

ФАКУЛЬТЕТ УПРАВЛІННЯ ПРОЦЕСАМИ ПЕРЕВЕЗЕНЬ

Кафедра вищої математики

**МАТЕМАТИЧНІ МЕТОДИ В ЗАДАЧАХ
УПРАВЛІННЯ ТРАНСПОРТНИМИ
СИСТЕМАМИ**

Конспект лекцій з дисципліни

***«МАТЕМАТИЧНІ МЕТОДИ В ЗАДАЧАХ УПРАВЛІННЯ
ТРАНСПОРТНИМИ СИСТЕМАМИ»***

Харків - 2013

Математичні методи в задачах управління транспортними системами: Конспект лекцій / Н.С. Юрчак, С.Д. Бронза,

О.О. Гончарова. – Харків: УкрДАЗТ, 2013. – 65 с.

Конспект лекцій призначений для студентів-магістрів денної форми навчання спеціальності “Організація перевезень і управління на транспорті (залізничний транспорт).

Конспект лекцій розглянуто й рекомендовано до друку на засіданні кафедри вищої математики УкрДАЗТ, протокол № 4 від 13 листопада 2012 р.

Рецензент

проф. В.О. Золотарьов

МАТЕМАТИЧНІ МЕТОДИ В ЗАДАЧАХ УПРАВЛІННЯ
ТРАНСПОРТНИМИ СИСТЕМАМИ

Конспект лекцій з дисципліни

*«МАТЕМАТИЧНІ МЕТОДИ В ЗАДАЧАХ УПРАВЛІННЯ
ТРАНСПОРТНИМИ СИСТЕМАМИ»*

Відповідальний за випуск Юрчак Н.С.

Редактор Решетилова В.В.

Підписано до друку 05.03.13 р.

Формат паперу 60x84 1/16. Папір писальний.

Умовн.-друк.арк. 2,0. Тираж 50. Замовлення №

Видавець та виготовлювач Українська державна академія залізничного транспорту,
61050, Харків-50, майдан Фейербаха, 7.
Свідоцтво суб'єкта видавничої справи ДК № 2874 від 12.06.2007 р.

**УКРАЇНСЬКА ДЕРЖАВНА АКАДЕМІЯ
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ**

Кафедра вищої математики

**МАТЕМАТИЧНІ МЕТОДИ В ЗАДАЧАХ УПРАВЛІННЯ
ТРАНСПОРТНИМИ СИСТЕМАМИ**

Конспект лекцій

**з дисципліни «Математичні методи в задачах управління
транспортними системами»**

**для студентів-магістрів спеціальності «Організація
перевезень і управління на транспорті (залізничний
транспорт)»**

Харків – 2012

Математичні методи в задачах управління транспортними системами: Конспект лекцій / Н.С. Юрчак, С.Д. Бронза, О.О. Гончарова. – Харків: УкрДАЗТ, 2013. – 65 с.

Конспект лекцій призначений для студентів-магістрів денної форми навчання спеціальності “Організація перевезень і управління на транспорті (залізничний транспорт).

Конспект лекцій розглянуто й рекомендовано до друку на засіданні кафедри вищої математики УкрДАЗТ, протокол № 4 від 13 листопада 2012 р.

Рецензент
проф. В.О. Золотарьов

План дисципліни “Математичні методи в задачах управління транспортними системами”

1 Класифікація та зміст задач.

2 Класичні приклади формулювання загальних задач лінійного програмування як задач управління транспортними системами (УТС).

3 Задачі нелінійного програмування: загальна теорія, опис постановки задач управління запасами, приклади використання градієнтних методів.

4 Елементи теорії графів та огляд задач УТС у сітьових постановках:

а) транспортні задачі;

б) задачі про максимальний потік на сітках;

в) задачі сітьового планування та управління на залізничних станціях;

г) сітки Петрі.

5 Задачі динамічного програмування, особливості та типові приклади:

а) задачі найвигіднішого режиму ведення потяга на відрізку шляху та задача про найкоротший шлях;

б) задача розподілу ресурсів;

в) динамічна задача прийняття рішень.

6 Задачі теорії масового обслуговування в широкому колі їх постановок та прикладів; метод динаміки середніх.

7 Основи класичної логіки.

ВСТУП

Конспект лекцій з дисципліни “Математичні методи в задачах управління транспортними системами” призначений для студентів-магістрів 5 курсу факультету «Управління процесами перевезень».

Він є логічним продовженням, а також поширенням дисципліни “Дослідження операцій в транспортних системах”, курс лекцій з якої читається студентам 2 курсу факультету УПП.

Конспект лекцій більш детально висвітлює теми, які розглядатимуться на лекціях, а також надає можливість самостійно опрацювати лекційний матеріал.

КЛАСИФІКАЦІЯ ТА ЗМІСТ ЗАДАЧ

Для розв’язання практичних задач дослідження операцій у транспортних формулюваннях використовується арсенал математичних методів, частина яких була розглянута у відповідному початковому курсі, а частина буде викладатися в процесі ознайомлення з новим матеріалом.

До таких математичних засобів відносяться: *математичні методи оптимізації*, а саме *лінійне програмування, нелінійне програмування, динамічне програмування; теорія графів* для розв’язання деяких задач, що відносяться до *транспортних моделей*, а також до *задач сітьового планування та управління; теорія масового обслуговування; основи класичної логіки*, що складають фундамент для вивчення більш сучасного продовження цього розділу – нечіткої логіки.

1 ПРИКЛАДИ ВИКОРИСТАННЯ МОДЕЛЕЙ ЛІНІЙНОГО ПРОГРАМУВАННЯ ДЛЯ РОЗВ'ЯЗАННЯ ЗАДАЧ УПРАВЛІННЯ НА ЗАЛІЗНИЧНОМУ ТРАНСПОРТІ

Приклад 1.1 Класичним прикладом загальної задачі лінійного програмування є задача *оптимального використання ресурсів*, яка може бути сформульована так. Підприємство має m видів ресурсів (сировина, обладнання, робоча сила та інше) у розмірах відповідно b_1, b_2, \dots, b_m ($b_i, i=1 \div m$) та випускає продукцію n типів. Одиниця продукції кожного виду забезпечує підприємству прибуток відповідно c_1, c_2, \dots, c_n ($c_j, j=1 \div n$). Відомі витрати кожного виду ресурсів на виробництво одиниці продукції кожного типу a_{ij} ($i=1 \div m; j=1 \div n$). Необхідно визначити такі розміри випуску продукції кожного типу x_j , які забезпечать підприємству максимальний прибуток.

Побудова математичної моделі та методи розв'язання (графічний для двовимірної задачі та симплекс-метод) розглянуті в основному курсі дослідження операцій. Слід відмітити наявність *двоїстої задачі лінійного програмування*, яка використовується для аналізу оптимальних розв'язків та дослідження їх чутливості до зміни кількості дефіцитних ресурсів, тобто таких, запас яких повністю використовується для виробництва продукції.

Приклад 1.2 *Задача про побудову моделі оптимального використання вагонів під завантаження або розвантаження* може бути сформульована так. На станції необхідно розвантажити маршрут однорідного вантажу, який складається з b_1 вагонів. Всі необхідні дії щодо розвантаження виконує один локомотив, який працює b_2 години на добу. Вагони розташовуються на трьох вантажних фронтах, які можуть умістити відповідно b_3, b_4, b_5 вагонів. Відомі витрати локомотиво-годин маневрової роботи на один вагон для кожного вантажного фронту a_1, a_2, a_3 . Визначена платня за розвантаження одного вагона, яка різниться для кожного фронту розвантаження з причини їх різного технічного оснащення і складає c_1, c_2, c_3 грошових одиниць за вагон. Необхідно

розподілити вагони по вантажних фронтах так, щоб забезпечити максимальне розвантаження та максимальний прибуток.

Приклад 1.3 Значний обсяг задач, математичною моделлю яких є модель лінійного програмування, складають задачі забезпечення *максимального пасажиропотоку*. На план формування категорій потягів для здійснення означеної мети впливає багато факторів, які призводять до значного ускладнення математичної моделі. Розглянемо приклад у спрощеній постановці, де, зокрема, не враховуються можливості проїзду пасажирів не до кінцевих, а до проміжних пунктів, а також заповнення вільних місць на проміжних станціях.

З вокзалу можна щоденно відправляти пасажирські та швидкі потяги. Наявний парк вагонів в цілому складається з b_1 поштово-вантажних, b_2 загальних, b_3 плацкартних, b_4 купейних та b_5 вагонів-люкс. Кількість відповідних вагонів у пасажирському потязі складає a_{11} , a_{21} , a_{31} , a_{41} , a_{51} одиниць, а у швидкому - a_{12} , a_{22} , a_{32} , a_{42} , a_{52} одиниць. Відома загальна кількість пасажирських місць у кожному типі вагонів c_{01} , c_{02} , c_{03} , c_{04} , c_{05} . Потрібно побудувати математичну модель задачі, використавши яку, можна обрати таке співвідношення між кількістю пасажирських та швидких потягів, аби число пасажирів, яке можна відправити щоденно, було б максимальним.

До наведених прикладів можна додати ще достатню кількість задач з моделлю лінійного програмування, а саме задачу про *вибір зонних станцій на приміських залізничних ділянках* для раціонального використання рухомого складу в тому сенсі, щоб частину потягів відправляти не до останньої станції, а до станцій, що розташовані всередині ділянки, – зонних. Цільова функція відображає вимогу максимального пробігу всіх приміських потягів для створення найліпших умов проїзду пасажирів, а обмеження враховують: а) неперевищення економічно обґрунтованої планової величини максимально припустимого рівня загального пробігу потягів; б) кількість потягів на кожен зону не повинна бути меншою за мінімально припустимий розмір руху. Розв'язком задачі є вибір зонних станцій на ділянках шляху та розподіл розмірів руху по зонах.

Також розповсюдженим типом задач лінійного програмування на залізничному транспорті є так звані *розподільчі задачі*, наприклад, *задача розподілу порожніх вагонів під завантаження*. Необхідно забезпечити максимальне завантаження мінімальною кількістю порожніх вагонів, що і відображає цільова функція задачі. Обмеження включають в себе вимоги: а) неперевіщення ресурсів порожніх вагонів; б) необхідність виконання заявок на завантаження.

Слід відмітити, що частка пропонованих задач лінійного програмування природно потребує цілочислових значень оптимального розв'язку. Така додаткова вимога або задовольняється округлюванням отриманих розв'язків, або застосуванням спеціальних методів *цілочислового програмування*, які розглядалися в основному курсі дослідження операцій.

Опис методів розв'язання задач лінійного програмування наданий, наприклад, у методичних вказівках [1, 2].

2 ЗАДАЧІ НЕЛІНІЙНОГО ПРОГРАМУВАННЯ

Наведені задачі лінійного програмування є частиною задач класичної оптимізації, які перетворюються в *задачі нелінійного програмування*, якщо математичну модель задачі складають нелінійні функції. *Методи розв'язання* задач нелінійного програмування можна класифікувати як *прямі та непрямі*. Прикладами прямих методів є *градієнтні алгоритми*, в яких пошук максимуму (мінімуму) в оптимізаційній задачі ведеться у напрямі найшвидшого збільшення (зменшення) значення цільової функції. У непрямих методах початкова задача оптимізації замінюється допоміжною, оптимальний розв'язок якої сприймається як розв'язок початкової. Частковими випадками таких методів є *опукле програмування*, зокрема *квадратичне*.

Для предметного розгляду теорії методів розв'язання задач нелінійного програмування наведемо приклад задачі, що приводить до математичної моделі нелінійного програмування.

Підприємство може випускати два типи виробів A та B . На їх виготовлення витрачається три типи ресурсів з відомими запасами b_1 , b_2 , b_3 . Планові норми витрат ресурсів на одиницю

продукції складають a_{ij} од. ($i=1\div 3, j=1,2$). Відомими також у прийнятій методиці розрахунків вважаються планова собівартість виробів c_j та їх оптові ціни d_j ($j=1,2$). Через наявність браку в процесі виробництва витрати ресурсів залежать від обсягу x_j ($j=1,2$) виробництва виробів та в першому наближенні відбиваються лінійною функцією $a_{ij} + x_j$, а собівартість продукції – функцією $c_j + kx_j$, $k < 1$. Вироби можуть випускатися в будь-яких співвідношеннях, оскільки збут їх забезпечений. Необхідно скласти план випуску продукції, який би забезпечив отримання максимального прибутку.

Для складання математичної моделі задачі математичного програмування формалізуємо основну вимогу задачі у вигляді її *цільової функції*. А саме: від реалізації одиниці виробу A підприємство отримає прибуток $d_1 - (c_1 + kx_1)$ грошових одиниць, одиниці виробу B – прибуток $d_2 - (c_2 + kx_2)$ гр. од. Загальний прибуток підприємства складе (із зазначенням основної вимоги задачі)

$$z = (d_1 - (c_1 + kx_1))x_1 + (d_2 - (c_2 + kx_2))x_2 \rightarrow \max. \quad (2.1)$$

Далі формулюємо *обмеження задачі* у вигляді системи нерівностей (або рівностей). На виготовлення виробів A та B , що заплановані підприємством, буде витрачено:

ресурсу I - $(a_{11} + x_1)x_1 + (a_{12} + x_2)x_2$ одиниць,

ресурсу II - $(a_{21} + x_1)x_1 + (a_{22} + x_2)x_2$ одиниць,

ресурсу III - $(a_{31} + x_1)x_1 + (a_{32} + x_2)x_2$ одиниць.

Враховуючи обмеженість кількості кожного ресурсу, формалізована система обмежень відбивається системою трьох нерівностей

$$(a_{i1} + x_1)x_1 + (a_{i2} + x_2)x_2 \leq b_i, \quad i = 1 \div 3. \quad (2.2)$$

Додаючи до двох частин постановки задачі природну вимогу *невід'ємності складових плану*

$$x_1 \geq 0, x_2 \geq 0, \quad (2.3)$$

отримуємо повну математичну модель задачі, яка зводиться до знаходження невід'ємних компонент оптимального плану x_1^* , x_2^* , які задовольняють *нелінійні обмеження* (2.2) та надають максимум *нелінійній цільовій функції* (2.1).

Типовим прикладом задач, що призводять до математичної моделі нелінійного програмування, є *задачі управління запасами*. Під запасами розуміють той вид продукції, який підлягає *збереженню* перед його використанням. Таким чином, витрати на створення запасів складаються з витрат на їх виробництво, збереження, а також можливі штрафні санкції за їх нестачу. Основне протиріччя задачі підлягає такому поясненню. При *неперервному процесі виробництва*, яке є *дешевшим*, ніж періодичний цикл виробництва з додатковими витратами на зупинку та запуск виробничого процесу, *збільшуються витрати на зберігання* внаслідок непохитного росту продукції в місцях збереження і відсутності можливості корегування виробництва потребами торгівлі. В протилежному випадку, тобто при випуску продукції *партіями*, різко *знижується платня за зберігання*, а також зменшується ризик застарілості та втрати конкурентоспроможності продукції, але *вартість безпосередньо виробництва зростає*. Комплексне дослідження проблеми управління запасами виробило стратегію випуску продукції дискретно, тобто *партіями*, знаходження *оптимального розміру* яких і є метою розв'язання задачі управління запасами. Необхідно визначити такі оптимальні розміри випуску запасів продукції, щоб загальні витрати на весь цикл виробництва, зберігання, включаючи можливі штрафи за нестачу, були мінімальними. В загальному випадку такі задачі зводяться до задач нелінійного програмування з різноманітними методами розв'язання [5].

Математична модель загальної задачі нелінійного програмування формується через необхідність знайдення оптимального значення вектора параметрів управління (компонент плану) $X = (x_1, x_2, \dots, x_n)$, який максимізує (мінімізує) цільову функцію

$$z = f(X) \rightarrow \max(\min) \quad (2.4)$$

і задовольняє m обмеження, кожне з яких може мати один зі знаків $\leq, =, \geq$, тобто

$$\varphi_i(X) \{ \leq, =, \geq \} b_i, \quad i = 1 \div m. \quad (2.5)$$

Припускається, що вигляд всіх функцій відомий і хоча б одна з них є нелінійною. Зазвичай до деяких параметрів управління висувається вимога невід'ємності.

На відміну від лінійних задач, у нелінійних – цільова функція може досягати свого екстремального значення не тільки в її кутових точках, інакше, вершинах (якщо такі точки є) області обмежень або області припустимих розв'язків, але й на межах області та всередині неї. Крім того, вона може мати декілька локальних екстремумів. Цим пояснюється відсутність загального методу розв'язання задач нелінійного програмування (ЗНП), подібного симплекс-методу для ЗЛП. Разом з тим для досить поширеного класу ЗНП, а саме *задач опуклого програмування*, існують достатньо добре розроблені методи розв'язання. Основу математичної моделі опуклого програмування складають *опуклі (увігнуті) функції*, визначені на *опуклих множинах* у постановці (2.4)-(2.5).

