Модуль 4
МЕРЕЖНІ МОДЕЛІ

4.1. Задачі мережної структури

У рамках теорії дослідження операцій розглядають велику кількість практичних задач, які можна сформулювати і розв'язати як мережні моделі. Наведемо кілька конкретних прикладів:

1. Проектування газопроводу, що з'єднує свердловини морського базування з приймальною станцією, розташованою на березі. Цільова функція відповідної моделі має мінімізувати вартість будівництва газопроводу.

2. Знаходження найкоротшого маршруту між двома містами по наявній мережі доріг.

3. Визначення оптимальної пропускної спроможності трубопроводу для транспортування вугільної пульпи від вугільних шахт до електростанцій.

4. Визначення максимальної пропускної спроможності нафтопроводу від пунктів нафтовидобутку до нафтопереробних заводів із мінімальною вартістю транспортування.

5. Складання тимчасового графіка будівельних робіт (визначення дати початку і завершення окремих етапів робіт).

Рішення приведених задач (як і багатьох інших подібних задач) вимагає застосування різних мережних оптимізаційних алгоритмів. У цьому розділі розглянемо такі алгоритми:

1. Алгоритм визначення мінімального остового дерева.
2. Алгоритм визначення найкоротшого шляху.
3. Оптимальний розподіл товарообігу по мережі.

Задачі, що випливають із перерахованих прикладів, можна сформулювати і розв'язувати як задачі лінійного програмування. Однак специфічна структура цих задач дозволяє розробити спеціальні мережні алгоритми ефективніше, ніж стандартний симплекс-метод.

4.2. Основні визначення мережних моделей

Мережа складається з множини **вузлів**, зв'язаних **дугами** (або **ребрами**). Таким чином, мережу можна описати двома множинами (N, A), де N — множина вузлів, A — множина ребер. Наприклад, мережу, показану на рис. 4.1, описуємо в такий спосіб:

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

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

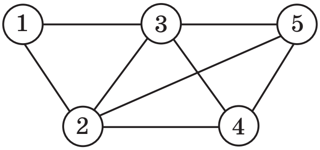


Рис. 4.1. Приклад мережі

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 спроможністю її ребер, що може бути як скінченною, так і нескінченною.

Ребро називається **спрямованим**, або **орієнтованим** (у цьому разі ребро будемо називати **дугою**), якщо в одному напрямку можливий тільки потік додатних значень, 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 кожнім ребрі. Шлях формує **цикл**, якщо початковий і кінцевий вузли збігаються. Наприклад, на рис. 4.1 ребра (2,3), (3,4) і (4,2) складають цикл.

Орієнтований цикл — це цикл, у якому всі дуги орієнтовані у визначеному напрямку.

Зв'язна мережа — така мережа, в якій будь-які два вузли зв'язані принаймні одним шляхом.

Деревом називається зв'язна мережа, що містить підмножину вузлів вихідної мережі і не має циклів. **Остове дерево** — це дерево, що містить *усі* вузли мережі. На рис. 4.2 показані дерево й остове дерево для мережі з рис. 4.1.

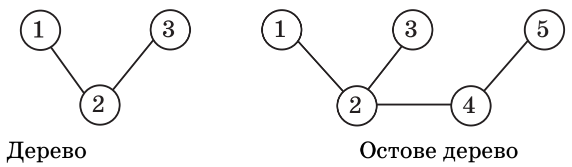


Рис. 4.2. Дерево + остове дерево

4.3. Побудова мінімального остового дерева

Алгоритм побудови мінімального остового дерева припускає з'єднання всіх вузлів мережі за допомогою шляхів найменшої довжини. Типовою задачею, для розв'язання якої потрібно такий алгоритм, є створення (проектування) мережі доріг із твердим покриттям, що з'єднують населені пункти сільської місцевості, де дороги, що з'єднують два яких-небудь пункти, можуть проходити через інші населені пункти. Найбільш економічний проект дорожньої системи має мінімізувати загальну довжину доріг із твердим покриттям, до того ж, бажаний результат можна одержати шляхом застосування алгоритму побудови мінімального остового дерева.

Опишемо процедуру виконання цього алгоритму. Позначимо через $N = \{1, 2, ..., n\}$ множину вузлів мережі і введемо нові позначення:

C_k — множина вузлів мережі, з'єднаних алгоритмом після виконання $\boldsymbol{k}$-ї ітерації цього алгоритму;

$\overline{C_k}$ — множина вузлів мережі, з'єднаних із вузлами безлічі C_k після виконання $\boldsymbol{k}$-ї ітерації цього алгоритму.

Крок 0. Припускаємо $C_0 = \varnothing$ і $\overline{C_0} = \boldsymbol{N}$.

Крок 1. Вибираємо *будь-який* вузол і з множини $\varnothing$ визначаємо $C_1 = \{i\}$, тоді $\overline{C_1} = N - \{i\}$. Припускаємо, $k = 2$.

Основний крок $\boldsymbol{k}$. У множини $\overline{C_{k-1}}$ вибираємо 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 $\overline{C_k}$ порожня, то виконання алгоритму припиняють. В іншому разі припускаємо $k = k + 1$ і повторюємо останній крок.

Приклад 4.1

Телевізійна компанія планує підключення до своєї кабельної мережі п'яти нових районів. На рис. 4.3 показано структуру планованої мережі і відстані (у милях) між районами і телецентром. Треба спланувати економічно найефективнішу кабельну мережу.

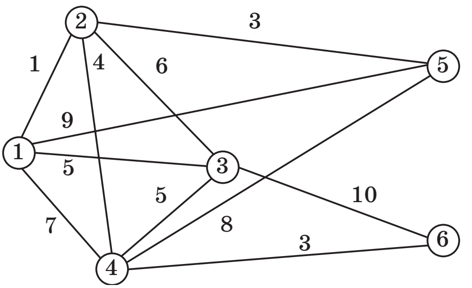


Рис. 4.3. Структура планованої мережі

Почнемо виконання алгоритму побудови мінімального остового дерева з вибору вузла 1 (або будь-якого іншого вузла). Тоді $C_1 = \{1\}$ і $\overline{C_1} = \{2, 3, 4, 5, 6\}$.

Послідовні ітерації виконання алгоритму показано на рис. 4.4. Тут тонкими лініями зображені ребра, які з'єднують вузли, що належать множинам C_k і $\overline{C_k}$, серед яких шукаємо ребро з мінімальною вартістю (довжиною). Це знайдене ребро позначене пунктирною лінією. Товстими суцільними лініями показані ребра, що з'єднують вузли множини C_k (і які раніше позначалися пунктирними лініями).

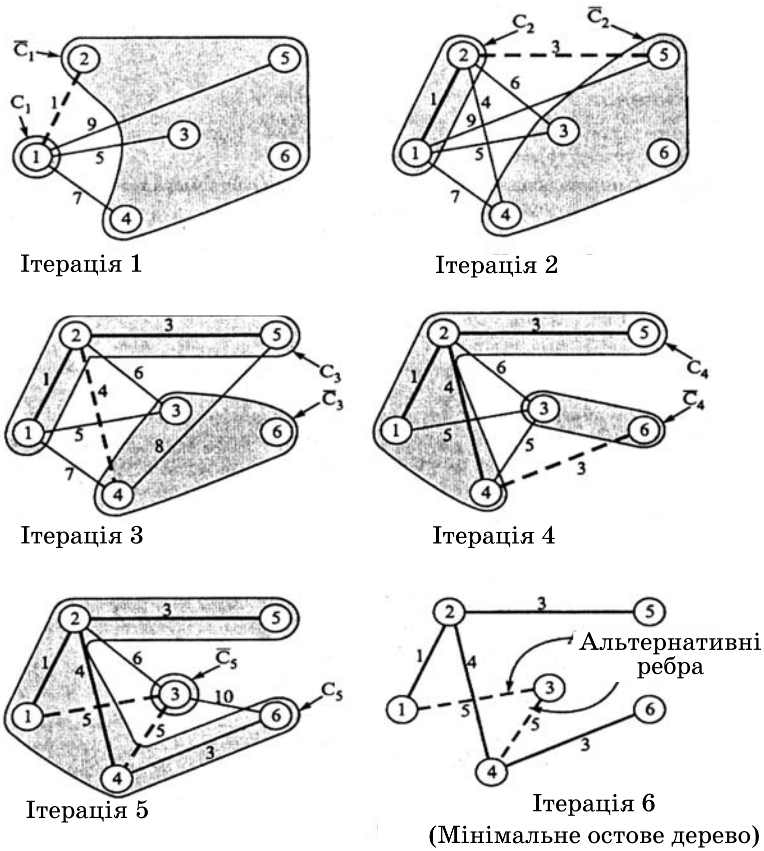


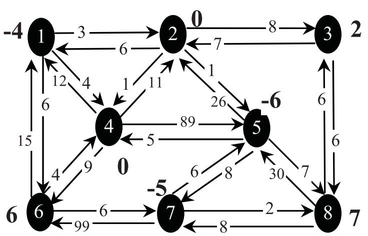
Рис. 4.4. Послідовність ітерацій виконання алгоритму

Наприклад, на першій ітерації ребро (1,2) має найменшу вартість (тобто найменша відстань між пунктами мережі) серед усіх інших ребер, що з'єднують вузол 1 із вузлами множини (зазначимо, що вузол 6 не має ребра, які безпосередньо з'єднували б його з вузлом 1). Тому $j = 2$ і $C_2 = \{1, 2\}$, а $\bar{C}_2 = \{3, 4, 5, 6\}$.

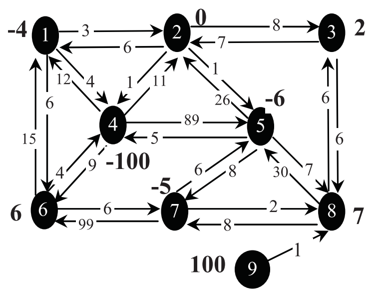
Розв'язок у формі мінімального остового дерева отримано на 6-й ітерації (рис. 4.4). *Мінімальна довжина кабелю для побудови такої мережі дорівнює* **1 + 3 + 4 + 3 + 5 = 16** *у. о.*

4.4. Індивідуальні завдання моделі «Побудова мінімального остового дерева»

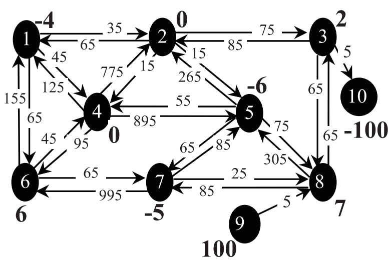
Варіант 1



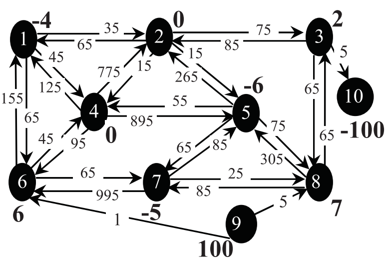
Варіант 2



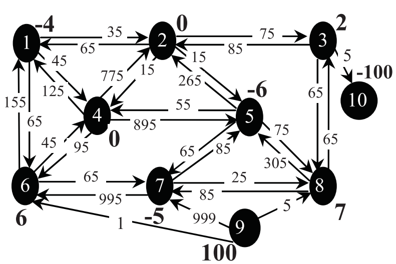
Варіант 3



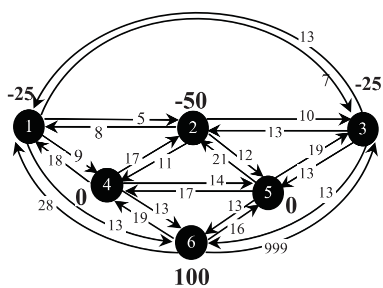
Варіант 4



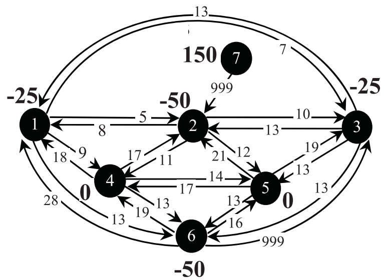
Варіант 5



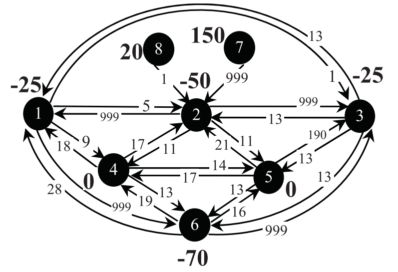
Варіант 6



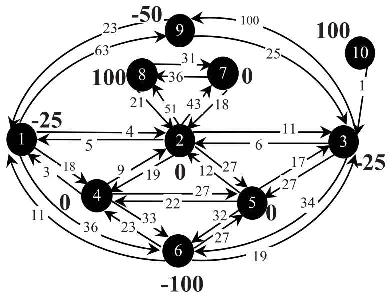
Варіант 7



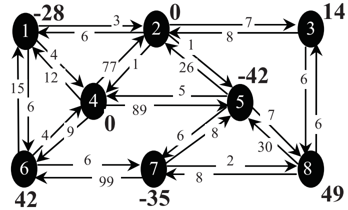
Варіант 8



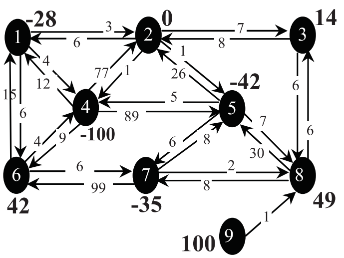
Варіант 9



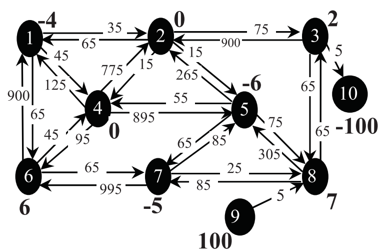
Варіант 10



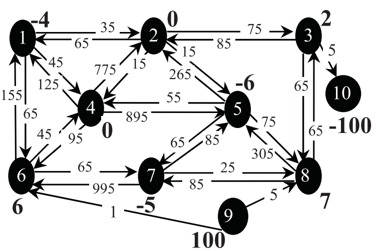
Варіант 11



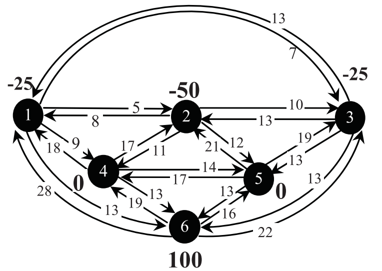
Варіант 12



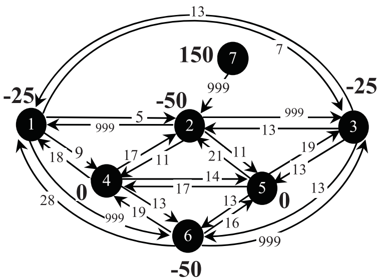
Варіант 13



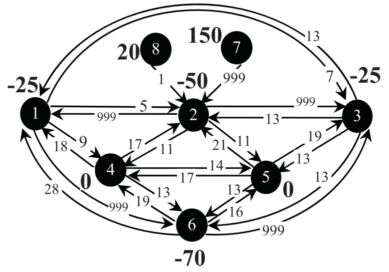
Варіант 14



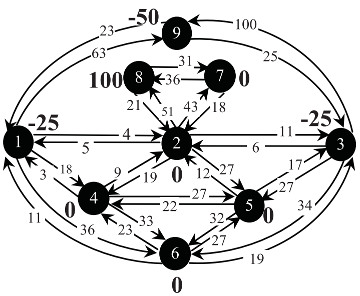
Варіант 15



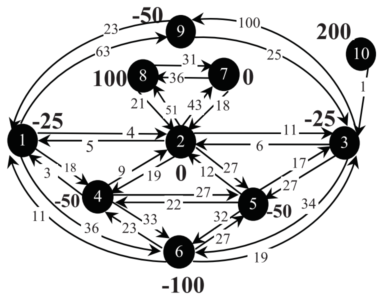
Варіант 16



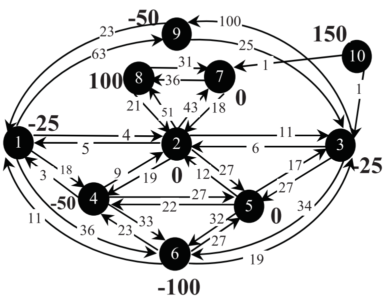
Варіант 17



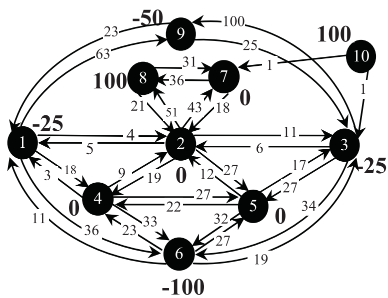
Варіант 18



Варіант 19



Варіант 20



4.5. Модель Флойда

Модель Флойда визначає найкоротші шляхи між будь-якими двома вузлами мережі. У цьому алгоритмі мережа подана у формі квадратної матриці з ***n*** рядками і ***n*** стовпчиками. Елемент *(i, j)* дорівнює відстані d_{ij} від вузла *i* до вузла *j*, що має кінцеве значення, якщо існує дуга *(i, j)*, або дорівнює нескінченності в разі її відсутності.

