Рівненський державний гуманітарний університет
Факультет математики та інформатики
Кафедра вищої математики

Кваліфікаційна робота
магістерського рівня
на тему

Детерміновані моделі динамічного програмування

Виконала: студент II курсу магістратури, групи М-21
спеціальності 014 Середня освіта (Математика)
Власюк Яна Володимирівна
Керівник: доктор технічних наук, професор
Бичков Олексій Сергійович
Рецензент ___________________

Рівне - 2022 року

ЗМІСТ

ВСТУП

Сучасна наука широко використовує математику. Використання математичних моделей стало складовою частиною науки, що створює нові можливості.

У найрізноманітніших областях теоретичної та практичної діяльності часто виявляється доцільним знаходити розв’язок не відразу, а поступово крок за кроком. Таким чином, розв’язування розглядається не як одиничний акт, а як процес, що складається з кількох етапів.

Динамічне програмування визначає оптимальний розв’язок багатовимірної задачі, розбиваючи її на n етапів, кожен з яких містить підзадачу з однією змінною. Перевага даного підходу полягає в тому, що в процесі розв’язування більшої багатовимірної задачі займаємося розв’язуванням одновимірних задач, що набагато простіше. Оскільки природа кожного етапу розв'язання залежить від конкретної оптимізаційної задачі, динамічне програмування не рекомендує безпосередньо для кожного етапу обчислювальних алгоритмів. Обчислювальні аспекти проектуються і реалізовуються окремо на кожному етапі розв’язування оптимізаційних під задач.

Актуальність обраної теми зумовлена тим що, детерміновані моделі динамічного програмування можна застосовувати для розв'язування практичних задач, таких як: задача про загрузку, задача планування робочої сили, задача заміни обладнання, задача інвестування, моделі керування запасами та інші

Метою роботи є аналіз детермінованих моделей динамічного програмування і застосування їх при розв’язуванні задач динамічного програмування; виявлення найкращого способу дії під час розв’язування тієї чи іншої задачі.

Об'єктом дослідження виступає процес побудови детермінованих моделей динамічного програмування.

Предметом дослідження є особливості застосування теорії детермінованих моделей динамічного програмування під час розв’язування задач.

Таким чином можна сформулювати основні завдання даної роботи:

1) розглянути загальні відомості що пов'язані із динамічним програмуванням;

2) розглянути основні поняття, пов'язанні з детермінованими моделями динамічного програмування, проаналізувати елементи моделей динамічного програмування;

3) проаналізувати використання детермінованих моделей динамічного програмування при розв'язуванні задач;

4) розробити методичні матеріали для розв'язування задач, що приводять до детермінованих моделей динамічного програмування.

В роботі систематично викладено теорію застосування детермінованих моделей динамічного програмування до знаходження розв'язку задач.

Робота складається з трьох розділів. У першому розділі розкрито основні поняття динамічного програмування; розглянуто приклад побудови обчислювальної схеми та побудови моделі динамічного програмування. Також в даному розділі міститься опис побудови обчислювальної схеми та процесу моделювання динамічного програмування. У другому розділі йдеться про рекурентну природу обчислень динамічного програмування, алгоритм прямої та зворотної прогонки. Також в даному розділі розглянуто деякі застосування динамічного програмування. У третьому розділі демонструється використання детермінованих моделей динамічного програмування при розв'язуванні задач.

В ході роботи вивчено теоретичні основи питання, що вивчається, підібрано та розв'язано приклади знаходження розв'язку задач.

Для написання роботи використовувалися матеріали з навчальної літератури в списку використаної літератури.

Апробація роботи: основні положення магістерської роботи доповідались на XV Всеукраїнській науково-практичній конференції здобувачів вищої освіти та молодих учених РДГУ у травні 2022 році.

РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ДИНАМІЧНОГО ПРОГРАМУВАННЯ

1.1. Модель динамічного програмування

Динамічне програмування — метод оптимізації, який пристосований до операцій, в яких процес прийняття рішень може бути розбитим на окремі етапи (кроки). Такі операції називаються багатоетапними.

Як розділ математичного програмування, динамічне програмування почало розвиватися в 50-х роках XX ст. завдяки працям Р. Беллмана та його співробітників. Спочатку цим методом розв'язувалися задачі оптимального керування запасами, потім клас задач значно розширився. Як практичний метод оптимізації, метод динамічного програмування став можливим лише при використанні сучасної обчислювальної техніки [1, с.102].

Принцип оптимальності, сформульований Беллманом, лежить в основі методу динамічного програмування. Цей принцип і ідея включення конкретної задачі оптимізації в сімейство аналогічних багатокрокових задач призводять до рекурентних співвідношень — функціональних рівнянь — відносно оптимального значення цільової функції. Їх розв'язок дозволяє послідовно отримати оптимальне керування для вихідної задачі оптимізації.

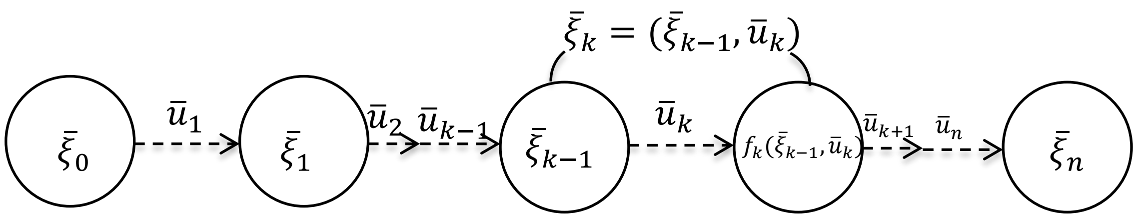

Рис. 1.1

Розглянемо загальний опис моделі динамічного програмування.

Аналізується керована система, яка під впливом керування переходить з початкового стану $\bar{\xi}_0$ в кінцевий стан $\bar{\xi}_n$. Припустимо, що можна розбити на n кроків процес керування системою. Нехай $\bar{\xi}_1, \bar{\xi}_2, \dots, \bar{\xi}_n$ — стан системи після першого, другого, ... , $n-$ го кроку. Схематично показано на рис. 1. 1.

Стан $\bar{\xi}_k$ системи після $k -$го кроку $(k = 1, 2, \dots, n)$ характеризується параметрами $\xi_k^{(1)}, \xi_k^{(2)}, \dots, \xi_k^{(s)}$, що називаються фазовими координатами. Стан $\bar{\xi}_k$ можна зобразити за допомогою точки s-вимірного простору, який називають фазовим. Послідовне перетворення системи (по кроках) відбувається за допомогою окремих заходів $\bar{u}_1, \bar{u}_2, \dots, \bar{u}_n$, які складають керування системою

$$U = (\bar{u}_1, \bar{u}_2, \dots, \bar{u}_n),$$

де $\bar{u}_k$ — керування на $k -$ му кроці, переводить систему від стану $\bar{\xi}_{k-1}$ у стан $\bar{\xi}_k$ (рис. 1.1). Керування $\bar{u}_k$ на $k -$ му кроці зводиться до вибору значень визначених керуючих змінних ${u_k}^{(1)}, {u_k}^{(2)}, \dots, {u_k}^{(n)}$.

Нехай стан системи в кінці $k -$го кроку залежить лише від попереднього стану системи $\bar{\xi}_{k-1}$ та керування $\bar{u}_k$ на даному кроці (рис. 1.1). Така властивість має назву відсутність післядії. Позначимо дану залежність у вигляді

$$\bar{\xi}_k = F_k(\bar{\xi}_{k-1}, \bar{u}_k), \tag{1.1}$$

Рівність (1.1) має назву рівняння стану. Функції $F_k(\bar{\xi}_{k-1}, \bar{u}_k)$ вважаємо заданими.

Варіюючи рівняння U, маємо різну «ефективність» процесу, яка буде оцінюватися кількісно цільовою функцією Z. Вона залежить від вибраного керування U та від початкового стану системи $\bar{\xi}_0$:

$$Z = \Phi(\bar{\xi}_0, U). \tag{1.2}$$

Показник ефективності процесу керування $k -$го кроку, який залежить від стану $\bar{\xi}_{k-1}$ та керування $\bar{u}_k$, вибраного на початку цього кроку, позначимо через $f_k(\bar{\xi}_{k-1}, \bar{u}_k)$ (рис. 1.1). У задачі поетапної оптимізації цільова функція (1.2) повинна бути адитивною, а саме

$$Z = \sum_{k=1}^{n} f_k(\bar{\xi}_{k-1}, \bar{u}_k). \tag{1.3}$$

Якщо не виконується властивість адитивності цільової функції Z, то її виконання можна досягти певними перетвореннями функції. Наприклад, якщо $Z -$ мультиплікативна функція, що задана у вигляді

$$Z = \prod_{k=1}^{n} f_k(\bar{\xi}_{k-1}, \bar{u}_k),$$

тоді можна розглянути функцію $Z' = \log Z$, яка є адитивною.

Як правило, умовами процесу на керування накладаються деякі обмеження на кожному кроці $\bar{u}_k$. Керування, що задовольняють дані обмеження, називається допустимими.

Можна сформулювати задачу покрокової оптимізації так: визначити сукупність допустимих керувань $\bar{u}_1, \bar{u}_2, \dots, \bar{u}_n$, що переводять систему із початкового стану $\bar{\xi}_0$ в кінцевий стан $\bar{\xi}_n$ і максимізують або мінімізують показник ефективності [17, с.6].

Для однаковості формулювань (але не обчислювальних процедур) надалі говоритимемо лише про задачу максимізації, маючи на увазі, що якщо необхідно мінімізувати Z, то, замінивши Z на $Z' = -Z$, перейдемо до максимізації Z' [5, с.247].

Початковий стан $\bar{\xi}_0$ та кінцевий стан $\bar{\xi}_n$ можуть бути задані однозначно або можуть бути вказані множиною Ω_0 початкових станів і множиною Ω_n кінцевих станів так, що $\bar{\xi}_0 \in \Omega_0, \bar{\xi}_n \in \Omega_n$. В останньому випадку в задачі покрокової оптимізації потрібно визначити сукупність допустимих керувань, що переводять систему з початкового стану $\bar{\xi}_0 \in \Omega_0$ в кінцевий стан $\bar{\xi}_n \in \Omega_n$ і максимізують цільову функцію (1.3). Керування, при якому досягається максимум цільової функції (1.3), називається оптимальним керуванням і позначається через $U^* = (\bar{u}_1^*, \bar{u}_2^*, \dots, \bar{u}_n^*)$.

Дискретною називається модель динамічного програмування, якщо змінні керування $\bar{u}_k$ набувають дискретних значень. Неперервною називається модель динамічного програмування, якщо змінні змінюються неперервно. Залежно від числа керуючих змінних на кожному кроці (r) і від числа параметрів станів (s) розрізняють багатовимірні та одновимірні моделі динамічного програмування. У задачі число кроків може або бути нескінченним, або скінченним.

Динамічне програмування застосовується при оптимізації як детермінованих, так і стохастичних процесів.

У деяких задачах, що розв'язуються з використанням методу динамічного програмування, процес керування розбивається на кроки. Наприклад, при розподілі ресурсів діяльності підприємства на кілька років кроком вважається часовий період; при розподілі коштів між n підприємствами номером кроку вважається номер чергового підприємства. В інших задачах розбиття на кроки вводиться штучно . Наприклад, неперервний керований процес можна розглядати як дискретний, умовно розбивши його на деякі часові відрізки – кроки. Виходячи з умов кожної конкретної задачі, довжину кроку вибирають таким чином, щоб на кожному кроці отримати просту задачу оптимізації та забезпечити необхідну точність обчислень [14, с.161].

1.2. Рівняння Беллмана. Принцип оптимальності

Метод динамічного програмування полягає в тому, що оптимальне керування будується крок за кроком. На кожному кроці оптимізується керування тільки цього кроку. Водночас на кожному кроці керування вибирається з урахуванням наслідків, оскільки керування, яке оптимізує цільову функцію лише для даного кроку, може призвести до неоптимального ефекту всього процесу. Керування на кожному кроці має бути оптимальним з точки погляду процесу в цілому [7, с.489].

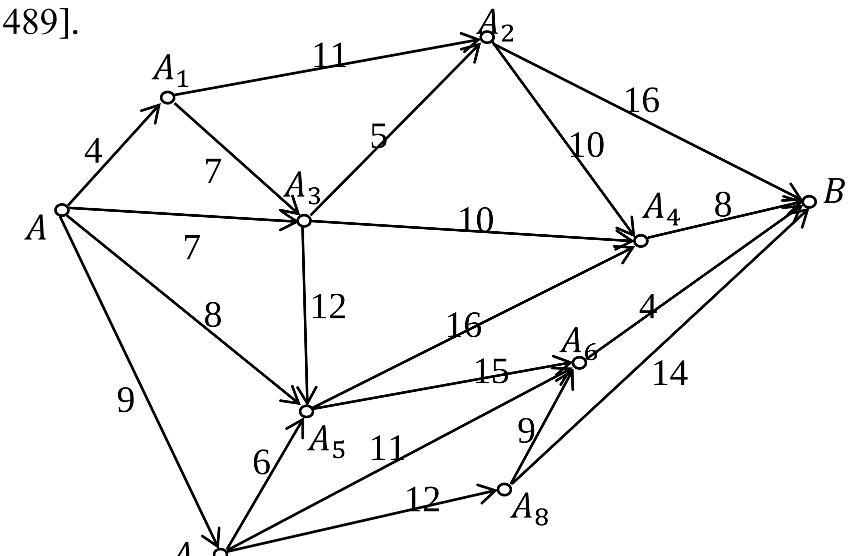

Рис.1.2

До сказаного вище ілюстрацією може служити задача про вибір найкоротшого шляху для переходу від точки A до точки B, якщо маршрут повинен пройти через деякі пункти. Пункти на рис. 1.2 позначені колами, а відрізками, що з'єднують їх - дороги, поряд з якими проставлені відповідні відстані.

З погляду інтересів оптимізації тільки кожного найближчого кроку − вибору найкоротшого шляху з даної точки до сусідньої − необхідно рухатися маршрутом, який проходить через точки A, A_1, A_3, A_2, A_4, B. Довжина якого дорівнює 34. Такий шлях з A в B не є найкоротшим. Наприклад, маршрут, який проходить через точки A, A_3, A_4, B має меншу довжину, рівну 25 [5, с.250].

Розв'язавши дану задачу, можна переконатися, що оптимальним другий шлях також не буде.

Розглянутий приклад багатокрокової операції показує, що на кожному кроці треба вибирати керування з урахуванням його впливу на наступні кроки. Це основне правило сформульоване Р. Беллманом у динамічному програмуванні, яке називається принципом оптимальності.

Оптимальне керування має таку властивість, що хоч би який був початковий стан на будь - якому кроці та керування, обране на цьому кроці, наступні керування повинні вибиратися оптимальними щодо стану, до якого прийде система в кінці даного кроку [5, с.251].

Використання даного принципу гарантує, що керування, яке обране на будь-якому кроці, є кращим з погляду на процес в цілому.

Отже, якщо система на початку $k-$го кроку знаходиться у стані $\bar{\xi}_{k-1}$, а обрано довільне керування $\bar{u}_k$, то система перейде в новий стан $\bar{\xi}_k = F(\bar{\xi}_{k-1}, \bar{u}_k)$, і подальше керування $\bar{u}_{k+1}, \dots, \bar{u}_n$ повинне вибиратися оптимальним щодо стану $\bar{\xi}_k$. Це означає, що при цих керуваннях показник ефективності максимізується до кінця процесу на наступних кроках $k+1, \dots, n,$ тобто величина

$$\sum_{i=k+1}^{n} f_i(\bar{\xi}_{i-1}, \bar{u}_i).$$

Показник, який характеризує сумарну ефективність з даного $k-$го до останнього $n-$го кроку, буде позначатися через Z_k, тобто

$$Z_k = \sum_{i=k}^{n} f_i(\bar{\xi}_{i-1}, \bar{u}_i).$$

Задача оптимізації процесу, розпочинаючи з $k-$го до останнього $n-$го кроку (рис.1.3), подібна на вихідну задачу при початковому стані системи $\bar{\xi}_{k-1}$, керування $U_k = (\bar{u}_k, \dots, \bar{u}_n)$ і показником ефективності $Z_k = \Phi(\bar{\xi}_{k-1,} U_k)$. Вибравши оптимальне керування U_k^* на $n-k+1$ кроках, що залишилися, отримаємо величину $Z_k^* = \max Z_k$, яка залежить тільки від $\bar{\xi}_{k-1}$, тобто

$$Z_k^*(\bar{\xi}_{k-1}) = \max_{U_k} \Phi(\bar{\xi}_{k-1,} U_k) = \Phi(\bar{\xi}_{k-1,} {U_k}^*). \qquad (1.4)$$

Величина $Z_k^*(\bar{\xi}_{k-1})$ називається умовним максимумом. Якщо тепер вибрати на $k-$му кроці деяке довільне керування $\bar{u}_k$, то система прийде в стан $\bar{\xi}_k$. Відповідно до принципу оптимальності, яке б $\bar{u}_k$ не вибрали, на наступних кроках керування $(\bar{u}_{k+1}, \dots, \bar{u}_n)$ має вибиратися так, щоб показник ефективності Z_{k+1} досягав максимального значення, що дорівнює $Z_{k+1}^*(\xi_k)$. Залишається вибрати керування $\bar{u}_k$. Його не можна вибирати з умови локальної максимізації показника ефективності на даному $k-$му кроці, аби отримати $\max f_k(\bar{\xi}_{k-1}, \bar{u}_k)$. Такий підхід був би недалекоглядним, оскільки від вибору $\bar{u}_k$ залежить новий стан $\bar{\xi}_k$, а від останнього - максимально можлива ефективність, яка може бути досягнута в подальшому, тобто величина $Z_{k+1}^*(\xi_k)$. Тому необхідно вибирати керування $\bar{u}_k$ так, щоб воно в сукупності з оптимальним керуванням на наступних кроках (починаючи з $(k+1)$) призводило до загального максимуму показника ефективності на $n-k+1$ кроках, починаючи з $k-$го до закінчення. Це твердження в аналітичній формі можна записати у вигляді наступного співвідношення:

$$Z_k^*(\bar{\xi}_{k-1}) = \max_{\bar{u}_k}\{f_k(\bar{\xi}_{k-1}, \bar{u}_k) + {Z^*}_{k+1}(\bar{\xi}_k)\} \qquad (1.5)$$

що отримало назву основного функціонального рівняння динамічного програмування, або рівняння Беллмана. Схематично співвідношення (1.5) ілюструється на рис.1.3 [5, с.252].

