— перехід за рівнем складності задля відслідкування глибини розуміння поняття, оскільки однією з характеристик тестового завдання є ключове поняття певної теми та при неправильній відповіді надається тест з таким же ключовим поняттям;

— перехід між ключовими поняттями теми зі зміною рівня складності тестового завдання для відслідкування глибини розуміння теми;

— формування індивідуального сценарію тестування студента;

— аналіз структурованості знань студента на основі результатів тестування та побудова індивідуальної карти прогалин знань.

Подальші дослідження спрямовані на реалізацію індивідуальних сценаріїв теоретичного навчання, які будуть враховувати результати тестування та індивідуальні карти прогалин знань і доповнювати теоретичний матеріал погано засвоєними поняттями.

Література

1. *Звонников В.И., Челышкова М.Б.* Современные средства оценивания результатов обучения. — М.: Издательский центр «Академия», 2007. — 224 с.

2. *Сметанюк Л.В., Кравцов Г.М.* К теории и практике использования адаптивных тестов // Інформаційні технології в освіті. Випуск 3. — Херсон. — 2009. — С. 148 -155.

3. *Федорук П.І., Масловський С.М.* Модель адаптивного тестування з нечіткою логікою // Математичні машини і системи. — 2009. — № 1. — С. 131-137.

УДК 656.13

АЛГОРИТМ ДИНАМІЧНОГО ПРОГРАМУВАННЯ ОПТИМАЛЬНОГО РОЗПОДІЛУ РЕСУРСІВ

Доктор фізико-математичних наук Гавриленко В.В.,
Цуканов О.І.,
Шумейко О.А.

У статті пропонується алгоритм оптимального розподілу ресурсів з метою одержання максимального економічного ефекту за допомогою динамічного програмування.

The paper proposes the optimal allocation of resources to obtain maximum economic benefit by using dynamic programming.

Вступ. Динамічне програмування є одним із методів оптимального програмування, у якому процес прийняття рішень і управління може бути розбитий на окремі етапи (кроки). Плануючи багатокроковий процес, на кожному кроці обирається управління з урахуванням його майбутніх наслідків на тих кроках, які ще є попереду. Лише на останньому кроці можна прийняти рішення, яке дасть максимальний ефект.

Коли всі умовно-оптимальні управління на всіх кроках відомі, то це означає, що визначено, як необхідно керувати на кожному кроці, яким би не був процес на початку. В такому разі можна знайти не умовно-оптимальне, а й оптимальне управління.

Динамічне програмування, використовуючи поетапне планування, дозволяє значно спростити алгоритм розв'язання задачі та розв'язувати ті із них, до яких не можна застосовувати методи математичного аналізу.

Однак динамічне програмування має свої особливості. На відміну від лінійного програмування, у якому симплексний метод є універсальним, у динамічному програмуванні такого методу не існує. Кожна задача є специфічною і у кожному випадку необхідно розробляти підходящий алгоритм розв'язання.

Основна частина. У загальному вигляді задача динамічного програмування ставиться так. Нехай аналізується деякий керований процес, представлення якого допускає декомпозицію на послідовні стани (кроки), кількість яких n задана. Ефективність всього процесу F може бути представлена як сума ефектив-

© Гавриленко В.В., Цуканов О.І., Шумейко О.А., 2010

ностей F_j $(j=1,...,n)$ окремих кроків. З кожним етапом (кроком) задачі пов'язане прийняття певного рішення x_j $(j=1,...,n)$, що визначає ефективність як даного етапу, так і всього процесу в цілому.

Розв'язування задачі динамічного програмування полягає в знаходженні такого управління процесом у цілому $X=(x_1,...,x_n)$, яке максимізує загальний прибуток:

$$\max F=\sum_{j=1}^{n} F_j .$$

Оптимальним розв'язком цієї задачі є управління X^*, що складається з сукупності оптимальних покрокових управлінь $X^*=(x^*_1,...,x_n^*)$, і забезпечує досягнення максимальної ефективності.

Загальний алгоритм розв'язування задачі динамічного програмування складається із послідовності таких операцій:

1. Визначають показники станів досліджуваної системи і множину параметрів, що описують ці стани.

2. Поділяють процес на етапи, які відповідають певним його станам (періодам планування динамічного процесу, або окремим об'єктам системи) для підготовки рішень у керуванні процесом.

3. Формулюють перелік управлінь x_j $(j=1,...,n)$ для кожного етапу і відповідні обмеження до них.