Опуклою називається множина у векторному просторі, яка разом з будь-якими двома точками містить всі точки відрізка, який їх з'єднує. Аналітично ця властивість опуклої множини описується формулою для радіус-вектора \vec{r} поточної точки відрізка через радіус-вектори кінцевих точок відрізка

$$\vec{r} = \lambda \vec{r}_1 + (1 - \lambda) \vec{r}_2, \quad (2.6)$$

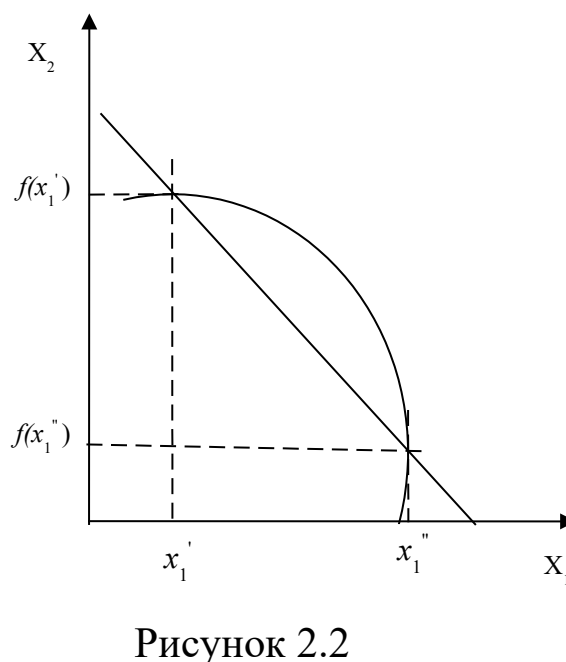
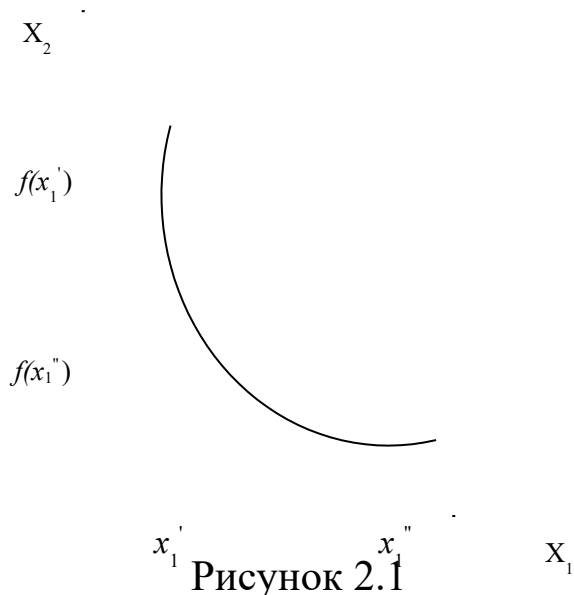
де коефіцієнт $0 \leq \lambda \leq 1$.

Функція $f(x)$, що визначена на опуклій множині, називається опуклою (увігнутою), якщо для будь-яких точок x_1 та x_2 з цієї множини і будь-якого $0 \leq \lambda \leq 1$ правильною є нерівність

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \quad (2.7)$$

$$(f(\lambda x_1 + (1 - \lambda)x_2) \geq \lambda f(x_1) + (1 - \lambda)f(x_2)). \quad (2.7')$$

Порівнюючи формули (2.7) ((2.7')) з формулою (2.6), можна уявити їх геометричне тлумачення. Нерівність (2.7) ((2.7')) виражає властивість будь-якої ординати опуклої (увігнутої) функції на відрізку $[x_1, x_2]$ набувати меншого (більшого) значення, ніж ордината хорди, що стягує дугу кривої функції $f(x)$ (рисунки 2.1, 2.2). Слід відмітити, що в математичному програмуванні вживають терміни „опуклої” („увігнутої”) функції в протилежному до прийнятого в математичному аналізі розумінні.



Властивості опуклих (увігнутих) функцій:

- якщо функції $f_i(x)$ ($i=1 \div k$) є опуклими (увігнутими) на опуклій множині, то опуклою (увігнутою) на цій множині буде лінійна комбінація цих функцій;

- якщо $\varphi(x)$ – опукла (увігнута) функція при всіх значеннях $x \geq 0$, то опуклою буде і множина розв'язків системи $\varphi(x) \leq b$, $x \geq 0$ ($\varphi(x) \geq b$, $x \geq 0$);

- опукла (увігнута) функція $f(x)$, що визначена на опуклій множині, досягає свого глобального мінімуму (максимуму) в кожній точці x , в якій градієнт функції обертається в нуль;

- локальний мінімум (максимум) опуклої (увігнутої) функції на опуклій множині збігається з її глобальним мінімумом (максимумом) на цій множині.

Серед можливих методів розв'язання задачі опуклого програмування для двовимірних задач виділяють графічний метод, який, як і для задач лінійного програмування, передбачає геометричний спосіб знаходження оптимального розв'язання. В опуклій області, що є геометричним образом системи обмежень задачі (2.5), знаходиться точка, через яку проходить геометричний образ цільової функції (2.4) – лінія рівня поверхні – і координати якої відповідають максимальному (мінімальному) значенню цільової функції. Як відомо, рівняння сім'ї ліній рівня задається рівністю

$$f(x) = C, \quad (2.8)$$

де C – довільне стале значення.

Приклад 2.1 Знайти екстремальні значення функції $z = x_1^2 + x_2^2$ в області невід'ємних розв'язків системи нерівностей, тобто математична модель задачі має такий вигляд:

$$z = x_1^2 + x_2^2 \rightarrow \max(\min);$$

$$\begin{cases} (x_1 - 5)^2 + (x_2 - 3)^2 \leq 36; \\ x_1 + x_2 \geq 8; \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

1 Графічний образ системи нерівностей – область, що обмежена відрізком AB прямої $x_1 + x_2 = 8$, дугою $B\bar{D}$ кола $(x_1 - 5)^2 + (x_2 - 3)^2 = 36$ з центром у точці $O_1(5;3)$ радіуса 6 та відрізком осі Ox_1 - DA (рисунок 2.3).

2 Сім'я ліній рівня – сукупність концентричних кіл $x_1^2 + x_2^2 = C$ з центром O , що міститься в початку координат.

3 Визначаємо точку мінімуму функції як точку дотику кола лінії рівня з відрізком межі AB – т. M , координати якої знаходимо як точку перетину відрізка AB та перпендикулярного йому радіуса OM , рівняння якого формуємо як канонічне рівняння прямої через т. O у напрямі нормального до прямої AB вектора $\vec{n}_{AB} = (1;1)$:

$$\frac{x_1}{1} = \frac{x_2}{1}.$$

Розв'язуємо систему рівнянь

$$\begin{cases} x_1 + x_2 = 8; \\ x_1 - x_2 = 0, \end{cases}$$

отримуючи координати точки мінімуму цільової функції $M(4;4)$.
Отже,

$$z_{\min} = z(4;4) = 4^2 + 4^2 = 32.$$

Точку максимуму цільової функції фіксуємо як останню точку дотику кола із сім'ї ліній рівня з областю припустимих розв'язків – т. N , координати якої визначаємо як точку перетину кола з системи обмежень та прямої OO_1 , яка описується рівнянням прямої, що проходить через дві точки $O(0;0)$ та $O_1(5;3)$. Складаємо та розв'язуємо систему рівнянь для знаходження т. N

$$\begin{cases} (x_1 - 5)^2 + (x_2 - 3)^2 = 36 \\ \frac{x_1}{5} = \frac{x_2}{3} \end{cases}; \quad \begin{cases} (x_1 - 5)^2 + (x_2 - 3)^2 = 36 \\ x_2 = \frac{3}{5}x_1 \end{cases};$$

$$\begin{cases} (x_1 - 5)^2 + \left(\frac{3}{5}x_1 - 3\right)^2 = 36 \\ x_2 = \frac{3}{5}x_1 \end{cases}; \quad \begin{cases} x_1^2 - 10x_1 + 25 + \frac{9}{25}x_1^2 - \frac{18}{5}x_1 + 9 = 36 \\ x_2 = \frac{3}{5}x_1 \end{cases};$$

$$\begin{cases} 17x_1^2 - 170x_1 - 25 = 0 \\ x_2 = \frac{3}{5}x_1 \end{cases}.$$

Розв'язуючи квадратне рівняння, обираємо додатний корінь. Таким чином знаходимо координати точки $N(10,14;6,08)$, а також відповідне значення цільової функції задачі на максимум

$$z_{\max} = z(10,14;6,08) = 10,14^2 + 6,08^2 \approx 139,83.$$

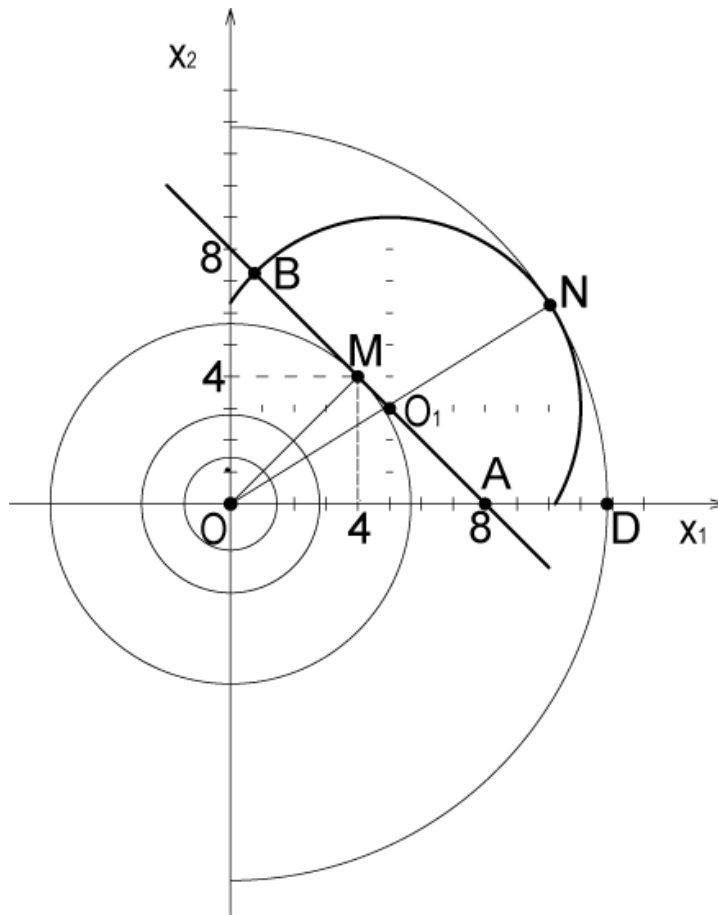


Рисунок 2.3

Серед задач опуклого програмування виділяють частковий випадок таких задач – задачі *квадратичного програмування*. В таких задачах цільова функція містить квадратичні доданки, а обмеження – лінійні. Приклади розв’язання таких задач графічним методом наведені в методичних вказівках [3, 4].

Розвинення логіки розв’язання задач опуклого програмування *аналітичними методами* буде усього, починаючи з простішого випадку постановки задачі на умовний екстремум. Якщо серед обмежень (2.5) немає нерівностей, умов невід’ємності та дискретності змінних, $m < n$, а функції $f(x)$ і $\varphi_i(x)$ неперервні та мають частинні похідні хоча б другого порядку, то задача приводиться до вигляду

$$z = f(x_1, x_2, \dots, x_n) \rightarrow \max(\min); \quad (2.9)$$

$$\varphi_i(x_1, x_2, \dots, x_n) = b_i \quad (i = 1 \div m). \quad (2.10)$$

Такі нелінійні задачі називають *класичними задачами оптимізації* і для їх розв’язання використовують *метод*

множників Лагранжа. Головною ідеєю цього методу є перехід до задачі на безумовний екстремум відносно *функції Лагранжа*, яка включає в себе безпосередньо цільову функцію (2.9) та систему обмежень (які ще називають *рівняннями зв'язку*) з так званими *ваговими коефіцієнтами* або множниками Лагранжа λ_i ($i = 1 \div m$). Отже, функція Лагранжа є функцією двох типів змінних: компонент вектора параметрів управління (компонент плану) задачі $X = (x_1, x_2, \dots, x_n)$ та компонент вектора множників Лагранжа $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ - і має вигляд

$$L(X, \Lambda) = f(X) + \sum_{i=1}^m \lambda_i (b_i - \varphi_i(X)). \quad (2.11)$$

Далі, як для задачі на безумовний екстремум, знаходять стаціонарні точки функції Лагранжа з системи рівнянь, які відбивають необхідні умови існування екстремуму

$$\begin{cases} \frac{\partial L(X, \Lambda)}{\partial x_j} = 0 \\ \frac{\partial L(X, \Lambda)}{\partial \lambda_i} = 0 \end{cases} \Rightarrow \begin{cases} \frac{\partial f(X)}{\partial x_j} - \sum_{i=1}^m \lambda_i \frac{\partial \varphi_i(X)}{\partial x_j} = 0; (j = 1 \div n) \\ b_i - \varphi_i(X) = 0; (i = 1 \div m) \end{cases}. \quad (2.12)$$

Наступним кроком із стаціонарних точок без значень λ_i ($i = 1 \div m$) обирають такі, в яких функція $f(X)$ має умовні екстремуми з урахуванням обмежень (2.10). Цей відбір здійснюють, наприклад, за допомогою достатніх умов існування екстремуму, але при розв'язанні конкретних задач дослідження спрощується за рахунок використання штучних прийомів, що прийняті за методику обчислень.

У методичних вказівках [5] описані методи розв'язання задач управління запасами в рамках постановки класичної задачі оптимізації з використанням спеціальних прийомів, що придатні саме для таких задач.

Метод множників Лагранжа узагальнюється на випадок, коли в математичній моделі (2.4)-(2.5) змінні невід'ємні та деякі обмеження задані у формі нерівностей. Суть методу полягає в тому, що для математичної моделі

$$z = f(X) \rightarrow \max, \quad (2.13)$$

$$\varphi_i(X) \geq 0; i = 1 \div m, \quad (2.14)$$

$$x_j \geq 0; j = 1 \div n \quad (2.15)$$

створюється функція Лагранжа

$$L(X, \Lambda) = f(X) + \sum_{i=1}^m \lambda_i \varphi_i(X), \quad (2.16)$$

необхідні умови існування екстремуму для якої мають вигляд умов Куна-Таккера. Вони використовуються для знаходження так званої *сідлової точки* функції Лагранжа (X^*, Λ^*) , яка визначається як така, що задовольняє ланцюжок нерівностей

$$L(X, \Lambda^*) \leq L(X^*, \Lambda^*) \leq L(X^*, \Lambda). \quad (2.17)$$

Ця формула виражає основну властивість сідлової точки бути одночасно точкою мінімуму функції відносно вектора однієї змінної та точкою максимуму відносно вектора іншої.

За *теоремою Куна-Таккера* доводиться, що при виконанні деяких умов вектор $X^* \geq 0$ є оптимальним розв'язком задачі (2.13)-(2.15) тоді і тільки тоді, коли для нього можна вказати значення вектора $\Lambda^* \geq 0$ таке, що у сукупності ці вектори створюють сідлову точку (X^*, Λ^*) функції Лагранжа (2.17). Практично теорема Куна-Таккера зводиться до умов *Куна-Таккера* для знаходження сідлової точки

$$\left(\frac{\partial L}{\partial x_j} \right)^* \leq 0; \quad x_j^* \left(\frac{\partial L}{\partial x_j} \right)^* = 0; \quad x_j^* \geq 0, \quad j = 1 \div n; \quad (2.18)$$

$$\left(\frac{\partial L}{\partial \lambda_i} \right)^* \geq 0; \quad \lambda_i^* \left(\frac{\partial L}{\partial \lambda_i} \right)^* = 0; \quad \lambda_i^* \geq 0, \quad i = 1 \div m.$$

Приклади, які поєднують використання графічного методу розв'язання задачі квадратичного програмування та умов Куна-Таккера для знаходження сідлової точки цієї задачі, детально розглянуті в методичних вказівках [3, 4].

Градiєнтні методи можна застосовувати для розв'язання будь-якої задачі нелінійного програмування, але вони приводять лише до локального екстремуму. І знов особливе місце серед ЗНП займають задачі опуклого програмування, для яких кожний локальний екстремум є одночасно і глобальним.

Градiєнтні методи використовують властивість *вектора-градiєнта (антиградiєнта)* функції, координати якого в даній точці визначаються частинними похідними функції в цій точці, визначати *напрямок найшвидшого зростання (спадання) цієї функції*, тобто для функції $z = f(X) = f(x_1, x_2, \dots, x_n)$

$$\text{grad} f(X_0) = \nabla f(X_0) = \left(\left. \frac{\partial f(X)}{\partial x_1} \right|_{m.X_0}; \left. \frac{\partial f(X)}{\partial x_2} \right|_{m.X_0}; \dots; \left. \frac{\partial f(X)}{\partial x_n} \right|_{m.X_0} \right). \quad (2.19)$$

Для опуклої функції необхідною та достатньою умовою оптимальності точки X^* є рівність нулю градiєнта функції в цій точці.