Покажемо спочатку основну ідею методу Флойда. Нехай є три вузли i, j і k і задано відстані між ними (рис. 4.5). Якщо виконується нерівність $d_{ij} + d_{jk} < d_{ik}$, то доцільно замінити шлях $i \to k$ шляхом $i \to j \to k$. Така заміна (далі її будемо умовно називати **трикутним оператором**) виконується автоматично у процесі виконання алгоритму Флойда.

Алгоритм Флойда потребує виконання таких дій.

Крок 0. Визначаємо початкову матрицю відстаней $\boldsymbol{D_0}$ і матрицю послідовності вузлів $\boldsymbol{S_0}$. Діагональні елементи обох матриць позначаємо знаком «$-$», що показує, що ці елементи в обчисленнях не враховують. Припускаємо $k = 1$.

Основний крок $\boldsymbol{k}$. Задаємо рядок $\boldsymbol{k}$ і стовпчик $\boldsymbol{k}$ як розв'язувальний рядок і розв'язувальний стовпчик. Розглядаємо можливість застосування *трикутного оператора* до всіх елементів $\boldsymbol{d_{ij}}$ матриці $\boldsymbol{D_{k-1}}$. Якщо виконується нерівність $\boldsymbol{d_{ik} + d_{kj} < d_{ij}}$, $\boldsymbol{(i \neq k}$, $\boldsymbol{j \neq k}$ **і** $\boldsymbol{i \neq j)}$, тоді виконуємо такі дії:

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

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

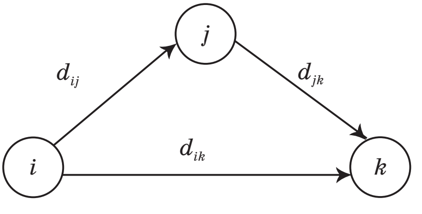


Рис. 4.5. Трикутний оператор

Після реалізації $\boldsymbol{n}$ кроків алгоритму визначення за матрицями $\boldsymbol{D_n}$ і $\boldsymbol{S_n}$ найкоротшого шляху між вузлами $\boldsymbol{i}$ і $\boldsymbol{j}$ виконують за такими правилами:

1) відстань між вузлами $\boldsymbol{i}$ і $\boldsymbol{j}$ дорівнює елементові $\boldsymbol{d_{ij}}$ у матриці $\boldsymbol{D_n}$;

2) проміжні вузли шляху від вузла $\boldsymbol{i}$ до вузла $\boldsymbol{j}$ визначаємо за матрицею $\boldsymbol{S_n}$. Нехай $\boldsymbol{s_{ij} = k}$, тоді маємо шлях $\boldsymbol{i—k—j}$. Якщо далі $\boldsymbol{s_{ik} = k}$ і $\boldsymbol{s_{kj} = j}$, тоді вважаємо, що весь шлях визначений, тому що знайдені всі проміжні вузли. В іншому разі повторюємо описану процедуру для шляхів від вузла $\boldsymbol{i}$ до вузла $\boldsymbol{k}$ і від вузла $\boldsymbol{k}$ до вузла $\boldsymbol{j}$.

Приклад 4.2

Знайдемо для мережі, показаної на рис. 4.6, найкоротші шляхи між будь-якими двома вузлами. Відстані між вузлами цієї мережі позначені на малюнку біля відповідних ребер. Ребро (3,5) орієнтовано, тому не допускається рух від вузла 5 до вузла 3. Усі інші ребра допускають рух в обидва боки.

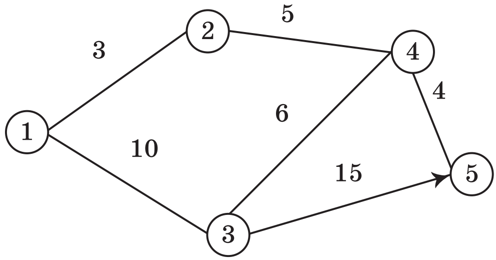


Рис. 4.6. Задана мережа

Крок 0. Початкові матриці $\boldsymbol{D_0}$ і $\boldsymbol{S_0}$ будують безпосередньо за заданою схемою мережі. Матриця $\boldsymbol{D_0}$ симетрична, за винятком пари елементів $\boldsymbol{d_{35}}$ і $\boldsymbol{d_{53}}$, де $\boldsymbol{d_{53} = \infty}$ (оскільки неможливий перехід від вузла 5 до вузла 3).

	D_0						S_0				
	1	2	3	4	5		1	2	3	4	5
1	—	3	10	∞	∞	1	—	2	3	4	5
2	3	—	∞	5	∞	2	1	—	3	4	5
3	10	∞	—	6	15	3	1	2	—	4	5
4	∞	5	6	—	4	4	1	2	3	—	5
5	∞	∞	∞	4	—	5	1	2	3	4	—

Крок 1. У матриці $\boldsymbol{D_0}$ виділені кольором ведучий рядок і стовпчик ($\boldsymbol{k = 1}$). Подвійною рамкою виокремлено елементи $\boldsymbol{d_{23}}$ і $\boldsymbol{d_{32}}$, єдині серед елементів матриці $\boldsymbol{D_0}$, значення яких можна поліпшити за допомогою *трикутного оператора*. Щоб на основі матриць $\boldsymbol{D_0}$ і $\boldsymbol{S_0}$ одержати матриці $\boldsymbol{D_1}$ і $\boldsymbol{S_1}$, виконуємо такі дії:

1. Заміняємо $\boldsymbol{d_{23}}$ на $d_{21} + d_{13} = 3 + 10 = 13$ і встановлюємо $\boldsymbol{s_{23}} = 1$;

2. Заміняємо $\boldsymbol{d_{32}}$ на $d_{31} + d_{12} = 10 + 3 = 13$ і встановлюємо $\boldsymbol{s_{32}} = 1$.

Матриці $\boldsymbol{D_1}$ і $\boldsymbol{S_1}$ мають такий вигляд:

	D_1						S_1				
	1	2	3	4	5		1	2	3	4	5
1	—	3	10	∞	∞	1	—	2	3	4	5
2	3	—	**13**	5	∞	2	1	—	**1**	4	5
3	10	**13**	—	6	15	3	1	**1**	—	4	5
4	∞	15	6	—	4	4	1	2	3	—	5
5	∞	∞	∞	4	—	5	1	2	3	4	—

Крок 2. Припускаємо $\boldsymbol{k_2}$ =2; у матриці $\boldsymbol{D_1}$ виділені кольором ведучий рядок і стовпець. Трикутний оператор застосовують до елементів матриць $\boldsymbol{D_1}$ і $\boldsymbol{S_1}$, виокремлених подвійною рамкою. У результаті одержуємо матриці $\boldsymbol{D_2}$ і $\boldsymbol{S_2}$:

	D_2						S_2				
	1	2	3	4	5		1	2	3	4	5
1	—	3	10	**8**	∞	1	—	2	3	**2**	5
2	3	—	13	5	∞	2	1	—	1	4	5
3	10	13	—	6	15	3	1	1	—	4	5
4	**8**	5	6	—	4	4	**2**	2	3	—	5
5	∞	∞	∞	4	—	5	1	2	3	4	—

Крок 3. Припускаємо $\boldsymbol{k}$ = 3; у матриці $\boldsymbol{D_2}$ виділено кольором розв'язувальний рядок і стовпчик. Трикутний оператор застосовується до елементів матриць $\boldsymbol{D_2}$ і $\boldsymbol{S_2}$, виокремлених жирною рамкою. У результаті одержуємо матриці $\boldsymbol{D_3}$ і $\boldsymbol{S_3}$:

	D_4						S_4				
	1	2	3	4	5		1	2	3	4	5
1	—	3	10	8	12	1	—	2	3	2	4
2	3	—	11	5	9	2	1	—	4	4	4
3	10	11	—	6	10	3	1	4	—	4	4
4	8	5	6	—	4	4	2	2	3	—	5
5	12	9	10	4	—	5	4	4	4	4	—

Крок 4. Припускаємо $\boldsymbol{k}$ = 4; розв'язувальний рядок і стовпчик у матриці $\boldsymbol{D_3}$ виділені. Одержуємо нові матриці $\boldsymbol{D_4}$ і $\boldsymbol{S_3}$.

	D_3						S_3				
	1	2	3	4	5		1	2	3	4	5
1	—	3	10	8	25	1	—	2	3	2	3
2	3	—	13	5	28	2	1	—	1	4	3
3	10	13	—	6	15	3	1	1	—	4	5
4	8	5	6	—	4	4	2	2	3	—	5
5	∞	∞	∞	4	—	5	1	2	3	4	—

Крок 5. Приймаємо $k = 5$; розв'язувальний рядок і стовпчик у матриці D_4 виділені кольором. Ніяких дій на цьому кроці не виконуємо; обчислення закінчені.

Кінцеві матриці D_4 і S_4 містять усю інформацію, потрібну для визначення найкоротших шляхів між будь-якими двома вузлами мережі. Наприклад, найкоротша відстань між вузлами 1 і 5 дорівнює $d_{15} = 12$.

Для вибору відповідних маршрутів нагадаємо, що сегмент маршруту (i, j) складається з ребра (i, j) тільки в тому разі, коли $s_{ij} = j$. В іншому разі вузли i і j зв'язані принаймні через один проміжний вузол. Наприклад, оскільки $s_{15} = 4$ і $s_{45} = 5$, спочатку найкоротший маршрут між вузлами 1 і 5 матиме вигляд: $1 \rightarrow 4 \rightarrow 5$. Але тому що $s_{14} \neq 4$, вузли 1 і 4 в обумовленому шляху не зв'язані одним ребром (але у вихідній мережі вони можуть бути зв'язані безпосередньо). Далі варто визначити проміжний вузол (вузли) між першим і четвертим вузлами. Маємо $s_{14} = 2$ і $s_{24} = 4$, тому маршрут $1 \rightarrow 4$ заміняємо: $1 \rightarrow 2 \rightarrow 4$. Оскільки $s_{12} = 2$ і $s_{24} = 4$, інших проміжних вузлів немає. Комбінуючи визначені сегменти маршруту, остаточно одержуємо такий найкоротший шлях від вузла 1 до вузла 5:**1 → 2 → 4 → 5.** *Довжина цього шляху дорівнює* **12 у. о.**

4.6.Індивідуальні завдання моделі «Модель Флойда»

Завдання взяти з пункту 4.1.1.

4.7. Оптимальний розподіл товарообігу по мережі

Постановка задачі (наближена до реальності).

Компанія має вісім великих оптових складів, розміщених у декількох областях. Відділ збуту ухвалив рішення значно знизити ціну одного дорогого виробу з метою ліквідації запасу виробів, що утворився. Перед початком рекламної кампанії керівництво має намір розмістити наявні запаси на зазначених восьми складах відповідно до прогнозів збуту в цих районах. Для здійснення цього плану потрібно перерозподілити деяку частину запасів (рис. 4.7).

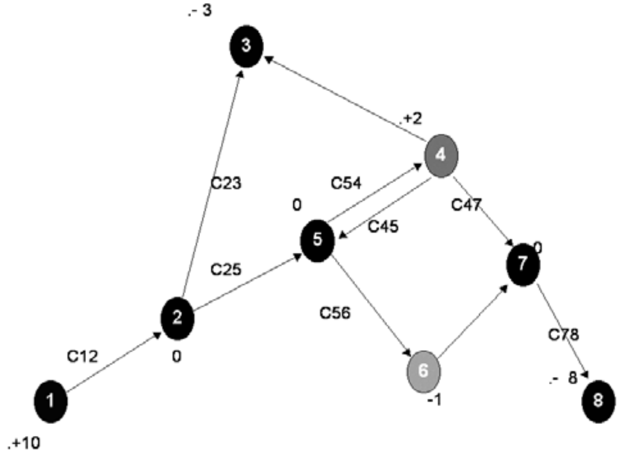


Рис. 4.7. Мережа оптових складів

Пронумеровані вузли в колах відповідають восьми складам. Додатне число, що стоїть біля номера складу, позначає надлишкову кількість запасів, що має бути розподілена в системі. Від'ємне число позначає потребу складу в додаткових запасах. Запаси на складах 2, 5 і 7 мають залишатися незмінними.

Зазначимо, що виріб може транспортуватися через склади 2, 4, 5, 6, 7; саме тому ці пункти називають ***проміжними***. Усі інші склади називають ***джерелами***, якщо в них є надлишок запасів, і ***стоками***, якщо потрібен додатковий запас => *склад **1** є джерелом, а склади **3** і **8** — стоками.*

Зауваження. Модель допускає перевезення між складами 4 і 5 в обох напрямках. Припустимо в цьому разі, що $c_{45} \neq c_{54}$ з такої причини: у момент відвантаження виробу біля компанії є можливість використовувати власний автотранспорт тільки на маршруті 4—5. Підприємство має більші витрати під час перевезень у зворотному напрямку, тому що воно змушене користуватися послугами найманого транспорту.

Аналізуючи схему, можна переконатися в тому, що поки не розв'язана вся задача, не можна виключити з розгляду жоден із можливих маршрутів перевезень між складами 4 і 5.

