

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»
АВТОМОБІЛЬНО-ДОРОЖНІЙ ІНСТИТУТ

Кафедра «Прикладна математика і інформатика»

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИВЧЕННЯ ДИСЦИПЛІНИ «ДОСЛІДЖЕННЯ
ОПЕРАЦІЙ» (ДЛЯ СТУДЕНТІВ НАПРЯМУ 6.070101
«ТРАНСПОРТНІ ТЕХНОЛОГІЇ (ЗА ВИДАМИ
ТРАНСПОРТУ)». ЧАСТИНА 2**

22/15-2012-11

Горлівка – 2012

ДВНЗ «ДонНТУ» Автомобільно-дорожній інститут

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ»
АВТОМОБІЛЬНО-ДОРОЖНИЙ ІНСТИТУТ

«ЗАТВЕРДЖУЮ»
Директор АДІ ДВНЗ «ДонНТУ»
М. М. Чальцев
2012 р.

Кафедра «Прикладна математика і інформатика»

**МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИВЧЕННЯ ДИСЦИПЛІНИ «ДОСЛІДЖЕННЯ
ОПЕРАЦІЙ» (ДЛЯ СТУДЕНТІВ НАПРЯМУ 6.070101
«ТРАНСПОРТНІ ТЕХНОЛОГІЇ (ЗА ВИДАМИ
ТРАНСПОРТУ»)). ЧАСТИНА 2**

22/15-2012-11

«РЕКОМЕНДОВАНО»
Навчально-методична комісія
факультету «Транспортні
технології»
Протокол № 9 від 19.05.2011р.

«РЕКОМЕНДОВАНО»
кафедра «Прикладна
математика і інформатика»
Протокол № 23 від 11.05.2011р.

Горлівка – 2012

УДК 519.47 +512.98(071)

Методичні вказівки до вивчення дисципліни «Дослідження операцій» (для студентів напряму 6.070101 «Транспортні технології (за видами транспорту)»). Частина 2 [Електронний ресурс] / укладачі: М. Є. Корольов, Т. В. Непомняща. – Електрон. дані – Горлівка: ДВНЗ «ДонНТУ» АДІ, 2012. – 1 електрон. опт. диск (CD-R); 2 см. – Систем. вимоги: Pentium; 32 RAM;

WINDOWS 98/2000/NT/XP; MS Word 2000. – Назва з титул. екрану.

Вказівки відповідають розділам «Мережеві моделі», «Комбінаторні моделі», «Моделі керування запасами», «Теорія ігор», «Прийняття рішень в умовах невизначеності», «Марківські процеси», «Системи масового обслуговування» дисципліни «Дослідження операцій» для підготовки бакалаврів. Містять реальні моделі оптимального планування в транспортних галузях, об'єднаннях і підприємствах, індивідуальні завдання за основними розділами курсу. Методичні вказівки призначено для студентів третього курсу напряму 6.070101 «Транспортні технології (за видами транспорту)».

Укладачі:

Корольов М. Є., к.ф.-м.н., доц.
Непомняща Т. В.

Відповідальний за випуск:

Хребет В. Г., к.ф.-м.н., доц.

Рецензент:

Сокирко В. М., к.т.н., доц.
каф. «Транспортні технології»

© Державний вищий навчальний заклад
«Донецький національний технічний університет»
Автомобільно-дорожній інститут, 2012

ДВНЗ «ДонНТУ» Автомобільно-дорожній інститут

Корольов Марк Євгенович
Непомняща Тетяна Володимирівна

МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИВЧЕННЯ ДИСЦИПЛІНИ «ДОСЛІДЖЕННЯ
ОПЕРАЦІЙ» (ДЛЯ СТУДЕНТІВ НАПРЯМУ 6.070101
«ТРАНСПОРТНІ ТЕХНОЛОГІЇ (ЗА ВИДАМИ
ТРАНСПОРТУ)»). ЧАСТИНА 2

Підписано до друку 2012 р. Гарнітура Times New.
Умов. друк. арк. 8,4. Зам. №

Державний вищий навчальний заклад
«Донецький національний технічний університет»
Автомобільно-дорожній інститут
84646, м. Горлівка, вул. Кірова, 51
E-mail: drukfn@rambler.ru

Редакційно-видавничий відділ

Свідоцтво про внесення до Державного реєстру видавців, виготовників і розповсюджувачів видавничої продукції ДК № 2982 від 21.09.2007 р.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1 МЕРЕЖЕВІ МОДЕЛІ	11
1.1 Загальні відомості про мережеві моделі	11
1.2 Задача про максимальний потік.....	12
1.2.1 Приклад розв’язання задачі про максимальний потік	12
1.2.2 Індивідуальні завдання. Максимальний потік.....	21
1.3 Побудова мінімального остового дерева	31
1.4 Модель Флойда.....	34
1.5 Оптимальний розподіл товарообігу по мережі	43
1.6 Індивідуальні завдання (мережеві моделі)	47
РОЗДІЛ 2 КОМБІНАТОРНІ МОДЕЛІ. МОДЕЛЬ «БРОДЯЧОГО ТОРГОВЦЯ».....	62
2.1 Загальні відомості про задачу комівояжера	62
2.2 Реалізація моделі «бродячого торговця»	63
2.3 Індивідуальні завдання (модель «бродячого торговця»)	70
РОЗДІЛ 3 МОДЕЛІ КЕРУВАННЯ ЗАПАСАМИ	73
3.1 Основні поняття.....	73
3.2 Детерміновані моделі керування запасами	74
3.2.1 Реалізація детермінованої моделі керування запасами	75
3.2.2 Індивідуальні завдання. Детерміновані моделі керування запасами	76
3.3 Стохастичні моделі керування запасами	76
3.3.1 Реалізація стохастичної моделі керування запасами	76
3.3.2 Індивідуальні завдання. Стохастичні моделі керування запасами	79
3.4 Автоматизація моделі керування запасами	81
3.4.1 Приклад автоматизації моделі керування запасами.....	81
3.4.2 Індивідуальні завдання. Автоматизовані моделі керування запасами	83
РОЗДІЛ 4 ЕЛЕМЕНТИ ТЕОРІЇ ІГОР	85
4.1 Основні поняття теорії ігор	85
4.1.1 Загальні поняття теорії ігор	85
4.1.2 Платіжна матриця. Нижня й верхня ціна гри	87
4.1.3 Принцип мінімакса	88
4.2 Ігрові моделі 2x2. Розв’язання ігор у змішаних стратегіях.	

Графічне розв'язання гри із заданою платіжною матрицею	90
4.2.1 Загальні відомості.....	90
4.2.2 Індивідуальні завдання. Ігрові моделі 2x2.....	96
4.3 Ігрові моделі 2x <i>n</i>	97
4.3.1 Загальні відомості.....	97
4.3.2 Індивідуальні завдання. Ігрові моделі 2x <i>n</i>	100
4.4 Розв'язання матричних ігор методами лінійного програмування	102
4.4.1 Загальні відомості.....	102
4.4.2 Індивідуальні завдання для розв'язання матричних ігор методами лінійного програмування.....	107
РОЗДІЛ 5 ПРИЙНЯТТЯ РІШЕНЬ В УМОВАХ НЕВИЗНАЧЕНОСТІ ...	110
5.1 Загальні відомості.....	110
5.2 Критерії прийняття рішень в умовах невизначеності.....	111
5.2.1 Критерій Лапласа (принцип недостатнього обґрунтування)	111
5.2.2 Мінімаксний критерій (найкраще з найгіршого).....	111
5.2.3 Критерій Севіджа (жалю).....	112
5.2.4 Критерій Гурвіца (баланс між оптимізмом і песимізмом)...	113
5.2.5 Приклад застосування критеріїв прийняття рішення в умовах невизначеності.....	114
5.2.6 Індивідуальні завдання. Прийняття рішень в умовах невизначеності.....	116
РОЗДІЛ 6 МАРКІВСЬКІ ПРОЦЕСИ.....	126
6.1 Теоретичні відомості про марківські процеси.....	126
6.1.1 Поняття марківського випадкового процесу.....	126
6.1.2 Потоки подій.....	128
6.1.3 Рівняння Колмогорова. Граничні ймовірності станів	129
6.2 Практична реалізація дослідження марківських процесів.....	131
6.2.1 Приклад дослідження марківського процесу	131
6.2.2 Індивідуальні завдання. Марківські процеси	135
РОЗДІЛ 7 СИСТЕМИ МАСОВОГО ОБСЛУГОВУВАННЯ	140
7.1 Загальні відомості про системи масового обслуговування.....	140
7.2 Процес загибелі й розмноження	141
7.2.1 Теоретичні відомості.....	141
7.2.2 Приклад дослідження СМО (процес загибелі й розмноження)	

.....	143
7.2.3 Індивідуальні завдання. Процес загибелі і розмноження....	143
7.3 СМО з відмовами	146
7.3.1 Одноканальна система з відмовами	146
7.3.2 Багатоканальна система з відмовами.....	148
7.3.3 Індивідуальні завдання. СМО з відмовами	155
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ.....	156

ВСТУП

Сьогодні, у зв'язку з інтенсифікацією темпів технічного прогресу, відбувається ускладнення задач керування. Безумовно, ця тенденція стосується й транспортних систем. Зміни в реальних виробничих процесах і системах керування підштовхують розвиток науки, сприяють пошуку нових методів вивчення систем. Наука намагається допомогти органам, що займаються плануванням і управлінням, у прийнятті рішень.

Сукупність теорії та методів, які застосовуються для побудови моделей систем і подальшого їх вивчення, отримала назву *дослідження операцій*.

Дослідження операцій використовує ідеї багатьох інших наук, але найтісніше воно пов'язане з математичними методами, оскільки переважна більшість розділів прикладної математики виникла саме при вивченні різноманітних економічних систем і технологічних процесів.

Мережеве планування, комбінаторні моделі, керування запасами, теорія ігор, прийняття рішень в умовах невизначеності, системи масового обслуговування – одні з найважливіших розділів дослідження операцій. Кожен із них знаходить власне застосування в промисловості, має свої типи моделей і математичні методи дослідження.

У центрі уваги посібника знаходяться загальні принципи функціонування економічних і технічних систем різного ступеня складності; формалізація процесів у цих системах шляхом побудови моделей, які відображають структуру й динаміку взаємозв'язків у реальному об'єкті; проведення обчислювальних експериментів і інтерпретація результатів.

Посібник містить приклади задач технічного характеру, які описуються за допомогою математичної моделі задля отримання оптимального рішення, виявлення факторів, що мають найбільший вплив на результат. У кожному розділі методичних вказівок наведені реальні моделі оптимального планування в транспортних галузях, об'єднаннях і підприємствах. Для розвитку навичок самостійного розв'язування практичних задач у методичні вказівки включені індивідуальні завдання за основними розділами курсу. Запропонований до Вашої уваги навчальний посібник може використовуватися студентами при підготовці до практичних занять з дисципліни «Дослідження операцій», а також при підготовці до модульних контрольних робіт.

Бажаємо успіхів!

РОЗДІЛ 1 МЕРЕЖЕВІ МОДЕЛІ

1.1 Загальні відомості про мережеві моделі

У рамках теорії дослідження операцій розглядається велика кількість практичних задач, які можна сформулювати й вирішити як мережеві моделі. Наведемо декілька конкретних прикладів:

1. Проектування мережі міських доріг, що з'єднує всі райони міста. Цільова функція відповідної математичної моделі повинна мінімізувати вартість будівництва доріг.
2. Знаходження найкоротшого маршруту між двома містами в межах існуючої мережі доріг.
3. Визначення оптимальної пропускної здатності доріг міста.
4. Складання тимчасового графіка ремонтних робіт (визначення дат початку й завершення окремих етапів робіт).

Розв'язання наведених задач (як і багатьох інших подібних задач) вимагає застосування різних мережевих оптимізаційних алгоритмів. У цьому розділі будуть розглянуті такі алгоритми:

1. алгоритм знаходження максимального потоку;
2. алгоритм побудови мінімального остового дерева;
3. алгоритм відшукування найкоротшого шляху;
4. оптимальний розподіл товарообігу по мережі.

Задачі, що виникають із перерахованих прикладів, можна переформулювати й вирішувати як задачі лінійного програмування. Проте специфічна структура цих задач дозволила розробити спеціальні мережеві алгоритми, які є ефективнішими за стандартний симплекс - метод.

Основні означення теорії мережевих моделей

Мережа складається з множини *вузлів*, що пов'язані *дугами* (або *ребрами*). Таким чином, мережа описується парою множин (N, A) ,

де N – множина вузлів, а A – множина ребер. Наприклад, мережа, показана на рисунку 1.1, описується в такий спосіб:

$$N = \{1, 2, 3, 4, 5\},$$

$$A = \{(1,3), (1,2), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5)\}.$$

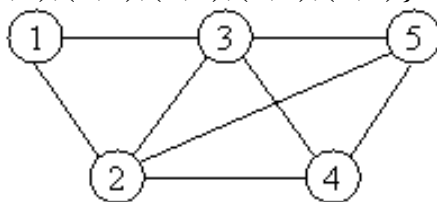


Рисунок 1.1 – Приклад мережі

З кожним типом мережі пов'язаний певний тип потоків (наприклад, транспортний потік нафти в нафтопроводах або автомобільні потоки в

мережі міських доріг). У загальному випадку потоки в мережі обмежені пропускною здатністю її ребер, що може бути як скінченною, так і нескінченною.

Ребро називається *спрямованим* або *орієнтованим* (і в цьому випадку ребро будемо називати *дугою*), якщо в одному напрямку можливий тільки додатній потік, а в протилежному – тільки нульовий. В *орієнтованій мережі* всі ребра орієнтовані.

Шляхом називається послідовність різних ребер, що з'єднують два вузли, незалежно від напрямку потоку в кожному ребрі. Шлях формує *цикл*, якщо початковий і кінцевий вузли збігаються. Наприклад, на рисунку 1.1 ребра (2,3), (3,4) і (4,2) складають цикл. *Орієнтований цикл* – це цикл, у якому всі дуги орієнтовані в одному напрямку.

1.2 Задача про максимальний потік

1.2.1 Приклад розв'язання задачі про максимальний потік

Постановка задачі: у межах підготовки до чемпіонату Євро-2012 проводиться модернізація (розширення) мережі доріг міста Донецька. Зокрема, було прийнято рішення збільшити пропускну здатність доріг з південного напрямку до стадіону «Донбас Арена», оскільки на шляху від перехрестя бульвару Шевченка з вулицею Челюскінців до перехрестя проспекту Миру з вулицею Челюскінців часто виникають затори. Кожна ділянка доріг, які планується реконструювати, має свою пропускну здатність (кількість автомобілів, які можуть проїхати в одиницю часу). Дороги можуть бути як з двостороннім, так і з одностороннім рухом. На ділянках доріг з одностороннім рухом передбачається додатна пропускну здатність в одному напрямку й нульова в іншому. Необхідно визначити максимальну пропускну здатність (максимальний потік) вулиць, які планується реконструювати. На карті (рис. 1.2) зображені напрямки руху транспорту на ділянках доріг, які планується реконструювати, а також пропускну здатність кожного напрямку.

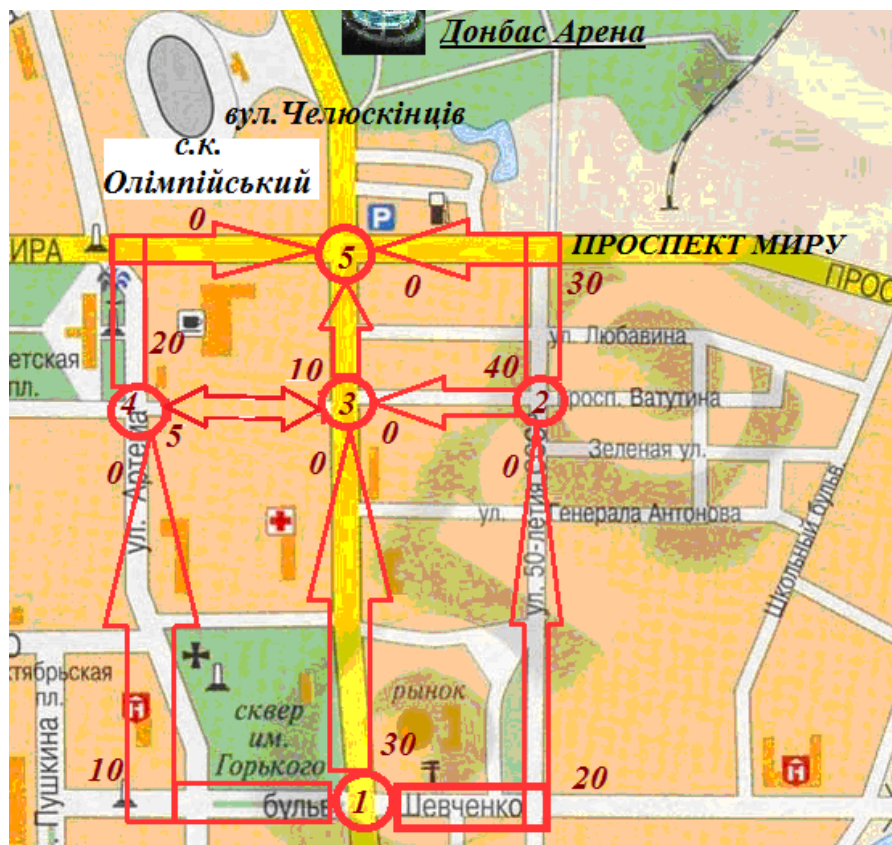


Рисунок 1.2 – Карта

Дані з карти (рис. 1.2) переносимо на мережу (рис. 1.3). На рисунку 1.3 при позначенні пропускних здатностей дотримувалися правила, що пропускна здатність із пункту i в пункт j розташовується на ребрі ij ближче до i , а пропускна здатність із пункту j в пункт i ближче до j .

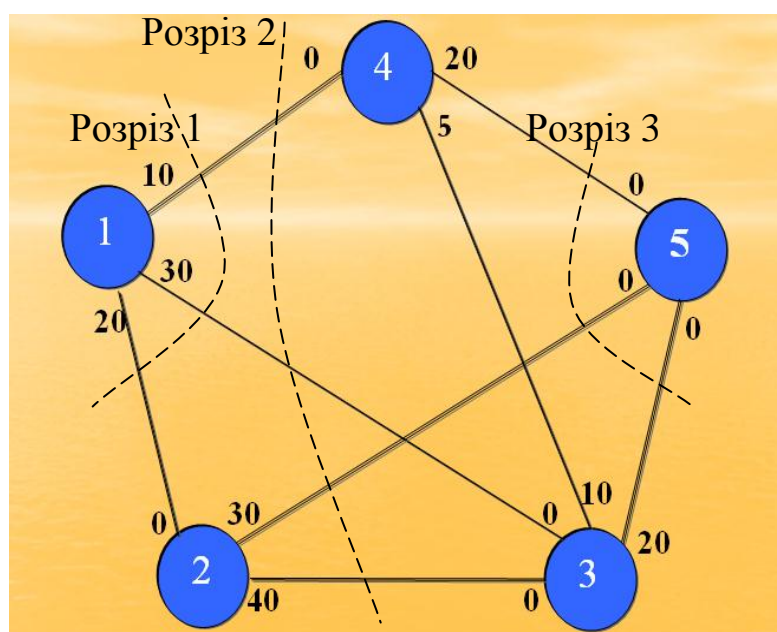


Рисунок 1.3 – Мережа

У наведеній задачі фактично потрібно знайти максимальний потік від початкового пункту (пункт 1, перетин бульвару Шевченка з вулицею Челюскінців) до кінцевого (пункт 5, зона стоянки біля стадіону «Донбас Арена»).

Перебір розрізів. Розріз визначається множиною ребер, які мають таку властивість: якщо всі ребра перекрити одночасно, то потік від джерела до стоку повністю припиняється. Пропускна здатність розрізу дорівнює сумі пропускних здатностей ребер, які «розрізаються». Серед усіх розрізів мережі розріз із мінімальною пропускною здатністю визначає максимальний потік у мережі.

Пропускна здатність розрізів, що зображені на рис. 1.3, представимо в таблиці.

Таблиця 1.1 – Пропускна здатність розрізів

Розріз	Ребра розрізу	Пропускна здатність
1	(1,2), (1,3), (1,4)	$10 + 30 + 20 = 60$
2	(1,3), (1,4), (2,3), (2,5)	$30 + 10 + 40 + 30 = 110$
3	(2,5), (3,5), (4,5)	$30 + 20 + 20 = 70$

Висновок, який можна зробити з цих трьох розрізів, полягає в тому, що максимальний потік не може перевищувати 60 машин. Проте ми не можемо сказати яким є максимальний потік насправді, оскільки не перебрали всі можливі розрізи мережі. Перебір усіх розрізів мережі є нелегкою справою. Сучасна наука не має універсального правила, за яким можна було б називати кількість розрізів для будь-якої мережі, не перебираючи їх. Тому для визначення максимального потоку в мережі не використовуються алгоритми, які базуються на повному переборі розрізів.

Алгоритм знаходження максимального потоку

Ідея даного алгоритму полягає у знаходженні сквозних шляхів від джерела до стоку.

У процесі виконання алгоритму частина пропускних здатностей «забирається» потоками, які проходять через дане ребро, внаслідок чого кожне ребро буде мати залишкову пропускну здатність. Мережа, в якій всі ребра мають залишкову пропускну здатність, називається залишковою.

Продемонструємо алгоритм для мережі, що зображена на рисунку 1.3.

Крок 1. Біля вузла 1 ставимо позначку $[\infty, -]$ (рис. 1.4). Це означає, що для нас не важливо яким шляхом автомобілі потрапляють у вузол 1. З

усіх шляхів, що виходять із вузла 1, обираємо шлях із найбільшою пропускну здатністю. У нашому випадку це шлях до вузла 3 (рис. 1.4).

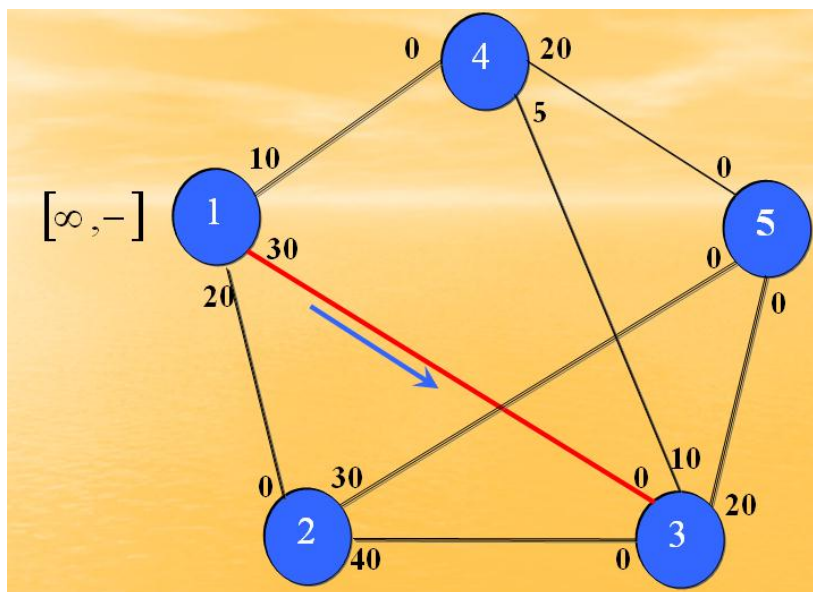


Рисунок 1.4 – Крок 1

Крок 2. Знаходимо шлях з найбільшою пропускну здатністю, перебираючи всі шляхи, що виходять із пункту 3. У нашому випадку це шлях у пункт 5 (рис. 1.5).

Біля вузлів мережі зазначаємо номер вузла, з якого ми потрапили в даний вузол і пропускну здатність відповідного ребра. Наприклад, ставлячи біля вузла 3 позначку $[30, 1]$ (рис. 1.5), ми зазначаємо, що у вузол 3 ми потрапили із вузлу 1 і пропускну здатність ребра мережі, яким ми рухалися, становить 30.

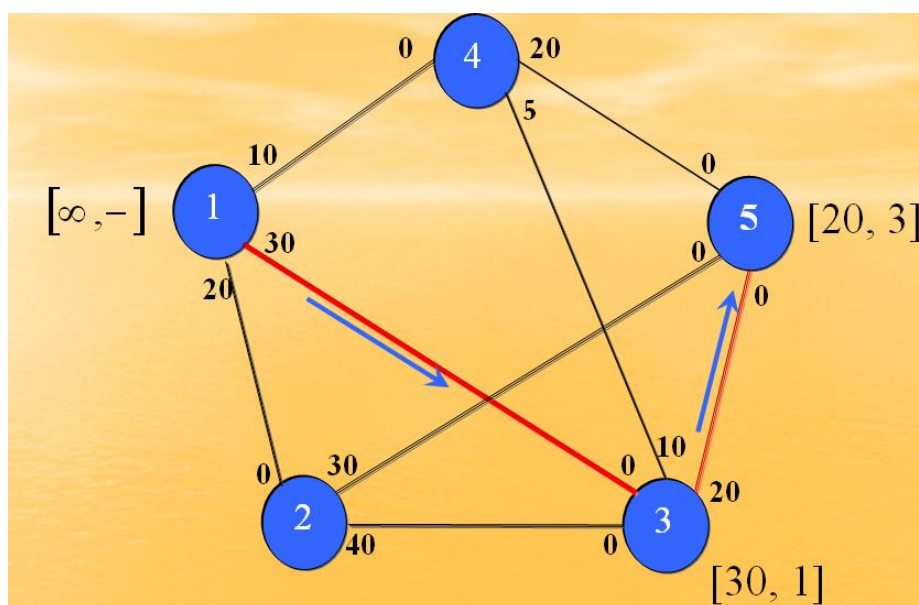


Рисунок 1.5 – Крок 2

Крок 3. Продовжуючи даний алгоритм, ми потрапляємо в останній пункт (стік). У цьому випадку необхідно скласти залишкову мережу, для чого знаходимо максимальний потік, який проходить через побудований нами шлях. Для отримання максимального потоку знаходимо мінімум значень пропускної здатності, що записані біля всіх вузлів сквозного шляху. Для нашої мережі максимальний потік дорівнює:

$$f_1 = \min\{\infty, 30, 20\} = 20.$$

Залишкові пропускні здатності ребер, які утворюють сквозний шлях, зменшуються на f_1 у напрямку руху потоку й збільшуються на ту ж саму величину в протилежному напрямку.

Отже, отримаємо залишкову мережу (рис. 1.6).

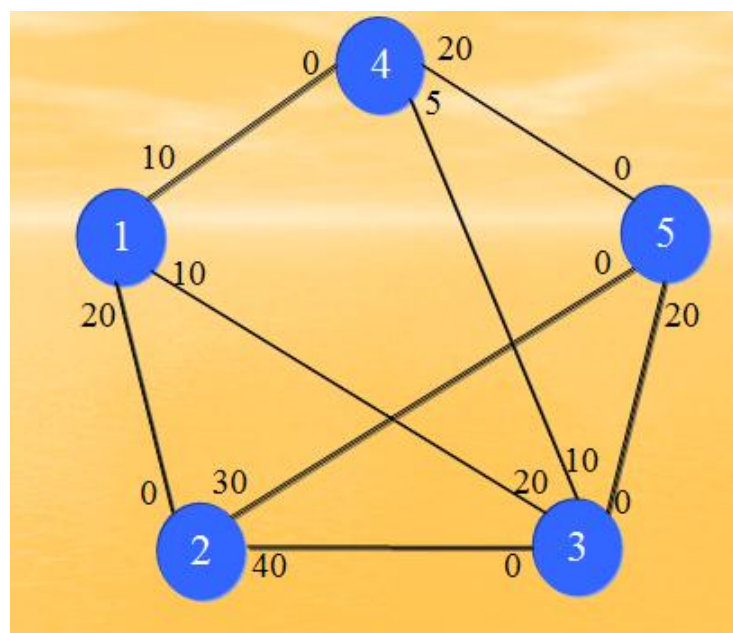


Рисунок 1.6 – Крок 3

Крок 4. Повторюємо алгоритм, що записаний у кроках 1,2 для залишкової мережі (рис. 1.7).

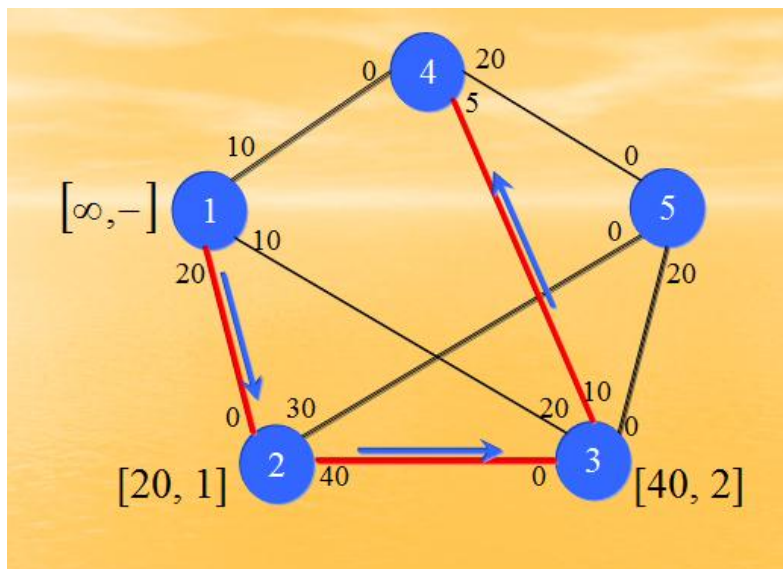


Рисунок 1.7 – Виконання алгоритму для залишкової мережі

Зверніть увагу! Діючи за алгоритмом, ми мали б піти з вузла 3 до вузла 1, проте такий шлях нам не підходить, оскільки вузол 3 — наш початковий пункт, повернувшись в який ми зробимо коло й не потрапимо до пункту призначення.

Продовжуючи алгоритм, знайдемо сквозний шлях (рис. 1.8).

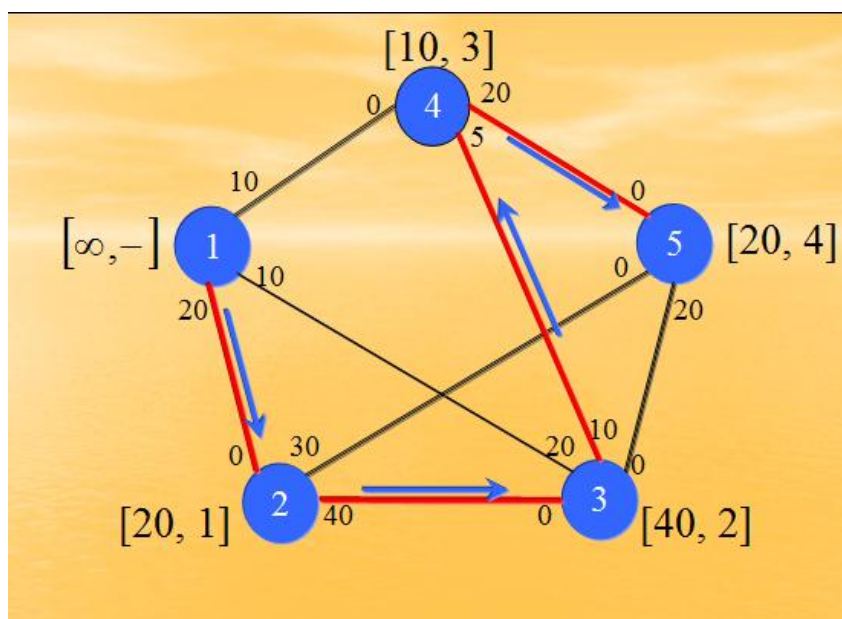


Рисунок 1.8 – Сквозний шлях

Підраховуємо максимальний потік:

$$f_2 = \min\{\infty, 20, 40, 10, 20\} = 10.$$

Далі будемо нову залишкову мережу, як описано в кроці 3. Шукана мережа зображена на рисунку 1.9

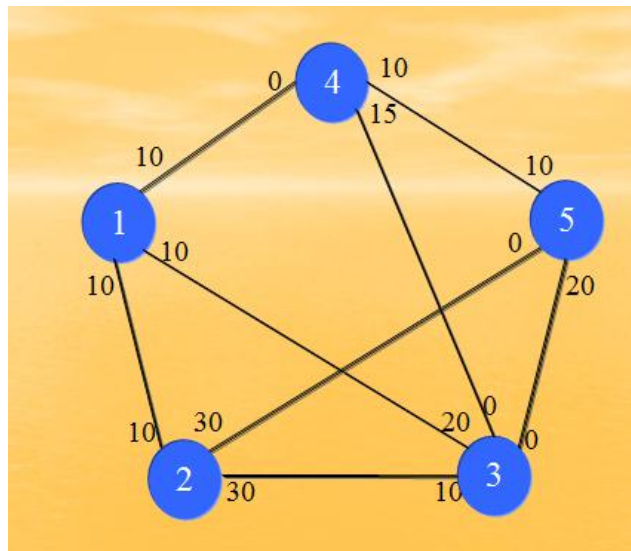


Рисунок 1.9 – Залишкова мережа

Продовжуючи алгоритм, будемо отримувати нові сквозні шляхи.

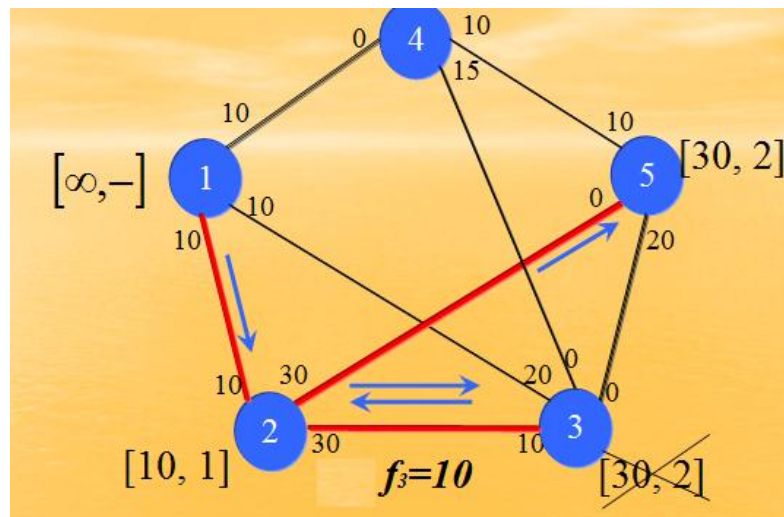


Рисунок 1.10 – Сквозний шлях

Зверніть увагу! На рисунку 1.10 ми мали два альтернативних ребра. У такому випадку, зазвичай, обирають те ребро, яке приводить до вузла з меншим номером. Проте у нашому прикладі такий шлях привів до руху по колу, тому довелося обрати інше ребро.

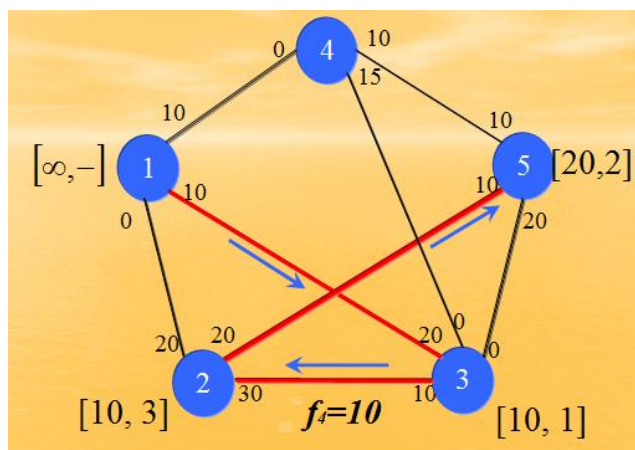


Рисунок 1.11 – Сквозний шлях

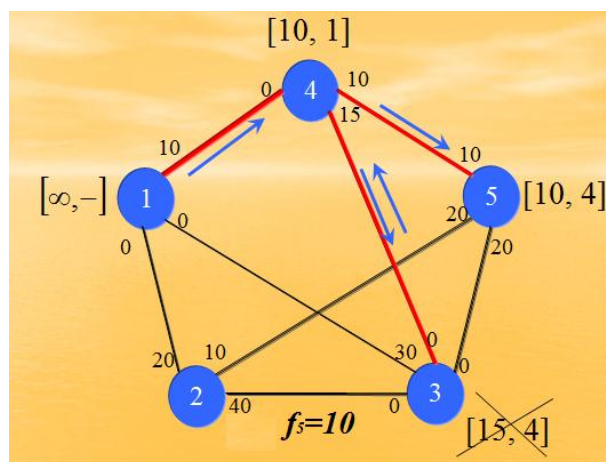


Рисунок 1.12 – Сквозний шлях

Зі сквозного шляху (рис. 1.12) отримали мережу, в якій заборонений рух із вузла 1 уздовж будь-якого ребра, тобто виконується процес відмови назад.

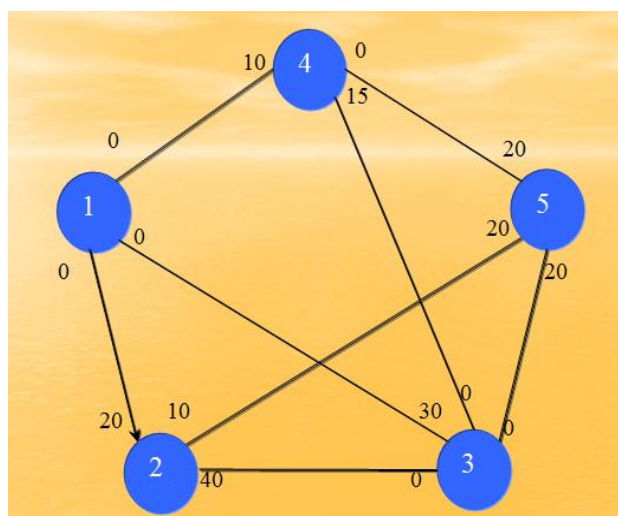


Рисунок 1.13 – Залишкова мережа

Тепер можемо обчислювати значення потоків на різних ребрах. Для цього віднімаємо значення залишкових пропускних здатностей (рис. 1.13) з початкових значень пропускних здатностей (рис. 1.3). Результати відображені на рисунку 1.14.

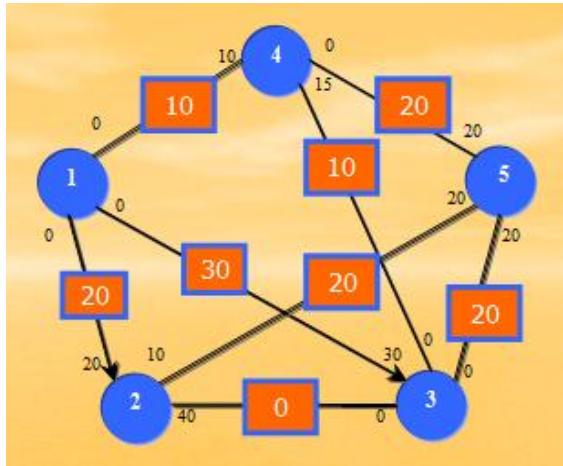


Рисунок 1.14 – Остаточні пропускні здатності

У таблиці 1.2 продемонструємо детальне обчислення остаточних пропускних здатностей.

Таблиця 1.2 – Обчислення пропускних здатностей

Ребро	Обчислення	Величина потоку	Напрямок
(1,2)	$(20,0) - (0,20) = (20, -20)$	20	1 → 2
(1,3)	$(30,0) - (0,30) = (30, -30)$	30	1 → 3
(1,4)	$(10,0) - (0,10) = (10, -10)$	10	1 → 4
(2,3)	$(40,0) - (40,0) = (0,0)$	0	–
(2,5)	$(30,0) - (10,20) = (20, -20)$	20	2 → 5
(3,4)	$(10,5) - (0,15) = (10, -10)$	10	3 → 4
(3,5)	$(20,0) - (0,20) = (20, -20)$	20	3 → 5
(4,5)	$(20,0) - (0,20) = (20, -20)$	20	4 → 5

Зверніть увагу! Величина потоку на ребрі (2,3) дорівнює нулеві. Це не значить, що автомобілі не можуть рухатися цією дорогою, проте, основні шляхи не будуть проходити через цю вулицю. Отже, ділянку проспекту Ватутіна від перехрестя з вулицею Челюскінців до перехрестя з вулицею п'ятдесятиріччя ССРСР розширювати недоцільно (рис. 1.2).

1.2.2 Індивідуальні завдання. Максимальний потік

Визначіть максимальний потік мережі.