Алгоритм пошуку екстремуму функції $f(X)$ організований так. У припустимій області розв'язків обирається довільна точка X_0 та за допомогою градiєнта (антиградiєнта) визначається напрям, у якому $f(X)$ зростає (спадає) з найбільшою швидкістю. Зробивши певний невеликий крок у знайденому напрямі, переходимо до наступної точки X_1 . Процес повторюється доти, поки не відбудеться збігання до точки екстремуму X^* , тобто для послідовності точок X_0, X_1, X_2, \dots необхідне виконання умов $f(X_0) < f(X_1) < f(X_2) < \dots$ в задачі на максимум ($f(X_0) > f(X_1) > f(X_2) > \dots$ - в задачі на мінімум).

Величина кроку, наприклад, з точки X_{i-1} до точки X_i в напрямі градiєнта $\nabla f(X_{i-1})$ (антиградiєнта $-\nabla f(X_{i-1})$) визначається величиною параметра λ у рівнянні прямої

$$X_i = X_{i-1} + \nabla f(X_{i-1})\lambda \quad (X_i = X_{i-1} - \nabla f(X_{i-1})\lambda), \quad (2.20)$$

що проходить через точку X_i паралельно градiєнту (антиградiєнту). Значення λ обирають або з конкретних умов задачі, або з таких міркувань: переміщення уздовж прямої (2.20) до нової точки супроводжується зміною функції $f(X)$ на величину $\Delta f(X)$

$$\Delta f(X) = f(X_i) - f(X_{i-1}) = f(X_{i-1} + \nabla f(X_{i-1})\lambda) - f(X_{i-1}), \quad (2.21)$$

яка залежить від обраного значення λ . Можна знайти оптимальне значення λ^* , яке надає приросту функції $\Delta f(X)$ найбільшої величини. Це значення можна знайти з необхідних умов існування екстремуму

$$\begin{aligned} \frac{d\Delta f}{d\lambda} = 0 &\Rightarrow \frac{d(f(X_i) - f(X_{i-1}))}{d\lambda} = 0 \Rightarrow \frac{d(f(X_{i-1} + \nabla f(X_{i-1})\lambda) - f(X_{i-1}))}{d\lambda} = 0 \\ &\Rightarrow \nabla f(X_i) \cdot \nabla f(X_{i-1}) = 0. \end{aligned} \quad (2.22)$$

Приклад 2.2 Максимізувати функцію

$$f(X) = 2x_1 - 0,2x_1^2 + 2,4x_2 - 0,2x_2^2 \rightarrow \max$$

при наявності обмежень

$$\begin{cases} x_1 + 2x_2 \geq 10; \\ x_1^2 + x_2^2 \leq 100; \\ x_1 \geq 0; \\ x_2 \leq 8. \end{cases}$$

Обираємо у припустимій області розв'язків довільну точку, наприклад, $X_0 = (6,4)$, яка дійсно належить області обмежень, що можна перевірити простою підстановкою:

$$\begin{aligned} 6 + 2 \cdot 4 = 14 > 10 &\text{ — правильно;} \\ 6^2 + 4^2 = 36 + 16 = 52 < 100 &\text{ — правильно;} \\ 6 > 0 &\text{ — правильно;} \\ 4 < 8 &\text{ — правильно.} \end{aligned}$$

Знаходимо

$$f(X_0) = 2 \cdot 6 - 0,2 \cdot 6^2 + 2,4 \cdot 4 - 0,2 \cdot 4^2 = 12 - 7,2 + 9,6 - 3,2 = 11,2.$$

Обчислюємо координати градієнта

$$\nabla f(X) = \left(\frac{\partial f}{\partial x_1}; \frac{\partial f}{\partial x_2} \right) = (2 - 0,4x_1; 2,4 - 0,4x_2),$$

і тоді в точці X_0 $\nabla f(X_0) = (2 - 0,4 \cdot 6; 2,4 - 0,4 \cdot 4) = (-0,4; 0,8) \neq 0$, тобто точка X_0 не є точкою екстремуму.

Знайдемо рівняння прямої, уздовж якої перемістимося до нової точки X_1 у напрямі градієнта, за формулами (2.20) для кожної координати точки X_1 , а саме

$$X_1 = (6 - 0,4\lambda; 4 + 0,8\lambda).$$

Для визначення координат точки X_1 скористуємося умовами (2.22) для знаходження значення λ . Для цього сформуємо вираз для $\nabla f(X_1)$:

$$\nabla f(X_1) = (2 - 0,4 \cdot (6 - 0,4\lambda); 2,4 - 0,4 \cdot (4 + 0,8\lambda)) = (-0,4 + 0,16\lambda; 0,8 - 0,32\lambda).$$

І тоді, записуючи необхідні умови існування екстремуму (2.22), отримаємо рівняння відносно параметра λ :

$$(-0,4; 0,8) \cdot (-0,4 + 0,16\lambda; 0,8 - 0,32\lambda) = 0;$$

$$0,16 - 0,064\lambda + 0,64 - 0,256\lambda = 0;$$

$$0,8 - 0,32\lambda = 0;$$

$$\lambda = \frac{0,8}{0,32} = 2,5.$$

Тобто координати точки X_1 такі: $(X_1) = (6 - 0,4 \cdot 2,5; 4 + 0,8 \cdot 2,5) = (5; 6)$, причому вона належить області обмежень (в чому легко переконатися перевіркою правильності нерівностей у системі обмежень). Перевіряємо значення градієнта функції в цій точці

$$\nabla f(X_1) = (2 - 0,4 \cdot 5; 2,4 - 0,4 \cdot 6) = (0; 0).$$

Це означає, що більш ніякими переміщеннями не можна збільшити функцію $f(X)$, і тому X_1 – точка максимуму X^* . Отже,

$$X^* = (5;6), \quad f_{\max}(X^*) = 2 \cdot 5 - 0,2 \cdot 5^2 + 2,4 \cdot 6 - 0,2 \cdot 6^2 = 10 - 5 + 14,4 - 7,2 = 12,2.$$

Розв'язок задачі проілюстровано на рисунку 2.4.

Треба відмітити, що у розглянутому прикладі точний розв'язок отриманий за кінцеву кількість кроків (що зокрема відбулося за рахунок вдалого обрання початкової точки), але, як правило, градієнтні методи дають точний розв'язок за нескінченну кількість кроків.

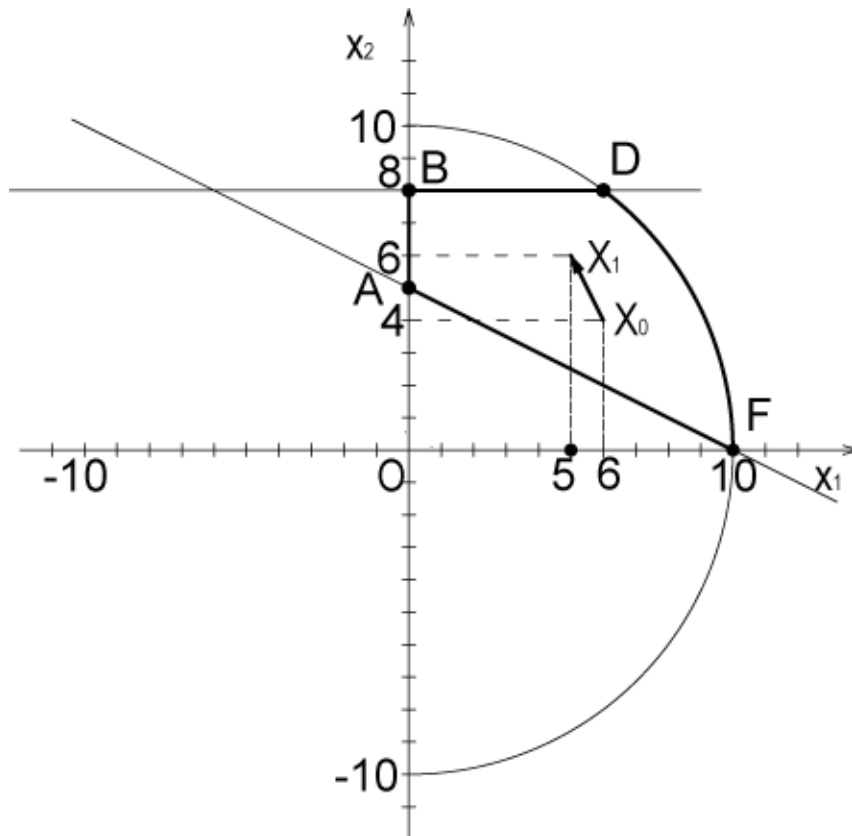


Рисунок 2.4

3 СІТЬОВІ ЗАДАЧІ УПРАВЛІННЯ ТРАНСПОРТНИМИ СИСТЕМАМИ

Існує цілий клас задач, які вимагають розв'язання як важливі задачі на залізничному транспорті, математичні моделі яких зручно задавати у вигляді *графів* або при їх розв'язанні використовувати так звані *сітьові способи*, які побудовані на *теорії графів*.

3.1 ЕЛЕМЕНТИ ТЕОРІЇ ГРАФІВ

Для завдання *графів* використовується два типи множин: множина *вершин графа*

$$v = (v_1, v_2, \dots, v_n)$$

та множина *ребер або дуг графа*

$$u = (u_1, u_2, \dots, u_m).$$

Кожний елемент u_k ($k = 1 \div m$) визначається певною парою вершин, які він поєднує, тобто

$$u_k = (v_i, v_j).$$

Отже, граф – це і є сукупність пар елементів (u, v) . Первинним, а тому невизначеним, є поняття вершини, яке можна розуміти як деякий „вузол”, який позначається крапкою, кільцем, квадратом або трикутником. В залежності від типу графа: графа *станів* або *географічного* – вузол може означати стан, умову або пункт призначення. В першому випадку граф являє собою зображення процесу зміни станів об’єкта, у другому – граф будується подібним до „географії” об’єкта, що моделюється. *Редра або дуги* – це відрізки прямих або кривих, які з’єднують вершини. Послідовність ребер утворює *ланцюг*. Ребро u_j називається *інцидентним* вершині v_i , якщо воно входить або виходить з цієї вершини. *Степінь (валентність)* вершини – кількість інцидентних до неї ребер. Граф називається *орієнтовним* або *орграфом*, якщо вказано, яка вершина є початковою, тобто показаний напрям дуг. Якщо з кожним ребром графа пов’язані якісь числові характеристики, або інакше – ребрам графа приписана певна вага, то граф називається *зваженим*. Кожному графу можна поставити у відповідність квадратну *матрицю суміжності*, розмір якої визначається кількістю вершин і яка складається з елементів s_{ij} , що дорівнюють сумі чисел

орієнтованих або неорієнтованих ребер, які поєднують вершини v_i та v_j . Важливим поняттям в теорії графів є поняття зв'язності графа, яке пов'язане з наявністю принаймні одного шляху для будь-яких двох вершин графа. Граф називають *петлею*, якщо його початок і кінець збігаються. *Циклічним* називають граф, який являє собою кінцевий ланцюг, у якого збігаються початкова і кінцева вершини. Ще деякі відомості про графи на початковому рівні можна брати з методичних вказівок [3, 4].

Взагалі, графи, як інструмент моделювання, мають дуже широке використання саме у транспортних задачах, завдяки таким перевагам, як наочність, гнучкість апарату. Розглянемо спочатку деякі з найбільш традиційних сітьових задач.

3.2 СІТЬОВІ МОДЕЛІ ТРАНСПОРТНИХ ЗАДАЧ

Як відомо з курсу дослідження операцій, транспортна задача – це частковий випадок загальної задачі лінійного програмування, для якої були введені специфічні методи розв'язання, зокрема метод потенціалів. Цей метод є значно простішим, ніж симплекс-метод, і основа його ґрунтується на перетворенні так званої транспортної таблиці, або матриці. *Класична транспортна задача* формулювалася як задача про найбільш економний спосіб перевезень саме на транспорті, але ця назва досить умовна, оскільки за її допомогою можна розв'язувати різноманітні задачі з інших галузей.

Сітьові методи не вимагають створення матриці перевезень, а дозволяють робити обчислення безпосередньо на *схемі шляхів сполучення* – на графі. Вершини графа відповідають пунктам перевезень (зокрема, станціям), а ребра – шляхам між пунктами. Граф є зваженим з вагою ребра – вартістю перевезення c_{ij} . Обсяги вантажу від постачальників a_i та для споживачів b_j показані в дужках біля вершин графа відповідно зі знаком „плюс” та „мінус”. Ребро вважається навантаженим, якщо вздовж нього проходить будь-який потік вантажу $x_{ij} > 0$. Основна ознака методу потенціалів – саме *система потенціалів* – вводиться і в сітьовий алгоритм цього методу. Це система умовних чисел, які певним чином приписуються кожній вершині.

Суть методу полягає у знаходженні системи потенціалів, які задовольняють умови оптимальності

$$|v_i - v_j| \leq c_{ij}, \quad x_{ij} = 0 \quad \text{— для ненавантажених ребер;}$$

$$|v_i - v_j| = c_{ij}, \quad x_{ij} > 0 \quad \text{— для навантажених ребер.} \quad (3.1)$$

Отже, алгоритм сітьового методу полягає в такій послідовності дій:

1 Побудування початкового плану. Принцип створення транспортної таблиці переноситься на граф так. Потоки вантажів зображують стрілками, які відображають напрям, і цифрами - їх величини біля відповідних ребер. Загальна величина потоку вантажу, що виходить з вершини, дорівнює запасу, який завантажили на цій станції, плюс транзитні потоки, а величина потоку, що входить у вершину, дорівнює відповідно вказаній потребі у вантажі плюс транзитні потоки.

2 Обчислення потенціалів. Якійсь вершині надають довільний потенціал та визначають потенціали всіх інших вершин з умов оптимальності (3.1) для навантажених ребер. Для цього, рухаючись по навантажених ребрах від вершин з відомими потенціалами до вершин з невідомими потенціалами, визначаючи останні, додаючи вартість перевезень на ребрі до попереднього потенціалу, якщо рух збігається з напрямом потоку вантажу, та віднімаючи з попереднього потенціалу, якщо рух зворотний. Можна зустрітися з випадком, коли сітка по потоках вантажу поділяється на декілька не пов'язаних між собою систем (аналогічно випадку виродженості плану транспортної задачі). Тоді вздовж деяких ділянок пропускають умовний нульовий потік.

3 Перевірка ненавантажених ребер за умовами оптимальності (3.1). Порушення цієї умови, тобто перевищення абсолютною величиною різниці потенціалів вартості перевезень, записують біля відповідних ребер зі знаком "плюс".

4 Цикл перерахування. Ребро з найбільшою величиною „порушення” оптимальності навантажують, обираючи напрям потоку від меншого потенціалу до більшого. Для визначення величини корегувального потоку будується замкнений контур по

навантажених ребрах, і, рухаючись вдовж контуру, визначаємо мінімальний зустрічний призначеному напрямку потік. Присвоюємо „гіршому” потоку на ненавантаженому ребрі значення корегувального потоку, а потім додаємо це значення до всіх попутних і віднімаємо від усіх зустрічних потоків. Економія від поліпшення плану відбивається добутком величини „порушення” на потенціальному ребрі, що навантажувється, на потужність призначеного корегувального потоку.

Як бачимо, наведений алгоритм дуже наочно корелює з відомим алгоритмом методу потенціалів як методу перетворення транспортної таблиці.

До транспортних моделей можна віднести так звані *задачі про призначення*, які потребують знаходження розподілу n працівників за n типами робіт, якщо робітникам притаманний різний ступінь кваліфікації (відома продуктивність праці кожного) з метою досягнення максимального виробітку. Сітьова модель транспортної задачі також влучно використовується у випадку, коли між початковими та кінцевими пунктами перевезень є проміжні пункти для тимчасового зберігання вантажу (транзитні пункти). Це так звана *транспортна модель з проміжними пунктами*, яка за змістом перетинається із задачами керування запасами. Приклади розв’язання цих задач надані у [6, 7].

3.3 ЗАДАЧА ПРО МАКСИМАЛЬНИЙ ПОТІК НА СІТКАХ

Ці задачі відносяться до дуже розповсюджених задач, в яких відбиваються транспортні моделі і використовується теорія графів.

По сітці магістралей (транспортних, трубопроводів), тобто по ребрах (i, j) , де i, j – номери вершин, які поєднуються відповідними ребрами ($i < j$), направляєється ресурс (речовина, вантаж, інформація) від *входу (витоку)* I графа до його *виходу (стоку)* S . Максимальна кількість речовини r_{ij} , яку може пропустити за одиницю часу ребро (i, j) , називається його *пропускною спроможністю*. В загальному випадку $r_{ij} \neq r_{ji}$. Якщо вершини i та j на сітці не з’єднані, то $r_{ij} = r_{ji} = 0$. Пропускні

спроможності ребер вздовж прямого та зворотного напрямку зазвичай показують у дужках на графі, а також їх можна задати квадратною матрицею, розмір якої визначається кількістю вершин на сітці.

