Міністерство освіти і науки України

Сумський державний університет

Кафедра комп’ютерних наук

Секція інформаційно-комунікаційних технологій

Лабораторна робота №12

З дисципліни

«Математичні методи дослідження операцій»

Викладач О. А. Шовкопляс

Студент Бабич З. А.

Група ІН-11.2

Варіант 3

Суми 2023

Теоретична частина

Динамічне програмування - спосіб вирішення складних завдань шляхом розбиття їх на більш прості підзадачі. Найбільш доцільно динамічне програмування застосовувати для вирішення таких практичних задач, в яких пошук оптимального рішення вимагає покрокового підходу.

Динамічне програмування, як і парадигма «Розділяй і володарюй», дозволяє вирішувати завдання, комбінуючи вирішення допоміжних завдань. Відмінність цього методу в тому, що допоміжні завдання не є незалежними, тобто різні допоміжні завдання використовують рішення одних і тих же підзадач.

Принцип оптимальності Белмана передбачає, що управління на кожному окремому кроці повинно обиратися з урахуванням наслідків від його впровадження в майбутньому. Таким чином, принцип динамічного програмування вимагає знаходження на кожному кроці умовного оптимального управління для будь-якого з можливих результатів попереднього кроку [1].

Динамічне програмування дістало великого поширення для задач дослідження операцій, в яких фактор часу не враховується.

На основі принципу оптимальності Белмана будується схема рішення багатокрокової завдання, що складається з 2-х частин:

1) Зворотний хід: від останнього кроку до першого отримують множину можливих оптимальних («умовно-оптимальних») управлінь.

2) Прямий хід: від відомого початкового стану до останнього стану з отриманої множини «умовно-оптимальних» управлінь складається шукане оптимальне управління для всього процесу в цілому.

Типовий алгоритм вирішення завдань методом динамічного програмування:

1. Описати будову оптимальних рішень.
2. Виписати рекурентні співвідношення, що зв'язують оптимальні значення параметра для підзадач.
3. Рухаючись від кінця до початку, обчислити оптимальне значення параметра для підзадач.
4. Користуючись отриманою інформацією, побудувати оптимальне рішення.

Типові задачі з використанням методу динамічного програмування:

1. Задача про обчислення чисел Фібоначчі;
2. Задача про пошук найбільшої загальної підпослідовності;
3. Задача пошуку найбільшої зростающої підпослідовності;
4. Задача про відстань Левенштейна;
5. Задача про порядок множення матриць;
6. Задачи про вибір траєкторії;
7. Задача послідовного прийняття рішення;
8. Задача про використання робочої сили;
9. Задача управління запасами;
10. Задача про ранці;
11. Алгоритм Флойда-Уоршелла: знайти найкоротші відстані між усіма вершинами зваженого орієнтованого графа;
12. Алгоритм Белмана-Форда: знайти найкоротший шлях в зваженому графі між двома заданими вершинами;

Метод Белмана - це алгоритм, який використовується для знаходження найкоротшого маршруту в орієнтованому або неорієнтованому графі. Цей метод заснований на принципі динамічного програмування та може бути застосований до графів з зважаними ребрами.

Основна ідея методу Белмана полягає в тому, щоб спочатку оцінити відстань від початкової вершини до всіх інших вершин у графі. Алгоритм повторює цю оцінку для кожної вершини у графі, оновлюючи оцінки, поки не будуть знайдені найкоротші маршрути до всіх вершин.

Процес знаходження найкоротшого маршруту методом Белмана складається з наступних кроків:

1. Початкова вершина має вагу 0, а всі інші вершини мають нескінченну вагу. Крім того, створюється масив для збереження найкоротших відстаней від початкової вершини до кожної іншої вершини.
2. Алгоритм ітеративно проглядає всі ребра графа та оновлює найкоротші відстані до вершин, якщо відстань через поточне ребро є коротшою. Цей крок повторюється V-1 разів (де V - кількість вершин у графі), щоб гарантувати знаходження найкоротшого маршруту.
3. Виявлення циклів з від'ємною вагою: Після V-1 ітерацій необхідно перевірити, чи є у графі цикли з від'ємною вагою. Це можна зробити, виконавши ще одну ітерацію алгоритму. Якщо під час цієї ітерації ваги вершин знову оновлюються, то це свідчить про наявність циклу з від'ємною вагою.
4. Після закінчення V-1 ітерацій, найкоротший маршрут від початкової вершини до будь-якої іншої вершини вже буде знайдений. Якщо після V-1 ітерацій відбулося оновлення ваги хоча б однієї вершини, то це означає наявність циклу з від'ємною вагою, і найкоротший маршрут не існує.

