи покращення роботи транспортних систем у всьому світі.

2.2 Методи пошуку найкоротшого маршруту

2.2.1 Використання алгоритму Дейкстри

Алгоритм Дейкстри [7, 8] є одним із найпоширеніших методів для знаходження найкоротшого шляху в графах з невід'ємними вагами ребер. Він працює за принципом поступового розширення області досяжності, починаючи з початкової вершини і розширюючи її на сусідні вершини з найменшою сумарною вагою.

Основні кроки даного алгоритму:

Крок 1. Задається початкова вершина, для якої значення найкоротшого шляху встановлюється як 0, а для інших вершин значення встановлюється як нескінченність (ініціалізація).

Крок 2. На кожному кроці вибирається вершина з найменшим значенням ваги (найближча невідвідана вершина).

Крок 3. Оновлюються значення найкоротшого шляху для сусідніх вершин, якщо новий шлях коротший за відомий раніше.

Крок 4. Вказані кроки повторюються до тих пір, поки всі вершини не будуть відвідані.

2.2.2 Використання алгоритму Флойда-Воршелла

Даний алгоритм [7, 8] дозволяє знайти найкоротші шляхи між всіма парами вершин у графі і підходить для випадків, коли потрібно провести багато запитів щодо найкоротших шляхів.

Основні кроки алгоритму:

Крок 1. Ініціалізується матриця відстаней, де початкові значення задаються як ваги ребер (якщо немає прямого ребра, то вказуємо «∞»).

Крок 2. Послідовно для кожної пари вершин перевіряється, чи не покращиться найкоротший шлях через проміжну вершину.

Крок 3. Даний процес повторюється до тих пір, поки не будуть опрацьовані всі проміжні вершини.

Крок 4. Читання результату оптимізації.

2.2.3 Використання алгоритму A*:

Алгоритм A* (A-star) [9] є евристичним методом і використовується в умовах, коли необхідно знайти найкоротший шлях швидко, навіть у дуже великих графах. Він поєднує в собі переваги алгоритмів Дейкстри та пошуку в ширину, додаючи евристику, яка оцінює "вартість" проходження до кінцевої точки.

Основні кроки алгоритму:

Крок 1. Ініціалізується початкова вершина, для якої обчислюється загальна оцінка шляху, що складається з поточного шляху до вершини і оцінки вартості до кінцевої точки.

Крок 2. Вибирається вершина з найменшою оцінкою.

Крок 3. Оновлюються оцінки для сусідніх вершин.

Крок 4. Процес повторюється доти, поки не буде знайдений шлях до кінцевої вершини.

2.2.4 Використання алгоритму Беллмана-Форда

Цей алгоритм [7, 8] підходить для пошуку найкоротшого шляху для графів з від'ємними вагами ребер, що здатний обробляти графи з негативними циклами.

Основні кроки алгоритму:

Крок 1. Ініціалізація аналогічна алгоритму Дейкстри, але з обробкою можливості наявності від'ємних ваг.

Крок 2. Повторюється процес релаксації ребер для всіх вершин на кожному кроці.

Крок 3. Процедура триває до тих пір, поки всі вершини не будуть оброблені, зазвичай в кілька ітерацій.

2.2.5 Використання евристичних методів

Для великих транспортних мереж (тобто мереж зі значною кількість вершин), де класичні методи можуть бути надто ресурсомісткими, більш доцільним може бути використання евристичних методів [10], такі як генетичні алгоритми або метод імітації відпалу.

Вказані методи не гарантують знаходження абсолютно оптимального рішення, але дозволяють знайти достатньо хороші рішення за менший час у порівнянні з іншими методами.

2.3. Приклад розв’язання задач. Задача 1

Умова задачі

Є схема умовної транспортної мережі, на якій вузли графа представляють собою пункти мережі, а дуги – прямі шляхи між пунктами. На дугах даної транспортної мережі вказані їх ваги (в даному випадку – це відстані між пунктами, однак вагами дуг можуть бути й інші критерії, наприклад, тривалість чи вартість руху по конкретній дузі).

Необхідно знайти найкоротший маршрут між початковим пунктом n та кінцевим пунктом k.

Вихідні дані

Схема умовної транспортної мережі наведена на рисунку 2.2.

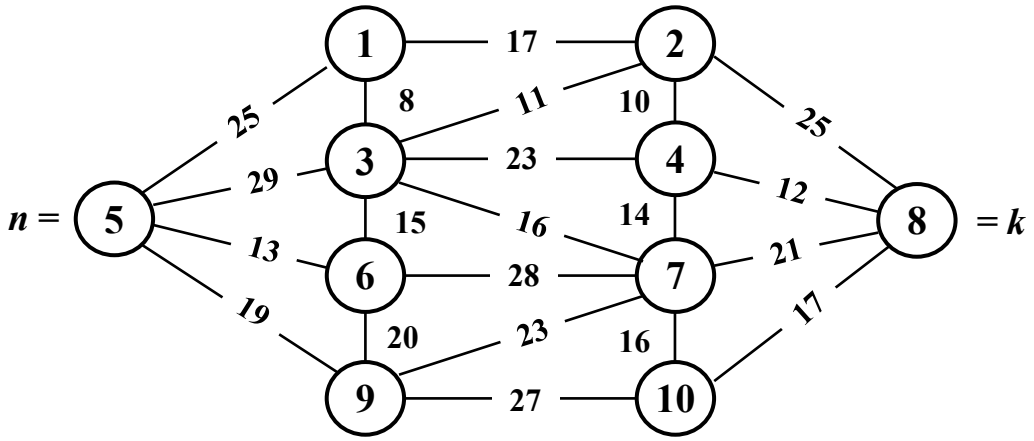


Рисунок 2.2 – Схема умовної транспортної мережі

Розв’язання задачі

Вирішимо дану задачу за допомогою алгоритму Флойда-Воршелла, який детально описаний в [7]. Складемо матрицю відстаней між пунктами (пишемо число, якщо прямий зв’язок є, та « ∞» в іншому разі.

Матриця відстаней наведена у таблиці 2.1.

Таблиця 2.1

Заповнення матриці відстаней

λ ⇒ ⇓											
	i \ j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	1	**x**	17	8	∞	25	∞	∞	∞	∞	∞
	2	17	**x**	11	10	∞	∞	∞	25	∞	∞
	3	8	11	**x**	23	29	15	16	∞	∞	∞
	4	∞	10	23	**x**	∞	∞	14	12	∞	∞
	5	25	∞	29	∞	**x**	13	∞	∞	19	∞
	6	∞	∞	15	∞	13	**x**	28	∞	20	∞
	7	∞	∞	16	14	∞	28	**x**	21	23	16
	8	∞	25	∞	12	∞	∞	21	**x**	∞	17
	9	∞	∞	∞	∞	19	20	23	∞	**x**	27
	10	∞	∞	∞	∞	∞	∞	16	17	27	**x**

Розв’язання задачі можна розглядати у вигляді алгоритму Флойда-Воршелла, який складається з декількох кроків.

Крок 1. Присвоїмо наступні потенціали пунктам:

– для початкового пункту λ_n=0;

– для всіх інших пунктів λ_i=∞.

Проставляємо потенціали біля всіх пунктів по горизонталі і вертикалі.

Згідно умови задачі, початковий пункт n = 5, а кінцевий k = 8.

Тоді потенціал рядка 5 і стовпця 5 дорівнює 0, а всіх інших ∞.

Крок 2. Кожну клітинку матриці перевіряємо за умовою

$$\lambda_j - \lambda_i \le l_{ij}, \tag{2.1}$$

де λ_i, λ_j – відповідно потенціал i-го рядка та j-го стовпця;

l_{ij} – відстань між i-м і j-м пунктами.

Перевірку умови (2.1) починають з рядка матриці n, який відповідає початковому пункту.

У випадку, якщо умова (2.1) виконується, то потенціал залишаємо без зміни, в іншому випадку – розраховується нове значення потенціалу за допомогою виразу

$$\lambda_j = \lambda_i + l_{ij}. \tag{2.2}$$

Нове значення λ_j записується в стовпчик j та рядок $i = j$.

Результат розв’язання задачі наведено в таблиці 2.2.

Таблиця 2.2

λ ⇒ ⇓										25 ∞	39 40 42 ∞	28 29 ∞	49 50 52 ∞	0	13 ∞	41 45 ∞	61 62 64 67 ∞	19 ∞	46 57 ∞
										1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
			25	∞						**1**	**x**	17	8	∞	25	∞	∞	∞	∞	∞
	39	40	42	∞						**2**	17	**x**	11	10	∞	∞	∞	25	∞	∞
		28	29	∞						**3**	8	11	**x**	23	29	15	16	∞	∞	∞
	49	50	52	∞						**4**	∞	10	23	**x**	∞	∞	14	12	∞	∞
				0						**5**	25	∞	29	∞	**x**	13	∞	∞	19	∞
			13	∞						**6**	∞	∞	15	∞	13	**x**	28	∞	20	∞
		41	45	∞						**7**	∞	∞	16	14	∞	28	**x**	21	23	16
61	62	64	67	∞						**8**	∞	25	∞	12	∞	∞	21	**x**	∞	17
			19	∞						**9**	∞	∞	∞	∞	19	20	23	∞	**x**	27
		46	57	∞						**10**	∞	∞	∞	∞	∞	∞	16	17	27	**x**

Після перевірки рядка 5 починають перевірку з максимально високо розташованого рядка, потенціал якого відмінний від ∞ (в даному випадку наступним буде рядок 1, потім рядок 2 і т. д.).

Крок 2 повторюють до тих пір, поки умова (2.1) не буде виконуватися для усіх клітин.

Крок 3. Якщо умова (2.1) виконується для усіх клітин, то треба прочитати результат розв'язання задачі.

Загальні принципи

а) Починаємо читати результат з кінцевого пункту *k*.

По стовпцю *k* повинна бути хоча б одна клітинка, для якої

$$\lambda_j - \lambda_i = l_{ij}. \qquad (2.3)$$

Пункт у рядку *i* є проміжним пунктом P_1. Матимемо $P_1 \to k$.

Тобто маршрут формується у зворотному порядку.

б) Продовжимо читати результат для проміжного пункту P_1.

По стовпчику $j = P_1$ знаходимо клітинку, для якої виконується умова (2.3). Пункт у рядку *i* є проміжним пунктом P_2. Матимемо $P_2 \to P_1 \to k$.

в) Продовжимо читати результат для проміжного пункту P_2 (і далі).

Якщо черговий $P_m = n$, то маршрут буде: $n \to P_{m-1} \to \ldots \to P_2 \to P_1 \to k$.

Якщо в стовпці дві і більше клітинок, для яких виконується умова (2.3), то це альтернативні маршрути – можна обрати будь-який з них.

Застосування загальних принципів для даної задачі

У даному прикладі кінцевим є пункт 8.

Для клітинки, яка знаходиться на перетинанні стовпця 8 зі строчкою 4 виконується умова $\lambda_8 - \lambda_4 = 61 - 49 = 12$.

Тому пункт у рядку 4 є першим проміжним пунктом.

Отже, поточне рішення буде 4→8.

Для клітинки, яка знаходиться на перетинанні стовпця 4 з рядком 2 виконується умова $\lambda_4 - \lambda_2 = 49 - 39 = 10$.

Тому пункт у рядку 2 є другим проміжним пунктом.

Тепер актуальне рішення буде 2→4→8.

Далі діємо аналогічно і отримуємо результат: 5→6→3→2→4→8.

Крок 4. Для знайденого найкоротшого маршруту визначається його довжина. У даному прикладі $L = 13 + 15 + 11 + 10 + 12 = 61$ *км*.

Крок 5. Перевірити по графу транспортної мережі вірність розв'язання задачі, зіставивши знайдений маршрут з деякими можливими маршрутами.

Контрольні запитання

1. Що таке вузол на графі транспортної мережі?
2. Що таке ребро на графі транспортної мережі?
3. Які характеристики можуть позначати вагу ребра у графі?
4. Для чого використовують алгоритм Дейкстри?
5. Що обчислює алгоритм Флойда-Воршелла?
6. Як алгоритм A* допомагає знайти шлях?
7. Яка основна особливість алгоритму Беллмана-Форда?
8. Що таке граф у контексті транспортної мережі?
9. Для чого потрібні методи знаходження найкоротших маршрутів?
10. В яких сферах може бути корисним пошук найкоротшого маршруту?

РОЗДІЛ 3

Оптимізація розподілу потоків на транспортній мережі

3.1 Теоретичні відомості

Оптимізація розподілу потоків на мережі – це комплексний процес визначення найбільш ефективних маршрутів для руху транспортних засобів, що спрямований на мінімізацію заторів, скорочення часу перебування в дорозі, зниження експлуатаційних витрат та покращення екологічної ситуації. Він включає аналіз транспортної інфраструктури, прогнозування руху та впровадження сучасних технологій управління дорожнім трафіком [3].

Основні завдання оптимізації:

1. Розподіл транспортних потоків – визначення оптимальних маршрутів для різних типів транспорту (легкового, вантажного, громадського) з урахуванням рівня завантаженості автодоріг, щоб зменшити затори та підвищити пропускну здатність автодорожньої мережі. Використання інтелектуальних транспортних систем (ITS) дозволяє автоматично регулювати потоки за допомогою датчиків руху та адаптивних світлофорів.

2. Управління попитом – застосування заходів, спрямованих на рівномірний розподіл навантаження на транспортну систему протягом доби. Це може включати впровадження гнучкого графіка роботи, заохочення використання громадського транспорту, розвиток інфраструктури для велосипедистів і пішоходів, а також використання електронних квитків і систем тарифного регулювання для розвантаження пікових годин.

3. Інтеграція різних видів транспорту – створення єдиної транспортної системи, що забезпечує ефективну взаємодію між автомобільним, залізничним, водним і громадським транспортом. Це сприяє підвищенню доступності транспортних послуг, зниженню заторів та

покращенню мобільності населення. Важливими заходами в цьому напрямку є будівництво транспортно-пересадкових вузлів, розвиток мультимодальних перевезень та запровадження єдиних електронних квитків для різних видів транспорту.

Сучасні методи оптимізації включають використання штучного інтелекту та аналізу великих даних для прогнозування завантаженості доріг, що дозволяє завчасно реагувати на зміни у транспортних потоках. Впровадження автоматизованих систем управління дозволяє не лише підвищити ефективність транспортної мережі, а й сприяти зниженню рівня шкідливих викидів, що повинно позитивно впливати на стан навколишнього середовища.

3.2 Приклад розв'язання задач. Задача 2

Умова задачі

Є деяка умовна транспортна мережа, схема якої наведена на рисунку 3.1.

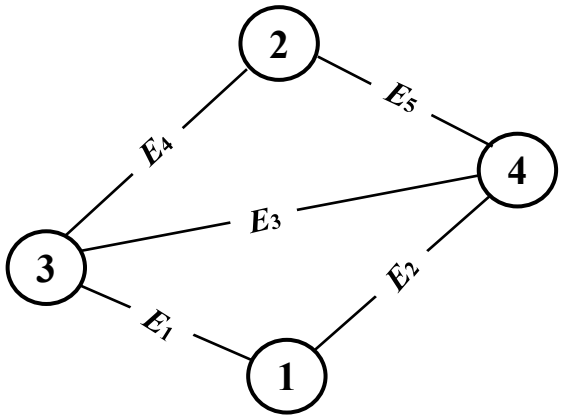


Рисунок 3.1 – Схема транспортної мережі

На даному рисунку вузли графа представляють собою пункти мережі, а дуги – це прямі шляхи між пунктами.

На дугах даної транспортної мережі вказані їх ваги (в даному випадку – це функції витрат на переміщення транспортної маси).

Необхідно оптимально розподілити потік M одиниць транспортних мас (ТМ) з пункту n у пункт k.

Вихідні дані

Функції витрат на переміщення транспортної маси наведені в таблиці 3.1.

Таблиця 3.1

Функції витрат на переміщення транспортної маси

Позначення	E_1	E_2	E_3	E_4	E_5
Значення	$6M^2$	$35M + M^2$	$10M + M^2$	$14M^2$	$15M + 2M^2$

Початковий пункт $n = 1$, кінцевий пункт $k = 2$.

Розв'язання задачі

Для вирішення даного завдання використовуємо метод покрокового розподілу за мінімумом сумарних диференційних витрат.

Алгоритм розв'язання задачі

Крок 0. Спочатку для кожної дуги транспортної мережі визначимо відповідні диференційні функції витрат $E' = f(M_{ij})$.

Крок 1. Початкове завантаження усіх ділянок приймається рівним $M_{ij} = 0$ одиниць транспортної маси.

На кожному етапі вирішення задачі буде розподілятись лише 1 одиниця транспортної маси ($\Delta M = 1$).

Матрицю заповнюємо симетрично відносно головної діагоналі.

У разі, якщо прямого зв'язку між пунктами немає (а також по головній діагоналі матриці), слід ставити знак «∞».

Якщо прямий зв'язок між пунктами є, то розраховуємо диференціальні витрати $E' = f(M_{ij})$ між цими пунктами для поточного значення M_{ij} одиниць транспортної маси.

Результат записуємо у клітинку на перетині рядка i та стовпця j, а також у клітинку на перетині рядка j та стовпця i поточної матриці.

Крок 2. Знаходимо найкоротший маршрут (за мінімумом $\Sigma E'$) з початкового пункту n в кінцевий пункт k методом потенціалів.

а) Приймемо потенціал для початкового пункту 1 рівним 0, а для інших пунктів – рівним «∞».

б) Визначимо нові потенціали для всіх клітинок, які містять число, за допомогою виразу

$$\lambda_j - \lambda_i \le l_{ij} \qquad (3.1)$$

де λ_i, λ_j – відповідно потенціал i-го рядка та j-го стовпця;

l_{ij} – відстань між i-м і j-м пунктами.

Якщо умова (3.1) не виконується то визначаємо нове значення потенціалу за формулою

$$\boldsymbol{\lambda_j = \lambda_i + l_{ij}} \qquad (3.2)$$

Нове значення λ_j записується в стовпчик j та рядок $i = j$.

Якщо умова (3.1) виконується, то потенціал залишається без змін.

Умова (3.1) повинна виконуватись для усіх клітинок таблиці.

в) Прочитати результат зі стовпчика з номером пункту прибуття, знайшовши клітинку, для якої справедлива рівність (поки не отримаємо $i = n$):

$$\boldsymbol{\lambda_j - \lambda_i = l_{ij}} \qquad (3.3)$$

Крок 3. Складемо таблицю розподілу потоків, додаючи $\Delta M = 1$ до ситуації перед цим кроком у строгій відповідності до знайденого маршруту. Дана таблиця заповнюється без симетрії відносно діагоналі.

Крок 4. Якщо перевезені всі одиниці транспортної маси, то розв'язання задачі закінчується і слід прочитати кінцевий результат.

Якщо перевезені не всі одиниці транспортної маси, то складемо нову матрицю диференціальних витрат $E'=f(M_{ij})$, значення якої розраховуються згідно зі значеннями M_{ij}, наведеними в попередній таблиці розподілу потоків (нова матриця заповнюється симетрично відносно діагоналі).

Далі переходимо до кроку 2.

Застосування алгоритму для розв'язання даної задачі

Перед початком власне вирішення задачі (крок 0 алгоритму) визначимо для кожної дуги диференціальну функцію витрат $E'=f(M_{ij})$ в таблиці 3.2 (вказані функції підходять для руху в обидві сторони).

Таблиця 3.2

Диференціальна функція витрат $E'=f(M_{ij})$

Дуга	E	E'
1 – 3	$6M^2$	$12M$
1 – 4	$35M + M^2$	$35 + 2M$
2 – 3	$14M^2$	$28M$
2 – 4	$15M + 2M^2$	$15 + 4M$
3 – 4	$10M + M^2$	$10 + 2M$

Початкове завантаження усіх ділянок транспортної мережі приймаємо рівним $M_{ij} = 0$ одиниць транспортної маси (див. таблицю 3.3).

Таблиця 3.3

Початкове завантаження усіх ділянок (*M*=0)

$i \backslash j$	1	2	3	4
1		0	0	0
2	0		0	0
3	0	0		0
4	0	0	0	

Розподілимо перевезення 1-ї одиниці транспортної маси.

Використовуючи схему транспортної мережі (див. рисунок 3.1) та значення диференційної функції витрат (див. таблицю. 3.2) заповнюємо матрицю витрат у таблиці 3.4 (симетрично відносно діагоналі).

Таким чином, в результаті виконання кроку 1 та пунктів *а*) та *б*) кроку 2 алгоритму розв'язання задачі отримаємо заповнену таблицю 3.4 з усіма визначеними потенціалами.

Таблиця 3.4

$\lambda_i \Downarrow$			$\lambda_j \Rightarrow$	0	0	0	10
				∞	∞	35	∞
			$i \backslash j$	1	2	3	4
		0	1	∞	∞	0	35
	0	∞	2	∞	∞	0	15
	0	∞	3	0	0	∞	10
10	35	∞	4	35	15	10	∞

M=1 *Таблиця 3.5*

$i \backslash j$	1	2	3	4
1		0	1	0
2	0		0	0
3	0	1		0
4	0	0	0	

Далі переходимо до пункту *в*) кроку 2 алгоритму розв'язання задачі, а саме до читання результату.

Згідно умов задачі кінцевим пунктом є пункт $k = 2$, тому читання результату починаємо зі стовпчика 2, в якому зверху вниз кожну клітинку з числом перевіряємо за умовою (3.3).

Так, для клітинки у рядку 3 перевірка за умовою (3.3) дає 0 – 0 = 0, а отже умова виконується, тому дана клітинка певним чином виділяється.

Далі визначимо пункт, з якого слід їхати на пункт 3.

У стовпчику 3 знаходимо першу ж клітинку, для якої умова (3.3) виконується, оскільки 0 – 0 = 0.

В результаті, найкоротший маршрут 1-ї одиниці ТМ такий $1 \rightarrow 3 \rightarrow 2$.

Згідно кроку 3 алгоритму розв'язання задачі складемо таблицю розподілу потоків додавши ΔM =1 до ситуації перед цим кроком (див. таблицю 3.3) у строгій відповідності до знайденого маршруту та без симетрії відносно діагоналі.

Так, для рядка 1 та стовпчика 3 матимемо 0 + 1 = 1. Аналогічно для рядка 3 та стовпчика 2 матимемо 0 + 1 = 1. В результаті отримали нову таблицю розподілу потоків (див. таблицю 3.5).

Розподілимо перевезення 2-ї одиниці транспортної маси

Таблиця 3.6

λ_i ⇓			λ_j ⇒	0	37 40 ∞	12 ∞	22 35 ∞
			i \ j	1	2	3	4
		0	1	∞	∞	12	35
37	40	∞	2	∞	∞	28	15
	12	∞	3	12	28	∞	10
22	35	∞	4	35	15	10	∞

M=2 *Таблиця 3.7*

i \ j	1	2	3	4
1		0	2	0
2	0		0	0
3	0	1		1
4	0	1	0	

Аналогічно знаходимо найкоротший маршрут перевезення 2-ї одиниці ТМ, який згідно таблиці 3.6 буде таким: $1 \to 3 \to 4 \to 2$.

Крок 4 повторюється до тих пір, поки не буде розподілена вся транспортна маса в обсязі 6 одиниць (див. таблиці 3.7 – 3.15).

Розподілимо перевезення 3-ї одиниці транспортної маси

Таблиця 3.8

λ_i ⇓			λ_j ⇒	0	52 ∞	24 ∞	35 ∞
			i \ j	1	2	3	4
		0	1	∞	∞	24	35
	52	∞	2	∞	∞	28	19
	24	∞	3	24	28	∞	12
	35	∞	4	35	19	12	∞

M=3 *Таблиця 3.9*

i \ j	1	2	3	4
1		0	3	0
2	0		0	0
3	0	2		1
4	0	1	0	

За таблицею 3.8 найкоротший маршрут 3-ї одиниці ТМ: $1 \to 3 \to 2$

Розподілимо перевезення 4-ї одиниці транспортної маси

Таблиця 3.10

λ_i ⇓			λ_j ⇒	0	54 92 ∞	36 ∞	35 ∞
			i \ j	1	2	3	4
		0	1	∞	∞	36	35
54	92	∞	2	∞	∞	56	19
	36	∞	3	36	56	∞	12
	35	∞	4	35	19	12	∞

M=4 *Таблиця 3.11*

i \ j	1	2	3	4
1		0	3	1
2	0		0	0
3	0	2		1
4	0	2	0	

З таблиці 3.10 найкоротший маршрут 4-ї одиниці ТМ: $1 \to 4 \to 2$

Розподілимо перевезення 5-ї одиниці транспортної маси

Таблиця 3.12

λ_i ⇓			λ_j ⇒	0	60 92 ∞	36 ∞	37 ∞
			i \ j	1	2	3	4
		0	1	∞	∞	36	37
60	92	∞	2	∞	∞	56	23
	36	∞	3	36	56	∞	12
	37	∞	4	37	23	12	∞

M=5 *Таблиця 3.13*

i \ j	1	2	3	4
1		0	3	2
2	0		0	0
3	0	2		1
4	0	3	0	

З таблиці 3.12 найкоротший маршрут 5-ї одиниці ТМ: 1 → 4 → 2

Розподілимо перевезення 6-ї одиниці транспортної маси

Таблиця 3.14

λ_i ⇓			λ_j ⇒	0	66 92 ∞	36 ∞	39 ∞
			i \ j	1	2	3	4
		0	1	∞	∞	36	39
66	92	∞	2	∞	∞	56	27
	36	∞	3	36	56	∞	12
	39	∞	4	39	27	12	∞

M=6 *Таблиця 3.15*

i \ j	1	2	3	4
1		0	3	3
2	0		0	0
3	0	2		1
4	0	4	0	

З таблиці 3.14 найкоротший маршрут 6-ї одиниці ТМ: 1 → 4 → 2 .

Для наочності оптимальний розподіл потоків 6-ти одиниць транспортної маси, який отримано в таблиці 3.15, наведемо на рисунку 3.2:

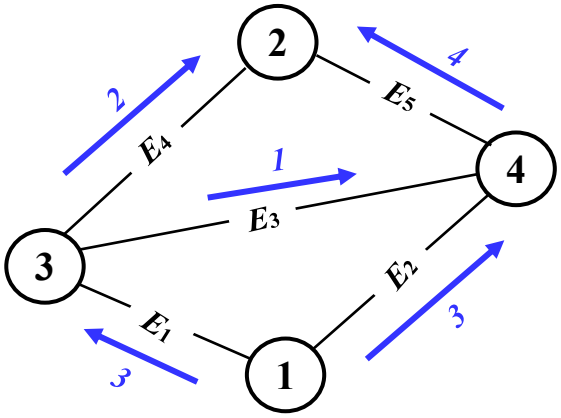


Рисунок 3.2 – Схема оптимального розподілу потоків на мережі

Отже, вся задана кількість транспортної маси була розподілена вказаним на рисунку 3.2 чином. Так, сума потоків, що виходять з пункту 1 та потоків, що входять в пункт 2 однакова та становить 6 одиниць.

Контрольні запитання

1. Що таке оптимізація розподілу потоків на мережі?
2. Які є основні завдання оптимізації розподілу потоків?
3. Що включають сучасні методи оптимізації розподілу потоків?
4. Яким методом вирішується розподіл потоків в Задачі 2?
5. Що саме визначається для кожної дуги транспортної мережі в Задачі 2?
6. Скільки одиниць транспортної маси ΔM розподіляється на кожному етапі вирішення Задачі 2?
7. Як саме відносно головної діагоналі заповнюється матриця витрат при вирішенні Задачі 2?
8. Як саме відносно головної діагоналі заповнюється таблиця розподілу потоків при вирішенні Задачі 2?
9. Яким приймається початкове завантаження транспортної мережі при вирішенні Задачі 2?
10. Як саме виконується базова перевірка розподілу транспортної маси біля вихідної та вхідної вершин при вирішенні Задачі 2?

РОЗДІЛ 4

Розрахунок раціональної ширини пасажирських платформ

4.1 Теоретичні відомості

Пасажирські платформи повинні забезпечувати зручну, швидку та безпечну висадку пасажирів, їх безперешкодний вихід у місто без різкого зниження швидкості пересування.

В основному пасажирські платформи поділяються на:

Бічна платформа – пасажирська платформа, яка розташована з одного боку від колії. Доступ до такої платформи може бути як в одному рівні (в тому числі і без перетину колій), так і в різних рівнях.

Острівна платформа – платформа, що розташована між двома коліями та обслуговує поїзди з обох боків. Доступ до такої платформи може бути в одному рівні (з перетином колій), так і в різних рівнях.

На середніх, великих і особливо великих вокзалах ширина платформи встановлюється за розрахунками залежно від інтенсивності та характеру пасажиропотоків (прямі, місцеві, приміські), швидкостей руху пасажирських поїздів, кількості виходів із платформи та місця їх розташування, а також розмірів пристроїв, що повинні розміщуватися на них (сходи, павільйони та ін.).

Ширина бічної пасажирської платформи, згідно з Державними будівельними нормами [6], має бути:

– не менше 6 *м* – у межах будівлі вокзалу в нормальних умовах;

– не менше 5 *м* – у межах будівлі вокзалу у важких умовах;

– не менше 4 *м* – поза межами будівлі вокзалу місткістю не менше 200 *чол*;

– не менше 3 *м* – поза межами будівлі вокзалу місткістю менше 200 *чол*.

Ширина острівної платформи повинна бути:

– не менше 8 *м* – в нормальних умовах проектування в разі розташування острівної платформи між головними коліями;

– не менше 6 *м* – в особливо важких умовах проектування; для забезпечення безпеки пасажирів слід передбачати відповідні заходи (улаштування поручнів уздовж осі платформ із розривом для проходу, оповіщувальної сигналізації про підхід швидкісних поїздів та ін.).

– не менше 4 *м* – на лініях III і IV категорій.

За наявності споруд (павільйонів, сходів з тунелів та пішохідних мостів і т. д.):

– не менше 2 *м* від краю платформи до краю споруди – на лініях, де передбачається рух пасажирських поїздів зі швидкістю не більше 140 км/год;

– не менше 3 *м* від краю платформи до краю споруди – на лініях, де передбачається рух пасажирських поїздів зі швидкістю більше 140 км/год;

– не менше 3,10 *м* від осі ближньої колії до краю споруди – в окремих випадках з відповідним обґрунтуванням при спорудженні споруд на платформі, для встановлення стовпів освітлення і т. ін..

Ширина платформи $b_{\text{о}}$ залежить від населеності M поїздів, що прибувають, довжини поїзда $L_{\text{с}}$ і розрахункової площі p, яку займає один пасажир на платформі. На ширину платформи впливає також співвідношення часу проходу пасажира від останнього вагона до виходу з платформи та часу висадки пасажирів у зв'язку з різними місцями їх скупчення [6, 11].

Час висадки пасажирів з поїзда розраховується за формулою

$$T_{\text{в}} = \frac{M}{mcn_{\text{в}}}, \tag{4.1}$$

де m – кількість вагонів у пасажирському поїзді;

c – кількість виходів, що одночасно відкриваються в одному пасажирському вагоні;

$n_{\text{в}}$ – кількість пасажирів, що висаджуються через один вихід за одиницю часу.

Час проходу поїзда пасажиром з останнього вагона

$$T_{\text{пр}} = \frac{L_{\text{с}}}{60v}, \tag{4.2}$$

де v – швидкість руху пасажирів, *м/хв*.

При розрахунку ширини платформи можливі два випадки:

– випадок 1 $T_{\text{в}} < T_{\text{пр}}$;

– випадок 2 $T_{\text{в}} \geq T_{\text{пр}}$.

Випадок 1

Розглянемо розрахункову схему для 1-го випадку, яка наведена на рисунку 4.1.

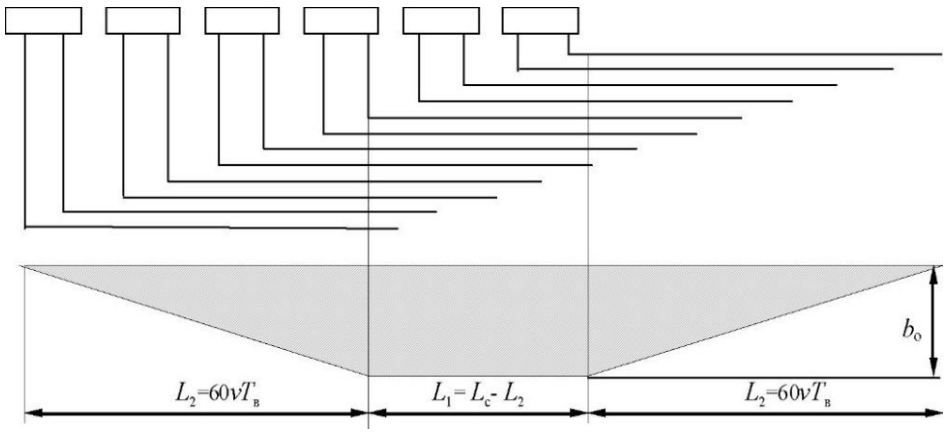


Рисунок 4.1 – Схема для розрахунку ширини пасажирської платформи для випадку $T_в < T_{пр}$

Як видно з даного рисунку, максимальний потік утворюється біля перших вагонів в момент закінчення висадки та зберігається до точки, яка знаходиться на відстані L_2 від кінця поїзда, яка дорівнює $L_2 = 60vT_в$ (довжина проходу пасажира за час $T_в$). З достатньою точністю потік пасажирів може бути представлений у вигляді трапеції.

Площа трапеції, з одного боку, пропорційна населеності поїзда, а з іншого – може бути розрахована геометрично.

Таким чином, отримаємо

$$Mp = \left(L_c - L_2\right)b_o + \frac{L_2 b_o}{2},$$

Звідки отримаємо

$$b_o = \frac{Mp}{L_c}.$$

Випадок 2

Розглянемо розрахункову схему для 2-го випадку на рисунку 4.2.

Як видно з рисунку 4.2, максимальний потік утворюється за першим вагоном в момент підходу до нього пасажирів з останнього і зберігається протягом часу $T_в - T_{пр}$.

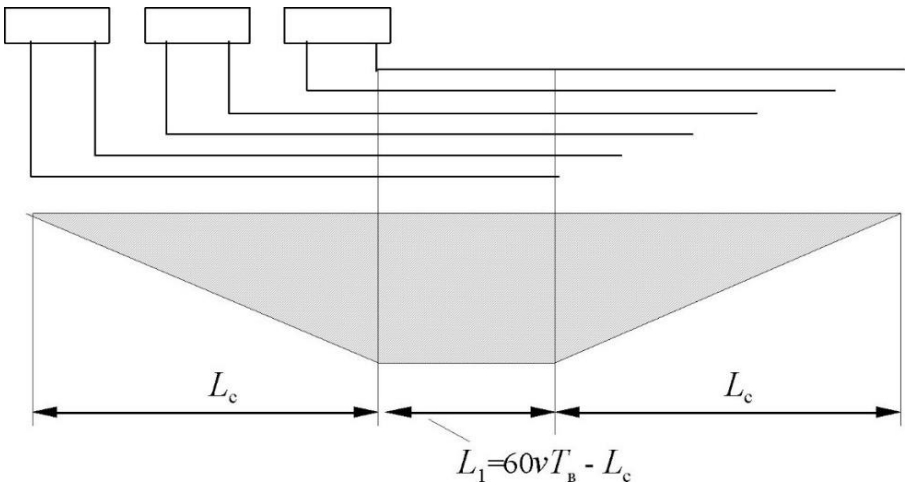


Рисунок 4.2 – Схема для розрахунку ширини пасажирської платформи для випадку $T_в > T_{пр}$

Із міркувань відповідності площ, матимемо

$$Mp = \left(60vT_в - L_с\right)b_о + 2\frac{Lb_о}{2}.$$

Використовуючи вираз (4.2), замінюємо v на $L_с / 60T_{пр}$.

Виконуючи перетворення, отримаємо

$$b_о = \frac{Mp}{L_с}k,$$

де $k = T_{пр} / T_в$.

При розрахунку ширини пасажирської платформи також необхідно врахувати такі умови:

1. Якщо передбачається одночасна посадка та висадка пасажирів, то ширина платформи збільшується на 2,0 .. 2,5 м для приміських і на 3,0 .. 3,5 м для дальніх поїздів.

2. Якщо передбачається одночасна висадка з двох поїздів, то значення M збільшується в два рази, а p зменшується на 25 .. 30 %.

При цьому розрахунок $T_в$ та $T_{пр}$ виконується без змін. Також вважається, що ширина платформи для поїздів, які прибувають буде достатньою і для поїздів, що відправляються.

3. При розрахунку ширини платформи також слід враховувати наявність осіб, які зустрічають пасажирів, носіїв та ширину смуг безпеки. При цьому, враховують лише ту частину поїзда, що тяжіє до даного сходу з платформи.

Таким чином, якщо схід з платформи розташований посередині платформи, та можливий з двох сторін, то необхідно враховувати лише половину состава, а якщо схід з платформи можливий лише з однієї сторони платформи, то необхідно враховувати повну довжину состава.

Урахування осіб, які зустрічають пасажирів, носіїв і ширини смуг безпеки при розрахунку ширини платформи відбувається наступним чином:

– для приміських поїздів

$$b_{\text{пр}} = \frac{M_{\text{пр}} p_{\text{пр}}}{L_{\text{с}}} k + 1 \quad , \tag{4.3}$$

– для дальніх поїздів

$$b_{\text{д}} = 1{,}25 \frac{M_{\text{д}} p_{\text{д}}}{L_{\text{с}}} k + 2 \, , \tag{4.4}$$

де $k = \begin{cases} 1, \text{ якщо } T_{\text{в}} < T_{\text{пр}}, \\ T_{\text{пр}} / T_{\text{в}}, \text{ якщо } T_{\text{пр}} < T_{\text{в}} \end{cases}$.

4. Якщо на платформі розміщені будь-які пристрої або споруди, то ширину платформ розраховують з урахуванням габариту наближення споруд (наприклад, ширини сходів у тунель або на пішохідний міст). При цьому на ширину платформ накладаються такі обмеження:

– якщо схід розташований посередині пасажирської платформи (див. рисунок 4.3), тоді

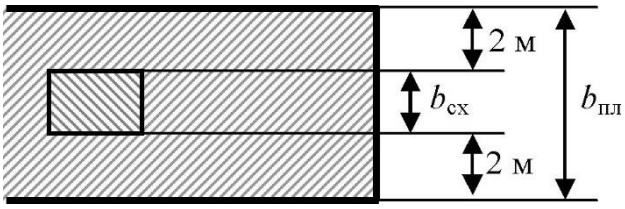


Рисунок 4.3 – Схема для розрахунку ширини пасажирської платформи з урахуванням розміщення на ній споруд чи пристроїв посередині

$$b_{\text{пл}} > b_{\text{сх}} + 4 . \tag{4.5}$$

– якщо схід розташований у кінці платформи (див. рисунок 4.4), тоді

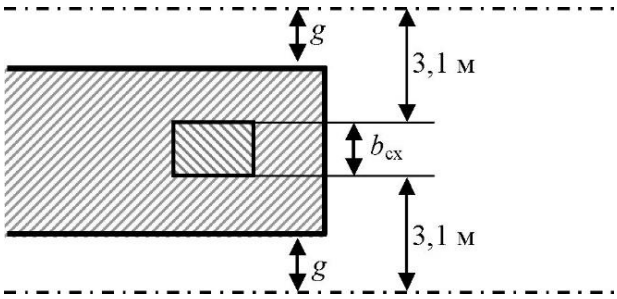


Рисунок 4.4 – Схема для розрахунку ширини пасажирської платформи з урахуванням розміщення на ній споруд чи пристроїв у кінці

$$b_{пл} \geq b_{сх} + 2(3{,}1 - g), \tag{4.6}$$

де g – габаритна відстань до платформи; згідно [6] для високих платформ габаритна відстань рівна 1,920 м, а для низьких – 1,745 м.

Нормативні дані для виконання розрахунків подані в таблиці 4.1.

Таблиця 4.1

Нормативні дані для виконання розрахунків ширини платформи

Категорія поїзда	c, шт	$n_в$, чол./хв		v, м/с	p, м/с
		високі	низькі		
приміський	2	70–75	35–40	1,2–1,3	0,9–1,0
дальній	1	25–30	12–15	1,0–1,1	1,2–1,3

4.2 Приклад розв’язання задач. Задача 3

Умова задачі

На залізничній станції проектується пасажирська платформа на якій передбачається певний тип операцій з пасажирськими поїздами.

Кількість вагонів у составі пасажирських поїзді складає m, довжина пасажирського вагона $l_в$, а середня населеність вагона $M_в$ пасажири. Посередині платформи розташований тунель, ширина сходів у який $b_{сх}$, а вхід у тунель можливий тільки з одного боку платформи.

Необхідно розрахувати ширину такої пасажирської платформи.

Вихідні дані

Вихідні дані для вирішення даної задачі наведені в таблиці 4.2.

Таблиця 4.2

Вихідні дані для вирішення задачі

Позначення	Категорія поїзда	Тип платформи	m, *ваг*	$l_{в}$, *м*	$M_{в}$, *пас.*	Перехід (входів 1/2)	$b_{сх}$, *м*
Значення	дальній	низька	9	25,0	43	тунель (1)	3,0

Заданий тип операцій: одночасна посадка та висадка пасажирів.

Розв'язання задачі

Для кількості вагонів $m = 9$ у пасажирському поїзді та середній населеності одного пасажирського вагона рівній $M_{в} = 43$ пасажири, тоді населеність поїзда становитиме $M = mM_{в} = 9 \cdot 43 = 387$ пасажирів.

Оскільки ширина платформ розраховується для дальніх поїздів, то згідно з нормативними даними (див. таблицю 4.1) кількість виходів, що відкриваються в одному вагоні, приймається $c = 1$, а кількість пасажирів, що висаджуються через один вихід за хвилину, приймаємо $n_{в} = 15$ (враховуючи, що платформа низька).

Час висадки пасажирів з поїзда визначаємо за формулою (4.1)

$$T_{в} = \frac{387}{9 \cdot 1 \cdot 15} = 2{,}9 \text{ хв.}$$

Довжина пасажирського поїзда складе $L_{с} = 9 \cdot 25 = 225$ м.

Швидкість пересування пасажира по платформі для дальніх поїздів згідно з нормативними даними приймаємо v = 1,1 м/с.

Вихід розташовано посередині платформи, а тому до розрахунку приймається половина довжини состава. Тоді час проходу поїзда пасажиром з останнього вагона до виходу з платформи за формулою (4.2)

$$T_{пр} = \frac{0{,}5 \cdot 225}{60 \cdot 1{,}1} = 1{,}7 \text{ хв.}$$

Розрахуємо відношення часу проходу пасажира від останнього вагона до часу висадки пасажирів

$$k = \frac{1,7}{2,9} = 0,59 .$$

Ширину платформи з урахуванням осіб, що проводжають, носіїв та смуг безпеки визначимо, враховуючи те, що поїзди дальні

$$b_{\text{Д}} = 1,25 \cdot \frac{387 \cdot 1,1}{225} \cdot 0,59 + 2 = 3,4 \text{ м.}$$

Далі при розрахунку ширини пасажирської платформи необхідно врахувати додаткові умови, а саме:

1. У даному випадку тунель розташований посередині платформи, а вхід у тунель можливий лише з одного боку, то при розрахунку ширини платформи за виразом (4.3) або (4.4) до розрахунку приймається лише половина довжину состава, тобто

$$b_{\text{Д}} = 1,25 \cdot \frac{387 \cdot 1,1}{0,5 \cdot 225} \cdot 0,59 + 2 = 4,8 \text{ м.}$$

2. У даному випадку передбачається одночасна посадка і висадка пасажирів з дальніх поїздів, тому ширину платформи, що була розрахована за формулою (4.4), необхідно збільшити на 3,0 .. 3,5 м, отже:

$$b_{\text{Д}} = 4,8 + 3,0 = 7,8 \text{ м.}$$

Оскільки посередині платформи розташований тунель, то ширина платформи повинна відповідати умові (4.5) $b_{\text{пл}} \geq b_{\text{сх}} + 4$, що в даному випадку $b_{\text{пл}} \geq 7$.

Умова виконується, а тому до проектування необхідно прийняти ширину платформи 7,80 м.

Контрольні запитання

1. Які основні функції повинна забезпечувати пасажирська платформа?
2. Чим відрізняється бічна платформа від острівної?
3. Якими нормативними документами регламентується мінімальна ширина пасажирських платформ?
4. Яка мінімальна ширина бічної платформи в межах вокзалу в нормальних умовах?
5. Яка мінімальна ширина острівної платформи між головними коліями в нормальних умовах?
6. Які фактори впливають на розрахункову ширину платформи?
7. Як впливають умови проектування на ширину платформи?

8. Як впливає наявність споруд чи сходів на ширину платформи?

9. Чим відрізняються два випадки розрахунку ширини платформи (час висадки більше часу проходу пасажирів та не більше за нього)?

10. Як впливає розташування та кількість виходів з платформи на її ширину?

11. Як враховуються смуги безпеки та носії при розрахунку ширини?

12. Як на ширину платформи впливає одночасна посадка та висадка пасажирів для приміських поїздів?

13. Як на ширину платформи впливає одночасна посадка та висадка пасажирів для дальніх поїздів?

14. Як на ширину платформи впливає одночасна висадка з 2-х поїздів?

РОЗДІЛ 5

Сітьове планування та управління. Правила побудови сітьових графіків та розрахунок їх параметрів

5.1 Теоретичні відомості

5.1.1 Загальна інформація про сітьове планування

Для багатьох галузей виробництва і деяких невиробничих сфер характерна наявність численних комплексів взаємопов'язаних робіт, операцій, процесів, які повинні виконуватися в певній послідовності і, по можливості, в найкоротші терміни. Так, крупні об'єкти транспортного будівництва (нові лінії, другі колії, і т. ін.) включають сотні споруд, будівництво яких вимагає виконання в певному взаємозв'язку тисяч окремих робіт.

Сітьове планування та управління (СПУ) є важливим інструментом у сфері управління проектами, який дозволяє ефективно планувати, координувати та контролювати різні процеси і ресурси для досягнення поставлених цілей. Воно використовується для управління складними проектами в будівництві, промисловості, інформаційних технологіях та інших галузях [1 – 5].

Основні концепції СПУ включають побудову сітьових моделей, аналіз шляхів і оптимізацію ресурсів.

Сітьова модель являє собою графічне відображення всіх робіт і завдань проекту у вигляді вузлів та ребер. Вона допомагає візуально уявити структуру проекту, взаємозв'язок між завданнями та час їх виконання:

– вузли (події) позначають початок і кінець певних робіт або завдань у проекті;

– ребра (дуги) відображають залежності між подіями та показують послідовність виконання робіт.

Існує декілька методів сітьового планування, які дозволяють створювати та аналізувати сітьові моделі:

– метод критичного шляху (CPM) використовується для визначення повного шляху з найбільшою тривалістю, який визначає загальний час виконання комплексу робіт.

– метод оцінки і аналізу програм (PERT) забезпечує аналіз часу виконання проекту з урахуванням невизначеності та ризиків.

Критичний шлях у сітьовій моделі визначає найбільш тривалий послідовний ланцюг завдань, які потрібно виконати для завершення проекту. Це основний елемент, який впливає на загальну тривалість проекту.

Резерв часу – це час, на який можна відкласти виконання певного завдання без впливу на загальний графік проекту.

Виділяють два основні види резерву:

Сітьове планування включає кілька основних етапів, які допомагають у створенні та аналізі сітьових моделей.

1. Визначення завдань та їх послідовності: на цьому етапі ідентифікуються всі завдання, необхідні для завершення проекту, та встановлюються залежності між ними. Кожне завдання визначається як вузол у сітьовій моделі, а залежності між ними відображаються як дуги.

2. Побудова сітьової діаграми: після визначення завдань і їхніх взаємозв'язків будується сітьова діаграма, яка відображає послідовність виконання завдань і допомагає візуально уявити структуру проекту.

3. Розрахунок часових параметрів: виконується розрахунок часу виконання кожного завдання, визначається критичний шлях, а також розраховуються резерви часу. Це дозволяє визначити найбільш тривалий шлях і зрозуміти, які завдання можна відкласти без впливу на загальний графік проекту.

4. Аналіз і оптимізація ресурсів: аналізуються наявні ресурси і розподіл часу для їх оптимального використання. Це дозволяє зменшити загальну тривалість проекту і підвищити його ефективність.

5.1.2 Переваги та недоліки сітьового планування

Перевагами сітьового планування є:

– візуалізація процесів, за якої сітьова модель дозволяє наочно побачити структуру проекту і взаємозв'язки між завданнями;

– контроль за терміном, яка сприяє точному визначенню термінів виконання завдань і загального проекту;

– оптимізація ресурсів допомагає ефективно розподілити ресурси і оптимізувати їх використання;

– прогнозування ризиків забезпечує можливість аналізу потенційних ризиків і визначення критичних завдань.

Недоліки сітьового планування такі:

– процес створення сітьової моделі може бути складним і трудомістким;

– потрібна велика кількість даних для точного планування, що може ускладнювати процес;

– сітьові моделі можуть бути менш гнучкими при змінах у проекті, що ускладнює їх коригування.

5.1.3 Висновок

Сітьове планування та управління є потужним інструментом для ефективного управління проектами. Воно дозволяє детально планувати і контролювати виконання завдань, оптимізувати використання ресурсів та забезпечувати своєчасне завершення проекту. Використання сітьових моделей допомагає краще розуміти структуру проекту, аналізувати залежності між завданнями і прогнозувати можливі ризики, що сприяє підвищенню загальної ефективності проектного управління [1 – 3, 7, 8, 11].

5.2 Призначення і область застосування сітьових графіків

Найважливішим завданням в плануванні і управлінні подібними складними процесами є визначення раціонального розкладу їх виконання з деталізацією аж до кожної окремої роботи, а згодом – контроль фактичного ходу робіт, поточне керівництво на основі складених розкладів і при необхідності – оперативне корегування розкладу.

Для виконання цього завдання використовувалися лінійні (або стрічкові) графіки, подібні графікам технологічних процесів [2, 3].

Проте можливості використання стрічкових графіків для визначення оптимального управління є обмеженими. У зв'язку з цим широке розповсюдження одержала інша форма моделювання зв'язків робіт в складних комплексах, а саме – сітьові графіки. Їх переваги, порівняно з лінійними графіками, полягають не тільки в наочності, але і в

можливості застосування для оптимізації управління різних засобів математичного вивчення мереж, а також автоматичних обчислювальних машин.

Сітьові графіки досить широко застосовуються на транспорті і в транспортному будівництві. З їх допомогою здійснювалося управління будівництвом багатьох залізничних об'єктів, зокрема декількох залізничних ліній.

Сітьові графіки використовуються також в плануванні капітального ремонту колії, в розробці оптимальної технології ремонту локомотивів і вагонів в депо і на заводах, при складанні технологічних процесів вантажних і сортувальних станцій.

Методи організації робіт, засновані на аналізі сітьових графіків, одержали загальну назву – сітьове планування і управління (СПУ).

Загальною метою СПУ є підвищення ефективності планованих процесів за рахунок:

– максимально можливого скорочення термінів їх виконання;

– якнайкращого використання ресурсів (наприклад, зниження простоїв працівників і механізмів);

– зниження вартості робіт.

В методах СПУ зберігається загальний принцип всіх завдань на оптимум: єдність критерію оптимальності в межах однієї задачі.

У різних завданнях СПУ ставиться за мету або мінімізація часу при заданих ресурсах, або мінімізація ресурсів (вартості) при заданому максимально допустимому часі, або, нарешті, мінімізація витрат в ширшому розумінні з урахуванням як ресурсів, що витрачаються, так і вартісної оцінки ефекту прискорення введення об'єкту.

Робота по застосуванню методів СПУ зазвичай включає такі стадії:

1) конкретне вивчення зв'язків технологічних операцій в плановаому комплексі, аналіз технологічної послідовності і взаємних залежностей робіт, визначення норм часу на кожну роботу;

2) побудова сітьового графіка;

3) визначення параметрів сітьового графіка;

4) ув'язка розробленого графіка з наявними ресурсами (робоча сила, механізми і т. д.), оптимізація побудованого графіка, тобто зміна окремих його характеристик з метою прискорення комплексу робіт, кращого використання ресурсів, зниження вартості;

5) практичне використання сітьового графіка і розроблених на його основі нормативів для потреб планувань, контролю і управління виробництвом.

5.3 Основні поняття

Сітьовий графік – комплекс взаємозв'язаних робіт, що має вигляд мережі односторонніх дуг, які відповідають деяким операціям (роботам), а також вершин (подій), які відповідають початку і закінченню відповідних робіт.

Основними елементами сітьового графіка є подія та робота. У сітьових графіках розрізняють такі види робіт: *дійсні*, *фіктивні*, *очікування*.

Дійсна робота – це робота, яка вимагає витрат праці, матеріальних ресурсів і часу (наприклад – технічне обслуговування вагонів);

Очікування – це робота, яка не вимагає витрат праці і матеріальних ресурсів, але займає деякий час;

Фіктивна робота (залежність) – це робота, яка показує зв'язок між двома або більше подіями та яка не вимагає витрат праці, матеріальних ресурсів і часу, але вказує, що можливість початку однієї операції безпосередньо залежить від виконання іншої. Тривалість такої роботи дорівнює нулю.

Будь-яка робота в мережі з'єднує дві події: попередню (яка є для неї початковою) і наступну за нею (яка є для неї кінцевою).

На сітьовому графіку розрізняють такі види подій:

Початкова подія – це момент початку виконання всього комплексу робіт;

Завершальна подія – це кінцева подія, яка означає досягнення кінцевої мети всього комплексу робіт;

Проміжна подія – це результат однієї або декількох робіт, що представляють можливість почати одну або кілька безпосередньо наступних робіт. Тривалість проміжної події в часі завжди дорівнює нулю (адже подія визначає стан, а не процес).

Будь-яка послідовність робіт в мережевому графіку, в якій кінцева подія кожної роботи цієї послідовності збігається з початковою подією наступної за нею роботою, називається *шляхом*. Шляхи на сітьовому графіку можуть бути трьох видів:

Повний шлях – це такий шлях, початок якого збігається з початковою подією мережі, а кінець – із завершальною, тобто це будь-яка послідовність робіт, що поєднує початкову та кінцеву подію;

Шлях, що передує події – шлях від вихідної події мережі до даної;

Шлях, наступний за подією – шлях, що з'єднує подію з завершальною подією;

Шлях між подіями *i* та *j* – шлях, що з'єднує будь-які дві події *i* та *j*, з яких жодна не є вихідною або завершальною подією сітьового графіка;

Критичний шлях – це такий шлях, який має найбільшу тривалість від початкової події до завершальної, тобто це повний шлях найбільшої тривалості.

5.4 Правила побудови сітьових графіків

Роботи, що входять до загального комплексу, можуть виконуватися як паралельно, так і послідовно (див. рисунок 5.1). Також роботи можуть бути як залежними, так і незалежними ввід інших. Переваги паралельного виконання робіт в тому, що вони виконуються одночасно, при цьому загальна тривалість виконання усього комплексу операцій скорочується. Але подібна організація робіт може потребувати додаткових ресурсів.

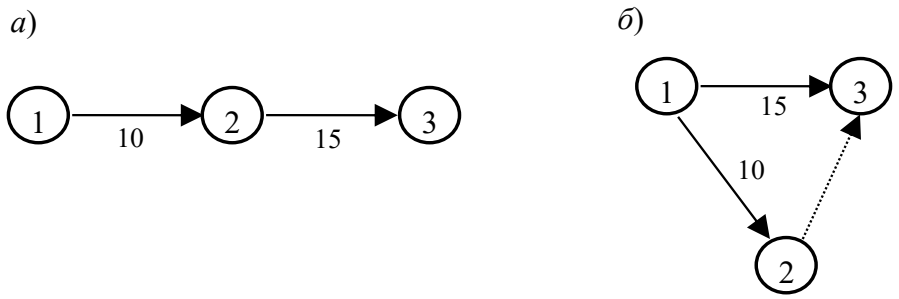


Рисунок 5.1 – Способи організації робіт на сітьовому графіку:
а) при послідовному виконанні (сумарно t = 25 *хв*);
б) при паралельному виконанні (сумарно t = 15 *хв*).

Сітьовий графік будують зліва направо. При цьому, по можливості, необхідно уникати перетинів ребер.

Послідовність побудови сітьового графіка така:.

