МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

ХАРКІВСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ МІСЬКОГО ГОСПОДАРСТВА

Я. В. Санько

КОНСПЕКТ ЛЕКЦІЙ

З ДИСЦИПЛІНИ

ДОСЛІДЖЕННЯ ОПЕРАЦІЙ В ТРАНСПОРТНИХ СИСТЕМАХ

(*для студентів 3 курсу денної та заочної форм навчання за напрямом підготовки 6.070101 «Транспортні технології (за видами транспорту)»*)

ХАРКІВ – ХНАМГ – 2012

Санько Я. В. Конспект лекцій з дисципліни «Дослідження операцій в транспортних системах» (для студентів 3 курсу денної та заочної форм навчання за напрямом підготовки 6.070101 «Транспортні технології (за видами транспорту)») / Я. В. Санько; Харк. нац. акад. міськ. госп-ва. – Х.: ХНАМГ, 2012. – 74 с.

Автор: Я. В. Санько

Рецензент: доц. О. В. Прасоленко

Рекомендовано кафедрою транспортних систем і логістики,
протокол № 2 від 30.08.2011 р.

ЗМІСТ

Вступ ……..... 4

Тема 1. Лінійне програмування ……..... 6
1.1. Задача розподілу ресурсів ……..... 6
1.2. Динамічне планування ……..... 8
1.3. Задача вибору оптимального транспортного маршруту …..... 10
1.4. Алгебраїчне формулювання задачі лінійного програмування у загальному вигляді …..... 12
1.5. Геометрична інтерпретація ……..... 13
1.6. Симплексний алгоритм ……..... 16
1.7. Класична транспортна задача ……..... 19

Тема 2. Цілочисельне програмування ……..... 21
2.1. Постановки задач цілочисельного програмування ……..... 23
2.2. Загальні відомості про методи рішення задач цілочисельного програмування ……..... 27
2.3. Метод гілок і границь ……..... 30
2.4. Задача комівояжера ……..... 32
2.5. Метод часткового (неявного) перебору ……..... 37

Тема 3. Динамічне програмування ……..... 40
3.1. Аналіз динамічних процесів ……..... 40
3.2. Модель розподілу зусиль ……..... 50
3.3. Модель заміни обладнання ……..... 53

Тема 4. Теорія масового обслуговування ……..... 56
4.1. Функції та узагальнена структура систем масового обслуговування ... 56
4.2. Класифікація систем масового обслуговування ……..... 58
4.3. Характеристики та критерії ефективності систем масового обслуговування …….. 60

Тема 5. Сітьове планування і управління комплексами робіт ……..... 62
5.1. Поняття та терміни ……..... 62
5.2. Порядок і правила побудови графів ……..... 63
5.3. Побудова правильної нумерації вершин графа ……..... 64
5.4. Часові параметри сітьового графіка ……..... 65
5.5. Упорядкування графа, обчислення основних параметрів подій та робіт ……. 66
5.6. Діаграма Гантта ……..... 71

Список джерел ……..... 73

ВСТУП

Під операцією можна розуміти сукупність дій, заходів, спрямованих на досягнення певної мети, тобто сукупність цілеспрямованих дій. Таке визначення операцій є надзвичайно широким та охоплює значну частину діяльності людей. Основним завданням дослідження операцій є пошук шляхів досягнення мети.

У кожній окремій сфері діяльності, у кожній конкретній операції людство, використовуючи досвід та інтуїцію, із давніх часів створювало мистецтво вироблення найкращих рішень. Найбільш яскравими прикладами є економіка, військові дії, спорт.

Проте, наука про прийняття рішень, особливо математична теорія, почала створюватися порівняно недавно і наразі розвивається досить швидко, хоча є далекою від довершеності.

Очевидно, що результати тільки кількісного аналізу ніколи не можуть стати повною підставою для прийняття того чи іншого стратегічного рішення.

Побудова моделей є квінтесенцією операційного підходу до вирішення організаційних завдань. У дослідженні операцій моделювання відіграє роль, аналогічну лабораторному експериментові у природничих науках.

Побудова моделі допомагає звести складні і часом невизначені фактори, пов'язані з проблемою ухвалення рішення, у логічно струнку схему, доступну для детального аналізу. Така модель дозволяє виявити альтернативи рішення задач та оцінити результати, до яких вони призводять, а також дає можливість визначити, які дані необхідні для оцінки наявних альтернатив. У підсумку це забезпечує одержання обґрунтованих висновків. Коротше кажучи, модель є засобом формування чіткого подання дійсності.

У дослідженні операцій модель здебільшого належить до класу математичних та обов'язково є певним наближеним відображенням дійсності. Вона повинна будуватися таким чином, щоб відбивати сутність проблеми організаційного керування. У той же час модель має бути довільною від несуттєвих деталей, що дозволяє відшукувати ефективніше рішення, яке можна реалізувати на практиці. Визначення правильного балансу між ступенем адекватності, моделі тієї дійсності, яку вона описує, і можливістю одержання з моделі реалізованого рішення в більшості випадків становить складну задачу, тому побудова моделей може виявитися справою дуже нелегкою.

У дослідженні операцій можна виділити чотири основні напрямки:

1. Створення й опис способів дії, що можуть призвести до досягнення мети; серед них і необхідно проводити вибір «найкращих» способів.

2. Створення моделі операції, що дає математичний опис мети, процесу та результатів проведення операції.

3. Оцінка та порівняння ефективності способів, що конкурують, на основі створеної моделі.

4. Розробка розуміння оптимального вибору дій і математичних методів їхнього пошуку.

Перший напрямок є сферою конкретних досліджень, що враховують

специфіку конкретної операції та спираються на відповідні розділи науки; математикам тут майже нема чого робити. Розробка моделі майже завжди пов'язана з боротьбою двох по суті протилежних бажань: якомога точніше відобразити в моделі реальні процеси й одержати модель досить просту, щоб можна було сподіватися вирішити задачу до кінця й досягти очікуваних результатів. Саме через це необхідна співдружність „фахівців ” і „математиків ”.

Другим розділом математики, на якому базується дослідження операцій, безсумнівно, є теорія ймовірностей, і особливо, математична теорія надійності та теорія масового обслуговування, теорія статистичних рішень, у якій статистика поєднана з теорією гри.

Оскільки суттєвою частиною дослідження операцій є пошук оптимальних рішень, в орбіту цієї науки, природно, входять і методи такого пошуку, як лінійне та нелінійне програмування.

У запропонованому конспекті наголос зроблено не стільки на суворому викладенні численних конкретних результатів і напрямків (це роль навчальних посібників), скільки на демонстрації загальних підходів і формулювань задач.

Підготовка майбутнього спеціаліста до вирішення проблем транспортних систем із урахуванням механізмів функціонування ринкової економіки, широкого впровадження дослідження операцій до всіх сфер діяльності транспортних технологій здійснюється за допомогою навчальної дисципліни „Дослідження операцій ”.

Об'єктом вивчення цієї дисципліни є складні транспортні системи.

Предметом вивчення дисципліни є транспортні процеси, що відбуваються в різних видах діяльності фахівців транспортних технологій, оцінка стану та динаміки їхнього розвитку.

Зміст дисципліни „Дослідження операцій ” полягає в розкритті теоретичних основ проектування й експлуатації великих і складних транспортних систем, управління транспортними процесами, методів аналізу стану, оцінки їхніх характеристик та ефективності.

ТЕМА 1. ЛІНІЙНЕ ПРОГРАМУВАННЯ

Лінійне програмування, безумовно, належить до найбільш розповсюджених методів дослідження операцій, що використовуються під час вирішення виробничих і комерційних завдань.

Попередні зауваження. Ціль даного розділу полягає в тому, щоб познайомити читача з низкою задач оптимізації, рішення яких припускає побудову лінійних моделей. Щоб здійснити це, не відволікаючи від головного, ми не будемо торкатися тут методів чисельного рішення задач лінійного програмування.

Важливий клас моделей, використовуваних при рішенні так званих транспортних задач, ілюструється на прикладі, приведеному в розділі 1.2. Багато складних моделей лінійного програмування являють собою комбінацію транспортної задачі і задачі „вибору компонентів ” (для складання рідких сумішей, для виготовлення харчових продуктів і т.д.). Крім того, вони можуть бути „мультичасовими ”, тобто містити елементи динаміки.

1.1. Задача розподілу ресурсів

На підприємстві, що випускає неоднорідну продукцію, керівник прагне визначити, якими повинні бути рівні виробництва для кожного продукту протягом якогось наперед заданого періоду. Ці рівні обмежені технологічними й іншими («внутрішніми» для даного підприємства) умовами, заданими у виді лінійних співвідношень (рівностей або нерівностей). У рамках цих обмежень керівництво даного підприємства намагається оптимізувати деяку конкретну цільову функцію, У розглянутому тут прикладі метою є одержання максимального прибутку.

Уявимо собі фірму (назвемо її умовно „Мультиконвеєр ”), що має можливість реалізувати від одного до чотирьох різних типів виробничо-технологічних процесів, при цьому в неї є право вибору того або іншого варіанта. Технологічні процеси першого і другого типів орієнтовані на одержання продукції А, а технологічні процеси третього і четвертого типів – на одержання продукції В. Витрати, пов’язані з кожним з технологічних процесів, визначаються трудовитратами (вимірюваними в людино-тижнях), кількістю (в одиницях ваги) споживаного протягом тижня матеріалу Y и кількістю (у шухлядах) споживаного протягом тижня матеріалу Z. Оскільки витрати, пов’язані з різними технологічними процесами, не однакові, прибутковість процесів виявляється різною навіть у тому випадку, коли вони використовуються для одержання продукції того самого виду. При складанні виробничого плану на тиждень діапазон можливостей підприємства обмежений як за рахунок людських ресурсів, так і за рахунок споживаної сировини (тобто матеріалів Y і Z). Виробничо-економічні показники і всі наявні обмеження подані у табл. 1.1.

Таблиця 1.1 – Виробничо-економічні показники роботи підприємства

	На одиницю продукції А		**На одиницю продукції В**		**Наявні ресурси (разом)**
	технологічний процес 1	**технологічний процес 2**	**технологічний процес 3**	**технологічний процес 4**	
Кількість людино-тижнів	1	1	1	1	≤25
Кількість матеріалу Y (у кілограмах)	4	5	3	2	≤120
Кількість матеріалу Z (одиниця виміру – ящик)	3	5	13	16	≤200
Дохід із одиниці продукції (у гривнях)	14	5	9	21	Максимізувати
Обсяг продукції, що випускається	x_1	x_2	x_3	x_4	

Розглянемо, насамперед, які допущення щодо технології виробництва необхідно прийняти, щоб забезпечити лінійність моделі.

Аксіоми лінійності. Приймемо наступні два допущення, що відіграють винятково важливу роль при аналізі (як у технологічному, так і в економічному аспектах) згаданих вище виробничо-технологічних процесів:

1. Подільність. Для кожного виробничо-технологічного процесу сумарна кількість кожного зі споживаних ресурсів і відповідний прибуток строго пропорційні обсягові продукції, що випускається, (тобто в розрахунку на одиницю часу пропорційні відповідної виробничої потужності або, як кажуть, рівню виробничої активності). Іншими словами, усі показники виробничо-технологічного процесу можуть бути збільшені або зменшені при збереженні їхньої взаємної пропорційності.

2. Адитивність. Якщо значення кожної з керованих змінних x_j визначено (тобто зазначений відповідний рівень виробничої активності), то повна кількість кожного зі спожитих ресурсів дорівнює сумі однойменних ресурсів, витрачених при реалізації всіх технологічних процесів, що застосовувалися, а повний прибуток дорівнює сумі прибутків, одержуваних в результаті реалізації цих технологічних процесів.

У розглянутому прикладі мається три лінійних нерівності (обмеження на трудовитрати, обмеження на матеріал Y і обмеження на матеріал Z). Прибуток, що підлягає максимізації, задається лінійним співвідношенням. Більш конкретно задача зводиться до наступного:

$$\text{максимізувати прибуток} = \text{максимізувати } (4x_1 + 5x_2 + 9x_3 + 11x_4) \qquad (1.1)$$

при обмеженнях

$$1x_1 + 1x_2 + 1x_3 + 1x_4 \le 15 \text{ (людино-тижні)},$$
$$7x_1 + 5x_2 + 3x_3 + 2x_4 \le 120 \text{ (матеріал Y)}, \qquad (1.2)$$
$$3x_1 + 5x_2 + 10x_3 + 15x_4 \le 100 \quad \text{(матеріал Z)}.$$

Негативні значення рівнів виробництва (обсягів продукції, що випускається), як, наприклад, x_1 = - 4,2, не мають фізичного змісту; таким чином, будемо вимагати, щоб виробництво не було „негативним ”. Іншими словами, будемо вважати, що кожна з керованих змінних x_j (j = 1, 2, 3, 4) або дорівнює нулеві, або приймає позитивні значення, тобто ненегативна:

$$x_1 \ge 0, x_2 \ge 0, x_3 \ge 0, x_4 \ge 0, \qquad (1.3)$$

Задача організаційного управління полягає в тому, щоб знайти значення всіх невідомих x_j, що задовольняє співвідношення (1.2) і (1.3) і максимізують прибуток (1.1). Взагалі говорячи, такі значення не обов’язково є єдино можливими. Можуть існувати альтернативні оптимальні рішення.

Одночасно з бажанням визначити оптимальні значення для кожної невідомої x_j може виникнути намір з’ясувати, яким чином відіб’ється на одержуваному прибутку збільшення кожного зі споживаних ресурсів, удосконалювання того або іншого технологічного процесу, зміна вартості споживаної сировини (і, отже, зміна прибутковості виробничо-технологічних процесів) або використання в процесі виробництва якого-небудь іншого ресурсу, що не є дефіцитним. При рішенні багатьох практичних задач методом лінійного програмування подібні питання стають важливішими, ніж визначення оптимальних значень кожної з змінних x_j.

1.2. Динамічне планування

Кожну з розглянутих вище моделей можна узагальнити на той випадок, коли задача планування носить мультичасовий характер. Так, наприклад, якщо в задачі розподілу ресурсів сировинні і людські ресурси, а також прибуток з одиниці продукції змінюються у часі, то задача оптимізації здобуває динамічний характер. При цьому вона не зводиться цілком до задачі оптимізації для послідовних періодів часу, розглянутих ізольовано одне від одного.

Загальним для всіх моделей цієї категорії є те, що поточні керуючі рішення „виявляються ” як у період, що відноситься безпосередньо до моменту ухвалення рішення, так і в наступні періоди. Отже, найбільш важливі економічні наслідки виявляються у різні періоди, а не тільки протягом одного періоду.

Багато важливих моделей лінійного програмування мають так звану мережну структуру, що призводить до дуже простого табличного подання.

Приклад. Велика молочна фірма має m заводів, що знаходяться в різних районах однієї області. Щодня виробництво молочної продукції на заводі і (i = 1, 2, ..., m) не перевищує S_i літрів. Щоб задовольнити наявний попит, фірма повинна щодня поставляти на кожний з n пунктів збуту не менш D_j (j = 1, 2, ..., n) літрів свіжої продукції. Економічна задача полягає в тому, щоб визначити, які

зливальні пункти якими заводами варто забезпечити, щоб транспортні витрати були мінімальними.

Нехай x_{ij} – кількість літрів молока, що постачається на j-й зливальний пункт і-м заводом, а c_{ij} – відповідні транспортні витрати у розрахунку на один літр.

Завод	**Пункт збуту**					
	1	**2**	**3**	...	**n**	**Пропозиція**
1	c_{11} x_{11}	c_{12} x_{12}	c_{13} x_{13}	...	c_{1n} x_{1n}	S_1
2	c_{21} x_{21}	c_{22} x_{22}	c_{23} x_{23}	...	c_{2n} x_{2n}	S_2
3	c_{31} x_{31}	c_{23} x_{23}	c_{33} x_{33}	...	c_{3n} x_{3n}	S_3
⋮	⋮	⋮	⋮	⋱	⋮	⋮
m	C_{m1} x_{m1}	c_{m2} x_{11}	C_{m3} x_{m3}	...	c_{mn} x_{mn}	S_m
Попит	D_1	D_2	D_3	...	D_n	

Рис. 1.1. Транспортний розклад

Математично задача формулюється в такий спосіб:

мінімізувати $$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \qquad (1.4)$$

при обмеженнях

$$\sum_{j=1}^{n} x_{ij} \le S_i \ \ (i = 1, 2, \ldots, m) \text{ (для пропозиції)}, \qquad (1.5)$$

$$\sum_{i=1}^{m} x_{ij} \le D_j \ \ (j = 1, 2, \ldots, n) \ \text{ (для попиту)}, \qquad (1.6)$$

$$x_{ij} \ge 0 \ \ (i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n). \qquad (1.7)$$

Зручне для використання табличне подання розглянутої ситуації дане на рис. 1.1. Якщо $\sum_{i=1}^{m} S_i \ge \sum_{j=1}^{n} D_j$, так що повний попит, принаймні, не перевищує сумарної пропозиції (тобто повного обсягу фірмою продукції, що випускається), то завжди можна знайти припустимий графік перевезень, коли використовується не більш $m + n - 1$ маршрутів. Як показано у розділі 2, при сформульованих вище умовах існує таке оптимальне рішення, що використовується не більше $m + n - 1$ маршрутів.

Для побудови одного з можливих графіків такого роду з числом маршрутів, що не перевищують $m + n - 1$, варто почати з верхнього лівого (або, як іноді кажуть, з північно-західного) кута таблиці і розподіляти продукцію в обсязі S_1 по пунктах збуту (починаючи з пункту, споживаного D_1 літрів), поки S_1 не буде цілком вичерпано. Потім аналогічні дії проводяться з S_2 і т.д. Таблиця на рис. 1.2 є ілюстрацією цієї процедури. Немає необхідності пояснювати, що одержуваний при цьому пробний маршрутний графік може

виявитися (і, як правило, виявляється далеко не оптимальним).

Завод	**Пункт збуту**				
	1	**2**	**3**	**4**	**Пропозиція**
1	c_{11} 2	c_{12} 8	c_{13}	c_{14}	10
2	c_{21}	c_{22} 1	c_{23} 3	c_{24} 2	6
3	c_{31}	c_{23}	c_{33}	c_{34} 15	15
Попит	2	9	3	17_n	

Рис. 1.2. Попереднє допустиме рішення

Характерною рисою мережної моделі, обумовленої співвідношеннями (1.4) – (1.7), є те, що у випадку, коли має місце хоча б одне припустиме рішення, завжди існує оптимальне рішення, для якого всі x_{ij} приймають цілочисельні значення (за умови, якщо S_i і D_j приймають цілі значення).

1.3. Задача вибору оптимального транспортного маршруту

Уявимо собі фірму, що займається перевезенням вантажів. Припустимо, що даній фірмі необхідно взяти в оренду на чотири роки визначену кількість транспортних засобів. Фірма може задовольнити свої потреби, узявши потрібну кількість транспортних засобів в оренду рівно на чотири роки (з початку першого року до початку п’ятого року). Відповідні витрати C_{15} включають орендну плату, оплату за пробіг і експлуатаційні витрати, пов’язані з використанням транспортних засобів протягом чотирьох років. Можливий інший варіант: фірма може орендувати додаткову кількість транспортних засобів на термін з початку першого до початку третього року, а потім узяти під оренду інша кількість транспортних засобів на термін з початку третього до початку п'ятого року. В другому варіанті витрати, що включають ті ж компоненти, що й у першому випадку, дорівнюють $C_{13} + C_{35}$. У більшості випадків $C_{15} \neq C_{13} + C_{35}$. У пошуках стратегії, що забезпечує мінімальні витрати, можна, природно, розглянути і ряд інших варіантів. У цьому змісті мережна модель, зображена на рис. 1.3, є ілюстрацією задачі динамічного планування.

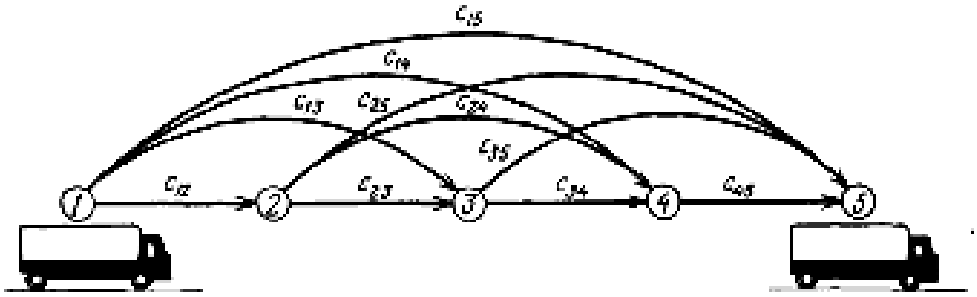

Рис. 1.3. Задача визначення найкоротшого маршруту

Питання, відповіді на які потрібно дати у зв’язку з розглядом мережі, поданої на рис. 1.3, пов’язані з перебуванням найкращого маршруту з вузла 1 у вузол 5. Ці

питання можна поставити в двох різних, хоча і взаємозалежних варіантах:

1) Чому дорівнюють витрати для найдешевшого маршруту від вузла 1 до вузла 5?

2) Який маршрут від вузла 1 до вузла 5 пов'язаний з найменшими витратами?

Інтуїція підказує, що, відповідаючи на одне питання, ми одночасно одержуємо відповідь і на інше питання. Однак, переходячи до побудови моделі лінійної оптимізації, ми будемо шукати відповідь на кожне питання окремо, а потім проаналізуємо взаємозв'язок між сформульованими вище задачами. Почнемо з першого питання.

Витрати, пов'язані з використанням найбільш дешевого маршруту. Нехай:

y_i – витрати, пов'язані з використанням найбільш дешевого маршруту від вузла i до вузла 5, де i приймає одне зі значень i =1, 2 ,3 ,4.

Можна визначити y_i, підрахувавши витрати для прямого маршруту від вузла i до вузла j (j ~ i) і додавши до них витрати для оптимального маршруту від вузла j до вузла 5. Найменші з усіх вироблених при цьому витрат визначають значення y_i. Математично це можна сформулювати в такий спосіб:

$$y_i = \underset{j=i+1,\ldots,5.}{\text{мінімум}} \left[c_{ij} + y_{ij}\right] (i = 1, 2, 3, 4),\ y_5 \equiv 0. \qquad (1.8)$$

Зокрема, для i = 1 маємо

$$y_1 = \text{мінімум}\left[c_{12} + y_2, c_{13} + y_3, c_{14} + y_4, c_{15}\right]. \qquad (1.9)$$

Таким чином, витрати у випадку вибору найбільш дешевого маршруту від вузла 1 до вузла 5 можна визначити, якщо відомі витрати, зв'язані з використанням найбільш дешевих маршрутів від вузла 2 до вузла 5, від вузла 3 до вузла 5 і від вузла 4 до вузла 5. Отже, у цьому випадку задача зводиться до порівняння витрат, пов'язаних з використанням прямого маршруту від вузла 1 до вузла 5, з витратами при використанні маршруту від вузла 1 через кожний із проміжних маршрутів і вибору найкращого шляху від вузла 1 до вузла 5.

З позицій лінійного програмування задача полягає в тому, щоб

максимізувати **1y** (1.10)

при обмеженнях

$$\begin{aligned} 1y_1 &\le c_{12} + 1y_2, \\ 1y_1 &\le c_{13} + 1y_3, \\ 1y_1 &\le c_{14} + 1y_4, \\ 1y_1 &\le c_{15}. \end{aligned} \qquad (1.11)$$

Маршрут, пов'язаний з найменшими витратами. Для рішення задачі вибору маршруту потрібно лише трохи узагальнити ті ідеї, що були основою при розгляді транспортної мережної моделі. Як і раніше, припустимо

x_{ij} = інтенсивність потоку від вузла i до вузла j (i = 1, 2, 3, 4), j > i.

Накладемо обмеження $x_{ij} \ge 0$. Для знаходження найкращого маршруту (шляху) розглянемо одиничний потік, що виходить з вузла 1 і входить в інші вузли:

$$1x_{12} + 1x_{13} + 1x_{14} + 1x_{15} = 1. \qquad (1.12)$$

Будемо вважати, що всі потоки, що входять у вузли 2, 3 і 4, повинні

випливати з них з наступним надходженням у вузол з більш високим порядковим номером:

$$\begin{aligned} -1x_{12} + 1x_{23} + 1x_{24} + 1x_{25} = 0 \text{ (вузол 2)}, \\ -1x_{13} - 1x_{23} + 1x_{34} + 1x_{35} = 0 \text{ (вузол 3)}, \\ -1x_{14} - 1x_{24} - 1x_{34} + 1x_{45} = 0 \text{ (вузол 4)}. \end{aligned} \quad (1.13)$$

Задача полягає в тому, щоб

мінімізувати $$\sum_{i=1}^{4}\sum_{j=i+1}^{5} c_{ij}x_{ij}. \quad (1.14)$$

1.4. Алгебраїчне формулювання задачі лінійного програмування у загальному вигляді

Математичні подання, сформульовані для ряду можна узагальнити в такий спосіб. Нехай x_j є j-а керована змінна (j = 1, 2, ..., n). Потрібно визначити такі значення x_j, щоб вираз $p_1x_1 + p_2x_2 + ... + p_nx_n$ залежно від змісту задачі було максимізовано або мінімізовано.