Кількість x_{ij} речовини, що проходить через ребро (i, j) за одиницю часу, називається *поток* по ребру. В теорії потоків припускається, що, якщо від вершини i до вершини j направляється потік величиною x_{ij} , величина потоку в зворотному напрямі дорівнює $-x_{ij}$, тобто

$$x_{ji} = -x_{ij} . \quad (3.2)$$

Приймається також, $x_{ii} = 0$.

Сукупність $X = \{x_{ij}\}$ потоків вздовж усіх ребер (i, j) сітки називають *поток* на сітці, або просто *поток*.

Потік вздовж кожного ребра не перебільшує його пропускну спроможність

$$x_{ij} \leq r_{ij} \quad (i, j = 1 \div n), \quad (3.3)$$

де n – кількість вершин сітки.

Кількість речовини, що втікає у вершину, дорівнює кількості речовини, що витікає з неї,

$$\sum_{j=1}^n x_{ij} = 0 \quad (i \neq I, S). \quad (3.4)$$

Властивості (3.2) ÷ (3.4), які вважаються *основними обмеженнями* потоків по ребрах, необхідно враховувати при формуванні потоку на сітці.

Алгоритм знаходження максимального потоку вимагає введення поняття *розрізу* сітки, що визначає множину ребер, при вилученні яких повністю припиняється потік від витіку до стоку, тобто фактично розріз складається з двох підмножин ребер, які не перетинаються: в одній підмножині A міститься витік I , у другій B – стік S . *Пропускна спроможність розрізу* дорівнює сумі пропускну спроможностей „розрізаних” ребер і позначається $R(A/B)$, тобто

$$R(A/B) = \sum_{i \in A} \sum_{j \in B} r_{ij}. \quad (3.5)$$

Якщо на сітці заданий деякий потік $X = \{x_{ij}\}$, то ребро (i, j) називають *насиченим*, якщо величина потоку по цьому ребру збігається з його пропускною спроможністю, і *ненасиченим*, якщо потік менше його пропускної спроможності.

Величина

$$X(A/B) = \sum_{i \in A} \sum_{j \in B} x_{ij},$$

яка складає суму потоків x_{ij} по всіх ребрах розрізу, називається *потокотом через розріз*.

У теорії задач на максимальний потік на сітці важливе значення має *теорема Форда-Фалкерсона*: на будь-якій сітці максимальна величина потоку від витoku I до стоку S дорівнює мінімальній пропускній спроможності розрізу, що відділяє I від S .

Наведемо *алгоритм побудування максимального потоку*, для розуміння теоретичної обґрунтованості якого варто додатково ознайомитись з описанням більш детального алгоритму, пов'язаного зі складанням матриць пропускних спроможностей R , потоків X та матриці їх різниць $R - X$, наприклад, в роботі [1]. Отже,

1 Будуємо деякий початковий потік $X^0 = \{x_{ij}^0\}$. Як рекомендацію можна взяти до уваги, що цей потік доцільно починати від витoku I по ребру з більшою пропускною спроможністю. Потік ілюструється ланцюгом відповідних вершин, що поєднуються відрізками прямих, а саме ребер, над якими показують перенесені з сітки пропускні спроможності (r_{ij}, r_{ji}) .

2 Фіксуємо мінімальну пропускну спроможність ланцюга першому значенню потоку x_0 , який стане першим доданком сумарного потоку і одночасно значенням Δ , на яке корегуються пропускні спроможності ланцюга, тобто $\Delta = x_0 = \min r_{ij}$.

3 Створюємо скорегований ланцюг пропускних спроможностей у виписаному ланцюзі під відрізками, що позначають ребра, віднімаючи Δ від пропускної спроможності у

напрямі руху потоку і додаючи значення Δ до пропускної спроможності у протилежному русі: $(r_{ij} - \Delta, r_{ji} + \Delta)$.

4 Якщо хоча б одне зі значень $r_{ij} - \Delta = 0$, то ребро стає насиченим, помічаємо його хрестиком і обираємо наступний потік в обхід насиченого ребра. Якщо жодна різниця не дорівнює нулю, наступний потік може завершити доведення одного з ребер початкового потоку до насиченого.

5 Процес створення потоків x_1, x_2, \dots, x_n в рамках наведених пунктів алгоритму продовжується до тих пір, доки на шляху до стоку S сформується хоча б один розріз, який проходить по насичених ребрах, тобто використані всі можливі шляхи формування нових потоків.

6 Знаходиться отриманий сумарний потік $X = \sum_{k=0}^n x_k$, який і повинен бути максимальним, що треба підтвердити перевіркою теореми Форда-Фалкерсона.

3.4 СІТЬОВЕ ПЛАНУВАННЯ ТА УПРАВЛІННЯ НА ЗАЛІЗНИЧНОМУ ТРАНСПОРТІ

Важливим напрямком використання теорії графів є розповсюдження її на задачі планування та управління виробничими процесами, зокрема, на залізничному транспорті. Особливості *методів сітьового планування та управління* (СПУ) полягають у можливості в графічній формі відбити виробничий процес, показати послідовність та логічний взаємозв'язок окремих робіт, які створюють процес, виявити *критичні*, тобто ті, що визначають його хід, *роботи* та зосередити на них увагу.

СПУ ґрунтується на побудуванні *орієнтовного графа станів виробничого процесу*, в якому *вершини* – це *події*, які відбивають *умови* можливості початку або закінчення різноманітних робіт, а *ребра* – операції, які складають саме виробничий процес, тобто безпосередньо *роботи*.

Події нумерують послідовно. Кожну роботу позначають номерами подій, що її обмежують. Ребра графа є *зваженими тривалістю роботи* у певному масштабі часу. Будь-яка послідовність робіт визначає *шлях*. СПУ має на меті такі задачі:

1 *Науковий аналіз і розробка технологічних процесів.* Це означає *оптимізацію сітьового графіка*, тобто знаходження такого порядку всіх робіт та розподілу між ними ресурсів таким чином, щоб забезпечити виконання всього процесу за мінімальний термін (оптимізація за часом) та з мінімальними витратами (оптимізація за вартістю). Це досягається порівнянням варіантів сітьового графіка та його послідовного поліпшення.

2 *Оперативний контроль за ходом процесу.* Методи СПУ дозволяють на кожному етапі виділяти головні (критичні) роботи, на які треба звернути особливу увагу. До розряду критичних на кожному етапі можуть потрапляти різні роботи в залежності від обстановки та фактичного ходу процесу.

Основу сітьового графіка складають три елементи: *робота, подія, шлях.*

Робота – це самостійна трудова операція. Серед робіт розрізняють такі типи:

- *дійсна* – трудовий процес, який вимагає часу і ресурсів (наприклад, технічний огляд составу, подача вагонів під навантаження ін.);

- *очікування* – технологічна операція, яка потребує часу без витрат ресурсів (наприклад, технологічна перерва у маневровій роботі через пропускання пасажирського потяга, операція зупинки якогось приладу при підготовці його до ремонту та ін.);

- *фіктивна* – указує тільки на логічний зв'язок між роботами і не потребує ані часу, ані ресурсів.

Дійсну роботу та очікування відмічають на графіку суцільними стрілками, а фіктивну – штриховими.

Подія – фіксований факт, який визначає закінчення або можливість початку декількох робіт, момент, який фіксує певний стан виробничого процесу (не має тривалості). Її треба сформулювати в категоричній формі. Наприклад: «Технічний огляд потяга завершений». Це означає готовність до виконання наступної роботи. У кожній роботі є початкова та кінцева подія. Роботу не можна почати, доки не здійсниться її початкова подія. Подія же не може статися, доки не будуть виконані всі роботи, для яких вона вважається кінцевою. Розрізняють *проміжні, первинні (початкові) та завершальні* події.

Шлях – будь-яка послідовність робіт, яка поєднує деякі дві події. Тривалість його дорівнює сумарній тривалості робіт, які складають його. Розрізняють такі шляхи:

- *повний* – послідовність робіт, які поєднують первинну та завершальну події;
- *передуючий події*;
- *наступний за подією*;
- *проміжний між подіями*;
- *критичний* – найбільший повний шлях.

На сітьовому графіку завжди декілька повних шляхів. Максимальний з них є критичним. Його тривалість визначає *загальний термін виконання процесу*. Формулюється задача *оптимізації сітьового графіка за критерієм мінімізації цього терміну*. Для цього необхідно вишукати способи скорочення критичного шляху, контролювати виконання робіт, які утворюють критичний шлях, та вживати оперативних заходів для попередження їх зриву. Саме *наочність зображення послідовності робіт, якою визначається загальний термін виконання процесу*, і є важливою перевагою сітьового графіка перед іншими методами планування і контролю.

Існують певні правила побудови сітьових графіків.

1 У кожній роботі номер початкової події повинен бути меншим за номер кінцевої. Послідовність подій встановлюють за *рангом*, тобто після викреслювання робіт, які виходять з попередньо пронумерованих подій (починаючи з початкової, до якої не входить жодної роботи), до одного рангу відносять ті, до яких не входить жодної роботи, і нумерують їх у довільному порядку і так далі.

2 Кожна проміжна подія повинна мати хоча б одну передуючу та наступну роботу (за виключенням постачання матеріалів, обладнання тощо, які зображують як роботи, що виходять з події, до якої не входить жодна робота).

3 Жоден шлях не повинен проходити двічі крізь одну і ту ж подію, тобто на сітці не повинен утворюватись замкнений цикл.

4 Між подіями може знаходитись тільки одна робота. Для зображення паралельних робіт до сітки вводять фіктивні роботи та додаткові події (рисунок 3.1).

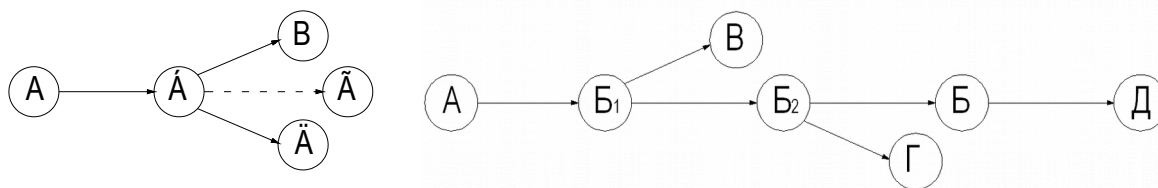


неправильно

правильно

Рисунок 3.1

5 Правило зображення *складених робіт*, тобто таких, які можуть бути розпочаті тільки після виконання деякого комплексу робіт, полягає у тому, що ці роботи зображують як сукупність послідовних робіт, результати яких є необхідними і достатніми для початку таких робіт: якщо роботу *БД* можна розгорнути тільки після виконання роботи *АБ*, а *БВ* і *БГ* – до її закінчення, то роботу *АБ* треба розбити додатковими подіями на такі частки, виконання яких забезпечить можливість початку робіт *БВ* і *БГ* (рисунок 3.2).

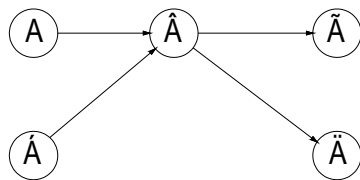


неправильно

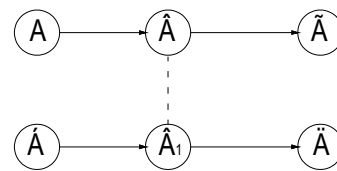
правильно

Рисунок 3.2

6 Правило зображення *диференціально-залежних робіт*, тобто таких, що для виконання однієї або декількох з групи необхідно отримати результати всіх робіт, які входять до її початкової події, а для іншої (або інших) – тільки декількох з них, тоді в сітку вводять додаткові події та фіктивні роботи: якщо для виконання роботи *ВГ* необхідні результати робіт *АВ* і *БВ*, а для *БД* – тільки результати *БВ*, то в сітку вводять додаткову подію *В₁* та фіктивну роботу *В₁В* (рисунок 3.3).



неправильно



правильно

Рисунок 3.3

Серед параметрів, якими характеризується сітьовий графік, можна відзначити такі:

- *тривалість критичного шляху* $T_{кр.}$;
- *найбільш ранній термін здійснення події i* - мінімально необхідний час між настанням початкової події та події i , тобто максимальний шлях, який передуює події i : $T_{Ei} = \max T_{перед.i}$. Зазвичай для початкової події $T_{Ei} = 0$;

- *найбільш пізній припустимий термін здійснення події i* - максимально припустимий час між початковою подією та подією при незмінному критичному шляху: $T_{Li} = T_{кр} - \max T_{слід.i}$, де $\max T_{слід.i}$ - максимальний шлях, який проходить за подією. Якщо подія i розташована на критичному шляху, то $T_{Ei} = T_{Li}$. Для завершальної події $T_{Ei} = T_{Li} = T_{кр.}$;

- *резерв часу для шляху* - це різниця між тривалістю критичного шляху та тривалістю T шляхів, що розглядаються: $R_T = T_{кр.} - T$. Цей параметр символізує час, на який можна збільшити тривалість усіх робіт без зміни загального терміну реалізації процесу, тобто гранично припустиме збільшення тривалості шляху T , при подальшому збільшенні якої шлях стає критичним;

- *резерв часу для події i* - це показник часу, на який можна затримати здійснення події, не змінюючи загального терміну виконання процесу: $R_i = T_{Li} - T_{Ei}$. Для подій, що містяться на критичному шляху, $R_i = 0$, тому що $T_{Ei} = T_{Li}$. Аналіз резервів часу для виконання кожної роботи допомагає ефективно розподілити ресурси часу між роботами. Розрізняють такі види резервів часу для роботи ij (рисунок 3.4):

- *повний* - резерв часу максимального зі шляхів, що проходять через роботу, $R_{ij} = T_{кр} - \max T_{ij}$. Він відбиває припустиме збільшення тривалості роботи (або спізнювання її початку), при

якому довжина максимального зі шляхів, що проходять крізь неї, не перебільшить тривалості критичного шляху. При використанні цього резерву максимальний шлях через роботу ij стає критичним, і всі розташовані на ньому роботи втрачають резерви часу. Величину R_{ij} зручно визначати через ранній та пізній терміни здійснення подій: $R_{ij} = T_{Lj} - T_{Ei} - t_{ij}$;

- *вільний (окремий)* – максимальний час, на який можна збільшити тривалість роботи ij (або відкласти її початок), якщо її початкова та кінцева події настануть у свої ранні терміни: $R_{ij}^c = T_{Ej} - T_{Ei} - t_{ij}$. При використанні вільного резерву часу для однієї роботи не порушуються вільні резерви часу інших робіт, якщо всі події настають у свої ранні терміни;

- *незалежний* – утворюється лише у деяких роботах і відображає максимальний час, на який можна збільшити тривалість роботи ij (або відкласти її початок) незалежно від термінів настання її початкової та кінцевої подій: $R_{ij}^n = T_{Ej} - T_{Li} - t_{ij}$. Використання незалежного резерву в будь-якому випадку не порушує резервів часу інших робіт.

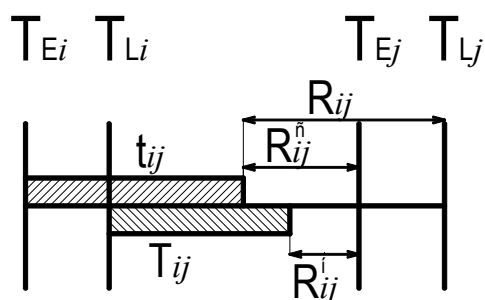


Рисунок 3.4

Роботи, що розташовані на критичному шляху, не мають жодних резервів часу. Таким чином, знаючи ранні та пізні терміни здійснення подій та тривалість робіт, можна визначити параметри сітьового графіка.

Для обчислення параметрів сітьового графіка застосовують *табличний, матричний та інші способи*. Для порівняно невеликих графіків, які використовують при експлуатаванні залізниць, зручно обчислювати параметри саме на графіках. Для цього кожен подію – кружечок – поділяють на чотири сектори (рисунок 3.5).