1. Роботи, з яких починається виконання загального комплексу операцій і яким ніякі інші роботи не передують, зображуються дугами, що виходять з початкової події (входу) графіка.

2. Роботи, що завершують проект, зображуються дугами, що закінчуються в завершальній вершині (виході) графіка.

3. Кожна робота повинна бути направлена від події з меншим номером до події з більшим номером.

4. Між двома подіями може знаходитись тільки одна робота (див. рисунок 5.2).

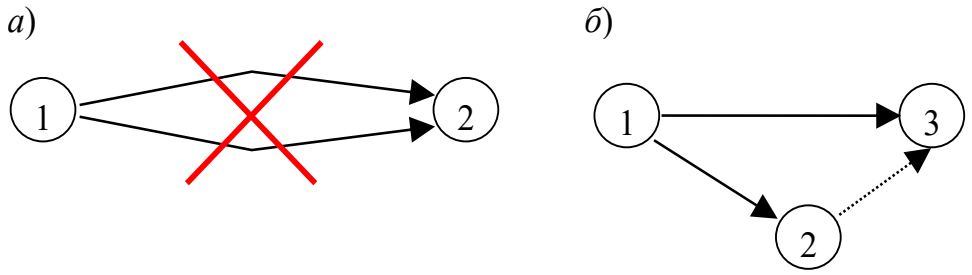


Рисунок 5.2 – Паралельне виконання робіт: а) невірно; б) вірно

5. Жоден шлях не може проходити двічі через одну і ту ж подію (див. рисунок 5.3).

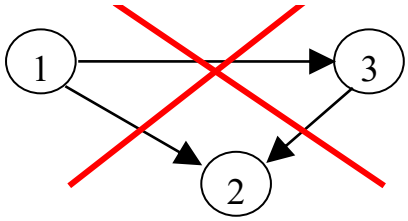


Рисунок 5.3 – Невірна побудова сітьового графіка

6. Кожна проміжна подія повинна мати хоча б одну попередню та наступну роботу, а якщо виникають тупики, то їх потрібно з'єднати з завершальною подією фіктивними роботами (див. рисунок 5.4).

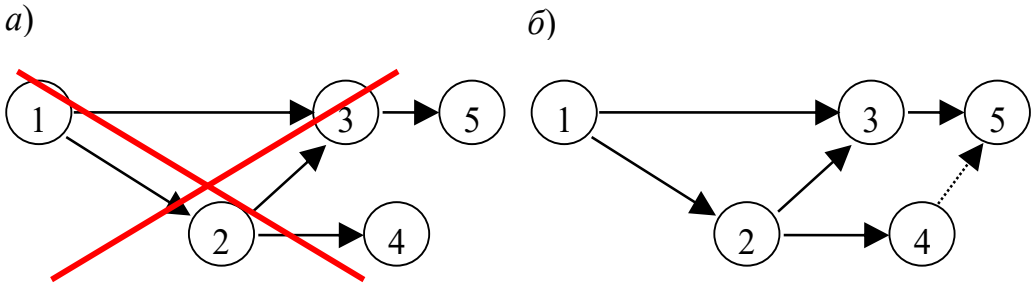


Рисунок 5.4 – Ув'язка робіт на графіку: а) невірно; б) вірно

7. Деякі роботи b_1, b_2, ... $b_п$ можуть бути розпочаті лише після закінчення інших a_1, ... a_m. В окремих випадках дотримання цього правила неможливе, якщо не ввести додаткові фіктивні дуги (недійсні роботи), що означають змістовні залежності між роботами.

8. Після побудови сітьового графіка роботи перенумеровуються двома цифрами (по номерам суміжних з даною роботою подій).

5.5 Нумерація подій сітьового графіка методом викреслювання

Алгоритм метода викреслювання

З метою забезпечення правильної нумерації подій застосовується алгоритм метода викреслювання, який формулюється таким чином:

Крок 1. Початковій події присвоїти номер 1;

Крок 2. Якщо пронумеровані всі події, то кінець алгоритму, інакше викреслити всі роботи, що виходять із пронумерованих подій.

Крок 3. Пронумерувати послідовними номерами всі події в які не входить жодна робота (роботи, що були викреслені, тепер не рахуються) і перейти на крок 2.

Застосування методу викреслювання для нумерації подій сітьового графіка

1. Нумеруємо початкову подію (див. рисунок 5.5)

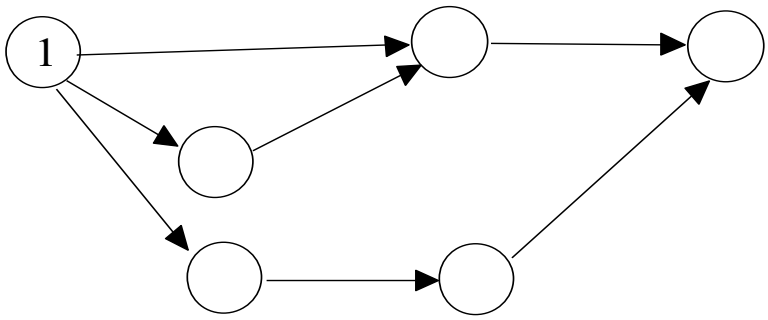


Рисунок 5.5 – Нумерація початкової події

2. Викреслюємо всі роботи, які виходять з пронумерованих подій (див. рисунок 5.6)

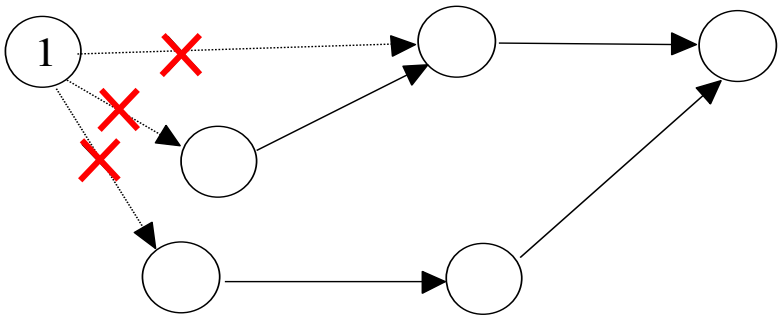


Рисунок 5.6 – Викреслення робіт, які виходять з пронумерованих подій (1 раз)

3. Нумеруємо всі ті події, в які не входить жодна робота (див. рисунок 5.7)

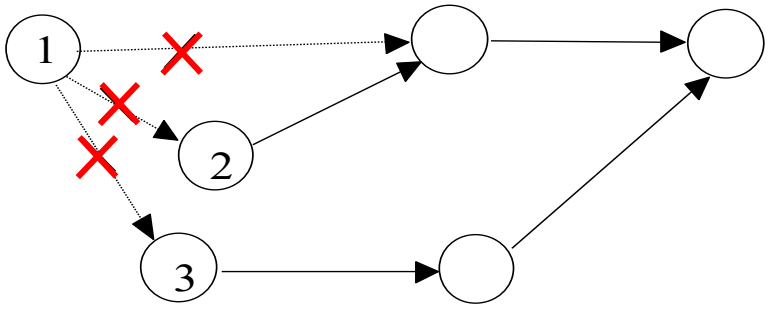


Рисунок 5.7 – Нумерація подій, в які не входить жодна робота (1 раз)

4. Викреслюємо всі роботи, які виходять з пронумерованих подій (див. рисунок 5.8)

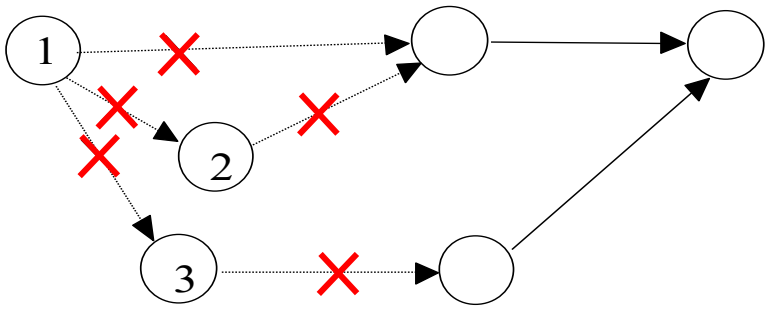


Рисунок 5.8 – Викреслення робіт, які виходять з пронумерованих подій (2 раз)

5. Нумеруємо всі ті події, в які не входить жодна робота (див. рисунок 5.9)

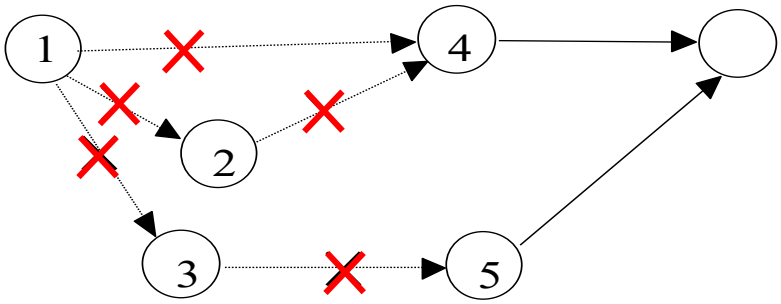


Рисунок 5.9 – Нумерація подій, в які не входить жодна робота (2 раз)

6. Викреслюємо всі роботи, які виходять з пронумерованих подій (див. рисунок 5.10)

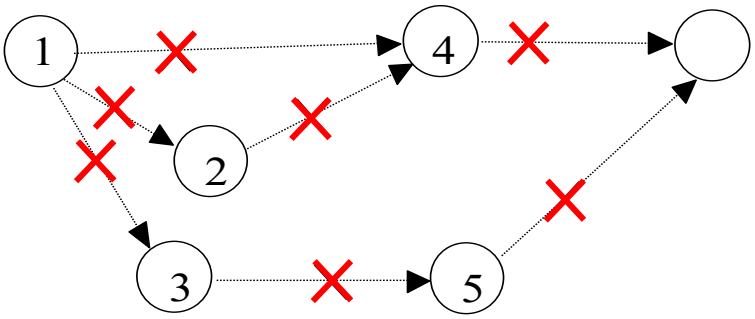


Рисунок 5.10 – Викреслення робіт, які виходять з пронумерованих подій (3 раз)

7. Нумеруємо всі ті події, в які не входить жодна робота (див. рисунок 5.11)

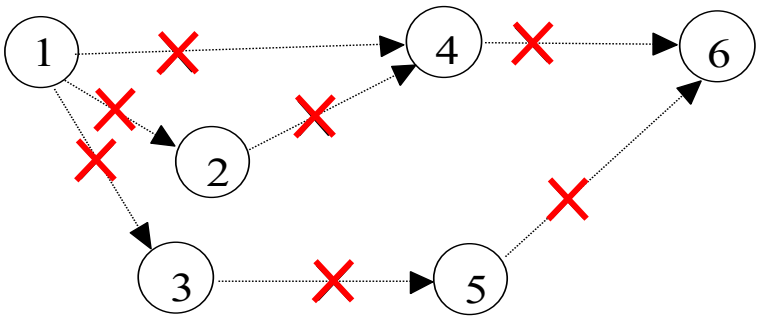


Рисунок 5.11 – Нумерація подій, в які не входить жодна робота (3 раз)

8. Оскільки всі події сітьового графіку пронумеровані, то виконання даного алгоритму закінчене.

5.6 Особливості сітьових графіків та їх параметрів

Оскільки метою складання графіка є календарне планування робіт, то кожній дузі приписується довжина, що дорівнює наперед зазначеній нормі тривалості відповідної роботи (наприклад, в днях або годинах). Умовним дугам, що зображають залежності між роботами, приписується нульова довжина.

Реальні сітьові графіки будівництва крупних об'єктів, що включають багато сотень і навіть тисячі робіт, складаються по частинам, які потім з'єднуються між собою. Складання графіків, тобто визначення всіх залежностей між роботами, їх графічне зображення і розрахунок

норм тривалості, виконується працівниками всіх будівельних і проектних організацій, що беруть участь в будівництві даного об'єкту. Бажано включення в графік моментів постачань устаткування, конструкцій і матеріалів, а також планування термінів постачань відповідно до вимог для аналізу сітьового графіка.

У багатьох випадках точно встановити норму тривалості для кожної роботи (наприклад, методами технічного нормування) не вдається. Тоді користуються приблизними даними, а при необхідності – експертними оцінками. В останньому випадку рекомендується отримати декілька незалежних експертних оцінок від різних осіб і встановити норму тривалості роботи виходячи з так званих оптимістичної $t_{опт}$, песимістичної $t_{пес}$ і середньої $t_{сер}$ оцінок:

$$T = \frac{t_{опт} + 4t_{сер} + t_{пес}}{6} \tag{5.1}$$

Надалі будемо позначати кожну роботу індексом ij, (де i – номер початкової, а j – номер кінцевої події даної роботи). Тривалість виконання роботи будемо позначати символом t_{ij}.

Сітьовий графік характеризується такими параметрами: тривалість критичного шляху, тобто тривалість виконання всього комплексу робіт $T_{кр}$; найбільш ранній можливий та найбільш пізній допустимий строк здійснення події; резерви часу для подій та робіт, які не знаходяться на критичному шляху.

5.7 Розрахунок параметрів сітьових графіків

Параметри невеликих сітьових графіків визначають секторним методом. При цьому події зображують у вигляді кругів, поділених на чотири сектори (див. рисунок 5.12).

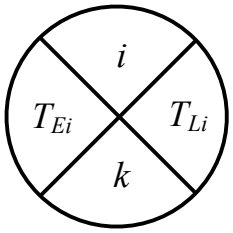


Рисунок 5.12 – Зображення події секторним методом

У верхньому секторі записують номер події i; у лівому та правому секторах відповідно найбільш ранній можливий T_{Ei} та найбільш пізній допустимий строк T_{Li} здійснення події; у нижньому секторі – номер попередньої події k.

Розрахунок починають з початкової події, проставляючи $T_{Ei} = 0$.

Потім розраховують T_{Ej}, послідовно переходячи від подій з меншими номерами до подій з більшими номерами, за формулою:

$$T_{Ej} = \max_{i}(T_{Ei} + t_{ij}), \tag{5.2}$$

де i – номер попередньої події;

t_{ij} – тривалість роботи ij.

Номер події i, з якої було розраховано T_{Ej}, проставляється в нижньому секторі.

Дійшовши до завершальної події з лівого в правий сектор проставляють тривалість всього комплексу робіт, тобто, тривалість критичного шляху $T_{Li} = T_{Ei} = T_{\text{кр}}$.

Далі розраховують T_{Li}, послідовно переходячи від подій з більшими номерами до подій з меншими номерами (рухаючись у зворотному напрямку), за формулою:

$$T_{Li} = \min_{j}(T_{Lj} - t_{ij}), \tag{5.3}$$

де j – номери всіх подальших подій за кількістю робіт, що виходять з події i.

У початковій події повинно вийти $T_{Li} = 0$. Положення критичного шляху визначають, починаючи із завершальної події за номерами, проставленими в нижніх секторах.

Резерви подій визначають за формулою:

$$R_i = T_{Li} - T_{Ei}. \tag{5.4}$$

Повний резерв – показує допустиме збільшення тривалості роботи (або запізнення її початку), при якому тривалість критичного шляху залишається незмінною.

Повний резерв роботи визначають за формулою:

$$R_{ij}^{\text{п}} = T_{Lj} - T_{Ei} - t_{ij}. \tag{5.5}$$

Вільний резерв – показує допустиме збільшення тривалості роботи (або запізнення її початку), при якому її кінцева подія настає у свій ранній строк.

Вільний резерв роботи визначають за формулою:

$$R_{ij}^{\text{В}} = T_{Ej} - T_{Ei} - t_{ij}. \quad (5.6)$$

Незалежний резерв – показує максимальний час, на який можна змінити тривалість виконання роботи, незалежно від строків її початкової та кінцевої події.

Незалежний резерв роботи визначають за формулою:

$$R_{ij}^{\text{Н}} = T_{Ej} - T_{Li} - t_{ij}. \quad (5.7)$$

Роботи, що знаходяться на критичному шляху, не мають резервів.

5.8 Приклад розв'язання задач. Задача 4

Умова задачі

Задана структурно-часова таблиця виконання робіт.

Необхідно побудувати сітьовий графік за даною структурно-часовою таблицею.

Вихідні дані

Структурно-часова таблиця виконання робіт наведена в таблиці 5.1

Таблиця 5.1

Структурно-часова таблиця виконання робіт

№ роботи	Роботи, які повинні бути виконані раніше	Тривалість, хв
1	–	2
2	1	15
3	1	15
4	3	10
5	2	5
6	4	5
7	5,6	10

Розв’язання задачі

1. Рисуємо початкову подію (див. рисунок 5.13).

Рисунок 5.13 – Початкова подія сітьового графіку

2. На підставі структурно-часової таблиці виконуємо побудову сітьового графіка (див. рисунок 5.14).

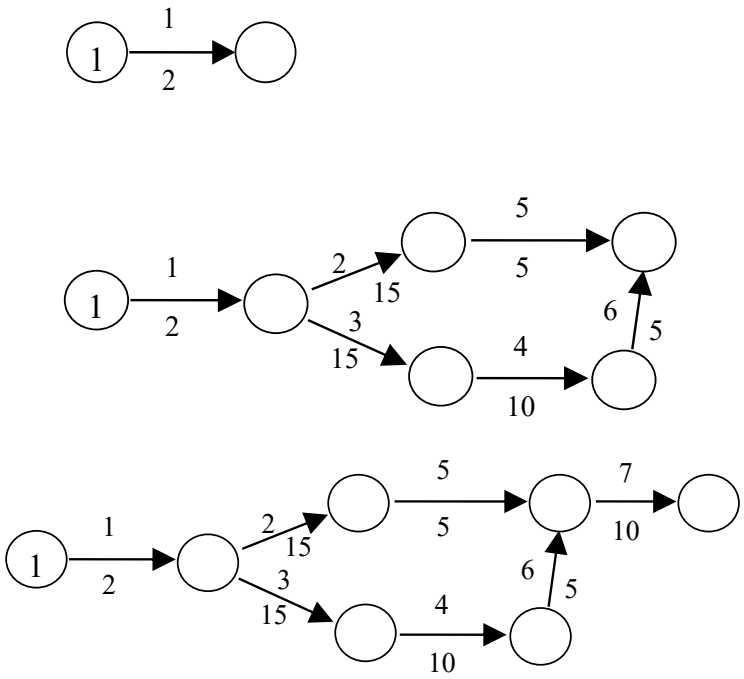


Рисунок 5.14 – Етапи побудови сітьового графіку

Виконуємо нумерацію подій (див. рисунок 5.15).

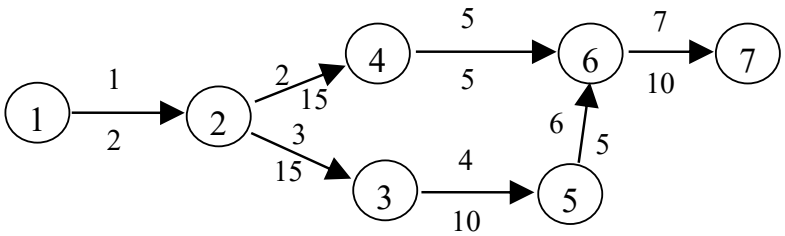


Рисунок 5.15 – Нумерація подій сітьового графіку

Скористаємось секторним методом позначення подій.

Перш за все проставимо тривалість виконання робіт на графіку (див. рисунок 5.16).

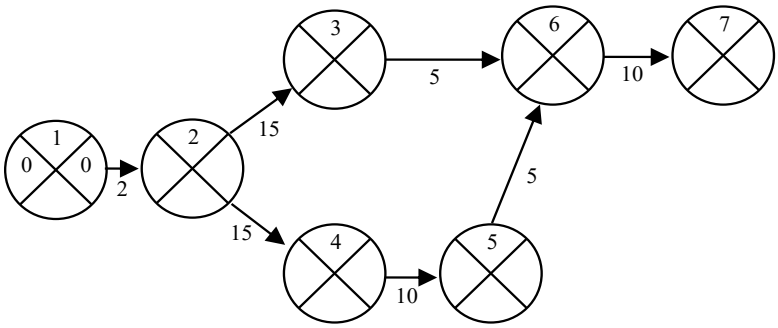


Рисунок 5.16 – Наведення тривалості виконання робіт сітьового графіку

Виконаємо розрахунок раннього часу початку виконання робіт T_E (див. рисунки 5.17 – 5.18).

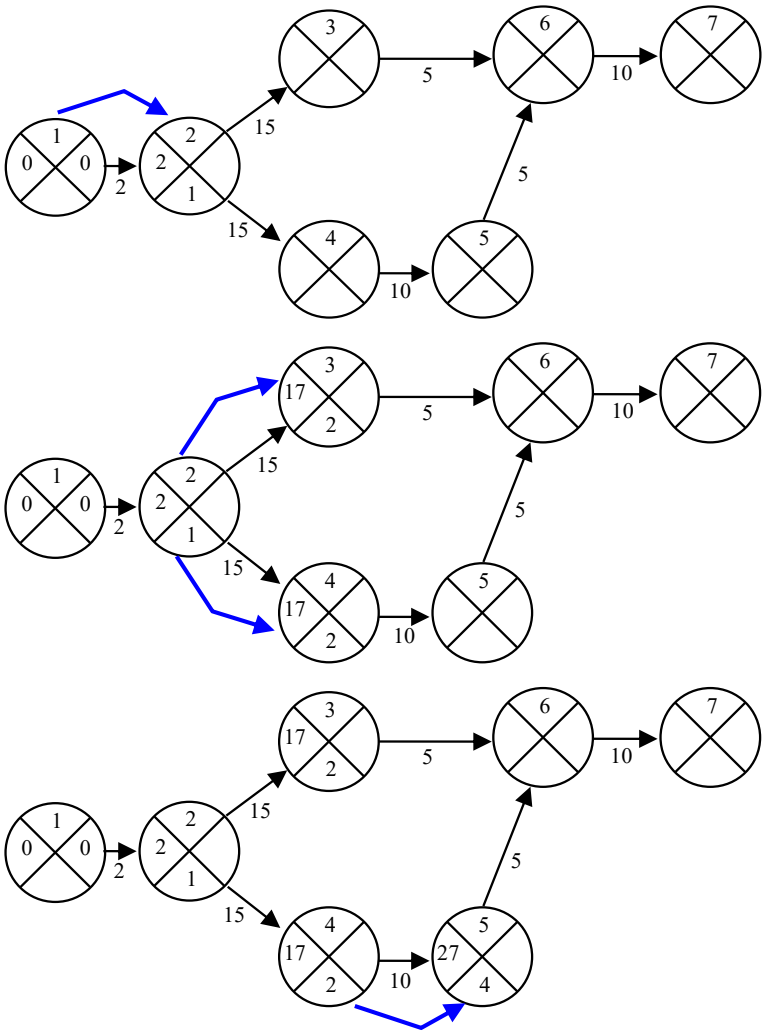


Рисунок 5.17 – Розрахунок раннього часу початку виконання робіт (*кроки 1-3*)

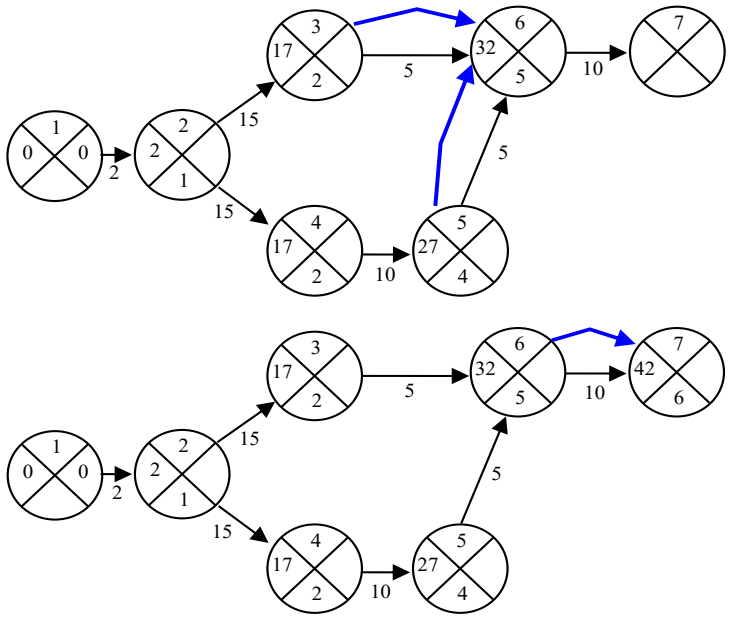


Рисунок 5.18 – Розрахунок раннього часу початку виконання робіт (*кроки 4-5*)

У завершальній події проставляємо з лівого у правий сектор значення часу виконання всіх робіт $T_{Li} = T_{Ei} = T_{кр}$ (див. рисунок 5.19).

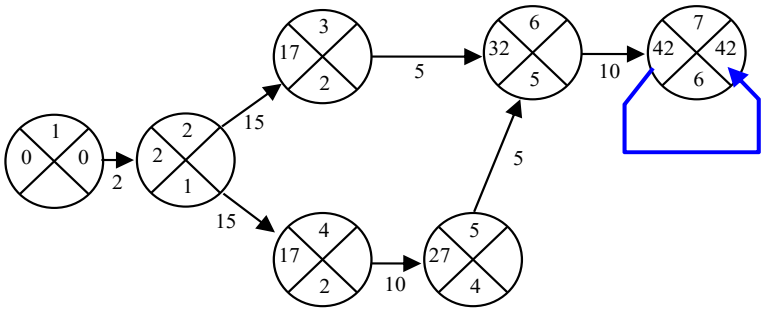


Рисунок 5.19 – Визначення параметрів завершальної події

Далі виконаємо розрахунок пізнього часу початку виконання робіт T_L (див. рисунки 5.20 – 5.21).

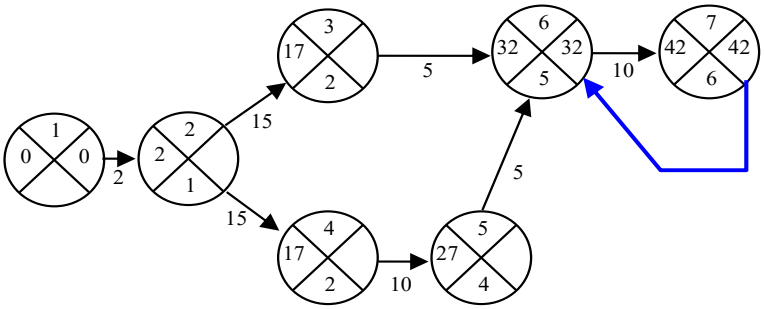


Рисунок 5.20 – Розрахунок пізнього часу початку виконання робіт (*крок 1*)

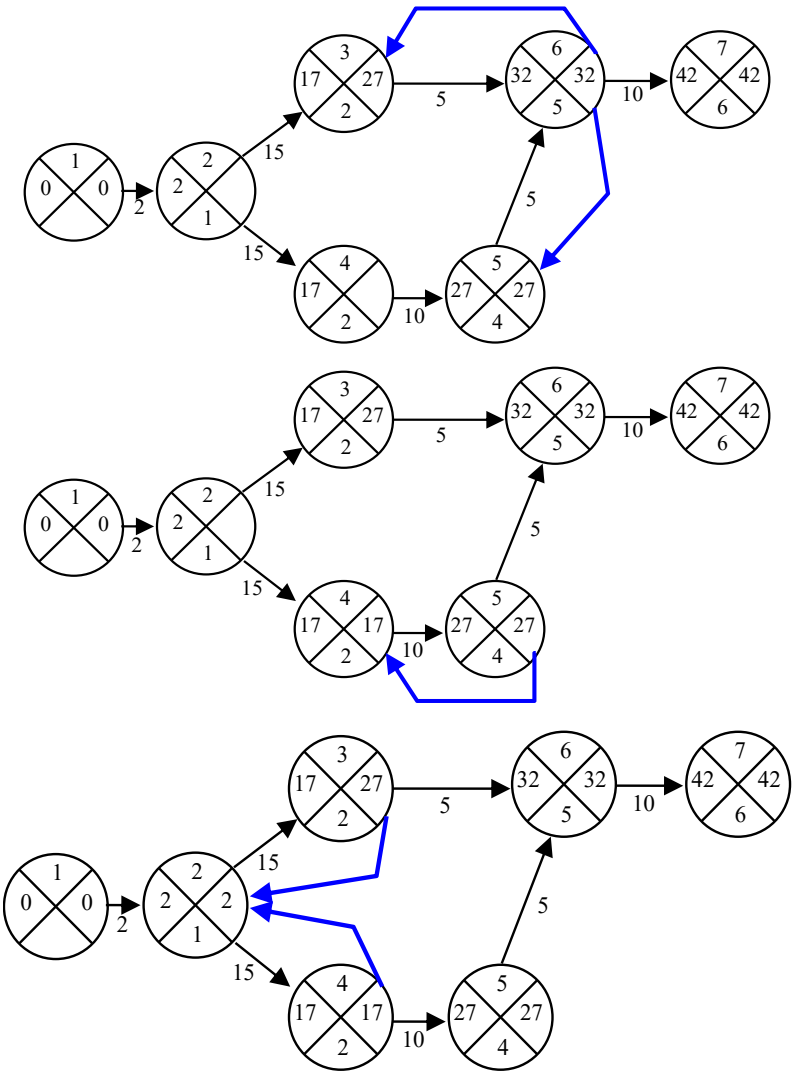


Рисунок 5.21 – Розрахунок пізнього часу початку виконання робіт (*кроки 2-4*)

Визначимо положення критичного шляху, починаючи із завершальної події за номерами в нижніх секторах подій (див. рисунок 5.22).

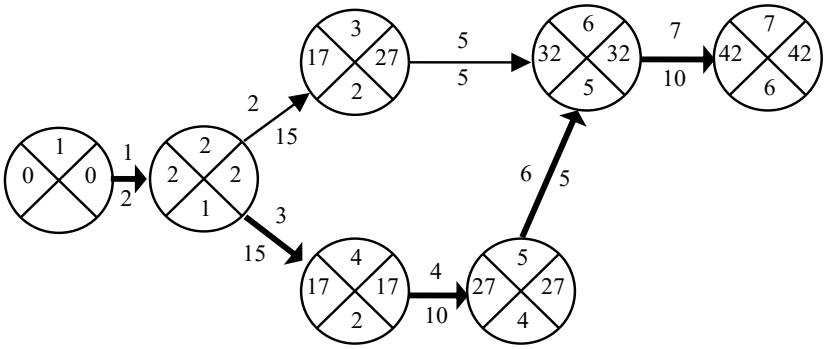


Рисунок 5.22 – Завершений сітьовий графік

Розрахунок резервів подій та робіт виконано в таблицях 5.2 – 5.3.

Таблиця 5.2

Розрахунок резервів подій

Номери робіт	Резерв
3	10

Таблиця 5.3

Розрахунок резервів робіт

Номери робіт	Резерви		
	повний	вільний	незалежний
2–3	10	–	–
3–6	10	10	–

5.9 Приклад розв’язання задач. Задача 5

Умова задачі

На деяку вантажну станцію прибувають вивізні поїзди з місцевими вагонами призначенням на ряд під’їзних колій (ПК1, ПК2, ПК3), які обслуговує 1 локомотив. Після обробки на вантажних фронтах із цих вагонів формують вивізні поїзди призначенням на найближчу технічну станцію.

Необхідно:

1. Розробити графік подачі та прибирання вагонів таким чином, щоб простій вагонів на станції був мінімальним.

2. Визначити тривалість перебування вагонів на станції та під’їзну колію, яка її лімітує.

Вихідні дані

Часові параметри обслуговування ПК наведені в таблиці 5.4

Таблиця 5.4

Часові параметри обслуговування під'їзних колій

Під'їзна колія	$t_{под/приб}$, *год*	$t_{ван}$, *год*	$t_р$, *год*
1	0,3	3,0	0,2
2	0,5	2,5	0,3
3	0,2	4,0	0,1

Розв'язання задачі

Алгоритм вирішення задачі

Вирішити дану задачу можна виконавши наступні кроки:

Крок 1. Визначення порядку подачі та прибирання вагонів на ПК.

Крок 2. Складання структурно-часової таблиці.

Крок 3. Побудова сітьового графіка обслуговування під'їзних колій та визначення усіх його основних параметрів.

Крок 4. Визначення резервів подій та робіт.

Визначення порядку подачі та прибирання вагонів на під'їзні колії.

Тривалість подачі вагонів $t_п$, їхньої обробки на вантажних фронтах $t_{ван}$ та ходу локомотива резервом $t_р$ записують в таблиці 5.5. В першу чергу на вантажні фронти подають ті групи вагонів, величина $t_п + t_{ван}$ у яких найбільша.

В даному прикладі вагони слід подавати в такій черговості: ПК3, ПК1, ПК2.

Тривалість обслуговування під'їзної колії локомотивом визначається як

$$t_л = t_п + t_р \tag{5.8}$$

Час закінчення вантажних операцій на під'їзних коліях розраховується за формулою:

$$t_з = t_п + t_{ван} + \sum_{i=1}^{n-1} t_{л,i}, \tag{5.9}$$

де $\sum_{i=1}^{n-1} t_{л,i}$ – момент початку подачі вагонів на вантажний фронт.

Вагони з під'їзних колій забирають відповідно до порядку закінчення вантажних операцій. В даній задачі вагони прибирають у такій черговості: ПК1, ПК2, ПК3.

Розрахунки виконуються в таблиці 5.5.

Таблиця 5.5

Визначення порядку обслуговування під'їзних колій

Під'їзна колія	$t_п$	$t_{ван}$	$t_р$	$t_п + t_{ван}$	Черговість подачі	$t_л$	$\sum_{i=1}^{n-1} t_{л,i}$	$t_з$	Черговість прибирання
1	0,3	3,0	0,2	3,3	2	0,5	0,3	3,6	1
2	0,5	2,5	0,3	3,0	3	0,8	0,8	3,8	2
3	0,2	4,0	0,1	4,2	1	0,3	0	4,2	3

Складання структурно-часової таблиці.

Структурно-часова таблиця містить інформацію про порядок та тривалість проведення технологічних операцій, а також визначає взаємозв'язок між роботами.

Структурно-часову таблицю обслуговування під'їзних колій наведено в таблиці 5.6.

Таблиця 5.6

Структурно-часова таблиця обслуговування під'їзних колій

№ роботи	Зміст роботи	Тривалість, *год*	Попередні роботи
1	Подача вагонів на ПК3	0,2	–
2	Вантажні операції на ПК3	4,0	1
3	Повернення локомотива на станцію з ПК3	0,1	1
4	Подача вагонів на ПК1	0,3	3
5	Вантажні операції на ПК1	3,0	4
6	Повернення локомотива на станцію з ПК1	0,2	4
7	Подача вагонів на ПК2	0,5	6
8	Вантажні операції на ПК2	2,5	7
9	Повернення локомотива на станцію з ПК2 і прямування на ПК1.	0,5	7
10	Прибирання вагонів з ПК1	0,3	5,9
11	Прямування локомотива на ПК2	0,3	10
12	Прибирання вагонів з ПК2	0,5	8,11
13	Прямування локомотива на ПК3	0,1	12
14	Прибирання вагонів з ПК3	0,2	2,13

Побудова сітьового графіка обслуговування під'їзних колій та визначення усіх його основних параметрів.

На підставі таблиці 5.6 будується сітьовий графік обслуговування під'їзних колій (див. рисунок 5.23). На графіку додатково пунктиром вказують простої вагонів на станції в очікуванні подачі на вантажні фронти та формування.

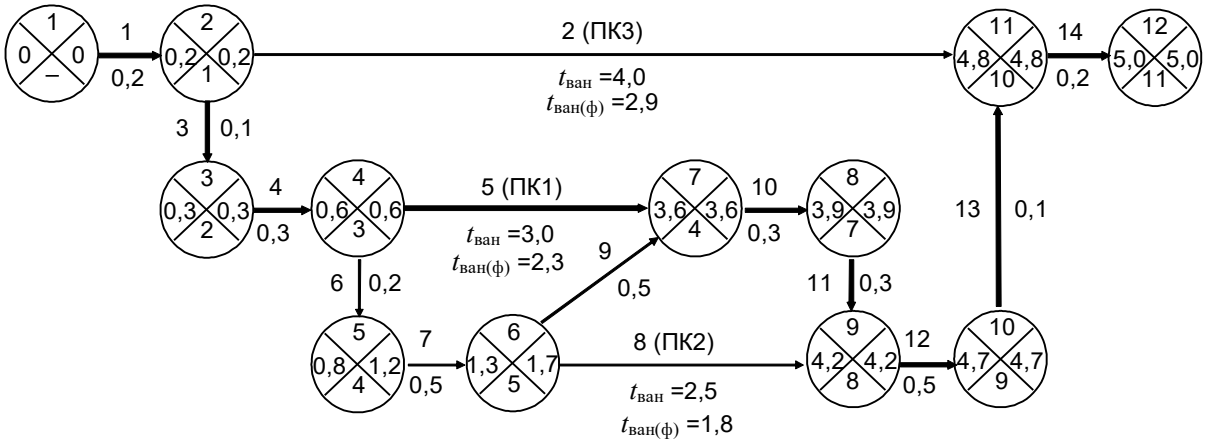


Рисунок 5.23 – Сітьовий графік обслуговування під'їзних колій

Після побудови сітьового графіка розраховують усі T_{Ei} та T_{Li} і встановлюють критичний шлях (потовщені лінії на рисунку 5.23).

Визначення резервів подій та робіт.

Величину резервів подій визначено за формулами (5.4).
Результати розрахунків наведено в таблиці 5.7.

Таблиця 5.7

Резерви подій

Номер події	R_i
5	0,4
6	0,4

Величину резервів робіт визначено за формулами (5.5) – (5.7); результати розрахунків наведено в таблиці 5.8.

Слід відзначити, що в даній таблиці нумерація робіт наведена через дріб як в традиційній формі (чисельник – номер роботи) так і в універсальній формі (знаменник – номери подій, які обмежують дану роботу). У випадку, якщо резерв дорівнює нулю, або від’ємний, то в даній таблиці ставиться прочерк «–».

Таблиця 5.8

Резерви робіт

Робота	$R_{ij}^{п}$	$R_{ij}^{в}$	$R_{ij}^{н}$
2 / 2-11	0,6	0,6	–
7 / 5-6	0,4	–	–
9 / 6-7	1,8	1,8	1,4
8 / 6-9	0,4	0,4	–

Сітьовий графік та його параметри можуть використовуватись окремо (як інструмент аналізу певного комплексу робіт), так і бути основою для подальшої оптимізації комплексу робіт, що розглядається.

Контрольні запитання

1. Для вирішення яких саме задач доцільно використовувати методи сітьового планування та управління?
2. Які є основні задачі сітьового планування та управління?
3. Які є основні елементи сітьового графіка?
4. Навести види та основні особливості робіт в теорії сітьового планування та управління.
5. Навести види та основні особливості подій в теорії сітьового планування та управління.
6. Навести види та основні особливості шляхів в теорії сітьового планування та управління.
7. Які вихідні дані потрібні для вирішення задач сітьового планування та управління?
8. Що таке структурно-часова таблиця та яке її призначення?
9. Які основні правила побудови сітьового графіка?
10. Які основні правила нумерації подій сітьового графіка?
11. Які дані слід вказувати в подіях сітьового графіка при використанні секторного способу?
12. Які є резерви робіт сітьового графіка та як вони визначаються?
13. Які є резерви подій сітьового графіка та як вони визначаються?
14. Чому дорівнюють резерви подій та робіт, які лежать на критичному шляху сітьового графіка?

РОЗДІЛ 6

Оптимізація технології роботи методами сітьового планування та управління

6.1 Теоретичні відомості

Методи сітьового планування та управління, як зазначалось раніше, можуть використовуватись для оптимізації технологічних процесів роботи як магістральних так і промислових станцій [12, 13].

У багатьох випадках одна і та ж виробнича операція може виконуватися різними технологічними та організаційними методами, які вимагають різних термінів виконання. У загальному випадку як час, так і вартість кожної роботи залежать від методу її виконання.

Назвемо в цьому випадку більш повільний і дешевий метод «нормальним» режимом, а більш швидкий і дорогий – «форсованим» режимом виконання роботи. Можливість використання форсованих режимів дозволяє прискорити виконання усього комплексу робіт, у порівнянні з критичним терміном, за рахунок деякого подорожчання проекту.

У зв'язку з цим в різних випадках можуть виникнути 2 типи завдань вибору оптимальних режимів робіт [2, 3].

Завдання 1-го типу: задано ліміт фінансових витрат *Л*, достатній для того, щоб деякі (але не всі) роботи перевести з нормальних режимів на форсовані. При цьому необхідно визначити мінімальний термін закінчення проекту, якщо допустити повне використання ліміту; встановити, на яких роботах повинні за цих умов застосовуватися форсовані режими.

Завдання 2-го типу: задано термін виконання проекту *Т*, в який неможливо укластися, виконуючи всі роботи в нормальних режимах. При цьому потрібно визначити, які з робіт необхідно перевести на форсовані режими, для того, щоб виконати проект у заданий термін з найменшими витратами.

6.2 Приклад розв'язання задач. Задача 6

Умова задачі

Розглянемо приклад оптимізації умовної мережі (транспортної або виробничої) з використанням сітьового планування та управління.

Нехай задана структурно-часова таблиця виконання деяких операцій (див. таблицю 6.1). При цьому деякі роботи можуть виконуватися у форсованому режимі.

Необхідно побудувати сітьовий графік та виконати оптимізацію технологічного процесу виконання усього комплексу операцій, якщо ліміт коштів складає $Л_0$.

Вихідні дані

Структурно-часова таблиця виконання технологічних операцій наведена в таблиці 6.1.

Таблиця 6.1

Структурно-часова таблиця виконання технологічних операцій

№ роботи	Роботи, які повинні бути виконані раніше	Тривалість виконання роботи у «нормальному» режимі, T, хв.	Тривалість виконання роботи у «форсованому» режимі, $T_ф$, хв.	Вартість скорочення тривалості роботи на 1 хв., у.о./хв.
1	–	22	20	4
2	–	5	–	–
3	2	6	–	–
4	3	4	–	–
5	4	8	5	10
6	1, 5	16	14	6
7	4	16	15	3

Ліміт коштів слід прийняти рівним $Л_0$ = 100 у.о.

Розв'язання задачі

За даними структурно часової таблиці (див. таблицю 6.1) побудовано сітьовий графік, який наведено на рисунку 6.1.

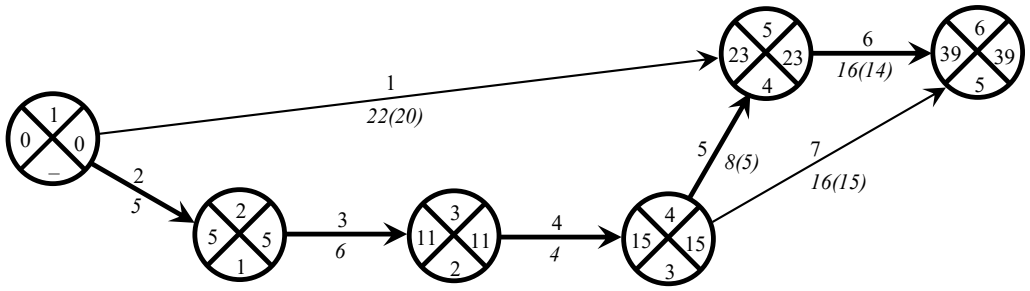


Рисунок 6.1 – Сітьовий графік виконання комплексу операцій

Тривалість критичного шляху складає $T_{кр}$ =39 хв.

Для того, щоб скоротити виконання усього комплексу операцій, необхідно перевести у форсовані режими ті роботи, які знаходяться на критичному шляху графіка, адже скорочення тривалості виконанні інших робіт (роботи 1–5 та 4–6) не призведе до скорочення критичного шляху. Отже, у даному прикладі доцільно скорочувати роботи 4–5 та 5–6, тому що є можливість перевести їх у форсовані режими.

Оптимізацію сітьових графіків (особливо великих) зручніше виконувати, якщо представити їх у спрощеному вигляді. Крім того, слід пам’ятати, що якщо частиною сітьового графіка (підграф) є множина робіт, пов'язаних з іншими роботами лише в двох подіях, що є початковою і кінцевою подіями даного підграфа, то цей підграф можна замінити на графіку однією умовною комплексною роботою, що включає всі інші, а її тривалість може бути визначена знаходженням критичного шляху на замінюваному підграфі.

У даному прикладі комплекс робіт 1–3–4, який пов'язаний з рештою сітьового графіка тільки у двох подіях (1 та 4), можна замінити однією комплексною роботою 1–4, тривалість якої дорівнюватиме загальній тривалості робіт 1, 3 та 5, які на графіку розташовуються послідовно. Таким чином тривалість роботи 1–4 складатиме: $T_{1\text{-}4}$ = 5+6+4 = 15 хв. У спрощеному вигляді сітьовий графік, зображений на рисунку 6.1, матиме такий вигляд (див. рисунок 6.2):

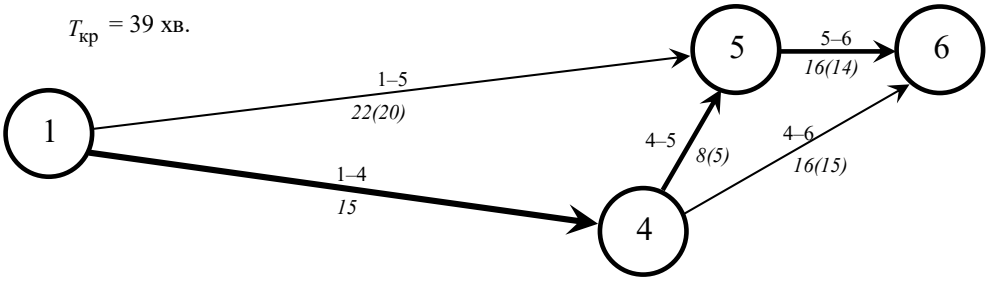


Рисунок 6.2 – Сітьовий графік у спрощеному вигляді

Номери робіт позначено двома цифрами (за номерами суміжних з даною роботою вершин). Отже, необхідно перевести у форсовані режими ті роботи, які знаходяться на критичному шляху графіка (роботи 4–5 та 5–6). При чому розпочнемо з тієї роботи, вартість скорочення якої менша, тобто з роботи 5–6.

При скороченні тривалості виконання будь-якої роботи, що лежить на критичному шляху, необхідно пам'ятати, що за певних умов, критичний шлях може змінитися (якщо тривалість виконання робіт на критичному шляху стане меншою, ніж на паралельному шляху графа). Тому при виконанні оптимізації необхідно враховувати резерви, які мають роботи, що лежать на паралельних шляхах. Тому для подальшої оптимізації розрахуємо величину повного резерву для робіт, які не лежать на критичному шляху (роботи 1–5 та 4–6), за формулою (2.5).

$$T^{\text{п}}_{1\text{-}5} = 23 - 0 - 22 = 1\ хв;$$

$$T^{\text{п}}_{4\text{-}6} = 39 - 15 - 16 = 8\ хв.$$

У даному прикладі, з вершини 4 у вершину 6 можна потрапити двома шляхами (рисунок 6.2): через роботи 4–5 та 5–6, а також іншим паралельним шляхом – через роботу 4–6, повний резерв якої складає 8 хв. За умови, якщо прийняти момент початку події 4 рівним 0, то на часовій осі це виглядатиме таким чином (див. рисунок 6.3).

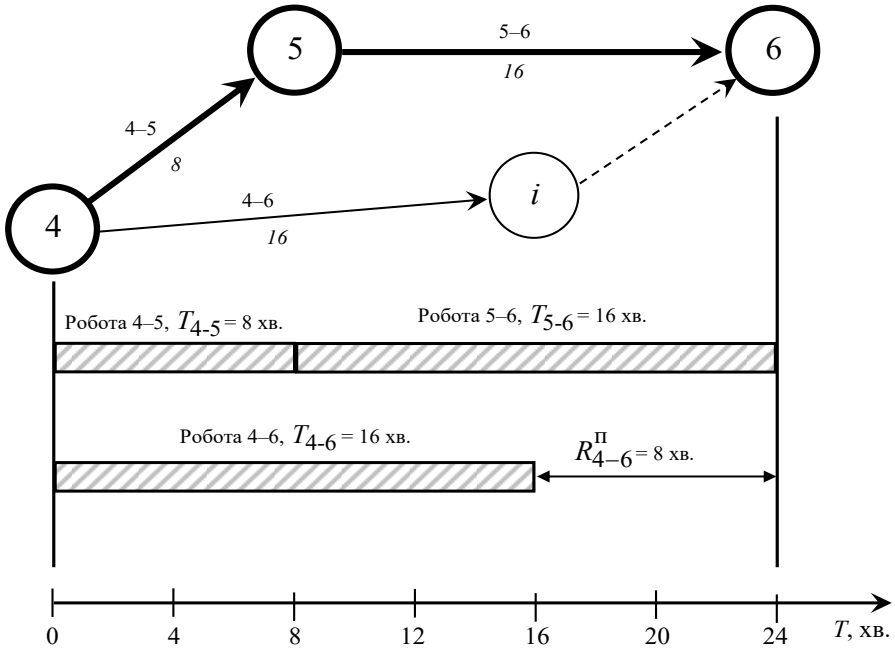


Рисунок 6.3 – Фрагмент сітьового графіка на часовій осі

У випадку, якщо роботу 5–6 зменшити на величину $\Delta t > 8$ хв., то критичний шлях проходитиме через роботу 4–6 (див. рисунок 6.4).

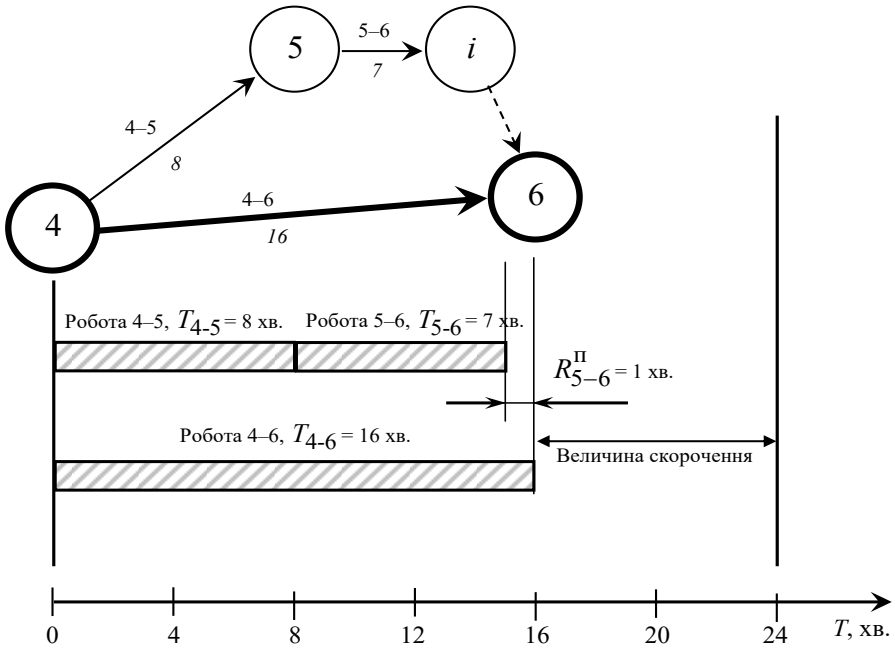


Рисунок 6.4 – Зміна критичного шляху на сітьовому графіку

У випадку, якщо роботу 5–6 зменшити на величину Δt = 8 хв., то критичний шлях розгалужується і проходитиме одночасно через роботу 4–6 та через роботи 4–5 та 5–6 (див. рисунок 6.5).

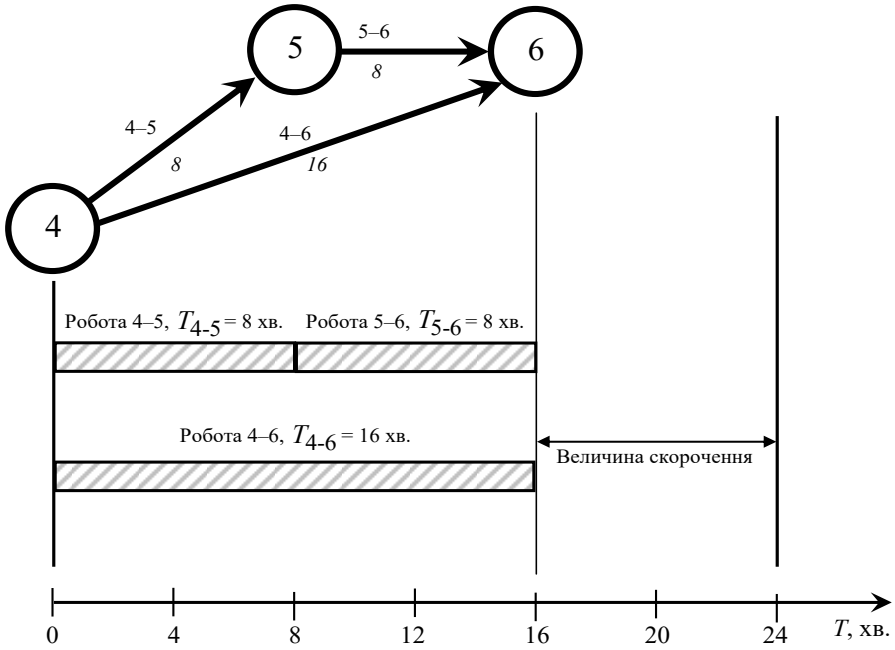


Рисунок 6.5 – Розгалуження критичного шляху на сітьовому графіку

У такому випадку для подальшого скорочення усього комплексу операцій необхідно буде скорочувати одночасно роботи 5–6 та 4–6 або роботи 4–6 та 4–5, або усі три роботи одразу.

Крім того, перед черговою ітерацією необхідно слідкувати на наявною кількістю коштів. Так, наприклад, якщо на черговому етапі необхідно скорочувати тривалість виконання роботи 5–6 (вартість скорочення на одиницю часу складає 6 у.о./хв.), при цьому в наявності є тільки 11 у.о., то це означає, що на даному етапі можна буде скоротити тривалість виконання даної роботи не більше, ніж на 1 хв. Отже, величину зменшення тривалості виконання операцій на черговому кроці γ можна визначити з виразу:

$$\Delta t_k = \min\left[T - T_{(ф)}; R_{ij}^{п}; \frac{Л_k}{C_k}\right], \qquad (6.1)$$

де $T, T_{(ф)}$ – тривалість виконання операцій відповідно при нормальному і форсованому режимах виконання;

$R_{ij}^{п}$ – величина повного резерву по паралельних шляхах графіку;

$Л_k$ – ліміт коштів, виділених на скорочення тривалості виконання операцій;

C_k – вартість скорочення тривалості операцій на одиницю часу.

Важливо зауважити, що кожну з величин у квадратних дужках виразу (6.1) необхідно округлювати до меншого цілого.

Отже, на першому етапі величина скорочення операцій складатиме:

$$\Delta t_1 = \min\,[16–14;\ 8;\ 100/6] = \min\,[2;\ 8;\ 16{,}67] = 2 \text{ хв.}$$

Таким чином, після першої ітерації, тривалість роботи 5–6, а відповідно і тривалість критичного шляху ($T_{кр}$), вдалося скоротити на Δt_1 = 2 хв. Загальна тривалість виконання усього комплексу робіт тепер складає $T_{кр}$ = 37 хв.

Сітьовий графік після першої ітерації наведено на рисунку 6.6.

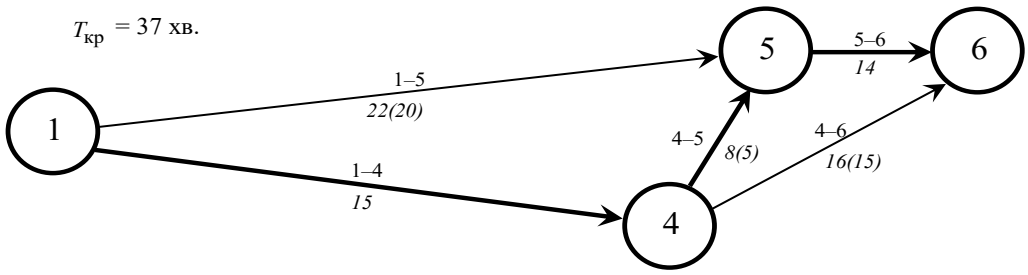


Рисунок 6.6 – Сітьовий графік після першої ітерації

Сума коштів, яка була витрачена на скорочення робіт, складає:

$$\sum C_k^{\text{скор}} = \Delta t_k \cdot C_{ij}^{\text{скор}} \qquad (6.2)$$

де Δt_k – величина скорочення операцій на кроці *k*;

$C_{ij}^{\text{скор}}$ – вартість скорочення роботи *i-j* на одиницю часу на кроці *k*.