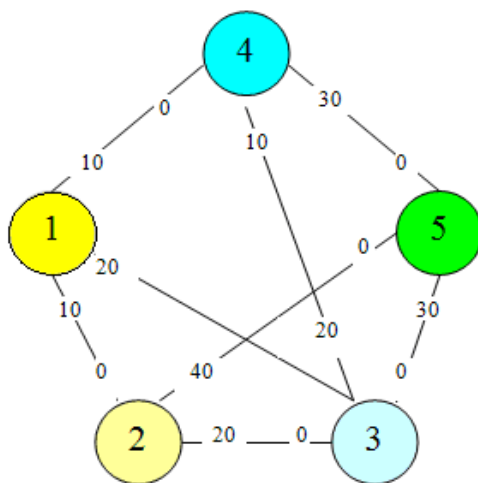


Рисунок 1.15 – Мережа (варіант 1)

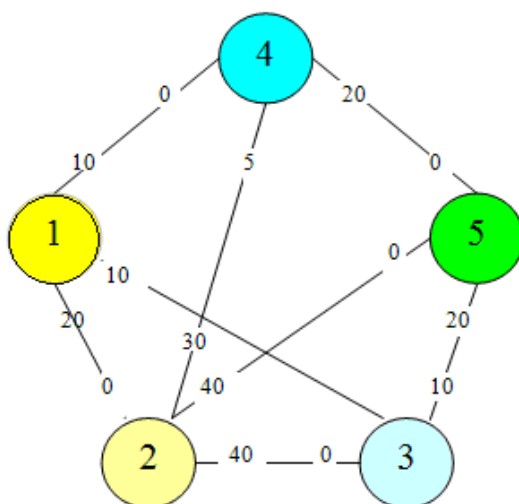


Рисунок 1.16 – Мережа (варіант 2)

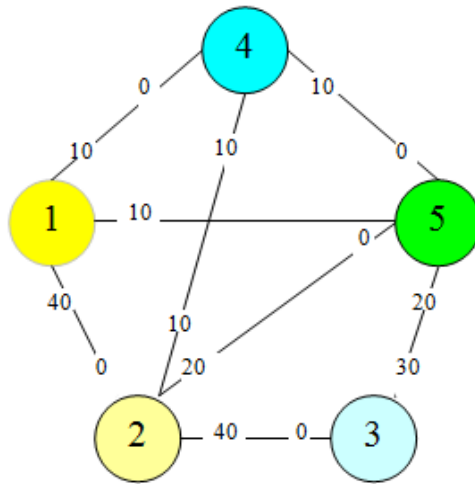


Рисунок 1.17 – Мережа (варіант 3)

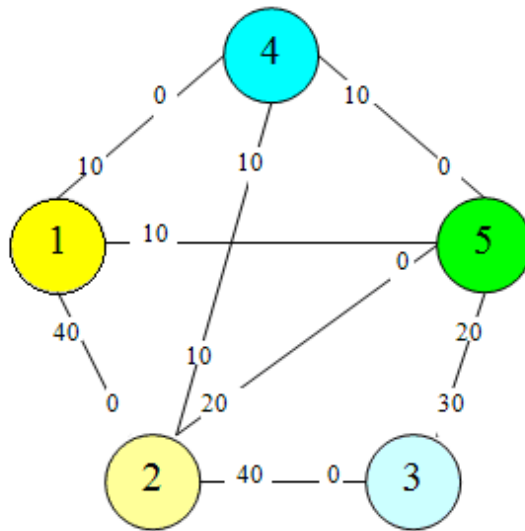


Рисунок 1.18 – Мережа (варіант 4)

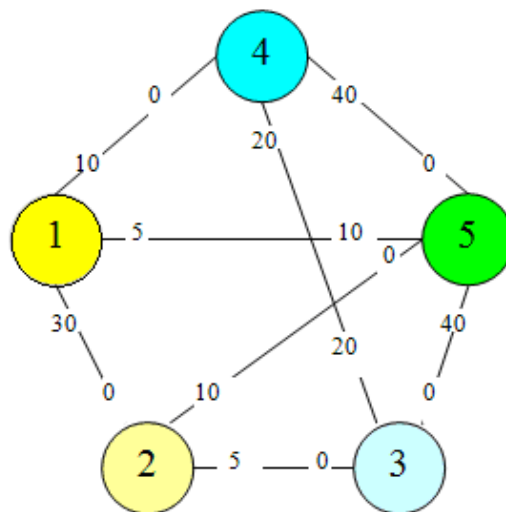


Рисунок 1.19 – Мережа (варіант 5)

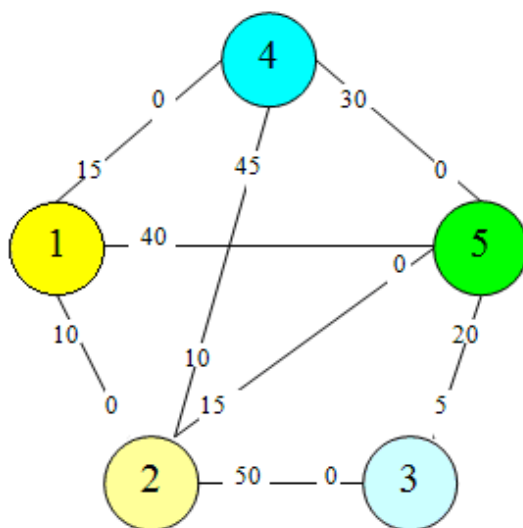


Рисунок 1.20 – Мережа (варіант 6)

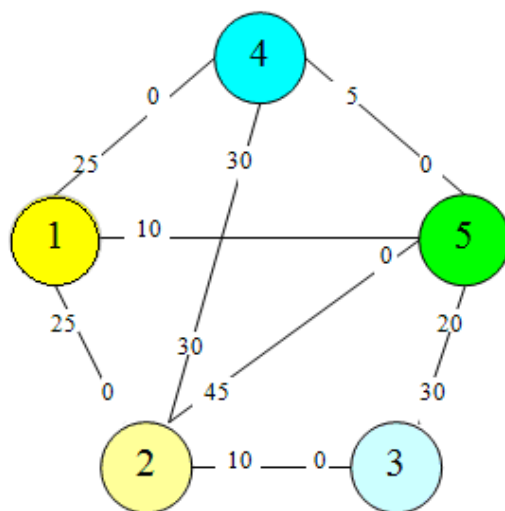


Рисунок 1.21 – Мережа (варіант 7)

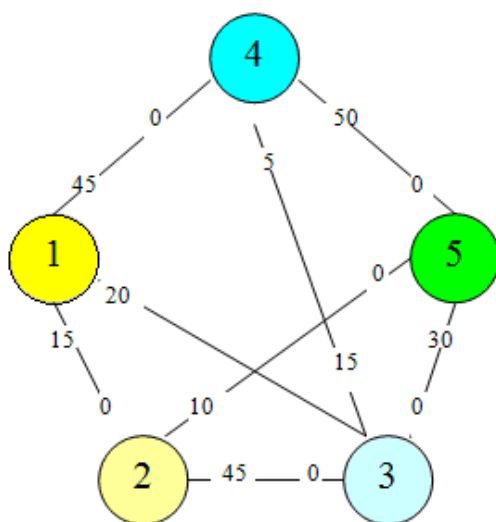


Рисунок 1.22 – Мережа (варіант 8)

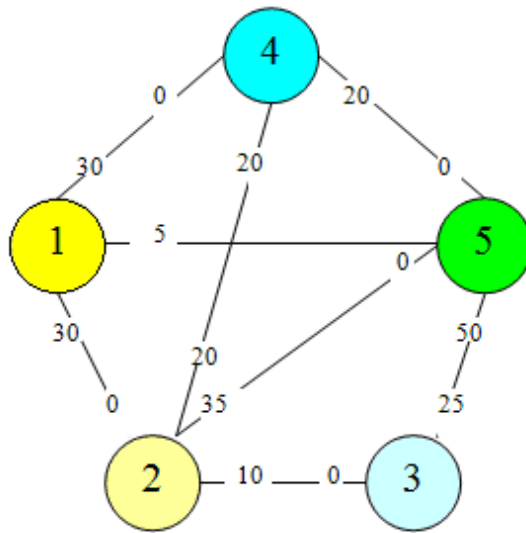


Рисунок 1.23 – Мережа (варіант 9)

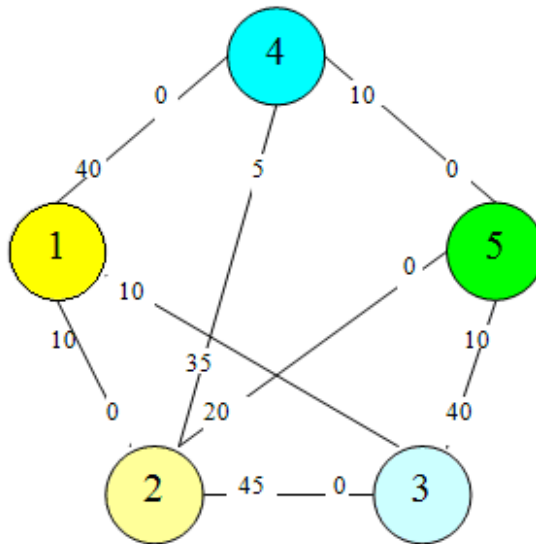


Рисунок 1.24 – Мережа (варіант 10)

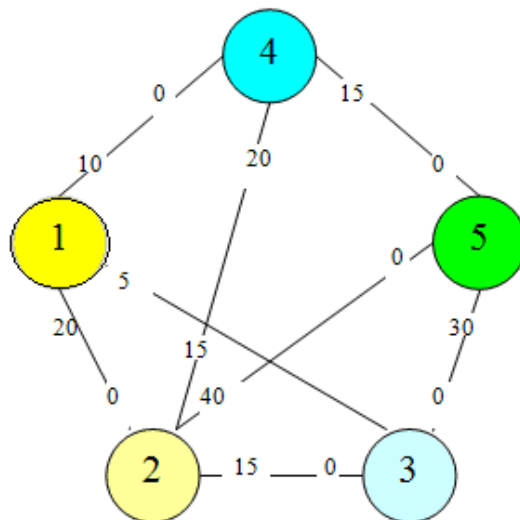


Рисунок 1.25 – Мережа (варіант 11)

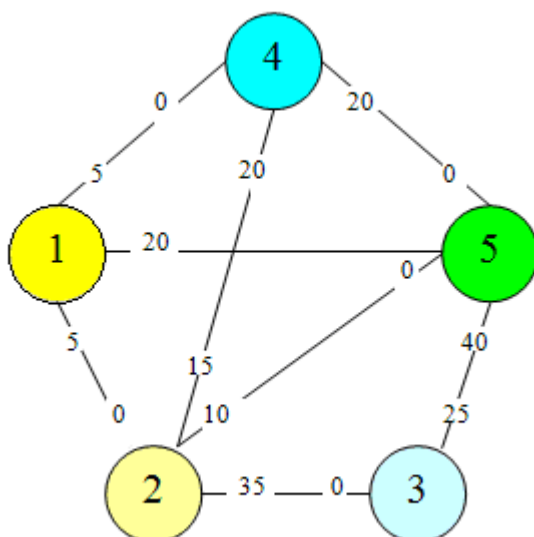


Рисунок 1.26 – Мережа (варіант 12)

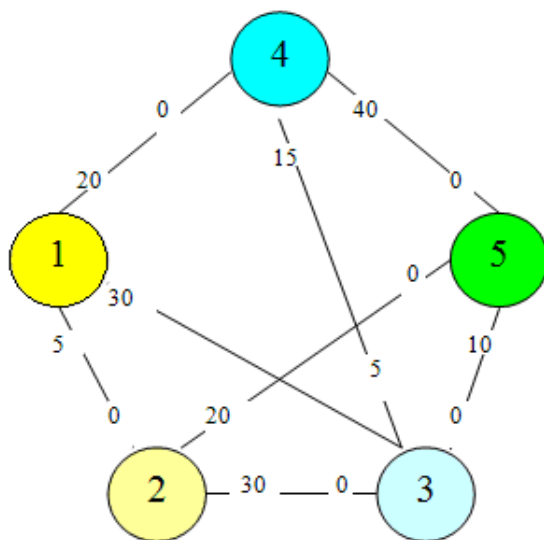


Рисунок 1.27 – Мережа (варіант 13)

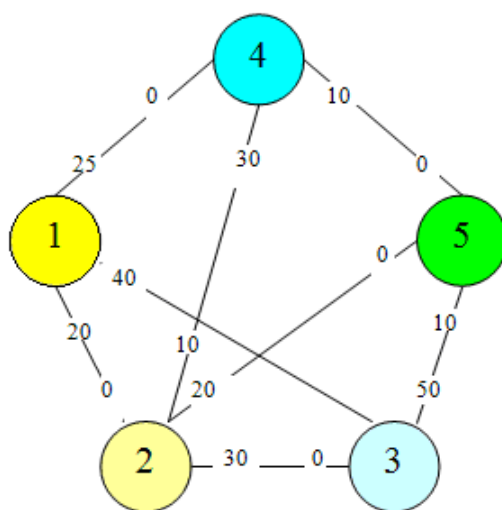


Рисунок 1.28 – Мережа (варіант 14)

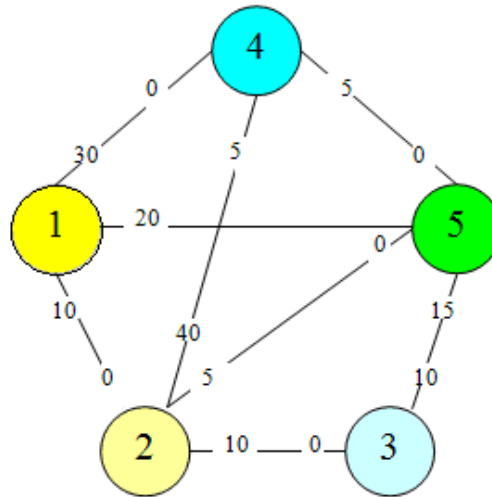


Рисунок 1.29 – Мережа (варіант 15)

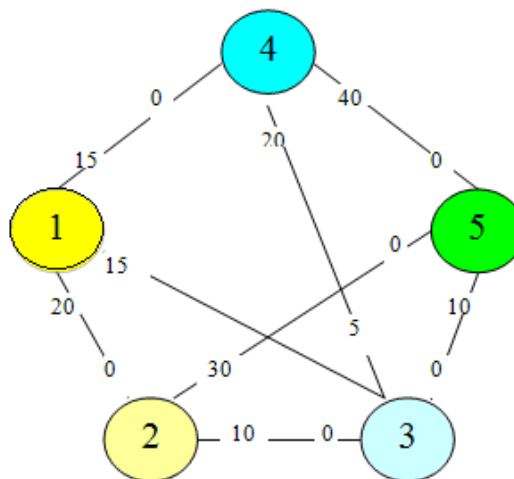


Рисунок 1.30 – Мережа (варіант 16)

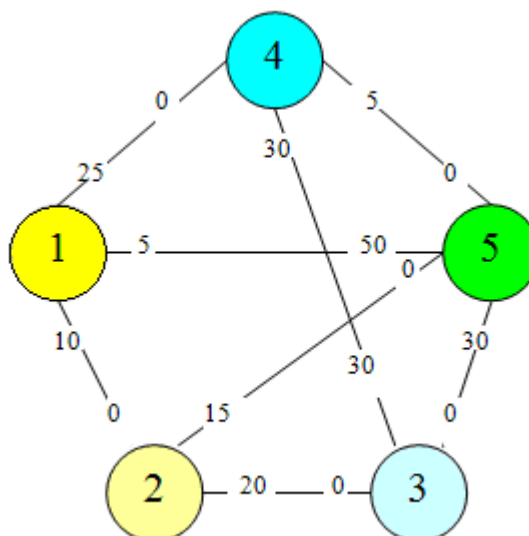


Рисунок 1.31 – Мережа (варіант 17)

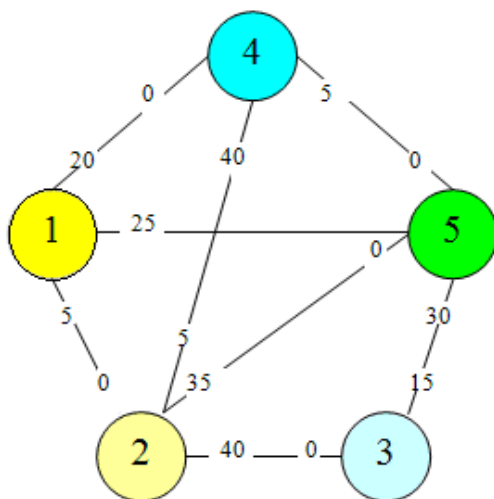


Рисунок 1.32 – Мережа (варіант 18)

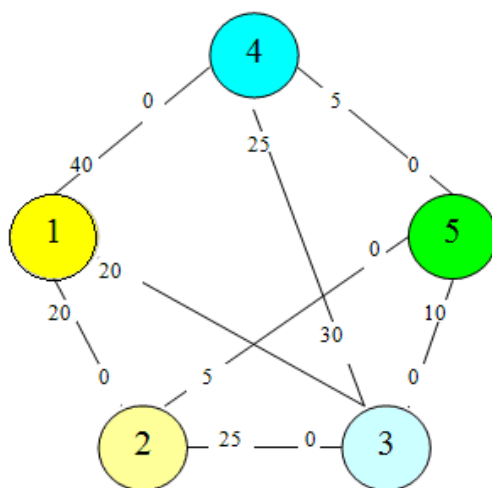


Рисунок 1.33 – Мережа (варіант 19)

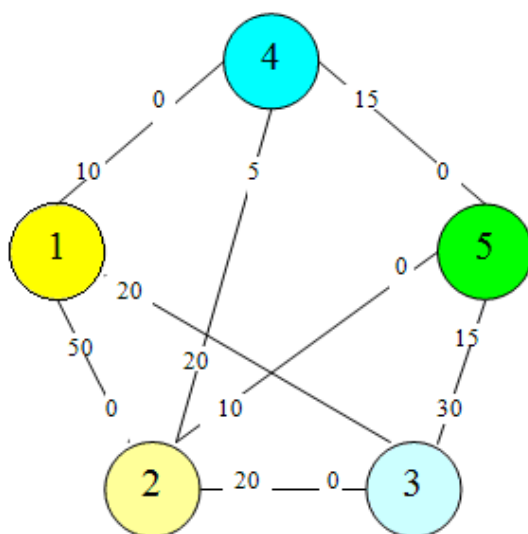


Рисунок 1.34 – Мережа (варіант 20)

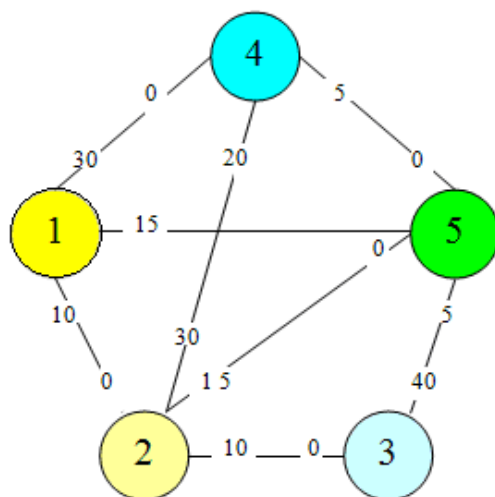


Рисунок 1.35 – Мережа (варіант 21)

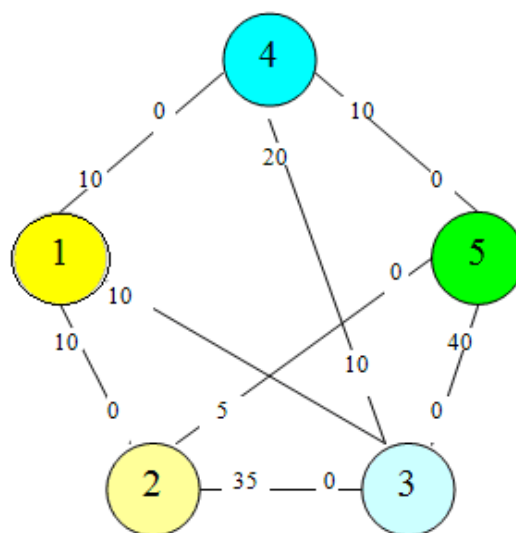


Рисунок 1.36 – Мережа (варіант 22)

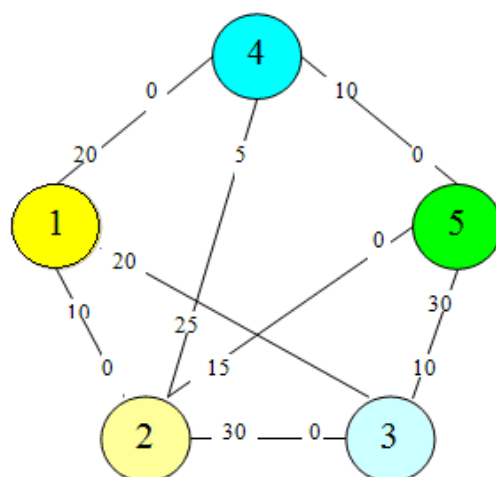


Рисунок 1.37 – Мережа (варіант 23)

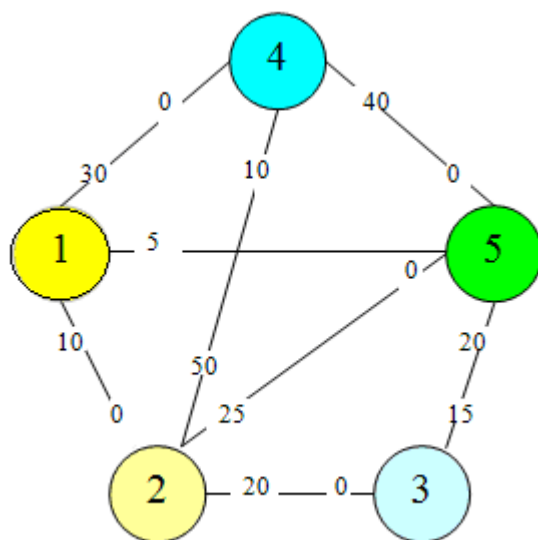


Рисунок 1.38 – Мережа (варіант 24)

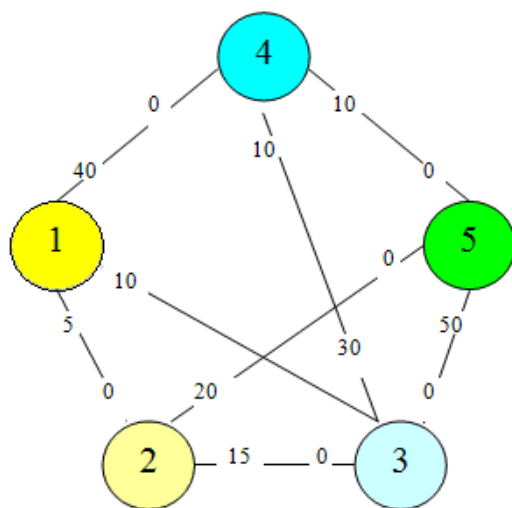


Рисунок 1.39 – Мережа (варіант 25)

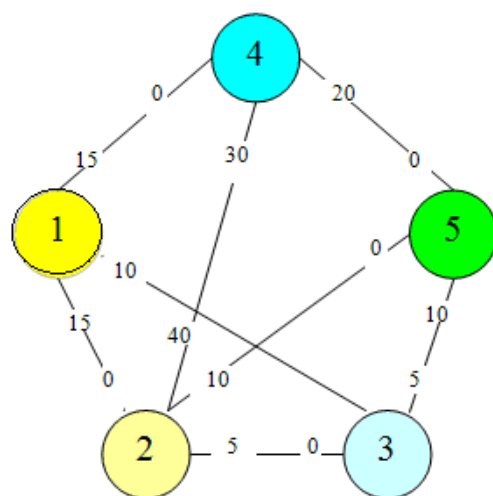


Рисунок 1.40 – Мережа (варіант 26)

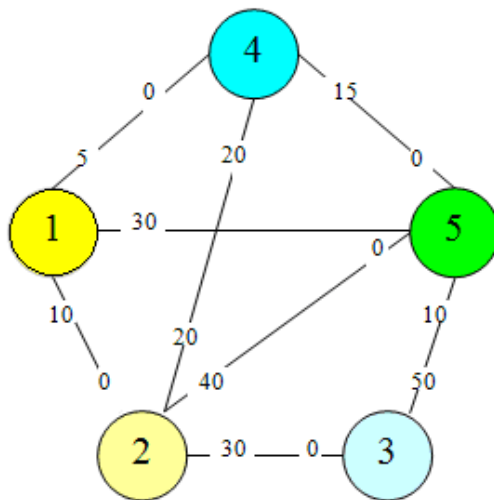


Рисунок 1.41 – Мережа (варіант 27)

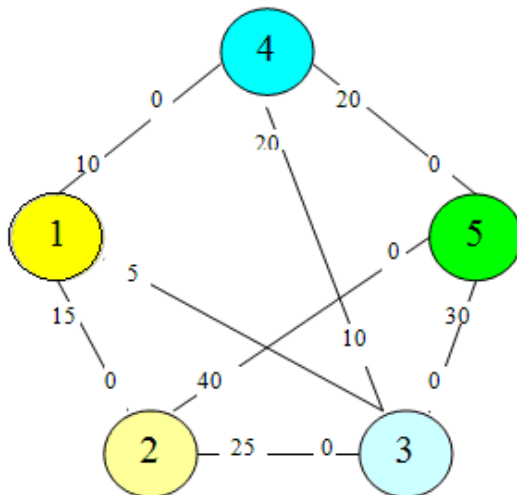


Рисунок 1.42 – Мережа (варіант 28)

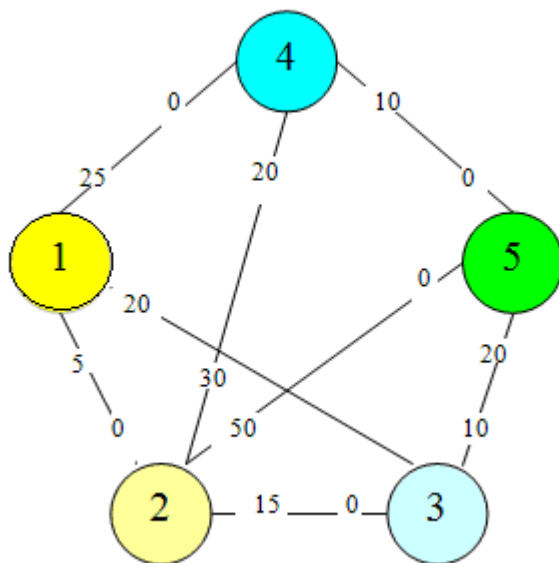


Рисунок 1.43 – Мережа (варіант 29)

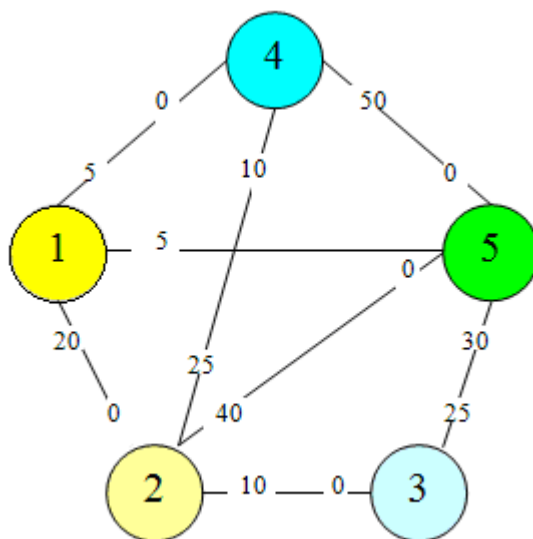
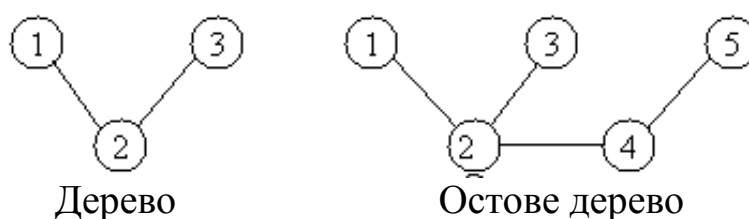


Рисунок 1.44 – Мережа (варіант 30)

1.3 Побудова мінімального остового дерева

Зв'язана мережа – така мережа, в якій будь-які два вузли зв'язані принаймні одним шляхом. На рисунку 1.1 показаний саме такий тип мережі. *Деревом* називається зв'язана мережа, що містить підмножину вузлів вихідної мережі й не має циклів. *Остове дерево* – це дерево, що містить усі вузли мережі. На рисунку 1.45 показано дерево й остове дерево для мережі з рисунку 1.1.



Дерево

Остове дерево

Рисунок 1.45 – Дерево + остове дерево

Побудова мінімального остового дерева передбачає з'єднання всіх вузлів мережі за допомогою шляхів найменшої довжини.

Опишемо процедуру виконання алгоритму побудови мінімального остового дерева.

Позначимо через $N=\{1,2,\dots,n\}$ множину вузлів мережі й введемо нові позначення:

S_k – множина вузлів мережі, що з'єднані алгоритмом після виконання k -ї ітерації цього алгоритму,

$\overline{C_k}$ – множина вузлів мережі, що не з'єднані з вузлами множини C_k після виконання k -ї ітерації цього алгоритму.

Крок 0. Покладемо $C_0 = \emptyset$ і $\overline{C_0} = N$.

Крок 1. Обираємо будь-який вузол множини $\overline{C_0}$, визначаємо:

$$C_1 = \{i\}, \text{ тоді } \overline{C_1} = N - \{i\}.$$

Переходимо до $k = 2$.

Основний крок k . У множині $\overline{C_{k-1}}$ обираємо вузол j^* , що з'єднаний найкоротшою дугою з яким-небудь вузлом із множини C_{k-1} . Вузол j^* приєднується до множини C_{k-1} і видаляється з множини $\overline{C_{k-1}}$. Таким чином, $C_k = C_{k-1} + \{j^*\}$, $\overline{C_k} = \overline{C_{k-1}} - \{j^*\}$.

Якщо множина C_k порожня, то виконання алгоритму закінчується. У протилежному випадку вважаємо $k = k + 1$ і повторюємо останній крок.

Алгоритм побудови мінімального остового дерева продемонструємо на прикладі проектування мережі доріг із твердим покриттям.

Постановка задачі: план підготовки до чемпіонату Євро-2012 передбачає спорудження нового терміналу аеропорту «Донецьк» (рис. 1.46). Інфраструктура нового терміналу включає такі компоненти: сам термінал, триповерховий паркінг на 600 місць, дві естакади, привокзальну площу зі стоянкою, авіадиспетчерську вежу.



Рисунок 1.46 – Будівництво нового терміналу (фото)

Необхідно спланувати мережу доріг із твердим покриттям, що з'єднують об'єкти інфраструктури між собою.

Зазначимо, що в даній задачі дороги, що з'єднують два яких-небудь об'єкти інфраструктури, можуть проходити через інші об'єкти. Найбільш економний проект дорожньої системи має мінімізувати загальну довжину

доріг із твердим покриттям, при цьому бажаний результат можна одержати шляхом застосування алгоритму побудови мінімального остового дерева.

На рисунку 1.47 зображено структуру планової мережі й відстані (у км) між об'єктами інфраструктури.

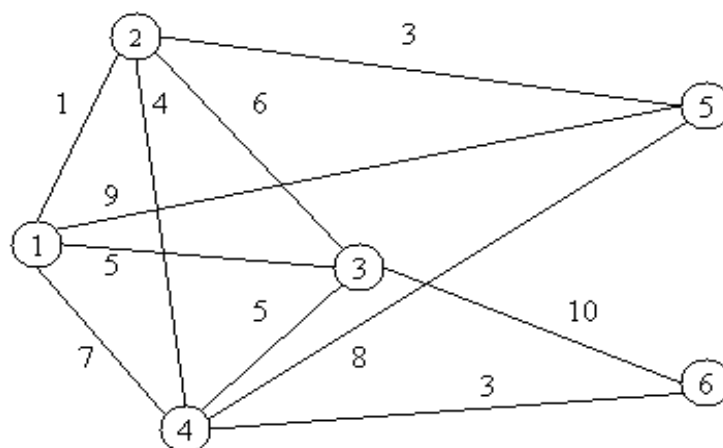


Рисунок 1.47 – Структура планової мережі
Пояснення до мережі наведені в таблиці 1.3.

Таблиця 1.3 – Розшифровка структури планової мережі

Номер вузлу	Об'єкт інфраструктури
1	Будівля терміналу
2	Авіадиспетчерська вежа
3	Привокзальна площа
4	Паркінг на 600 місць
5	Початок естакади № 1
6	Початок естакади № 2

Почнемо виконання алгоритму побудови мінімального остового дерева з вибору вузла 1 (або будь-якого іншого вузла). Тоді

$$C_1 = \{1\} \text{ і } \bar{C}_1 = \{2,3,4,5,6\}.$$

Виконання послідовності ітерацій алгоритму представлено на рисунку 1.48. Тут тонкими лініями показані ребра, що з'єднують вузли, які належать множинам C_k і \bar{C}_k , серед яких обираємо ребро з мінімальною вартістю (довжиною). Таке ребро позначене пунктирною лінією. Товстими суцільними лініями позначені ребра, що з'єднують вузли множини C_k (і які раніше позначалися пунктирними лініями).

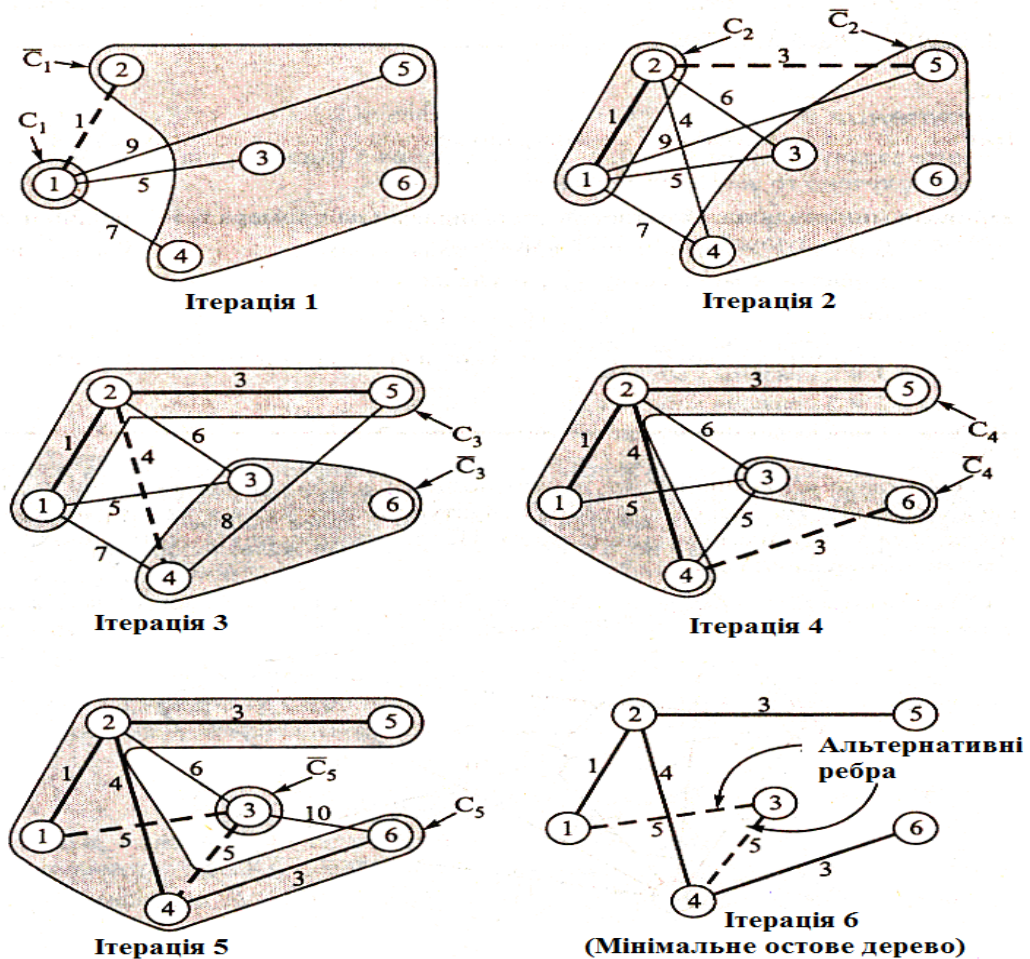


Рисунок 1.48 – Послідовність ітерацій виконання алгоритму

Зверніть увагу! Прийняття рішення з вибору альтернативних ребер (рис. 1.48, ітерація 6) виходить за межі математичної моделі. Подібні рішення приймає експерт – конструктор споруди.

Пояснення до ітерації 1: ребро (1,2) має найменшу вартість (тобто найменшу відстань між пунктами мережі) порівняно з усіма іншими ребрами, які з'єднують вузол 1 з вузлами множини (відзначимо, що вузол 6 не має ребра, що безпосередньо з'єднує його з вузлом 1). Тому $j = 2$ і $C_2 = \{1,2\}$, $\bar{C}_2 = \{3,4,5,6\}$.

Рішення у вигляді мінімального остового дерева отримане на 6-й ітерації (рисунок 1.17). *Мінімальна довжина доріг для з'єднання об'єктів інфраструктури нового терміналу аеропорту «Донецьк» дорівнює $1 + 3 + 4 + 3 + 5 = 16$ (км).*

1.4 Модель Флойда

Дозволяє знайти найкоротші шляхи між будь-якими двома вузлами мережі. У цьому алгоритмі мережа представлена у вигляді квадратної

матриці з n рядками і n стовпцями. Елемент (i,j) дорівнює відстані d_{ij} від вузла i до вузла j . Зазначена відстань має кінцеве значення, якщо існує дуга (i,j) , і дорівнює нескінченності в протилежному випадку.

Покажемо спочатку *основну ідею методу Флойда*. Нехай є три вузли i, j і k та задані відстані між ними (рис. 1.49). Якщо виконується нерівність $d_{ij} + d_{jk} < d_{ik}$, то доцільно замінити шлях $i \rightarrow k$ шляхом $i \rightarrow j \rightarrow k$. Така заміна (далі її будемо умовно називати *трикутним оператором*) виконується автоматично в процесі виконання алгоритму Флойда.

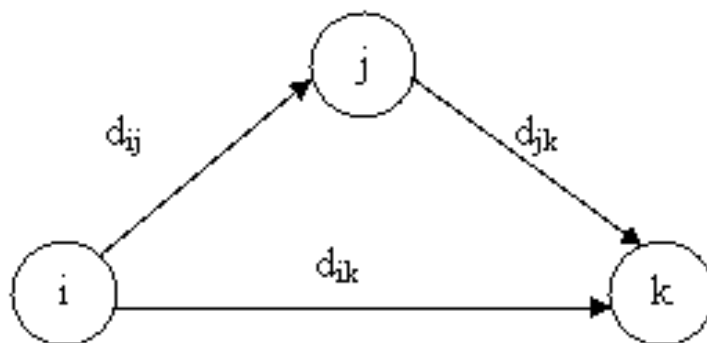


Рисунок 1.49 – Трикутний оператор

Алгоритм Флойда вимагає виконання наступних дій.

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

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

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

б) створюємо матрицю S_k шляхом заміни в матриці S_{k-1} елемента S_{ij} на k . Переходимо до $k = k + 1$ і повторюємо крок k .

Після реалізації n кроків алгоритму визначення за матрицями D_n і S_n найкоротшого шляху між вузлами i і j виконується за такими правилами:

1. Відстань між вузлами i і j дорівнює елементові d_{ij} у матриці D_n .
2. Проміжні вузли шляху від вузла i до вузла j визначаємо за матрицею S_n . Нехай $S_{ij} = k$, тоді маємо шлях $i - k - j$. Якщо далі $S_{ik} = k$ і

$S_{kj} = j$, тоді вважаємо, що весь шлях визначений, тому що знайдені всі проміжні вузли. У протилежному випадку повторюємо описану процедуру для шляхів від вузла i до вузла k і від вузла k до вузла j .

Продемонструємо реалізацію моделі Флойда.

Постановка задачі: компанія «Автолюкс» надає послуги з перевезення вантажів, а також кур'єрські послуги по Україні. Нещодавно компанія уклала договір із мережею туристичних агенцій про надання кур'єрських послуг (перевезення закордонних паспортів і документів, необхідних для отримання віз). Агенції знаходяться в містах: Київ, Кіровоград, Полтава, Харків, Донецьк, Горлівка, Кривий Ріг, Луганськ (рис. 1.50). У кожному з цих міст є представництва компанії «Автолюкс». Перевезення документів планується здійснювати постійними рейсами компанії, що створені для кур'єрської доставки, але забезпечуючи додаткову охорону. Витрати на додаткову охорону (в євро) на кожному з наявних маршрутів компанії наведені в таблиці 1.4 (буквою N позначені напрями, якими не користується компанія «Автолюкс»). Зазначимо, що витрати не пропорційні довжинам автомобільних доріг, оскільки залежать від розцінок охоронних компаній різних регіонів. Необхідно знайти найдешевші (стосовно вартості охоронних послуг) маршрути між будь-якими двома містами.

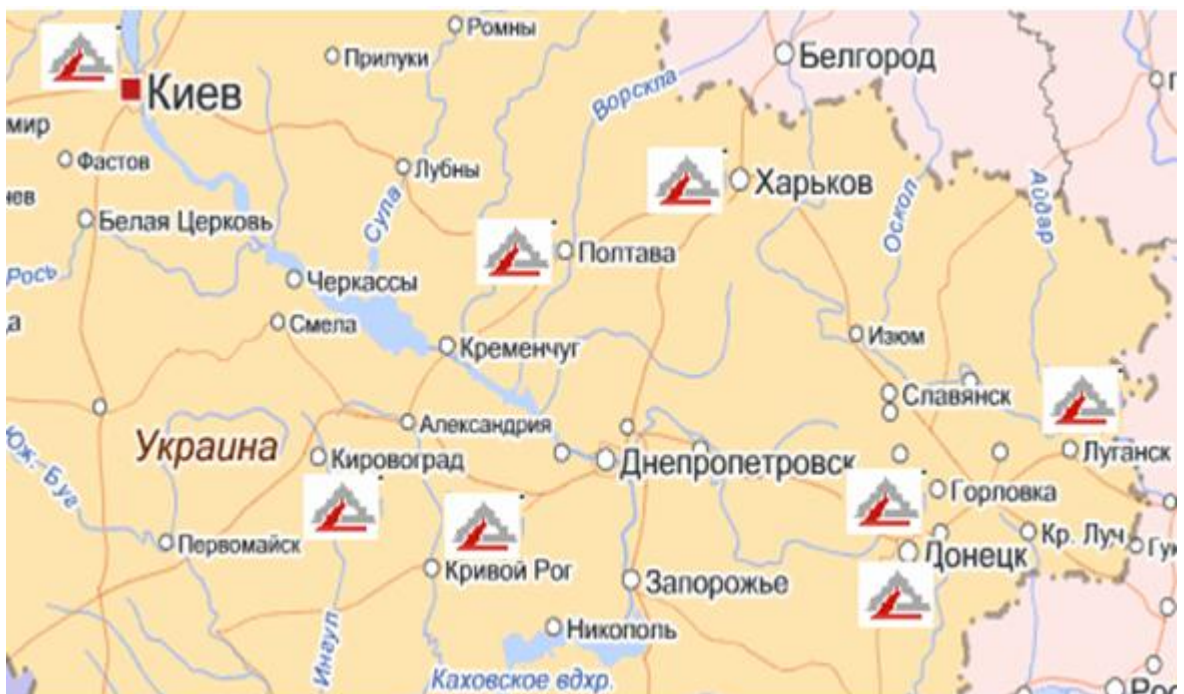


Рисунок 1.50 – Розміщення представництв компанії «Автолюкс»

Таблиця 1.4 – Витрати на додаткову охорону

	Луганськ	Горлівка	Донецьк	Кривий Ріг	Кіровоград	Полтава	Харків	Київ
Луганськ	–	3	5	N	N	4	4	2
Горлівка	3	–	N	5	N	N	3	N
Донецьк	5	N	–	6	2	3	N	1
Кривий Ріг	N	5	6	–	4	N	1	3
Кіровоград	N	N	N	4	–	1	5	1
Полтава	5	N	5	3	N	–	N	4
Харків	4	4	N	3	2	N	–	1
Київ	4	N	4	N	N	2	3	–

Пронумеруємо представництва компанії «Автолюкс»:

- 1 – представництво міста Луганськ;
- 2 – представництво міста Горлівка;
- 3 – представництво міста Донецьк;
- 4 – представництво міста Кривий Ріг;
- 5 – представництво міста Кіровоград;
- 6 – представництво міста Полтава;
- 7 – представництво міста Харків;
- 8 – представництво міста Київ.

За даними карти (рис. 1.50) і таблиці 1.4 побудуємо мережу (рис. 1.51). На рисунку 1.51 при позначенні вартості охоронних послуг дотримувалися правила, що вартість послуг із пункту i у пункт j розташовується на ребрі ij ближче до i , а вартість послуг із пункту j у пункт i ближче до j .

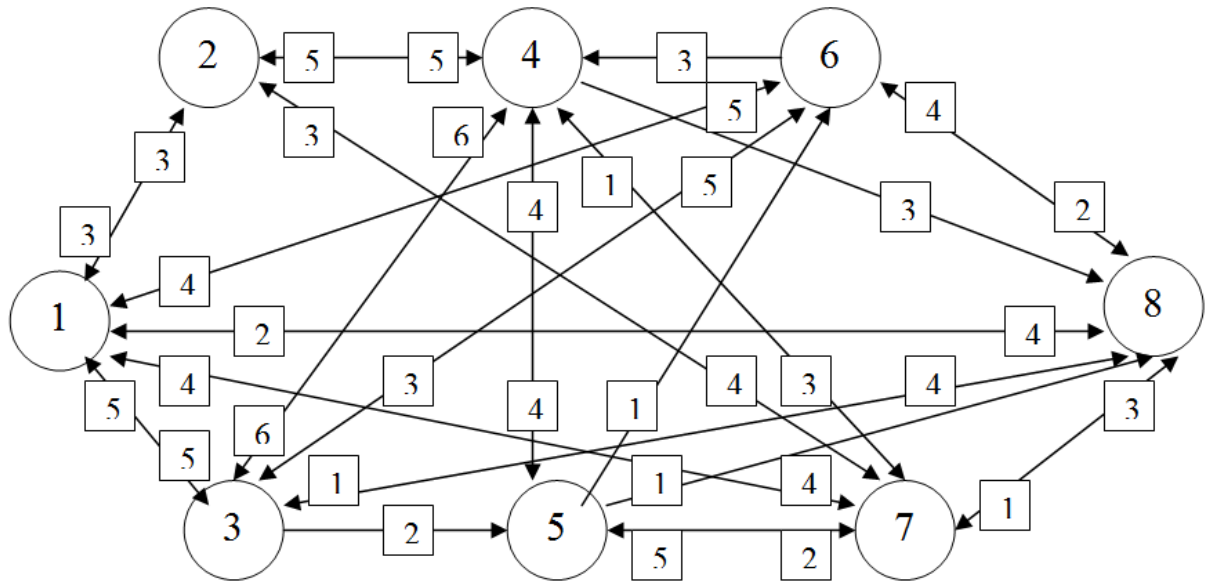


Рисунок 1.51 – Схема мережі

Крок 0. Початкові матриці \mathbf{D}_0 і \mathbf{S}_0 будуються безпосередньо за заданою схемою мережі. Матриця \mathbf{D}_0 містить елементи N (це значить, що компанія «Автолюкс» не використовує ці напрями), які при виконанні алгоритму вважаємо рівними нескінченності.