Побудову таблиці виконаємо в такому порядку:

1) виділити рядок для кожного ***джерела***. Значення A_i для джерела визначається кількістю виробів, що їх поставляють з цього складу (наприклад, ***склад* 1** має A_1 **= 10*);***

2) виділити стовпчик для кожного ***стоку***. Значення B_j для стоку визначається потребою у виробах (наприклад, для ***складу* 3** маємо $B_3 = -3$, а для ***складу* 8** маємо $B_8 = -8$);

3) виділити рядок і стовпчик для кожного ***проміжного пункту***. Нехай T_k — чистий запас розглянутого пункту. Якщо в цьому пункті є надлишок запасів, то T_k — додатне число, якщо ж у цьому пункті потрібне поповнення запасів, то T_k — від'ємне число. Припустимо тепер, що для будь-якого пункту $\boldsymbol{k}$ виконуються співвідношення: $A_k = T_k + B$ і $B_k = B$, де B — сумарні запаси наявні у ***всіх*** пунктах (***склади* — 2, 4, 5, 6, 7** мають $B = 12$*);*

4) задати величини $\boldsymbol{x_{ij}}$ $i \neq j$ тільки для дуг, що існують у заданій мережі.

Для проміжного пункту $\boldsymbol{k}$ задати $\boldsymbol{x_{kk}}$ при $\boldsymbol{c_{kk}}$ **= 0.**

Завершуючи аналіз, перенесемо дані рис. 4.8 на рис. 4.7, одержимо рис. 4.9.

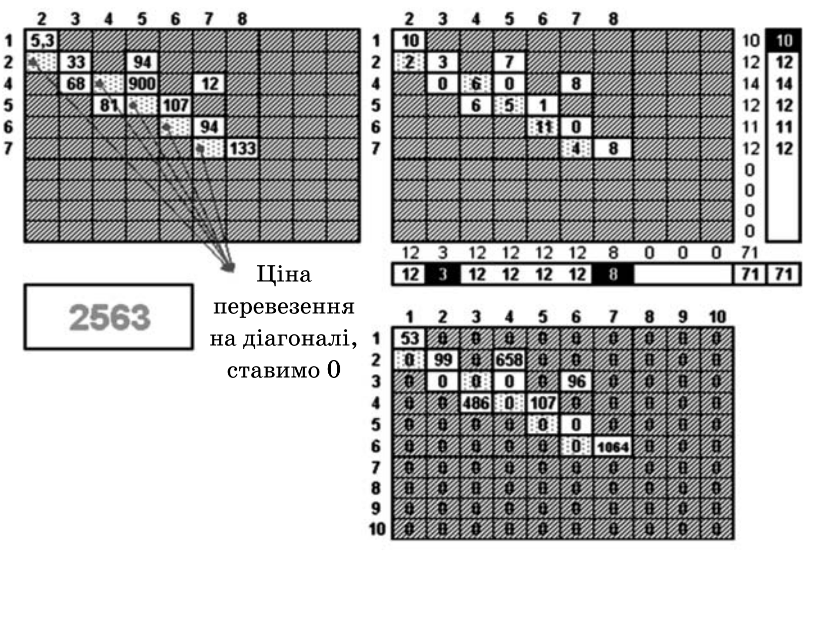


Рис. 4.8. Розрахунок моделі на MS EXCEL 2000

У результаті маємо:

Склад «1» відправляє три вироби на склад «3» через проміжний пункт 2, один виріб на склад «6» через пункти 2 і 5 і шість виробів — на склад «8» через пункти 2, 5, 4 і 7. Склад «4» відправляє два вироби на склад «8» через пункт 7.

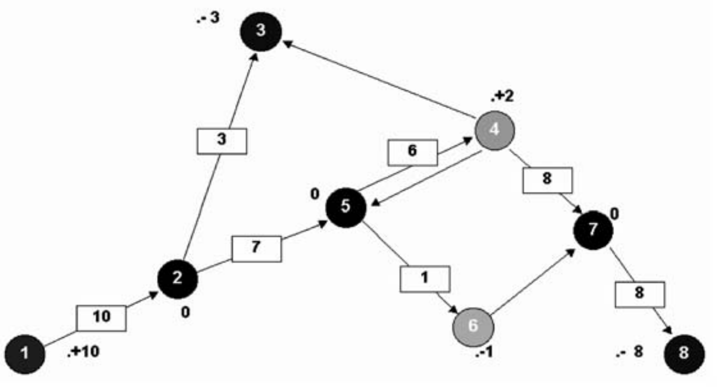


Рис. 4.9. Мережа оптових складів після реалізації моделі

4.8. Індивідуальні завдання моделі «Оптимальний розподіл товарообігу по мережі»

Розв’яжіть задачу для даних пункту 4.4

Модуль 5
КОМБІНАТОРНІ МОДЕЛІ

5.1. Метод гілок і меж для задачі комівояжера (Алгоритм-1)

Постановка задачі комівояжера (роз'їзного торгового агента) дуже проста. Маємо $\boldsymbol{n}$ міст і значення всіх відстаней між містами. Вважаємо, що кожне місто має шлях до кожного іншого. Відсутність шляху можна розглядати як шлях нескінченної відстані. Комівояжер повинен обійти всі міста і повернутися в початкове. Вибраний маршрут має бути мінімальної довжини. Кількість усіх можливих маршрутів дорівнює $(\boldsymbol{n} - \mathbf{1})!$ Простий перебір можливий лише для невеликих значень $\boldsymbol{n}$. Кожний шлях може характеризуватися не відстанню, а вартістю проїзду з одного пункту в інший. У такому разі шукаємо мінімальну вартість проїзду комівояжера.

Визначення 1.

$\min f(x)$ — ? $x \in K$ — кінцева або рахункова $\xrightarrow{def}$ комбінаторна задача.

Визначення 2. $\forall f^0 \mid f^0 \geq f(x^*) \xrightarrow{def}$ верхня межа задачі.

Визначення 3.

$$f_1 \leq f(x) \forall x \in K_1; \qquad K_1, K_2 \,\Big|\, \begin{cases} K = K_1 \cup K_2 \\ K_1 \cap K_2 = \emptyset \end{cases}$$
$$f_2 \leq f(x) \forall x \in K_2;$$

$\xrightarrow{def}$ нижні межі для підмножин K_1, K_2.

Твердження 1: $f_2 > f^0 \Rightarrow x^* \in K_1$.

Побудова алгоритму-1:

1. Вибираємо пункти «i» і «j»:

$$\left.\begin{array}{l} K_1 \xrightarrow{def} \{\text{маршр. } i \to j\} \\ K_2 \xrightarrow{def} \{\text{маршр. не з } i \to j\} \end{array}\right\} \Rightarrow \begin{cases} K = K_1 \cup K_2 \\ K_1 \cap K_2 = \emptyset. \end{cases}$$

2. Вибираємо $\forall$ допустимий маршрут «x»: $f(x) \geq f(x^*)$ верхня *межа* $\xleftarrow{def} f^0$.

3. Нижня межа — ?

Алгоритм вибору нижніх меж:

$$\boxed{(A)\text{:матриця відстаней}} \xrightarrow{def} \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} \begin{array}{l} \xrightarrow{\min} K_1 \\ \xrightarrow{\min} K_2 \\ \\ \xrightarrow{\min} K_n \end{array} \Rightarrow \overset{\circ}{A}. \begin{pmatrix} a_{11}-K_1 & a_{12}-K_1 & \dots & a_{1n}-K_1 \\ a_{21}-K_2 & a_{22}-K_2 & \dots & a_{2n}-K_2 \\ \dots & \dots & \dots & \dots \\ a_{n1}-K_n & a_{n2}-K_n & \dots & a_{nn}-K_n \end{pmatrix}$$

$$\xrightarrow{\min} \mathbf{L_1} \quad \xrightarrow{\min} \mathbf{L_2} \quad \xrightarrow{\min} \mathbf{L_n}$$

$$\begin{pmatrix} a_{11}-K_1-L_1 & \dots & a_{1n}-K_1-L_n \\ a_{21}-K_2-L_1 & \dots & a_{2n}-K_2-L_n \\ \dots\dots\dots\dots & \dots & \dots\dots\dots\dots \\ a_{n1}-K_n-L_n & \dots & a_{nn}-K_n-L_n \end{pmatrix} = \overset{\circ\circ}{A} \longleftarrow$$

де $A \xrightarrow{def}$ матриця відстаней;

$\overset{\circ}{A} \xrightarrow{def}$ матриця зменшених відстаней;

$\overset{\circ\circ}{A} \xrightarrow{def}$ 2-а матриця зменшених відстаней;

S — довжина р-го маршруту, розрахованого на А;

S_1 — довжина р-го маршруту, розрахованого на $\overset{\circ}{A}$.

$$K := \sum_{i=1}^{n} K_i \Rightarrow \boldsymbol{S = S_1 + K}; \quad (*)$$

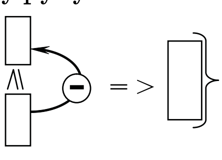
$$\left.\begin{array}{l} S_2 \text{ — довжина } p\text{-го маршруту, розрахованого на } \overset{\circ\circ}{A}; \\ L := \sum_{i=1}^{n} L_i; \end{array}\right\} \Rightarrow$$

$\Rightarrow \boldsymbol{S_1 = S_2 + L} \rightarrow (*)$

$\Rightarrow S = S_2 + L + K = [S_2 \geq 0] \Rightarrow \forall x \in K : S\,(x) \geq L + K \rightarrow$

$\rightarrow$ основна нерівність покладена в основу одержання нижніх меж.

Алгоритм вибору початку руху — ?



1. $\min\limits_{j}$ a

2. min (із залишків) $\underset{j}{\text{a}}$

max — ? a

a рядок перебування **max** $\rightarrow$ у цьому рядку вибираємо **min** елемент $\Rightarrow$ клітинка початку руху.

5.2. Метод гілок і меж для задачі комівояжера (Алгоритм-2)

На відміну від алгоритму-1 — бінарного розподілення, в алгоритмі-2 безліч K розподілимо на m підмножин, таких, що:

$$K = K_1 \cup K_2 \cup \ldots \cup K_m; \quad K_i \cap K_j = \emptyset; \; i \neq j.$$

Визначення 5.2.1: $f_i \leq f(x) \;\; \forall x \in K_i \xrightarrow{def}$ нижня межа для K_i.

Визначення 5.2.2: $\forall f^0 \,|\, f^0 \geq f(x^*) \xrightarrow{def}$ верхня межа задачі.

Твердження 5.2.1: $f_i > f^0 \Rightarrow x^* \notin K_i$. Аналогічно: $S(x) \geq L + K$.

Наведемо приклади реалізації алгоритму-1 — рис. 5.1, 5.2 і алгоритму-2 — рис. 5.3, 5.4, 5.5.

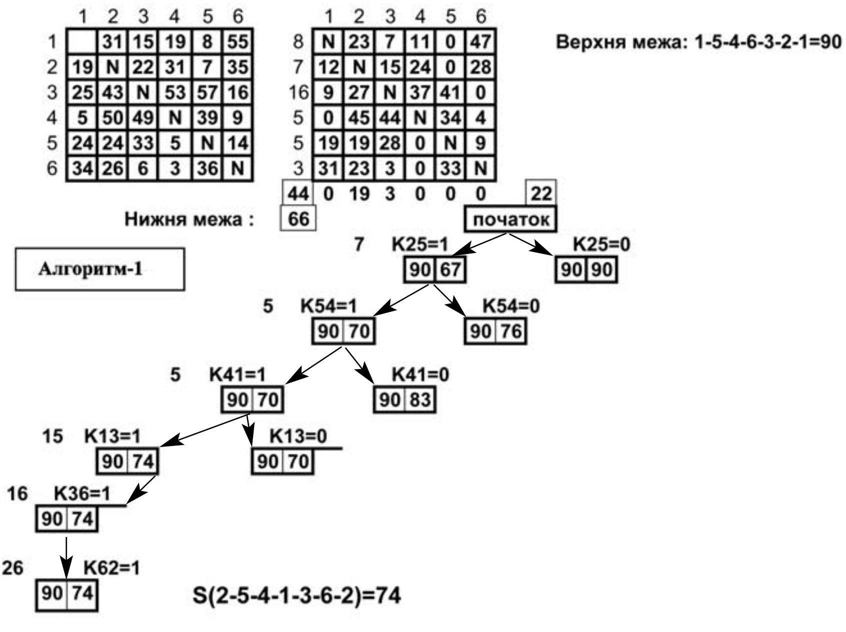


Рис. 5.1. Дерево алгоритму-1,
перша ітерація

За верхню межу візьмемо отриманий шлях $S = \mathbf{74}$.

Піднімаємося вгору по гілках, знаходимо необірвані гілки, для яких знайдене значення шляху S не перевищує нижню межу, і повторюємо процес — рис. 5.2.

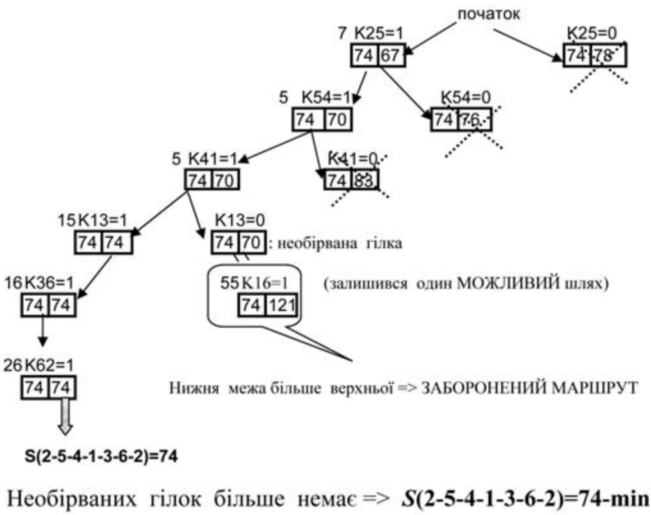


Рис. 5.2. Дерево алгоритму-1,
друга ітерація

Необірваних гілок більше немає => **S(2-5-4-1-3-6-2) = 74-min.**

5.3. Індивідуальні завдання (Алгоритм-1)

Реалізуйте алгоритм-1 для матриць відстаней, наведених у табл. 5.1.

Таблиця 5.1

Матриці відстаней

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
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

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

5.4. Метод гілок і меж для задачі комівояжера (Алгоритм-2)

	1	2	3	4	5	6
1	N	31	15	19	8	55
2	19	N	22	31	7	35
3	25	43	N	53	57	16
4	5	50	49	N	39	9
5	24	24	33	5	N	14
6	34	26	6	3	36	N

	1	2	3	4	5	6	
8	N	23	7	11	0	47	
7	12	N	15	24	0	28	
16	9	27	N	37	41	0	
5	0	45	44	N	34	4	
5	19	19	28	0	N	9	
3	31	23	3	0	33	N	
44	0	19	3	0	0	0	22

Верхня межа:
1-5-4-6-3-2-1 = 90

Алгоритм-2.