Тоді, для першого кроку $k = 1$, $\Delta t_1 = 2\ хв$, $C_{5-6} = 6\ у.о.$ і отримаємо

$$\sum C_1^{\text{скор}} = \Delta t_1 \cdot C_{5-6} = 2 \cdot 6 = 12\ у.о.$$

Ліміт коштів, який залишається після даного кроку скорочення, розраховується за формулою

$$Л_k = Л_{k-1} - \sum C_k^{\text{скор}}, \qquad (6.3)$$

де $Л_k$ – наявний ліміт коштів на кроці *k*;

$\sum C_k^{\text{скор}}$ – загальна вартість скорочення робіт на одиницю часу (1 хв) на кроці *k*.

Ліміт коштів, який залишається після даного скорочення становить:

$$Л_1 = Л_0 - \sum C_1^{\text{скор}} = 100 - 12 = 88\ у.о.$$

Крім того, необхідно врахувати, що скоротилася величина резерву по паралельному шляху.

Величина резерву роботи 4–6 тепер складе:

$$R_{4-6}^{\text{п}} = 8 - \Delta t_1 = 8 - 2 = 6\ хв.$$

Графічно величина резерву роботи 4–6 після скорочення показана на рисунку 6.7.

Тривалість виконання роботи 5–6 після першого етапу складає $T_{5\text{-}6} = T_{5\text{-}6(ф)} = 14$ хв.

Подальше її скорочення неможливе. Тому надалі необхідно скорочувати роботу 4–5.

У даному прикладі, з вершини 1 у вершину 5 можна потрапити двома шляхами (див. рисунок 6.6): через роботи 1–4 та 4–5, а також іншим паралельним шляхом – через роботу 1–5, повний резерв якої розрахований раніше і складає 1 хв.

За умови, якщо прийняти момент початку події 1 рівним 0, то на часовій осі це виглядатиме таким чином (див. рисунок 6.8).

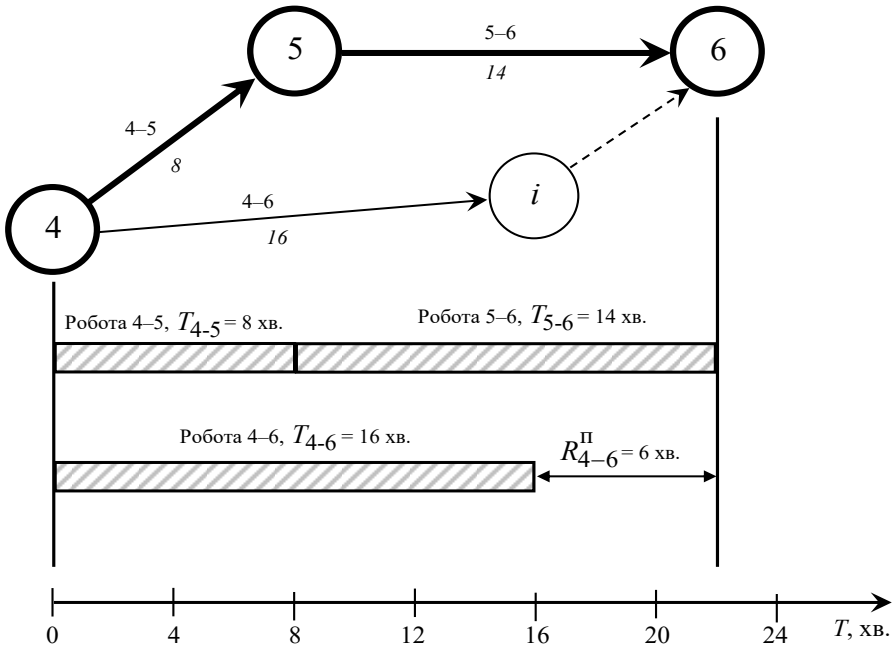


Рисунок 6.7 – Скорочення резерву для роботи 4–6

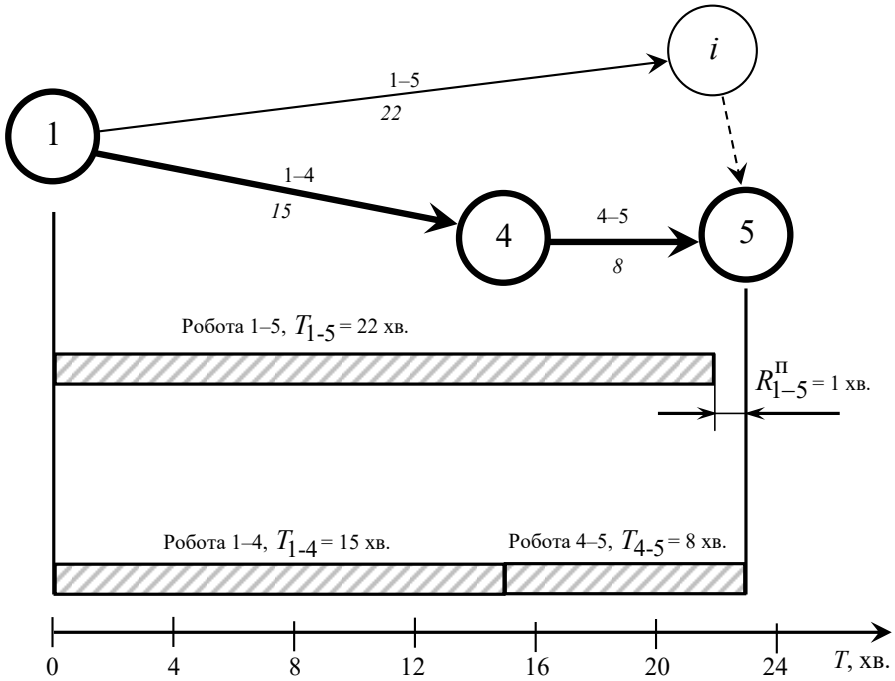


Рисунок 6.8 – Фрагмент сітьового графіка на часовій осі

У випадку, якщо роботу 4–5 зменшити на величину $\Delta t > 1$ *хв*, то критичний шлях проходитиме через роботу 1–5 (див. рисунок 6.9).

У випадку, якщо роботу 4–5 зменшити на величину $\Delta t = 1$ *хв*, то критичний шлях розгалужуватиметься (див. рисунок 6.10).

Отже, на другому етапі величина скорочення операцій складатиме:

Δt_2 = min [8 – 5; min(1; 6); 88 / 10] = min [3; 1; 8,8] = 1 хв.

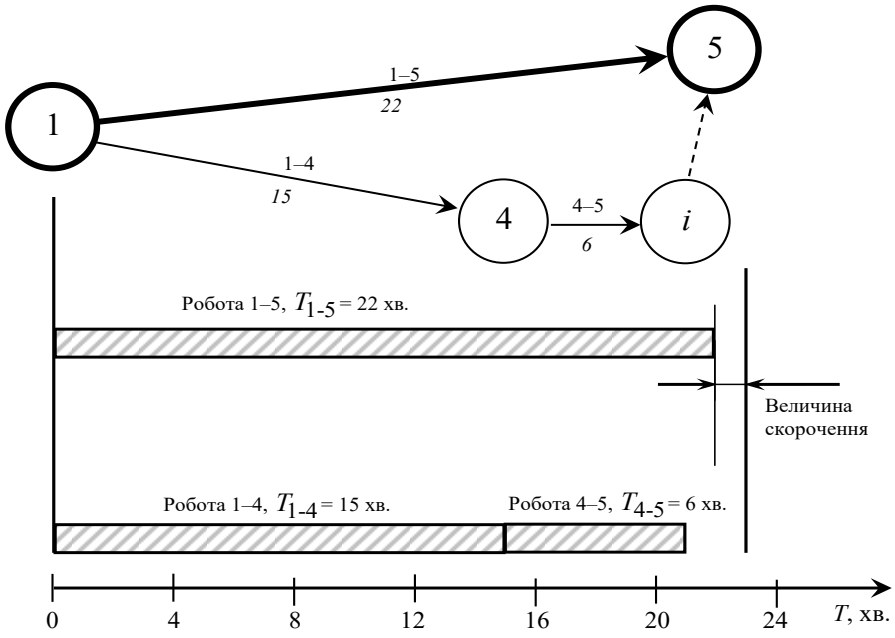


Рисунок 6.9 – Зміна критичного шляху на сітьовому графіку

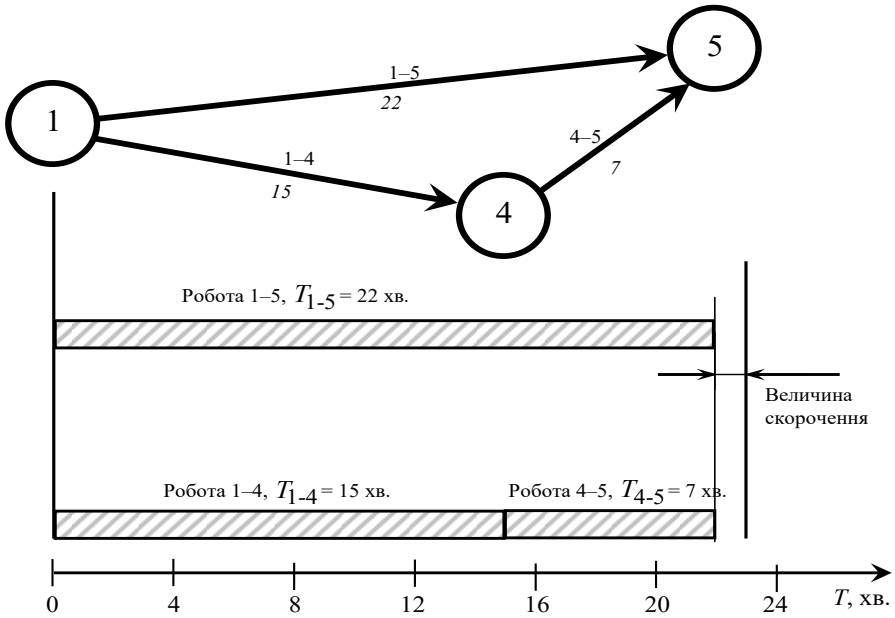


Рисунок 6.10 – Розгалуження критичного шляху на сітьовому графіку

Таким чином, після другої ітерації, тривалість роботи 4–5, а відповідно і тривалість критичного шляху ($T_{кр}$), вдалося скоротити на Δt_2 = 1 хв. Загальна тривалість виконання усього комплексу робіт тепер складає $T_{кр}$ = 36 хв.; тривалість роботи 4–5 складає $T_{4\text{-}5}$ = 7 хв., а резерву у роботи 1–5 більше немає.

Це означає, що критичний шлях розгалужується на два шляхи так, як показано на рисунку 6.11.

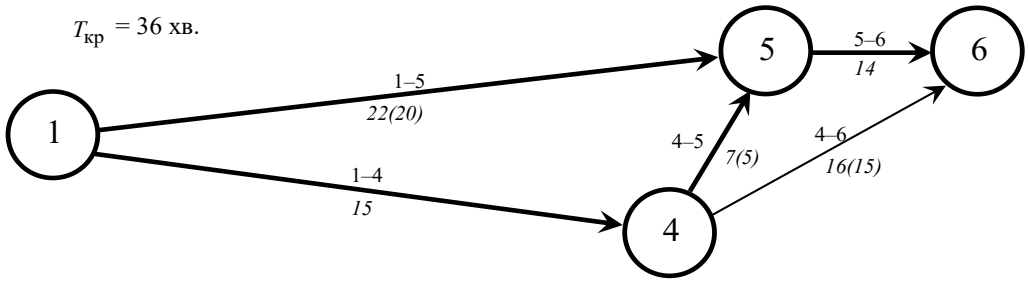


Рисунок 6.11 – Сітьовий графік після другої ітерації

Сума коштів, яка була витрачена на скорочення робіт, складає:

$$\sum C_2^{скор} = \Delta t_2 \cdot C_{4-5} = 1 \cdot 10 = 10\ у.о.$$

Ліміт коштів, який залишається після цього, становить:

$$Л_2 = Л_1 - \sum C_2^{скор} = 88\ -\ 10\ =\ 78\ у.о.$$

Крім того, необхідно відмітити, що робота 4–6 паралельна роботі 4–5. Це означає, що резерв роботи 4–6 також скоротиться з 6 до 5 хв. ($R^{п}_{4\text{-}6}$ = 5 хв.).

Далі необхідно одночасно скорочувати тривалість робіт 4–5 та 1–5 для того, щоб скоротити загальну тривалість виконання всіх операцій, тобто тривалість критичного шляху $T_{кр}$. Вартість скорочення на одиницю часу буде дорівнювати сумі вартості скорочення робіт 4–5 та 1–5, тобто C_3 = 10 + 4 = 14 у.о. / хв.

$$\Delta t_3 = \min\ [\min(7–5;\ 22–20);\ 5;\ 78/14] = \min\ [2;\ 5;\ 5{,}57] = 2\ хв.$$

Таким чином, після третьої ітерації, тривалість робіт 4–5 та 1–5, а відповідно і тривалість критичного шляху ($T_{кр}$), вдалося скоротити на Δt_3 = 2 хв. (див. рисунок 6.12).

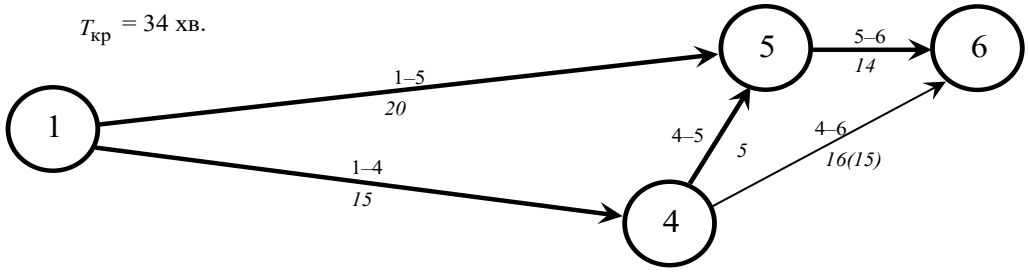


Рисунок 6.12 – Сітьовий графік після третьої ітерації

Загальна тривалість виконання усього комплексу робіт тепер складає $T_{кр}$ = 34 хв; тривалість роботи 4–5 складає $T_{4\text{-}5}$ = 5 хв, тривалість роботи 1–5 складає $T_{1\text{-}5}$ = 5 хв, а резерв роботи 4–6 буде $R^{п}_{4-6} = 3\ хв$.

Подальше скорочення тривалості виконання операцій неможливе, оскільки у всіх робіт на критичному шляху величина T дорівнює величині $T_{(ф)}$ ($T = T_{(ф)}$).

Скорочення тривалості виконання операцій також припиняється і в тому випадку, коли наявної кількості коштів не вистачає на скорочення операцій навіть на одиницю часу.

6.2 Приклад розв'язання задач. Задача 7

Умова задачі

Розглянемо приклад оптимізації технологічного процесу транспортних мереж з використанням сітьового планування та управління, на прикладі виконання обслуговуванні під'їзних колій локомотивом вантажної станції.

Як було відмічено вище, в різних випадках можуть виникнути 2 типи завдань вибору оптимальних режимів робіт.

Завдання 1-го типу: на основі даних таблиці 6.2 та розробленого сітьового графіка обслуговування під'їзних колій (див. рисунок 5.7), а також маючи певний ліміт коштів $Л_0$, необхідно максимально скоротити сумарну тривалість обслуговування під'їзних колій ПК 1, ПК 2 та ПК 3. При цьому деякі роботи сітьового графіка, а саме вантажні операції на під'їзних коліях, можуть виконуватися у форсованому режимі.

Завдання 2-го типу: маючи необмежений резерв коштів та не враховуючи обмеження зі скорочення тривалості вантажних операцій до величини $t_{ван(ф)}$, на основі даних таблиці 6.2 та розробленого сітьового графіка обслуговування під'їзних колій (див. рисунок 5.7) слід визначити, на скільки необхідно скоротити виконання вантажних операцій на кожній ПК для того, щоб сумарна тривалість обслуговування під'їзних колій ПК 1, ПК 2 та ПК 3 скоротилась на $dt = 0{,}9$ год.

Вихідні дані

Тривалість виконання вантажних операцій в нормованому ($t_{\text{ван}}$) та форсованому ($t_{\text{ван(ф)}}$) режимах, а також вартість скорочення тривалості операцій на 0,1 год (C) наведені в таблиці 6.2.

Таблиця 6.2

Характеристика під'їзних колій

Під'їзна колія	$t_{\text{ван}}$, *год*	$t_{\text{ван(ф)}}$, *год*	C, *у. о.* за *0,1 год*
1	3,0	2,3	4
2	2,5	1,8	3
3	4,0	2,9	5

Сітьовий графік обслуговування під'їзних колій, який було побудовано на рисунку 5.23 на основі структурно-часової таблиці 5.6, продублюємо на рисунку 6.13.

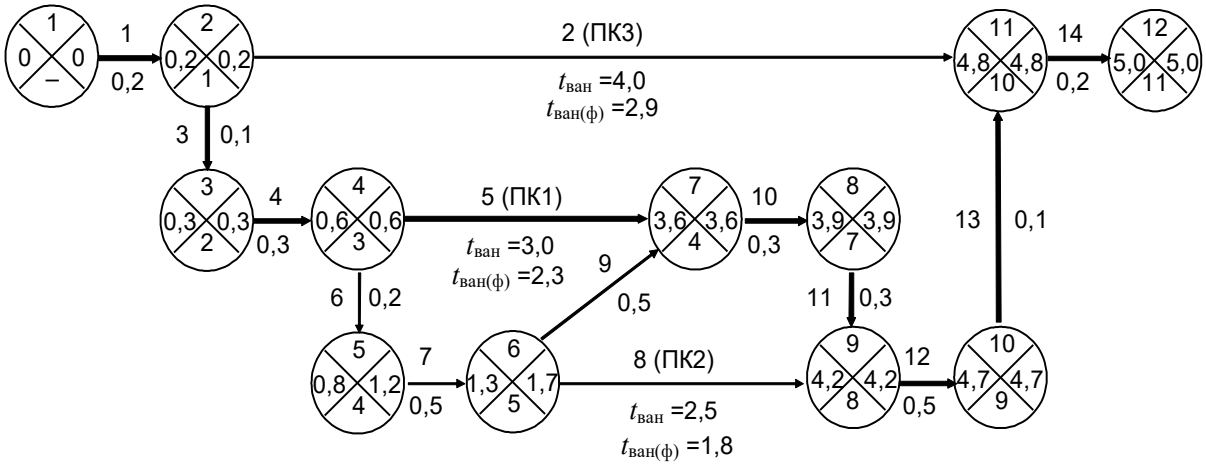


Рисунок 6.13 – Сітьовий графік обслуговування під'їзних колій

Ліміт коштів слід прийняти рівним $Л_0$ = 29 у.о.

Розв'язання задачі

Завдання 1-го типу: оптимізація сітьових графіків за умови ліміту коштів.

Крок 1. На критичному шляху лежить робота 5 (вантажні операції на ПК 1). Вартість скорочення тривалості операцій на 0,1 год на даній ПК становить 4 у.о. (див. вихідні дані у таблиці 6.1).

Величину зменшення тривалості виконання операцій на під'їзній колії 1 можна визначити з дещо модифікованого виразу (6.1)

$$\Delta t_{\gamma} = \min\left[t_{\text{ван}} - t_{\text{ван(ф)}}; R_{ij}^{\text{п}}; 0{,}1\left(Л_i / C_i\right) \right], \qquad (6.4)$$

де $t_{\text{ван}}$, $t_{\text{ван(ф)}}$ – тривалість виконання вантажних операцій на даній під'їзній колії в нормальному і форсованому режимах відповідно;

$R_{ij}^{\text{п}}$ – величина повного резерву по паралельних шляхах графіку (див. рисунок 6.14), у цьому випадку це шляхи через роботу 2 (коли обмежуючими будуть вантажні операції на під'їзній колії 3) та через роботу 8 (коли обмежуючими будуть вантажні операції на ПК 2);

$Л_i$ – ліміт коштів, виділених на скорочення тривалості операцій;

C_i – вартість скорочення тривалості операцій на 0,1 год на i-й ПК;

γ – порядковий номер етапу скорочення.

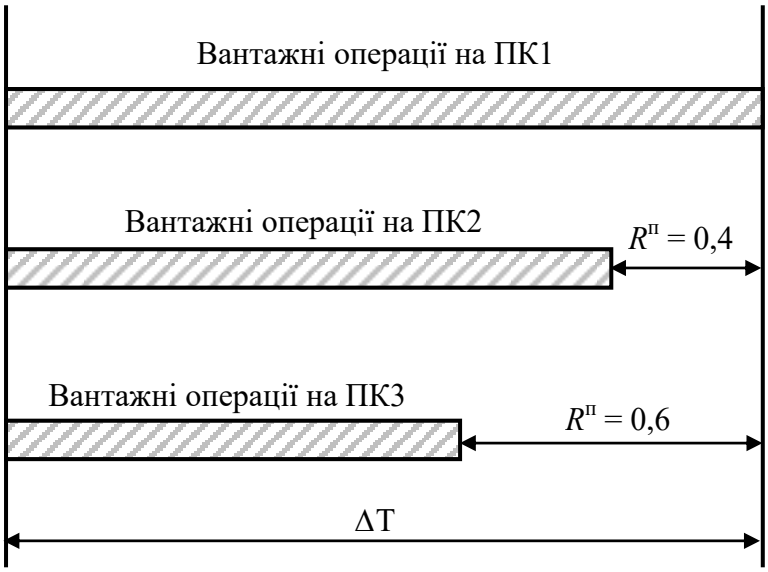


Рисунок 6.14 – Тривалість виконання вантажних операцій на під'їзних коліях до моменту виконання оптимізації

Слід зауважити, що кожну складову виразу (6.4) необхідно округлювати до десятих у менший бік.

Отже, на першому кроці загальну тривалість виконання операцій можна скоротити на

$$\Delta t_1 = \min\left[3{,}0 - 2{,}3; \min(0{,}6; 0{,}4); 0{,}1 \cdot (29/4)\right] = \left[0{,}7; 0{,}4; 0{,}7\right] = 0{,}4 \ \textit{год}.$$

На першому кроці скорочуємо тривалість виконання вантажних операцій на ПК 1 на 0,4 год.

Сума коштів, яка була витрачена на скорочення робіт, визначається за формулою (6.2).

Для першого кроку $k = 1$, $\Delta t_1 = 0{,}4\ год$, $C_{4-7} = 4\ у.о.$ і отримаємо

$$\sum C_1^{скор} = \Delta t_1 \cdot C_{4-7} = 0{,}4 \cdot 4 = 1{,}6\ у.о.$$

Ліміт коштів, який залишається після цього, розраховується за модифікованою формулою (6.3)

$$Л_k = Л_{k-1} - \frac{\sum C_k^{скор}}{0{,}1} \cdot, \qquad (6.5)$$

де $Л_k$ – наявний ліміт коштів на кроці k;

$\sum C_k^{скор}$ – загальна вартість скорочення робіт на 0,1 год на кроці k.

$$Л_1 = Л_0 - \frac{\sum C_1^{скор}}{0{,}1} = 29 - \frac{1{,}6}{0{,}1} = 13 \text{ у.о.}$$

Тепер критичний шлях розгалужується і проходить одночасно по двох шляхах, а саме через роботу 5 та роботу 8 (див. рисунок 6.15).

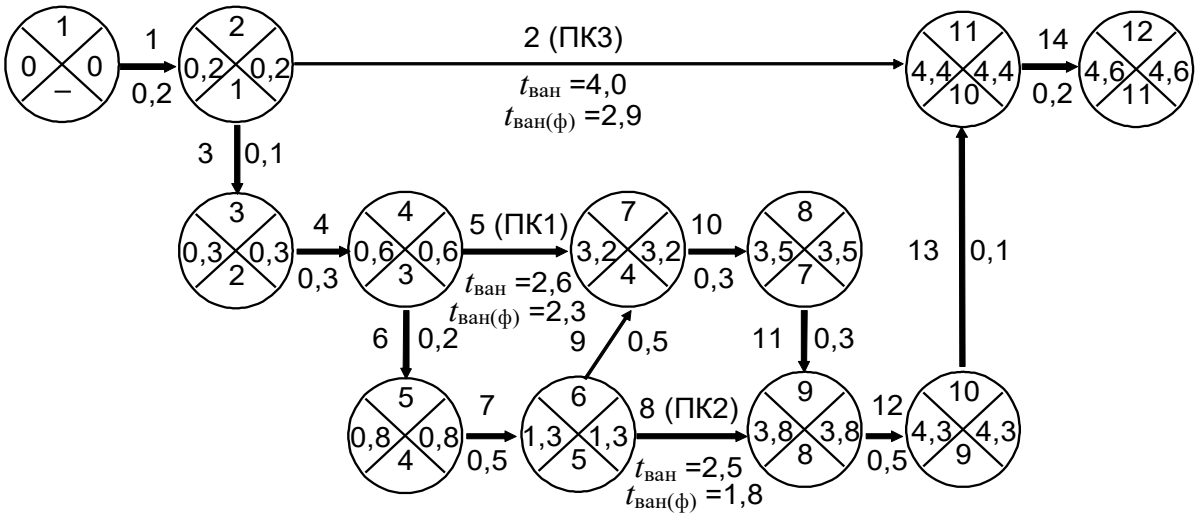


Рисунок 6.15 – Сітьовий графік обслуговування під'їзних колій після першого етапу скорочення тривалості вантажних операцій

У разі скорочення тривалості виконання вантажних операцій на ПК 1 більше, ніж резерв виконання вантажних операцій на ПК 2, необхідно одночасно скорочувати тривалість операцій на ПК 1 та на ПК 2 для того, що скоротити загальну тривалість виконання всіх операцій (див. рисунок 6.16).

Величина резерву після скорочення тривалості операцій на попередньому кроці становитиме $R_{\text{п}} = 0,6 - 0,4 = 0,2$ год.

Крім того, тривалість виконання вантажних операцій на ПК 1 теж скоротилася на 0,4 год і тепер дорівнює: $t'_{\text{ван}} = 3,0 - 0,4 = 2,6$ год.

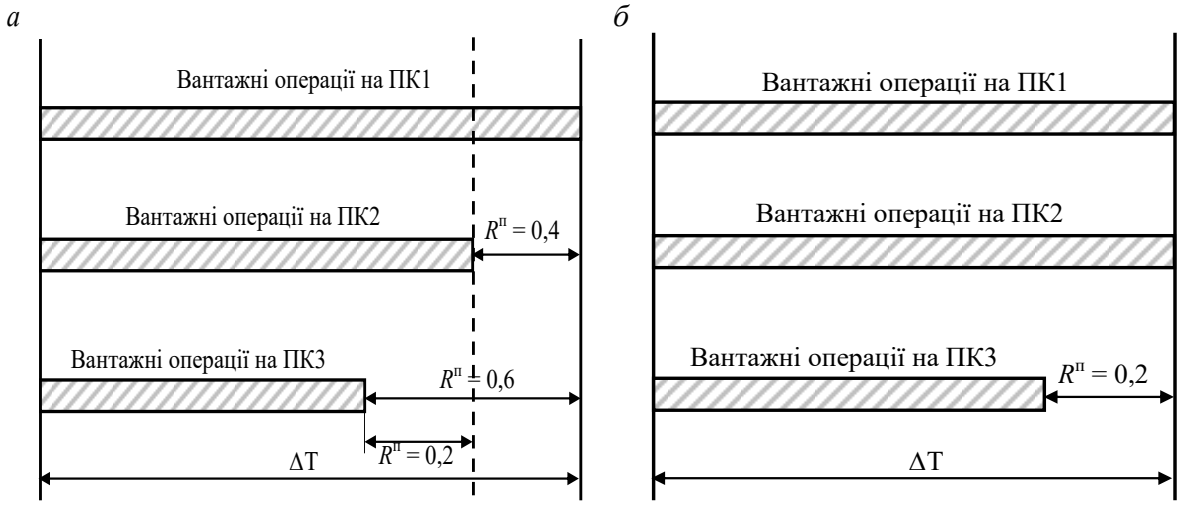


Рисунок 6.16 – Тривалість виконання вантажних операцій на під'їзних коліях
а) до першого скорочення тривалості виконання
б) після скорочення тривалості виконання операцій на ПК2

Крок 2. На другому кроці будемо скорочувати тривалість виконання операцій на ПК 1 та на ПК 2.

Вартість скорочення операцій на 0,1 год буде дорівнювати сумі вартостей скорочення операцій на цих двох під'їзних коліях і складатиме

$$\sum C_i^{\text{скор}} = 4 + 3 = 7 \text{ у.о.}$$

Крім того, резерв у роботи 8 (вантажні операції на ПК 2) вже відсутній, отже, цю роботу надалі теж можна зменшувати до відповідної величини $t_{\text{ван(ф)}}$, тобто до 1,8 год. Залишається тільки резерв для роботи 2 (вантажні операції на ПК 3), який зменшився після першого етапу скорочення і дорівнює 0,2 год.

Отже, використовуючи вираз (6.4), знаходимо значення Δt_2:

$$\Delta t_2 = \min\left[\min(2,6 - 2,3; 2,5 - 1,8); 0,2; 0,1 \cdot (13/7)\right] = \min\left[0,4; 0,2; 0,1\right] = 0,1 \text{ год.}$$

При розрахунку Δt_1 величина $t_{\text{ван}} - t_{\text{ван(ф)}}$ розрахована для ПК 1 та ПК 2, а величина резерву $R_{ij}^{\text{п}}$ – для ПК 3.

Сітьовий графік обслуговування під'їзних колій після другого етапу скорочення тривалості виконання вантажних операцій наведений на рисунку 6.17.

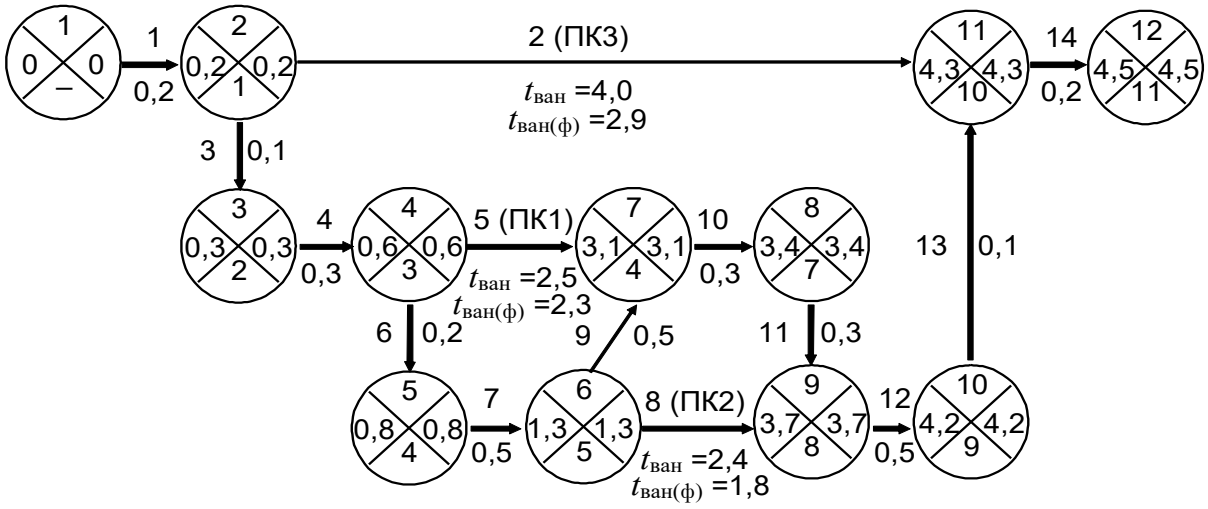


Рисунок 6.17 – Сітьовий графік обслуговування під'їзних колій після другого етапу скорочення тривалості вантажних операцій

Отже, після другого кроку додатково скорочуємо загальну тривалість виконання всіх операцій ще на 0,1 год.

Ліміт коштів, який залишається після даного етапу скорочення, становить $Л_2 = 13 - (0{,}1 \cdot 7) / 0{,}1 = 6$ у.о.

Подальше скорочення тривалості виконання вантажних операцій на під'їзних коліях неможливе, оскільки ліміту коштів, що залишився, недостатньо для скорочення операцій навіть на 0,1 год, тому що на третьому етапі необхідно скорочувати тривалість виконання вантажних операцій одночасно на трьох під'їзних коліях. Вартість скорочення на 0,1 год становитиме $\sum C_i^{скор} = 4 + 3 + 5 = 12$ у.о., що більше за ліміт коштів, що залишився ($Л_2 = 6$ у.о.).

Оптимізація скорочення тривалості операцій за сітьовим графіком припиняється у таких випадках:

– якщо ліміту коштів, що залишився, недостатньо для подальшого скорочення тривалості операцій;

– якщо на одному з можливих варіантів критичного шляху досягнуто величини $t_{ван(ф)}$.

Завдання 2-го типу: оптимізація сітьових графіків за умови ліміту часу.

Крок 1. На критичному шляху лежить робота 5 (вантажні операції на ПК 1). Вартість скорочення тривалості операцій на 0,1 год на цій ПК становить 4 у.о. (див. вихідні дані).

Величину зменшення тривалості виконання операцій на ПК 1 можна визначити з виразу

$$\Delta t_{\gamma} = \min\left[R_{ij}^{\text{п}} \right], \tag{6.6}$$

де $R_{ij}^{\text{п}}$ – величина повного резерву по паралельних шляхах графіка: у даному випадку це шляхи через роботу 2 (у випадку, якщо обмежуючими будуть вантажні операції на ПК 3) та через роботу 8 (у випадку, якщо обмежуючими будуть вантажні операції на ПК 2);

$$\Delta t_1 = \min\left[0,6; 0,4\right] = 0,4 \text{ год.}$$

На першому кроці скорочуємо тривалість виконання вантажних операцій на ПК 1 на 0,4 год. Тепер необхідно скоротити тривалість виконання всіх операцій на $0,9 - 0,4 = 0,5$ год.

Крок 2. Далі будемо скорочувати тривалість виконання операцій одночасно на ПК 1 і на ПК 2. Причому необхідно мати на увазі, що резерву у роботи 8 (вантажні операції на ПК 2) більше немає. Залишається тільки резерв для роботи 2 (вантажні операції на ПК 3), який зменшився після першого етапу скорочення і дорівнює 0,2 год, (див. рисунок 6.18).

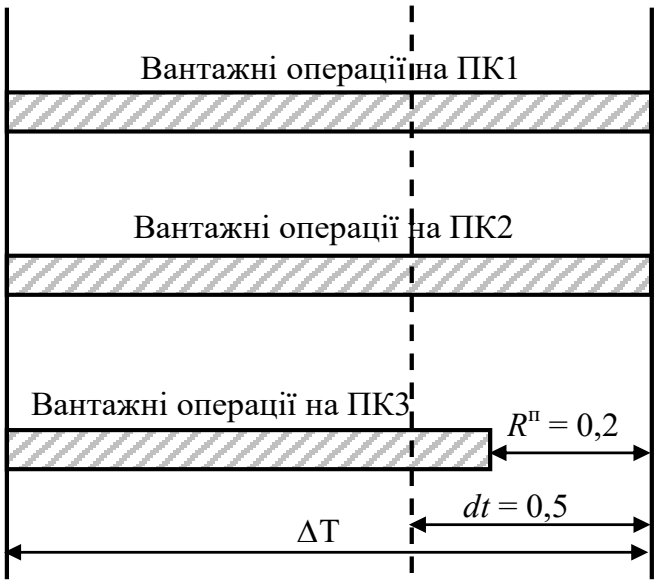


Рисунок 6.18 – Тривалість виконання вантажних операцій на під'їзних коліях перед другим етапом скорочення

Отже, подальше скорочення тривалості операцій повинно виконуватися в такому порядку: спочатку скорочуємо тривалість виконання

вантажних операцій на ПК 1 та ПК 2 на 0,2 год (величину резерву, що залишився), а потім скорочуємо тривалість виконання вантажних операцій одночасно на трьох під'їзних коліях на величину $0,5-0,2=0,3$ год.

Отже, $\Delta t_2 = \min[0,2;0,5] = 0,2$ год.

Крок 3. Скорочуємо тривалість виконання вантажних операцій на ПК 1, ПК 2 і ПК 3. $\Delta t_3 = 0,3$ год.

Подальше скорочення тривалості виконання операцій на під'їзних коліях не потрібне, тому що за три кроки вдалося скоротити загальну тривалість на

$$\sum \Delta t = 0,4+0,2+0,3=0,9 \text{ год.}$$

Тривалість виконання вантажних операцій на під'їзних коліях становитиме:

ПК 1: $t_{\text{ван}} = 3,0-(0,4+0,2+0,3)=2,1$ год (тривалість виконання операцій скорочували на кожному кроці);

ПК 2: $t_{\text{ван}} = 2,5-(0,2+0,3)=2,0$ год (тривалість виконання операцій скорочували на другому та третьому кроках);

ПК 3: $t_{\text{ван}} = 4,0-0,3=3,7$ год (тривалість виконання операцій скорочували тільки на третьому кроці).

Вартість скорочення тривалості виконання операцій визначимо наступним чином:

$$C = \frac{0,4+0,2+0,3}{0,1}\cdot 4 + \frac{0,2+0,3}{0,1}\cdot 3 + \frac{0,3}{0,1}\cdot 5 = 66 \text{ у.о.}$$

Таким чином, на задане скорочення тривалості виконання операцій було витрачено 66 у. о.

Контрольні запитання

1. Навести приклад оптимізаційних задач, які можна вирішувати, використовуючи сітьове планування та управління?
2. За рахунок чого можна скоротити тривалість комплексу робіт?
3. Які бувають режими виконання робіт?
4. Чи для кожної роботи характерна наявність «формованого» режиму її виконання?
5. Які типи завдань вибору оптимальних режимів робіт існують в сітьовому плануванні та управлінні?
6. З якою метою при вирішенні оптимізаційних задач може бути доцільним замінити певний підграф сітьового графіка умовною комплексною роботою?

7. Яким двом умовам повинна відповідати робота, яку доцільно в першу чергу переводити у «форсований» режим?

8. Які основні фактори впливають на величину скорочення тривалості роботи на кожному кроці при вирішенні задачі оптимізації сітьових графіків за умови ліміту коштів?

9. Які основні фактори впливають на величину скорочення тривалості роботи на кожному кроці при вирішенні задачі оптимізації сітьових графіків за умови ліміту часу?

10. В якому випадку (чи випадках) подальше вирішенні задачі оптимізації сітьових графіків за умови ліміту коштів неможливе?

11. В якому випадку (чи випадках) подальше вирішенні задачі оптимізації сітьових графіків за умови ліміту часу неможливе?

РОЗДІЛ 7

Використання методики вирішення розподільчої задачі для оптимізації технологічних процесів

7.1 Теоретичні відомості

До числа поширених і важливих економіко-математичних задач відноситься вибір найкращих варіантів використання різноманітних видів взаємозамінних ресурсів для виконання різних видів робіт [2, 3, 12].

У задачах розподілу взаємозамінних ресурсів зазвичай вихідними даними є:

– обсяги B_j різноманітних робіт j, які повинні бути виконані;

– наявна кількість ресурсів A_i різних постачальників i;

– витрати ресурсів кожного типу i на одиницю робіт кожного виду j – a_{ij}.

Необхідно при заданих значеннях A_i, B_j, a_{ij}, визначити обсяг кожного виду робіт j, який виконується за допомогою ресурсу i – x_{ij}, і кількість одиниць ресурсу i, який використовується на роботах виду j – $y_{ij} = a_{ij}\, x_{ij}$.

При розподілі ресурсів можуть ставитися задачі двох видів:

– загальної наявності ресурсів недостатньо для повного задоволення потреб (наприклад, наявної кількості транспортних засобів (вагонів, вантажних автомобілів) недостатньо для вивозу всього вантажу). У цьому випадку максимальне збільшення загального обсягу роботи є важливішим за зниження витрат. Тому в таких задачах критерієм оптимальності є максимальний обсяг роботи, що виконується.

– загальної наявності ресурсів достатньо для повного задоволення всіх потреб. Критерієм оптимальності в цьому випадку стає зведення до мінімуму загальної суми витрат на виконання заданого обсягу роботи.

Для залізничної станції типовою задачею такого роду є розподіл вагонів різних видів під завантаження різного роду вантажів. Якщо порожніми пробігами вагонів до місць навантаження знехтувати, то критерієм оптимального розподілу вагонів є досягнення максимуму середнього статичного навантаження.

Аналогічним чином вирішується розподільча задача і для підприємства, яке має парк вантажних автомобілів та займається вивезенням вантажів з підприємства до споживачів.

7.2 Математична постановка розподільчої задачі

Нехай відома кількість різних вантажів B_j у тонах, які необхідно буде транспортувати, а також наявність порожніх ТЗ різних типів A_i та їх технічні норми навантаження p_{ij}. При неможливості відправлення певного роду вантажу в даному типі ТЗ приймаємо значення $p_{ij} = 0$.

Необхідно знайти такий розподіл ТЗ кожного типу під завантаження вантажів кожного роду, що забезпечить мінімальну кількість завантажених ТЗ.

Система обмежень при цьому матиме такий вигляд:

$$\begin{aligned} &\sum_i x_{ij} p_{ij} = B_j \ (j = 1,\ 2,\ ...) \\ &\sum_j x_{ij} \le A_i \ (i = 1,\ 2,\ ...) \\ &\sum_i \sum_j x_{ij} \to \min \end{aligned} \quad , \qquad (7.1)$$

де $\sum_i x_{ij} p_{ij} = B_j \ (j = 1,\ 2,\ ...)$ – весь вантаж повинен бути вивезений;

$\sum_j x_{ij} \le A_i \ (i = 1,\ 2,\ ...)$ – кількість ТЗ кожного роду, що зайняті під завантаження, не повинна перевищувати їх наявної кількості;

$\sum_i \sum_j x_{ij} \to \min$ – кількість завантажених ТЗ має бути мінімальною.

Задача розподілу ТЗ під завантаження може бути розв’язана **методом розв’язувальних множників**.

Зведемо вихідні дані в таблицю 7.1, де рядки відповідають типам ТЗ, а стовпці – родам вантажів. Ще є стовпці, які відведені для суми ТЗ кожного виду, що були використані, величини надлишку та нестачі,

а також для міток (проставляється «*» для недостатнього рядка). У правих верхніх кутах основних клітинок вказані фактичні технічні норми навантаження p_{ij}.

Таблиця 7.1

Формалізація вихідних даних у табличному вигляді

ТЗ	Вантажі			Зайнято ТЗ	Баланс	Мітка
	1-й рід **350**	2-й рід **250**	3-й рід **150**			
1-й тип **25**	**30**	**25** 10/250	**10**	10	+15	
2-й тип **20**	**35** 10/350	**20**	**15** 15/150	25	-5	*

Таким чином, задача розподілу ТЗ під навантаження вирішується з метою визначення мінімальної кількості рухомого складу, зайнятого під навантаження.

7.3 Алгоритм розв'язання розподільчої задачі

Крок 1. Скласти початковий варіант розподілу ТЗ таким чином, щоб кожен вантаж був завантажений лише у ті з них, для яких норма навантаження є максимальною (за наявності однакових норм навантаження – у менш дефіцитні ТЗ).

Кількість ТЗ кожного виду, що використовуються, визначаються за формулою

$$x_{ij} = \lceil B_j / p_{ij} \rceil, \tag{7.2}$$

де B_j – обсяг вантажів j-го роду, *тони*;

p_{ij} – технічні норми навантаження ТЗ i-го типу вантажами j-го роду.

У відповідній клітинці таблиці записуємо через дріб кількість зайнятих ТЗ (чисельник), а також обсяг вантажу, що слід в цих ТЗ перевозити (знаменник).

Крок 2. Визначити кількість зайнятих ТЗ кожного типу, а також баланс їх наявності R_i : надлишок ($R_i > 0$) та нестача ($R_i < 0$).

Баланс наявності ТЗ визначимо за формулою

$$R_i = A_i - \sum_j x_{ij} \ (i = 1,\ 2,\ ...)\,, \qquad (7.3)$$

Якщо баланси наявності для всіх типів ТЗ невід'ємні ($R_i \geq 0$), то вирішення завершено і даний розв'язок є оптимальним.

Крок 3. Класифікувати рядки на надлишкові та недостатні.

Недостатніми є всі рядки, у яких $R_i < 0$, а також рядки, у яких $R_i = 0$, і, при цьому, які відповідають обидвом вимогам:

– поточне значення навантаження в одному зі стовпців дорівнює поточному значенню навантаження в допустимій клітинці, що знаходиться на перетині цього стовпця та одного з недостатніх рядків;

– ця допустима клітинка повинна бути зайнятою (з перевезенням).

Якщо хоча б одна з вказаних вище вимог не виконується, то рядок, у якому $R_i = 0$, не буде недостатнім.

Так, в таблицях 7.2 – 7.3 рядки з $R_i = 0$ не є недостатніми.

Таблиця 7.2

Приклад №1 таблиці, у яких рядок з $R_i = 0$ не є недостатнім

ТЗ	Вантажі			Зайнято ТЗ	Баланс	Мітка
	1-й рід **350**	2-й рід **250**	3-й рід **150**			
1-й тип **10**	**30**	**25** 10/250	**10**	10	0	
2-й тип **20**	**35** 10/350	**20**	**15** 15/150	25	-5	*

У таблиці 7.2 поточне значення навантаження в жодному зі стовпців не дорівнює поточному значенню навантаження в клітинці, що розташована на перетині цього стовпця та недостатнього рядка, тому рядок з $R_i = 0$ не є недостатнім.

В таблиці 7.3 клітинка, що знаходиться на пересіченні стовпця, де збіглися значення поточного навантаження цієї клітинки та недостатнього рядка, не є зайнятою, тому рядок з $R_i = 0$ в цій таблиці теж не є недостатнім.

Таблиця 7.3

Приклад №2 таблиці, у яких рядок з R_i = 0 не є недостатнім

ТЗ	Вантажі			Зайнято ТЗ	Баланс	Мітка
	1-й рід **350**	2-й рід **250**	3-й рід **150**			
1-й тип **13**	**30**	**20** 13/250	**10**	13	0	
2-й тип **20**	**35** 10/350	**20**	**15** 15/150	25	-5	*

В той же час, в таблицях 7.4 та 7.5 рядки з R_i = 0 є недостатніми.

Таблиця 7.4

Приклад №1 таблиці, у яких рядок з R_i = 0 є недостатнім

ТЗ	Вантажі			Зайнято ТЗ	Баланс	Мітка
	1-й рід **350**	2-й рід **250**	3-й рід **150**			
1-й тип **10**	**35**	**25** 10/250	**10**	10	0	*
2-й тип **20**	**35** 10/350	**20**	**15** 15/150	25	-5	*

Таблиця 7.5

Приклад №2 таблиці, у яких рядок з R_i = 0 є недостатнім

ТЗ	Вантажі			Зайнято ТЗ	Баланс	Мітка
	1-й рід **350**	2-й рід **250**	3-й рід **150**			
1-й тип **12**	**35** 2/70	**25** 10/250	**10**	12	0	*
2-й тип **20**	**35** 8/280	**20**	**15** 15/150	23	-3	*

Так, в таблицях 7.4 та 7.5 обидві вимоги виконуються, тому рядки з R_i = 0 будуть недостатніми.

Усі недостатні рядки слід помітити зірочкою (*) в стовпчику «Мітка».

Якщо всі рядки недостатні (як $R_i < 0$, так і рядки з $R_i = 0$), то вирішення закінчено. В цьому випадку вивезти весь вантаж наявними ТЗ неможливо, а отримане рішення дозволяє вивезти максимальний обсяг заданого вантажу.

Крок 4. Для кожного стовпця, для якого $x_{ij} > 0$ у недостатньому рядку, визначити відношення

$$\lambda_j = \frac{\max\limits_{i} \tilde{p}_{ij}}{\max\limits_{i \in H} \tilde{p}_{ij}}, \tag{7.4}$$

де $\max\limits_{i} \tilde{p}_{ij}$ – максимальне значення навантаження по стовпчику;

$\max\limits_{i \in H} \tilde{p}_{ij}$ – максимальне значення навантаження по стовпчику, але лише серед клітинок надлишкових рядків.

Отримані значення λ_j записуємо в нижній рядок.

Крок 5. Визначити мінімальне значення серед λ_j ($\lambda = \min \lambda_j$).

Крок 6. Перетворити матрицю навантаження, розділивши всі значення p_{ij} в недостатніх рядках на λ. Значення p_{ij} із надлишкових рядків переноситься без змін.

Крок 7. В одному з надлишкових рядків матриці знайти клітинку *st*, для якої величина поточного навантаження дорівнювала максимальній у стовпці. Ця клітинка розглядається як допустима та до неї переноситься додатній об'єм вантажу x_{st}.

Важливо:

– слід враховувати, що поява нових від'ємних балансів і зменшення використання ресурсів не допускається;

– кількість ТЗ, що використовуються, визначається за фактичними значеннями їх завантаження p_{ij}.

Крок 8. Перейти до кроку 2.

Важливо зазначити, що закінчення вирішення розподільчої задачі відбувається у двох випадках:

1) в усіх рядках баланс $R_i \geq 0$, тобто немає рядків з від'ємним балансом. Це означає, що кількість ТЗ є достатньою для перевезення заданих обсягів вантажів, а отриманий розподіл, який забезпечує мінімальну кількість завантажених ТЗ;

2) в усіх рядках баланс $R_i \leq 0$, тобто немає рядків з додатнім балансом. Це означає, що наявної кількості ТЗ недостатньо для перевезення заданих обсягів вантажів, а отриманий розподіл дозволяє вивезти максимальний обсяг вантажів наявними ТЗ.

7.4. Приклад розв'язання задач. Задача 8

Умова задачі

Для деякого умовного транспортного підприємства характерно:

– існує певна кількість родів вантажів (з обсягами), які слід транспортувати з даного підприємства;

– є обмежена кількість наявних транспортними засобами (ТЗ) різних типів, які можуть використовуватись для перевезення вказаних вантажів (всіх родів вантажів чи лише деяких із них);

– для кожного роду вантажу характерна фактична норма завантаження кожного типу транспортних засобів (якщо даний тип транспортних засобів не може перевозити певний рід вантажу умовно приймаємо фактичну норму навантаження рівною нулю).

Необхідно розподілити задані обсяги вантажів між наявними транспортними засобами (з врахуванням їх сумісності) таким чином, щоб загальна кількість транспортних засобів, що будуть зайняті перевезенням вантажів, була мінімальною.

Вихідні дані

Вихідні дані для даної задачі наведені в таблиці 7.6.

Таблиця 7.6

Вихідні дані

ТЗ	Вантажі				Наявна кількість ТЗ
	1-й рід	2-й рід	3-й рід	4-й рід	
1-й тип	27	13	15	7	30
2-й тип	30	16	17	8	35
3-й тип	30	21	22	10	25
Обсяг вантажів	600	128	500	100	

В таблиці 7.6 наведені наступні дані:

– наявна кількість ТЗ кожного типу (останній стовпчик);

– заданий обсяг (в тонах) вантажів кожного роду, які потрібно транспортувати з даного підприємства (нижній рядок);

– фактична норма завантаження (в тонах) *i*-го типу ТЗ вантажем *j*-го роду (центральна частина таблиці).

Розв'язання задачі

Вирішимо дану розподільчу задачу методом розв'язувальних множників.

Спочатку формалізуємо вихідні дані у таблиці 7.7.

Таблиця 7.7

Початковий розподіл вантажів у ТЗ

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**			
2-й тип **35**	**30** 20/600	**16**	**17**	**8**			
3-й тип **25**	**30**	**21** 7/128	**22** 23/500	**10** 10/100			

Далі в таблиці 7.7 складемо початковий план перевезення.

Для цього в кожному стовпчику даної таблиці обираємо ТЗ з максимальною технічною нормою навантаження і визначаємо кількість потрібних ТЗ поділивши обсяг вантажу, який слід перевезти, на норму навантаження за виразом (7.2).

Якщо ж технічна норма навантаження дорівнює нулю, то даний тип ТЗ для такого роду вантажу використовувати заборонено.

У випадку, якщо в результаті розрахунків буде отримано дробовий результат, то згідно виразу (7.2) його необхідно округлити до цілого в більшу сторону.

Таким чином, для 1-го роду вантажу максимальне навантаження складає 30 т (завантажувати цей вантаж слід у менш дефіцитні ТЗ):

$$x_{ij} = \lceil 600 / 30 \rceil = 20 \textit{ одиниць},$$

Так, для перевезення вантажу 1-го роду обрано ТЗ 2-го типу, так як їх кількість складає 35 одиниць, а ТЗ 3-го типу є лише 25 одиниць.

Для кожного типу ТЗ слід розрахувати баланс.

Наприклад, 3-го типу ТЗ зайнято 7 + 23 = 40 *одиниць*, а їх баланс складе 25 – 40 = -15 *одиниць*.

Результати розрахунків наведені в таблицю 7.8.

Таблиця 7.8

Розрахунок балансу зайнятих ТЗ

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17**	**8**	20	**+15**	
3-й тип **25**	**30**	**21** 7/128	**22** 23/500	**10** 10/100	40	**–15**	

Рядки з від'ємним балансом позначаємо міткою (*). Ці рядки є недостатніми (див. таблицю 7.9).

Оскільки в даній таблиці рядків з нульовим балансом немає, то ж й інших недостатніх рядків не буде.

Таблиця 7.9

Визначення недостатніх рядків

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17**	**8**	20	**+15**	
3-й тип **25**	**30**	**21** 7/128	**22** 23/500	**10** 10/100	40	**–15**	*

Для кожного стовпчика, який містить хоч одну зайняту клітинку в недостатніх рядках, за формулою (7.4) розраховуємо дозволяючий множник λ.

Так, наприклад, для 2-го роду вантажу

$$\lambda = 21 / 16 = 1{,}313$$

Для першого стовпчика λ не розраховується, оскільки в цьому стовпчику зайнята клітинка знаходиться у рядку, який не помічений *.

Результати розрахунків наведені в таблиці 7.10.

Таблиця 7.10

Розрахунок коефіцієнтів λ

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17**	**8**	20	**+15**	
3-й тип **25**	**30**	**21** 7/128	**22** 23/500	**10** 10/100	40	**–15**	*
λ_j	–	1,313	1,294	1,250			

Слід відзначити, що значення коефіцієнту λ не може бути меншим за одиницю (λ≥1). Якщо ж отримане значення коефіцієнту λ<1, то це означає, що мала місце помилка розрахунків.

Серед усіх розрахованих λ необхідно обрати мінімальний (λ_{min}=1,250) та виділити його (див. таблицю 7.11).

Таблиця 7.11

Визначення мінімального серед розрахованих коефіцієнтів λ

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17**	**8**	20	**+15**	
3-й тип **25**	**30**	**21** 7/128	**22** 23/500	**10** 10/100	40	**–15**	*
λ_j	–	1,313	1,294	**1,250**			

Клітинка з поточним значенням навантаження p_{ij} =8 з надлишкового рядка буде допустимою для переносу деякого обсягу вантажу. Показуємо стрілкою, з якої клітинки у яку буде перенесено вантаж на наступному кроці (див. таблицю 7.12); поточне навантаження в обох цих клітинках на наступному кроці повинні безумовно співпадати (див. таблицю 7.13).

Таблиця 7.12

Визначення стовпця, по якому буде перерозподіл вантажу

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17**	**8** ↑	20	**+15**	
3-й тип **25**	**30**	**21** 7/128	**22** 23/500	**10** 10/100	40	**–15**	*
λ_j	–	1,313	1,294	**1,250**			

Перетворюємо матрицю навантаження, розділивши всі значення p_{ij} в усіх недостатніх рядках на λ (див. таблицю 7.13). При цьому нецілі значення записуються з точністю до тисячних (або у вигляді звичайного дробу).

Значення p_{ij} із надлишкових рядків переноситься без змін.

Таблиця 7.13

Перетворення матриці після першої ітерації

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**			
2-й тип **35**	**30** 20/600	**16**	**17**	**8** ↑			
3-й тип **25**	**24**	**16,800** 7/128	**17,600** 23/500	**8** 10/100			

У допустиму клітинку переноситься додатковий обсяг вантажу (див. таблицю 7.14).