З рівняння (1.5) отримаємо функцію $Z^*_{n-1}(\bar{\xi}_{n-2})$, якщо відома функція $Z^*_n(\bar{\xi}_{n-1})$; аналогічно можна отримати $Z^*_{n-2}(\bar{\xi}_{n-3})$, якщо знайдена $Z^*_{n-1}(\bar{\xi}_{n-2})$, і т. д., поки не визначено величину $Z^*_1(\bar{\xi}_0)$, що за визначенням представляє максимальне значення показника ефективності процесу в цілому:

$$Z^*_1(\bar{\xi}_0) = \max_U \Phi(\bar{\xi}_0, U).$$

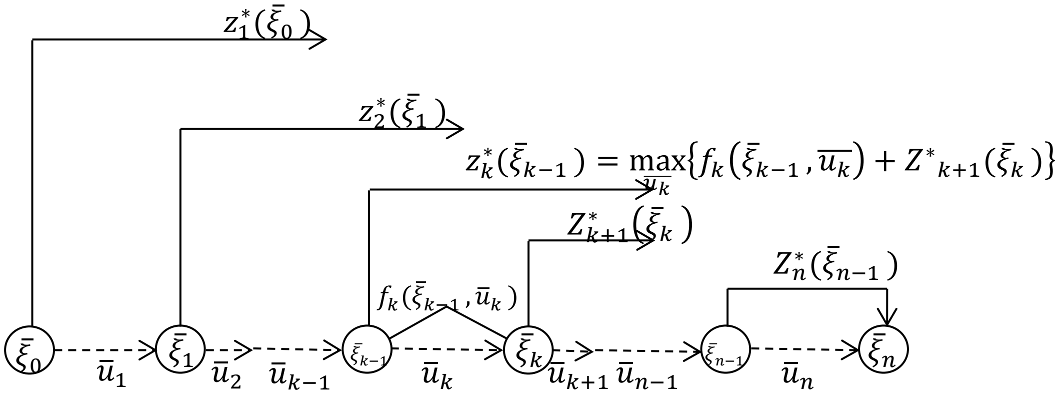

Рис.1.3

Співвідношення (1.5) для визначення послідовності функцій $Z^*_k(\bar{\xi}_{k-1})$ через $Z^*{}_{k+1}(\bar{\xi}_k)$ $(k = n, n-1, \dots, 1)$ називають основними рекурентними рівняннями Беллмана.

Розв'язуючи рівняння (1.5) визначення умовного максимуму показника ефективності за $n-k+1$ кроків, починаючи з $k-$го, визначаємо відповідне оптимальне керування $\bar{u}_k$, у якому цей максимум досягається. Це керування також залежить від $\bar{\xi}_{n-1}$. Будемо позначати таке керування через $u^*_k(\bar{\xi}_{n-1})$ і називати умовним оптимальним керуванням на k – му кроці [5, с.254].

Головне значення рівняння (1.5), у якому втілена ідея динамічного програмування, зводиться до того, що розв'язання вихідної задачі визначення максимуму функції (1.2) n змінних $\overline{u_1}, \overline{u_2}, \dots, \overline{u_n}$ зводиться до розв'язування послідовності n задач, що задаються співвідношеннями (1.5), кожна з яких є задачею максимізації функції однієї змінної $\bar{u}_k$. Дані задачі виявляються взаємопов'язаними, через те що у співвідношенні (1.5) при визначенні $Z_k^*(\bar{\xi}_{k-1})$ враховується знайдена під час розв’язку попередньої задачі функція $Z^*{}_{k+1}(\bar{\xi}_k)$.

1.3. Приклад побудови обчислювальної схеми та побудови моделі динамічного програмування

Перш ніж розглянути загальну обчислювальну схему динамічного програмування, потрібно розглянути конкретну задачу, на прикладі якої видно, як виконуються розрахунки за рівняннями (1.5) [6, с.13].

Задача 1. Планується розподіл початкової суми коштів ξ_0 між n підприємствами $\Pi_1, \Pi_2, \dots, \Pi_n$. Вважається, що виділені кошти x_k підприємству Π_k на початку планового періоду приносять дохід $f_k(x_k)$ $(k = 1, 2, \dots, n)$. Вважатимемо, що:

1) дохід, який отриманий від вкладення коштів у підприємство Π_k, не залежить від вкладення в інші підприємства;

2) дохід, який отриманий від різних підприємств, представлений в однакових одиницях;

3) загальний дохід рівний сумі доходів, що отримані від розподілу на всіх підприємствах всіх коштів.

Визначити, скільки коштів необхідно виділити кожному підприємству, щоб був максимальний сумарний дохід.

Необхідно скласти математичну модель задачі. Загальний дохід виражається цільовою функцією

$$Z = \sum_{k=1}^{n} f_k(x_k).$$

Змінні x_k повинні задовольняти умови

$$x_1 + x_2 + \dots + x_n = \xi_0;\ x_k \geq 0 \qquad (k = 1, \dots, n).$$

Необхідно визначити змінні $x_1, \dots, x_n$, які задовольняють обмеженням і змінюють у максимум цільову функцію.

Аналогічна задача оптимізації розв'язується в класичному аналізі за допомогою множників Лагранжа. У прикладних задачах класичний метод Лагранжа, як правило, не застосовується з багатьох причин і перш за все - через розмірність. Шукати абсолютний максимум функції n змінних, навіть якщо ця функція диференційована, справа трудомістка. Якщо до того ж врахувати, що екстремум може досягатися на межі, то до дослідження стаціонарних точок усередині області додається дослідження стаціонарних точок на її межі. У практичних задачах змінні x_k можуть набувати дискретних значень (наприклад, кошти виділяються в розмірах, кратних 10 од.), а функції доходу $f_k(x_k)$ можуть бути недиференційованими або навіть заданими таблицею. У всіх цих випадках класичні методи оптимізації не застосовуються. До розв'язання задачі можна застосувати методи нелінійного програмування. Однак і вони виявляються ефективними лише при додаткових властивостях функцій $f_k(x_k)$, які на практиці часто не виконуються. Нарешті, іноді буває важливо не тільки одержати розв'язання конкретної задачі за певних ξ_0 та n, але й дослідити чутливість розв'язку до зміни цих вихідних даних, що при використанні класичних методів важко [5, с.21].

Ззазначені труднощі легко долаються шляхом методів динамічного програмування.

Перейдемо до відтворення задачі як моделі динамічного програмування. Властивість процесу розподілу між n підприємствами коштів дозволяє розглядати його як $n-$ кроковий процес. Номер підприємства візьмемо за номер $k-$го кроку, якому виділяються кошти x_k. На 1-му кроці виділяємо 1-му підприємству кошти x_1, на 2-му кроці - 2-му підприємству виділяємо кошти x_2 з тих, що залишилися і т.д. Зрозуміло, що змінні x_k $(k = 1, \dots, n)$ розглядаються як керуючі змінні. Початковий стан системи характеризується величиною ξ_0 коштів, які підлягають розподілу. Після виділення x_1 залишається $\xi_1 = \xi_0 - x_1$ коштів і т.д. Величини ξ_0, $\xi_1, \dots, \xi_n$, які на попередніх кроках після розподілу характеризують залишок коштів, будемо визначати як параметри стану. Рівності які служать рівняннями стану:

$$\xi_k = \xi_{k-1} - x_k \quad (k = 1, \dots, n). \tag{1.6}$$

Сумарний дохід за n кроків складає

$$Z = \Phi(\xi_0, U) = \sum_{k=1}^{n} f_k(x_k) \tag{1.7}$$

і є показником ефективності процесу, який має, як видно з рівності, адитивну форму.

Якщо залишок коштів дорівнює ξ_{k-1} до початку $k -$ го кроку, тоді дохід, що можна отримати на $n - k + 1$ кроках, які залишилися (тобто від розподілу коштів підприємствам $\Pi_k, \Pi_{k+1}, \dots, \Pi_n$), складе

$$Z_k = \sum_{i=k}^{n} f_i(x_i).$$

Максимальний дохід за ці $n - k + 1$ кроків залежить від того, скільки коштів залишилося від попередніх $k - 1$ кроків, тобто від величини ξ_{k-1}. Тому будемо його позначати через $Z_k^* = (\xi_{k-1})$. Очевидно, що $Z_1^*(\xi_0) = Z_{\max}$, тобто, $Z_1^*(\xi_0)$ становить сумарний максимальний дохід за n кроків (дохід, що отриманий при розподілі коштів ξ_0 між n підприємствами).

Розглянемо будь-який $k -$ й крок. Очевидно, що x_k можна вибирати з умови $0 \le x_k \le \xi_{k-1}$. Значення x_k називається допустимим, якщо воно задовольняє дану подвійну нерівність. Принцип оптимальності в даному конкретному випадку означає, якщо виділити величину x_k і отримавши від $k -$ го підприємства дохід $f_k(x_k)$, нам необхідно розпорядитися коштами, що залишилися $\xi_k = \xi_{k-1} - x_k$ найкращим чином і отримати максимальний дохід $Z_{k+1}^*(\xi_k)$ від підприємств $\Pi_{k+1}, \dots, \Pi_n$. Тоді величину x_k необхідно визначати з умов максимізації суми $f_k(x_k) + Z_{k+1}^*(\xi_k)$. Отже, отримуємо рівняння

$$Z_k^*(\xi_{k-1}) = \max_{0 \le x_k \le \xi_{k+1}} \{f_k(x_k) + Z_{k+1}^*(\xi_k)\}, \tag{1.8}$$

що має назву рівнянням Беллмана [6, с.15].

Перейдемо до схеми обчислень. Нас цікавить $Z_1^*(\xi_0)$, але якщо почати з $1 -$ го кроку, тобто з розв'язку задачі

$$Z_1^*(\xi_0) = \max_{0 \le x_1 \le \xi_0} \{f_1(x_1) + Z_2^*(\xi_1)\},$$

то необхідно знати $Z_2^*(\xi_1)$. У свою чергу, при визначенні $Z_2^*(\xi_1)$ потрібно знати $Z_3^*(\xi_2)$ і так далі. Однак є крок, за яким немає наступних. Таким є n-й крок, у якому виділяються кошти останньому підприємству Π_n. Для нього рівність (1.8) має вигляд

$$Z_{n-1}^*(\xi_{n-1}) = \max_{0 \le x_n \le \xi_{n-1}} \{f_n(x_n)\}. \quad (1.9)$$

Вважатимемо, що функція доходу $f_n(x_n)$ монотонно зростає, отже розв'язком даної задачі є умовне оптимальне керування $x_n^*(\xi_{n-1})$, при якому досягається умовний максимум $Z_n^*(\xi_{n-1}) = f_n(x_n^*)$. Отже, підприємству Π_n виділяються всі кошти що залишилися ξ_{n-1}, які приносять дохід $f_n(\xi_{n-1})$. Повернемося до попереднього, $(n-1)$ – го кроку, на початку якого є залишок коштів ξ_{n-2}. Рівняння (1.8) у цьому випадку набуде вигляду

$$Z_{n-1}^*(\xi_{n-2}) = \max_{0 \le x_{n-2} \le \xi_{n-2}} \{f_{n-1}(x_{n-1}) + Z_n^*(\xi_{n-1})\}.$$

Тут оптимальний вибір x_{n-1} не такий очевидний, як при розв'язуванні попередньої задачі (1.9). Спочатку, виразимо з рівняння стану ξ_{n-1} через $\xi_{n-2} - -x_{n-1}$, отримаємо

$$Z_{n-1}^*(\xi_{n-2}) = \max_{0 \le x_{n-1} \le \xi_{n-2}} \{f_{n-1}(x_{n-1}) + Z_n^*(\xi_{n-2} - x_{n-1},)\}. \quad (1.10)$$

Обидва доданки у фігурних дужках - це відомі функції, які залежать від керуючої змінної x_{n-1}. Параметр ξ_{n-2} є початковим станом для задачі. Здійснивши дослідження на максимум функції $Z_{n-1}(x_{n-1},\xi_{n-2}) = f_{n-1}(x_{n-1}) + +Z_n^*(\xi_{n-2} - x_{n-1},)$ від однієї змінної x_{n-1}, отримаємо умовне оптимальне керування $x_{n-1}^*(\xi_{n-2})$ та відповідний умовний максимум сумарного доходу $Z_{n-1}^*(\xi_{n-2})$. На мові даної задачі цей розв'язок означає, що якщо перед виділенням коштів підприємству Π_{n-1} в нашому розпорядженні є залишок ξ_{n-2}, то підприємству Π_{n-1} необхідно виділити $x_{n-1}^*(\xi_{n-2})$ коштів. При цьому сума доходів від підприємств Π_{n-1} та Π_n досягає максимуму.

Завершивши розв'язок задачі (1.10), приступимо до наступного з кінця $(n-2)-$го кроку, аналогічно знайдемо умовне оптимальне керування $x_{n-2}^{*}(\xi_{n-3})$ і відповідний залишку ξ_{n-3} умовний максимум $Z_{n-2}^{*}(\xi_{n-3})$ і т.д.

Отже, пройшовши послідовно всі кроки процесу розподілу з кінця до його початку (тобто до 1-го кроку), маємо дві послідовності функцій:

$$Z_n^{*}(\xi_{n-1}), Z_{n-1}^{*}(\xi_{n-2}), \dots, Z_2^{*}(\xi_1), Z_1^{*}(\xi_0)$$

(умовні максимальні доходи) та

$$x_n^{*}(\xi_{n-1}), x_{n-1}^{*}(\xi_{n-2}), \dots, x_2^{*}(\xi_1), x_1^{*}(\xi_0)$$

(умовні оптимальні керування).

На цьому завершився перший та основний етап обчислювального процесу, що отримав назву умовної оптимізації.

Далі перейдемо до другого етапу обчислювальної схеми – безумовної оптимізації. На даному етапі, спочатку, знаючи функцію $Z_1^{*}(\xi_0)$, за заданим значенням ξ_0^{*} визначаємо $Zmax = Z_1^{*}(\xi_0)$. Тепер, звертаємося до послідовності $x_k^{*}(\xi_{n-1})$ яку проходимо від початку до кінця процесу. Виділяємо $x_1^{*} = x_1^{*}(\xi_0^{*})$ 1-му підприємству; тоді для розподілу залишається $\xi_1^{*} = \xi_0^{*} - x_1^{*}$. За цією величиною визначаємо оптимальну кількість коштів $x_2^{*} = x_2^{*}(\xi_1^{*})$, що виділяються 2-му підприємству. Знову знаходимо $\xi_2^{*} = \xi_1^{*} - x_2^{*}$ після чого визначаємо x_3^{*}, і т. д., поки не буде визначено шукане оптимальне керування $(x_1^{*},\ x_2^{*}, \dots, x_n^{*})$ [24, с.32].

1.4. Числовий приклад

Розглянемо розв'язок задачі про розподіл коштів за заданих конкретних умов. На числовому прикладі буде продемонстровано загальну обчислювальну схему, розглянуту у пункті 1.3.

Задача 2. Розв'язати задачу 1 за такими даними: 1) $\xi_0 = 200$ млн. грн.; 2) $n = 4$; 3) кошти виділяються лише у розмірах, кратних 40 млн. грн.; 4) функції доходу кожному з чотирьох підприємств задані у табл. 1.1

Таблиця 1.1

x \ f	$f_1(x)$	$f_2\,(x)$	$f_3\,(x)$	$f_4\,(x)$
40	8	6	3	4
80	10	9	4	6
120	11	11	7	8
160	12	13	11	13
200	18	15	18	16

Таблиця 1.1

Умова 3) визначає дискретність задачі, а наявність на кожному кроці лише однієї змінної керування та одного параметра стану показують, що задача є одновимірною. З огляду на цю умову, можна було б прийняти за одиницю масштабу 40 млн. грн. Внаслідок цього $\xi_0 = 5$ од. і можливі значення для керуючих змінних могли б дорівнювати 0, 1, 2, 3 і 4. Однак збережемо реальні одиниці, в яких зазначені вихідні дані.

Відповідно до позначень підрозділу 1.3, маємо чотири керуючі змінні x_1, x_2, x_3, x_4 і п'ять параметрів стану $\xi_0, \xi_1, \xi_2, \xi_3, \xi_4$.