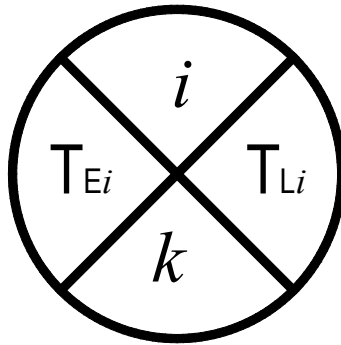


Рисунок 3.5

У верхньому проставляють номер події i , інші заповнюють протягом обчислення. У лівому секторі записують ранній термін здійснення події T_{Ei} , у правому – пізній T_{Lj} , номер попередньої події на максимальному передуючому шляху. Починають облік з початкової події, відмічаючи $T_{Ei}=0$, а потім обчислюють T_{Ej} , послідовно переходячи від події з меншим номером до наступної за формулою $T_{Ej} = \max(T_{Ei} + t_{ij})$, де i – номери всіх передуючих подій за кількістю вхідних до події j робіт. Пересуваючись до останньої події, заповнюють у неї і правий сектор, оскільки $T_{Ei} = T_{Li} = T_{кр}$. Цим визначена величина критичного шляху. Потім обчислюють T_{Li} , послідовно пересуваючись у протилежному напрямі, а саме від події з більшим номером до події з меншим номером, за формулою $T_{Li} = \min(T_{Lj} - t_{ij})$, де j – номери всіх наступних подій за кількістю вихідних з події i робіт. У початковій події при цьому повинні отримати $T_{Li} = 0$, тому що для неї $T_{Ei} = T_{Li} = 0$. Розташування критичного шляху графіка установлюється, починаючи з завершеної події, за номерами, що містяться у нижніх секторах.

Після розробки сітьового графіка для календарного планування термінів початку та закінчення робіт складають лінійну діаграму з демонстрацією кількості робіт, що виконуються в кожний момент часу. За допомогою діаграми можна рівномірно розподіляти ресурси часу. Лінійна діаграма являє собою графік, по горизонтальній осі якого відкладається час, по вертикальній – смугами з номерами початкової та кінцевої події – роботи (ті, що розташовані на критичному шляху, заштриховані). Резерви часу для кожної роботи показані штриховими лініями. За результатами діаграми корегується навантаження робіт на бригади та вишуковуються можливості їх

більш рівномірної роботи, а також скорочення кількості задіяних бригад.

Сітьові графіки використовують також для *оптимізації технологічного процесу на сортувальних станціях*, який полягає у скороченні тривалості переробки груп вагонів для накопичування їх у поїзди, і відповідно у скороченні тривалості простою вагонів при очікуванні переробки. Важливим напрямком використання сітьових графіків є *оптимізація технології обслуговування під'їзних колій навантажувальних станцій* з метою зменшення часу простою вагонів.

Але в сучасних умовах підвищених вимог до гнучкості схем роботи залізничних станцій, збільшення числа компонентів, що входять до системи, яка проектується, а також ускладнення причинно-наслідкових зв'язків у комплексі компонентів, які взаємодіють, взаємопов'язані та взаємозалежні між собою, описані графи вже не задовольняють ці вимоги.

Останнім досягненням в теорії графів є поява на початку 1960-х років теорії сітей Петрі, які на сучасний момент вже мають широке практичне застосування в багатьох галузях (інформатика, соціологія, хімія, медицина тощо), зокрема, в транспортних системах.

3.5 ПОНЯТТЯ ПРО СІТКИ ПЕТРІ ТА ПРИНЦИПИ МОДЕЛЮВАННЯ ТРАНСПОРТНОЇ СИСТЕМИ ЗА ДОПОМОГОЮ СІТОК ПЕТРІ

Особливістю сіток Петрі є їх придатність бути інструментом графічного зображення системи у вигляді блок-схеми, у якій діють *паралельні процеси*, тобто *паралельна обробка інформації*, а також об'єкти, що безпосередньо *паралельно діють*.

Сітки Петрі визначаються як модифікований, багатофункціональний, зважений дводольний граф. Останнє означає, що всі вершини графа відносяться до двох типів: *позиції (місця) P та переходи V* . Позиції зображуються кільцем, а переходи – відрізками прямих. Дуги у сітках Петрі – напрямні. Причому кожна дуга зв'язує вершини тільки різних типів. Дуги позначаються відповідними значеннями ваги (цілими числами), і

дугу з вагою k можна вважати еквівалентною k паралельним дугам. У задачах моделювання, де використовуються поняття умов (станів) і подій, позиції (місця) відповідають умовам, а переходи – подіям. Кожний перехід (подія) пов'язаний з певним числом вхідних і вихідних позицій – аналогів відповідно передумов та післяумов цієї події. На рисунку 3.6 наведені приклади зображення дводольних графів.

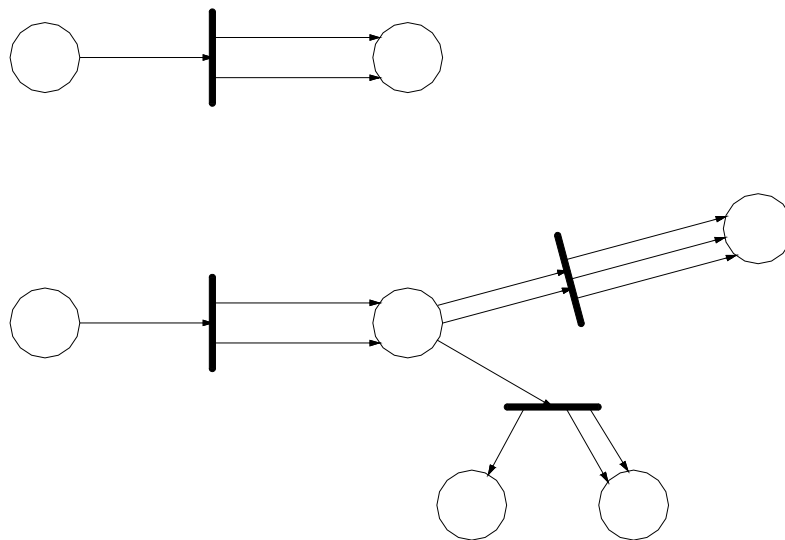


Рисунок 3.6

Оригінальним поняттям сіток Петрі є поняття «фішки». Фішки зображуються точками, розташованими всередині позицій. Таким чином, кожній позиції сітки ставиться у відповідність натуральне число, яке вказує кількість фішок у даній позиції – *розмітку позиції*. Сукупність таких чисел для всіх позицій сітки називають *розміткою сітки*. Якщо позиція не містить фішок, то вона має нульову розмітку. Приклад сітки з розміткою наведений на рисунку 3.7.

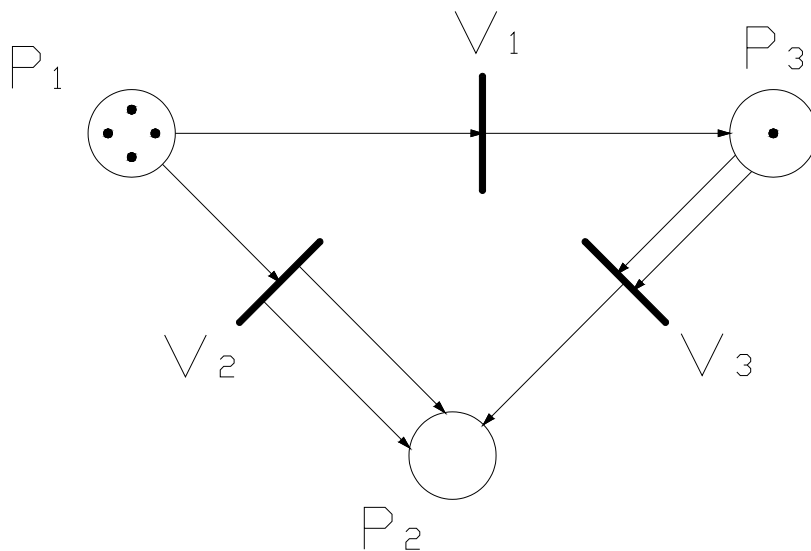


Рисунок 3.7

Важливим поняттям сіток Петрі є поняття «спрацьовування переходів», яке полягає у вилученні фішок з кожної вхідної позиції та переміщенні їх у кожну вихідну позицію. Причому кількість фішок, які вилучаються з конкретної позиції або переміщуються у конкретну позицію, дорівнює кількості дуг, що з'єднують перехід, який спрацьовує, із даною конкретною позицією. Так, наприклад, перехід V_1 на рисунку 3.7 при спрацьовуванні вилучає з позиції P_1 одну фішку та збільшує кількість фішок у позиції P_3 на одну. Перехід V_2 вилучає одну фішку з позиції P_1 та переміщує у позицію P_2 дві фішки.

Сітка Петрі функціонує, переходячи від розмітки до розмітки, наприклад, від деякої початкової до деякої проміжної або до заключної – тупикової розмітки, тобто такої, коли не може спрацьовувати жодний перехід. Це означає, що спрацьовування переходів змінює розмітку сітки. Власно кажучи, предметом теоретичного дослідження сіток Петрі є процес їх функціонування, тобто аналіз можливих послідовностей спрацьовування переходів і властивості розміток сітки, які при цьому отримують.

Інтерпретація спрацьовування переходів може бути сформульована так. Перехід може спрацьовувати тоді і тільки тоді, коли всі його вхідні місця заповнені хоча б однією фішкою. Перехід при спрацьовуванні вилучає рівно по одній фішці з кожного свого вхідного місця і утворює рівно по одній новій

фішці в кожному своєму вихідному місці. Якщо в сітці Петрі є один перехід, який може спрацьовувати, він обов'язково це зробить. Якщо можуть спрацьовувати два або більше переходів, то вони спрацьовуватимуть послідовно, але не детерміновано, тобто не можна заздалегідь знати, який перехід повинен спрацьовувати. Наприклад, на рисунку 3.7 перехід V_3 не може спрацьовувати, оскільки у позиції P_3 міститься тільки одна фішка, а дуг, що пов'язують P_3 і V_3 – дві. Але якщо спрацьовував перехід V_1 , то внаслідок збільшення кількості фішок у позиції P_3 перехід V_3 спрацьовуватиме. Тобто саме введення поняття «спрацьовування переходів» дає підставу для моделювання процесів, які пов'язані між собою причинно-наслідковим зв'язком.

Сіткам Петрі притаманні певні *властивості*. Деякі з них:

Безпека. Місце сітки Петрі, яке містить не більше однієї фішки, називається безпечним. Сітка Петрі безпечна, коли безпечні всі місця сітки.

Обмеженість. Властивість не допускати перевищення кількості фішок у конкретній або довільній позиції деякого довільного числа. Місце є k -безпечним або k -обмеженим, якщо кількість фішок у ньому не перебільшує цілого числа k .

Одна з основних проблем у теорії сіток Петрі – задача про скінченність функціонування сітки, тобто про досяжність тупикової розмітки.

Формально сітку Петрі N визначають п'ятіркою значень

$$N = \langle P, V, F, H, M_0 \rangle, \quad (3.6)$$

де P – скінченна, непуста множина *місць* або *позицій* $\{P_1, P_2, \dots, P_n\}$; V – скінченна непуста множина *переходів* $\{V_1, V_2, \dots, V_m\}$; F – *функція інцидентності*, яка вказує на наявність дуг, що з'єднують *місця* з *переходами*; H – *функція інцидентності*, яка вказує на наявність дуг, що з'єднують *переходи* з *місцями*; M_0 – початкова розмітка сітки.

Функції інцидентності $F(P_i, V_j)$ ($H(V_i, P_j)$) складаються з множини двох значень $\{0, 1\}$: $F(P_i, V_j) = 1$, коли з вершини-місця P_i прямує дуга до вершини-переходу V_j ($H(V_i, P_j) = 1$, коли з вершини-переходу V_i прямує дуга до вершини-місця P_j), у

протилежних випадках, тобто коли відповідних дуг не існує, $F(P_i, V_j) = 0$ ($H(V_i, P_j) = 0$).

Розглянемо довільну сітку Петрі (рисунок 3.8), яка відповідно до форми завдання (3.1) може бути описаною так:

$$P = \{P_1, P_2, P_3, P_4\} \text{ – множина позицій (місць);}$$

$$V = \{V_1, V_2, V_3\} \text{ – множина переходів;}$$

початкова розмітка M_0 :

$M_0 =$	P_1	P_2	P_3	P_4
	2	0	1	0

функції інцидентності:

$F:$	V_1	V_2	V_3
P_1	1	1	0
P_2	0	0	1
P_3	0	0	1
P_4	0	1	0

$H:$	P_1	P_2	P_3	P_4
V_1	0	1	0	1
V_2	0	0	1	0
V_3	1	0	0	0

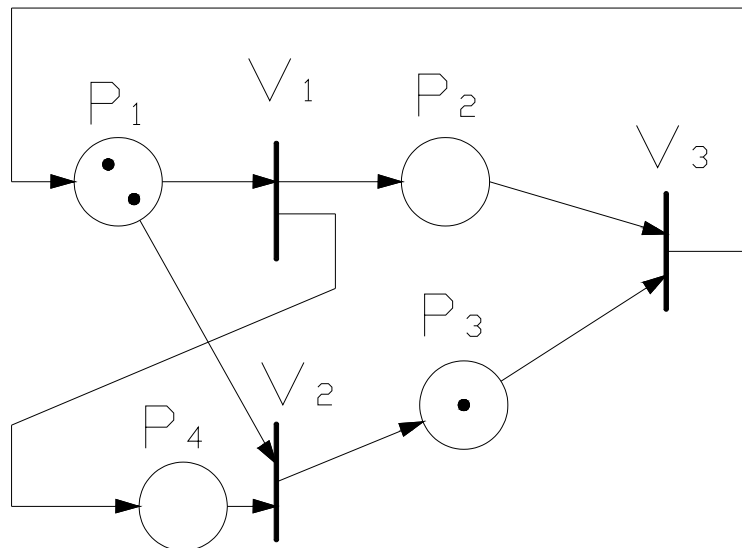


Рисунок 3.8

Результати функціонування сітки Петрі зведені у таблицю 3.1, де наведені декілька можливих змін розміток сітки, починаючи з M_0 .

Таблиця 3.1

Розмітка		Позиції (місця)			
		P_1	P_2	P_3	P_4
M_0		2	0	1	0
M_1	$V_1 \downarrow$	1	1	1	1
M_2	$V_2 \downarrow$	0	1	2	0
M_3	$V_3 \downarrow$	1	0	1	0
M_4	$V_1 \downarrow$	0	1	1	1
M_5	$V_3 \downarrow$	1	0	0	1
M_6	$V_1 V_2$	0	1	0	2
M_7		0	0	1	0

Пояснення до таблиці. При початковій розмітці (перший рядок) лише один перехід V_1 може спрацьовувати. Відбувається зміна розміток (другий рядок). При розмітці M_1 до спрацьовування готові всі три переходи V_1 , V_2 , V_3 , але може спрацьовувати лише один, нехай "випадково" спрацює перехід V_2 , тоді утворилась розмітка M_2 , при якій може спрацьовувати тільки перехід V_3 , і т. ін. Останній крок при розмітці M_5 у випадках спрацьовування одного з двох можливих переходів призводить до тупикових розміток: відповідно M_6 або M_7 .

Можна повернутися до розмітки M_1 , коли до спрацьовування готові всі три переходи, і самотійно розглянути кроки функціонування сіток Петрі у випадках, коли можуть спрацьовувати переходи V_1 або V_3 , і побудувати аналогічні таблиці розміток.

Дерево досяжності – це множина досяжності розміток сітки Петрі. Початковою вершиною дерева є вершина, яка відповідає початковій розмітці M_0 . Кожна можлива послідовність розміток до своєї тупикової утворює одну гілку дерева досяжності (наприклад, послідовність у наведеній таблиці 3.1) з вершинами, які відповідають розміткам на кожному кроці і поєднані стрілками. Дерево "розростається" при розгляданні спрацьовування всіх можливих переходів на кожному кроці.

4 ЗАДАЧІ ДИНАМІЧНОГО ПРОГРАМУВАННЯ

Задачі динамічного програмування (ЗДП) – це особливий клас задач математичного програмування, зокрема, лінійного програмування, математичний апарат яких дозволяє здійснювати планування *керованих процесів*. До означених процесів відносяться такі, на рух яких можна цілеспрямовано впливати всередині задачі, що розглядається. Тобто метод динамічного програмування – це метод оптимізації, спеціально пристосований до операцій, у яких процес прийняття рішень може бути розподілений на окремі етапи (кроки). Такі *операції* називають *багатокроковими*.

До *особливостей* ЗДП можна віднести такі:

1 Нехай описується деяка система S , яка характеризується певним набором параметрів X , які вважаються *множиною станів* системи. У такій системі відбуваються деякі процеси (виробничі, економічні, тощо), які представляють як *багатокрокові*. Це дає підставу подавати набір параметрів X на кожному кроці розв'язання задачі

$$X = \{X_1, X_2, \dots, X_k, \dots, X_N\},$$

де N – кількість етапів (кроків), на які розбивається задача; X_k – множина станів системи S на початку k -ого етапу,

$$k = 1 \div N.$$

2 Оскільки керування системою здійснюється для досягнення конкретної мети, то вводиться показник ефективності управління або *цільова функція на кожному етапі* або кроці оптимізації $f(X_k)$.

3 Розрізняють розв'язання ЗДП з алгоритмом *прямої прогонки* (а) та алгоритм *зворотної прогонки* (б).

