УДК 519.651.3

М.Е. Боднюк, А.Н. Знайдюк, Л.А. Байкова

Иркутский государственный университет путей сообщения, г. Иркутск, Российская Федерация

РЕШЕНИЕ ЗАДАЧИ УПРАВЛЕНИЯ МЕТОДОМ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Аннотация. *В статье описывается понятие динамического программирования. Рассмотрена задача, в которой применены знания математического метода динамического программирования. Решение задачи представлено с помощью применения программного кода.*

Ключевые слова: *математика, динамическое программирование.*

M.E. Bodnyuk, A.N. Znaydiuk, L.A. Baykova

Irkutsk State Transport University, Irkutsk, the Russian Federation

SOLVING THE CONTROL PROBLEM BY DYNAMIC PROGRAMMING METHOD

Abstract. *The article describes the concept of dynamic programming. The problem in which the knowledge of mathematics by the method of dynamic programming is applied is considered. The solution of the problem is presented using the application of program code.*

Keywords: *mathematics, dynamic programming.*

Введение

Метод динамического программирования – это метод решения задач оптимизации, который позволяет разбить задачу на более простые подзадачи и решить их последовательно. Этот метод широко используется в различных областях, включая экономику, финансы, управление производством, биологию, транспорт и другие. Разработка и изучение математических методов решения задач представляет большой теоретический интерес. Также весьма важным и актуальным является вопрос о практической роли и значимости математических методов для решения задач. Подобные вопросы, в настоящее время, как никогда актуальны и важны.

История вопроса

Понятие «динамическое программирование» первым использовал Ричард Беллман в 50-е годы 20 века. Он внес значительный вклад в развитие теории оптимизации и динамического программирования. Некоторые известные задачи, которые были разработаны Р. Беллманом: задача о рюкзаке, о поиске кратчайшего пути, о нахождении максимальной последовательности, о расписании работ, о редакционном расстоянии. Эти задачи динамического программирования имеют широкое применение в различных областях и являются основой для решения более сложных задач.

Основная часть

Одним из разделов прикладной математики «исследование операций» является понятие динамического программирования.

«Исследование операций – это построение и разработка математических моделей по поиску и принятию оптимальных решений». Каждый набор значений x1,x2,…,xn, удовлетворяющий всем ограничениям задач, называется ее решением. Областью допустимых решений называется множество всех решений задачи. «Оптимальное решение задачи – это аспект исследования операций, в который включается анализ и решение математических задач выбора в заданном множестве допустимых решений X элемента, удовлетворяющего определенным критериям оптимальности». Решить заданную задачу математического программирования, значит найти оптимальный план и значение искомой функции.

Математическое программирование (оптимизация) – это область математики, которая занимается разработкой методов и алгоритмов для решения задач оптимизации. «Оптимизация включает в себя максимизацию или минимизацию целевой функции, которая зависит от набора переменных и подчиняется определенным ограничениям». Цель математического программирования заключается в поиске оптимального решения задачи, которое удовлетворяет заданным ограничениям и достигает наилучшего результата.

В математическом программировании существуют следующие разделы, такие как линейное программирование, где целевая функция и ограничения являются линейными; динамическое программирование – это метод решения задач, в которых решение строится из решений подзадач; нелинейное программирование используется в задачах нахождения минимума или максимума нелинейной функции нескольких переменных, причем на переменные либо есть ограничения в виде равенств или неравенств, либо нет; выпуклое программирование – это раздел, в котором целевая функция и ограничения являются выпуклыми и т.д.

Оптимальная структура в динамическом программировании представляет идею того, что решение сложной задачи может быть разбито на более простые подзадачи, решение которых может быть использовано для получения решения исходной задачи. При этом, оптимальное решение каждой подзадачи может быть сохранено и использовано в дальнейшем, что позволяет избежать повторных вычислений и ускорить процесс решения задачи. Оптимальная структура также предполагает, что решение каждой подзадачи зависит только от решений более простых подзадач, а не от решений более сложных задач. Это позволяет использовать принципы декомпозиции и модульности при проектировании алгоритмов динамического программирования.

Виды методов динамического программирования:

– нисходящее динамическое программирование: производится запоминание результатов вычисления некоторых подзадач, т.е. тех, которые могут повториться при дальнейших решениях;

– восходящее динамическое программирование: производится переводом сложной задачи в последовательность простых подзадач.

Алгоритм метода динамического программирования включает следующие шаги:

– определение целевой функции, которую необходимо оптимизировать;
– разбиение задачи на более мелкие подзадачи;
– определение зависимостей между подзадачами;
– разработка рекурсивной формулы для вычисления значений подзадач;
– разработка алгоритма для вычисления значений целевой функции путем комбинации решений более мелких подзадач;
– вычисление оптимального решения.

Задача

Существует лестница, которая состоит из n ступеней. Перед этой лестницей находится человек, которому необходимо по ней подняться. За один шаг он может подняться на одну ступень выше, либо перейти через одну и подняться на 2 ступени. Сколько способов может применить человек, для того чтобы добраться до n ступени?