Нижня межа: **66**

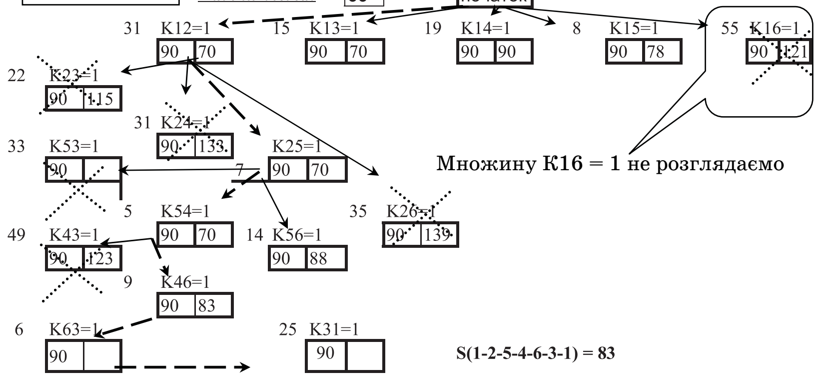


Рис. 5.3. Дерево алгоритму-2, перша ітерація

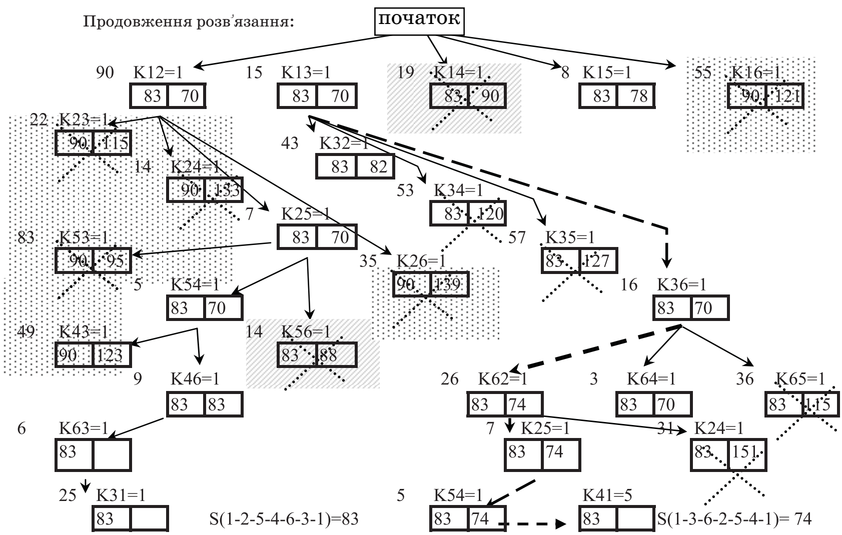


Рис. 5.4. Дерево алгоритму-2, друга ітерація

Після першої ітерації (рис. 5.3) маємо шлях S = 83. Піднімаємося вгору по гілках (замінимо межу 90 на 83) і продовжуємо процес (рис. 5.4). Після другої ітерації маємо шлях S = 74. Знаходимо необірвані гілки і повторюємо процес — третю ітерацію (рис. 5.5).

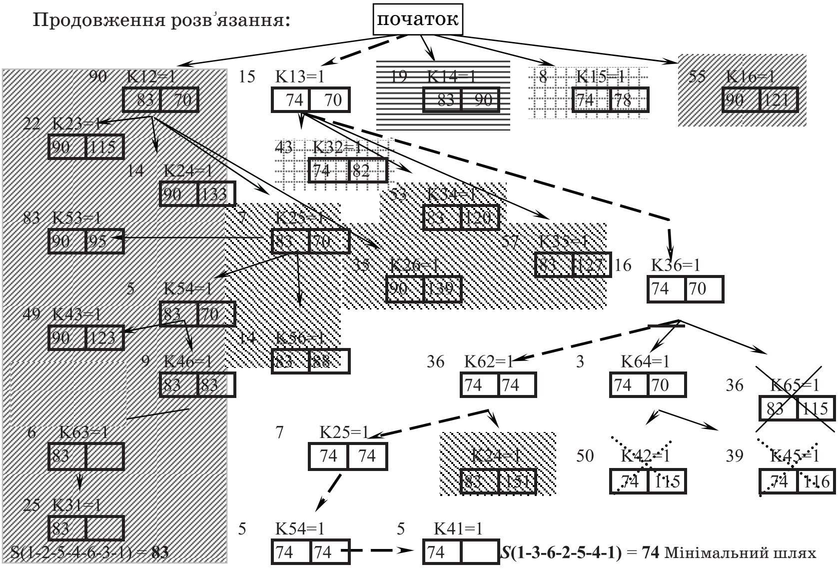


Рис. 5.5. Дерево алгоритму-2, третя ітерація

Отриманий шлях має також величину 74, але тепер відсутні необірвані гілки. Отриманий результат S = 74 (як і попередній) — оптимальний і є розв'язком задачі.

5.5. Індивідуальні завдання задачі комівояжера (Алгоритм-2)

Реалізуйте алгоритм-2 для матриць відстаней, наведених у таблиці 5.1 розділу 5.3. Результати першого та другого алгоритмів порівняйте.

Модуль 6
МОДЕЛІ КЕРУВАННЯ ЗАПАСАМИ

6.1. Основні поняття проблеми керування запасами

Задачі керування запасами належать до найчисленнішого класу економічних задач з дослідження операцій, розв'язання яких має важливе значення. Правильне й своєчасне визначення оптимальної стратегії керування запасами, а також нормативного рівня запасів дозволяє вивільнити значні оборотні кошти, заморожені у вигляді запасів, що в кінцевому підсумку підвищує ефективність зареєстрованих ресурсів.

Розглянемо основні характеристики моделей керування запасами.

Попит. Попит на продукти, які запасають, може бути детермінованим (у найпростішому випадку — постійним у часі) або випадковим. Випадковість попиту описують або випадковим моментом попиту, або випадковим обсягом попиту в детерміновані або випадкові моменти часу.

Поповнення складу. Поповнення складу можна здійснювати або періодично через певні інтервали часу, або в міру вичерпання запасів, тобто зниження їх до деякого рівня.

Обсяг замовлення. За умови періодичного поповнення і випадкового вичерпання запасів обсяг замовлення може залежати від того стану, що спостерігається в момент подачі замовлення. Замовлення зазвичай подають на ту саму величину у разі досягнення запасом заданого рівня — так званої *точки замовлення*.

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

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

Витрати збереження. У більшості моделей керування запасами обсяг складу вважають практично необмеженим, а контрольною величиною є обсяг збережених запасів. У такому разі вважають, що за збереження кожної одиниці запасу на одиницю часу стягується певна плата.

Штраф за дефіцит. Будь-який склад створюють для того, щоб запобігти дефіцитові певного типу виробів у системі, яку обслуговують. Відсутність запасу в потрібний момент призводить до збитків, пов'язаних із простоєм устаткування, неритмічністю виробництва тощо. Ці збитки надалі будемо називати *штрафом за дефіцит*.

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

Структура складської системи. Найповніше розроблені математичні моделі одиночного складу. Однак на практиці трапляються і складніші структури: ієрархічні системи складів із різними періодами поповнення і часом доставки замовлень, з можливістю обміну запасами між складами одного рівня ієрархії тощо.

Критерієм ефективності прийнятої стратегії керування запасами є *функція витрат,* що становить сумарні витрати на збереження й постачання продукту, що його запасають (у тому числі збитки від псування продукту під час збереження і в разі його морального старіння, втрати прибутку від заморожування капіталу тощо), і витрати на штрафи.

Керування запасами полягає у пошуку такої стратегії поповнення й витрати запасів, за якої функція витрат набуває мінімального значення.

Нижче розглянемо найпростіші моделі керування запасами.

6.2. Детерміновані моделі керування запасами

Нехай функції $A(t)$, $B(t)$ і $R(t)$ виражають, відповідно, поповнення запасів, їхню витрату й попит на продукт, що його запасають, за проміжок часу $[0, t]$. У моделях керування запасами звичайно використовують похідні цих функцій за часом $a(t)$, $b(t)$, $r(t)$, названі, відповідно, *інтенсивностями поповнення, витратами й попитом*.

Якщо функції $a(\mathrm{t})$, $b(t)$, $r(t)$ — не випадкові величини, то модель керування запасами вважається ***детермінованою,*** якщо хоча б одна з них має випадковий характер — *стохастичною*. Якщо всі параметри моделі не змінюються з часом, її називають *статичною,* в іншому разі — *динамічною.* Статичні моделі використовують, коли приймається разове рішення про рівень запасів на визначений період, а динамічні — у разі прийняття послідовних рішень про рівні запасу або коректування попередніх рішень з урахуванням поточних змін.

Рівень запасу в момент t визначається основним рівнянням запасів:

$$J(t) = J_0 + A(t) - B(t), \qquad (6.1)$$

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

Рівняння (6.1) частіше використовують в інтегральній формі:

$$J(t) = J_o + \int_0^t a(t)dt - \int_0^t b(t)dt. \qquad (6.2)$$

Приклад 6.1

Інтенсивність надходження деталей на склад готової продукції цеху становить на початку зміни 5 деталей/хв; протягом першої години лінійно зростає, досягає до кінця її 10 деталей/хв, і потім залишається сталою. Припустимо, що деталі надходять на склад безупинно протягом усіх семи годин робочої зміни, а вивозять деталі зі складу тільки наприкінці роботи. Треба записати рівняння для рівня запасу в довільний момент часу і, використовуючи його, визначити кількість деталей на складі: а) через 30 хв після початку роботи; б) наприкінці зміни.

Розв'язання. За умовою протягом зміни не відбувається видача деталей зі складу, тобто $b(t) = 0$. Інтенсивність поповнення запасу протягом першої години лінійно зростає, тобто $a(t) = kt + b$. З огляду на те, що $a(0) = 5$, одержуємо $b = 5$. Якщо наприкінці першої години, тобто при $t = 60$, $a(60) = 10$, то $10 = k * 60 + 5$, звідки $k = 1/12$. Таким чином, для першої години зміни $a(t) = (1/12) * t + 5$, або $a(t) = 10$.

З огляду на тривалість зміни (7 годин = 420 хв) і співвідношення (6.2), одержуємо:

$$J(t) = \int_0^t (t/12 + 5)dt = t^2/24 + 5t, \text{ якщо } 0 \le t \le 60, \text{ і}$$

$$J(t) = \int_0^{60} (t/12 + 5)dt + \int_{60}^t 10dt = (t^2/24 + 5t)\Big|_0^{60} + 10t\Big|_{60}^t =$$

$$= 450 + 10t - 600 = 10t - 150, \text{ якщо } 60 \le t \le 420.$$

Кількість деталей на складі через 30 хв після початку роботи:

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

а наприкінці зміни: $J(420) = 10*420 - 150 = 4050$.

6.3. Індивідуальні завдання «Детерміновані моделі керування запасами»

Реалізуйте модель прикладу 6.1 і визначте кількість деталей на складі через ((№ варіанту)*10) хвилин після початку роботи.

6.4. Стохастичні моделі керування запасами

Розглянемо *стохастичні моделі керування запасами,* в яких попит є *випадковим*. Цей факт істотно позначається на характері відповідних моделей і значно ускладнює їхній аналіз, тому у рамках цього посібника обмежимося розглядом найпростіших моделей.

Припустимо, що попит r за інтервал часу T є випадковим і заданий його закон (ряд) розподілу $p(r)$ або щільність імовірності $\varphi(r)$ (звичайно 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, то це призводить до штрафу за дефіцит c_3 на одиницю продукції.

Як функцію сумарних витрат, що є в стохастичних моделях випадковою величиною, розглядають її середнє значення або математичне очікування.

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

попит нижче рівня запасу

попит вище рівня запасу

$$C(s) = c_2 \sum_{r=0}^{s} (s - r)\, p(r) + c_3 \sum_{r=s+1}^{\infty} (r - s)\, p(r). \tag{6.3}$$

У рівнянні (6.3) перший доданок враховує витрати на придбання (збереження) надлишку $\boldsymbol{s - r}$ одиниць продукту (при $r \le s$), а другий доданок — штраф за дефіцит на $\boldsymbol{r - s}$ одиниць продукту (при $r > s$).

У разі безперервного випадкового попиту, що задається щільністю ймовірностей $\varphi(r)$, рівняння $C(s)$ набуває вигляду:

$$C(s)=c_2\int_0^s (s-r)\varphi(r)dr+c_3\int_{s+1}^{\infty}(r-s)\varphi(r)dr. \qquad (6.4)$$

Задача керування запасами полягає у відшуканні такого запасу ***s****, за якого математичне чекання сумарних витрат (6.3) або (6.4) набуває мінімального значення.*

Доведено, що в разі дискретного випадкового попиту r рівняння **(6.3)** мінімальне при запасі s_0, що задовольняє нерівностям

$$F(s_0)<\rho<F(s_0+1), \qquad (6.5)$$

а в разі безперервного випадкового попиту r рівняння **(6.4)** мінімальне при значенні s_0, обумовленого з рівнянням

$$F(s_0)=\rho, \qquad (6.6)$$

де

$$F(s)=p(r<s) \qquad (6.7)$$

є функція розподілу попиту r; $F(s_0)$ і $F(s_0+1)$ — її значення; ρ — щільність збитків через незадоволений попит, обумовлена формулою:

$$\rho=\frac{c_3}{c_2+c_3}. \qquad (6.8)$$

Приклад 6.2

Підприємство купує агрегат із запасними блоками до нього. Вартість одного блоку становить 5 грошових одиниць. У разі виходу агрегату з ладу через полом блоку, відсутнього в запасі, простій агрегату і термінове замовлення нового блоку до нього обійдеться в 100 грошових одиниць. Розподіл агрегатів за кількістю блоків, що потребують заміни, наведено в таблиці 6.1.

Таблиця 6.1

Розподіл агрегатів по кількості блоків, 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

Потрібно визначити оптимальну кількість запасних блоків, які варто придбати разом з агрегатом.

Розв'язання. За умовою $c_2 = 5$, c_3, $= 100$. Обчислимо щільність збитків через брак запасних блоків за формулою (6.8): $\rho = 100/(5 + 100) = 0{,}952$.

З огляду на (6.7), знайдемо значення функції розподілу попиту (табл. 6.2).

Таблиця 6.2

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	0,1

Очевидно (див. таблицю 6.2), що оптимальний запас становить $s_0 = 2$, тому що він задовольняє нерівність (6.5): $F(2) < 0{,}952 < F(3)$.

Відповідь: *рекомендація — придбати **2** запасних блоки.*

Приклад 6.3

Розв'язати *приклад 6.2* за умови безперервного випадкового попиту ***r***, розподіленого за показовим законом із функцією розподілу $F(r) = 1 - e^{-\lambda r}$ при $\lambda = 0{,}98$.

Розв'язання. Оптимальне число запасних блоків s_0 визначимо з рівняння (6.6): $1 - e^{-\lambda s_0} = \rho$, звідки $e^{-\lambda s_0} = 1 - \rho$ і $s_0 = -\frac{1}{\lambda}\ln(1-\rho)$.

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

Відповідь: *рекомендація — придбати **4** запасних блоки.*

6.5. Індивідуальні завдання «Стохастичні моделі керування запасами»

Задача 1

Реалізуйте демонстраційну модель (приклад 6.2) при змінених умовах:

а) вартості блоку; б) вартості термінового замовлення; в) статистичної імовірності блоків, що потребують заміни. Дані наведено в таблиці 6.3.

Задача 2

Реалізуйте демонстраційну модель (приклад 6.3) при змінених умовах: а) параметра λ; б) вартості блоку; в) вартості термінового замовлення. Дані наведено в таблиці 6.3.

Таблиця 6.3

Варіанти завдань задач 1, 2

№ варіанту	Число замінених блоків, N							Вартість блоку	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
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
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

6.6. Автоматизація моделі керування запасами

Розглянемо таку задачу:

Підприємство купує товар за ціною 20 грн за одиницю, а продає за ціною 23 грн за одиницю. У разі, якщо товар не реалізували, його повертають за ціною 17 грн. Потрібно визначити кількість одиниць продукції для закупівлі, щоб очікуваний прибуток був максимальним.

Реалізацію моделі виконаємо засобами *MS Excel* 2000 (рис. 6.1).