4 Цільові функції на кожному етапі носять *накопичувальний* або *рекурентний характер*. Це означає, що в залежності від обраного алгоритму вони: у випадку а) виражають вимоги до задачі на першому етапі планування ($k=1, f_1(X_1)$); перших двох етапах планування ($k=2, f_2(X_2)$), ..., на перших k етапах планування ($k=k, f_k(X_k)$) та остаточно – на всіх N етапах планування ($k=N, f_N(X_N)$); у випадку б) – вимоги до задачі на N -

му етапі планування ($k = N, f_N(X_N)$), на N -му та $(N-1)$ -му етапах планування ($k = N-1, f_{N-1}(X_{N-1})$), ..., на всіх N етапах планування ($k = 1, f_1(X_1)$). Таким чином, останнє значення цільової функції одразу дає її сумарну оптимальну величину за весь період планування.

Зауваження 1. Здатність цільових функцій до накопичувального характеру досягається включенням у вираз для цільової функції поточного етапу оптимального значення цільової функції попереднього етапу, за виключенням першої за порядком цільової функції.

Зауваження 2. Вибір алгоритму, що означений пунктом 3, пов'язаний з початковими умовами, які дозволяють записати вираз для $f_1(X_1)$ (для алгоритму прямої прогонки) або для $f_N(X_N)$ (для алгоритму зворотної прогонки) у спрощеному вигляді, тобто без урахування її попереднього значення.

5 В ЗДП вводиться множина U_k – *множина управлінь задачею на кожному етапі планування*, під впливом якої система S переходить до свого наступного стану.

6 Обчислювані процедури динамічного програмування ґрунтуються на *рекурентних співвідношеннях Беллмана*, побудованих на принципах, означених у пунктах 1...5.

7 Розв'язання ЗДП включає в себе знаходження не тільки *оптимального значення цільової функції* за весь період планування, але й *вектора оптимальних управлінь задачею*

$$U^* = (U_1^*, U_2^*, \dots, U_N^*),$$

або стратегії управління задачею.

Отже, запишемо у найбільш загальному вигляді цільові функції задачі детермінованого динамічного програмування у вигляді співвідношень Беллмана.

Для алгоритму прямої прогонки

$$f_1(X_1) = \max_{U_1} (\min) \{z_1(X_1, U_1)\}; \tag{4.1}$$

$$f_k(X_k) = \max_{U_k} (\min) \{z_k(X_k, U_k) + f_{k-1}(X_k)\}, \quad k = 2 \div N.$$

Для алгоритму зворотної прогонки

$$f_N(X_N) = \max_{U_N} (\min) \{z_N(X_N, U_N)\};$$

$$f_k(X_k) = \max_{U_k} (\min) \{z_k(X_k, U_k) + f_{k+1}(X_k)\}, \quad k = 1 \div N - 1.$$
(4.2)

4.1 ЗАДАЧА ПРО НАЙКОРОТШИЙ ШЛЯХ ЯК СІТЬОВА ЗДП

Задача про найкоротший шлях як сітьова задача надає наочності сутності методу динамічного програмування та висвітлює всі особливості методу.

Формулювання задачі зводиться до знаходження на сітці доріг найбільш економічного маршруту перевезення вантажу з початкового пункту до кінцевого. Сітка доріг зображується зваженим графом, де у ролі ваги ребер обирається або вартість доставки вантажу по відповідному маршруту, або відстань чи час руху між відповідними пунктами.

Концепція методу динамічного програмування передбачає надання процесові оптимізації покрокового характеру. З цією метою всі пункти сітки розбиваються на групи так, що до кожної i -тої групи як початкові пункти входять кінцеві пункти $i-1$ -ї групи, а як кінцеві – тільки ті пункти, що пов'язані шляхами з початковими тієї ж групи.

Для формування цільових функцій задачі можна обирати алгоритми як прямої, так і зворотної прогонки, тому що відсутні будь-які вимоги до економічності шляху до початкового і після кінцевого пунктів. Отже, обираємо алгоритм прямої прогонки. І тоді при наявності k кроків планування маршруту, використовуючи вирази (4.1), цільові функції для першого етапу планування і для інших $k-1$ етапів запишуться у вигляді співвідношень Беллмана

$$f_1(X^1) = \min_{U_1} \{d(x_{i-1}^1; x_i^1)\},$$

$$f_k(X^k) = \min_{U_k} \{d(x_{i-1}^k; x_i^k) + f_{k-1}(X^k)\}, \quad k = 2 \div N,$$
(4.3)

де $d(x_{i-1}^k; x_i^k)$ – відстань між відповідними початковими та кінцевими пунктами k -го кроку маршруту; сукупність U_k компонент вектора управління відбиває оптимальний маршрут (перелік пунктів проходження) на k -му кроці планування.

Приклад 4.1. На заданій сітці доріг є декілька маршрутів, по яких можна доставляти вантаж від пункту 1 до пункту 7. Відомі вартості перевезень одиниці вантажу між пунктами сіті, які позначені як ваги ребер графа (рисунок 4.1). Методом динамічного програмування знайти на сітці найбільш економний маршрут доставки вантажу з пункту 1 до пункту 7.

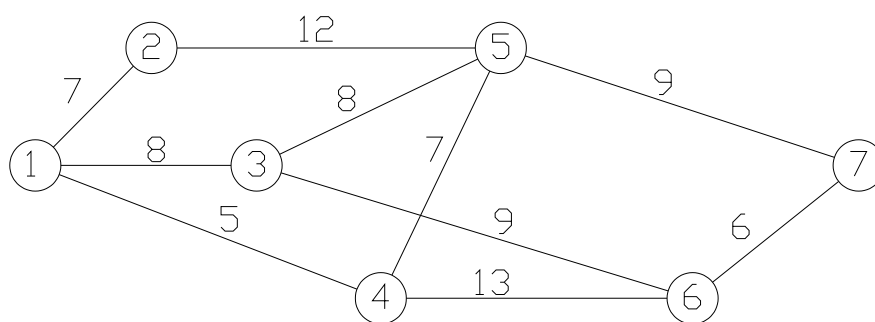


Рисунок 4.1

На сітці рисунка 4.2 показано, як маршрут перевезення вантажу розбивається на три кроки.

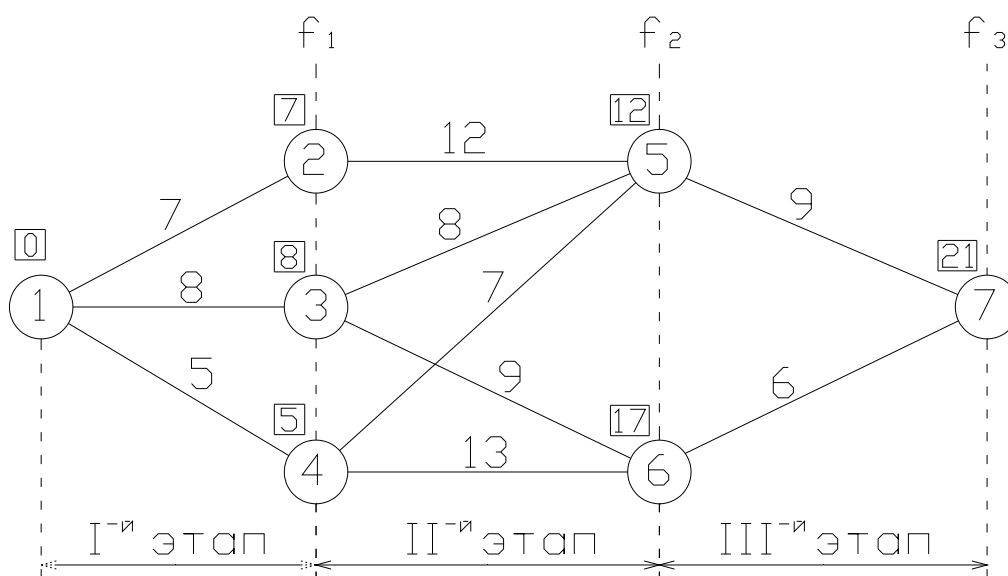


Рисунок 4.2

Запишемо цільові функції задачі, розкриваючи зміст позначень у виразах (4.5).

1 $k = 1$ – планування першого етапу, який включає три цільові функції

$$\left\{ \begin{array}{l} \text{найкоротший} \\ \text{шлях} \\ \text{до } n.2 \end{array} \right\} = \min_{U_1^1: i=1} \{ \text{шлях } _ \text{до } n.2 \} = 7, \quad U_1^1 : 1-2;$$

$$\left\{ \begin{array}{l} \text{найкоротший} \\ \text{шлях} \\ \text{до } n.3 \end{array} \right\} = \min_{U_1^2: i=1} \{ \text{шлях } _ \text{до } n.3 \} = 8, \quad U_1^2 : 1-3;$$

$$\left\{ \begin{array}{l} \text{найкоротший} \\ \text{шлях} \\ \text{до } n.4 \end{array} \right\} = \min_{U_1^3: i=1} \{ \text{шлях } _ \text{до } n.4 \} = 5, \quad U_1^3 : 1-4.$$

2 $k = 2$ – планування перших двох етапів, який включає дві цільові функції

$$\left\{ \begin{array}{l} \text{найкоротший} \\ \text{шлях} \\ \text{до } n.5 \end{array} \right\} = \min_{U_2^1: i=2,3,4} \left\{ \left(\begin{array}{l} \text{найкоротший} \\ \text{шлях} \\ \text{до } _ \text{вузла } _ i \end{array} \right) + \left(\begin{array}{l} \text{відстань} \\ \text{від } _ \text{вузла } _ i \\ \text{до } n.5 \end{array} \right) \right\} = \min_{U_2^1: i=2,3,4} \left\{ \begin{array}{l} 7+12 \\ 8+8 \\ 5+7 \end{array} \right\} = 12;$$

$$U_2^1 : 1-2-5;$$

$$\left\{ \begin{array}{l} \text{найкоротший} \\ \text{шлях} \\ \text{до } n.6 \end{array} \right\} = \min_{U_2^2: i=3,4} \left\{ \left(\begin{array}{l} \text{найкоротший} \\ \text{шлях} \\ \text{до } _ \text{вузла } _ i \end{array} \right) + \left(\begin{array}{l} \text{відстань} \\ \text{від } _ \text{вузла } _ i \\ \text{до } n.6 \end{array} \right) \right\} = \min_{U_2^2: i=3,4} \left\{ \begin{array}{l} 8+9 \\ 5+13 \end{array} \right\} = 17;$$

$$U_2^2 : 1-3-6.$$

3 $k = 3$ – планування всіх трьох етапів

$$\left\{ \begin{array}{l} \text{найкоротший} \\ \text{шлях} \\ \text{до } n.7 \end{array} \right\} = \min_{U_3: i=5,6} \left\{ \left(\begin{array}{l} \text{найкоротший} \\ \text{шлях} \\ \text{до } _ \text{вузла } _ i \end{array} \right) + \left(\begin{array}{l} \text{відстань} \\ \text{від } _ \text{вузла } _ i \\ \text{до } n.7 \end{array} \right) \right\} = \min_{U_3: i=5,6} \left\{ \begin{array}{l} 12+9 \\ 17+6 \end{array} \right\} = 21;$$

$$U_3 : 1-4-5-9.$$

Отже, *найкоротший шлях* від пункту 1 до пункту 7 складає

$$f_{\min} = 21 \text{од.},$$

який досягається такою *оптимальною стратегією управління задачею*

$$U^* : 1-4-5-9.$$

Серед *інших задач динамічного програмування* виділимо *задачу розподілу ресурсів та задачу керування запасами*, що відносяться, як і попередня, до *лінійних детермінованих ЗДП*.

До *лінійних стохастичних ЗДП* можна віднести *марковську задачу прийняття рішень*. Постановка та приклади перелічених задач надані у методичних вказівках [3, 5]. Але для того, щоб сприйняти логіку постановки динамічної задачі прийняття рішення, треба зануритися в усю тему “Задачі прийняття рішень”, яка включає в себе три типи постановок задач, які відрізняються рівнем доброякісності даних, які використовуються при прийнятті рішень, а саме [7]:

- *задачі прийняття рішень в умовах визначеності;*
- *задачі прийняття рішень в умовах ризику;*
- *задачі прийняття рішень в умовах невизначеності.*

У першому випадку всі ідеї, почуття, емоції відносно прийняття рішення можна охарактеризувати значеннями *вагових коефіцієнтів на користь тієї або іншої альтернативи*, які знаходяться за допомогою складання *матриці порівнянь*. При цьому розповсюдженням є *метод ієрархій прийняття рішень в умовах невизначеності*.

Ситуація прийняття рішень в умовах ризику вимагає визначення *платежів*, тобто *грошових критеріїв* прийняття рішення в тих чи інших обставинах, в яких вартості альтернативних рішень зазвичай описуються ймовірнісними розподілами. З цієї причини рішення, що приймається, ґрунтується на використанні *критерію очікуваного значення*, відповідно до якого альтернативні рішення порівнюються з точки зору максимізації очікуваного прибутку або мінімізації очікуваних витрат.

Прийняття рішень в умовах невизначеності, як і в умовах ризику, вимагає визначення альтернативних дій, яким відповідають платежі, що залежать від випадкових обставин – станів природи. Коло станів природи є відомим, але їх неможливо описати ані у детермінованому, ані у ймовірнісному сенсі. Постановка та можливі критерії прийняття рішень надані у методичних вказівках [5].

Задачі прийняття рішень докладно розглядаються в рамках дисципліни “Дослідження операцій у транспортних системах”.

Розглянемо задачу прийняття рішення в умовах невизначеності у динамічній постановці.

4.2 ПРИКЛАД ПОСТАНОВКИ ТА РОЗВ’ЯЗАННЯ МАРКОВСЬКОЇ ЗАДАЧІ ПРИЙНЯТТЯ РІШЕНЬ

Приклад 4.2 Транспортне підприємство висуває на ринок новий вид продукції. Якщо обсяг збуту високий, то з ймовірністю 0.5 він залишиться таким же в наступному місяці. Якщо він невисокий, то ймовірність того, що він стане високим у наступному місяці, дорівнює тільки 0.2. Підприємство може провести рекламну кампанію. Якщо вона прийме таке рішення при високому рівні збуту, то ймовірність того, що він залишиться високим і в наступному місяці, зростає до 0.8. При низькому рівні реклама збільшує цю ймовірність тільки до 0.4.

Якщо при високому рівні збуту реклама не використовується, то очікуваний прибуток складе 10 гр. од. за умови, що обсяг збуту залишиться високим у наступному місяці, і 4 гр. од. – у протилежному випадку. Якщо первинно спостерігається низький рівень збуту, то відповідні прибутки дорівнюють відповідно 7 гр. од. та -2 гр. од. (фактично це означає втрату прибутку). При використанні реклами прибуток складає 7 гр. од., якщо первинно рівень збуту був і залишився високим, і стає рівним 6 гр. од., якщо рівень збуту знижується. Якщо первинний рівень збуту низький, то прибутки дорівнюють 3 гр. од. та -5 гр. од. у залежності від того, підвищується він чи ні наступного місяця.

Необхідно визначити оптимальну стратегію поведінки транспортного підприємства відносно використання реклами впродовж наступних трьох місяців та відповідні сумарні доходи у випадках різних первинних станів ринку збуту.

Нагадаємо, що сформульована задача – це задача прийняття рішень в умовах невизначеності станів системи, тобто об'єктивних обставин, з якими стикається фізична чи юридична особа, що приймає рішення. Динамічність задачі полягає у відомості того, що ці стани мають тенденцію змінюватися на кінцевих етапах планування задачі, що сумарно утворюють весь період планування. Тенденція зміни носить відомий ймовірнісний характер.

Цей приклад, у протилежність розглянутому у методичних вказівках [5], де початкові умови формалізовані, потребує складення математичної моделі постановки задачі.

Визначаємо наявність двох альтернатив прийняття рішення:

$m_1 = 1$ – невикористання реклами;

$m_2 = 2$ – використання реклами.

Математична модель постановки задачі потребує створення двох типів матриць для кожної з можливих альтернатив поведінки:

P^m – перехідних матриць станів системи;

R^m – матриць прибутків системи.

Перехідні матриці складаються на основі припущення про марковський характер випадкового процесу – випадкової зміни станів системи, які утворюють ситуацію прийняття рішень в умовах невизначеності наявності того чи іншого стану, а саме обставин на ринку – високого чи низького рівня збуту. Нехай

S_1 – стан високого рівня збуту;

S_2 – стан низького рівня збуту.

