

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра математики факультету інформатики

Курсова робота
за спеціальністю 113 Прикладна математика

“Принцип Беллмана у динамічному програмуванні”

Керівник курсової роботи
доц. Щестюк Н. Ю.
(підпис)

“ ” 2020 р.

Виконав студент 4 курсу
Факультету
інформатики

Коваль А. В.

“ ” 2020 р.

Київ 2020

Тема: Принцип Беллмана у динамічному програмуванні

Календарний план виконання роботи:

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	02.11.2019	
2.	Огляд технічної літератури за темою роботи.	15.11.2019	
3.	Опрацювання матеріалів	17.11.2019	
4.	Проведення аналізу	20.01.2019	
5.	Написання роботи	06.03.2019	
6.	Перевірка та погодження роботи	06.05.2019	
7.	Створення презентації та написання доповіді	10.05.2019	
8.	Остаточне оформлення роботи	15.05.2019	
9.	Захист курсової роботи	20.05.2019	

Студент _____

Керівник _____

“ ”

ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ МЕТОДУ ДИНАМІЧНОГО ПРОГРАМУВАННЯ.....	5
1.1. Основи динамічного програмування	5
1.2. Математична постановка задачі динамічного програмування.	
Методи функціональних рівнянь Р. Беллмана.....	13
РОЗДІЛ 2. ПРАКТИЧНЕ ЗАСТОСУВАННЯ МЕТОДУ ДИНАМІЧНОГО ПРОГРАМУВАННЯ	
2.1. Використання методу динамічного програмування для розв’язання задачі про знаходження найкоротшого маршруту.....	
ВИСНОВКИ.....	
СПИСОК ВИКОРИСТАНИХ	
ДЖЕРЕЛ.....	

ВСТУП

У даній курсовій роботі було досліджено теоретичні аспекти методу динамічного програмування. По своїй суті динамічне програмування є методом прийняття рішення, яке відрізняється тим, що бере до уваги критерій часу та порядок виконання операцій, які розглядаються. Даний метод вважається більш складним, ніж методи статичної оптимізації задач.

Актуальність курсової роботи зумовлена тим, що на сьогодні у багатьох економічних та виробничих завдань виникає необхідність врахування зміни процесу, який моделюється, у часі та впливу часу на критерій оптимальності. Метод динамічного програмування, запропонований на початку 50-х років американським вченим Р. Беллманом, передбачає вирішення поставлених задач оптимального управління.

Використовуючи метод динамічного програмування для певного процесу, яким можна керувати, серед множини усіх можливих варіантів управління процесом шукають найбільш оптимальний, керуючись деяким заданим критерієм. Такий оптимальний варіант управління полягає у знаходженні екстремуму, тобто максимального або мінімального значення цієї цільової функції. У свою чергу, функція є числовою характеристикою процесу.

Особливістю методу є багатоступеневість, або розподіл прийняття управлінських рішень на ряд послідовних етапів, що відповідають, як правило, різним моментам часу.

Загалом, методи динамічного програмування можуть бути використані не тільки у дискретних, але і в неперервних процесах, якими можна керувати. Наприклад, у таких процесах, де потрібно приймати рішення періодично в певний інтервал часу.

Метою даної курсової роботи є вивчення теоретичних аспектів динамічного програмування і вирішення задачі на побудову оптимального транспортного маршруту.

Об'єктом практичного дослідження виступає логістична задача, ціль якої - знаходження найкоротшого шляху з використанням динамічного програмування, а саме метод Беллмана, який забезпечує оптимальний маршрут.

Для досягнення мети були окреслені наступні завдання: вивчити емпіричний матеріал за даною темою; ознайомитись з алгоритмом математичної постановки завдань методом динамічного програмування; розв'язати задачу на знаходження найкоротшого маршруту.

Предметом дослідження є теоретико-методичні засади методу динамічного програмування та його практичне застосування на прикладі задачі на знаходження найкоротшого маршруту.

РОЗДІЛ 1

ТЕОРЕТИЧНІ АСПЕКТИ МЕТОДУ ДИНАМІЧНОГО ПРОГРАМУВАННЯ

1.1. Основи динамічного програмування

Більшість методів дослідження операцій спрямовані на розв'язання задач, які мають конкретне завдання. Виявилось, що класичний математичний апарат не може повністю задовольнити потреби багатьох задач оптимізації, які містять багато змінних або обмежень у вигляді нерівностей або систем.