4. Визначають ефект, який забезпечує управління x_j на j-му кроці, якщо перед тим система була у стані S, у вигляді функції ефективності $g_j(S, x_j)$.

5. Визначають, як змінюється стан S системи під впливом управління x_j на j-му кроці, тобто як здійснюється перехід до нового стану $S^*=\phi_j(S, x_j)$.

6. Визначають рекурентну залежність задачі динамічного програмування, що визначає умовний оптимальний ефект $F_j(S)$, починаючи з 1-го кроку і до останнього, через вже відому функцію $F_{j-1}(S^*)$:

$$F_j(S)=\max_{x_j}\{g_j(S, x_j)+F_{j-1}(S^*, x_j)\}. \tag{1}$$

Цьому ефекту відповідає умовне оптимальне управління на і-му кроці $x_j(S)$.

7. Використовуючи умовну оптимізацію останнього $n-1$-го кроку, визначаємо множину станів S, з яких можна за один крок дійти до кінцевого стану. Умовний оптимальний ефект на n-му кроці обчислюють за формулою: $F_n(S)=\max\limits_{x_n}\{g_n(S, x_n)\}$.

Потім знаходять умовне оптимальне управління $x_n(S)$, в результаті реалізації якого цей максимум було досягнуто.

8. Починаючи з 1-го кроку, проводять умовну оптимізацію 2-го, 3-го та інших кроків за рекурентними залежностями (1) і визначають для кожного кроку умовне оптимальне управління: $x_j^*(S_j)$.

9. Виконавши обчислення для всіх n етапів, проводять безумовну оптимізацію управління у зворотному напрямку від кінцевого стану S_n до початкового S_1. Для цього, з урахуванням визначеного оптимального управління на останньому кроці x_n^*, визначають стан системи на попередньому кроці. Потім для

цього нового стану знаходять оптимальне управління x_{n-1}^* і аналогічно повторюють ці дії до першого етапу. В результаті знаходять оптимальне покрокове управління $X^* = (x_1^*,...,x_n^*)$, що забезпечує максимальний ефект F^*.

Згідно з технологією Mathcad, у якому робота алгоритму реалізується на числових даних, розглянемо приклад оптимізації проекту розподілу інвестицій у транспортній системі.

Припустимо, що розглядається n варіантів нарощування провізних здатностей транспортної фірми для збільшення об'ємів перевезень і відповідно зростання прибутку на 4-х об'єктах. За умови оновлення рухомого складу і модернізації сервісного обслуговування фірма може досягти певного приросту прибутку.

Протягом цього періоду вона має можливість вкладати кошти у розмірі k гр. одиниць. Вкладення коштів буде здійснюватись за чотирма варіантами проекту в обсягах k_i, $i = 1,2,...,m$ гр. од. Приріст прибутку фірми від вкладених коштів за варіантами проекту представлені у наступній таблиці.

Приріст прибутку фірми за варіантами проекту

Кошти, гр. од. k_i	Варіант 1 $g_1(k_i)$	Варіант 2 $g_2(k_i)$	Варіант 3 $g_3(k_i)$	Варіант 4 $g_4(k_i)$
10	8	10	12	11
20	18	20	21	23
30	25	28	26	30
40	36	40	38	35
50	46	48	47	50

де $g_j(k_i)$ – прибуток проекту за j-м варіантом за умови інвестування в нього коштів об'ємом k_i гр. од..

Основою динамічного програмування є функціональне рівняння, яке рекурентно зв'язує значення цільових функцій $f_j(x)$ і $f_{j-1}(x)$ відповідно на j-му і j–1-му кроці. Це рівняння має вигляд:

$$f_1(x) = g_1 x),$$
$$f_j(x) = \max\left\{g_j(k_i) + f_{j-1}(x - k_i)\right\}, \quad i = 1,..., m; \quad j = 2,..., n, \qquad (2)$$

де x – загальна сума виділених коштів, k_i – сума коштів, яка виділяється на j-му кроці.

Для представлення функціонального рівняння (2) у векторно-матричному вигляді, що потрібно для запису алгоритму у Mathcad, перейдемо від функцій $g_j(x_i)$ до індексованих величин $g_{i,j}$, де $i = 1,...,m$ відповідає об'єму коштів k_i, $j = 1,...,n$ — номеру варіанту проекту. При цьому величина $f_{i,j-1}$ відповідає значенню функції $f_{j-1}(x - k_i)$.