\mathbf{D}_0	1	2	3	4	5	6	7	8
1	–	3	5	N	N	4	4	2
2	3	–	N	5	N	N	3	N
3	5	N	–	6	2	3	N	1
4	N	5	6	–	4	N	1	3
5	N	N	N	4	–	1	5	1
6	5	N	5	3	N	–	N	4
7	4	4	N	3	2	N	–	1
8	4	N	4	N	N	2	3	–

\mathbf{S}_0	1	2	3	4	5	6	7	8
1	–	2	3	N	N	6	7	8
2	1	–	N	4	N	N	7	N
3	1	N	–	4	5	6	N	8
4	N	2	3	–	5	N	7	8
5	N	N	N	4	–	6	7	8
6	1	N	3	4	N	–	N	8
7	1	2	N	4	5	N	–	8
8	1	N	3	N	N	6	7	–

Рисунок 1.52 – Результат кроку 0

Крок 1. У матриці \mathbf{D}_0 виділяємо ведучий рядок і ведучий стовпець (перший рядок і перший стовпець). Кольором на рисунку 1.21 позначені ведучий рядок, ведучий стовпець і елементи матриці \mathbf{D}_0 , значення яких можна поліпшити за допомогою *трикутного оператора*. Наприклад, елемент d_{67} замінюємо сумою його проєкцій на напрямний рядок і напрямний стовпець, тобто $d_{61} + d_{17} = 5 + 4 = 9$ і встановлюємо $S_{67} = 1$.

D_1	1	2	3	4	5	6	7	8	S_1	1	2	3	4	5	6	7	8
1	–	3	5	N	N	4	4	2	1	–	2	3	N	N	6	7	8
2	3	–	8	5	N	7	3	5	2	1	–	1	4	N	1	7	1
3	5	8	–	6	2	3	9	1	3	1	1	–	4	5	6	1	8
4	N	5	6	–	4	N	1	3	4	N	2	3	–	5	N	7	8
5	N	N	N	4	–	1	5	1	5	N	N	N	4	–	6	7	8
6	5	8	5	3	N	–	9	4	6	1	1	3	4	N	–	1	8
7	4	4	9	3	2	8	–	1	7	1	2	1	4	5	1	–	8
8	4	7	4	N	N	2	3	–	8	1	1	3	N	N	6	7	–

Рисунок 1.53 – Результат кроку 1

Крок 2. У матриці D_1 виділяємо ведучий рядок і ведучий стовпець (другий рядок і другий стовпець). Застосовуємо трикутний оператор до елементів матриць D_1 і S_1 . У результаті отримаємо матриці D_2 і S_2 .

D_2	1	2	3	4	5	6	7	8	S_2	1	2	3	4	5	6	7	8
1	–	3	5	8	N	4	4	2	1	–	2	3	2	N	6	7	8
2	3	–	8	5	N	7	3	5	2	1	–	1	4	N	1	7	1
3	5	8	–	6	2	3	9	1	3	1	1	–	4	5	6	1	8
4	8	5	6	–	4	12	1	3	4	2	2	3	–	5	2	7	8
5	N	N	N	4	–	1	5	1	5	N	N	N	4	–	6	7	8
6	5	8	5	3	N	–	9	4	6	1	1	3	4	N	–	1	8
7	4	4	9	3	2	8	–	1	7	1	2	1	4	5	1	–	8
8	4	7	4	12	N	2	3	–	8	1	1	3	2	N	6	7	–

Рисунок 1.54 – Результат кроку 2

Крок 3. У матриці D_2 виділяємо ведучий рядок і ведучий стовпець (третій рядок і третій стовпець). Застосовуємо трикутний оператор до елементів матриць D_2 і S_2 . У результаті отримаємо матриці D_3 і S_3 .

D_3	1	2	3	4	5	6	7	8
1	–	3	5	8	N	4	4	2
2	3	–	8	5	N	7	3	5
3	5	8	–	6	2	3	9	1
4	8	5	6	–	4	12	1	3
5	N	N	N	4	–	1	5	1
6	5	8	5	3	N	–	9	4
7	4	4	9	3	2	8	–	1
8	4	7	4	12	N	2	3	–

S_3	1	2	3	4	5	6	7	8
1	–	2	3	2	N	6	7	8
2	1	–	1	4	N	1	7	1
3	1	1	–	4	5	6	1	8
4	2	2	3	–	5	2	7	8
5	N	N	N	4	–	6	7	8
6	1	1	3	4	N	–	1	8
7	1	2	1	4	5	1	–	8
8	1	1	3	2	N	6	7	–

Рисунок 1.55 – Результат кроку 3

Крок 4. У матриці D_3 виділяємо ведучий рядок і ведучий стовпець (четвертий рядок і четвертий стовпець). Застосовуємо трикутний оператор до елементів матриць D_3 і S_3 . У результаті отримаємо матриці D_4 і S_4 .

D_4	1	2	3	4	5	6	7	8
1	–	3	5	8	7	4	4	2
2	3	–	8	5	10	7	3	5
3	5	8	–	6	2	3	9	1
4	8	5	6	–	4	9	1	3
5	N	N	N	4	–	1	5	1
6	5	8	5	3	7	–	9	4
7	4	4	9	3	2	8	–	1
8	4	7	4	10	6	2	3	–

S_4	1	2	3	4	5	6	7	8
1	–	2	3	2	3	6	7	8
2	1	–	1	4	3	1	7	1
3	1	1	–	4	5	6	1	8
4	2	2	3	–	5	3	7	8
5	N	N	N	4	–	6	7	8
6	1	1	3	4	3	–	1	8
7	1	2	1	4	5	1	–	8
8	1	1	3	3	3	6	7	–

Рисунок 1.56 – Результат кроку 4

Крок 5. У матриці D_4 виділяємо ведучий рядок і ведучий стовпець (п'ятий рядок і п'ятий стовпець). Застосовуємо трикутний оператор до елементів матриць D_4 і S_4 . У результаті отримаємо матриці D_5 і S_5 .

D_5	1	2	3	4	5	6	7	8	S_5	1	2	3	4	5	6	7	8
1	–	3	5	8	7	4	4	2	1	–	2	3	2	3	6	7	8
2	3	–	8	5	9	7	3	5	2	1	–	1	4	4	1	7	1
3	5	8	–	6	2	3	7	1	3	1	1	–	4	5	6	4	8
4	8	5	6	–	4	9	1	3	4	2	2	3	–	5	3	7	8
5	12	9	10	4	–	1	5	1	5	4	4	4	4	–	6	7	8
6	5	8	5	3	7	–	4	4	6	1	1	3	4	3	–	4	8
7	4	4	9	3	2	8	–	1	7	1	2	1	4	5	1	–	8
8	4	7	4	10	6	2	3	–	8	1	1	3	3	3	6	7	–

Рисунок 1.57 – Результат кроку 5

Крок 6. У матриці D_5 виділяємо ведучий рядок і ведучий стовпець (шостий рядок і шостий стовпець). Застосовуємо трикутний оператор до елементів матриць D_5 і S_5 . У результаті отримаємо матриці D_6 і S_6 .

D_6	1	2	3	4	5	6	7	8	S_6	1	2	3	4	5	6	7	8
1	–	3	5	8	7	4	4	2	1	–	2	3	2	3	6	7	8
2	3	–	8	5	9	7	3	5	2	1	–	1	4	4	1	7	1
3	5	8	–	6	2	3	7	1	3	1	1	–	4	5	6	4	8
4	8	5	6	–	4	5	1	3	4	2	2	3	–	5	5	7	8
5	12	9	10	4	–	1	5	1	5	4	4	4	4	–	6	7	8
6	5	8	5	3	7	–	4	4	6	1	1	3	4	3	–	4	8
7	4	4	9	3	2	3	–	1	7	1	2	1	4	5	5	–	8
8	4	7	4	10	6	2	3	–	8	1	1	3	3	3	6	7	–

Рисунок 1.58 – Результат кроку 6

Крок 7. У матриці D_6 виділяємо ведучий рядок і ведучий стовпець (сьомий рядок і сьомий стовпець). Застосовуємо трикутний оператор до елементів матриць D_6 і S_6 . У результаті отримаємо матриці D_7 і S_7 .

D_7	1	2	3	4	5	6	7	8	S_7	1	2	3	4	5	6	7	8
1	–	3	5	7	7	4	4	2	1	–	2	3	6	3	6	7	8
2	3	–	8	5	9	7	3	5	2	1	–	1	4	4	1	7	1
3	5	8	–	6	2	3	7	1	3	1	1	–	4	5	6	4	8
4	8	5	6	–	4	5	1	3	4	2	2	3	–	5	5	7	8
5	6	9	6	4	–	1	5	1	5	6	4	6	4	–	6	7	8
6	5	8	5	3	7	–	4	4	6	1	1	3	4	3	–	4	8
7	4	4	8	3	2	3	–	1	7	1	2	6	4	5	5	–	8
8	4	7	4	5	6	2	3	–	8	1	1	3	6	3	6	7	–

Рисунок 1.59 – Результат кроку 7

Крок 8. У матриці D_7 виділяємо ведучий рядок і ведучий стовпець (восьмий рядок і восьмий стовпець). Застосовуємо трикутний оператор до елементів матриць D_7 і S_7 . У результаті отримаємо матриці D_8 і S_8 .

D_8	1	2	3	4	5	6	7	8	S_8	1	2	3	4	5	6	7	8
1	–	3	5	7	6	4	4	2	1	–	2	3	6	7	6	7	8
2	3	–	8	5	5	6	3	4	2	1	–	1	4	7	7	7	7
3	5	8	–	6	2	3	7	1	3	1	1	–	4	5	6	4	8
4	5	5	6	–	3	4	1	2	4	7	2	3	–	7	7	7	7
5	6	9	6	4	–	1	5	1	5	6	4	6	4	–	6	7	8
6	5	8	5	3	6	–	4	4	6	1	1	3	4	7	–	4	8
7	4	4	8	3	2	3	–	1	7	1	2	6	4	5	5	–	8
8	4	7	4	5	5	2	3	–	8	1	1	3	6	7	6	7	–

Рисунок 1.60 – Результат кроку 8

Крок 9. У матриці D_8 виділяємо ведучий рядок і ведучий стовпець (дев'ятий рядок і дев'ятий стовпець). Застосовуємо трикутний оператор до елементів матриць D_8 і S_8 . У результаті отримаємо матриці D_9 і S_9 .

D_9	1	2	3	4	5	6	7	8	S_9	1	2	3	4	5	6	7	8
1	–	3	5	7	6	4	4	2	1	–	2	3	6	7	6	7	8
2	3	–	8	5	5	6	3	4	2	1	–	1	4	7	7	7	7
3	5	8	–	6	2	3	4	1	3	1	1	–	4	5	6	8	8
4	5	5	6	–	3	4	1	2	4	7	2	3	–	7	7	7	7
5	5	8	5	4	–	1	4	1	5	8	8	8	4	–	6	8	8
6	5	8	5	3	6	–	4	4	6	1	1	3	4	7	–	4	8
7	4	4	5	3	2	3	–	1	7	1	2	8	4	5	5	–	8
8	4	7	4	5	5	2	3	–	8	1	1	3	6	7	6	7	–

Рисунок 1.61 – Результат кроку 9

Кінцеві матриці D_9 і S_9 містять всю інформацію, необхідну для визначення найдешевшого (стосовно вартості охоронних послуг) маршруту між будь-якими двома містами.

Наприклад, перевезення за найдешевшим маршрутом (стосовно вартості охоронних послуг) між Кіровоградом і Харковом (пунктами 5 і 7) коштуватиме 1 євро. При цьому, оптимальний маршрут: Кіровоград \rightarrow Київ \rightarrow Харків (5 \rightarrow 8 \rightarrow 7), оскільки $S_{57} = 8$, тобто наявний проміжний пункт 8 (Київ). $S_{87} = 7$, тому з пункту 8 (з Києва) доцільно одразу прямувати в пункт 7 (до Харкова).

1.5 Оптимальний розподіл товарообігу по мережі

Постановка задачі: мережа автозаправних станцій складається з 9 пунктів у місті Донецьку. На початку літнього періоду на деяких станціях закінчилося дизпаливо. Керівництво прийняло рішення не докуповувати паливо, а перерозподілити наявне між автозаправними станціями, тим самим поповнити нестачу й ліквідувати залишки. На карту (рис. 1.62) нанесені автозаправні станції, зазначені запаси на них (у м³) і стрілками позначені можливі напрямки руху між пунктами й ціна перевезення 1 м³ дизпалива на кожному з них (у гривнях). Необхідно спланувати перерозподіл, щоб сумарні витрати були мінімальними.

На рисунку 1.62 при позначенні ціни перевезення 1 м³ дизпалива дотримувалися правила, що ціна перевезення з пункту i у пункт j розташовується на ребрі ij ближче до i , а з пункту j у пункт i ближче до j .

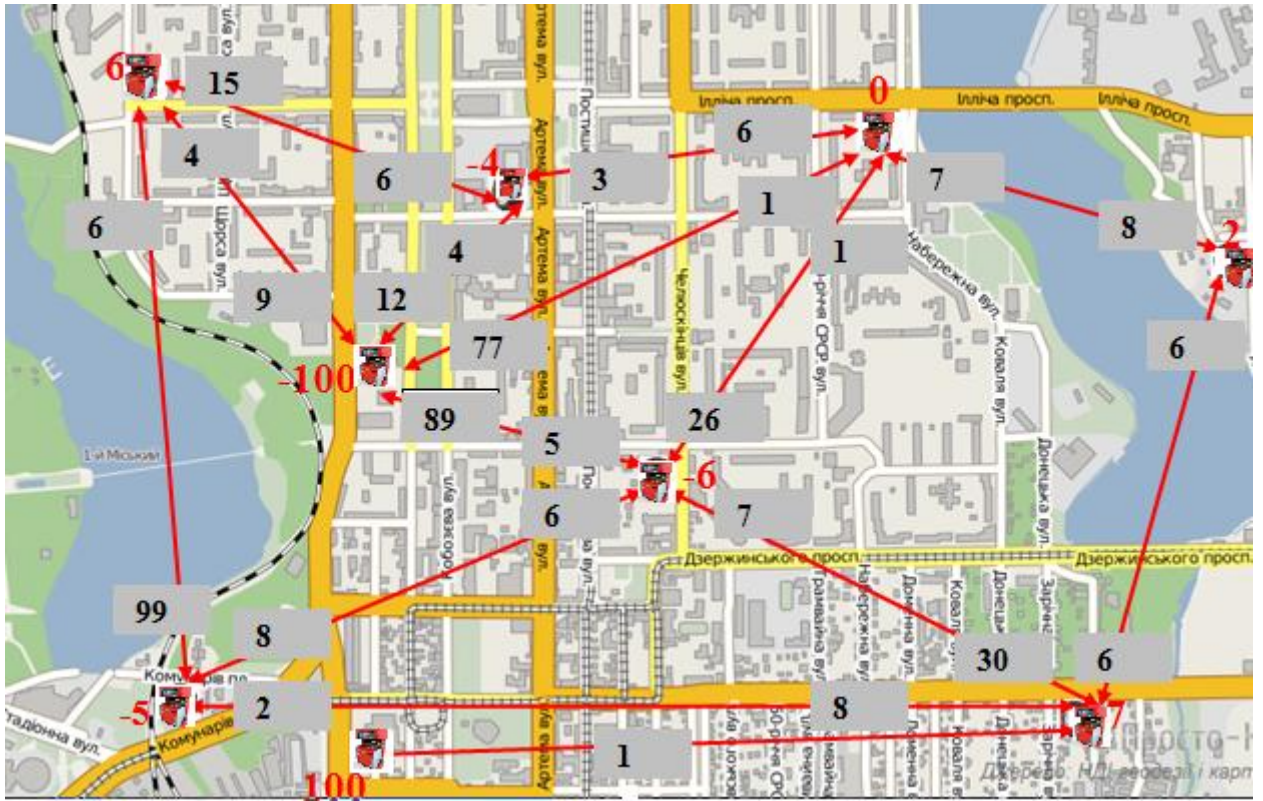


Рисунок 1.62 – Карта Донецька

За картою складемо схему (рис. 1.63).

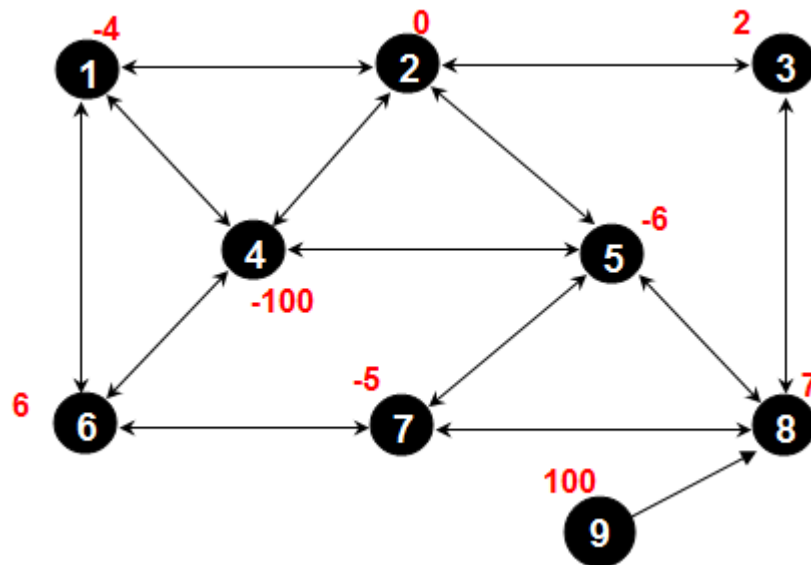


Рисунок 1.63 – Схема

На схемі пронумеровані вузли (автозаправні станції). Додатне число, яке стоїть біля номеру станції, позначає надлишок дизпалива. Від'ємне число позначає потребу даної станції у додатковому дизпаливі.

Дизпаливо може транспортуватися через автозаправні станції 1, 2, 3, 4, 5, 6, 7, 8, тому ці станції називаються *проміжними*. Усі інші

станції називаються *джерелами*, якщо в них є надлишок запасів, і *стоками*, якщо потрібен додатковий запас. У нашій задачі стоків немає, а пункт 9 є джерелом.

Зверніть увагу! Модель допускає перевезення між деякими станціями в двох напрямках, наприклад, між 1 і 2. При цьому ціна перевезення 1 м^3 дизпалива з пункту 1 у пункт 2 дорівнює 3 грн., а у зворотному напрямку – 6 грн. Відмінність у ціні пояснюється тим, що в напрямку 1-2 для перевезення можна використати автомобіль компанії, а в іншому напрямку доводиться користуватися послугами найманого транспорту.

Розв'язуємо задачу за допомогою Microsoft Excel.

Побудову таблиць проводимо у такій послідовності:

1) виділяємо рядок для кожного джерела. Значення A_i для джерела визначається кількістю виробів, що поставляються з нього;

2) виділяємо стовпець для кожного стоку. Значення B_j для стоку визначається потребою у виробках;

3) виділяємо рядок і стовпець для кожного проміжного пункту. Нехай T_k – «чистий» запас пункту, що розглядається. Якщо в цьому пункті є надлишок запасів, то T_k – додатне число, якщо ж потрібно поповнення запасів, то T_k – від'ємне число. Прийmemo тепер, що для будь-якого пункту k виконується співвідношення: $A_k = T_k + B$ і $B_k = B$, де B — сумарні запаси, що наявні у всіх пунктах;

4) вводимо величини $X_{ij}, i \neq j$, для дуг, які існують у вихідній мережі.

Розв'язання задачі за допомогою Microsoft Excel наведено у таблицях 1.5, 1.6, 1.7.

Таблиця 1.5 – Таблиця у Microsoft Excel (1)

	1	2	3	4	5	6	7	8
1	0	3		4		6		
2	6	0	7	1	1			
3		8	0					6
4	12	77		0	89	9		
5		26		5	0		6	7
6	15			4		0	6	
7					8	99	0	2
8			6		30		8	0
9								1

Ціна перевезення «в себе» дорівнює 0

Таблиця 1.6 – Таблиця у Microsoft Excel (2)

	1	2	3	4	5	6	7	8				B = 115
1	111	0		0		0					111	111
2	4	11	0	94	6						115	115
3		104	13					0			117	117
4	0	0		15	0	0					15	15
5		0		0	109		0	0			109	109
6	0			6		115	0				121	121
7					0	0	110	0			110	110
8			102		0	5	15				122	122
9								100			100	100
											0	
	115	115	115	115	115	115	115	115	0	0	920	
	115	115	115	115	115	115	115	115			920	920

Сума елементів стовпця

Сума елементів рядка

Обираємо Сервіс — Пошук розв'язку

Таблиця 1.7 – Таблиця у Microsoft Excel (3)

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	24	0	0	94	6	0	0	0	0	0
3	0	832	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	24	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	612	0	0	0	40	0	0	0
9	0	0	0	0	0	0	0	100	0	0
	0	0	0	0	0	0	0	0	0	0

За допомогою Microsoft Excel знаходимо суму елементів останньої таблиці: **1732**. Отже, мінімальна ціна перевезення становитиме 1732 грн.

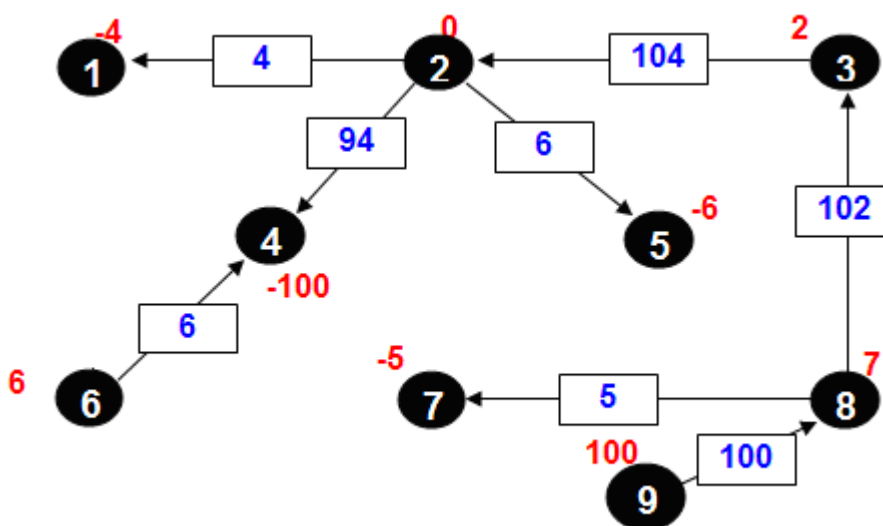


Рисунок 1.64 – Мережа після оптимізації в Excel

З рисунку 1.64 видно, що необхідно вивезти 100 м^3 дизпалива із пункту 9 у пункт 8. З пункту 8 забрати ще 7 м^3 дизпалива. З отриманих 107 м^3 5 м^3 відвезти у пункт 7, ліквідуючи там нестачу, а 102 м^3 повезти у пункт 3. Продовжуючи аналогічно перерозподіляти дизпаливо (рис. 1.64), ми ліквідуємо залишки й нестачу.

1.6 Індивідуальні завдання (мережеві моделі)

За даною мережею знайдіть максимальний потік, побудуйте

мінімальне остове дерево, знайдіть найкоротші шляхи між будь-якими вузлами мережі, розподіліть товарообіг по мережі.

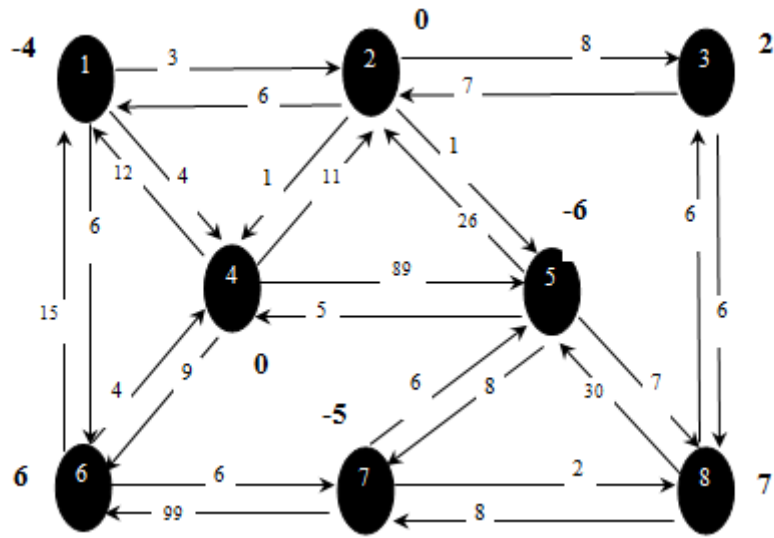


Рисунок 1.65 – Мережа (варіант 1)

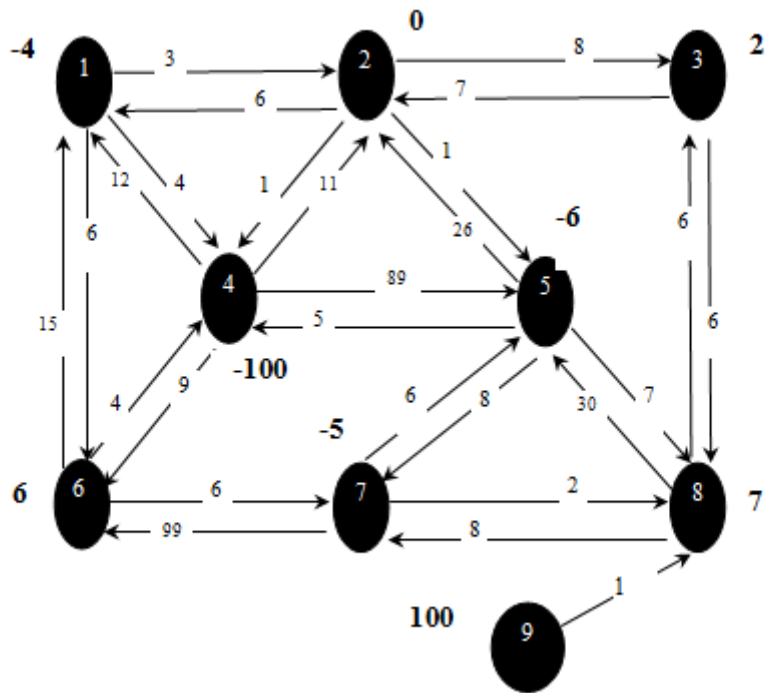


Рисунок 1.66 – Мережа (варіант 2)

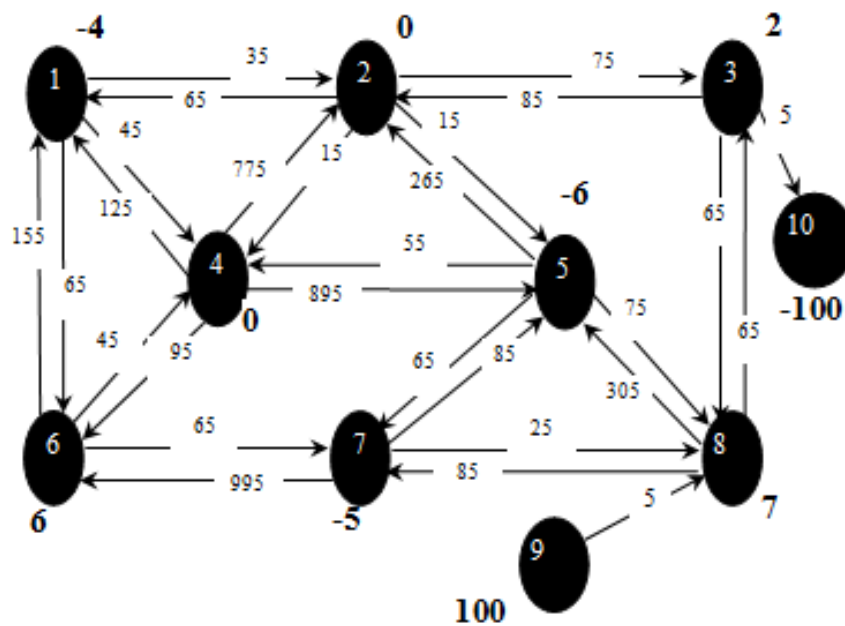


Рисунок 1.67 – Мережа (варіант 3)

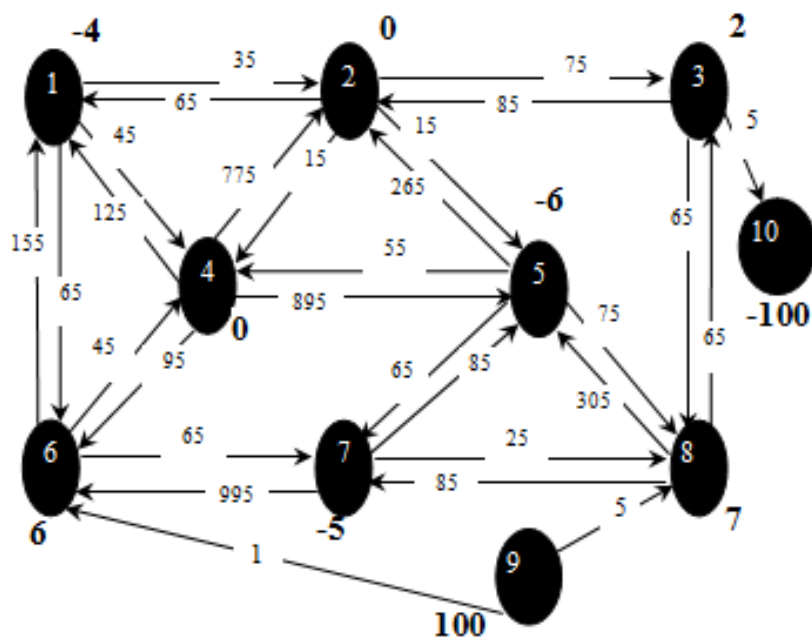


Рисунок 1.68 – Мережа (варіант 4)

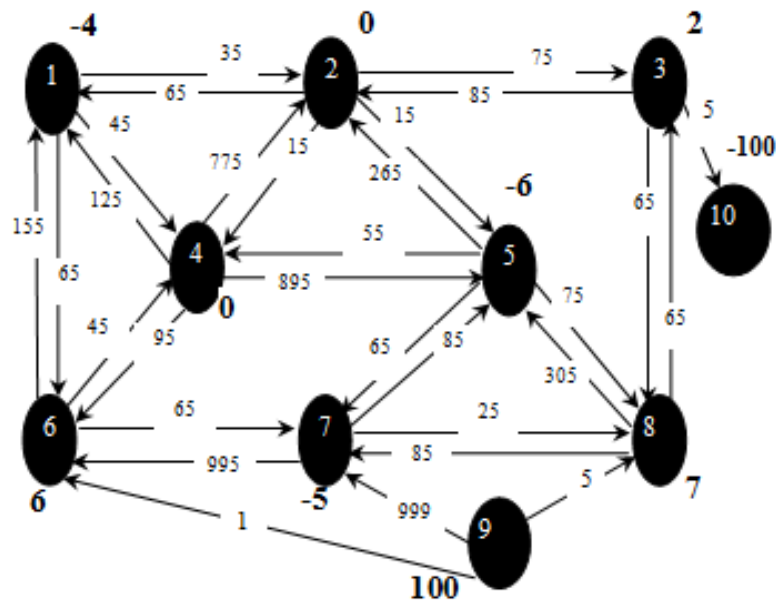


Рисунок 1.69 – Мережа (варіант 5)

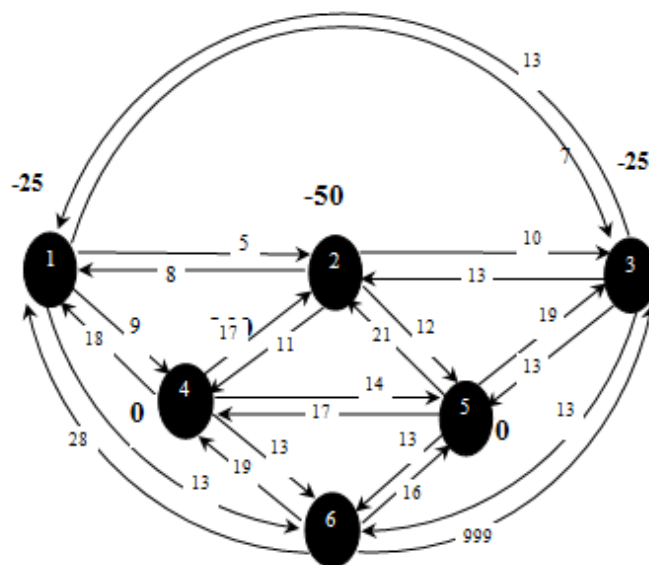


Рисунок 1.70 – Мережа (варіант 6)

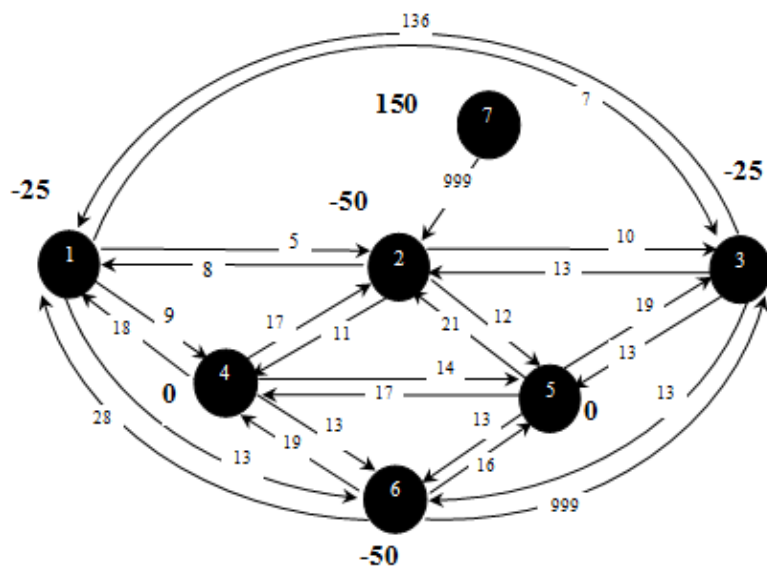


Рисунок 1.71 – Мережа (варіант 7)

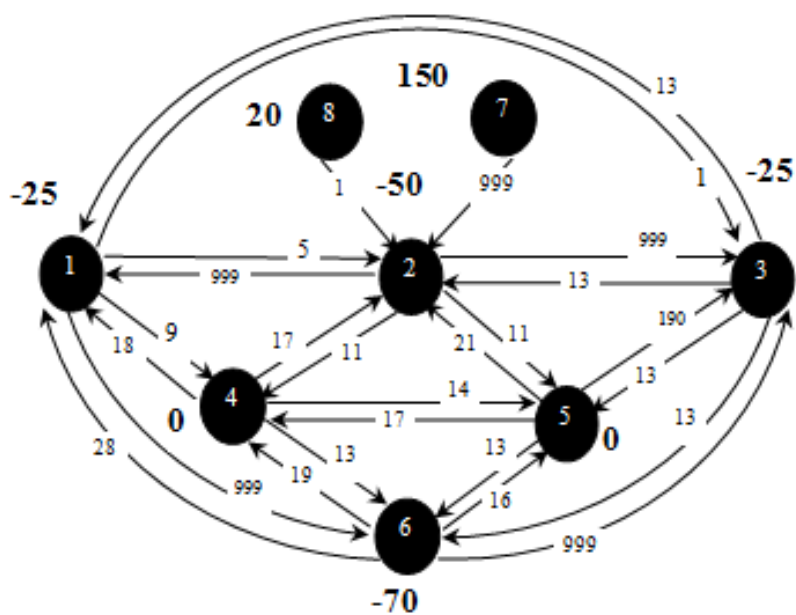


Рисунок 1.72 – Мережа (варіант 8)

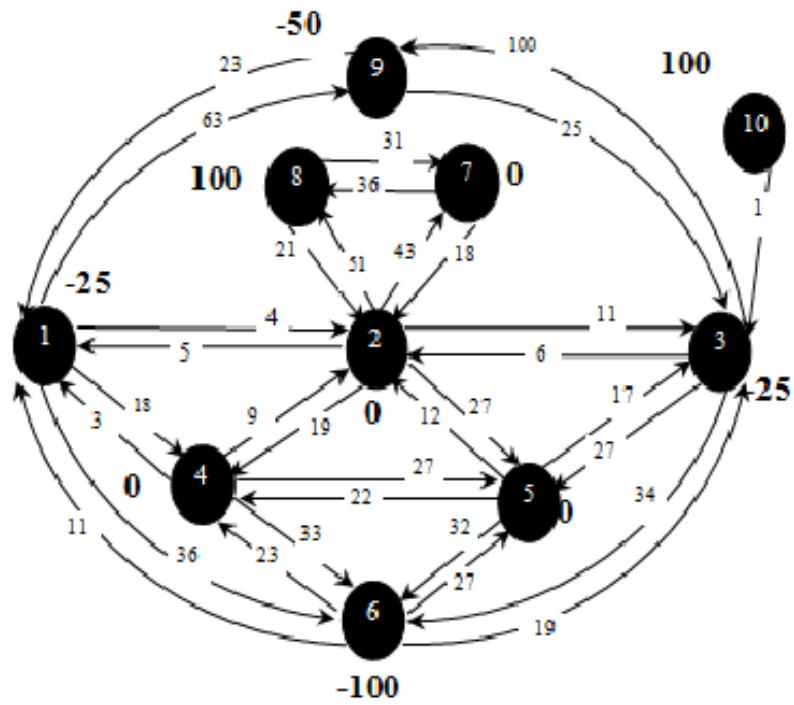


Рисунок 1.73 – Мережа (варіант 9)

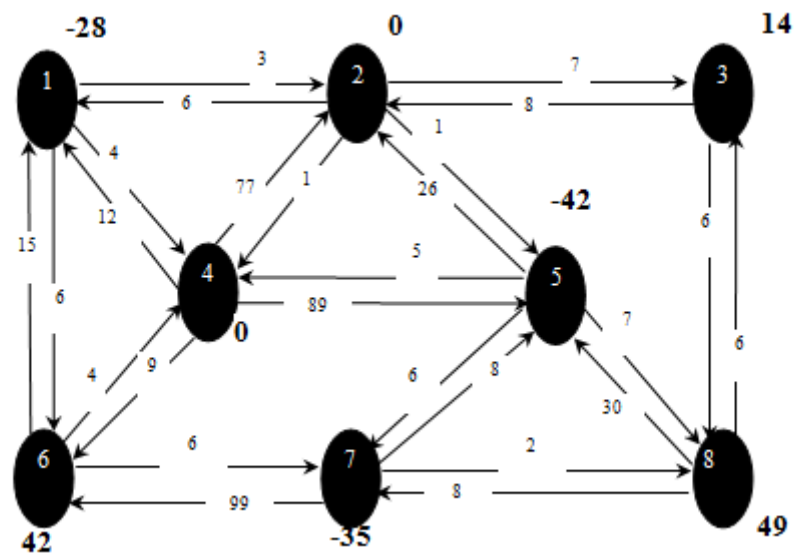


Рисунок 1.74 – Мережа (варіант 10)

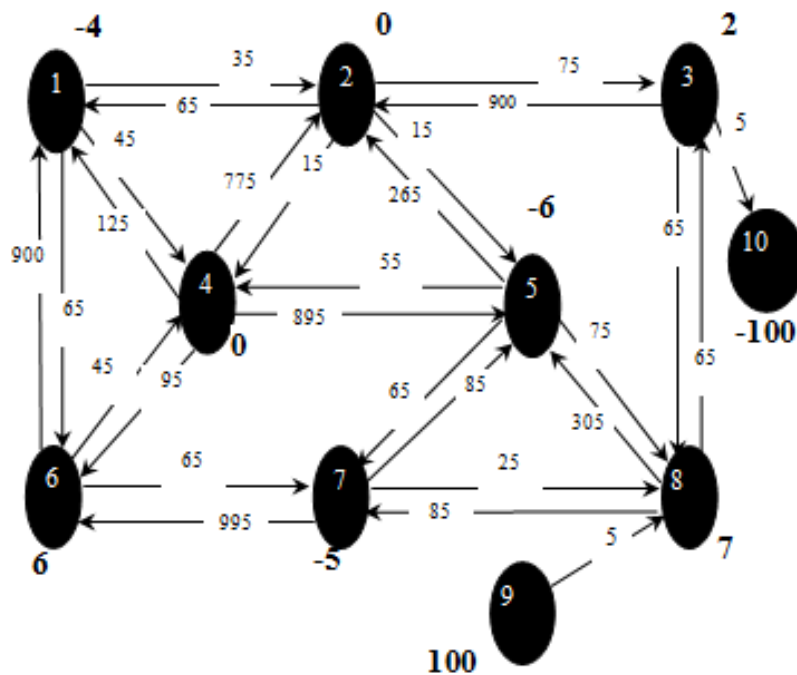


Рисунок 1.75 – Мережа (варіант 11)

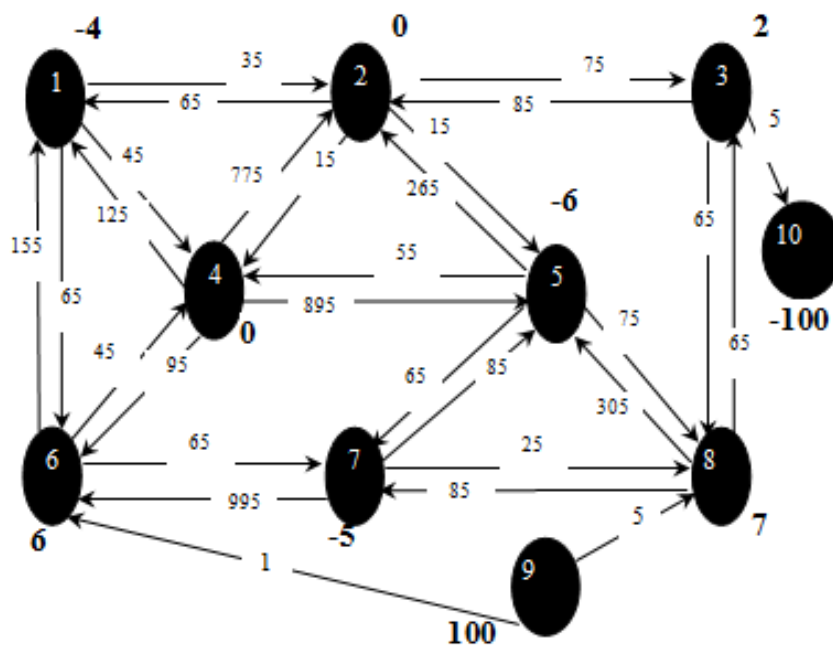


Рисунок 1.76 – Мережа (варіант 12)