Рівняннями стану є такі рівності:

$$\xi_1 = \xi_0 - x_1; \quad \xi_2 = \xi_1 - x_2; \quad \xi_3 = \xi_2 - x_3; \quad \xi_4 = \xi_3 - x_4. \qquad (1.11)$$

Рівняння Беллмана записується у формі (1.8). Даний процес є чотирикроковим. При цьому, так як на останньому кроці (при $k = 5$) процес завершується і прибуток на наступних кроках відсутній, то і $Z_5^*(\xi_4) = 0$. Рівняння (1.8) для останнього кроку:

$$Z_4^*(\xi_3) = \max_{0 \le x_4 \le \xi_3} \{f_4(x_4)\} \qquad (1.12)$$

і для всіх попередніх ($k = 3, 2, 1$) кроків:

$$Z_k^*(\xi_k - 1) = \max_{0 \le x_k \le \xi_3} \{f_k(x_k) + Z_{k+1}^*(\xi_k)\}.$$

Вираз, що стоїть у фігурних дужках останньої рівності, позначається через

$$Z_k(\xi_{k-1}, x_k)\text{=}f_k(x_k) + Z_{k+1}^*(\xi_k). \quad (1.13)$$

де $\xi_k = \xi_{k-1} - x_k$. Тоді рівняння Беллмана для будь-якого кроку запишеться у вигляді

$$Z_k^*(\xi_{k-1}) = \max_{0 \le x_k \le \xi_{k-1}} \{Z_k(\xi_{k-1}, x_k)\}. \quad (1.14)$$

Відповідні розрахунки наводяться у двох таблицях – основній (табл. 1.2), в якій розміщуються результати умовної оптимізації, послідовність функцій $Z_k^*(\xi_{k-1})$ і $x_k^*(\xi_{k-1})$, і допоміжній (табл. 1.3), в якій визначається $Z_k(\xi_{k-1}, x_k)$ та виконується умовна оптимізація.

В основній таблиці входом є параметр ξ, для якого можливі значення 0, 40, 80, 120, 160 і 200 (див. табл. 1.2).

Умовна оптимізація починається з розрахунку 4-го кроку, для чого використовується рівняння (1.12). Так як функція доходу $f_4(x_4)$ монотонно зростає (чим більше вкладається коштів, тим більший дохід), то її максимум досягається при найбільшому значенні $x_4(\xi_3) = \xi_3$. При цьому отримаємо $Z_4^*(\xi_3) = f_4(\xi_3)$, де $0 \le \xi_3 \le 200$. Цей результат умовної оптимізації 4-го кроку розміщується безпосередньо у 2-м та 3-му стовпцях табл. 1.2, переписавши необхідні дані з останнього стовпця табл. 1.1. Умовна оптимізація 3, 2 та 1-го кроків виконується спочатку в табл. 1.3. Розрахунок ведеться за формулами (1.13) та (1.14) при $k = 3, k = 2, k = 1$.

Таблиця 1.2

	4-й крок		3-й крок		2-й крок		1-й крок	
ξ	$Z_4^*(\xi_3)$	$x_4^*(\xi_3)$	$Z_3^*(\xi_2)$	$x_3^*(\xi_2)$	$Z_2^*(\xi_1)$	$x_2^*(\xi_1)$	$Z_1^*(\xi_0)$	$x_1^*(\xi_0)$
40	4	40	4	0	6	40	8	40
80	6	80	7	40	10	40	14	40
120	8	120	9	40	13	80(40)	18	40

160	13	160	13	0	16	80	21	40
200	16	200	18	200	19	40	24	40

Роз’яснимо побудову табл. 1.3 та послідовність проведення розрахунків. Оскільки умовна оптимізація триває на всіх кроках за однотипними рівностями (1.13) і (1.14), то перші три стовпці табл.1.3 є спільними для всіх трьох кроків. Стани на початку і в кінці $k-$го кроку та керування на $k-$му кроці позначені через ξ_{k-1}, ξ_k і x_k. Пояснимо спочатку порядок заповнення цих стовпців. Якщо $\xi_{k-1}=40$, то відповідні керування можуть бути $x_k=0$ і $x_k=40$ (2-й стовпець).

$k=3,2,1$			$3-$й крок ($k=3$)			$2-$й крок ($k=2$)			$1-$й крок($k=1$)		
ξ_{k-1}	x_k	$\xi_k=\xi_{k-1}-x_k$	$f_3(x_3)$	$Z_4^*(\xi_3)$	$Z_3(\xi_2,x_3)$	$f_2(x_2)$	$Z_3^*(\xi_2)$	$Z_2(\xi_1,x_2)$	$Z_1(\xi_0,x_1)$	$Z_1(\xi_0,x_1)$	$Z_1(\xi_0,x_1)$
1	2	3	4	5	6	7	8	9	10	11	12
40	0	40	0	4	0+4=$\underline{4}$	0	4	0+4=4	0	6	0+6=6
	40	0	3	0	3+0=3	6	0	6+0=$\underline{6}$	8	0	8+0=$\underline{8}$
	0	80	0	6	0+6=6	0	7	0+7=7	0	10	0+10=10
80	40	40	3	4	3+4=$\underline{7}$	6	4	6+4=$\underline{10}$	8	6	8+6=$\underline{14}$
	80	0	4	0	4+0=4	9	0	9+0=9	10	0	10+0=10
	0	120	0	8	0+8=8	0	9	0+9=9	0	13	0+13=13
120	40	80	3	6	3+6=$\underline{9}$	6	7	6+7=$\underline{13}$	8	10	8+10=$\underline{18}$
	80	40	4	4	4+4=8	9	4	9+4=13	10	6	10+6=16
	120	0	7	0	7+0=7	11	0	11+0=11	11	0	11+0=11
	0	160	0	13	0+13=$\underline{13}$	0	13	0+13=13	0	16	0+16=16
	40	120	3	8	3+8=11	6	9	6+9=15	8	13	8+13=$\underline{21}$
160	80	80	4	6	4+6=10	9	7	9+7=$\underline{16}$	10	10	10+10=20
	120	40	7	4	7+4=11	11	4	11+4=15	11	6	11+6=17
	160	0	11	0	11+0=11	13	0	13+0=13	12	0	12+0=12
	0	200	0	16	0+16=16	0	18	0+18=18	0	19	0+19=19
	40	160	3	13	13+3=16	6	13	6+13=$\underline{19}$	8	16	8+16=$\underline{24}$
200	80	120	4	8	4+8=12	9	9	9+9=18	10	13	10+13=23
	120	80	7	6	7+6=13	11	7	11+7=18	11	10	11+10=21
	160	40	11	4	11+4=15	13	4	13+4=17	12	6	12+6=18
	200	0	18	0	18+0=$\underline{18}$	15	0	15+0=15	18	0	18+0=18

Відповідні стани в кінці кроку визначаються за допомогою рівняння стану $\xi_k=\xi_{k-1}-x_k$ і приймають відповідне значення $40-0=40$ і $40-40=0$ (3-й стовпець). Так же само послідовно заповнюються три стовпці для $\xi_{k-1}=40;80;120;160;200$.

Тепер виконуємо умову оптимізації на 3-му кроці. Використавши формули $Z_3(\xi_2,x_3)=f_3(x_3)+Z_4^*(\xi_3)$ і $Z_3^*(\xi_2)=\max\limits_{0\le x_3\le \xi_2} Z_3(\xi_2,x_3)$, які послідовно розгорнуті за рядками у 4, 5 і 6-му стовбцях. Якщо $\xi_2 = 40, x_3 = 0$ і $\xi_3 = 40$ (1-й рядок перших трьох стовпців), то отримаємо $f_3(x_3) = f_3(0) = 0$ (з табл.1.1), $Z_4^*(\xi_3) = = Z_4^*(40) = 4$ (з табл.1.2) та $Z_3(\xi_2, x_3) = 0 + 4 = 4$. Ці числа заносяться до 1-го рядка 4, 5 та 6-го стовпців. Аналогічно заповнюються всі інші рядки цих стовпців.

Порівнявши величини $Z_3(\xi_2, x_3)$ при тому самому значенні ξ_2, вибирається найбільше число, яке дорівнює величині $Z_3^*(\xi_2)$ (у табл. 1.3 це значення підкреслено). Відповідні їм умовні оптимальні керування $x_3^*(\xi_2)$ стоять у тому ж рядку табл. 1.3 (у 2-му стовпці). Виконавши умовну оптимізацію 3-го кроку, перенесемо до табл. 1.2 проти відповідних значень $\xi = \xi_2$ підкреслені значення $Z_3^*(\xi_2)$ і відповідні їм $x_3^*(\xi_2)$.

Аналогічним чином умовна оптимізація 2-го кроку виконується у всіх стовпцях табл. 1.3 по 7, 8 та 9-му рядкам.

Розглянемо, як приклад, детальний розрахунок для $\xi_1 = 160$ (четверта секція табл. 1.3). При $x_2 = 0$ і $\xi_2 = \xi_1 - x_2 = 160$ отримаємо $f_2(x_2) = f(0) = 0$, $Z_3^*(\xi_2) = = Z_3^*(160) = 13$ (з табл. 1.2), тому $Z_2(\xi_1, x_2) = Z_2(160, 0) = 13$. При $x_2 = 40$ і $\xi_2 = \xi_1 - x_2 = 160 - 40 = 120$ отримаємо $f_2(40) = 6$ (з табл. 1.1), $Z_3^*(120) = 9$ (з табл. 1.2), звідки $Z_2(160, 40) = 6 + 9 = 15$ і т.д. Після заповнення четвертої секції в 9-му стовпці отримуємо п'ять чисел: 13, 15, 16, 15 і 15, з яких найбільше $Z_2^*(160) = 16$, а відповідне $x_2^*(100) = 80$. Ці числа і занесені до основної таблиці, як результати виконання умовної оптимізації 2-го кроку при $\xi_1 = 160$.

Умовну оптимізацію 1-го кроку (10, 11 і 12- стовпці табл. 1.3) можна було б виконати лише при заданому стані ξ_0 =200 (остання секція табл. 1.3). Однак, маючи на увазі подальший аналіз, проводимо розрахунки для всіх можливих значень ξ_0 (тобто для 40, 80, 120, 160 і 200).

Переходимо до другого етапу розрахунку - безумовної оптимізації. З 1-го (останнього за порядком дій) кроку умовної оптимізації отримуємо $Z_1^*(200) = = 24$ тобто максимальний дохід, який може бути, дорівнює 24. Тут же отримуємо $x_1^*(200) = x_1^* = 40$, отже підприємству I слід виділити 40 млн. грн.

Подальші (безумовні) оптимальні керування визначаються з табл. 1.2 за наступним ланцюжком. При $x_1^* = 40$ з рівняння стану отримуємо $\xi_1^* = 200 - -40 = 160$. У відповідному (7-му) стовпці табл. 1.2 отримуємо $x_2^*(160) = 80 = = x_2^*$. Обчислюємо $\xi_2^* = 160 - 80 = 80$ (залишок коштів перед виділенням підприємству III). У 5-му стовпці табл. 1.2 знаходимо $x_3^* = x_3^*(80) = 40$. Тоді $\xi_3^* = \xi_2^* - x_3^* = 80 - 40 = 40$. Нарешті, з 3-го стовпця таблиці 2 отримуємо $x_4^* = x_4^*(40) = 40$.

Отже, максимальний дохід, що дорівнює 24 млн. грн., буде отримано, якщо розподіляти кошти між підприємствами наступним чином: підприємству I виділити $x_1^* = 40$ млн. грн., підприємству II - $x_2^* = 80$ млн. грн., підприємству III - $x_3^* = 40$ млн. грн., підприємству IV - $x_1^* = 40$ млн. грн.

Слід зауважити, що основну складність при розв'язанні задачі складає перший етап. Що стосується безумовної оптимізації, то вона в основному пов'язана з послідовним розв'язуванням рівнянь стану і тому у обчислювальному плані не викликає особливих труднощів.

Якби дискретний характер задачі не диктувався її умовами, то дискретність можна було б ввести штучно, обравши крок зміни змінних керування x_k та зберігши в іншому наведену розрахункову схему. Цей крок можна було б вибрати виходячи з необхідної точності розрахунків та умов задання функцій $f_k(x_k)$, маючи на увазі необхідність інтерполювання в табл. 1.1.

Обчислювальна схема, по суті, дає розв'язок не однієї задачі, а безлічі задач з будь-якими значеннями ξ_0 від 0 до 200. Для цього достатньо було розрахувати повністю 10, 11 і 12-й стовпці табл. 1.3. В результаті ми отримали б не єдине значення $Z_1^*(200)$, а функцію $Z_{max} = Z_1^*(\xi_0)$, яку можна використати для дослідження чутливості Z_{max} до варіації вихідного параметра ξ_0 (наприклад, визначити $\Delta Z_{max}/\Delta\xi_0$).

Якщо початкова сума зменшилася (наприклад, ξ_0=160), то використовуючи четверту секцію табл. 1.3 (стовпці 10-12), аналогічно отримаємо $Z_{max} = = Z_1^*(160) = 23$, $x^* = x_1^*(160) = 40$, а виконавши безумовну оптимізацію у тій же табл. 1.2, отримаємо $U^* = (40{,}80{,}0{,}40)$. При збільшенні ξ_0, наприклад до величини 240, необхідно на кожному кроці в табл. 1.3 додати ще одну секцію, що відповідає $\xi_0 = 240$, і результат умовної оптимізації помістити в додатковому рядку в табл. 1.2. Таким чином, варіація ξ_0 не вимагає перерахунку всієї задачі заново, а лише деякого доповнення до вже наведених обчислень.

Наведена модель динамічного програмування дозволяє також без перерахування всієї задачі врахувати зміну числа n кроків. Нехай додатково до даних нашої задачі є ще одне підприємство Π_5 із функцією доходу, заданою табл. 1.4:

Таблиця 1.4

x	40	80	120	160	200
$f_5(x)$	6	9	12	15	19

Оскільки при першій нумерації кроків, що відповідають номеру підприємства на 4-му кроці, закінчується розподіл коштів і умовна оптимізація почалася з цього кроку, добудовувати схему вправо, додаючи підприємство V, не можна: це призвело б до зміни всіх розрахунків, починаючи з останнього 4-го кроку. Тому схема добудовується шляхом розташування нового підприємства зліва і вибору для нього індексу 0. Отже нумерація кроків починається з $k = 0$ (нульового кроку), а вихідну суму коштів необхідно позначати через ξ_{-1}. Сумарний максимальний дохід за 5 - кроковий процес складає $Z_{max} = Z_0^*(\xi_{-1})$.

Так як умовна оптимізація на 1, 2, 3 і 4-му кроках вже виконана в табл. 1.3, то залишилося добудувати таблицю для виконання умовної оптимізації на 0-му кроці, причому тільки для $\xi_{-1} = 200$. Зазначені розрахунки виконані у табл. 1.5. З оптимізації 0-го кроку на етапі безумовної оптимізації отримуємо $Z_{max} = Z_0^*(200) = 27$. При цьому максимум досягається при двох оптимальних керуваннях $x_0^* = 40$ і $\tilde{x}_0^* = 80$. Прийнявши перший варіант, отримуємо послідовно $\xi_0^* = \xi_{-1}^* - x_0^* = 200 - 40 = 160$, та $x_1^* = 40$ (з табл. 1.2). Далі, $\xi_1^* = 160 - 40 = 120$, з табл.1.2 отримаємо $x_2^* = 80$ (або 40). Знову знаходимо $\xi_2^* = 120 - 80 = 40$ (або $\hat{\xi}_2^* = 120 - 40 = 80$) і відповідно $x_3^* = 0$ (або $\hat{x}_3^* = 40$). Нарешті, $\xi_3^* =$

$40 - 0 = 40$ (або $80 - 40 = 40$) і $x_4^* = 40$. Таким чином, отримано два альтернативні оптимальні керування $U_1^* = (40, 40, 80, 0, 40)$ і $U_2^* = (40, 40, 40, 40, 40)$. Аналогічно, вибираючи на 0-му кроці $\hat{x}_0^* = 80$, отримуємо третє альтернативне оптимальне керування $U_3^* = (80, 40, 40, 0, 40)$.

Таблиця 1.5

			0-й крок		
ξ_{-1}	x_0	ξ_0	$f_0(x_0)$	$Z_1^*(\xi_0)$	$Z_0(\xi_{-1}, x_0)$
1	2	3	13	14	15
200	0	200	<u>0</u>	24	0+24=24
	<u>40</u>	100	<u>6</u>	21	6+21=<u>27</u>
	<u>80</u>	<u>120</u>	9	18	9+18=<u>27</u>
	120	80	12	14	12+14=26
	160	40	15	8	15+8=23
	200	0	19	0	19+0=19

Таким чином, обчислювальний процес динамічного програмування виявляється гнучким і зручно пристосовується до зміни вихідних даних (величин ξ_0 і n) без перебудови всіх раніше виконаних розрахунків.

1.5. Загальний опис побудови обчислювальної схеми та процесу моделювання динамічного програмування

Розглянувши у двох попередніх підрозділах метод динамічного програмування на конкретних прикладах, проведемо тепер опис обчислювальної схеми в загальному випадку.

Повторимо постановку задачі, сформульованої в підрозділі 1.1. Потрібно визначити допустиме керування U деякою системою, що переводить її з початкового стану $\xi_0 \in \Omega_0$ в кінцевий стан $\xi_n \in \Omega_n$, при якому показник ефективності (цільова функція) $Z = \Phi(\xi_0, U)$ досягає найбільшого або найменшого значення.