На x_j накладений ряд обмежень, кожне з яких відноситься до одному з наступних типів:

$$a_1x_1 + a_2x_2 + ... + a_nx_n \geq a,$$
$$b_1x_1 + b_2x_2 + ... + b_nx_n = b,$$
$$c_1x_1 + c_2x_2 + ... + c_nx_n \leq c.$$

Крім того, може мати місце обмеження $x_{ij} \geq 0$.

Задача оптимізації при такого виду обмеженнях:

1) може не мати жодного припустимого рішення, тобто може не існувати таких значень змінних x_j (j = 1, 2, ..., n), що задовольняли б всім обмеженням;

2) може мати єдине припустиме оптимальне рішення;

3) може мати кілька припустимих оптимальних рішень;

4) може мати таке припустиме рішення, для якого цільова функція виявляється необмеженою, тобто значення цільової функції може бути зроблене як завгодно великим для задачі максимізації або як бажано малим для задачі мінімізації за рахунок вибору відповідного припустимого рішення.

Будь-яку задачу лінійного програмування можна розглядати як задачу:

максимізації $$\sum_{j=1}^{n} c_jx_j \quad (1.15)$$

при наявності обмежень

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \qquad (i = 1, 2, ..., m), \quad (1.16)$$

$$x_{ij} \geq 0 \qquad (j = 1, 2, ..., n). \quad (1.17)$$

Одночасно будь-яку задачу лінійного програмування можна звести до:

мінімізації $$\sum_{j=1}^{n} c_j x_j \qquad (1.18)$$

за умови $$\sum_{j=1}^{n} a_{ij} x_j = b_i \qquad (i = 1, 2, \ldots, m),\ b_i \geq 0, \qquad (1.19)$$

$$x_{ij} \geq 0 \qquad (j = 1, 2, ..., n). \qquad (1.20)$$

Як правило (хоча зовсім не обов'язково), у (1.19) має місце нерівність $n > m$.

1.5. Геометрична інтерпретація

Кращого розуміння алгебраїчних властивостей лінійних оптимізаційних моделей можна добитися за допомогою геометричної інтерпретації понять, що використовуються в лінійному програмуванні.

Задача.

Максимізувати $$12x_1 + 15x_2, \qquad (1.21)$$

при наявності обмежень:

$$4x_1 + 3x_2 \leq 12, \qquad (1.22)$$

$$2x_1 + 5x_2 \leq 10, \qquad (1.23)$$

$$x_1 \geq 0,\ x_2 \geq 0. \qquad (1.24)$$

Ця задача графічно подана на рис. 1.4. Зважимо, що обмеження (1.22 – відображено цифрою 2 у колі) і (1.23 – відображено цифрою 3 у колі) зображені графічно (прямі лінії) як рівняння, отримані заміною у (1.22) у (1.23) нерівностей, що фігурують, на рівності. Відповідні нерівності зображені стрілками, спрямованими убік припустимих значень x_1 і x_2.

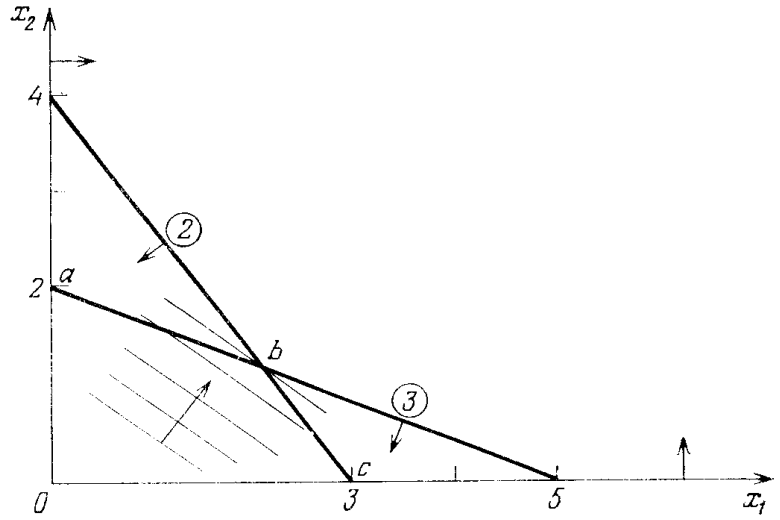

Рис. 1.4. Простір рішень

Оскільки жодна з цих змінних не може приймати негативних значень, область припустимих значень x_1 і x_2 обмежена, крім того, осями координат. Таким чином, багатокутник $\overline{Oabc}$ містить у собі область значень x_1 і x_2, що задовольняють усім наявним обмеженням. Множина точок, що належать області, обмеженої $\overline{Oabc}$ (разом із граничними точками), називають множиною

рішень. Ця множина є опуклою, тобто будь-який відрізок, що з'єднує дві довільним чином обрані точки даної множини, лежить усередині або проходить уздовж межі $\overline{Oabc}$ (іншими словами, належить згаданій множині). Вершини O, a, b і c називають екстремальними точками – вони не можуть належати внутрішній частині жодного з відрізків, що з'єднують дві різні точки розглянутої множини.

Рівнобіжні прямі на рис. 1.5 є графічним зображенням різних значень цільової функції. Стрілка, що перетинає ці прямі, спрямована убік зростаючих значень цільової функції. Оптимальне рішення визначається екстремальною точкою b, для якої

$$x_1 = 15/7,\ x_2 = 8/7,\ 12x_1 + 15x_2 = 300/7.$$

Альтернативні оптимальні рішення. Якщо у виразі для цільової функції змінити коефіцієнти (при цьому на рис. 1.4 кут нахилу рівнобіжних ліній стосовно осі абсцис також буде іншим), то в результаті точка, що задає оптимальне рішення, може, природно, переміститися. Однак завжди існує оптимальне рішення, обумовлене деякою екстремальною точкою. Продемонструємо справедливість цього твердження на одному конкретному прикладі. Нехай має місце деяке цілком визначена зміна кутових коефіцієнтів згаданих вище рівнобіжних прямих, тобто

максимізувати $4x_1 + 10x_2,$ (1.25)

при наявності тих же самих обмежень (1.22), (1.23) і (1.24).

Обмеження 1.22 – відображено цифрою 2 у колі, а 1.23 – цифрою 3 у колі, та зображені графічно (прямі лінії) на рис. 1.5.

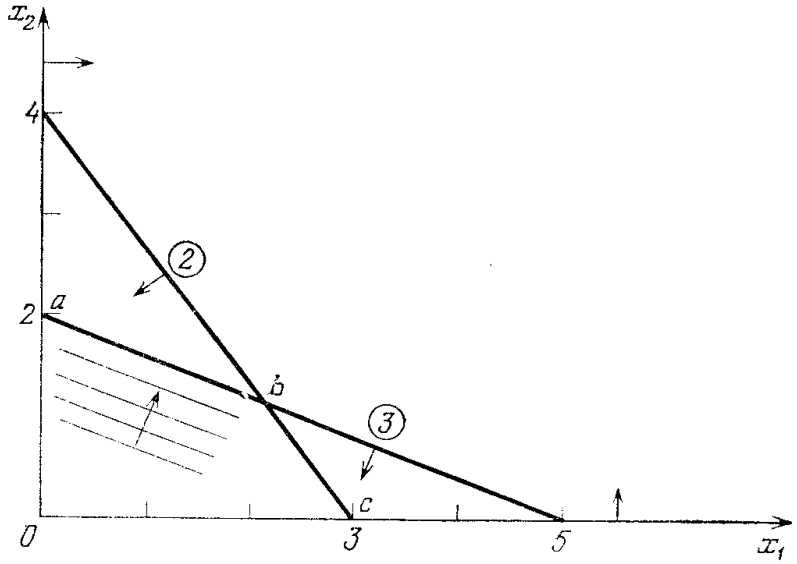

Рис. 1.5. Альтернативні оптимальні рішення

У цьому випадку всі точки (безкінечна множина точок), що лежать на відрізку $\overline{ab}$ (рис. 1.5), є оптимальними. Отже, рішення x_1= 15/7, x_2 = 8/7 як і раніше оптимальне. Але тепер оптимальним є і рішення x_1 = 0, x_2 = 2. Те ж саме

можна сказати і про будь-яке позитивно-зважене середнє двох зазначених рішень. При цьому оптимальне значення цільової функції дорівнює 20.

Необмежені оптимальні рішення. У прикладі, графічно поданому на рис. 1.6, розглядається наступна модель:

Максимізувати $-2x_1 + 6x_2$, (1.26)

при наявності обмежень

$$-1x_1 - 1x_2 \le 2, \quad (1.27)$$

$$-1x_1 - 1x_2 \le 1, \quad (1.28)$$

$$x_1 \ge 0,\ x_2 \ge 0. \quad (1.29)$$

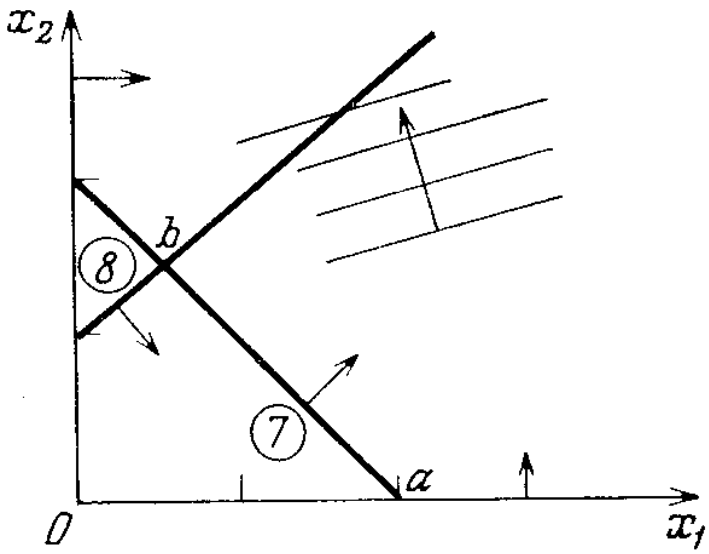

Рис. 1.6. Необмежене оптимальне рішення

Обмеження 1.27 – відображено цифрою 7 у колі, а 1.28 – цифрою 8 у колі, та зображені графічно (прямі лінії) на рис. 1.6.

Ця задача має необмежену множину рішень. Неважко переконатися, що вона опукла і що не існує екстремальних точок, крім a і b. Для даної задачі значення цільової функції може бути зроблене як завгодно великим, тобто для будь-якого заданого значення цільової функції у просторі рішень завжди існує точка, у якій цільова функція приймає ще більше значення. Така точка лежить на прямій, рівняння якої має вигляд $-1x_1 + 1x_2 = 1$.

Задача, що не має рішення. Розглянемо ще один приклад (графічно поданий на рис. 1.7). Нехай необхідно

максимізувати $1x_1 + 1x_2$, (1.30)

при наявності наступних обмежень:

$$-1x_1 - 1x_2 \le -1, \quad (1.31)$$

$$1x_1 - 1x_2 \le -1, \quad (1.32)$$

$$x_1 \ge 0,\ x_2 \ge 0. \quad (1.33)$$

Легко переконатися графічно (рис. 1.7), що дана задача не має жодного оптимального рішення.

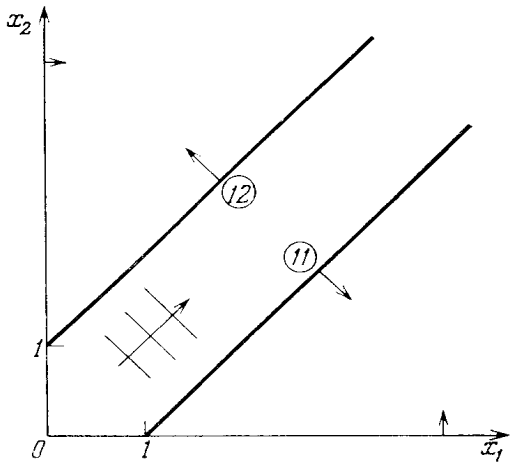

Рис. 1.7. Випадок, коли не існує оптимального рішення

1.6. Симплексний алгоритм

Для рішення задач лінійного програмування запропоновано чимало алгоритмів. Однак найбільш ефективним серед них виявляється алгоритм, розглянутий нижче. При цьому варто підкреслити, що при рішенні деяких часткових задач (як, наприклад, задач, пов'язаних з оптимізацією потоків у мережах) можуть виявитися більш ефективними інші алгоритми.

Спочатку припустимо, що рішення існує, причому оптимальне значення цільової функції кінцево (потім ці обмеження будуть зняті). У цьому випадку обчислювальна процедура буде виглядати у такий спосіб:

Крок 1. Виберемо m змінних, що задають припустиме спробне рішення. Виключимо ці змінні з виразу для цільової функції.

Крок 2. Перевіримо, чи не можна за рахунок однієї зі змінних, прирівняної спочатку до нуля, поліпшити значення цільової функції, додаючи їй відмінні від нуля (причому позитивні) значення. Якщо це можливо, перейдемо до кроку 3. У протилежному випадку припинимо обчислення.

Крок 3. Знайдемо граничне значення змінної, за рахунок якої можна поліпшити значення цільової функції. Збільшення значення цієї змінної дозволимо доти, поки одна з m змінних, що ввійшли у спробне рішення, не обернеться в нуль. Виключимо з виразу для цільової функції тільки що згадану змінну, за рахунок якої результат може бути поліпшений.

Крок 4. Вирішимо систему m рівнянь щодо змінних, що ввійшли в нове спробне рішення. Виключимо ці змінні з виразу для цільової функції. Повернемося до кроку 2.

Важливо відзначити, що при однозначному розумінні даного припису запропонований алгоритм дійсно приводить до оптимального рішення для будь-якої моделі лінійного програмування за кінцеве число інтеграцій. Такий спосіб рішення задач лінійного програмування часто називають симплексним алгоритмом.

Задача кроку 1 полягає в тому, щоб вибрати первинне припустиме рішення. Іншими словами, будується перше спробне рішення за допомогою тільки вільних змінних.

Аналіз моделей на чутливість і двоїста задача. Досвідчений керівник, що

використовує при рішенні задач організаційного керування методи лінійного програмування, рідко задовольняється лише чисельними значеннями керованих змінних, при яких досягається оптимум, якщо та або інша модель не застосовувалася їм багаторазово і, отже, діапазон її можливостей заздалегідь не відомий. У більшості ж випадків керівник хоче знати, у якому інтервалі можна змінювати вхідні параметри без істотного відхилення від знайденого оптимуму і без значного порушення структури базису. Дослідження, що дозволяє відповісти на ці питання, називається аналіз моделі на чутливість (або аналіз моделі при відомому оптимальному рішенні).

Вихідна і двоїста задачі. Розглянемо дві наступні задачі лінійного програмування:

максимізувати $$\sum_{j=1}^{n} c_j x_j \qquad (1.34)$$

при обмеженнях

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \qquad (i = 1, 2, \ldots, m), \qquad (1.35)$$

$$x_{ij} \geq 0 \qquad (j= 1, 2, ..., n). \qquad (1.36)$$

та

мінімізувати $$\sum_{j=1}^{m} b_j y_j \qquad (1.37)$$

при обмеженнях

$$\sum_{i=1}^{m} a_{ij} y_j \geq c_j \quad (j = 1, 2, \ldots, n), \qquad (1.38)$$

$$y_i \geq 0 \qquad (i = 1, 2, \ldots, m). \qquad (1.39)$$

Для визначеності умовно назвемо першу задачу [співвідношення (1.34) – (1.36)] вихідною, а другу [співвідношення (1.37) – (1.39)] двоїстою (стосовно першої).

У якості прикладу розглянемо наступні дві задачі:

вихідна задача:

максимізувати $$4x_1 + 5x_2 + 9x_3, \qquad (1.40)$$

при наступних обмеженнях:

$$\begin{aligned} &1x_1 + 1x_2 + 2x_3 \leq 16, \\ &7x_1 + 5x_2 + 3x_3 \leq 25, \\ &x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0; \end{aligned} \qquad (1.41)$$

двоїста задача:

мінімізувати $$16x_1 + 25x_2, \qquad (1.42)$$

при наступних обмеженнях:

$$\begin{aligned} &1y_1 + 7y_2 \geq 4, \\ &1y_1 + 5y_2 \geq 5, \\ &2y_1 + 3y_2 \geq 9, \end{aligned} \qquad (1.43)$$

$$y_1 \geq 0,\ y_2 \geq 0.$$

Інакше кажучи, двоїста задача – це на 90° обернена вихідна задача.

Дійсно,

1) j-й стовпець, складений з коефіцієнтів, що фігурують в обмеженнях вихідної моделі, збігається з j-й рядком, складеної з коефіцієнтів, що фігурують в обмеженнях двоїстої моделі;

2) рядок, складений з коефіцієнтів у виразі для цільової функції, збігається зі стовпцем, складеним з констант, що фігурують у правих частинах обмежень двоїстої системи;

3) стовпець, складений з констант, що фігурують у правих частинах обмежень вихідної моделі, збігається з рядком, складеної з коефіцієнтів у виразі для цільової функції двоїстої моделі;

4) напрямок знаків нерівності у вихідній моделі протилежно напрямку знаків нерівності в двоїстій моделі; вимога максимізації у вихідній задачі замінено вимогою мінімізації.

Має місце наступна важлива теорема (теорема подвійності):

а) Якщо вихідна і двоїста їй задачі мають припустиме рішення, то:

1) існує оптимальне рішення x_j^* (j = 1, 2, …, n) вихідної задачі;

2) існує оптимальне рішення y_j^* (i = 1, 2, …, m) двоїстої задачі;

3) має місце наступне співвідношення:

$$\sum_{j=1}^{n} c_j x_j^* = \sum_{i=1}^{m} c_i y_i^*. \qquad (1.44)$$

б) Якщо вихідна (двоїста) задача допускає оптимальне рішення, для якого значення цільової функції обмежено, то відповідна їй двоїста (вихідна) задача допускає оптимальне рішення при тому ж значенні цільової функції.

Оптимальні значення змінних двоїстої задачі.

а) Коефіцієнти при вільних змінних у рядку 0 на останній симплекс-ітерації при рішенні задачі максимізації збігаються з оптимальними значеннями змінних двоїстої задачі.

б) Коефіцієнт при x_j у рядку 0 на останній симплекс-ітерації являє собою різниця між лівою та правою частинами j-го обмеження двоїстої задачі, що відповідає оптимальному рішенню останньої.

Інтерпретація змінних двоїстої задачі. Оптимальне значення кожної змінної двоїстої задачі визначає позитивне або негативне збільшення значення цільової функції за рахунок одиничного збільшення (позитивного або негативного) значення константи в правій частині відповідного обмеження за умови, що розглянутий базис залишається допустимим.

Оптимальне значення x_0 = Σ (Константи в правих частинах обмежень) ×

× (Оптимальні значення змінних двоїстої задачі).

Оптимальні значення змінних двоїстої задачі часто називають схованими доходами. У випадку коли константи в правих частинах обмежень задають обсяги наявних ресурсів, сховані доходи визначають внесок у прибуток,

отриманий за рахунок одиниці кожного з ресурсів, відповідно до виду оптимального рішення вихідної задачі.

1.7. Класична транспортна задача

Транспортна задача (або задача прикріплення постачальників до споживачів) є одним з перших прикладів оптимізації на лінійних мережах. У даний час ця задача стала типовою для промислових фірм, що мають кілька підприємств, складів, ринків збуту та оптових баз. Модель застосовується головним чином при рішенні планових задач. У цьому випадку стратегічні рішення зводяться до вибору транспортних маршрутів, якими продукція різних підприємств доставляється на кілька складів або у різні кінцеві пункти призначення.

Математична постановка класичної транспортної задачі має вигляд

максимізувати $$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}. \qquad (1.45)$$

при обмеженнях

$$\sum_{j=1}^{n} x_{ij} \le S_i, \text{, } i = 1, 2, \ldots, m \text{ (наявні ресурси)}, \qquad (1.46)$$

$$\sum_{j=1}^{m} x_{ij} \ge D_j, \text{, } j = 1, 2, \ldots, n \text{ (попит)}, \qquad (1.47)$$

$$x_{ij} \ge 0, \text{ для всіх i та j} \qquad (1.48)$$

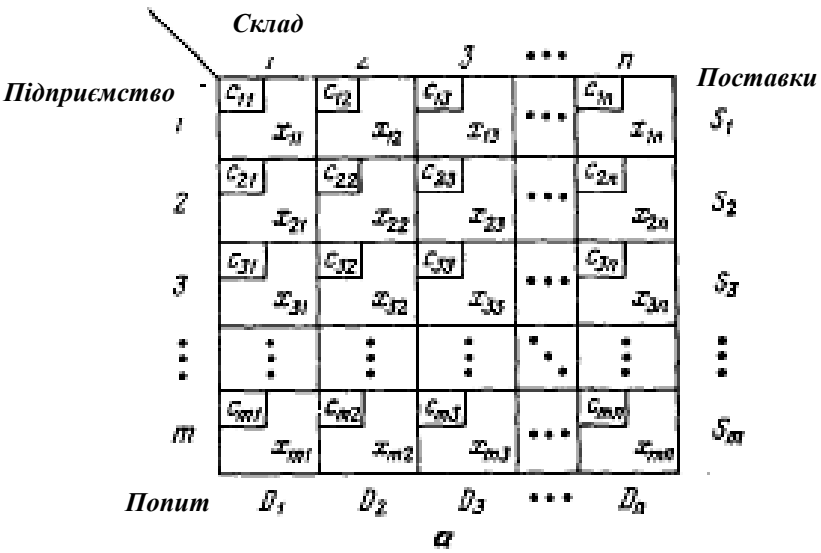

Рис. 1.8. Матриця умов транспортної задачі

У класичній інтерпретації цієї моделі прийнято вважати, що наявні m різних постачальників (підприємств або пунктів відправлення), що мають деякі вироби, які вони можуть відправити n споживачам (у n пунктів призначення). Зокрема, передбачається, що підприємство i може відвантажити не більш S_i виробів, а споживачеві j потрібно не менш D_j виробів. Величини S_i та D_j на

розглянутому інтервалі часу або плановому періоді приймаються постійними. Витрати на перевезення одиниці вантажу з пункту відправлення і у пункт призначення j дорівнюють C_{ij}. Цільовою функцією є вибір плану перевезень для заданого інтервалу часу, що мінімізує загальні транспортні витрати (рис. 1.9).

При аналізі звичайної транспортної задачі і побудові алгоритму її рішення зручно прийняти, що загальна потужність постачальників дорівнює загальному попитові споживачів, тобто

$$\sum_{i=1}^{m} S_i \geq \sum_{j=1}^{n} D_j. \qquad (1.49)$$

Таким чином, надалі передбачається, що виконується умова (1.49), а обмеження (1.46) і (1.47) є рівностями (якщо тільки спеціально не передбачається що-небудь інше). Звідси модель транспортної задачі приймає вид

мінімізувати $$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}. \qquad (1.50)$$

при обмеженнях

$$\sum_{j=1}^{n} x_{ij} = S_i, \text{, } i = 1, 2, \ldots, m \text{ (наявні ресурси)}, \qquad (2.7)$$

$$\sum_{j=1}^{m} x_{ij} = D_j, \text{, } j = 1, 2, \ldots, n \text{ (попит)}, \qquad (2.8)$$

$$x_{ij} = 0,1,2,\ldots \text{ для всіх i та j} \qquad (2.9)$$

де S_i та D_j – позитивні цілі числа, що задовольняють умові (1.49).

Запитання для самоконтролю:

1. В чому полягає задача організаційного управління?
2. Що таке мережна структура моделей?
3. В чому полягає динамічне планування?
4. Поняття про задачу оптимального транспортного маршруту?
5. Як визначаються витрати на маршрутах?
6. Якою є сутність задачі оптимізації?
7. Для чого використовують геометричну інтерпретацію алгебраїчних властивостей лінійних оптимізаційних моделей?
8. Які особливості симплексного алгоритму для рішення задач лінійного програмування?
9. Сформулюйте класичну транспорту задачу.

ТЕМА 2. ЦІЛОЧИСЕЛЬНЕ ПРОГРАМУВАННЯ

Розглянемо наступну модель:

Оптимізувати $$\sum_{j=1}^{n} c_j x_j \qquad (2.1)$$

при обмеженнях

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \quad i = 1,2,\ldots,m, \qquad (2.2)$$

$$x_j \ge 0, \quad j = 1,2,\ldots,n, \qquad (2.3)$$

$$x_j - \text{цілі}, \quad j = 1,2,\ldots,p\,(\le n). \qquad (2.4)$$

Задачі оптимізації такого класу називаються задачами цілочисельного (або діофантового, або дискретного) програмування. Якщо p = n, тобто всі змінні повинні бути цілими числами, то модель визначає цілком цілочисельну задачу. У протилежному випадку, тобто коли p<n, маємо частково цілочисельну задачу.

Задача цілочисельного програмування відрізняється від будь-якої задачі лінійного програмування умовами дискретності (2.4). У випадку коли лінійні обмеження (2.2) відповідають деякої мережної моделі, існує оптимальне рішення задачі (2.1), (2.2) і (2.3), що задовольняє обмеженням цілочисельності (2.4). Цей результат уже відомий з теореми про безперервність для потоків у мережах. Однак у загальному випадку умова (2.4) накладає додаткові обмеження, унаслідок яких максимальне значення цільової функції задачі цілочисельного програмування виявляється менше значення цільової функції відповідної задачі лінійного програмування.