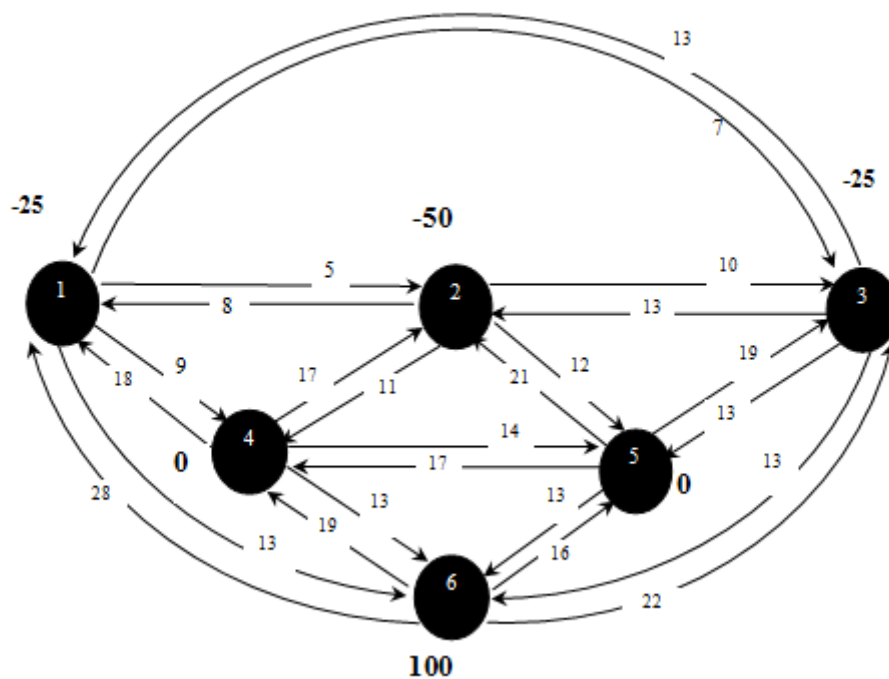


Рисунок 1.77 – Мережа (варіант 13)

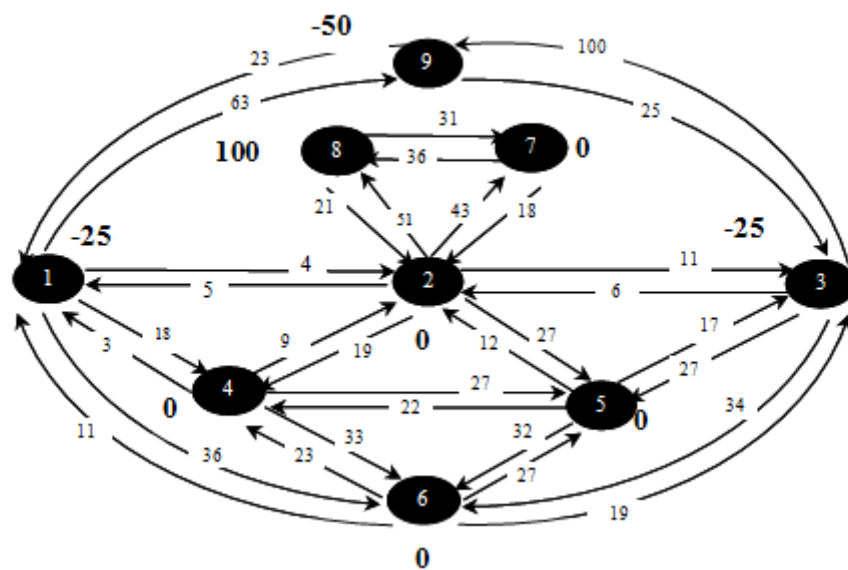


Рисунок 1.78 – Мережа (варіант 14)

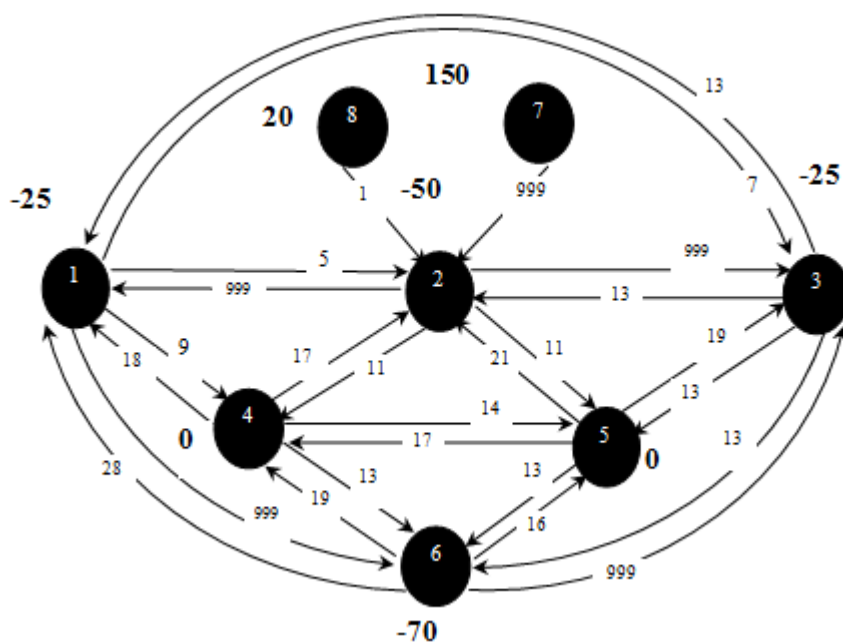


Рисунок 1.79 – Мережа (варіант 15)

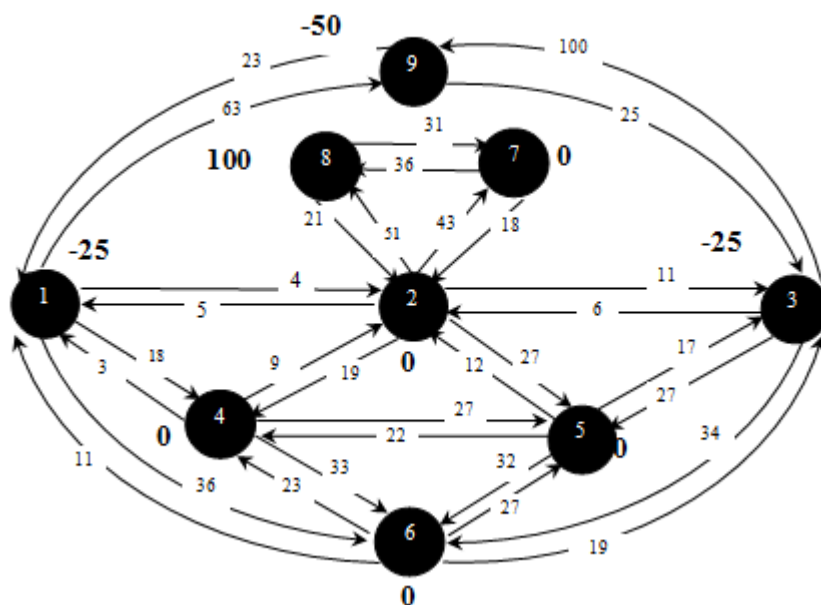


Рисунок 1.80 – Мережа (варіант 16)

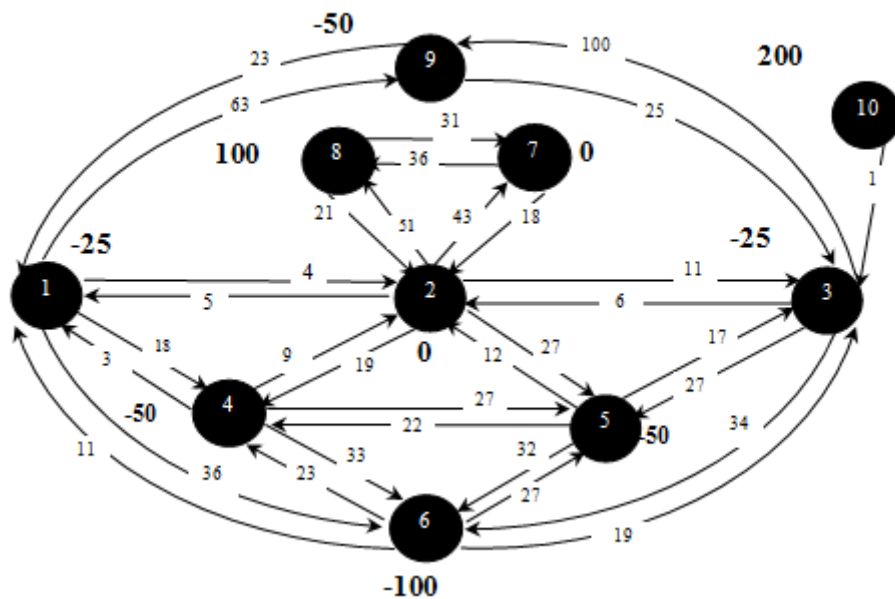


Рисунок 1.81 – Мережа (варіант 17)

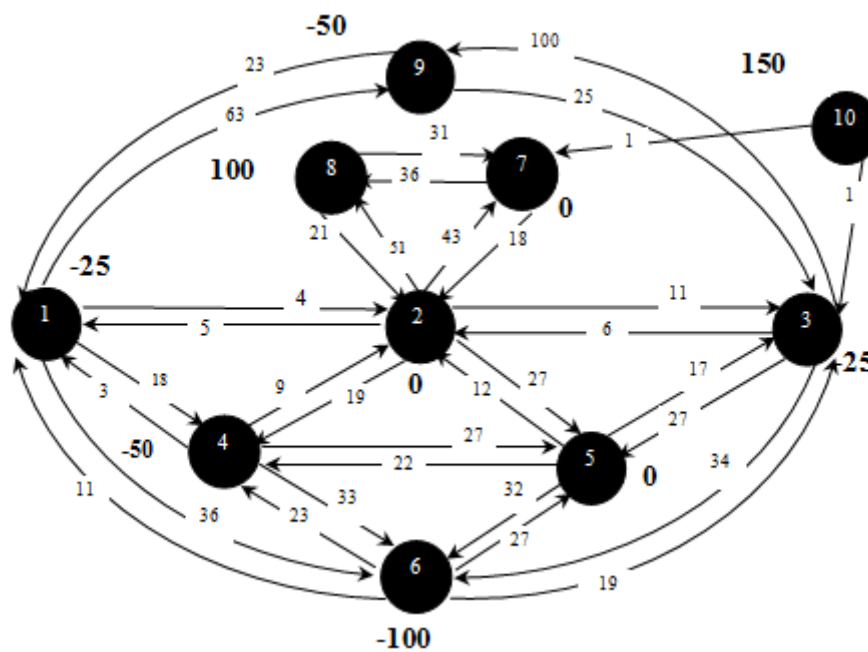


Рисунок 1.82 – Мережа (варіант 18)

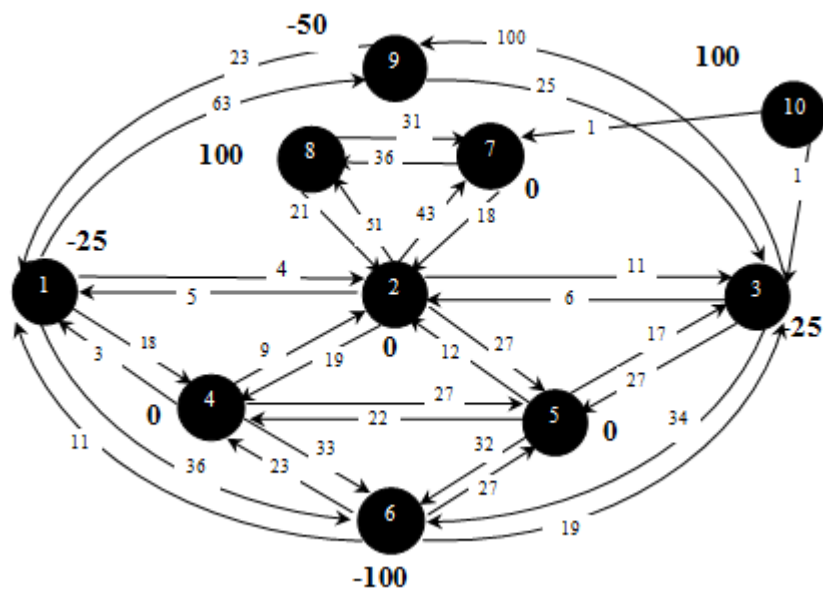


Рисунок 1.83 – Мережа (варіант 19)

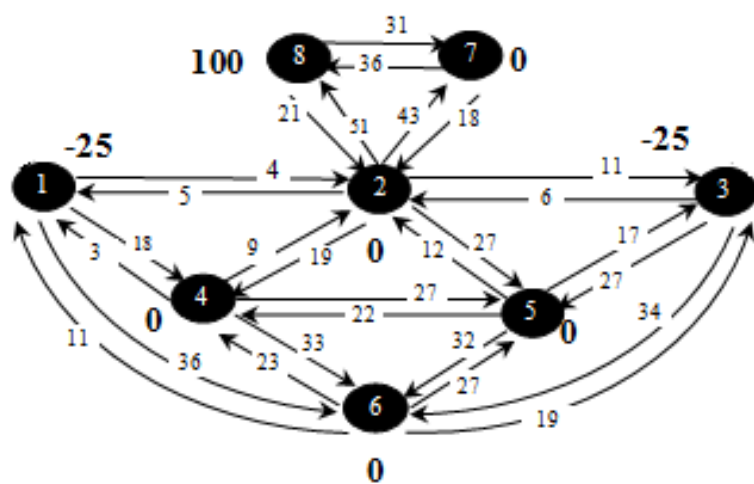


Рисунок 1.84 – Мережа (варіант 20)

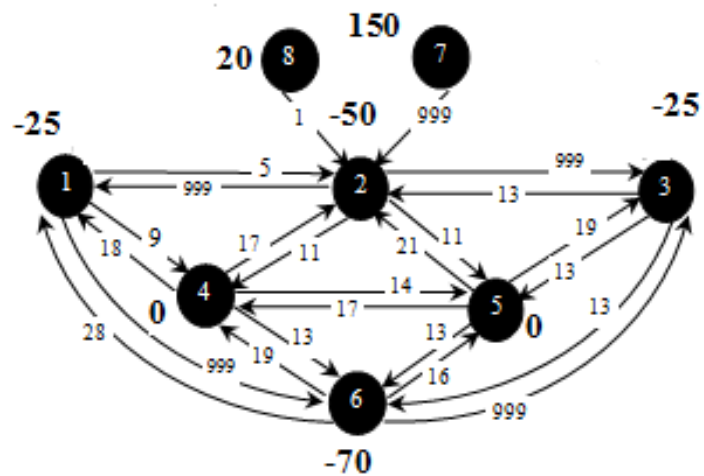


Рисунок 1.85 – Мережа (варіант 21)

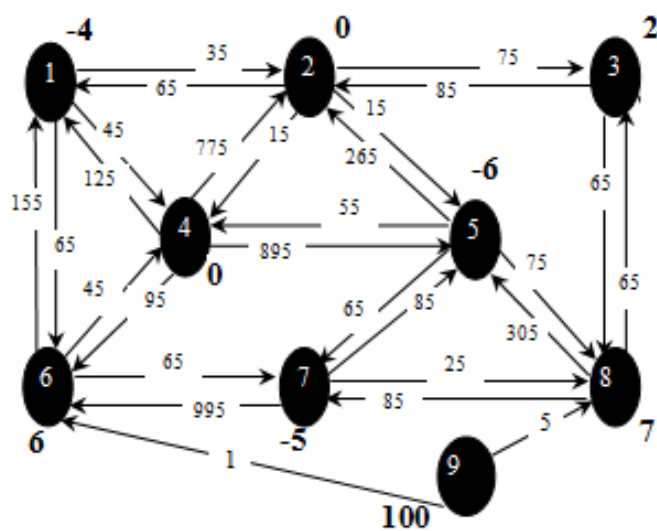


Рисунок 1.86 – Мережа (варіант 22)

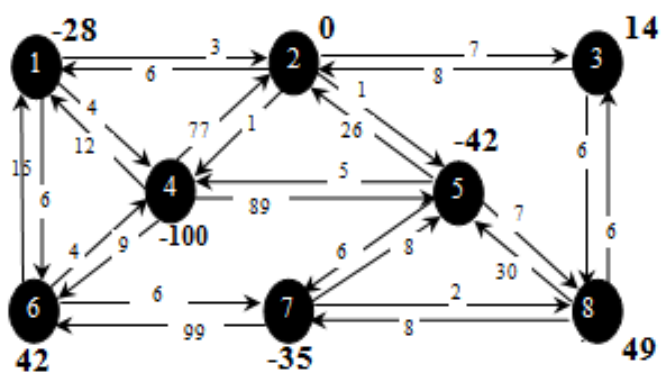


Рисунок 1.87 – Мережа (варіант 23)

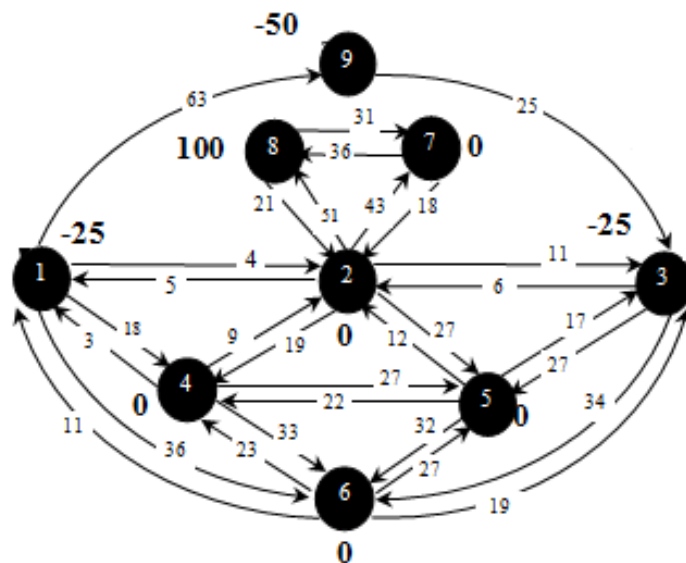


Рисунок 1.88 – Мережа (варіант 24)

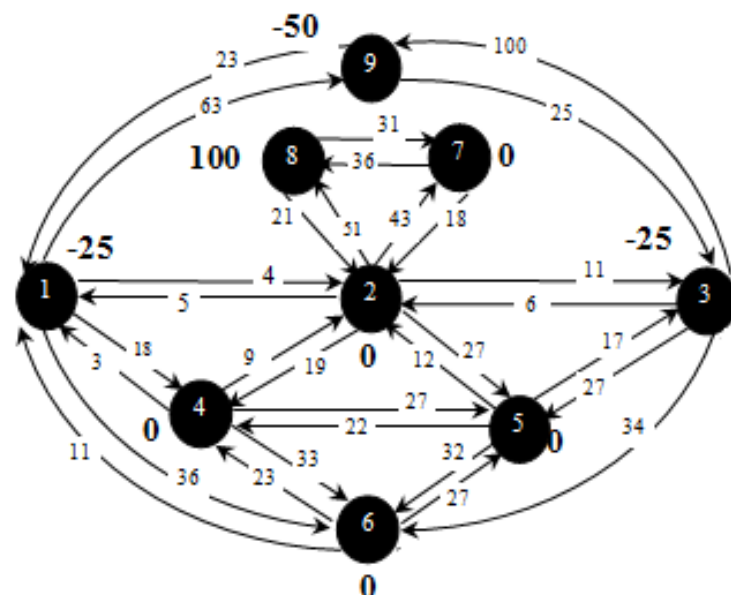


Рисунок 1.89 – Мережа (варіант 25)

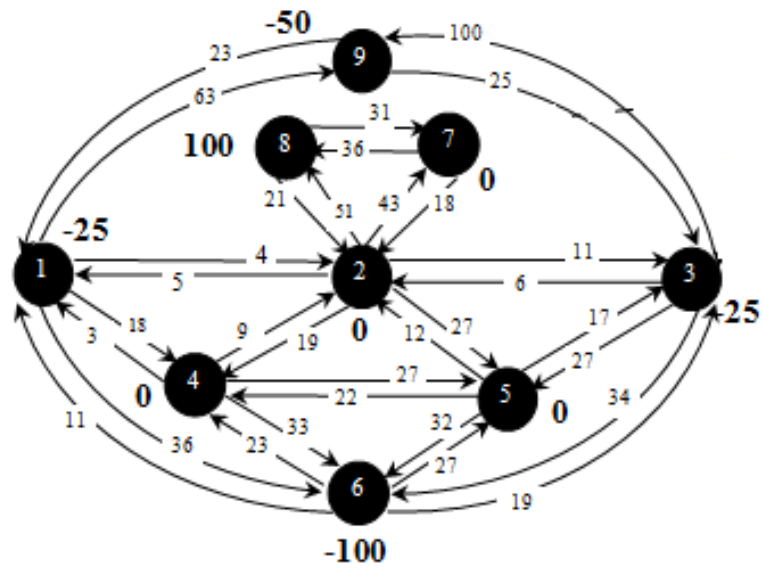


Рисунок 1.90 – Мережа (варіант 26)

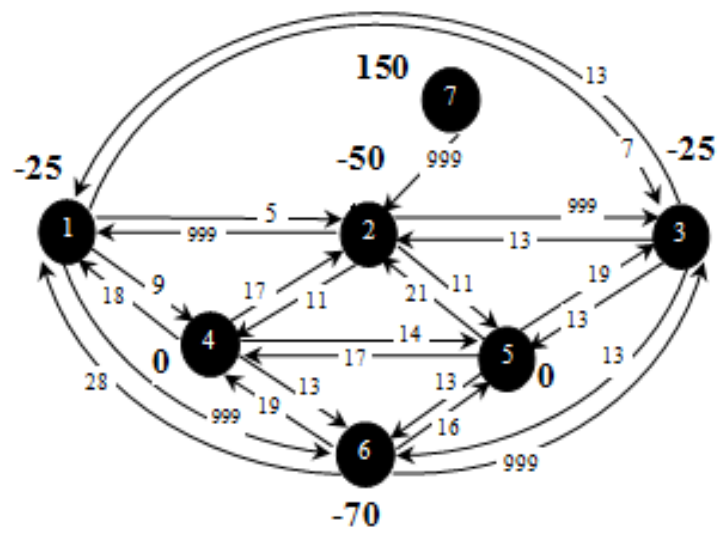


Рисунок 1.91 – Мережа (варіант 27)

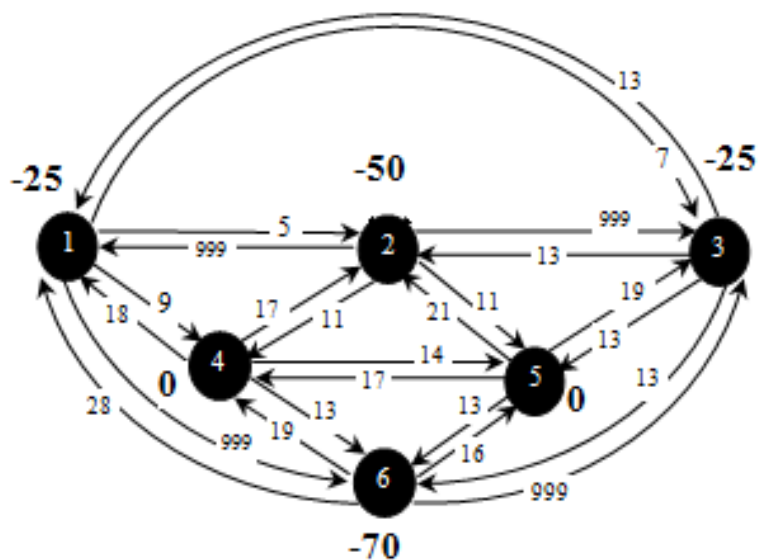


Рисунок 1.92 – Мережа (варіант 28)

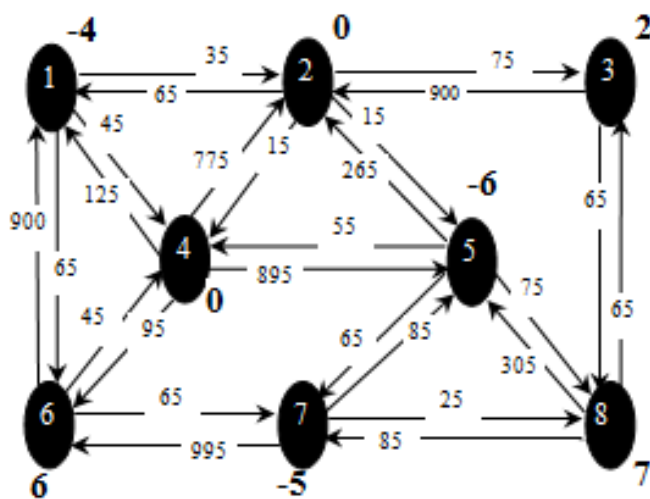


Рисунок 1.93 – Мережа (варіант 29)

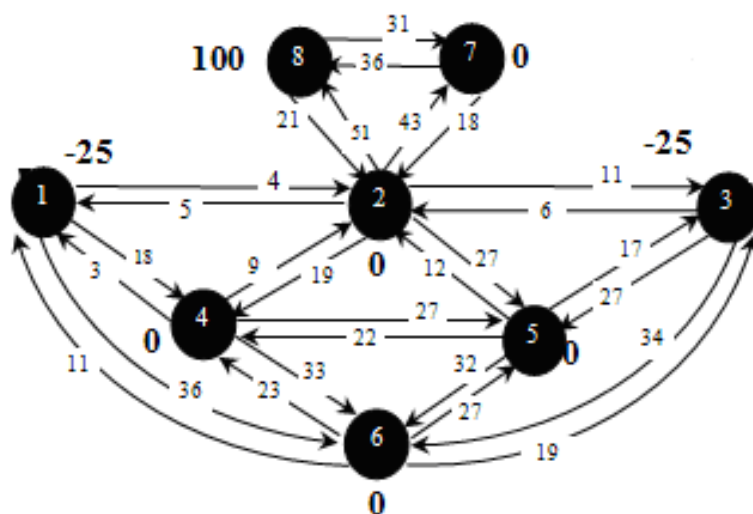


Рисунок 1.94 – Мережа (варіант 30)

РОЗДІЛ 2 КОМБІНАТОРНІ МОДЕЛІ. МОДЕЛЬ «БРОДЯЧОГО ТОРГОВЦЯ»

2.1 Загальні відомості про задачу комівояжера

Постановка задачі. Задача комівояжера (комівояжер — бродячий торговець) полягає в знаходженні найвигіднішого маршруту, що проходить через вказані міста хоча б по одному разу. В умовах завдання вказуються критерій вигідності маршруту (найкоротший, найдешевший тощо) і відповідні матриці відстаней. Зазвичай задано, що маршрут повинен проходити через кожне місто тільки один раз.

Існує багато різновидів узагальненої постановки задачі, зокрема, геометрична задача комівояжера (коли матриця відстаней відображає відстані між точками на площині), трикутна задача комівояжера (коли на матриці вартостей виконується нерівність трикутника), симетрична та асиметрична задачі комівояжера.

Прості методи розв'язання задачі комівояжера: повний лексичний перебір, метод включення найближчого міста, метод найдешевшого включення, метод мінімального кістяка дерева. На практиці застосовують різні модифікації зазначених методів: метод гілок і границь, метод генетичних алгоритмів, а також алгоритм мурашиної колонії.

Історична довідка. Невідомо, коли проблему комівояжера було досліджено вперше. Відомо лише, що в 1832 році була видана книжка з назвою «Комівояжер — як він має поводитись і що має робити для того, аби доставляти товар та мати успіх у своїх справах — поради старого Кур'єра», в якій описано проблему, але математичний апарат для її розв'язання не застосовується. Натомість, у ній запропоновано приклади маршрутів для деяких регіонів Німеччини та Швейцарії.

Перші згадки про проблему комівояжера в якості математичної задачі на оптимізацію належать Карлу Менгеру, який сформулював її в математичному колоквиумі в 1930 році так: «Ми називаємо проблемою жєнця (оскільки це питання виникає в кожного листоноші, зокрема, її вирішують багато мандрівників) завдання віднайти найкоротший шлях між скінченною множиною місць, відстань між якими відома».

Невдовзі з'явилась відома зараз назва «Задача мандруючого продавця», яку запропонував Гаслер Вітні з Принстонського Університету.

Разом із простотою формулювання та порівняною простотою знаходження гарних розв'язків задача комівояжера відрізняється тим, що визначення насправді оптимального шляху є досить складним завданням.

Через це, починаючи з другої половини 20-го століття, дослідження задачі комівояжера має не стільки практичне значення, скільки теоретичне (розробка нових алгоритмів оптимізації).

Багато сучасних методів дискретної оптимізації (метод ділення площиною, метод «гілок та границь» і різноманітні варіанти евристичних алгоритмів) було розроблено на прикладі задачі комівояжера.

У 1950-ті і 1960-ті роки задача комівояжера привернула увагу науковців у США та Європі. Важливий внесок у дослідження задачі зробили Джордж Данціг, Делберт Рей Фалкерсон і Селмер Джонсон, котрі в 1954 році в інституті RAND Corporation сформулювали проблему у вигляді задачі дискретної оптимізації та розробили метод відсікаючої площини для її розв'язання. Використовуючи новий метод, вони обчислили шлях для окремого набору вузлів (екземпляру проблеми) з 49 міст та довели, що не існує ще коротшого шляху. У 1960-ті та 1970-ті роки численні групи дослідників вивчали задачу з точки зору математики та її застосування, наприклад, в інформатиці, економіці, хімії та біології.

2.2 Реалізація моделі «бродячого торговця»

Реалізацію моделі «бродячого торговця» продемонструємо на конкретному прикладі.

Постановка задачі: у межах підготовки до чемпіонату Євро-2012 організовується автобусний екскурсійний маршрут по Донецьку. Екскурсія має починатися біля готелю «Донбас Палас». Основні пункти маршруту: дельфінарій «Немо», Донецький парк кованих фігур, планетарій, ботанічний сад, стадіон «Донбас Арена». Необхідно так спланувати екскурсійний маршрут, щоб довжина шляху, який проходить автобус, була мінімальною.

Пронумеруємо основні пункти маршруту:

- 1 – готель «Донбас Палас»;
- 2 – дельфінарій «Немо»;
- 3 – Донецький парк кованих фігур;
- 4 – планетарій;
- 5 – ботанічний сад;
- 6 – стадіон «Донбас Арена».

У таблиці 2.1 вказані відстані між пунктами екскурсії (у км).

Таблиця 2.1 – Відстані між пунктами екскурсії

	1	2	3	4	5	6
1	N	21	34	25	8	23
2	12	N	19	22	45	35
3	15	13	N	24	40	38
4	37	42	9	N	14	23
5	29	36	18	6	N	43
6	26	31	30	16	7	N

Для розв'язання даної задачі використаємо два алгоритма методу гілок і границь реалізації моделі «бродячого торговця».

Алгоритм 1

1. Знаходження верхньої границі

Починаємо зі знаходження верхньої границі задачі (за будь-яким припустимим маршрутом):

$$S(1 - 5 - 4 - 6 - 3 - 2 - 1) = 92.$$

Важливо, щоб маршрут, за яким знаходимо верхню границю, проходив через усі пункти й в жоден пункт не доводилося заходити двічі.

2. Побудова дерева (перша ітерація) (рис. 2.1).

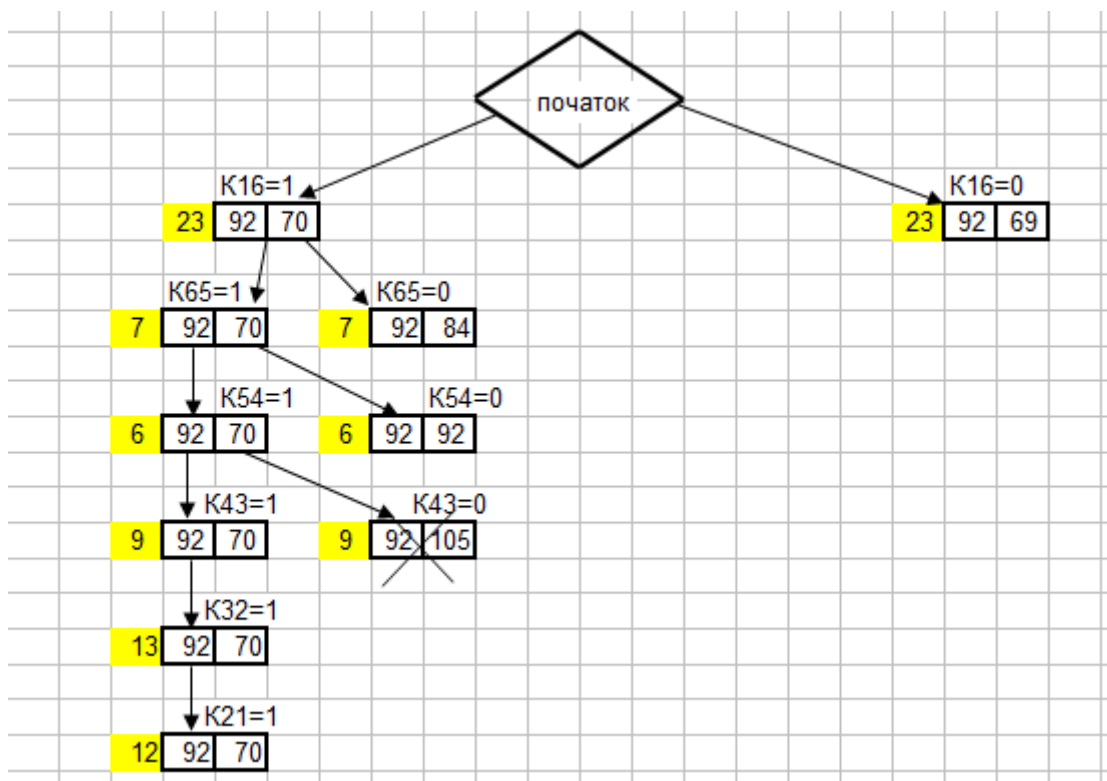


Рисунок 2.1 – Дерево алгоритму 1, перша ітерація

Пояснення до дерева алгоритму:

1) На рисунку 2.1 символ $K16 = 1$ означає множину маршрутів,

що містять ребро 1-6, а $K_{16} = 0$ – множину маршрутів, що не містять ребро 1-6.

- 2) Прямокутники біля кожної стрілки розбиті на дві комірки: у тій, що знаходиться ліворуч, записана верхня границя задачі (див. пункт 1 алгоритму), а в тій, що праворуч – нижня границя.
- 3) Послідовність руху визначається за найменшими відстанями, зокрема, після K_{16} (обрали згідно з клітинкою початку руху) ми розглядаємо K_{65} , оскільки найближчим до пункту 6 є пункт 5. При цьому не забуваємо, що у кожний пункт можемо заходити лише один раз.
- 4) Гілка перекреслюється (обривається), якщо нижня границя більша за верхню.

Нижні границі (ij) знаходимо за таким алгоритмом:

у випадку, коли $K_{ij} = 1$:

- 1) у матриці відстаней (таблиця 2.1) елементи i -го рядка замінюємо символами N (вважаємо, що в цих комірках символ нескінченності), окрім елемента з індексами ij ;
- 2) у матриці відстаней (таблиця 2.1) елементи j -го стовпця замінюємо символами N , окрім елемента з індексами ij ;
- 3) у матриці відстаней (таблиця 2.1) елемент з індексами ji замінюємо символом N ;
- 4) у матриці, що була отримана в результаті перетворень, що описані вище, знаходимо мінімальні елементи рядків і віднімаємо їх від елементів цих рядків;
- 5) у новій матриці знаходимо мінімальні елементи стовпців;
- 6) знаходимо суму мінімальних елементів, що отримані у пунктах 4 і 5. Це і буде нижня границя.

У випадку, коли $K_{ij} = 0$:

1) у матриці відстаней (таблиця 2.1) елемент з індексами ij замінюємо символом N (вважаємо, що у цих комірках символ нескінченності);

2) виконуємо пункти 4,5,6 попереднього алгоритму.

Продемонструємо на прикладі нашої задачі знаходження нижньої границі (65).

Маємо матрицю відстаней:

	1	2	3	4	5	6
1	N	21	34	25	8	23
2	12	N	19	22	45	35
3	15	13	N	24	40	38
4	37	42	9	N	14	23
5	29	36	18	6	N	43
6	26	31	30	16	7	N

Рисунок 2.2 – Матриця відстаней (1)

Виконуємо перетворення 1, 2, 3:

	1	2	3	4	5	6
1	N	21	34	25	N	23
2	12	N	19	22	N	35
3	15	13	N	24	N	38
4	37	42	9	N	N	23
5	29	36	18	6	N	N
6	N	N	N	N	7	N

Рисунок 2.3 – Матриця відстаней (2)

Знаходимо мінімальні елементи рядків:

	1	2	3	4	5	6	min
1	N	21	34	25	N	23	21
2	12	N	19	22	N	35	12
3	15	13	N	24	N	38	13
4	37	42	9	N	N	23	9
5	29	36	18	6	N	N	6
6	N	N	N	N	7	N	7

Рисунок 2.4 – Матриця відстаней (3)

Віднімаємо мінімальні елементи рядків від інших елементів:

	1	2	3	4	5	6
1	N	0	13	4	N	2
2	0	N	7	10	N	23
3	2	0	N	11	N	25
4	28	33	0	N	N	14
5	23	30	12	0	N	N
6	N	N	N	N	0	N

Рисунок 2.5 – Матриця відстаней (4)

Знаходимо мінімальні елементи стовпців:

	1	2	3	4	5	6
1	N	0	13	4	N	2
2	0	N	7	10	N	23
3	2	0	N	11	N	25
4	28	33	0	N	N	14
5	23	30	12	0	N	N
6	N	N	N	N	0	N
min	0	0	0	0	0	2

Рисунок 2.6 – Матриця відстаней (5)

Знаходимо нижню границю (65):

$$21 + 12 + 13 + 9 + 6 + 7 + 0 + 0 + 0 + 0 + 0 + 2 = 70.$$

3. Побудова дерева (друга ітерація) (рис. 2.7).

Піднімаємося вгору за гілками й змінюємо 92 на 70. Знаходимо необірвані гілки й продовжуємо процес заново.

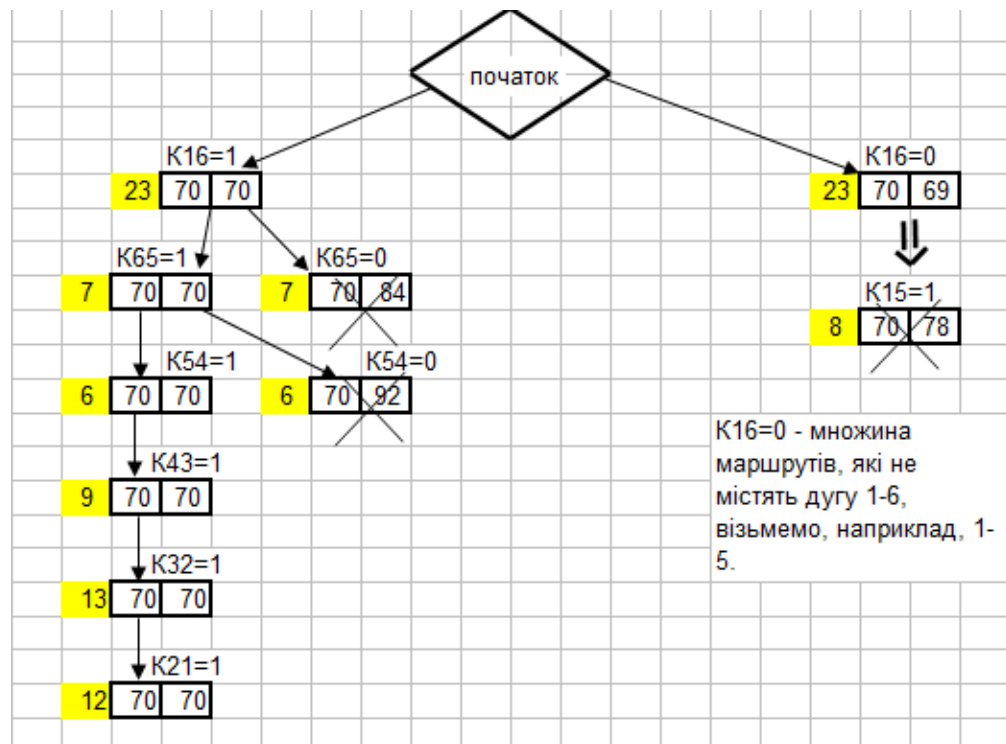


Рисунок 2.7 – Дерево алгоритму 1, друга ітерація

Необірваних гілок немає, тому знайдений маршрут є оптимальним.

Отже, оптимальний шлях 1 – 6 – 5 – 4 – 3 – 2 – 1. Його довжина: 70.

Зверніть увагу! У більшості випадків кількість ітерацій буде меншою, якщо застосувати алгоритм знаходження клітинки початку руху:

- 1) знаходимо мінімальний елемент у кожному рядку;
- 2) знаходимо мінімальний елемент у кожному рядку з тих, що залишилися (зрозуміло, що ці елементи будуть не меншими за ті, що знайдені у попередньому пункті цього алгоритму);
- 3) знаходимо різницю між елементами, що отримані в пункті 2, і елементами, що отримані в пункті 1. У результаті матимемо стовпець з чисел;
- 4) у стовпці, що знайдений в пункті 3, обираємо найбільший елемент;
- 5) у рядку, якому відповідає найбільший елемент, знаходимо найменший. Клітинка перебування цього елемента й буде клітинкою початку руху.

У таблиці 2.2 наведемо реалізацію даного алгоритму для нашої задачі.

Таблиця 2.2 – Реалізація алгоритму знаходження клітинки початку руху

	1	2	3	4	5	6	<i>min</i> рядка	<i>min</i> рядка з тих, що залишилися	різниця мінімумів
1	N	21	34	25	8	23	8	21	13
2	12	N	19	22	45	35	12	19	7
3	15	13	N	24	40	38	13	15	2
4	37	42	9	N	14	23	9	14	5
5	29	36	18	6	N	43	6	18	12
6	26	31	30	16	7	N	7	16	9

Отже, доцільно починати рух із клітинки 15, тобто клітинки першого рядка п'ятого стовпця.

Алгоритм 2

На відміну від алгоритму 1 (бінарної розбивки), алгоритм 2 передбачає побудову дерева, в якому відображаються всі можливі напрямки руху від початкового пункту. Проте в іншому алгоритми не відрізняються. На рисунках 2.8 і 2.9 зображені дерева алгоритму 2.

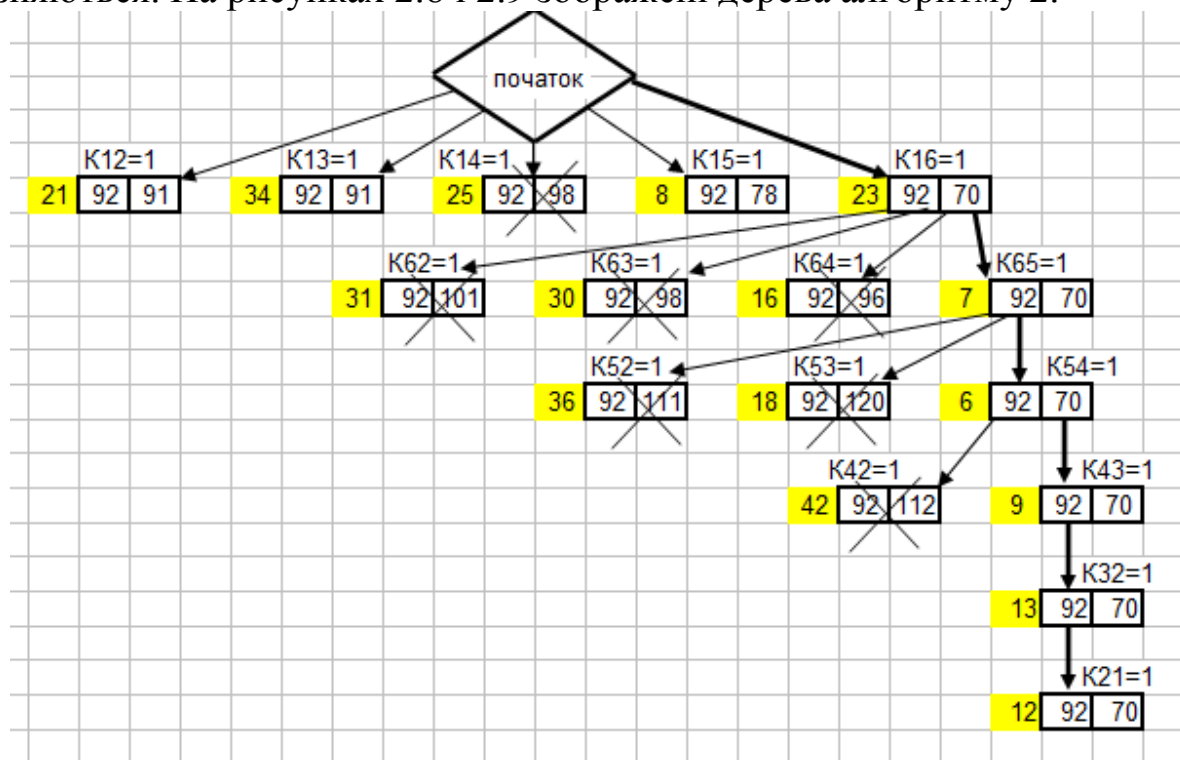


Рисунок 2.8 – Дерево алгоритму 2, перша ітерація

На рисунку 2.8 ми продовжуємо саме шлях 1 – 6, оскільки йому відповідає найменша нижня границя.