Саме тому з'являється ідея, на якій ґрунтується метод динамічного програмування: розбити задачі більшої розмірності на, так звані, підзадачі з невеликою кількістю змінних, та подальшого розв'язання всієї задачі по частинах.

Термін динамічного програмування був запропонований Річардом Беллманом, ще в 40-х роках минулого століття. Такою назвою він хотів привернути увагу до своєї роботи. Під терміном “динамічне” він мав на увазі, що проблема оптимального управління, яку він вирішував цим методом є залежною від часу, тобто вона постійно змінюється. Водночас, “програмування” в цьому контексті є синонімом слова планування, тобто розуміється як послідовність дій, які заплановані.

Динамічне програмування є методом спрощення задачі, тобто математичною оптимізацією. В основу метода покладений, так званий, принцип оптимальності. На його основі будується поетапне розв'язання

задачі, що допускає декомпозицію. Дія цього методу полягає в тому, щоб розбити об'ємну задачу на підзадачі, використовуючи в більшій частині метод рекурсії для розбиття. Якщо задачі будуть розбиті таким методом, то буде існувати певна залежність між розв'язком загальної задачі, і розв'язком підзадач, тобто підзадачі можуть бути розв'язані необмежену кількість раз, але загальна задача буде мати один розв'язок.

Отже, можна сформулювати “принцип оптимальності”, покладений в основу усіх задач ДП:

“Незважаючи на стан системи S перед наступним етапом прийняття управлінського рішення, необхідно вибрати такий варіант управління, щоб сума доходу на цьому та всіх послідовуючих кроках були максимальними”.

Наступний алгоритм слід використовувати для постановки завдань динамічного програмування:

- обрати змінні, що задають певний стан керованої системи S перед кожним наступним її кроком;
- визначити етапи операції;
- визначити множину варіантів прийняття рішень X_i для кожного кроку й обмеження, що накладають на них;
- скласти функцію, яка відповідає нашому виграшу, іншими словами, визначити, який виграш можуть принести системі S варіанти прийняття рішень кроку X_i ;
- зрозуміти, як змінюється стан S системи під впливом управлінських рішень на кожному кроці;
- записати основне рекурентне рівняння, що виражає певний умовний оптимальний виграш $W_i(S)$ через функцію $W_{i+1}(S)$. Цьому виграшу відповідає умовне оптимальне управлінське рішення на i -му кроці $X_i(S)$.

- умовно оптимізувати останній (m -ий) крок, використавши множину станів S . Визначивши для кожного окремого стану суму виграшу по формулі, можна обчислити виграш для кінцевого стану.
- зробити умовну оптимізацію.

Якщо ми знаємо стартовий стан системи, якою можемо керувати і яка потребує прийняття рішення щодо подальшого управління нею, то варіювати таку систему на стартовому кроці не потрібно. Ми відразу можемо знайти оптимальний виграш для цього стартового стану S_0 . Він і буде нашим оптимальним виграшем за всю операцію.

Автор вище згадує, так званий, початок етапу, який являє собою час ухвалення рішення (наприклад: в економічному контексті це передбачає величину капітальних вкладень, про заміну устаткування певного виду й інші).

За допомогою цього методу можливо розв'язати задачі, до яких ми не можемо застосувати інші методи. Значне спрощення операції розв'язання забезпечується за рахунок того, що ми зменшуємо кількість наших варіантів.

Використовуючи поетапний процес ухвалення рішень, ми не повинні забувати, що на кожному конкретному етапі повинні обдумати вплив нашого рішення на весь процес в цілому, тобто маючи на увазі кінцеву мету.

Проте слід сказати, що динамічного програмування має свої недоліки. Якщо порівнювати його з лінійним програмуванням, де ми можемо використовувати один універсальний метод (симплексний метод), то у динамічного такого універсального методу немає. Кожна задача викликає певні, притаманні лише їй, труднощі, і кожного разу необхідно

знаходити найбільш оптимальну методику рішення. Інший недолік методу - це трудомісткість розв'язків задач, які є досить багатовимірними.

Враховуючи велику кількість змінних, розв'язок завдання навіть на сучасних комп'ютерах може обмежуватися пам'яттю й швидкістю приладу.