Значення задач цілочисельного програмування. Вище було показано, що, як правило, практичне використання моделей математичного програмування пов’язано з прийняттям планових рішень у складних ситуаціях. Часто зустрічаються умови, коли моделі планування містять цілочисельні змінні.

1. Використання обладнання. Змінними x_j можна позначити одиниці устаткування, що повинні функціонувати протягом планового періоду, описуваного моделлю. У такому випадку на значення x_j приходиться накласти вимога цілочисельності.

2. Витрати на підготовку виробництва. Може виникнути необхідність розгляду операції, виконання якої пов’язане з так названими постійними витратами (або витратами на підготовку виробництва) C_j завжди, коли відповідне значення $x_j > 0$, причому C_j не залежить від фактичного значення x_j.

3. Розміри партій. При розробці деяких виробничих планів на значення x_j можуть накладатися обмеження виду $x_j = 0$ або $x_j \ge L_i$.

4. Рішення типу „так – ні ”. Може виникнути необхідність визначення ситуацій „або – або ” іншого характеру. З цією метою на змінні x_j

накладаються обмеження $x_j = 1$ або $x_j = 0$, що відповідають рішенням „прийняти ” або „відкинути ” чи „так ” або „ні ”. Досить часто альтернативи такого роду відносяться до розряду рішень про розподіл капіталовкладень, оскільки їхня реалізація пов’язана з великими витратами коштів і ресурсів. Це можна вважати основною причиною, що пояснює, чому цілочисельне програмування грає настільки важливу роль в організаційних рішеннях. Оптимальне рішення задачі розподілу капіталовкладень може принести фірмі набагато більший прибуток, ніж наближене або вгадане.

У ряді інших задач прийняття рішень може знадобитися побудова моделей цілочисельного програмування. Один клас таких задач пов’язаний з рішеннями, що визначають вибір упорядкувань, календарних планів (розкладів) і маршрутів. Прикладом задачі, що належить до цього класу, може служити задача комівояжера. У цій задачі потрібно відшукати найкоротший маршрут комівояжера, що повинен побувати в кожнім з n міст, причому маршрут починається та закінчується містом 1. Як інший приклад задач зазначеного класу можна взяти задачу складання розкладу обробки деталей на верстатах. Найбільш простим варіантом цієї задачі є випадок, коли кожна з n деталей повинна оброблятися на кожнім з k верстатів. Припустимо, що деталь не можна обробляти на верстаті j, поки не закінчена її обробка на верстаті j-1. Припустимо, далі, що час обробки кожної деталі на кожному верстаті строго фіксований. Тоді оптимальна послідовність визначається розкладом обробки, що мінімізує загальну тривалість обробки всіх деталей на усіх верстатах. Іншими прикладами задач упорядкування, вибору маршруту і календарного планування є задачі балансу складальних ліній, мережного планування при обмежених ресурсах, попереджувального ремонту при обмеженнях на робочу силу і диспетчирування автотранспорту.

Задачі упорядкування, календарного планування і вибору маршруту є окремими випадками комбінаторних задач.

Комбінаторна оптимізаційна задача полягає у пошуку серед кінцевої множини альтернатив однієї, якій відповідає екстремальне значення прийнятої цільової функції. Так, наприклад, у задачі комівояжера при n містах ця кінцева безліч містить (n–1)! різних допустимих маршрутів, що починаються та закінчуються в місті 1. У найпростішій модифікації задачі обробки деталей кінцева множина альтернатив включає $(n!)^k$ допустимих послідовностей обробки n деталей на k верстатах.

Існує один підхід до пошуку цілочисельних рішень, що часто пропонують початківці. Цей підхід зводиться до того, що спочатку не враховуються умови цілочисельності і визначається оптимальне рішення звичайної задачі лінійного програмування. Якщо отримане рішення задовольняє обмеженням цілочисельності, то воно є оптимальним рішенням і вихідній цілочисельній задачі. У протилежному випадку пропонується переходити до цілочисельного рішення за допомогою округлення рішення звичайної задачі лінійного програмування до цілих чисел.

Якщо деякі коефіцієнти a_{ij} у лінійних обмеженнях (2.2) у якій-небудь

конкретній моделі негативні, то сама задача округлення звичайного дрібного рішення до припустимого цілочисельного може виявитися досить складною. Таким чином, хоча за допомогою методу округлення у деяких випадках і вдається відшукувати рішення цілочисельних задач, не можна розраховувати, що цей метод можна використовувати як універсальний.

Ще більш неприємна особливість, що є властивою задачам цілочисельного програмування, полягає в тому, що немає простого способу, який дозволяє визначити, чи є дане припустиме рішення оптимальним. У цьому полягає одне з важливих помітних розходжень між задачами цілочисельного та лінійного програмування.

Практично ефективні алгоритми. З усього викладеного вище очевидно, що загальний алгоритм рішення задач цілочисельного програмування повинний виключати необхідність явного перебору всіх допустимих альтернатив. Потрібні методи, що забезпечують частковий перебір порівняно невеликого числа допустимих варіантів і неявний перебір всіх інших. Нагадаємо, що симплексний метод, застосовуваний для рішення звичайних задач лінійного програмування, має саме такі характеристики. Цей метод передбачає оцінку лише невеликого числа всіх припустимих базисних рішень. Аналогічно в рекурентних співвідношеннях, використовуваних у методі динамічного програмування, застосовують принцип оптимальності, що дозволяє усунути необхідність перебору всіх припустимих рішень. Ефективність цих методів оптимізації, заснованих на частковому переборі, виправдує постановку питання про пошук аналогічних підходів до рішення задач цілочисельного програмування.

2.1. Постановки задач цілочисельного програмування

Розміри партій. Приклад з альтернативними обмеженнями.

Припустимо, що x_j є кількість виробів, що випускається протягом деякого планового періоду. При цьому можливі дві ситуації: $x_j \geq L_j$, де L_j – найменший допустимий розмір партії, або $x_j = 0$ протягом усього розглянутого періоду. Припустимо, що можна задати досить велике число U_j, таке, що обмеження $(x_j \leq U_j)$ в оптимальному рішенні напевно виконується. Тоді дихотомію ($x_j = 0$ або $x_j \geq L_j$) можна виразити, ввівши булєву змінну ($y_j = 0$ або $y_j = 1$) і два лінійних обмеження

$$x_j - U_j y_j \leq 0, \tag{2.5}$$

$$x_j - L_j y_j \geq 0. \tag{2.6}$$

При $y_j = 0$ з обмежень (2.5) та (2.6) походить, що вироби не випускаються ($x_j = 0$). При $y_j = 1$ обмеження (2.5) втрачає сенс, а з обмеження (2.6) походить задана умова на розмір партії.

Викладений підхід можна узагальнити, поширивши його на випадки, коли рішення повинне задовольняти принаймні k з p обмежень:

$$\sum_{j=1}^{n} a_{ij}x_j \le b_i, \quad i = 1,2,\ldots,p. \tag{2.7}$$

Введемо *p* булєвих змінних $y_i = 0$ або $i = 1,2,\ldots,p$ та накладемо 1+p лінійних обмежень:

$$\sum_{i=1}^{p} y_i \ge k, \tag{2.8}$$

$$\sum_{j=1}^{m} a_{ij}x_j \le b_i y_i + U_i(1 - y_i), \tag{2.9}$$

або

$$\sum_{j=1}^{n} a_{ij}x_j - (b_i - U_i)y_i \le U_i, \quad i = 1,2,\ldots,p,$$

де значення U_i вибирається настільки великим, щоб умова $(\sum_{j=1}^{n} a_{ij}x_j \le U_i)$ обов'язково виконувалося в оптимальному рішенні.

Розподіл капіталовкладень. Приклад із взаємозалежними альтернативами. Мабуть, найбільш важливими організаційними рішеннями, що приводять до задач цілочисельного програмування, є альтернативи „так – ні ". Альтернативи такого роду часто виникають у стратегічних моделях планування, що відносяться до декількох планових інтервалів. Звичайне значення змінної x_j визначає вибір конкретного проекту (операції або варіанта капіталовкладень), де

$$x_j = \begin{cases} 1, & \text{якщо проект } j \text{ виконується,} \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Відповідний коефіцієнт a_{ij} в обмеженні часто виражає кількість ресурсу, що обмежує, скажемо готівкових коштів, які потрібні для виконання проекту j на інтервалі i, де загальна кількість цього ресурсу, використовувана на кожному інтервалі, обмежена. Обмеження подібного роду аналогічні обмеженням, що зустрічаються в звичайних моделях лінійного програмування, призначених для рішення задач довгострокового стратегічного планування. Тому конкретні приклади таких задач тут не приводяться.

Однак, унаслідок того, що змінна x_j є двоїстою, її можна використовувати для відображення обмежень комбінаторного типу, що часто характеризує задачі розподілу капіталовкладень. Припустимо, приміром, що потрібно накласти обмеження на рішення, припустивши, щоб воно допускало вибір не більш k з перших p проектів. (Можливо, що для реалізації кожного проекту потрібне призначення провідного інженера, а є можливість виділити усього k інженерів такої кваліфікації.) Тоді ця умова може бути виражена за допомогою лінійного обмеження:

$$\sum_{j=1}^{p} x_j \le k. \tag{2.10}$$

Якщо записати обмеження (2.10) у вигляді рівності, то будуть прийняті до виконання в точності k проектів. Особливий випадок наявності p взаємовиключних проектів можна врахувати, прийнявши в (2.10) k=1.

Задача комівояжера. Комівояжеру потрібно побувати в кожному з n міст, починаючи й закінчуючи свій маршрут містом 1. Він не має права двічі заїжджати в жодне з міст. Позначимо через $c_{ij} \geq 0$ відстань між містами i та j, прийнявши $c_{ij} = \infty$, якщо прямого маршруту між містами i та j не існує. У деяких випадках $c_{ij} \neq c_{ji}$. Оптимізаційна задача полягає у пошуку найкоротшого циклу.

Запропоновано кілька математичних формулювань цієї задачі. У формулюванні, що нижче приводиться, використовується відносно невелика кількість змінних. Визначимо булєві змінні у такий спосіб:

$$x_j = \begin{cases} 1, & \text{якщо цикл включає переїзд з міста } i \text{ у місто } j, \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Цільова функція записується в наступному вигляді:
мінімізувати

$$\sum_i^n \sum_j^m c_{ij} x_{ij}, \quad \text{де } c_{ij} = \infty, \quad i = 1,2,\ldots,n. \tag{2.11}$$

Змінні повинні задовольняти обмеженням:

$$\sum_i^n x_{ij} = 1, \quad i = 1,2,\ldots,n \ \text{(від`їзд)}, \tag{2.12}$$

$$\sum_j^n x_{ij} = 1, \quad j = 1,2,\ldots,n \ \text{(прибуття)}, \tag{2.13}$$

$$x_{ij} \text{ – ненегативні цілі при будь-якому i та j.} \tag{2.14}$$

(Умова $c_{ij} = \infty$ приймається для того, щоб виключити можливість появи в оптимальному рішенні значень $x_{ij} = 1$, що не мають сенсу. З іншого боку, можна просто виключити змінні x_{ij} з умов задачі.) Обмеження (2.12) – (2.14) забезпечують рівність кожної змінної x_{ij} або 0, або 1. Співвідношення (2.12) вимагають, щоб цикл включав у точності один виїзд із кожного міста, аналогічно співвідношення (2.13) вимагають у точності одного прибуття в кожне місто.

На жаль, хоча змінні x_{ij} у будь-якому циклі повинні задовольняти зазначеним вище обмеженням, припустиме при цих обмеженнях рішення не обов’язково є циклом. Зокрема, допустиме рішення, що задовольняє умовам (2.12), (2.13) і (2.14), може включати два або більш незв’язаних між собою циклів або підциклів, наприклад,

$$x_{12} = x_{23} = x_{31} = 1,$$

та

$$x_{45} = x_{56} = \ldots = x_{n4} = 1.$$

Тому необхідно ввести додаткові обмеження, що забезпечують одержання рішення у вигляді циклу. Можна застосувати витончений спосіб завдання лінійних обмежень, що виключає виникнення всіх підциклів.

Введемо змінні u_i, $i = 1,2,\ldots,n$ та накладемо на них наступні $(n-1)^2 - (n-1)$ умов:

$$u_i - u_f + nx_{ij} \le n - 1, \quad \begin{array}{l} i = 1,2,\ldots,n, \\ j = 1,2,\ldots,n \ (i \ne j) \end{array} \qquad (2.15)$$

Щоб усунути виникнення підциклів, на змінні u_i не потрібно накладати ніяких додаткових обмежень, однак накладення умови незаперечності та цілочисельності не може принести ніякого збитку. В умови (2.12) – (2.15) входять $n^2 - n + 1$ лінійні обмеження та $n^2 + n - 1$ цілочисельних змінних, з яких n змінних x_{ii} повинні дорівнювати нулю.

Для демонстрації ефективності обмежень (2.15) потрібно довести два твердження: а) ці обмеження дійсно виключають усі підцикли; б) жоден повний цикл не виключається. Ці два твердження розглядаються послідовно.

Насамперед обмеження (2.12) – (2.15) виключають можливість появи всіх підциклів між містами 2, 3, ..., n. Щоб переконатися в цьому, розглянемо будь-який цикл між k з цих міст, що визначається значеннями $x_{ij} = 1$ для k змінних. Додамо в (2.15) k обмежень, що відповідають перемінним, утворюючим підцикл. Такому загальному обмеженню повинне також задовольняти будь-яке рішення, припустиме за умовою (2.15). Варто переконатися в тому, що для кожного з *k* міст в отриманій сумі фігурують величини u_i та $(-u_i)$. Отже, у загальному обмеженні немає жодної величини u_i. Однак це неприпустимо для всіх $x_{ij} = 1$ у цьому обмеженні, тому що в такому випадку сума в лівій частині нерівності, що дорівнює nk, буде більше суми в правій частині нерівності, що дорівнює (n–1)k. Отже, умова (2.15) виключає можливість появи будь-якого підциклу.

Покажемо тепер, що обмеження (2.12) – (2.15) не приводять до усунення будь-якого повного циклу. Для цього досить установити, що існують значення u_i, що задовольняють обмеженню (2.15), для будь-якого заданого циклу. Розглянемо такий цикл, що по визначенню полягає у відвідуванні кожного міста 2, 3, ..., n один і тільки один раз. Нехай t_i є положення в такому циклі, коли комівояжер попадає в місто i, причому для міста 1 приймемо t_1–1. Тоді в циклі місто 1 – місто 3 – місто 5 ... значення будуть дорівнювати: $t_1 = 1$, $t_3 = 2$, $t_5 = 3$, ... При цій умові $u_i = t_i$, $i = 2,3,\ldots,n$, є допустима множина значень t_i. Щоб переконатися в цьому, припустимо, що в даному циклі $x_{ij} = 1$, так що $t_j = t_i + 1$. Тоді обмеження для цього x_{ij} в (2.15) задовольняється, оскільки $t_i - (t_i + 1) + n(1) \le n - 1$.

Припустимо тепер, що $x_{ij} = 0$. Тоді відповідна нерівність у (2.15) приймає вигляд $(u_i - u_j \le n - 1)$. Це нерівність повинна виконуватися, оскільки $u_i < n$ та $u_j > 1$.

2.2. Загальні відомості про методи рішення задач цілочисельного програмування

Відомими є два основних підходи до пошуку точного оптимального рішення задач цілочисельного програмування.

1. Методи площин, що відсікають, (методи відсікання). Запропоновано кілька варіантів цього підходу до рішення цілочисельних задач. Один з них, що викладається в наступному розділі, призначений для цілком цілочисельної моделі. Вихідним моментом є оптимальне рішення відповідної задачі лінійного програмування, отриманої в результаті відкидання умов цілочисельності. На кожній ітерації додається лінійне обмеження, що задовольняє цілочисельному рішенню вихідної задачі, але таке, що виключає поточне нецілочисельне рішення. Обчислювальний процес припиняється, як тільки буде досягнуте будь-яке цілочисельне рішення. Збіжність забезпечується за кінцеве, але іноді дуже велике число ітерацій.

2. Методи, повернення. У цій групі методів також маються різні модифікації. Перший метод, названий методом „галузей і границь”, призначений для рішення частково цілочисельних задач. Як і в методі відсікання, рішення задачі починається з пошуку оптимального рішення відповідної регулярної задачі лінійного програмування. Потім формується сімейство зв'язаних, але різних задач лінійного програмування. На прикладі задачі комівояжера розглядається кілька варіантів цього підходу. Нарешті викладається алгоритм (неявного) часткового перебору, що застосовується тут тільки до задач, що містять булєві цілочисельні змінні. Завдяки такій особливій структурі задачі обчислювальні процедури істотно спрощуються. Термін „повернення” визначає специфічний спосіб формування і рішення послідовності задач.

Сутність алгоритмів, заснованих на методі відсікання, легко усвідомити, звернувшись до геометричних подань у просторі рішень. Визначимо опуклу оболонку множини припустимих цілочисельних точок (рішень) як мінімальна опукла множина, що містить усі ці точки (або, що є такою ж, як множина точок евклідового n–мірного простору, що складається з усіх опуклих комбінацій вигляду:

$$[wx_1 + (1-w)y_1,\ wx_2 + (1-w)y_2,\ \ldots,\ wx_n + (1-w)y_n],$$

де $0 < w < 1$, а також $(x_1, x_2, \ldots, x_n)$ і $(y_1, y_2, \ldots, y_n)$ – припустимі цілочисельні рішення).

Припустимо, що оптимальне значення лінійної цільової функції є кінцевим. Тоді легко показати, що цільова функція досягає оптимального значення в одній з вершин цієї опуклої оболонки. Така екстремальна точка є однією з припустимих цілочисельних рішень.

У свою чергу опуклу оболонку можна представити кінцевою множиною лінійних обмежень. В алгоритмі відсікання спочатку розглядається опукла множина, визначена лінійними обмеженнями й умовами незаперечності змінних вихідної задачі, й відшукується екстремальна точка цієї множини.

Якщо таке рішення виявляється нецілочисельним, то додають обмеження,

що відсікає поточну екстремальну точку і зменшує „обсяг ” опуклої множини. Однак нове обмеження не відсікає жодної екстремальної точки опуклої оболонки, що належить допустимим цілочисельним рішенням. Зрештою вводиться таке число додаткових обмежень, що екстремальна точка усіченої опуклої множини збігається з екстремальною точкою опуклої оболонки.

Докладний виклад алгоритму. Допустимо, що цілком цілочисельна задач має такий вигляд:

максимізувати $$\sum_{j=1}^{n} c_j x_j \qquad (2.16)$$

при обмеженнях

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1,2,\ldots,m, \qquad (2.17)$$

всі x_j – ненегативні цілі. (2.18)

Припустимо, що при обмеженнях (2.17) і (2.18) існує припустиме рішення і що в оптимальному рішенні цільова функція приймає кінцеве значення. Перш ніж сформулювати конкретні кроки алгоритму, покажемо в загальному виді, як побудувати додаткові лінійні обмеження, яким повинне задовольняти будь-яке рішення, припустиме за умовами (2.17) і (2.18).

Розглянемо будь-яке лінійне рівняння, яке можна одержати за допомогою алгебраїчних операцій над лінійними обмеженнями (2.17), наприклад, шляхом додавання двох або більше рівнянь або множення рівняння на постійну, відмінну від нуля. Тоді, якщо деяке рішення задовольняє умові (2.17), вона має також задовольняти й новому рівнянню. Приймемо, що вираз

$$\sum_{j=1}^{n} a_j x_j = b, \qquad (2.19)$$

визначає саме таке нове обмеження. Можливо, одна або кілька величин a_j та b є дробовими, скажемо 2,5 або 10/3.

Позначимо символом [d] цілу частину d, тобто найбільше ціле число, менше або рівне дійсному числу d. Наприклад,

$$[2,5] = 2, \qquad [\frac{10}{3}] = 3, \qquad [4] = 4,$$

$$[-2,5] = -3, \qquad [-\frac{10}{3}] = -4, \qquad [-4] = -4. \qquad (2.20)$$

Оскільки на величини x_j накладене обмеження (2.18), звідки походить, що вони повинні бути ненегативними цілими, будь-які значення x_j, що задовольняють обмеженню (2.17), а отже, і (2.18), повинні, мабуть, задовольняти більш слабкому обмеженню

$$\sum_{j=1}^{n} [a_j]\, x_j = b. \qquad (2.21)$$

Далі, оскільки сума в лівій частині (2.21) повинна бути цілочисельною, потрібно довести, що (2.21) можна підсилити в такий спосіб:

$$\sum_{j=1}^{n}[a_j]\ x_j = [b]. \qquad (2.22)$$

Нарешті, (2.22) можна перетворити в рівність, додавши вільну змінну

$$\sum_{j=1}^{n}[a_j]\ x_j + x = [b] \quad \text{(цілі частини)}, \qquad (2.23)$$

тобто на x накладена умова незаперечності та цілочисельності. Таким чином, якщо до умови (2.17) додати лінійне обмеження (2.23), задача все ж залишиться повністю цілочисельною.

У результаті отримане обмеження (2.23), в якому величини a_j та b отримані за допомогою алгебраїчних операцій над обмеженнями (2.17) та якому задовольняє будь-яке припустиме за умовами (2.17) і (2.18) рішення.

Для спрощення словесного опису алгоритму виконаємо елементарні перетворення над (2.19) і (2.23). Визначимо значення f_j та f тотожностями

$$[a_j] + f_j \equiv a_j \qquad \text{та} \qquad [b] + f \equiv b, \qquad (2.24)$$

такими, що $0 \le f_j < 1$ та $0 \le f < 1$. Ці величини називаються дробовими частинами a_j та b. (Для приклада дробова частина величин 2,5 і –2,5 дорівнює 0,5, величини 10/3 дорівнює 1/3 і величини –10/3 дорівнює 2/3.)

Віднімаючи (2.19) з (2.23), одержимо обмеження, що складається тільки з дробових частин

$$\sum_{j=1}^{n}(-f_j)\,x_j + x = -f \quad \text{(дробові частини)}. \qquad (2.25)$$

В алгоритмі, що нижче приводиться, замість (2.23) використовується обмеження (2.25).

Алгоритм відсікання складається з наступних кроків.

Крок 1. Знайти оптимальне рішення задачі лінійного програмування з цільовою функцією (2.16) і лінійних обмежень (2.17), не з огляду на умови цілочисельності (2.18), але вимагаючи, щоб усі $x_j \ge 0$.

Крок 2. Припинити обчислення, якщо поточне рішення задачі лінійного програмування є цілочисельним. У протилежному випадку вибрати яку-небудь дробову базисну змінну. Скласти обмеження (2.25) з рівняння, що містить цю базисну змінну в поточному оптимальному рішенні задачі лінійного програмування.

Крок 3. Додати до вихідної задачі лінійного програмування це нове обмеження, знайти нове оптимальне рішення задачі з додатковим обмеженням і повернутися до кроку 2.

Заключні зауваження. Практичний досвід реалізації обчислювальних схем складних варіантів алгоритмів, побудованих на методі відсікання, ні в якій мері не можна визнати успішним у всіх випадках. Зокрема, ці алгоритми виявилися мало ефективними при рішенні комбінаторних задач, поданих у цілочисельній

формулюванні. Виникли також труднощі при рішенні задач, у яких величини a_j та b мають великі значення. Однак алгоритми цього класу добре зарекомендували себе при рішенні задач, в яких оптимальне рішення задачі лінійного програмування вже містить цілочисельні значення багатьох змінних.

Найбільші обчислювальні труднощі виникають при пошуку оптимального рішення на кроці 3. Після виконання декількох ітерацій на кроці 3 можуть з'явитися численні альтернативні оптимальні рішення задачі лінійного програмування. Внаслідок цього цільова функція залишається постійною (зазвичай приймаючи неоптимальне значення) протягом багатьох ітерацій, поки альтернативні рішення поступово не будуть виключені в силу обмежень, формованих на кроці 2. На жаль, це явище часто виникає в задачах середньої та великої розмірності. Існує також багато прикладів задач малої розмірності (не більше 10 змінних і рівнянь), при рішенні яких для досягнення збіжності будуть потрібні тисячі ітерацій.

2.3. Метод гілок і границь

Цей метод можна застосовувати для рішення як повністю, так і частково цілочисельних задач дискретного програмування. Припустимо для визначеності, що розглянута модель має такий вигляд:

максимізувати $$\sum_{j=1}^{n} c_j x_j \tag{2.26}$$

при обмеженнях

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1,2,\ldots,m, \tag{2.27}$$

$$x_j \text{ – цілі}, \quad j = 1,2,\ldots,p\,(\le n), \tag{2.28}$$

$$x_j \ge 0, \qquad j = p+1,\ldots,n. \tag{2.29}$$

Крім того, допустимо, що для кожної цілочисельної змінної можна задати верхню і нижню межі, у межах яких безумовно знаходяться її оптимальні значення

$$L_j \le x_j \le U_j, \qquad j = 1,2,\ldots,p. \tag{2.30}$$

Звичайно $L_j = 0$, але ця умова не обов'язкова. (Припущення, що всі L_j, не привзодить до втрати спільності. Однак, користуючись загальним символом L_j, алгоритм простіше описати).