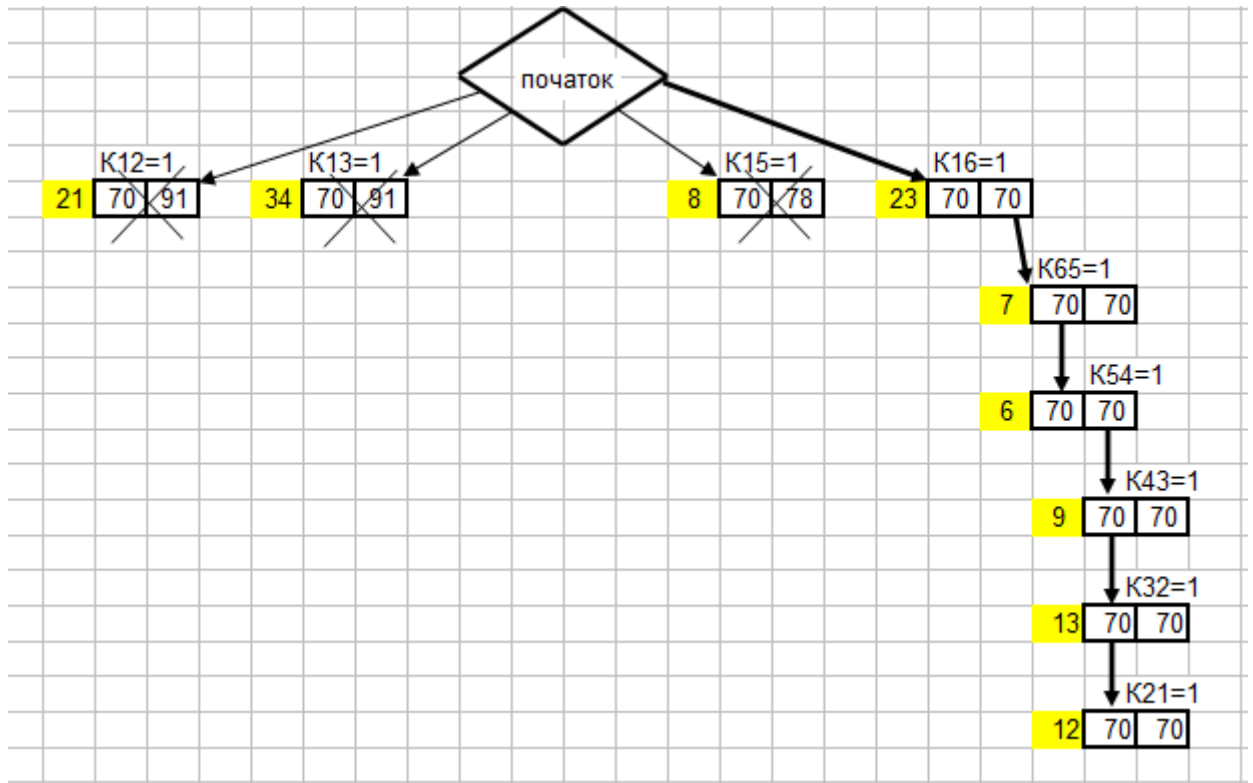


Рисунок 2.9 – Дерево алгоритму 2, друга ітерація

Оптимальний шлях 1 – 6 – 5 – 4 – 3 – 2 – 1 (готель «Донбас Палас» – стадіон «Донбас Арена» – ботанічний сад – планетарій – Донецький парк кованих фігур – дельфінарій «Немо» – готель «Донбас Палас».

Його довжина: 70 км.

2.3 Індивідуальні завдання (модель «бродячого торговця»)

Реалізуйте алгоритм-1 і алгоритм 2 для матриць відстаней, що наведені у таблиці 2.3. Результати порівняйте.

Таблиця 2.3 – Індивідуальні завдання

1	N	7	81	34	17	9	2	N	5	7	45	12	23	3	N	15	21	12	45	30
	10	N	56	42	18	12		8	N	22	14	17	25		9	N	14	25	42	33
	27	35	N	37	19	4		16	33	N	35	40	27		11	8	N	16	20	29
	31	2	29	N	56	67		19	20	34	N	42	28		6	17	40	N	27	24
	61	8	17	23	N	51		23	44	31	13	N	20		13	43	35	18	N	22
	24	60	77	21	16	N		36	43	18	26	10	N		36	41	28	34	10	N

Продовження таблиці 2.3

4	N	21	34	25	8	23	5	N	33	44	22	11	15	6	N	5	9	14	10	21
	12	N	19	22	45	35		36	N	25	21	41	16		33	N	32	27	15	11
	15	13	N	24	40	38		8	23	N	34	27	30		34	15	N	20	22	41
	37	42	9	N	14	23		12	9	17	N	20	31		36	24	28	N	45	42
	29	36	18	6	N	43		32	10	14	24	N	45		25	17	40	44	N	39
	26	31	30	16	7	N		19	28	18	40	26	N		12	18	38	45	6	N
7	N	16	21	31	45	9	8	N	11	14	21	32	17	9	N	6	9	21	19	27
	10	N	22	37	30	12		10	N	12	33	25	28		20	N	33	41	22	31
	14	20	N	29	41	32		23	9	N	24	40	45		29	32	N	10	15	17
	12	17	8	N	23	33		20	8	13	N	41	29		23	18	42	N	45	34
	26	28	34	19	N	44		30	15	19	27	N	37		16	35	11	43	N	37
	25	8	38	36	40	N		34	39	18	43	7	N		12	14	27	30	40	N
10	N	33	16	19	25	40	11	N	22	14	10	19	27	12	N	27	23	12	29	40
	29	N	34	17	41	42		25	N	15	28	31	42		41	N	34	18	24	33
	34	9	N	15	24	35		21	9	N	41	23	33		43	28	N	14	35	38
	43	21	14	N	8	37		26	29	39	N	40	45		9	15	19	N	36	17
	39	45	22	18	N	23		24	20	11	16	N	34		10	20	30	45	N	39
	10	15	30	20	11	N		7	6	8	37	36	N		11	21	22	37	6	N
13	N	24	33	20	12	19	14	N	14	22	33	44	15	15	N	11	17	33	27	30
	9	N	18	21	8	34		21	N	16	23	32	8		15	N	31	40	18	21
	40	31	N	29	17	25		25	41	N	35	27	30		36	41	N	16	32	23
	30	16	11	N	7	35		34	12	9	N	17	36		8	12	34	N	37	44
	45	39	15	23	N	26		18	26	28	40	N	37		14	25	39	15	N	9
	27	35	41	44	10	N		11	19	29	39	45	N		19	45	10	22	29	N
16	N	23	16	41	29	35	17	N	41	23	15	30	26	18	N	9	22	36	41	42
	33	N	40	36	24	15		9	N	16	25	31	39		23	N	10	29	35	16
	8	17	N	25	45	39		17	10	N	33	27	37		25	28	N	11	31	40
	37	9	27	N	38	34		41	44	8	N	35	12		38	15	18	N	8	30
	26	28	22	10	N	11		32	38	45	11	N	40		45	44	33	27	N	7
	31	32	42	14	12	N		16	18	19	20	14	N		12	18	19	20	21	N
19	N	22	8	34	41	27	20	N	44	39	8	27	30	21	N	32	12	39	24	27
	14	N	30	39	28	19		15	N	28	17	21	23		14	N	22	23	25	45
	18	23	N	29	10	45		24	35	N	37	12	38		40	41	N	37	38	10
	44	12	20	N	24	35		26	40	41	N	22	10		17	18	19	N	8	26
	16	17	26	33	N	11		23	11	32	25	N	18		28	11	31	33	N	44
	37	38	21	9	25	N		16	20	9	29	43	N		42	36	35	9	34	N

Закінчення таблиці 2.3

22	N	10	15	25	35	45	23	N	11	17	21	32	22	24	N	35	12	39	22	19
	16	N	26	36	37	9		23	N	24	9	35	45		7	N	18	21	38	28
	12	27	N	38	40	28		33	37	N	42	19	12		20	23	N	10	33	40
	29	30	14	N	33	22		25	26	18	N	8	27		15	16	25	N	27	11
	17	23	32	8	N	41		38	41	10	44	N	28		29	9	34	37	N	42
	42	43	44	34	11	N		16	43	19	20	30	N		43	44	45	36	8	N
25	N	17	12	22	34	41	26	N	10	18	38	40	20	27	N	18	22	35	12	37
	25	N	37	40	28	14		22	N	32	35	12	41		36	N	14	42	27	26
	35	45	N	10	26	27		23	26	N	36	37	11		16	17	N	8	32	23
	31	30	20	N	9	23		8	30	31	N	15	17		15	13	19	N	45	44
	29	8	39	42	N	43		16	19	14	39	N	42		40	30	20	25	N	10
	15	16	38	44	33	N		34	44	43	9	45	N		28	11	29	33	38	N
28	N	40	22	35	36	9	29	N	44	34	14	24	16	30	N	34	11	25	30	40
	25	N	11	33	39	40		22	N	42	32	43	8		10	N	23	36	33	45
	26	28	N	17	10	19		23	11	N	39	40	41		27	37	N	41	20	9
	16	23	34	N	38	20		25	26	10	N	19	18		16	27	29	N	14	30
	30	45	44	8	N	37		15	45	30	20	N	21		45	15	17	19	N	24
	20	12	15	30	29	N		36	37	38	28	9	N		34	22	26	12	38	N

РОЗДІЛ 3 МОДЕЛІ КЕРУВАННЯ ЗАПАСАМИ

3.1 Основні поняття

Задачі керування запасами – значний клас економічних задач дослідження операцій, розв’язання яких має важливе значення для промислових підприємств і комерційних фірм. Правильне й своєчасне визначення оптимальної стратегії керування запасами, а також нормативного рівня запасів дозволяє визволити значні оборотні кошти, що заморожені у вигляді запасів, що підвищує ефективність використання ресурсів.

Розглянемо основні поняття й характеристики моделей керування запасами.

Попит. Попит на продукт, що запасується, може бути *детермінованим* (у найпростішому випадку – постійним у часі) або *випадковим*. Випадковість попиту описується або випадковим моментом попиту, або випадковим обсягом попиту в детерміновані або випадкові моменти часу.

Поповнення складу. Поповнення складу може здійснюватися або періодично через певні проміжки часу, або після вичерпання запасів, тобто зниження їх до деякого рівня.

Обсяг замовлення. При періодичному поповненні й випадковому вичерпанні запасів обсяг замовлення може залежати від того стану, що спостерігається в момент подачі замовлення.

Час доставки. В ідеалізованих моделях керування запасами передбачається, що поповнення доставляється на склад миттєво. В інших моделях розглядається затримка постачань на фіксований або випадковий інтервал часу.

Вартість постачання. Як правило, передбачається, що вартість кожного постачання складається з двох компонентів – разових витрат, що не залежать від обсягу партії, і витрат, що залежать (найчастіше – лінійно) від обсягу партії.

Витрати на зберігання. У більшості моделей керування запасами вважають обсяг складу практично необмеженим, а в якості контролюючої величини виступає обсяг збережених запасів. При цьому вважають, що за збереження кожної одиниці запасу в одиницю часу стягується певна плата.

Штраф за дефіцит. Будь-який склад створюється для того, щоб запобігти дефіциту певного типу виробів у системі, що обслуговується. Відсутність запасу в потрібний момент призводить до збитків, що

пов'язані з простим устаткуванням, неритмічністю виробництва тощо. Ці збитки надалі будемо називати *штрафом за дефіцит*.

Номенклатура запасу. У найпростіших випадках передбачається, що на складі зберігається запас однотипних виробів або однорідного продукту. У більш складних випадках розглядається *багатономенклатурний запас*.

Структура складської системи. У науці найповніше досліджені математичні моделі одиночного складу. Однак на практиці зустрічаються й більш складні структури: ієрархічні системи складів із різними періодами поповнення й часом доставки замовлень, з можливістю обміну запасами між складами одного рівня ієрархії тощо.

В якості критерію ефективності прийнятої стратегії керування запасами виступає *функція розходів (витрат)*, що встановлює сумарні витрати на збереження й постачання продукту, що запасється (в тому числі збитки від псування продукту при збереженні та його моральному старінні, втрати прибутку від омертвіння капіталу тощо) і витрати на штрафи.

Керування запасами полягає у відшуванні такої стратегії поповнення й витрати запасів, за якої функція витрат приймає мінімальне значення.

Нижче розглядаються найпростіші моделі керування запасами.

3.2 Детерміновані моделі керування запасами

Нехай функції $A(t)$, $B(t)$ і $R(t)$ виражають відповідно поповнення запасів, їхню витрату й попит на продукцію, що запасється, за проміжок часу $[0, t]$. У моделях керування запасами зазвичай використовуються похідні цих функцій за часом $a(t)$, $b(t)$, $r(t)$, що називають відповідно *інтенсивностями поповнення, витрати й попиту*.

Якщо функції $a(t)$, $b(t)$, $r(t)$ – не випадкові величини, то модель керування запасами вважається *детермінованою*. Якщо хоча б одна з них носить випадковий характер – *стохастичною*.

Якщо всі параметри моделі не змінюються в часі, вона називається *статичною*, у протилежному випадку – *динамічною*. Статичні моделі використовуються, коли приймається разове рішення про рівень запасів на визначений період, а динамічні – у випадку прийняття послідовних рішень про рівні запасу або коректування раніше прийнятих рішень з урахуванням поточних змін.

3.2.1 Реалізація детермінованої моделі керування запасами

Рівень запасу в момент t визначається основним рівнянням запасів:

$$J(t) = J_0 + A(t) - B(t), \quad (3.1)$$

де J_0 — початковий запас у момент $t = 0$.

Рівняння (3.1) найчастіше використовується в інтегральній формі:

$$J(t) = J_0 + \int_0^t a(t) dt - \int_0^t b(t) dt. \quad (3.2)$$

Приклад 3.1. На підприємстві «ЗАЗ» інтенсивність надходження деталей із багатопозиційного прес-автомату «FAGOR» до складу готової продукції становить 5 дет/хв на початку зміни. Протягом першої години цей показник лінійно зростає, досягаючи 10 дет/хв, і потім залишається незмінним до кінця зміни (після семи годин роботи автомат вимикають і інженер з експлуатації проводить його огляд). Надходження деталей на склад відбувається безупинно протягом усіх семи годин зміни, а вивезення деталей зі складу виконується тільки наприкінці роботи. Необхідно записати рівняння для рівня запасу в довільний момент часу t , використовуючи його, знайти кількість деталей на складі:

- а) через 30 хв після початку роботи;
- б) наприкінці зміни.

Розв'язання. За умовою протягом зміни не відбувається видачі деталей зі складу, тобто $b(t) = 0$. Інтенсивність поповнення запасу протягом першої години лінійно зростає, тобто

$$a(t) = kt + b. \quad (3.3)$$

Крім того,

$$a(0) = 5, a(60) = 10, \quad (3.4)$$

Рівності (3.4) були отримані з умови, що на початку зміни ($t = 0$) інтенсивність становить 5 дет/хв, а наприкінці години ($t = 60$) інтенсивність становить 10 дет/хв.

Знаючи залежність (3.3) і умови (3.4), можемо знайти k і b :

$$\begin{cases} 5 = k * 0 + b, \\ 10 = k * 60 + b. \end{cases}$$

Звідси: $b = 5, k = \frac{1}{12}$.

Отже, для першої години зміни:

$$a(t) = \frac{1}{12}t + 5,$$

а потім

$$a(t) = 10.$$

З огляду на тривалість зміни (7 год = 420 хв) і співвідношення (3.2), одержуємо:

$$J(t) = \int_0^t (t/12 + 5)dt = t^2 / 24 + 5t,$$

якщо $0 \leq t \leq 60$, і

$$\begin{aligned} J(t) &= \int_0^{60} (t/12 + 5)dt + \int_{60}^t 10dt = (t^2 / 24 + 5t)|_0^{60} + 10t|_{60}^t = \\ &= 450 + 10t - 600 = 10t - 150, \end{aligned}$$

якщо $60 \leq t \leq 420$.

Кількість деталей на складі через 30 хв. після початку роботи:

$$J(30) = 900/24 + 5 \cdot 30 = 187,5,$$

а наприкінці зміни:

$$J(420) = 10 \cdot 420 - 150 = 4050.$$

3.2.2 Індивідуальні завдання. Детерміновані моделі керування запасами

Реалізуйте модель прикладу 3.1, і знайдіть кількість деталей на складі через ((№ варіанта)·10) хвилин після початку роботи.

3.3 Стохастичні моделі керування запасами

3.3.1 Реалізація стохастичної моделі керування запасами

Розглянемо *стохастичні моделі керування запасами*, в яких попит є *випадковим*. Цей факт істотно позначається на характері відповідних моделей і значно ускладнює їх аналіз.

Припустимо, що попит r за проміжок часу T є випадковим і заданий закон (ряд) розподілу $p(r)$ або щільність імовірності $\varphi(r)$ (зазвичай функції $p(r)$ і $\varphi(r)$ оцінюються на підставі дослідних або статистичних даних). Якщо попит r нижче рівня запасу s , то придбання (збереження, продаж) надлишку продукту вимагає додаткових витрат c_2 на одиницю продукту; навпаки, якщо попит r вище рівня запасу s , те це призводить до штрафу за дефіцит c_3 на одиницю продукції.

В якості функції сумарних витрат, яка є випадковою величиною в стохастичних моделях, розглядають її середнє значення або математичне

очікування.

У розглянутій моделі при дискретному випадковому попиті r , що має закон розподілу $p(r)$, математичне очікування сумарних витрат має вигляд:

$$C(s) = c_2 \sum_{r=0}^s (s-r)p(r) + c_3 \sum_{r=s+1}^{\infty} (r-s)p(r). \quad (3.5)$$

У рівнянні (3.5) перший доданок враховує витрати на придбання (збереження) надлишку $s-r$ одиниць продукту (при $r \leq s$), а другий доданок – штраф за дефіцит на $r-s$ одиниць продукту (при $r > s$).

У випадку безперервного випадкового попиту, що задається щільністю ймовірностей $\varphi(r)$, рівняння для $C(s)$ приймає вигляд:

$$C(s) = c_2 \int_0^s (s-r)\varphi(r)dr + c_3 \int_{s+1}^{\infty} (r-s)\varphi(r)dr. \quad (3.6)$$

Задача керування запасами полягає у відшукуванні такого запасу s , при якому математичне чекання сумарних витрат (3.5) або (3.6) приймає мінімальне значення.

Доведено, що при дискретному випадковому попиті r математичне очікування $C(s)$ (рівняння 3.5) є мінімальним при запасі s_0 , що задовольняє нерівностям:

$$F(s_0) < \rho < F(s_0 + 1), \quad (3.7)$$

а при безперервному випадковому попиті r математичне сподівання $C(s)$ (рівняння 3.5) є мінімальним при запасі s_0 , що задовольняє нерівностям:

$$F(s_0) = \rho, \quad (3.8)$$

де

$$F(s) = p(r < s), \quad (3.9)$$

є функцією розподілу попиту r , $F(s_0)$ і $F(s_0 + 1)$ – її значення; ρ – щільність збитків через незадоволений попит, що знаходиться за формулою

$$\rho = \frac{c_3}{c_2 + c_3}. \quad (3.10)$$

Приклад 3.2. СТО купує стенд розвал-сходження 3D Hunter DSP 600 із запасними світлодіодами до нього. Вартість одного світлодіоду дорівнює 5 у.о. У випадку виходу стенду з ладу через поломку світлодіоду, відсутнього в запасі, простій стенду й термінове замовлення нового світлодіоду до нього обійдеться у 100 у.о.д. Емпіричні дані щодо кількості змінених світлодіодів наведені в таблиці 3.1.

Таблиця 3.1 – Розподіл кількості світлодіодів, що потребують заміни

Кількість замінених блоків r	0	1	2	3	4	5	6
Статистична ймовірність (частка) агрегатів $p(r)$, яким потрібна була заміна r блоків	0,90	0,05	0,02	0,01	0,01	0,01	0,00

Необхідно визначити оптимальну кількість запасних світлодіодів, що варто придбати разом із стендом.

Розв'язання. За умовою $c_2 = 5$, $c_3 = 100$. Обчислимо щільність збитків через недостачу запасних блоків за формулою (3.10):

$$\rho = 100/(5+100) = 0,952.$$

З огляду на (3.9), знайдемо значення функції розподілу попиту (дивись таблицю 3.2).

Таблиця 3.2 – Значення функції розподілу попиту

s	0	1	2	3	4	5	6
$F(s)$	0,00	0,90	0,95	0,97	0,98	0,99	1

Очевидно (дивись таблицю 3.2), що оптимальний запас складе $s_0 = 2$, тому що він задовольняє нерівності (3.7):

$$F(2) < 0,952 < F(3).$$

Відповідь: *рекомендація – придбати два запасних світлодіоди.*

Приклад 3.3. Вирішити приклад 3.2, за умови безперервного випадкового попиту r , що має показниковий закон із функцією розподілу

$$F(r) = 1 - e^{-\lambda r}, \text{ при } \lambda = 0,98.$$

Розв'язання. Оптимальне число запасних блоків s_0 знайдемо з рівняння (3.8): $1 - e^{-\lambda s_0} = \rho$, звідки $e^{-\lambda s_0} = 1 - \rho$ і $s_0 = -\frac{1}{\lambda} \ln(1 - \rho)$.

При $\lambda = 0,98$; $s_0 = -(1/0,98) \ln 0,05 \approx 4$ (світлодіоди).

Відповідь: *рекомендація – придбати чотири запасних блоки.*

3.3.2 Індивідуальні завдання. Стохастичні моделі керування запасами

Задача 1.

Реалізуйте демонстраційну модель (приклад 3.2), але зі своїми даними про:

- а) вартість світлодіоду;
- б) вартість термінового замовлення;
- в) статистичні дані.

Дані взяти з таблиці 3.3.

Задача 2.

Реалізуйте демонстраційну модель (приклад 3.3) але з іншим значенням параметру λ , вартості світлодіоду, вартості термінового замовлення. Дані наведено в таблиці 3.3.

Таблиця 3.3 – Варіанти завдань

№ варіанта	Число замінених світлодіодів, N							Вартість світлодіода	Вартість термінового замовлення	λ
	0	1	2	3	4	5	6			
	Статистична імовірність заміни блоків									
1	0,90	0,01	0,02	0,03	0,02	0,01	0,01	5	10	0,10
2	0,87	0,02	0,01	0,02	0,01	0,04	0,03	4	12	0,12
3	0,85	0,01	0,04	0,04	0,03	0,01	0,02	3	9	0,23
4	0,80	0,05	0,04	0,04	0,02	0,03	0,02	6	15	0,34
5	0,81	0,04	0,05	0,03	0,02	0,01	0,04	7	20	0,45
6	0,83	0,03	0,04	0,02	0,03	0,02	0,03	9	21	0,56
7	0,91	0,02	0,01	0,02	0,01	0,02	0,01	8	23	0,67

Продовження таблиці 3.3

8	0,92	0,01	0,01	0,02	0,02	0,01	0,01	7	18	0,78
9	0,84	0,04	0,03	0,02	0,01	0,02	0,04	6	19	0,89
10	0,86	0,03	0,02	0,02	0,03	0,02	0,02	5	14	0,13
11	0,88	0,05	0,02	0,02	0,01	0,01	0,01	4	13	0,25
12	0,89	0,04	0,02	0,02	0,01	0,01	0,01	9	25	0,46
13	0,82	0,06	0,04	0,03	0,02	0,01	0,02	10	30	0,57
14	0,83	0,04	0,02	0,03	0,01	0,02	0,05	10	12	0,68
15	0,84	0,05	0,04	0,02	0,02	0,01	0,02	11	26	0,79
16	0,86	0,04	0,02	0,04	0,02	0,01	0,01	12	14	0,80
17	0,88	0,02	0,01	0,02	0,03	0,02	0,02	9	27	0,91
18	0,90	0,02	0,02	0,02	0,02	0,01	0,01	8	29	0,97
19	0,92	0,02	0,01	0,02	0,01	0,01	0,01	13	26	0,85
20	0,93	0,02	0,01	0,01	0,01	0,01	0,01	14	18	0,74
21	0,80	0,04	0,05	0,03	0,03	0,04	0,01	11	16	0,63
22	0,81	0,05	0,04	0,03	0,03	0,01	0,03	15	20	0,41
23	0,85	0,06	0,03	0,02	0,01	0,01	0,02	12	17	0,20
24	0,86	0,02	0,03	0,01	0,04	0,02	0,02	9	10	0,17
25	0,89	0,03	0,03	0,02	0,01	0,01	0,01	8	11	0,83

Закінчення таблиці 3.3

26	0,91	0,01	0,01	0,02	0,03	0,01	0,01	7	13	0,94
27	0,82	0,02	0,03	0,05	0,02	0,03	0,03	4	10	0,14
28	0,89	0,01	0,02	0,04	0,02	0,01	0,01	5	7	0,27
29	0,85	0,02	0,03	0,04	0,04	0,01	0,01	2	10	0,35
30	0,83	0,01	0,05	0,04	0,03	0,02	0,02	10	17	0,40

3.4 Автоматизація моделі керування запасами

3.4.1 Приклад автоматизації моделі керування запасами

Автосалон отримує автомобілі марки Kia Ceed SW від постачальника за ціною 20 тис. доларів за одиницю, а продає за ціною 23 тис. доларів за одиницю. У випадку, якщо товар не вдається реалізувати протягом місяця, виконується повернення за ціною 17 тис. доларів за одиницю. Умови контракту із постачальником передбачають, що автосалон купує 5, 10, 15 або 20 автомобілів щомісяця. Відомі статистичні дані щодо кількості реалізованих автомобілів марки Kia Ceed SW за останні 37 місяців (табл. 3.4). Необхідно визначити скільки потрібно закуповувати автомобілів щомісяця, щоб очікуваний прибуток був максимальним.

Таблиця 3.4 – Статистичні дані

Об'єм реалізації	0	5	10	15	20
Кількість подій	0	7	9	15	6

Автоматизацію моделі виконуємо у середовищі MS Excel.

На робочому листі MS Excel доцільно створити керуючі кнопки: «Пуск» – за допомогою якої можна задавати вихідні дані (продаж, купівля, повернення) і «Матриця результатів» – натискання на яку робить автоматичне заповнення матриці фінансових наслідків при всіх можливих варіантах подій купівлі автомобілів і їхньої реалізації. За

даними кнопками закріплюється наступний код в VBA, що записаний в окремих модулях:

Модуль 1 (для кнопки «Пуск»):

```
Public Sub vvod()
    Dim r As Double, v As Double, a As Integer, r1 As String
    Range("продаж")=InputBox("Введіть вартість продажу")
    Range("покупка")=InputBox("Введіть вартість покупки")
    Range("повернення")=InputBox("Введіть вартість повернення")
    r = Range("максимальний_прибуток").Value
    v = Range("оптимальний_об'єм").Value
    a=MsgBox("максимальний прибуток:" & r & " оптимальний
об'єм" & v)
End Sub
```

Модуль 2 (для кнопки «Матриця результатів»)

```
Option Base 1
Function априбуток (покуп)
    Dim N As Integer, цпрод As Double, цпок As Double, цсдач As Double
    Dim i As Integer, j As Integer, res() As Double
    N = покуп.Rows.Count
    цпрод = Range("продаж").Value
    цпок = Range("покупка").Value
    цсдач = Range("повернення").Value
    ReDim res(N, N)
    For i = 1 To N
        For j = 1 To N
            If i <= j Then res(i, j) = покуп(i) * (цпрод - цпок)
            If i > j Then res(i, j) = покуп(j)*(цпрод - цпок) - (покуп(i) -
покуп(j))*(цпок - цсдач)
        Next j
    Next i
    априбуток = res
End Function
```

Після запуску даних кодів заповнимо стовпець «**Прибуток**», знайдемо в ньому максимальне значення й відповідне значення партії, що реалізується, зі стовпця «**Покупка**» за формулами, що наведені на рисунку 3.1.

1	B	C	D	E	F	G	H	I	J	K	L	M
2	продаж	покуп	повер									
3	23	20	17									
5	об'єм реалізації	0	5	10	15	20						
6	число подій	0	7	9	15	6						
7	ймовірність подій	0.00	0.19	0.24	0.41	0.16						
8				Продаж								
9				0	5	10	15	20				
10		Покупка	0	0	0	0	0	0			прибуток	
11			5	-15	15	15	15	15			0	
12			10	-30	0	30	30	30			15	
13			15	-45	-15	15	45	45			24.32	
14			20	-60	-30	0	30	60			26.35	
16			максимальний прибуток								16.22	
17			оптимальний об'єм									
18												
19												

Формули:

 =C6/СУММ(\$C\$6:\$G\$6)

 (=МУМНОЖ(E10:K14;ТРАНСП(C7:G7)))

 =МАКС(K10:K14)

 =(ПОИСКПОЗ(МАКС(K10:K14);K10:K14;0)-1)*5

Рисунок 3.1 – Реалізація моделі в Excel

3.4.2 Індивідуальні завдання. Автоматизовані моделі керування запасами

Реалізуйте модель пункту 3.4.1, використовуючи дані таблиці 3.5.
Таблиця 3.5 – Варіанти завдань

Вар.	Продаж	Покупка	Повернення	Число подій				
				0 шт.	5 шт.	10 шт.	15 шт.	20 шт.
1	12	10	8	0	6	8	10	6
2	14	12	10	0	6	10	9	5
3	16	14	12	0	10	11	9	7
4	18	16	14	0	9	9	8	7
5	21	18	16	0	7	11	11	6
6	23	20	18	0	5	9	12	5
7	25	22	20	0	5	8	7	3
8	27	24	22	0	4	10	9	3

Продовження таблиці 3.5

9	29	26	24	0	6	14	12	4
10	31	28	26	0	3	9	15	4
11	17	15	12	0	7	12	14	5
12	20	18	15	0	5	12	11	3
13	24	21	18	0	4	15	12	6
14	27	24	21	0	3	10	13	4
15	30	27	24	0	8	11	14	3
16	33	30	27	0	6	11	14	7
17	36	33	30	0	4	10	12	6
18	39	36	33	0	7	12	15	4
19	42	39	36	0	5	9	12	3
20	45	42	39	0	8	15	16	7
21	20	16	13	0	7	15	13	5
22	22	18	15	0	4	12	14	5
23	24	20	17	0	3	11	14	6
24	26	22	19	0	5	16	15	7
25	28	24	21	0	6	10	12	4
26	30	26	22	0	5	14	9	3
27	32	28	24	0	4	12	15	6
28	34	30	26	0	6	14	17	7
29	36	32	28	0	7	16	14	6
30	38	34	30	0	5	15	18	4

РОЗДІЛ 4 ЕЛЕМЕНТИ ТЕОРІЇ ІГОР

4.1 Основні поняття теорії ігор

4.1.1 Загальні поняття теорії ігор

На практиці часто виникають так звані конфліктні ситуації, в яких дві (або більше) сторони переслідують різні цілі, а результати будь-якої дії кожної зі сторін залежать від дій партнера. Такі ситуації нагадують гру в шахи, шашки, доміно тощо, оскільки результат кожного ходу гравця залежить від відповідного ходу супротивника, ціль гри – виграш одного з учасників. В економіці конфліктні ситуації зустрічаються дуже часто й мають різноманітний характер. До них належать, наприклад, взаємини між постачальником і споживачем, покупцем і продавцем, банком і клієнтом. В усіх цих прикладах конфліктна ситуація породжується розходженням інтересів сторін і прагненням кожної з них приймати оптимальні рішення, що реалізують поставлені цілі найбільшою мірою. При цьому кожному доводиться враховувати не тільки власні цілі, але й цілі іншої сторони, а також орієнтуватися на невідомі заздалегідь рішення «супротивника».

Для раціонального розв'язання задач із конфліктними ситуаціями необхідні науково обґрунтовані методи. Такі методи розроблені математичною теорією конфліктних ситуацій, що отримала назву *теорії ігор*.

Ознайомимося з основними поняттями теорії ігор. Математична модель конфліктної ситуації називається *грою*, сторони, що беруть участь у конфлікті, – *гравцями*, а результат конфлікту – *виграшем*.

Для кожної формалізованої гри вводяться *правила*, тобто система умов, що визначає:

- 1) варіанти дій гравців;
- 2) обсяг інформації кожного гравця про дії партнерів;
- 3) виграш, до якого приводить кожна сукупність дій. Як правило, виграш (або програш) може бути заданий кількісно. Наприклад, можна оцінити програш нулем, виграш – одиницею, а нічию – $1/2$.

Гра називається *парною*, якщо в ній беруть участь два гравці, і *множинною*, якщо кількість гравців більше двох. Ми будемо розглядати тільки парні ігри. У них беруть участь два гравці A і B , інтереси яких протилежні, а під грою будемо розуміти ряд дій з боку A і B .

Гра називається *грою з нульовою сумою* або *антагоністичною*, якщо виграш одного з гравців дорівнює програшеві іншого, тобто для повного задання гри досить вказати величину одного з них. Якщо позначити a — виграш одного з гравців, b — виграш іншого, то для гри з нульовою сумою $b = -a$, тому досить розглядати, наприклад a .

Вибір і здійснення однієї з передбачених правилами гри дій називається *ходом* гравця. Ходи можуть бути особистими й випадковими.

Особистий хід — це свідомий вибір гравцем однієї з можливих дій (наприклад, хід у шаховій грі).

Випадковий хід — це випадково обрана дія (наприклад, вибір карти з перетасованої колоди). Надалі ми будемо розглядати тільки особисті ходи гравців.

Стратегією гравця називається сукупність правил, що визначають вибір його дії під час кожного особистого ходу в залежності від ситуації, що сформувалася. Безумовно, в процесі гри гравець робить вибір у залежності від конкретної ситуації. Проте можливо використовувати заздалегіть продумані рішення (у відповідь на будь-яку сформовану ситуацію). Це означає, що гравець вибрав визначену стратегію, що може бути задана у вигляді переліку правил або програми, наприклад, можна здійснити гру за допомогою комп'ютера.

Гра називається *кінцевою*, якщо в кожного гравця є кінцеве число стратегій, і *нескінченною* — у протилежному випадку.

Для того, щоб *вирішити* гру або знайти рішення гри, необхідно для кожного гравця вибрати стратегію, що задовольняє умові *оптимальності*, тобто один із гравців повинен одержати *максимальний виграш* коли другий додержується своєї стратегії. У той же час другий гравець повинен мати *мінімальний програш*, якщо перший додержується своєї стратегії.

Такі стратегії називаються *оптимальними*. Оптимальні стратегії повинні також задовольняти умові *стійкості*, тобто кожному з гравців має бути не вигідно відмовитися від своєї стратегії в цій грі.

Якщо гра повторюється досить багато разів, то гравців може цікавити не виграш і програш у кожній конкретній партії, а *середній виграш (програш)* у всіх партіях.

Метою теорії ігор є визначення оптимальної стратегії для кожного гравця. При виборі оптимальної стратегії природно припускати, що обидва гравці поведуться розумно, тобто орієнтуючись на свої інтереси. Найважливіше обмеження теорії ігор — виграш є єдиним показником ефективності, у той час як у більшості реальних економічних задач є більш одного показника ефективності. Крім того, в економіці, як правило, виникають задачі, в яких інтереси партнерів не обов'язково антагоністичні. Апарат теорії ігор для розв'язання задач із багатьма

учасниками, що мають несуперечливі інтереси, виходить за межі нашого курсу.

4.1.2 Платіжна матриця. Нижня й верхня ціна гри

Розглянемо парну кінцеву гру. Нехай гравець A має m особистих стратегій, що позначимо A_1, A_2, \dots, A_m . Нехай у гравця B є n особистих стратегій, позначимо їх B_1, B_2, \dots, B_n . Кажуть, що гра має розмірність $m \times n$. У результаті вибору гравцями будь-якої пари стратегій:

$$A_i \text{ і } B_j, \quad (i = 1, 2, \dots, m; j = 1, 2, \dots, n)$$

однозначно визначається результат гри, тобто виграш a_{ij} гравця A (додатний чи від'ємний) і програш $(-a_{ij})$ гравця B . Припустимо, що значення a_{ij} відомі для будь-якої пари стратегій A_i і B_j . Матриця P , елементами якої є виграші, що відповідають стратегіям A_i і B_j , називається *платіжною матрицею* або *матрицею гри* (табл. 4.1):

Таблиця 4.1 – Платіжна матриця

	B_1	B_2	...	B_n
A_1	a_{11}	a_{12}	...	a_{1n}
A_2	a_{21}	a_{22}	...	a_{2n}
...
A_m	a_{m1}	a_{m2}	...	a_{mn}

Рядки цієї таблиці відповідають стратегіям гравця A , а стовпці – стратегіям гравця B .

Приклад складання платіжної матриці.

Необхідно скласти платіжну матрицю для гри «пошук»: гравець A може сховатися в одному з двох пунктів (I і II); гравець B шукає гравця A , і якщо знайде, то одержує штраф 1 грош. од. від A , в іншому випадку платить гравцеві A 1 грош. од. Необхідно побудувати платіжну матрицю гри.

Розв'язання.

Якщо гравець A знаходиться у місці I, і там його виявляє гравець B , тобто здійснюється пара стратегій (A_1, B_1) , то гравець A платить штраф, тобто $a_{11} = -1$. Аналогічно одержуємо $a_{22} = -1$ для пари стратегій

(A_2, B_2) . Очевидно, що стратегії (A_1, B_2) і (A_2, B_1) дають гравцеві A вигравш 1, тому $a_{12} = a_{21} = 1$.

Таким чином, для гри «пошук» розмірності 2×2 одержуємо платіжну матрицю:

$$P = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}.$$

4.1.3 Принцип мінімакса

Розглянемо гру m на n з матрицею $P = (a_{ij})$, $(i = 1, 2, \dots, m; j = 1, 2, \dots, n)$ і визначимо найкращу серед стратегій A_1, A_2, \dots, A_m . Вибираючи стратегію A_i , гравець A повинен розраховувати, що гравець B відповість на неї тією зі стратегій B_j для якої вигравш для гравця A мінімальний (гравець B прагне «нашкодити» гравцеві A). Позначимо через λ_i , найменший вигравш гравця A при виборі ним стратегії A_i , для всіх можливих стратегій гравця B (найменше число в i -му рядку платіжної матриці), тобто

$$\lambda_i = \min_{j=1, \dots, n} a_{ij}.$$

Серед усіх чисел λ_i ($i = 1, 2, \dots, m$) оберемо найбільше:

$$\lambda = \max_{i=1, \dots, m} \lambda_i.$$

Назвемо λ нижньою ціною гри або *максимальним вигравшем* (*максиміном*). Це гарантований вигравш гравця A за будь-якої стратегії гравця B . Отже *максимін*:

$$\lambda = \max_{i=1, m} \min_{j=1, n} a_{ij}$$

є *гарантованим вигравшем* для A .

Стратегія, що відповідає максиміну, називається *максимінною стратегією*. Гравець B зацікавлений у тому, щоб зменшити вигравш гравця A . Обираючи стратегію B_j , він враховує максимально можливий при цьому вигравш для A . Нехай $\beta_j = \max_{i=1, m} a_{ij}$.

Серед усіх чисел β_j , оберемо найменше

$$\beta = \min_{j=1, n} \beta_j$$

і назвемо β верхньою ціною гри або *мінімаксом*. Це гарантований програш гравця B . Отже *мінімакс*:

$$\beta = \min_{j=1,n} \max_{i=1,m} a_{ij}$$

є гарантованим програшем для B . Стратегія, що відповідає мінімаксу, називається *мінімаксною стратегією*. Принцип, що диктує гравцям вибір найбільш «обережних» мінімаксної та максимінної стратегій, називається *принципом мінімакса*. Цей принцип впливає з розумного припущення, що кожен гравець прагне досягти мети, протилежної меті супротивника.

Приклад знаходження нижньої та верхньої ціни гри.

Необхідно визначити нижню та верхню ціни гри й відповідні стратегії для гри «пошук», що описана в попередньому пункті.

У грі «пошук», що описана в попередньому пункті, верхня й нижня ціни гри різні: $\lambda \neq \beta$.

Якщо верхня й нижня ціни гри збігаються, то загальне значення верхньої та нижньої ціни гри $\lambda = \beta = v$ називається *чистою ціною гри* або *ціною гри*. Мінімаксні стратегії, що відповідають ціні гри, є *оптимальними стратегіями*, а їхня сукупність – *оптимальним рішенням* або *рішенням* гри. У цьому випадку гравець A одержує максимальний гарантований (незалежний від поведження гравця B) виграш V , а гравець B намагається отримати мінімальний гарантований (незалежний від поведження гравця A) програш V .

Кажуть, що рішення гри має *стійкість*: якщо один із гравців додержується своєї оптимальної стратегії, при цьому для іншого не може бути вигідним відхилитися від своєї оптимальної стратегії.

Пара чистих стратегій A і B дає оптимальне рішення гри тоді й тільки тоді, коли відповідний їй елемент a_{ij} є одночасно найбільшим у своєму стовпці й найменшим у своєму рядку. Така ситуація, якщо вона існує, називається *сідловою точкою* (за аналогією з поверхнею сідла, що викривляється нагору в одному напрямку й вниз – в іншому).

Приклад знаходження нижньої та верхньої ціни гри.

Необхідно визначити нижню та верхню ціну гри, що задана платіжною матрицею:

$$P = \begin{pmatrix} 0,5 & 0,6 & 0,8 \\ 0,9 & 0,7 & 0,8 \\ 0,7 & 0,6 & 0,6 \end{pmatrix}.$$

Розв'язання.

Таблиця 4.2 — Розв'язання задачі

	B_1	B_2	B_3	λ_i
A_1	0,5	0,6	0,8	0,5
A_2	0,9	0,7	0,8	0,7
A_3	0,7	0,6	0,6	0,6
β_j	0,9	0,7	0,8	$\lambda = \beta = 0,7$

Сідлова точка

4.2 Ігрові моделі 2x2. Розв'язання ігор у змішаних стратегіях. Графічне розв'язання гри із заданою платіжною матрицею

4.2.1 Загальні відомості

Якщо гра не має сідлової точки, то застосування чистих стратегій не дає оптимального рішення гри. Так, у грі «пошук» (див. пункт 4.1.2) $\lambda \neq \beta$, тому сідлова точка відсутня. У такому випадку можна одержати оптимальне рішення, чергуючи випадковим чином чисті стратегії.

Змішаною стратегією S_A гравця A називається застосування чистих стратегій A_1, A_2, \dots, A_m з ймовірностями p_1, p_2, \dots, p_m , причому сума ймовірностей дорівнює 1:

$$\sum_{i=1}^m p_i = 1.$$

Змішані стратегії гравця A записуються у вигляді матриці:

$$S_A = \begin{pmatrix} A_1 & A_2 & \dots & A_i & \dots & A_m \\ p_1 & p_2 & \dots & p_i & \dots & p_m \end{pmatrix}$$

або у вигляді рядка: $S_A = (p_1, p_2, \dots, p_i, \dots, p_m)$.

