

Василь Янішевський, кандидат фізико-математичних наук,
старший викладач

Дрогобицький державний педагогічний університет
ім. І. Франка

ЗАДАЧІ ДИНАМІЧНОГО ПРОГРАМУВАННЯ ТА ЦІЛОЧИСЛОВІ МОДЕЛІ

У статті розглянуті типові задачі динамічного програмування, що є предметом вивчення в ряді дисциплін з економіко-математичного моделювання. Показано, що для кожної з моделей можна скласти цілочислову модель (лінійну чи нелінійну), для розв'язання яких можна використати широко доступні пакети оптимізаційних програм. Наведено також приклад розв'язку динамічної задачі знаходження найкоротшого шляху з використанням математичного пакету Mathcad.

Постановка проблеми та аналіз останніх досліджень і публікацій. Метод динамічного програмування (ДП) – математичний метод сучасної теорії управління і знаходить застосування у вирішенні різноманітних проблем теорії прийняття рішень в різноманітних галузях знання. Він був запропонований американським математиком Р. Беллманом, який його чітко сформулював і пропагував у своїх книгах [1, 2, 3]. Методам динамічного програмування присвячені також сучасні дослідження, які представлені в багатьох монографіях та журнальних статтях [8, 13, 14]. Даний метод вивчається студентами економічних спеціальностей в таких дисциплінах, як “Математичне програмування”, “Дослідження операцій” та інших пов'язаних з оптимізацією та моделюванням економічних, екологічних, соціальних та інших процесів. Основи цього методу представлені в багатьох підручниках та навчальних посібниках, зокрема [6, 7, 12]. Метод динамічного програмування застосовний до задач, у яких процес прийняття рішення можна розбити на етапи і при цьому виконується принцип оптимальності [1]. Принцип оптимальності в свою чергу є наслідком адитивності критерію оптимальності та відсутності зворотного зв'язку між етапами. Така ситуація властива багатьом практичним економіко-виробничим задачам. Зокрема, моделі динамічного програмування застосовуються в задачах управління запасами, розподілу дефіцитних ресурсів, ремонту обладнання та його заміни, планування зайнятості на виробництві та інших. Популярність ДП зумовлена достатньо простою структурою алгоритму, що дозволяє досить успішно розв'язувати оптимізаційні задачі ручним способом. Очевидно, така простота була досить суттєвою перевагою в період, коли комп'ютерні технології ще не набули широкого застосування.

Сьогодні, коли комп'ютер став доступний практично кожному, коли наявні пакети прикладних програм для розв'язання різноманітних оптимізаційних задач, ситуація зазнає суттєвих змін.

Мета статті. У даній статті висловлюються певні зауваження та рекомендації щодо викладання методів ДП, що мають місце сьогодні в багатьох посібниках та підручниках. Слід зауважити, що, по-перше, як викладання методу ДП та розв'язування з його допомогою задач в основному супроводжується ручними розрахунками [7, 10]; по-друге, не завжди встановлюється зв'язок з іншими методами, які можуть бути застосовні до задач, що традиційно “закріплені” за методом ДП (за невеликим винятком для задачі про завантаження наводиться еквівалентна лінійна цілочислова модель [12]). Отже, з метою усунення згаданих прогалин, у даній статті для найбільш типових задач ДП (дискретного) формулюються еквівалентні цілочислові моделі математичного програмування. Слід зауважити, що більшість практичних оптимізаційних задач можна сформулювати у вигляді цілочислових чи частково цілочислових моделей. Тому розвиток вмінь та навичок аналізу з допомогою згаданих моделей математичного програмування різноманітних економіко-виробничих задач є досить актуальною складовою економіко-математичної підготовки студентів.

Виклад основного матеріалу. Моделі ДП та цілочислові моделі.

1. Задача найкоротшого шляху.

Безумовно, дана задача є найбільш вдалим вибором для першого ознайомлення із методом ДП. Задача полягає у знаходженні найкоротшого шляху між двома пунктами [12]. При цьому наявна мережа доріг, що проходить через проміжні пункти і з'єднує дані два пункти. Спочатку задача

ЗАДАЧІ ДИНАМІЧНОГО ПРОГРАМУВАННЯ ТА ЦІЛОЧИСЛОВІ МОДЕЛІ

розбивається на етапи. В кожен етап входить певна кількість пунктів, які з'єднані із пунктами наступного етапу і так далі. Графічно це можна зобразити у вигляді схеми (Рис. 1), де пункти зображаються вузлами і вузли сусідніх етапів з'єднані зв'язками. У загальному випадку маємо $n+1$ вузлів, з яких $n-1$ згруповані в m етапів.