Ідея методу гілок і границь заснована на наступному елементарному факті. Розглянемо будь-яку змінну x_j і приймемо, що I є деяке ціле число, де $L_j \le I \le U_j - 1$. Тоді оптимальне рішення задачі (2.26) – (2.30) буде задовольняти або лінійному обмеженню

$$x_j \ge I + 1, \tag{2.31}$$

або лінійному обмеженню

$$x_j \le I. \tag{2.32}$$

Алгоритм. На кожній ітерації (позначимо її t) існує нижня границя (оцінка), скажемо x_0^t, оптимального значення цільової функції. Для спрощення викладу припустимо, що на першій ітерації значення x_0^1 або строго менше оптимального значення, або дорівнює значенню цільової функції, що відповідає фіксованому припустимому рішенню. При відсутності якоїсь додаткової інформації про особливості розглянутої задачі в самому несприятливому випадку можна прийняти $x_0^1 = -\infty$. Крім нижньої межі існує основний список задач лінійного програмування, що підлягають рішенню. Єдина відмінність цих задач однієї від одної пов'язано зі зміною умов (2.30). На ітерації 1 основний список містить всього одну задачу (2.26), (2.27), (2.29) та (2.30). На довільній ітерації t виконуються наступні кроки.

Крок 1. Припинити обчислення, якщо основний список порожній. У противному випадку вибрати одну задачу лінійного програмування з основного списку.

Крок 2. Розв'язати обрану задачу. Якщо вона не має припустимого рішення або якщо отримане у результаті оптимальне значення цільової функції менше або дорівнює x_0^t, то прийняти $x_0^{t+1} = x_0^t$ і повернутися до кроку 1. У протилежному випадку перейти до кроку 3.

Крок 3. Якщо отримане оптимальне рішення задачі лінійного програмування задовольняє цілочисельним обмеженням, то зафіксувати його, прийняти x_0^{t+1} рівним відповідному оптимальному значенню цільової функції і повернутися до кроку 1. У протилежному випадку перейти до кроку 4.

Крок 4. Вибрати будь-яку змінну x_j, $j = 1, 2, \ldots, p$, що не має цілого значення в отриманому оптимальному рішенні обраної задачі лінійного програмування. Нехай b_j відповідає цьому значенню, а $[b_j]$ визначає найбільше ціле число, що менше або дорівнює b_j. Додати дві задачі лінійного програмування в основний список. Ці дві задачі ідентичні задачі, обраній на кроці 1, за винятком того, що в одній з них нижня границя x_j замінена на $[b_j]+1$, а в іншій верхня границя x_i замінена на $[b_j]$. Прийняти $x_0^{t+1} = x_0^t$ та повернутися до кроку 1.

При зупинці алгоритму у випадку, коли фіксується припустиме рішення, що дає x_0^t, це рішення оптимальне, у протилежному випадку припустимого рішення не існує. Можна одержати цілочисельне рішення, не дійшовши до останньої ітерації, однак при цьому не відомо, чи є воно дійсно оптимальним. З цієї причини алгоритм можна назвати приблизно двоїстим.

Заключні зауваження. Метод гілок і границь поряд з методом відсікання з обчислювальної точки зору має істотні достоїнства. Справа в тому, що алгоритми, побудовані на цих методах, порівняно легко програмуються для ЕОМ і тому зазначені алгоритми реалізуються на будь-якій ітерації без втручання людини. Метод гілок і границь ефективний при рішенні задач, що містять невелике число цілочисельних змінних. Однак, якщо число таких

змінних є великим або рішення задачі лінійного програмування є далеким від оптимального рішення цілочисельної задачі (як це мало місце в розглянутому прикладі), то число ітерацій, необхідних для одержання рішення, може виявитися занадто великим і для реалізації алгоритму будуть потрібні невиправдані з практичної точки зору витрати машинного часу.

2.4. Задача комівояжера

Покажемо тепер, як доцільно модифікувати метод галузей і границь, викладений у попередньому розділі, щоб мати можливість використовувати особливі структурні властивості комбінаторних задач. З цією метою обрана задача комівояжера, на прикладі якої показані три різні модифікації методу гілок і границь.

Задачу комівояжера зручно записувати в формулюванні, розглянутої раніше:

мінімізувати

$$\sum_{i}^{n}\sum_{j}^{m} c_{ij}x_{ij}, \quad \text{де } c_{ij} = \infty, \quad i = 1,2,\ldots,n, \tag{2.33}$$

При обмеженнях

$$\sum_{i}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n \ \text{(від`їзд)}, \tag{2.34}$$

$$\sum_{j}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n \ \text{(прибуття)}, \tag{2.35}$$

x_{ij} – ненегативні цілі при всіх і та j, (2.36)

рішення є цикл. (2.37)

Нагадаємо, що $x_{ij} = 1$ означає, що комівояжер переїжджає з міста *i* безпосередньо в місто *j*, де $c_{ij} \geq 0$ – відповідна відстань між містами. Якщо виключити умову (2.37), то (2.33) – (2.36) є задачею про призначення, яку можна вирішити кожним з методів, розглянутих раніше. (Відзначимо, однак, що в даному випадку метою є „мінімізація ”, в той час як раніше розглядалися задачі „максимізації ”. Модифікації методу гілок і границь, що викладаються нижче, відрізняються цілком очевидними деталями від приведеної раніше модифікації, що викликано зміною змісту цільової функції).

Три різні модифікації методу гілок і границь розглядаються на прикладі конкретної задачі, вихідні дані якої зведені в таблицю рис. 2.1. Підкреслимо, що єдине рішення відповідної задачі про призначення $x_{15} = x_{51} = x_{23} = x_{34} = x_{42} = 1$ не задовольняє умові (2.37), тому що воно містить два цикли. Тому сумарна відстань, що обумовлена цим рішенням і дорівнює 60, строго менше довжини оптимального циклу. Існує усього (n-1)! = 4! = 24 можливих циклів.

Метод виключення підциклів. З трьох розглянутих надалі модифікацій методу гілок і границь ця в найменшому ступені відрізняється від викладеного раніше алгоритму. На початку ітерації t відома верхня границя (оцінка) x_0^t оптимального значення цільової функції.

З міста \ В місто	1	2	3	4	5
1	∞	10	25	25	10
2	1	∞	10	15	2
3	8	5	∞	20	10
4	14	10	24	∞	15
5	10	8	25	27	∞

Відстань c_{ij}

Оптимальне значення: $C_{15}+C_{23}+C_{34}+C_{42}+C_{51}=60$

Рис. 2.1. Умови задачі комівояжера

Можна прийняти x_0^t рівним досить великому числу, скажемо сумі ($c_{12} + c_{23} + ... + c_{nl}$), що відповідає циклу, що включає проїзд з міста 1 у місто 2... у місто *n*, в місто 1. Крім того, існує основний список, що містить ряд задач про призначення. Всі ці задачі мають вигляд (2.33) – (2.36), але відрізняються одна від одної тим, що в них різні величини c_{ij} дорівнюють ∞. На ітерації 1 основний список складається із задачі про призначення (2.33) – (2.36), що назвемо задачею 1. Визначимо підцикл як цикл, що містить не всі n міст, скажемо $x_{15} = x_{51} = 1$ або $x_{23} = x_{34} = x_{42} = 1$.

На ітерації t виконуються наступні кроки.

Крок 1. Припинити обчислення, якщо основний список порожній. У протилежному випадку вибрати одну задачу, викресливши її з основного списку.

Крок 2. Вирішити обрану задачу про призначення. Якщо оптимальне значення цільової функції (яке може бути дорівнювати ∞) більше або дорівнювати x_0^t, то прийняти $x_0^{t+1} = x_0^t$ і повернутися до кроку 1. У противному випадку перейти до кроку 3.

Крок 3. Якщо отримане оптимальне рішення обраної задачі про призначення є циклом, то зафіксувати його, прийняти x_0^{t+1} рівним відповідному оптимальному значенню цільової функції та повернутися до кроку 1. У протилежному випадку перейти до кроку 4.

Крок 4. Зупинитися в отриманому оптимальному рішенні обраної задачі про призначення на підциклі, що містить мінімальне число міст. Кожному $x_{ij} = 1$ в обраному підциклі поставити у відповідність задачу про призначення, вносячи її в основний список й прийнявши відповідне значення $c_{ij} = \infty$. Залишити всі інші коефіцієнти тими ж, що в задачі, вибраній на кроці 1. Прийняти $x_0^2 = x_0^1 = 65$ та повернутися до кроку 1.

Метод завдання маршрутів. Викладений вище алгоритм дозволяє обмежити

число розгалужень, що переглядаються, що досягається за рахунок обчислення ефективної нижньої оцінки (межі) цільової функції для будь-якого циклу, породжуваного кожною задачею. Однак для одержання цієї оцінки приходиться відшукувати оптимальне рішення задачі про призначення. Покажемо тепер, як можна обчислювати оцінки більш простим способом. Однак ціна, яку приходиться платити за таке спрощення, визначається тим, що в цьому алгоритмі потрібно досліджувати більше число гілок відповідного дерева.

На початку будь-якої ітерації t відома верхня оцінка x_0^t оптимального значення цільової функції. Значення x_0^t можна визначити, виходячи з тих же розумінь, що викладалися вище. Крім того, мається основний список задач, у якому деяка підмножина значень c_{ij} змінена та прийнята рівною ∞, а підмножина $x_{ij} = 1$. Серед значень $x_{ij} = 1$ відсутні набори, що утворять підцикли. На ітерації 1 основний список містить дві задачі: в одній з них значення обраного c_{ij} змінене на ∞, в іншій – відповідна змінна $x_{ij} = 1$, а $c_{ij} = \infty$. (Вважаючи $c_{ij} = \infty$, забороняють тим самим $x_{ij} = 1$, що приводило б до утворення підциклу місто i – місто j – місто i).

Положення кожної задачі, що входить в основний список, зручно визначати у такий спосіб. Візьмемо матрицю c_{ij}, подібну приведеній на рис. 2.1. Якщо $x_{ij} = 1$, викреслимо k-й рядок та h-й стовпець. Інші c_{ij} (з яких деякі дорівнюють ∞) відповідають ще не призначеним перемінним x та j. Нижню оцінку оптимального значення цільової функції для будь-якого циклу, що містить задану підмножину $x_{ij} = 1$, можна обчислити різними способами. У цілому, чим більше нижня оцінка, тим менше число гілок приходиться досліджувати.

Один простий, але досить ефективний метод обчислення цієї оцінки заснований на тому, що відстань повинна бути принаймні рівним сумі c_{ij} при $x_{ij} = 1$ плюс сума найменших c_{ij} у кожному з невикреслених рядків. Можна (і потрібно) збільшити цю оцінку в ще більшому ступені, віднімаючи мінімальний коефіцієнт c_{ij} у кожнім невикресленому рядку з всіх інших c_{ij} цього рядка. Таким чином, отримуються так називані зменшені відстані. Далі, до зазначеної вище суми додається мінімальне зі зменшених відстаней по кожному з невикреслених стовпців. Приклад, що ілюструє цю процедуру, приведену у таблиці (рис. 2.2). Відмітимо, що значення величини x_{23} було прийнято рівним одиниці. У верхній таблиці зазначені мінімальні відстані по кожному рядку, крім другого. Кожне з цих відстаней віднімається з кожного елемента у відповідному рядку, що дає нижню таблицю.

(Насправді, можна знайти нижню оцінку, вирішуючи задачу про призначення, утворену з величин c_{ij}, що належать невикресленим рядкам і стовпцям, додаючи отримане оптимальне значення цільової функції до суми

коефіцієнтів c_{ij}, для яких задана підмножина $x_{ij} = 1$. Такий підхід, мабуть, більш трудомісткий з обчислювальної точки зору, ніж тільки що описана модифікація методів гілок і границь або викладений простий прийом обчислення оцінок). На ітерації t виконуються наступні кроки.

Крок 1. Припинити обчислення, якщо основний список порожній. У протилежному випадку вибрати одну задачу і викреслити її з основного списку.

Крок 2. Визначити нижню оцінку цільової функції для будь-якого циклу, породжуваного обраною задачею. Якщо нижня оцінка більше або дорівнює x_0^t, то прийняти $x_0^{t+1} = x_0^t$ та повернутися до кроку 1. У протилежному випадку перейти до кроку 3.

Крок 3. Якщо поточне рішення визначає цикл, то зафіксувати його, прийняти x_0^{t+1} рівним відповідному значенню цільової функції та повернутися до кроку 1. У протилежному випадку перейти до кроку 4.

Крок 4. При наявності можливості вибрати змінну x_{hk}, що не входить у поточне рішення, таку, що $c_{hk} < \infty$ за умови, що $x_{hk} = 1$ не призводить до утворення підциклу на змінних, що вже увійшли в рішення. При такому виборі внести до основного списку дві задачі. Кожну з цих задач прийняти ідентичній задачі, обраній на кроці 1, за винятком лише того, що в одну з них ввести зміну $c_{hk} = \infty$, а в іншу – умову $x_{hk} = 1$ та зміну $c_{hk} = \infty$. Прийняти $x_0^{t+1} = x_0^t$ та повернутися до кроку 1.

Метод часткових циклів. Головною причиною викладу цієї модифікації методу гілок і границь є ознайомлення з підходом, використовуваним для рішення всіляких комбінаторних задач, наприклад, задач упорядкування (у яких $x_{ij} = 1$ відповідає розміщенню i-го елемента на j-му місці деякої послідовності). Для цього буде потрібно використання поняття частковий цикл, що визначається як послідовність, що містить менше n різних міст і починається з міста 1 (наприклад, місто 1 – місто i– місто j– місто k, де n > 4).

На початку будь-якої ітерації t відома верхня оцінка x_0^t оптимального значення цільової функції. Значення x_0^t визначається загальноприйнятим способом. Крім того, заданий основний список задач, що містить деяку підмножину $x_{ij} = 1$, що визначає частковий цикл, і підмножина значень c_{ij}, прийнятих у результаті перегляду рівними ∞. На ітерації 1 основний список містить n-1 задач, по одній для кожного $x_{1j} = 1$ і переглянутих $c_{j1} = \infty,\ j = 2,3,\ldots,n.$ Для обчислення нижньої оцінки оптимального значення цільової функції, що відповідає циклу, що є доповненням часткового циклу, можна застосувати той же метод, що й в алгоритмі завдання маршрутів. (З іншого боку, можна визначати оптимальне рішення задачі про призначення, включивши в цю задачу коефіцієнти c_{ij}, що належать рядкам і стовпцям, не пов’язаним з підмножиною $x_{ij} = 1$, що входять у частковий цикл.)

$x_{21} = 1$ $(c_{23} = 10)$

З міста \ У місто	1	2	3	4	5	Мінімум по рядках
1	∞	10		25	10	10
2						
3	8	∞		20	10	8
4	14	10		∞	15	10
5	10	8		27	∞	8

Зменшені відстані

З міста \ У місто	1	2	3	4	5
1	∞	0		15	0
2					
3	0	∞		12	2
4	4	0		∞	5
5	2	0		13	∞
Мінімум по стовпцях	0	0		12	0

Оцінка: $c_{23} + \Sigma$*(Мінімум по рядках)*$+\Sigma$*(Мінімум по стовпцях) =*

= 10+10+8+10+8+0+0+12+0 = 58

Рис. 2.2. Метод визначення оцінок в алгоритмі завдання маршрутів

Розглянута модифікація відрізняється від методу завдання маршрутів тільки кроком 4.

Крок 4. По кожному місту k, що не входить у частковий цикл задачі, обраної на кроці 1, внести додаткову задачу до основного списку, розширивши частковий цикл із міста j, що є останнім містом, включеним у частковий цикл, до міста k, змінивши при цьому c_{ij} на ∞. Прийняти $x_0^{t+1} = x_0^t$ та повернутися до кроку 1.

У цьому алгоритмі у випадку, коли частковий цикл містить m міст (включаючи місто 1), на кроці 4 в основний список включається п – т задач. Відзначимо, що на кроці 4 відсутній довільний вибір. (Надлишкові обчислення в даному алгоритмі можуть мати місце, якщо в основний список входять кілька задач, що містять у своїх часткових циклах ті самі т міст, наприклад місто 1 – місто 2 – місто 3 та місто 1 – місто 3 – місто 2. У цьому випадку обчислення оцінок на кроці 2 повторюються).

2.5. Метод часткового (неявного) перебору

Багато важливих задач цілочисельного програмування можна описати у такий спосіб:

максимізувати $$\sum_{j=1}^{n} c_{ij} x_{ij} \tag{2.38}$$

при обмеженнях

$$\sum_{j=1}^{n} a_{ij} x_{ij} \le b_i, \quad i = 1,2,\ldots,m, \tag{2.39}$$

де умови цілочисельності зведені до

$$x_j = \begin{cases} 0, \\ 1, \end{cases} \qquad j = 1,2,\ldots,n. \tag{2.40}$$

Припустимо, що будь-який коефіцієнт із c_j є ціле число (цього завжди можна добитися, вибравши правильний масштаб цільової функції за умови, що вихідні значення коефіцієнтів задані раціональними числами).

Моделі розподілу капіталовкладень часто можна представити у виді (2.38)–(2.40). Крім того, багато з повністю цілочисельних задач можна перетворити таким чином, щоб кожна змінна задовольняла умовам (2.40). Припустимо, наприклад, що змінна x має визначну верхню оцінку U. Тоді x у всіх вираженнях можна замінити еквівалентним двоїчним поданням

$$x \equiv 1w_1 + 2w_2 + 4w_3 + \ldots + 2^{k-1} w_k, \tag{2.41}$$

де $w_j = 1$ або 0 і значення k вибирається так, щоб виконувалася умова $2^{k-1} \ge U$.

Нагадаємо, що у випадку застосування алгоритму гілок і границь, приведеного раніше, до рішення задачі (2.38)–(2.40) приходиться вирішувати послідовність задач лінійного програмування. В алгоритмі, що нижче викладається, використовується умова (2.40), унаслідок чого обчислення обмежуються операціями додавання (і віднімання). З цієї причини такий метод іноді називають адитивним алгоритмом.

Якщо враховувати тільки умову (2.40), то існує 2^n можливих виборів значень (x_1, x_2, …, x_n). Зрозуміло, багато з них неприпустимі через лінійні обмеження (2.39) і лише дуже невелике число з них є оптимальними. Розглянемо деяку підмножину x_j, в якому кожному x_j поставлено у відповідність визначене числове значення (або нуль, або одиниця). Така підмножина називається частковим рішенням. Змінні x_j, що не входять у часткове рішення, називають вільними змінними. Будь-який конкретний вибір числових значень вільних змінних називається доповненням відповідного часткового рішення. Якщо часткове рішення містить *s* змінних, то існує 2^{n-s} доповнень. У розглянутому алгоритмі кожна задача основного списку відповідає частковому рішенню, а можливі доповнення утворюють гілки дерева.

Зміст розглянутих прикладів полягає у наступному. При заданому частковому рішенні представимо лінійні обмеження у вигляді

$$\sum_{\substack{\text{по всіх}\\ \text{вільних}\\ \text{змінних}}} a_{ij}x_j \le b_i - \sum_{\substack{\text{по змінних}\\ \text{часткового}\\ \text{рішення}}} a_{ij}x_j, \qquad i = 0,1,2,\ldots,m. \tag{2.42}$$

де кожній змінній x_j у частковому рішенні поставлено у відповідність визначене значення, що входить під знак суми в правій частині нерівності (2.42), а коефіцієнти в обмеженні при i=0 дорівнюють $a_{0j} = -c_j$ та $b_0 = -x_0^t - 1$. Тоді припустимого доповнення, якому відповідає значення цільової функції, що перевищує нижню оцінку, не існує, якщо

$$\sum_{\substack{\text{по всіх}\\ \text{вільних}\\ \text{змінних}}} \min(a_{ij},0) \le b_i - \sum_{\substack{\text{по змінних}\\ \text{часткового}\\ \text{рішення}}} a_{ij}x_j \quad \text{при будь-якому } i. \tag{2.43}$$

Член у лівій частині нерівності (2.43) є просто сумою всіх негативних коефіцієнтів при вільних змінних. Якщо ця сума більше правої частини нерівності, то, навіть вважаючи $x_j = 1$, для кожної вільної змінної, де $a_{ij} < 0$, неможливо виконати *i*-е обмеження (2.42).

Іншим важливим моментом є те, що при заданому частковому рішенні іноді удається визначити, яке значення повинне мати вільна змінна при будь-якому допустимому доповненні у випадку, коли значення цільової функції перевищує поточну нижню оцінку.

Зокрема, для будь-який вільної змінної x_k у випадку, коли

$$\sum_{\substack{\text{по всіх}\\ \text{вільних}\\ \text{змінних}}} \min(a_{ij},0) + |a_{ik}| > b_1 - \sum_{\substack{\text{по змінних}\\ \text{часткового}\\ \text{рішення}}} a_{ij}x_j \text{ при будь-якому } i, \tag{2.44}$$

$x_k = 0$, якщо $a_{ik} > 0$, та $x_k = 1$, якщо $a_{ik} < 0$. [Перевірки (2.42) і (2.44) можна також застосувати до складених обмежень, утвореним шляхом додавання позитивних комбінацій обмежень (2.39), а саме

$$\sum_{j=1}^{n}\left(\sum_{i=1}^{m} y_i a_{ij}\right) x_j \le \sum_{i=1}^{m} y_i b_i, \tag{2.45}$$

де $y_i > 0$. Щоб показати, як можна використовувати такі складені або заміщені обмеження, розглянемо умови $-x_1 - x_2 - x_3 \le -1$, $2x_2 - 2x_4 + 2x_5 \le 0$, $2x_3 + 2x_4 - 2x_5 \le 0$. При частковому рішенні $x_1 = 0$ застосування перевірки (2.43) до кожного обмеження не вказує на неприпустимість доповнення. Однак легко переконатися в тому, що застосування (2.43) до суми цих обмежень показує, що припустимого доповнення не існує. Аналогічно при $x_1 = 1$ застосування (2.45) до кожного обмеження не дає інформації про значення вільних змінних. Однак перевірка

(2.45) суми обмежень показує, що в будь-якому припустимому доповненні $x_2 = x_3 = 0$].

Перейдемо тепер до опису алгоритму. На ітерації t виконуються наступні кроки.

Крок 1. Припинити обчислення, якщо основний список порожній. У протилежному випадку вибрати задачу з основного списку і викреслити її з нього.

Крок 2. Якщо можна знайти вільні змінні, які повинні мати визначені значення при будь-якому припустимому доповненні, коли значення цільової функції перевищує x_0^t, то відповідним чином розширити обране часткове рішення. Якщо можна установити, що не існує припустимого доповнення, у якого значення цільової функції перевищує x_0^t, то прийняти $x_0^{t+1} = x_0^t$ і повернутися до кроку 1. У протилежному випадку перейти до кроку 3.

Крок 3. Якщо (розширене) часткове рішення є повним (тобто містить усі n змінних), зафіксувати його, прийняти x_0^{t+1} рівним відповідному значенню цільової функції та повернутися до кроку 1. У протилежному випадку перейти до кроку 4.

Крок 4. Вибрати будь-яку вільну змінну x_k, що не входить у (розширене) часткове рішення. Внести дві задачі в основний список. В одній з них прийняти $x_k = 0$ (у розширеному) частковому рішенні, в іншій прийняти $x_k = 1$. Прийняти $x_0^{t+1} = x_0^t$ та повернутися до кроку 1.

При припиненні виконання операцій алгоритму у випадку, коли зафіксоване припустиме рішення, що дає x_0^t, це рішення є оптимальним. У протилежному випадку припустимого рішення не існує. Алгоритм повинний забезпечувати збіжність за кінцеве число ітерацій.

Заключні зауваження. Варто звернути увагу на два основних розходження між викладеним методом і методом гілок і границь, розглянутим раніше. По-перше, в адитивному алгоритмі потрібне виконання тільки операцій додавання і віднімання (не потрібно ні множення, ні ділення). Вибір на кроках 1 та 4 може ґрунтуватися на інформації, отриманій з оптимального рішення задачі лінійного програмування (2.38), (2.39) і обмежень $0 \le x_j \le 1$.

Запитання для самоконтролю:

1. Що таке цілочисельні змінні?
2. Що таке розподіл капіталовкладень в задачах цілочисельного програмування?
3. Які є методи рішення цілочисельного програмування?
4. Для яких випадків використовується метод гілок і границь?
5. Сформулюйте задачу комівояжера?
6. У яких випадках використовується метод часткового (неявного) перебору?

ТЕМА 3. ДИНАМІЧНЕ ПРОГРАМУВАННЯ

3.1. Аналіз динамічних процесів

Як було показано в попередніх темах, моделі лінійного програмування в більшості випадків використовуються у промисловості для прийняття великомасштабних планових рішень в складних ситуаціях. У той же час моделі динамічного програмування зазвичай застосовуються при рішенні задач значно меншого масштабу. От деякі типові області застосування моделей динамічного програмування при прийнятті рішень:

– розробка правил управління запасами, які встановлюють момент поповнення запасів і розмір замовлення, що поповнює. Розробка принципів календарного планування виробництва і вирівнювання зайнятості в умовах змінного попиту на продукцію.

– визначення необхідного обсягу запасних частин, що гарантує ефективне використання дорогого устаткування.

– розподіл дефіцитних капітальних вкладень між можливими новими напрямками їхнього використання.