Для того щоб загальну задачу оптимізації можна було описати моделлю динамічного програмування необхідні такі умови:

1) задача може інтерпретуватися як $n -$ кроковий процес керування, а показник ефективності процесу може бути представлений в адитивній формі, тобто як сума показників ефективності на кожному кроці;

2) структура задачі є інваріантною щодо числа кроків n, тобто повинна бути визначена для будь-якого n і не залежати від цього числа;

3) на кожному кроці стан системи визначається скінченним числом s параметрів стану і управляється скінченним числом r змінних керування, причому s і r не залежать від числа кроків n;

4) Вибір керування на $k -$ му кроці не впливає на попередні кроки, а стан на початку цього кроку є функція тільки попереднього стану та обраного на ньому керування (відсутність післядії).

Зауважимо, що виконання зазначених умов іноді виявляється очевидним, а іноді досягається після відповідних перетворень. Так, у підрозділі 1.1 зазначено, як може бути перетворений мультиплікативний показник ефективності в адитивний. Іноді наявність післядії може бути подолано введенням додаткового параметра стану. Так, наприклад, якщо дохід на $k -$ му кроці в задачі 1 залежить не тільки від виділених коштів x_k і залишку до початку кроку ξ_{k-1}, але і від деякої величини y_{k-1}, що залежить від попереднього періоду, то $\xi_k = F(\xi_{k-1}, x_k, y_{k-1})$. У такому разі будемо характеризувати стан системи після $(k - 1)$-го кроку двома параметрами стану ξ_{k-1} та $\eta_{k-1} = y_{k-1}$.

Те ж зауваження стосується розбиття процесу на окремі кроки, яке інколи випливає з властивостей процесу, а інколи вводиться штучно. Все це говорить про те, що формулювання задачі у формі моделі динамічного програмування інколи вимагає застосування штучних прийомів.

Таким чином побудова моделі динамічного програмування зводиться до наступних основних моментів:

1) обирають спосіб поділу процесу на кроки;

2) вводять параметри стану $\bar{\xi}_k = (\bar{\xi}_k^{(1)}, \bar{\xi}_k^{(2)}, \dots, \bar{\xi}_k^{(s)})$ і змінні керування $\bar{u}_k(\bar{u}_k^{(1)}, \bar{u}_k^{(2)}, \dots, \bar{u}_k^{(s)})$ на кожному кроці процесу;

3) записують рівняння стану

$$\xi_k = F(\xi_{k-1}, u_k); \tag{1.15}$$

4) вводять показники ефективності на $k-$му кроці $f_k = (\xi_{k-1}, u_k)$ і сумарний показник - цільову функцію

$$Z = \sum_{k=1}^{n} f_k(\xi_{k-1}, u_k); \tag{1.16}$$

5) водять у розгляд умовні максимуми $Z_k^*(\xi_{k-1})$ показника ефективності від $k-$го кроку (включно) до кінця процесу та умовні оптимальні керування на $k-$му кроці $u_k^*(\xi_{k-1})$;

6) з обмежень задачі визначають для кожного кроку множини D_k допустимих керувань на цьому кроці;

7) записують основні для обчислювальної схеми динамічного програмування функціональні рівняння Беллмана

$$Z_k^*(\xi_{k-1}) = \max_{u_k \in D_k} \{f_k(\xi_{k-1}, u_k) + Z_{k+1}^*(\xi_k)\} \tag{1.17}$$

і

$$Z_n^*(\xi_{n-1}) = \max_{u_n \in D_n} \{f_n(\xi_{n-1}, u_n)\}. \tag{1.18}$$

Незважаючи на однотипність у загальній побудові моделі динамічного програмування, наведеній вище, обчислювальна схема будується в залежності від розмірності задачі, характеру моделі (дискретної або неперервної), виду функцій (1.15), (1.16) та інших характеристик моделі. При всій різноманітності обчислювальних схем динамічного програмування можна відзначити в них наступні спільні риси.

1) Розв'язання рівнянь (1.17) проводять послідовно, починаючи з (1.18). Цей етап отримав назву умовної оптимізації.

2) У результаті послідовного розв'язання n частинних задач на умовний максимум визначають дві послідовності функцій: $\{Z_k^*(\xi_{k-1})\}$ - умовні максимуми та відповідні їм $\{u_k^*(\xi_{k-1})\}$ - умовні оптимальні керування.

3) Зазначені послідовності функцій у дискретних задачах отримують у табличній формі, а в неперервних моделях їх можна отримати аналітично.

4) Після виконання першого етапу (умовної оптимізації) приступають до другого етапу - безумовної оптимізації.

а) Якщо початковий стан ξ_0^* задано ($\xi_0 = \xi_0^*$), то безпосередньо визначають максимум цільової функції

$$Z_{max} = Z_1^*(\xi_0^*), \tag{1.19}$$

а потім шукане безумовне оптимальне керування по ланцюжку

$$\xi_0^* \to u_1^* - -\to \xi_1^* \to u_2^* - -\to \xi_2^* \to \cdots \to u_n^* - -\to \xi_n^*. \tag{1.20}$$

У цьому ланцюжку перехід, вказаний суцільною лінією, проводять за послідовністю $\{u_k^*(\xi_{k-1})\}$, а пунктирною - за допомогою рівнянь станів.

б) Якщо задано множину Ω_0 початкових станів, $\xi_0 \in \Omega_0$, то додатково розв’язують ще одну задачу на максимум:

$$Z_{max} = \max_{\xi_0 \in \Omega_0}\{Z_1^*(\xi_0)\}, \tag{1.21}$$

звідки знаходять ξ_0^*, а потім, як і в попередньому а), по ланцюжку (1.20) - безумовне оптимальне керування.

Іноді на етапі умовної оптимізації обчислювальний процес зручно будувати у напрямку, зворотному описаному вище, тобто від 1-го кроку до $n-$го. Цей спосіб отримав назву *прямого ходу* обчислень на відміну від вищевикладеного, який називається *зворотним ходом*. Рівняння станів для прямого ходу зручно записувати у вигляді

$$\xi_{k-1} = T_k(\xi_k, u_k) \quad (k = 1, \dots, n). \tag{1.22}$$

Вони можуть бути отримані шляхом розв’язування рівнянь (1.1) щодо ξ_{k-1}. Введемо у розгляд умовні максимуми показника ефективності за k кроків, від 1-го до $k-$го включно – величини $Z_k^*(\xi_k)$. Повторивши міркування підрозділу 1.2, прийдемо до наступної форми рівнянь Беллмана:

$$Z_k^*(\xi_k) = \max_{u_k \in D_k} \{f_k(\xi_k, u_k) + Z_{k-1}^*(\xi_{k-1})\} \quad (k = 2, \dots, n); \tag{1.23}$$

$$Z_1^*(\xi_1) = \max_{u_1 \in D_1} \{f_1(\xi_1, u_1)\}. \tag{1.24}$$

В результаті розв'язання цих рівнянь отримаємо послідовність

$$Z_1^*(\xi_1), Z_2^*(\xi_2), \dots, Z_n^*(\xi_n); \qquad u_1^*(\xi_1), u_2^*(\xi_2), \dots, u_n^*(\xi_n). \tag{1.25}$$

Етап безумовної оптимізації не відрізняється принципово від аналогічного етапу у зворотному ході обчислень: $Z_{max} = Z_n^*(\xi_n^*)$, якщо $\xi_n = \xi_n^*$ задано, або

$$Z_{max} = \max_{\xi_0 \in \Omega_0} \{Z_n^*(\xi_n)\}, \tag{1.26}$$

якщо вказано множину Ω_n можливих кінцевих станів. Далі визначаємо безумовне оптимальне керування по ланцюжку

$$\xi_n^* \to u_n^* - -\to \cdots - -\to \xi_{k-1}^* \to u_{k-1}^* - -\to \cdots - -\to \xi_1^* \to u_1^* - -\to \xi_0^*. \tag{1.27}$$

З погляду обсягу обчислень обидві схеми рівноцінні, але за деяких додаткових дослідженнях кращою стає та чи інша. Так, наприклад, додавання нових кроків зручніше проводити при використанні прямого ходу обчислень, для дослідження чутливості до варіацій ξ_0 зручніший зворотний хід, до варіацій ξ_n – прямий хід і т. д.

Наведений загальний опис обчислювальної схеми динамічного програмування має дещо абстрактний характер.

РОЗДІЛ 2. ДЕТЕРМІНОВАНІ МОДЕЛІ ДИНАМІЧНОГО ПРОГРАМУВАННЯ

2.1. Рекурентна природа обчислень динамічного програмування

Динамічне програмування визначає оптимальний розв’язок багатовимірної задачі, розбиваючи її на n етапів, кожен з яких містить підзадачу з однією змінною. Перевага даного підходу полягає в тому, що в процесі розв’язування більшої багатовимірної задачі займаємося розв’язуванням одновимірних задач, що набагато простіше. Оскільки природа кожного етапу розв'язання залежить від конкретної оптимізаційної задачі, динамічне програмування не рекомендує безпосередньо для кожного етапу обчислювальних алгоритмів. Обчислювальні аспекти проектуються і реалізовуються окремо на кожному етапі розв’язування оптимізаційних під задач.

Таким чином, обчислення в динамічному програмуванні виконується рекурентно в тому сенсі, що розв’язок однієї підзадачі використовується як вихідні дані для наступної. Оптимальний розв’язок вихідної задачі, отримують розв'язавши останню підзадачу. Спосіб виконання рекурентних обчислень залежить від того, як проводиться декомпозиція вихідної задачі. Зокрема, під задачі зазвичай зв’язані між собою спільними обмеженнями. Ці обмеження повинні враховуватися, якщо відбувається перехід від однієї під задачі до іншої.

По суті стан системи на етапі i - це інформація, що пов’язує етапи разом, при цьому оптимальні розв’язки для інших етапів, можуть бути прийняті без повторного перегляду того, як були отримані розв’язки для попередніх етапів. Дане визначення дозволяє розглядати кожен етап окремо.

Визначення стану системи веде до наступного уніфікованого положення, яке Називається принципом оптимальності. На кожному етапі оптимальна стратегія визначається незалежно від стратегій, застосованих на попередніх етапах [18, с.444].

Реалізація принципу оптимальності очевидно демонструється обчисленнями в задачі 2 розділу 1. Наприклад, на етапі 3 ми використовуємо лише найкоротші відстані до вузлів 5 і 6 і не цікавимося про те, як ці вузли досягаються з вузла 1.

У цій задачі обчислення проводилися від останнього етапу до першого. Така послідовність відома як алгоритм зворотної прогонки. Цю задачу можна розв’язати за допомогою алгоритму прямої прогонки, відповідно до якого обчислення проводяться від першого етапу до останнього [18, с.445].

Пряма і зворотна прогонка дають один і той самий розв’язок. Хоча алгоритм прямого прогону здається більш логічним, у літературі динамічного програмування використовується зворотна прогонка. Причина такої переваги полягає в тому, що загальний алгоритм зворотного прогону з точки зору обчислень може бути більш ефективним.

2.2. Деякі застосування динамічного програмування

2.2.1. Задача про завантаження

У цьому підрозділі представлено приклади, кожен з яких демонструє методи динамічного програмування. При розгляді кожного прикладу слід звертати особливу увагу на три основні елементи моделі динамічного програмування:

1) визначення етапів;

2) визначення варіантів розв’язку на кожному етапі;

3) визначення станів на кожному етапі [18, с.446].

Із перерахованих вище елементів поняття стану зазвичай найскладніше для сприйняття. Представлені в даному розділі програми показують, що визначення стану змінюється в залежності від ситуації, що моделюється. Тим не менше, при розгляді кожної програми буде корисно розглянути такі запитання:

1) які відносини пов'язують етапи?

2) Яка інформація потрібна для того, щоб отримати допустимі розв’язки на поточному етапі без повторної перевірки розв’язків, отриманих на попередніх етапах? [18, с.447]

Розуміння поняття стану вдається глибше усвідомити поставивши під сумнів визначення стану, наведене в даному розділі. Краще спробувати скористатися іншим визначенням, яке може здатися "більш логічним", і використати його в рекурентних обчисленнях. Зрештою виявиться, що визначення наведені тут у розділі забезпечують коректне розв’язання задачі. Водночас запропонований підхід має покращити розуміння самої концепції стану.

Задача про завантаження – це задача про раціональне завантаження судна (літака, автомашини тощо), яке має обмеження за об’ємом чи вантажопідйомністю. Кожен поміщений на судно вантаж приносить певний прибуток. Задача полягає у визначенні завантаження судна такими вантажами, які приносять найбільший сумарний прибуток.

Перш ніж представити співвідношення динамічного програмування, зауважимо, що задача відома в літературі як задача про спорядження, в якій пілот реактивного літака повинен визначити найбільш цінні предмети для взяття на борт літака, або як задача про рюкзак, в якому солдат (або турист) повинен визначити найцінніші предмети, що підлягають завантаженню в рюкзак. Схоже, що ці три назви були придумані для однієї і тієї ж задачі, для того, щоб забезпечити рівне представництво трьох видів збройних сил: військово-морського флоту, повітряних сил та армії.

Рекурентне рівняння зворотного прогону розроблено для загальної задачі завантаження судна вантажопідйомністю W предметами n найменувань. Нехай $m_i -$ кількість предметів $i -$ го найменування, що підлягають завантаженню, $r_i -$ прибуток, який приносить один завантажений предмет $i -$ го найменування, $w_i -$ вага одного предмета $i -$ го найменування. Загальна задача має вигляд наступної цілочислової задачі лінійного програмування [18, с.447].

Максимізувати

$$z = r_1 m_1 + r_2 m_2 + \cdots + r_n m_n$$

за умови, що

$$w_1 m_1 + w_2 m_2 + \cdots + w_n m_n \leq W,$$

$$m_1, m_2, \ldots, m_n \geq 0 \text{ і цілі.}$$

Трьома елементами моделі є

1) етап і представлений елементами предмета $i -$ го найменування, $i = 1, 2, \ldots, n;$

2) варіанти розв’язку на етапі і представлені кількістю m_i, предметів $i-$го найменування, що завантажуються. Відповідний прибуток дорівнює $r_i m_i$. Значення m_i, заключено в межах від 0 до $[W/w_i]$, $де$ $[W/w_i]-$ ціла частина числа W/w_i.

3) Стан x_i на етапі і виражає сумарну вагу предметів, рішення про завантаження яких прийнято на етапах $i, i+1, \dots, n$. Це визначення відображає той факт, що обмеження за вагою є єдиним, яке пов'язує n етапів разом.

Нехай $f_i(x_i)$ –максимальний сумарний прибуток від етапів $i, i+1, \dots, n$ при заданому стані x_i. Найпростіше рекурентне рівняння визначається за допомогою наступної двоетапної процедури.

Крок 1. Виразимо $f_i(x_i)$ як функцію $f_{i+1}(x_{i+1})-$ у такий спосіб

$$f_i(x_i) = \max_{\substack{m_i=0,1,\dots,\left[\frac{W}{W_i}\right] \\ x_i=0,1,\dots,W}} \{r_i m_i + f_{i+1}(x_{i+1})\}, \quad i = 1, 2, \dots, n,$$

де $f_{n+1}(x_{n+1}) \equiv 0$.

Крок 2. Виразимо x_{i+1} як функцію x_i що переконатися, що ліва частина є функцією лише x_i. За визначенням $x_i - x_{i+1}$ представляє вагу, що завантажена на етапі i, тобто $x_i - x_{i+1} = w_i m_i$, або $x_{i+1} = x_i - w_i m_i$. Таким чином, рекурентне рівняння подано у вигляді

$$f_i(x_i) = \max_{\substack{m_i=0,1,\dots,\left[\frac{W}{W_i}\right] \\ x_i=0,1,\dots,W}} \{r_i m_i + f_{i+1}(x_i - w_i m_i)\}, \quad i = 1, 2, \dots, n$$

Задача про завантаження вантажу представляє типову модель задачі розподілу ресурсів, у якій обмежений ресурс розподіляється між скінченною кількістю (економічних) видів діяльності. При цьому ціллю є максимізація відповідної функції прибутку. У таких моделях визначення стану на кожному етапі буде подібним до визначення задачі про завантаження вантажу. А саме, стан на етапі i - це загальна кількість ресурсу, що розподіляється на етапах $i, i+1, \dots, n$.

2.2.2. Задача планування робочої сили

У деяких будівельних проектах найм і звільнення робітників здійснюються для підтримки робочої сили, яка відповідає потребам проекту. Зважаючи на те, що як наймання, так і звільнення спричиняють додаткові витрати, потрібно визначити, як регулювати чисельність робітників протягом реалізації проекту [15, с.455].

Припустимо, що проект виконуватиметься протягом n тижнів і що мінімальна робоча сила необхідна на $i -$й тиждень, становить b_i робітників. Теоретично за ідеальних умов хотілося б використовувати протягом $i -$ го тижня рівно b_i робочих. Крім того, може бути економнішим відхилення чисельності робочої сили як в одну, так і в іншу сторону від мінімальних потреб. Враховуючи, що $x_i -$ фактична кількість працюючих протягом $i -$го тижня, то можуть виникнути витрати двох видів: 1) $C_1(x_i - b_i) -$ вартість утримання надлишкової робочої сили $x_i - b_i$ і 2) $C_2(x_i - x_{i-1}) -$ витрати на найм додаткових $x_i - x_{i-1}$ робітників.