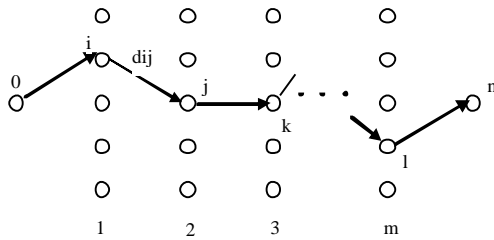


Рис. 1

З метою систематизації наступних викладок введемо певні позначення. Нехай вектор v_i містить множину вузлів, що належать i -му етапу, відстані між вузлами етапів містяться в матриці d_{ij} , ij , елемент якої визначає відстань між i та j вузлами. Динамічний метод розв'язку даної задачі наведений у [12]. Суть його полягає в тому, що послідовно знаходять найкоротші відстані до вузлів першого етапу з наступним використанням цих відстаней як вихідних даних для наступного етапу і так далі. Якщо позначити $f_i(j)$ найкоротшу відстань до вузла j на i -му етапі, тоді можна скласти рекурентне рівняння для знаходження $f_i(j)$ (рівняння Беллмана для даної задачі)

$$f_i(j) = \min_{\text{всі } j-k \text{ зв'язки}} \{d_{jk} + f_{i-1}(k)\}$$

для початкового вузла $f_0(0) = 0$

Якщо кількість вузлів та етапів невелика, то розрахунки за рекурентним рівнянням можна здійснити ручним способом. Однак, якщо кількість вузлів немала, то ручний спосіб стає дуже трудомістким. У цьому випадку можна скласти програму за алгоритмом певною мовою програмування. З цією метою можна скористуватися згаданим математичним пакетом Mathcad [11, 12]. Зокрема, в додатку наведений приклад програми, що реалізує алгоритм даної задачі, складеної в системі Mathcad.

Як вже підкреслювалось, метод динамічного програмування визначає основну ідею знаходження розв'язку, однак сам алгоритм розв'язку залежить від конкретної задачі. Для

різних задач, алгоритми розв'язку потрібно розробляти самостійно, причому вони бувають зазвичай складніші, ніж у наведеному вище прикладі, це залежить від структури обмежень задачі, від кількості параметрів моделі та інше. Тому доречним є розгляд інших формулювань даної задачі. Ми скористаємось методами та моделями цілочислового програмування.

Зокрема, введемо двійкові змінні x_i для кожного вузла. Ці змінні приймають значення 1, якщо шлях проходить через даний вузол, і значення 0 в протилежному випадку. Знаходження мінімального шляху тоді відповідатиме мінімуму цільової функції

$$F = \sum_{i \in v_1} x_0 d_{0i} x_i + \sum_{\substack{i \in v_1 \\ j \in v_2}} x_i d_{ij} x_j + \dots + \sum_{i \in v_m} x_i d_{in} x_n \rightarrow \min$$

при обмеженнях

$$\sum_{i \in v_1} x_i = 1, \sum_{i \in v_2} x_i = 1, \sum_{i \in v_3} x_i = 1, \dots, \sum_{i \in v_m} x_i = 1$$

$$x_i = 0, 1 \quad i \in (v_1, v_2, \dots, v_m), \quad x_0 = 1, \quad x_n = 1.$$

Обмеження вказують, що найкоротший шлях пройде лише через один вузол кожного етапу. Записана модель є нелінійною, а саме квадратичною. Кількість змінних моделі збігається з кількістю вузлів, а кількість обмежень рівна числу етапів. Безумовно, при ручних розрахунках слід віддати перевагу динамічному методу розв'язання, а не шукати розв'язок цілочислової нелінійної моделі. Однак для розв'язання нелінійної моделі можна скористатись нелінійними алгоритмами поширеного пакету оптимізації (Solver), вбудованого в табличний процесор Microsoft Excel, чи пакетом Лінго, навчальна версія якого вільно поширюється через Internet [10]. Зауважимо, що в доступному математичному пакеті Mathcad зазвичай відсутні цілочислові алгоритми, для цього потрібне розширення пакету (Solving and Optimization Extension Pack), що реалізується на комерційній основі.