Тоді, вважаючи одним часовим кроком – місяць, формалізуємо початкові дані відносно перехідних ймовірностей марковського ланцюга з дискретним часом, тобто в задачі відомо:

$$p_{11}^1 = 0,5; \quad p_{21}^1 = 0,2 .$$

Знаходимо

$$p_{12}^1 = 1 - 0,5 = 0,5; \quad p_{22}^1 = 1 - 0,2 = 0,8 .$$

Аналогічно сприймаємо дані для другої альтернативи, а саме

$$p_{11}^2 = 0,8 \Rightarrow p_{12}^2 = 1 - 0,8 = 0,2 ;$$

$$p_{21}^2 = 0,4 \Rightarrow p_{22}^2 = 1 - 0,4 = 0,6 .$$

Таким чином, визначені елементи перехідних матриць станів ринку за один місяць, тобто матриці мають вигляд

$$P^1 = \begin{pmatrix} 0,5 & 0,5 \\ 0,2 & 0,8 \end{pmatrix}; \quad P^2 = \begin{pmatrix} 0,8 & 0,2 \\ 0,4 & 0,6 \end{pmatrix}.$$

Відповідні матриці прибутків у рамках обраних альтернатив поведінки відносно проведення рекламної кампанії, виходячи з умов задачі, такі:

$$R^1 = \begin{pmatrix} 10 & 4 \\ 7 & -2 \end{pmatrix}; \quad R^2 = \begin{pmatrix} 7 & 6 \\ 3 & -5 \end{pmatrix}.$$

Нагадаємо (дивись методичні вказівки [5]) правило створення співвідношень Беллмана для стохастичної динамічної задачі прийняття рішень.

Якщо для m можливих альтернатив прийняття рішення в рамках стикання з k можливими об'єктивними обставинами (станами природи) перехідні матриці станів та матриці платежів мають вигляд

$$P^m = \left\{ p_{ij}^m \right\}_{[k \times k]}; \quad R^m = \left\{ r_{ij}^m \right\}_{[k \times k]},$$

то для алгоритму зворотної прогонки (який використовується в цій задачі) у структуру виразів (4.2) вводиться ймовірнісний множник, а саме

$$f_N(i) = \max_i \left\{ \sum_{j=1}^k p_{ij}^m \cdot r_{ij}^m \right\}; \quad (4.4)$$

$$f_n(i) = \max_i \left\{ \sum_{j=1}^k p_{ij}^m (r_{ij}^m + f_{n+1}(j)) \right\}, \quad n = 1 \div N - 1,$$

де N – номер етапу планування, i (j) – номер стану системи.

У виразах (4.4) основний доданок, а саме цільову функцію, яка відбиває середньоймовірнісний прибуток за один етап планування, позначимо v_i^m , тобто

$$v_i^m = \sum_{j=1}^k p_{ij}^m \cdot r_{ij}^m.$$

Прорахуємо ці значення окремо, ще раз пояснюючи позначення: наприклад, v_1^2 – це середньомісячний прибуток (у ймовірнісному сенсі) при ситуації високого рівня збуту на ринку ($i = 1$) з проведенням рекламної кампанії ($m = 2$).

$$\begin{aligned} v_1^1 &= 0,5 \cdot 10 + 0,5 \cdot 4 = 7; \\ v_2^1 &= 0,2 \cdot 7 + 0,8 \cdot (-2) = -0,2; \\ v_1^2 &= 0,8 \cdot 7 + 0,2 \cdot 6 = 6,8; \\ v_2^2 &= 0,4 \cdot 3 + 0,6 \cdot (-5) = -1,8. \end{aligned}$$

Використовуючи алгоритм зворотної прогонки, приступаємо до планування кожного етапу, починаючи з останнього

1 $k = 3$ – планування третього місяця:

Стан системи (i)	$f_3(i) = v_i^m$		Оптимальний розв'язок	
	Альтернативи		$f_3^*(i) - \max$	m^*
	$m = 1$	$m = 2$		

1	7	6,8	7	1
2	-0,2	-1,8	-0,2	1

2 $k = 2$ – планування другого та третього місяців:

Стан системи (i)	$f_2(i) = v_i^m + \sum_{j=1}^2 p_{ij}^m \cdot f_3^*(j)$		Оптимальний розв'язок	
	Альтернативи			
	m = 1	m = 2	$f_2^*(i) - \max$	m^*
1	$7 + 0,5 \cdot 7 + 0,5 \cdot (-0,2) = 10,4$	$6,8 + 0,8 \cdot 7 + 0,2 \cdot (-0,2) = 12,36$	12,36	2
2	$-0,2 + 0,2 \cdot 7 + 0,8 \cdot (-0,2) = 1,04$	$-1,8 + 0,4 \cdot 7 + 0,6 \cdot (-0,2) = 0,88$	1,04	1

3 $k = 1$ – планування всіх трьох місяців:

Стан системи (i)	$f_1(i) = v_i^m + \sum_{j=1}^2 p_{ij}^m \cdot f_2^*(j)$		Оптимальний розв'язок	
	Альтернативи			
	m = 1	m = 2	$f_1^*(i) - \max$	m^*
1	$7 + 0,5 \cdot 12,36 + 0,5 \cdot 1,04 = 13,7$	$6,8 + 0,8 \cdot 12,36 + 0,2 \cdot 1,04 = 16,896$	16,896	2
2	$-0,2 + 0,2 \cdot 12,36 + 0,8 \cdot 1,04 = 3,104$	$-1,8 + 0,4 \cdot 12,36 + 0,6 \cdot 1,04 = 3,768$	3,768	2

Таким чином, у випадку сприятливої ситуації на ринку на початок планування прибуток транспортного підприємства за три місяці складе майже 16,9 гр. од. за рахунок проведення рекламної кампанії в перші два місяці виробничої діяльності (на початку третього місяця реклама не використовується). При несприятливому збігу обставин на споживчому ринку на початку планування сумарний прибуток, що очікується, складе майже 3,8 гр. од., причому для досягнення оптимального результату достатньо провести рекламну кампанію тільки в перший місяць виробничої діяльності.

Ще раз наголошуємо, що відповідь задачі динамічного програмування містить оптимальне значення цільової функції та стратегію управління задачею для досягнення цього значення.

5 ЗАДАЧІ ТЕОРІЇ МАСОВОГО ОБСЛУГОВУВАННЯ В ШИРОКОМУ КОЛІ ЇХ ПОСТАНОВОК ТА ПРИКЛАДІВ

Цей розділ конспекту лекцій ґрунтується на вже надрукованих конспектах лекцій з тем “Елементи теорії марковських ланцюгів” [7] та “Елементи теорії масового обслуговування” [8], а також на методичних вказівках [5]. Ці роботи призначені для вивчення розділу „Елементи теорії масового обслуговування” в курсі „Дослідження операцій у транспортних системах” другого курсу і повинні стати обов’язковим матеріалом для повторення як дуже важлива частина освіти студента-магістра в рамках курсу, який є темою даного конспекту лекцій.

Нагадаємо, що предметом дослідження теорії масового обслуговування (ТМО) є організація роботи систем масового обслуговування (СМО), до яких відносяться майже всі технологічні складові функціонування залізничного транспорту. Стисло за мету організації роботи СМО вважаємо розв’язання проблеми черг, тобто мінімізацію часу очікування в системі, не допускаючи простою безпосередньо каналів обслуговування.

Зручною є класифікація СМО за двома напрямками:

1) за типом черги:

- з необмеженою чергою;
- з обмеженою чергою;
- з відмовами;

2) за кількістю каналів обслуговування:

- одноканальні СМО;
- багатоканальні СМО.

Саме у ракурсі такої класифікації слід розглядати повторення відповідного матеріалу за вказаною методичною літературою.

Включимо у зміст даного конспекту лекцій постановку, методи розв'язання деяких практичних задач ТМО, які використовують та доповняють вже отримані знання з цієї теми.

5.1 ПРИКЛАД ПОСТАНОВКИ І РОЗВ'ЯЗАННЯ ЗАДАЧІ ЗНАХОДЖЕННЯ ЙМОВІРНОСТЕЙ СТАНІВ У СИСТЕМІ ОБІГУ ВАГОНІВ НА ЗАЛІЗНИЧНОМУ ВУЗЛІ

Даний приклад відбиває можливість опису роботи залізничного вузла за допомогою випадкового процесу – *марковського ланцюга з неперервним часом*, який додатково має властивості *однорідності та ординарності*.

Нагадаємо, що для системи S , яка може перебувати у кінцевій кількості дискретних станів, ймовірнісні характеристики переходу зі стану в стан у момент часу t залежать тільки від того стану, в якому система знаходилась в цей момент, і не залежать від того, як і коли система потрапила у цей стан. Ця властивість, яку можна висловити *логічним висловлюванням* „майбутнє залежить тільки від теперішнього, а не від минулого”, і є основною *властивістю марковського ланцюга*.

Властивість переходити зі стану в стан у будь-який момент часу інтервалу, що розглядається, відносить цей процес до *марковського ланцюга з неперервним часом* (МЛ з НЧ).

Властивість *однорідності*, або *стаціонарності* МЛ з НЧ, пов'язана із тим, що його ймовірнісні характеристики переходу зі стану в стан не залежать від відлікового часу, а залежать тільки від тривалості інтервалу часу.

І насамкінець, властивість *ординарності* полягає у припущенні, що за малий проміжок часу, який розглядається, система може здійснити не більше одного переходу зі стану в стан, або, іншими словами, здійснити не більше одного стрибка.

Математичний апарат теорії марковських ланцюгів з неперервним часом дає можливість скласти лінійні диференціальні рівняння Колмогорова для знаходження ймовірностей станів $\vec{p}(t) = (p_1(t), p_2(t), \dots)$

$$\bar{p}'(t) = \bar{p}(t) \cdot \Lambda, \quad (5.1)$$

(Λ - матриця-генератор інтенсивностей станів), а також лінійні алгебраїчні рівняння для фінальних ймовірностей станів $\bar{p}^* = (p_1^*, p_2^*, \dots)$

$$\bar{p} \cdot \Lambda = 0, \quad (5.2)$$

відбивають відносний час перебування системи у кожному з цих станів для граничного або усталеного режиму функціонування системи.

Приклад 5.1. Нехай система S – вантажна станція, вагони на якій можуть знаходитися у семи станах S_i $i = 1 \div 7$: S_1 – вагони у стані розвантаження; S_2 – вагони у стані навантаження; S_3 – вагони у стані сортування; S_4 – резерв порожніх вагонів; S_5 – вагони у складі парку непрацюючих вагонів; S_6 – вагони у ремонтному депо; S_7 – стан знаходження вагонів поза вузлом. Скласти рівняння Колмогорова для визначення ймовірностей P_i , $i = 1 \div 7$ знаходження системи у станах S_i у довільний момент часу t при відомих значеннях інтенсивностей λ_{ij} переходів системи у відповідні стани за певний часовий інтервал спостережень.

Можливі переходи зі стану в стан зображуються у вигляді зваженого графа марковського ланцюга системи S , який інакше можна назвати графом станів обігу вагонів. Загальний вигляд графа обігу в з усіма можливими переходами показаний на рисунку 5.1.

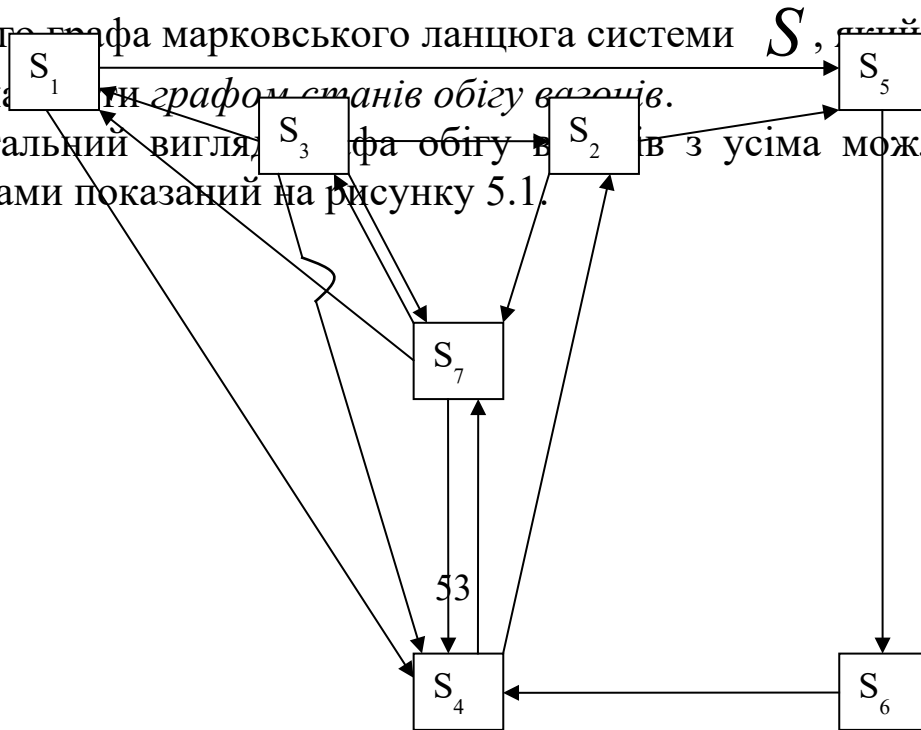


Рисунок 5.1

Згідно з формулою (5.1) диференціальне матричне рівняння Колмогорова для ймовірностей станів $p_i(t)$ запишеться у вигляді (випускаємо у запису залежність від t)

$$(p'_1, p'_2, p'_3, p'_4, p'_5, p'_6, p'_7) = (p_1, p_2, p_3, p_4, p_5, p_6, p_7) \times$$

$$\begin{pmatrix} \lambda_{11} & 0 & 0 & \lambda_{14} & \lambda_{15} & 0 & 0 \\ 0 & \lambda_{22} & 0 & 0 & \lambda_{25} & 0 & \lambda_{27} \\ \lambda_{31} & \lambda_{32} & \lambda_{33} & \lambda_{34} & 0 & 0 & \lambda_{37} \\ 0 & \lambda_{42} & 0 & \lambda_{44} & 0 & 0 & \lambda_{47} \\ 0 & 0 & 0 & 0 & \lambda_{55} & \lambda_{56} & 0 \\ 0 & 0 & 0 & \lambda_{64} & 0 & \lambda_{66} & 0 \\ \lambda_{71} & 0 & \lambda_{73} & \lambda_{74} & 0 & 0 & \lambda_{77} \end{pmatrix}, \quad (5.3)$$

де згідно з правилом формування матриці-генератора,

$$\lambda_{11} = -(\lambda_{14} + \lambda_{15}); \quad \lambda_{22} = -(\lambda_{25} + \lambda_{27}); \quad \lambda_{33} = -(\lambda_{31} + \lambda_{32} + \lambda_{34} + \lambda_{37}); \quad \lambda_{44} = -(\lambda_{42} + \lambda_{47});$$

$$\lambda_{55} = -\lambda_{56}; \quad \lambda_{66} = -\lambda_{64}; \quad \lambda_{77} = -(\lambda_{71} + \lambda_{73} + \lambda_{74}).$$

Розписуючи поелементно матричне рівняння (5.3), отримаємо систему лінійних диференціальних рівнянь

$$\left\{ \begin{array}{l} \frac{dp_1}{dt} = -(\lambda_{14} + \lambda_{15})p_1 + \lambda_{31}p_3 + \lambda_{71}p_7, \\ \frac{dp_2}{dt} = -(\lambda_{25} + \lambda_{27})p_2 + \lambda_{32}p_3 + \lambda_{42}p_4, \\ \frac{dp_3}{dt} = -(\lambda_{31} + \lambda_{32} + \lambda_{34} + \lambda_{37})p_3 + \lambda_{73}p_7, \\ \frac{dp_4}{dt} = \lambda_{14}p_1 + \lambda_{34}p_3 - (\lambda_{42} + \lambda_{47})p_4, \\ \frac{dp_5}{dt} = \lambda_{15}p_1 + \lambda_{25}p_2 - \lambda_{56}p_5, \\ \frac{dp_6}{dt} = \lambda_{56}p_5 - \lambda_{64}p_6, \\ \frac{dp_7}{dt} = \lambda_{27}p_2 + \lambda_{37}p_3 + \lambda_{47}p_4 - (\lambda_{71} + \lambda_{73} + \lambda_{74})p_7. \end{array} \right. \quad (5.4)$$

Приклад ілюструє принцип складання системи диференціальних рівнянь Колмогорова для будь-якого випадкового процесу, обминаючи запис матричного рівняння: зі знаком „мінус” у i -тому рівнянні записані доданки, які відбивають властивості виходу зі стану S_i , зі знаком „плюс” – доданки, що відбивають властивості входу до стану S_i .

Загальний розв’язок такої системи, тобто вирази для функцій $p_i(t)$, знаходиться заміною одного з рівнянь рівнянням, яке відбиває основну властивість компонент вектора станів системи, а саме для даного випадку,

$$\sum_{i=1}^7 p_i = 1. \quad (5.5)$$

Один з методів розв’язання сформульованого прикладу показаний у роботі [10].

Зазвичай у таких задачах предметом дослідження є поведінка системи в усталеному, або стаціонарному режимі, тобто через деякий час, коли ймовірнісні характеристики станів

системи вже не залежать від часу, а складають нерухомий вектор фінальних ймовірностей, який знаходиться з рівняння (5.2).