Елементи моделі динамічного програмування визначаються наступним чином:

1) етап і представляється тижнем $i -$ го найменування, $i = 1, 2, \dots, n$;

2) варіантами розв'язку на етапі i є $x_i -$ кількість робітників за $i -$й тиждень;

3) станом на етапі i є $x_{i-1} -$ кількість робітників протягом $(i - 1) -$ го тижня (етапу).

Рекурентне рівняння динамічного програмування має вигляд

$$f_i(x_{i-1}) = \min_{x_i \geq b_i} \{C_1(x_i - b_i) + C_2(x_i - x_{i-1}) + f_{i+1}(x_i)\}, \quad i = 1, 2, \dots, n,$$

де $f_{n+1}(x_n) \equiv 0$.

Обчислення починаються на етапі n при $x_n = b_n$, і закінчуються на етапі 1.

2.2.3. Задача заміни обладнання

Чим довше механізм залишається в експлуатації, тим вища вартість обслуговування і нижча продуктивність. Коли механізм досягає певного віку, може виявитися економнішою його заміна. Таким чином, задача заміни обладнання, зводиться до визначення найбільш оптимального віку експлуатації механізму.

Припустимо, що ми вивчаємо проблему заміни механізмів протягом n років. На початку кожного року вирішуємо, чи залишити механізм в експлуатації ще один рік, чи замінити його новим. Нехай $r(t)$ і $c(t)$ представляють річний дохід, експлуатаційні витрати та t - річної машини. Далі нехай $s(t)-$ вартість продажу механізму, який експлуатувався t років. Вартість придбання нового механізму в будь-який рік залишається незмінною і дорівнює I.

Елементи моделі динамічного програмування є:

1) етап i представлений порядковим номером року $i, i = 1, 2, \dots, n$;

2) варіантами розв'язку на $i-$му етапі (тобто для $i-$го року) є альтернативи: продовжити експлуатацію або замінити механізм на початку $i-$го року;

3) станом на $i-$му етапі є термін експлуатації t (вік) механізму до початку $i-$го року [5, с.268].

Нехай $f_i(t)-$ максимальний прибуток, який одержаний за роки від i до n за умови, що на початку $i-$го року є механізм $t-$річного віку.

Рекурентне рівняння має такий вигляд

$$f_i(t) = \max\begin{Bmatrix} r(t) - c(t) + f_{i+1}(t+1), \text{якщо використовувати механізм,} \\ r(0) - s(t) - I - c(0) + f_{i+1}(1), \text{якщо замінити механізм,} \end{Bmatrix},$$

де $f_{n+1}(.) \equiv 0$.

2.2.4. Задача інвестування

Припустимо, що ви хочете інвестувати суми $P_1, P_2, \dots, P_n$ на початку кожного з наступних n років відповідно. У вас є дві можливості для інвестування у два банки: перший банк сплачує річний складний відсоток r_1, а другий банк $-r_2$. Для заохочення депозитів обидва банки виплачують бонуси новим інвесторам у вигляді відсотка від вкладеної суми. Відповідні відсотки бонусів змінюються з року в рік, та для $i-$го року становлять q_{i1} і q_{i2} у першому та другому банках відповідно. Бонуси виплачуються в кінці кожного року, протягом якого зроблено інвестицію, і можуть бути інвестовані в будь якому банку в наступному році. Це означає, що лише зазначені відсотки та нові гроші можуть бути інвестовані в один із двох банків. Однак, як тільки інвестиція внесена у банку, то вона залишається там до кінця розглянутого періоду. Розробіть стратегію інвестицій на наступні n років [5, с.269].

Елементами моделі динамічного програмування є:

1) етап i представлений номером року $i, i = 1, 2, \dots, n$;

2) варіантами розв'язку на $i -$ му етапі (для $i -$ го року) є суми I_i і $\bar{I}_i$ інвестовані в перший і другий банк;

3) станом x_i на $i -$ му етапі є кількість капіталу на початок $i -$ го року, які доступні для інвестування;

Зауважимо, що за визначенням $\bar{I}_i = x_i - I_i$. Отже,

$$x_i = P_i + q_{i-1.1}I_{i-1} + q_{i-1.2}(x_{i-1} - I_{i-1}) =$$
$$= P_i + (q_{i-1.1} - q_{i-1.2})I_{i-1} + q_{i-1.2}x_{i-1},$$

де $i = 2, 3, \dots, n, x_1 = P_1$.

Сума інвестованих грошей x_i, включає лише нові гроші плюс будь-який бонус від інвестицій, зроблені протягом $(i - 1)$-го року.

Нехай $f_i(x_i)$ – оптимальне значення суми інвестицій для інтервалу від $i -$ го до $n -$ го року задане за умови, що грошова сума x_i є на початку $i -$ го року. Далі позначимо s_i як накопичену суму на кінець $n -$ го року враховуючи, що I_i та $(x_i - I_i) -$ це інвестиції здійснені протягом $i -$ го року у перший та другий банк відповідно. Якщо позначити $\alpha_i = (1 + r_i)$, $i = 1,2$, задачу можна сформулювати так: максимізувати

$$z = s_1 + s_2 + \dots + s_n,$$

де

$$s_i = I_i\alpha_1^{n+1-i} + (x_i - I_i)\alpha_2^{n+1-i} = (\alpha_1^{n+1-i} - \alpha_2^{n+1-i})I_i + \alpha_2^{n+1-i}x_i,$$

$i = 1, 2, \dots, n - 1,$

$$s_n = (\alpha_1 + q_{n1} - \alpha_2 - q_{n2})I_n + (\alpha_2 + q_{n2})x_n.$$

Так як бонус за $n -$ й рік є частиною остаточної накопиченої грошової суми від інвестицій, у вирази для s_n додано q_{n1} і q_{n2}.

Таким чином, рекурентне рівняння в алгоритмі динамічного програмування для зворотної прогонки у цьому випадку матиме вигляд:

$$f_i(x_i) = \max_{0 \le I_i \le x_i} \{s_i + f_{i+1}(x_{i+1})\}, \qquad i = 1, 2, \dots, n-1,$$

де x_{i+1} виражено через x_i відповідно до формули, наведеної вище а $f_{n+1}(x_{n+1}) \equiv 0$.

2.3. Детерміновані моделі керування запасами

2.3.1. Загальна модель керування запасами

Як у бізнесі, так і у виробництві зазвичай прийнято підтримувати розумний запас матеріальних ресурсів або комплектуючих для забезпечення безперервності виробничого процесу. Традиційно запас сприймається як неминучі витрати, коли надто низький його рівень призводить до дорогих зупинок виробництва, а занадто високий - до "омертвіння" капіталу. Задача керування запасами - визначити рівень запасу, який врівноважує два згадані крайні випадки [7, с.583].

Основним фактором, що впливає на формулювання та розв'язання задачі, є те, що у реальному житті обсяг попиту на запас може бути або детермінованим, або імовірнісним.

Задача керування запасами передбачає періодичне розміщення та отримання замовлень заданих розмірів продукції. З цієї точки зору стратегія запасів відповідає на два питання:

1) скільки замовити?

2) Коли замовляти?

Основою для відповіді на ці запитання є мінімізація такої функції витрат на запаси:

$$\left\{\begin{matrix}\text{Сумарні}\\ \text{витрати системи}\\ \text{керування}\\ \text{запасами}\end{matrix}\right\} = \left\{\begin{matrix}\text{Витрати}\\ \text{на придбання}\end{matrix}\right\} + \left\{\begin{matrix}\text{Витрати на}\\ \text{оформлення}\\ \text{замовлення}\end{matrix}\right\} +$$

$$+\left\{\begin{matrix}\text{Витрати на}\\ \text{зберігання}\\ \text{замовлення}\end{matrix}\right\}+\left\{\begin{matrix}\text{Втрати від}\\ \text{дефіциту}\\ \text{запасу}\end{matrix}\right\}.$$

Ціни повинні бути виражені як функції шуканого обсягу замовлення, та інтервалом часу між замовленнями.

Витрати на придбання – це ціна за одиницю придбаної продукції. Ця сума може бути постійною або зі знижкою, якщо обсяг замовлення перевищив певну суму.

Витрати на оформлення – це фіксовані витрати, пов'язані з його розміщенням на інших виробництвах. Від обсягу замовлення ці витрати не залежать.

Витрати на зберігання запасу - це вартість утримання запасів на складі. Він включає відсотки на інвестований капітал, і вартість зберігання, догляду та утримання.

Втрати від нестачі запасу - це витрати, які нараховуються, коли відсутній запас необхідної продукції. Він включає потенційну втрату доходу, та більш суб'єктивну вартість, що пов'язана із втратою довіри клієнтів.

Час замовлення залежить від системи керування запасами. Система може базуватися на періодичному контролі стану запасу (наприклад, замовлення щотижня або щомісяця), у якому нові замовлення збігається з початком кожного періоду. Як альтернатива система може базуватися на постійному контролі стану запасу, де нові замовлення розміщуються коли рівень запасів падає до заздалегідь визначеного рівня, який називається точкою повторного замовлення.

2.3.2. Задачі економічного розміру замовлення

У цьому підрозділі представлена модель задачі, у якій рівень запасів періодично контролюється протягом скінченного числа рівних періодів і обсяг попиту протягом періоду хоч і є детермінованим, але водночас він динамічний, оскільки може періодично змінюватися [7, с.586].

Ситуація, в якій виникає змінний динамічний детермінований попит, називається плануванням потреб ресурсів. Ідею розв'язування такої задачі розглянемо на прикладі. Припустимо, що квартальний попит протягом наступного року для моделей M_1 та M_2 певної продукції становить 100 та 150 одиниць відповідно. Поставки квартальних партій здійснюються в кінці кожного кварталу. Термін виготовлення замовлення дорівнює 2 місяці для моделі M_1 та 1 місяць для моделі M_2. Кожна одиниця моделі M_1 та M_2 використовує 2 одиниці комплектуючих деталей S. Термін виконання комплектуючих для виробництва дорівнює одному місяцю.

На рис. 2.1 схематично зображено календарне планування виробництва моделей M_1 та M_2. Побудова плану починається з відображення у вигляді суцільних стрілок квартального попиту на дві моделі, який має місце в кінці 3-го, 6-го,-го 9 та 12-го місяців. Потім при відомих квартальних термінах пунктирні стрілки вказують початок виробництва кожної партії продукції M_1 та M_2 в 1-й та 2-й місяці.

Щоб вчасно розпочати виробництво двох моделей, необхідно щоб постачання комплектуючих S збігалося з початком виробництва M_1 та M_2. Дана інформація показана пунктирними стрілками на схемі у планах їх виробництва. Термін виробництва представлено суцільними стрілками на S −схемі, де попит на комплектуючі S становить 2 одиниці на кожну одиницю M_1 та M_2 продукції. Термін виготовлення комплектуючих рівний одному місяцю, на S −схемі пунктирні стрілки визначають план виробництва комплектуючих. З цих двох планів, можна відповідний визначити сумарний попит на S, як показано внизу на рис. 2.1. Результуючий змінний але відомий попит на комплектуючі S є типовою ситуацією, коли виникає застосування динамічної моделі економічного розміру замовлення. По суті, при вказаному змінному попиті на комплектуючі S задача зводиться до визначення обсягу виробництва на початку кожного місяця, щоб зменшити витрати виробництва та зберігання продукції.

На рис. 2.1 представлено дві моделі. Перша модель не передбачає вартість розміщення замовлення, а друга враховує дані витрати. Ця здавалося б, "маленька" деталь продовжує відповідні відмінності у складності моделей.

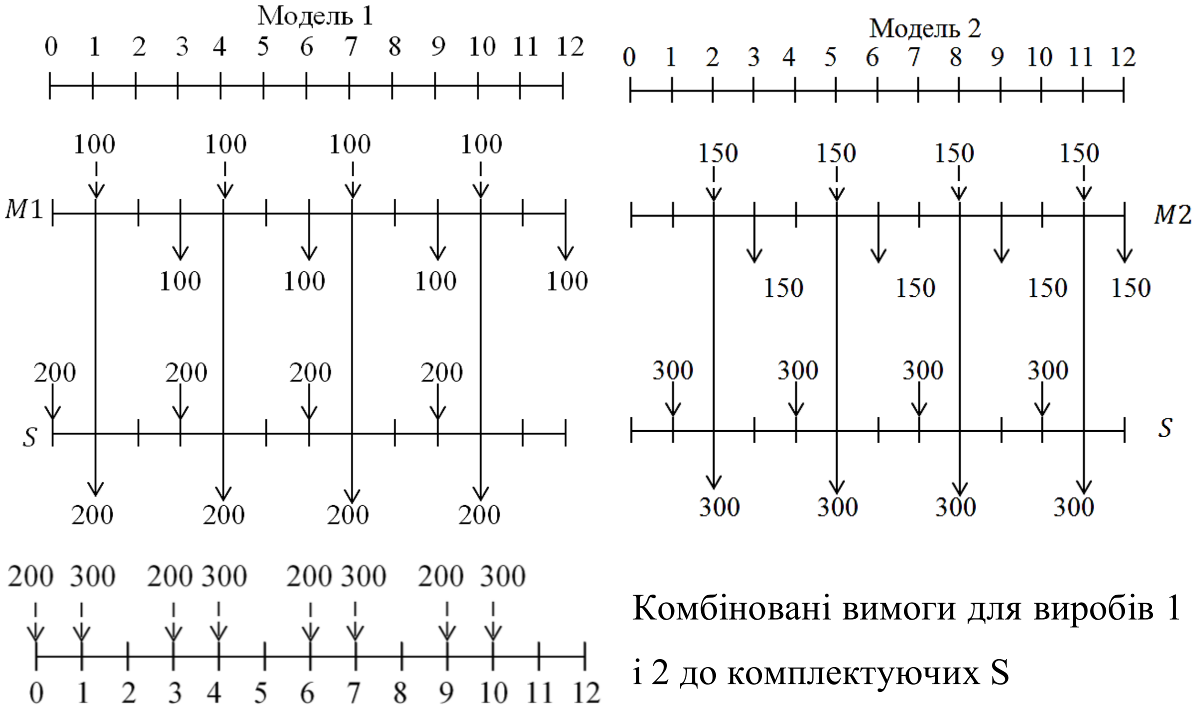

Рис. 2.1

2.3.3. Модель без витрат на оформлення замовлення

Ця модель передбачає задачу яка розрахована на n рівних періодів календарного планування виробництва. Кожен період має обмеження виробничої потужності, який може включати кілька рівнів (наприклад, звичайний режим роботи та надурочна робота відповідно). У поточний період можуть вироблятися вироби для пізніших періодів, але у цьому випадку витрати на їх зберігання повинні враховуватися.

Загальні припущення моделі такі:

1) за будь-який період витрати на оформлення замовлення відсутні;

2) дефіцит не допускається;

3) собівартість одиниці продукції в будь-який період є постійна, або має зростаючі (опуклі) граничні витрати;

4) вартість зберігання одиниці продукції в будь-який період є постійною величиною.

Відсутність дефіциту означає, що виробництво у наступних періодах не може бути задоволене попитом продукції протягом поточного періоду. Для цього припущення потрібно, щоб сумарна виробнича потужність за періоди $1, 2, \ldots, i$ дорівнювала сумарному попиту за цей же час [7, с.587].

Рисунок 2.2 ілюструє, коли собівартості одиниці продукції із збільшенням рівня виробництва зростають. Наприклад звичайний режим роботи та понаднормова продуктивність відповідають двом можливим об'ємам виробництва, вартість виробництва продукції в надурочний час вища, ніж у звичайному режимі роботи.

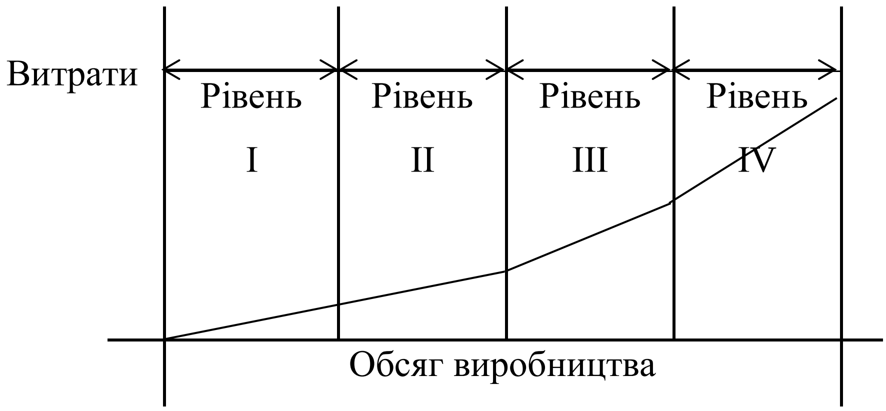

Рис. 2.2

Задача $n-$етапного планування, що розглядається, може бути сформульована як транспортна задача з kn пунктами виробництва і n споживачами, де k – кількість рівнів виробництва за період (наприклад, якщо у кожному періоді використовується основний і надурочний режими роботи, тоді $k = 2$). Виробничі потужності кожного із kn пунктів на рівні виробництва визначають обсяги поставок. Обсяги попиту уточнюються за потребою кожного періоду. Собівартість одиниці «транспортування» від пункту виробництва до місця призначення є сумою відповідних витрат виробничого процесу та витрат на зберігання одиниці продукції. Розв'язання такої транспортної задачі визначає об'єм виробництва продукції для кожного рівня, які мінімізують витрати на зберігання та виробництво.