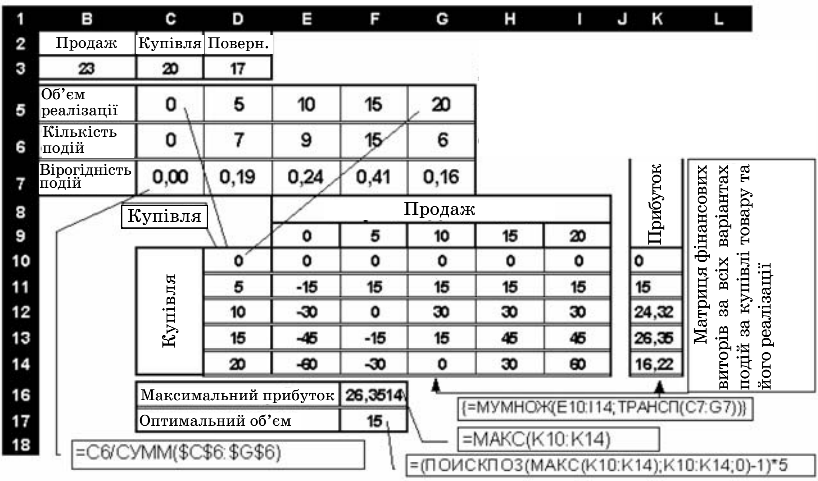


Рис. 6.1. Електронна форма таблиці в *Exel* при реалізації моделі

Sub vvod **()**:' *процедура уведення даних*

Range («продаж»**)** = **InputBox (**«Уведіть вартість продажу»**)**

Range («купівля»**)** = **InputBox (**«Уведіть вартість купівлі»**)**

Range («повернення»**)** = **InputBox (**«Уведіть вартість повернення»**)**

End Sub

Тут матриця різноманітних результатів реалізована за допомогою процедури-функції **(Function** прибуток **( ))**

Код процедури-функції:

Option Base 1

Function прибуток (покуп)

N = купів. **Rows. Count**

```
      цпрод. = Range («продаж»). Value
      црешт. = Range («купівля»). Value
      црешт = Range («повернення»). Value
   ReDim res (N, N)
   For i = 1 To N
      For j = 1 To N
        If i <= j Then res (i, j) = купів (i) * (цпрод — цкупів)
        If i > j Then res (i, j) = купів (j) * (цпрод — цкупів) — –
      – (купів (i) — купів (j)) * (цкуп — црешт)
      Next j
   Next i
прибуток = res
   End Function
```

6.7. Індивідуальні завдання «Автоматизація моделі»

Реалізуйте модель пункту 6.3. Дані наведені в таблиці 6.4.

Таблиця 6.4

Варіанти завдань

Вар.	Продаж	Купівля	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
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

Закінчення табл. 6.4

Вар.	Продаж	Купівля	Повернення	Число подій				
				0 шт.	5 шт.	10 шт.	15 шт.	20 шт.
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

Модуль 7
ЕЛЕМЕНТИ ТЕОРІЇ ІГОР

7.1. Задачі теорії ігор

На практиці часто виникають задачі, в яких потрібно виносити рішення в умовах невизначеності, тобто виникають ситуації, коли дві (або більше) сторони мають різні цілі, а результати будь-якої дії кожної зі сторін залежать від дій партнера. Такі ситуації, що виникають під час гри в шахи, шашки, доміно тощо, належать до конфліктних: результат кожного ходу гравця залежить від відповідного ходу супротивника, мета гри — виграш одного з партнерів. В економіці конфліктні ситуації виникають дуже часто і 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 партнерів і прагненням кожного з них приймати оптимальні рішення. При цьому кожен має враховувати цілі свої та партнера, а також ймовірні дії та обставини.

Для успішного розв'язання задач із конфліктними ситуаціями застосовують науково обґрунтовані методи. Такі методи розроблені математичною теорією конфліктних ситуацій, що зветься *теорія 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 поводження партнерів;

3) виграш, до якого призводить кожна сукупність дій. Як правило, виграш (або програш) може бути заданий кількісно; наприклад, можна оцінити програш нулем, виграш — одиницею, а нічию — 0,5.

Гра називається *парною*, якщо в ній беруть участь два гравці, і *множинною*, якщо кількість гравців більше двох. Ми розглянемо тільки парні ігри. У них беруть участь два гравці — $\boldsymbol{A}$ й $\boldsymbol{B}$, інтереси яких протилежні, а під грою будемо розуміти низку 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 завдання гри досить указати величину одного з них. Якщо позначити a — виграш одного з гравців, b — виграш іншого гравця, то для гри з нульовою сумою $b = -a$, тому досить розглядати, наприклад, a.

Вибір і здійснення одного з передбачених правилами дій називається *ходом* гравця. Ходи можуть бути особистими і 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 розглядатимемо тільки особисті ходи гравців.

Стратегією гравця називається сукупність правил, за якими він обирає особистий хід залежно від сформованої ситуації. Звичайно, в процесі гри, під час кожного особистого ходу гравець робить вибір залежно від конкретної ситуації. Однак у принципі можливо, що всі рішення будуть спрогнозовані гравцем заздалегідь (у відповідь на будь-яку сформовану ситуацію). Це означає, що гравець вибрав визначену стратегію, що може бути задана у формі списку правил або програми. (Так можна здійснити гру за допомогою ЕОМ.)

Гра називається *скінченною,* якщо кожний гравець має скінченне число стратегій, і *нескінченною* — в іншому разі.

Для того щоб *вирішити* гру або знайти рішення гри, кожному гравцеві потрібно вибрати стратегію, що задовольняє умові *оптимальності,* тобто один із гравців має одержати *максимальний виграш,* а інший — *мінімальний програш,* за умови, що кожен додержується своєї стратегії.

Такі стратегії називаються *оптимальними.* Оптимальні стратегії мають також задовольняти умові *стійкості,* тобто кожному гравцеві має бути невигідно 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. Вибираючи оптимальну стратегію, природно припускати, що обидва гравця поводяться розумно стосовно своїх інтересів. Найважливіше обмеження теорії ігор — одиничність виграшу як показника ефективності, хоча більшість реальних економічних задач має декілька показників ефективності. Крім того, в економіці, як правило, виникають задачі, в яких інтереси партнерів не обов'язково антагоністичні. Розв'язання задач із багатьма учасниками, що мають несуперечливі інтереси, виходить за рамки нашого курсу.

7.2. Платіжна матриця. Нижня і верхня ціна гри

Розглянемо парну скінченну гру. Нехай гравець ***A*** має ***m*** особистих стратегій, які позначимо $A_1, A_2, \dots A_m$. Нехай у гравця ***B*** є ***n*** особистих стратегій, позначимо їх $B_1, B_2, \dots B_n$. За таких умов кажуть, що гра має розмірність $\boldsymbol{m \times n}$. У результаті вибору гравцями будь-якої пари стратегій

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

однозначно визначається результат гри, тобто виграш $\boldsymbol{a_{ij}}$ гравця ***A*** (позитивний або негативний) і програш $(-\boldsymbol{a_{ij}})$ гравця ***B***. Припустимо, що значення $\boldsymbol{a_{ij}}$ відомі для будь-якої пари стратегій $(A_i, B_j,)$. Матриця $P = (\boldsymbol{a_{ij}})$, $i = 1, 2, \dots m$; $j = 1, 2, \dots, n$, елементами якої є виграші, що відповідають стратегіям A_i і B_j, називається *платіжною матрицею* або *матрицею гри* (табл. 7.1).

Таблиця 7.1

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 стовпчики — стратегіям гравця ***B***.

Задача № 1

Гра «Пошук».

Скласти платіжну матрицю для такої гри.

Гравець ***A*** може сховатися в одному з двох притулків (I і II); гравець ***B*** шукає гравця ***A***, і якщо знайде, то одержує **1** грошову одиницю від A (штраф), у іншому разі платить гравцеві A 1 грошову одиницю. Потрібно побудувати платіжну матрицю гри.

Розв'язання. Для складання платіжної матриці проаналізуємо поводження кожного з гравців.

Гравець A може сховатися в притулок I — позначимо цю стратегію через A_1, або в притулок II — стратегія A_2. Гравець B може шукати першого гравця в притулку I — стратегія B_1, або в притулку II — стратегія B_2.

Якщо гравець A знаходиться в притулку 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 $a_{22} = -1$ (A_2, B_2). Очевидно, що стратегії (A_1, B_2) і (A_2, B_1) дають гравцеві A виграш 1, тому $a_{12} = a_{21} = 1$.

Таким чином, для гри «Пошук» обсягом 2×2 маємо таку платіжну матрицю:

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

7.3. Принцип мінімаксу

Розглянемо гру $m \times n$ з матрицею $P = (a_{ij})$, $i = 1, 2, ..., m$; $j = 1, 2,..., n$ і визначимо найкращу серед стратегій $A_1, A_2, ..., A_m$. Вибираючи стратегію A_i, гравець A має розраховувати, що гравець B відповість на неї тою зі стратегій B_j, для якої виграш для гравця A буде мінімальним (гравець B прагне «нашкодити» гравцеві A). Позначимо через λ_i, найменший виграш гравця A за стратегії A_i, незалежно від стратегій гравця B (найменше число в i-му рядку платіжної матриці), тобто

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

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

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

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

максимін: $\lambda = \max\limits_{i=\overline{1,m}} \min\limits_{j=\overline{1,n}} a_{ij}$ *(гарантований виграш для A).*

Стратегія, що відповідає максиміну, називається *максимінною стратегією.* Гравець ***B*** зацікавлений у тому, щоб зменшити виграш гравця ***A***, він ураховує максимально можливий виграш для ***A*** за умови стратегії β_j.

Позначимо $\beta_j = \max\limits_{i=\overline{1,m}} a_{ij}$.

Серед усіх чисел β_j виберемо найменше $\beta = \min\limits_{j=\overline{1,n}} \beta_j$ і назвемо β верхньою ціною гри або мінімаксним виграшем (*мінімаксом*). Це гарантований програш гравця ***B***. Отже:

мінімакс: $\beta = \min\limits_{j=\overline{1,n}} \max\limits_{i=\overline{1,m}} a_{ij}$ (*гарантований програш для* ***B***).

Стратегія, що відповідає мінімаксу, називається *мінімаксною стратегією.* Принцип, що диктує гравцям вибір мінімаксної і максимінної стратегій, називається ***принципом мінімаксу***. Цей принцип випливає зі зрозумілого припущення, що кожен гравець прагне досягти мети, протилежної меті супротивника.

Задача № 2

Визначимо нижню і верхню ціни гри і відповідні стратегії в задачі 1. Розглянемо платіжну матрицю із задачі 1.

У задачі **1**, розглянутій вище, верхня і нижня ціни гри різні: $\lambda \neq \beta$.

Якщо верхня і нижня ціни гри збігаються, то загальне значення верхньої і нижньої ціни гри $\lambda = \beta = \nu$ 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 разі гравець ***A*** одержує максимальний гарантований (не залежний від дій гравця ***B***) виграш ν, а гравець ***B*** домагається мінімального гарантованого (незалежно від дій гравця ***A***) програшу ν.

Вважають, що рішення гри має *стійкість,* тобто якщо один із гравців додержується своєї оптимальної стратегії, то для іншого не може бути вигідним відхилення від своєї оптимальної стратегії.

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

Задача № 3

Визначити нижню і верхню ціну гри заданою платіжною матрицею:

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

Для цього знайдемо мінімум у кожному рядку та максимум у кожному стовпці і визначимо, чи має гра сідлову точку. Розв'язок подано у формі матричної таблиці на рис. 7.1.

A_j \ B_j	B_1	B_2	B_3	α_j
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	$\alpha = \beta = 0,7$

Рис. 7.1. Сідлова точка в платіжній матриці

7.4. Ігрові моделі 2×2. Рішення ігор у змішаних стратегіях

7.4.1. Графічне рішення гри із заданою платіжною матрицею

Якщо гра не має сідлової точки, то застосування чистих стратегій не дає оптимального рішення гри. Так, у задачі 1: $\lambda \neq \beta$, тобто, сідлова точка відсутня. У такому разі можна одержати оптимальне рішення, якщо випадковим чином змінювати чисті стратегії.

Визначення 7.4.1. Змішаною стратегією S_A гравця A називається застосування чистих стратегій $A_1, A_2, ..., A_m$ з імовірностями $p_1, p_2, ..., p_m$, причому сума ймовірностей дорівнює одиниці: $\sum_{i=1}^{m} p_i = 1$.

Змішані стратегії гравця A записують 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: $S_A = (P_1\ P_2\ ...\ P_i\ ...\ P_m)$.

Аналогічно змішані стратегії гравця ***B*** позначають:

$$S_B = \begin{pmatrix} B_1 & B_2 & \dots & B_j & \dots & B_n \\ P_1 & P_2 & \dots & P_j & \dots & P_n \end{pmatrix}$$

або у вигляді рядка: $S_B = (q_1\ q_2\ \dots\ q_i\ \dots\ q_n)$,
де сума ймовірностей появи стратегій дорівнює одиниці:

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

Чисті стратегії можна вважати частковим випадком змішаних, в яких $p = 1$ 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 має межі: $\lambda \leq \nu \leq \beta$, де λ і β — *нижня і верхня ціни гри.*

Теорема «Неймана» (Джон фон Нейман (1903–1957) — американський математик).

Кожна скінченна гра має принаймні одне оптимальне рішення, можливо, серед змішаних стратегій:

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

Визначення 7.1.2. Якщо чиста стратегія належить до оптимальної змішаної стратегії 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 дорівнює ціні гри, за умови, що інший 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 найпростішою скінченної грою.

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

2. Гра, у якій *відсутня сідлова точка*, відповідно до теореми **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, то його середній виграш дорівнює ціні гри ν, хоч би якою 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 додержується оптимальної змішаної стратегії:

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

а гравець ***B*** застосовує чисту стратегію B_1 (це відповідає 1-му стовпчику платіжної матриці P), дорівнює ціні гри υ:

$$a_{11}p_1^* + a_{21}p_2^{*}* = \upsilon. \tag{7.1}$$

Той же середній виграш одержує гравець ***A***, якщо 2-й гравець застосовує стратегію B_2, тобто $a_{12}p_1^* + a_{22}p_2^* = \upsilon$. З огляду на те, що $p_1^* + p_2^* = 1$, одержуємо систему рівнянь для пошуку оптимальної стратегії S_A^*, і ціни гри ν:

$$\begin{cases} a_{11}P_1^* + a_{21}P_2^* = \upsilon, \\ a_{12}P_1^* + a_{22}P_2^* = \upsilon, \\ \quad P_1^* + P_2^* = 1. \end{cases}$$

Розв'яжемо цю систему і отримаємо оптимальну стратегію

$$P_1^* = \frac{a_{22} - a_{21}}{a_{11} + a_{22} - a_{12} - a_{21}}; \quad P_2^* = \frac{a_{11} - a_{12}}{a_{11} + a_{22} - a_{12} - a_{21}}$$

і ціну гри:

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

Застосовуючи теорему про активні стратегії при пошуку 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 ν, 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:

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

Задача № 4

Знайти оптимальні стратегії гри, за умови виграшу в задачі 1.

Розв'язання. Гра «пошук» задана платіжною матрицею без сідлової точки:

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

Тому шукаємо рішення в змішаних стратегіях:

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

Розвязуючи ці системи, одержуємо: $p_1^* = p_2^* = q_1^* = q_2^* = \dfrac{1}{2},\ v = 0.$

Це означає, що оптимальна стратегія кожного гравця полягає в тому, щоб чергувати свої чисті стратегії випадковим чином, вибираючи кожний із притулків з імовірністю 0,5; при цьому середній виграш дорівнює 0.