1.2. Математична постановка задачі динамічного програмування.

Методи функціональних рівнянь Р. Беллмана

Можна висловити припущення, що існує якась керована система S , яка знаходиться у стартовому стані (S_0). Завдяки управлінню U система переходить з свого стартового стану (S_0) в кінцевий стан (S_k). Ми можемо охарактеризувати якість прийнятих рішень щодо управління заданою системою певним значенням функції $W(U)$. Наша задача полягає у тому, щоб знайти одне найбільш оптимальне рішення щодо управління системою U^* серед безлічі інших рішень, при якому б функція $W(U)$ приймала значення екстремуму, тобто досягала максимуму або мінімуму.

Завдання методу можна інтерпретувати геометрично. Сам стан фізичної системи S можна охарактеризувати числовими значеннями, наприклад швидкістю або іншими. Ми можемо назвати такі значення координатами системи. Отже, стан досліджуваної системи можна позначити точкою S , а перехід із одного стану в інший – траєкторією точки S . Управління U означає вибір певної траєкторії переміщення точки S з точки S_1 в точку S_2 , тобто встановлення певного закону руху крапки S .

Взагалі всю сукупність станів, яких може набувати система, називають множиною можливих станів. Відповідно до кількості значень,

що описують стан системи, множина можливих станів може бути різною. Можна розглянути приклад (рис. 1.1), при якому стан системи S описується лише одним параметром, – координатою x . Якщо координата змінюється, точка S переміщується по осі Ox . Таким чином, множиною можливих станів системи є певна сукупність значень x , а керуванням – закон руху точки S з стартового стану S_0 у кінцеве S_k по осі Ox .

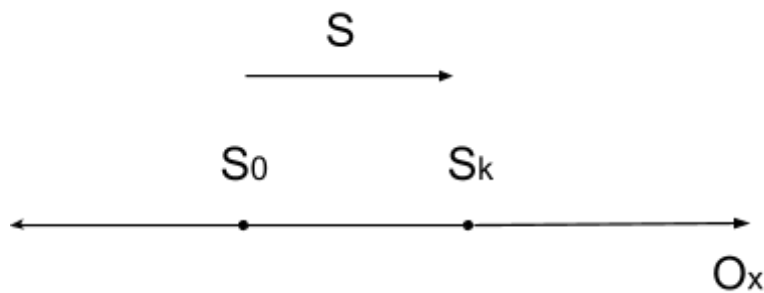


Рис. 1.1. Графічне зображення переходу системи S

Отже, ми можемо сформулювати певну геометричну інтерпретацію завдання. Необхідно обрати траєкторію, яка належить сукупності станів системи S , при якій функція W прийматиме найбільш оптимальне значення.

Необхідно ввести позначення і припущення, що для того, щоб дослідити розв'язання задач методом динамічного програмування.

Якщо взяти K -ий етап ($k=1, n$) досліджуваної системи S , то його можна визначити множиною значень $X(k) = (x_1(k), x_2(k), \dots, x_n(k))$ при втіленні управління u_k . При цьому система S перейшла зі стану $X(k-1)$ у стан $X(k)$. Можемо припустити, що новий стан $X(k)$ досліджуваної системи залежить від моделі управління та даного етапу $X(k-1)$ і не залежить від способу, завдяки якому система прийшла в новий етап.

Якщо як результат втілення k -го кроку ми отримуємо якийсь дохід або виграш, які залежать від стартового стану системи $X(k-1)$ обраного керування u_k і рівний $W_k(X(k-1), u_k)$, то загальний дохід за n кроків становить:

$$F = \sum_{k=1}^n W_k(X^{(k-1)}, u_k) \quad (1.1)$$

Таким чином, задача має відповідати двом умовам. Перша умова зазвичай називається умовою відсутності післядії, а друга – умовою адитивності цільової функції завдання.

Для задачі можна сформулювати принцип оптимальності Беллмана, якщо задовольняється виконання першої умови. Оптимальна стратегія керування - це множина керувань $U^*=(u_1^*, u_2^*, \dots, u_n^*)$, при умові реалізації яких система S за n кроків переходить зі стартового стану $X(0)$ у кінцевий $X(k)$ і при цьому функція (1.1) приймає максимальне значення.