– вибір методів проведення рекламної кампанії, що знайомить покупця з продукцією фірми.

– систематизація методів пошуку коштовного виду ресурсів.

– складання календарних планів поточного і капітального ремонту складного устаткування.

– розробка довгострокових правил заміни основних фондів, що вибувають з експлуатації.

На алегоричному прикладі пояснимо деякі важливі ідеї динамічного програмування, а також побудуємо систему умовних позначок динамічних моделей.

Приклад. Жив колись містер М., який вирішив відправитися на пошуки щастя до Сан-Франциско. Тоді диліжанси були єдиним видом громадського транспорту для поїздки зі східних штатів, де проживав містер М., на Захід. У бюро подорожей йому показали карту Сполучених Штатів (рис. 3.1) з нанесеними на ній диліжансовими маршрутами, що обслуговувалися в той час. Кожен квадрат на карті зображує один зі штатів (станів); для зручності штати пронумеровані. Відмітимо, що, який би з варіантів шляху від штату 1 (Схід) до штату 10 (Захід) ми ні вибрали, він включає 4 диліжансових маршрути — або 4 „кроки ”.

Оскільки містеру М. було відомо, що подорож пов’язана із серйозною небезпекою для здоров’я і життя, перед від’їздом він вирішив застрахуватися. Ставка страхового платежу (іншими словами, вартість, що відповідає прийнятій стратегії вибору шляху) залежала від диліжансових маршрутів, що обираються, й вона була тим вище, чим небезпечніше був маршрут. Позначимо через c_{ij} вартість страхового поліса для переїзду зі штату *i* у штат *j*. Умовні чисельні значення c_{ij} проставлені на рис. 3.1.

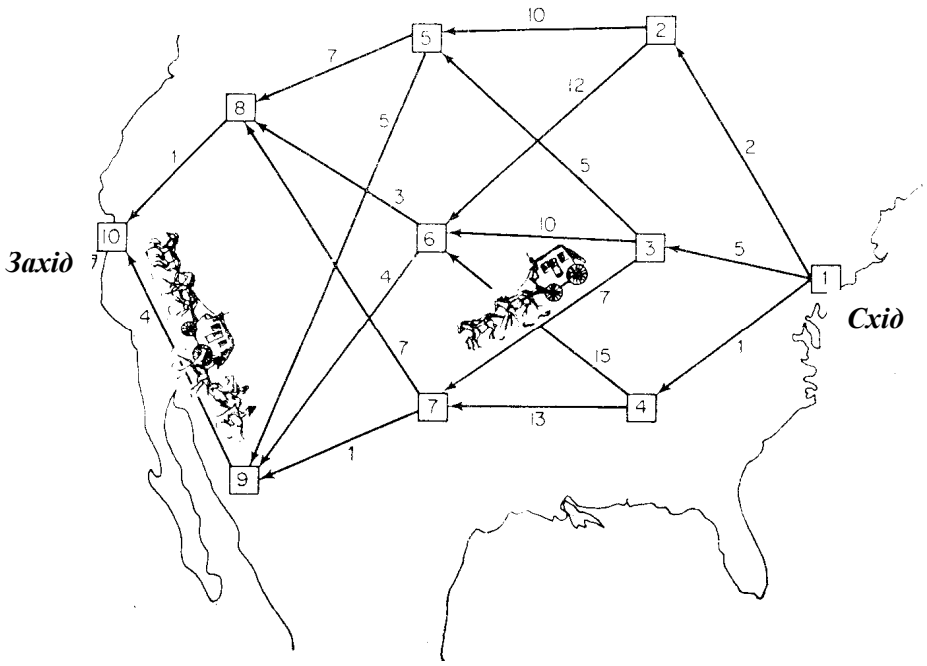

Рис. 3.1. Задача про диліжанси

Ціль містера М. — вибрати такий шлях від штату 1 до штату 10, для якого загальна вартість страхування є мінімальною (рекомендується відшукати оптимальний шлях самостійно). Містер М. почав аналіз проблеми з того, що визнав істотним сформульований нижче принцип.

Принцип оптимальності. Оптимальна стратегія має ту властивість, що, яким би не був шлях досягнення деякого штату (або стану), наступні рішення повинні належати оптимальній стратегії для частини шляху, що починається з цього стану.

Для того щоб врахувати сформульований принцип оптимальності та його обчислювальний зміст, зручним є використання наступних позначень:

$f_n(s)$ – вартість, що відповідає стратегії мінімальних витрат для шляху від штату (стану) s, якщо до кінцевого штату залишається n кроків;

$j_n(s)$ – рішення, що дозволяє досягти $f_n(s)$.

Всі ці букви та індекси необхідні і несуть важливі сенсові навантаження: f означає, що дане число є значенням цільової функції, s – що це значення залежить від стану системи, підрядковий індекс n несе динамічну інформацію про те, що зі стану s необхідно зробити ще n кроків; нарешті, символ j залежить як від кроку n, так і від стану s та відповідає деякому фіксованому шляху.

У підході містера М. можна знайти відому методичність. Коротко описуваний метод можна представити у вигляді так називаного динамічного рекурентного співвідношення:

$$f_n(s) = \min_{\text{по всіх s та j на мережі}} [c_{sj} + f_{n-1}(f)], \quad n = 1, 2, 3, 4. \tag{3.1}$$

Вираз (3.1) означає, що містер М. повинний обчислити всі можливі значення вартості, що відповідають різним стратегіям, підсумовуючи відповідну вартість для чергового кроку шляху (переїзду зі штату s у штат j), і вартість, що відповідає оптимальній стратегії вибору шляху, від штату j, з якого до кінця шляху залишається тільки (n-1) кроків. Обчислені суми варто порівняти між собою та

вибрати таке j, для якого значення цієї суми мінімально.

Приклад. Фірма повинна розробити календарну програму випуску деякого виду виробів на плановий період, що складається з N відрізків. Передбачається, що для кожного з цих відрізків існує точний прогноз попиту на продукцію, що випускається.

Час виготовлення партії виробів настільки малий, що їм можна нехтувати; відповідно продукція, що виготовляється протягом відрізка t, може бути використана для повного або часткового покриття попиту протягом цього відрізка. Для різних відрізків попит не однаковий; крім того, на економічні показники виробництва впливають розміри виготовлених партій. Тому фірмі нерідко буває вигідно виготовляти протягом деякого місяця продукцію в обсязі, що перевищує попит у межах цього відрізка, і зберігати надлишки, використовуючи їх для задоволення наступного попиту. Разом з тим, збереження виникаючих при цьому запасів пов'язано з визначеними витратами. У залежності від обставин витрати, обумовлені такими факторами, як відсотки на капітал, взятий у борг для створення запасів, орендна плата за складські приміщення, страхові внески і витрати по утриманню запасів. Ці витрати необхідно враховувати також при встановленні програми випуску.

Мета фірми – розробити таку програму, при якій загальна сума витрат на виробництво та на утримання запасів мінімізується за умови повного і своєчасного задоволення попиту на продукцію. Аналіз цієї задачі почнемо з перетворення якісного її опису в математичну модель.

Побудова моделі. Введемо змінні:

x_t – випуск продукції протягом відрізка t;

i_t – рівень запасів на кінець відрізка t.

Попит на продукцію для відрізка t позначимо D_t; передбачається, що величини D_t для всіх t відображаються невід'ємними цілими числами і що до початку планового періоду всі D_t відомі.

Припустимо також, що для кожного відрізка t витрати залежать від випуску продукції x_t, рівня запасів i_t на кінець відрізка t і, крім того, можливо, від значення t; позначимо витрати на відрізку t через $C_t(x_t, i_t)$. Тоді цільову функцію можна записати в наступному вигляді:

$$\sum c_t(x_t, i_t) \to \min. \qquad (3.2)$$

На значення змінних x_t та i_t накладено кілька обмежень.

По-перше, передбачається цілочисельність обсягів випуску:

$$x_t = 0, 1, 2, 3... \quad (t = 1, 2, ..., N). \qquad (3.3)$$

По-друге, передбачається, що для адміністрації фірми бажаним є нульовий рівень запасів на кінець відрізка N:

$$i_N = 0 \text{ (кінцевий запас дорівнює нулю)} \qquad (3.4)$$

По-третє, ставиться умова повного і своєчасного задоволення попиту в межах кожного місяця.

Якщо скористатися прийнятими умовними позначками, то це обмеження можна буде записати у наступному вигляді:

$$i_t = i_{t-1} + x_t - D_t$$

або, у зручнішому для нас вигляді:

$$i_{t-1} + x_t - i_t = D_t \quad (t = 1, 2, ..., N), \tag{3.5}$$

де i_0 – заданий рівень запасів на початок планового періоду.

Таким чином, потрібно, щоб

$$i_t = 0, 1, 2, 3... \quad (t = 1, 2, ..., N-1). \tag{3.6}$$

Для того щоб вирішити задачу при нелінійності кожної з величин $c_t(x_t, i_t)$, сформулюємо її в термінах динамічного програмування.

Нехай N=4. Складемо балансове рівняння (3.5) для t = 1, 2, 3, 4 (рис. 3.2).

	x_1	i_1	x_2	i_2	x_3	i_3	x_4	
Періоди	1	−1						$= D_1 - i_0$
		1	1	−1				$= D_2$
				1	1	−1		$= D_3$
						1	1	$= D_4$

Рис. 3.2. Матриця обмежень задачі управління запасами

Побудуємо п’яте рівняння, просумувавши чотири рівняння, а потім складемо систему з п’яти рівнянь, яка включає поряд з п’ятим чотири вихідних рівняння, що помножені на -1. побудована система адекватна мережі, що зображена на рис. 3.3:

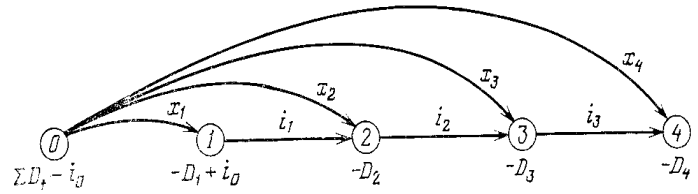

Рис. 3.3. Мережа, що відображає виробництво продукції та рух запасів

Тут кінцевим станом буде початок останнього відрізка планового періоду, а вихідним – початковий момент першого відрізка (попереду ще N відрізків).

При складанні математичної моделі зручно використовувати систему індексів, при якій підрядковий індекс „1” відповідає кінцевому, а „N” – початковому стану. Застосуємо наступні позначення:

d_n – попит на продукцію на відрізку n, що відстає від кінця планового періоду на n відрізків (включаючи розглянутий);

$c_n(x, j)$ – витрати на відрізку n, пов’язані з випуском x одиниць продукції та з утриманням запасів, рівень яких на закінчення відрізка дорівнює j одиниць.

У цій системі позначень $d_1 \equiv D_n$ та $d_n \equiv D_1$, а $c_1(x, j) \equiv c_N(x, j)$.

Що ж визначає стан системи на початку будь-якого відрізка? Можна вважати, що це рівень запасів на початок відрізка. Для ухвалення поточного рішення про обсяг випуску не потрібно знати, яким чином досягнутий

початковий рівень. З огляду на цю обставину, введемо наступні позначення:

$f_n(i)$ – вартість, що відповідають стратегії мінімальних витрат на n відрізків, що залишилися, при початковому рівні запасів і;

$x_n(i)$ – випуск, що забезпечує досягнення $f_n(i)$.

Згідно з (3.4), рівень запасів на кінець планового періоду дорівнює нулю. тому можна записати, що

$$f_0(0) = 0 \quad (n = 0). \tag{3.7}$$

Потім перейдемо до n=1. Початковий рівень запасів і може визначатися будь-яким ненегативним цілим числом, не більшим, ніж d_1; поза залежністю від значення і для повного задоволення потреби в межах останнього відрізка обсяг випуску повинний дорівнювати $(d_1 - i)$.

Отже,

$$f_1(i) = c_1(d_1 - i), \quad i = 0,1,...,d_1. \tag{3.8}$$

Перейдемо до n = 2. Відмітимо, що якщо початковий рівень запасів дорівнює i, а обсяг випуску – x, то загальні витрати для двох місяців складають:

$$c_2(x, i + x - d_2) + f_1(i + x - d_2)$$

причому передбачається, що обрана стратегія для n=1 була оптимальною. Відмітимо, що величина $(i + x - d_2)$ є рівнем запасів на закінчення відрізка 2. Величина і може приймати будь-які ненегативні цілочисельні значення, що не перевищують (d_1 + d_2); питання: поясніть, чому? При заданому і цілочисельне значення х має бути не менше, ніж (d_2 – i), що забезпечує повне задоволення потреби на відрізку 2, але не більше, ніж (d_1 + d_2 – i) тому, що кінцевий запас дорівнює нулю. Оптимальному обсягу випуску відповідає таке значення х, при якому мінімізується зазначена вище сума. Виконаний нами аналіз ситуації для n=2 можна виразити наступним загальним виразом:

$$f_2(i) = \min_x [c_2(x, i + x - d_2) + f_1(i + x - d_2)],$$

де i=0, 1, ... , d_1 + d_2, при чому для пошуку мінімуму перебираються всі ненегативні цілі значення х, що знаходяться у межах

$$d_2 - i \le x \le d_1 + d_2 - i.$$

Як і в задачі про диліжанси, значення $f_3(i)$ можна обчислити, якщо відомі значення $f_2(i)$ і т.д. Зрештою, у даній задачі можна обчислити $f_n(i_0)$, де i_0 – рівень запасів на початок планового періоду. Загальне рекурентне співвідношення записується в наступному вигляді:

$$f_n(i) = \min_x [c_n(x, i + x - d_n) + f_{n-1}(i + x - d_n)], \; n = 1, 2, ..., N, \tag{3.9}$$

де i = 0, 1, . . ., d_1 + . . . + d_n, причому для пошуку мінімуму перебираються всі ненегативні цілі значення х, що знаходяться у межах:

$$d_n - i \le x \le d_1 + d_2 + ... + d_n - i.$$

Відмітимо, що, оскільки початковий рівень запасів і розглядається як змінна величина, яка цілком характеризує стан системи, єдиною незалежною управляючою перемінною в рекурентному співвідношенні (3.9) є х, тому що

рівень запасів на закінчення відрізка дорівнює ($i + x - d_n$). Відмітимо також, що, оскільки $f_0(0)$ та $f_1(i)$ легко обчислюються по формулах (3.7) і (3.8), можна за чергою обчислити значення $f_2(0)$, $f_2(1)$, ... , $f_2(d_1)$, а потім $f_3(0)$, $f_3(1)$, ... , $f_3(d_1 + d_2)$. Послідовно переходячи до все більших значень n, дійдемо до обчислення $f_{N-1}(0)$, $f_{N-1}(1)$,..., $f_{N-1}(d_1 + d_2 + \ldots + d_{N-1})$ і, нарешті, до $f_n(i_0)$.

Для спрощення аналізу будемо вважати, що попит і функція витрат однакові для усіх відрізків планового періоду. Для конкретності приймемо

$$D_t = 3 \text{ одиницям (попит постійний у часі).} \qquad (3.10)$$

Припустимо також, що витрати дорівнюють сумі двох елементів; перший з них відноситься до виробництва, а другий визначається вартістю утримання запасів, що є лінійною функцією обсягу запасів. Таким чином, для усіх відрізків

$$C_t(x_t, i_t) = C_t(x_t) + hi_t, \qquad (3.11)$$

$$\text{де } C(0) = 0,\ C(1) = 15,\ C(2) = 17,\ C(3) = 19,\ C(4) = 21,\ C(5) = 23, \qquad (3.12)$$

$$h = 1. \qquad (3.13)$$

У свою чергу, виробничі витрати можна розглядати як суму умовно-постійних витрат на операції по переналагодженню (ці витрати дорівнюють 13 умовним одиницям) і пропорційних витрат (вони дорівнюють 2 умовним одиницям на кожну одиницю продукції). Оскільки h = 1, витрати на утримання запасів чисельно дорівнюють рівневі запасів на закінчення відрізка.

Далі, виробничі потужності і складські площі фірми обмежені; це накладає на задачу додаткове ускладнення. Приймемо, що випуск протягом одного відрізка не може перевищити 5 одиниць, а рівень запасів на кінець відрізка – 4 одиниці. Іншими словами, для усіх відрізків

$$\left.\begin{aligned} x_t &= 0,1,\ldots,5 \\ i_t &= 0,1,\ldots,4 \end{aligned}\right\} \qquad (3.14)$$

Формулювання задачі динамічного програмування. При наявності приведених вище даних про умови діяльності фірми можна скласти динамічне рекурентне співвідношення, що відображає специфіку задачі. Нагадаємо, що використовуються наступні позначення:

$f_n(i)$ – мінімальні витрати протягом n останніх відрізків планового періоду при початковому рівні запасів i;

$x_n(i)$ – випуск, що дозволяє досягти $f_n(i)$.

Для n = 1

$$\left.\begin{aligned} f_1(i) &= C(3-i) \\ x_1(i) &= 3-i \end{aligned}\right\}, \quad i = 0, 1, 2, 3, \qquad (3.15)$$

оскільки рівень запасів на кінець планового періоду дорівнює нулю. У загальному вигляді рекурентне співвідношення можна записати в такий спосіб:

$$f_n(i) = \min_x [c_x + 1(i + x - 3) + f_{n-1}(i + x - 3)],\ n = 2, 3,\ldots, \qquad (3.16)$$

де i = 0, 1, 2, 3, 4 і для пошуку мінімуму перебираються всі ненегативні цілі значення x, що знаходяться у межах $3 - i \le x \le \min(5, 7 - i)$. Обмеженість виробничих потужностей, відображувана першим з умов (3.14), не дозволяє x перевищити 5,

а обмеженість рівня запасів на закінчення відрізка, відображувана другою частиною (3.14), не дозволяє x перевищити (7 – i).

Для того щоб аналіз був змістовним, необхідно мати у розпорядженні всі значення функцій $f_n(i)$; у зв'язку з цим їхнє обчислення є нашою черговою задачею. Для кожного кроку n побудована одна таблиця; у ній передбачене по одному рядку для кожного можливого значення початкового рівня запасів i та по одному стовпцю – для кожного можливого значення випуску x. Оскільки попит на продукцію у межах кожного відрізка повинний бути цілком задоволений, а рівень запасів на закінчення відрізка не може перевищити 4 одиниці, деякі клітинки у таблицях „заборонені " – вони відповідають неприпустимим сполученням i та x. Кожне з проставлених у таблиці чисел являє собою суму витрат для розглянутого відрізка n та оптимальних витрат для усіх (n–1) наступних відрізків. У двох правих стовпцях таблиці проставлені: мінімальна по рядку сума [у стовпці $f_n(i)$] та відповідний їй оптимальний випуск у стовпці x(i)].

Значення $f_1(i)$, обчислені за формулою (3.15), приведені в таблиці рис. 3.4, а значення функції $f_2(i)$ – у таблиці рис. 3.5. Розглянемо структуру останньої таблиці більш докладно. У ній є 5 рядків, по одній для кожного допустимого значення i. Клітинки, що відповідають деяким сполученням i та x – „заборонені ". Так, якщо i = 1, то попит удасться задовольнити тільки за умови x ≥ 2.

$f_1(i) = C(3-i)$

	i	$x_1(i)$	$f_1(i)$
Початковий запас	0	3	19
	1	2	17
	2	1	15
	3	0	0

Рис. 3.4. Задача фірми (n = 1)

Якщо i = 4, то x ≤ 2, інакше порушиться умова нульового рівня запасів на кінець планового періоду. Перше з доданків у кожній клітинці – значення C(x), обчислене за формулою (3.12). Другий доданок – витрати на утримання запасів, що дорівнюють рівню запасів на закінчення відрізка, помноженому на h = 1. Так, наприклад, при i = 3 та x = 0 рівень запасів на закінчення відрізка також дорівнює нулю; через це дорівнює нулю також другий доданок у відповідній клітинці.

При i = 3 та x = 1 рівень запасів на кінець місяця дорівнює 1; у відповідній клітинці другий доданок також дорівнює 1. Аналогічним чином значення других доданків обчислюються й для інших клітинок третього рядка (i = 3). Третій доданок – це значення $f_1(i + x - 3)$, раніше обчислене та приведене у таблиці рис. 3.4.

Для кожного фіксованого i значення функції $f_2(i)$ являє собою мінімальну з усіх сум у „клітинках " даного рядка, а $x_2(i)$ – відповідний випуск. Так, при i = 1 і n = 2 оптимальний випуск дорівнює 5 одиницям; він дозволяє за два місяці досягти витрат, що дорівнюють 26. Будь-яке інше значення x обумовлює більш високі витрати.

Розрахунок значень $f_3(i)$ приведений у таблиці рис. 3.6. Тут перший доданок дорівнює [C(x) + 1(i + x – 3)], а другий доданок є значення $f_2(i + x – 3)$, взяте з таблиці рис. 3.5. Інші значення $f_n(i)$ для n = 4, 5, 6 подані у зведеній таблиці рис. 3.7. Вам варто перевірити, наскільки освоєні рекурентні обчислювальні операції методу динамічного програмування, побудувавши повністю розрахункову таблицю для $f_4(i)$, аналогічну таблиці рис. 3.6, та порівнявши отримані результати з тими даними, що приведені у таблиці рис. 3.7. Помітимо, що для n = 4 оптимальними є два значення випуску – 3 одиниці та 4 одиниці.

$C(x)+1(i+x-3)+f_2(i+x-3)$

Випуск

Початковий запас i \ x	0	1	2	3	4	5	$x_2(i)$	$f_2(i)$
0				19 + 0 + 19	21 + 1 + 17	23 + 2 + 15	3	38
1			17 + 0 + 19	19 + 1 + 17	21 + 2 + 15	23 + 3 + 0	5	26
2		15 + 0 + 19	17 + 1 + 17	19 + 2 + 15	21 + 3 + 0		4	24
3	0 + 0 + 19	15 + 1 + 17	17 + 2 + 15	19 + 3 + 0			0	19
4	0 + 1 + 17	15 + 2 + 15	17 + 3 + 0				0	18

Рис. 3.5. Задача фірми (n = 2)

$[C(x)+1(i+x-3)]+f_2(i+x-3)$

Випуск

Початковий запас i \ x	0	1	2	3	4	5	$x_3(i)$	$f_3(i)$
0				19 + 38	22 + 26	25 + 24	4	48
1			17 + 38	20 + 26	23 + 24	26 + 19	5	45
2		15 + 38	18 + 26	21 + 24	24 + 19	27 + 18	4	43
3	0 + 38	16 + 26	19 + 24	22 + 19	25 + 18		0	38
4	1 + 26	17 + 24	20 + 19	23 + 18			0	27

Рис. 3.6. Задача фірми (n = 3)

Початковий запас	n=1		n=2		n=3		n=4		n=5		n=6	
	$x_1(i)$	$f_1(i)$	$x_2(i)$	$f_2(i)$	$x_3(i)$	$f_3(i)$	$x_4(i)$	$f_4(i)$	$x_5(i)$	$f_5(i)$	$x_6(i)$	$f_6(i)$
0	3	19	3	38	4	48	3,4	67	5	79	4	96
1	2	17	5	26	5	45	5	64	5	74	5	93
2	1	15	4	24	4	43	5	54	4	72	4	91
3	0	0	0	19	0	38	0	48	0	67	0	79
4			0	18	0	27	0	46	0	65	0	75

Рис. 3.7. Задача фірми

Тривалість планового періоду. Для визначеності будемо вважати, що плановий період починається в січні. Нас цікавить зміна оптимальних місячних обсягів випуску при збільшенні числа місяців N у плановому періоді і, зокрема, зміну січневого випуску. Результати аналізу, заснованого на даних рис. 3.7, приведені в таблиці рис. 3.8; при цьому передбачається, що вихідний рівень запасів на початок січня дорівнює нулю.

Тривалість планового періоду, N	Січень	Лютий	Березень	Квітень	Травень	Червень	Загальна сума витрат	Середньомісячні витрат
1	3						19	19
2	3	3					38	19
3	4	5	0				48	16
4	3 4	4 5	5 0	0 3			67	$16\frac{3}{4}$
5	5	5	0	5	0		79	$15\frac{4}{5}$
6	4	5	0	4	5	0	96	16

Рис. 3.8. Програма випуску продукції фірмою $(t_0 = 0)$

Таблиця рис. 3.8 побудована у такий спосіб. Січневий обсяг виробництва (3 одиниці) для N = 1 взятий з першого рядка таблиці рис. 3.7 при n, що дорівнює 1. Січневий обсяг виробництва (3 одиниці) для N = 2 взятий з того ж рядка при n= 2 і т.д. У випадку N = 6 січневий випуск дорівнює 4 одиницям, а рівень запасів на початок лютого –1 одиниці (він дорівнює i + x – d = 0 + 4 – 3). Отже, лютневий випуск (5 одиниць) знайдемо з другого рядка таблиці рис. 3.7 при n = 5, оскільки рівень запасів на початок місяця тепер дорівнює 1 одиниці. У свою чергу це означає, що рівень запасів па початок березня складе 3 одиниці

(i + x – d = 1 + 5 – 3 = 3); тому випуск продукції в березні буде нульовим, як це показано в таблиці рис. 3.7 для i = 3 та n = 4. На основі аналогічних міркувань визначимо, що виробництво у квітні (n = 3) повинно дорівнювати 4 одиницям, через те, що рівень запасів на початок квітня – нуль (i + x – d = 3 + 0 – 3 = 0). При такому випуску рівень запасів на начало травня буде дорівнювати 0 + 4 – 3 = 1, так що травневий випуск (n = 2) складе 5 одиниць. Отже, у червні оптимальний нульовий випуск, оскільки рівень запасів на початок місяця (n = 1) дорівнює 1 + 5 – 3 = 3 одиницям. Отже може на переконатися в тому, що мінімальна загальна сума витрат для N = 6 складе: (21 + 1) + (23 + 3) + +(0 + 0) + (21 + 1) +(23 + 3) +(0 + 0) = 96; ця сума й проставлена в клітинці таблиці рис. 3.7, відповідної $f_6(0)$.