Таблиця 7.14

Перерозподіл вантажу після першої ітерації

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17**	**8** 13/100	33	**+2**	
3-й тип **25**	**24**	**16,800** 7/128	**17,600** 23/500	**8**	30	**–5**	*

При цьому необхідно враховувати, що поява нових від'ємних балансів і зменшення використання ресурсів не допускається. Це означає, що баланс рядків не може переходити з від'ємного у додатній і навпаки, а лише в нульовий баланс (якщо умови задачі дозволяють). Отже, якщо при вирішенні задачі з'являється рядок з балансом $R_i = 0$, то цей рядок повинен залишитися нульовим до кінця вирішення задачі.

Для кожної зайнятої клітинки з недостатніх рядків розраховуємо коефіцієнти (див. таблицю 7.15).

Таблиця 7.15

Визначення стовпця, по якому буде перерозподіл вантажу

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17**	**8** 13/100	33	**+2**	
3-й тип **25**	**24**	**16,800** 7/128	**17,600** 23/500	**8**	30	**–5**	*
λ_j	–	1,050	**<u>1,035</u>**	–			

На другому кроці коефіцієнт λ розраховується для стовпців 2 та 3. Виділяємо серед розрахованих λ мінімальний ($\lambda_{min} = 1{,}035$) і показуємо, з якої клітинку в яку на наступному кроці буде виконуватися перерозподіл вантажу (див. таблицю 7.15).

Перетворюємо матрицю навантаження, розділивши всі значення p_{ij} в усіх недостатніх рядках на λ (див. таблицю 7.16). Значення p_{ij} із рядків таблиці 7.15, які не помічені зірочкою, переноситься без змін.

Таблиця 7.16

Перерозподіл вантажу після другої ітерації

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17** 2/34	**8** 13/100	35	**0**	
3-й тип **25**	**23,188**	**16,232** 7/128	**17** 22/466	**7,727**	29	**–4**	

Кількість ТЗ, що використовуються, визначається за фактичними значеннями навантаження p_{ij}.

У ТЗ 2-го типу можна зайняти лише 2 одиниці (у іншому випадку баланс 2-го рядка перейде з 0 у від'ємний). Кількість вантажу, яка може бути завантажена в ці ТЗ, складає 2 · 17 = 34 *т*.

Решту вантажу (500 – 34 = 466 *т*) необхідно залишити у ТЗ 3-го типу. Необхідна кількість ТЗ складає 466 / 22 = 21,182 одиниць, а після округлення в більшу сторону отримаємо 22 одиниці.

Для кожного типу ТЗ необхідно розрахувати баланс, після чого усі рядки з від'ємним балансом позначити зірочкою (*). Рядок з балансом R_i=0 також є недостатнім, тому що поточне значення навантаження p_{23}=17 дорівнює поточному значенню навантаження в клітинці, що знаходиться на перетині цього стовпця та недостатнього рядка із навантаженням p_{33}=17, тому рядок 2 також помічаємо зірочкою (*).

Результати всіх вказаних дій наведені в таблиці 7.17.

Далі у відповідності до вищенаведеного алгоритму розраховуємо λ, обираємо серед них мінімальний та визначаємо, з яких клітинок у які буде виконуватися перерозподіл вантажу (див. таблицю 7.18).

Але у даному випадку перенесення вантажу з 2-го рядка у 1-й спричинить зміну балансу 2-го рядка з нульового на додатній, що також суперечить алгоритму вирішення задачі. Тому, з метою збереження нульового балансу 2-го рядка, необхідно паралельно перенести вантаж 3-го роду з ТЗ 3-го типу у ТЗ 2-го типу (див. таблиці 7.17–7.18).

Таблиця 7.17

Розрахунок балансу зайнятих ТЗ

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17** 2/34	**8** 13/100	35	**0**	*
3-й тип **25**	**23,188**	**16,232** 7/128	**17** 22/466	**7,727**	29	**–4**	*

Таблиця 7.18

Визначення стовпця, по якому буде перерозподіл вантажу

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17** 2/34	**8** 13/100	35	**0**	*
3-й тип **25**	**23,188**	**16,232** 7/128	**17** 22/466	**7,727**	29	**–4**	*
λ_j	1,111	1,249	1,333	1,143			

Перерозподіл вантажу необхідно виконувати таким чином, щоб зберегти нульовий баланс 2-го рядка, при цьому необхідно враховувати, що поява нових від’ємних балансів і зменшення використання ресурсів не допускається (див. таблицю 7.19).

Слід звернути увагу, що перерозподіляти вантаж можна у межах як одного стовпчика так і декількох стовпчиків, але тільки із клітинки в клітинку, у яких на даному кроці співпадає поточне завантаження.

Таблиця 7.19

Визначення стовпця, по якому буде перерозподіл вантажу

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27**	**13**	**15**	**7**	0	**+30**	
2-й тип **35**	**30** 20/600	**16**	**17** 2/34	**8** 13/100	35	**0**	*
3-й тип **25**	**23,188**	**16,232** 7/128	**17** 22/466	**7,727**	29	**–4**	*
λ_j	**<u>1,111</u>**	1,249	1,333	1,143			

Так, в таблиці 7.20 при перетворенні матриці необхідно перенести вантаж 3-го роду з ТЗ 3-го типу у ТЗ 2-го типу, але для цього їх необхідно звільнити від вантажу 1-го роду. При цьому ТЗ 3-го типу необхідно звільнити лише 4 одиниці.

Таблиця 7.20

Перерозподіл вантажу після третьої ітерації

ТЗ	Вантажі				Зайнято ТЗ	Баланс	Мітка
	1-й рід **600**	2-й рід **128**	3-й рід **500**	4-й рід **100**			
1-й тип **30**	**27** 6/150	**13**	**15**	**7**	6	**+24**	
2-й тип **35**	**27** 15/450	**14,401**	**15,302** 7/104	**7,201** 13/100	35	**0**	
3-й тип **25**	**20,872**	**14,610** 7/128	**15,302** 18/396	**6,955**	25	**0**	
Всього					**66**		

Отже, для вантажу 3-го роду звільняємо 4 одиниці 3-го типу ТЗ: при цьому залишаємо зайнятими лише 18 одиниць 3-го типу ТЗ, у яких знаходитиметься 18 · 22 = 396 т.

Решту вантажу (500 – 396 = 104 т) переміщуємо у ТЗ 2-го типу, яких необхідно 104 / 17 ≈ 7 одиниць.

Якщо для вантажу 3-го роду зайняти 7 одиниць 2-го типу ТЗ, то для вантажу 1-го роду можна буде використати 35 – (13 + 7) = 15 одиниць ТЗ. Всі ці 15 одиниць повністю завантажуємо (див. таблицю 7.20). В них сумарно буде 15 · 30 = 450 т вантажу.

Решту (600 – 450 = 150 т) вантажу переміщуємо у ТЗ 1-го типу, яких треба 150 /24 ≈ 6 одиниць.

Даний план завантаження ТЗ є оптимальним, тому що всі $R_i \geq 0$. Загальна кількість зайнятих перевезеннями ТЗ склала 66 одиниць.

Для умов даного прикладу кінець розв'язку наступив тому, що в усіх рядках баланс $R_i \geq 0$, тобто немає рядків з від'ємним балансом. Це означає, що кількість ТЗ є достатньою для перевезення заданих обсягів вантажів і досягнуто оптимальний розподіл вантажу в ці ТЗ, а кількість зайнятих ТЗ є мінімально можливою.

Контрольні запитання

1. Які транспортно-логістичні задачі можуть бути вирішені за методикою розподільчої задачі?
2. Яким методом вирішується розподільча задача?
3. Які вихідні дані потрібні для вирішення розподільчої задачі?
4. Як саме виконується побудова початкового плану перевезень при вирішенні розподільчої задачі?
5. Чи враховується наявна кількість ТЗ під час побудови початкового плану перевезень при вирішенні розподільчої задачі?
6. За яким показником визначаються «недостатні» чи «надлишкові» рядки?
7. За яких умов рядки з додатнім ($R_i > 0$) та з від'ємним балансом ($R_i < 0$) є недостатніми?
8. За яких умов рядки з нульовим балансом ($R_i = 0$) є недостатніми?
9. Для чого використовуються розв'язувальні множники при вирішенні розподільчої задачі?
10. Як слід виконувати перерозподіл вантажу певного роду між ТЗ різного типу при вирішенні розподільчої задачі?

РОЗДІЛ 8

Оптимізація техніко-технологічних параметрів транспортних систем із застосуванням методів динамічного програмування

8.1 Теоретичні відомості

У багатьох випадках управлінські рішення на транспортних об'єктах приймають виходячи із ситуації, яка склалася на поточний момент: наявної кількості готових до роботи локомотивів, вільних бригад ПТО, вільних колій у парку, вільних місць на СТО чи наявних перевантажувальних механізмів і т. ін. [2, 3, 12].

Управлінське рішення за такої ситуації має сенс або за умов стабільності системи, або на короткий проміжок у часі. У реальності всі процеси на транспорті змінюються в часі, тобто за своєю природою є динамічними. Це вимагає від управлінців розв'язання практичних задач, в яких необхідно враховувати можливі зміни процесів у часі за умов, що процесом можна керувати, тобто впливати на хід його розвитку.

Динамічне програмування – це математичний метод оптимізації рішень, який пристосований до багатоетапних або багатокрокових операцій.

Особливістю у такому програмуванні є те, що для управління складним процесом серед множини всіх допустимих рішень шукають оптимальне за якимось критерієм, тобто таке рішення, яке дає екстремум (максимум або мінімум) цільової функції з урахуванням зміни ситуації у часі. Під багатоетапністю розуміють або багатокрокову структуру процесу, або розподіл управління на ряд послідовних етапів, що відповідають, як правило, різним моментам часу, тобто розв'язок виконується у декілька етапів. Таким чином, слово "програмування" у даному контексті означає прийняття управлінських рішень, а слово "динамічне" вказує на суттєве значення фактору часу та порядку виконання

операцій у процесах і методах, що розглядаються. Отже, процес динамічного програмування виконується у декілька етапів від кінця операції до її початку.

Отже, при оптимізації процесів із застосуванням динамічного програмування операція розбивається на декілька кроків, на кожному з яких приймається рішення, від якого залежить успіх даного кроку і всієї операції в цілому. Таке рішення називають управлінням на ***i***-му кроці.

Управління операцією в цілому являє собою сукупність всіх покрокових управлінь.

$$U = (U_1, U_2, ..., U_m) \tag{8.1}$$

де U_i – управління на ***i***-му кроці.

Ефект операції залежить від управління операцією:

$$W = f(U) = f(U_1, U_2, ..., U_m) \tag{8.2}$$

Виникає питання – як вибрати покрокові управління U_1, U_2, ..., U_m таким чином, щоб ефект був максимальним.

При цьому ефект від операції складатиме:

$$W = \sum_{i=1}^{m} \omega_i \tag{8.3}$$

де ω_i – ефект на ***i***-му кроці.

Суть динамічного програмування в тому, що операція здійснюється поетапно або покроково. Складність полягає в тому, що результат управління на кожному кроці впливає на всі подальші кроки. Крім того, при оптимізації останнього *n*-го кроку невідомо, як закінчився *n-1* крок (невідомий стан системи перед останнім кроком).

Для того, щоб вийти з цього ускладнення, здійснюється припущення по результатам *n-1* кроку про стан системи. Таке припущення називають **умовно-оптимальним**.

Рухаючись від кінця до початку можна знайти **умовно-оптимальне рішення**, яке забезпечує оптимальне продовження процесу відносно стану, який досягнуто в поточний момент.

Після цього, якщо відомий початковий стан системи, можна **знайти безумовно оптимальне рішення** на першому кроці.

Після першого кроку система перейде в другий стан, для якого нам вже також відомо оптимальне управління.

Таким чином, в процесі оптимізації управління методом динамічного програмування багатокроковий процес проходиться двічі.

Перший раз – від початку до кінця, внаслідок чого визначаються умовно-оптимальні управління на кожному кроці, а також умовно-оптимальний ефект на етапі від даного кроку і до кінця операції.

Другий раз проходимо процес від початку до кінця, внаслідок чого визначаємо безумовні оптимальні крокові управління на всіх кроках операції.

8.2. Приклад розв'язання задач. Задача 9

Умова задачі

Розглянемо застосування методів динамічного програмування на прикладі задачі вибору черговості розформування составів.

Розформування составів на сортувальних станціях представляє собою багатокроковий процес, при чому кількість можливих варіантів на кожному кроці є змінною величиною. При цьому, мінімальна кількість варіантів складає один, максимальна дорівнює кількості колій в парку прийому.

Вибір черговості розформування составів впливає на такі параметри станції:

– простій вагонів під накопиченням;

– звільнення колій в парку для прийому нових поїздів;

– простій локомотивів у парку прийому у зв'язку з перетинанням маршрутів;

– черговість витягування составів у парк відправлення.

Якщо вирішувати дану задачу як задачу динамічного програмування, то із зазначених критеріїв необхідно вибрати лише один: простій вагонів під накопиченням.

Необхідно обрати таку черговість розформування составів на сортувальній станції, що забезпечує досягнення мінімального простою вагонів під накопиченням.

Вихідні дані

Норма составів по накопиченню m=50 вагонів. Порядок надходження составів у розформування наведений в таблиці 8.1:

Таблиця 8.1

Порядок надходження составів у розформування

Початковий стан	Кроки	
	1	2
1, 2	3	

Початковий стан сортувального парку та характеристика составів наведені в таблиці 8.2:

Таблиця 8.2

Початковий стан сортувального парку та характеристика составів

Розподіл вагонів		Призначення											Всього	$T_р$, год
		1	2	3	4	5	6	7	8	9	10	11		
Залишок на сортувальних коліях		22	32	7	1	10	1	14	6	0	46	0	139	
Состави	1	8	31	1	2	2	0	5	0	0	0	3	52	0,29
	2	5	8	6	2	0	8	0	14	9	1	2	55	0,31
	3	1	0	0	2	0	0	20	4	9	16	2	54	0,30

Розв’язання задачі

Прийняті поняття та позначення

Під **операцією** будемо розуміти розформування групи составів, які готові до розпуску на початковий момент (состави 1 та 2), а також тих, що будуть підготовлені в процесі розформування (состав 3).

Кроком операції є розформування одного з составів. Кількість кроків k визначається кількістю составів, що будуть готові до розформування в процесі операції (k=3).

Процес функціонування сортувального комплексу є керованим; управління процесом полягає в виборі состава для розформування на кожному кроці операції.

При цьому від вибору состава залежить простій вагонів під накопиченням на кожному кроці і взагалі за період операції.

Необхідно вибрати таку черговість розпуску составів, при якій вказаний простій буде мінімальним.

Поетапні розрахунки

1. Будуємо граф можливих станів парку прийому за час розформування составів, що знаходяться у парку у початковому стані, а також надходять у розформування за цей період (див. рисунок 8.1).

Робота парку розглядається протягом 3 кроків, на кожному з яких розформовується один состав з тих, що знаходяться у парку і готові до розформування.

Стани парку прийому на моменти початку розформування составів позначають колами, поділеними на дві частини. У нижній частині кола вказують номери составів готових до розформування.

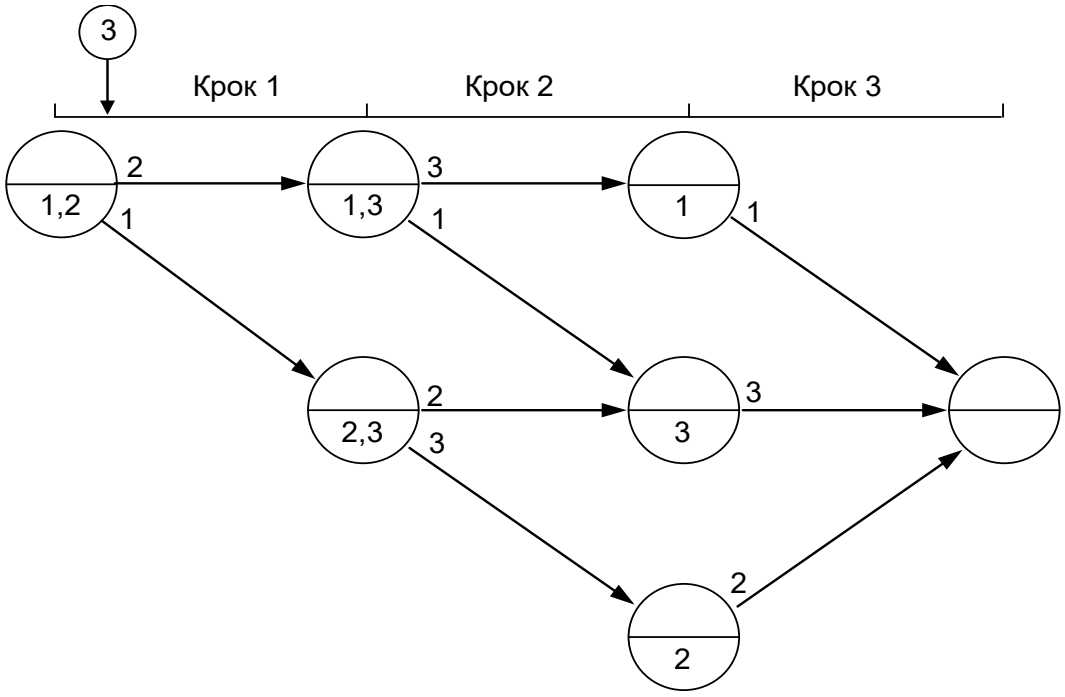


Рисунок 8.1 – Граф можливих станів парку прийому

Переходи із стану в стан позначають лініями (ребрами графа) між колами; над лінією вказують номер состава, який розформовується на даному кроці. Під час 1-го кроку, згідно з вихідними даними, в парку може бути підготовлений до розформування ще один состав. Закінчення підготовки нових составів до розформування необхідно показати стрілками на відповідних кроках над графом станів парку і вказати номери цих составів. Їх необхідно також враховувати при відображенні станів парку у колах.

2. Розраховуємо вагоно-години простою та залишок вагонів у сортувальному парку у випадку, якщо першим буде розформовано состав №1.

Якщо у поточному стані парку прибуття s розформувати i-й состав, то простій вагонів у сортувальному парку за час його розформування T_p знаходимо за таким виразом:

$$nt_s^i = n_s \cdot T_{\text{p}i}, \tag{8.4}$$

де n_s – кількість вагонів у сортувальному парку в стані s;

T_{pi} – тривалість розформування i-го составу.

Загальна кількість вагонів у сортувальному парку у початковому стані s=(1,2) складає:

$$n_{(1,2)} = 22 + 32 + 7 + 1 + 10 + 1 + 14 + 6 + 0 + 46 + 0 = 139 \textit{ ваг.}$$

У випадку, якщо першим буде розформовано состав №1 вагоно-години простою становлять:

$$nt^1_{(1,2)} = 139 \cdot 0{,}29 = 40{,}31 \textit{ ваг-год.}$$

Отримане значення необхідно записати над ребром графа, що позначає перехід із стану (1,2) в стан (2,3) (див. рисунок 8.2).

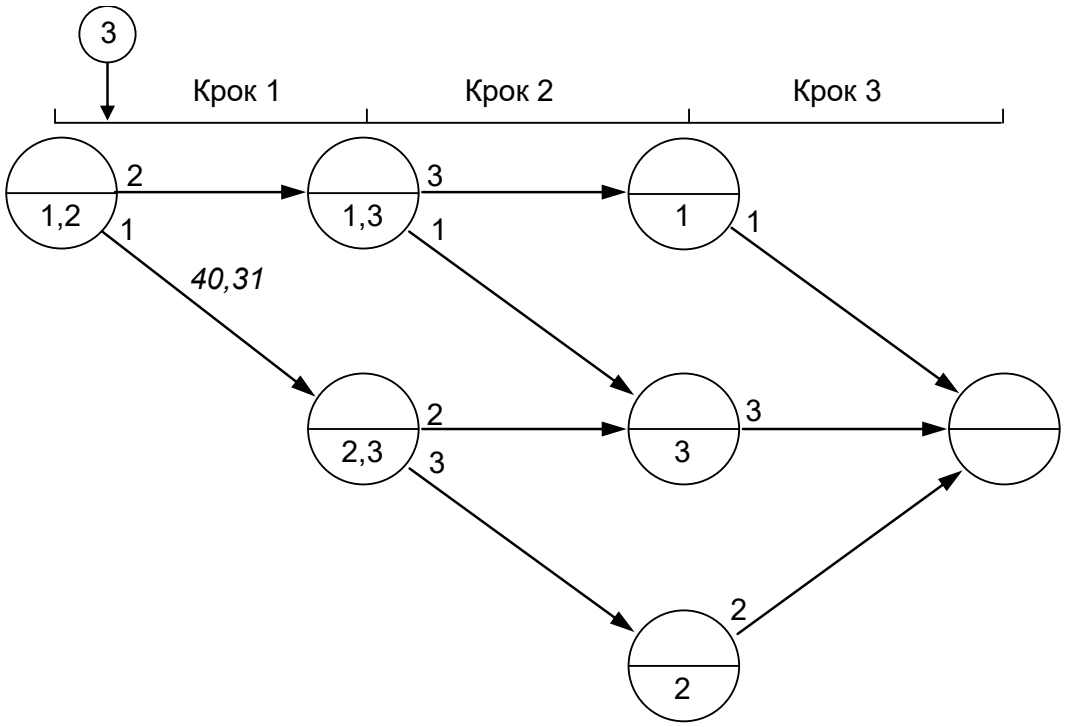


Рисунок 8.2 – Розрахунок простою вагонів у сортувальному парку, якщо першим розформовується состав №1

Залишок вагонів на коліях сортувального парку після розформування состава 1 визначаємо у таблиці 8.3.

3. Розраховуємо вагоно-години простою та залишок вагонів у сортувальному парку у випадку, якщо першим буде розформовано состав №2.

$$nt^{1}_{(1,2)} = 139 \cdot 0{,}31 = 43{,}09 \textit{ ваг-год}.$$

Таблиця 8.3

Залишок вагонів на коліях сортувального парку

Залишок вагонів на сортувальних коліях перед кроком.	Призначення											Всього
	1	2	3	4	5	6	7	8	9	10	11	
	22	32	7	1	10	1	14	6	0	46	0	139
Состав 1	8	31	1	2	2	0	5	0	0	0	3	52
Кількість вагонів на коліях після розформування состава 1	31	65	11	7	17	7	26	14	9	56	14	257
Залишок вагонів після завершення накопичення составу на 2-й колії	30	13	8	3	12	1	19	6	0	46	3	141

Отримане значення необхідно записати над ребром графа, що позначає перехід із стану (1,2) в стан (1,3) (див. рисунок 8.3).

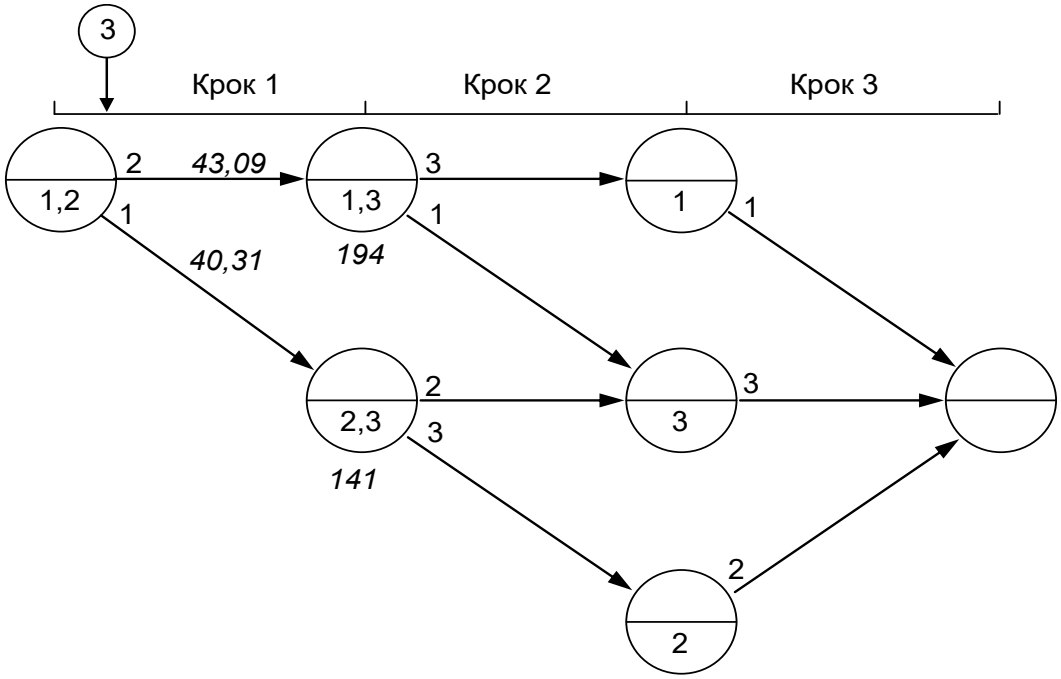


Рисунок 8.3 – Розрахунок простою вагонів у парку наприкінці кроку 1

Залишок вагонів на коліях сортувального парку після розформування состава 2 визначаємо у таблиці 8.4.

Таблиця 8.4

Залишок вагонів на коліях сортувального парку

Залишок вагонів на сортувальних коліях перед кроком.	Призначення											Всього
	1	2	3	4	5	6	7	8	9	10	11	
	22	32	7	1	10	1	14	6	0	46	0	139
Состав 2	5	8	6	2	0	8	0	14	9	1	2	55
Кількість вагонів на коліях після розформування состава 2	27	40	13	3	10	9	14	20	9	47	2	194

4. Аналогічним чином розраховуємо вагоно-години простою та залишок вагонів у сортувальному парку на інших кроках (див. рисунок 8.4).

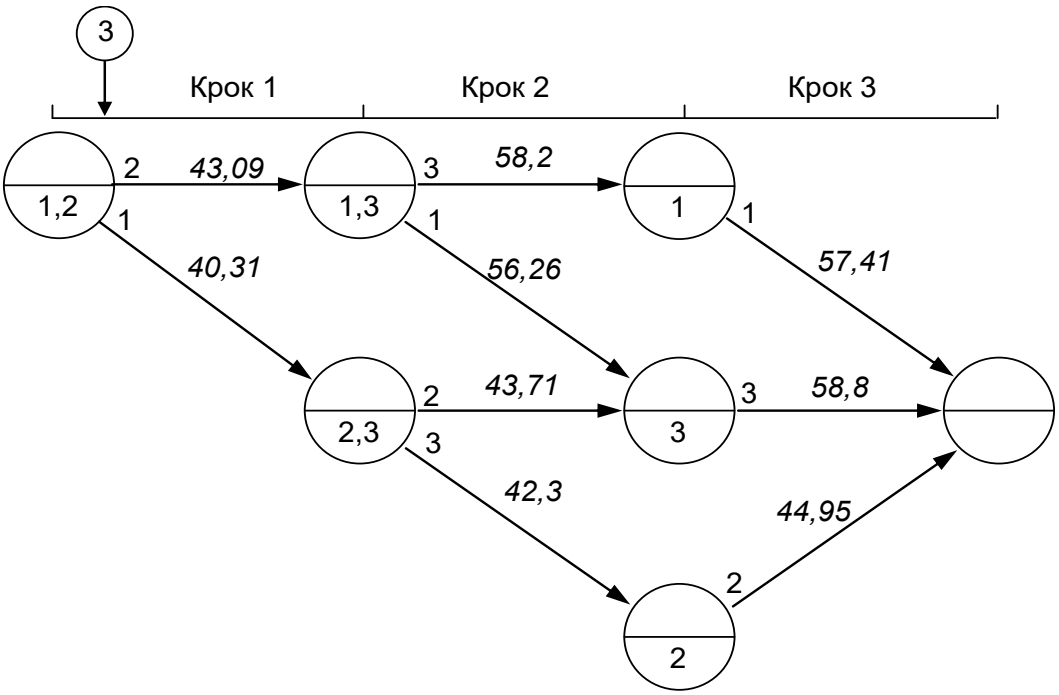


Рисунок 8.4 – Граф можливих станів парку прийому з можливим значенням простоїв вагонів

5. Методом динамічного програмування знайдемо такий порядок розформування составів, який забезпечує мінімальний простій вагонів у сортувальному парку.

Власне рішення задачі

Рішення задачі виконується в два етапи.

Перший етап виконується від кінця операції до її початку; на цьому етапі визначається умовно-оптимальне управління (УОУ) на кожному кроці (номер состава, який розформується), при якому простій вагонів (умовно-оптимальний ефект – УОЕ) буде мінімальним.

У прикладі на початку останнього (3-го) кроку в парку може залишатись тільки один готовий для розформування состав: 1, 2 чи 3 (див. рисунок 8.4). Таким чином, на цьому кроці вибір состава для розформування єдиний, він же є УОУ. Також однозначно визначається і простій вагонів (для состава 1 – 57,41 ваг-год; для состава 2 – 44,95 ваг-год; для состава 3 – 58,8 ваг-год); цей УОЕ записується у верхню частину відповідного кола (див. рисунок 8.5).

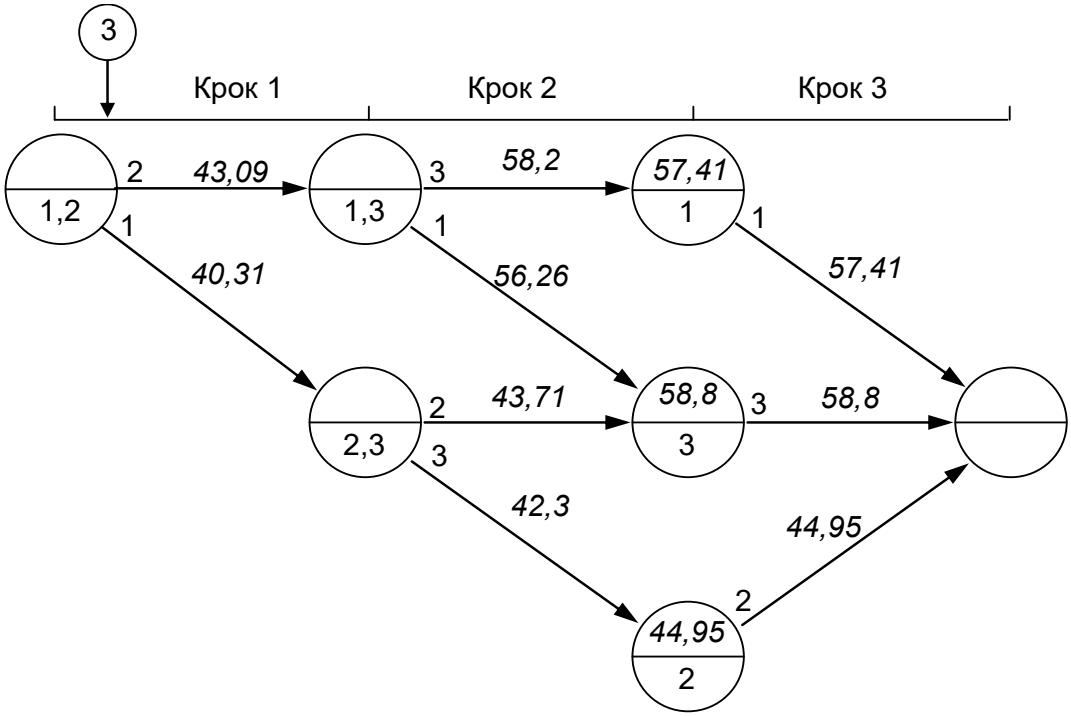


Рисунок 8.5 – Визначення умовно-оптимального рішення

Переходимо до передостаннього (2-го) кроку, до початку якого парк може бути в одному з двох станів (1.3) і (2.3). Спочатку розглянемо стан (1.3) в якому можна вибрати для розпуску состав 1 чи 3.

Загальний простій до кінця операції у цьому випадку складає:

розпуск состава 1: $\sum nt^{1}_{(1,3)} = nt^{1}_{(1,3)} + nt_{(3)} = 56,25 + 58,8 = 115,06$ *ваг-год*;

розпуск состава 3: $\sum nt^{3}_{(1,3)} = nt^{3}_{(1,3)} + nt_{(1)} = 58,20 + 57,41 = 115,61$ *ваг-год*.

Отже, якщо до початку 2-го кроку парк знаходиться у стані (1.3) то УОЕ на другому кроці – состав 1. При цьому УОЕ до кінця операції 115,05 (записати 1 коло, що відповідає стану (1.3)) (див. рисунок 8.6).

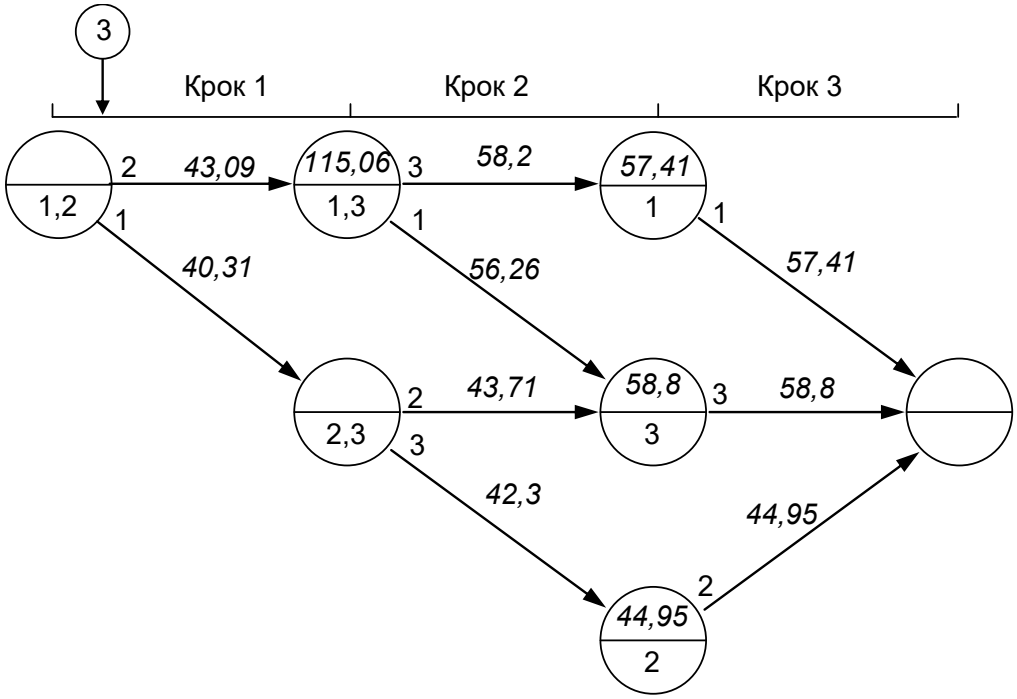


Рисунок 8.6 – Визначення умовно-оптимального рішення

В результаті аналогічних розрахунків для стану (2.3) встановлено УОУ – состав 3; УОЕ складає $\Sigma nt_{(23)}$ = 87,25 ваг-год (див. рисунок 8.7.)

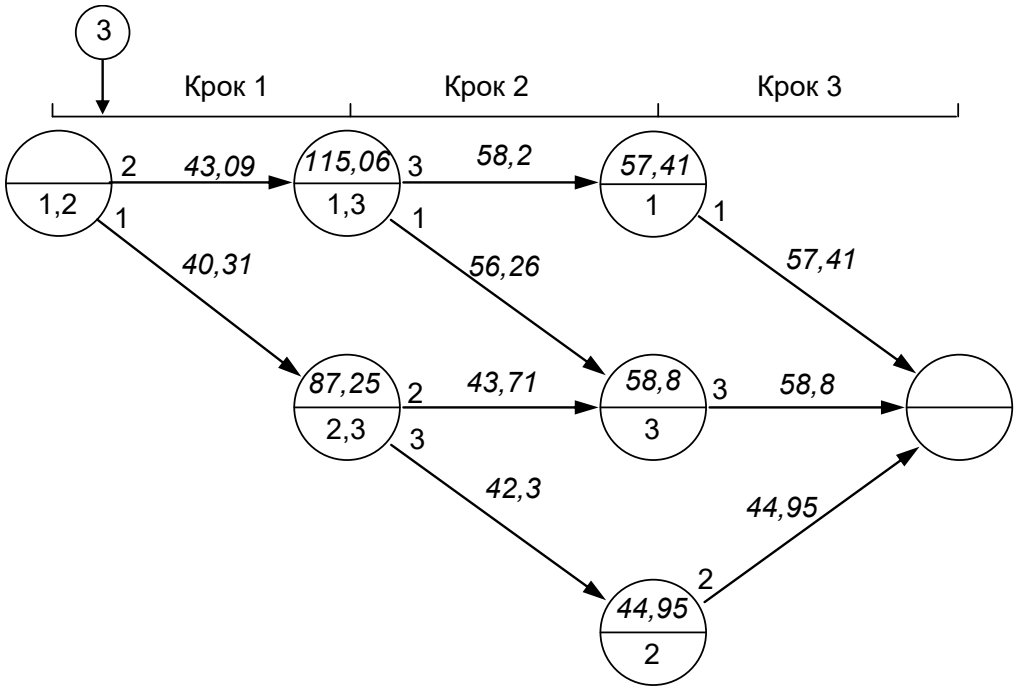


Рисунок 8.7 – Визначення умовно-оптимального рішення

На початку 1-го кроку парк може знаходитись в єдиному стані – (1.2). При цьому можливо вибрати для розпуску состав 1 чи 2. Знайдемо простій до кінця операції в цих випадках.

розпуск состава 1: $\sum nt^1_{(1.2)} = nt_1 + nt_{(2.3)} = 40,31 + 87,25 = 127,56$ ваг-год;

розпуск состава 2: $\sum nt^2_{(1.2)} = nt_2 + nt_{(1.3)} = 43,09 + 115,06 = 158,15$ ваг-год.

Таким чином на першому кроці УОЕ є розформування состава 1 (позначимо стрілкою ребро графа (1.2)–(2.3), а УОЕ 127,56 запишемо у коло, що відповідає початковому стану) (див. рисунок 8.8).

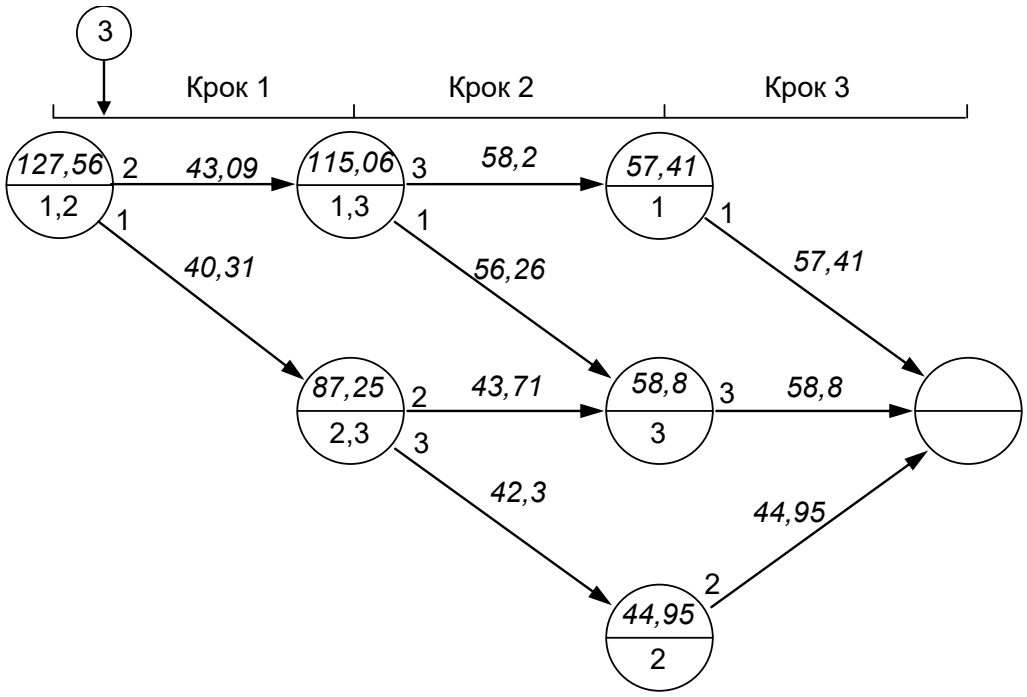


Рисунок 8.8 – Визначення умовно-оптимального рішення

<u>Другий етап</u> виконується від початку до кінця операції; на цьому етапі знаходиться безумовно-оптимальне управління (БОУ) на кожному кроці, а також безумовно-оптимальний ефект від всієї операції. Початковий стан операції (1.2) визначено однозначно; в цьому стані відомо УОУ (розпуск состава 1), воно ж є і БОУ на першому кроці. На другому кроці відомо УОУ для всіх станів. Так як на першому кроці БОУ є розпуск состава 1, то перед другим кроком парк буде в стані (2.3). Для нього БОУ є состав 3 (позначений стрілкою).

Рухаючись таким чином до кінця операції, послідовно знаходимо оптимальну черговість розформування 1-3-2, при цьому простій вагонів під накопиченням в сортувальному парку складає 127,56 ваг-год (див. рисунок 8.9).

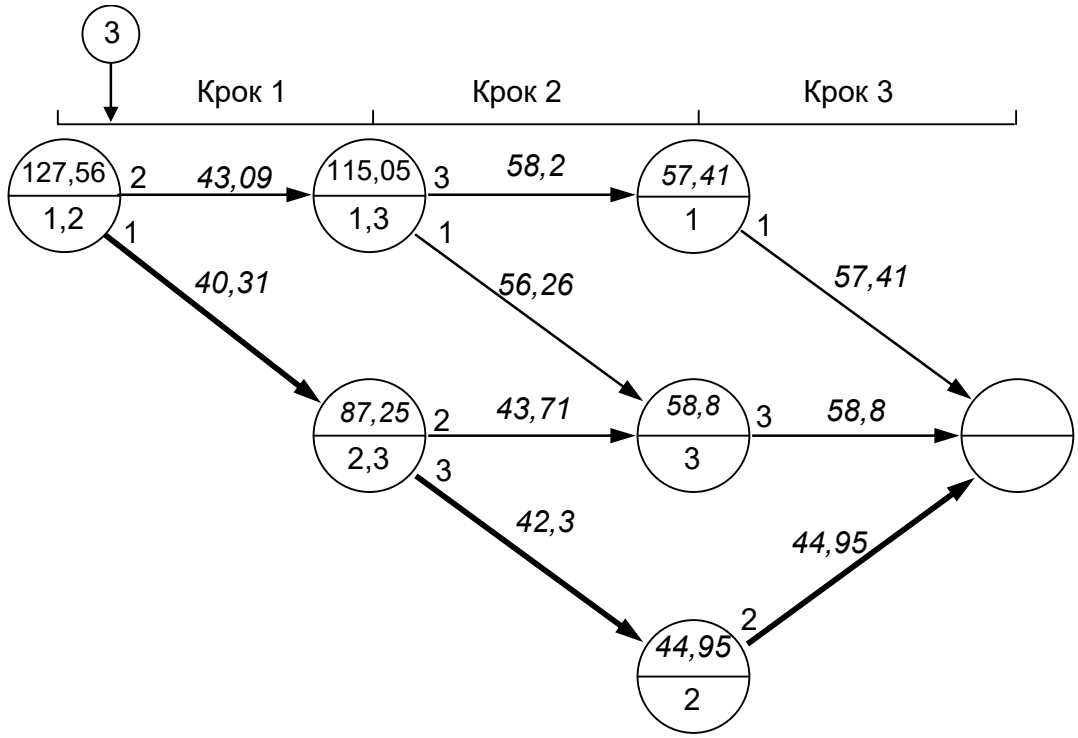


Рисунок 8.9 – Визначення безумовно-оптимального рішення

Графік накопичення составів у сортувальному парку при оптимальній черговості розпуску наведено на рисунку 8.10.

Переваги методів динамічного програмування

1. Методи динамічного програмування можна застосовувати за будь-якого способу завдання цільової функції та з будь-якою припустимою множиною станів та керувань. Цієї переваги позбавлені класичні методи оптимізації та інші обчислювальні методи математичного програмування.

2. Обчислювальні схеми методів динамічного програмування в дискретному випадку пов'язані з перебиранням оптимальних значень показника ефективності та керування на кожному кроці для всіх можливих значень змінної стану, але обсяг розрахунків при цьому значно менший, ніж при прямому перебирання варіантів. Це пов'язано з тим, що на етапі умовної оптимізації невдалі варіанти відразу відкидаються, а зберігаються лише умовно-оптимальні на даному кроці.

3. Методи динамічного програмування дають можливість аналізу чутливості до зміни вихідних даних станів s_k та їх кількості n. Фактично тут на кожному кроці розв'язується не одна задача, а множина однотипних задач для різних станів s_k $(1 < k < n)$.

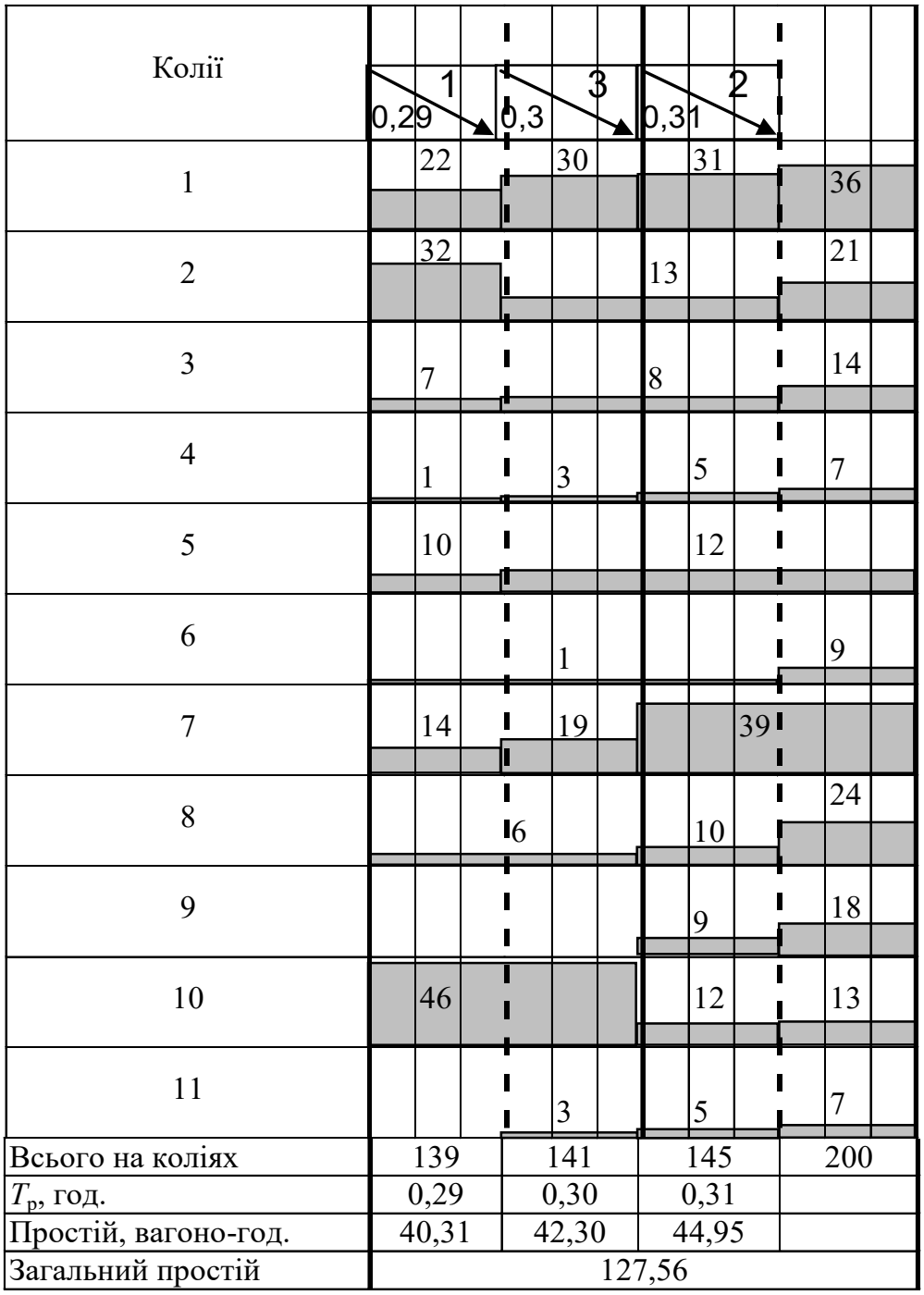


Рисунок 8.10 – Графік накопичення вагонів у сортувальному парку при оптимальній черговості розформування

Тому зі зміною вихідних даних можна не розв'язувати задачу заново, а зробити лише нескладні додавання до вже виконаних розрахунків, тобто продовжити вже розв'язану задачу за рахунок збільшення кількості кроків n або кількості значень s_k.

Недоліки методів вибору черговості розпуску составів

Система критеріїв ефективності черговості розпуску составів потребує удосконалення.

В більшості випадках мінімізуються загальні експлуатаційні витрати окремої сортувальної станції. Жоден із відомих методів вибору черговості розпуску составів не враховує можливості створення більш сприятливих умов переробки вагонопотоків на наступних технічних станціях.

Отже, сортувальну станцію необхідно розглядати як частину транспортної мережі, що складається із багатьох станцій.

У зв'язку з цим необхідно розробити систему критеріїв, яка мінімізуватиме загальні експлуатаційні витрати в цілому на шляху прямування вагонопотоків.

Створення сприятливих умов роботи наступних технічних станцій полігону за допомогою вибору черговості розпуску составів можливе за рахунок формування поїздів із меншою кількістю відчепів, що позитивно впливає на показники роботи підсистем розформування цих станцій.

Різні методи, присвячені вирішенню задачі вибору черговості розпуску составів, розглядають сортувальну станцію ізольовано від загальної залізничної мережі.

Це являється певним недоліком, оскільки станції пов'язані між собою. Вихідний потік поїздів однієї станції є вхідним потоком для іншої. Покращення показників роботи однієї станції може відбуватись ціною погіршення показників інших станцій та в цілому залізничної мережі.

Тому черговість розпуску составів повинна враховувати можливий вплив на умови роботи наступних технічних станцій.

На рисунку 8.11 наведено приклад, що демонструє, яким чином змінюється середня кількість відчепів у составі свого формування при різних послідовностях розпуску.

Перед початком розпуску в парку прийому знаходяться два состави.

У випадку розформування составів у порядку їх надходження, у сортувальному парку накопичується два состави, кожен із яких складається із п'яти відчепів.

У випадку розформування составів у зворотному порядку, в сортувальному парку накопичується два состави, кожен із яких складається із чотирьох відчепів.

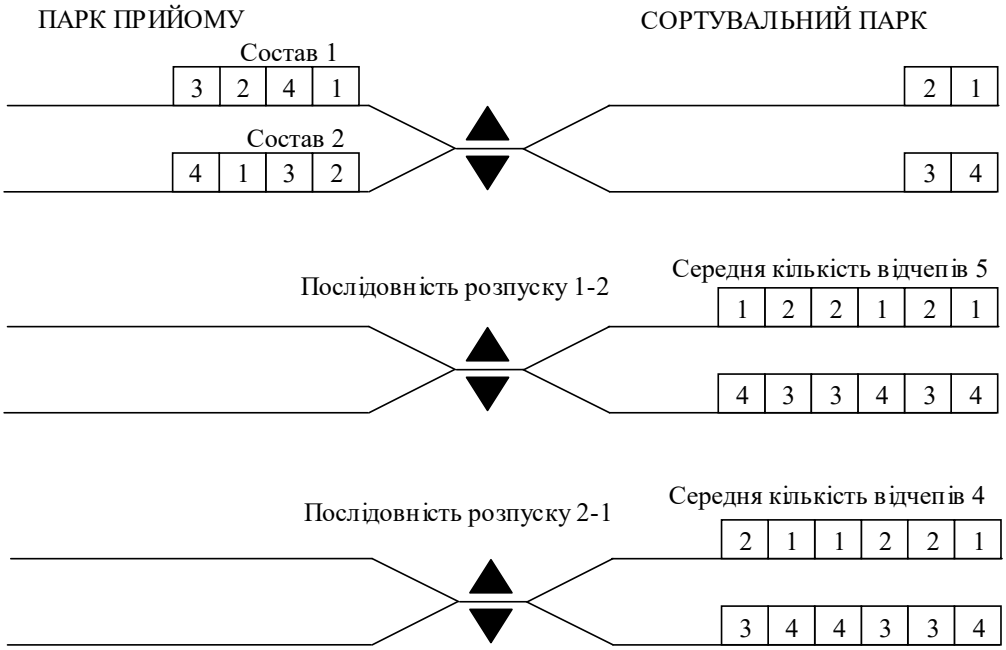


Рисунок 8.11 – Приклад впливу черговості розпуску составів на кількість відчепів у составах свого формування

Отже, якщо всю множину черговостей позначимо як $X = \{X^{(t)}\}$. Серед усіх варіантів $X^{(t)}$ слід відібрати такий, який забезпечує мінімальні загальні експлуатаційні витрати сортувальної станції. В цьому випадку цільова функція задачі вибору черговості розпуску составів формулюється наступним чином:

$$C(X^{(t)}) = C_{ваг-год}(X^{(t)}) + C_{п-год}(X^{(t)}) + C_{лок-год}(X^{(t)}) + C_{ман}(X^{(t)}) \to \min \quad (8.5)$$

де $C_{ваг\text{-}год}(X^{(t)})$ – витрати, що пов’язані із простоєм вагонів на станції при реалізації черговості розпуску $X^{(t)}$;

$C_{п\text{-}год}(X^{(t)})$ – витрати, що пов’язані із простоєм поїздів по неприйому на станцію при реалізації черговості розпуску $X^{(t)}$;

$C_{лок\text{-}год}(X^{(t)})$ – витрати, що пов’язані із простоєм локомотивів на станції при реалізації черговості розпуску $X^{(t)}$;

$C_{ман}(X^{(t)})$ – витрати, що пов’язані із додатковою маневровою роботою на станції при реалізації черговості розпуску $X^{(t)}$ (дана складова витрат виникає при переповненні сортувальних колій).

З погляду на загальні експлуатаційні витрати сортувальної станції, ідеальною є така послідовність розпуску, яка забезпечує:

– обробку в парку прийому составів із замикаючими групами без непродуктивних простоїв;

– відсутність затримок поїздів по неприйому на станцію;

– відсутність затримок поїзних локомотивів при їх прибиранні від поїздів в розформування;

– направлення усіх вагонів на спеціалізовані сортувальні колії;

– розформування останнього составу у мінімально можливий термін (для врахування ефекту післядії черговості розпуску на показники).

Розглянемо на прикладі як можна скоротити об’єм розрахунків, пов’язаних із аналізом варіантів черговості розпуску, при використанні поняття ідеальної послідовності.

На рисунку 8.12 наведено приклад обробки трьох составів у парку прийому.