2.3.4. Модель із витратами на оформлення замовлення

У даній моделі дефіцит не допускається, і витрати на оформлення замовлення враховуються щоразу, коли розпочато виробництво нової партії. Розглянемо точний метод динамічного програмування.

Дану задачу керування запасами схематично представлено на рис. 2.3. На ньому використано символи, для позначення наступних величин, визначених для кожного етапу $i, i = 1, 2, \dots, n$:

z_i – кількість (обсяг) замовленої продукції,

D_i – потреба в продукції (попит),

x_i – запаси на початок етапу i.

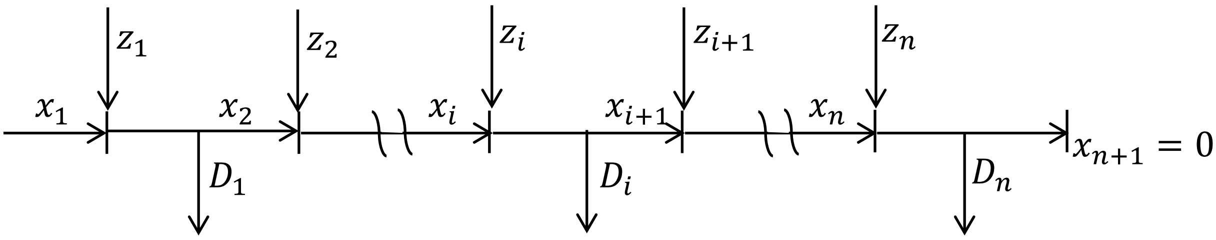

Рис. 2.3

Вартісні елементи у задачі визначаються як:

K_i – витрати на оформлення замовлення,

h_i – витрати на зберігання одиниці запасів продукції від етапу i до етапу $i + 1$.

Функція виробничих витрат задається для етапу i формулою

$$C_i(z_i) = \begin{cases} 0, & z_i = 0, \\ K_i + c_i(z_i), & z_i > 0, \end{cases}$$

де $c_i(z_i)$ є функцією граничних виробничих витрат при заданому значенні z_i.

Оскільки дефіцит не допускається, задача керування запасами зводиться до обчислення значень z_i, що мінімізують сумарні витрати, пов'язані з розміщенням замовлень, закупівлею та зберіганням продукції протягом n етапів. Витрати на зберігання на i –му етапі для простоти передбачаються пропорційними величині

$$x_{i+1} = x_i + z_i - D_i,$$

яка представляє обсяг запасу, перехідного з етапу i в етап $i + 1$.

Для процедури прямої прогонки рекурентного рівняння стан на етапі (період) i визначається як обсяг запасів x_{i+1}, на кінець періоду, де, як показано на рис. 2.3,

$$0 \le x_{i+1} \le D_{i+1} + \cdots + D_n.$$

Дана нерівність означає, що у граничному випадку запаси x_{i+1}, що залишилися, можуть задовольнити попит на всі інші періоди.

Нехай $f_i(x_{i+1})$ буде мінімальною вартістю запасів для періодів $1, 2, \dots, i$ на кінець етапу i при заданій величині запасу x_{i+1}. Таким чином рекурентне рівняння буде записано як

$$f_1(x_2) = \min_{0 \le z_1 \le D_1 + x_2} \{ C_1(z_1) + h_1 x_2 \},$$

$$f_i(x_{i+1}) = \min_{0 \le z_i \le D_i + x_{i+1}} \{ C_i(z_i) + h_i x_{i+1} + f_{i-1}(x_{i+1} + D_i - z_i) \}, \quad i = 2,3, \dots, n.$$

Модель динамічного програмування, наведену вище, можна використовувати з будь-якою функцією витрат. Однак, окремим випадком цієї моделі є така модель, в якій на етапі i як вартість закупівлі одиниці продукції, так і витрати на її зберігання -незростаючі (увігнуті) функції обсягу продукції, що закуповується та зберігається відповідно. Така ж ситуація виникає, коли функція вартості, віднесена до одиниці продукції, є постійною або коли надається оптова знижка [18, с.467].

За даних умов можна довести, що:

1) враховуючи нульовий початковий запас ($x_1 = 0$), оптимальною стратегією буде задоволення попиту для будь-якого періоду i за рахунок нового закупленого запасу, або продукції, але не від обох джерел, тобто $z_i x_i = 0$. (Для випадку з позитивним початковим рівнем запасу ($x_1 > 0$) ця сума може бути списана з вимог наступних періодів поки не вичерпається)

2) оптимальна кількість виробництва z_i на будь-якому періоді i повинна або дорівнювати нулю, або задовольняти точний попит протягом одного або кількох наступних періодів.

Використання зазначених вище двох властивостей з рекурентним рівнянням для алгоритму прямого прогону динамічного програмування дозволяє спростити схему обчислень.

2.4. Проблема розмірності

У всіх задачах динамічного програмування, які були представлені, стан на будь-якому етапі представлено єдиною змінною. Наприклад, у задачі рюкзака єдиним обмеженням є вага предмета. Більш реалістично, що об'єм рюкзака також може бути іншою обмежувальною величиною. У такому випадку стан на будь-якому етапі системи називається двовимірним, оскільки він формується з двох змінних: ваги та об'єму.

Збільшення кількості змінних стану збільшує кількість обчислень на кожному етапі. Це особливо чітко видно в табличних обчисленнях динамічного програмування, оскільки кількість рядків у кожній таблиці відповідає всім можливим комбінаціям значень змінних стану [18, с.466].

Цю обчислювальну складність іноді називають у літературі прокляттям розмірності.

Наступний приклад вибрано для демонстрації прокляття розмірності. Він також служить для демонстрації зв'язку між лінійним і динамічним програмування.

Приклад 1. Підприємство виробляє два види продукції. Добова продуктивність виробничого процесу складає 430 хв. Продукт першого виду вимагає 2 хвилини, а продукт другого виду – 1 хвилину. Немає обмежень на кількість виробленого продукту першого виду, але максимальна щоденна потреба в другому продукті становить 230 одиниць. Прибуток від одиниці продукції першого виду становить 2 долари, а другого - 5 доларів. Знайдіть оптимальний розв'язок за методами динамічного програмування.

Дана задача є такою задачею лінійною програмування: максимізувати

$$z = 2x_1 + 5x_2$$

при обмеженнях

$$2x_1 + x_2 \leq 430,$$

$$x_2 \leq 230,$$

$$x_1, x_2 \geq 0.$$

Елементами моделі динамічного програмування є:

1) етап i відповідає продукту $i, i = 1, 2$;

2) альтернативою x_i на $i-$му етапі є кількість виробництва продукції $i, i = 1, 2$;

3) стан (v_1, w_1) представляє кількість ресурсів, необхідну для виробництва продукції виду 1 і 2 (виробничий час і обмеження на попит) і використовується на етапах 1 і 2;

4) стан (v_2, w_2) представляє кількість ресурсів, необхідну для виробництва продукції виду 1 і 2 (виробничий час та обмеження на попит) і використовується на етапі 2.

Етап 2.

Нехай $f_2(v_2, w_2)$ – це максимальний прибуток для етапу 2 (продукція виду 2) для стану (v_2, w_2). Тоді

$$f_2(v_2, w_2) = \max_{\substack{0 \le x_2 \le v_2 \\ 0 \le x_2 \le w_2}} \{5x_2\}$$

Отже, $\max\{5x_2\}$ має місце при $x_2 = \min\{v_2, w_2\}$. Звідси маємо наступне розв'язання для 2-го етапу

Стан	Оптимальний розв'язок	
	$f_2(v_2, w_2)$	x_2
(v_2, w_2)	$5\min\{v_2, w_2\}$	$\min\{v_2, w_2\}$

Етап 1.

$$f_1(v_1, w_1) = \max_{0 \le 2x_1 \le v_1} \{2x_1 + f_2(v_1 - 2x_1, w_1)\} = \max_{0 \le 2x_1 \le v_1} \{2x_1 + 5\min(v_1 - 2x_1, w_1)\}.$$

Оптимізація на першому етапі вимагає розв'язання задачі, що у загальному випадку є досить складною. Для цієї задачі ми встановлюємо $v_1 = 430$ і $w_1 = 230$, що дає $0 \le 2x_1 \le 430$. Оскільки $\min(430 - 2x_1, 230)$ є нижньою обвідною двох прямих, що перетинаються то

$$\min(430 - 2x_1, 230) = \begin{cases} 230, & 0 \le x_1 \le 100, \\ 430 - 2x_1, & 100 \le x_1 \le 215. \end{cases}$$

$$f_1(430,230) = \max_{x_1} \begin{cases} 2x_1 + 1150, & 0 \le x_1 \le 100, \\ -8x_1 + 2150, & 100 \le x_1 \le 215. \end{cases}$$

Можна графічно перевірити, що максимальне значення $f_1(430,230)$ виникає при $x_1 = 100$. Таким чином, отримуємо

Стан	Оптимальний розв’язок	
	$f_1(v_1, w_1)$	x_1
$(430,230)$	1350	100

Для визначення оптимального значення x_1 зазначаємо, що

$$v_2 = v_1 - 2x_1 = 430 - 200 = 230,$$

$$w_2 = w_1 - 0 = 230.$$

Отже,

$$x_2 = \min\{v_2, w_2\} = 230.$$

Оптимальний розв’язок має вигляд: $x_1 = 100$ одиниць, $x_2 = 230$ одиниць, $z = 1350$ доларів.

РОЗДІЛ 3. ВИКОРИСТАННЯ ДЕТЕРМІНОВАНИХ МОДЕЛЕЙ ДИНАМІЧНОГО ПРОГРАМУВАННЯ ПРИ РОЗВ'ЯЗУВАННІ ЗАДАЧ

3.1. Задача про завантаження

4-тонне судно можна завантажити предметами трьох найменувань. У таблиці 3.1 наведено дані про вагу одного предмета w_i (у тоннах) і дохід від одного завантаженого предмета r_i (у тисячах доларів. Як має бути завантажене судно, щоб отримати максимальний прибуток?

Таблиця 3.1

Предмет i	w_i	r_i
1	2	31
2	3	47
3	1	14

Оскільки вага одного предмета w_i та максимальна вага W є цілі значення, стан x_i приймає лише цілі значення.

Етап 3. Точна вага, що буде завантажена на етапі 3, заздалегідь невідома, але може приймати одне зі значень 0, 1, …, 4 (оскільки $\mathrm{W} = 4$ тонни). Стани $x_3 = 0$ і $x_3 = 4$ відповідно, являють собою крайні випадки, коли товари третього найменування взагалі не відправляються, а для нього призначається ціле судно. Решта значень $x_3(= 1, 2 \text{ або } 3)$ означають часткове завантаження судна предметами третього найменування. По суті, заданий діапазон значень для x_3 охоплює всі ємності судна, що залишилися, їх можна заповнити предметами третього найменування.

Враховуючи що вага w_3 одного предмета дорівнює 1 тонну, максимальна кількість одиниць третього типу, яку можна завантажити, дорівнює [4/1] = 4, що означає, що можливими значеннями x_3 є 0, 1, 2, 3 і 4. Таким чином, розв'язок m_3 є допустимим за умови, що $w_3 m_3 \leq x_3$. Отже, всі неприпустимі альтернативи (ті, для яких $w_3 m_3 > x_3$) виключаються. Основою для порівняння альтернатив етапу 3 є наступне рівняння

$$f_3(x_3) = \max_{m_3}\{14m_3\}, \quad \max\{m_3\} = \left[\frac{4}{1}\right] = 4.$$

У таблиці 3.2 порівнюються допустимі розв’язання для кожного значення x_3.

Таблиця 3.2

x_3	m_3					Оптимальний розв’язок	
	$m_3 = 0$	$m_3 = 1$	$m_3 = 2$	$m_3 = 3$	$m_3 = 4$	$f_3(x_3)$	m_3^*
0	0	-	-	-	-	0	0
1	0	14	-	-	-	14	1
2	0	14	28	-	-	28	2
3	0	14	28	42	-	42	3
4	0	14	28	42	56	56	4

Етап 2.

$$f_2(x_2) = \max_{m_2}\{47m_2 + f_3(x_2 - 3m_2)\}, \quad \max\{m_2\} = \left[\frac{4}{3}\right] = 1.$$

x_2	$47m_2 + f_3(x_2 - 3m_2)$		Оптимальний розв’язок	
	$m_2 = 0$	$m_2 = 1$	$f_2(x_2)$	m_2^*
0	0+0=0	-	0	0
1	0+14=14	-	14	0
2	0+28=28	-	28	0
3	0+42=42	47+0=47	47	1
4	0+56=56	47+14=61	61	1

Етап 1.

$$f_1(x_1) = \max_{m_1}\{31m_1 + f_2(x_1 - 2m_1)\}, \quad \max\{m_1\} = \left[\frac{4}{2}\right] = 2.$$

x_1	$31m_1 + f_2(x_1 - 2m_1)$			Оптимальний розв’язок	
	$m_1 = 0$	$m_1 = 1$	$m_1 = 2$	$f_1(x_1)$	m_1^*
0	0+0=0	-	-	0	0
1	0+14=14	-	-	14	0
2	0+28=28	31+0=31	-	31	1
3	0+47=47	31+14=45	-	47	0

4	0+61=61	31+28=59	62+0=62	62	2

Оптимальний розв'язок визначається таким чином. Якщо $W = 4$ тонни, з першого етапу розв'язання задачі при $x_1 = 4$ дає оптимальний розв'язок $m_1^* = 2$, що означає, що завантажені на судно будуть два предмети першого найменування. Цей розподіл залишає $x_2 = x_1 - 2m_1^* = 4 - 2 \times 2 = 0$. Розв'язання на другому етапі при $x_2 = 0$ дає $m_2^* = 0$, що, у свою чергу, дає $x_3 = x_2 - 3m_2^* = 0 - 3 \times 0 = 0$. Далі, починаючи з етапу 3 при $x_3 = 0$ призводить до $m_3^* = 0$. Таким чином, оптимальним розв'язком задачі є $m_1^* = 2, m_2^* = 0$ і $m_3^* = 0$. Відповідний прибуток дорівнює 62 000 доларів.

У таблиці для першого етапу нам фактично потрібно отримати оптимальний розв'язок $x_1 = 4$ лише тому, що це останній етап, який потрібно розглянути. Однак обчислення для $x_1 = 0, 1, 2$ і 3 включені до таблиці, щоб дозволити провести аналіз чутливості розв'язання. Наприклад, що станеться, якщо місткість судна становитиме 3 тонни замість 4? Новий оптимальний розв'язок може бути визначено як, починаючи з $x_1 = 3$ на першому етапі і продовжуючи так, як було зроблено при $x_1 = 4$.

3.2. Задача планування робочої сили

Будівельний підрядник оцінює, що кількість робочої сили, необхідної протягом наступних 5 тижнів, становитиме 5, 7, 8,4 і 6 робітників відповідно. Надлишкова робоча сила, яка утримується в штаті, коштуватиме 300 доларів США на одного працівника на тиждень, а нове наймання кожного тижня матиме фіксовану вартість 400 доларів США плюс 200 доларів США на працівника на тиждень.

Виражаючи C_1 і C_2 у сотнях доларів, маємо

$$b_1 = 5, b_2 = 7, b_3 = 8, b_4 = 4, b_5 = 6,$$

$$C_1(x_i - b_i) = 3(x_i - b_i), x_i > b_i, i = 1, 2, \dots, 5,$$

$$C_2(x_i - x_{i-1}) = 4 + 2(x_i - x_{i-1}), x_i > x_{i-1}, i = 1, 2, \dots, 5.$$

Етап 5. $(b_5 = 6)$.

x_4	$C_1(x_5-6)+C_2(x_5-x_4)$	Оптимальний розв'язок	
	$x_5=6$	$f_5(x_4)$	x_5^*
4	3(0)+4+2(2)=8	8	6
5	3(0)+4+2(1)=6	6	6
6	3(0)+0=0	0	6

Етап 4. $(b_4=4)$.

x_3	$C_1(x_4-4)+C_2(x_4-x_3)+f_5(x_4)$			Оптимальний розв'язок	
	$x_4=4$	$x_4=5$	$x_4=6$	$f_4(x_3)$	x_4^*
	3(0)+0+8=8	3(1)+0+6=9	3(2)+0+0=6	6	6

Етап 3. $(b_3=8)$.

x_2	$C_1(x_3-8)+C_2(x_3-x_2)+f_4(x_3)$	Оптимальний розв'язок	
	$x_3=8$	$f_3(x_2)$	x_3^*
	3(0)+4+2(1)+6=12	12	8
	3(0)+0+6=6	6	8

Етап 2. $(b_2=7)$.

x_1	$C_1(x_2-7)+C_2(x_3-x_2)+f_3(x_2)$		Оптимальний розв'язок	
	$x_2=7$	$x_2=8$	$f_2(x_1)$	x_2^*
	3(0)+4+2(2)+12=20	3(1)+4+2(3)+6=19	19	8
	3(0)+4+2(1)+12=18	3(1)+4+2(2)+6=17	17	8
	3(0)+0+12=12	3(1)+4+2(1)+6=15	12	7
	3(0)+0+12=12	3(1)+0+6=9	9	8

Етап 1. $(b_1=5)$.

x_0	$C_1(x_1-5)+C_2(x_1-x_0)+f_2(x_1)$				Оптимальний розв'язок	
	$x_1=5$	$x_1=6$	$x_1=7$	$x_1=8$	$f_1(x_0)$	x_1^*
	3(0)+4+2(5)+ +19=33	3(1)+4+2(6)+ +17=36	3(2)+4+2(7)+ +12=36	3(2)+4+2(8)+ +9=35	33	5

Оптимальний розв'язок визначається наступним чином:

$$x_0 = 0 \rightarrow x_1^* = 5 \rightarrow x_2^* = 8 \rightarrow x_3^* = 8 \rightarrow x_4^* = 6 \rightarrow x_5^* = 6.$$

Наступний план відповідає отриманому розв'язку:

Номер неділі (i)	Мінімум робочої сили b_i	Кількість фактично працюючих x_i	Розв'язання
1	5	5	Найняти 5 робітників
2	7	8	Найняти 3 робітників
3	8	8	Нічого не міняти
4	4	6	Звільнити 2 робітників
5	6	6	Нічого не міняти

3.3. Задача заміни обладнання

Компанії необхідно визначити оптимальну політику заміни поточного трирічного механізму протягом наступних 4 років $(n = 4)$. У наступній таблиці наведені дані, що відносяться до задачі. Компанія вимагає заміни 6-річного механізму. Вартість нового механізму становить 100 000 доларів.