Здійснимо обчислення цього вектора для матриці-генератора, сформованого за результатами спостереження за обігом вагонів на станції за певний місяць:

$$\Lambda = \begin{pmatrix} -2 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & -2 \end{pmatrix}.$$

У заданому вигляді марковський ланцюг не є регулярним (у системі виключений перехід до стану S_2), і тому для виконання обчислень доцільним буде розглядання системи без стану S_2 (навантаження вагонів на станції тимчасово припиняється), тобто заздалегідь вважаємо $P_2^* = 0$. Отже, згідно з (5.2),

$$\left\{ \begin{array}{l} (p_1^*, p_3^*, p_4^*, p_5^*, p_6^*, p_7^*) \cdot \begin{pmatrix} -2 & 0 & 1 & 1 & 0 & 0 \\ 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 2 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 2 & 0 & 0 & 0 & -2 \end{pmatrix} = 0, \\ p_1^* + p_3^* + p_4^* + p_5^* + p_6^* + p_7^* = 1 \end{array} \right.$$

Перетворюємо матричне рівняння у систему алгебраїчних рівнянь

$$\left\{ \begin{array}{l} -2p_1^* + 2p_3^* = 0 \\ -2p_3^* + 2p_7^* = 0 \\ p_1^* - 2p_4^* + p_6^* = 0 \\ p_1^* - p_5^* = 0 \\ p_5^* - p_6^* = 0 \\ 2p_4^* - 2p_7^* = 0 \\ p_1^* + p_3^* + p_4^* + p_5^* + p_6^* + p_7^* = 1 \end{array} \right.$$

Розв'яжемо систему методом Гауса, для чого випишемо поширену матрицю системи, переміщуючи останнє рівняння на перший рядок та виключаючи, наприклад, третє рівняння. Далі типовими діями прямого ходу методу Гауса з перетворення поширеної матриці системи приходимо до її трикутного вигляду.

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 \\ -2 & 2 & 0 & 0 & 0 & 0 & \cdot & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 & \cdot & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & \cdot & 0 \\ 0 & 0 & 2 & 0 & 0 & -2 & \cdot & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 \\ 0 & 4 & 2 & 2 & 2 & 2 & \cdot & 2 \\ 0 & -2 & 0 & 0 & 0 & 2 & \cdot & 0 \\ 0 & -1 & -1 & -2 & -1 & -1 & \cdot & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 & \cdot & 0 \\ 0 & 0 & 2 & 0 & 0 & -2 & \cdot & 0 \end{pmatrix} \sim \\
\sim \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 \\ 0 & -1 & -1 & -2 & -1 & -1 & \cdot & -1 \\ 0 & -2 & 0 & 0 & 0 & 2 & \cdot & 0 \\ 0 & 4 & 2 & 2 & 2 & 2 & \cdot & 2 \\ 0 & 0 & 0 & 1 & -1 & 0 & \cdot & 0 \\ 0 & 0 & 2 & 0 & 0 & -2 & \cdot & 0 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 \\ 0 & -1 & -1 & -2 & -1 & -1 & \cdot & -1 \\ 0 & 0 & 2 & 4 & 2 & 4 & \cdot & 2 \\ 0 & 0 & -2 & -6 & -2 & -2 & \cdot & -2 \\ 0 & 0 & 0 & 1 & -1 & 0 & \cdot & 0 \\ 0 & 0 & 2 & 0 & 0 & -2 & \cdot & 0 \end{pmatrix} \sim \\
\sim \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 \\ 0 & -1 & -1 & -2 & -1 & -1 & \cdot & -1 \\ 0 & 0 & 2 & 4 & 2 & 4 & \cdot & 2 \\ 0 & 0 & 0 & -2 & 0 & 2 & \cdot & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & \cdot & 0 \\ 0 & 0 & 0 & -4 & -2 & -6 & \cdot & -2 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 \\ 0 & -1 & -1 & -2 & -1 & -1 & \cdot & -1 \\ 0 & 0 & 2 & 4 & 2 & 4 & \cdot & 2 \\ 0 & 0 & 0 & 1 & -1 & 0 & \cdot & 0 \\ 0 & 0 & 0 & -2 & 0 & 2 & \cdot & 0 \\ 0 & 0 & 0 & -4 & -2 & -6 & \cdot & -2 \end{pmatrix} \sim \\
\sim \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 \\ 0 & -1 & -1 & -2 & -1 & -1 & \cdot & -1 \\ 0 & 0 & 2 & 4 & 2 & 4 & \cdot & 2 \\ 0 & 0 & 0 & 1 & -1 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 0 & -2 & 2 & \cdot & 0 \\ 0 & 0 & 0 & 0 & -6 & -6 & \cdot & -2 \end{pmatrix} \sim \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdot & 1 \\ 0 & -1 & -1 & -2 & -1 & -1 & \cdot & -1 \\ 0 & 0 & 2 & 4 & 2 & 4 & \cdot & 2 \\ 0 & 0 & 0 & 1 & -1 & 0 & \cdot & 0 \\ 0 & 0 & 0 & 0 & -2 & 2 & \cdot & 0 \\ 0 & 0 & 0 & 0 & 0 & -12 & \cdot & -2 \end{pmatrix} .$$

Отже, отримана відповідна система рівнянь відносно компонент вектора фінальних ймовірностей має такий вигляд:

$$\begin{cases} p_1^* + p_3^* + p_4^* + p_5^* + p_6^* + p_7^* = 1 \\ -p_3^* - p_4^* - 2p_5^* - p_6^* - p_7^* = -1 \\ 2p_4^* + 4p_5^* + 2p_6^* + 4p_7^* = 2 \\ p_5^* - p_6^* = 0 \\ -2p_6^* - 2p_7^* = 0 \\ -12p_7^* = -2 \end{cases} .$$

Звідки зворотним ходом методу Гауса знаходимо вектор фінальних ймовірностей

$$\bar{p}^* = \left(\frac{1}{6}; 0; \frac{1}{6}; \frac{1}{6}; \frac{1}{6}; \frac{1}{6}; \frac{1}{6} \right),$$

значення якого можна тлумачити так, що при заданих умовах в усталеному режимі можна вважати рівноможливими всі технологічні ланки технологічного процесу обігу вагонів, окрім їх навантаження.

5.2 МЕТОД ДИНАМІКИ СЕРЕДНІХ

Розглянутий приклад є типовим для демонстрування можливостей математичного апарату – теорії неперервних марковських ланцюгів. Але цей математичний апарат вважається зручним, поки кількість можливих станів системи S порівняно невелика. У випадку, коли кількість станів зростає (до порядку декількох десятків або сотень), для обробки результатів обчислень переходять до узагальнених характеристик процесів, а саме до *середніх значень*, які відомі як показники ефективності роботи СМО, наприклад, „*середня кількість заяв у системі*”, „*середній час перебування заяв у черзі*”, „*середня кількість зайнятих каналів обслуговування*” та інші. Всі ці характеристики знаходились через ймовірності станів.

Метод динаміки середніх дозволяє скласти і розв'язати рівняння безпосередньо для середніх характеристик, які і є кінцевою метою багатьох досліджень. Теоретичним підґрунтям використання методу є глобальні висновки закону великих чисел теорії ймовірностей: масовий характер явищ, що вивчаються, дозволяє встановити у них порівняно прості закономірності.

Розглянемо ідею методу динаміки середніх на прикладі.

Маємо систему S , яка складається з великої кількості N однорідних елементів, кожний з яких може випадково переходити зі стану в стан згідно з пуасонівським потоком подій. Це означає, що процес, який протікає у системі, буде марковським.

Розглянемо поведінку одного елемента. Припустимо, що елемент може знаходитись у будь-якому з N можливих станів: E_1, E_2, \dots, E_n , а стан системи S у кожний момент часу характеризується кількістю елементів, які знаходяться у кожному зі станів.

Вводимо у розгляд випадкову величину *чисельності станів* $X_k(t)$, яка відбиває кількість одиниць елементів, які знаходяться в момент t у стані E_k . Тоді для будь-якого моменту t сума чисельності всіх станів дорівнює загальній кількості елементів

$$X_1(t) + X_2(t) + \dots + X_n(t) = \sum_{k=1}^n X_k(t) = N. \quad (5.6)$$

Ставиться задача знаходження для будь-якого t $X_k(t)$, а саме її *математичного сподівання* $m_k(t)$ – середнього значення чисельності кожного стану та дисперсії $D_k(t)$ – розкиду фактичної чисельності навколо середнього значення.

Для того, щоб знайти ці характеристики, треба знати інтенсивності всіх потоків подій, які переводять *елемент* (а не систему – це важливо) зі стану в стан. Вважаючи відомими ці інтенсивності, а також графи станів кожного елемента, чисельність кожного стану $X_k(t)$ можна подати як суму випадкових величин, кожна з яких пов'язана з окремим (i -им) елементом так:

$$X_k^{(i)}(t) = \begin{cases} 1, \text{ якщо } i\text{-ий елемент у момент } t \text{ знаходиться у стані } E_k; \\ 0, \text{ якщо } i\text{-ий елемент у момент } t \text{ не знаходиться у стані } E_k. \end{cases} \quad (5.7)$$

Очевидно, що для будь-якого моменту t загальна чисельність стану E_k дорівнює сумі відповідних величин $X_k^{(i)}(t)$:

$$X_k(t) = X_k^{(1)}(t) + X_k^{(2)}(t) + \dots + X_k^{(N)}(t) = \sum_{i=1}^N X_k^{(i)}(t). \quad (5.8)$$

Якщо інтенсивності λ_{ij} потоків подій, які переводять кожний елемент зі стану в стан, є відомими, тобто не випадковими, випадкові величини $X_k^{(i)}(t)$ – доданки у виразі (5.4) для окремих елементів незалежні між собою. І цей висновок є важливим для використання теореми додавання математичних сподівань, а головне, теореми додавання дисперсій:

$$m_k(t) = \sum_{i=1}^N M(X_k^i(t)), \quad (5.9)$$

$$D_k(t) = \sum_{i=1}^N D(X_k^i(t)).$$

Числові характеристики випадкової величини $X_k^{(i)}(t)$ знайдемо, створивши її закон розподілу з урахуванням того, що будемо вважати значення ймовірності знаходження кожного з однорідних елементів у стані E_k рівною $p_k(t)$, а також (5.4):

0	1
$1 - p_k(t)$	p_k

Тоді згідно з визначеними формулами обчислення математичного сподівання і дисперсії

$$M(X_k^i(t)) = 0 \cdot (1 - p_k(t)) + 1 \cdot p_k(t) = p_k(t), \quad (5.10)$$

$$D(X_k(t)) = (0 - p_k(t))^2 \cdot (1 - p_k(t)) + (1 - p_k(t))^2 \cdot p_k(t) = p_k(t) \cdot (1 - p_k(t)).$$

Підставляючи вирази (5.7) у формули (5.6), отримаємо математичне сподівання, дисперсію та середнє квадратичне відхилення чисельності k -го стану:

$$m_k(t) = N \cdot p_k(t),$$

$$D_k(t) = N \cdot p_k(t) \cdot (1 - p_k(t)), \quad (5.11)$$

$$\sigma_k(t) = \sqrt{N p_k(t) (1 - p_k(t))}.$$

Формули (5.8) дозволяють вказати орієнтовно діапазон практично всіх можливих значень чисельності у будь-який момент часу

$$m_k(t) \pm 3\sigma_k(t).$$

Таким чином, не визначаючи ймовірностей станів системи S в цілому, а тільки ймовірності станів окремих її елементів, можна визначити для будь-якого моменту часу t середню чисельність кожного стану та у яких межах знаходиться фактична чисельність. Тобто сформульована задача зводиться до визначення ймовірностей станів *одного окремого елемента*. Ці ймовірності можуть бути знайдені як розв'язки диференціальних рівнянь Колмогорова, для складання яких повинні бути відомими інтенсивності потоків подій, які переводять кожний елемент зі стану в стан.

Зауважимо, що диференціальні рівняння станів можна записувати *безпосередньо для середніх чисельності станів*. Дійсно, з формул (5.11) виходить, що середня чисельність кожного стану пропорційна ймовірності цього стану (з коефіцієнтом пропорційності N), і тоді, склавши диференціальні рівняння відносно ймовірностей, множенням на N переводимо їх в рівняння відносно середніх чисельності станів $m_k(t)$. Формули (5.11) дозволяють знайти дисперсії чисельності станів $D_k(t)$, а також їх середні квадратичні відхилення $\sigma_k(t)$.

Знання числових характеристик чисельності станів відкриває можливість оцінювати ймовірності станів системи в цілому, наприклад, знаходити ймовірність того, що чисельність якогось стану буде знаходитись у певних межах. У припущенні, що кількість елементів N досить велика, чисельність будь-

якого k -того стану можна вважати розподіленою за нормальним законом. Тоді ймовірність того, що випадкова величина X_k (чисельність k -того стану) буде обмежена значеннями від α до β , відбивається формулою

$$P(X_k \in (\alpha, \beta)) = \Phi\left(\frac{\beta - m_k}{\sigma_k}\right) - \Phi\left(\frac{\alpha - m_k}{\sigma_k}\right), \quad (5.12)$$

де m_k , σ_k – математичне сподівання та середнє квадратичне відхилення чисельності k -того стану, $\Phi(x)$ – функція Лапласа.

6 ЕЛЕМЕНТИ КЛАСИЧНОЇ ЛОГІКИ

Завершальний розділ викладання дисципліни виник як вимога ознайомити студентів з основами *математичної, або формальної, логіки*, засвоєння мови якої сприяє набуттю навичок правильних міркувань і переконливої аргументації, чіткого формулювання думок і висновків, формування загальної культури мислення. Очевидно, що перелічені якості повинні бути притаманні будь-якому фахівцю та управлінцю. Тобто, з одного боку, цей розділ можна розглядати як інструмент, який використовується при постановці та коректному розв'язанні розглянутих у наданому курсі задач. А з другого боку, його можна розглядати як окрему наукову дисципліну, яка закладає фундамент для вивчення *узагальненої класичної логіки*, а саме *теорії нечітких множин та нечіткої логіки*, причиною появи якої стала наявність нечітких наближених міркувань при описі процесів, систем, об'єктів.

Класичні розділи математичної логіки складаються з: *алгебри висловлювань; числення висловлювань; логіки предикатів*. Всі ці поняття у достатньому обсязі вводяться, обговорюються та наповнюються прикладами у конспекті лекцій “Дискретна

математика. Висловлювання” [11], написаному викладачами кафедри.

СПИСОК ЛІТЕРАТУРИ

1 Математичне програмування: Методичні вказівки і завдання з теми для студентів економічних спеціальностей заочної скороченої форми навчання. / І.В. Ковалішина. – Харків: ХарДАЗТ, 1999. – Ч. 1. – 48 с. [№3534].

2 Математичне програмування: Завдання і методичні вказівки до виконання контрольної роботи. / О.О. Думіна, О.І. Удодова. – Харків: УкрДАЗТ, 2007. – Ч. 1. – 54 с. [№1358].

3 Математичне програмування: Методичні вказівки і завдання з теми для студентів економічних спеціальностей заочної скороченої форми навчання. / І.В. Ковалішина. – Харків: ХарДАЗТ, 1999. – Ч. 2. – 46 с. [№3535].

4 Математичне програмування: Завдання і методичні вказівки до виконання контрольної роботи. / О.О. Думіна, М.Є. Резуненко, О.І. Удодова, Ю.С. Шувалова. – Харків: УкрДАЗТ, 2009. – Ч. 2. – 54 с. [№1441].

5 Дослідження операцій: Методичні вказівки для студентів економічних спеціальностей денної та безвідривної форми навчання / Н.С. Юрчак, Н.І. Волохова, О.О. Думіна. – Харків: УкрДАЗТ, 2005. – 58 с. [№ 1298].

6 Акулиничев В.М. Математические методы в эксплуатации железных дорог: учеб. пособие для вузов ж.-д. трансп. / В.М. Акулиничев, В.А. Кудрявцев, А.Н. Корешков – М.: Транспорт, 1981. – 223 с.

7 Таха Хэмди А. Введение в исследование операций / Пер. с англ. – М.: Издательский дом "Вильямс", 2001. – 912 с.

8 Єфременко Р.О. Елементи теорії марковських ланцюгів: Конспект лекцій / Р.О. Єфременко, Г.Ю. Глушакова, М.Є. Резуненко. – Харків: УкрДАЗТ, 2004. – 42 с.

9 Елементи теорії масового обслуговування: Конспект лекцій / Р.О. Єфременко, Г.Ю. Глушакова, М.Є. Резуненко. – Харків: УкрДАЗТ, 2004. – 42 с.

10 Розподіл імовірності станів системи обороту вагонів на залізничному вузлі. Загальне рішення / Д.В. Ломотько, С.Д. Бронза, М.Ж. Овчів // Інформаційно-керуючі системи на залізничному транспорті. – 2012. – № 6. – Ч. 1. – С. 15-18.

11 Дискретна математика. Висловлювання обслуговування: Конспект лекцій / О.О. Думіна, Є.Ю. Колбасіна, О.І. Удодова, Ю.С. Шувалова. – Харків: УкрДАЗТ, 2012. – Ч. 1. – 26 с.