Метод Белмана ефективно працює з графами будь-якого типу і розміру, але має часову складність O(V\*E), де V - кількість вершин, E - кількість ребер у графі.

Якщо порівнювати цей алгоритм с алгоритмом Дейкстри, то алгоритм Белмана-Форда буде повільнішим, але є і переваги в тому, що він є універсальний, оскільки може працювати с графами у яких вага ребра може бути від’ємного значення. Але якщо граф має від’ємний цикл, тоді для даного графу не існує дерево найкоротших шляхів.

Практична частина

Постановка задачі:

На розвиток трьох підприємств виділено 5 млн грн. Відома ефективність капітальних вкладень у кожне підприємство, задана значенням нелінійної функції gі(хі), поданої в таблиці. Необхідно розподілити виділені кошти між підприємствами таким чином, щоб одержати максимальний сумарний дохід.  
Для спрощення розрахунків припускаємо, що розподіл коштів здійснюється в цілих числах хі = {0, 1, 2, 3, 4, 5} млн грн.

**Таблиця 1-Вхідні дані**

|  |  |  |  |
| --- | --- | --- | --- |
| x | g1(x) | g2(x) | g3(x) |
| 0 | 0 | 0 | 0 |
| 1 | 2,3 | 1,9 | 4 |
| 2 | 4,5 | 3,8 | 5 |
| 3 | 6 | 5 | 5,4 |
| 4 | 8,7 | 6,8 | 7,2 |
| 5 | 9,5 | 8 | 9 |

Розв’язання:

**I етап. Умовна оптимізація.**

*1-й крок: k = 1.*

Припустимо, що всі засоби в кількості δ4 = 100 ум. од. віддані першому підприємству. У цьому випадку, як видно з таблиці 2, максимальний дохід становитиме 9,5 ум. од.

Отже *F1(C1)=g1(x1)*.

**Таблиця 2-Крок 1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| õ1  C1 | 0 | 1 | 2 | 3 | 4 | 5 | F1(C1) | õ1\* |
| 0 | 0 | – | – | – | – | – | 0 | 0 |
| 1 | – | 2,3 | – | – | – | – | 2,3 | 1 |
| 2 | – | – | 4,5 | – | – | – | 4,5 | 2 |
| 3 | – | – | – | 6 | – | – | 6 | 3 |
| 4 | – | – | – | – | 8,7 | – | 8,7 | 4 |
| 5 | – | – | – | – | – | 9,5 | 9,5 | 5 |

(*õn\**- кількість ресурсів відданих n-тому підприємству)

*2-й крок*: *k = 2.*

Визначаємо оптимальну стратегію при розподілі коштів між першим і другим підприємствами. При цьому рекурентне співвідношення Белмана має вигляд:

*F2(C2)=*{*g2(x2)+F1(C2-x2)*}, на основі якого складено таблицю 3.

**Таблиця 3-Крок 2**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| õ2  C2 | 0 | 1 | 2 | 3 | 4 | 5 | F2(C2) | õ2\* |
| 0 | 0+0 | – | – | – | – | – | 0 | 0 |
| 1 | 0+2,3 | 1,9+0 | – | – | – | – | 2,3 | 0 |
| 2 | 0+4,5 | 1,9+2,3 | 3,8+0 | – | – | – | 4,5 | 0 |
| 3 | 0+6 | 1,9+4,5 | 3,8+2,3 | 5+0 | – | – | 6,4 | 1 |
| 4 | 0+8,7 | 1,9+6 | 3,8+4,5 | 5+2,3 | 6,8+0 | – | 8,7 | 0 |
| 5 | 0+9,5 | 1,9+8,7 | 3,8+6 | 5+4,5 | 6,8+2,3 | 8+0 | 10,6 | 1 |

*3-й крок*: *k = 3.*

Визначаємо оптимальну стратегію при розподілі коштів між першим, другим і третім підприємствами. При цьому рекурентне співвідношення Белмана має вигляд:

*F3(C3)=*{*g3(x3)+F2(C3-x3)*}, на основі якого складено таблицю 4.