Вік t	Прибуток $r(t)(\$)$	Вартість обслуговування $c(t)(\$)$	Залишкова вартість $s(t)(\$)$
0	20 000	200	-
1	19 000	600	80 000
2	18 500	1200	60 000
3	17 200	1500	50 000
4	15 500	1700	30 000
5	14 000	1800	10 000
6	12 200	2200	5 000

Визначення можливих значень віку механізму на кожному етапі є дещо складним завданням. На рис.3.1 представлено у вигляді сітки узагальнену задачу заміни обладнання. На початку першого року у нас є трирічний механізм. Ми можемо або замінити його (З), або залишити (С) ще на рік. На початку другого року, якщо відбудеться заміна, новому механізму буде один рік, інакше його вік буде 4 роки. Така ж логіка діє на початку кожного року, на початку з другого до четвертого.

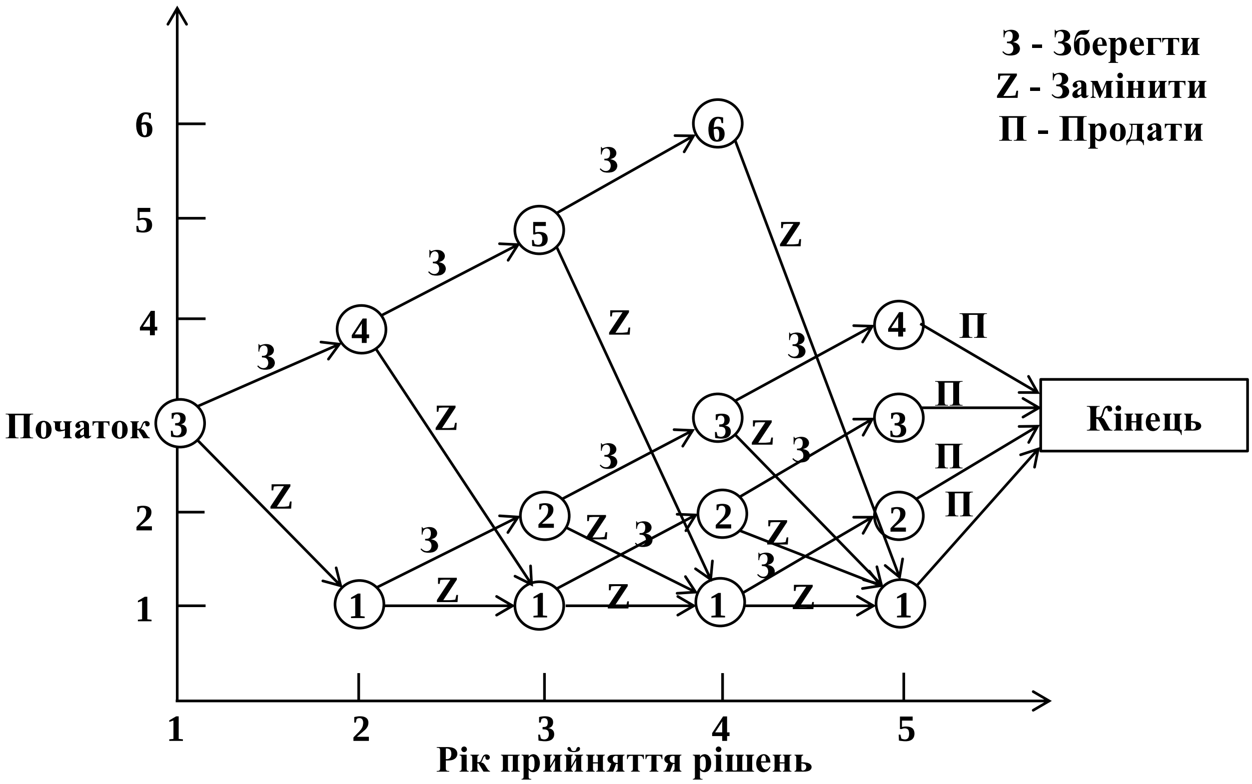

Рис. 3.1

Якщо однорічний механізм замінено на початку другого, третього або четвертого року, то механізм, що його замінив, також буде однорічним до початку наступного року. Крім того, на початку 4-го року необхідно замінити 6-річний механізм, якщо він ще експлуатується, а наприкінці 4-го року в обов'язковому порядку всі механізми продаються (П).

Мережа також показує, що на початку другого року можливий вік механізму з терміном експлуатації становить 1 або 4 роки. Для початку третього року можливий вік 1, 2 або 5 років, а для початку четвертого року можливий вік 1, 2, 3 або 6 років.

Розв’язок мережі даної задачі еквівалентно знаходженню найдовшого маршруту (тобто максимального прибутку) від початку першого року до кінця четвертого року, показаного на рис 3.1. Використано табличну форму запису при розв'язанні задачі.

Етап 4.

t	З	Z	Оптимум	
	$r(t)+s(t+1)-c(t)$	$r(0)+s(t)+s(1)-c(0)-I$	$f_4(t)$	Рішення
	19.0+60-0.6=78.4	20+80+80-0.2-100=79.8	79.8	Z
	18.5+50-1.2=67.3	20+60+80-0.2-100=59.8	67.3	З
	17.2+30-1.5=45.7	20+50+80-0.2-100=49.8	49.8	Z
	Необхідна заміна	20+5+80-0.2-100=4.8	4.8	Z

Етап 3.

t	З	Z	Оптимум	
	$r(t)-c(t)+f_4(t+1)$	$r(0)+s(t)+c(0)-I+f_4(1)$	$f_3(t)$	Рішення
	19.0-0.6+67.3=85.7	20+80-0.2-100+79.8=79.6	85.7	З
	18.5-1.2+49.8=67.1	20+60-0.2-100+79.8=59.6	67.1	З
	14.0-1.8+4.8=17.0	20+10-0.2-100+79.8=9.6	17.0	З

Етап 2.

t	З	Z	Оптимум	
	$r(t)-c(t)+f_3(t+1)$	$r(0)+s(t)+c(0)-I+f_3(1)$	$f_2(t)$	Рішення
	19.0-0.6+67.1=85.5	20+80-0.2-100+85.7=85.5	85.5	З або Z
	15.5-1.7+17.0=30.8	20+30-0.2-100+85.7=35.5	35.5	Z

Етап 1.

t	З	Z	Оптимум	
	$r(t)-c(t)+f_2(t+1)$	$r(0)+s(t)+c(0)-I+f_2(1)$	$f_1(t)$	Рішення
	17.2-1.5+35.5=51.2	20+50-0.2-100+85.5=55.3	55.3	Z

На рис. 3.2 підсумовано отримання оптимального розв'язку. На початку першого року, якщо $t=3$ оптимальним розв'язком є заміна механізму. Таким чином, новий механізм перебуватиме в експлуатації 1 рік до початку другого року. При $t=1$ на початку другого року оптимальним розв'язком буде збереження або заміна механізму. Якщо його замінити, то новий механізм перебуватиме в експлуатації 1 рік до початку третього року, інакше збереженому механізму буде 2 роки. Процес продовжується таким чином до четвертого року.

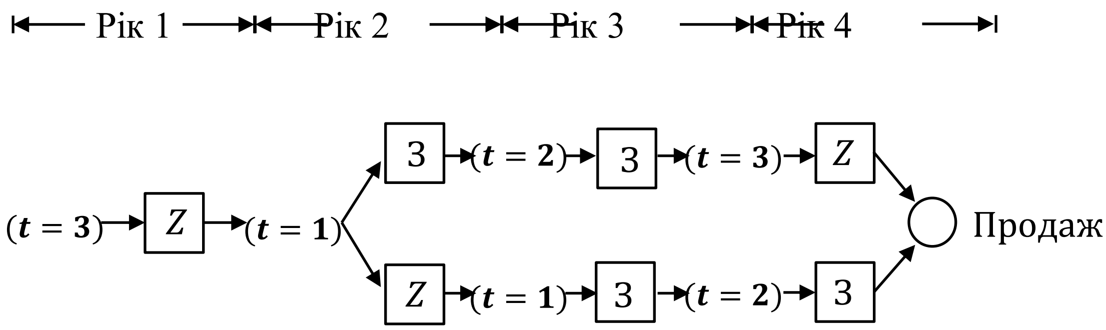

Рис. 3.2

Отже, альтернативними оптимальними стратегіями починаючи з першого року щодо заміни механізму є $(Z, З, З, Z)$ та $(Z, Z, З, З)$. Загальний прибуток 55 300 доларів.

3.4. Задача інвестування

Припустимо, нехай ви хочете інвестувати 4000 доларів на даний момент і 2000 доларів на початку від другого до четвертого років. Відсоткова ставка, яку пропонує перший банк становить 8% щорічно та бонуси протягом наступних чотирьох років становлять 1,8%, 1,7%, 2,1% та 2,5% відповідно. Річна відсоткова ставка, яку пропонує другий банк, на 0,2% нижча, ніж у першого банку, але бонус на 0,5% вищий. Метою є максимізація накопиченого капіталу в кінці чотирьох років.

Використовуючи позначення, введені у підрозділі 2.2.4, маємо

$$P_1 = \$4\,000, P_2 = P_3 = P_4 = \$2\,000,$$

$\alpha_1 = (1 + 0.08) = 1.08,$

$\alpha_2 = (1 + 0.078) = 1.078,$

$q_{11} = 0.018, q_{21} = 0.017, q_{31} = 0.021, q_{41} = 0.025,$

$q_{12} = 0.023, q_{22} = 0.022, q_{32} = 0.026, q_{42} = 0.030.$

Етап 4.

$$f_4(x_4) = \max_{0 \le I_4 \le x_4} \{s_4\},$$

де

$$s_4 = (\alpha_1 + q_{41} - \alpha_2 - q_{42})I_4 + (\alpha_2 + q_{42})x_4 = -0.003I_4 + 1.108x_4.$$

Функція s_4 є лінійною по I_4 в діапазоні $0 \le I_4 \le x_4$, а, її максимум виникає при $I_4 = 0$ через від'ємний коефіцієнт I_4. Таким чином, оптимальний розв'язок для етапу 4 можна підсумувати як

Стан	Оптимальний розв'язок	
	$f_4(x_4)$	I_4^*
x_4	$1.108x_4$	0

Етап 3.

$$f_3(x_3) = \max_{0 \le I_3 \le x_3} \{s_3 + f_4(x_4)\},$$

де

$$s_3 = (1.08^2 - 1.078^2)I_3 + 1.078^2 x_3 = 0.00432I_3 + 1.1621x_3,$$

$$x_4 = 2000 - 0.005I_3 + 0.026x_3.$$

Отже,

$$f_3(x_3) = \max_{0 \le I_3 \le x_3} \{0.00432I_3 + 1.1621x_3 + 1.108(2000 - 0.005I_3 + 0.026x_3)\}$$

$$= \max_{0 \le I_3 \le x_3} \{2216 - 0.00122I_3 + 1.1909x_3\}.$$

Стан	Оптимальний розв'язок	
	$f_3(x_3)$	I_3^*
x_3	$2216 + 1.1909x_3$	*0*

Етап 2.

$$f_2(x_2) = \max_{0 \le I_2 \le x_2} \{s_2 + f_3(x_3)\},$$

де

$$s_2 = (1.08^3 - 1.078^3)I_2 + 1.078^3 x_2 = 0.006985I_2 + 1.25273x_2,$$

$$x_3 = 2000 - 0.005I_2 + 0.022x_2.$$

Отже,

$$f_2(x_2) = \max_{0 \le I_2 \le x_2} \{0.006985I_2 + 1.2527x_2 + 2216 + 1.1909(2000 - 0.005I_2$$

$$+ 0.022x_2)\} = \max_{0 \le I_2 \le x_2} \{4597.8 + 0.0010305I_2 + 1.27893x_2\}.$$

Стан	Оптимальний розв'язок	
	$f_2(x_2)$	I_2^*
x_2	$4597.8 + 1.27996x_2$	x_2

Етап1

$$f_1(x_1) = \max_{0 \le I_1 \le x_1} \{s_1 + f_2(x_2)\},$$

де

$s_1 = (1.08^4 - 1.078^4)I_1 + 1.078^4 x_1 = 0.01005I_1 + 1.3504x_1,$

$x_2 = 2000 - 0.005I_1 + 0.023x_1.$

Отже,

$$f_1(x_1) = \max_{0 \le I_1 \le x_1} \{0.01005I_1 + 1.3504x_1 + 4597.8 + 1.27996(2000 - 0.005I_1 + 0.023x_1)\} = \max_{0 \le I_1 \le x_1} \{7157.7 + 0.00365I_1 + 1.37984x_1\}.$$

Стан	Оптимальний розв'язок	
	$f_1(x_1)$	I_1^*
$x_1 = \$4000$	$7157.7 + 1.38349x_1$	$\$4000$

Під час обчислень у зворотному напрямку маємо наступне

$$x_2 = 2000 - 0.005 \times 4000 + 0.023 \times 4000 = \$2072,$$

$$x_{3=}2000 - 0.005 \times 2072 + 0.022 \times 2072 = \$2035.22,$$

$$x_4 = 2000 - 0.005 \times 0 + 0.026 \times 2035.22 = 2052.92.$$

Таким чином, оптимальний розв'язок підсумовується як

Оптимальний розв'язок	Рішення, що приймається інвестором
$I_1^* = x_1$	Інвестувати $x_1 = \$\ 4000$ в перший банк
$I_2^* = x_2$	Інвестувати $x_2 = \$\ 2072$ в перший банк
$I_3^* = 0$	Інвестувати $x_3 = \$\ 2035.22$ в другий банк
$I_4^* = 0$	Інвестувати $x_4 = \$\ 2052.92$ в другий банк

3.5 Задачі економічного розміру замовлення

Модель без витрат на оформлення замовлення.

Компанія виробляє спеціальні витяжки для використання в домашніх камінах протягом місяців з грудня до березня. Попит починається повільно, досягає піку в середині сезону та зменшується до кінця сезону. Тому що враховуючи популярність продукції, компанія може використовувати понаднормовий час, щоб задовольнити попит на продукцію. Наступна таблиця має дані про виробничі потужності та потреби на чотири зимові місяці.

Можливості виробництва			
Місяць	Звичайний режим роботи (одиниці)	Надурочні (одиниці)	Попит (одиниці)
1	90	50	100
2	100	60	190
3	120	80	210
4	110	70	160

Вартість одиниці виробництва продукції в будь який період дорівнює 6 доларів протягом основного режиму роботи та 9 доларів протягом понаднормового часу. Вартість зберігання за одиницю продукції на місяця дорівнює 0,10 доларів.

Щоб гарантувати, що можливі розв’язки, коли не допускається дефіцит, необхідно, щоб сумарна пропозиція продукції (можливості виробництва) до початку кожного місяця щонайменше дорівнювала сумарному попиту. Про це свідчить таблиця 3.3 [6, с.67].

Таблиця 3.3

Місяць	Сумарна пропозиція	Сумарний попит
1	90+50=140	100
2	140+100+60=300	100+90=290
3	300+120+80=500	290+210=500
4	500+110+70=680	500+160=660

Таблиця 3.4 містить дані розглянутої задачі, та її розв'язання. Символи R_i і O_i позначають звичайний і надурочний режим роботи в період $i, i = 1, 2, 3, 4$.

Оскільки в четвертому періоді сумарна пропозиція перевищує сумарний попит, то додано штучний пункт споживання, щоб збалансувати модель. Усі «транспортні» маршрути від попереднього до поточного періоду є заблоковані, оскільки дефіцит відсутній.

«Транспортні» витрати на одиницю продукції є сумою відповідних витрат на виробництво та зберігання. Наприклад, собівартість від R_i до першого періоду дорівнює тільки собівартості виготовлення одиниці продукції 6 доларів, собівартість від O_i, до четвертого періоду дорівнює вартості виготовлення плюс собівартість зберігання від першого до четвертого періоду, тобто 9+(0,1+0,1+0,1)= = 9,30 доларів. Нарешті, собівартість перевезення до штучного пункту споживання (надлишок) дорівнює нулю.