Тепер рівняння (2) буде представлене у векторно-матричному вигляді таким чином:

$$f_{i,1} = g_{i,1}, \quad i = 1,..., m;$$
$$f_{i,j} = \max_{0 \le k_i \le k} \left(g_{i,j} + f_{i,j-1}\right), \quad j = 2,..., 4. \qquad (3)$$

Задача розв'язується поетапно. Етапами розв'язання задачі будуть послідовні вкладення коштів у сумісні варіанти проекту. Отже маємо чотирьохетапну задачу динамічного програмування. Алгоритм розв'язання задачі складається із чотирьох етапів.

1-й етап. Задачу починаємо розв'язувати із визначення кількості коштів у 1-й варіант проекту. Якщо маємо в розпорядження кошти k_i, $(i = 1,...,m)$ гр. од., то їх ефективність відповідає прибутку $f_{i,1} = g_{i,1}$, що буде отриманий від інвестування в перший варіант проекту k_i — гр. од. Максимальний прибуток (ефективність 1-го етапу) дорівнюватиме

$$f_1(k_i) = \max_{k_i} \{g_{i,1}\}, \quad i = 1,...,m.$$

2-й етап. На цьому етапі розподіляємо кошти k_i $(i = 1,...,m)$ між двома варіантами проекту і зіставляємо ефективності прийнятих рішень на попередньому і поточному етапах. Якщо позначити ефективності другого етапу через $f_{i,2}$ $i = 1,...,m$, то маємо рівняння:

$$f_{i,2} = \max_{k_i} \{g_{i,2} + f_{i,1}\}, \quad i = 1,...,m.$$

За цим рівнянням визначаємо оптимальні рішення на 2-му етапі.

3-й етап. Кошти k_i $(i = 1,...,m)$ розподіляються між трьома варіантами проекту. Знову необхідно зіставити ефективності попереднього та поточного етапів. Отже, використовуємо дані, що визначають прибуток, який можна отримати від вкладення коштів одразу в перший та другий варіант проекту $(f_{i,2},\ i = 1,...,m)$ та прибуток від вкладення коштів в третій варіант проекту. Використовуємо формулу

$$f_{i,3} = \max_{k_i} \{g_{i,3} + f_{i,2}\}, \quad i = 1,...,m.$$

4-й етап. Аналогічно продовжуємо процес розрахунків для 4-го етапу, розподіляючи кошти k_i $(i = 1,...,m)$ між чотирма варіантами проекту.

В результаті цих кроків одержуємо умовно-оптимальні управління процесом розподілу інвестицій від 1-го до 4-го етапу. Згідно з методом динамічного програмування тепер потрібно розглянути процес безумовної оптимізації, розглядаючи результати етапів у зворотному порядку від 4-г до 1-го.

$$\mathrm{ORIGIN} := 1$$

$$k := \begin{pmatrix} 10 \\ 20 \\ 30 \\ 40 \\ 50 \end{pmatrix} \qquad g := \begin{pmatrix} 8 & 10 & 12 & 11 \\ 18 & 20 & 21 & 23 \\ 25 & 28 & 26 & 30 \\ 36 & 40 & 38 & 35 \\ 46 & 48 & 47 & 50 \end{pmatrix}$$

Пошук умовно-оптимальних управлінь (прямий хід алгоритму)

1-й етап

$$i := 1..5 \quad f1_i := g_{i,1} \qquad f1 = (8\ \ 18\ \ 25\ \ 36\ \ 46) \quad F1 := \max(f1) \qquad F1 = 46$$

2-й етап

$$k_1 = 10 \quad f2_{1,1} := g_{1,1} + 0 \quad f2_{1,2} := 0 + g_{1,2} \quad f2 = (8\ \ 10) \quad F2_1 := \max(f2) \quad F2_1 = 10$$

$k_2 = 20$ $f2_{2,1} := g_{2,2} + 0$ $f2_{2,2} := g_{1,2} + F1_1$ $f2_{2,3} := F1_2 + 0$

$f2 := \text{submatrix}(f2, 2, 2, 1, 3)$ $f2 = (20\ \ 18\ \ 18)$ $F2_2 := \max(f2)$ $F2_2 = 20$

$k_3 = 30$ $f2_{3,1} := g_{3,2} + 0$ $f2_{3,2} := g_{2,2} + F1_1$ $f2_{3,3} := g_{1,2} + F1_2$ $f2_{3,4} := 0 + F1_3$