Геометрична інтерпретація гри 2×2 зображена на рис. 7.2.

Розглянемо подібні трикутники рис. 7.2:

$$\Delta M_1B_1T \infty \Delta \mathring{B}_1B_1D \Rightarrow \frac{B_1T}{M_1T} = \frac{B_1D}{\mathring{B}_1D} = \frac{a_{21} - \nu}{p_1} = \frac{a_{21} - a_{11}}{1} \Rightarrow$$

$$\Rightarrow a_{21} - \nu = p_1(a_{21} - a_{11}) \Rightarrow \nu = a_{21} - p_1a_{21} + p_1a_{11} \Rightarrow \nu = a_{21}(1 - p_1) +$$

$$+ p_1a_{11} \Rightarrow \boxed{\nu = a_{21}p_2 + a_{11}p_1}$$ *порівняйте з формулою* (7.1)

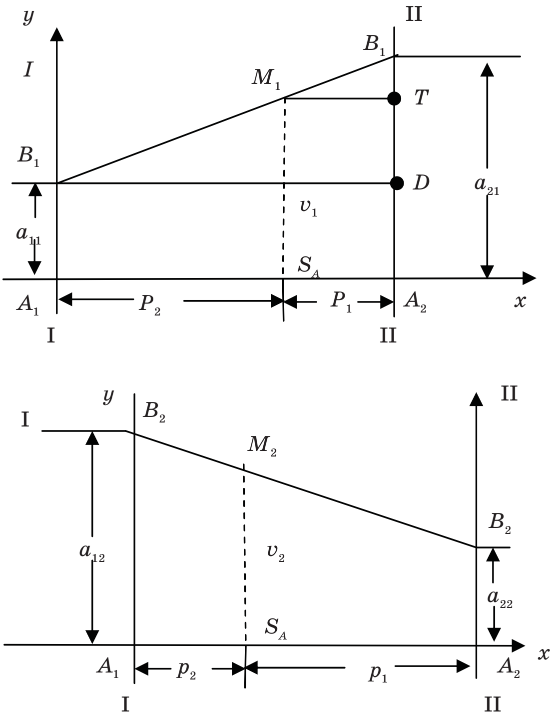


Рис. 7.2. Геометрична інтерпретація гри 2×2

Поєднуючи I-й і II-й рисунки 7.2, одержуємо загальний фрагмент:

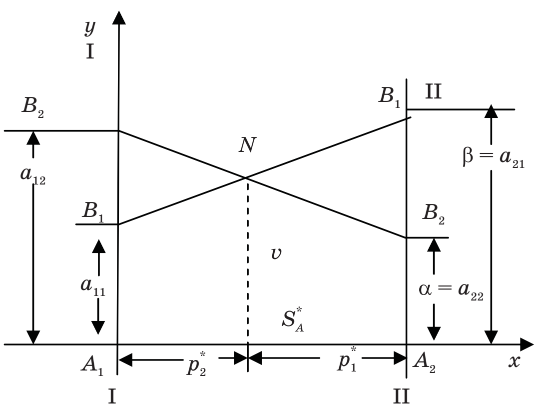


Рис. 7.3. Графічне рішення ігрової моделі

Наведемо рішення конкретної задачі. Для заданої платіжної матриці графічно знайти ціну гри з платіжною матрицею P:

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

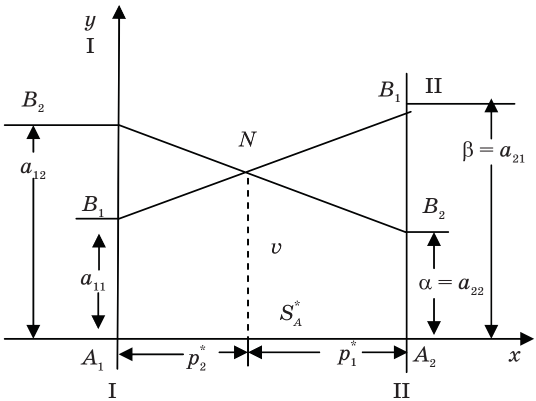


Рис. 7.4. Реалізація моделі із заданою платіжною матрицею

$$B_1B_1 \cap B_2B_2 = N:$$

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

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

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

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

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

$$x = \frac{1,5}{2,5} = 0,6 = p_2;$$

$$p_1 = 1 - 0,6 = 0,4.$$

Відповідь: Таким чином, $p_1^* = 0,6$; $p_2^* = 1 - 0,6 = 0,4$; оптимальна стратегія $S_A^* = (0,6\ ;\ 0,4)$, ціна гри $\nu = 1,8$.

7.5. Індивідуальні завдання ігрових моделей 2×2

Знайдіть оптимальні стратегії гри із заданою платіжною матрицею:

а) аналітично;

б) графічно;

в) порівняйте результати.

Перевірте, чи має гра сідлову точку.

Таблиця 7.2

Платіжні матриці 2×2

Варіант 1

2	3
4	1

Варіант 2

1	3
4	0

Варіант 3

1	4
3	1

Варіант 4

1	5
2	1

Варіант 5

0	2
4	1

Варіант 6

2	0
1	9

Варіант 7

0	3
4	0

Варіант 8

0	3
3	2

Варіант 9

1	6
3	1

Варіант 10

0	3
5	1

Варіант 11

2	5
4	1

Варіант 12

3	4
7	2

Варіант 13

1	4
7	0

Варіант 14

5	9
7	4

Варіант 15

5	2
0	4

Варіант 16

4	2
0	3

Варіант 17

6	2
1	3

Варіант 18

7	0
1	3

Варіант 19

8	2
1	4

Варіант 20

4	5
8	2

Варіант 21

3	0
1	3

Варіант 22

5	0
3	4

Варіант 23

1	7
3	2

Варіант 24

2	1
0	4

Варіант 25

2	4
5	3

Варіант 26

1	5
6	3

Варіант 27

2	4
6	1

Варіант 28

3	5
7	1

Варіант 29

4	5
7	2

Варіант 30

3	5
6	1

7.6. Ігрові моделі $2 \times n$

Рішення матричних ігор у змішаних стратегіях можна знайти 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 із нульовою сумою.

Графічне рішення ігор

Розглянемо гру $2 \times n$, в якій гравець $\boldsymbol{A}$ має дві стратегії:

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

Гра припускає, що гравець $\boldsymbol{A}$ змішує стратегії A_1 і A_2 з відповідними ймовірностями x_1 і $1 - x_1$, $0 \leq x\ 1 \leq 1$. Гравець $\boldsymbol{B}$ змішує стратегії B_1, B_2, ..., B_n з імовірностями y_1, y_2, ..., y_n, де $y_1 \geq 0$ *і* $y_1 + y_2 + ... + y_n = 1$. У цьому разі очікуваний виграш гравця $\boldsymbol{A}$, що відповідає j-тій чистій стратегії гравця $\boldsymbol{B}$, обчислюється у вигляді $(a_{1j} - a_{2j})x_1 + a_{2j}$, $j = 1, 2, ..., n$.

Отже, гравець $\boldsymbol{A}$ шукає величину x_1, що максимізує мінімум очікуваних виграшів:

$$\max_{x_1} \min_{j} \left\{ \left(a_{1_j} - a_{2_j} \right) x_1 + a_{2_j} \right\}.$$

Приклад 7.1

Розглянемо таку гру 2×4, у якій платежі виплачуються гравцеві $\boldsymbol{A}$.

	y_1	y_2	y_3	y_4
A_1	2	2	3	−1
A_2	4	3	2	6

Гра не має рішення в чистих стратегіях, і отже, стратегії мають бути змішаними. Очікувані виграші гравця $\boldsymbol{A}$, що відповідають чистим стратегіям гравця $\boldsymbol{B}$, записані в таблиці 7.3.

Таблиця 7.3

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 7.5 зображені чотири прямі лінії, що відповідають чистим стратегіям гравця ***B***. Щоб визначити найкращий результат із найгірших, побудована нижня межа чотирьох зазначених прямих (зображена на малюнку товстими лінійними сегментами), що є мінімальним (найгіршим) виграшем для гравця ***A***, незалежно від дій гравця ***B***. Максимум (найкраще) нижньої, що огинає, відповідає максимальному рішенню в точці $x_1^* = 0,5$. Ця точка визначається перетинанням прямих 3 і 4. Отже, оптимальним рішенням для гравця ***A*** є змішування стратегій A_1 і A_2 з імовірностями 0,5 і 0,5 відповідно. Ціна гри ν визначається підстановкою $x_1 = 0,5$ у рівняння або прямої 3, або 4, що приводить до такого:

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

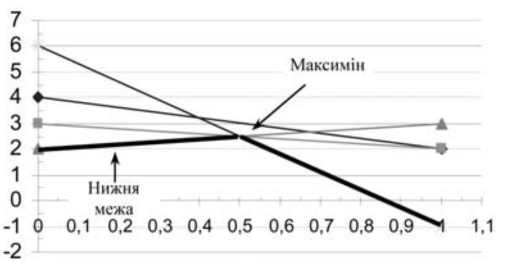


Рис. 7.5. Побудова нижньої огинаючої

Оптимальна змішана стратегія гравця ***B*** визначається двома стратегіями, що визначають нижню межу графіка. Це значить, що гравець ***B*** може змішувати стратегії B_3 і B_4. У цьому разі $y_1 = y_2 = 0$ і $y_4 = 1 - y_3$. Отже, очікувані платежі гравця ***B***, що відповідають чистим стратегіям гравця ***A***, мають такий вигляд (табл. 7.4):

Таблиця 7.4

Очікувані платежі гравця *B*

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

Найкраще розв’язання з найгірших для гравця $\boldsymbol{B}$ є точкою мінімуму верхньої, що огинає задані дві прямі. Ця процедура еквівалентна розв’язанню рівняння: $4y_3 - 1 = -4y_3 + 6$. Його розв’язанням буде $y_3 = 7/8$, що визначає ціну гри $\nu = 4 \times (7/8) - 1 = 3/4$.

Таким чином, розв’язанням задачі теорії гри для гравця $\boldsymbol{A}$ є змішування стратегій A_1 і A_2 з рівними імовірностями 0,5, а для гравця $\boldsymbol{B}$ — змішування стратегій B_3 і B_4 з імовірностями 7/8 і 1/8.

7.7 Індивідуальні завдання ігрових моделей 2×n

Реалізуйте графічно гру 2×4 з такими платіжними матрицями:

Таблиця 7.5

Платіжні матриці прикладу 7.1

№ 1			
5	2	6	4
2	3	1	3
№ 2			
−1	2	1	−6
−4	−3	−6	−2
№ 3			
1	5	3	4
3	2	1	0
№ 4			
2	−1	−5	−4
−3	−3	0	1
№ 5			
−1	0	3	4
3	2	0	−1
№ 6			
−2	1	−5	−1
−5	−4	3	−3
№ 7			
7	1	1	4
−1	2	3	1
№ 8			
−2	−1	−5	1
−5	−3	1	−4
№ 9			
5	2	1	4
−2	7	3	1
№ 10			
0	0	−5	1
−2	−4	0	−4

№ 11			
1	5	−6	4
3	2	5	3
№ 12			
−1	2	−3	−1
1	−3	1	−4
№ 13			
7	0	−1	5
0	4	3	3
№ 14			
−2	0	−4	−1
2	−1	2	−2
№ 15			
−2	2	7	6
5	6	−1	1
№ 16			
4	−3	−4	−2
−2	1	2	−3
№ 17			
3	2	5	0
−2	6	−1	1
№ 18			
0	−1	−3	5
−5	1	3	−3
№ 19			
2	−1	2	3
4	2	1	0
№ 20			
−4	0	−3	2
1	−5	3	−4

№ 21			
0	−1	2	3
4	2	−1	−3
№ 22			
−6	1	3	2
1	−3	−5	−1
№ 23			
4	−1	3	5
0	3	−2	−3
№ 24			
0	−3	1	−5
−6	1	−4	−3
№ 25			
1	−1	3	5
7	5	1	0
№ 26			
3	1	−3	6
−4	3	0	−3
№ 27			
1	−1	3	5
7	5	2	0
№ 28			
−4	1	−3	6
0	2	4	−3
№ 29			
7	2	−2	3
3	4	2	0
№ 30			
1	−3	−2	4
−1	0	4	−3

7.8. Розв'язання матричних ігор методами лінійного програмування

Теорія ігор зв'язана з лінійним програмуванням, тому будь-яку кінцеву гру двох осіб із нульовою сумою можна подати у формі задачі лінійного програмування. Розділ 7.4 ілюструє розв'язання задач матричних ігор методами лінійного програмування.

Оптимальні значення ймовірностей x_i, $i = 1, 2, \ldots, m$, гравця $\boldsymbol{A}$ можна визначити розв'язанням такої максимінної задачі:

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

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

$$x_i \ge 0,\ \ i = 1, 2, \ldots, m.$$

Щоб сформулювати цю задачу як задачу лінійного програмування, припустимо:

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

Звідси випливає, що

$$\sum_{i=1}^{m} a_{ij}x_i \ge v,\ j = 1, 2, \ldots, n.$$

Отже, задача гравця $\boldsymbol{A}$ може бути записана у вигляді:

максимізувати $z = v$
при обмеженнях

$$\sum_{i=1}^{m} a_{ij}x_i \ge v,\ j = 1, 2, \ldots, n$$

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

$$x_i \ge 0,\ \ i = 1, 2,\ \ldots,\ m,$$

v не обмежено знаком.

Зазначимо останню умову, що ціна гри v може набувати як додатного, так і від'ємного значення.

Оптимальні стратегії y_1, y_1, ..., y_n гравця $\boldsymbol{B}$ визначаються шляхом розв'язання задачі:

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

$$y_1+y_2+\ldots+y_n=1,\ \ y_j\geq 0,\ \ j=1,2,\ ...,\ n.$$

Використовуючи процедуру, аналогічну наведеній вище для гравця $\boldsymbol{A}$, дійдемо висновку, що задача для гравця $\boldsymbol{B}$ зводиться до такого:

мінімізувати $w = v$

при обмеженнях

$$\sum_{j=1}^{n}a_{ij}y_j \leq v, i=1,2,\ldots,m.$$

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

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

v не обмежено знаком.

Обидві задачі оптимізують ту ж саму (не обмежену знаком) змінну v, що є ціною гри. Причиною цього є те, що задача гравця $\boldsymbol{B}$ є двоїстою щодо задачі гравця $\boldsymbol{A}$. Це означає, що оптимальне розв'язання однієї із задач автоматично визначає оптимальне розв'язання іншої.

Приклад 7.2

Вирішимо таку матричну гру методами лінійного програмування:

		Y_1	Y_2	Y_3	Мінімуми рядків
	A_1	3	−1	−3	−3
	A_2	−2	4	−1	−2
	A_3	−5	−6	2	−6
Максимуми стовпчиків		3	4	2	

Значення ціни гри $\boldsymbol{v}$ знаходиться між −2 і 2.

Задача лінійного програмування для гравця $\boldsymbol{A}$:

максимізувати $z = v$.

при обмеженнях

$$3x_1-2x_2-5x_3-v\geq 0,$$

$$-x_1+4x_2-6x_3-v\geq 0,$$

$$-3x_1 - x_2 + 2x_3 - v \geq 0,$$
$$x_1 + x_2 + x_3 = 1,$$
$$x_1, x_2, x_3 \geq 0,$$
v не обмежено знаком.