Принцип оптимальності свідчить, що для знаходження показника оптимальної стратегії керування спочатку необхідно визначити оптимальну стратегію для останнього кроку n , далі для двох останніх кроків, для трьох останніх кроків тощо, і так до першого кроку.

Для того, щоб знайти розв'язок ймовірно необхідно передбачити результат передостаннього кроку і вибрати стратегію керування, яка забезпечує максимізацію значення функції $W_n(X(n-1), u_n)$. Такий вид керування, яке обирається, звертаючи увагу на припущення щодо результатів попереднього кроку, називається показник умовно оптимальним керуванням. Таким чином, принцип оптимальності потребує

знаходження показника умовно оптимального керування на кожному етапі з можливих варіантів попереднього етапу.

Таким чином, ми визначили, що задача методу динамічного програмування має задовольняти дві умови. Перша умова зазвичай називається умовою відсутності післядії, а друга – умовою адитивності цільової функції завдання.

Для задачі можна сформулювати принцип оптимальності Беллмана, якщо задовольняється виконання першої умови. Оптимальна стратегія керування - це множина стратегій керування $U^*=(u_1^*, u_2^*, \dots, u_n^*)$, при умові втіленні яких система S за n кроків переходить із стартового стану $X(0)$ у кінцевий $X(k)$ і при цьому функція (1.1) досягає екстремуму, тобто набуває максимального значення.

Принцип оптимальності свідчить, що для знаходження оптимальної стратегії керування спочатку необхідно знайти оптимальну стратегію для останнього кроку n , далі для двох останніх кроків, для трьох останніх кроків тощо, і так до першого кроку.

Для того, щоб знайти розв'язок ймовірно необхідно передбачити результат передостаннього кроку і вибрати стратегію керування, яка забезпечує максимізацію значення функції $W_n(X(n-1), u_n)$. Такий вид керування, яке обирається, звертаючи увагу на припущення щодо результатів попереднього кроку, називається умовно оптимальним керуванням. Таким чином, принцип оптимальності вимагає знаходити на кожному кроці умовно оптимальне керування для кожного з можливих варіантів попереднього кроку.

Тепер ми можемо сформулювати принцип оптимальності математично, щоб мати змогу використати його практично. Для цього вводимо додаткові позначення. Визначимо максимальний дохід за n етапів

після переходу системи S зі стартового стану $X(0)$ у кінцевий стан $X(k)$ як $F_n(X_0)$.

$F_{n-k}(X^{(k)})$ – максимальний дохід, який можна отримати при зміні стану з $X(k)$ у кінцевий стан $X^{(n)}$, використовуючи оптимальну стратегію керування. Тоді:

$$F_n(X^0) = \max[W_1(X^{(0)}, u_1) + \dots + W_n(X^{(n-1)}, u_n)] \quad (1.2)$$

$$F_{n-k}(X^{(k)}) = \max[W_{k+1}(X^{(k)}, u_{k+1}) + F_{n-k-1}(X^{(k+1)})] \quad (k=0, n-1) \quad (1.3)$$

Останній вираз є математичним записом принципу оптимальності, і називається основним функціональним рівнянням Беллмана, або ж рекурентним співвідношенням. Саме за допомогою даного рівняння можна розв'язати задачі методом динамічного програмування.

Беручи $k=n-1$ у рекурентному співвідношенні (1.3), отримуємо наступне функціональне рівняння:

$$F_1(X^{(n-1)}) = \max[W_n(X^{(n-1)}, u_n) + F_0(X^{(n)})] \quad (1.4)$$

Значення функції для останнього кроку будемо вважати відомим. Тепер, розглядаючи всі можливі параметри системи S на $(n-1)$ – м кроці $X_1^{(n-1)}, X_2^{(n-1)}, \dots, X_m^{(n-1)}, \dots$, знайдемо умовні оптимальні рішення $u_n^0(x_1^{(n-1)}), u_n^0(x_2^{(n-1)}), \dots, u_n^0(x_m^{(n-1)}), \dots$