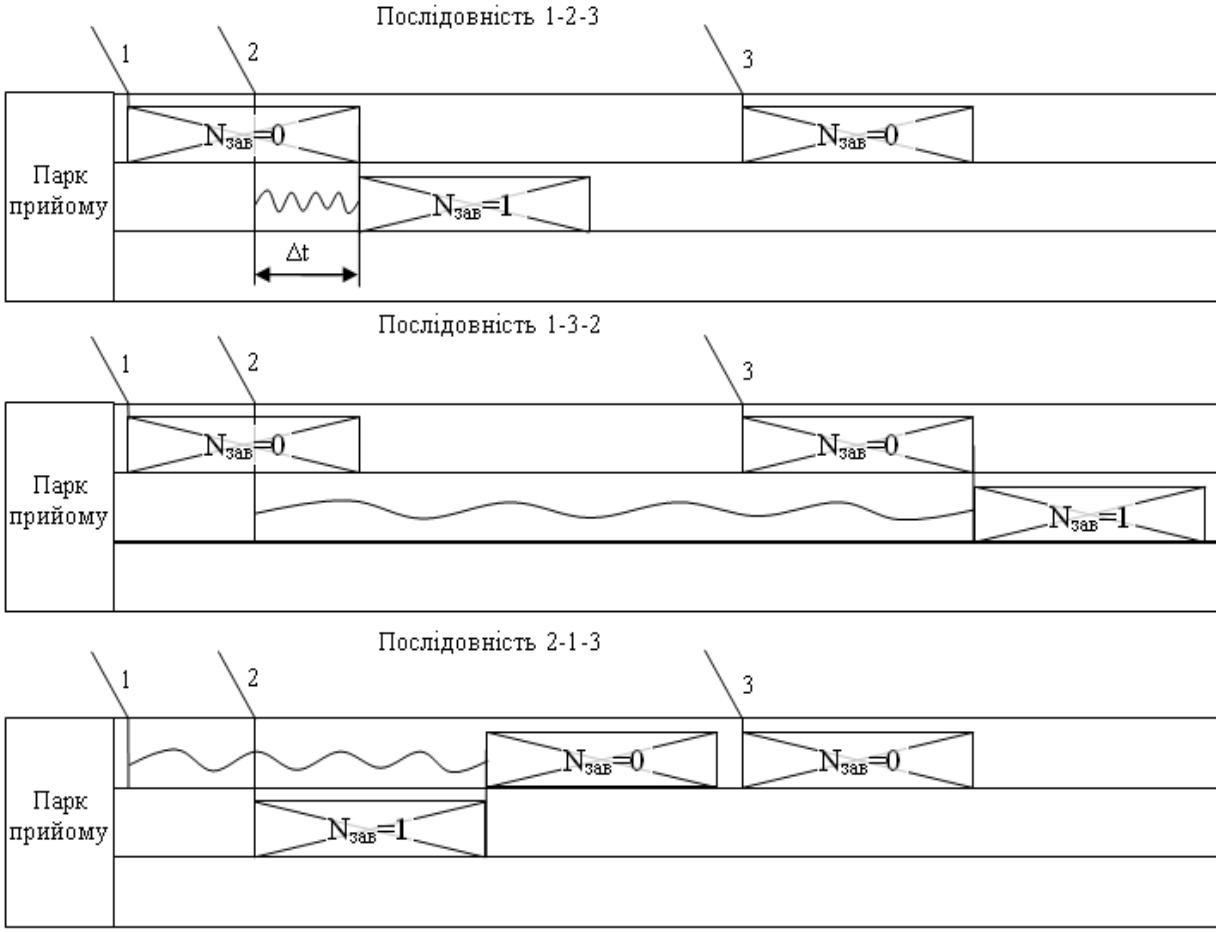


Рисунок 8.12 – Приклад використання критерію ідеальної послідовності розпуску при аналізі варіантів послідовності

Другий состав має в своєму складі одну замикаючу групу, решта составів замикаючих груп не мають. Виконується аналіз можливих варіантів черговості обробки составів із вибором найкращого. При трьох составах маємо 6 варіантів черговості розпуску (див. таблицю 8.5).

Таблиця 8.5

Варіанти черговості розпуску трьох составів

№ варіанту	Послідовність розпуску
1	1–2–3
2	1–3–2
3	2–1–3
4	2–3–1
5	3–1–2
6	3–2–1

Аналіз починається із послідовності 1-2-3 (відповідає принципу обробки FI-FO). Далі поступово перебираються всі варіанти рухаючись у напрямку зворотної послідовності розпуску 3-2-1 (відповідає принципу обробки LI-FO).

Для першого варіанту єдиний состав із завершальною групою затримується на величину часу Δt, отже така послідовність не є ідеальною, і її можна спробувати покращити. Наступний варіант, що обирається – 1-3-2. При такій послідовності другий состав має ще більші затримки, тому обирається третій варіант 2-1-3.

В цьому випадку состав із замикаючою групою обробляється в парку прийому без непродуктивних простоїв, крім того, останній состав послідовності розформовується також без затримок і тому не погіршує умови роботи в парку прийому після реалізації обраної послідовності (обробка составів 4, 5 і т. д.).

Отже послідовність 2-1-3 є ідеальною і не може бути поліпшена, тому оцінку решти варіантів можна не визначати.

Контрольні запитання

1. Для вирішення якого типу задач доцільно використовувати методи динамічного програмування?
2. Яка основна особливість динамічного програмування?
3. Що розуміють під багатоетапністю операцій?
4. Яка основна складність динамічного програмування?
5. Які види оптимальних рішень задач динамічного програмування та в якій послідовності вони знаходяться?
6. Скільки разів проходиться багатокроковий процес при оптимізації управління методом динамічного програмування?

7. Який критерій було обрано при вирішенні задачі вибору черговості розформування составів?

8. Які вихідні дані потрібні для вирішення задачі вибору черговості розформування составів?

9. Що записується у верхній та нижній частинах кола, яке позначає стан роботи парку прийому, при вирішенні задачі вибору черговості розформування составів?

10. Назвати основні переваги методів динамічного програмування.

11. Назвати основні недоліки методів вибору черговості розпуску составів.

РОЗДІЛ 9

Методи формування багатогрупних составів на залізничних станціях

9.1 Теоретичні відомості

Формування составів, особливо багатогрупних, є одним з найбільш трудомістких елементів технологічного процесу переробки вагонів на станціях [14 – 17]. Ця задача широко вирішується під час закінчення формування збірних поїздів, коли вагони необхідно підібрати у порядку призначення на проміжні станції дільниці і у разі добірки вагонів по фронтах вантажної роботи на вантажних станціях.

Тривалість виконання сортування состава залежить від різних факторів: початкової упорядкованості состава, типу сортувального пристрою та маневрового локомотива; методу сортування (осаджуванням, одиночними або серійними поштовхами) та порядку виконання маневрів.

В оперативних умовах тривалість формування состава в основному залежить від порядку виконання маневрових напіврейсів. Цей порядок визначає складач поїздів, в основному спираючись на власний досвід. Природно, що тривалість формування може змінюватися в широких межах.

Тим часом існують математичні методи, використання яких дозволяє досить ефективно виконувати формування.

Попередньо встановимо деякі угоди та визначення. Будемо вважати, що для формування поїздів використовується сортувальний пристрій (гірка, витяжна колія) та певна кількість сортувальних колій m ($2 \le m \le \theta$). Формування состава здійснюється за N етапів, кількість яких залежить від методу формування, кількості сортувальних колій, що використовуються, m і кількості груп θ у невпорядкованому составі. Кожен етап формування, незалежно від методу, який використовується, включає дві операції: збирання вагонів із $m_{сб}$ колій і їх сортування на $m_с$ колій.

Прийнято, що перед початком формування вагони состава знаходяться на витяжній колії, так що на першому етапі збирання вагонів не виконується. На останньому етапі вагони повинні бути зібрані на одній із сортувальних колій (m_{cN}=1). На інших етапах кількість колій, що використовується, залежить від методу формування та від їх наявної кількості *M*.

9.2 Комбінаторний метод формування составів

Характеристика методу

Комбінаторний метод ґрунтується на представленні чисел у системі числення Фібоначчі.

Номери груп γ = 0, …, 16 у зазначеній формі запису наведено в табличній формі, в якій вказано кількість етапів формування *N*, яка необхідна, якщо максимальний номер групи в составі дорівнює γ.

Ряд Фібоначчі складають числа, кожне з яких представляє собою суму *m* попередніх чисел. Наприклад, при *m* =3 маємо ряд, який наведено у таблиці 9.1.

Таблиця 9.1

Ряд Фібоначчі при *m*=3

	$\overbrace{\quad}^{m\text{-}1}$							
0	0	1	1					
	0	1	1	2				
		1	1	2	4			
			1	2	4	7		
				2	4	7	13	
					4	7	13	24

Для розробки схем формування составів використовується система кодування чисел у двійковій формі.

Так, при *m*=3 коди виглядають так, як наведено у таблиці 9.2.

Таблиця 9.2

Коди Фібоначчі при *m*=3

γ	Ряд Фібоначчі					код $\Phi_3(\gamma)$
	13	7	4	2	1	
0	0	0	0	0	0	0
1	0	0	0	0	1	1
2	0	0	0	1	0	10
3	0	0	0	1	1	11
4	0	0	1	0	0	100
5	0	0	1	0	1	101
6	0	0	1	1	0	110
7	0	1	0	0	0	1000
8	0	1	0	0	1	1001

Номери груп γ = 0, …, 16 у системі запису Фібоначчі наведені у таблиці 9.3; у цій же таблиці показано кількість етапів формування *N*, яка необхідна якщо максимальний номер групи в составі дорівнює γ.

Для визначення логічних номерів колій при сортуванні, при сортуванні номерам груп {γ = 0, 1, …, θ–1} ставляться у відповідність числа із системи запису Фібоначчі, які наведено у таблиці 9.3 у зворотному порядку: $\Phi_m(q-1), \Phi_m(q-2), ..., \Phi_m(0)$.

Зворотна послідовність вказаних чисел використовується для того, щоб у сформованому составі групи були розташовані у прямому порядку (0, 1,…, θ–1).

Як видно, у кожному стовбці таблиці 9.3 знаходяться не більше, ніж *m*-1 одиниць поспіль.

Числа $\Phi_m(\gamma)$ використовуються для визначення логічного номера колії (ЛНК) μ_i, на яку потрібно направити ***i***-й відчеп з логічним номером групи (ЛНГ) γ_i при сортуванні на кожному етапі формування багатогрупного состава.

Для визначення ЛНК μ_i на *j*-му етапі виконується послідовний аналіз цифр у запису $\Phi_m(\gamma)$:

$$\mu_i = k, \text{ якщо } \varphi_{k+j}=0,\ \varphi_{k+j-1} = \ldots = \varphi_{j+1} = \varphi_j = 1;\ (0 \le \mu_i \le m-1) \qquad (9.1)$$

Таблиця 9.3

Запис номерів груп у системі Фібоначчі

Номер групи γ	Кількість сортувальних колій *m*							
	2		3		4		5	
	$\Phi_2(\gamma)$	*N*	$\Phi_3(\gamma)$	*N*	$\Phi_4(\gamma)$	*N*	$\Phi_5(\gamma)$	*N*
0	0	2	0	2	0	2	0	2
1	1		1		1		1	
2	10	3	10	3	10	3	10	3
3	100	4	11		11		11	
4	101		100	4	100	4	100	4
5	1000	5	101		101		101	
6	1001		110		110		110	
7	1010		1000	5	111		111	
8	10000	6	1001		1000	5	1000	5
9	10001		1010		1001		1001	
10	10010		1011		1010		1010	
11	10100		1100		1011		1011	
12	10101		1101		1100		1100	
13	100000	7	10000	6	1101		1101	
14	100001		10001		1110		1110	
15	100010		10010		10000	6	1111	
16	100100		10011		10001		10000	6

При комбінаторному методі після сортування вагонів на ***j***- му етапі на наступному (j+1)-му етапі сортують вагони, що знаходяться на колії, яка на j-му етапі мала ЛНК μ=0. Відсутність операцій збирання вагонів після кожного сортування є істотною перевагою комбінаторного методу, порівняно з іншими. При цьому, після кожного етапу логічні номери усіх колій, які використовуються, циклічно змінюються.

Для визначення ЛНК може використовуватися такий рекурентний вираз:

$$\mu_{j+1} = (\mu_j + m - 1) \bmod m. \qquad (9.2)$$

Алгоритм формування состава комбінаторним методом

1. На основі таблиці 9.3 кожній групі вагонів відповідно до її умовного номера та кількості сортувальних колій присвоїти двійковий код Фібоначчі $\Phi_m(\gamma)$.

Для того, щоб групи вагонів були відсортовані у прямому порядку, коди груп необхідно присвоювати у зворотному порядку, тобто γ визначається за виразом

$$\gamma = v - k - 1, \tag{9.3}$$

де v – загальна кількість умовних призначень;

k – умовний номер призначення.

2. Сортувальним коліям присвоїти умовні номери 0, 01, 011, …

3. У процесі сортування відправити групи вагонів на ті колії, номером яких закінчується їх код (при перегляді справа наліво до першого нуля, що зустрінеться).

4. Витягнути на витяжну колію вагони, що стоять на колії 0.

5. Циклічно змістити нумерацію колій вниз.

6. Викреслити останню цифру кодів призначень груп вагонів і перейти до кроку 3.

Кроки 3–6 повторюються циклічно до моменту формування состава на колії № 0 сортувального парку.

9.3 Розподільчий метод формування составів

Характеристика методу

В основі розподільчого методу лежить представлення чисел в позиційній системі числення.

Основа системи числення обирається рівною кількості сортувальних колій m.

Наприклад, для представлення чисел у трійковій системі використовуються три числа 0, 1, 2 (див. таблицю 9.4).

Таблиця 9.4

Представлення чисел у трійковій системі

Десяткова	0	1	2	3	4	5	6	7	8	9
Трійкова	0	1	2	10	11	12	20	21	22	100

Записи чисел в позиційних системах з різною основою та кількість етапів формування составів N наведені в таблиці 9.5.

Таблиця 9.5

Запис номерів груп в позиційних системах числення

Номер групи γ	Кількість сортувальних колій m							
	2		3		4		5	
	$\psi_2(\gamma)$	N	$\psi_3(\gamma)$	N	$\psi_4(\gamma)$	N	$\psi_5(\gamma)$	N
0	0	2	0	2	0	2	0	2
1	1		1		1		1	
2	10	3	2		2		2	
3	11		10	3	3		3	
4	100	4	11		10	3	4	
5	101		12		11		10	3
6	110		20		12		11	
7	111		21		13		12	
8	1000	5	22		20		13	
9	1001		100	4	21		14	
10	1010		101		22		20	
11	1011		102		23		21	
12	1100		110		30		22	
13	1101		111		31		23	
14	1110		112		32		24	
15	1111		120		33		30	
16	10000	6	121		100	4	31	

Числа $\psi_m(\gamma)$ використовуються для визначення логічного номеру колії (ЛНК) μ_i, на яку потрібно направити i-й відчеп с ЛНГ γ_i при сортуванні на кожному етапі формування. Для визначення ЛНК μ_i на j-му етапі виконують послідовний запис цифр. При цьому номер колії, на яку направляється відчеп i-го призначення на j-му етапі сортування визначається j-ю цифрою в запису коду i-го призначення. Після кожного етапу сортування вагони збираються на витяжній колії зі всіх сортувальних колій.

Алгоритм формування состава розподільчим методом

1. На основі таблиці 9.2 кожній групі вагонів відповідно до її умовного номера та кількості сортувальних колій присвоїти код позиційної системи числення $\Psi_m(\gamma)$.

2. Сортувальним коліям присвоїти умовні номери 0, 1, 2, 3, …

3. У процесі сортування відправити групи вагонів на ті колії, логічним номером яких закінчується їх поточний фрагмент коду.

4. Зібрати вагони зі всіх сортувальних на витяжній колії, починаючи з колії, що має найбільший порядковий номер і закінчуючи колією № 0.

5. Викреслити останню цифру з коду призначення та перейти до кроку 3.

Кроки 3–5 повторюються циклічно до моменту формування состава на колії № 0 сортувального парку.

9.4 Приклад розв’язання задач. Задача 10

Умова задачі

Виконати формування багатогрупного состава на витяжній колії комбінаторним методом при кількості сортувальних колій $m = 3$.

Порядок розміщення вагонів у початковому составі наступний:

3	5	1	3	2	3	5	6	0	4

Розв’язання задачі

1. Присвоюємо коди призначенням вагонів (див. таблицю 9.6):

Таблиця 9.6

Коди призначень вагонів при сортуванні комбінаторним методом

Призначення вагонів	K	$\Phi_3(\gamma)$
0	6	110
1	5	101
2	4	100
3	3	011
4	2	010
5	1	001
6	0	000

2. Далі відповідно до алгоритму формування состава комбінаторним методом записуємо весь початковий (до сортування) состав на витяжну колію, а сортувальним коліям присвоюємо умовні номери 0, 01, 011 (див. рисунок 9.1):

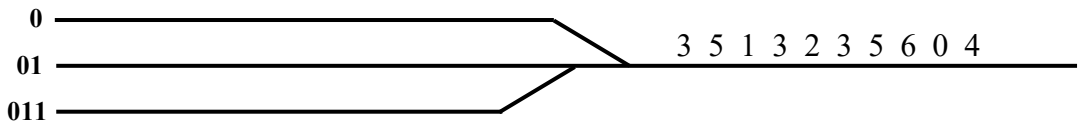


Рисунок 9.1 – Положення состава перед сортуванням (етап 1)

3. Після цього в процесі сортування відправляємо групи вагонів на ті колії, номером яких закінчується їх код (при перегляді справа наліво до першого нуля, що зустрінеться). Так, вагони призначенням 3, потрібно спрямувати на сортувальну колію 011 (див. рисунок 9.2), тому що цифра 0 зустрілася після двох одиниць у двійковому коді 011.

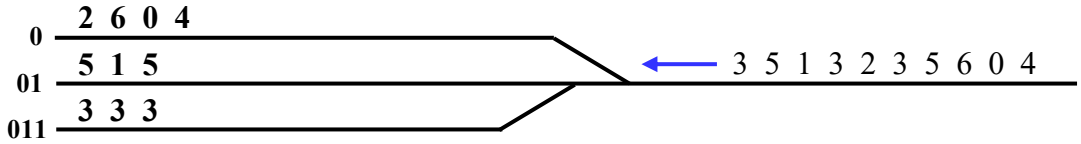


Рисунок 9.2 – Стан сортувального парку після сортування (етап 1)

Вагони призначенням 5 необхідно спрямувати на сортувальну колію № 1 (двійковий код 001, в першому розряді маємо 1, а в другому розряді маємо 0).

4. Після першого етапу, відповідно до алгоритму, вагони з колії 0 витягуємо на витяжну колію формування, тобто на витяжній колії вагони будуть розташовуватися у такому порядку: 2, 6, 0, 4. Решта вагонів залишаються на сортувальних коліях (див. рисунок 9.3).

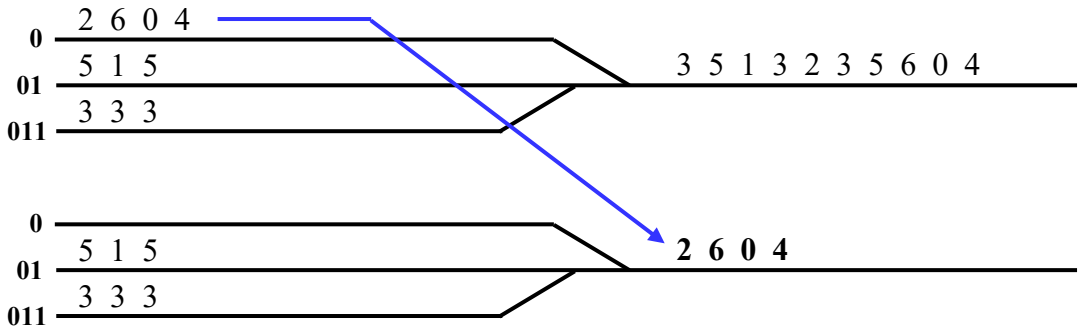


Рисунок 9.3 – Збирання вагонів після сортування (етап 1)

5. Далі циклічно зміщуємо нумерацію колій вниз (див. рисунок 9.4).

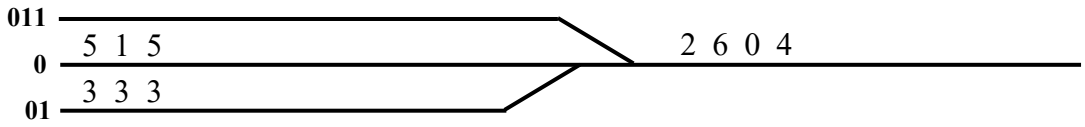


Рисунок 9.4 – Перенумерація сортувальних колій після збирання (етап 1)

6. Після цього, викреслюємо останню цифру коду (див. таблицю 9.5) і переходимо до наступного етапу сортування вагонів (див. рисунок 9.7).

Таблиця 9.7

Коди призначень вагонів на другому етапі

Призначення вагонів	$Ф_3(\gamma)$
0	011~~0~~
1	010~~1~~
2	010~~0~~
3	001~~1~~
4	001~~0~~
5	000~~1~~
6	000~~0~~

Повторюємо ці дії декілька разів (до повного формування составу).

На другому етапі вагони призначенням 2 і 6 потрібно спрямувати на сортувальну колію 0, вагони призначенням 0 – на сортувальну колію 011, а вагони призначенням 4 – на сортувальну колію 01 (див. рисунок 9.5).

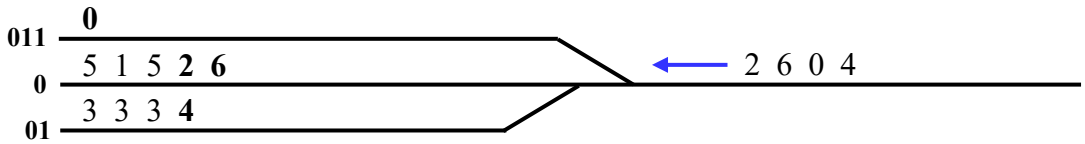


Рисунок 9.5 – Стан сортувального парку після сортування (етап 2)

Після чергового етапу викреслюємо всюди по одній цифрі коду.

На третьому етапі вагони розподіляються за поточним фрагментом коду. Останній етап – це збирання вагонів, який виконується без використання цифр коду (див. рисунок 9.6).

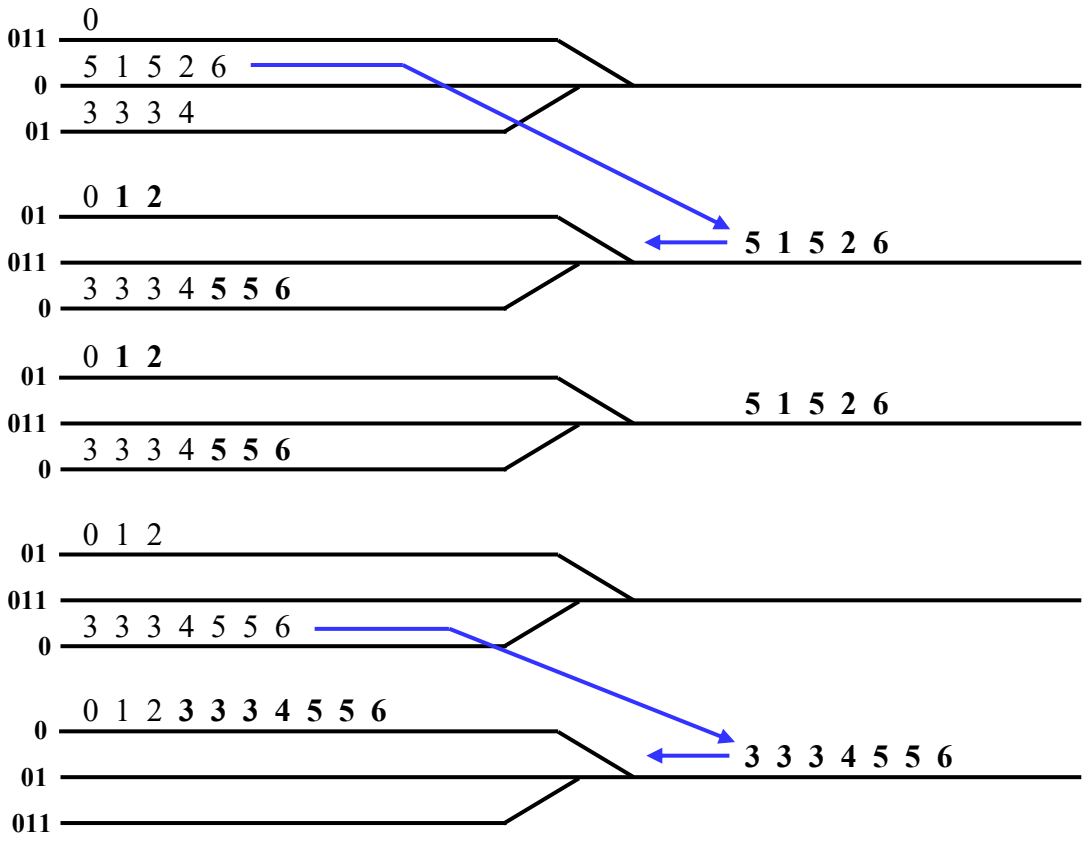


Рисунок 9.6 – Стан сортувального парку після сортування (етапи 3-4)

Отже, процес формування багатогрупного состава комбінаторним методом завершено за 4 етапи.

9.5 Приклад розв’язання задач. Задача 11

Умова задачі

Виконати формування багатогрупного состава на витяжній колії розподільчим методом при кількості сортувальних колій $m = 3$.

Порядок розміщення вагонів у составі наступний:

3	5	1	3	2	3	5	6	0	4

Розв’язання задачі

1. Присвоюємо коди призначенням вагонів (див. таблицю 9.8):

Таблиця 9.8

Коди призначень вагонів при сортуванні розподільчим методом

Призначення вагонів	Ψ_3 (γ)
0	000
1	001
2	002
3	010
4	011
5	012
6	020

Таблиця 9.8 заповнюється аналогічно до таблиці 9.6, але коди присвоюються на основі таблиці 9.5. Замість старших розрядів, яких не вистачає, записують нулі.

2. Відповідно до алгоритму записуємо весь початковий (не відсортований) состав на витяжну колію, а сортувальним коліям присвоюємо номери 0, 1, 2 (див. рисунок 9.7).

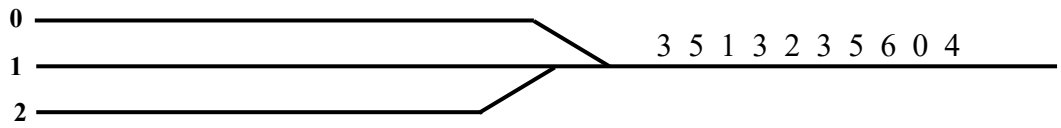


Рисунок 9.7 – Положення состава перед сортуванням (етап 1)

3. Вагони у процесі сортування відправляємо на ті колії, номером яких закінчується їх код. Отже, після першого етапу сортування отримаємо стан сортувального парку, що наведено на рисунку 9.8.

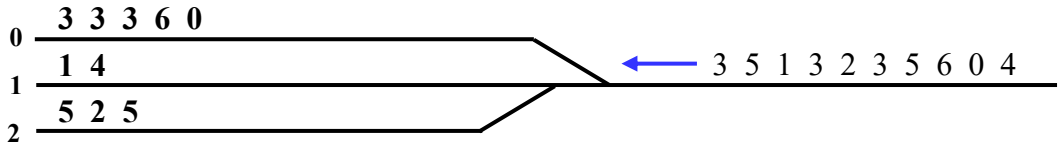


Рисунок 9.8 – Стан сортувального парку після сортування (етап 1)

4. На відміну від комбінаторного методу, після чергового етапу сортування вагони на витяжній колії збираємо зі всіх колій, починаючи з колії, що має найбільший логічний номер. Отже, маневровий локомотив спочатку заїжджає на колію 2, збирає вагони 5, 2, 5, далі

виїжджає з ними на витяжну колію, після чого заїжджає на колію 1 за групою вагонів 1, 4. Після збирання всіх вагонів на першому етапі сортування на витяжній колії буде знаходитись весь состав у такому порядку: 3, 3, 3, 6, 0, 1, 4, 5, 2, 5 (див. рисунок 9.9).

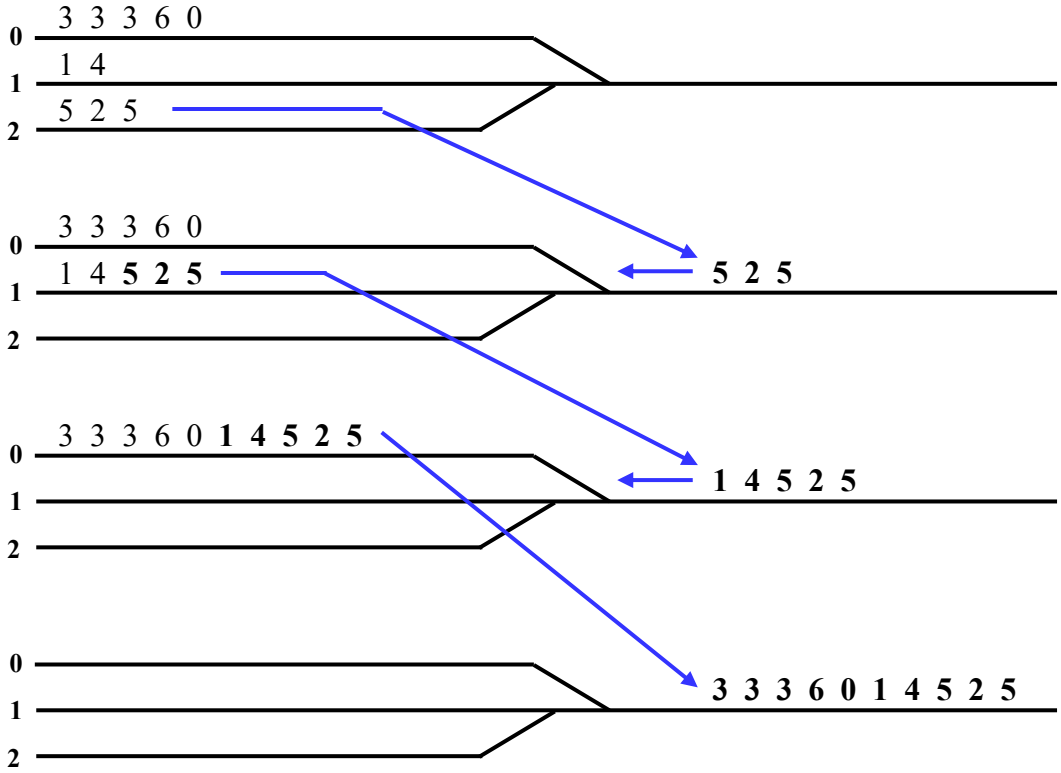


Рисунок 9.9 – Збирання вагонів після сортування (етап 1)

5. Далі викреслюємо останню цифру коду в таблиці 9.8 і переходимо до чергового етапу сортування. Тепер вагони призначенням 3 необхідно спрямувати на колію 1 (на першому етапі – колія 0), а вагони призначенням 6 необхідно спрямувати на колію 2 (на першому етапі – колія 0) і т. д. (див. рисунок 9.10).

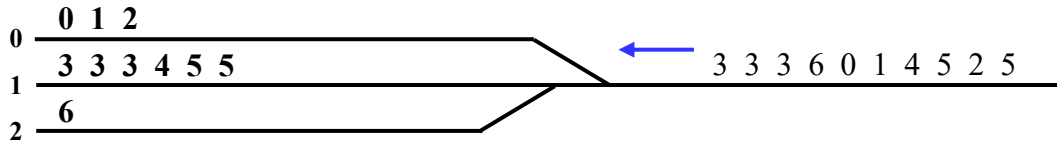


Рисунок 9.10 – Стан сортувального парку після сортування (етап 2)

У розподільчому методі на останньому етапі (збирання вагонів) усі вагони на витяжну колію витягувати немає необхідності. Так, в даному прикладі на останньому етапі достатньо витягнути вагони призначенням 6 з колії 2 та вагони призначенням 3, 4, 5 з колії 1 на витяжну, а

потім осадити їх на колію 0 (див. рисунок 9.11). При цьому логічні номери колій не змінюються на жодному етапі.

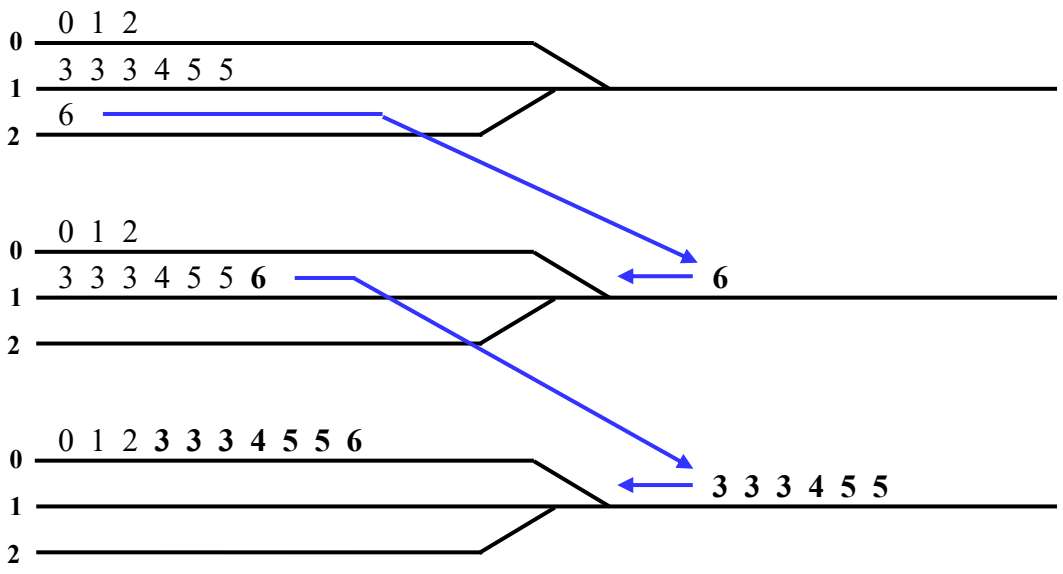


Рисунок 9.11 – Стан сортувального парку після сортування (етап 3)

Отже, процес формування багатогрупного состава розподільчим методом завершено за 3 етапи.

9.6 Умовна нумерація груп вагонів

Застосування умовної нумерації груп вагонів дозволяє виявити початкову упорядкованість состава та скоротити обсяг роботи з формування. Так, проаналізувавши состав із прикладу видно, що состав частково вже є упорядкованим, тобто одиниці знаходяться перед двійками, трійки перед четвірками, п'ятірки перед шістками. Тому, змінивши нумерацію груп на умовну, можна скоротити обсяг роботи з формування.

Присвоєння умовних номерів зручно виконувати в таблиці (див. таблицю 9.9) за таким алгоритмом:

1. Кожному вагону надати порядковий номер (0, 1, 2, 3, ...);
2. Відсортувати состав у визначеному порядку;
3. Виділити групи порядкових номерів, що зростають;
4. Присвоїти умовні номери кожній такій групі (0, 1, 2, ...).

Таблиця 9.9

Присвоєння умовних номерів

Вихідний состав	3	5	1	3	2	3	5	6	0	4
Порядкові номери	0	1	2	3	4	5	6	7	8	9
Відсортований состав	0	1	2	3	3	3	4	5	5	6
Порядкові номери	8	2	4	0	3	5	9	1	6	7
Умовні номери груп	0	1		2				3		

Виконаємо формування состава комбінаторним методом з урахуванням умовної нумерації груп.

Присвоюємо коди призначенням вагонів (див. таблицю 9.9):

Таблиця 9.9

Коди призначень вагонів при сортуванні состава комбінаторним методом з урахуванням умовної нумерації груп

Призначення вагонів	УНГ	К	Φ_3 (γ)
0	0	3	011
1, 2	1	2	010
3, 4	2	1	001
5, 6	3	0	000

Сформуємо состав комбінаторним методом (див. рисунок 9.12):

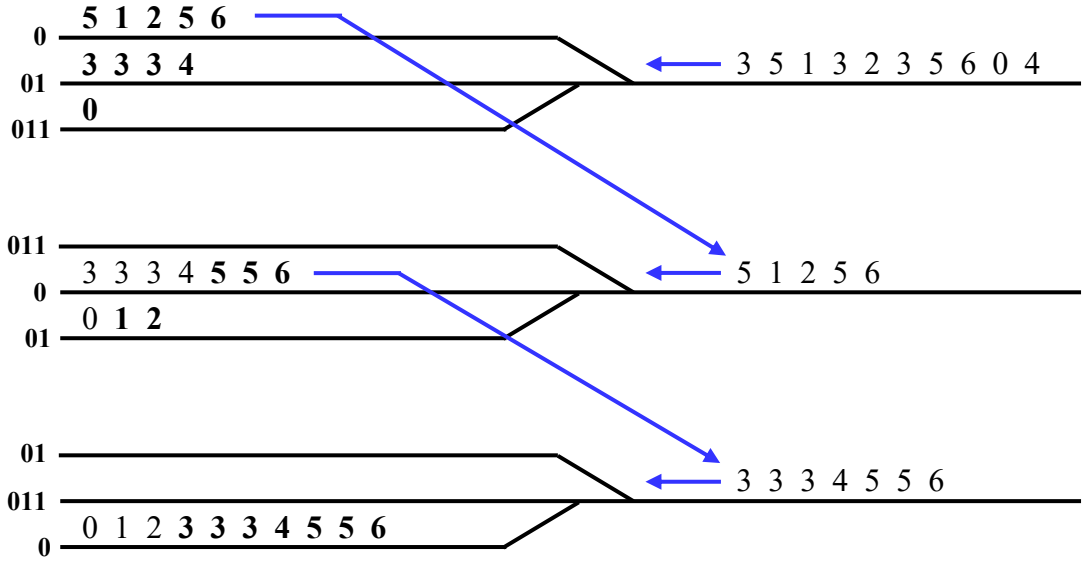


Рисунок 9.12 – Стан сортувального парку після сортування (етапи 1-3)

Таким чином, при застосуванні умовної нумерації вдалося скоротити кількість етапів формування состава з 4-х до 3-х.

Виконаємо формування состава розподільчим методом з урахуванням умовної нумерації груп.

Присвоюємо коди призначенням вагонів (див. таблицю 9.10).

Таблиця 9.10

Коди призначень вагонів при сортуванні состава розподільчим методом з врахуванням умовної нумерації груп

Призначення вагонів	УНГ	$\Psi_3(\gamma)$
0	0	000
1, 2	1	001
3, 4	2	002
5, 6	3	010

Виконуємо формування багатогрупного составу розподільчим методом (див. рисунок 9.13):

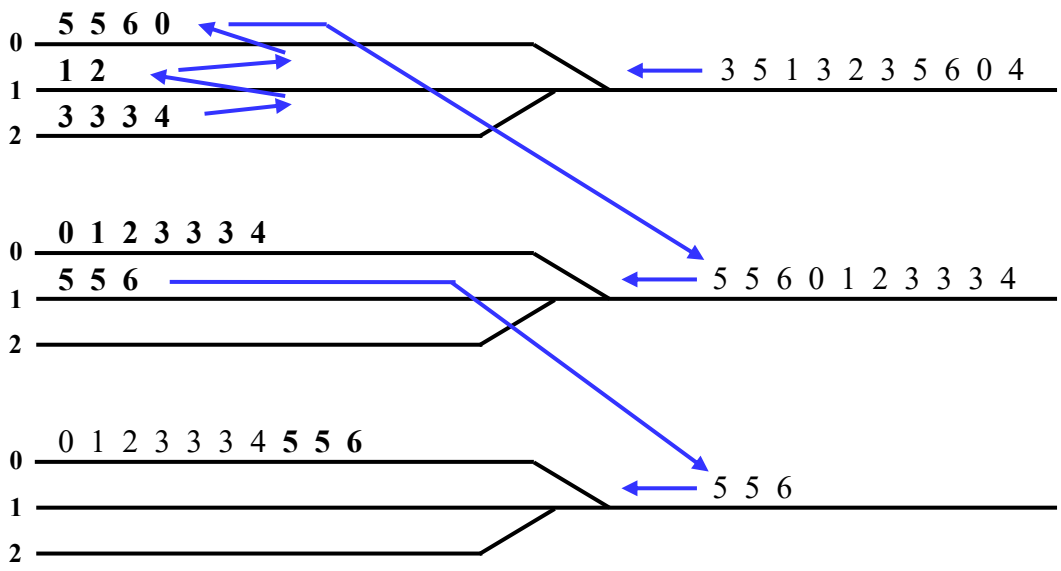


Рисунок 9.13 – Стан сортувального парку після сортування (етапи 1-3)

Як видно, на останньому етапі замість збирання вагонів з двох колій у разі використання умовної нумерації можна виконувати збирання з однієї колії.

Для аналізу звернемо увагу на принципову різницю між двома методами формування (див. таблицю 9.11).

Таблиця 9.11

Принципова різниця між комбінаторним і розподільчим методами

Метод формування	Різниця		
	Нумерація	Перенумерація після чергового етапу сортування	Витягування вагонів після чергового етапу сортування
Комбінаторний метод	0, 1, 11, 111,..	Циклічно вниз	З колії 0
Розподільчий метод	0, 1, 2, 3,..	Відсутня	З усіх колій, починаючи з колії, що має найбільший порядковий номер

Більш детальні дослідження комбінаторного та розподільчого методу показує, що комбінаторний метод має перевагу у разі малої кількості колій (2, 3) через відсутність необхідності збирання вагонів зі всіх сортувальних колій, в той час, як розподільчий метод вміє ефективно використовувати велику кількість колій.

Контрольні запитання

1. Що таке багатогрупний состав та які його принципові відмінності від одногрупного составу при розформуванні-формуванні?
2. З якою метою доцільно використовувати методи формування багатогрупних составів?
3. Які основні особливості комбінаторного методу формування багатогрупних составів?
4. Які основні особливості розподільчого методу формування багатогрупних составів?
5. Що таке система числення Фібоначчі та як саме вона використовується при формуванні багатогрупних составів?
6. Що таке позиційна система числення та як саме вона використовується при формуванні багатогрупних составів?
7. Які основні кроки алгоритму комбінаторного методу формування багатогрупних составів?

8. Які основні кроки алгоритму розподільчого методу формування багатогрупних составів?

9. З якою метою доцільно виконувати врахування початкової упорядкованості груп вагонів вихідного багатогрупного составу за рахунок визначення умовної нумерації груп?

10. Для яких технічних засобів доцільно використовувати комбінаторний та розподільчий методи формування багатогрупних составів?

РОЗДІЛ 10

Задача комівояжера

10.1 Теоретичні відомості

10.1.1 Історична довідка

Історія задачі комівояжера, відомої також під англійською назвою Traveling Salesman Problem (TSP), розпочинається в середині XIX століття, коли ірландський математик Уільям Гамільтон вигадав так звану гру «Навколосвітню подорож», мета якої відшукати такий шлях, що проходить через усі вершини (міста, пункти призначення) певного графа, щоб відвідати кожну вершину одноразово і повернутися у вихідну [18 – 20]. Дана гра стала основою ідеї майбутніх циклів Гамільтона, які отримали широке практичне застосування та приклад яких наведені на рисунку 10.1.

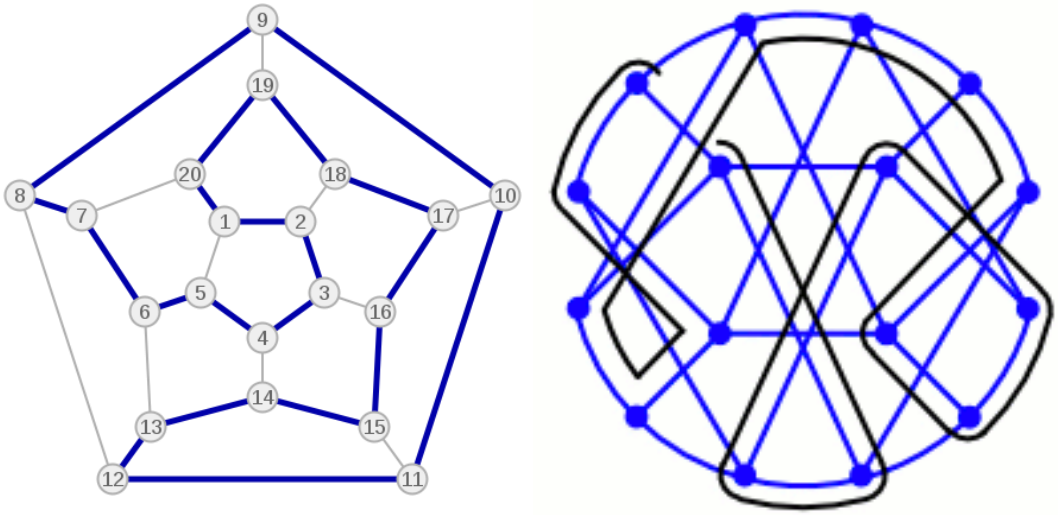


Рисунок 10.1 – Цикли Гамільтона

Теоретичне підґрунтя цієї концепції почало розвиватися в кінці XIX – на початку XX століття, хоча назва «задача комівояжера» увійшла у вжиток лише в 1930-х роках.

У вказаний період науковці, переважно із США, розглядали практично значиму логістичну проблему визначення оптимального маршруту для комерційного агента, який мав відвідати різні міста з мінімальними витратами часу та коштів [18, 19].

Поступово задача набула строгої математичної форми: маючи набір міст та відстаней між кожною парою з них, потрібно знайти найкоротший замкнений маршрут, який проходить через кожне місто рівно один раз.

У 1950-х та 1960-х роках з'явилися перші відомі методи розв'язування, серед яких вирізняється робота Джорджа Данціга, Рея Фалкерсона й Селмера Джонсона над «різанням площин», що поєднувалося з лінійним програмуванням. Цей підхід допоміг розв'язати на той час складні випадки задачі, а згодом його вдосконалювали завдяки розвитку обчислювальної техніки та появі нових алгоритмів на основі методу гілок і меж, динамічного програмування та евристичних пошуків.

Згодом задачу комівояжера визнали NP-складною, що означає відсутність відомих на сьогодні поліноміальних алгоритмів для її точного розв'язання при великих розмірах вхідних даних.

В той же час задача комівояжера залишається одним із найважливіших об'єктів досліджень в області комбінаторної оптимізації та операційного дослідження: вона не лише стимулює пошук нових методів, а й знаходить численні практичні застосування в логістиці, організації виробничих процесів, плануванні маршрутів і навіть у проєктуванні комп'ютерних мереж.

10.1.2 Постановка задачі комівояжера в загальному випадку

Задача комівояжера полягає в тому, щоб знайти найкоротший маршрут, який починатиметься з першого міста, відвідає всі інші міста строго один раз у будь-якому порядку та повернеться до початкової точки. Відстані між містами відомі. Необхідно визначити такий порядок відвідування міст, який забезпечить мінімальну загальну довжину маршруту [2, 3, 18 – 23].

Задача формулюється як проблема знаходження замкнутого маршруту, що проходить через усі точки один раз і повертається до початкової, при цьому мінімізуючи загальну вартість маршруту.

Нехай $P = \{P_1, P_2, \ldots, P_n\}$ – множина точок, а d_{ij} – відстань між точками P_i та P_j.

Цільова функція тоді матиме наступний вигляд

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} \to \min ,$$

де x_{ij} дорівнює 1, якщо маршрут проходить між i і j, і дорівнює 0 в іншому випадку.

Задача має бути розв'язана з обмеженнями, що кожна точка відвідується лише один раз, і транспорт залишає кожну точку рівно один раз:

$$\sum_{j=1, j\neq i}^{n} x_{ij} = 1, \forall i; \quad \sum_{i=1, i\neq j}^{n} x_{ij} = 1, \forall i$$

Для врахування додаткових обмежень, таких як часові вікна, вводять змінні t_i для часу прибуття до точки i, з умовою:

$$a_i \le t_i \le b_i, \forall i$$

10.1.3 Загальна характеристика методів розв'язання

Реалізація рішень задачі комівояжера в транспортній логістиці дозволяє досягти кількох важливих результатів.

По-перше, це зниження витрат на перевезення завдяки мінімізації відстаней і часу маршруту.

По-друге, підвищується точність дотримання графіків доставки, що позитивно впливає на репутацію компанії.

По-третє, зменшується екологічний вплив через скорочення використання палива, що відповідає сучасним трендам екологічної логістики.

Оптимізація маршрутів із використанням задачі комівояжера є фундаментальною складовою транспортної логістики. Різноманіття методів, як точних, так і наближених, дозволяє ефективно вирішувати задачі різних розмірностей і складності, адаптуючи їх до специфічних вимог торгівельної мережі.

Задача комівояжера є класичною проблемою оптимізації в транспортній логістиці, яка належить до NP-складних задач. Це означає, що для її вирішення за допомогою точних методів обчислювальні витрати швидко зростають із збільшенням кількості точок. Однак розроблено низку підходів, які дозволяють ефективно вирішувати задачу залежно від її розмірності та вимог.

Методи можна умовно поділити на [2, 3, 18 – 23]:

– точні, які забезпечують оптимальне рішення;

– наближені, які знаходять рішення близьке до оптимального за менший час.

Точні методи забезпечують оптимальні рішення, але підходять лише для задач із невеликою кількістю точок.

Наближені методи, навпаки, є ефективними для великих задач, але можуть вимагати додаткової оптимізації для досягнення прийнятної точності.

Вибір конкретного методу для вирішення задачі комівояжера залежить від розміру задачі, вимог до точності та обмежень у часі.

Для задач середньої розмірності часто використовуються гібридні підходи, які поєднують переваги різних методів.

Завдяки розвитку обчислювальних потужностей і програмних інструментів, сучасні методи вирішення задачі комівояжера дозволяють враховувати широкий спектр реальних обмежень і застосовуються в багатьох сферах логістики та планування [23].

10.2 Короткий опис точних методів вирішення задачі комівояжера

10.2.1 Метод меж і гілок

Метод гілок і меж [2, 3, 7, 8, 12, 18 – 23] є фундаментальним підходом до вирішення задачі комівояжера, який поєднує простоту розуміння з можливістю значного скорочення обчислювальних витрат. Він базується на ідеї побудови дерева рішень, де кожен вузол дерева відповідає певному стану задачі (наприклад, маршрут, який проходить через підмножину точок), а гілки – можливим виборам продовження цього маршруту.

Основні етапи алгоритму наступні:

<u>Етап 1. Побудова початкового дерева рішень.</u>

Початкове дерево рішень є базовою структурою, яка представляє всі можливі маршрути задачі комівояжера. У кореневій вершині дерева міститься вихідна задача, яка описує весь набір точок, що мають бути відвідані, але ще не визначено порядок їх обходу. Тобто, коренева вершина відповідає повній задачі без жодного часткового маршруту або вибраних ребер.

Етап 2. Розгалуження.

На кожному кроці алгоритму гілок і меж вибирається вершина дерева рішень, яка підлягає розгалуженню. Ця вершина відповідає певному частковому рішенню задачі, тобто маршруту, який уже містить деяку підмножину точок, що були відвідані. Розгалуження полягає в розбитті цього часткового рішення на кілька підзадач, кожна з яких представляє один із можливих виборів наступної точки для відвідування.

Етап 3. Обчислення меж.

Для кожної гілки дерева рішень обчислюється нижня межа можливого маршруту, що дозволяє оцінити, наскільки перспективним є цей варіант. Нижня межа визначається як мінімальна вартість, необхідна для завершення маршруту, включаючи вже обрані елементи. Найчастіше використовується метод редукції матриці відстаней, де мінімальні значення по рядках і стовпчиках віднімаються для отримання нових меж.

Етап 4. Відсікання гілок.

Для кожної гілки дерева рішень обчислюється нижня межа можливого маршруту, що дозволяє оцінити, наскільки перспективним є цей варіант. Нижня межа визначається як мінімальна вартість, необхідна для завершення маршруту, включаючи вже обрані елементи. Найчастіше використовується метод редукції матриці відстаней, де мінімальні значення по рядках і стовпчиках віднімаються для отримання нових меж.

Етап 5. Продовження пошуку.

Алгоритм вибирає наступну вершину для розгалуження з найнижчою межею серед усіх гілок дерева, що залишилися. Цей процес повторюється, доки всі гілки дерева рішень не будуть або розглянуті, або відсічені. У результаті вдається знайти оптимальний маршрут, що має мінімальну вартість серед усіх можливих.

Метод гілок і меж є точним і гарантує знаходження оптимального рішення. Однак, його обчислювальна складність стрімко зростає зі збільшенням кількості точок nnn, оскільки кількість можливих маршрутів дорівнює $n!$. Тому метод є ефективним для задач із кількістю точок до 30-40, залежно від структури задачі та апаратних ресурсів.

Так, наприклад, якщо в задачі є 5 точок, то дерево рішень матиме 5! = 120 можливих маршрутів. Однак, метод гілок і меж дозволяє відсікати більшість неефективних варіантів, зменшуючи кількість реальних обчислень до кількох десятків.

Метод меж і гілок є загальновідомим точним методом розв'язання задачі комівояжера, і для нього розроблено багато варіантів оптимізації.

10.2.2 Метод динамічного програмування

Метод динамічного програмування [2, 3, 7, 8, 12, 18 – 20] базується на принципі оптимальності Беллмана, що формулюється так: "Оптимальний маршрут містить у своєму складі оптимальні підмаршрути". Цей підхід дозволяє розбити вихідну задачу на підзадачі меншого розміру, розв’язання яких використовується для побудови загального оптимального рішення.

Формулювання задачі наступне:

Для множини точок $S = \{P_1, P_2, \ldots, P_n\}$ і точки $P_j \in S$ мінімальна вартість маршруту, який проходить через усі точки множини SSS і закінчується у P_j, позначається як $C(S, j)$, та обчислюється рекурентно:

$$C(S, j) = \min_{i \in S, i \neq j} [C(S \setminus \{j\}, i) + d_{ij}],$$

де d_{ij} – відстань між точками i і j, а $C(\{i\}, i) = 0$ для базового випадку.

Переваги даного методу полягають в тому, що він гарантує оптимальне рішення та може бути використаний для модифікованих задач комівояжера (наприклад, із часовими вікнами).

Недоліком методу є те, що його обчислювальна складність становить $O(n^2 \cdot 2^n)$, що робить метод обчислювально витратним для задач із кількістю точок більше 20.

Наприклад, якщо є 4 точки, то множини для підзадач матимуть розміри $2^n = 2^4 = 16$. Для кожної підмножини обчислюється мінімальна вартість маршруту, поступово будується оптимальний шлях.

10.2.3 Метод лінійного програмування

Метод лінійного програмування [7, 8, 12, 18 – 20] дає можливість розв’язувати задачу комівояжера через формулювання її у вигляді задачі оптимізації з лінійною цільовою функцією та набором

обмежень. Змінними є x_{ij} , які дорівнюють 1, якщо маршрут включає ребро $i \to j$, і 0 – в іншому випадку.

Цільова функція має наступний вигляд:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} ,$$

Обмеження для даного методу такі:

Кожна точка повинна бути відвідана один раз:

$$\sum_{j=1, j \neq i}^{n} x_{ij} = 1, \forall i$$

Кожна точка повинна залишатися кінцевою для одного маршруту:

$$\sum_{i=1, i \neq j}^{n} x_{ij} = 1, \forall j$$

Забороняються цикли, які не включають усі точки:

$$\sum_{i, j \in S} x_{ij} \leq |S| - 1, \forall S \subset P, |S| < n .$$

Розв’язання задачі здійснюється за допомогою симплекс-методу або інших чисельних підходів.

Переваги цього методу полягають в його універсальності (він може враховувати часові вікна, обмеження на вантажопідйомність тощо) та може використовуватись для модифікованих варіантів задачі комівояжера.

Недоліком можна відзначити те, що його обчислювальна складність швидко зростає з розміром задачі.

10.2.4 Алгоритм Літтла

Алгоритм Літтла [7, 8, 12, 18 – 20] є спеціалізованим методом для задачі комівояжера, що використовує матричне подання відстаней між точками.

Основний принцип полягає у поступовому зменшенні розмірності матриці через виключення гілок, які не можуть бути частиною оптимального маршруту.

Етапи алгоритму:

1. Редукція рядків і стовпців матриці шляхом віднімання мінімального елемента кожного рядка і стовпчика.

2. Побудова дерева рішень, де кожна вершина відповідає вибору певної гілки.

3. Виключення гілок із високою вартістю на кожному етапі, які не можуть бути частиною оптимального маршруту.

Алгоритм Літтла ефективний для задач із невеликою кількістю точок (до 10-15), оскільки швидко визначає оптимальний маршрут. Він є основою для багатьох інших методів і використовується як базовий підхід у дослідженнях задачі комівояжера.

10.3 Короткий опис наближених методів вирішення задачі комівояжера

Наближені методи є практичним рішенням для задачі комівояжера, особливо коли її розмірність значно зростає, і точні методи стають обчислювально недоцільними [9, 10, 12, 18 – 20, 23]. Ці методи забезпечують швидке знаходження рішень, які, хоча і не гарантують оптимальності, є близькими до неї. У транспортній логістиці вони широко використовуються для великих мереж, де потрібна оперативна оптимізація маршрутів.

10.3.1 Метод найближчого сусіда

Метод найближчого сусіда [9, 10, 12, 18 – 20] є найпростішим евристичним підходом, який будує маршрут, вибираючи наступну точку на основі мінімальної відстані від поточної.

Алгоритм починається зі стартової точки, яка може бути обрана випадковим чином або відповідно до заданого критерію (наприклад, найближчої до складу).

З цієї точки метод послідовно додає до маршруту найближчу невідвідану точку. Формально вибір наступної точки j описується виразом:

$$j = \arg\min_{k \in U} d_{ik},$$

де U – множина невідвіданих точок;

i – поточна точка;

d_{ik} – відстань між точками i та k.