Аналогічно змішані стратегії гравця B позначаються:

$$S_B = \begin{pmatrix} B_1 & B_2 & \dots & B_j & \dots & B_n \\ p_1 & p_2 & \dots & p_j & \dots & p_n \end{pmatrix}$$

або у вигляді рядка $S_B = (q_1, q_2, \dots, q_i, \dots, q_m)$,

де сума ймовірностей використання стратегій дорівнює 1:

$$\sum_{j=1}^n q_j = 1.$$

Чисті стратегії можна вважати частковим випадком змішаних і задавати рядом, у якому $p = 1$ відповідає чистій стратегії. На підставі принципу мінімаксу визначається *оптимальне рішення* гри: це пари оптимальних стратегій S_A^*, S_B^* , у загальному випадку змішаних, що мають таку властивість: якщо один із гравців додержується своєї оптимальної стратегії, то іншому не може бути вигідно відступати від своєї. Виграш, що відповідає оптимальному рішенню, називається *ціною гри* v .

Ціна гри задовольняє нерівності:

$$\lambda \leq v \leq \beta \text{ де } \lambda \text{ і } \beta - \text{нижня і верхня ціни гри.}$$

Теорема Неймана (Джон фон Нейман (1903 – 1957) – американський математик). Кожна кінцева гра має принаймні одне оптимальне рішення, можливо, серед змішаних стратегій.

$$S_A^* = (p_1^*, p_2^*, \dots, p_n^*) \text{ і } S_B^* = (q_1^*, q_2^*, \dots, q_n^*) - \text{пари оптимальних стратегій.}$$

Якщо чиста стратегія входить в оптимальну змішану стратегію з відмінною від нуля ймовірністю, то вона називається *активною*.

Теорема про активні стратегії: якщо один із гравців додержується своєї оптимальної змішаної стратегії, то виграш залишається незмінним і рівним ціні гри v , якщо другий гравець не виходить за межі своїх активних стратегій.

Ця теорема має велике практичне значення – вона дає конкретні моделі перебування оптимальних стратегій за відсутності сідлової точки.

Розглянемо гру розміру 2×2 , що є найпростішим випадком кінцевої гри.

1. Якщо така гра має *сідлову точку*, то оптимальне рішення – це пари чистих стратегій, що відповідають цій точці.

2. Гра, в якій *відсутня сідлова точка*, за теоремою Неймана має оптимальне рішення, яке визначається парою змішаних стратегій $S_A^* = (p_1^*, p_2^*)$ і $S_B^* = (q_1^*, q_2^*)$.

Для того щоб їх знайти, скористаємося теоремою про активні стратегії: якщо гравець A додержується своєї оптимальної стратегії S_A^* , то його середній виграш дорівнює ціні гри v , якою би активною стратегією не користувався гравець B .

Нехай гра задана платіжною матрицею: $P = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$. Середній виграш гравця A , якщо він використовує оптимальну змішану стратегію:

$$S_A^* = \begin{pmatrix} A_1 & A_2 \\ \dots & \dots \\ P_1 & P_2 \end{pmatrix},$$

а гравець B – чисту стратегію B_1 (це відповідає 1-му стовпцеві платіжної матриці P), дорівнює ціні гри v :

$$a_{11}p_1^* + a_{21}p_2^* = v. \quad (4.1)$$

Той же середній вигреш одержує гравець A , якщо 2-й гравець застосовує стратегію B_2 , тобто

$$a_{12}p_1^* + a_{22}p_2^* = v.$$

З огляду на те, що $p_1^* + p_2^* = 1$, одержуємо систему рівнянь для визначення оптимальної стратегії s_A^* , і ціни гри v :

$$\begin{cases} a_{11}p_1^* + a_{21}p_2^* = v, \\ a_{12}p_1^* + a_{22}p_2^* = v, \\ p_1^* + p_2^* = 1. \end{cases}$$

Розв'язавши цю систему, одержимо оптимальну стратегію:

$$p_1^* = \frac{a_{22} - a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}};$$

$$p_2^* = \frac{a_{11} - a_{12}}{a_{11} + a_{22} - a_{12} - a_{21}};$$

і ціну гри

$$v = \frac{a_{22}a_{11} - a_{12}a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}.$$

Застосовуючи теорему про активні стратегії при відшуканні S_B^* (оптимальної стратегії гравця B), одержуємо, що за будь-якої чистої стратегії гравця A (A_1 або A_2) середній програш гравця B дорівнює ціні гри v , тобто

$$\begin{cases} a_{11}q_1^* + a_{12}q_2^* = v, \\ a_{21}q_1^* + a_{22}q_2^* = v, \\ q_1^* + q_2^* = 1. \end{cases}$$

Тоді оптимальна стратегія $S_B^* = (q_1^*, q_2^*)$ визначається формулами:

$$q_1^* = \frac{a_{22} - a_{12}}{a_{11} + a_{22} - a_{12} - a_{21}}; \quad q_2^* = \frac{a_{11} - a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}.$$

Приклад знаходження оптимальних стратегій гри.

Необхідно знайти оптимальні стратегії для ситуації з автотранспортним підприємством і постачальником, що описана

вище (див. пункт 4.1.2).

Гра «пошук» задана платіжною матрицею без сідлової точки:

$$P = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}, \lambda = -1, \beta = 1.$$

Тому шукаємо рішення в змішаних стратегіях:

$$\begin{cases} (-1)p_1^* + 1 \cdot p_2^* = v, \\ 1 \cdot p_1^* - 1 \cdot p_2^* = v, \\ p_1^* + p_2^* = 1, \end{cases} \quad \begin{cases} (-1)q_1^* + 1 \cdot q_2^* = v, \\ 1 \cdot q_1^* - 1 \cdot q_2^* = v, \\ q_1^* + q_2^* = 1. \end{cases}$$

Розв'язуючи ці системи, одержуємо:

$$p_1^* = p_2^* = q_1^* = q_2^* = \frac{1}{2}, v = 0.$$

Це означає, що оптимальна стратегія кожного гравця полягає в тому, щоб чергувати свої чисті стратегії випадковим чином, обираючи кожне з місць з імовірністю $\frac{1}{2}$, при цьому середній виграш дорівнює 0.

Графічне розв'язання гри з заданою платіжною матрицею

Геометрична інтерпретація гри 2x2 представлена на рисунках 4.1 і 4.2.

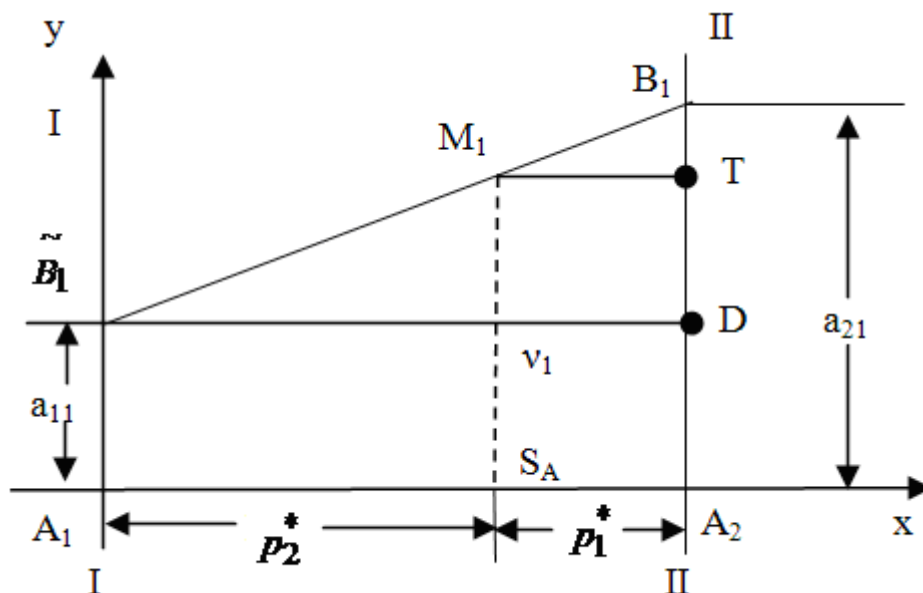


Рисунок 4.1 – Геометрична інтерпретація (1)

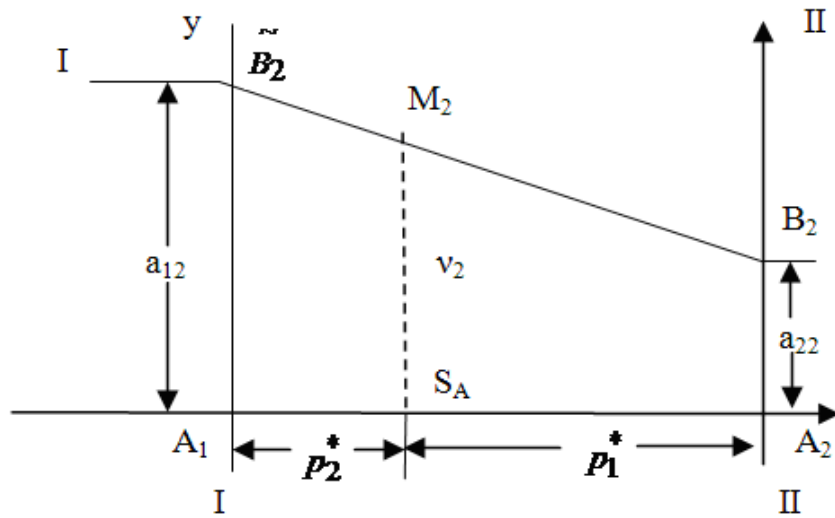


Рисунок 4.2 – Геометрична інтерпретація (2)

Розглянемо подібні трикутники рисунка 5.1:

$$\Delta M_1 B_1 T \propto \Delta \tilde{B}_1 B_1 D \Rightarrow \frac{B_1 T}{M_1 T} = \frac{B_1 D}{\tilde{B}_1 D} = \frac{a_{21} - v}{p_1^*} = \frac{a_{21} - a_{11}}{1} \Rightarrow$$

$$\Rightarrow a_{21} - v = p_1^* (a_{21} - a_{11}) \Rightarrow v = a_{21} - p_1^* a_{21} + p_1^* a_{11} \Rightarrow v = a_{21} (1 - p_1^*) + p_1^* a_{11} \Rightarrow v = a_{21} p_2^* + a_{11} p_1^*.$$

Об'єднуючи рисунки 4.1 і 4.2, отримаємо графічний розв'язок ігрової моделі (рис. 4.3):

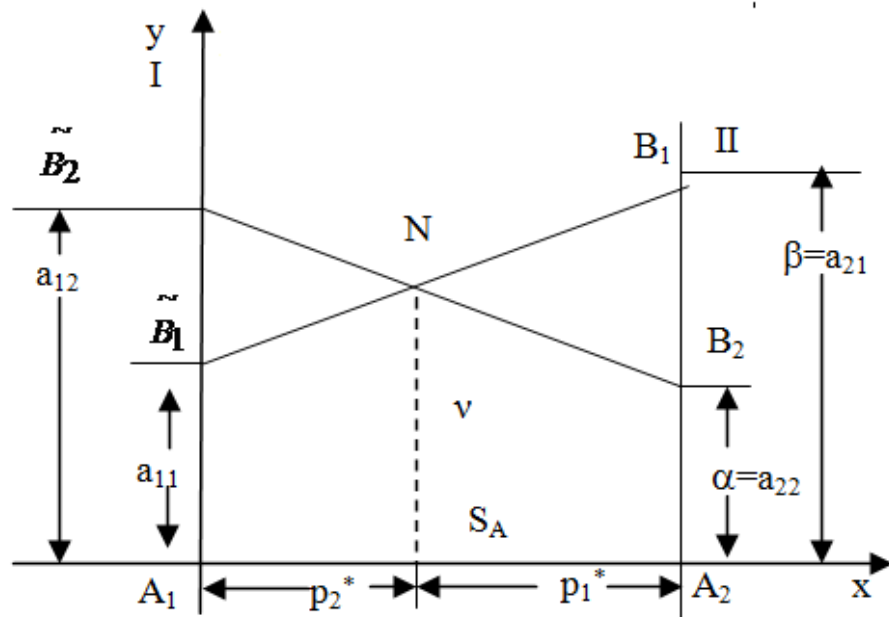


Рисунок 4.3 – Графічний розв'язок ігрової моделі

Наведемо конкретний приклад графічного розв'язання гри.
Необхідно графічно розв'язати гру для даної платіжної матриці:

$$P = \begin{pmatrix} 1,5 & 3 \\ 2 & 1 \end{pmatrix}.$$

Реалізацію моделі наведено на рисунку 4.4.

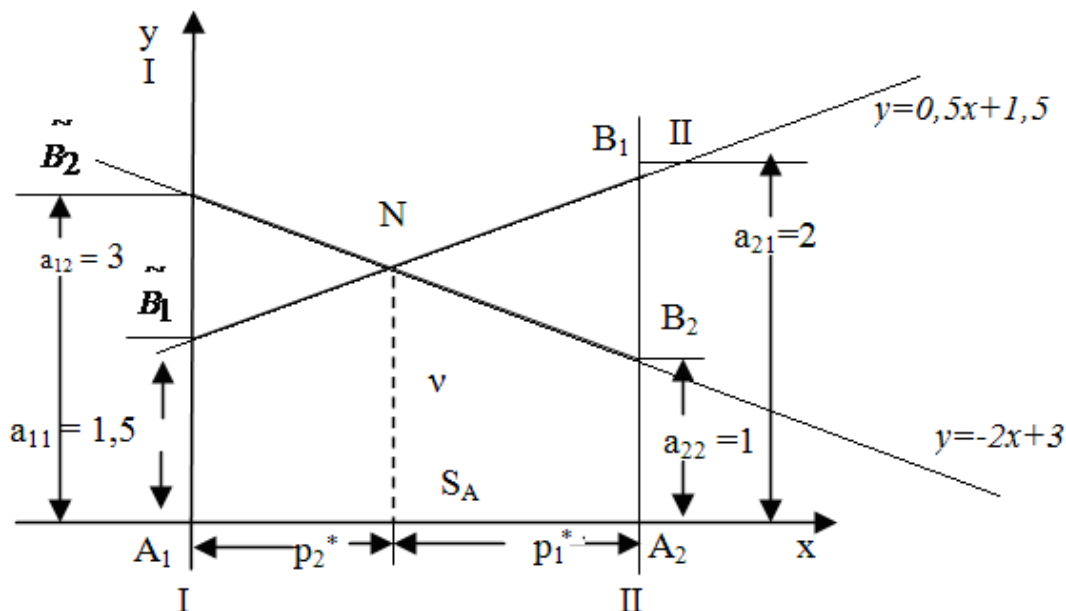


Рисунок 4.4 – Реалізація моделі із заданою платіжною матрицею

$$\tilde{B}_1 B_1 \cap \tilde{B}_2 B_2 = N;$$

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}; \quad \frac{x - 0}{1 - 0} = \frac{y - 1,5}{2 - 1,5} \Rightarrow$$

$$\Rightarrow \frac{x}{1} = \frac{y - 1,5}{0,5} \Rightarrow y = 0,5x + 1,5.$$

$$\frac{x - 0}{1 - 0} = \frac{y - 3}{1 - 3} \Rightarrow y - 3 = -2x \Rightarrow y = -2x + 3.$$

$$\begin{cases} y = 0,5x + 1,5, \\ y = -2x + 3. \end{cases}$$

$$0,5x + 1,5 = -2x + 3;$$

$$2,5x = 1,5;$$

$$x = \frac{1,5}{2,5} = 0,6 = p_2, \quad p_1 = 1 - 0,6 = 0,4.$$

Відповідь: таким чином, $p_1^* = 0,4$; $p_2^* = 1 - 0,4 = 0,6$. Оптимальна стратегія $S_A^* = (0,4; 0,6)$, ціна гри $v = 1,8$.

4.2.2 Індивідуальні завдання. Ігрові моделі 2x2

Знайдіть оптимальні стратегії гри із заданою платіжною матрицею:

а) аналітично;

б) графічно, порівнявши результати з пунктом а).

Перевірте, чи має гра сідлову точку.

Таблиця 4.3 – Платіжні матриці 2x2

Варіант 1	Варіант 11	Варіант 21												
<table border="1"><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>1</td></tr></table>	2	3	4	1	<table border="1"><tr><td>2</td><td>5</td></tr><tr><td>7</td><td>1</td></tr></table>	2	5	7	1	<table border="1"><tr><td>3</td><td>0</td></tr><tr><td>1</td><td>3</td></tr></table>	3	0	1	3
2	3													
4	1													
2	5													
7	1													
3	0													
1	3													
Варіант 2	Варіант 12	Варіант 22												
<table border="1"><tr><td>1</td><td>3</td></tr><tr><td>4</td><td>0</td></tr></table>	1	3	4	0	<table border="1"><tr><td>3</td><td>4</td></tr><tr><td>7</td><td>2</td></tr></table>	3	4	7	2	<table border="1"><tr><td>5</td><td>0</td></tr><tr><td>3</td><td>4</td></tr></table>	5	0	3	4
1	3													
4	0													
3	4													
7	2													
5	0													
3	4													
Варіант 3	Варіант 13	Варіант 23												
<table border="1"><tr><td>1</td><td>4</td></tr><tr><td>3</td><td>1</td></tr></table>	1	4	3	1	<table border="1"><tr><td>1</td><td>4</td></tr><tr><td>7</td><td>0</td></tr></table>	1	4	7	0	<table border="1"><tr><td>1</td><td>7</td></tr><tr><td>3</td><td>2</td></tr></table>	1	7	3	2
1	4													
3	1													
1	4													
7	0													
1	7													
3	2													
Варіант 4	Варіант 14	Варіант 24												
<table border="1"><tr><td>1</td><td>5</td></tr><tr><td>2</td><td>1</td></tr></table>	1	5	2	1	<table border="1"><tr><td>5</td><td>9</td></tr><tr><td>7</td><td>4</td></tr></table>	5	9	7	4	<table border="1"><tr><td>2</td><td>1</td></tr><tr><td>0</td><td>4</td></tr></table>	2	1	0	4
1	5													
2	1													
5	9													
7	4													
2	1													
0	4													
Варіант 5	Варіант 15	Варіант 25												
<table border="1"><tr><td>0</td><td>2</td></tr><tr><td>4</td><td>1</td></tr></table>	0	2	4	1	<table border="1"><tr><td>5</td><td>2</td></tr><tr><td>0</td><td>4</td></tr></table>	5	2	0	4	<table border="1"><tr><td>2</td><td>4</td></tr><tr><td>5</td><td>3</td></tr></table>	2	4	5	3
0	2													
4	1													
5	2													
0	4													
2	4													
5	3													
Варіант 6	Варіант 16	Варіант 26												
<table border="1"><tr><td>2</td><td>0</td></tr><tr><td>1</td><td>9</td></tr></table>	2	0	1	9	<table border="1"><tr><td>4</td><td>2</td></tr><tr><td>0</td><td>3</td></tr></table>	4	2	0	3	<table border="1"><tr><td>1</td><td>5</td></tr><tr><td>6</td><td>3</td></tr></table>	1	5	6	3
2	0													
1	9													
4	2													
0	3													
1	5													
6	3													
Варіант 7	Варіант 17	Варіант 27												
<table border="1"><tr><td>0</td><td>3</td></tr><tr><td>4</td><td>0</td></tr></table>	0	3	4	0	<table border="1"><tr><td>6</td><td>2</td></tr><tr><td>1</td><td>3</td></tr></table>	6	2	1	3	<table border="1"><tr><td>2</td><td>4</td></tr><tr><td>6</td><td>1</td></tr></table>	2	4	6	1
0	3													
4	0													
6	2													
1	3													
2	4													
6	1													
Варіант 8	Варіант 18	Варіант 28												
<table border="1"><tr><td>0</td><td>3</td></tr><tr><td>3</td><td>2</td></tr></table>	0	3	3	2	<table border="1"><tr><td>7</td><td>0</td></tr><tr><td>1</td><td>3</td></tr></table>	7	0	1	3	<table border="1"><tr><td>3</td><td>5</td></tr><tr><td>7</td><td>1</td></tr></table>	3	5	7	1
0	3													
3	2													
7	0													
1	3													
3	5													
7	1													

Варіант 9	
1	6
3	1

Варіант 19	
8	2
1	4

Варіант 29	
4	5
7	2

Варіант 10	
0	3
5	1

Варіант 20	
4	5
8	2

Варіант 30	
3	5
6	1

4.3 Ігрові моделі $2 \times n$

4.3.1 Загальні відомості

Розв'язок матричних ігор у змішаних стратегіях може бути знайдене або графічно, або методами лінійного програмування. Графічний метод застосуємо для ігор, у яких хоча б один гравець має дві чисті стратегії. Цей метод графічно пояснює поняття сідлової точки. Методами лінійного програмування може бути вирішена будь-яка гра двох осіб з нульовою сумою.

Графічне розв'язання ігор

Розглянемо гру $2 \times n$, в якій гравець A має дві стратегії.

		y_1	y_2	\dots	y_n
		B_1	B_2	\dots	B_n
x_1 :	A_1	a_{11}	a_{12}	\dots	a_{1n}
$1 - x_1$:	A_2	b_{11}	b_{12}	\dots	b_{1n}

Рисунок 4.5 – Платіжна матриця (загальний вигляд)

Гра передбачає, що гравець A змішує стратегії A_1 і з імовірностями x_1 і $1 - x_1$, $0 \leq x_1 \leq 1$. Гравець B змішує стратегії B_1, B_2, \dots, B_n з імовірностями y_1, y_2, \dots, y_n , де $y_i \geq 0, i = 1, 2, \dots, n$ і $y_1 + y_2 + \dots + y_n = 1$. У цьому випадку очікуваний виграш гравця A , що відповідає j -й чистій стратегії гравця B обчислюється за формулою:

$$(a_{1j} - a_{2j})x_1 + a_{2j}, j = 1, 2, \dots, n.$$

Отже, гравець A шукає таку величину x_1 , що максимізує мінімум очікуваних виграшів.

$$\max_{x_1} \min_j \{(a_{1j} - a_{2j})x_1 + a_{2j}\}.$$

Розглянемо таку ігрову ситуацію: на літній період директор СТО (гравець A) хоче взяти в оренду частину складських приміщень, що знаходяться неподалік СТО, і встановити там полиці для сезонного зберігання зимових шин (стратегія A_1) або обладнання з обслуговування автомобільних кондиціонерів (стратегія A_2). Власник складських приміщень, проаналізувавши свої витрати на звільнення приміщень і прибуток від оренди, зробив висновок, що може виділити 100, 120, 150 чи 200 кв. м площі своїх приміщень (стратегії B_1, B_2, B_3, B_4). Прибуток СТО, що відповідає рішенням щодо використання приміщень і площі, що дасть в оренду власник приміщень, що наведений у платіжній матриці:

	B_1	B_2	B_3	B_4
A_1, x_1	2	2	3	-1
$A_2,$	4	3	2	6
$1 - x_1$				

Рисунок 4.6 – Платіжна матриця

Необхідно визначити оптимальну стратегію СТО й власника складських приміщень.

Гра не має розв'язку в чистих стратегіях, тому стратегії мають бути змішаними. Очікувані виграші гравця A , що відповідають чистим стратегіям гравця B , що наведені в таблиці 4.4.

Таблиця 4.4 – Очікувані виграші гравця A

Чисті стратегії гравця B	Очікувані виграші гравця A
1	$-2x_1 + 4$
2	$-x_1 + 3$
3	$x_1 + 2$
4	$-7x_1 + 6$

На рисунку 4.7 зображені чотири прямі лінії, що відповідають чистим стратегіям гравця B . Щоб визначити найкращий результат з найгірших, побудована нижня огинаюча чотирьох зазначених прямих (зображена на рисунку товстими лінійними сегментами), що представляє

мінімальний (найгірший) виграш для гравця A незалежно від того, що робить гравець B . Максимум (найкраще) нижньої огинаючої, відповідає максимальному розв'язку в точці $x_1 = 0,5$. Ця точка визначається перетинанням прямих 3 і 4. Отже, оптимальним для гравця A є змішування стратегій A_1 і A_2 з ймовірностями 0,5 і 0,5 відповідно. Відповідну ціну гри v знайдемо шляхом підстановки $x_1 = 0,5$ у рівняння або прямої 3, або 4:

$$v = \begin{cases} \frac{1}{2} + 2 = \frac{5}{2}, & \text{з рівняння прямої 3,} \\ -7\left(\frac{1}{2}\right) + 6 = \frac{5}{2}, & \text{з рівняння прямої 4.} \end{cases}$$

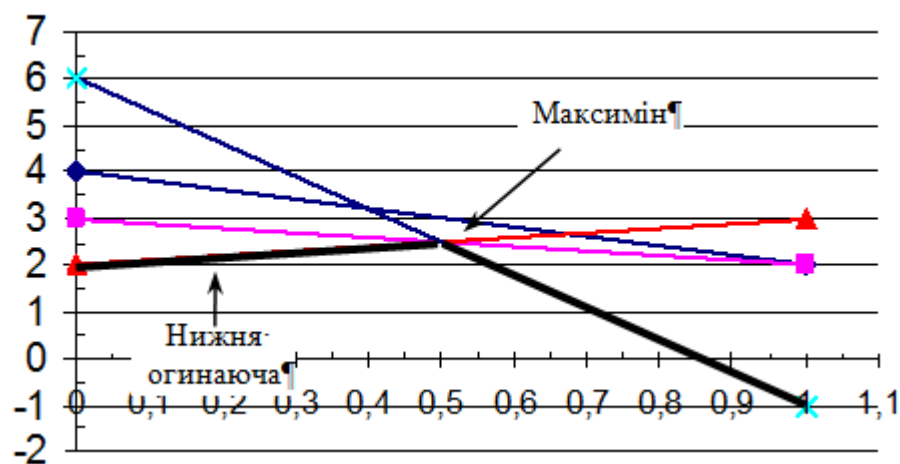


Рисунок 4.7 – Побудова нижньої огинаючої

Оптимальна змішана стратегія гравця B визначається двома стратегіями, що визначають нижню огинаючу графіка. Це значить, що гравець B може змішувати стратегії B_3 і B_4 , у цьому випадку $y_1 = y_2 = 0$ і $y_4 = 1 - y_3$. Отже, очікувані платежі гравця B , що відповідають чистим стратегіям гравця A , мають такий вигляд (таблиця 4.5).

Таблиця 4.5 – Очікувані платежі гравця B

Чисті стратегії гравця A	Очікувані платежі гравця B
1	$4y_3 - 1$
2	$-4y_3 + 6$

Найкраще рішення з найгірших для гравця B є точкою мінімуму верхньої огинаючої. Для її знаходження необхідно розв'язати рівняння:

$$4y_3 - 1 = -4y_3 + 6.$$

Його розв'язком буде $y_3 = \frac{7}{8}$, що визначає ціну гри

$$v = 4 \cdot \frac{7}{8} - 1 = \frac{5}{2}.$$

Таким чином, розв'язком гри для гравця А (СТО) є змішування стратегій A_1 і A_2 з рівними ймовірностями 0,5 і 0,5, а для гравця В – змішування стратегій B_3 і B_4 з ймовірностями $\frac{7}{8}$ і $\frac{1}{8}$.

4.3.2 Індивідуальні завдання. Ігрові моделі 2х1

Необхідно реалізувати графічно гру з платіжною матрицею, що відповідає Вашому номеру варіанта (табл. 4.6).

Таблиця 4.6 – Варіанти завдань

Варіант 1				Варіант 2			
5	2	6	4	-1	2	1	-6
2	3	1	3	-4	-3	-6	-2
Варіант 3				Варіант 4			
1	5	3	4	2	-1	-5	-4
3	2	1	0	-3	-3	0	1
Варіант 5				Варіант 6			
-1	0	3	4	-2	1	-5	-1
3	2	0	-1	-5	-4	3	-3
Варіант 7				Варіант 8			
7	1	1	4	-2	-1	-5	1
-1	2	3	1	-5	-3	1	-4
Варіант 9				Варіант 10			
5	2	1	4	0	0	-5	1
-2	7	3	1	-2	-4	0	-4

Закінчення таблиці 4.6

Варіант 11				Варіант 12			
1	5	-6	4	-1	2	-3	-1
3	2	5	3	1	-3	1	-4
Варіант 13				Варіант 14			
7	0	-1	5	-2	0	-4	-1
0	4	3	3	2	-1	2	-2
Варіант 15				Варіант 16			
-2	2	7	6	4	-3	-4	-2
5	6	-1	1	-2	1	2	-3
Варіант 17				Варіант 18			
3	2	5	0	0	-1	-3	5
-2	6	-1	1	-5	1	3	-3
Варіант 19				Варіант 20			
2	-1	2	3	-4	0	-3	2
4	2	1	0	1	-5	3	-4
Варіант 21				Варіант 22			
0	-1	2	3	-6	1	3	2
4	2	-1	-3	1	-3	-5	-1
Варіант 23				Варіант 24			
4	-1	3	5	0	-3	1	-5
0	3	-2	-3	-6	1	-4	-3
Варіант 25				Варіант 26			
1	-1	3	5	3	1	-3	6
7	5	1	0	-4	3	0	-3
Варіант 27				Варіант 28			
1	-1	3	5	-4	1	-3	6
7	5	2	0	0	2	4	-3
Варіант 29				Варіант 30			
7	2	-2	3	1	-3	-2	4
3	4	2	0	-1	0	4	-3

4.4 Розв'язання матричних ігор методами лінійного програмування

4.4.1 Загальні відомості

Теорія ігор тісно пов'язана з лінійним програмуванням. Будь-яку кінцеву гру двох сторін із нульовою сумою можна представити у вигляді задачі лінійного програмування.

Нехай задана платіжна матриця (рис. 4.8)

:

		B_1	B_2	...	B_n
		y_1	y_2	...	y_n
A_1	x_1	a_{11}	a_{12}	...	a_{1n}
A_2	x_2	a_{21}	a_{22}	...	a_{2n}
...
A_m	x_m	a_{m1}	a_{m2}	...	a_{mn}

Рисунок 4.8 – Платіжна матриця

Оптимальні значення ймовірностей x_i , де $i = 1, 2, \dots, m$ можна визначити шляхом розв'язання такої максимінної задачі:

$$\max_{x_i} \left\{ \min \left(\sum_{i=1}^m a_{i1} x_i, \sum_{i=1}^m a_{i2} x_i, \dots, \sum_{i=1}^m a_{in} x_i \right) \right\},$$

$$x_1 + x_2 + \dots + x_m = 1,$$

$$x_i \geq 0, \quad i = 1, 2, \dots, m.$$

Щоб сформулювати наведену задачу у вигляді задачі лінійного програмування, покладемо:

$$v = \min \left(\sum_{i=1}^m a_{i1} x_i, \sum_{i=1}^m a_{i2} x_i, \dots, \sum_{i=1}^m a_{in} x_i \right).$$

Звідси випливає, що

$$\sum_{i=1}^m a_{ij} x_i \geq v, \quad j = 1, 2, \dots, n.$$

Отже, задача гравця A може бути записана у вигляді: максимізувати $z = v$, враховуючи обмеження:

$$\begin{cases} \sum_{i=1}^m a_{ij}x_i \geq v, j = 1, 2, \dots, n. \\ x_1 + x_2 + \dots + x_m = 1, \\ x_i \geq 0, i = 1, 2, \dots, m. \\ v \text{ не обмежене у знаку.} \end{cases}$$

Оптимальні стратегії y_1, y_2, \dots, y_n гравця B визначаються шляхом розв'язання такої задачі:

$$\min_{y_j} \left\{ \max \left(\sum_{j=1}^n a_{1j}y_j, \sum_{j=1}^n a_{2j}y_j, \dots, \sum_{j=1}^n a_{mj}y_j \right) \right\},$$

$$y_1 + y_2 + \dots + y_n = 1,$$

$$y_j \geq 0, j = 1, 2, \dots, n.$$

Використовуючи процедуру, аналогічну той, що використовувалася для гравця A , приходимо до висновку, що задача для гравця B зводиться до такої: мінімізувати $F = v$, враховуючи обмеження:

$$\begin{cases} \sum_{j=1}^n a_{ij}y_j \leq v, i = 1, 2, \dots, m, \\ y_1 + y_2 + \dots + y_n = 1, \\ y_j \geq 0, j = 1, 2, \dots, n. \\ v \text{ не обмежене у знаку.} \end{cases}$$

Обидві задачі, що отримані вище, оптимізують одну й ту саму змінну v , що не має обмежень у знаку. Тому оптимальний розв'язок однієї з задач автоматично визначає оптимальний розв'язок іншої.

Розв'яжемо таку матричну гру методами лінійного програмування:

СТО пропонує комплекс послуг із заміни ходової частини (передня підвіска): заміна амортизатору (A_1); заміна кермової тяги (A_2); заміна втулки стабілізатора (A_3), заміна ричагу (A_4). Замовляючи заміну запчастини, клієнт обирає вид гарантії на неї: без гарантії (B_1), гарантія на 5 тис. кілометрів пробігу (B_2), гарантія на два роки (B_3). У платіжній матриці (рис. 4.9) наведений прибуток СТО у випадку замовлення певної послуги з відповідним видом гарантії:

	B_1	B_2	B_3
A_1	-3	1	2
A_2	5	2	4
A_3	-3	3	5

Рисунок 4.9 – Платіжна матриця

Спочатку шукаємо ціну гри v (рис. 4.10).

	B_1	B_2	B_3	\min_j
A_1	-3	1	2	-3
A_2	5	2	4	2
A_3	-3	3	5	-3
\max_i	5	3	5	

$$\max_i \min_j (a_{ij}) = 2 = \lambda.$$

$$\min_j \max_i (a_{ij}) = 3 = \beta.$$

Рисунок 4.10 – Пошук ціни гри

Оскільки $0 < \beta \leq v \leq \lambda > 0$, можемо зробити висновок, що

$$v > 0. \quad (4.2)$$

Отже, $v = (3; 2)$, тому розв'язок гри шукаємо у змішаних стратегіях.

Задача для гравця A може бути записана двома способами:

1. Максимізувати $F = x_4$, $v = x_4$,

де v – невід'ємна величина ($x_4 \geq 0$), враховуючи обмеження:

$$\begin{cases} -3x_1 + 5x_2 - 3x_3 - x_4 \geq 0, \\ x_1 + 2x_2 + 3x_3 - x_4 \geq 0, \\ 2x_1 + 4x_2 + 5x_3 - x_4 \geq 0, \\ x_1 + x_2 + x_3 = 1, \\ x_1, x_2, x_3 \geq 0. \end{cases}$$

2. Максимізувати $F = -x_4$, $v = -x_4$,

де v – недодатна величина ($x_4 \geq 0$), враховуючи обмеження:

$$\begin{cases} -3x_1 + 5x_2 - 3x_3 + x_4 \geq 0, \\ x_1 + 2x_2 + 3x_3 + x_4 \geq 0, \\ 2x_1 + 4x_2 + 5x_3 + x_4 \geq 0, \\ x_1 + x_2 + x_3 = 1, \\ x_1, x_2, x_3 \geq 0. \end{cases}$$

Другу задачу не розв'язуємо, враховуючи (4.2).

Розв'яжемо першу задачу (табл. 4.6):

Таблиця 4.6 – Розв'язання задачі лінійного програмування (для гравця A)

Базис	x_1	x_2	x_3	x_4	Var1	Var2	Var3	Вільний член
Var1	3	-5	3	1	1	0	0	0
Var2	-1	-2	-3	1	0	1	0	0
Var3	-2	-4	-5	1	0	0	1	0
	1	1	1	0	0	0	0	1
F	0	0	0	-1	0	0	0	0

Var1	8	0	8	1	1	0	0	5
Var2	1	0	-1	1	0	1	0	2
Var3	2	0	-1	1	0	0	1	4
x_2	1	1	1	0	0	0	0	1
F	0	0	0	-1	0	0	0	0

Var1	7	0	9	0	1	-1	0	3
x_4	1	0	-1	1	0	1	0	2
Var3	1	0	0	0	0	-1	1	2
x_2	1	1	1	0	0	0	0	1
F	1	0	-1	0	0	1	0	2

x_3	7/9	0	1	0	1/9	-1/9	0	1/3
x_4	16/9	0	0	1	1/9	8/9	0	7/3
Var3	1	0	0	0	0	-1	1	2
x_2	2/9	1	0	0	-1/9	1/9	0	2/3
F	16/9	0	0	0	1/9	8/9	0	7/3

Отже, цільова функція досягає свого максимального значення:

$$F = \frac{7}{3}, \text{ за умов оптимального плану: } X_{opt} = \left\{ 0, \frac{2}{3}, \frac{1}{3}, \frac{7}{3} \right\}, \text{ тобто, коли}$$

$$x_1 = 0, x_2 = \frac{2}{3}, x_3 = \frac{1}{3}, v = \frac{7}{3}.$$

Задача для гравця B може бути записана двома способами:

1. Мінімізувати $F = u_4, v = u_4,$

де v – невід’ємна величина ($y_4 \geq 0$), враховуючи обмеження:

$$\begin{cases} -3y_1 + y_2 + 2y_3 - y_4 \leq 0, \\ 5y_1 + 2y_2 + 4y_3 - y_4 \leq 0, \\ -3y_1 + 3y_2 + 5y_3 - y_4 \leq 0, \\ y_1 + y_2 + y_3 = 1, \\ y_1, y_2, y_3 \geq 0. \end{cases}$$

2. Мінімізувати $F = -y_4$, $v = -y_4$,

де v – недодатна величина ($y_4 \geq 0$), враховуючи обмеження:

$$\begin{cases} -3y_1 + y_2 + 2y_3 + y_4 \leq 0, \\ 5y_1 + 2y_2 + 4y_3 + y_4 \leq 0, \\ -3y_1 + 3y_2 + 5y_3 + y_4 \leq 0, \\ y_1 + y_2 + y_3 = 1, \\ y_1, y_2, y_3 \geq 0. \end{cases}$$

Другу задачу не розв’язуємо, враховуючи (4.2).

Розв’яжемо першу задачу (табл. 4.7):

Таблиця 4.7 – Розв’язання задачі лінійного програмування (для гравця B)

Базис	y_1	y_2	y_3	y_4	Var1	Var2	Var3	Вільний член
Var1	-3	1	2	-1	1	0	0	0
Var2	5	2	4	-1	0	1	0	0
Var3	-3	3	5	-1	0	0	1	0
	1	1	1	0	0	0	0	1
$-F$	0	0	0	1	0	0	0	0
Var1	0	11/5	22/5	-8/5	1	3/5	0	0
y_1	1	2/5	4/5	-1/5	0	1/5	0	0
Var3	0	21/5	37/5	-8/5	0	3/5	1	0
	0	3/5	1/5	1/5	0	-1/5	0	1
$-F$	0	0	0	1	0	0	0	0
Var1	0	7	6	0	1	-1	0	8
y_1	1	1	1	0	0	0	0	1
Var3	0	9	9	0	0	-1	1	8
y_4	0	3	1	1	0	-1	0	5
$-F$	0	-3	-1	0	0	1	0	-5

Продовження таблиці 4.7

Var1	0	0	-1	0	1	-2/9	-7/9	16/9
y_1	1	0	0	0	0	1/9	-1/9	1/9
y_2	0	1	1	0	0	-1/9	1/9	8/9
y_4	0	0	-2	1	0	-2/3	-1/3	7/3
$-F$	0	0	2	0	0	2/3	1/3	-7/3

Отже, цільова функція досягає свого максимального значення:
 $F = -\left(-\frac{7}{3}\right) = \frac{7}{3}$, за умов оптимального плану: $Y_{opt} = \left\{\frac{1}{9}, \frac{8}{9}, 0, \frac{7}{3}\right\}$, тобто,
 коли

$$y_1 = \frac{1}{9}, y_2 = \frac{8}{9}, y_3 = 0, v = \frac{7}{3}.$$

Загальний розв'язок гри: ціна гри: $v = \frac{7}{3}$;

$$\text{стратегія гравця } A = \left\{0, \frac{2}{3}, \frac{1}{3}\right\};$$

$$\text{стратегія гравця } B = \left\{\frac{1}{9}, \frac{8}{9}, 0\right\}.$$

4.4.2 Індивідуальні завдання для розв'язання матричних ігор методами лінійного програмування

Розв'яжіть задачу, що відповідає номеру Вашого варіанта (табл. 4.8), методами лінійного програмування:

- як задачу лінійного програмування для гравця A ;
- як задачу лінійного програмування для гравця B ;
- порівняйте результати.

Таблиця 4.8 – Варіанти завдань

Варіант 1				Варіант 2			
	-3	1	2		2	1	2
	5	2	4		1	4	0
	-3	3	5		-3	-1	2
Варіант 3				Варіант 4			
	3	-6	3		15	2	10
	0	3	-5		-2	9	2
	1	2	1		-5	-6	1

Продовження таблиці 4.8

Варіант 5				Варіант 6			
	9	5	12		-9	-6	3
	-1	3	-6		-15	-3	-10
	0	8	2		-6	2	-9
Варіант 7				Варіант 8			
	-10	-6	3		-6	-4	-2
	-25	-2	-10		4	-1	-3
	-6	2	-9		-5	-2	-1
Варіант 9				Варіант 10			
	-8	7	2		10	20	5
	5	1	7		-15	-5	15
	-3	-2	-1		5	-10	-5
Варіант 11				Варіант 12			
	-7	5	6		-12	5	-7
	8	9	5		10	4	2
	-3	-2	7		-4	-2	6
Варіант 13				Варіант 14			
	-12	5	-7		2	-7	4
	10	4	2		5	-9	5
	-4	-2	6		-2	-2	-7
Варіант 15				Варіант 16			
	-4	-2	3		-1	-2	-3
	10	5	12		-5	7	3
	1	10	-3		-6	4	-2
Варіант 17				Варіант 18			
	8	-7	4		12	1	2
	1	10	5		8	13	9
	-2	-6	3		-4	-1	3
Варіант 19				Варіант 20			
	-3	4	2		5	3	30
	5	2	7		-1	6	2
	-2	-1	-3		-4	2	10
Варіант 21				Варіант 22			
	6	3	-7		5	3	4
	-1	6	-5		-1	-7	3
	-4	2	-2		-2	4	2

Закінчення таблиці 4.8

Варіант 23				Варіант 24			
	5	-6	-8		-6	-3	-5
	-1	-10	3		-2	5	-5
	-6	-4	5		-5	-7	3
Варіант 25				Варіант 26			
	-6	3	-5		-6	3	6
	-1	5	-3		2	5	1
	-5	7	3		1	7	-3
Варіант 27				Варіант 28			
	-2	-5	4		1	6	2
	-4	2	-1		-1	-4	-3
	-6	5	-3		2	-5	-4
Варіант 29				Варіант 30			
	3	2	-9		7	1	2
	-1	-5	-1		5	10	9
	-3	-5	-4		-4	-1	3

РОЗДІЛ 5 ПРИЙНЯТТЯ РІШЕНЬ В УМОВАХ НЕВИЗНАЧЕНОСТІ

5.1 Загальні відомості

Основна відмінність моделей прийняття рішень в умовах невизначеності від моделей, що розглядаються в теорії ігор, полягає у тому, що особі, яка приймає рішення, протистоїть так звана «природа». Це означає, що супротивник не переслідує власних цілей, протилежних цілям особи, яка приймає рішення. «Природа» не намагається своїми діями зашкодити «супротивнику». Натомість у теорії ігор супротивник приймає свідомі рішення, що спрямовані на відстоювання власних інтересів, зазвичай протилежних інтересам іншого гравця. В якості «природи» можуть виступати економічні чинники (наприклад, курси валют, попит на продукцію), погодні явища тощо.

Дані, необхідні для прийняття рішень в умовах невизначеності, зазвичай задаються у формі матриці, рядки якої відповідають можливим діям особи, яка приймає рішення, а стовпці – можливим станам системи (оскільки вважаємо, що «природа» не приймає рішення, а знаходиться в певних станах).