Відомо, що математична модель нерідко є досить загальною для того, щоб охопити безліч різних реальних ситуацій. Тому змістовна оцінка моделі, приведеної в цьому розділі, повинна, в першу чергу, ґрунтуватися на аналізі системи вихідних припущень, а не конкретних умов діяльності фірми. Перелічимо найважливіші з цих припущень.

1) Прогноз попиту є точним. Хоча фірмі рідко вдається зовсім точно визначити попит на кілька місяців уперед, розміри помилки часто досить мала і детермінована модель дає гарну апроксимацію дійсності. (У подібних обставинах звичайно застосовується ковзне планування, однак іноді перегляд програми при цьому здійснюється не щомісяця, а раз на кілька місяців.) Якщо ж помилки прогнозування істотні, необхідним є перехід до моделей такого типу.

2) Тривалість виготовлення продукції дуже мала. У реальних умовах робиться інше припущення, відповідно до якого можна визначати тривалість періоду виробництва з дуже малою помилкою. Так, нехай виготовлення партії виробів триває два тижня. Якщо виробнича програма побудована з використанням приведених у цьому розділі рекурентних співвідношень, причому відповідно до цієї програми лютневий попит задовольняється лютневим випуском, то, насправді, запуск партії необхідно здійснити па два тижні раніше, тобто в другій половині січня.

Іншим аспектом даного припущення є можливість визначення тривалості виготовлення партії виробів поза зв’язком з виготовленням інших замовлень. Якщо кілька різних видів виробів обробляються на тому самому устаткуванні, виробнича потужність якого обмежена, то сукупність програм випуску, кожна з яких отримана за допомогою відособленої моделі динамічного програмування, може виявитися несумісною.

Моделі, приведені в цьому розділі, часто корисні в тих випадках, коли вироби замовляють у зовнішнього постачальника, який має необхідні запаси. Тоді затримка у часі пов’язана з інтервалом постачання, а витрати включають не витрати виробництва, а витрати на закупівлю виробів.

3) Витрати по кожному відрізку залежать від поточного випуску та від рівня запасів на кінець відрізка; попит на кожному відрізку повністю і вчасно задовольняється. Ці два припущення легко можна узагальнити на значно більш

широке коло ситуацій. Методи такого узагальнення розглядаються нижче.

Не вводячи істотно нових обмежень, можна переписати рекурентне співвідношення динамічного програмування у наступному вигляді:

$$f_n(i) = \min[c_n(x, i, i + x - d_n) + f_{n-1}(i + x - d_n). \qquad (3.17)$$

При такому записі співвідношення можна поширити на ситуацію, при якій витрати на утримання запасів обчислюються виходячи із середнього рівня

$$\frac{i + (i + x - d_n)}{2}. \qquad (3.18)$$

Далі можна ввести допущення, відповідно до якого своєчасно незадоволений попит повністю зберігається і зміщується на наступні відрізки. При такому допущенні „балансове рівняння ”

$$i_t = i_{t-1} + x_t - D_t \qquad (3.19)$$

як і раніше залишається в силі, але величина i_t може приймати також негативні значення, що характеризують незадоволений попит, що накопичується. Можливо й інше допущення, при якому вчасно незадоволений попит повністю втрачається; у цьому випадку:

$$i_t = \max(i_{t-1} + x_t - D_t,\ 0). \qquad (3.20)$$

Тоді

$$f_n(i) = \min_{x=0,1,\ldots} \{c_n(x, i, i + x - d_n) + f_{n-1}[\max(i + x - d_n,\ 0)]\}. \qquad (3.21)$$

Як у виразі (3.19), так і у виразі (3.20) цільова функція витрат $c_n(x, i, i + x - d_n)$ повинна включати штраф за затримку задоволення попиту або за повну його втрату. Такою функцією може бути функція (3.22):

$$c_n(x, i, i + x - d_n) = c(x) + h[\max(i + x - d_n,\ 0)] - p[\min(i + x - d_n)], \qquad (3.22)$$

де $p(p > 0)$ являє собою штраф за одиницю попиту, не задоволеного протягом того відрізка, коли цей попит виник. Очевидно, у випадку допущення втрати попиту скорочення виторгу від реалізації буде враховуватися у значенні p.

3.2. Модель розподілу зусиль

Розглянемо нижченаведений гіпотетичний, але, проте повчальний приклад так називаної моделі розподілу зусиль. Власник торговельної фірми повинний розподілити наявний у нього тижневий запас у кількості N яєць по s магазинах. З досвіду відомо, що якщо направити y_i, яєць у магазин j, то прибуток становитиме $R_j(y_j)$. Власник фірми вважає, що для максимізації загального прибутку не слід направляти всі яйця в один магазин і прагнути знайти оптимальний розподіл наявних у наявності яєць по магазинах.

Цю задачу можна сформулювати у такий спосіб:

максимізувати $$\sum_{j=1}^{s} R_j(y_j), \qquad (3.23)$$

при обмеженнях

$$\sum_{j=1}^{s} y_j = N \text{ (наявний запас яєць)}, \qquad (3.24)$$

y_j = 0, 1, 2, ... для будь-якого j (враховуючи тільки цілі яйця). (3.25)

Для перетворення постановки задачі (3.23) – (3.25) у задачу динамічного програмування приймемо:

$g_j(n)$ – прибуток при оптимальному розподілі n яєць по

магазинах 1, 2, ... , j, (3.26)

$y_j(n)$ – кількість яєць, що спрямовані у магазин j, які дають прибуток $g_j(n)$.

При заданих числових значеннях функцій прибутку $R_j(y_j)$ і відомому значенні N обчислення при рішенні задачі починаються з визначення значень $g_1(0)$, $g_1(1)$,…, $g_2(N)$ при j = 1. Потім знаходяться значення $g_2(0)$, $g_2(1)$, ... , $g_2(N)$. Обчислення продовжуються по тій же схемі при послідовному зростанні значень j поки, зрештою, не визначаються значення $g_s(N)$. Оптимальний розподіл у принципі знаходиться шляхом здійснюваного у зворотному порядку вибору значень y_j, що дають у сукупності $g_s(N)$.

Модель загального виду. Приклад торговельної фірми відноситься до часткового випадку задачі розподілу зусиль. Цей приклад має частковий характер у силу того, що єдине обмеження (3.24) є лінійним. Точно такий же динамічний підхід також є справедливим і у випадку, коли єдине обмеження нелінійне. Припустимо для визначеності, що модель описується наступними співвідношеннями:

максимізувати $$\sum_{j=i}^{s} R_j(y_j), \qquad (3.27)$$

при обмеженнях

$$\sum_{j=i}^{s} H_j(y_j) = N, \qquad (3.28)$$

y_j = 0, 1, 2, ... при будь-якому j. (3.29)

Припустимо, що кожна функція H_j (y_j) є не спадною, що приймає цілочисельні значення при будь-якому y_j = 0, 1, 2,… й задовольняючій умові $H_j(0) = 0$. Для спрощення міркувань приймемо, що $H_1(y_2) = y_1$, унаслідок чого допустиме рішення існує при будь-якому значенні N. На кожну величину y_j можна також накласти обмеження зверху, як це буде показано у числовому прикладі, що приводиться нижче.

Рекурентне співвідношення динамічного програмування, що відповідає задачі (3.27) – (3.29), має такий вигляд:

$$g_j(n) = \max_{y_j}\{R_1(y_j) + g_{j-1}[n - H_j(y_i)]\}, \quad j = 1, 2, ..., s, \qquad (3.30)$$

$$g_0(n) \equiv 0, \ j = 0, \qquad (3.31)$$

де n = 0, 1, ..., N, а максимум береться тільки по ненегативних цілочисельних значеннях

y_j, що задовольняє умові $H_j(y_j) \le n$. Відшукується значення $g_s(N)$. Для виконання обчислень потрібно визначити за виразом (3.30) значення кожної функції $g_j(n)$ при $n = 0, 1, \ldots, N$, починаючи з $j = 1$ та закінчуючи $j = s$.

Числовий приклад. Проілюструємо обчислювальну схему рішення задачі про розподіл зусиль, задавши числовими даними для моделі (3.27) – (3.29). Приймемо $s = 3$ та $N = 8$. Значення функцій $R_j(y_j)$ та $H_j(y_j)$ приведені в таблиці рис. 3.9. Відмітимо, що

$$R_1(y_1) = 2y_1, \quad R_2(y_2) = 3y_2,$$
$$H_1(y_1) = y_1, \quad H_2(y_2) = 2y_2, \quad H_3(y_3) = 3y_3, \qquad (3.32)$$

Як видно, це лінійні функції, але функція $R_3(y_3)$ нелінійна.

Крім того, накладемо обмеження зверху на всі

$$y_j \le 2 \quad \text{при будь-якому } j. \qquad (3.33)$$

(У цій задачі можна було б також прийняти обмеження $y_3 \le 1$, оскільки при $y_3 = 2$ значення цільової функції не зростає).

y_j	$R_1(y_1)$	$H_1(y_1)$	$R_2(y_2)$	$H_2(y_2)$	$R_3(y_3)$	$H_3(y_3)$
0	0	0	0	0	0	0
1	2	1	3	2	4,5	3
2	4	2	6	4	4,5	6

Рис. 3.9. Приклад задачі про розподіл зусиль

Значення $g_j(n)$ та відповідні їм оптимальні рішення $y_j(n)$ зведені в таблицю рис. 3.10. Проаналізуємо, яким чином отримані ці величини.

Обчислення значень $y_1(n)$ та $g_1(n)$ тривіальні. Так, наприклад, якщо $n = 2$, то з таблиці рис. 3.9 легко побачити, що $H_1(2) = 2$, звідки $y_1(2) = 2$ і відповідно $g(2) = R_1(2) = 4$.

	j=1		j=2		j=3	
n	$y_1(n)$	$g_1(n)$	$y_2(n)$	$g_2(n)$	$y_3(n)$	$g_3(n)$
0	0	0	0	0	0	0
1	1	2	0	2	0	2
2	2	4	0	4	0	4
3			1	5	0	5
4			1	7	0	7
5			2	8	1	8,5
6			2	10	0	10
7					1	11,5
8					1	12,5

Рис. 3.10. Оптимальні стратегії для приклада задачі про розподіл зусиль (N = 8)

Аналогічно обчислення значень $y_1(n)$ та $g_2(n)$ при $n = 0, 1$ майже настільки ж тривіальні, оскільки при цих значеннях n обмеження $H_2(y_2) \le n$ може

задовольнятися тільки при $y_2 = 0$. Отже, маємо

$$g_2(0) = 0,\ y_2(0) = 0, \tag{3.34}$$
$$g_2(1) = R_2(0)+g_1(1) = 0 + 2 = 2,$$

та

$$y_2(1) = 0. \tag{3.35}$$

Більш типовими за своїм характером є обчислення значень $y_2(2)$ та $g_2(2)$, оскільки при n = 2 обидва значення $y_2 = 2$ та $y_2 = 1$, як видно з таблиці рис. 3.9, є припустимими. Таким чином, з (3.30) одержуємо

$$g_2(2) = \max\,[R_2(0) + g_1(2),\ R_2(1) + g_1(0)] =$$
$$= \max\,(0 + 4,\ 3 + 0)= 4 \text{ та } y_2(2) = 0. \tag{3.36}$$

Для перевірки засвоєння обчислювальної схеми читачеві рекомендується самостійно визначити значення інших елементів таблиці рис. 3.10. (Відзначимо, що при збільшенні n від 0 до 8 оптимальні значення y_3 коливаються).

Оптимальна стратегія при N = 8 знаходиться у такий спосіб. Починаючи з j = 3, маємо $y_3(8) = 1$. Отже, для визначення оптимального значення y_2 потрібно обчислити $n = 8 - 3y_3(8)= 5$. Зробивши це, одержуємо $y_2(5) = 2$. Тоді нове значення $n =5 - 2y_2(5) = 1$ і, отже, $y_1(1) = 1$. В результаті оптимальний розподіл визначається наступними значеннями змінних: $y_1 = 1$, $y_2 = 2$, $y_3 = 1$. Відповідне цьому розподілові оптимальне значення цільової функції $g_3(8) = 12{,}5$.

3.3. Модель заміни обладнання

Раніше було показано, як можна використовувати ациклічну мережу для опису задачі визначення стратегії заміни обладнання, що мінімізує витрати у рамках планового періоду з (N – 1) відрізків.

Величина c_{ij} являла собою повні експлуатаційні витрати, починаючи з відрізка і та закінчуючи відрізком j, на якому це обладнання замінюється. Хоча наведений раніше приклад відноситься до випадку, коли обладнання береться в оренду, модель цілком придатна також у ситуації, коли фірма закуповує обладнання. В останньому випадку величина c_{ij} є сумою покупної ціни та експлуатаційних витрат нового обладнання, за винятком залишкової вартості цього обладнання на початок відрізка j.

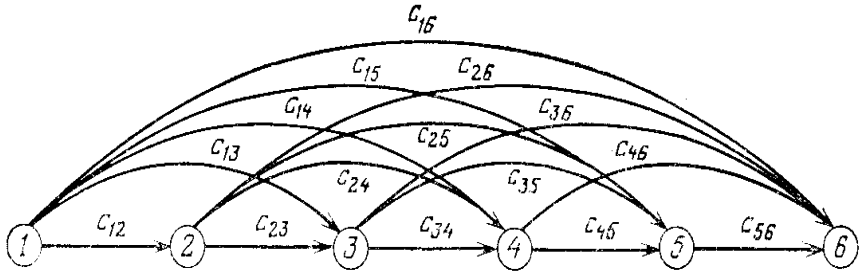

Рис. 3.11. Сітка заміни обладнання

Задача відноситься до категорії задач про відновлення в тому сенсі, що визначення стратегії заміни рівносильне складанню списку відрізків, у яких виробляється закупівля нового обладнання. Приймемо наступні позначення:

f_n – стратегія заміни, що мінімізує витрати на відрізках

n, n + 1, ... , N – 1, у припущенні, що нове обладнання купується на початку відрізка n. (3.37)

Тоді відшукується значення f_1 та рекурентне співвідношення, що відповідає цій умові, має вигляд

$$f_n = \min_{k=n+1,\dots,N}[c_{nk} + f_k], \quad n = N-1,\ N-2,\ \dots,\ 1, \tag{3.38}$$

$$f_n = 0. \tag{3.39}$$

Корисно описати словами вираження (3.38). Відповідна ациклічна мережа при N – 1 = 5 приведена на рис. 3.11.

До цієї задачі можна підійти з іншої точки зору. Припустимо, що витрати, що відповідають деякій стратегії заміни, включають дві складові:

p_{in} – чисті вартості заміни обладнання віку i на відрізку n новим обладнанням за умови продажу на цьому відрізку старого обладнання за його залишковою вартістю;

k_{nt} – вартість експлуатації обладнання на відрізку n, вік якого наприкінці цього відрізка дорівнює t.

Приймемо, що

$f_n(i)$ – стратегія, що мінімізує витрати на відрізках

n, n + 1, ..., N – 1, за умови, що на початку відрізка n вік обладнання дорівнює i рокам. (3.41)

Відзначимо, що зміст, що дається індексу в (3.41), у даному прикладі відрізняється від змісту, що надавався йому в попередніх прикладах. Справді, тут n відноситься до відрізка n і всім наступним відрізкам до кінця планового періоду.

Якщо оптимальне рішення зводиться до збереження устаткування і попадає на відрізок n, то

$$f_n(i) = k_{n\ \ i=1} + f_{n+1}(i+1), \tag{3.42}$$

але якщо оптимальне рішення зводиться до його заміни, то

$$f_n(i) = p_{in} + k_{n1} + f_{n+1}(1). \tag{3.43}$$

Таким чином, маємо

$$f_n = \min[k_{n\ \ i+1} + f_{n+1}(i+1),\ \ p_{in} + k_{n1} + f_{n+i}(1)], \quad n = 1,\ 2,\ \ N-1, \tag{3.44}$$

де

$$f_N(i) = 0. \tag{3.45}$$

Ми шукаємо значення $f_1(i_0)$, де i_0 – вік обладнання на початок планового періоду. Якщо в цей час розглянута одиниця обладнання відсутній, то величина $p_{i_0 1}$ являє собою просто покупну ціну нового обладнання і рішення зберегти обладнання при n = 1 не має сенсу.

В основі методу динамічного програмування, використовуваного для рішення оптимізаційних задач з багатьма обмеженнями і великим числом змінних,

лежить поділ задачі на послідовність кроків, на кожному з яких вирішується оптимізаційна задача меншої розмірності. На відміну від цього методу в більшості методів лінійного та нелінійного програмування починаються спроби рішення таких задач шляхом одночасного обліку всіх обмежень.

У методі динамічного програмування задача зводиться до наступного вигляду:

I) Керовані змінні та відповідні обмеження групуються по кроках, і багатокроковий процес прийняття рішень досліджується у визначеній послідовності.

II) Єдина інформація про попередні кроки, що використовується для вибору оптимальних значень змінних на розглянутому кроці, знаходиться у так називаній змінній стану, що представляє собою іноді n-мірний вектор.

III) Розглянуте рішення, прийняте при заданому поточному стані системи, впливає на стан системи на наступному кроці.

IV) Оптимальність поточного рішення оцінюється в термінах прогнозованого економічного ефекту для розглянутого кроку і всіх наступних кроків.

На відміну від моделі лінійного програмування, що відноситься до спеціального класу математичних моделей, які дозволяють одержати рішення різними методами (серед цих методів самим універсальним є симплексний), динамічне програмування є особливим аналітичним методом, застосованим до всіляких математичних моделей. Існує канонічна форма запису задачі у термінах динамічного програмування, що наочно ілюструє перераховані у пп. I) – IV) структурні властивості.

Позначимо символом s деякий стан системи і символом S_n – множину всіх можливих станів на кроці n. Далі приймемо, що d_n позначає рішення, обране на кроці n, а $D_n(s)$ – всі припустимі значення d_n за умови, що система знаходиться в стані s. Нарешті, нехай функція $R_n(s, d_n)$ позначає безпосередній економічний ефект, що досягається у результаті вибору рішення d_n при заданому стані системи s, а $T_n(s, d_n)$ – перетворений стан системи на кроці n – 1. Тоді рекурентне співвідношення динамічного програмування (РДП) запишеться в наступному вигляді:

$$f_n(s) = \underset{d_n \text{ у } D_n(s)}{\text{optimum}} \{R_n(s_1 d_n) + f_{n-1}[T_n(s_1 d_n)]\} \quad \text{при будь – якому } s \text{ у } S_n,$$

де „optimum” означає максимум або мінімум у залежності від постановки задачі.

Потрібно враховувати важливу особливість РДП. Для ухвалення оптимального рішення на кроці n не потрібно значення керованих змінних, що зумовлюють ефект $f_{n-1}[T_п (s, d_n)]$. Такий характер рекурентних співвідношень означає, що на кожному кроці оптимізація проводиться лише стосовно обмеженого числа керованих змінних і в наступних обчисленнях використовуються тільки значення цільової функції. Саме ця простота обчислювальної схеми обумовлює привабливість методу динамічного програмування. Якщо користуватися поняттями лінійного програмування, то обчислення, виконувані при використанні методу динамічного програмування, можна розглядати як послідовне визначення значень двоїстих змінних. Ця аналогія для прикладів, викладених у даному розділі, є дуже точною в силу

причин, що далі роз'ясняються.

В усіх приведених прикладах керовані змінні, а також змінні стану та кроку приймали тільки цілочисельні значення. (Задачі такого роду називають задачами дискретного програмування.) Крім того, процес прийняття рішень відносився до кінцевого проміжку часу або кінцевого числа кроків. Нарешті, на результати і переходи з одного стану в інший не робили ніякого впливу випадкові фактори.

Запитання для самоконтролю:

1. Які типові області застосування моделей динамічного програмування?
2. В чому полягає принцип оптимальності?
3. Сформулюйте задачу динамічного програмування.
4. Які є моделі розподілу зусиль?
5. Для яких процесів використовується модель заміни обладнання?

ТЕМА 4. ТЕОРІЯ МАСОВОГО ОБСЛУГОВУВАННЯ

Теорія масового обслуговування (ТМО) займається аналізом таких процесів і систем, в яких з різних причин виникають черги на обслуговування. Такими причинами можуть бути:

- кількість вимог на обслуговування в одиницю часу, що змінюється випадково;

- час обслуговування, що також є випадковою величиною.

Системи, в яких формуються потоки вимог на обслуговування (вхідні потоки), здійснюється їх послідовне обслуговування, в результаті чого формуються потоки обслужених вимог (потоки обслуговувань або вихідні потоки) називають системами масового обслуговування (СМО).

4.1. Функції та узагальнена структура систем масового обслуговування

Узагальнена структурна схема СМО приведена на рис. 4.1. Вимоги на обслуговування $(\lambda_1 \ldots \lambda_m)$ надходять на вхід СМО та, в залежності від типу вимог за допомогою дисципліни черги (D_1), ставляться у відповідні черги $(Ч_1 \ldots Ч_m)$. Їх обслуговування виконується каналами обслуговування $(К_1 \ldots К_m)$ (або універсальними, або спеціалізованими, що обслуговують лише окремі типи вимог). Для кожного K_j вважаються відомими функції розподілу часу обслуговування.

Призначення кожної з вимог черги $Ч_k\left(K=\overline{1,N}\right)$ до каналу $K_j\left(j=\overline{1,m}\right)$ здійснюється в загальному випадку за допомогою дисципліни обслуговування D_2, що призначає черговість обслуговування. В результаті роботи $K_j\left(j=\overline{1,m}\right)$ формується вихідний потік обслужених вимог $\left(\lambda_1^0 \ldots \lambda_m^0\right)$ (потік обслуговувань). Якщо у момент надходження вимоги є вільні канали, обслуговування

починається без очікування. Але у випадку, коли канали зайняті, вимога за допомогою D_1 ставиться в одну з черг ($Ч_1 \ldots Ч_N$), при цьому черга може бути або загальною (N = 1), або роздільною. Розподілення здійснюється в цьому випадку за пріоритетним принципом.

На кількість місць очікування можуть накладатися обмежування, що викликає у певних обставинах відмови в формулюванні в чергу очікування. При цьому можливі також окремі конфліктні ситуації, що призводять до рішення щодо зняття окремих вимог з черги (при цьому утворюється потік виштовхувань).

Вимоги, що надійшли до черги очікування, можуть бути „терплячі ” або „нетерплячі ”, тобто такі, які втративши терпіння, покидають СМО не обслуженими.

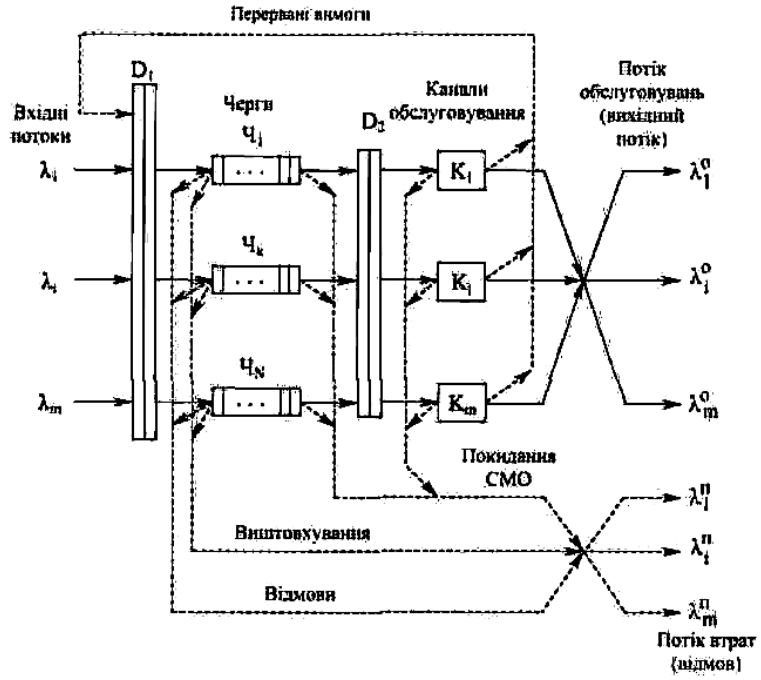

Рис. 4.1. Узагальнена структура СМО

Ці вимоги також входять до потоку відмов. Аналогічні ситуації можуть мати місце в каналах обслуговування, при цьому також утворюється потік покидання СМО з каналів обслуговування. В загальному випадку потоки відмов, виштовхування та покидання СМО утворюють вихідний потік втрат СМО. Звичайно, в залежності від типу СМО окремі складові СМО та складові вхідних та вихідних потоків можуть бути відсутні.