Цей процес триває, поки всі точки не будуть включені до маршруту, після чого маршрут завершується поверненням до стартової точки.

Сила методу полягає в його швидкості й простоті. Він здатний обробляти великі графи за дуже короткий час, що робить його корисним у ситуаціях, де час є критично важливим фактором. Наприклад, у разі термінової доставки товарів метод дозволяє швидко скласти приблизний маршрут, який одразу можна використовувати.

Однак метод має значний недолік – він часто генерує неефективні маршрути через те, що зосереджується лише на локальній оптимальності, не враховуючи глобальної структури задачі.

Наприклад, після включення близької точки маршрут може "віддалити" транспортний засіб від наступних точок, збільшуючи загальну довжину.

10.3.2 Метод парних заміщень

Метод парних заміщень [9, 10, 18 – 20, 23] є вдосконаленням початкового маршруту шляхом поступового обміну порядку відвідування двох точок у маршруті. Спочатку будується базовий маршрут, наприклад, за допомогою методу найближчого сусіда або випадкової перестановки точок. Далі метод починає перевіряти всі можливі заміщення пар точок у маршруті. Якщо після обміну парами довжина маршруту зменшується, зміни зберігаються, і алгоритм продовжує роботу.

$$L_{\text{новий}} = L_{\text{поточний}} - (d_{i-1,i} + d_{j,j+1}) + (d_{i-1,j} + d_{i,j+1}),$$

де i та j – точки, що обмінюються;

$L_{\text{поточний}}$ – початкова довжина маршруту.

Цей метод дозволяє поступово покращувати початкове рішення, уникаючи деяких помилок локальної оптимізації. Його перевагою є здатність знайти значно коротший маршрут порівняно з базовим методом, особливо якщо базовий маршрут має недоліки. Однак метод може застрягти в локальному мінімумі, тобто в рішенні, яке неможливо покращити лише за допомогою парних замін, але яке все ще далеке від глобального оптимуму.

У транспортній логістиці цей метод може використовуватися для швидкої адаптації існуючого маршруту до нових умов, наприклад, додавання нової торгівельної точки або зміни розташування однієї з точок.

10.3.3 Метод випадкового пошуку

Метод випадкового пошуку [9, 10, 18 – 20, 23] передбачає створення великої кількості маршрутів шляхом випадкового перемішування точок, після чого обирається найкращий із них за заданим критерієм, наприклад, мінімальною довжиною маршруту. Цей метод є простим у реалізації і часто використовується як початковий етап для інших алгоритмів, наприклад, генетичних або методу парних заміщень.

Випадковий пошук може бути ефективним для задач із невеликими або середніми графами, але його точність значно залежить від кількості згенерованих маршрутів. Він також не враховує жодних принципів оптимізації, що робить його менш придатним для складних логістичних задач.

10.3.4 Табу-пошук

Табу-пошук [9, 10, 18 – 20, 23] є одним із методів локальної оптимізації, який уникає застрягання в локальних мінімумах шляхом введення списку заборонених дій (таблиці табу).

Алгоритм починається зі створення початкового маршруту і поступово покращує його шляхом локальних змін (наприклад, обміну двох точок).

Якщо зміна не призводить до покращення, вона може бути прийнята, якщо її немає у списку табу.

Список табу оновлюється з кожним кроком, що дозволяє уникати повторного відвідування неефективних рішень.

Оцінка нового маршруту:

$$f(R_{\text{новий}}) = \text{довжина}(R_{\text{новий}}), R_{\text{новий}} \in / \text{ таблиця табу.}$$

Цей метод добре підходить для задач із великою кількістю точок, оскільки він комбінує швидкість локальних методів із можливістю виходу за межі локальних мінімумів.

Проте для досягнення високої ефективності потрібно ретельно налаштовувати параметри, такі як розмір списку табу.

10.3.5 Локальний пошук (Local Search)

Метод локального пошуку [9, 10, 18 – 20, 23] спрямований на покращення існуючого маршруту шляхом локальних змін. Це може бути перестановка точок, зміна напрямку маршруту або видалення/додавання точок.

Оцінка зміни маршруту виконується за допомогою даного виразу:

$$\Delta L = L_{\text{новий}} - L_{\text{поточний}},$$

де $L_{\text{новий}}$ – довжина маршруту після зміни;

$L_{\text{поточний}}$ – довжина початкового маршруту.

Алгоритм працює ітеративно, поступово покращуючи маршрут. Якщо знайдене рішення краще за попереднє, воно приймається як новий початковий маршрут.

Цей метод є основою для багатьох інших наближених алгоритмів, таких як табу-пошук або метод парних заміщень. Він добре працює для невеликих графів, але може застрягати в локальних мінімумів.

10.3.6 Метод інсерції (метод вставки)

Метод інсерції [9, 10, 18 – 20] побудований на ітеративному формуванні маршруту шляхом поступового додавання точок до існуючого часткового маршруту.

Початково вибирається невеликий базовий маршрут (наприклад, дві або три точки), після чого до нього додаються нові точки так, щоб мінімізувати приріст загальної довжини маршруту.

Новий приріст обчислюється як:

$$\Delta L = d_{i,j} + d_{j,k} - d_{i,k},$$

де i та k – сусідні точки, між якими вставляється точка j.

Точки можуть додаватися в будь-яку позицію в маршруті залежно від критерію мінімізації, що дозволяє враховувати як географічну близькість, так і інші фактори.

Перевагою цього методу є його швидкість і можливість отримати якісні результати для задач із середньою кількістю точок.

Проте метод може бути менш ефективним для великих задач, де часткові рішення часто стають локально оптимальними.

10.3.7 Метод мінімального остовного дерева (MST-based heuristic)

Даний метод ґрунтується на побудові мінімального остовного дерева (MST – Minimum Spanning Tree) для графа, що представляє задачу комівояжера [9, 10, 18 – 20].

Остовне дерево – це підграф, який з'єднує всі точки (вузли) графа з мінімальною можливою сумарною вагою (довжиною ребер).

Алгоритм будує мінімальне остовне дерево і використовує його як основу для побудови приблизного маршруту.

$$MST = \arg\min \sum_{(i,j)\in T} d_{ij},$$

де T – остовне дерево;

d_{ij} – відстань між точками.

Цей метод дуже швидкий і дає гарні результати для задач із рівномірно розподіленими точками, але він не завжди враховує особливості задачі, такі як часові вікна чи обмеження на вантажопідйомність.

10.3.8 Генетичні алгоритми

Генетичні алгоритми [9, 10, 18 – 20, 23] – це метаевристичний підхід, натхненний процесом природного відбору. Вони працюють із популяцією можливих рішень (маршрутів), які еволюціонують через послідовні покоління.

Кожен маршрут оцінюється за допомогою функції пристосованості, яка визначає, наскільки добре він виконує задачу мінімізації довжини маршруту.

Кожен маршрут оцінюється за функцією пристосованості:

$$f(R) = \frac{1}{L(R)},$$

де R – маршрут;

$L(R)$ – загальна довжина маршруту.

Нові маршрути створюються за допомогою кросовера, наприклад:

$$R_{\text{новий}} = R_1[1:k] + R_2[k+1:n],$$

де R_1, R_2 – батьківські маршрути;

k – точка розділення.

Найкращі маршрути мають більшу ймовірність бути обраними для створення нових рішень за допомогою операцій кросовера (комбінування частин двох маршрутів) і мутації (випадкові зміни в маршруті).

Генетичні алгоритми є надзвичайно гнучкими й добре справляються з великими задачами.

Вони дозволяють уникати застрягання в локальних мінімумах завдяки варіативності, яку забезпечують мутації.

Наприклад, якщо базові методи створили неефективний маршрут, генетичний алгоритм може випадково модифікувати його таким чином, що відкриються нові, більш оптимальні варіанти.

Однак цей підхід вимагає значних обчислювальних ресурсів для обробки великої кількості маршрутів у кожному поколінні.

У транспортній логістиці він часто застосовується для складних систем із багатьма обмеженнями, такими як часові вікна, вантажопідйомність транспорту тощо.

10.3.9 Алгоритм мурашиної колонії

Алгоритм мурашиної колонії [9, 10, 18 – 20, 23] заснований на моделюванні поведінки реальних мурах, які прокладають шляхи до джерел їжі, залишаючи феромонні мітки.

У задачі комівояжера мурахи представляють можливі маршрути, а феромони – значення, які вказують на ефективність цих маршрутів.

На початку всі ребра графа мають однаковий рівень феромонів, але з кожним кроком маршрути, які мінімізують довжину, отримують більше феромонів, що підвищує ймовірність їх вибору іншими "мурахами".

Імовірність вибору ребра $i \to j$ визначається формулою:

$$P_{ij} = \frac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{k \in U} \tau_{ik}^{\alpha} \cdot \eta_{ik}^{\beta}}$$

де τ_{ij} – рівень феромону;

$\eta_{ij} = \dfrac{1}{d_{ij}}$ – інверсія відстані;

α і β – вагові коефіцієнти;

U – множина невідвіданих точок.

Цей підхід є ітеративним: на кожній ітерації мурахи створюють маршрути, які потім оцінюються, і феромони оновлюються відповідно до якості маршрутів. З часом алгоритм сходиться до одного або кількох найкращих маршрутів.

Алгоритм мурашиної колонії особливо корисний для великих задач, де потрібно знайти якісне рішення за обмежений час. Він є гнучким і легко адаптується до специфічних вимог задачі, таких як багатовимірні обмеження.

Однак алгоритм може бути повільним на початкових етапах через необхідність навчання системи.

10.3.10 Метод імітаційного відпалу

Метод імітації відпалу [9, 10, 18 – 20, 23] моделює процес охолодження металів, де рішення покращуються через поступове зменшення ймовірності прийняття гірших рішень.

Імовірність прийняття такого рішення задається формулою:

$$P = e^{-\Delta E/T}$$

де ΔE – приріст довжини маршруту;

T – поточна "температура".

У задачі комівояжера це означає, що алгоритм починає з випадкового маршруту й поступово покращує його шляхом невеликих змін.

На кожному кроці метод може прийняти гірше рішення (збільшення довжини маршруту) з певною ймовірністю, яка зменшується в процесі "охолодження". Це дозволяє уникати локальних мінімумів і знаходити більш оптимальні рішення.

Метод імітаційного відпалу є ефективним для задач середньої розмірності, де необхідно швидко знайти рішення з високою якістю. Його недоліком є залежність результату від параметрів алгоритму, таких як початкова температура, швидкість охолодження тощо.

10.3.11 Гібридні алгоритми

Гібридні алгоритми [9, 10, 18 – 20, 23] об’єднують у собі особливості кількох методів для досягнення більш збалансованого результату між точністю і швидкістю обчислень. Наприклад, для створення початкового рішення може використовуватися метод найближчого сусіда,

після чого застосовується генетичний алгоритм для його вдосконалення. Інший варіант – поєднання алгоритму мурашиної колонії з локальним пошуком, що дозволяє швидко знайти наближений оптимальний маршрут і доопрацювати його.

Гібридні алгоритми є особливо корисними для великих задач із багатьма обмеженнями, оскільки вони комбінують швидкість наближених методів із здатністю враховувати складні умови, наприклад, часові вікна чи вантажопідйомність транспорту.

Ці наближені методи, хоч і демонструють на 10% (і більше) гірший раціональний маршрут, ніж точні методи, проте дозволяють вирішувати задачі комівояжера у випадках, коли точні методи недоступні через їхню складність. Завдяки їх гнучкості вони стали незамінними в транспортній логістиці, плануванні маршрутів та інших сферах.

10.4. Приклад розв'язання задач. Задача 12

Умова задачі

На контейнерний майданчик для вивантаження прибуває вагон (або група вагонів) з контейнерами. Кожний контейнер (1, 2, 3) необхідно вивантажити на відповідну секцію складу (4, 5, 6), (рисунок 10.2).

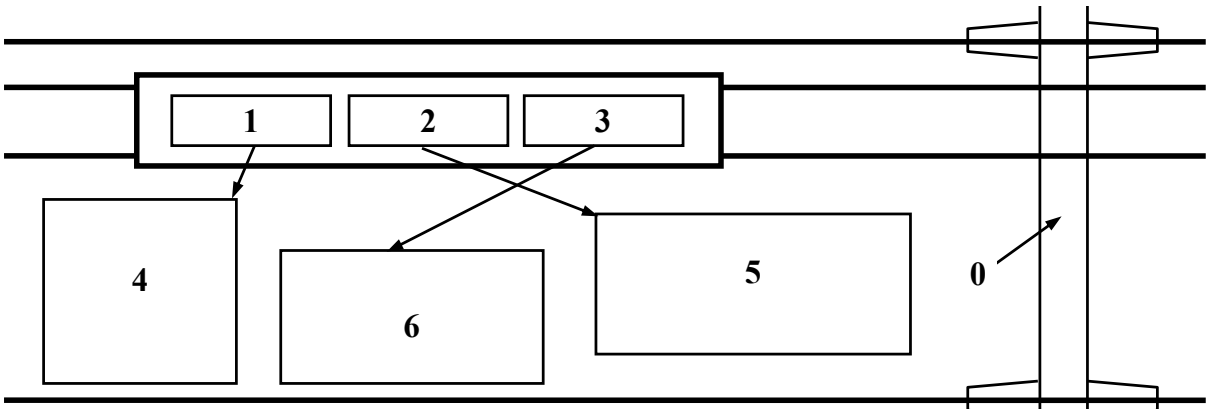


Рисунок 10.2 – Схема розвантаження контейнерів на складі (*вихідна*)

Необхідно визначити оптимальну черговість вивантаження, яка забезпечить мінімальну тривалість вантажних операцій, якщо в початковий момент кран розташований у положенні 0.

Вихідні дані

Тривалості пробігу крана наведені в таблиці 10.1.

Таблиця 10.1

Тривалості пробігу крану від точки до точки

Від точки	До точки						
	0	1	2	3	4	5	6
0	0	4,2	3,4	2,6	-	-	-
1	-	-	-	-	1,7	-	-
2	-	-	-	-	-	2,3	-
3	-	-	-	-	-	-	2,9
4	0	-	1,1	2,3	-	-	-
5	0	2,6	-	0,8	-	-	-
6	0	1,8	1,3	-	-	-	-

Розв’язання задачі

Тривалість виконання операцій вивантаження контейнерів однакова для всіх варіантів, тому достатньо оптимізувати напрямок повернення крана за черговим контейнером після вивантаження попереднього. Операції руху візка крана до контейнера та його переміщення до місця розташування на складі поєднані в одну таблицю (див. таблицю 10.2).

Таблиця 10.2

Тривалість руху візка крана до контейнера та його переміщення до місця розташування на складі

Від точки	До точки (через точку)			
	0	4 (1→4)	5 (2→5)	6 (3→6)
0	-	5,9	5,7	5,5
4	0	-	3,4	5,2
5	0	4,3	-	3,7
6	0	2,9	3,6	-

Візок крана з положення 0 забирає контейнер № 1 та вивантажує його на місце № 4, тривалість перевантаження складає

$$T_{0\to4} = t_{0\to1} + t_{1\to4} = 4{,}2 + 1{,}7 = 5{,}9 \text{ хв.}$$

Повернення після вивантаження з місця № 5 за контейнером № 1 і переміщення його на місце вивантаження № 4 триває

$$T_{5\to4} = t_{5\to1} + t_{1\to4} = 2{,}6 + 1{,}7 = 4{,}3 \text{ хв.}$$

Повернення після вивантаження з місця № 8 за контейнером № 1 і переміщення його на місце вивантаження № 4 триває

$$T_{6\to4} = t_{6\to1} + t_{1\to4} = 1{,}2 + 1{,}7 = 2{,}9 \text{ хв.}$$

Інші значення у таблиці 10.2 заповнимо аналогічно.

Задача комівояжера розв'язується **методом гілок та меж**.

При цьому перед черговою ітерацією визначається верхня межа розв'язку *B*, тобто найкраще рішення, яке відоме на даний момент.

Далі виконується спрямований перебір варіантів, на кожному кроці якого визначається нижня межа розв'язку *H* – тобто найменше можливе значення показника при прийнятому на даному кроці порядку обходу вершин.

Важливо:

Якщо на будь-якому кроці нижня межа перебільшить верхню, то подальший перебір варіантів припиняється та виконується аналіз альтернативних розв'язків.

Якщо отримано допустимий розв'язок, то змінюється верхня межа та виконується аналіз альтернативних рішень.

Метод гілок та меж реалізовується наступним чином:

1. Скласти довільний план переміщення контейнерів, наприклад:

$$(0–6) \to (6–5) \to (5–4) \to (4–0).$$

Тривалість повернення крана за контейнерами становитиме

$$T = 5{,}5 + 3{,}6 + 4{,}3 + 0 = 13{,}4 \textit{ хв}.$$

Отриманий результат являє собою верхню межу $B = 13{,}4$ *хв*.

2. Виконати редукцію рядків та стовпців матриці.

Для цього в кожному стовпці та в кожному рядку необхідно визначити мінімальний елемент (див. таблицю 10.3):

Таблиця 10.3

Визначення початку редукції

Від точки	До точки (через точку)				
	0	4 (1→4)	5 (2→5)	6 (3→6)	
0	-	5,9	5,7	5,5	**5,5**
4	0	-	3,4	5,2	**0**
5	0	4,3	-	3,7	**0**
6	0	2,9	3,6	-	**0**
	0	**2,9**	**3,4**	**3,7**	

Редукцію необхідно розпочинати зі стовпців у випадку, якщо сума мінімальних елементів стовпців більша за суму мінімальних елементів рядків. У іншому випадку редукцію необхідно розпочинати з рядків.

У даному випадку сума мінімальних елементів стовпців складає:

$$Q_j^* = 0 + 2{,}9 + 3{,}4 + 3{,}7 = 10{,}0 \text{ хв.}$$

Сума усіх мінімальних елементів рядків складає:

$$C_i^* = 5{,}5 + 0 + 0 + 0 = 5{,}5 \text{ хв.}$$

Оскільки сума мінімальних елементів стовпців (10,0) більша за суму мінімальних елементів рядків (5,5), то редукцію будемо виконувати починаючи зі стовпців.

Для цього в кожному стовпці слід визначити мінімальний елемент та відняти його від усіх елементів стовпця (див. таблицю 10.4).

Таблиця 10.4

Редукція матриці по стовпцях (*етап 1*)

Від точки	До точки			
	0	4	5	6
0	–	5,9	5,7	5,5
4	0	–	3,4	5,2
5	0	4,3	–	3,7
6	0	2,9	3,6	–
$\boldsymbol{Q_j}$	**0**	**2,9**	**3,4**	**3,7**

Після цього аналогічна операція виконується з рядками (див. таблицю 10.5).

Таблиця 10.5

Редукція матриці по рядках (*етап 1*)

Від точки	До точки				C_i
	0	4	5	6	
0	–	3,0	2,3	1,8	**1,8**
4	0	–	0	1,5	**0**
5	0	1,4	–	0	**0**
6	0	0	0,2	–	**0**
Q_j	**0**	**2,9**	**3,4**	**3,7**	

3. Визначити величину нижньої межі H, для цього необхідно знайти суму мінімальних елементів, за допомогою яких виконувалася редукція матриці, для даного прикладу

$$H = 0 + 2{,}9 + 3{,}4 + 3{,}7 + 1{,}8 + 0 + 0 + 0 = 11{,}8 \text{ хв.}$$

Таким чином, тривалість вивантаження не може бути меншою за $H = 11{,}8$ хв.

На дереві пошуку зображується початкова вершина (корінь дерева), на якій позначають нижню та верхню межі (див. рисунок 10.3).

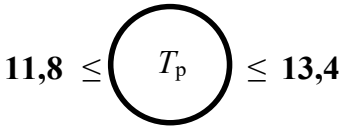


Рисунок 10.3 – Дерево пошуку після визначення нижньої та верхньої межі

4. Додати в маршрут одне ребро.

Згідно з алгоритмом методу гілок та меж оптимальне рішення з великою ймовірністю (втім не 100-відсотковою) передбачає включення у граф такого ребра, виключення якого максимально збільшить тривалість процесу вивантаження контейнерів.

Для визначення цього ребра необхідно для всіх нульових елементів матриці розрахувати штрафи Φ_{ij}, які представляють собою додатковий пробіг крану при невключенні того чи іншого ребра; вказані штрафи,

в свою чергу, залежать від альтернативних мінімальним (наступних за мінімальним) значень i-го рядка A_i та j-го стовпчика B_j.

Результати розрахунків альтернативних мінімальним значень наведені в таблиці 10.6.

Таблиця 10.6

Визначення альтернативних елементів (*етап 1*)

Від точки	До точки				C_i	A_i
	0	4	5	6		
0	–	1,2	0,5	0	**1,8**	0,5
4	0	–	0	1,5	**0**	0
5	0	1,4	–	0	**0**	0
6	0	0	0,2	–	**0**	0
Q_j	**0**	**2,9**	**3,4**	**3,7**	**11,8**	
B_j	0	1,2	0,2	0		

Штраф Φ_{ij}, який викликаний невключенням ребра ij, знаходимо за формулою:

$$\Phi_{ij} = A_i + B_j, \tag{10.1}$$

де A_i – альтернатива в рядку;
B_j – альтернатива в стовпці.
Розрахунок штрафів:

$\Phi_{0-6} = 0{,}5 + 0 = 0{,}5$; $\Phi_{4-0} = 0 + 0 = 0$; $\Phi_{4-5} = 0 + 0{,}2 = 0{,}2$;

$\Phi_{5-0} = 0 + 0 = 0$; $\Phi_{5-6} = 0 + 0 = 0$; $\Phi_{6-0} = 0 + 0 = 0$;

$\underline{\Phi_{6-4} = 0 + 1{,}2 = 1{,}2.}$

На графі покажемо дві гілки вирішення задачі (див. рисунок 10.4).

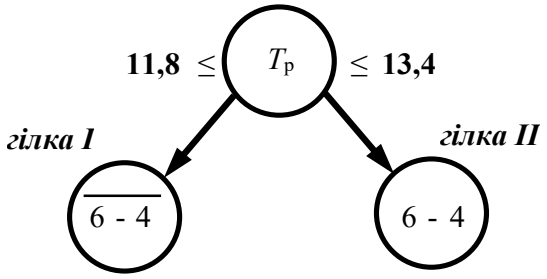


Рисунок 10.4 – Дерево пошуку з врахування ребра (6–4)

Одна (гілка *I*) відповідає невключенню ребра до розв'язку, а інша (гілка *II*) – включенню ребра 6–4 до розв'язку.

Так, при невключенні ребра (6–4) у розв'язок із матриці таблиці 10.6 необхідно лише ребру (6–4) замість «0» поставимо «–» (див. таблицю 10.7).

Таблиця 10.7

Невключення ребра 6–4 до розв'язку

Від точки	До точки			
	0	4	5	6
0	–	1,2	0,5	0
4	0	–	0	1,5
5	0	1,4	–	0
6	0	–	0,2	–

Для того, щоб знайти нижню межу розв'язку для гілки *I* необхідно до попередньої нижньої межі (H = 11,8 *хв*) додати максимальний штраф (Φ_{6-4} = 1,2 *хв*).

Тоді нижня межа для гілки *I* складе:

$$H^I = 11{,}8 + 1{,}2 = 13{,}0 \text{ хв}$$

Наведемо отримане значення на графі (див. рисунок 10.5):

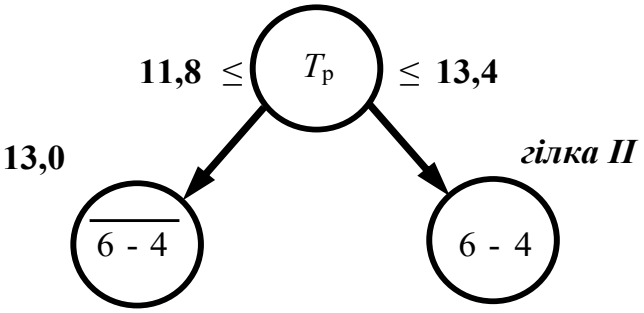


Рисунок 10.5 – Визначення нижньої межі для гілки *I*

При включенні ребра 6–4 до розв'язку із таблиці 10.6 необхідно видалити рядок № 6 та стовпець № 4 (див. таблицю 10.8).

Таблиця 10.8

Включення ребра 6–4 до розв'язку (крок 1)

Від точки	До точки			
	0	4	5	6
0	–	1,2	0,5	0
4	0	–	0	–
5	0	1,4	–	0
6	0	0	0,2	–

В результаті цього розміри матриці зменшаться з 4х4 до 3х3 (див. таблицю 10.9).

Таблиця 10.9

Включення ребра 6–4 до розв'язку (крок 2)

Від точки	До точки		
	0	5	6
0	–	0,5	0
4	0	0	1,5
5	0	–	0

Крім того, в матриці з таблиці 10.9, слід поставити прочерк («–») на ребрі або ребрах, що спричинюють зустрічний пробіг. Отже, ребру (4–6), зворотному до включеного до розв'язку ребра (6–4) (див. таблицю 10.9), замість «1,5» поставимо «–» (див. таблицю 10.10).

Таблиця 10.10

Включення ребра 6–4 до розв'язку (крок 3)

Від точки	До точки		
	0	5	6
0	–	0,5	0
4	0	0	–
5	0	–	0

Для того, щоб знайти нижню межу розв'язку для гілки *II* необхідно виконати редукцію матриці таблиці 10.10, в результаті отримаємо таблицю 10.11.

Далі визначимо суму мінімальних елементів усіх рядків та стовпців нової матриці.

Таблиця 10.11

Редукція матриці та визначення альтернативних елементів (*етап 2*)

Від точки	До точки			C_i	A_i
	0	5	6		
0	–	0,5	0	**0**	0,5
4	0	0	–	**0**	0
5	0	–	0	**0**	0
Q_j	**0**	**0**	**0**	**0**	
B_j	0	0,5	0		

У даному випадку сума мінімальних елементів як рядків (C_i) так і стовпців (Q_j) нової матриці дорівнює 0.

Отже нова нижня межа для гілки *II* складає

$$H^{II} = 11{,}8 + 0 = 11{,}8 \text{ хв.}$$

Отриману нову нижню межу наносимо на граф (див. рисунок 10.6).

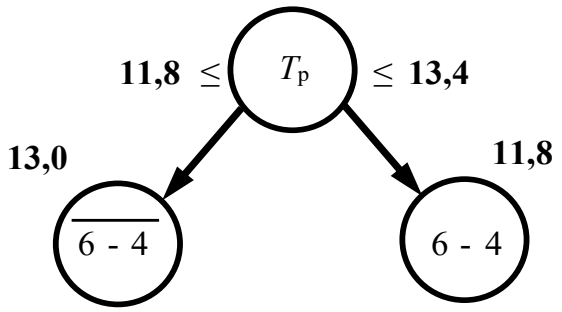


Рисунок 10.6 – Визначення нижньої межі для гілки *II*

Знову розраховуємо штрафи для визначення ще одного ребра, яке слід додати до розв'язку:

$\Phi_{0-6} = 0{,}5 + 0 = 0{,}5;$ $\Phi_{4-0} = 0 + 0 = 0;$ $\Phi_{4-5} = 0 + 0{,}5 = 0{,}5;$

$\Phi_{5-0} = 0 + 0 = 0;$ $\Phi_{5-6} = 0 + 0 = 0.$

Максимальний штраф складає 0,5 при виключенні з матриці ребра (0–6) або ребра (4–5).

Обираємо будь-яке з них, наприклад, ребро 0–6 і продовжуємо вирішення (див. таблицю 10.12).

Отримане нове значення нижньої межі розв’язку покажемо на графі (див. рисунок 10.7):

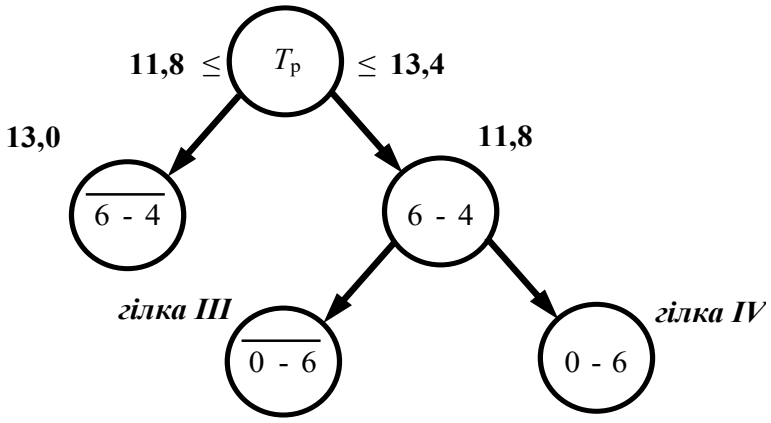


Рисунок 10.7 – Дерево пошуку після врахування ребра (0–6)

Для того, щоб знайти нижню межу розв’язку для гілки *III* необхідно до попередньої нижньої межі (H^{II} = 11,8 *хв*) додати максимальний штраф ($Ф_{0–6}$ = 0,5 *хв*).

Таким чином, нижня межа для гілки *III* складе

$$H^{III} = 11{,}8 + 0{,}5 = 12{,}3 \text{ хв}$$

Наведемо отримане значення на графі (див. рисунок 10.8):

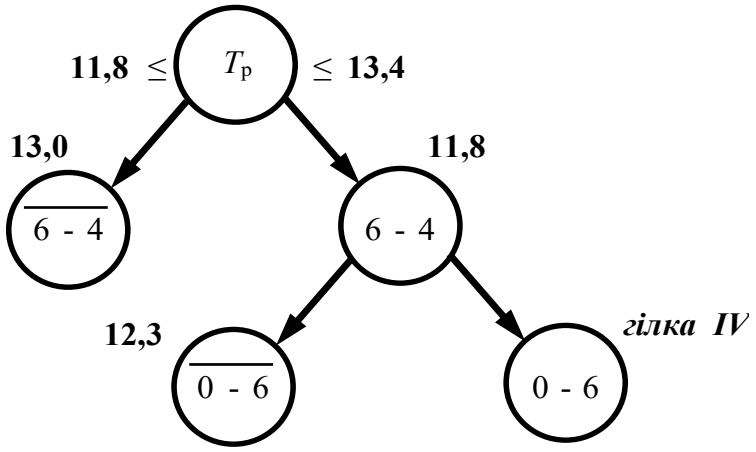


Рисунок 10.8 – Визначення нижньої межі для гілки *III*

Для того, щоб знайти нижню межу розв’язку для гілки *IV* необхідно виконати редукцію матриці у таблиці 10.8 та визначити суму мінімальних елементів рядків та стовпців (див. таблицю 10.9).

Таблиця 10.12

Включення ребра 0–6 до розв'язку (крок 1)

Від точки	До точки			C_i	A_i
	0	5	6		
~~0~~	~~–~~	~~0,5~~	~~0~~	~~**0**~~	~~0,5~~
4	0	0	–	**0**	0
5	0	–	0	**0**	0
Q_j	**0**	**0**	**0**	**0**	
B_j	0	0,5	0		

Таблиця 10.13

Включення ребра 0–6 до розв'язку (крок 2)

Від точки	До точки		C_i
	0	5	
4	-	0	**0**
5	0	-	**0**
Q_j	**0**	**0**	**0**

Для ребра (4–0) таблиці 10.13 поставлено прочерк («–») з таких міркувань.

До маршруту на даний момент додано два ребра: (6–4) та (0–6), які складають ланцюг: 0 – 6 – 4.

Ребра, які можуть спричинити зустрічний пробіг, наступні: (6–0), (4–6) та (4–0). Саме тому для ребра (4–0) в таблиці 10.13 замість «0» поставлено «–».

В таблиці 10.13 сума мінімальних елементів як рядків (C_i) так і стовпців (Q_j) нової матриці дорівнює 0.

Отже нова нижня межа для гілки *IV* складає

$$H^{IV} = 11{,}8 + 0 = 11{,}8 \text{ хв.}$$

Отриману нову нижню межу наносимо на граф (див. рисунок 10.9).

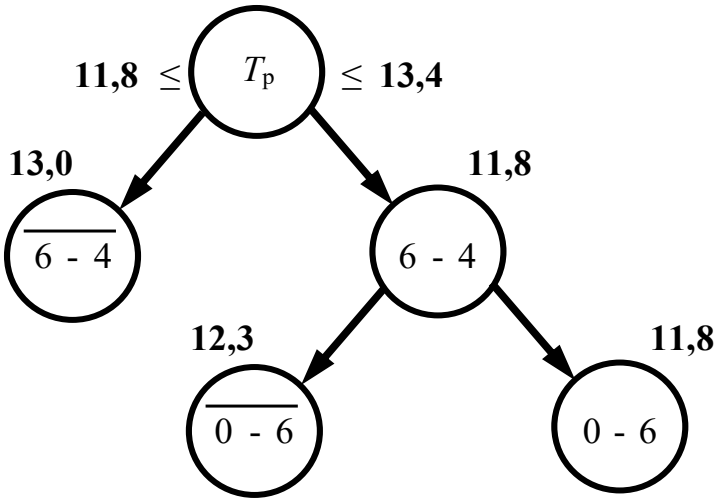


Рисунок 10.9 – Визначення нижньої межі для гілки *IV*

В отриманій матриці 2х2 виконувати редукцію немає сенсу, ребра (4–5) та (5–0) додаються до дерева розв'язку. Дані ребра є безальтернативними, а тому для варіантів розв'язку, які їх не містять, нижня межа становитиме «∞».

Усі ребра, які включено до рішення, повинні складати замкнений маршрут комівояжера, для даного прикладу:

$$(0–6) \rightarrow (6–4) \rightarrow (4–5) \rightarrow (5–0).$$

Оскільки отримано новий розв'язок, який кращий, ніж знайдений на першому кроці розв'язок (11,8 < 13,4), тому змінюємо верхню межу (B) з 13,4 *хв* на 11,8 *хв*.

Після цього порівнюємо усі знайдені нижні межі з новою верхньою межею. У випадку, якщо будь-яка нижня межа більше, ніж нова верхня межа (B^* = 11,8 *хв*), то така гілка дерева повністю виключається з розв'язку.

В умовах даного прикладу всі альтернативні варіанти передбачають час більший ніж B^* = 11,8 *хв*, тому всі ці альтернативні варіанти ігноруються, а отриманий основний варіант розвантаження контейнерів з транспортного засобу буде оптимальним (див. рисунок 10.10).

Таким чином, оптимальний маршрут руху візку крана такий:

$$0 \rightarrow 6 \rightarrow 4 \rightarrow 5 \rightarrow 0.$$

При цьому, тривалість розвантаження контейнерів з залізничних платформ на майданчик складе $T_р$ = 11,8 *хв*.

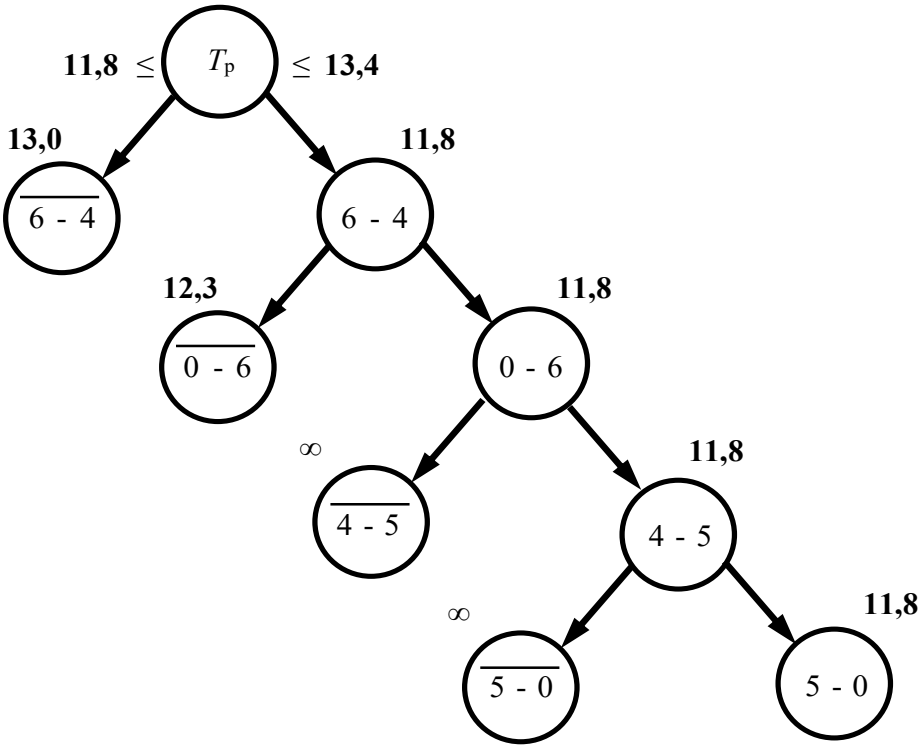


Рисунок 10.10 – Дерево пошуку розв’язку з усіма варіантами руху візка крана

Оптимальна послідовність вивантаження контейнерів козловим краном на визначені місця складу за даними рисунку 10.10 наведена на рисунку 10.11.

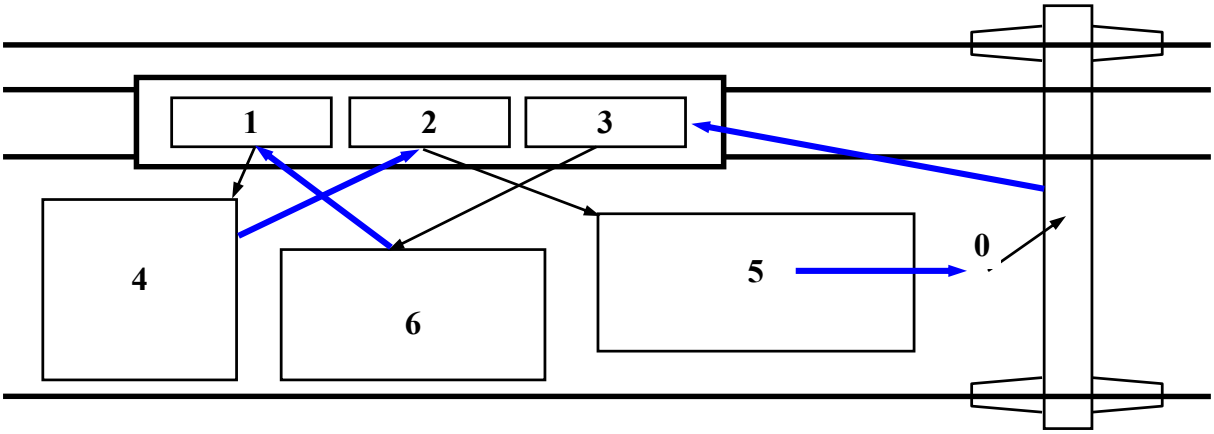


Рисунок 10.11 – Схема розвантаження контейнерів на складі (*оптимальна*)

На рисунку 10.11 тонкими лініями зі стрілками показані задані маршрути руху козлового крану, а товстими лініями зі стрілками показані визначені в результаті вирішення задачі комівояжера маршрути руху цього крану.

10.5. Приклад розв’язання задач. Задача 13

Умова задачі

Деяке автотранспортне підприємство обслуговує окремі (1–7) пункти (магазини, поштові відділення) міста у певній черговості. Найближчим часом планується розширення початкової мережі на 3 нові пункти (8–10).

Початкова транспортна мережа із 7-ми пунктів та 3 нові пункти наведені на рисунку 10.12.

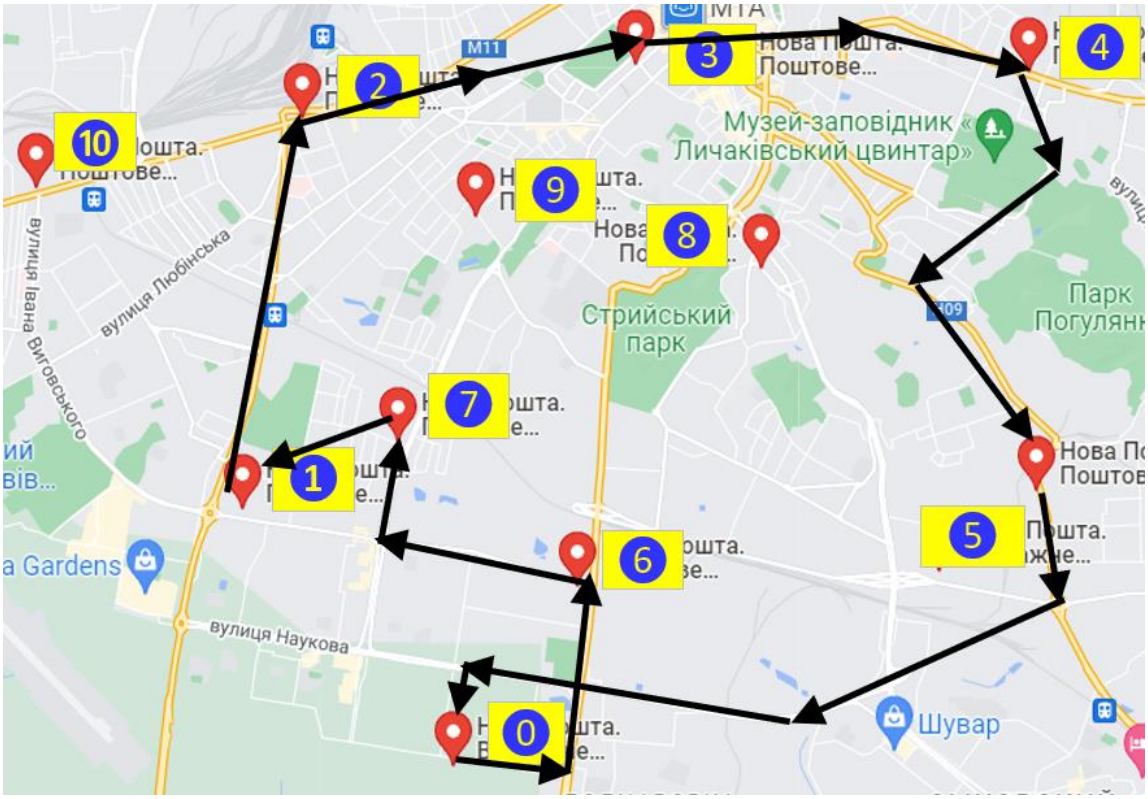


Рисунок 10.12 – Початкова транспортна мережа із 7-ми пунктів та 3 нові пункти

Необхідно визначити раціональну черговість обслуговування розширеної мережі із 10-ти точок.

Вихідні дані

Відстані між точками розширеної торгової мережі наведені в таблиці 10.14.

Таблиця 10.14

Матриця відстаней між точками розширеної торгової мережі

i \ *j*	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	∞	3,3	∞	∞	10,7	6,4	2,3	2,8	6,7	∞	7,2
1	3,3	∞	3,4	∞	∞	8,8	3,2	2,3	∞	4,6	3,0
2	∞	3,0	∞	2,9	∞	∞	∞	3,5	∞	2,2	2,0
3	∞	∞	2,9	∞	3,6	6,1	4,9	6,7	2,1	1,8	4,2
4	10,7	∞	∞	3,6	∞	5,1	∞	6,4	3,7	5,1	∞
5	6,4	8,8	∞	6,1	5,1	∞	4,8	6,9	4,0	∞	∞
6	2,5	3,8	∞	4,0	∞	5,3	∞	2,9	3,3	3,2	∞
7	2,8	2,1	4,3	6,7	6,4	7,7	2,3	∞	4,4	2,2	4,6
8	6,7	∞	∞	2,1	3,7	3,9	3,9	4,1	∞	2,4	6,1
9	∞	4,6	2,2	1,7	5,1	∞	4,0	2,2	2,4	∞	3,9
10	7,2	3,0	2,0	4,2	∞	∞	∞	4,1	5,6	4,2	∞

Розв’язання задачі

Визначимо раціональну черговість обслуговування розширеної мережі із 10-ти точок двома методами:

1. Методом найближчого сусіда (наближений метод);
2. Методом гілок та меж (точний метод).

Метод 1. Визначення раціонального маршруту доставки вантажів методом найближчого сусіда

Визначимо за допомогою методу найближчого сусіда точку мережі до якої слід рухатись з точки 0 у таблиці 10.15.

Таблиця 10.15

Пошук точки призначення на 1-му етапі

i \ *j*	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	$L_{\min}$	**До**
0	–	3,3	∞	∞	10,7	6,4	2,3	2,8	6,7	∞	7,2	2,3	6

Таким чином, на даному етапі відвідані наступні пункти:

$$0 \rightarrow \mathbf{6}$$

Визначимо точку мережі до якої слід рухатись з точки 6 у таблиці 10.16.

Таблиця 10.16

Пошук точки призначення на 2-му етапі

i \ *j*	0	1	2	3	4	5	6	7	8	9	10	L_{min}	До
6	–	3,8	∞	4,0	∞	5,3	–	2,9	3,3	3,2	∞	2,9	7

Таким чином, на даному етапі відвідані наступні пункти:

$$0 \rightarrow 6 \rightarrow \mathbf{7}$$

Визначимо точку мережі до якої слід рухатись з точки 7 у таблиці 10.17.

Таблиця 10.17

Пошук точки призначення на 3-му етапі

i \ *j*	0	1	2	3	4	5	6	7	8	9	10	L_{min}	До
7	–	2,1	4,3	6,7	6,4	7,7	–	–	4,4	2,2	4,6	2,1	1

Таким чином, на даному етапі відвідані наступні пункти:

$$0 \rightarrow 6 \rightarrow 7 \rightarrow \mathbf{1}$$

Визначимо точку мережі до якої слід рухатись з точки 1 у таблиці 10.18.

Таблиця 10.18

Пошук точки призначення на 3-му етапі

i \ *j*	0	1	2	3	4	5	6	7	8	9	10	L_{min}	До
1	–	–	3,4	∞	∞	8,8	–	–	∞	4,6	3,0	3,0	10

Таким чином, на даному етапі відвідані наступні пункти:

$$0 \rightarrow 6 \rightarrow 7 \rightarrow 1 \rightarrow \mathbf{10}$$

Визначимо точку мережі до якої слід рухатись з точки 10 у таблиці 10.19.

Таблиця 10.19

Пошук точки призначення на 4-му етапі

i \ *j*	0	1	2	3	4	5	6	7	8	9	10	L_{min}	До
10	–	–	2,0	4,2	∞	∞	–	–	5,6	4,2	–	2,0	2

Таким чином, на даному етапі відвідані наступні пункти:

$$0 \to 6 \to 7 \to 1 \to 10 \to \mathbf{2}$$

Визначимо точку мережі до якої слід рухатись з точки 2 у таблиці 10.20.

Таблиця 10.20

Пошук точки призначення на 5-му етапі

i \ *j*	0	1	2	3	4	5	6	7	8	9	10	L_{min}	До
2	–	–	–	2,9	∞	∞	–	–	∞	2,2	–	2,2	9

Таким чином, на даному етапі відвідані наступні пункти:

$$0 \to 6 \to 7 \to 1 \to 10 \to 2 \to \mathbf{9}$$

Визначимо точку мережі до якої слід рухатись з точки 9 у таблиці 10.21.

Таблиця 10.21

Пошук точки призначення на 6-му етапі

i \ *j*	0	1	2	3	4	5	6	7	8	9	10	L_{min}	До
9	–	–	–	1,7	5,1	∞	–	–	2,4	–	–	1,7	3

Таким чином, на даному етапі відвідані наступні пункти:

$$0 \to 6 \to 7 \to 1 \to 10 \to 2 \to 9 \to \mathbf{3}$$

Визначимо точку мережі до якої слід рухатись з точки 3 у таблиці 10.22.

Таблиця 10.22

Пошук точки призначення на 7-му етапі

i \ *j*	0	1	2	3	4	5	6	7	8	9	10	L_{min}	До
3	–	–	–	–	3,6	6,1	–	–	2,1	–	–	2,1	8

Таким чином, на даному етапі відвідані наступні пункти:

$$0 \to 6 \to 7 \to 1 \to 10 \to 2 \to 9 \to 3 \to \mathbf{8}$$

Визначимо точку мережі до якої слід рухатись з точки 8 у таблиці 10.23.

Таблиця 10.23

Пошук точки призначення на 8-му етапі

i \ *j*	0	1	2	3	4	5	6	7	8	9	10	L_{min}	До
8	–	–	–	–	3,7	3,9	–	–	–	–	–	3,7	4

Таким чином, на даному етапі відвідані наступні пункти:

$$0 \to 6 \to 7 \to 1 \to 10 \to 2 \to 9 \to 3 \to 8 \to \mathbf{4}$$

Визначимо точку мережі до якої слід рухатись з точки 4 у таблиці 10.24.

Таблиця 10.24

Пошук точки призначення на 9-му етапі

i \ *j*	0	1	2	3	4	5	6	7	8	9	10	L_{min}	До
4	–	–	–	–	–	5,1	–	–	–	–	–	5.1	5

Таким чином, на даному етапі відвідані наступні пункти:

$$0 \to 6 \to 7 \to 1 \to 10 \to 2 \to 9 \to 3 \to 8 \to 4 \to \mathbf{5}$$

На останньому етапі додаємо точку 0, з якої починали рух по даному маршруту.

$$0 \to 6 \to 7 \to 1 \to 10 \to 2 \to 9 \to 3 \to 8 \to 4 \to 5 \to \mathbf{0}$$

Використовуючи матрицю відстаней між роздрібними точками заданої торгової мережі (див. таблицю 10.14) визначимо загальну довжину маршруту їх обслуговування:

$$L_м = 2{,}3 + 2{,}9 + 2{,}1 + 3{,}0 + 2{,}0 + 2{,}2 + 1{,}7 + 2{,}1 + 3{,}7 + 5{,}1 + 6{,}4 = 33{,}5 \text{ км}$$

Таким чином, за допомогою методу найближчого сусіда було визначено раціональний маршрут обслуговування роздрібних точок торгової мережі, довжина якого склала 33,5 км.

У той же час, як було зазначено вище, метод найближчого сусіда відноситься до групи наближених методів і може давати не найбільш точний результат.

Метод 2. Визначення оптимального маршруту доставки вантажів методом гілок та меж

Використовуючи дані таблиці 10.14, визначимо мінімальні відстані по кожному рядку та стовпчику у таблиці 10.25.

Приймаємо отриманий раніше раціональний маршрут за методом найближчого сусіда за початковий маршрут розвезення вантажів до роздрібних точок заданої торгової мережі:

$$0 \to 6 \to 7 \to 1 \to 10 \to 2 \to 9 \to 3 \to 8 \to 4 \to 5 \to 0$$

Загальна довжина цього маршруту становитиме:

$$L_м = 2{,}3 + 2{,}9 + 2{,}1 + 3{,}0 + 2{,}0 + 2{,}2 + 1{,}7 + 2{,}1 + 3{,}7 + 5{,}1 + 6{,}4 = 33{,}5 \text{ км}$$

Даний результат є верхньою межею $B = L_м = 33{,}5$ км.

Таблиця 10.25

Матриця відстаней між точками розширеної торгової мережі

i \ *j*	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	d_i
0	∞	3,3	∞	∞	10,7	6,4	2,3	2,8	6,7	∞	7,2	2,3
1	3,3	∞	3,4	∞	∞	8,8	3,2	2,3	∞	4,6	3,0	2,3
2	∞	3,0	∞	2,9	∞	∞	∞	3,5	∞	2,2	2,0	2,0
3	∞	∞	2,9	∞	3,6	6,1	4,9	6,7	2,1	1,8	4,2	1,8
4	10,7	∞	∞	3,6	∞	5,1	∞	6,4	3,7	5,1	∞	3,6
5	6,4	8,8	∞	6,1	5,1	∞	4,8	6,9	4,0	∞	∞	4,0
6	2,5	3,8	∞	4,0	∞	5,3	∞	2,9	3,3	3,2	∞	2,5
7	2,8	2,1	4,3	6,7	6,4	7,7	2,3	∞	4,4	2,2	4,6	2,1
8	6,7	∞	∞	2,1	3,7	3,9	3,9	4,1	∞	2,4	6,1	2,1
9	∞	4,6	2,2	1,7	5,1	∞	4,0	2,2	2,4	∞	3,9	1,7
10	7,2	3,0	2,0	4,2	∞	∞	∞	4,1	5,6	4,2	∞	2,0
d_j	2,5	2,1	2,0	1,7	3,6	3,9	2,3	2,2	2,1	1,8	2,0	26,4 / 26,2

Для того, щоб визначити нижню межу множини потрібно виконати редукцію рядків та стовпців. Оскільки сума мінімальних елементів по стовпцях більша ніж сума мінімальних елементів у рядках, то розпочнемо виконувати редукцію зі стовпців.

Матриця з мінімальними елементами за рядками подана в таблиці 10.26.

Таблиця 10.26

Матриця відстаней з мінімальними елементами по рядках

i \ *j*	0	1	2	3	4	5	6	7	8	9	10
0	–	1,0	–	–	8,4	4,1	0,0	0,5	4,4	–	4,9
1	1,0	–	1,1	–	–	6,5	0,9	0,0	–	2,3	0,7
2	–	1,0	–	0,9	–	–	–	1,5	–	0,2	0,0
3	–	–	1,1	–	1,8	4,3	3,1	4,9	0,3	0,0	2,4
4	7,1	–	–	0,0	–	1,5	–	2,8	0,1	1,5	–
5	2,4	4,8	–	2,1	1,1	–	0,8	2,9	0,0	–	–
6	0,0	1,3	–	1,5	–	2,8	–	0,4	0,8	0,7	–
7	0,7	0,0	2,2	4,6	4,3	5,6	0,2	–	2,3	0,1	2,5
8	4,6	–	–	0,0	1,6	1,8	1,8	2,0	–	0,3	4,0
9	–	2,9	0,5	0,0	3,4	–	2,3	0,5	0,7	–	2,2
10	5,2	1,0	0,0	2,2	–	–	–	2,1	3,6	2,2	–
d_j	**0**	**0**	**0**	**0**	**1,1**	**1,5**	**0**	**0**	**0**	**0**	**0**

Аналогічно редукції по рядках виконуємо дану операцію по стовпцях, після чого записуємо результат зміненої матриці у таблицю 10.27.

Вказуємо мінімальні елементи рядків матриці, та продублюємо мінімальні елементи по стовпцях із таблиці 10.26.

Нижню межу розв’язку *H* визначимо за формулою (3.1):

$$H = 1{,}1+1{,}5+2{,}3+2{,}3+2{,}0+1{,}8+3{,}6+4{,}0+2{,}5+2{,}1+2{,}1+1{,}7+2{,}0 = 29{,}0 \text{ км}$$

Таблиця 10.27

Матриця відстаней після редукції по стовпцях

i \ *j*	0	1	2	3	4	5	6	7	8	9	10	d_{ij}
0	–	1,0	–	–	7,3	2,6	0,0	0,5	4,4	–	4,9	2,3
1	1,0	–	1,1	–	–	5,0	0,9	0,0	–	2,3	0,7	2,3
2	–	1,0	–	0,9	–	–	–	1,5	–	0,2	0,0	2,0
3	–	–	1,1	–	1,8	4,3	3,1	4,9	0,3	0,0	2,4	1,8
4	7,1	–	–	0,0	–	0,0	–	2,8	0,1	1,5	–	3,6
5	2,4	4,8	–	2,1	0,0	–	0,8	2,9	0,0	–	–	4,0
6	0,0	1,3	–	1,5	–	1,3	–	0,4	0,8	0,7	–	2,5
7	0,7	0,0	2,2	4,6	4,3	5,6	0,2	–	2,3	0,1	2,5	2,1
8	4,6	–	–	0,0	0,5	0,3	1,8	2,0	–	0,3	4,0	2,1
9	–	2,9	0,5	0,0	2,3	–	2,3	0,5	0,7	–	2,2	1,7
10	5,2	1,0	0,0	2,2	–	–	–	2,1	3,6	2,2	–	2,0
d_j	0,0	0,0	0,0	0,0	1,1	1,5	0,0	0,0	0,0	0,0	0,0	**29,0**

Отже, довжина маршруту доставки вантажів до роздрібних точок торгової мережі не повинна бути меншою за 29,0 *км*.

Визначаємо перше ребро, яке необхідно буде додати до маршруту за величиною штрафу. Так, знайдемо в таблиці 10.28 альтернативні елементи по рядках A_i й стовпцях B_j.