і відповідні значення функції (1.4)

$$F_1^0(X_1^{(n-1)}), F_1^0(X_2^{(n-1)}), \dots, F_1^0(X_m^{(n-1)}), \dots$$

Отже, ми знаходимо показник для умовно оптимального керування для досліджуваної системи S . Що б не сталося в системі після даного $(n-1)$ -го етапу, нам буде зрозуміло, яке прийняти нам рішення на n -му етапі. Тепер ми розглянемо функціональне рівняння при якому $k=n-2$:

$$F_0(X^{(n-2)}) = \max[W_n(X^{(n-2)}, u_{n-1}) + F_0(X^{(n-1)})]$$

Тобто, нам треба знайти значення функцій $W_{n-1}(X^{(n-2)}, u_{n-1})$ і $F_1(X^{(n-1)})$.. Якщо врахувати, що значення другої функції вже визначено, то залишається обчислити першу для деякої множини значень X та стратегій. За допомогою цих обчислень можливо буде визначити показник умовно оптимального керування $u_0(n-1)$ для всіх значень $X(n-2)$. Кожний показник керування у сумі з керуванням, яке було обране на останньому етапі, забезпечить досягнення максимального значення доходу на двох останніх етапах.

Отже, повторюючи процес, який ми описали вище, можна поступово дійти до стартового етапу. Після цього ми знатимемо, у якому стані знаходиться система. Після цього залишається обрати найкращий показник умовно оптимальних керувань, які були прийняті на всіх наступних етапах.

Отже, завдяки послідовному проходженню усіх етапів визначається максимальний виграш за n етапів, а потім знаходиться показник умовно оптимального керування для кожного з них.

Для того, щоб визначити найбільш оптимальну стратегію керування, тобто розв'язати задачу, тепер необхідно пройти весь алгоритм етапів від початку до кінця. Це можна зробити наступним чином: на першому етапі беремо показник умовно оптимального керування u_0 у якості

оптимального керування $u1^*$. Далі шукаємо, який стан має система при $X1^*$. Цей стан забезпечує керування $u1^*$ і визначає знайдене умовно оптимальне $u20$, що тепер вважається оптимальним. Знаючи $u2^*$, знаходимо $X2^*$ тощо.

Отже, проаналізувавши дане розв'язання, ми можемо бачити, що його процес тривалим та складним і містить багато обчислень. Це є причиною, чому більш складні завдання розв'язують за допомогою програмного забезпечення.

РОЗДІЛ 2

ПРАКТИЧНЕ ЗАСТОСУВАННЯ МЕТОДУ ДИНАМІЧНОГО ПРОГРАМУВАННЯ

2.1. Використання методу динамічного програмування для розв’язання задачі про знаходження найкоротшого маршруту

З метою практичного вивчення методу динамічного програмування була поставлена задача на знаходження найкоротшого маршруту. Для цього були зібрані дані про проміжні точки між будинком автора та будівлею НаУКМА.

Для детальнішого вивчення маршруту обрали програмний продукт Google Карти. Далі автор розглядає три можливі варіанти маршруту (рис. 2.1).

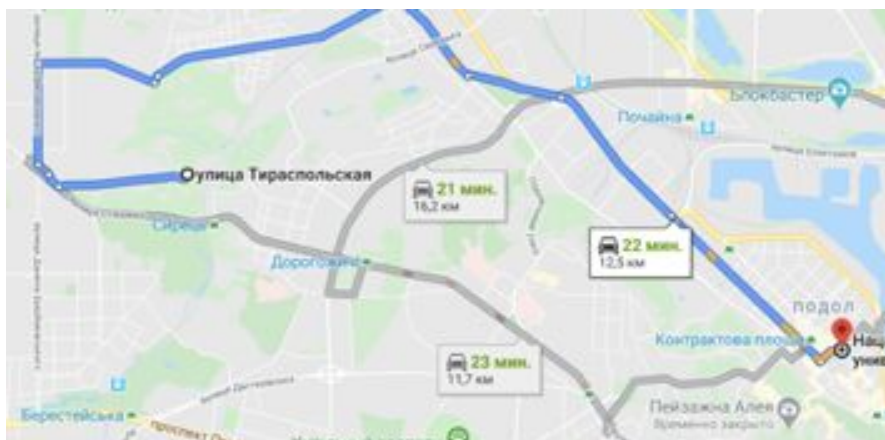


Рис. 2.1 - Карта маршрутів, використаних для розв'язання задачі про знаходження найкоротшого маршрута

Наступним кроком є встановлення проміжних точок для побудови графа. Вони будуть позначатися латинськими буквами для скорочення (А, В, С тощо). Вага ребер графа буде визначатися відстанню між ними в кілометрах, тобто, якщо довжина частини маршруту становитиме один кілометр, то вага ребра, який відповідає цьому маршруту буде дорівнювати одиниці.