Оптимальне розв'язання: $x_1 = 0{,}3945$, $x_2 = 0{,}3119$, $x_3 = 0{,}2936$ і $v = -0{,}9083$.

Відповідні двоїсті змінні $y_1 = -0{,}3211$, $y_2 = -0{,}0826$, $y_3 = -0{,}5963$. Причина того, що змінні y_1, y_2, y_3 не є додатними, як це має бути, полягає в тому, що задача лінійного програмування для гравця $\boldsymbol{A}$ є задачею максимізації з обмеженнями виду «$\geq$». За цих умов, як відомо, двоїсті змінні мають бути від'ємними. Щоб переконатися в тому, перетворимо всі обмеження виду «$\geq$» у задачі лінійного програмування для гравця $\boldsymbol{A}$ в обмеження виду «$\leq$» шляхом множення кожної нерівності на -1. Відповідні двоїсті змінні будуть невід'ємними, як і потрібно. Дійсно, побудова двоїстої задачі безпосередньо із задачі лінійного програмування для гравця $\boldsymbol{A}$ показує, що у двоїстій задачі, що є відповідною задачею лінійного програмування для гравця $\boldsymbol{B}$, мають бути $y_j \leq 0$, але в той же час потрібне виконання умови — $y_1 - y_2 - y_3 - \ldots y_j = 1$, що відповідає вимогам $y_j \geq 0$.

Перетворимо обмеження-нерівності виду «$\geq$» у задачі лінійного програмування для гравця $\boldsymbol{A}$ в обмеження-нерівності виду «$\leq$».

Задача лінійного програмування для гравця $\boldsymbol{B}$: мінімізувати $z = v$ при обмеженнях:

$$3y_1 - y_2 - 3y_3 - v \leq 0,$$
$$-2y_1 + 4y_2 - y_3 - v \leq 0,$$
$$-5y_1 - 6y_2 + 2y_3 - v \leq 0,$$
$$y_1 + y_2 + y_3 = 1,$$
$$y_1 + y_2 + y_3 \geq 0, \text{ а } v \text{ не обмежене знаком.}$$

Оптимальне розв'язання:

$$y_1 = 0{,}3211,\ y_2 = 0{,}0826,\ y_3 = 0{,}5963 \text{ і } v = -0{,}9083.$$

Відповідними двоїстими змінними є:

$$x_1 = -0{,}3945,\ x_2 = -0{,}3119,\ x_3 = -0{,}2936.$$

Зауваження 1. Доцільно розв'язувати задачі лінійного програмування пункту 7.4, застосовуючи пакет аналізу (пошук розв'язання) із програмного продукту *MS Excel*.

7.9. Індивідуальні завдання для рішень матричних ігор методами лінійного програмування

Розв'яжіть таку матричну гру методами лінійного програмування:

— як задачу лінійного програмування для гравця $\boldsymbol{A}$;

— як задачу лінійного програмування для гравця $\boldsymbol{B}$;

— порівняйте результати.

1. $\begin{pmatrix} -3 & 1 & 2 \\ 1 & 4 & 4 \\ -3 & -1 & 2 \end{pmatrix}$. 2. $\begin{pmatrix} -7 & 1 & 2 \\ 1 & 4 & 5 \\ -6 & -1 & 2 \end{pmatrix}$. 3. $\begin{pmatrix} -7 & 3 & 1 \\ -8 & 4 & -6 \\ -6 & -1 & 2 \end{pmatrix}$.

4. $\begin{pmatrix} -2 & 1 & 0 \\ -6 & 3 & -6 \\ -4 & -1 & 2 \end{pmatrix}$. 5. $\begin{pmatrix} -3 & -3 & 2 \\ -6 & 4 & -6 \\ -5 & -1 & 1 \end{pmatrix}$. 6. $\begin{pmatrix} 2 & 3 & 4 \\ -6 & 2 & -5 \\ -1 & 4 & 2 \end{pmatrix}$.

7. $\begin{pmatrix} -5 & -6 & 3 \\ -4 & -2 & -5 \\ -3 & 2 & -1 \end{pmatrix}$. 8. $\begin{pmatrix} -6 & -4 & -2 \\ 4 & -1 & 3 \\ -5 & -2 & -1 \end{pmatrix}$. 9. $\begin{pmatrix} -3 & 2 & 1 \\ 3 & 4 & 5 \\ -5 & -2 & -1 \end{pmatrix}$.

10. $\begin{pmatrix} 4 & 5 & 6 \\ -1 & -2 & 6 \\ 2 & -3 & 2 \end{pmatrix}$. 11. $\begin{pmatrix} -7 & 5 & 6 \\ 2 & 4 & 5 \\ -3 & -2 & 7 \end{pmatrix}$. 12. $\begin{pmatrix} -6 & 5 & -7 \\ 2 & 5 & 5 \\ -4 & -2 & 6 \end{pmatrix}$.

13. $\begin{pmatrix} -5 & -7 & 4 \\ 5 & 3 & 4 \\ -2 & 1 & 6 \end{pmatrix}$. 14. $\begin{pmatrix} 4 & -7 & 4 \\ 5 & -4 & 5 \\ -2 & -2 & 3 \end{pmatrix}$. 15. $\begin{pmatrix} 3 & 2 & 1 \\ -1 & -4 & -3 \\ -3 & -5 & -4 \end{pmatrix}$.

16. $\begin{pmatrix} 4 & -5 & 6 \\ -2 & -4 & -3 \\ -3 & -7 & -4 \end{pmatrix}$. 17. $\begin{pmatrix} -4 & -2 & 3 \\ 5 & 7 & 4 \\ 1 & 2 & -3 \end{pmatrix}$. 18. $\begin{pmatrix} 1 & -1 & -4 \\ -5 & 7 & -3 \\ 1 & 4 & -2 \end{pmatrix}$.

19. $\begin{pmatrix} 4 & -7 & 4 \\ 5 & -4 & 5 \\ -2 & -2 & 3 \end{pmatrix}$. 20. $\begin{pmatrix} -5 & 1 & 2 \\ 2 & 7 & 4 \\ -4 & -1 & 3 \end{pmatrix}$. 21. $\begin{pmatrix} -3 & 4 & 2 \\ 2 & 7 & 4 \\ -4 & -1 & -3 \end{pmatrix}$.

22. $\begin{pmatrix} 6 & 3 & 3 \\ -1 & 6 & 2 \\ -4 & 2 & 3 \end{pmatrix}$. 23. $\begin{pmatrix} 6 & 3 & 3 \\ -1 & -6 & 2 \\ -5 & 1 & 2 \end{pmatrix}$. 24. $\begin{pmatrix} 5 & 1 & 4 \\ -1 & -3 & 3 \\ 4 & 1 & 2 \end{pmatrix}$.

25. $\begin{pmatrix} 5 & 2 & 4 \\ -1 & -7 & 3 \\ -2 & 1 & 2 \end{pmatrix}$. 26. $\begin{pmatrix} 5 & 6 & 4 \\ -1 & 5 & 3 \\ 6 & 7 & 5 \end{pmatrix}$. 27. $\begin{pmatrix} -6 & -3 & -5 \\ -4 & 5 & 2 \\ -5 & -7 & 3 \end{pmatrix}$.

28. $\begin{pmatrix} -6 & -3 & -5 \\ -4 & 5 & 2 \\ -5 & 7 & 3 \end{pmatrix}$. 29. $\begin{pmatrix} -6 & 3 & 5 \\ -4 & 5 & -2 \\ -7 & 7 & -3 \end{pmatrix}$. 30. $\begin{pmatrix} -3 & 1 & 4 \\ -4 & 2 & -1 \\ -6 & 5 & -3 \end{pmatrix}$.

Модуль 8
МАРКІВСЬКІ ПРОЦЕСИ

8.1. Поняття марківського випадкового процесу

Процес роботи системи масового обслуговування (СМО) — це *випадковий процес*.

Під *випадковим (ймовірним* або *стохастичним) процесом* розуміємо 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 ... можна заздалегідь перелічити, а перехід системи з одного стану в інший відбувається миттєво (стрибком). Процес називається *процесом із безперервним часом*, якщо моменти можливих переходів системи із одного стану в інший є випадковими, а не фіксованими заздалегідь. Процес роботи СМО являє собою випадковий процес із дискретними станами і безперервним часом. Це означає, що стан СМО змінюється стрибком у випадкові моменти появи якихось подій (наприклад, надходження нової заявки, закінчення обслуговування).

Математичний аналіз роботи СМО істотно спрощується, якщо процес цієї роботи — марківський. Випадковий процес називається *марківським* або *випадковим процесом без наслідку*, якщо для будь-якого моменту часу t_0 імовірність характеристики процесу в майбутньому залежить тільки від його стану в певний момент t_0 і не залежать від того, коли і як система його набула.

Приклад марківського процесу. Система S — лічильник у таксі. Стан системи в момент t характеризується кількістю кілометрів (десятих часток кілометрів), пройдених автомобілем до певного моменту. Нехай у момент t_0 лічильник показує S_0. Імовірність того, що в момент $t > t_0$ лічильник покаже ту чи іншу кількість кілометрів (точніше, вартість проїзду) 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 в шахи; система S — група шахових фігур. Стан системи характеризується кількістю фігур супротивника, що збереглися на дошці в момент t_0. Імовірність того, що в момент $t > t_0$ матеріальна перевага буде на боці одного із супротивників, залежить у першу чергу від того, в якому стані знаходяться система у певний момент t_0, а не від того, коли і в якій послідовності зникли фігури з дошки до моменту t_0.

У ряді випадків передісторію розглянутих процесів можна просто не враховувати і застосовувати для їхнього вивчення марківські моделі.

Аналізуючи випадкові процеси з дискретними станами зручно користуватися геометричною схемою — так званим *графом станів*. Звичайно стани системи зображують прямокутниками (колами), а можливі переходи з одного стану в інший — стрілками (орієнтованими дугами), що з'єднують стани.

Приклад 8.1

Побудувати граф станів такого випадкового процесу: пристрій S складається з двох вузлів, кожний із яких у випадковий момент часу може вийти з ладу, після чого миттєво починається ремонт вузла, що продовжується заздалегідь невідомий випадковий час.

Розв'язок. Можливі стани системи: S_0 — обидва вузли справні; S_1 — перший вузол ремонтується, другий справний; S_2 — другий вузол ремонтується, перший справний; S_3 — обидва вузли ремонтуються. Граф системи наведений на рис. 8.1.

Стрілка, спрямована, наприклад, із S_0 у S_1, означає перехід системи в момент відновлення першого вузла, з S_1 у S_0 — перехід у момент закінчення ремонту цього вузла.

На графі відсутні стрілки з S_0 у S_3 і з S_1 у S_2. Це означає, що виходи вузлів із ладу передбачають незалежними один від одного, і наприклад, імовірністю одночасного виходу з ладу двох вузлів (перехід з S_0 у S_3) або одночасного закінчення ремонтів двох вузлів (перехід з S_3 у S_0) можна знехтувати.

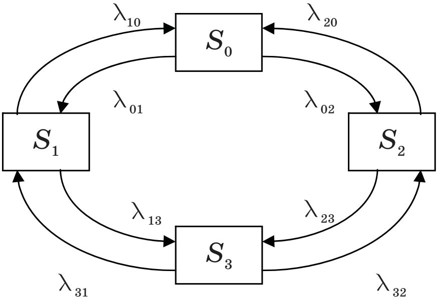


Рис. 8.1. Граф станів випадкового процесу

Для математичного опису марківського випадкового процесу з дискретними станами і безперервним часом, що відбувається в СМО, розглянемо одне з важливих понять теорії ймовірностей — поняття *потоку подій.*

8.2. Потоки подій

Визначення 8.1. Під потоком подій розуміють послідовність однорідних подій, що надходять одна за одною в якісь випадкові моменти часу (наприклад, потік викликів на телефонній станції, потік відмовлень ЕОМ, потік покупців тощо).

Визначення 8.2. Потік характеризується інтенсивністю λ — частотою появи подій або середнім числом подій, що надходять у СМО за одиницю часу.

Визначення 8.3. ***Регулярним*** *називається потік подій, що надходять одна за одною через визначені рівні проміжки часу*. Наприклад, потік виробів на конвеєрі складального цеху (з постійною швидкістю руху) є регулярним.

Визначення 8.4. *Потік подій називається* ***стаціонарним****, якщо його ймовірнісні характеристики не залежать від часу*. Зокрема, інтенсивність стаціонарного потоку є величина постійна: $\lambda(t) = \lambda$. Наприклад, потік автомобілів на міському проспекті не є стаціонарним протягом доби, але цей потік можна вважати стаціонарним, скажемо, у години пік. Зазначимо, що в останньому разі фактична кількість автомобілів за одиницю часу (наприклад, щохвилини) може помітно відрізнятися одна від одної, але середня їхня кількість буде постійна і не залежатиме від часу.

Визначення 8.5. *Потік подій називається* ***потоком без наслідку****, якщо для будь-яких двох непересічних ділянок часу* τ_1 *і* τ_2 *число подій, що припадають на один з них, не залежить від кількості подій, що припадають на інші*. Наприклад, потік пасажирів, що входять у метро, практично не має наслідку. А, скажемо, потік покупців, що відходять з покупками від прилавка, має наслідок (хоча б тому, що інтервал часу між окремими покупцями не може бути менше, ніж мінімальний час обслуговування кожного з них).

Визначення 8.6. *Потік подій називається* ***одинарним****, якщо ймовірність потрапляння на малий (елементарний)* *відтинок часу Δt двох і більше подій нескінченно мала порівнянно з імовірністю влучення однієї події.* Іншими словами, потік подій ординарний, якщо події з'являються в ньому поодинці, а не групами. Наприклад, потік потягів, що підходять до станції, одинарний, а потік вагонів не одинарний.

Визначення 8.7. *Потік подій називається* ***найпростішим*** *(або* ***стаціонарним пуассонівським****), якщо він одночасно стаціонарний, одинарний і не має наслідку.* Назва «найпростіший» застосована тому, що такі потоки мають найпростіший математичний опис.

8.3. Рівняння Колмогорова. Граничні ймовірності станів

Розглянемо математичний опис марківського процесу з дискретними станами і безперервним часом на прикладі 8.1 випадкового процесу з прикладу 8.1, граф якого зображений на рисунку 8.1. Вважаємо, 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, перехід системи зі стану S_0 у S_1 відбуватиметься під впливом потоку відмовлень першого вузла, а зворотний перехід зі стану S_1 у S_0 — під впливом потоку «закінчень ремонтів» першого вузла тощо. Граф станів системи з проставленими біля стрілок інтенсивностями будемо називати *розміченим*. Ця система S має чотири можливих стани: S_0, S_1, S_2, S_3.

Імовірністю i-го стану називається імовірність $p_i(t)$, що в момент t система буде перебувати в стані S_i. Очевидно, що для будь-якого моменту t сума ймовірностей усіх станів дорівнює одиниці:

$$\sum_{i=0}^{3} p_i(t) = 1. \tag{8.1}$$

Доведемо це твердження:

Дійсно: $A_i \xrightarrow{def}$ подія в тім, що система знаходиться в стані $S_i \Rightarrow \sum_{i=0}^{3} A_i$ — достовірна подія $\xrightarrow{def} P(\sum_{i=0}^{3} A_i) = 1,$

$[A_i$ — неспільна подія$] \Rightarrow P(\sum_{i=0}^{3} A_i) = \sum_{i=0}^{3} P(A_i) = 1; \ (P(A_i) = P_i(t)),$

що і потрібно було довести.