Таким чином, будь-яка СМО включає до себе дві частини: що обслуговується та що обслуговує. При цьому частина, що обслуговується, включає до себе сукупність джерел вимог, що створюють вхідний потік вимог СМО. Частина, що обслуговує, включає до себе, у загальному випадку, деякий накопичувач вимог на обслуговування (вимог, що чекають на обслуговування) і механізм обслуговування, що формує вихідний потік вимог після обслуговування. Механізм обслуговування може функціонувати паралельно, обслуговуючи одночасно кілька вимог. При цьому йдеться про багатоканальну СМО. Але механізм обслуговування може мати також деякі операції

обслуговування, що виконуються послідовно для кожної вимоги. В цьому випадку йдеться про наявність багатофазового каналу обслуговування й про багатофазну СМО. Очевидно, що в загальному випадку мова йде про багатоканальні та багатофазні СМО, де здійснюється паралельна багатофазна обробка кількох вимог.

Наприклад, система організації завантаження автомобілів на крупному вантажному терміналі уявляє собою багатоканальну та багатофазну СМО. У цієї СМО завантаження автомобілів може здійснюватися одночасно кількома автонавантажувачами, після завантаження необхідно зробити кілька допоміжних операцій: оформлення документів, зважування вантажу, перевірка на виїзді та інші, що визначає багатофазність каналів.

З точку зору ТМО таку систему необхідно формалізувати, представляючи її у вигляді деякої узагальненої структури, проаналізувати закон надходження вимог в СМО, визначити час знаходження вимог у кожній фазі, кількість автомобілів у черзі, кількість зайнятих каналів тощо. При цьому необхідно оцінити втрати від очікування та часу обслуговування, а також втрати від простою механізмів обслуговування, якщо вимог немає. В результаті аналізу можливо визначити найбільш раціональну організацію роботи терміналу (збільшити або зменшити кількість каналів обслуговування, змінити порядок обслуговування або його термін та ін.) Саме такі питання доводиться вирішувати при розробці будь-якої СМО. Таким чином, методи ТМО дозволяють визначити характер функціонування та структуру СМО за характеристиками її окремих частин (вхідний потік, накопичувач, механізм обслуговування, вихідний потік) ще на етапі проектування СМО, коли вона фізично ще не існує.

Предметом ТМО є побудова математичних моделей процесів, пов’язаних з масовим обслуговуванням та оцінка ефективності функціонування СМО. У якості показників ефективності СМО використовують середню кількість вимог, що очікують у черзі обслуговування; імовірність того, що кількість вимог у черзі перевищуватиме певне значення, та інші.

4.2. Класифікація систем масового обслуговування

Відрізняють два основних види СМО: з відмовами та з чергою чекання.

В СМО з відмовами вимога на обслуговування, що надходить, коли всі канали обслуговування зайняті, покидає СМО не обслугованою й більше не розглядається. В СМО з чергою чекання при зайнятості всіх каналів вимога ставиться у чергу очікування. При цьому її обслуговування, як правило, здійснюється за правилом черги: першим прийшов – обслуговується першим (так зване безпріоритетне обслуговування).

В окремих випадках здійснюється пріоритетне обслуговування вимог (наприклад, машини швидкої допомоги на АЗС обслуговуються позачергово і т.д.).

Пріоритет може бути абсолютним, коли вимога, що щойно надійшла, „виштовхує” тих, які чекають, а також відносним, коли більш „вагома ” вимога займає краще місце у черзі (наприклад, автомобілі учасників війни на СТО обслуговуються швидше, ніж автомобілі звичайних клієнтів). Слід зауважити,

що визначення пріоритетності заявок (її ще називають дисципліною обслуговування) поки що є недостатньо розробленою й тому у транспортних системах частіше застосовується безпріоритетне обслуговування.

СМО з чергою чекання діляться на СМО з обмеженою довжиною черги або необмеженою. Наприклад, якщо поблизу є декілька АЗС, то автомобіль, що прибув для заправки, може її покинути, зважаючи на досить довгу чергу та рушити на іншу. У даному випадку йдеться про обмеження черги через час очікування. Те ж саме трапляється іноді через нестачу місць для очікування, тоді йдеться про обмеження довжини черги.

Інша ситуація, коли поблизу немає інших АЗС – тоді автомобіль мусить чекати заправки, незважаючи на величезну чергу чекання. У цьому випадку йдеться про СМО з необмеженою чергою чекання.

Як вже відмічалось раніше, СМО бувають одноканальні або багатоканальні (за кількістю паралельних каналів обслуговування (наприклад, станції технічного обслуговування автомобілів можуть бути як одно -, так і багатоканальними, враховуючи кількість ліній обслуговування). Як правило, СМО не класифікуються за числом фаз обслуговування в одному каналі, але при детальному аналізі кожного каналу це обов’язково береться до уваги.

В залежності від кількості джерел вимог на обслуговування відрізняють розімкнуті та замкнуті СМО. В розімкнутих СМО кількість вимог, що надходить до СМО для обслуговування, вважається необмеженою, тобто їх кількість не залежить від режиму роботи СМО та її стану. Для замкнутих СМО характерним є обмежена кількість джерел вимог обслуговування, які багаторазово повертаються в СМО, хоча період повернення є випадковою величиною. Властивість замкнутості СМО проявляється тим сильніше, чим менша кількість джерел вимог на обслуговування. Очевидно, що потік вимог у таких СМО залежить від того, скільки вимог в поточний час пов’язані із СМО (обслуговуються або знаходяться у черзі очікування). Прикладом замкнутої СМО може бути сумісна робота навантажувально-розвантажувального агрегату на вантажному терміналі, котрий обслуговує визначену кількість транспортних засобів. Розімкнутою системою можна вважати, наприклад, бюро прийому телефонних замовлень на здійснення перевезень, де кількість замовлень не залежить від етапу СМО тощо.

В залежності від умов формування вхідного та вихідного потоків СМО діляться також на:

- СМО з дискретним часом, коли прибуття вимоги на обслуговування або її покидання СМО обслугованою здійснюється в певні моменти часу, відстані між якими є кратними деякій фіксованій величині. Подібний режим змін стану СМО характерний здебільшого для ПК, що здійснює обробку різноманітної інформації лише за командами пристрою управління, який синхронізує роботу усіх блоків ПК та алгоритмів обробки інформації;

- СМО з безперервним часом, коли прибуття вимог в СМО здійснюється у будь-які моменти часу, без жодних обмежень на інтервал між ними. Подібний режим змін стану СМО характерний, наприклад, для роботи автозаправних

станцій та станцій технічного обслуговування (СТО), коли автомобілі прибувають і покидають такі СМО у будь-який момент часу.

4.3. Характеристики та критерії ефективності систем масового обслуговування

Будь-яка система масового обслуговування характеризується параметрами, характеристиками та критеріями ефективності. До параметрів СМО відносяться показники структури (одно - або багатоканальна СМО), умови її роботи (з чеканням або з відмовами, замкнута або розімкнута СМО). Ці параметри розглядалися раніше.

До характеристик СМО належать такі властивості СМО, які можуть бути оцінені кількісно. До них належать:

1) інтенсивність вхідного потоку (λ, од/од. часу), тобто середня кількість вимог на обслуговування, що надходять в СМО за одиницю часу. Ця усереднена характеристика має обов'язково супроводжуватись характеристикою закону розподілу потоку (пуассонівський, ерлангівський, гауссівський та ін.);

2) інтенсивність вихідного потоку (μ, од/од. часу), що характеризує пропускну здатність механізму обслуговування та визначається як величина, зворотна середньому часу обслуговування при умові, що наступна вимога надходить саме у момент, коли закінчується обслуговування попередньої вимоги. Тобто при вихідному потоці μ СМО завжди є в роботі й не має вільного часу. Тому не слід змішувати поняття μ з інтенсивністю реального потоку обслуговування в СМО, яка завжди є меншою за μ, зважаючи на випадковий характер надходження вимог на обслуговування, що обумовлює певні інтервали часу, коли СМО є у вільному стані;

3) абсолютна пропускна здатність СМО (А) – це середня кількість обслужених СМО вимог за одиницю часу. По суті, ця характеристика і є характеристикою реального потоку обслуговування в СМО та залежить як від інтенсивності вхідного потоку λ, так і від пропускної здатності механізму обслуговування μ. Тому, говорячи про абсолютну пропускну здатність СМО, слід розуміти, що це не є певний параметр СМО, а лише характеристика СМО при визначеному вхідному потоці вимог на обслуговування. Очевидно, що в СМО без відмов $A = \lambda$;

4) відносна пропускна здатність СМО (q) – це відношення кількості обслужених вимог (інтенсивності обслуговування) до загальної кількості вимог, що надійшли в СМО. Це визначення q дозволяє вважати також, що вона є також імовірністю обслуговування вимог, що надходять в СМО. Очевидно, що в СМО без відмов $q = 1$, з відмовами $q < 1$.

Наступні характеристики СМО можуть розглядатись одночасно як критерії ефективності їх роботи – змінюючи або структуру, або режим роботи СМО, можливо міняти певні параметри СМО з метою поліпшення саме цих характеристик:

- середня кількість зайнятих каналів;
- середня кількість вимог у черзі очікування;

- середній час очікування у черзі;

- середній час обслуговування;

- середній час знаходження вимоги в СМО (тобто час очікування + час обслуговування);

- імовірність відмов в обслуговуванні та інші.

В окремих СМО (наприклад, в інформаційно-обчислювальних системах) у якості критерію ефективності використовують деякі комплексні показники, що характеризують оцінку пристосованості СМО виконувати притаманні їй функції. У якості такого показника вибирають:

$$E = \sum_{i=1}^{M} l_{ci} \bar{t}_{ci}, \tag{4.1}$$

де l_{ci} – штраф за одиницю часу перебування в СМО вимоги і-го типу, $\bar{t}_{ci}$ – середній час перебування і-ї вимоги в СМО. Очевидно, що при однорідності вимог (М = 1) матимемо критерій ефективності роботи СМО $E = l_{ci} \bar{t}_{ci}$.

Під потоками подій розуміють послідовність однорідних подій, що слідують одна за одною у деякі, в загальному випадку випадкові, моменти часу. Стосовно СМО під вхідним потоком подій розуміють потік вимог на обслуговування (наприклад, потік автомобілів, що прибувають на АЗС для заправки, де подією є прибуття одного будь-якого автомобіля, моментом здійснення події – момент його прибуття на АЗС, часовий інтервал між подіями – інтервал між моментами прибуття на АЗС цього та попереднього автомобіля), а також вихідний потік обслуговування (наприклад, потік заправлених автомобілів, що покидає АЗС). У деяких випадках розглядають також потік вимог, що не обслуговуються через дуже велику чергу очікування або відсутність вільних місць очікування. Цей потік часто зветься потоком втрат СМО (або потоком не обслужених вимог).

Як вже відмічалось, потік подій характеризується інтенсивністю λ, тобто частотою появи подій або середньою кількістю подій за одиницю часу. Потік подій є регулярним, якщо події відбуваються через рівні інтервали часу. Наприклад, пересічення „стоп – лінії ” на регульованому перехресті впродовж роз’їзду черги автомобілів розглядають як регулярний. Регулярним також буде потік автомобілів з конвеєра збирального цеху, якщо швидкість конвеєра є незмінною.

На відміну від регулярного, випадковий потік характеризується нерівномірністю інтервалів часу слідування подій. Головною характеристикою випадкового потоку є імовірність попадання інтервалу часу між подіями в задане значення. При цьому розглядають, як правило, дві характеристики:

- імовірність появи поточного інтервалу t_i, меншого заданого значення τ, де τ та t_i, відраховуються від моменту здійснення (і – 1)-ї події. Ця характеристика $F(\tau) = P[t_i < \tau]$ має назву інтегральної функції розподілу імовірностей часових інтервалів слідування подій у потоці;

- щільність розподілу часових інтервалів слідування подій f(τ), що

представляє собою похідну від F(τ), тобто:

$$f(\tau) = \frac{dF(\tau)}{d\tau}$$

і характеризує швидкість зміни F(τ) при зміні значення τ. Ця взаємозалежність свідчить, що при аналізі потоку подій досить знайти або F(τ), або f(τ).

В ТМО найбільш поширені саме випадкові процеси подій, при цьому він формалізується наступним чином.

Нехай $t_0, t_1, t_2, \ldots, t_N$ – моменти надходження вимог на обслуговування в СМО (або моменти покидання СМО обслуженими вимогами). Величина t_0 – умовний момент розгляду початку потоку. Позначимо $\tau_1 = t_i - t_{i-1}$ – інтервали часу між моментами звершення і-ї та (і – 1)-ї подій. Потік вважається заданим, якщо відомі значення $\tau_i (i \in \overline{1,n})$.. Додамо, що оскільки потоки випадкові, то необхідним для опису потоку є не стільки значення інтервалів, скільки їх імовірнісні характеристики F(τ) або P(τ).

Запитання для самоконтролю:

1. Яка узагальнена структура систем масового обслуговування?
2. Яка класифікація систем масового обслуговування?
3. Які критерії ефективності систем масового обслуговування?

ТЕМА 5. СІТЬОВЕ ПЛАНУВАННЯ І УПРАВЛІННЯ КОМПЛЕКСАМИ РОБІТ

5.1. Поняття та терміни

Сітьові моделі призначені для формування календарного плану реалізації проекту.

Проект – це сукупність робіт, які необхідно виконати для досягнення певного результату.

Основою сітьової моделі є сітьовий графік, що дає наочне уявлення про комплекс робіт.

Сітьовий графік – це графічне зображення планованого процесу (проекту, розробки), що відображає взаємозв’язок і послідовність робіт, що входять до нього.

З математичної точки зору сітьовий графік є орієнтований кінцевий граф без контурів, вершини якого зображають події, а дуги – елементарні операції (роботи).

Під подією розуміємо факт, результат закінчення однієї або декількох робіт, необхідний і достатній для початку наступних робіт.

Розрізняють висхідні, проміжні та завершальні події.

Робота – це процес, необхідний для переходу від однієї події до іншої. Кожна робота має попередню подію, до настання якого вона не може початися.

Роботи бувають декількох видів.

Дійсними називаються роботи, які потребують витрат часу і ресурсів, наприклад, розробка технічного проекту, закладка фундаменту, заправка повітряного судна паливом і т.п. Фіктивні роботи (залежності) зображають

логічні зв'язки між роботами та не потребують витрат часу і ресурсів. Очікуванням називається робота, яка потребує витрат часу, але не потребує витрат ресурсів (наприклад, процес твердіння бетону, сушка забарвлених поверхонь і т.п.).

Події в сітьовому графіку відображаються кружечками і нумеруються числом і. Кожна робота зображається в сітьовому графіку дугою і відображається парою чисел (i, j), перше з яких показує номер попереднього, а друге – подальшої за роботою події.

Правильна нумерація подій в сітьовому графіку – це така нумерація, за якої номер кожної наступної події більше номера будь-якого попереднього.

Шляхом в сітьовому графіку називається будь-яка послідовність дуг, у котрої кінцева вершина кожної попередньої дуги співпадає з початковою вершиною наступної дуги. Повний шлях – це шлях, початкова вершина якого співпадає з результатним, а кінцева – із завершальною подією.

5.2. Порядок і правила побудови графів

При побудові графа необхідно дотримуватися певних правил, щоб в подальшому можна було досліджувати його. Графічне зображення вимог щодо побудови графів приведене на рис. 5.1.

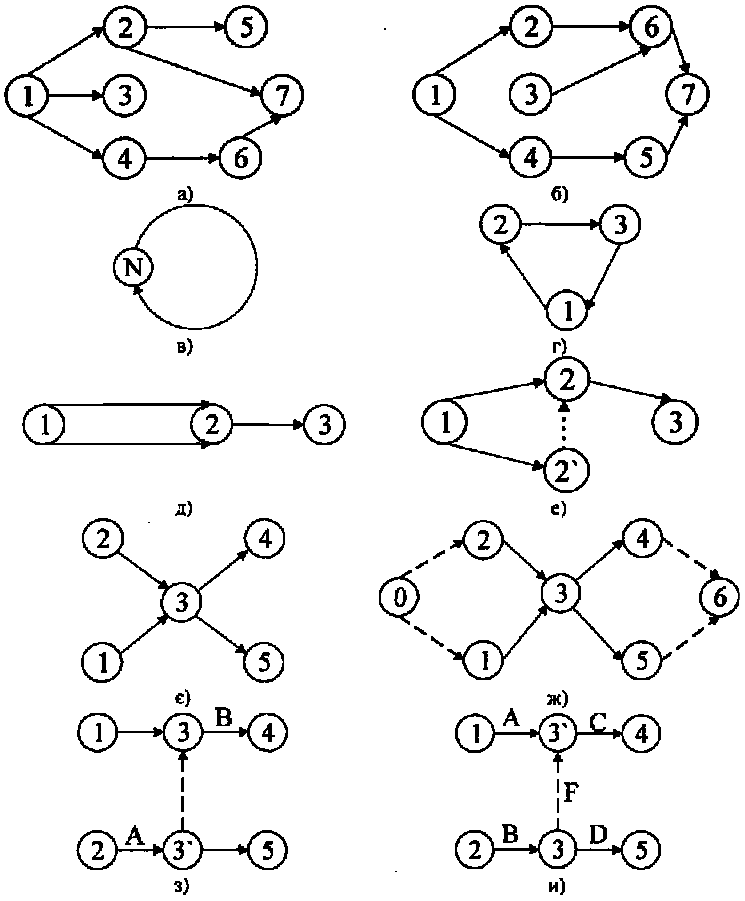

Рис. 5.1. Графічне зображення вимог до побудови графів

1) Граф не повинен мати „безвиході ", тобто подій, з яких не виходить жодної роботи, окрім завершальної події (рис. 5.1 а). Поява „безвиході " подій свідчить про недостатньо ретельно виконаний аналіз робіт та їх взаємозв'язки.

2. На графі не може бути „хвостових ” подій (крім висхідної події), тобто подій, яким не передує жодна робота. На рис. 5.1 такою „хвостовою ” подією є подія 3; вона не може відбутися, отже, не можуть відбутися й наступні події.

3. Граф не може мати замкнутих контурів і петель, тобто шляхів, що сполучають певні події з ними ж (рис. 5.1 в, г). Поява замкнутих контурів вимагає перегляду складу робіт та їх взаємозв’язків, після змістовного аналізу яких завжди з’являється можливість уникнути замкнутих контурів і петель.

4. Дві довільні події повинні бути безпосередньо зв’язаними не більше ніж однією дугою-роботою. Ця вимога обумовлена тим, що роботи позначають двома індексами (i, j), що відповідають подіям „i” та „j”. Якщо насправді потрібно виконати декілька робіт, які починаються та завершуються одночасно, при одних й тих же початкових і кінцевих подіях, то в таких випадках необхідно ввести фіктивні події та фіктивні роботи (рис. 5.1 д), паралельні роботи при цьому замикаються на фіктивні події (рис. 5.1 е).

5. На графі повинно бути лише одне результатне і лише одна завершальна подія. Якщо це об’єктивно не так (початок реалізації комплексу робіт можна починати паралельно з декількох робіт – рис. 5.1 ж, е), то необхідно ввести фіктивні події та роботи, як це показано на рис. 5.1 з.

5.3. Побудова правильної нумерації вершин графа

Взагалі вершини графа можна нумерувати довільно, але для вирішення багатьох практичних завдань зручно виконувати так звану правильну нумерацію вершин. При такій нумерації будь-який шлях від вершини з меншим номером до вершини з великим номером проходитиме лише через вершини з номерами, що ростуть. Правильна нумерація вершин виконується за так званим „алгоритмом викреслювання дуг ”.

Опишемо зміст цього алгоритму, пояснюючи його етапи щодо графа, зображеного на рис. 5.2, для якого в квадратах приводиться деяка довільна нумерація вершин.

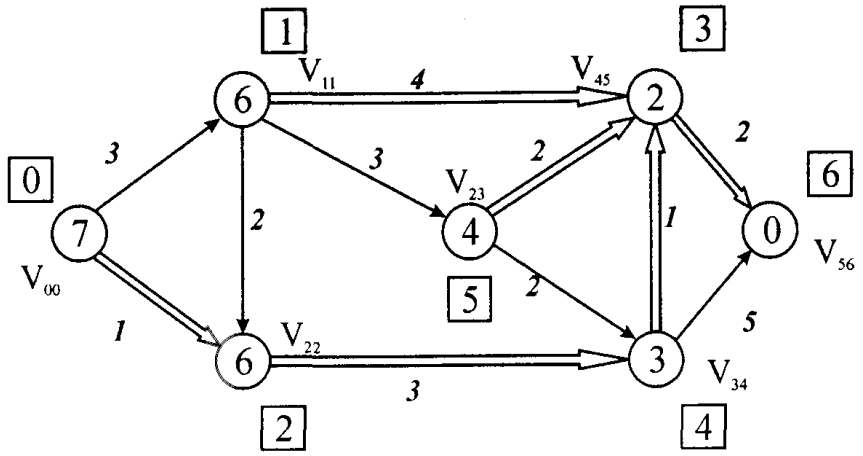

Рис. 5.2. Побудова вірної нумерації вершин графу

Знаходимо відправну крапку – висхідна подія, у яку не входить жодна робота. Привласнюємо цій події 0-й ранг. Викреслюємо всі роботи, які виходять з події 0-го рангу. Далі шукаємо події, у які не входять ніякі роботи, окрім викреслених. Їм привласнюємо 1-й ранг. Після цього викреслюємо всі

роботи, які виходять з подій 1-го рангу. Таким чином повторюємо викреслювання і привласнення рангів подіям до тих пір, поки всі події не отримають свій ранг. Нумерація подій здійснюється згідно номеру рангу. Так, висхідну подію завжди має 1-й номер. Якщо декілька подій мають однаковий ранг, їх можна нумерувати в будь-якій послідовності.

I. Умовно виділимо всі дуги, що виходять з початкової вершини, назвемо її вершиною нульового рангу і дамо їй номер „00” (V_{00}). Тепер розглянемо вершини, в які не входять ніякі дуги, окрім викреслених. Такі вершини назвемо вершинами 1-го рангу і пронумеруємо їх у довільному порядку, дотримуючись безперервності в нумерації. За цих умов кожній вершині надаватимемо два індекси: перший – ранг вершини, другий – її порядковий номер серед множини вершин однакового рангу. На графі рис. 5.2 маємо одну вершину „11” (V_{11}) першого рангу.

II. Умовно викреслимо всі дуги, що виходять з вершин 1-го рангу. Вершини, у які не входять ніякі дуги, окрім вже позначених, назвемо вершинами 2-го рангу і пронумеруємо їх в довільній послідовності, зберігаючи безперервність в нумерації дотично раніше використаних чисел натурального ряду. Для графа рис. 5.2 це вершини „22” (V_{22}) та „23” (V_{23}).

Допустимо, що пройдений n-1 етап і визначені вершини (n-1)-го рангу. Викреслимо всі дуги, що виходять з вершин (n-1)-го рангу. Розглянемо всі вершини, у яких закінчуються викреслені на цьому етапі дуги вершини, у які входять лише дуги, визначені на (n-1)-му етапі, утворюють множину вершин n-го рангу. Пронумеруємо їх у довільному порядку, безперервно використовуючи числа натурального ряду, починаючи з найменшого, яке не було використано для нумерації вершин (n-1)-го рангу. Алгоритм завершується після досягнення кінцевої вершини. Для нашого прикладу (рис. 5.2) це вершина „56” (V_{56}), де індекс 5 означає ранг кінцевої вершини, а індекс 6 – її порядковий номер. Виконавши правильно нумерацію вершин графа, надалі індекс рангу вершини можна не указувати.

5.4. Часові параметри сітьового графіка

Основними параметрами сітьових графіків є критичний шлях, ранні та пізні терміни здійснення подій, ранні та пізні терміни початку й закінчення робіт, резерви часу подій і робіт.

Повний шлях, що має найменшу тривалість, називається критичним. Роботи та події, що належать критичному шляху, називаються критичними. Тривалість критичного шляху характеризує мінімальну тривалість виконання всього комплексу робіт, тобто проекту.

Ранній термін здійснення події $t_p(i)$ дорівнює тривалості щонайдовшого з всіх шляхів від висхідної події до даної. Пізній термін здійснення події $t_n(i)$ дорівнює різниці між тривалістю критичного та тривалістю щонайдовшого з всіх шляхів від даної події до тієї, що завершує. Резерв часу події – це різниця між пізнім і раннім терміном здійснення події.

Ранній термін початку роботи дорівнює ранньому терміну здійснення її початкової події. Пізній термін закінчення роботи дорівнює пізньому терміну

здійснення її кінцевої події. Пізній термін початку роботи дорівнює пізньому терміну її закінчення мінус її тривалість. Ранній термін закінчення роботи дорівнює ранньому терміну початку роботи плюс її тривалість.

Повний резерв часу роботи є інтервал часу між раннім і пізнім термінами початку роботи. Він показує, у яких межах можна пересунути початок роботи (або наскільки можна розтягнути її тривалість), не змінюючи при цьому терміну виконання всього проекту.

Вільний резерв часу роботи – це запас часу, на який можна збільшити тривалість роботи або відкласти її початок у припущенні, що початкова та кінцева події цієї роботи здійснюються у свої ранні терміни.