$f2 := \text{submatrix}(f2, 3, 3, 1, 4)$ $f2 = (28\ \ 28\ \ 28\ \ 25)$ $F2_3 := \max(f2)$ $F2_3 = 28$

$k_4 = 40$ $f2_{4,1} := g_{4,2} + 0$ $f2_{4,2} := g_{3,2} + F1_1$ $f2_{4,3} := g_{2,2} + F1_2$ $f2_{4,4} := g_{1,2} + F1_3$

$f2_{4,5} := 0 + F1_4$

$f2 := \text{submatrix}(f2, 4, 4, 1, 4)$ $f2 = (40\ \ 36\ \ 38\ \ 35)$ $F2_4 := \max(f2)$ $F2_4 = 40$

$k_5 = 50$ $f2_{5,1} := g_{5,2} + 0$ $f2_{5,2} := g_{4,2} + F1_1$ $f2_{5,3} := g_{3,2} + F1_2$ $f2_{5,4} := g_{2,2} + F1_3$

$f2_{5,5} := g_{1,2} + F1_4$

$f2 := \text{submatrix}(f2, 5, 5, 1, 5)$ $f2 = (48\ \ 48\ \ 46\ \ 45\ \ 46)$

$F2_5 := \max(f2)$ $F2_5 = 48$

$$F2^T = (10\ \ 20\ \ 28\ \ 40\ \ 48)$$

3-й етап

$k_1 = 10$ $f3_{1,1} := g_{1,3} + 0$ $f3_{1,2} := F2_1 + 0$ $f3 = (12\ \ 10)$

$f3 := \text{submatrix}(f3, 1, 1, 1, 2)$ $f3 = (12\ \ 10)$ $F3_1 := \max(f3)$ $F3_1 = 12$

$k_2 = 20$ $f3_{2,1} := g_{2,3} + 0$ $f3_{2,2} := g_{1,3} + F2_1$ $f3_{2,3} := F2_2$

$f3 := \text{submatrix}(f3, 2, 2, 1, 3)$ $f3 = (21\ \ 22\ \ 20)$ $F3_2 := \max(f3)$ $F3_2 = 22$

$k_3 = 30$ $f3_{3,1} := g_{3,3} + 0$ $f3_{3,2} := g_{2,3} + F2_1$ $f3_{3,3} := g_{1,3} + F2_2$ $f3_{3,4} := F2_3 + 0$

$f3 := \text{submatrix}(f3, 3, 3, 1, 4)$ $f3 = (26\ \ 31\ \ 32\ \ 28)$ $F3_3 := \max(f3)$ $F3_3 = 32$

$k_4 = 40$ $f3_{4,1} := g_{4,3} + 0$ $f3_{4,2} := g_{3,3} + F2_1$ $f3_{4,3} := g_{2,3} + F2_2$ $f3_{4,4} := g_{1,2} + F2_3$

$f3_{4,5} := 0 + F2_4$

$f3 := \text{submatrix}(f3, 4, 4, 1, 5)$ $f3 = (38\ \ 36\ \ 41\ \ 38\ \ 40)$ $F3_4 := \max(f3)$ $F3_4 = 41$

$k_5 = 50$ $f3_{5,1} := g_{5,3} + 0$ $f3_{5,2} := g_{4,3} + F2_1$ $f3_{5,3} := g_{3,3} + F2_2$ $f3_{5,4} := g_{2,3} + F2_3$

$f3_{5,5} := g_{1,3} + F2_4$ $f3_{5,6} := 0 + F2_5$

$f3 := \text{submatrix}(f3, 5, 5, 1, 6)$ $f3 = (47\ \ 48\ \ 46\ \ 49\ \ 52\ \ 48)$ $F3_5 := \max(f3)$ $F3_5 = 52$

$$F3^T = (12\ \ 22\ \ 32\ \ 41\ \ 52)$$

4-й етап

$k_1 = 10$ $f4_{1,1} := g_{1,4} + 0$ $f4_{1,2} := F3_1 + 0$ $f4 = (11\ \ 12)$ $F4_1 := \max(f4)$ $F4_1 = 12$

$k_2 = 20 \quad f4_{2,1} := g_{2,4} + 0 \quad f4_{2,2} := g_{1,4} + F3_1 \quad f4_{2,3} := 0 + F3_2$

$f4 := \text{submatrix}(f4, 2, 2, 1, 3) \quad f4 = (23 \ \ 23 \ \ 22) \quad F4_2 := \max(f4) \quad F4_2 = 23$