Якщо ввести змінні для зв'язків між вузлами x_{ij} , то можна записати адекватну лінійну модель. Тоді цільова функція набуде вигляду

$$F = \sum_{i \in v_1} d_{0i} x_{0i} + \sum_{\substack{i \in v_1 \\ j \in v_2}} d_{ij} x_{ij} + \dots + \sum_{i \in v_m} d_{in} x_{in} \rightarrow \min$$

Для складання обмежень, слід врахувати всі можливі маршрути, що з'єднують початковий та кінцевий пункт, тобто

$$x_{0i} + x_{i_1} + \dots + x_{i_m} = m + 1$$

$$i_1 \in v_1, i_2 \in v_2, \dots, i_m \in v_m$$

Лінійності моделі досягнуто за рахунок збільшення кількості змінних та кількості обмежень (кількість змінних задачі рівна кількості можливих зв'язків між етапами, а кількість обмежень рівна кількості усіх маршрутів, що з'єднують початковий та кінцевий вузли). Оскільки задача є лінійною, то для її розв'язку можна застосовувати цілочислові алгоритми програм TORA ([12, 12]), Excel чи пакет Lindo [9]. Розглянута модель найкоротшого шляху може бути застосована для інших економіко-виробничих задач. Зокрема, якщо замість відстані ввести затрати на будівництво шляху між заданими пунктами, то дану модель можна використовувати для знаходження способу прокладання шляху з найменшими затратами [12, 10].

2. Задача про розподіл ресурсів між підприємствами.

Сюди належить ціла група задач, які розглядаються методами динамічного програмування. У загальному випадку задачі оптимального розподілу ресурсів можуть бути описані наступним чином. У наявності є певна кількість ресурсів (матеріальні, трудові, фінансові), котрі потрібно розподілити між рівними об'єктами їх використання по окремих проміжках планового періоду так, щоб отримати максимальну сумарну ефективність від вибраного способу розподілу. Ми обмежимося розглядом задачі наведеної в [7], в якій початкові ресурси S (вважається дискретною величиною) розподіляються між n підприємствами. За певний проміжок часу (наприклад, рік) засоби виділені k -му підприємству приносять прибуток $f_k(x)$. Функція $f_k(x)$ задається таблицею. Якщо позначити x_k – кількість засобів надану k -му підприємству, то математична модель матиме вигляд.

$$F = \sum_{k=1}^n f_k(x_k) \rightarrow \max$$

при обмеженнях

$$\sum_{k=1}^n x_k = S; \quad x_k \geq 0, \quad k = 1, \dots, n$$

У [7] стверджується, що до даної задачі неможливо застосувати методи лінійного цілочислового програмування, оскільки функції $f_k(x_k)$ задаються таблично. Однак тут слід ввести певні уточнення. Можливо, це твердження стосується вищенаведеної моделі, однак доволі

нескладно побудувати модель лінійного програмування (ЛП), використавши двійкові змінні. В даному випадку ситуація подібна до задачі про призначення (спеціальний тип транспортних задач) [12]. Зокрема, введемо двійкові змінні x_{jk} , де перший індекс вказує кількість виділених засобів, а другий – номер підприємства. В результаті отримаємо задачу лінійного програмування через двійкові змінні

$$F = \sum_j^S \sum_{k=1}^n f_k(j) \cdot x_{jk} \rightarrow \max$$

$$\sum_{j=1}^S x_{jk} = 1, \quad k = 1, \dots, n$$

$$\sum_j^S \sum_{k=1}^n j \cdot x_{jk} = S$$

$$x_k = 0, 1; \quad k = 1, \dots, n$$

Перші n обмежень вказують, що ресурси виділяються кожному підприємству, а наступне визначає обмеження на ресурси. Для розв'язку даної задачі лінійного програмування, як вже згадувалось, доступними є ряд засобів. Зокрема, легко отримати результати для прикладу наведеного в [7], з допомогою Excel.

3. Задача планування робочої сили.

Для задач такого типу також можливі різні формулювання, однак достатньо розглянути схему прийняту, наприклад, у [12]. Задача полягає в тому, що для виконання деякого проекту регулюється кількість робітників шляхом найму та звільнення. Нехай проект буде виконуватися протягом n умовних періодів (наприклад, тижнів) і мінімальна потреба в робочій силі протягом i -го періоду становить b_i робітників. Щонайменше упродовж i -го періоду необхідно мати рівно b_i робітників. Однак залежно від вартісних показників може бути вигідним відхилення чисельності робітників від мінімальних потреб. Якщо x_i – кількість працюючих протягом i -го періоду, то можливі, зокрема, витрати двох видів: 1) $C_1 \cdot (x_i - b_i)$ – витрати, які пов'язані з необхідністю утримувати надлишок робочої сили і 2) $f(x_i - x_{i-1})$ – витрати, пов'язані з необхідністю додаткового найму $x_i - x_{i-1}$ робітників. Для лінійної моделі $f(x_i - x_{i-1}) = C_0 + C_1 \cdot (x_i - x_{i-1})$. З динамічним методом розв'язку даної задачі можна