Решение:

До начала решения необходимо рассмотреть подзадачи, которые можно использовать для дальнейшего решения:

1. Лестница имеет всего одну ступень (рис. 1)

Рис.1. F(N) = 1

2.Лестница имеет 2 ступени (рис. 2)

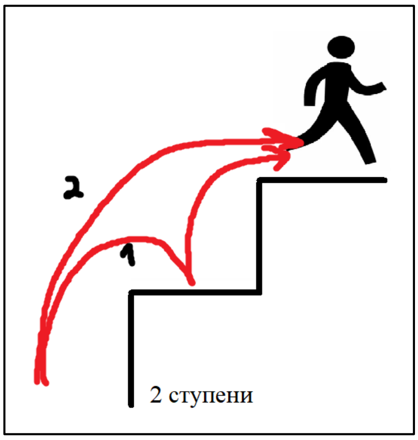

Рис. 2. F(N) = 2

3. Лестница состоит из 3-х ступеней (рис. 3)

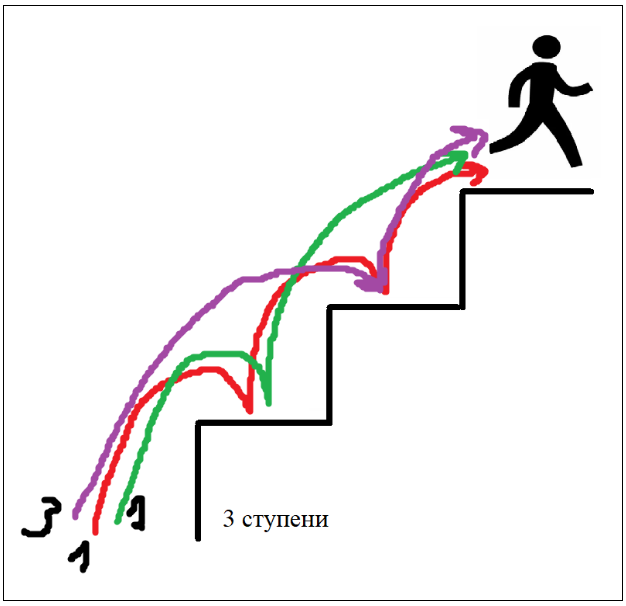

Рис. 3. F(N) = 3

4. Лестница состоит из 4-х ступеней (рис. 4)

Здесь мы можем воспользоваться примером шагов до 2 ступени, но при условии, что мы прибавим полный шаг через 1 ступень, к каждому способу.

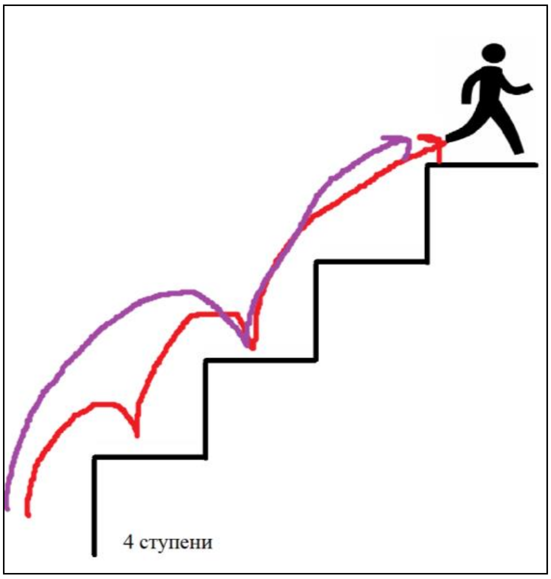

Рис. 4. $f(N_i) = f(N_i) - 2$

На данный момент мы имеем 3 способа, чтобы подняться на 3 ступень, если мы к каждому из них добавим шаг на одну ступеньку выше, то получим 3 способа попасть на 4 ступеньку (рис. 5).

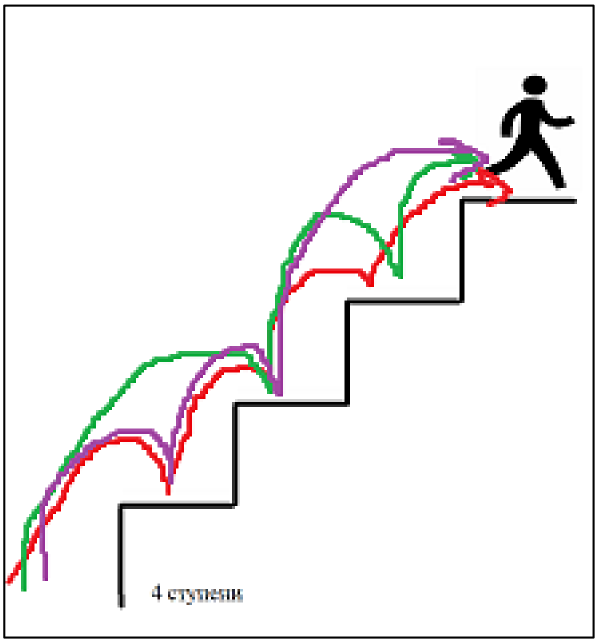

Рис. 5. $f(N_i) = f(N_i) - 1$

Отсюда следует, существует всего f(N) способов попасть на i ступень $f(N_i) = f(N_{i-1}) + f(N_{i-2})$.