$k_3 = 30 \quad f4_{3,1} := g_{3,4} + 0 \quad f4_{3,2} := g_{2,4} + F3_1 \quad f4_{3,3} := g_{1,4} + F3_2 \quad f4_{3,4} := 0 + F3_3$

$f4 := \text{submatrix}(f4, 3, 3, 1, 4) \quad f4 = (30 \ \ 35 \ \ 33 \ \ 32) \quad F4_3 := \max(f4) \quad F4_3 = 35$

$k_4 = 40 \quad f4_{4,1} := g_{4,4} + 0 \quad f4_{4,2} := g_{3,4} + F3_1 \quad f4_{4,3} := g_{2,4} + F3_2 \quad f4_{4,4} := g_{1,4} + F3_3$

$f4_{4,5} := 0 + F3_4$

$f4 := \text{submatrix}(f4, 4, 4, 1, 5) \quad f4 = (35 \ \ 42 \ \ 45 \ \ 43 \ \ 41) \quad F4_4 := \max(f4) \quad F4_4 = 45$

$k_5 = 50 \quad f4_{5,1} := g_{5,4} + 0 \quad f4_{5,2} := g_{4,4} + F3_1 \quad f4_{5,3} := g_{3,4} + F3_2 \quad f4_{5,4} := g_{2,4} + F3_3$

$f4_{5,5} := g_{1,4} + F3_4 \quad f4_{5,6} := 0 + F3_5$

$f4 := \text{submatrix}(f4, 5, 5, 1, 6) \quad f4 = (50 \ \ 47 \ \ 52 \ \ 55 \ \ 52 \ \ 52) \quad F4_5 := \max(f4) \quad F4_5 = 55$

$$F4^T = (12 \ \ 23 \ \ 35 \ \ 45 \ \ 55)$$

Пошук безумовних оптимальних управлінь (зворотний хід алгоритму) по кроках.

1. Отже оптимальний розв'язок на 4-му етапі розв'язання задачі дає максимальний прибуток у розмірі $F_{max} = 55$ гр. од.: $F_{max} := \max(F1, F2, F3, F4)$ $F_{max} = 55$. Це значення одержується із відповідного виразу на 4-му кроці алгоритму $f4_{5,4} := g_{2,4} + F3_3$ $f4_{5,4} = 55$, що і дає $F4_5 = 55$. Це означає, що за 4-м варіантом проекту за рахунок коштів $k_2 = 20$ гр. од. можна одержати $g_{2,4} = 23$ гр. од. прибутку. Залишок коштів складає 50 — 20=30 гр. од.

2. Продовжуючи пошук оптимального плану, із виразу $f4_{5,4} = g_{2,4} + F3_3$ маємо $F3_3 = 32$. Цьому значенню відповідає вираз $f3_{3,3} = g_{1,3} + F2_2$, який дає значення $f3_{3,3} = 32$. У вираз $f3_{3,3}$ входить величина $g_{1,3} = 12$, яка означає, що на 3-му кроці, тобто за 3-м варіантом проекту, одержується прибуток у сумі 12 гр. од. Сума витрачених коштів складає 10 гр. од., залишок — $30 - 10 = 20$ гр. од.

3. На цьому етапі маємо $F2_2 = 20$, що одержується із виразу $f2_{2,1} = g_{2,2} + 0$. Це означає, що на 2-му етапі, тобто за другим варіантом проекту, за рахунок 20 гр. од. коштів одержується прибуток у сумі $g_{2,2} = 20$ гр. од. Залишок коштів дорівнює 0. Таким чином, оптимальним розв'язком задачі буде:

$$k_1^* = 0, \ \ g_1^* = 0; \quad k_2^* = 20, \quad g_2^* = 20; \quad k_3^* = 10, \quad g_3^* = 12; \quad k_4^* = 20, \quad g_4^* = 23.$$

На цьому алгоритм розв'язання задачі оптимального розподілу ресурсів закінчується.

Висновок. У статті пропонується технологія реалізації моделей оптимального розподілу ресурсів з метою одержання максимального економічного ефекту у розглядуваній системі. Алгоритм реалізації моделі побудований у математичній системі Mathcad. Алгоритм має універсальний характер і може бути реалізований у будь-якій іншій системі. Він побудований на представлені основного функціонального рівняння динамічного програмування у матрично-векторному вигляді. Це дає можливість наочно представити та перебрати усі варіанти можливих рішень на кожному кроці алгоритму і, таким чином, представити його у вигляді досить простої структури.