ознайомитися у [12]. Ми зосередимо увагу на формулюванні моделі лінійного програмування. Змінними задачі є x_i , кількість яких рівна кількості періодів. Цільова функція визначає сумарні витрати пов'язані з наймом та звільненням робітників які необхідно мінімізувати. Тобто,

$$F = C_1 \cdot \sum_{i=1}^n (x_i - b_i) + \sum_{i=1}^n (C_0 + C_2 \cdot (x_i - x_{i-1})) \rightarrow \min$$

Другий доданок враховується лише у випадку $x_i > x_{i-1}$, тому зручно ввести додаткові вільні змінні $y_i = x_i - x_{i-1}$, і пов'язані з ними ($y_i = y_i^+ - y_i^-$) невід'ємні змінні $y_i^+, y_i^- \geq 0$. Одночасно є потреба ввести двійкові змінні t_i , такі, що $t_i = 1$ для $y_i^+ \geq 0$ та $t_i = 0$ для $y_i^- \geq 0$. Техніка введення таких змінних описана в [12]. Після необхідних перетворень отримасмо задачу ЛП

$$F = C_1 \cdot \sum_{i=1}^n (x_i - b_i) + \sum_{i=1}^n (C_0 \cdot t_i + C_2 \cdot y_i^+) \rightarrow \min,$$

з обмеженнями

$$\begin{cases} y_i^+ - y_i^- = x_i - x_{i-1}, i = 1..n \\ M \cdot (1 - t_i) + y_i^+ - y_i^- \geq 0, i = 1..n \\ M \cdot t_i + y_i^- - y_i^+ \geq 0, i = 1..n \\ x_i \geq b_i; y_i^+, y_i^- \geq 0; t_i = 0, 1; i = 1..n \\ x_0 = 0 \end{cases}$$

де M – достатньо велике додатне число. Таким чином, задача містить $3n$ цілочислових змінних та n двійкових. Використовуючи дану лінійну цілочислову модель нескладно отримати розв'язки відповідних прикладів, наведених у [12], зокрема з використанням програми TORA.

4. Задача заміни обладнання.

Досить широкий клас задач, для якого традиційно застосовуються методи динамічного програмування. У [5], зокрема, стверджується, що динамічний підхід – єдиний метод розв'язку задач про заміну. Однак, насправді, цю задачу можна звести до задачі мінімального шляху, подібно як в п.1. У загальному задача формулюється наступним чином [12]. Розглядається заміна механізму протягом n періодів (наприклад, років). На початку кожного періоду приймається рішення або про експлуатацію механізму ще на один рік, або про заміну його новим. Якщо

позначити через $r(t)$ і $c(t)$ прибуток від експлуатації t - річного механізму та затрати на його обслуговування за цей же період. Через $s(t)$ позначимо вартість продажу механізму, що експлуатувався t - років, вартість придбання нового механізму приймається однаковою протягом всіх років і рівна I . Для даної задачі можна побудувати схему, що складається з вузлів та зв'язків (детальніше див. [12]). За такою схемою можна скласти цілочислову модель подібно як в п.1, де роль відстаней між відповідними вузлами будуть відігравати вартісні показники. Оскільки необхідні викладки багато в чому повторюють п.1, то їх пропонується здійснити читачеві самостійно.

Висновки. У даній роботі наведено приклад застосування математичного пакету Mathcad до розв'язування однієї із задач динамічного програмування. Наведені також моделі математичного програмування (в більшості випадків лінійного) для задач, які традиційно розглядаються динамічним методом. По-перше – розгляд однієї тієї ж задачі з різних сторін дозволяє глибше проникати в суть поставленої задачі; по-друге, наведені в статті моделі можна використовувати також як тестування результатів отриманих динамічним методом. Отже, матеріали наведені а даній статті можуть бути використані студентами економічних спеціальностей для поглибленого вивчення цілочислових моделей математичного програмування, а також викладачами в лекційних матеріалах.