Таблиця 5.1 – Матриця моделі прийняття рішень в умовах невизначеності

	θ_1	θ_2	...	θ_n
a_1	$v(a_1, \theta_1)$	$v(a_1, \theta_2)$...	$v(a_1, \theta_n)$
a_2	$v(a_2, \theta_1)$	$v(a_2, \theta_2)$...	$v(a_2, \theta_n)$
...
a_m	$v(a_m, \theta_1)$	$v(a_m, \theta_2)$...	$v(a_m, \theta_n)$

Для прийняття рішень в умовах невизначеності використовуються такі критерії:

- 1) критерій Лапласа (принцип недостатнього обґрунтування);
- 2) мінімаксний критерій (найкраще з найгіршого);
- 3) критерій Севіджа (жалю);

4) критерій Гурвіца (баланс між оптимізмом і песимізмом).

5.2 Критерії прийняття рішень в умовах невизначеності

5.2.1 Критерій Лапласа (принцип недостатнього обґрунтування)

Цей критерій спирається на відомий принцип недостатнього обґрунтування. Оскільки ймовірності станів $\theta_1, \theta_2, \dots, \theta_n$ невідомі, відсутня необхідна інформація для того, щоб зробити висновок, що ці ймовірності різні. У протилежному випадку можна було б визначити ці ймовірності й ситуація не відповідала би моделі прийняття рішень в умовах невизначеності. Згідно з принципом недостатнього обґрунтування, стани $\theta_1, \theta_2, \dots, \theta_n$ мають рівні ймовірності. Отже, задачу можна розглядати як задачу прийняття рішення в умовах ризику, коли обирається дія a_i , що дає найбільший очікуваний виграш.

Іншими словами, необхідно знайти дії a_i^* , що відповідають

$$\max_{a_i} \left\{ \frac{1}{n} \sum_{j=1}^n v(a_i, \theta_j) \right\},$$

де $\frac{1}{n}$ – ймовірність реалізації стану θ_j , ($j = 1, 2, \dots, n$).

Така постановка задачі стосується оптимізації прибутку. Якщо б мова про оптимізацію витрат, то необхідно було б знаходити мінімум функції.

5.2.2 Мінімаксний критерій (найкраще з найгіршого)

Цей критерій вважається «найобережнішим», оскільки він базується на виборі найкращої серед найгірших можливостей. Якщо результат $v(a_i, \theta_j)$ представляє собою витрати особи, яка приймає рішення, для дії a_i найбільші витрати незалежно від можливого стану θ_j будуть дорівнювати

$$\max_{\theta_j} \{v(a_i, \theta_j)\}.$$

За мінімаксним критерієм має обиратися дія a_i , що дає

$$\min_{a_i} \max_{\theta_j} \{v(a_i, \theta_j)\}.$$

Аналогічно в тому випадку, коли $v(a_i, \theta_j)$ представляє виграш, згідно з мінімаксним критерієм, обирається дія a_i , що дає

$$\max_{a_i} \min_{\theta_j} \{v(a_i, \theta_j)\}.$$

У цьому випадку критерій називається максимінним.

5.2.3 Критерій Севіджа (жалю)

Критерій мінімакса, що розглянутий у попередньому пункті, є настільки «песимістичним», що інколи призводить до нелогічних висновків. Розглянемо таку матрицю витрат:

$$v(a_i, \theta_j) =$$

	θ_1	θ_2
a_1	11.000 дол.	90 дол.
a_2	10.000 дол.	10.000 дол.

Використання мінімаксного критерію приводить до вибору стратегії a_2 . Але інтуїтивно ми схилиємося до вибору a_1 , оскільки не виключено, що θ буде дорівнювати θ_2 і витрати становитимуть лише 90 доларів, тоді як вибор a_2 завжди призводить до витрат 10.000 доларів.

Критерій Севіджа «виправляє» ситуацію шляхом введення нової матриці жалю (витрат), в якій $v(a_i, \theta_j)$ змінюємо на $r(a_i, \theta_j)$, які знаходимо за формулою

$$r(a_i, \theta_j) = \begin{cases} \max_{a_k} \{v(a_k, \theta_j)\} - v(a_i, \theta_j), \\ v(a_i, \theta_j) - \min_{a_k} \{v(a_k, \theta_j)\}. \end{cases}$$

Перший рядок функції $r(a_i, \theta_j)$ відповідає прибутку, а другий – витратам.

Це означає, що $r(a_i, \theta_j)$ є різницею між найкращим значенням у стовпці θ_j і значенням $v(a_i, \theta_j)$ при тому ж θ_j . По суті $r(a_i, \theta_j)$ виражає «переживання» особи, яка приймає рішення, з приводу того, що він не обрав найкращої дії відносно стану θ_j . Функція $r(a_i, \theta_j)$ називається матрицею жалю.

Щоб показати, як введені величини $r(a_i, \theta_j)$ допомагають дійти логічного висновку в наведеному вище прикладі, розглянемо матрицю жалю (з точки зору витрат):

$$v(a_i, \theta_j) =$$

	θ_1	θ_2
a_1	1.000 дол.	0 дол.
a_2	0 дол.	9.910 дол.

Застосовуючи мінімаксний критерій, робимо висновок, що необхідно обрати a_1 .

Варто зазначити, що незалежно від того, є $v(a_i, \theta_j)$ матрицею прибутку чи витрат, матриця жалю $r(a_i, \theta_j)$ завжди визначає витрати. Отже, до матриці $r(a_i, \theta_j)$ можна застосовувати тільки мінімаксний (а не максимінний) критерій.

5.2.4 Критерій Гурвіца (баланс між оптимізмом і песимізмом)

Цей критерій охоплює різні підходи до прийняття рішень: від найбільш оптимістичного до найбільш песимістичного. При найбільш оптимістичному підході можна обрати дію, що дає $\max_{a_i} \max_{\theta_j} \{v(a_i, \theta_j)\}$.

(Припускаючи, що $v(a_i, \theta_j)$ представляє прибуток.) Аналогічно при найбільш песимістичних припущеннях дія, що обирається, відповідає $\max_{a_i} \min_{\theta_j} \{v(a_i, \theta_j)\}$. Критерій Гурвіца встановлює баланс між випадками

крайнього оптимізму й крайнього песимізму шляхом зважування обох способів поведінки з відповідними вагами α і $1-\alpha$, де $0 \leq \alpha \leq 1$. У тому випадку, коли $v(a_i, \theta_j)$ представляє прибуток, критерій обирає дію, що дає

$$\max_{a_i} \left\{ \alpha \max_{\theta_j} v(a_i, \theta_j) + (1-\alpha) \min_{\theta_j} v(a_i, \theta_j) \right\}.$$

У тому випадку, коли $v(a_i, \theta_j)$ представляє витрати, критерій обирає дію, що дає

$$\max_{a_i} \left\{ \alpha \min_{\theta_j} v(a_i, \theta_j) + (1-\alpha) \max_{\theta_j} v(a_i, \theta_j) \right\}.$$

Перший доданок визначає долю оптимізму, а другий – долю песимізму.

Параметр α визначається як показник оптимізму: при $\alpha = 1$ критерій занадто оптимістичний; при $\alpha = 0$ – занадто песимістичний. Значення α

між 0 і 1 може визначатися залежно від схильності особи, що приймає рішення, до песимізму чи оптимізму. За відсутності яскраво вираженої схильності $\alpha = \frac{1}{2}$ здається найбільш розумним.

5.2.5 Приклад застосування критеріїв прийняття рішення в умовах невизначеності

Застосування критеріїв прийняття рішень продемонструємо на такій задачі: перед початком зимового сезону шиномонтажне підприємство планує виділити частину свого приміщення під обладнання для надання однієї з послуг (балансування коліс, встановлення зимових шин, шиповка шин, ліквідація проколів коліс). Точна кількість клієнтів невідома, але очікується, що вона може приймати одне з чотирьох значень (200, 250, 300 чи 350). Для кожного з цих можливих значень існує найкраща послуга (з точки зору можливих витрат). Встановлення обладнання для надання інших послуг призводить до додаткових витрат або через більший рівень пропозицій порівняно з попитом, або через неповне задоволення попиту.

Нижче наведено матрицю, що визначає витрати на расходні матеріали в сотнях гривень.

Таблиця 5.2 – Матриця витрат

		Клієнти				
		200	250	300	350	
Послуги		θ_1	θ_2	θ_3	θ_4	
	балансування коліс	a_1	5	10	18	25
	встановлення зимових шин	a_2	8	7	8	23
	шиповка шин	a_3	21	18	12	21
	ліквідація проколів коліс	a_4	30	22	19	15

1. Застосуємо критерій Лапласа (принцип недостатнього обґрунтування).

Принцип Лапласа припускає, що $\theta_1, \theta_2, \theta_3, \theta_4$ рівноймовірні. Отже, $P\{\theta = \theta_j\} = \frac{1}{4}$, $j = 1, 2, 3, 4$, й очікувані витрати від різних дій a_1, a_2, a_3, a_4 становлять:

$$M\{a_1\} = \left(\frac{1}{4}\right)(5 + 10 + 18 + 25) = 14,5;$$

$$M\{a_2\} = \left(\frac{1}{4}\right)(8 + 7 + 8 + 2 \setminus 3) = 11,5;$$

$$M\{a_3\} = \left(\frac{1}{4}\right)(21 + 18 + 12 + 21) = 18,0;$$

$$M\{a_4\} = \left(\frac{1}{4}\right)(30 + 22 + 19 + 15) = 21,5.$$

З отриманих значень обираємо найменше – 11,5. Таким чином, найкраща пропозиція, відповідно до критерію Лапласа, a_2 , тобто, необхідно встановити обладнання для встановлення зимових шин.

2. Застосуємо мінімаксий критерій (найкраще з найгіршого).

Таблиця 5.3 – Реалізація мінімаксного критерію

	θ_1	θ_2	θ_3	θ_4	max
a_1	5	10	18	25	25
a_2	8	7	8	23	23
a_3	21	18	12	21	21
a_4	30	22	19	15	30

В останньому стовпці обираємо найменше значення – 21. Йому відповідає стратегія a_3 , тобто необхідно встановити обладнання для шиповки шин.

3. Застосуємо критерій Севіджа (жалю).

Складемо матрицю жалю (таблиця 5.4). Для цього знайдемо мінімальний елемент кожного стовпця й віднімемо його з усіх елементів цього стовпця.

Таблиця 5.4 – Матриця жалю

	θ_1	θ_2	θ_3	θ_4	max
a_1	0	3	10	10	10
a_2	3	0	0	8	8
a_3	16	11	4	6	16
a_4	25	15	11	0	25

В останньому стовпці обираємо найменше значення – 8. Йому відповідає стратегія a_2 , тобто необхідно розмістити обладнання для встановлення зимових шин.

4. Застосуємо критерій Гурвіца (баланс між оптимізмом і песимізмом).

Вважаємо, що $\alpha = \frac{1}{2}$.

Таблиця 5.5 – Реалізація критерію Гурвіца

	$\min_{\theta_j} v(a_i, \theta_j)$ мінімум рядка	$\max_{\theta_j} v(a_i, \theta_j)$ максимум рядка	$\alpha \cdot \min_{\theta_j} v(a_i, \theta_j) + (1 - \alpha) \cdot \max_{\theta_j} v(a_i, \theta_j)$
a_1	5	25	15
a_2	7	23	15
a_3	12	21	16,5
a_4	15	30	22,5

В останньому стовпці обираємо найменше значення – 15. Йому відповідають дві стратегії a_1 і a_2 , тобто необхідно розмістити обладнання для балансування коліс або встановлення зимових шин.

Результати використання чотирьох методів наведемо в таблиці 5.6.

Таблиця 5.6 – Результати

Метод	Результат
Критерій Лапласа (принцип недостатнього обґрунтування)	a_2
Мінімаксний критерій (найкраще з найгіршого)	a_3
Критерій Севіджа (жалю)	a_2
Критерій Гурвіца (баланс між оптимізмом і песимізмом)	a_1 чи a_2

Оскільки більшість критеріїв дали відповідь a_2 , обираємо послугу: встановлення зимових шин.

Зауваження 5.1. Ми розглянули випадок, коли матриця виражає витрати. Знайдіть оптимальну стратегію з використання приміщення шиномонтажного підприємства, вважаючи, що матриця виражає прибуток.

5.2.6 Індивідуальні завдання. Прийняття рішень в умовах невизначеності

Реалізуйте модель прийняття рішень в умовах невизначеності за

допомогою критеріїв Лапласа, Севіджа, Гурвіца й мінімаксного критерію, знаючи платіжну матрицю (таблиця 5.7).

Таблиця 5.7 – Варіанти завдань

Варіант 1	θ_1	θ_2	θ_3	θ_4
a_1	5	10	22	25
a_2	8	7	8	22
a_3	21	17	12	21
a_4	30	22	18	15
Варіант 2	θ_1	θ_2	θ_3	θ_4
a_1	5	18	25	25
a_2	6	7	8	22
a_3	21	14	12	21
a_4	30	22	18	15
Варіант 3	θ_1	θ_2	θ_3	θ_4
a_1	6	3	7	1
a_2	8	2	4	5
a_3	3	0	4	5
a_4	6	7	8	9
Варіант 4	θ_1	θ_2	θ_3	θ_4
a_1	6	2	3	5
a_2	8	4	5	3
a_3	6	1	5	8
a_4	6	7	8	9

Продовження таблиці 5.7

Варіант 5	θ_1	θ_2	θ_3	θ_4
a_1	3	8	1	4
a_2	7	5	6	2
a_3	8	2	9	12
a_4	13	5	11	7
Варіант 6	θ_1	θ_2	θ_3	θ_4
a_1	11	2	23	4
a_2	8	17	33	1
a_3	12	25	4	85
a_4	16	7	15	14
Варіант 7	θ_1	θ_2	θ_3	θ_4
a_1	26	18	24	3
a_2	12	8	13	6
a_3	18	8	53	1
a_4	24	41	6	21
Варіант 8	θ_1	θ_2	θ_3	θ_4
a_1	29	19	24	3
a_2	12	8	13	6
a_3	18	16	51	1
a_4	23	45	6	21

Продовження таблиці 5.7

Варіант 9	θ_1	θ_2	θ_3	θ_4
a_1	16	5	21	4
a_2	3	54	21	25
a_3	11	6	87	3
a_4	24	8	11	3
Варіант 10	θ_1	θ_2	θ_3	θ_4
a_1	3	21	4	62
a_2	2	51	7	9
a_3	5	3	14	6
a_4	18	8	24	7
Варіант 11	θ_1	θ_2	θ_3	θ_4
a_1	13	5	17	8
a_2	24	6	51	9
a_3	12	5	14	2
a_4	3	52	7	48
Варіант 12	θ_1	θ_2	θ_3	θ_4
a_1	3	12	5	41
a_2	63	17	9	5
a_3	15	28	10	30
a_4	2	18	63	7

Продовження таблиці 5.7

Варіант 13	θ_1	θ_2	θ_3	θ_4
a_1	22	3	14	5
a_2	4	27	12	9
a_3	24	17	36	32
a_4	5	67	4	12
Варіант 14	θ_1	θ_2	θ_3	θ_4
a_1	5	17	52	9
a_2	18	42	6	85
a_3	13	2	31	7
a_4	8	14	5	9
Варіант 15	θ_1	θ_2	θ_3	θ_4
a_1	2	17	52	9
a_2	14	42	6	85
a_3	18	2	31	7
a_4	8	21	5	9
Варіант 16	θ_1	θ_2	θ_3	θ_4
a_1	11	6	87	3
a_2	18	1	86	4
a_3	14	2	13	5
a_4	3	17	51	2

Продовження таблиці 5.7

Варіант 17	θ_1	θ_2	θ_3	θ_4
a_1	11	6	87	14
a_2	18	1	86	2
a_3	21	2	27	26
a_4	25	17	35	31
Варіант 18	θ_1	θ_2	θ_3	θ_4
a_1	31	7	54	6
a_2	23	14	5	11
a_3	3	24	58	61
a_4	36	2	14	12
Варіант 19	θ_1	θ_2	θ_3	θ_4
a_1	42	17	54	6
a_2	23	14	14	11
a_3	15	24	58	61
a_4	36	2	29	12
Варіант 20	θ_1	θ_2	θ_3	θ_4
a_1	21	3	54	7
a_2	18	4	21	24
a_3	7	12	35	4
a_4	11	27	2	6

Продовження таблиці 5.7

Варіант 21	θ_1	θ_2	θ_3	θ_4
a_1	32	3	37	7
a_2	18	4	21	24
a_3	17	12	35	4
a_4	11	26	15	6
Варіант 22	θ_1	θ_2	θ_3	θ_4
a_1	32	3	28	16
a_2	45	13	21	24
a_3	17	12	35	4
a_4	19	26	21	23
Варіант 23	θ_1	θ_2	θ_3	θ_4
a_1	16	18	25	40
a_2	20	4	12	5
a_3	5	20	53	7
a_4	3	65	26	15
Варіант 24	θ_1	θ_2	θ_3	θ_4
a_1	21	19	25	42
a_2	15	4	12	5
a_3	5	18	53	7
a_4	3	65	11	15

Продовження таблиці 5.7

Варіант 25	θ_1	θ_2	θ_3	θ_4
a_1	15	1	29	42
a_2	16	4	8	35
a_3	16	18	53	7
a_4	3	26	11	15
Варіант 26	θ_1	θ_2	θ_3	θ_4
a_1	28	17	29	42
a_2	12	53	8	32
a_3	14	18	75	17
a_4	3	26	33	15
Варіант 27	θ_1	θ_2	θ_3	θ_4
a_1	31	5	17	6
a_2	25	31	55	18
a_3	32	15	46	7
a_4	2	34	35	19
Варіант 28	θ_1	θ_2	θ_3	θ_4
a_1	5	12	3	17
a_2	6	15	24	3
a_3	7	21	6	15
a_4	2	14	17	24

Закінчення таблиці 5.7

Варіант 29	θ_1	θ_2	θ_3	θ_4
a_1	5	16	8	14
a_2	24	15	75	3
a_3	7	21	62	5
a_4	2	13	29	24
Варіант 30	θ_1	θ_2	θ_3	θ_4
a_1	9	19	21	3
a_2	13	24	76	5
a_3	15	8	19	32
a_4	29	7	64	5

РОЗДІЛ 6 МАРКІВСЬКІ ПРОЦЕСИ

6.1 Теоретичні відомості про марківські процеси

6.1.1 Поняття марківського випадкового процесу

Система масового обслуговування (СМО) — система, яка виконує обслуговування вимог за допомогою обслуговуючих приладів. Класична СМО містить від одного до нескінченного числа приладів. Прикладами таких систем є телефонні мережі, ремонтні майстерні, магазини, перукарні тощо.

Процес роботи СМО є випадковим. *Випадковим (імовірностним або стохастичним) процесом* називається процес зміни в часі стану якої-небудь системи відповідно до ймовірних закономірностей. Процес називається *процесом із дискретними станами*, якщо його можливі стани S_1, S_2, S_3, \dots можна заздалегідь перелічити, а перехід системи зі стану в стан відбувається миттєво (стрибком). Процес називається *процесом із безперервним часом*, якщо моменти можливих переходів системи зі стану в стан не фіксовані заздалегідь, а випадкові. Процес роботи СМО являє собою випадковий процес із дискретними станами й безперервним часом. Це означає, що стан СМО змінюється стрибком у випадкові моменти появи якихось подій (наприклад, приходу нової заявки, закінчення обслуговування).

Проводити математичний аналіз роботи СМО легше, якщо процес її функціонування є марківським. Випадковий процес називається *марківським або випадковим процесом без наслідку*, якщо для будь-якого моменту часу t_0 імовірність характеристики процесу в майбутньому залежать тільки від його стану в даний момент t_0 і не залежать від того, коли й як система прийшла в цей стан.

Приклад марківського процесу: система S – лічильник у таксі. Стан системи в момент t характеризується кількістю кілометрів (десятих частин кілометрів), що пройдені автомобілем до даного моменту. Нехай у момент t_0 лічильник показує S_0 . Імовірність того, що в момент $t > t_0$ лічильник покаже ту або іншу кількість кілометрів S_1 , залежить від S_0 , але не залежить від того, в які моменти часу змінювалися показання лічильника до моменту t_0 .

Багато процесів можна приблизно вважати марківськими. Наприклад, процес гри у шахи; система S – група шахових фігур. Стан

системи характеризується кількістю фігур супротивника, що збереглися на дощі в момент t_0 . Імовірність того, що в момент $t > t_0$ матеріальна перевага буде на боці одного з супротивників, залежить у першу чергу від того, в якому стані знаходяться система в даний момент t_0 , а не від того, коли й в якій послідовності зникли фігури з дошки до моменту t_0 .

У ряді випадків передісторією розглянутих процесів можна просто зневажити й застосовувати для їхнього вивчення марківські моделі.

При аналізі випадкових процесів із дискретними станами зручно користуватися геометричною схемою – так званим *графом станів*. Зазвичай стани системи зображуються прямокутниками (кружками), а можливі переходи зі стану в стан – стрілками (орієнтованими дугами), що з'єднують стани.

Наприклад, побудуємо граф станів такого випадкового процесу: станція технічного обслуговування (СТО) пропонує комплект послуг (перевірку ходової частини й перевірку кутів розвал-сходження), для чого встановлені відповідні стенди. Кожна з послуг у випадковий момент часу може бути недоступною для клієнтів, що прибули на СТО (стенд зайнятий чи вийшов з ладу). Час, на протязі якого неможливо перевірити ходову частину чи кути розвал-сходження, заздалегідь невідомий.

Можливі стани системи:

S_0 – обидва стенди можна використовувати;

S_1 – стенд для перевірки ходової частини недоступний, а стенд для перевірки кутів розвал-сходження доступний;

S_2 – стенд для перевірки кутів розвал-сходження недоступний, а стенд для перевірки ходової частини доступний;

S_3 – недоступні обидва стенди.

Граф описаної системи наведений на рисунку 6.1.

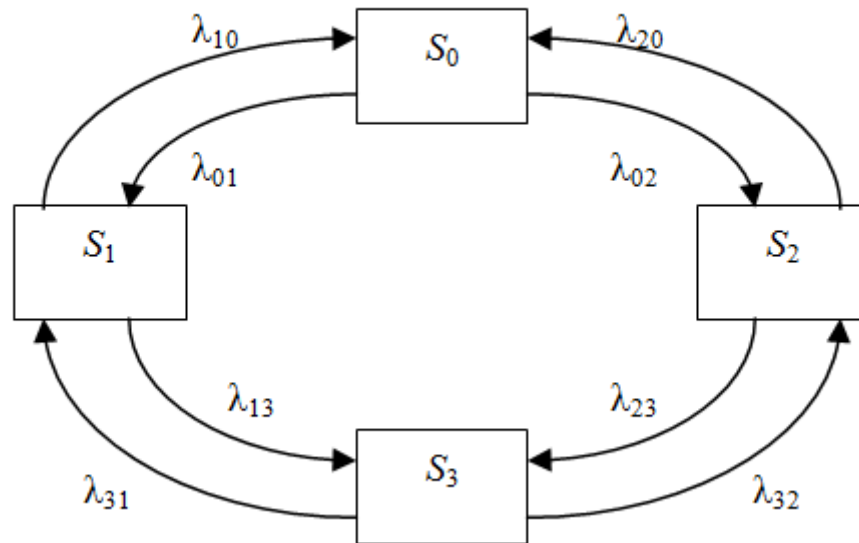


Рисунок 6.1 – Граф станів випадкового процесу

Стрілка, що спрямована, скажімо, з S_0 у S_1 , означає перехід системи в інший стан у момент, коли стенд перевірки ходової частини стає недоступним, а з S_1 у S_0 – перехід системи в інший стан, коли цей стенд стає доступним.

На графі відсутні стрілки з S_0 у S_3 і з S_1 у S_2 , оскільки недоступність стенду перевірки ходової частини й стенду перевірки кутів розвал-сходження вважаються незалежними, тому, наприклад, імовірністю одночасної недоступності обох стендів (перехід з S_0 у S_3) можна зневажити.

6.1.2 Потоки подій

Для математичного описання марківського випадкового процесу з дискретними станами й неперервним часом, що протікає у СМО, використовують поняття *потіку подій*.

Під потоком подій розуміють послідовність однорідних подій, що ідуть одне за одним у випадкові моменти часу (наприклад, потік викликів на телефонній станції, потік покупців тощо).

Потік характеризується інтенсивністю λ , тобто частотою появи подій або середнім числом подій, що надходять у СМО за одиницю часу.

Потік називається *регулярним*, якщо події ідуть одне за одним через визначені рівні проміжки часу. Наприклад, потік виробів на конвеєрі певного цеху (з постійною швидкістю руху) є регулярним.

Потік подій називається *стаціонарним*, якщо його імовірнісні

характеристики не залежать від часу. Зокрема, інтенсивність стаціонарного потоку є величиною сталою $\lambda(t) = \lambda$. Наприклад, потік автомобілів на міському проспекті не є стаціонарним протягом доби, але цей потік можна вважати стаціонарним, скажімо, у годину пік. Звертаємо увагу на те, що в останньому випадку фактична кількість автомобілів, що проїжджають в одиницю часу (наприклад, щохвилини), може не бути сталою, але середня їх кількість буде сталою й не залежатиме від часу.

Потік подій називається *потоким без наслідку*, якщо для будь-яких двох проміжків часу, що не перетинаються (τ_1 і τ_2) кількість подій, що відбувається в один з них, не залежить від кількості подій, що відбувається в інший. Наприклад, потік пасажирів, що входять у метро, практично не має наслідку. А, скажімо, потік покупців, що відходять з покупками від прилавку, вже має наслідок (хоча б тому, що інтервал часу між окремими покупцями не може бути меншим за мінімальний час обслуговування кожного з них).

Потік подій називається *ординарним*, якщо імовірність потрапляння на малу (елементарну) ділянку часу Δt двох і більше подій нескінченно мала в порівнянні з імовірністю потрапляння однієї події. Іншими словами, потік подій ординарний, якщо події з'являються в ньому поодиноці, а не групами. Наприклад, потік потягів, що підходять до станції, ординарний, а потік вагонів не ординарний.

Потік подій називається *найпростішим* (або *стаціонарним пуассонівським*), якщо він одночасно стаціонарний, ординарний і не має наслідку. Назва «найпростіший» пояснюється тим, що СМО з найпростішими потоками має порівняно просте математичне описання.

6.1.3 Рівняння Колмогорова. Граничні ймовірності станів

Розглянемо математичне описання марківського процесу з дискретними станами й неперервним часом на прикладі випадкового процесу з пункту 6.1.1, граф якого зображено на рисунку 6.1.

Будемо думати, що всі переходи системи зі стану S_i у стан S_j проходять під впливом найпростіших потоків подій з інтенсивностями λ_{ij} ($i, j = 0, 1, 2, 3$). Так, наприклад, перехід системи зі стану S_0 у стан S_1 буде відбуватися під впливом потоку «зайнятості стенду перевірки ходової частини», а зворотній перехід зі стану S_1 у стан S_0 – під впливом потоку «звільнення стенду перевірки ходової частини».

Граф станів системи називається *розміченим*, якщо біля стрілок зазначені відповідні інтенсивності.

Система, яку ми розглядаємо, має чотири можливі стани: S_0, S_1, S_2, S_3 .

Імовірність i -го стану ($p_i(t)$) – ймовірність того, що в момент часу t система буде знаходитися в стані S_i .

Твердження 6.1. Для будь-якого моменту часу t сума ймовірностей усіх станів дорівнює одиниці:

$$\sum_{i=1}^3 p_i(t) = 1. \quad (6.1)$$

Доведемо це твердження.

Введемо події A_i ($i = 0, 1, 2, 3$) – система знаходиться в стані S_i . Сума подій A_i ($i = 0, 1, 2, 3$) є достовірною подією, тому її ймовірність дорівнює 1:

$$P\left(\sum_{i=0}^3 A_i\right) = 1.$$

Оскільки події A_i ($i = 0, 1, 2, 3$) є несумісними, можемо застосувати теорему про ймовірність суми несумісних подій. У результаті отримаємо:

$$P\left(\sum_{i=0}^3 A_i\right) = \sum_{i=0}^3 P(A_i) = 1.$$

Враховуючи, що $P(A_i) = p_i(t)$, можемо записати:

$$\sum_{i=1}^3 p_i(t) = 1.$$

Твердження доведено.

А. М. Колмогоров отримав систему диференціальних рівнянь для знаходження ймовірностей $p_i(t)$, яка для випадку можливих чотирьох станів системи запишеться у вигляді:

$$\begin{cases} p_0' = \lambda_{10} p_1 + \lambda_{20} p_2 - (\lambda_{01} + \lambda_{02}) p_0, \\ p_1' = \lambda_{01} p_0 + \lambda_{31} p_3 - (\lambda_{10} + \lambda_{13}) p_1, \\ p_2' = \lambda_{02} p_0 + \lambda_{32} p_3 - (\lambda_{20} + \lambda_{23}) p_2, \\ p_3' = \lambda_{13} p_1 + \lambda_{23} p_2 - (\lambda_{31} + \lambda_{32}) p_3. \end{cases} \quad (6.2)$$

У лівій частині кожного з рівнянь системи (6.2) стоїть похідна ймовірності i -го стану. У правій частині – сума добутків ймовірностей усіх станів, з яких ідуть стрілки в даний стан, на інтенсивності відповідних потоків подій мінус сумарна інтенсивність усіх потоків, що виводять систему з даного стану, помножена на ймовірність даного (i -го) стану.

Рівняння Колмогорова дають можливість знайти всі ймовірності

станів як *функції часу*. Особливий інтерес представляють ймовірності системи $p_i(t)$ у *граничному стаціонарному режимі*, тобто при $t \rightarrow \infty$, що називаються *граничними* (або *фінальними*) ймовірностями станів.

У теорії випадкових процесів доведено, що *якщо кількість станів системи скінченна й з кожного з них можна (за кінцеву кількість кроків) перейти в будь-який інший стан, то граничні ймовірності існують*.

Гранична ймовірність стану S_i має такий зміст: вона показує *середній відносний час перебування системи в цьому стані*. Наприклад, якщо гранична ймовірність стану S_0 дорівнює 0,5, тобто $p_0 = 0,5$, то це означає, що в середньому половину часу система знаходиться в стані S_0 .

Оскільки граничні ймовірності сталі, то, замінивши в рівняннях Колмогорова їх похідні нульовими значеннями, одержимо систему лінійних алгебраїчних рівнянь, що описують стаціонарний режим. Для системи S із графом станів, що зображений на рисунку 6.1, така система рівнянь має вигляд:

$$\begin{cases} (\lambda_{01} + \lambda_{02})p_0 = \lambda_{10}p_1 + \lambda_{20}p_2, \\ (\lambda_{10} + \lambda_{13})p_1 = \lambda_{01}p_0 + \lambda_{31}p_3, \\ (\lambda_{20} + \lambda_{23})p_2 = \lambda_{20}p_0 + \lambda_{32}p_3, \\ (\lambda_{31} + \lambda_{32})p_3 = \lambda_{13}p_1 + \lambda_{23}p_2. \end{cases} \quad (6.3)$$

Систему (6.3) можна скласти безпосередньо за розміченим графом, якщо керуватися таким правилом: *ліворуч у рівняннях стоїть гранична ймовірність даного стану p_i , помножена на сумарну інтенсивність усіх потоків, що ведуть із даного стану, а праворуч – сума добутків інтенсивностей усіх потоків, що входять у i -ий стан, на ймовірності тих подій, з яких ці потоки виходять*.

6.2 Практична реалізація дослідження марківських процесів

6.2.1 Приклад дослідження марківського процесу

Необхідно розв'язати такі задачі для системи S , граф станів якої зображений на рисунку 6.1:

- 1) знайти граничні ймовірності, якщо $\lambda_{01} = 1$, $\lambda_{02} = 2$, $\lambda_{10} = 2$, $\lambda_{13} = 2$, $\lambda_{20} = 3$, $\lambda_{23} = 1$, $\lambda_{31} = 3$, $\lambda_{32} = 2$;
- 2) знайти середній чистий прибуток від експлуатації в стаціонарному режимі системи S , якщо відомо, що в одиницю часу робота стану перевірки ходової частини й стану перевірки кутів розвал-сходження приносить

прибуток відповідно у 10 і 6 грош. од, а їх недоступність вимагає витрат відповідно у 4 і 2 грош. од;

- 3) оцінити економічну ефективність наявної можливості зменшення вдвічі середнього часу недоступності кожного стенду, якщо при цьому доведеться вдвічі збільшити витрати на їх недоступність.

Розв'язання:

1) для запису системи алгебраїчних рівнянь спираємося на систему (6.3), але замість «зайвого» останнього рівняння пишемо нормувальну умову (6.1):

$$\begin{cases} 3p_0 = 2p_1 + 3p_2, \\ 4p_1 = p_0 + 3p_3, \\ 4p_2 = 2p_0 + 2p_3, \\ p_0 + p_1 + p_2 + p_3 = 1. \end{cases}$$

Розв'язавши систему, отримаємо: $p_0 = 0,4$, $p_1 = 0,2$, $p_2 = 0,27$, $p_3 = 0,13$, тобто в граничному стаціонарному режимі система S буде у середньому 40 % часу знаходитись у стані S_0 (обидва стенди можна використовувати), 20 % – у стані S_1 (стенд для перевірки ходової частини недоступний, а стенд для перевірки кутів розвал-сходження доступний), 27 % – у стані S_2 (стенд для перевірки кутів розвал-сходження недоступний, а стенд для перевірки ходової частини доступний), 13 % – у стані S_3 (недоступні обидва стенди);

2) дані другої частини задачі представимо у вигляді таблиці 6.1

Таблиця 6.1 – Дані задачі

Стенди	Прибуток	Витрати
Стенд для перевірки ходової частини	10	4
Стенд для перевірки кутів розвал-сходження	6	2

Дані про граничні ймовірності нанесемо на рисунок 6.2.

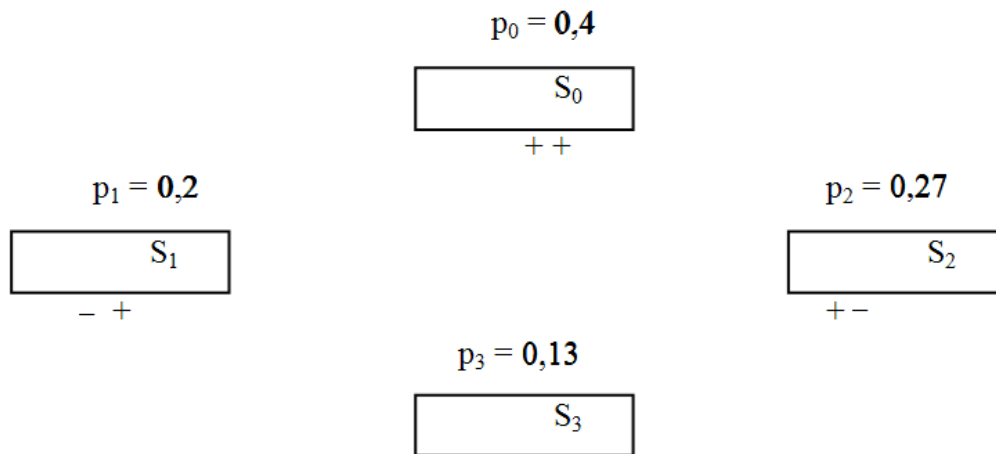


Рисунок 6.2 – Стан системи з граничними ймовірностями

Прибуток від роботи стану перевірки ходової частини:

$$(p_0 + p_2) \cdot 10 = (0,4 + 0,27) \cdot 10 = 0,67 \cdot 10 = 6,7.$$

Прибуток від роботи стану перевірки кутів розвал-сходження:

$$(p_0 + p_1) \cdot 6 = (0,4 + 0,2) \cdot 6 = 0,6 \cdot 6 = 3,6.$$

Прибуток від роботи стану перевірки ходової частини й стану перевірки кутів розвал-сходження:

$$6,7 + 3,6 = 10,3.$$

Витрати від поломки стану перевірки ходової частини:

$$(p_1 + p_3) \cdot 4 = (0,2 + 0,13) \cdot 4 = 0,33 \cdot 4 = 1,32.$$

Витрати від поломки стану перевірки кутів розвал-сходження:

$$(p_2 + p_3) \cdot 2 = (0,27 + 0,13) \cdot 2 = 0,4 \cdot 2 = 0,8.$$

Витрати від поломки стану перевірки ходової частини й стану перевірки кутів розвал-сходження:

$$1,32 + 0,8 = 2,12.$$

Відповідь:

загальний прибуток системи = прибуток – витрата = $10,3 - 2,12 = 8,18$ (грш.од)

3) крок 1. Розмічений граф задачі зображений на рисунку 6.3.

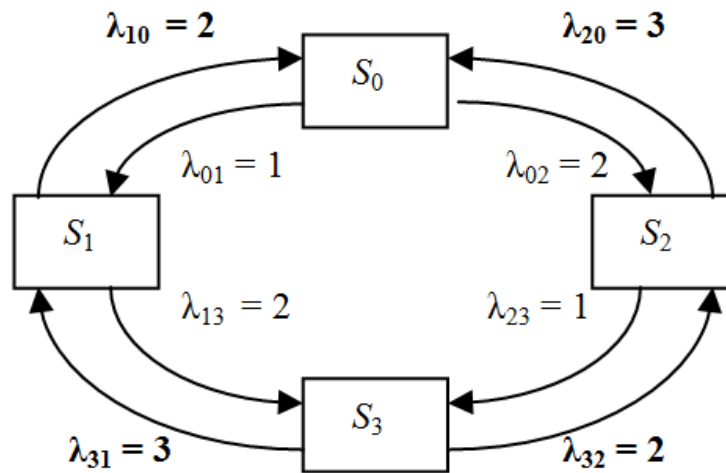


Рисунок 6.3 – Розмічений граф

Зменшення вдвічі середнього часу недоступності приводить до збільшення частоти повернення до роботи, тому інтенсивності зовнішнього кола розміченого графу збільшаться вдвічі, а інтенсивності внутрішнього кола не зміняться (рис. 6.4).

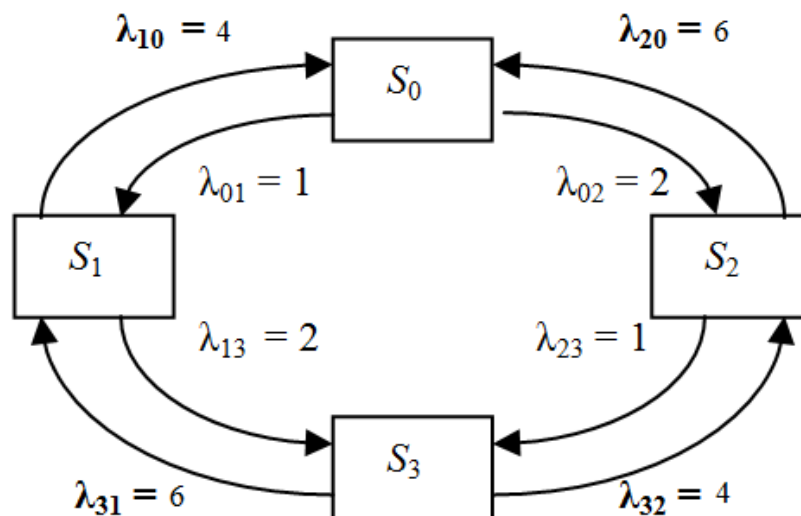


Рисунок 6.4 – Змінений розмічений граф

Враховуючи нові значення інтенсивності, отримаємо:

$$\begin{cases} (1+2)p_0 = 4p_1 + 6p_2, \\ (4+2)p_1 = 1p_0 + 6p_3, \\ (6+1)p_2 = 2p_0 + 4p_3, \\ p_0 + p_1 + p_2 + p_3 = 1. \end{cases} \Rightarrow \begin{cases} 3p_0 = 4p_1 + 6p_2, \\ 6p_1 = 1p_0 + 6p_3, \\ 7p_2 = 2p_0 + 4p_3, \\ p_0 + p_1 + p_2 + p_3 = 1. \end{cases} \Rightarrow \begin{cases} p_0 = 0,6, \\ p_1 = 0,15, \\ p_2 = 0,2, \\ p_3 = 0,05. \end{cases}$$

Крок 2. Збільшення витрат на недоступність удвічі.

Таблиця 6.2 – Змінені дані задачі

Стенди	Прибуток	Витрати
Стенд перевірки ходової частини	10	8
Стенд перевірки кутів розвал-сходження	6	4

Далі виконуємо такі ж дії, як у пункті 2 даної задачі.

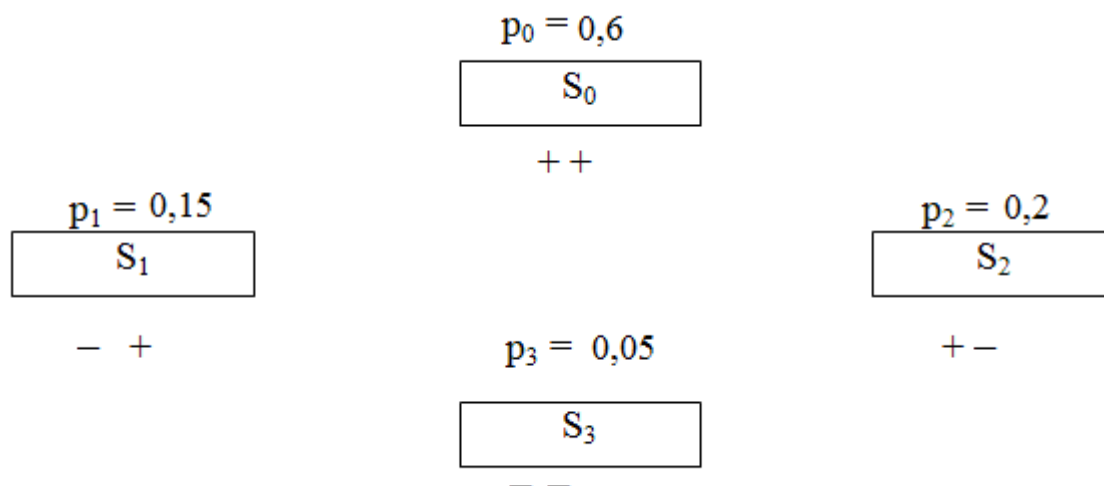


Рисунок 6.5 – Стан системи з граничними ймовірностями

Прибуток від роботи стенду перевірки ходової частини:

$$(p_0 + p_2) \cdot 10 = (0,6 + 0,2) \cdot 10 = 0,8 \cdot 10 = 8.$$

Прибуток від роботи стенду перевірки кутів розвал-сходження:

$$(p_0 + p_1) \cdot 6 = (0,6 + 0,15) \cdot 6 = 0,75 \cdot 6 = 4,5.$$

Витрати від поломки стенду перевірки ходової частини:

$$(p_1 + p_3) \cdot 8 = (0,15 + 0,05) \cdot 8 = 0,2 \cdot 8 = 1,6.$$

Витрати від поломки стенду перевірки кутів розвал-сходження:

$$(p_2 + p_3) \cdot 4 = (0,2 + 0,05) \cdot 4 = 0,25 \cdot 4 = 1.$$

$$\begin{aligned} \text{Загальний прибуток системи} &= \text{прибуток} - \text{витрата} = \\ &= 12,5 - 2,6 = 9,9 (\text{грош. од}). \end{aligned}$$

Відповідь: оскільки прибуток у третій частині задачі більший за прибуток у другій частині, можна стверджувати, що зменшення вдвічі середнього часу недоступності кожного стенду з одночасним збільшенням витрат на їх недоступність є економічно вигідним.

6.2.2 Індивідуальні завдання. Марківські процеси

1. Побудуйте граф станів і знайдіть граничні ймовірності такого процесу: пристрій S складається з двох вузлів, кожний з яких у випадковий момент часу може вийти з ладу, після чого миттєво

починається ремонт вузла, який триває заздалегіть невідомий випадковий проміжок часу. Інтенсивності наведено в таблиці 6.3.