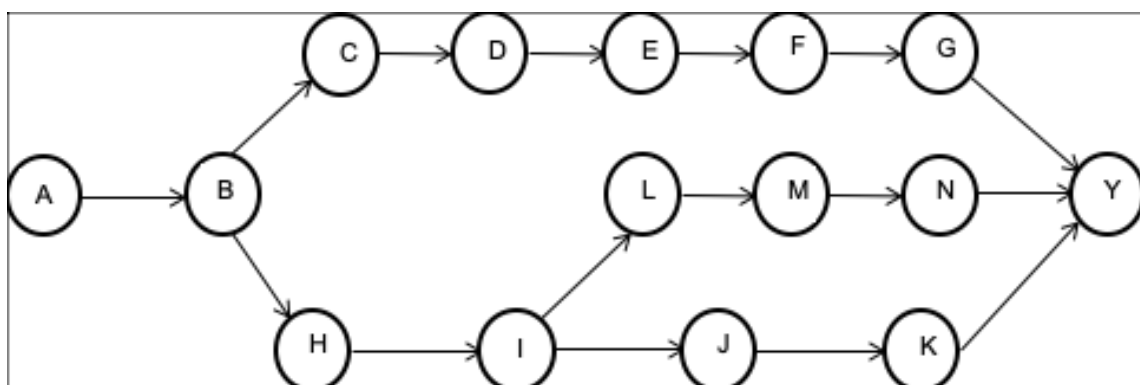


Рис. 2.2 - Граф проміжних точок використаних маршрутів

Вага ребер становить:

AB, BC, CD, GY, NY, HI = 1;

BH, IJ, KY, EF = 2;

DE, FG, MN, JK = 3;

LM = 4;

IL = 5.

Позначимо через d_{ij} відстань між суміжними вузлами i та j , а U_j – найкоротшу відстань між першим та j -им вузлами. За методом Беллмана запишемо формулу $f_{N-i}(s_i) = \min\{r_i(s_i, u_i) + f_{N-(i+1)}(s_{i+1}) : u_i\}$, $i=0, \dots, N-1$ такому вигляді:

$$U_j = \min\{U_i + d_{ij} : i=1, \dots, j-1\}, j=A, \dots, Y.$$

Знайдемо поетапно U_j .

$$1. U_A = 0$$

$$2. U_B = U_A + d_{AB} = 0 + 1 = 1,$$

$$3. U_C = U_B + d_{BC} = 1 + 2 = 3,$$

$$4. U_H = U_C + d_{CH} = 3 + 2 = 5,$$

Після виконання цих дій на кожному кроці ми отримуємо:

$$5. U_Y = \min\{U_G + d_{GY}, U_N + d_{NY}, U_K + d_{KY}\} = \min\{12, 16, 11\} = 11$$

Отже, ми розрахували всі можливі випадки побудови маршрутів. Після того, як ми отримали мінімальну відстань між вузлами А і Y, яка дорівнює 11, то найкоротшим маршрутом буде: А-Н-І-К-У.

Таким чином, ми розв'язали задачу, використавши метод динамічного програмування на практиці.

ВИСНОВКИ

На сьогодні у багатьох економічних та виробничих завдань виникає необхідність врахування зміни процесу, який моделюється, у часі та впливу часу на критерій оптимальності. Метод динамічного програмування, запропонований на початку 50-х років американським вченим Р. Беллманом, передбачає вирішення поставлених задач оптимального управління.

В основі методу лежить принцип оптимальності:

“Незважаючи на стан системи S перед черговим етапом прийняття управлінського рішення, необхідно вибрати такий варіант управління, щоб виграш на даному кроці плюс оптимальний виграш на всіх наступних кроках були максимальними”.

Для детальнішого вивчення методу динамічного програмування у курсовій роботі було розв'язано задачу на знаходження найкоротшого маршруту. Для цього спочатку побудували маршрути та встановили

проміжні вузли. Потім побудували граф, надавши ребрам значень, відповідно до довжини частини шляху. Використавши метод Беллмана, було розраховано найкоротший шлях, тим самим, практично використавши метод динамічного програмування.