**Таблиця 4-Крок 3**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| õ3  C3 | 0 | 1 | 2 | 3 | 4 | 5 | F3(C3) | õ3\* |
| 0 | 0+0 | – | – | – | – | – | 0 | 0 |
| 1 | 0+2,3 | 4+0 | – | – | – | – | 4 | 1 |
| 2 | 0+4,5 | 4+2,3 | 5+0 | – | – | – | 6,3 | 1 |
| 3 | 0+6,4 | 4+4,5 | 5+2,3 | 5,4+0 | – | – | 8,5 | 1 |
| 4 | 0+8,7 | 4+6,4 | 5+4,5 | 5,4+2,3 | 7,2+0 | – | 10,4 | 1 |
| 5 | 0+10,6 | 4+8,7 | 5+6,4 | 5,4+4,5 | 7,2+2,3 | 9+0 | 12,7 | 1 |

**II етап. Безумовна оптимізація**

Визначаємо компоненти оптимальної стратегії.

1-й крок. За даними таблиці 4 максимальний  прибуток при розподілі 5 ум. од. між трьома підприємствами становить: 12,7 ум. од. При цьому третьому підприємству потрібно виділити δ3\*=1 ум. од.

2-й крок. Визначаємо величину залишених грошових коштів, що припадають на частку другого і першого підприємств: 5 – 1 = 4 ум. од.

За даними таблиці 3 знаходимо, що оптимальний варіант розподілу коштів розміром 4 ум. од. між другим і першим підприємствами становить: F2(4) = 8,7 при виділенні другому підприємству δ2\*= 0 ум. од.

3-й крок. Визначаємо величину залишених грошових коштів, що припадають на частку першого підприємства: 4 – 0 = 4 ум. од.

Знаходимо: F1(4) = 8,7 при виділенні першому підприємству∙ δ1\*= 4 ум. од.

Таким чином, оптимальний план інвестування підприємств=(4,0,1) який забезпечить максимальний прибуток *F(5)=g1(4)+g2(0)+g3(1)=8,7+0+4=12,7* ум. од.

Висновки

Під час виконання лабораторного завдання було розв’язано задачу динамічного програмування про оптимальний розподіл ресурсів. Отримані знання про основні особливості методів дослідження операцій, умови їх правильного використання, можливості адаптації до конкретних задач певних галузей науки і техніки. Був опрацьований та успішно застосований метод Белмана.

Список джерел

1. Лавров Є.А. Математичні методи дослідження операцій: підручник / Є.А. Лавров, Л.П. Перхун, В.В. Шендрик , Е. Г. Кузнєцов, Ю. В. Парфененко, В. А. Сергієнко/ за ред. Є.А. Лаврова. –Суми : Сумський державний університет, 2017. –222 с.(86-87 стр.)
2. Платформа Mix: Алгоритми і структури даних / Тема 5. Методи та парадигми теорії алгоритмів / Конспект лекції [Електронний ресурс]. – Режим доступу: <https://mix.sumdu.edu.ua/textbooks/24090/379536/index.html#p2>
3. Принцип Белмана в динамічному програмуванні [Електронний ресурс]. – Режим доступу: <https://ekmair.ukma.edu.ua/server/api/core/bitstreams/33a061df-f901-4679-9aef-96b238583583/content>
4. Динамічне програмування [Електронний ресурс]. – Режим доступу: <http://uk.wikipedia.org/wiki/Динамічне_програмування>
5. Беллман Р. Динамическое програмирование. / Ричард Беллман [пер. с английского И.М. Андреезой, А.А Корбута, И. В. Романовского, И. Н. Соколовой]. – Москва: 1960. – 400с
6. Методи динамічного програмування - Мeтодичні вказівки [Електронний ресурс]. – Режим доступу: <https://web.posibnyky.vntu.edu.ua/fmbt/avto6_bilichenko_modelyuvtehproces_avtotransportu/p9.html>