Таблиця 6.3 – Інтенсивності подій

№ варіанта	λ_{10}	λ_{01}	λ_{20}	λ_{02}	λ_{13}	λ_{31}	λ_{32}	λ_{23}
1	1	2	3	1	1	3	4	5
2	2	3	4	2	2	7	8	9
3	1	2	3	1	2	3	4	5
4	9	8	7	6	5	4	3	2
5	8	5	2	7	4	1	9	6
6	3	5	7	9	5	1	4	6
7	2	3	4	3	2	1	4	5
8	1	2	1	3	4	5	2	2
9	2	2	3	3	4	4	2	1
10	7	7	5	5	9	6	2	1
11	6	6	2	1	2	3	4	5
12	4	5	6	4	1	3	2	7
13	2	4	3	1	6	5	1	2
14	2	3	4	1	5	6	2	3
15	3	5	4	6	2	1	1	3
16	2	1	3	5	6	4	2	3
17	8	2	3	4	5	6	2	1
18	9	6	3	8	5	2	7	4
19	1	3	8	4	5	3	7	6
20	2	5	9	1	4	8	7	6
21	3	5	9	1	6	8	4	2
22	7	4	2	8	6	1	9	3
23	4	6	2	8	7	3	9	1
24	1	6	7	9	4	3	8	2
25	1	3	4	5	9	7	6	8
26	8	2	6	4	7	5	3	1
27	9	1	6	8	4	2	3	7
28	9	6	4	8	5	1	7	6
29	1	2	4	5	1	3	2	5
30	2	4	1	1	5	5	3	2

2. Знайдіть середній чистий прибуток від експлуатації в стаціонарному режимі системи S в умовах попередньої задачі, якщо відомі витрати на ремонт кожного вузла й прибуток від їх експлуатації (табл. 6.4).

Таблиця 6.4 – Прибуток і витрати

Варіант 1			Варіант 11			Варіант 21		
Вузол	Приб.	Витр.	Вузол	Приб.	Витр.	Вузол	Приб.	Витр.
1	10	4	1	3	2	1	10	7
2	5	1	2	2	1	2	9	4
Варіант 2			Варіант 12			Варіант 22		
Вузол	Приб.	Витр.	Вузол	Приб.	Витр.	Вузол	Приб.	Витр.
1	9	5	1	5	3	1	4	3
2	4	3	2	2	1	2	7	5
Варіант 3			Варіант 13			Варіант 23		
Вузол	Приб.	Витр.	Вузол	Приб.	Витр.	Вузол	Приб.	Витр.
1	8	3	1	4	3	1	5	2
2	5	2	2	3	2	2	7	4
Варіант 4			Варіант 14			Варіант 24		
Вузол	Приб.	Витр.	Вузол	Приб.	Витр.	Вузол	Приб.	Витр.
1	9	7	1	7	6	1	8	7
2	6	4	2	5	3	2	5	2
Варіант 5			Варіант 15			Варіант 25		
Вузол	Приб.	Витр.	Вузол	Приб.	Витр.	Вузол	Приб.	Витр.
1	6	2	1	7	5	1	5	4
2	9	8	2	4	2	2	8	3
Варіант 6			Варіант 16			Варіант 26		
Вузол	Приб.	Витр.	Вузол	Приб.	Витр.	Вузол	Приб.	Витр.
1	7	5	1	4	2	1	5	3
2	8	4	2	7	3	2	6	4
Варіант 7			Варіант 17			Варіант 27		
Вузол	Приб.	Витр.	Вузол	Приб.	Витр.	Вузол	Приб.	Витр.
1	8	7	1	5	3	1	10	7
2	7	6	2	10	7	2	9	3

Продовження таблиці 6.4

Варіант 8			Варіант 18			Варіант 28		
Вузол	Приб.	Витр.	Вузол	Приб.	Витр.	Вузол	Приб.	Витр.
1	5	3	1	7	5	1	7	6
2	4	1	2	10	4	2	3	2
Варіант 9			Варіант 19			Варіант 29		
Вузол	Приб.	Витр.	Вузол	Приб.	Витр.	Вузол	Приб.	Витр.
1	9	2	1	10	5	1	3	1
2	10	3	2	3	2	2	7	5
Варіант 10			Варіант 20			Варіант 30		
Вузол	Приб.	Витр.	Вузол	Приб.	Витр.	Вузол	Приб.	Витр.
1	8	4	1	9	5	1	10	6
2	5	3	2	6	4	2	7	5

3. Оцініть економічну ефективність наявної можливості зменшення в k разів середнього часу ремонту кожного з вузлів, якщо при цьому доведеться збільшити в t разів витрати на ремонт кожного вузла (в одиницю часу). Значення k і t наведено в таблиці 6.5.

Таблиця 6.5 – Значення k і t

№ варіанта	Зменшення середнього часу ремонту кожного з вузлів, k	Збільшення витрат на ремонт кожного з вузлів, t
1	2	4
2	3	5
3	4	6
4	5	7
5	6	8
6	7	9
7	8	6
8	9	8
9	8	7
10	7	5
11	6	4
12	5	3

Продовження таблиці 6.5

13	4	2
14	3	1
15	2	3
16	1	2
17	2	4
18	3	5
19	4	6
20	5	7
21	6	8
22	7	9
23	8	2
24	9	3
25	8	4
26	7	5
27	6	3
28	5	7
29	4	8
30	3	9

РОЗДІЛ 7 СИСТЕМИ МАСОВОГО ОБСЛУГОВУВАННЯ

7.1 Загальні відомості про системи масового обслуговування

При дослідженні операцій часто доводиться мати справу з системами, що призначені для багаторазового використання при розв'язанні однотипних задач. Процеси, що виникають при цьому, отримали назву *процесів обслуговування*, а системи – *систем масового обслуговування (СМО)*. Прикладами СМО є телефонні системи, ремонтні майстерні, магазини, перукарні тощо.

Кожна СМО складається з певної кількості обслуговуючих одиниць (приладів, пристроїв, пунктів, станцій), які будемо називати *каналами обслуговування*. Каналами можуть бути лінії зв'язку, робочі точки, обчислювальні машини, продавці тощо. За кількістю каналів СМО поділяють на *одноканальні й багатоканальні*.

Зазвичай заявки надходять у СМО не регулярно, а випадково, утворюючи так званий *випадковий потік заявок (вимог)*. Обслуговування заявок триває якийсь випадковий проміжок часу. Випадковий характер потоку заявок і часу обслуговування призводить до того, що СМО виявляється завантаженою нерівномірно: в якісь періоди часу накопичується дуже велика кількість заявок (вони або стають у чергу, або так і не задовольняються), в інші ж періоди СМО працює з недовантаженням або простоює.

Завданням теорії масового обслуговування є побудова математичних моделей, що пов'язують задані умови роботи СМО (кількість каналів, їхню продуктивність, характер потоку заявок тощо) з показниками ефективності СМО, які описують її здатність справлятися з потоком заявок.

В якості *показника ефективності СМО* використовуються: середня кількість заявок (тут і надалі середні величини розуміються як математичні сподівання відповідних випадкових величин), що обслуговуються в одиницю часу; середня кількість заявок у черзі; середній час очікування на обслуговування; ймовірність відмови в обслуговуванні без очікування; ймовірність того, що кількість заявок у черзі перевищить певне значення тощо.

СМО поділяють на два основних типи (класи): СМО з *відмовами й СМО з очікуванням (чергою)*. У СМО з відмовами заявка, що надійшла в момент, коли всі канали зайняті, отримує відмову, залишає СМО й надалі в процесі обслуговування не бере участі (наприклад, заявка на телефонну розмову в момент, коли всі канали зайняті). У СМО з очікуванням заявка,

що прийшла в момент, коли всі канали зайняті, стає в чергу на обслуговування.

СМО з очікуванням поділяються на різні види в залежності від того, як організована черга: з обмеженою чи необмеженою довжиною черги, з обмеженим чи необмеженим часом очікування тощо.

Для класифікації СМО важливе значення має *дисципліна обслуговування*, що визначає порядок вибору заявок із тих, що надійшли, і порядок розподілу їх між вільними каналами. За цією ознакою обслуговування заявки може бути організоване за принципом «першою прийшла – першою обслуговується», «останньою прийшла – першою обслуговується» (такий порядок може застосовуватися, наприклад, при технічному обслуговуванні виробів зі складу, тому що ті вироби, які потрапили на склад останніми, часто виявляються доступнішими) або *обслуговування з пріоритетом* (коли в першу чергу обслуговуються найбільш важливі заявки). Пріоритет може бути як *абсолютним*, коли більш важлива заявка «витісняє» звичайну заявку (наприклад, у випадку аварійної ситуації планові роботи ремонтних бригад припиняються до моменту ліквідації аварії), так і *відносним*, коли більш важлива заявка отримує лише «краще» місце в черзі.

7.2 Процес загибелі й розмноження

7.2.1 Теоретичні відомості

У теорії масового обслуговування розглядається особливий клас випадкових процесів – процеси загибелі й розмноження. Назва пояснюється тим, що подібні моделі часто зустрічаються в процесі математичної інтерпретації біологічних задач про чисельність популяцій.

Граф станів процесу загибелі й розмноження зображено на рис. 7.1.

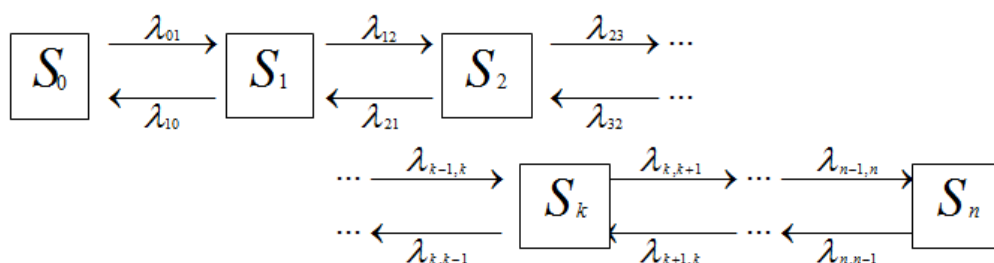


Рисунок 7.1 – Граф станів процесу загибелі й розмноження

На графі стани системи позначені так:

$$S_0, S_1, S_2, \dots, S_{k-1}, S_k, S_{k+1}, \dots, S_n.$$

Переходи з будь-якого стану можуть здійснюватися тільки в «сусідній» стан, тобто зі стану S_k можливий перехід у стан S_{k-1} або S_{k+1} .

Нехай усі потоки подій, що переводять систему зі стану в стан, є найпростішими з інтенсивностями $\lambda_{k,k+1}$ (якщо перехід відбувається зі стану k у стан $(k+1)$) і $\lambda_{k+1,k}$ (якщо перехід відбувається зі стану $(k+1)$ у стан k).

За графом, що представлений на рисунку 7.1, складемо й розв'яжемо систему алгебраїчних рівнянь для граничних імовірностей станів.

Згідно з правилом складання таких рівнянь, отримаємо для стану S_0 :

$$\lambda_{01}p_0 = \lambda_{10}p_1. \quad (7.1)$$

Для стану S_1 :

$$(\lambda_{12} + \lambda_{10})p_1 = \lambda_{01}p_0 + \lambda_{21}p_2. \quad (7.2)$$

З урахуванням (7.1) (7.2) приводиться до вигляду:

$$\lambda_{12}p_1 = \lambda_{21}p_2.$$

Записавши рівняння для граничних імовірностей інших станів, отримаємо систему рівнянь:

$$\begin{cases} \lambda_{01}p_0 = \lambda_{10}p_1, \\ \lambda_{12}p_1 = \lambda_{21}p_2, \\ \dots\dots\dots, \\ \lambda_{k-1,k}p_{k-1} = \lambda_{k,k-1}p_k, \\ \dots\dots\dots, \\ \lambda_{n-1,n}p_{n-1} = \lambda_{n,n-1}p_n. \end{cases} \quad (7.3)$$

До системи (7.3) додається умова:

$$p_0 + p_1 + p_2 + \dots + p_n = 1. \quad (7.4)$$

Розв'язуючи систему (7.3) та (7.4), отримаємо:

$$p_0 = \left(1 + \frac{\lambda_{01}}{\lambda_{10}} + \frac{\lambda_{12}\lambda_{01}}{\lambda_{21}\lambda_{10}} + \dots + \frac{\lambda_{n-1,n}\dots\lambda_{12}\lambda_{01}}{\lambda_{n,n-1}\dots\lambda_{21}\lambda_{10}} \right)^{-1}, \quad (7.5)$$

$$p_1 = \frac{\lambda_{01}}{\lambda_{10}} p_0, p_2 = \frac{\lambda_{12}\lambda_{01}}{\lambda_{21}\lambda_{10}} p_0, \dots, p_n = \frac{\lambda_{n-1,n}\dots\lambda_{12}\lambda_{01}}{\lambda_{n,n-1}\dots\lambda_{21}\lambda_{10}} p_0. \quad (7.6)$$

Легко помітити, що в формулах (7.6) коефіцієнти при p_0 є доданками з формули (7.5). Чисельники зазначених коефіцієнтів є добутком усіх інтенсивностей, які стоять біля стрілок, що ведуть зліва направо до даного стану, а знаменники – добутком усіх інтенсивностей, які стоять біля стрілок, що ведуть справа наліво до даного стану.

7.2.2 Приклад дослідження СМО (процес загибелі й розмноження)

Постановка задачі: проходження планового державного технічного огляду передбачає загальний огляд автомобіля, перевірку ходової частини, перевірку кутів розвал-сходження. Розмічений граф процесу проходження планового технічного огляду зображений на рисунку 7.2.

На рисунку 7.2 стан S_0 відповідає перебуванню автомобіля в місці, що відведене для загального огляду; стан S_1 – для перевірки ходової частини; стан S_2 – для перевірки кутів розвалу-сходження.

Необхідно знайти граничні ймовірності перебування автомобіля в кожному з пунктів.

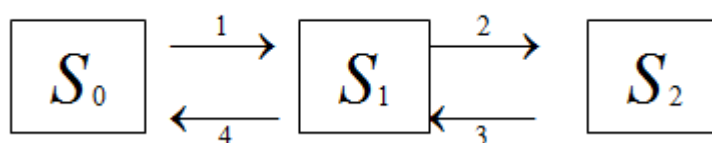


Рисунок 7.2 – Розмічений граф процесу проходження планового технічного огляду

Розв'язання.

За формулою (7.5) знайдемо p_0 :

$$p_0 = \left(1 + \frac{1}{4} + \frac{2 \cdot 1}{3 \cdot 4} \right)^{-1} \approx 0,706.$$

Використовуючи формули (7.6), знайдемо p_1 і p_2 :

$$p_1 \approx \frac{1}{4} \cdot 0,706 \approx 0,177;$$

$$p_2 \approx \frac{2 \cdot 1}{3 \cdot 4} \cdot 0,706 \approx 0,118.$$

Тобто, автомобіль, що прибуває для планового державного технічного огляду, у середньому 70,6% часу буде знаходитися на загальному огляді (обслуговуватися чи очікувати в черзі); 17,7% часу – у місці перевірки ходової частини (обслуговуватися чи очікувати в черзі); 11,8% часу – у місці перевірки кутів розвалу-сходження (обслуговуватися чи очікувати в черзі).

7.2.3 Індивідуальні завдання. Процес загибелі і розмноження

Граф станів процесу загибелі й розмноження наведено на

рисунку 7.3.

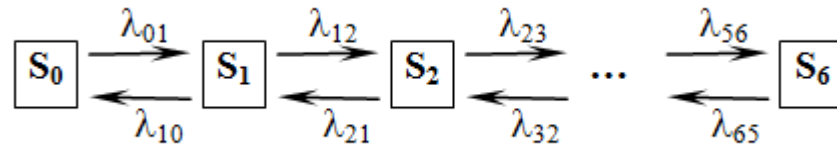


Рисунок 7.3 – Граф станів

Необхідно знайти граничні ймовірності станів S_2 і S_6 . Значення інтенсивностей наведено в таблиці 7.1.

Таблиця 7.1. – Значення інтенсивностей

№ варіанта								№ варіанта							
1	$\lambda_{k,k+1}$	1	3	2	4	4	2	16	$\lambda_{k,k+1}$	1	3	1	1	2	2
	$\lambda_{k+1,k}$	3	1	4	2	5	2		$\lambda_{k+1,k}$	3	2	2	2	3	2
2	$\lambda_{k,k+1}$	1	1	2	4	5	2	17	$\lambda_{k,k+1}$	1	1	2	4	5	2
	$\lambda_{k+1,k}$	3	2	1	2	1	4		$\lambda_{k+1,k}$	3	2	1	2	1	4
3	$\lambda_{k,k+1}$	2	3	2	1	1	4	18	$\lambda_{k,k+1}$	2	2	3	1	3	2
	$\lambda_{k+1,k}$	3	1	4	2	5	2		$\lambda_{k+1,k}$	1	1	1	4	5	2
4	$\lambda_{k,k+1}$	2	3	2	3	3	4	19	$\lambda_{k,k+1}$	1	3	2	1	3	4
	$\lambda_{k+1,k}$	3	4	1	1	5	2		$\lambda_{k+1,k}$	4	4	1	2	5	2
5	$\lambda_{k,k+1}$	3	3	2	2	1	1	20	$\lambda_{k,k+1}$	3	4	2	1	1	1
	$\lambda_{k+1,k}$	3	1	4	2	5	2		$\lambda_{k+1,k}$	1	1	2	2	4	3
6	$\lambda_{k,k+1}$	4	3	2	1	4	1	21	$\lambda_{k,k+1}$	4	4	2	2	5	1
	$\lambda_{k+1,k}$	3	1	4	3	5	2		$\lambda_{k+1,k}$	2	2	4	3	3	2

Продовження таблиці 7.1

7	$\lambda_{k,k+1}$	2	2	1	1	4	3	22	$\lambda_{k,k+1}$	2	2	4	4	4	3
	$\lambda_{k+1,k}$	5	1	4	1	5	2		$\lambda_{k+1,k}$	4	1	3	3	1	2
8	$\lambda_{k,k+1}$	4	3	2	5	4	2	23	$\lambda_{k,k+1}$	1	1	3	4	3	2
	$\lambda_{k+1,k}$	3	3	4	2	5	2		$\lambda_{k+1,k}$	2	3	2	3	3	2
9	$\lambda_{k,k+1}$	1	2	2	5	3	3	24	$\lambda_{k,k+1}$	2	2	2	3	3	2
	$\lambda_{k+1,k}$	3	1	1	2	2	3		$\lambda_{k+1,k}$	1	1	1	2	2	3
10	$\lambda_{k,k+1}$	1	1	1	4	4	2	25	$\lambda_{k,k+1}$	2	2	2	3	3	3
	$\lambda_{k+1,k}$	2	2	2	3	3	3		$\lambda_{k+1,k}$	1	1	1	4	4	2
11	$\lambda_{k,k+1}$	3	2	1	3	2	1	26	$\lambda_{k,k+1}$	1	3	1	4	1	2
	$\lambda_{k+1,k}$	1	1	2	3	5	4		$\lambda_{k+1,k}$	4	1	3	2	2	3
12	$\lambda_{k,k+1}$	3	3	3	4	4	2	27	$\lambda_{k,k+1}$	2	4	2	4	1	2
	$\lambda_{k+1,k}$	3	1	4	2	2	2		$\lambda_{k+1,k}$	3	2	4	1	3	2
13	$\lambda_{k,k+1}$	2	1	2	1	3	3	28	$\lambda_{k,k+1}$	1	1	2	2	3	3
	$\lambda_{k+1,k}$	2	3	4	1	3	2		$\lambda_{k+1,k}$	3	2	1	1	2	2
14	$\lambda_{k,k+1}$	1	2	3	1	4	2	29	$\lambda_{k,k+1}$	3	2	1	2	3	2
	$\lambda_{k+1,k}$	3	3	4	4	5	5		$\lambda_{k+1,k}$	1	2	4	4	2	3
15	$\lambda_{k,k+1}$	3	3	3	1	1	2	30	$\lambda_{k,k+1}$	1	1	2	2	4	4
	$\lambda_{k+1,k}$	1	1	4	3	3	2		$\lambda_{k+1,k}$	3	3	3	2	2	2

7.3 СМО з відмовами

Нагадаємо, що в СМО з відмовами заявка, яка надійшла в момент, коли всі канали були зайняті, отримавши відмову, залишає СМО і в подальшому процесі обслуговування не бере участі.

В якості показників ефективності СМО з відмовами будемо розглядати:

A – абсолютну пропускну здатність СМО (середнє число заявок, що обслуговуються в одиницю часу);

Q – відносну пропускну здатність (відношення середньої кількості заявок, що обслуговуються системою за одиницю часу, до середньої кількості заявок, що надходить за цей час);

$P_{\text{відм}}$ – імовірність відмови, тобто того, що заявка отримає відмову;

k – середня кількість зайнятих каналів (для багатоканальних систем).

7.3.1 Одноканальна система з відмовами

Розглянемо таку задачу: є один канал, на який надходить потік заявок з інтенсивністю λ . Потік обслуговування має інтенсивність μ . Необхідно знайти граничні ймовірності станів системи й показники її ефективності.

Система S (СМО) має два стани: S_0 – канал вільний, S_1 – канал зайнятий. Розмічений граф станів представлений на рисунку 7.4.

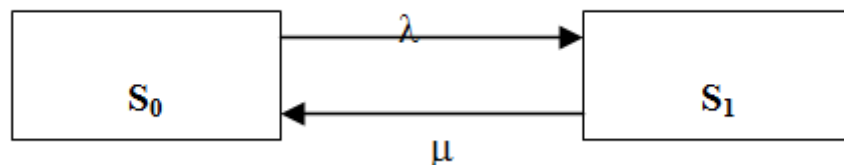


Рисунок 7.4 – Розмічений граф станів

У граничному стаціонарному режимі система алгебраїчних рівнянь для ймовірностей станів має вигляд:

$$\begin{cases} \lambda p_0 = \mu p_1, \\ \mu p_1 = \lambda p_0. \end{cases} \quad (7.7)$$

Отже, система виродилася в одне рівняння. Враховуючи умову

$$p_0 + p_1 = 1,$$

знайдемо з (7.7) граничні ймовірності станів:

$$p_0 = \frac{\mu}{\lambda + \mu}, p_1 = \frac{\lambda}{\lambda + \mu}, \quad (7.8)$$

які виражають середній відносний час перебування системи в стані S_0 (коли канал вільний) і S_1 (коли канал зайнятий), тобто визначають відповідно відносну пропускну здатність Q системи й ймовірність відмови $P_{відм}$:

$$Q = p_0 = \frac{\mu}{\lambda + \mu}, \quad (7.9)$$

$$P_{відм} = p_1 = \frac{\lambda}{\lambda + \mu}. \quad (7.10)$$

Абсолютну пропускну здатність знайдемо, помноживши відносну пропускну здатність Q на інтенсивність потоку відмов:

$$Q \cdot \lambda = A = \frac{\lambda \mu}{\lambda + \mu}. \quad (7.11)$$

Приклад одноканальної СМО з відмовами: автосалон складається з декількох приміщень, в кожному з яких представлені автомобілі певної марки. Якщо клієнт бажає провести тест-драйв, продавець-консультант телефонує до кабінету співробітника, який має знаходитися в автомобілі під час пробної поїздки. Якщо такий співробітник зайнятий на тест-драйві автомобіля іншої марки, телефон не відповідає, тобто на даний час проведення тест-драйву стає неможливим. За зміну до співробітника, який проводить тест-драйв, надходить у середньому 8 заявок, а провести він може 3 тест драйви. Необхідно знайти граничні ймовірності станів наведеної СМО й знайти показники ефективності.

Для даної задачі:

S_0 – співробітник може провести тест-драйв;

S_1 – співробітник зайнятий і не може провести тест-драйв;

$\lambda = 8$;

$\mu = 3$.

Граничні ймовірності знайдемо за формулами (7.8):

$$p_0 = \frac{3}{3+8} \approx 0,273, p_1 = \frac{8}{8+3} \approx 0,727.$$

Отже, приблизно 27,3 % часу співробітник готовий негайно провести тест-драйв, 72,7 % часу — не готовий.

Відносну пропускну здатність Q знайдемо за формулою (7.9)

$$Q = \frac{3}{8+3} \approx 0,273.$$

Ймовірність відмови знайдемо за формулою (7.10)

$$P_{відм} = \frac{8}{8+3} \approx 0,727.$$

Очевидно, що за наявності лише одного спеціаліста, який проводить тест-драйв, автосалон буде погано справлятися з потоком заявок на цю послугу.

Абсолютну пропускну здатність знайдемо за формулою (7.11)

$$A = \frac{8 \cdot 3}{8+3} \approx 2,182.$$

7.3.2 Багатоканальна система з відмовами.

Розглянемо таку задачу: система S (СМО) має стани $S_0, S_1, S_2, \dots, S_n$, де S_k – стан системи, коли в ній знаходиться k заявок, тобто зайнято k каналів. Розмічений граф станів системи (рис. 7.5) відповідає процесу загибелі й розмноження.

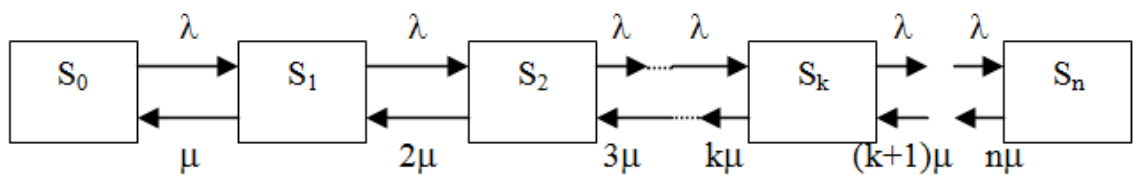


Рисунок 7.5 – Розмічений граф станів

Потік заявок послідовно переводить систему з будь-якого лівого стану в сусідній правий з однією й тією ж інтенсивністю λ . Інтенсивність потоку обслуговування, що переводить систему з будь-якого правого стану в сусідній лівий стан, постійно змінюється. Дійсно, якщо СМО знаходиться в стані S_2 (два канали зайняті), то вона може перейти в стан S_1 (один канал зайнятий), коли закінчить обслуговування або перший, або другий канал, тобто сумарна інтенсивність їхніх потоків обслуговування буде 2μ . Аналогічно сумарний потік обслуговування, що переводить СМО зі стану S_3 (три канали зайняті) у S_2 , буде мати інтенсивність 3μ , тобто може звільнитися кожний з трьох каналів тощо.

Дослідження описаної вище СМО має назву задачі Ерланга (на честь засновника теорії СМО – датського вченого А. К. Ерланга (1878 – 1929)).

Використовуючи формулу (7.5), знайдемо p_0 :

$$p_0 = \left(1 + \frac{\lambda}{\mu} + \frac{\lambda^2}{2!\mu^2} + \dots + \frac{\lambda^k}{k!\mu^k} + \dots + \frac{\lambda^n}{n!\mu^n} \right)^{-1}, \quad (7.12)$$

де члени $\frac{\lambda}{\mu}, \frac{\lambda^2}{2!\mu^2}, \dots, \frac{\lambda^n}{n!\mu^n}$ – коефіцієнти при p_0 у виразах для граничних ймовірностей p_1, p_2, \dots, p_n .

Величина

$$\rho = \frac{\lambda}{\mu} \quad (7.13)$$

називається *приведеною інтенсивністю потоку заявок* або *інтенсивністю навантаження каналу*. Вона виражає середню кількість заявок, що приходить за середній час обслуговування однієї заявки. Підставивши ρ у (7.12), отримаємо:

$$p_0 = \left(1 + \rho + \frac{\rho^2}{2!} + \dots + \frac{\rho^k}{k!} + \dots + \frac{\rho^n}{n!} \right)^{-1}, \quad (7.14)$$

$$p_1 = \rho p_0, p_2 = \frac{\rho^2}{2!} p_0, \dots, p_k = \frac{\rho^k}{k!} p_0, \dots, p_n = \frac{\rho^n}{n!} p_0. \quad (7.15)$$

Формули (7.14) і (7.15) для граничних ймовірностей отримали назву *формул Ерланга*.

Ймовірність відмови СМО є граничною ймовірністю того, що всі n каналів системи будуть зайняті, тобто

$$P_{\text{відм}} = \frac{\rho^n}{n!} p_0. \quad (7.16)$$

Відносна пропускна здатність:

$$Q = 1 - P_{\text{відм}} = 1 - \frac{\rho^n}{n!} p_0. \quad (7.17)$$

Абсолютна пропускна здатність:

$$A = \lambda Q = \lambda \left(1 - \frac{\rho^n}{n!} p_0 \right). \quad (7.18)$$

Приклад багатоканальної СМО з відмовами: для модернізації (встановлення нового обладнання) бензозаправної станції (рис. 7.6), що знаходиться на трасі Київ – Сімферополь, були отримані такі статистичні дані: у середньому на станцію протягом умовного інтервалу часу τ

(30 хвилин = $\frac{1}{2}$ години) під'їжджає 8 машин для заправки бензином АІ-95.

Від однієї колонки нового зразку протягом часу τ можуть заправитися 3 автомобіля. Необхідно визначити оптимальну кількість колонок нового зразку, що необхідно встановити. Критерієм оптимальності вважати виконання за допомогою бензоколонок нового зразку у середньому 95 заявок на заправку з кожних 100 заявок. Побудувати діаграму залежності показників ефективності від кількості встановлених бензозаправних колонок.

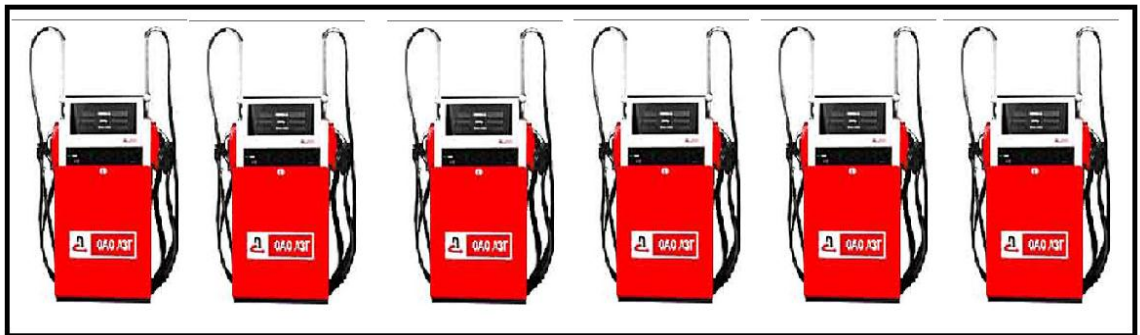


Рисунок 7.6 – Схема бензоколонки

Інтенсивність навантаження каналу визначимо за формулою (7.13), враховуючи, що в нашому випадку $\lambda = 8$; $\mu = 3$:

$$\rho = \frac{8}{3} \approx 2,667.$$

Будемо поступово збільшувати кількість бензозаправних колонок ($n = 2, 3, 4, \dots$).

Отримаємо граф станів (рис. 7.7):

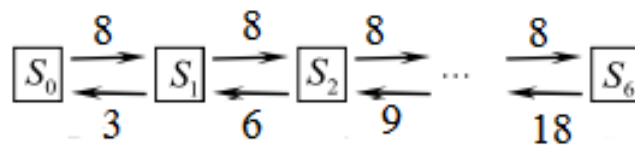


Рисунок 7.7 – Розмічений граф

На рисунку 7.7 стани описуються так:

S_0 – не встановлено жодної бензоколонки нового зразку;

S_1 – встановлена одна бензоколонка нового зразку;

S_2 – встановлені дві бензоколонки нового зразку;

S_3 – встановлені три бензоколонки нового зразку;

S_4 – встановлені чотири бензоколонки нового зразку;

S_5 – встановлені п'ять бензоколонок бензоколонки нового зразку;

S_6 – встановлені шість бензоколонок бензоколонки нового зразку.

Знайдемо граничні ймовірності СМО, якщо планується встановити тільки одну колонку (система у стані S_1), використовуючи формули (7.14) і (7.15)

$$p_0 = (1 + 2,667)^{-1} \approx 0,273;$$

$$p_1 = 2,667 \cdot 0,273 \approx 0,728.$$

Ймовірність відмови СМО знаходимо за формулою (7.16)

$$P_{відм} = \frac{\rho^1}{1!} p_0 = \frac{2,667}{1} \cdot 0,273 \approx 0,728.$$

Ймовірність того, що автомобіль зможе заправитися, знаходимо за формулою (7.17)

$$Q = 1 - 0,728 = 0,272.$$

Абсолютну пропускну здатність знаходимо за формулою (7.18)

$$A = 8 \cdot 0,272 = 2,176.$$

Аналогічно знайдемо граничні ймовірності й показники ефективності, якщо планується встановити дві, три, чотири, п'ять і шість колонок.

Для плану встановлення двох колонок (система у стані S_2):

$$p_0 = \left(1 + \rho + \frac{\rho^2}{2!} \right)^{-1} \approx 0,138;$$

$$p_1 \approx 0,368;$$

$$p_2 = \frac{\rho^2}{2!} \cdot p_0 \approx 0,49;$$

$$P_{відм} \approx 0,49;$$

$$Q = 0,51;$$

$$A \approx 4,08.$$

Для плану встановлення трьох колонок (система у стані S_3):

$$p_0 = \left(1 + \rho + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} \right)^{-1} \approx 0,096;$$

$$p_1 \approx 0,256;$$

$$p_2 = \frac{\rho^2}{2!} p_0 \approx 0,341;$$

$$p_3 = \frac{\rho^3}{3!} p_0 \approx 0,304;$$

$$P_{відм} \approx 0,304;$$

$$Q = 0,696;$$

$$A \approx 5,568.$$

Для плану встановлення чотирьох колонок (система у стані S_4):

$$p_0 = \left(1 + \rho + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} + \frac{\rho^4}{4!} \right)^{-1} \approx 0,08;$$

$$p_1 \approx 0,213;$$

$$p_2 = \frac{\rho^2}{2!} p_0 \approx 0,284;$$

$$p_3 = \frac{\rho^3}{3!} p_0 \approx 0,253;$$

$$p_4 = \frac{\rho^4}{4!} p_0 \approx 0,169;$$

$$P_{відм} \approx 0,169;$$

$$Q = 0,831;$$

$$A \approx 6,648.$$

Для плану встановлення п'яти колонок (система у стані S_5):

$$p_0 = \left(1 + \rho + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} + \frac{\rho^4}{4!} + \frac{\rho^5}{5!} \right)^{-1} \approx 0,073;$$

$$p_1 \approx 0,195;$$

$$p_2 = \frac{\rho^2}{2!} p_0 \approx 0,26;$$

$$p_3 = \frac{\rho^3}{3!} p_0 \approx 0,231;$$

$$p_4 = \frac{\rho^4}{4!} p_0 \approx 0,154;$$

$$p_5 = \frac{\rho^5}{5!} p_0 \approx 0,082;$$

$$P_{відм} \approx 0,082;$$

$$Q = 0,918;$$

$$A \approx 7,344.$$

Для плану встановлення шести колонок (система у стані S_6):

$$p_0 = \left(1 + \rho + \frac{\rho^2}{2!} + \frac{\rho^3}{3!} + \frac{\rho^4}{4!} + \frac{\rho^5}{5!} + \frac{\rho^6}{6!} \right)^{-1} \approx 0,071;$$

$$p_1 \approx 0,189;$$

$$p_2 = \frac{\rho^2}{2!} p_0 \approx 0,252;$$

$$p_3 = \frac{\rho^3}{3!} p_0 \approx 0,225;$$

$$p_4 = \frac{\rho^4}{4!} p_0 \approx 0,15;$$

$$p_5 = \frac{\rho^5}{5!} p_0 \approx 0,08;$$

$$p_6 = \frac{\rho^6}{6!} p_0 \approx 0,036;$$

$$P_{\text{відм}} \approx 0,036;$$

$$Q = 0,964;$$

$$A \approx 7,712.$$

Значення характеристик СМО запишемо у таблицю 7.2.

Таблиця 7.2 – Значення характеристик СМО

Показники ефективності	Кількість каналів (колонок)					
	1	2	3	4	5	6
$P_{\text{відм}}$	0. 728	0 .49	0. 304	0. 169	0. 082	0. 036
Q	0. 272	0 .51	0. 696	0. 831	0. 918	0. 964
A	2. 176	4 .08	5. 568	6. 648	7. 344	7. 712

За умовою оптимальності відносна пропускна здатність має бути не

меншою за $\frac{95}{100} \left(Q \geq \frac{95}{100} \right)$. Отже, необхідно спланувати встановлення не менш ніж 6 бензоколонок нового зразку.

Проведемо візуалізацію даних таблиці 7.2 (рис. 7.8).

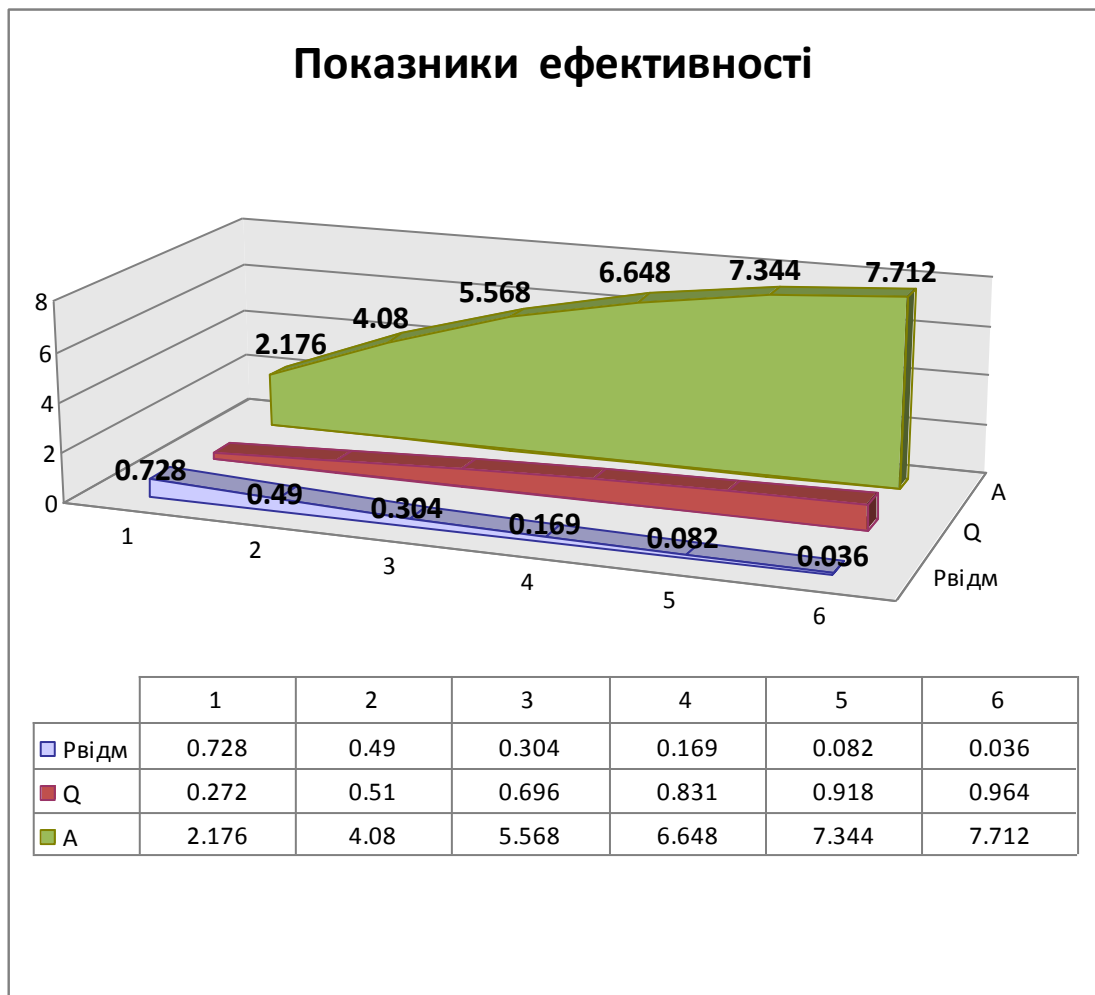


Рисунок 7.8 – Діаграма залежності показників ефективності від кількості встановлених бензоколонок

Висновки: виходячи з розрахунків (табл. 7.2) і візуалізації результатів (рис. 7.8), рекомендуємо запланувати встановлення шести бензоаправних колонок. При цьому середня кількість заявок, що обслуговуються в одиницю часу τ , наблизиться до восьми автомобілів із імовірністю відмови 0,036.

Оскільки кожний зайнятий канал обслуговує у середньому $\mu = 3$ заявок (в одиницю часу τ), то середня кількість зайнятих каналів (\bar{K}) дорівнює:

$$\bar{K} = \frac{A}{\mu} \approx \frac{7,712}{3} \approx 3.$$

7.3.3 Індивідуальні завдання. СМО з відмовами

Знайдіть граничні ймовірності й показники ефективності систем масового обслуговування, що мають розмічені графи, що зображені на рисунках 7.9 і 7.10. Значення інтенсивностей наведено в таблиці 7.3.

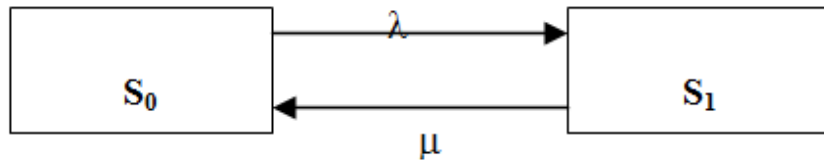


Рисунок 7.9 – Розмічений граф (1)

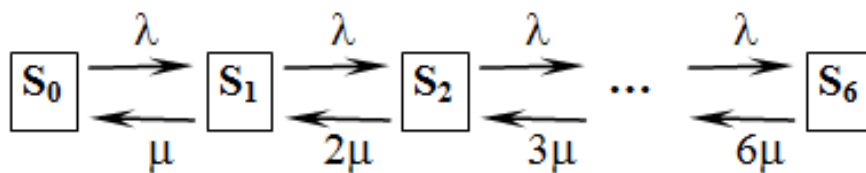


Рисунок 7.10 – Розмічений граф (2)

Таблиця 7.3 – Значення інтенсивностей подій

№ варіанта	λ	μ	№ варіанта	λ	μ	№ варіанта	λ	μ
1	2	7	11	6	5	21	8	9
2	3	6	12	5	4	22	9	8
3	4	5	13	4	3	23	8	7
4	5	4	14	3	2	24	7	4
5	6	3	15	2	3	25	6	3
6	7	2	16	3	4	26	5	2
7	8	3	17	4	9	27	4	7
8	9	4	18	5	6	28	3	8
9	8	5	19	6	7	29	2	9
10	7	6	20	7	8	30	3	5

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Абчук В. А. Экономико-математические методы: элементарная математика и логика. Методы исследования операций / В. А. Абчук. – СПб.: Союз, 1999. – 320 с.
2. Вентцель Е. С. Исследование операций. Задачи, принципы, методология: [учеб. пособие для студ. вузов] / Е. С. Вентцель. – М.: Высш. шк., 2001. – 208 с.
3. Исследование операций: [учеб. для вузов] / [И. К. Волков, Е. А. Загоруйко]; под ред. В. С. Зарубина, А. П. Крищенко. – М.: Изд-во МГТУ им. Н. Э. Баумана, 2000. – 436 с.
4. Исследование операций в экономике: [учебн. пособие для вузов] / [Н. Ш. Кремер, Б. А. Путко, И. М. Тришин, М. Н. Фридман]; под ред. Н. Ш. Кремера. – М.: Банки и биржи, ЮНИТИ, 1997. – 407 с.
5. Конюховский П. В. Математические методы исследования операций / П. В. Конюховский. – СПб: Питер, 2001. – 192 с.
6. Сигал И. Х. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы / И. Х. Сигал, А. П. Иванов. – М.: ФИЗМАТЛИТ, 2002. – 240 с.
7. Пинегина М. В. Математические методы и модели в экономике: [учебное пособие для студентов вузов экономических специальностей] / М. В. Пинегина. – М.: Издательство «Экзамен», 2004. – 128 с.
8. Хемди Таха А. Введение в исследование операций / А. Хемди Таха. – М.: Издательский дом «Вильямс», 2001. – 912 с.