Додаток

Один з можливих варіантів програми складеної в системі Mathcad, що реалізує метод динамічного програмування знаходження найкоротшого шляху, наводиться нижче

$$v(n, g) := \sum_{i=0}^n g_i$$

$$fdist(d, g) := \begin{cases} f_0 \leftarrow 0 \\ \text{for } j \in v(0, g)..v(1, g) \\ \quad \left| \begin{array}{l} f_{1,j} \leftarrow f_0 + d_{1,j} \\ a_{1,j} \leftarrow \text{concat}("1", "-", \text{num2str}(j)) \end{array} \right. \\ \text{for } n \in 2..rows(g) - 1 \\ \quad \text{for } j \in v(n-1, g) + 1..v(n, g) \\ \quad \quad \left| \begin{array}{l} i_0 \leftarrow v(n-2, g) + 1 \\ f_{\min} \leftarrow f_{n-1, i_0} + d_{i_0, j} \\ a_{\min} \leftarrow i_0 \\ \text{for } i \in v(n-2, g) + 1..v(n-1, g) \\ \quad \quad \left| \begin{array}{l} f_{1n, j} \leftarrow f_{n-1, i} + d_{i, j} \\ \text{if } f_{1n, j} < f_{\min} \\ \quad \quad \left| \begin{array}{l} f_{\min} \leftarrow f_{1n, j} \\ a_{\min} \leftarrow i \end{array} \right. \end{array} \right. \\ f_{n, j} \leftarrow f_{\min} \end{array} \right. \\ f_{n, j} \leftarrow \text{concat}(a_{n-1, a_{\min}}, "-", \text{num2str}(j)) \end{cases}$$

$$\begin{pmatrix} f_{n, j} \\ a_{n, j} \end{pmatrix}$$

ЗАДАЧІ ДИНАМІЧНОГО ПРОГРАМУВАННЯ ТА ЦІЛОЧИСЛОВІ МОДЕЛІ

Розв'язок задачі реалізований за допомогою програми - функції $fdist(d, g)$, аргументами якої є матриця відстаней між вузлами етапів d , та вектор g , що містить інформацію про кількість етапів та кількість вузлів в кожному етапі. Функція $v(n, g)$ введена для зручності та визначає сумарну кількість вузлів до n -го етапу включно. Результатом програми є знаходження найкоротшого шляху між двома пунктами та відповідної траєкторії. З решти функціями та операторами, що використовуються можна ознайомитися за відповідними посібниками присвяченими математичному пакету Mathcad ([11, 15]), та за довідкою самої системи. Нижче наведемо розв'язок задачі приведеної в [12] (ст. 413).

$$d_{1,2} := 7 \quad d_{1,3} := 8 \quad d_{1,4} := 5$$

$$d_{2,5} := 12 \quad d_{2,6} := \infty$$

$$d_{3,5} := 8 \quad d_{3,6} := 9$$

$$d_{4,5} := 7 \quad d_{4,6} := 13$$

$$d_{5,7} := 9 \quad d_{6,7} := 6$$

$$g := (1 \ 3 \ 2 \ 1)^T \quad fdist(d, g) = \begin{pmatrix} 21 \\ "1-4-5-7" \end{pmatrix}$$

1. Беллман Р. Динамическое программирование. – М.: Изд-во иностр. лит., 1960. – 400 с.

2. Беллман Р., Дрейфус С. Прикладные задачи динамического программирования. – М.: Наука,

1965. – 458 с.

3. Беллман Р., Калаба Р. Динамическое программирование и современная теория управления. – М.: Наука, 1969. – 118 с.

4. Воскобойников Ю. Е, Воскобойникова Т.Н. Программирование в математическом пакете Mathcad. Новосибирск. НГСУ –1999. – С. 35.

5. Динамическое программирование. // www.karelia.ru/psu/Faculties/Forest/courses/

6. Зайченко Ю.П. Дослідження операцій: Підручник. – К.: ВІПОЛ, 2000.

7. Исследование операций в экономике: Учеб. пособие. / Под ред. Н.Ш. Кремера. – М.: Банки и биржи, ЮНИТИ. – 1999.

8. Моисеев Н.Н. Элементы теории управления оптимальных систем. – М.: Наука, 1975. – 526 с.

9. Моделирование линейных процессов. Пакет Lindo. // <http://economy.mari.ru/test/Lect2/lec6.html>

10. Пакет Линго. // <http://economy.mari.ru/test/Lect2/lec9.html>

11. Плис А.И., Сливина Н.А. Mathcad: математический практикум. – М.: Финансы и Статистика. – 1999.

12. Таха Х. Введение в исследование операций. – М.: Издат. дом "Вильямс", 2001. – 912 с.

13. Черноусько Ф. Л., Меликян А.А. Игровые задачи управления и поиска. – М.: Наука, 1978. – 270 с.

14. Черноусько Ф. Л. Динамическое программирование. // Соросовский образовательный журнал, №2, 1998

15. Янішевський В.С. Вивчення основ лінійного програмування з допомогою математичного пакету Mathcad. // Молодь і ринок. – 2003., №3, С. 30–35.



Дрогобич. Дерев'яна церква св. Юра. 1654.