Визначення часових параметрів подій та робіт і критичного шляху сітьового графіка – це тільки частина планування. Наступним етапом сітьового планування та управління є перевірка фізичної реалізації проекту. Відправною точкою цього етапу є визначення загальної потреби у ресурсах для кожного одиничного інтервалу часу. Для вирішення цього завдання застосовуються лінійні графіки Гантта.

Завдання оптимізації в сітьовому графіку можуть бути такого вигляду:

1) при обмежених ресурсах розподілити їх так, щоб мінімізувати час виконання проекту. Іноді виникає потреба отримати час виконання менше критичного. Тоді оптимізація комплексу робіт полягає у перерозподілі внутрішніх резервів або за рахунок залучення додаткових коштів;
2) при заданому часі виконання проекту мінімізувати нерівномірність споживання ресурсів.

Для вирішення оптимізаційних завдань сітьового планування та управління використовуються методи математичного програмування.

5.5. Упорядкування графа, обчислення основних параметрів подій та робіт

Комплекс робіт (проект) викладений у вигляді сітьового графіка на рис. 5.3.

На стрілках (роботах) вказана тривалість робіт, а в кружечках (подіях) – параметри подій (рис. 5.4).

Розглянемо параметри подій. Певна подія j не може відбутися раніше, ніж завершаться всі попередні перед нею роботи. Отже, ранній термін $t_p(j)$ можливого звершення j-ої події визначається терміном максимального шляху.

Якщо подія j має декілька шляхів, які їй передують, отже, декілька попередніх подій i, то ранній термін $t_p(j)$ звершення події j зручно обчислювати за формулою:

$$t_p(j) = \max[t_p(i) + t(i, j)].$$

Формула показує, що обчислення параметра t_p доцільно починати з висхідної події, для якого t_p дорівнює нулю, розглядаючи наступні події в порядку збільшення їх номерів.

Затримка із звершенням i-ї події щодо свого раннього терміну не впливатиме на термін звершення завершальної події (отже, і на термін виконання досліджуваного комплексу робіт), поки сума термінів звершення i-ї події і терміну максимального зі всіх шляхів, такого, що йде від i-ї події до

того, що завершує, не перевищить терміну критичного шляху.

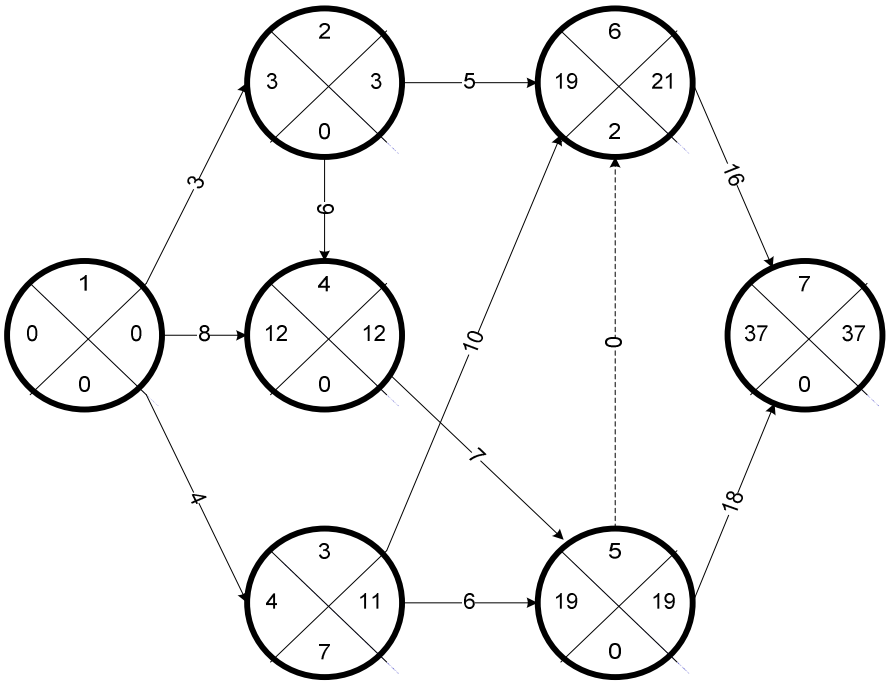

Рис. 5.3. Приклад сітьового графа

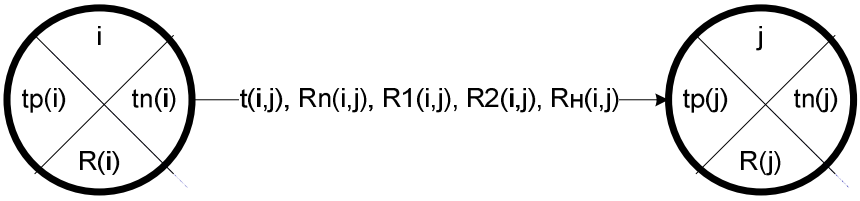

Рис. 5.4. Схема розташування параметрів подій

Якщо подію „i” має декілька наступних шляхів, пов’язаних з декількома наступними подіями „j”, то пізній термін звершення події „i” зручно обчислювати за формулою:

$$t_n(i) = \min\left[t_n(j) - t(i, j)\right].$$

Резерв часу R(i) i-ї події обчислюється як різниця пізнього та раннього термінів звершення i-ї події:

$$R(i) = t_n(i) - t_p(i).$$

Резерв часу події показує, на який допустимий термін можна затримати звершення події, не гальмуючи (не збільшуючи) при цьому термін виконання всього комплексу робіт мережі.

Критичні події не мають резервів часу, оскільки будь-яка затримка із звершенням подій, розташованих на критичному шляху, викличе таку ж затримку у виконанні завершальної події.

Отже, щоб визначити термін критичного шляху, необхідно і достатньо обчислити ранній термін завершальної події, її величину та визначити цей термін.

Події з нульовим резервом часу визначають роботи, що складають

критичний шлях.

Приклад. Обчислити параметри часу подій і критичний шлях графа рис. 5.1. Результати приведені в табл. 5.1.

Таблиця 5.1– Результати розрахунку параметрів подій графа

Номер події	**t_p – ранній термін звершення**	**t_n – пізній термін звершення**	**R_p – резерв часу**
2	0+3=3	$t_n = t_p$ (критична подія)	0
3	0+4=4	min{19-6;21-10}=11	11-4=7
4	max {0+8;3+9}=12	$t_n = t_p$ (критична подія)	0
5	max {4+6;12+7}=19	$t_n = t_p$ (критична подія)	0
6	max {3+5.4+10; 19+0}=19	37-16-21	21-19=2
7	max {19+16; 19+18}=37	37	0

Для початкової вершини i = 1 очевидно tp(i)= 0. Для i = 2 маємо $t_p(2)= 3$, оскільки існує лише одна робота, завершення якої відповідає події 2. З тієї ж причини $t_p(3)= 4$. Для обчислення $t_p(4)$ скористаємося формулою і заданими величинами t(2;4) = 9 та t(1;4) = 8:

$t_p(4) = \max \{t_p(1)+t(1;4); t_p(2)+t(2;4)\} = \max \{0+8; 3+9\} = 12.$

Отже, $t_p(4) = 12$.

Користуючись формулою, маємо:

$t_p(5)= \max \{t_p(3)+t(3;5); t_p(4)+t(4;5)\} = \max \{4+6; 12+7\} = 19.$

$t_p(6)= \max \{t_p(2)+t(2;6); t_p(5)+t(5;6); t_p(3)+t(3;6)\} = \max \{3+5; 19+0; 4+10\} = 19.$

$t_p(7)= \max \{t_{p(}5)+t(5;7); t_p(6)+t(6;7)\} = \max \{19+18; 19+16\} = 37.$

Отже, термін критичного шляху дорівнює 37 одиницям вимірювання часу.

При обчисленні пізніх термінів здійснення подій $t_n(i)$ розглядаємо вершини графа в порядку зменшення їх нумерації. Для завершальної події пізній термін його звершення дорівнює ранньому, тому $t_n(7)= 37$. Пізні терміни інших подій обчислюємо, користуючись відповідною формулою.

$t_n(6)= \min \{t_n(7) – t(6;7)\} = \min \{37 – 16\} = 21$, тому що для події 6 існує лише один наступний шлях (6;7).

$t_n(5)= t_n(7) – t(5;7)= 37 – 18 = 19.$

$t_n(4)= t_n(5) – t(4;5)= 19 – 7 = 12$, тому що для події 4 існує лише один наступний шлях, який починається з дуги (4;5).

$t_n(3)= \min \{t_n(5) – t(3;5); t_n(6) – t(3;6)\} = \min \{19 – 6; 21 – 10\}= 11.$

$t_n(2)= \min \{t_n(4) – t(2;4); t_n(6) – t(2;6)\} = \min \{12 – 9; 21 – 5\} = 3.$

$t_n(1)= 0.$

Обчислюємо резерв часу для кожної події:

R(1) = 0 – 0 = 0;

R(2) = 3 – 3 = 0;

R(3) = 11 – 4 = 7;

R(4) = 12 – 12 = 0;

R(5) = 19 – 19 = 0;

R(6) = 21 – 19 = 2;

R(7) = 37 – 37 = 0.

Відмінний від нуля (ненульовий) резерв часу певної події означає, що термін звершення такої події може бути збільшений на величину її резерву часу без затримки терміну виконання всього комплексу робіт.

Аналізуючи результати розрахунків, знаходимо, що події, що належать

критичному шляху, не мають резерву часу.

Перейдемо до розрахунків параметрів робіт.

Окрема робота може бути виконана за час відповідно до раннього, пізнього або будь-яким проміжним терміном, який відповідає події, що завершує цю роботу. Очевидно, що ранній термін $t_{pn}(i, j)$ почала роботи (i, j) співпадає з раннім терміном здійснення попередньої події i, тому:

$$t_{pn}(i, j) = t_p(i).$$

Таким чином, ранній термін $t_{pз}(i, j)$ завершення роботи (i, j) обчислюється так:

$$t_{pз}(i, j) = t_p(i) + t(i, j).$$

Для того, щоб фактичний термін виконання певної роботи не збільшив час виконання комплексу робіт, повинна виконуватися наступна вимога: кожну роботу слід завершувати не пізніше допустимого терміну її завершальної події j. За таких умов пізній термін $t_{pз}(i, j)$ завершення роботи (i, j) визначається формулою:

$$t_{nз}(i, j) = t_n(j),$$

а пізній термін $t_{nn}(i, j)$ початку цієї роботи – формулою:

$$t_{nn}(i, j) = t_n(j) - t(i, j).$$

Таким чином, моменти початку і завершення кожної роботи тісно пов’язані з сусідніми подіями.

Розглянемо резерви часу шляхів. Такі резерви мають всі некритичні шляхи. Резерв часу $R_t(L)$ шляху L визначається як різниця термінів критичного і даного шляхів:

$$R_t(L) = t_{кр} - t(L).$$

Величина $R_t(L)$ показує, на скільки можуть бути збільшені в сумі терміни всіх робіт, складових шлях L. Якщо збільшити терміни виконання робіт шляху L в сумі більше чим на величину $R_t(L)$, критичним шляхом буде L. Отже, будь-яка з робіт шляху L, якщо вона не є такою, що становить критичного шляху, має певний резерв часу.

Розглянемо чотири різновиди резервів часу робіт.

Повний резерв $R_n(i, j)$ часу роботи (i, j) показує, на скільки можна збільшити термін виконання даної роботи за умови, що термін виконання всього комплексу робіт не зміниться. Повний резерв $R_n(i, j)$ часу роботи (i, j) обчислюється за формулою:

$$R_n(i, j) = t_n(j) - t_p(i) - t(i, j).$$

Повний резерв часу роботи дорівнює резерву часу максимального з шляхів, що містять цю роботу. Цим резервом можна скористатися при виконанні роботи за умови, що її початкова подія відбудеться в найбільш ранній, а її завершальна подія – в найбільш пізній термін.

Істотною властивістю повного резерву часу роботи є те, що цей резерв стосується не тільки цієї роботи, але і всіх повних шляхів, що містять її. Якщо використовувати повний резерв часу лише для цієї однієї роботи, то резерви

часу решти робіт максимального шляху, що містить її, будуть повністю вичерпані. Резерви часу робіт, що належать іншим, не максимальним по термінах шляхам, які містять цю роботу, відповідно зменшаться на величину використаного резерву.

Решта резервів часу роботи, які розглянемо далі, є частинами його повного резерву.

Резервом частини першого виду R_1 часу роботи (i, j) є та частина повного резерву часу цієї роботи, на яку можна збільшити термін роботи, не змінивши при цьому пізній термін її початкової події. Цим резервом можна скористатися при виконанні певної роботи за умови, що її початкова і завершальна події відбудуться у свої пізні терміни. Величина $R_1(i, j)$ обчислюється за формулою:

$$R_1(i, j) = t_п(j) - t_п(i) - t(i, j) = R_п(i, j) - R(i).$$

Резервом частини другого виду R_2 часу роботи (i, j) є та частина повного резерву часу цієї роботи, на яку можна збільшити термін її виконання, не змінивши при цьому ранній термін її початкової події.

Цим резервом можна користуватися при виконанні певної роботи за умови, що початкова і завершальна події відбудуться у свої ранні терміни. Величина $R_2(i, j)$ обчислюється за формулою:

$$R_2(i, j) = t_р(j) - t_р(i) - t(i, j) = R_п(i, j) - R(j).$$

Резервом часу R_2 можна користуватися, щоб запобігти випадковостям, які можуть трапитися при виконанні відповідної роботи. За умови планування виконання робіт в ранні терміни їх початку і завершення завжди маємо можливість при необхідності перейти на пізні терміни початку і завершення робіт, тобто величина $R_2(i, j)$ показує, на скільки можна перенести початок або збільшити термін виконання роботи.

Незалежним резервом часу $R_н$ роботи (i, j) є та частина повного резерву часу цієї роботи, яка визначається за умови, що попередні роботи завершуються в пізні терміни, а всі наступні починаються в ранні терміни.

Величина $R_н$ обчислюється за формулою:

$$R_н(i, j) = t_р(j) - t_п(i) - t(i, j) = R_2(i, j) - R(i) = R_1(i, j) - R(j).$$

Використання незалежного резерву часу не впливає на величини резервів часу інших робіт. Незалежні резерви часу прагнуть використовувати тоді, коли завершення попередньої роботи відбулося в пізній термін, а наступні роботи бажано виконати в ранній термін. Таку можливість маємо, якщо величина незалежного резерву часу $R_н(i, j)$ дорівнює нулю або позитивна. За умови, що величина $R_н(i, j)$ негативна, немає можливості виконати наступні роботи в ранні терміни, оскільки попередня робота ще не завершена, тому негативні значення $R_н(i, j)$ не мають реального сенсу. Незалежний резерв часу має сенс лише для тих робіт, які не належать максимальним шляхам, що проходять через їх початкові та завершальні події.

Власний резерв першого виду часу роботи може бути використаний на збільшення терміну виконання даної та наступних робіт. Частковий резерв другого виду часу роботи може бути використаний на збільшення терміну

виконання даної та попередніх робіт без зміни (порушення) резерву часу наступних робіт. Незалежний резерв часу може бути використаний для збільшення терміну виконання лише даної роботи.

Роботи, що складають критичний шлях, і критичні події будь-яких резервів часу не мають. Якщо початкова подія і роботи (i, j) належить критичному шляху, то повний резерв часу

$$R_n(i, j) = R_1(i, j).$$

Якщо завершальна подія j роботи (i, j) належить критичному шляху, то повний резерв часу

$$R_n(i, j) = R_2(i, j).$$

Якщо початкова подія і та завершальна подія j належать критичному шляху, але сама робота не належить йому, то повний резерв часу визначається так:

$$R_n(i, j) = R_1(i, j) = R_2(i, j) = R_н(i, j).$$

Приведені співвідношення доцільно використовувати для контролю правильності розрахунків резервів термінів виконання окремих робіт.

Обчислимо тимчасові параметри робіт графа рис. 5.1.

Результати розрахунків подані в табл. 5.2; для кращої орієнтації приведені проміжні обчислення.

Таблиця 5.2 – Таблиця розрахунків параметрів робіт графа

Роботи	**Час робіт**	**R_n**	**R_1**	**R_2**	**R_n**
1; 2	3	3-0-3 = 0	3-0-3 = 0	3-0-3 = 0	3-0-3 = 0
1; 3	4	11-0-4 = 7	11-0-4 = 7	4-0-4 = 0	4-0-4 = 0
1; 4	8	12-0-8 = 4	12-0-8 = 4	12-0-8 = 4	12-0-8 = 4
2; 4	9	12-3-9 = 0	12-3-9 = 0	12-3-9 = 0	12-3-9 = 0
2; 6	5	21-3-5 = 13	21-3-5 = 13	19-3-5 = 11	19-3-5 = 11
3; 5	6	19-4-6 = 9	19-11-6 = 2	19-4-6 = 9	19-11-6 = 2
3; 6	10	21-4-10 = 7	21-11-10 = 0	19-4-10 = 5	19-11-10 = -2
4; 5	7	19-12-7 = 0	19-12-7 = 0	19-12-7 = 0	19-12-7 = 0
5; 6	0	21-19-0 = 2	21-19-0 = 2	19-19-0 = 0	19-19-0 = 0
5; 7	18	37-19-18 = 0	37-19-18 = 0	37-19-18 = 0	37-19-18 = 0
6; 7	16	37-19-16 = 2	37-21-16 =0	37-19-16 = 2	37-21-16 = 0

5.6. Діаграма Гантта

Найпростішим інструментом, що дозволяє отримати наочне уявлення про проект і визначити його тривалість, є діаграма Гантта. Такі діаграми Генрі Гантт вперше застосував на початку XX століття для мінімізації часу виконання послідовності машинних операцій.

Для побудови діаграми зображатимемо стадії прямокутниками, довжини яких пропорційні тривалості стадій (рис. 5.6). Причому відкладатимемо прямокутники, керуючись принципом: почати кожну стадію так рано, як тільки можливо. Наприклад, стадії А, В і З не мають попередників.

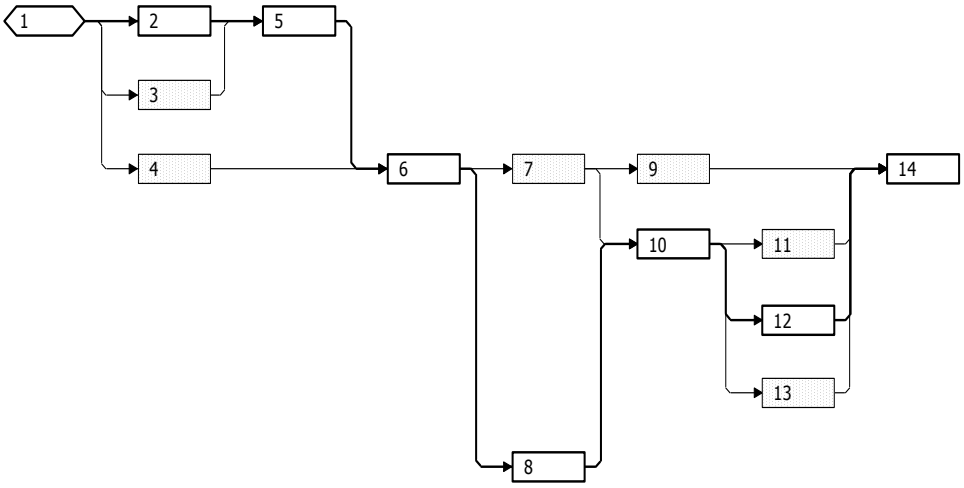

Рис. 5.5. Приклад сітьового графіка

Значить, їх можна почати одночасно у момент часу t = 0. Проте стадію D можна почати не раніше, ніж закінчиться найбільш тривала з її попередниць (A, B, З) – стадія A, тобто у момент t = 5. Стадії F та G також можна почати одночасно, після закінчення їх попередниці – стадії E (у момент часу t = 13) і т.д. Продовжуючи процес побудови до вичерпання всіх стадій, знайдемо тривалість проекту – 70 одиниць. Це число повинне співпадати з тривалістю критичного шляху.

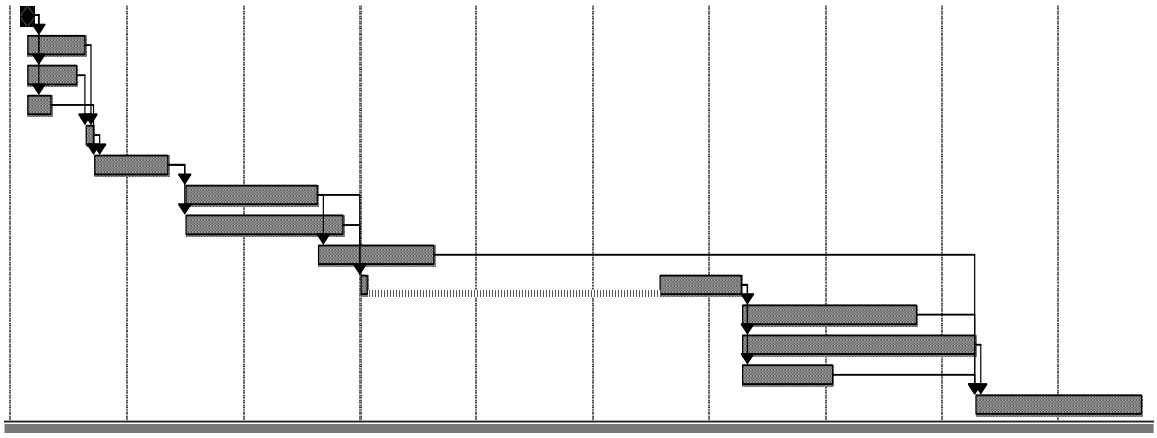

Рис. 5.6. Діаграма Гантта

Діаграма Гантта дає не тільки час виконання проекту, але також один з можливих його розкладів, коли кожна стадія починається так рано, як тільки можливо.

Разом з тим, уважно розглянувши діаграму Гантта, можна відмітити, що не всі стадії однаково впливають на час виконання проекту і, відповідно, не всі стадії слід прагнути починати (й закінчувати) так рано, як тільки можливо. Наприклад, ясно, що початок стадії L можна безболісно відсунути строком до 13 одиниць. Це не спричинить за собою подовження проекту в цілому. В той же час, стадію K неможливо відсунути (або затримати її закінчення) без того, щоб не продовжити проект, оскільки затримка з виконанням стадії K неминуче викличе затримку початку робіт на стадії M, що спричинить подовження проекту. Такі стадії називають „критичними ” стадіями, оскільки вони критично впливають на тривалість проекту.

Запитання для самоконтролю:

1. Що таке сітьове планування та управління?
2. Які правила побудови графів?
3. Що таке часові параметри сітьового графіку?
4. Для чого використовується діаграма Гантта?

СПИСОК ДЖЕРЕЛ

1. Системологія на транспорті. Дослідження операцій в транспортних системах [Гаврилов Е. В., Дмитриченко М. Ф., Доля В. К. та ін.]; під ред. М. Ф. Дмитриченка. – К.: Знання України, 2008. – 360 с. – (5 кн. / Гаврилов Е. В. Дмитриченко М. Ф., Доля В.К. та ін.; кн. 3).
2. Хемди А. Таха. Введение в исследование операций, 7-е издание: Пер. с англ. – М.: Издательский дом "Вильямс", 2005. – 912 с.
3. Зайченко Ю. П. Дослідження операцій: Підручник. – К.: ВІПОЛ, 2000.
4. Мур Дж., Уэдерфорд Л. Экономическое моделирование в Microsoft Excel. – М.: Издательский дом "Вильямс", 2004.
5. Лопатников Л. И. Экономико-математический словарь / Словарь современной экономической науки. Издание 4-е, переработанное и дополненное. Г.: Издательство «ABF», 1996. – 704 с.
6. Четверухін Б. М., Бакуліч О. О., Радкевич С. Д. Дослідження операцій в транспортних системах. Частина 2. Системи масового обслуговування. Навчальний посібник. – К.: НТУ, 2001. – 141 с.
7. Четверухін Б. М. Дослідження операцій в транспортних системах. Частина 1. Методи лінійного програмування та їх застосування. Навчальний посібник. – К.: УТУ, 2000. – 100 с.
8. Кремер Н. Ш. Исследование операций в экономике. – М.: ЮНИТИ, 1997. – 407 с.

Навчальне видання

САНЬКО Ярослав Володимирович

Конспект лекцій
з дисципліни

«ДОСЛІДЖЕННЯ ОПЕРАЦІЙ В ТРАНСПОРТНИХ СИСТЕМАХ»

(*для студентів 3 курсу денної та заочної форм навчання за напрямом підготовки 6.070101 «Транспортні технології* (*за видами транспорту*)*»*)

Відповідальний за випуск *В. К. Доля*

Редактор *К. В. Дюкар*

Комп'ютерне верстання *К. А. Алексанян*

План 2011, поз. 209Л

Підп. до друку 13.10.2011 р. Формат 60×84/16
Друк на ризографі. Ум. друк. арк. 4,3
Зам. № Тираж 50 пр.

Видавець і виготовлювач:
Харківська національна академія міського господарства,
вул. Революції, 12, Харків, 61002
Електронна адреса: rectorat@ksame.kharkov.ua
Свідоцтво суб'єкта видавничої справи:
ДК № 4064 від 12.05.2011 р.