Таблиця 10.28

Приведена матриця з альтернативними до мінімальних елементів

$i \backslash j$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	d_i	A_i
0	–	1,0	–	–	7,3	2,6	**0**	0,5	4,4	–	4,9	2,3	0,5
1	1,0	–	1,1	–	–	5,0	0,9	**0**	–	2,3	0,7	2,3	0,7
2	–	1,0	–	0,9	–	–	–	1,5	–	0,2	**0**	2,0	0,2
3	–	–	1,1	–	0,7	2,8	3,1	4,9	0,3	**0**	2,4	1,8	0,3
4	7,1	–	–	**0**	–	**0**	–	2,8	0,1	1,5	–	3,6	0
5	2,4	4,8	–	2,1	**0**	–	0,8	2,9	**0**	–	–	4,0	0
6	**0**	1,3	–	1,5	–	1,3	–	0,4	0,8	0,7	–	2,5	0,4
7	0,7	**0**	2,2	4,6	3,2	4,1	0,2	–	2,3	0,1	2,5	2,1	0,1
8	4,6	–	–	**0**	0,5	0,3	1,8	2,0	–	0,3	4	2,1	0,3
9	–	2,9	0,5	**0**	2,3	–	2,3	0,5	0,7	–	2,2	1,7	0,5
10	5,2	1,0	**0**	2,2	–	–	–	2,1	3,6	2,2	–	2,0	1,0
d_j	0	0	0	0	1,1	1,5	0	0	0	0	0	29,0	
B_j	0,7	1,0	0,5	0	0,5	0,3	0,2	0,4	0,1	0,1	0,7		

За формулою (10.1) саме для нульових елементів матриці визначимо додаткові пробіги за умови виключення цього ребра із маршруту:

$\Phi_{0\text{-}6} = 0{,}5 + 0{,}2 = 0{,}7$; $\Phi_{3\text{-}9} = 0{,}3 + 0{,}1 = 0{,}4$; $\Phi_{5\text{-}4} = 0 + 0{,}5 = 0{,}5$;
$\Phi_{7\text{-}1} = 0{,}1 + 1 = 1{,}1$; $\Phi_{1\text{-}7} = 0{,}7 + 0{,}4 = 1{,}1$; $\Phi_{4\text{-}3} = 0 + 0 = 0$;
$\Phi_{5\text{-}8} = 0 + 0{,}1 = 0{,}1$; $\Phi_{8\text{-}3} = 0{,}3 + 0 = 0{,}3$; $\Phi_{2\text{-}10} = 0{,}2 + 0{,}7 = 0{,}9$;
$\Phi_{4\text{-}5} = 0 + 0{,}3 = 0{,}3$; $\Phi_{6\text{-}0} = 0{,}4 + 0{,}7 = 1{,}1$; $\Phi_{9\text{-}3} = 0{,}5 + 0 = 0{,}5$;
$\underline{\Phi_{10\text{-}2} = 1{,}0 + 0{,}5 = 1{,}5}$.

Найбільший додатковий пробіг складе $\Phi_{10\text{-}2} = 1{,}5$ при виключенні ребра (10–2) з маршруту доставки вантажу. Отже, множину розбиваємо на дві підмножини (10–2) та $(\overline{10-2})$. Розраховуємо нижню межу множини рішення, яка у разі виключення ребра (10–2) становить $H^1 = 29{,}0 + 1{,}5 = 30{,}5$ *хв*. Включення ребра (10–2) до множини рішення здійснюється шляхом викреслювання всіх елементів 10-го рядка та 2-го стовпця. Крім того слід замінити елемент $d_{2\text{-}10}$ на символ ∞ або «–», щоб уникнути утворення зустрічного пробігу.

У таблиці 10.29 показано процес включення ребра (10–2).

Таблиця 10.29

Операція включення ребра (10–2) до множини рішення

i \ *j*	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	–	1,0	–	–	7,3	2,6	0	0,5	4,4	–	4,9
1	1,0	–	1,1	–	–	5,0	0,9	0	–	2,3	0,7
2	–	1,0	–	0,9	–	–	–	1,5	–	0,2	–
3	–	–	1,1	–	0,7	2,8	3,1	4,9	0,3	0	2,4
4	7,1	–	–	0	–	0	–	2,8	0,1	1,5	–
5	2,4	4,8	–	2,1	0	–	0,8	2,9	0	–	–
6	0	1,3	–	1,5	–	1,3	–	0,4	0,8	0,7	–
7	0,7	0	2,2	4,6	3,2	4,1	0,2	–	2,3	0,1	2,5
8	4,6	–	–	0	0,5	0,3	1,8	2,0	–	0,3	4,0
9	–	2,9	0,5	0	2,3	–	2,3	0,5	0,7	–	2,2
~~**10**~~	~~5,2~~	~~1,0~~	~~0~~	~~2,2~~				~~2,1~~	~~3,6~~	~~2,2~~	

У результаті включення ребра до множини рішення, отримаємо матрицю, розміром 10 на 10, для якої слід виконати операцію приведення.

У таблиці 10.30 наведена матриця відстаней після операції приведення з фіксацією мінімумів.

Таблиця 10.30

Скорочена матриця відстаней після приведення матриці

i \ *j*	**0**	**1**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	d_i
0	–	1,0	–	7,3	2,6	0	0,5	4,4	–	4,2	0
1	1,0	–	–	–	5,0	0,9	0	–	2,3	0	0
2	–	1,0	0,9	–	–	–	1,5	–	0,2	–	0,2
3	–	–	–	0,7	2,8	3,1	4,9	0,3	0	1,7	0
4	7,1	–	0	–	0	–	2,8	0,1	1,5	–	0
5	2,4	4,8	2,1	0	–	0,8	2,9	0	–	–	0
6	0	1,3	1,5	–	1,3	–	0,4	0,8	0,7	–	0
7	0,7	0	4,6	3,2	4,1	0,2	–	2,3	0,1	1,8	0
8	4,6	–	0	0,5	0,3	1,8	2,0	–	0,3	3,3	0
9	–	2,9	0	2,3	–	2,3	0,5	0,7	–	1,5	0
d_j	0	0	0	0	0	0	0	0	0	0,7	**0,9**

Нижня межа підмножини з включенням ребра (10–2) за (3.3) рівна:

$$H^{II} = 29{,}0 + 0{,}9 = 29{,}9 \text{ хв}$$

Інші розрахунки здійснюють аналогічно. У таблицях 10.31 – 10.46 наведені подальші етапи вирішення задачі комівояжера.

Таблиця 10.31

Матриця після редукції після викреслювання ребра (10–2)

$i \backslash j$	**0**	**1**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	d_i	A_i
0	–	1,0	–	7,3	2,6	0	0,5	4,4	–	4,2	0	0,5
1	1,0	–	–	–	5,0	0,9	0	–	2,3	0	0	0
2	–	0,8	0,7	–	–	–	1,3	–	0	–	0,2	0,7
3	–	–	–	0,7	2,8	3,1	4,9	0,3	0	1,7	0	0,3
4	7,1	–	0	–	0	–	2,8	0,1	1,5	–	0	0
5	2,4	4,8	2,1	0	–	0,8	2,9	0	–	–	0	0
6	0	1,3	1,5	–	1,3	–	0,4	0,8	0,7	–	0	0,4
7	0,7	0	4,6	3,2	4,1	0,2	–	2,3	0,1	1,8	0	0,1
8	4,6	–	0	0,5	0,3	1,8	2,0	–	0,3	3,3	0	0,3
9	–	2,9	0	2,3	–	2,3	0,5	0,7	–	1,5	0	0,5
d_j	0	0	0	0	0	0	0	0	0	0,7	**0,9**	
B_j	0,7	0,8	0	0,5	0,3	0,2	0,4	0,1	0	1,5		

Визначимо штрафи для всіх нульових елементів таблиці 10.31:

$\Phi_{0\text{-}6} = 0{,}5 + 0{,}2 = 0{,}7$; $\Phi_{2\text{-}9} = 0{,}7 + 0 = 0{,}7$; $\Phi_{4\text{-}5} = 0 + 0{,}3 = 0{,}3$;

$\Phi_{6\text{-}0} = 0{,}4 + 0{,}7 = 1{,}1$; $\Phi_{1\text{-}7} = 0 + 0{,}4 = 0{,}4$; $\Phi_{3\text{-}9} = 0{,}3 + 0 = 0{,}3$;

$\Phi_{5\text{-}4} = 0 + 0{,}5 = 0{,}5$; $\Phi_{7\text{-}1} = 0{,}1 + 0{,}8 = 0{,}9$; $\underline{\Phi_{1\text{-}10} = 0 + 1{,}5 = 1{,}5;}$

$\Phi_{4\text{-}3} = 0 + 0 = 0$; $\Phi_{5\text{-}8} = 0 + 0{,}1 = 0{,}1$; $\Phi_{8\text{-}3} = 0{,}3 + 0 = 0{,}3$;

$\Phi_{9\text{-}3} = 0{,}5 + 0 = 0{,}5$.

Нижня межа множини рішення при виключенні ребра (1–10) становитиме:

$$H^{III} = 29{,}9 + 1{,}5 = 31{,}4 \textit{ хв}$$

Матриця відстаней при виключенні ребра (1–10) наведена в таблиці 10.32.

Таблиця 10.32

Матриця відстаней при виключенні ребра (1–10)

i \ *j*	**0**	**1**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
0	–	1	–	7,3	2,6	0	0,5	4,4	–	4,2
1	1	–	–	–	5	0,9	0	–	2,3	0
2	–	–	0,7	–	–	–	1,3	–	0	–
3	–	–	–	0,7	2,8	3,1	4,9	0,3	0	1,7
4	7,1	–	0	–	0	–	2,8	0,1	1,5	–
5	2,4	4,8	2,1	0	–	0,8	2,9	0	–	–
6	0	1,3	1,5	–	1,3	–	0,4	0,8	0,7	–
7	0,7	0	4,6	3,2	4,1	0,2	–	2,3	0,1	1,8
8	4,6	–	0	0,5	0,3	1,8	2	–	0,3	3,3
9	–	2,9	0	2,3	–	2,3	0,5	0,7	–	1,5

Матриця після редукції при виключенні ребра (1–10) подана в таблиці 10.33.

Таблиця 10.33

Матриця після редукції при виключенні ребра (1–10)

i \ *j*	**0**	**1**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	d_i	A_i
0	–	1,0	–	7,3	2,6	0	0,1	4,4	–	0	0,1
2	–	–	0,7	–	–	–	0,9	–	0	0	0,7
3	–	–	–	0,7	2,8	3,1	4,9	0,3	0	0	0,3
4	7,1	–	0	–	0	–	2,4	0,1	1,5	0	0
5	2,4	4,8	2,1	0	–	0,8	2,5	0	–	0	0
6	0	1,3	1,5	–	1,3	–	0	0,8	0,7	0	0
7	0,7	0	4,6	3,2	4,1	0,2	–	2,3	0,1	0	0,1
8	4,6	–	0	0,5	0,3	1,8	1,6	–	0,3	0	0,3
9	–	2,9	0	2,3	–	2,3	0,1	0,7	–	0	0,1
d_j	0	0	0	0	0	0	0,4	0	0	**0,4**	
B_j	0,7	1,0	0	0,5	0,3	0,3	0,1	0,1	0		

Згідно таблиці 10.33 нижня межа при виключенні ребра (1–10) складе:

$$H^{IV} = 29{,}9 + 0{,}4 = 30{,}3\ хв$$

Визначимо штраф для всіх нульових елементів таблиці 10.33:

$\Phi_{0\text{-}6} = 0{,}1 + 0{,}3 = 0{,}4$; $\Phi_{4\text{-}3} = 0 + 0 = 0$; $\Phi_{5\text{-}8} = 0 + 0{,}1 = 0{,}1$;
$\underline{\Phi_{7\text{-}1} = 0{,}1 + 1 = 1{,}1;}$ $\Phi_{2\text{-}9} = 0{,}7 + 0 = 0{,}7$; $\Phi_{4\text{-}5} = 0 + 0{,}3 = 0{,}3$;
$\Phi_{6\text{-}0} = 0 + 0{,}7 = 0{,}7$; $\Phi_{8\text{-}3} = 0{,}3 + 0 = 0{,}3$; $\Phi_{3\text{-}9} = 0{,}3 + 0 = 0{,}4$;
$\Phi_{5\text{-}4} = 0 + 0{,}5 = 0{,}5$; $\Phi_{6\text{-}7} = 0 + 0{,}1 = 0{,}1$; $\Phi_{9\text{-}3} = 0{,}1 + 0 = 0{,}1$.

Нижня межа «дерева рішення» при виключенні ребра (7–1) буде:

$$H^{V} = 30{,}3 + 1{,}1 = 31{,}4\ хв$$

В таблиці 10.34 показане визначення нижньої межі при виключенні ребра (7–1).

Таблиця 10.34

Матриця після редукції з викреслюванням ребра (7–1)

i \ *j*	0	1	3	4	5	6	7	8	9
0	–	1	–	7,3	2,6	0	0,1	4,4	–
2	–	–	0,7	–	–	–	–	–	0
3	–	–	–	0,7	2,8	3,1	4,9	0,3	0
4	7,1	–	0	–	0	–	2,4	0,1	1,5
5	2,4	4,8	2,1	0	–	0,8	2,5	0	–
6	0	1,3	1,5	–	1,3	–	0	0,8	0,7
~~**7**~~	~~0,7~~	~~0~~	~~4,6~~	~~3,2~~	~~4,1~~	~~0,2~~	~~–~~	~~2,3~~	~~0,1~~
8	4,6	–	0	0,5	0,3	1,8	1,6	–	0,3
9	–	2,9	0	2,3	–	2,3	0,1	0,7	–

Таблиця 10.35

Матриця після включення ребра (7–1)

i \ *j*	0	3	4	5	6	7	8	9	d_i	A_i
0	–	–	7,3	2,6	0	0,1	4,4	–	0	0,1
2	–	0,7	–	–	–	–	–	0	0	0,7
3	–	–	0,7	2,8	3,1	4,9	0,3	0	0	0,3
4	7,1	0	–	0	–	2,4	0,1	1,5	0	0
5	2,4	2,1	0	–	0,8	2,5	0	–	0	0
6	0	1,5	–	1,3	–	0	0,8	0,7	0	0
8	4,6	0	0,5	0,3	1,8	1,6	–	0,3	0	0,3
9	–	0	2,3	–	2,3	0,1	0,7	–	0	0,1
d_j	0	0	0	0	0	0	0	0	**0**	
B_j	2,4	0	0,5	0,3	0,8	0,1	0,1	0		

Нижня межа підмножини при виключенні ребра (7–1) становить:

$$H^{VI} = 30{,}3 + 0 = 30{,}3\ хв$$

Визначимо штрафи для всіх нульових елементів таблиці 10.35:

$\Phi_{0\text{-}6} = 0{,}1 + 0{,}8 = 0{,}9$; $\Phi_{4\text{-}3} = 0 + 0 = 0$; $\Phi_{5\text{-}8} = 0 + 0{,}1 = 0{,}1$;
$\Phi_{8\text{-}3} = 0{,}3 + 0 = 0{,}3$; $\Phi_{2\text{-}9} = 0{,}7 + 0 = 0{,}7$; $\Phi_{4\text{-}5} = 0 + 0{,}3 = 0{,}3$;
$\Phi_{6\text{-}0} = 0 + 2{,}4 = 2{,}4$; $\Phi_{9\text{-}3} = 0{,}1 + 0 = 0{,}1$; $\Phi_{3\text{-}9} = 0{,}3 + 0 = 0{,}3$;
$\Phi_{5\text{-}4} = 0 + 0{,}5 = 0{,}5$; $\Phi_{6\text{-}7} = 0 + 0{,}1 = 0{,}1$.

Нижня межа множини «дерева рішення» при виключенні ребра (6–0) становитиме:

$$H^{VII} = 30{,}3 + 2{,}4 = 32{,}7\ хв$$

Визначення нижньої межі підмножини при виключенні ребра (6–0) виконано в таблицях 10.36 – 10.37.

Таблиця 10.36

Матриця відстаней після включення ребра (6–0) у рішення

i \ *j*	**0**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	–	–	7,3	2,6	–	0,1	4,4	–
2	–	0,7	–	–	–	–	–	0
3	–	–	0,7	2,8	3,1	4,9	0,3	0
4	7,1	0	–	0	–	2,4	0,1	1,5
5	2,4	2,1	0	–	0,8	2,5	0	–
6	0	1,5	–	1,3	–	0	0,8	0,7
~~**8**~~	~~4,6~~	~~0~~	~~0,5~~	~~0,3~~	~~1,8~~	~~1,6~~	~~–~~	~~0,3~~
9	–	0	2,3	–	2,3	0,1	0,7	–

Таблиця 10.37

Матриця після редукції з викреслюванням ребра (6–0)

i \ *j*	**3**	**4**	**5**	**6**	**7**	**8**	**9**	d_i	A_i
0	–	7,2	2,5	–	0	4,3	–	0,1	2,5
2	0,7	–	–	–	–	–	0	0	0,7
3	–	0,7	2,8	3,1	4,9	0,3	0	0	0,3
4	0	–	0	–	2,4	0,1	1,5	0	0
5	2,1	0	–	0	2,5	0	–	0	0
8	0	0,5	0,3	1,0	1,6	–	0,3	0	0,3
9	0	2,3	–	1,5	0,1	0,7	–	0	0,1
d_j	0	0	0	0,8	0	0	0	**0,9**	
B_j	0	0,5	0,3	1,0	0,1	0,1	0		

Нижня межа підмножини при виключенні ребра (6–0) складе:

$$H^{VIII} = 30{,}3 + 0{,}9 = 31{,}2\ хв$$

Визначимо штрафи для нульових елементів таблиці 10.37:

$\Phi_{0\text{-}7} = 2{,}5 + 0{,}1 = 2{,}6;$ $\Phi_{4\text{-}5} = 0 + 0{,}3 = 0{,}3;$ $\Phi_{5\text{-}8} = 0 + 0{,}1 = 0{,}1;$
$\Phi_{2\text{-}9} = 0{,}7 + 0{,}3 = 1{,}0;$ $\Phi_{5\text{-}4} = 0 + 0{,}5 = 0{,}5;$ $\Phi_{8\text{-}3} = 0{,}3 + 0 = 0{,}3;$
$\Phi_{3\text{-}9} = 0{,}3 + 0 = 0{,}3;$ $\Phi_{5\text{-}6} = 0 + 1{,}0 = 1{,}0;$ $\Phi_{9\text{-}3} = 0{,}1 + 0 = 0{,}1;$
$\Phi_{4\text{-}3} = 0 + 0 = 0.$

Нижня межа множини рішення при виключення ребра (0–7) становитиме:

$$H^{IX} = 31{,}2 + 2{,}6 = 33{,}8\ хв$$

Для визначення нижньої межі підмножини при виключенні ребра (0–7) визначимо у таблицях 10.38 – 10.39.

Таблиця 10.38

Матриця відстаней після включення ребра (0–7) до рішення

i \ *j*	3	4	5	6	7	8	9
0	–	7,2	2,5	–	0	4,3	–
2	0,7	–	–	–	–	–	0
3	–	0,7	2,8	2,3	4,9	0,3	0
4	0	–	0	–	2,4	0,1	1,5
5	2,1	0	–	0	2,5	0	–
8	0	0,5	0,3	1,0	1,6	–	0,3
9	0	2,3	–	1,5	0,7	–	2,2

Таблиця 10.39

Матриця відстаней після виконання операції редукції з викреслюванням ребра (0–7)

i \ *j*	3	4	5	6	8	9	d_i	A_i
2	0,7	–	–	–	–	0	0	0,7
3	–	0,7	2,8	2,3	0,3	0	0	0,3
4	0	–	0	–	0,1	1,5	0	0
5	2,1	0	–	0	0	–	0	0
8	0	0,5	0,3	1,0	–	0,3	0	0,3
9	0	2,3	–	1,5	0,7	–	0	0,7
d_j	0	0	0	0	0	0	**0**	
B_j	0	0,5	0,3	1,0	0,1	0		

Нижня межа підмножини у разі виключення ребра (0–7) становить:

$$H^{X} = 31{,}2 + 0 = 31{,}2\ хв$$

Визначимо штрафи для всіх нульових елементів таблиці 10.39:

$\Phi_{2\text{-}9} = 0{,}7 + 0 = 0{,}7$; $\Phi_{4\text{-}5} = 0 + 0{,}3 = 0{,}3$; $\Phi_{5\text{-}8} = 0 + 0{,}1 = 0{,}1$;
$\Phi_{3\text{-}9} = 0{,}3 + 0 = 0{,}3$; $\Phi_{5\text{-}4} = 0 + 0{,}5 = 0{,}5$; $\Phi_{8\text{-}3} = 0{,}3 + 0 = 0{,}3$;
$\Phi_{4\text{-}3} = 0 + 0 = 0$; $\underline{\Phi_{5\text{-}6} = 0 + 1{,}0 = 1{,}0;}$ $\Phi_{9\text{-}3} = 0{,}7 + 0 = 0{,}7$.

Нижня межа у разі виключення ребра (5–6) з множини рішень становитиме:

$$H^{XI} = 31{,}2 + 1{,}0 = 32{,}2\ хв$$

Для визначення нижньої межі підмножини при виключенні ребра (5–6) виконаємо розрахунки у таблицях 10.40 – 10.41.

Таблиця 10.40

Матриця відстаней після включення ребра (5–6) у маршрут

i \ *j*	**3**	**4**	**5**	**6**	**8**	**9**
2	0,7	–	–	–	–	0
3	–	0,7	2,8	2,3	0,3	0
4	0	–	0	–	0,1	1,5
5	2,1	0		0	0	
8	0	0,5	0,3	1	–	0,3
9	0	2,3	–	1,5	0,7	–

Таблиця 10.41

Матриця відстаней після здійснення редукції з викреслюванням ребра (5–6)

i \ *j*	**3**	**4**	**5**	**8**	**9**	d_i	A_i
2	0,7	–	–	–	0	0	0,7
3	–	0,2	2,8	0,2	0	0	0,2
4	0	–	0	0	1,5	0	0
8	0	0	0,3	–	0,3	0	0
9	0	1,8	–	0,6	–	0	0,6
d_j	0	0,5	0	0,1	0	**0,6**	
B_j	0	0,3	0,3	0,3	0		

Нижня межа підмножини при виключенні ребра (5–6) становитиме:

$$H^{XII} = 31{,}2 + 0{,}6 = 31{,}8\ хв$$

Визначимо штрафи для всіх нульових елементів таблиці 10.41:

$\underline{\Phi_{2\text{-}9} = 0{,}7 + 0 = 0{,}7;}$ $\Phi_{4\text{-}5} = 0 + 0{,}3 = 0{,}3;$ $\Phi_{8\text{-}4} = 0 + 0{,}3 = 0{,}3;$

$\Phi_{3\text{-}9} = 0{,}2 + 0 = 0{,}2;$ $\Phi_{4\text{-}8} = 0 + 0{,}3 = 0{,}3;$ $\Phi_{9\text{-}3} = 0{,}6 + 0 = 0{,}6.$

$\Phi_{4\text{-}3} = 0 + 0 = 0;$ $\Phi_{8\text{-}3} = 0 + 0 = 0;$

Нижня межа у разі виключення ребра (2–9) становитиме

$$H^{XIII} = 31{,}8 + 0{,}7 = 32{,}5\ хв$$

Для визначення нижньої межі підмножини при виключенні ребра (2–9) здійснимо розрахунки у таблицях 10.42 – 10.43.

Таблиця 10.42

Матриця відстаней після включення ребра (2–9) до рішення

i \ *j*	**3**	**4**	**5**	**8**	**9**
2	0,7	–	–	–	0
3	–	0,2	2,8	0,2	0
4	0	–	0	0	1,5
8	0	0	0,3	–	0,3
9	0	1,8	–	0,6	–

Таблиця 10.43

Матриця відстаней після виконання операції редукції з викреслюванням ребра (2–9)

i \ *j*	**3**	**4**	**5**	**8**	d_i	A_i
3	–	0	2,8	0	0,2	0
4	0	–	0	0	0	0
8	0	0	0,3	–	0	0
9	0	1,8	–	0,6	0	0,6
d_j	0	0	0	0	**0,2**	
B_j	0	0	0,3	0		

Нижня межа підмножини при виключенні ребра (2–9) становитиме

$$H^{XIV} = 31{,}8 + 0{,}2 = 32{,}0\ хв$$

Визначимо штрафи для всіх нульових елементів таблиці 10.43:

$\Phi_{3\text{-}4} = 0 + 0 = 0$; $\Phi_{3\text{-}8} = 0 + 0 = 0$; $\Phi_{4\text{-}3} = 0 + 0 = 0$;
$\Phi_{8\text{-}3} = 0 + 0 = 0$; $\Phi_{8\text{-}4} = 0 + 0 = 0$; $\Phi_{4\text{-}5} = 0 + 0{,}3 = 0{,}3$;
$\Phi_{9\text{-}3} = 0 + 0{,}6 = 0{,}6$; $\Phi_{4\text{-}8} = 0 + 0 = 0$.

Нижня межа при виключенні ребра (9–3) становитиме

$$H^{XV} = 32{,}0 + 0{,}6 = 32{,}6 \text{ хв}$$

Визначимо нижню межу при виключенні ребра (9–3) в таблицях 10.44 – 10.45.

Таблиця 10.44

Матриця відстаней після включення ребра (9–3)

i \ *j*	3	4	5	8
3	-	0	2,8	0
4	0	-	0	0
8	0	0	0,3	-
9	0	1,8	-	0,6

Таблиця 10.45

Приведена матриця відстаней з викресленням ребра (9–3)

i \ *j*	4	5	8	d_i	A_i
3	0	2,8	0	0	0
4	–	0	0	0	0
8	0	0,3	–	0	0,3
d_j	0	0	0	**0**	
B_j	0	0,3	0		

Нижня межа підмножини при виключенні ребра (9–3) становитиме

$$H^{XVI} = 32{,}0 + 0 = 32{,}0 \text{ хв}$$

Визначимо штрафи для всіх нульових елементів таблиці 10.45:

$\Phi_{3\text{-}4} = 0 + 0 = 0$; $\Phi_{8\text{-}4} = 0{,}3 + 0 = 0{,}3$; $\Phi_{4\text{-}8} = 0 + 0 = 0$
$\Phi_{3\text{-}8} = 0 + 0 = 0$; $\Phi_{4\text{-}5} = 0 + 0{,}3 = 0{,}3$;

Нижня межа при виключенні ребра (4–5) з множини рішень становитиме

$$H^{XVII} = 32{,}0 + 0{,}3 = 32{,}3 \text{ хв}$$

Згідно з таблицею 10.46 в маршрут включимо ребра (3–8) та (8–4).

Таблиця 10.46

Матриця відстаней після включення ребра (4–5)

i \ *j*	**4**	**8**
3	–	0
8	0	–

В результаті проведених розрахунків, оптимальний на даному етапі маршрут доставки вантажів з довжиною 32,0 км утворюють ребра:

(10–2), (1–10), (7–1), (6–0), (0–7), (5–6), (2–9), (9–3), (4–5), (3–8), (8–4).

Після розташування ребер маршруту в правильному порядку (з виїздом зі складу в точці 0 та поверненням у нього), матимемо наступний порядок обходу точок мережі:

$$0 \to 7 \to 1 \to 10 \to 2 \to 9 \to 3 \to 8 \to 4 \to 5 \to 6 \to 0$$

Довжина даного маршруту складе:

$$L_м = 2{,}8 + 2{,}1 + 3{,}0 + 2{,}0 + 2{,}2 + 1{,}7 + 2{,}1 + 3{,}7 + 5{,}1 + 4{,}8 + 2{,}5 = 32{,}0 \text{ км}$$

Отже, довжина оптимального маршруту обслуговування 10-ти роздрібних точок торгової мережі склала 32,0 км. Даний маршрут показано на рисунку 10.13, а дерево оптимального рішення даної задачі наведено на рисунку 10.14.

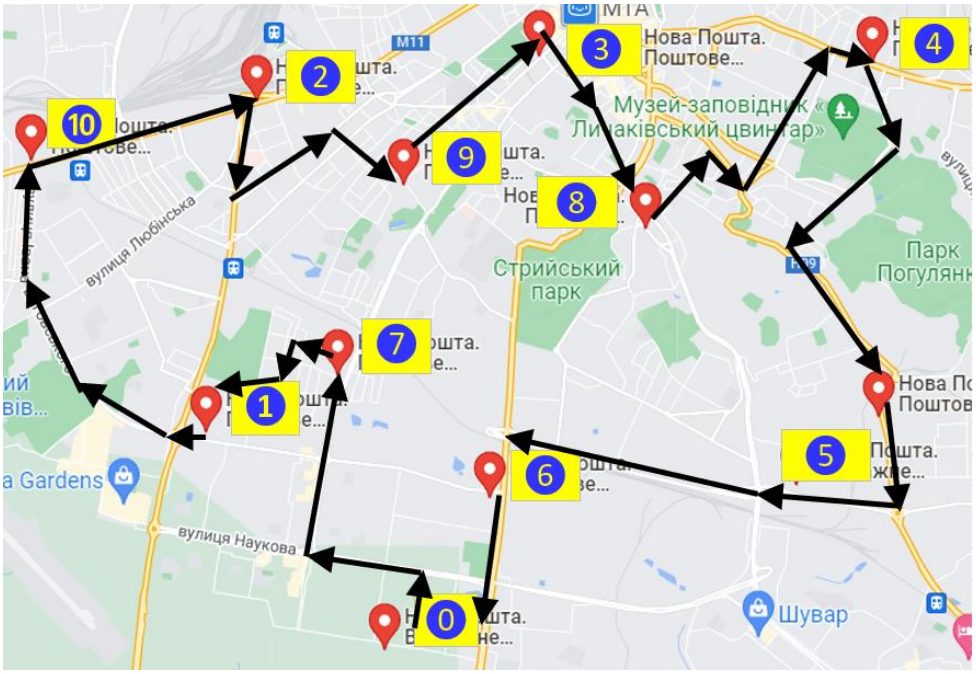


Рисунок 10.13 – Порядок обслуговування точок за оптимальним маршрутом

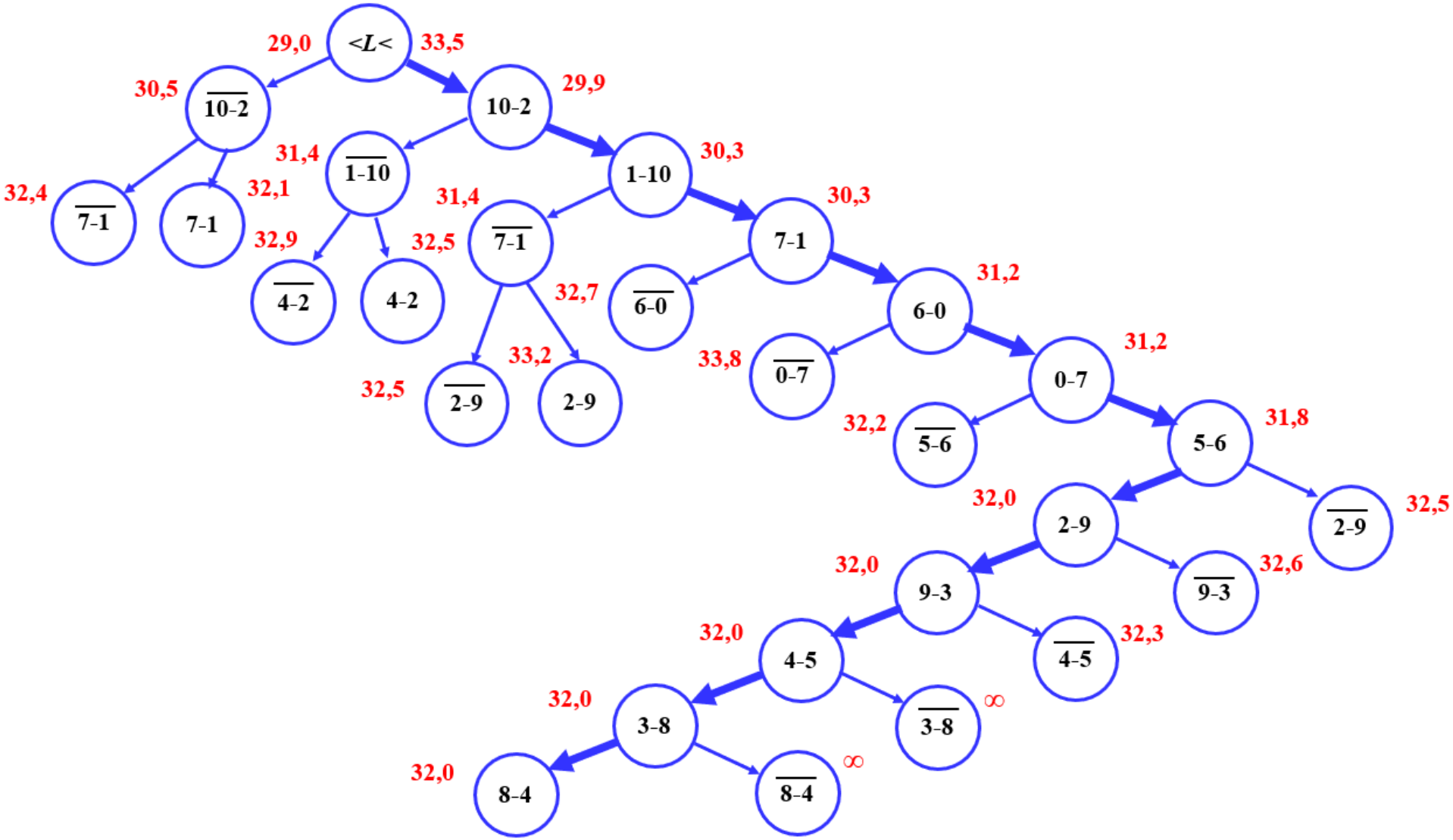


Рисунок 10.14 – Дерево пошуку рішення оптимальної черговості обслуговування розширеної мережі з 10-ти точок

Висновки до розділу

У зв'язку з необхідністю розширення мережі, яку обслуговує автотранспортне підприємство до 10-ти точок, виникла потреба у визначенні нової раціональної послідовності обслуговування роздрібних точок торгової мережі.

На основі вирішення задачі комівояжера методом найближчого сусіда було визначено кращий маршрут довжиною 33,5 км, а за допомогою методу гілок та меж, довжина кращого маршруту склала 32,0 км, що на 4,5% менше.

Доставка вантажів передбачається одним вантажним автомобілем до усіх 10-ти точок в кожному рейсі.

Контрольні запитання

1. Навести формулювання задачі комівояжера?
2. Навести приклад задач, які можуть вирішуватись як задача комівояжера?
3. Які групи методів існують для вирішення задачі комівояжера?
4. До якої групи методів вирішення задачі комівояжера відноситься метод гілок та меж?
5. В чому полягає підхід до вирішення задачі комівояжера за допомогою методу гілок та меж?
6. В чому полягає поняття редукції при вирішенні задачі комівояжера?
7. Як визначається верхня межа розв'язку методом гілок та меж?
8. Як визначається нижня межа розв'язку методом гілок та меж?
9. В якому випадку слід зупинити розв'язок задачі комівояжера за основною гілкою методом гілок та меж?
10. Що слід зробити після закінчення розв'язку за основною гілкою методом гілок та меж?

РОЗДІЛ 11

Багатокритеріальні задачі оптимізації та методи їх розв'язку

11.1 Вступ

Усі розглянуті в попередніх розділах оптимізаційні задачі мали всього один критерій оптимальності, а моделі, що їх описували, були *однокритеріальними*. В реальних задачах вибору найбільш пріоритетного рішення, що виникають на практиці, як правило, присутні кілька критеріїв оптимальності. Можна привести багато прикладів, коли потрібно знайти рішення, для якого досягалися найкращі значення відразу за кількома критеріями (найбільш поширене завдання, яке ми вирішуємо дуже часто – це пошук покупки, яка була б якомога якісніша і якомога дешевша).

11.2 Теоретичні відомості

11.2.1 Основні визначення та поняття

Завдання вибору деякого рішення з множини допустимих рішень з урахуванням декількох критеріїв оптимальності називають ***багатокритеріальною задачею оптимізації***.

Під багатокритеріальною задачею найчастіше розуміють не власне вербальний опис задачі, а її модель, тобто ***багатокритеріальна задача*** – це математична модель прийняття оптимального рішення за декількома критеріями [12, 24 – 27]. Ці критерії можуть відображати оцінки різних якостей об'єкта або процесу, з приводу яких приймається рішення.

Формально багатокритеріальна задача як модель задається так:

$$\begin{cases} F(x) \to max \\ x \in D \end{cases}, \qquad (11.1)$$

де D – множина допустимих рішень;

$F(x)$ – векторна функція аргументу x, яку можна представити у такому вигляді:

$$F(x) = \{ f_1(x), f_2(x), \ ..., f_k(x) \}, \qquad (11.2)$$

де $f_1(x), f_2(x), ..., f_k(x)$ – скалярні функції векторного аргументу x, кожна з яких є математичним виразом одного критерію оптимальності.

Так як в даній моделі використовується *векторна цільова функція*, її часто називають ***задачею векторної оптимізації***.

Очевидно, що задача (11.1) не належить до класу задач математичного програмування, тому що моделі цього класу задач завжди містять тільки одну цільову функцію векторного аргументу.

Інакше задачу (11.1) можна переписати у вигляді:

$$\begin{pmatrix} f_1(x) \\ f_2(x) \\ \dots \\ f_k(x) \end{pmatrix} \to \max, \text{ де } x \in D, \qquad (11.3)$$

Тут розглядається комплексний векторний критерій, за допомогою якого можна досягти максимального ефекту, при цьому необов'язково досягнення екстремуму у всіх функціях. Існування рішення, яке буквально максимізує всі цільові функції, є рідкісним винятком (якщо згадати приклад про пошук одночасно дуже якісної і дуже дешевої покупки, то стає зрозумілим, що знаходження такого рішення – рідкісна удача, але, набагато частіше, це – нерозв'язна задача).

Задача векторної оптимізації в загальному випадку не має чіткого математичного рішення. Для отримання того чи іншого її вирішення необхідно використовувати додаткову суб'єктивну інформацію фахівця в даній предметній області, якого прийнято називати особою, що приймає рішення (англ. *decisionmaker*). Це означає, що при вирішенні задачі різними фахівцями із залученням різних джерел інформації, скоріше всього будуть отримані різні відповіді.

Задачі векторної оптимізації в даний час прийнято розглядати в рамках теорії прийняття рішень, основною особливістю задач якої є наявність невизначеності, яка не може бути виключена за допомогою різних прийомів моделювання та об'єктивних розрахунків.

В багатокритеріальних задачах невизначеність полягає в тому, що невідомо, яким критерієм віддати перевагу і в якій мірі.

Для усунення цієї невизначеності необхідно:

1. сформулювати спеціальний принцип оптимальності;

2. залучити додаткову суб'єктивну інформацію особи, що приймає рішення, засновану на її досвіді та інтуїції.

Так, нехай x_1 та x_2 допустимі рішення задачі (11.3). Кажуть, що x_1 краще рішення в порівнянні з x_2, якщо $f_i(x_1) \geq f_i(x_2)$, для будь-якого $i = 1,k$, при чому існує i_0 такий, що $f_i(x_1) > f_i(x_2)$.

Іншими словами, будемо вважати, що рішення x_1 краще в порівнянні з рішенням x_2, якщо воно не гірше x_2 по всіх розглянутих критеріях, причому серед всіх критеріїв є хоча б один критерій з номером i_0, для якого рішення x_1 краще, ніж x_2.

Отже, деякий розв'язок x^* задачі (11.3) називається ***ефективним рішенням*** даної задачі, якщо для нього не існує більш кращих розв'язків. Інакше можна сказати, що ефективним рішенням називається таке рішення x^*, яке не можна поліпшити по якомусь із критеріїв, не погіршивши при цьому значення інших критеріїв.

Множина ефективних рішень називається множиною Парето і позначається $P(D)$. Очевидно, множина Парето є підмножиною множини допустимих рішень D, яка, в свою чергу, належить n-вимірному векторному простору E, тобто $P(D) \subset D \subset E^n$ (див. рисунок 11.1).

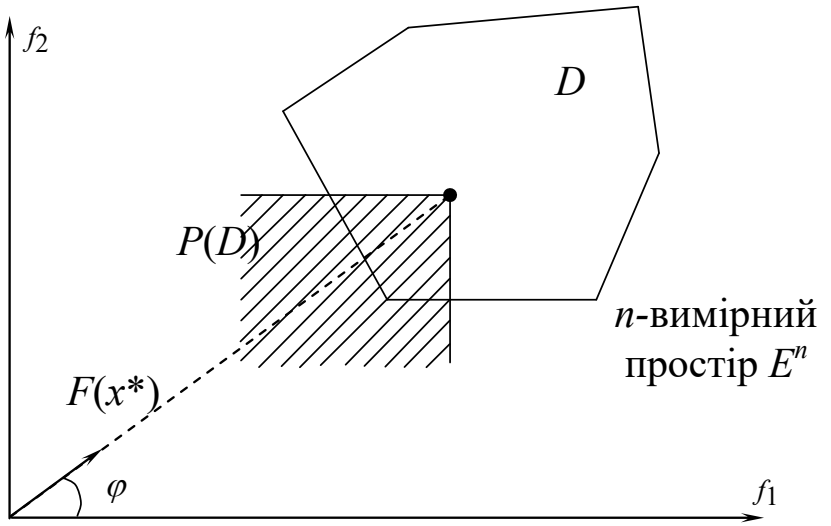


Рисунок 11.1 – Геометрична інтерпретація умови $P(D) \subset D \subset E^n$

Вектор значень критеріїв, обчислених для ефективного розв'язку $F(x^*)$, називається ***ефективної оцінкою***. Сукупність усіх ефективних

оцінок, тобто образ множини Парето в просторі критеріїв, називається ***множиною ефективних оцінок*** і, як правило, позначається як $F(P)$. Множина ефективних оцінок є підмножиною множини допустимих рішень в просторі критеріїв $F(D)$, яка, в свою чергу, є підмножиною k-вимірного векторного простору, тобто $F(P) \subset F(D) \subset E^k$. Тобто можна сказати, що множині Парето P, яка належить множині допустимих рішень D, за допомогою векторної функції F можна зіставити множину ефективних оцінок $F(P)$.

Сенс введеного поняття ефективного рішення полягає в тому, що оптимальне рішення слід шукати лише серед елементів множини Парето – множини $P(D)$. В іншому випадку завжди знайдеться точка x, що виявляється кращою, незалежно від розстановки пріоритетів важливості окремих критеріїв. У множині Парето існує поняття *узгодженого оптимуму*, при якому при якому жодний з можливих розв'язків неможна покращити за певним критерієм, не погіршивши при цьому інший. Множина розв'язків за Парето дозволяє звузити клас можливих претендентів на остаточне рішення і виключити з розгляду завідомо неконкурентоспроможні варіанти. А остаточний вибір здійснюється на основі додаткової інформації про перевагу особи, що приймає рішення.

Недолік принципу Парето в тому, що він пропонує в якості розв'язку множину рішень, що не завжди прийнятно. Для того, щоб вибрати з цієї множини єдине рішення потрібна якась додаткова інформація, припущення, домовленість про те, що ж вважати найкращим рішенням.

Тому потрібні додаткові процедури для пошуку одного розв'язку з множини Парето. Специфіка розв'язку таких завдань полягає в тому, що сам вибір такої процедури, методу знаходження остаточного рішення в якійсь мірі заснований на припущеннях особи, що приймає рішення, тобто на суб'єктивній інформації.

Методи вирішення задач багатокритеріальної оптимізації можна поділити на чотири групи:

– методи, засновані на згортанні критеріїв;
– методи, які використовують обмеження на критерії;
– методи цільового програмування;
– методи, засновані на знаходженні компромісного рішення.

Замість вихідної багатокритеріальної задачі відповідно до обраного методу, формується задача що її заміщає. До складу такої задачі входить один критерій, а до вихідної системи обмежень додається одне або декілька додаткових обмежень.

Розв'язок такої задачі називається *субоптимальним*.

Розглянемо деякі з вищезазначених методів розв'язку багатокритеріальних задач.

11.2.2 Метод згортання критеріїв

В методах, заснованих на згортанні критеріїв, з декількох локальних критеріїв формується один. Процедура згортання критеріїв така.

Нехай задано вектор вагових коефіцієнтів критеріїв $\alpha = (\alpha_1, \alpha_2, ..\alpha_n)$, що характеризують важливість критерію $\sum_{k=1}^{K} \alpha_k = 1,\ \alpha_k \geq 0,\ k = \overline{1,K}$

Лінійна функція в такому випадку буде являти собою суму критеріїв, помножених на вагові коефіцієнти. Задача математичного програмування зводиться до однокритеріальної і приймає такий вигляд:

$$\begin{gathered} F' = \sum_{k=1}^{K} \alpha_k f_k(x) \to \max \\ \sum_{k=1}^{K} \alpha_k = 1,\ \alpha_k \geq 0,\ k = \overline{1,K},\ x \in D \end{gathered}, \qquad (11.4)$$

Рішення, яке отримане в результаті оптимізації такого критерію можна вважати ефективним.

До недоліків методу можна віднести такі:

– малому приросту коефіцієнтів відповідає великий приріст функції, тобто рішення задачі не є стійким;

– необхідність визначення вагових коефіцієнтів і ступеню їх значимості (нормування критеріїв).

11.2.3 Метод послідовних поступок

Ще одним методом, який дозволяє розв'язувати багатокритеріальні задачі та отримувати при цьому ефективне рішення є метод послідовних поступок.

Алгоритм цього методу такий.

1. Критерії нумеруються в порядку зменшення важливості.
2. Розв'язується задача для першої цільової функції

$$f_1(x) \to \max$$
$$\begin{cases} x \geq 0, \\ x \in D \end{cases} \tag{11.5}$$

та визначається ефективне рішення f_1^*.

3. Встановлюється припустима поступка Δ_1 по цьому критерію.

4. Розв'язується задача для другої цільової функції з відповідним додатковим обмеженням:

$$f_2(x) \to \max$$
$$\begin{cases} f_1(x) \geq f_1^* - \Delta_1, \\ x \geq 0, \\ x \in D \end{cases} \tag{11.6}$$

Якщо в задачі більше двох критеріїв, то пункти 3 та 4 повторюються для $f_2(x), \ldots f_k(x)$.

11.3 Приклад розв’язання задач. Задача 14

Умова задачі

На деяку вантажну станцію прибуває состав передаточного поїзда із M вагонів, які необхідно вивантажити впродовж доби на трьох вантажних фронтах місткістю відповідно m_1, m_2 та m_3 вагонів.

У зв’язку з різним технічним обладнанням вантажних фронтів витрати залізниці, які пов’язані з вивантаженням вагонів на них, можуть певним чином відрізнятись; в рамках даної задачі вказані витрати прийняті рівними C_1, C_2 та C_3 умовних одиниць за 1 вагон.

Прийнято, що плата клієнтів за вивантаження вантажів складатиме $П_1$, $П_2$ та $П_3$ умовних одиниць за 1 вагон.

Необхідно знайти ефективне рішення, яке забезпечує мінімум витрат залізниці та клієнтів при виваженні вагонів.

Вихідні дані

Значення наведених в умові задачі параметрів задані в таблиці 11.1.

Таблиця 11.1

Значення параметрів для прикладу

M	m_1	m_2	m_3	C_1	C_2	C_3	$П_1$	$П_2$	$П_3$
50	25	28	20	20	30	35	60	55	45

Розв'язання задачі

При формалізації задачі в якості змінних обираємо кількість вагонів, які розвантажуються на вантажних фронтах: x_1 – на першому фронті; x_2– на другому фронті; x_3 – на третьому фронті.

Тоді отримаємо дві цільові функції, які матимуть такий вигляд:

$$C = 20x_1 + 30x_2 + 35x_3 \to min;$$

$$П = 60x_1 + 55x_2 + 45x_3 \to min.$$

Задача має такі обмеження:

– по загальній кількості вагонів, які розвантажуються:

$$x_1 + x_2 + x_3 = 50;$$

– по місткості вантажних фронтів:

$$x_1 \le 25; x_2 \le 28; x_3 \le 20;$$

Слід також зазначити, що кількість вагонів не може бути від'ємною:

$$x_i \ge 0.$$

задача розподілу вагонів по вантажним фронтам має такий вигляд:

$$\begin{cases} x_1 + x_2 + x_3 = 50; \\ x_1 \le 25; \\ x_2 \le 28; \\ x_3 \le 20; \\ x_i \ge 0. \end{cases} \qquad (11.1)$$

$$C = 20x_1 + 30x_2 + 35x_3 \to \min$$

$$П = 60x_1 + 55x_2 + 45x_3 \to \min$$

Або при використанні єдиного векторного критерію оптимізації:

$$\begin{pmatrix} C = 20x_1 + 30x_2 + 35x_3 \\ \Pi = 60x_1 + 55x_2 + 45x_3 \end{pmatrix} \to \min$$

$$\begin{cases} x_1 + x_2 + x_3 = 50; \\ x_1 \le 25; \\ x_2 \le 28; \\ x_3 \le 20; \\ x_i \ge 0. \end{cases} \qquad (11.2)$$

У такій постановці завдання пошуку таких значень кількості вагонів, які забезпечують мінімальне значення єдиного векторного критерію, зводиться до задачі векторної оптимізації.

Уся множина допустимих рішень, яка отримана повним перебором варіантів, представлена у векторній площині на рисунку 11.2.

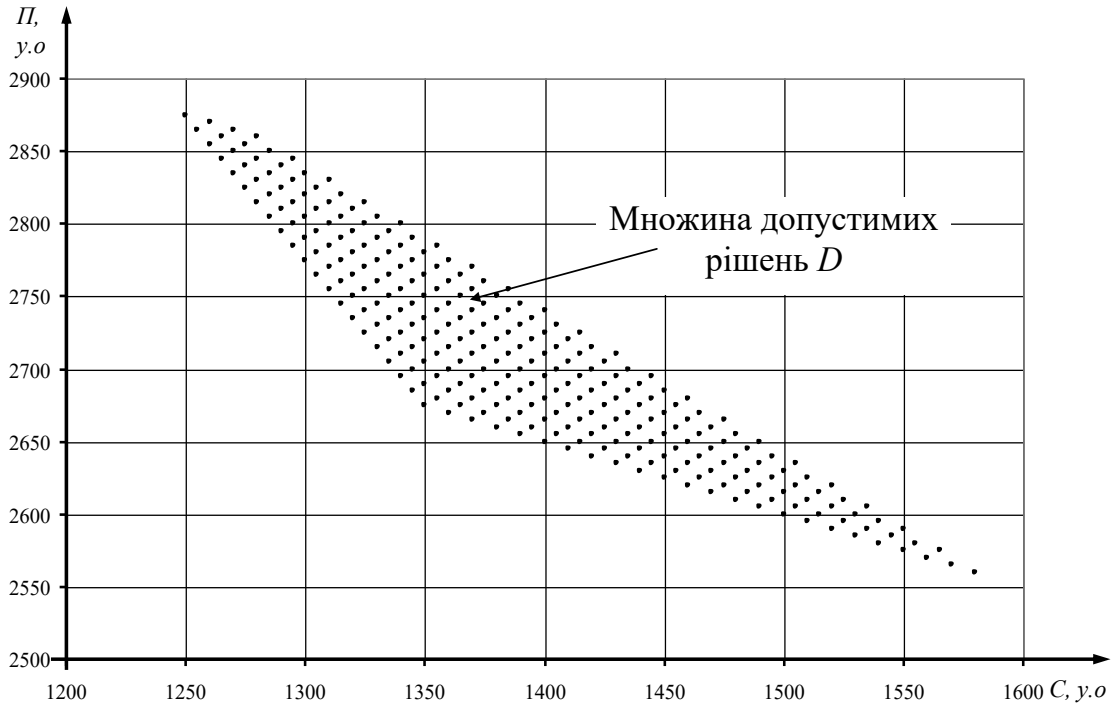


Рисунок 11.2 – Множина допустимих рішень у векторній площині

Таким чином, на рисунку 11.2 представлена множина, яка складається з 294 можливих допустимих варіантів розв'язку задачі (11.2). Ця множина є підмножиною множини усіх можливих рішень.

Графічно задачу можна розв'язати таким чином. Обираємо будь-яку точку на площині з множини допустимих рішень. Нехай це буде точка x_{ij} (див. рисунок 11.3) та направляємо її в тому ж напрямку, що й цільову функцію.

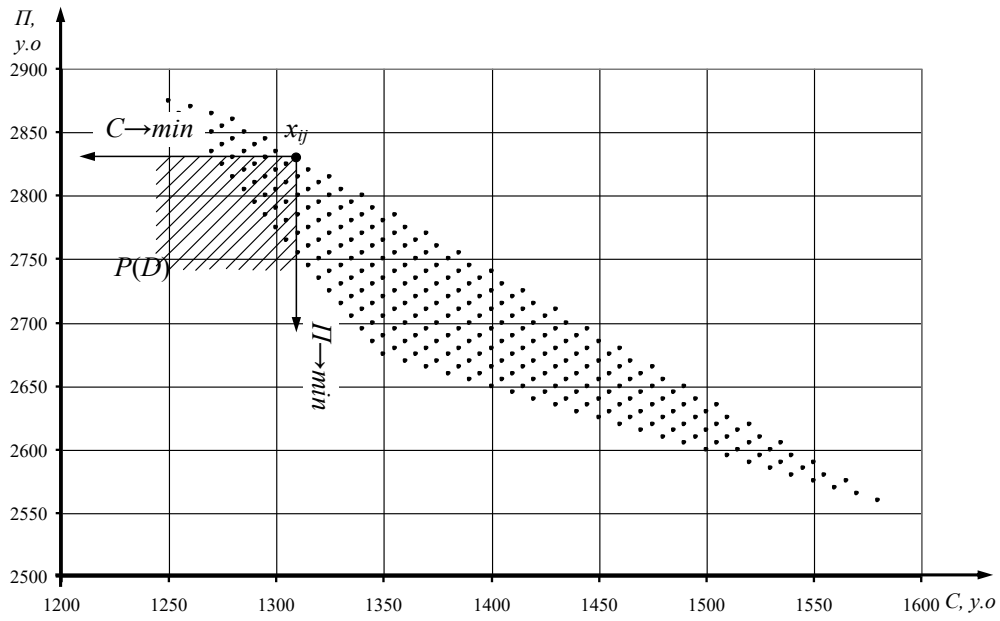


Рисунок 11.3 – Пошук ефективного за Парето рішення

Тобто, якщо цільова функція направлена до мінімуму, то й пошук оптимального рішення необхідно здійснювати в напрямку мінімальних значень.

У випадку, якщо до множини $P(D)$ попадають ще якісь точки x_{mn}, то розв'язок x_{ij} не є оптимальним; існують інші розв'язки, які кращі за x_{ij}. Отже, серед множини допустимих рішень необхідно обрати таку підмножину, кожен розв'язок якої неможливо покращити, не погіршивши при цьому один із компонентів векторного критерію.

Таку підмножину називають ***множиною Парето***, а варіанти рішення називають ***незрівняними за Парето***, тому що поліпшення якості рішення з одним локальним критеріям призводить до погіршення якості рішення з іншим.

Множина ефективних рішень Парето для задачі, яка розглядається, наведена на рисунку 11.4.

При цьому, крайня ліва точка з множини Парето відповідає рішенню, для якого витрати залізниці є мінімальними, а крайня права точка – рішенню, для якого витрати клієнтів є мінімальними.

Усі ефективні варіанти множини Парето наведено у таблиці 11.2.

Таким чином, рішення даної задачі дозволяє значно скоротити число усіх можливих варіантів розв'язку та допомогти серед незрівнянних за Парето варіантів, обрати той, який влаштовує особу, що приймає рішення. Але, нажаль, така постановка не дає однозначного об'єктивного рішення.