Таблиця 3.4

	1	2	3	4	Надлишок	
R_1	6 **90**	6,1	6,2	6,3	0	**90**
O_1	9 **10**	9,1 **30**	9,2 **10**	9,3	0	**50→40→10**
R_2		6 **100**	6,1	6,2	0	**100**
O_2		9 **60**	9,1	9,2	0	**60**
R_3			6 **120**	6,1	0	**120**
O_3			9 **80**	9,1	0	**80**
R_4				6 **110**	0	**110**
O_4				9 **50**	0 **20**	**70→20**
	100	**190**	**210**	**160**	**20**	
	↓	↓	↓	↓		
	10	**90**	**90**	**50**		
		↓	↓			
		30	**10**			

Оптимальний розв'язок виходить за один прохід, розпочинаючи з першого стовпця і рухатися у напрямку до стовпця «Надлишок». Для кожного стовпця попит задовольняється за допомогою використання найдешевшого маршруту.

Починаючи з першого стовпця маршрут (R_1, 1) має найдешевшу собівартість перевезення, і тоді ми призначаємо перевезення максимально можливого обсягу, а саме $\min(90, 100) = 90$ одиниць, що залишає 10 незадоволених одиниць попиту у першому стовпці. І наступним найдешевшим маршрутом у першому стовпці є $(O_1, 1)$, якому визначаємо перевезення $\min(50, 10) = 10$ одиниць, Попит для першого періоду тепер задоволено.

Далі переходимо до другого стовпця. Визначення перевезень у другому стовпці відбувається у такому порядку: 100 одиниць за (R_2, 2), 60 одиниць за $(O_2, 2)$ та 30 одиниць за $(O_1, 2)$. Відповідно маршрутам собівартості цих «перевезень» становить в 6, 9 і 9,10 доларів. При цьому маршрут $(R_1, 2)$, не розглядається, транспортні витрати якого дорівнюють 6,10 доларів на одиницю продукції, тому що весь запас R_1 витрачено для першого періоду.

Продовжуючи таким же чином, задовольняємо попит для третього, а потім і четвертого стовпців. Оптимальний розв'язок, виділено жирним шрифтом у таблиці 3.4, підсумовано таким чином:

Період	Виробництво
Період 1 (звичайний режим роботи)	Виготовити 90 одиниць продукції для першого періоду
Період 1 (надурочний режим роботи)	Виготовити 40 одиниць продукції: 10 для періоду 1, 30 для періоду 2 та 10 для періоду 3
Період 2 (звичайний режим роботи)	Виготовити 100 одиниць продукції для другого періоду
Період 2 (надурочний режим роботи)	Виготовити 60 одиниць продукції для періоду 2
Період 3 (звичайний режим роботи)	Виготовити 120 одиниць продукції для

	третього періоду
Період 3 (надурочний режим роботи)	Виготовити 80 одиниць продукції для періоду 3
Період 4 (звичайний режим роботи)	Виготовити 110 одиниць продукції для четвертого періоду
Період 4 (надурочний режим роботи)	Виготовити 50 одиниць продукції для періоду 4; залишилася невикористаною виробнича потужність на 20 одиниць продукції

Відповідні сумарні витрати становлять

$$90 \times 6 + 10 \times 9 + 30 \times 9{,}10 + 100 \times 6 + 60 \times 9 + 10 \times 9{,}20 + 120 \times 6 + 80 \times$$
$$\times 9 + 110 \times 6 + 50 \times 9 = 4685 \text{ доларів.}$$

Задача із загальною функцією вартості

У таблиці 3.5 наведено дані для триетапної системі керування запасами. Початковий запас становить $x_1 = 1$ одиниці продукції. Передбачається, що вартість одиниці продукції становить 10 доларів за перші тири одиниці та 20 доларів за кожну додаткову одиницю. Необхідно визначити оптимальну стратегію.

Таблиця 3.5

Період i	Попит, D_i(одиниці)	Витрати на оформлення замовлення K_i(\$)	Витрати на зберігання h_i(\$)
1	3	3	1
2	2	7	3
3	4	6	2

Для періоду i функція виробничих витрат дорівнює $C_i(z_i) = K_i + c_i(z_i)$ для $z_i > 0$, де

$$c_i(z_i) = \begin{cases} 10z_i, & 0 \le z_i \le 3, \\ 30 + 20(z_i - 3), & z_i \ge 4. \end{cases}$$

Етап 1.

$D_1 = 3, \quad 0 \le x_2 \le 2 + 4 = 6.$

x_2	h_1x_2	$C_1(z_1) + h_1x_2$							Оптимальний розв'язок	
		$z_1 = 2$	3	4	5	6	7	8		
		$C_1(z_1) = 23$	33	53	73	93	113	133	$f_1(x_2)$	z_1^*
0	0	23							23	2
1	1		34						34	3
2	2			55					55	4
3	3				76				76	5
4	4					97			97	6
5	5						118		118	7
6	6							139	139	8

Так як x_1 = 1, мінімальне значення z_1 дорівнює $D_1 - x_1 = 3 - 1 = 2.$

Етап 2. $D_2 = 2,\quad 0 \le x_3 \le 4.$

x_3	h_2x_3	$C_2(z_2) + h_2x_3 + f_1(x_3 + D_2 - z_2)$							Оптимальний розв'язок	
		$z_2 = 0$	1	2	3	4	5	6		
		$C_2(z_2)=$ $= 0$	17	27	37	57	77	97	$f_2(x_3)$	z_2^*
0	0	0+55= =55	17+34= =51	27+23= =50					50	2
1	3	3+76= =79	20+55= =75	30+34= =64	40+23= =63				63	3
2	6	6+97= =103	23+76= =99	33+55= =88	43+34= =77	63+23= =86			77	3
3	9	9+118= =127	26+97= =123	36+76= =112	46+55= =101	66+34= =100	86+23= =109		100	4
4	12	12+139= =151	29+118= =147	39+97= =136	49+76= =125	69+55= =124	89+34= =123	109+23= =123	123	5

Етап 3. $D_3 = 4,\quad x_4 = 0.$

x_4	h_3x_4	$C_3(z_3) + h_3x_4 + f_2(x_4 + D_3 - z_3)$					Оптимальний розв'язок	
		$z_3 = 0$	1	2	3	4		
		$C_3(z_3) = 0$	16	26	36	56	$f_3(x_4)$	z_3^*
	(	0+123=123	16+100=	26+77=103	36+63=99	56+50=	99	3

			=116			=106		

Оптимальний розв'язок визначається таким чином:

$$(x_4 = 0) \rightarrow [z_3 = 3] \rightarrow (x_4 = 0 + 4 - 3 = 1) \rightarrow [z_2 = 3] \rightarrow (x_2 = 1 + 2 - 3 = 0) \rightarrow [z_1 = 3].$$

Таким чином, отримуємо розв'язок: $z_1^* = 2, z_2^* = 3$ і $z_3^* = 3,$ при цьому підсумовуються загальні витрати 99 доларів.

Задача з постійними або спадаючими граничними витратами.

Чотириетапна модель керування запасами має такі вихідних даних.

Період i	Попит D_i (одиниці)	Витрати оформлення замовлення K_i($)
1	76	98
2	26	114
3	90	185
4	67	70

Початковий запас дорівнює $x_1 = 15$ одиниць. Виробнича собівартість одиниці продукції становить 2 долари, а вартість зберігання за один період становить 1 долари. (Для простоти передбачається, що витрати на виробництво одиниці продукції та зберігання не змінюються для всіх періодів.)

Розв'язання визначається за допомогою алгоритмом прямої прогонки, наведеного раніше, за винятком значення величини x_{i+1} і z_i, припустимі «загальні» платежі, що випливає з наведених властивостей функції витрат. Оскільки $x_1 = 15$, попит на перший період зменшується і становить 76-15=61 одиниць.

Етап 1. $D_1 = 61$

x_2	h_1x_2	$C_1(z_1) + h_1x_2$				Оптимальний розв'язок	
		$z_1 = 61$	87	177	244		
		$C_1(z_1) = 220$				$f_1(x_2)$	z_1^*
0	0	220	-	-	-	220	61
26	26	-	298	-	-	298	87

116 183	116 183	- -	- -	568 -	- 769	568 769	177 244
Замовлення на етапі 1 для етапів		1	1, 2	1, 2,	1, 2, 3,		

Етап 2. $D_2 = 26$

x_3	h_2x_3	$C_2(z_2) + h_2x_3 + f_1(x_3 + D_2 - z_2)$				Оптимальний розв'язок	
		$z_2 = 0$	26	116	183		
		$C_2(z_2)$=0	166	346	480	$f_2(x_3)$	z_2^*
90	0	0+298=298	166+220= =386	-	-	298	0
0	90	90+568=658	-	436+220= =656	-	656	166
15	157	157+769=92	-	-	637+220= =857	857	183
Замовлення на етапі 2 для етапів		-	2	2, 3	2, 3, 4		

Етап 3. $D_3 = 90$

x_4	h_3x_4	$C_3(z_3) + h_3x_4 + f_2(x_4 + D_3 - z_3)$			Оптимальний розв'язок	
		$z_3 = 0$	90	157		
		$C_3(z_3) = 0$	365	499	$f_3(x_4)$	z_3^*
0 67	0 67	0+656=656 67+857=924	365+298=663 -	- 566+298=864	656 864	0 157
Замовлення на етапі 3 для етапів		-	3	3, 4		

Етап 4. $D_4 = 67$

<table>
<tr><td rowspan="3">x_5</td><td rowspan="3">$h_4 x_5$</td><td colspan="2">$C_4(z_4) + h_4 x_5 + f_3(x_5 + D_4 - z_4)$</td><td colspan="2" rowspan="2">Оптимальний розв'язок</td></tr>
<tr><td>$z_4 = 0$</td><td>67</td></tr>
<tr><td>$C_4(z_4) = 0$</td><td>204</td><td>$f_4(x_5)$</td><td>z_4^*</td></tr>
<tr><td>0</td><td>0</td><td>0+864=864</td><td>204+656=860</td><td>860</td><td>67</td></tr>
<tr><td colspan="2">Замовлення на етапі 3 для етапів</td><td>-</td><td>4</td><td></td><td></td></tr>
</table>

Оптимальна стратегія визначається з таблиці таким чином:

$$(x_5 = 0) \to [z_4 = 67] \to (x_4 = 0) \to [z_3 = 0] \to (x_3 = 90) \to [z_2 = 116] \to (x_2 = 0) \to [z_1 = 61].$$

Звідси отримуємо розв'язок: $z_1^* = 61, z_2^* = 116, z_3^* = 0$ і $z_4^* = 67$ при сумарних витратах 860 доларів.

ВИСНОВКИ

Як розділ математичного програмування, динамічне програмування розпочало розвиватися у 50-х роках XX ст. завдяки працям Р. Беллмана та його співробітників.

Динамічне програмування являє собою теорію, яка поєднує однотипні ідеї та прийоми, що застосовуються для розв’язування різних за змістом задач. Переваги полягають у поетапному аналізі результатів задачі, що дозволяє більш об’єктивно, точно та повно розв’язати задачі цілеспрямованої людської діяльності.

У динамічному програмуванні можна виділити такі аспекти: теоретичний прикладний та обчислювальний. Характерні особливості та структуру задачі динамічного програмування найлегше засвоїти, якщо розв'язування задачі розпочинати зі складання моделі та завершувати виконанням розрахунків.

Детерміновані моделі динамічного програмування дають можливість знайти оптимальний розв’язок у задачах з багатокроковою структурою. Побудова таких моделей, є певною мірою мистецтвом. Тому у роботі наводяться різноманітні приклади, які дозволяють відчути побудову даних моделей.

Під час виконання роботи було проведено аналіз застосування детермінованих моделей динамічного програмування.

Робота містить виклад матеріалу про детерміновані моделі динамічного програмування. Її можна використовувати як конспект лекцій, або при підготовці до іспиту для студентів вузів, які вивчають детерміновані моделі динамічного програмування. Разом з тим, вона може використовуватися як задачник для розв'язування типових задач. В теоретичний матеріал включено всі основні поняття. Детально розглянуті алгоритми розв’язку задач.

Робота складається з трьох розділів. У першому розділі розглянуто основні поняття і визначення пов’язанні динамічним програмуванням. У другому розділі докладно розглянуті приклади детермінованих моделей динамічного програмування. Зокрема дається уявлення теоретичного дослідженням моделей динамічного програмування. У третьому розділі наводиться приклад задач динамічного програмування.

В результаті проведеної роботи була розкрита тема «Детерміновані моделі динамічного застосування», поставлена мета досягнута в повному обсязі і виконано поставленні завдання.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Агальцов В. П. Математические методы в программировании: учебник / Агальцов В. П. – [2-е изд., перераб. и доп.] – М. : ИД «ФОРУМ», 2015. –240 с. : ил. – (Профессиональное образование).

2. Вентцель Е. С. Исследование операций. Задачи, принципы, методология : учеб. пособие для студ. / Вентцель Е. С. – [2-е изд., стер.] – М. : Высш. шк., 2001. – 208 с. : ил.

3. Вітлінський В.В. Математичне програмування: навч.-метод. посібник для самост.вивч. дисц. / Вітлінський В.В., Наконечний С.І., Терещенко Т.О. – К.: КНЕУ, 2001. – 248 с.

4. Зайченко Ю.П. Исследование операций: учебное пособие. / Зайченко Ю.П. – [2-е изд., перераб. и доп.] – К. : Вища школа, - 1979. – 392с.

5. Исследование операций в экономике : учебное пособие для вузов / [Н.Ш. Кремер, Б.А. Путко, И.М. Тришин, М.Н. Фридман] ; – под ред. проф. Н.Ш. Кремера. – М. : ЮНИТИ, 2002. – 407 с.

6. Калихман И.Л. Динамическое программирование в примерах и задачах: учеб. пособие. / И. Л. Калихман, М. А. Войтенко - М. : Высш. школа, 1979. - 125 с. : ил.

7. Красс М. С. Основы математики и ее приложения в экономическом образовании : учеб. / М. С. Красс, Б. П. Чупрынов – [4-е изд. испр.] – М. : Дело, 2001. – 688 с.

8. Кузнецов А. В. Руководство к решению задач по математическому программированию : учеб. пособие для студ. / А. В. Кузнецов, Н.И. Холод, Л. С. Костевич ; под ред. А.В. Кузнецова. – [2-е изд., перераб. и доп.] – Мн. : Выш. шк., 2001. – 448, [2] с. : ил., табл.

9. Кузнецов Б. Т. Математика: учебник для студентов вузов, обучающихся по специальностям экономики и управления / Кузнецов Б. Т. – [2-е изд., перераб. и доп.] – М. : ЮНИТИ-ДАНА, 2004. – 719 с. – (Серия «Высшее профессиональное образование: Экономика и управление»).

10. Кулян В. Р. Математическое программирование (с элементами информационных технологий) : учеб. пособие для студ. нематематических спец. вузов / В. Р. Кулян, Е. А. Юнькова, А. Б. Жильцов. - К. : МАУП, 2000. - 121 с. : ил.

11. Лежнев А. В. Динамическое программирование в экономических задачах : учеб. пособие / Лежнев А. В. – М. : БИНОМ. Лаборатория знаний, 2010. – 176 с. : ил.

12. Лисенко О. І. «Математичні методи моделювання та оптимізації. Частина 1. Математичне програмування та дослідження операцій : підручник / О. І. Лисенко, О. М. Тачиніна, І. В. Алєксєєва ; за заг. ред. О. І. Лисенка. – К. : НАУ, 2017. – 212 с.

13. Математичні методи дослідження операцій : підручник / [Є. А. Лавров, Л. П. Перхун, В. В. Шендрик та ін.]. – Суми : Сумський державний університет, 2017. – 212 с.

14. Минюк С. А. Математические методы и модели в экономике : учеб. пособие / Минюк С. А., Ровба Е. А., Кузьмич К. К. – Мн. : ТетраСистемс, 2002. – 432 с.

15. Окулов С.М. Динамическое программирование / С. М. Окулов, О. А. Пестов. - М. : БИНОМ. Лаборатория знаний, 2015. - 300 с. - (Развитие интеллекта школьников).

16. Романовский И.В. Алгоритмы решения экстремальных задач / Романовский И.В. – М. : Наука, 1977. - 352 с. : ил.

17. Саркисян Р.Е. Детерминированные модели динамического программирования. Численные методы оптимизации : учебное пособие / Саркисян Р.Е. – М. : МИИТ, 2008. – 200 с.

18. Таха Х. М. Введение в исследование операций / Таха Х. М. – [2-е изд.] – М. : Вильямс, 2005. – 912 с. : ил. – Парал. тит. англ.

19. Трунов О. М. Приклади розв’язку практичних та ситуаційних задач з курсу «Дослідження операцій» / Трунов О. М., Волкова С. О. – Київ, 2008. – 116 с.

20. Хазанов Л.Э. Математическое моделирование в экономике: учебное пособие / Хазанов Л.Э. - М. : БЕК, 1998. – 141с.

21. Хэдли Дж. Нелинейное и динамическое программирование / Хэдли Дж. ; перевод с англ. Ю.И. Волкова и др. ; - под редакцией Г.П. Акимова. – М. : Мир, 1967. – 506 с.

22. Черноморов Г. А. Теория принятия решений : учеб. пособие / Черноморов Г. А. – Юж. –Рос. гос. техн. ун-т. Новочеркасск: Ред. Журн. «Изв. вузов Электромеханика», 2002. – 276 с.

23. Шикин Е. В. Математические методы и модели в управлении : учебное пособие / Е. В. Шикин, А. Г. Чхартишвили – [3-е изд.] – М. : Дело, 2004 – 440 с. – (Серия « Классический университетский учебник»).