Математик ***Колмогоров*** *отримав систему диференціальних рівнянь:*

$$\begin{cases} p_0' = \lambda_{10} p_1 + \lambda_{20} p_2 - (\lambda_{01} + \lambda_{02}) p_0; \\ p_1' = \lambda_{01} p_0 + \lambda_{31} p_3 - (\lambda_{10} + \lambda_{13}) p_1; \\ p_2' = \lambda_{02} p_0 + \lambda_{32} p_3 - (\lambda_{20} + \lambda_{23}) p_2; \\ p_3' = \lambda_{13} p_1 + \lambda_{23} p_2 - (\lambda_{31} + \lambda_{32}) p_3. \end{cases} \qquad (8.2)$$

Словесно: *У лівій частині кожного з них стоїть похідна імовірності і-го стану. У правій частині — сума добутків імовірностей усіх станів (з яких входять стрілки в цей стан) на інтенсивності відповідних потоків подій, мінус сумарна інтенсивність усіх потоків, що виводять систему із цього стану, помножена на імовірність цього (і-го) стану.*

Рівняння Колмогорова дають можливість знайти всі ймовірності станів як *функції часу*. Виокремлюють імовірності системи $p_i(t)$ у *граничному стаціонарному режимі*, тобто при $t \to \infty$, що називаються *граничними* (або *фінальними*) ймовірностями станів.

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

Гранична імовірність стану S_i показує *середній відносний час перебування системи в цьому стані*. Наприклад, якщо гранична імовірність стану S_0, тобто $p_0 = 0{,}5$, то це означає, що в середньому половину часу система знаходиться в стані S_0.

Якщо граничні імовірності постійні, то замінимо в рівняннях Колмогорова їхні похідні нульовими значеннями, одержимо систему лінійних алгебричних рівнянь, що описують стаціонарний режим. Для системи S із графом станів, зображеному на рисунку 8.1, така система рівнянь має вигляд:

$$\begin{cases} (\lambda_{01} + \lambda_{02}) p_0 = \lambda_{10} p_1 + \lambda_{20} p_2, \\ (\lambda_{10} + \lambda_{13}) p_1 = \lambda_{01} p_0 + \lambda_{31} p_3, \\ (\lambda_{20} + \lambda_{23}) p_2 = \lambda_{02} p_0 + \lambda_{32} p_3, \\ (\lambda_{31} + \lambda_{32}) p_3 = \lambda_{13} p_1 + \lambda_{23} p_2. \end{cases} \qquad (8.3)$$

Систему (8.3) можна скласти безпосередньо за розміченим графом станів, якщо керуватися правилом, відповідно до якого *ліворуч у рівняннях стоїть гранична ймовірність його стану* p_i, *помножена на сумарну інтенсивність усіх потоків, що ведуть із цього стану, а праворуч — сума добутків інтенсивностей усіх потоків, що входять у i-ий стан, на імовірності тих подій, з яких ці потоки виходять.*

8.4. «Марківський процес», «Рівняння Колмогорова»

Задача 8.1

Знайти граничні ймовірності для системи S із прикладу 8.1, граф станів якої наведений на рисунку 8.1, при $\lambda_{01} = 1$, $\lambda_{02} = 2$, $\lambda_{10} = 2$, $\lambda_{13} = 2$, $\lambda_{20} = 3$, $\lambda_{23} = 1$, $\lambda_{31} = 3$, $\lambda_{32} = 2$.

Розв'язання. Система алгебричних рівнянь, що описують стаціонарний режим для цієї системи, має вигляд (8.3) або:

$$\begin{cases} 3p_0 = 2p_1 + 3p_2, \\ 4p_1 = p_0 + 3p_3, \\ 4p_2 = 2p_0 + 2p_3, \\ p_0 + p_1 + p_2 + p_3 = 1. \end{cases} \tag{8.4}$$

Тут ми замість «зайвого» рівняння системи (8.3) записали нормувальну умову (8.1).

Розв'яжемо систему (8.4) і одержимо такі результати:

$$p_0 = 0{,}40,\ p_1 = 0{,}20,\ p_2 = 0{,}27,\ p_3 = 0{,}13,$$

тобто в граничному, стаціонарному режимі система S у середньому 40% часу перебуватиме у стані S_0 (обидва вузли справні), 20% — у стані S_1 (перший вузол ремонтується, інший працює), 27% — у стані S_2 (другий вузол ремонтується, перший працює) і 13% часу — у стані S_3 (обидва вузли ремонтуються).

Задача 8.2

Знайти середній чистий дохід від експлуатації в стаціонарному режимі системи S в умовах задачі 8.1, якщо відомо, що в одиницю часу справна робота першого і другого вузлів приносить прибуток відповідно в 10 і 6 грошових одиниць, а їхній ремонт вимагає витрат відповідно в 4 і 2 грошові одиниці.

Задача 8.3

Оцінити економічну ефективність наявної можливості зменшення вдвічі середнього часу ремонту кожного з вузлів, якщо при цьому доведеться вдвічі збільшити витрати на ремонт кожного вузла (в одиницю часу).

Розв'язання задачі 8.2:

Таблиця 8.1

Дані задачі 8.2

№ вузла	Дохід	Витрата
1 вузол (автомат)	10	4
2 вузол (автомат)	6	2

Випишемо граничні ймовірності з відповіді задачі 8.1 на рисунку 8.1 і знайдемо доходи і витрати вузлів:

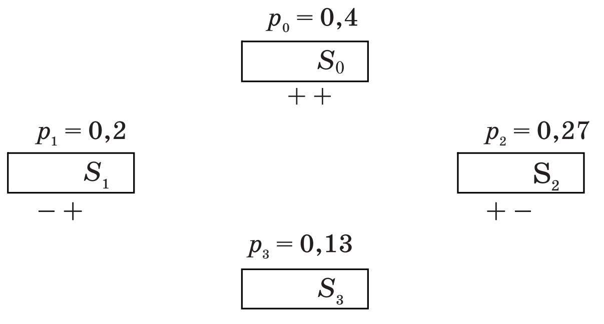


Рис. 8.2. Стан системи з граничними ймовірностями

Прибутки 1-го автомата:
$(p_0 + p_2) \cdot 10 = (0,4 + 0,27) \cdot 10 = 0,67 \cdot 10 = 6,7$
Прибутки 2-го автомата:
$(p_0 + p_1) \cdot 6 = (0,4 + 0,2) \cdot 6 = 0,6 \cdot 6 = 3,6$

$\Sigma = 10,3$ **прибутки**

Витрати 1-го автомата:
$(p_1 + p_3) \cdot 4 = (0,2 + 0,13) \cdot 4 = 0,33 \cdot 4 = 1,32$
Витрати 2-го автомата:
$(p_2 + p_3) \cdot 2 = (0,27 + 0,13) \cdot 2 = 0,4 \cdot 2 = 0,8$

$\Sigma = 2,12$ **витрати**

Відповідь: Загальний прибуток системи дорівнює

прибуток – витрати = 10,3 – 2,12 = **8,18** грошових одиниць.

Розв’язання задачі 8.3:

Крок 1

Маємо початковий граф і таблицю «прибуток-витрати» (задача 8.3).

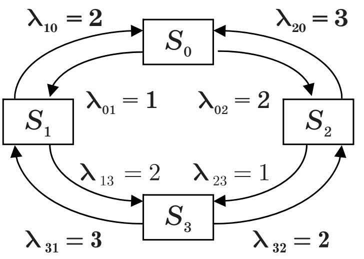


Рис. 8.3. Рис. до задачі 8.3

Зменшення вдвічі середнього часу ремонту призводить до збільшення частоти повернення до роботи (див. зовнішнє «кільце» рис. 8.2), тобто:

$$\begin{cases} \lambda_{10} = 4;\ \lambda_{20} = 6;\ \lambda_{31} = 6;\ \lambda_{32} = 4 \\ \text{Внутрішнє «кільце» графа залишилося попереднє:} \\ \lambda_{01} = 1;\ \lambda_{02} = 2;\ \lambda_{13} = 2;\ \lambda_{23} = 1 \end{cases}$$

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

Крок 2: Збільшення витрат на ремонт удвічі:

Таблиця 8.2

Змінені дані задачі 8.3

№ вузла	Дохід	Витрати
1 вузол (автомат)	10	**8**
2 вузол (автомат)	6	**4**

Розраховані частоти повернення до роботи покажемо на рис. 8.4.

Далі аналогічно до задачі 8.2 знайдемо доходи і витрати вузлів для графа рис. 8.3.

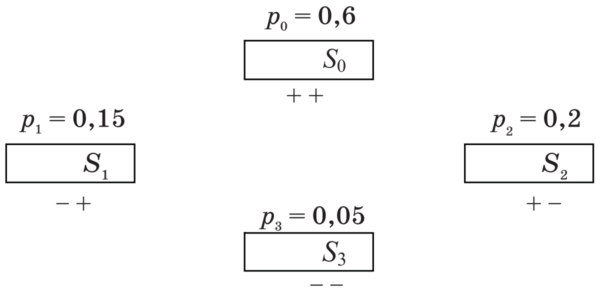


Рис. 8.4. Стан системи з граничними ймовірностями

Прибутки 1-го автомата:
$(p_0 + p_2) \cdot 10 = (0,6 + 0,2) \cdot 10 = 0,8 \cdot 10 = 8$
Прибутки 2-го автомата:
$(p_0 + p_1) \cdot 6 = (0,6 + 0,15) \cdot 6 = 0,75 \cdot 6 = 4,5$

$\sum = 12,5$ **прибутки**

Витрати 1-го автомата:
$(p_1 + p_3) \cdot 8 = (0,15 + 0,05) \cdot 8 = 0,2 \cdot 8 = 1,6$
Витрати 2-го автомата:
$(p_2 + p_3) \cdot 4 = (0,2 + 0,05) \cdot 4 = 0,25 \cdot 4 = 1$

$\sum = 2,6$ **витрати**

Загальний прибуток системи = прибуток − витрати = 12,5 − − 2,6 = **9,9** грошових одиниць.

Відповідь: Оскільки прибуток у задачі **8.3** більше прибутку в задачі **8.2** ($\approx 20\%$), то економічна доцільність прискорення ремонтів з одночасним збільшенням витрат на ремонт очевидно вигідна.

8.5. Індивідуальні завдання «Марківський процес», «Рівняння Колмогорова»

А. Реалізуйте демонстраційну задачу 8.1 за частот, наведених у таблиці 8.3:

Таблиця 8.3

Частоти подій

№ варіанту	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

Закінчення табл. 8.3

№ варіанта	λ_{10}	λ_{01}	λ_{20}	λ_{02}	λ_{13}	λ_{31}	λ_{32}	λ_{23}
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

Б. Реалізуйте демонстраційну задачу 8.2. за таких умов:

Завдання «прибуток — витрата» вузлів

Варіант 1

	Прибуток	Витрата
1 вузол	10	4
2 вузол	5	1

Варіант 2

	Прибуток	Витрата
1 вузол	9	5
2 вузол	4	3

Варіант 3

	Прибуток	Витрата
1 вузол	8	3
2 вузол	5	2

Варіант 4

	Прибуток	Витрата
1 вузол	9	7
2 вузол	6	4

Варіант 5

	Прибуток	Витрата
1 вузол	6	2
2 вузол	9	8

Варіант 6

	Прибуток	Витрата
1 вузол	7	5
2 вузол	8	4

Варіант 7

	Прибуток	Витрата
1 вузол	8	7
2 вузол	7	6

Варіант 8

	Прибуток	Витрата
1 вузол	5	3
2 вузол	4	1

Варіант 9

	Прибуток	Витрата
1 вузол	9	2
2 вузол	10	3

Варіант 10

	Прибуток	Витрата
1 вузол	8	4
2 вузол	5	3

Варіант 11

	Прибуток	Витрата
1 вузол	3	2
2 вузол	2	1

Варіант 12

	Прибуток	Витрата
1 вузол	5	3
2 вузол	2	1

Варіант 13

	Прибуток	Витрата
1 вузол	4	3
2 вузол	3	2

Варіант 14

	Прибуток	Витрата
1 вузол	7	6
2 вузол	5	3

Варіант 15

	Прибуток	Витрата
1 вузол	7	5
2 вузол	4	2

Варіант 16

	Прибуток	Витрата
1 вузол	4	2
2 вузол	7	3

Варіант 17

	Прибуток	Витрата
1 вузол	5	3
2 вузол	10	7

Варіант 18

	Прибуток	Витрата
1 вузол	7	5
2 вузол	10	4

Варіант 19

	Прибуток	Витрата
1 вузол	10	5
2 вузол	3	2

Варіант 20

	Прибуток	Витрата
1 вузол	9	5
2 вузол	6	4

Варіант 21

	Прибуток	Витрата
1 вузол	10	7
2 вузол	9	4

Варіант 22

	Прибуток	Витрата
1 вузол	4	3
2 вузол	7	5

Варіант 23

	Прибуток	Витрата
1 вузол	5	2
2 вузол	7	4

Варіант 24

	Прибуток	Витрата
1 вузол	8	7
2 вузол	5	2

Варіант 25

	Прибуток	Витрата
1 вузол	5	4
2 вузол	8	3

Варіант 26

	Прибуток	Витрата
1 вузол	5	3
2 вузол	6	4

Варіант 27

	Прибуток	Витрата
1 вузол	10	7
2 вузол	9	3

Варіант 28

	Прибуток	Витрата
1 вузол	7	6
2 вузол	3	2

Варіант 29

	Прибуток	Витрата
1 вузол	3	1
2 вузол	7	5

Варіант 30

	Прибуток	Витрата
1 вузол	10	6
2 вузол	7	5

В. Реалізуйте демонстраційну задачу 8.3, за таких умов:

Таблиця 8.4

Зміна умов задачі 8.3

№ варіанта	Зменшення середнього часу ремонту кожного з вузлів *у*	Збільшення витрат на ремонт кожного з вузлів *у*
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
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

Модуль 9
СИСТЕМИ МАСОВОГО ОБСЛУГОВУВАННЯ

9.1. Процеси обслуговування

Досліджуючи операції, часто доводиться стикатися із системами, призначеними для багаторазового використання в процесі розв'язування однотипних задач. Такі процеси одержали назву *процесів обслуговування*, а системи — *систем масового обслуговування (СМО)*. Прикладами таких систем є телефонні системи, ремонтні майстерні, магазини, перукарні тощо.

Кожна СМО складається з певної кількості обслуговувальних 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* і *багатоканальні*.

Заявки надходять у СМО звичайно не регулярно, а випадково, що утворює так званий *випадковий потік заявок (вимог)*. Обслуговування заявок також триває якийсь випадковий час. 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 залишають СМО необслугованими), в інші ж періоди СМО працює з недовантаженням або простоює.

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

Як *показники ефективності* СМО використовуються: середня кількість заявок (тут і надалі середні величини розуміються як математичні очікування відповідних випадкових величин), що їх обслуговують в одиницю часу; середня кількість заявок у черзі; середній час чекання обслуговування; імовірність відмовлення в обслуговуванні без чекання; імовірність того, що кількість заявок у черзі перевищить визначене значення тощо.

СМО поділяють на два основних типи (класи): СМО *з відмовленнями* і СМО *з чеканням (чергою)*. У СМО з відмовленнями заявка, що надійшла в момент, коли всі канали зайняті, одержує