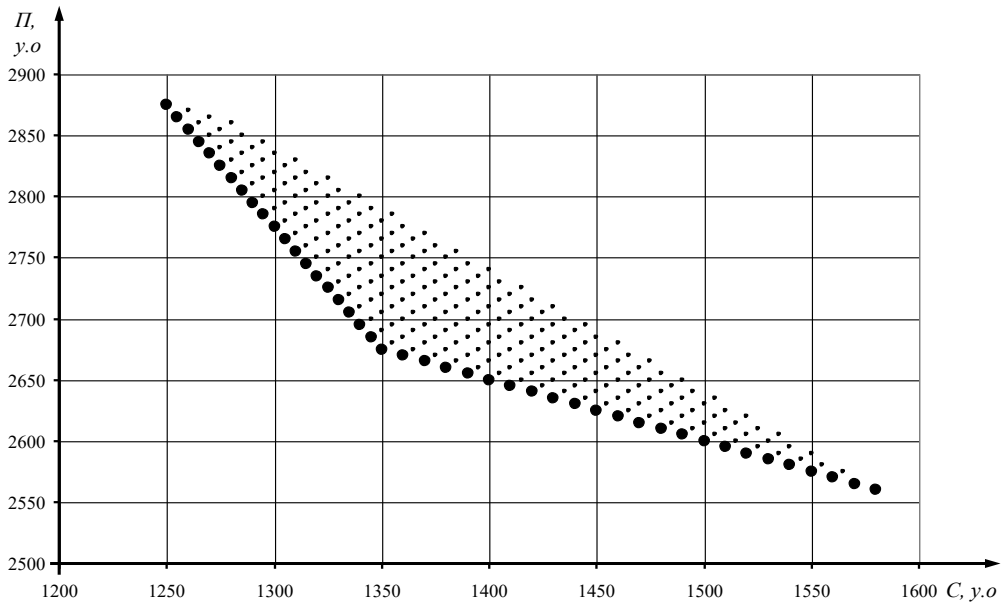


Рисунок 11.4 – Множина ефективних за Парето рішень

Таблиця 11.2

Множина ефективних за Парето рішень

№	x_1	x_2	x_3	*С*	*П*	№	x_1	x_2	x_3	*С*	*П*
1	25	5	20	1350	2675	23	23	7	20	1370	2665
2	25	6	19	1345	2685	24	22	8	20	1380	2660
3	25	7	18	1340	2695	25	21	9	20	1390	2655
4	25	8	17	1335	2705	26	20	10	20	1400	2650
5	25	9	16	1330	2715	27	19	11	20	1410	2645
6	25	10	15	1325	2725	28	18	12	20	1420	2640
7	25	11	14	1320	2735	29	17	13	20	1430	2635
8	25	12	13	1315	2745	30	16	14	20	1440	2630
9	25	13	12	1310	2755	31	15	15	20	1450	2625
10	25	14	11	1305	2765	32	14	16	20	1460	2620
11	25	15	10	1300	2775	33	13	17	20	1470	2615
12	25	16	9	1295	2785	34	12	18	20	1480	2610
13	25	17	8	1290	2795	35	11	19	20	1490	2605
14	25	18	7	1285	2805	36	10	20	20	1500	2600
15	25	19	6	1280	2815	37	9	21	20	1510	2595
16	25	20	5	1275	2825	38	8	22	20	1520	2590
17	25	21	4	1270	2835	39	7	23	20	1530	2585
18	25	22	3	1265	2845	40	6	24	20	1540	2580
19	25	23	2	1260	2855	41	5	25	20	1550	2575
20	25	24	1	1255	2865	42	4	26	20	1560	2570
21	25	25	0	**1250**	2875	43	3	27	20	1570	2565
22	24	6	20	1360	2670	44	2	28	20	1580	**2560**

Вирішимо поставлену задачу за допомого розглянутих вище методів. Для цього необхідно привести її до основної задачі лінійного програмування. Вводимо додаткові змінні: y_1, y_2, y_3 (резерви місткості вантажних фронтів), щоб перетворити нерівності на рівняння:

$$\begin{cases} x_1 + x_2 + x_3 = 50; \\ x_1 + y_1 = 25; \\ x_2 + y_2 = 28; \\ x_3 + y_3 = 20; \end{cases} \tag{11.3}$$

$$C = 20x_1 + 30x_2 + 35x_3 \to \min$$

$$П = 60x_1 + 55x_2 + 45x_3 \to \min$$

Спочатку, за допомогою симплекс методу вирішуємо задачу для першого критерію ($C \to \min$); отримаємо наступний розв'язок (1):

$$x_1 = 25 \text{ ваг}, x_2 = 25 \text{ ваг}, x_3 = 0 \text{ ваг.}$$

Витрати залізниці складуть $C^* = 1250$ *у.о.*, а клієнтів $П^* = 2875$ *у.о.*

Далі, за допомогою симплекс методу вирішуємо задачу для другого критерію ($П \to \min$), в результаті чого отримаємо такий розв'язок (2):

$$x_1 = 2 \text{ ваг}, x_2 = 28 \text{ ваг}, x_3 = 20 \text{ ваг.}$$

Витрати клієнтів складатимуть $П^* = 2560$ *у.о.*

Витрати залізниці складатимуть $C^* = 1580$ *у.о.*

Отже отримано 2 граничних рішення, при чому обидва вони є ефективними. На рисунку 11.5 наведено лінійну залежність витрат клієнтів на розвантаження вантажу від витрат залізниці.

Рисунок 11.5 – Лінійна залежність витрат клієнтів від витрат залізниці

Ця залежність демонструє, яким чином змінюються витрати *С* та *П*, але також не дає однозначного рішення. Крім того, в порівнянні з попереднім рішенням (див. рисунок 11.4), було втрачено 42 можливих ефективних розв'язки задачі.

Виконаємо розв'язок задачі за допомогою *методу згортки критеріїв*.

Для прикладу припустимо, що критерії *С* та *П* рівні за значенням, тобто $\alpha_1 = 0{,}5$ та $\alpha_2 = 0{,}5$.

Таким чином, замість системи (11.3) отримаємо таку систему рівнянь:

$$\begin{cases} x_1 + x_2 + x_3 = 50; \\ x_1 + y_1 = 25; \\ x_2 + y_2 = 28; \\ x_3 + y_3 = 20; \end{cases} \qquad (11.4)$$

$$C = 0{,}5(20x_1 + 30x_2 + 35x_3) \to \min$$

$$П = 0{,}5(60x_1 + 55x_2 + 45x_3) \to \min$$

Або після згортки критеріїв:

$$\begin{cases} x_1 + x_2 + x_3 = 50; \\ x_1 + y_1 = 25; \\ x_2 + y_2 = 28; \\ x_3 + y_3 = 20; \end{cases} \qquad (11.5)$$

$$СП = 40x_1 + 42{,}5x_2 + 40x_3 \to \min$$

За допомогою симплекс методу вирішуємо задачу для загального критерію (*СП*→min), в результаті чого отримаємо такий розв'язок (3):

$$x_1 = 25 \text{ ваг}, x_2 = 5 \text{ ваг}, x_3 = 20 \text{ ваг}.$$

Витрати залізниці складатимуть $C^* = 1350$ *у.о.*

Витрати клієнтів складатимуть $П^* = 2675$ *у.о.*

Виконаємо розв'язок задачі *методом послідовних поступок*.

При вирішенні задачі симплекс-методом для першого критерію (*С→min*) було отримано ефективний розв'язок, при якому $C^* = 1250$ *у.о.* Приймаємо максимальну величину поступки по цьому критерію Δ_1=20% та вирішуємо задачу у відповідності до алгоритму.

Для цього вводимо додаткове обмеження

$$C \le C^* + \Delta_1, \text{ де } C^* = 1250 \text{ у.о.}, \text{ а } \Delta_1 = 0{,}2 \cdot 1250 = 250 \text{ у.о.}$$

$$\begin{cases} x_1 + x_2 + x_3 = 50; \\ x_1 + y_1 = 25; \\ x_2 + y_2 = 28; \\ x_3 + y_3 = 20; \\ C \le C^* + \Delta_1 \end{cases} \qquad (11.6)$$

$$\Pi' = 60x_1 + 55x_2 + 45x_3 \to \min$$

$$\begin{cases} x_1 + x_2 + x_3 = 50; \\ x_1 + y_1 = 25; \\ x_2 + y_2 = 28; \\ x_3 + y_3 = 20; \\ 20x_1 + 30x_2 + 35x_3 \le 1500 \end{cases} \qquad (11.7)$$

$$\Pi' = 60x_1 + 55x_2 + 45x_3 \to \min$$

За допомогою симплекс методу вирішуємо задачу для другого критерію ($\Pi \to min$) при додатковому обмеженні $C \le C^* + \Delta_1$. Отримаємо такий розв'язок (4):

$$x_1 = 10 \text{ ваг}, x_2 = 20 \text{ ваг}, x_3 = 20 \text{ ваг}.$$

Витрати клієнтів складатимуть $\Pi^* = 2600$ *у.о.*

Витрати залізниці складатимуть $C^* = 1500$ *у.о.*

Покажемо усі отримані розв'язки на графічній інтерпретації задачі, (див. рисунок 11.6), або у векторній площині (див. рисунок 11.7).

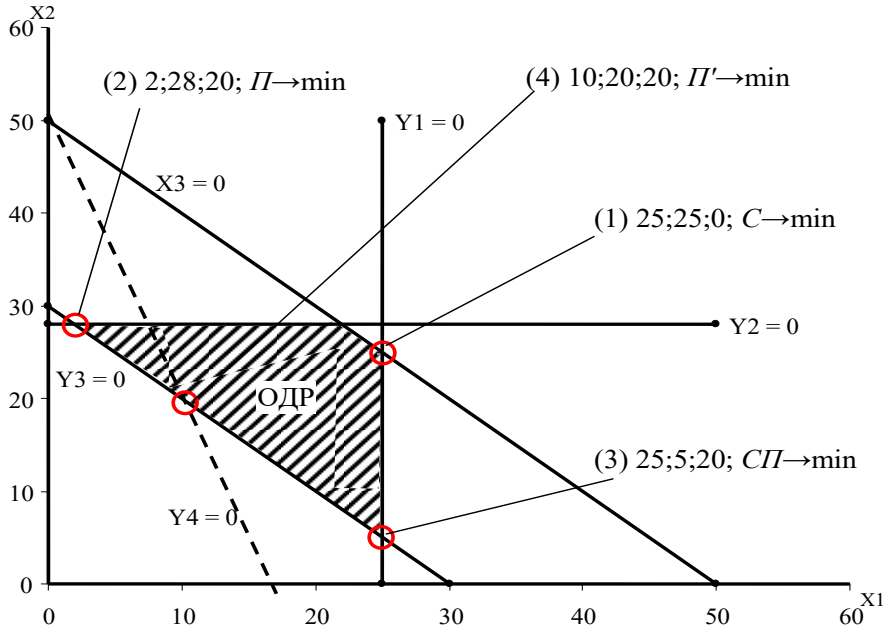


Рисунок 11.6 – Графічна інтерпретація розв'язку задачі

Висновки

Розглянуті методи рішення задач багатокритеріальної оптимізації мають свої переваги та недоліки.

1. Векторна оптимізація як частковий випадок багатокритеріальної оптимізації дозволяє знайти таку область компромісів, до якої входить підмножина допустимих ефективних розв'язків, які є незрівняними за Парето. Таку підмножину можна знайти геометрично у векторній площині. Таким чином, рішення задачі векторної оптимізації дозволяє значно скоротити число усіх можливих варіантів розв'язку та допомогти серед незрівнянних за Парето варіантів, обрати той, який влаштовує особу, що приймає рішення. Але, нажаль, така постановка не дає однозначного об'єктивного рішення.

2. Задачі багатокритеріальної оптимізації можна розв'язувати з використанням відомих методів лінійного програмування, наприклад симплекс-методу, отримуючи при цьому деяке ефективне рішення. Але таке рішення не є оптимальним за комплексним критерієм, тому потребує подальшого уточнення.

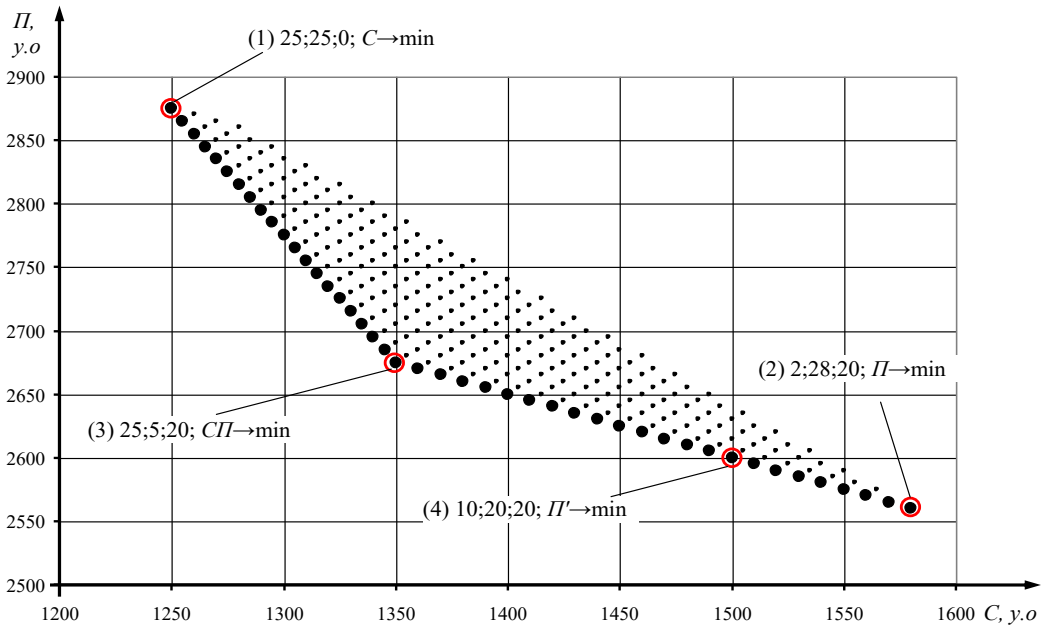


Рисунок 11.7 – Множина ефективних за Парето рішень з варіантами розв'язків

3. З використанням методу згортання критеріїв можна отримати єдине ефективне і, водночас, оптимальне рішення, яке звужує область компромісів Парето до одного єдиного розв'язку. Але слід пам'ятати, що таке рішення отримано за допомогою нормування критеріїв, тобто за допомогою суб'єктивної інформації.

4. З використанням методу послідовних поступок також можна отримати єдине ефективне рішення, але таке рішення буде результатом заздалегідь визначеної похибки.

Контрольні запитання

1. Який тип задач відносяться до багатокритеріальних?
2. Навести формальний запис моделі багатокритеріальної задачі.
3. Чим багатокритеріальна задача відрізняється від задач лінійного програмування?
4. В чому полягає невизначеність при вирішенні багатокритеріальних задач?
5. Які існують способи усунення невизначеності при вирішенні багатокритеріальних задач?
6. Який розв’язок можна назвати ефективним рішенням багатокритеріальної задачі?
7. Як називається множина ефективних рішень багатокритеріальної задачі?
8. Що називають ефективною оцінкою багатокритеріальної задачі?
9. Що дозволяє отримати множина розв'язків за Парето?
10. В чому полягає недолік принципу Парето?
11. На які групи поділяються методи вирішення задач багатокритеріальної оптимізації.
12. Що таке субоптимальний розв'язок багатокритеріальної задачі?
13. Які особливості методу згортання критеріїв?
14. Які особливості методу послідовних поступок?
15. Чи можна розв'язувати задачі багатокритеріальної оптимізації використовуючи методи лінійного програмування?
16. В чому полягає метод згортання критерії?

РОЗДІЛ 12

Удосконалення техніко-технологічних параметрів транспортних систем з використанням векторної оптимізації

12.1 Теоретичні відомості

Однією з основних проблем, що виникає при керуванні складними системами, до яких відносяться залізничні станції, є проблема удосконалення техніко-технологічних параметрів функціонування об'єкту при реалізації того чи іншого управлінського рішення [25 – 27].

При формалізації задачі станція або окрема її підсистема (парк) розглядається як складна система, яка складається з множини елементів, до яких можна віднести колійний розвиток, систему обслуговування і т. д. Комплекс заходів γ_i, направлених на удосконалення техніко-технологічних параметрів станції в загальному випадку передбачає проведення робіт на кожному такому елементі:

$$\gamma_i = \{\Theta_1, \Theta_2, \ldots \Theta_j \ldots \Theta_n\}, \tag{12.1}$$

де Θ_j – множина можливих заходів на j-му елементі станції (реконструкція горловин, зміна кількості колій, кількості виконавців технологічних операцій);

n – загальна кількість елементів станції, для яких планується модернізація.

На реалізацію кожного із заходів Θ_{ij} комплексу γ_i потрібні певні витрати коштів $C(\Theta_{ij})$. З іншого боку, реалізація заходу Θ_{ij} дозволяє отримати певне значення показника його ефективності $E(\Theta_{ij})$. Конкретні значення зазначених показників можуть бути отримані за результатами моделювання роботи станції. Таким чином, сумарні витрати коштів по відповідному варіанту γ_i визначаються як:

$$C(\gamma_i) = \sum C(\Theta_{ij}), \;\; \Theta_{ij} \in \gamma_i \tag{12.2}$$

Значення показника ефективності кожного варіанту складе:

$$E(\gamma_i) = \sum E(\Theta_{ij}), \quad \Theta_{ij} \in \gamma_i \tag{12.3}$$

Набір можливих варіантів складає безліч Г = {γ_1, γ_2, … γ_k }.

Рішенням задачі є така підмножина $\Gamma^* = \{\gamma_1^*, \gamma_2^*, \ldots \gamma_m^*\}$, для кожного з елементів якої виконується умова:

$$\begin{pmatrix} C(\gamma^*), \\ -E(\gamma^*) \end{pmatrix} \to \min \tag{12.4}$$

Вказане завдання у викладеній постановці є задачею векторної оптимізації. Результатом рішення є підмножина точок, кожна з яких при заданій величині витрат $C(\gamma^*)$ визначає найбільш ефективний комплекс заходів γ^*. При цьому отримана підмножина точок Γ^* включає лише так звані ефективні рішення, кожне з яких передбачає поліпшення значення якісного показника при збільшенні витрат.

При постановці завдання визначення раціонального комплексу заходів, спрямованих на вдосконалення конструкції і технології роботи парку прийому станції, кожен варіант серед множини організаційно-технічних рішень може оцінювався за такими показниками:

$C(\gamma_i)$ – витрати на реалізацію певного комплексу заходів, *тис.грн./рік*;

$T(\gamma_i)$ – величина середнього простою одного поїзда, яка може бути досягнута при реалізації варіанту γ_i, *хв./поїзд*.

При цьому завдання (12.4) представляється у вигляді:

$$\begin{pmatrix} C(\gamma^*), \\ T(\gamma^*) \end{pmatrix} \to \min \tag{12.5}$$

за умови, що кожен з елементів $\gamma_i \in \Gamma$ являє собою множину можливих заходів (12.1).

Будь-який варіант організаційно-технічних рішень $\gamma^* \in \Gamma$ будемо називати ефективним, якщо будь-яке відхилення від нього призводить до погіршення хоча б одного з показників, тобто до збільшення величини простою поїздів, або збільшення величини витрат. Під рішенням задачі (12.5) будемо розуміти деякий набір варіантів $\Gamma^* \in \Gamma$, в якому будь-який варіант є ефективним.

12.2 Приклад розв'язання задач. Задача 15

Умова задачі

Для парку прийому деякої сортувальної станції характерне таке:

– кількість колій у парку $N_{кол}$;

– корисна довжина колій становить $l_{кор}$ метрів.

– величина поїздопотоку складає N поїздів за добу;

– математичне очікування кількості вагонів у складі поїзду становить $M[m]$ вагонів;

– математичне очікування тривалості обробки одного вагону становить $M[\tau_{об}]$ хвилин на вагон;

– тривалість заїзду маневрового локомотиву під состав $t_з$ хвилин;

– тривалість насуву состава на гірку становить $t_н$ хвилин;

– швидкість розпуску $V_{роз}$ кілометрів за годину.

Слід визначити оптимальні техніко-технологічні параметри парку прийому сортувальної станції, при яких одночасно забезпечуються:

1. мінімальна тривалість простою поїздів у парку $\Sigma T_{зан} \rightarrow \min$;

2. мінімально можливі витрати на реалізацію певного варіанта техніко-технологічних рішень $\Sigma C \rightarrow \min$.

Для досягнення цього необхідно:

1. Вирішити задачу векторної оптимізації графічно за умови, що максимальна сума витрачених коштів не повинна перевищувати C_{max}.

2. Вирішити задачу за допомогою методу згортки критеріїв для прийнятих значень λ_1 та λ_2.

3. Вирішити задачу за допомогою методу послідовних поступок для прийнятого значення Δ_2.

Вихідні дані

Значення наведених в умові задачі параметрів задані в таблиці 12.1.

Таблиця 12.1

Значення параметрів для прикладу

$N_{кол}$	$l_{кор}$, *м*	N, *п-в/добу*	$M[m]$, *ваг*	$M[\tau_{об}]$, *хв/ваг*	$t_з$, *хв*	$t_н$, *хв*	$V_{роз}$, *км/год*	C_{max}, *тис. у. о.*	λ_1	λ_2	Δ_2, %
4	850	60	50	0,9	6,0	6,0	3,6	350	0,6	0,4	4

Розв’язання задачі

Множина можливих варіантів розв'язку поставленої задачі може бути досягнута за умови варіювання такими техніко-технологічними параметрами парку.

1. Кількість колій в парку будемо варіювати в межах [$N_{кол}$; $N_{кол}$+1], тобто W_1={4; 5}. Тут варіант W_1 = 4 відповідає існуючій кількості колій в парку, а варіант W_1 = 5 – передбачає добудову однієї колії для прийому поїздів.

2. Кількість локомотивів, які задіяні при розформуванні составів, будемо варіювати в межах W_2={1; 2}.

3. Кількість бригад ПТО у парку W_3={1; 2}.

4. Кількість груп оглядачів вагонів в бригаді ПТО W_4={1; 2; 3; 4}.

Таким чином, була отримана підмножина γ_i, яка складається з 32 різних варіантів, кожен з яких являє собою комбінацію вказаних вище заходів W_k (див. таблицю 12.2).

Таблиця 12.2

Визначення множини техніко-технологічних параметрів функціонування парку станції

№	Множина γ_i				№	Множина γ_i			
	W_1 ($N_{кол}$)	W_2 ($K_{лок}$)	W_3 ($S_{бр}$)	W_4 ($K_{гр}$)		W_1 ($N_{кол}$)	W_2 ($K_{лок}$)	W_3 ($S_{бр}$)	W_4 ($K_{гр}$)
1	4	1	1	1	17	5	1	1	1
2	4	1	1	2	18	5	1	1	2
3	4	1	1	3	19	5	1	1	3
4	4	1	1	4	20	5	1	1	4
5	4	1	2	1	21	5	1	2	1
6	4	1	2	2	22	5	1	2	2
7	4	1	2	3	23	5	1	2	3
8	4	1	2	4	24	5	1	2	4
9	4	2	1	1	25	5	2	1	1
10	4	2	1	2	26	5	2	1	2
11	4	2	1	3	27	5	2	1	3
12	4	2	1	4	28	5	2	1	4
13	4	2	2	1	29	5	2	2	1
14	4	2	2	2	30	5	2	2	2
15	4	2	2	3	31	5	2	2	3
16	4	2	2	4	32	5	2	2	4

Отже, була отримана вихідна множина варіантів **Г** = {γ_1, γ_2, … γ_{32} }, з якої необхідно вибрати підмножину ефективних рішень, що

забезпечує такі рішення, які одночасно задовольняють обидва компоненти векторного критерію (12.5).

Для отримання достовірних результатів щодо функціонування парку прийому при реалізації певного варіанту організаційно-технічних рішень необхідно виконувати дослідження з використанням функціональної моделі парку.

Але з метою засвоєння викладеної вище методики векторної оптимізації будемо користуватися спрощеною процедурою визначення показників функціонування.

Тривалість простою поїздів у парку будемо визначати за виразом:

$$\sum T_{\text{зан}} = T_{\text{обс}} + T_{\text{розф}} + \sum t_{\text{з}}\,, \tag{12.6}$$

де $T_{\text{обс}}$ – тривалість обслуговування составу бригадою ПТО;

$T_{\text{розф}}$ – тривалість розформування составу;

$\Sigma t_{\text{з}}$ – сумарна тривалість затримок, пов'язаних з міжопераційними простоями в парку.

Тривалість обслуговування составу бригадою ПТО визначимо так:

$$T_{\text{обс}} = \frac{1}{K_{\text{гр}}} M[\tau_{\text{обс}}] \cdot M[m] \tag{12.7}$$

де $K_{\text{гр}}$ – кількість груп оглядачів у бригаді ПТО;

$M\,[\tau_{\text{обс}}]$ – математичне очікування тривалості обробки одного вагону;

$M\,[m]$ – математичне очікування кількості вагонів у складі поїзду.

Тривалість розформування составу для умов задачі будемо визначати за таким виразом:

$$T_{\text{розф}} = \frac{t_{\text{з}}}{K_{\text{лок}}} + t_{\text{н}} + t_{\text{р}} + t_{\text{ос}} \tag{12.8}$$

де $t_{\text{з}}$ – тривалість заїзду маневрового локомотиву під состав; згідно з вихідними даними $t_{\text{з}} = 6$ *хв.*;

$K_{\text{лок}}$ – кількість маневрових локомотивів;

$t_{\text{н}}$ – тривалість насуву состава на гірку; згідно з вихідними даними $t_{\text{н}} = 5$ *хв*;

$t_{\text{р}}$ – тривалість розпуску состава;

$t_{\text{ос}}$ – тривалість осаджування составів у сортувальному парку.

Тривалість розпуску состава будемо визначати за таким виразом:

$$t_{\text{р}} = \frac{0{,}06 \cdot l_{\text{ваг}} \cdot M[m]}{V_{\text{роз}}} \cdot \left(1 - \frac{1}{2g}\right), \tag{12.9}$$

де $l_{ваг}$ – довжина вагона, приймаємо $l_{ваг}$ = 15 *м*;

$V_{роз}$ – швидкість розпуску; згідно з вихідними даними $V_{роз}$ = 3,6 *км/год*;

g – кількість відчепів у составі; для умов задачі $g = 0{,}8 \cdot M[m]$.

Тривалість осаджування составів у сортувальному парку знайдемо за виразом:

$$t_{ос} = 0{,}06 \cdot M[m] \qquad (12.10)$$

В свою чергу сумарну тривалість затримок, пов'язаних з міжопераційними простоями в парку, для умов задачі можна розрахувати за приблизним виразом:

$$\sum t_з = 78{,}4275 - 4{,}4875 N_{кол} - 2{,}3614 S_{бр} - 12{,}5475 K_{гр} + 0{,}2123 N_{кол} \cdot S_{бр} \cdot K_{гр} \qquad (12.11)$$

Виконаємо розрахунки за наступних параметрів:

$$N_{кол} = 4;\ K_{лок} = 1;\ S_{бр} = 1;\ K_{гр} = 1.$$

Тривалість обслуговування составу складатиме:

$$T_{обс} = \frac{1}{1} 0{,}9 \cdot 50 = 45{,}0\ хв.$$

Тривалість розпуску состава складатиме:

$$t_р = \frac{0{,}06 \cdot 15 \cdot 50}{3{,}6} \cdot \left(1 - \frac{1}{2 \cdot 50 \cdot 0{,}8}\right) = 12{,}3\ хв.$$

Тривалість осаджування составів складатиме:

$$t_{ос} = 0{,}06 \cdot 50 = 3{,}0\ хв.$$

Отже, тривалість розформування складатиме:

$$T_{розф} = \frac{6}{1} + 5 + 12{,}3 + 3{,}0 = 26{,}3\ хв.$$

При цьому величина міжопераційних простоїв складатиме:

$$\sum t_з = 78{,}4275 - 4{,}4875 \cdot 50 - 2{,}3614 \cdot 1 - 12{,}5475 \cdot 1 + 0{,}2123 \cdot 50 \cdot 1 \cdot 1 = 46{,}4\ хв.$$

Таким чином, тривалість простою поїздів у парку складатиме:

$$\sum T_{зан} = 45{,}0 + 26{,}3 + 46{,}4 = 117{,}7\ хв.$$

Розрахунки для решти варіантів виконуються аналогічно та заносяться до таблиці (див. таблицю 12.3)

Таблиця 12.3

Визначення множини техніко-технологічних параметрів функціонування парку станції

№	$N_{кол}$	$K_{лок}$	$S_{бр}$	$K_{гр}$	$T_{обс}$	$t_р$	$t_{ос}$	$T_{розф}$	$\sum t_з$	$\sum \boldsymbol{T}_{зан}$
1	4	1	1	1	45,0	12,3	3,0	26,3	46,4	117,7
2	4	1	1	2	22,5	12,3	3,0	26,3	34,7	83,5
3	4	1	1	3	15,0	12,3	3,0	26,3	23,0	64,3
4	4	1	1	4	11,3	12,3	3,0	26,3	11,3	48,9
5	4	1	2	1	45,0	12,3	3,0	26,3	44,9	116,2
6	4	1	2	2	22,5	12,3	3,0	26,3	34,1	82,9
7	4	1	2	3	15,0	12,3	3,0	26,3	23,2	64,5
8	4	1	2	4	11,3	12,3	3,0	26,3	12,4	49,9
9	4	2	1	1	45,0	12,3	3,0	23,3	46,4	114,7
10	4	2	1	2	22,5	12,3	3,0	23,3	34,7	80,5
11	4	2	1	3	15,0	12,3	3,0	23,3	23,0	61,3
12	4	2	1	4	11,3	12,3	3,0	23,3	11,3	45,9
13	4	2	2	1	45,0	12,3	3,0	23,3	44,9	113,2
14	4	2	2	2	22,5	12,3	3,0	23,3	34,1	79,9
15	4	2	2	3	15,0	12,3	3,0	23,3	23,2	61,5
16	4	2	2	4	11,3	12,3	3,0	23,3	12,4	46,9
17	5	1	1	1	45,0	12,3	3,0	26,3	42,1	113,4
18	5	1	1	2	22,5	12,3	3,0	26,3	30,7	79,5
19	5	1	1	3	15,0	12,3	3,0	26,3	19,2	60,5
20	5	1	1	4	11,3	12,3	3,0	26,3	7,7	45,2
21	5	1	2	1	45,0	12,3	3,0	26,3	40,8	112,1
22	5	1	2	2	22,5	12,3	3,0	26,3	30,4	79,2
23	5	1	2	3	15,0	12,3	3,0	26,3	20,0	61,3
24	5	1	2	4	11,3	12,3	3,0	26,3	9,6	47,1
25	5	2	1	1	45,0	12,3	3,0	23,3	42,1	110,4
26	5	2	1	2	22,5	12,3	3,0	23,3	30,7	76,5
27	5	2	1	3	15,0	12,3	3,0	23,3	19,2	57,5
28	5	2	1	4	11,3	12,3	3,0	23,3	7,7	42,2
29	5	2	2	1	45,0	12,3	3,0	23,3	40,8	109,1
30	5	2	2	2	22,5	12,3	3,0	23,3	30,4	76,2
31	5	2	2	3	15,0	12,3	3,0	23,3	20,0	58,3
32	5	2	2	4	11,3	12,3	3,0	23,3	9,6	44,1

Тепер необхідно оцінити витрати на реалізацію певного варіанту функціонування парку.

Для умов задачі загальну суму витрат (ΣC) можна знайти як суму окремих складових витрат, пов’язаних з: придбанням та утриманням

маневрових локомотивів ($C_{лок}$); утриманням бригади ПТО ($C_{пто}$); спорудженням та утриманням колій парку прийому ($C_{кол}$); укладкою та утриманням стрілочних переводів ($C_{стр}$); тривалістю знаходження вагонів у підсистемі розформування ($C_{пс}$):

$$\Sigma C = C_{пто} + C_{лок} + C_{кол} + C_{стр} + C_{пс}, \qquad (12.12)$$

Складові частини формули (12.12) можна знайти так:

$$C_{пто} = 4{,}5\ e_{гр}\ S_{бр}\ K_{гр}, \qquad (12.13)$$

$$C_{лок} = e_{лок}\ K_{лок}, \qquad (12.14)$$

$$C_{кол} = e_{кол}\ N_{кол}\ l_{кор}\ k\ 10^{-3}, \qquad (12.15)$$

$$C_{стр} = e_{стр}\ K_{стр}, \qquad (12.16)$$

$$C_{пс} = 365\frac{\sum T_{зан}}{60} e_{вг} \cdot N \cdot M[m], \qquad (12.17)$$

де e – витратна ставка на одиницю відповідного вимірника;

$l_{кор}$ – корисна довжина колії, *м*;

k – коефіцієнт відношення повної довжини колій до корисної; $k = 1{,}1$;

$K_{стр}$ – кількість стрілочних переводів;

N – кількість поїздів, які прибувають у парк за добу; згідно з вихідними даними величина поїздопотоку складає $N = 60$ *поїздів/добу*;

Кількість стрілочних переводів можна знайти як $K_{стр} = 2(N_{кол} - 1)$.

Величини витратних ставок приймаємо такі:

$e_{гр} = 7{,}2$ *тис. у.о.*; $e_{лок} = 44{,}3$ *тис. у.о.*; $e_{кол} = 10{,}7$ *тис. у.о.*;

$e_{стр} = 2{,}4$ *тис. у.о.*; $e_{вг} = 0{,}0001$ *тис. у.о.*;

Виконаємо розрахунки при таких параметрах:

$$N_{кол} = 4;\ K_{лок} = 1;\ S_{бр} = 1;\ K_{гр} = 1.$$

$$C_{пто} = 4{,}5 \cdot 7{,}2 \cdot 1 \cdot 1 = 32{,}4 \text{ тис. у.о.};$$

$$C_{лок} = 44{,}3 \cdot 1 = 44{,}3 \text{ тис. у.о.};$$

$$C_{кол} = 10{,}7 \cdot 4 \cdot 850 \cdot 1{,}1 \cdot 10^{-3} = 40{,}02 \text{ тис. у.о.};$$

$$K_{стр} = 2(4 - 1) = 6 \text{ стр. пер.};$$

$$C_{стр} = 2{,}7 \cdot 6 = 16{,}2 \text{ тис. у.о.};$$

$$C_{пс} = 365\frac{117{,}7}{60} \cdot 0{,}0001 \cdot 60 \cdot 50 = 214{,}83 \text{ тис. у.о.};$$

Таким чином, загальна величина витрат складатиме:

$$\Sigma C = 32{,}4 + 44{,}3 + 40{,}02 + 16{,}23 + 214{,}83 = 347{,}75 \text{ тис. у.о.}$$

Розрахунки для решти варіантів виконуються в таблиці 12.4.

Таблиця 12.4

Визначення множини економічних параметрів

№	$N_{кол}$	$K_{лок}$	$S_{бр}$	$K_{гр}$	$C_{пто}$	$C_{лок}$	$C_{кол}$	$K_{стр}$	$C_{стр}$	$C_{пс}$	$\sum C$
1	4	1	1	1	32,4	44,3	40,02	6	16,2	214,83	347,75
2	4	1	1	2	64,8	44,3	40,02	6	16,2	152,42	317,74
3	4	1	1	3	97,2	44,3	40,02	6	16,2	117,39	315,11
4	4	1	1	4	129,6	44,3	40,02	6	16,2	89,19	319,31
5	4	1	2	1	64,8	44,3	40,02	6	16,2	212,08	377,4
6	4	1	2	2	129,6	44,3	40,02	6	16,2	151,21	381,33
7	4	1	2	3	194,4	44,3	40,02	6	16,2	117,73	412,65
8	4	1	2	4	259,2	44,3	40,02	6	16,2	91,08	450,8
9	4	2	1	1	32,4	88,6	40,02	6	16,2	209,36	386,58
10	4	2	1	2	64,8	88,6	40,02	6	16,2	146,95	356,57
11	4	2	1	3	97,2	88,6	40,02	6	16,2	111,91	353,93
12	4	2	1	4	129,6	88,6	40,02	6	16,2	83,72	358,14
13	4	2	2	1	64,8	88,6	40,02	6	16,2	206,6	416,22
14	4	2	2	2	129,6	88,6	40,02	6	16,2	145,74	420,16
15	4	2	2	3	194,4	88,6	40,02	6	16,2	112,25	451,47
16	4	2	2	4	259,2	88,6	40,02	6	16,2	85,61	489,63
17	5	1	1	1	32,4	44,3	50,02	8	21,6	207,03	355,35
18	5	1	1	2	64,8	44,3	50,02	8	21,6	145,01	325,73
19	5	1	1	3	97,2	44,3	50,02	8	21,6	110,36	323,48
20	5	1	1	4	129,6	44,3	50,02	8	21,6	82,55	328,07
21	5	1	2	1	64,8	44,3	50,02	8	21,6	204,66	385,38
22	5	1	2	2	129,6	44,3	50,02	8	21,6	144,57	390,09
23	5	1	2	3	194,4	44,3	50,02	8	21,6	111,86	422,18
24	5	1	2	4	259,2	44,3	50,02	8	21,6	85,99	461,11
25	5	2	1	1	32,4	88,6	50,02	8	21,6	201,56	394,18
26	5	2	1	2	64,8	88,6	50,02	8	21,6	139,53	364,55
27	5	2	1	3	97,2	88,6	50,02	8	21,6	104,88	362,3
28	5	2	1	4	129,6	88,6	50,02	8	21,6	77,08	366,9
29	5	2	2	1	64,8	88,6	50,02	8	21,6	199,19	424,21
30	5	2	2	2	129,6	88,6	50,02	8	21,6	139,1	428,92
31	5	2	2	3	194,4	88,6	50,02	8	21,6	106,39	461,01
32	5	2	2	4	259,2	88,6	50,02	8	21,6	80,52	499,94

Уся множина рішень задачі векторної оптимізації представлена у таблиці 12.5.

Таблиця 12.5

Множина рішень задачі векторної оптимізації

№	$N_{кол}$	$K_{лок}$	$S_{бр}$	$K_{гр}$	$\sum T_{зан}$	$\sum C$	№	$N_{кол}$	$K_{лок}$	$S_{бр}$	$K_{гр}$	$\sum T_{зан}$	$\sum C$
1	4	1	1	1	117,7	347,75	17	5	1	1	1	113,4	355,35
2	4	1	1	2	83,5	317,74	18	5	1	1	2	79,5	325,73
3	4	1	1	3	64,3	315,11	19	5	1	1	3	60,5	323,48
4	4	1	1	4	48,9	319,31	20	5	1	1	4	45,2	328,07
5	4	1	2	1	116,2	377,40	21	5	1	2	1	112,1	385,38
6	4	1	2	2	82,9	381,33	22	5	1	2	2	79,2	390,09
7	4	1	2	3	64,5	412,65	23	5	1	2	3	61,3	422,18
8	4	1	2	4	49,9	450,80	24	5	1	2	4	47,1	461,11
9	4	2	1	1	114,7	386,58	25	5	2	1	1	110,4	394,18
10	4	2	1	2	80,5	356,57	26	5	2	1	2	76,5	364,55
11	4	2	1	3	61,3	353,93	27	5	2	1	3	57,5	362,30
12	4	2	1	4	45,9	358,14	28	5	2	1	4	42,2	366,90
13	4	2	2	1	113,2	416,22	29	5	2	2	1	109,1	424,21
14	4	2	2	2	79,9	420,16	30	5	2	2	2	76,2	428,92
15	4	2	2	3	61,5	451,47	31	5	2	2	3	58,3	461,01
16	4	2	2	4	46,9	489,63	32	5	2	2	4	44,1	499,94

Кінцевою метою вирішення зазначеного завдання є виділення з вихідної множини **Г** підмножини **Г***, яка включає тільки ефективні варіанти організаційно-технічних заходів.

Така підмножина наведена у таблиці 12.6 та на рисунку 12.1.

Знайдемо ефективний розв'язок (1) задачі векторної оптимізації за умови, що максимальна сума витрачених коштів не повинна перевищувати $C_{max} = 350$ *тис. у.о.*

Таблиця 12.6

Множина ефективних за Парето рішень

№	$N_{кол}$	$K_{лок}$	$S_{бр}$	$K_{гр}$	$\sum T_{зан}$	$\sum C$
3	4	1	1	3	64,3	315,11
4	4	1	1	4	48,9	319,31
20	5	1	1	4	45,2	328,07
28	5	2	1	4	42,2	366,90

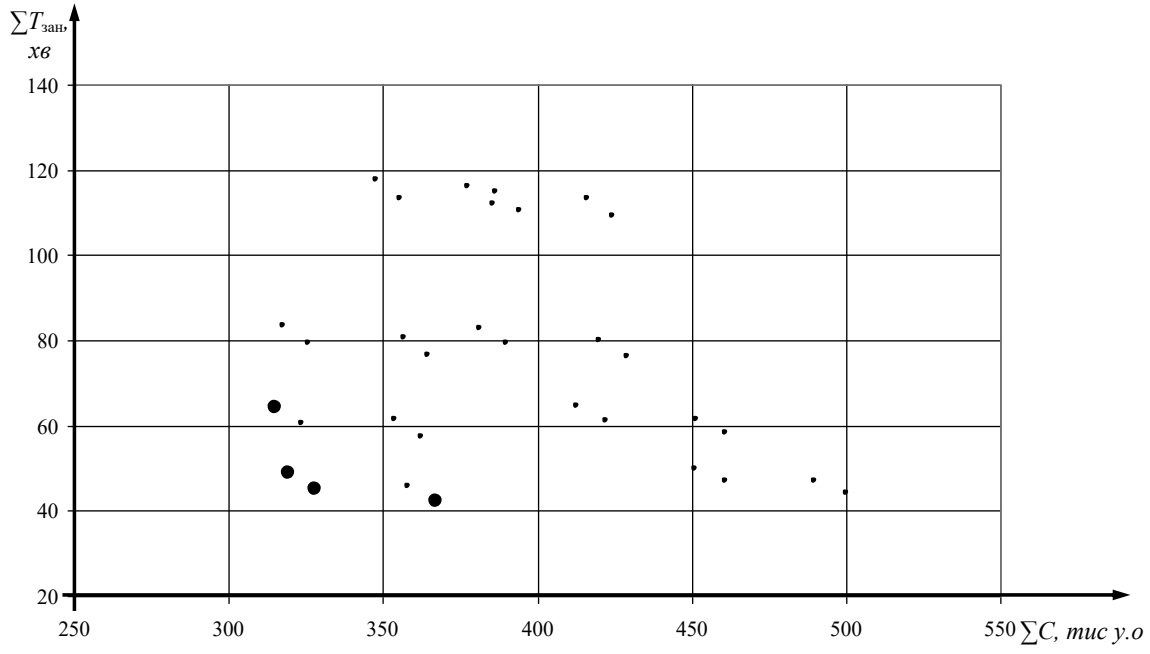


Рисунок 12.1 – Множина ефективних за Парето рішень у векторній площині

За такої умови найкращим з погляду мінімізації першого критерію ($\sum T_{зан}$ = 42,2 *хв*.) рішення буде за таких параметрів:

$$N_{кол} = 5;\ K_{лок} = 2;\ S_{бр} = 1;\ K_{гр} = 4.$$

Знайдемо ефективний розв'язок (2) задачі за допомогою методу згортки критеріїв при λ_1= 0,6 та λ_2=0,4.

За такої умови найкраще рішення буде за таких параметрів:

$$N_{кол} = 4;\ K_{лок} = 1;\ S_{бр} = 1;\ K_{гр} = 4.$$

Знайдемо ефективний розв'язок (3) задачі за допомогою методу послідовних поступок при Δ_2=4%.

Для цього серед усіх рішень (1) обираємо оптимальне за критерієм $\sum C$ рішення ($\sum C$ = 315,11 *тис. у.о*.).

Розраховуємо величину можливої поступки:

$$\Delta_2 = 0{,}04 \cdot 315{,}11 = 12{,}6\ тис.\ у.о.$$

Таким чином, величина витрат не повинна перевищувати

$$\sum C + \Delta_2 = 315{,}11 + 12{,}60 = 327{,}71\ тис.\ у.о.$$

За такої умови найкраще рішення буде за таких параметрів:

$$N_{кол} = 4;\ K_{лок} = 1;\ S_{бр} = 1;\ K_{гр} = 4.$$

Варіанти розв'язку задачі за допомогою різних методів оптимізації наведено у таблиці 12.7.

Таблиця 12.7

Варіанти розв'язку задачі

№	Техніко-технологічні параметри парку				Множина ефективних рішень		Варіанти розв'язку задачі					
							(1)		(2)	(3)		
	$N_{кол}$	$K_{лок}$	$S_{бр}$	$K_{гр}$	$\sum T_{зан}$	$\sum C$	$\sum T_{зан}$	$\sum C$	TC	$\sum C+\Delta_2$	$\sum C'_{зан}$	$\sum T'_{зан}$
3	4	1	1	3	64,3	315,11	64,3	315,11	164,64	346,62	315,11	64,3
4	4	1	1	4	48,9	319,31	48,9	319,31	**157,05**	346,62	319,31	48,9
20	5	1	1	4	45,2	328,07	**45,2**	**328,07**	158,37	346,62	**328,07**	**45,2**
28	5	2	1	4	42,2	366,90	-	-	172,10	346,62	-	-

Варіанти розв'язку задачі в графічному вигляді у векторній площині наведені на рисунку 12.2.

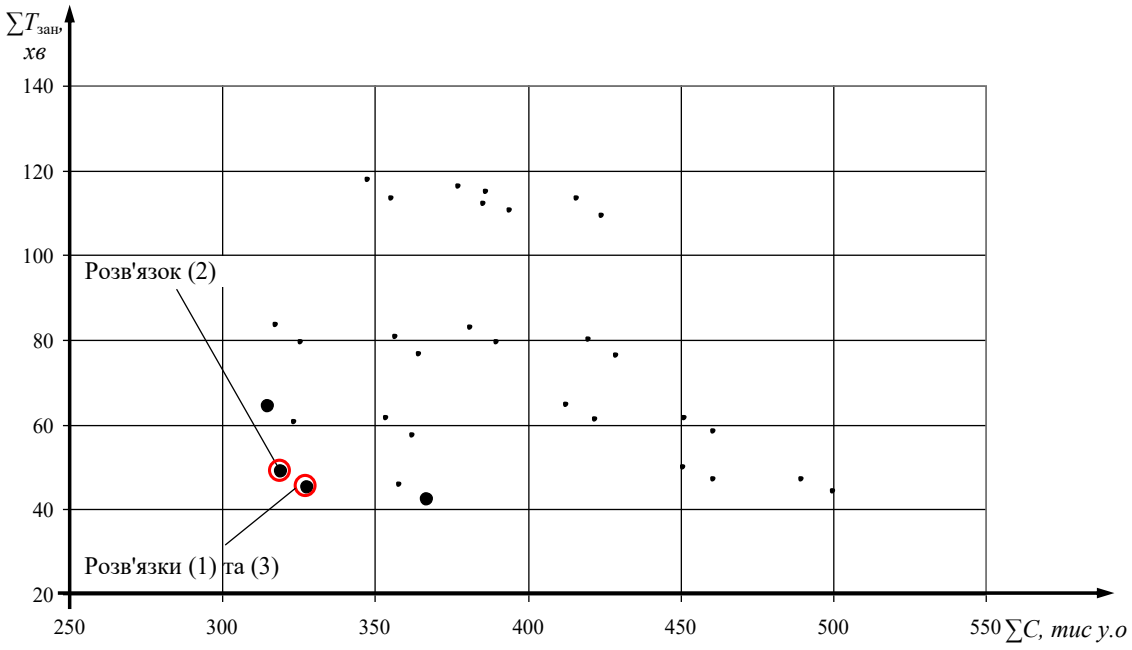


Рисунок 12.2 – Множина ефективних за Парето рішень у векторній площині з можливими варіантами розв'язку

Для визначення раціональних варіантів організаційно-технічних заходів, що забезпечують максимальну ефективність роботи парку станції в залежності від обсягів фінансування проекту, поставлене завдання вдосконалення техніко-технологічних параметрів парку станції було вирішено як задача векторної оптимізації.

Можливі варіанти розв'язку цієї задачі наведено на рисунку 12.2.

Контрольні запитання

1. Що називається задачею векторної оптимізації та чому?

2. Чим задача векторної оптимізації відрізняється від задач лінійного програмування?

3. Чим задача векторної оптимізації відрізняється від багатокритеріальної задачі?

4. Які дві цільові функції слід оптимізувати при вирішенні задачі векторної оптимізації на прикладі Задачі 15?

5. Як саме забезпечувалась оптимізація двох цільових функцій в Задачі 15?

6. Які вихідні дані потрібні для вирішення задачі векторної оптимізації на прикладі Задачі 15?

7. Що означає ефективне рішення у задачах векторної оптимізації?

8. Чому саме сортувальна станція є прикладом складної системи для векторної оптимізації?

9. У чому полягає метод згортки критеріїв і які значення λ_1 і λ_2 були використані в Задачі 15?

10. Як метод послідовних поступок визначає допустиму величину відхилення і яке рішення вважається найкращим для умов Задачі 15?

11. Що входить до вихідної множини можливих варіантів Г при вирішенні Задачі 15?

12. Які параметри в Задачі 15 змінювалися для формування множини техніко-технологічних рішень γ_i?

Література

1. Cascetta E. Transportation Systems Engineering: Theory and Methods. Springer Science & Business Media, 2001. 710 p.
2. Козаченко Д. М., Вернигора Р. В., Малашкін В. В. Основи дослідження операцій у транспортних системах: приклади та задачі : навч. посіб. Дніпропетровськ : ДНУЗТ, 2015. 277 с.
3. Ortúzar J. de D., Willumsen L. G. Modelling Transport. 5th ed. Hoboken (NJ) : Wiley-Blackwell, 2024. 720 p.
4. Методологія та організація наукових досліджень : навч. посіб. / Л. О. Єльнікова та ін. Дніпро : Укр. держ. ун-т науки і технологій, 2024. 228 с.
5. Law A. M. Simulation Modeling and Analysis. 6th ed. New York : McGraw-Hill, 2024. 704 p.
6. *ГБН В.2.3-37472062-2:2013. Службово-технічні будівлі і споруди станційно-вокзальних комплексів та зупинних пунктів залізничного транспорту. Проектування, будівництво*. Чинний від 2014-05-01. Київ : Міністерство інфраструктури України, 2013. 115 с.
7. Sen S., Kumar A. Design and Analysis of Algorithms: A Contemporary Perspective : textbook. Cambridge : Cambridge University Press, 2019. 367 p. DOI: https://doi.org/10.1017/9781108654937
8. Goodrich M. T., Tamassia R. Algorithm Design: Foundations, Analysis, and Internet Examples. New York : John Wiley & Sons, 2001. 720 p.
9. Russell S., Norvig P. Artificial Intelligence: A Modern Approach : textbook. 4th ed. Hoboken (NJ) : Pearson (NJ), 2020. 1136 c.
10. Handbook of Metaheuristics / ed. by M. Gendreau, J.-Y. Potvin. 3rd ed. Cham : Springer, 2019. 604 p. DOI: https://doi.org/10.1007/978-3-319-91086-4
11. Проектування станційних колій. Роз'їзди, обгінні пункти та проміжні станції: приклади та задачі : навч. посіб. / М. І. Березовий, М. П. Божко, В. В. Журавель, Є. Б. Демченко ; Дніпропетр. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна. Дніпро : Герда, 2017. 196 с.
12. Taha H. A. Operations Research: An Introduction : textbook. 10th ed. Boston : Pearson, 2017. 849 p.

13. Промисловий транспорт. Переробна спроможність вантажних фронтів: приклади та задачі : навч. посіб. / М. І. Березовий та ін. Дніпро : Укр. держ. ун-т науки і технологій, 2023. 135 с. DOI: https://doi.org/10.15802/978-617-8314-40-8

14. Козаченко Д. М., Коробйова Р. Г., Германюк Ю. М. Організація роботи сортувальної станції : навч. посіб. для ВНЗ. Львів : СПОЛОМ, 2023. 142 с.

15. Сковрон І. Я. Удосконалення технології та технічних засобів формування багатогрупних составів : автореф. дис. … канд. техн. наук : 05.22.20. Дніпропетровськ, 2015. 21 с.

16. Сковрон І. Я., Раджапова І. В., Москвітіна А. Р. Порівняльний аналіз технології формування багатогрупних составів на різних сортувальних пристроях. *Транспортні системи та технології перевезень*. 2024. Вип. 28. С. 43–51. DOI: https://doi.org/10.15802/tstt2024/312035

17. Сковрон І. Я., Демченко Є. Б., Дорош А. С., Малашкін В. В. Методи двостороннього формування багатогрупних составів. *Транспортні системи і технології перевезень* : зб. наук. пр. Дніпров. нац. ун-ту залізн. тр-ту ім. акад. В. Лазаряна. 2019. Вип. 18. С. 103–109. DOI: https://doi.org/10.15802/tstt2019/182615

18. Gutin G., Punnen A. P. The Traveling Salesman Problem and Its Variations. New York : Springer, 2002. 830 p. DOI: https://doi.org/10.1007/b101971

19. Stützle T., Hoos H. H. Stochastic Local Search: Foundations and Applications. San Francisco CA : Morgan Kaufmann, 2005. 658 p.

20. Dorigo M., Stützle T. Ant Colony Optimization : monohrafiia. Cambridge, MA : MIT Press, 2004. 319 p. DOI: https://doi.org/10.7551/mitpress/1290.001.0001

21. Коцюк О. Я., Бакуліч О. О., Мусатенко О. В. Оптимізація розподільчої системи доставки товарів. *Економіка та управління на транспорті*. 2016. Вип. 3. С. 32–41.

22. Сковрон І. Я., Дорош А. С., Демченко Є. Б., Болвановська Т. В., Малашкін В. В. Підвищення ефективності доставки збірних вантажів автомобільним транспортом. *Транспортні системи та технології перевезень*. 2020. Вип. 20. С. 36–44. DOI: https://doi.org/10.15802/tstt2020/217400

23. Іващенко Г. С., Онищенко О. І., Бондаренко М. Е., Здорик Н. В. Методи рішення задачі комівояжера на основі обчислювального інтелекту. *Системи управління, навігації та зв'язку*. 2024. Т. 2, № 76. С. 99–105. DOI: https://doi.org/10.26906/sunz.2024.2.099

24. Oliveira W., Fiorotto D., Song X., Jones D. An extended goal programming model for the multiobjective integrated lot-sizing and cutting stock problem. *European Journal of Operational Research*. 2021. Vol. 295, Iss. 3. P. 996–1007. DOI: https://doi.org/10.1016/j.ejor.2021.03.049

25. Herzel A., Ruzika S., Thielen C. Approximation Methods for Multi-objective Optimization Problems: A Survey. *INFORMS Journal on Computing*. 2021. Vol. 33, Iss. 4. P. 1284–1299. DOI: https://doi.org/10.1287/ijoc.2020.1028

26. Малярець Л. М., Мінєнкова О. В. Вирішення проблем багатокритеріальності в оцінці діяльності підприємства на основі методів багатокритеріальної оптимізації. *Проблеми економіки*. 2017. № 1. С. 421–427.

27. Anderluh A., Nolz P., Hemmelmayr V., Crainic T. G. Multi-objective optimization of a two-echelon vehicle routing problem with vehicle synchronization and ‘grey zone’ customers arising in urban logistics *European Journal of Operational Research*. 2021. Vol. 289, Iss. 3. P. 940–958. DOI: https://doi.org/10.1016/j.ejor.2019.07.049

Навчальне видання

Сковрон Ігор Ярославович, Дорош Андрій Сергійович,
Демченко Євген Борисович, Мазуренко Олександр Олександрович,
Малашкін Вячеслав Віталійович, Болвановська Тетяна Валентинівна

Методи оптимізації параметрів транспортних систем. Базові методи: приклади та задачі

Навчальний посібник

Електронне видання

Відповідальний редактор І. Я. Сковрон
Комп'ютерна верстка А. С. Дорош
Дизайн обкладинки В. В. Малашкін

Зареєстровано НМВ УДУНТ (№ 1.827 від 13.06.2025)

Формат 60×84 1/16. Ум. друк. арк. 12,09. Обл.12,24
Зам. № 41

Видавець: Український державний університет науки і технологій
вул. Лазаряна, 2, ауд. 2216, ауд. 263 (наукова бібліотека)
м. Дніпро, 49010.
Свідоцтво суб'єкта видавничої справи ДК № 7709 від 14.12.2022

СКОВРОН
ІГОР ЯРОСЛАВОВИЧ

ДОРОШ
АНДРІЙ СЕРГІЙОВИЧ

ДЕМЧЕНКО
ЄВГЕН БОРИСОВИЧ

МАЗУРЕНКО
ОЛЕКСАНДР ОЛЕКСАНДРОВИЧ

МАЛАШКІН
ВЯЧЕСЛАВ ВІТАЛІЙОВИЧ

БОЛВАНОВСЬКА
ТЕТЯНА ВАЛЕНТИНІВНА