Реализация решения с помощью языка программирования C# и Visual Studio показана на рис. 6.

```
// C # программа для подсчета числа
// способов добраться до лестницы, когда
// человек может подняться на 1, 2, ..м
// лестницы за один раз
using System;
Ссылок: 0
class GFG
{
    // рекурсивная функция
    // используется countWays
    Ссылок: 1
    static int countWaysUtil(int n, int m){
        int[] res = new int[n];
        res[0] = 1;
        res[1] = 1;
        for (int i = 2; i < n; i++){
            res[i] = 0;
            for (int j = 1; j <= m && j <= i; j++)
                res[i] += res[i - j];
        }
        return res[n - 1];
    }
    // Возвращает количество способов
    // чтобы добраться до лестницы
    Ссылок: 1
    static int countWays(int s, int m){
        return countWaysUtil(s + 1, m);

    }
    Ссылок: 0
    public static void Main(){
        int s = 0, m = 2;
        Console.Write("Введите количество ступеней - ");
        s = int.Parse(Console.ReadLine());
        Console.WriteLine("Количество способов = " + countWays(s, m));
    }
}
```

```
Консоль отладки Microsoft Visual Studio
Введите количество ступеней - 9
Количество способов = 55
```

Рис. 6. Реализация решения

Данный продукт не является совершенным и представлен, как образец возможной реализации процесса использования динамических методов для решения задач.

Заключение

Таким образом, метод динамического программирования позволяет решать задачи управления, разбивая их на более простые подзадачи и последовательно решая их. В данном случае, мы использовали метод динамического программирования для определения количества способов, которыми человек может подняться по лестнице. Этот метод широко используется в различных областях и позволяет получать оптимальные решения с минимальными затратами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бхаргава А. Грокаем алгоритмы. Иллюстрирование пособие для программистов и любопытствующих//Учебное пособие/ Питер, 2017. С.288.
2. Письменный Д.Т. Конспект лекций по высшей математике: полный курс//Учебное пособие/Москва Айрис пресс, 2013. С 608.
3. Щедрин Н.И., Кархов А.Н. Математические методы программирования в экономике//Учебное пособие/Москва, 1974. С 144.
4. А.А. Мицель. Методы оптимизации//Учебное пособие/Томск, 2016. С 68.
5. В. Д. Власенко. Динамическое и стохастическое программирование//Учебное пособие/Хабаровск, 2008 С 37.
6. Т.Е.Гришкина, Н.Н.Двоерядкина. Динамическое программирование//Учебное пособие//Амур, 2019. С 38.

7. Калихман И.Л. Войтенко М.А. Динамическое программирование в примерах и задачах//Учебное пособие/Москва, 1979. С 129.

REFERENCES

1. Bhargava A. Grokaem algorithms. Illustrating a manual for programmers and the curious//Study guide/ St. Petersburg, 2017. p.288.
2. Written D.T. Lecture notes on higher mathematics: full course//Study guide/Moscow Iris Press, 2013. P. 608.
3. Shchedrin N.I., Karkhov A.N. Mathematical methods of programming in economics//Study guide/Moscow, 1974. From 144.
4. A.A. Mizel. Optimization methods//Study guide/Tomsk, 2016. From 68.
5. V. D. Vlasenko. Dynamic and stochastic programming//Textbook/Khabarovsk, 2008 With 37.
6. T.E.Grishkina, N.N.Dvoeryadkina. Dynamic programming//Study guide//Amur, 2019. From 38.
7. Kalikhman I.L. Voitenko M.A. Dynamic programming in examples and problems//Study guide/Moscow, 1979. From 129.

Информация об авторах

Знайдюк Алексей Николаевич – студент 2 курса факультета «Управление на транспорте и информационные технологии» специальность «Разработка программно-информационных систем», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: Znaidyuk00@gmail.com

Боднюк Максим Евгеньевич – студент 2 курса факультета «Управление на транспорте и информационные технологии» специальность «Разработка программно-информационных систем», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: maks.bodnyuk@mail.ru

Байкова Людмила Анатольевна – старший преподаватель кафедры «Математика», Иркутский государственный университет путей сообщения, г. Иркутск, e-mail: baykova_la@irgups.ru

Information about the authors

Alexey Nicolayevich Znaydyuk – 1st year student of the Faculty «Transport Management and Information Technologies» specialty «Development of software and information systems», Irkutsk State Transport University, Irkutsk, e-mail: Znaidyuk00@gmail.com

Maxim Yevgenyevich Bodnyuk – 1st year student of the Faculty «Transport Management and Information Technologies» specialty «Development of software and information systems», Irkutsk State Transport University, Irkutsk, e-mail: maks.bodnyuk@mail.ru

Lyudmila Anatolievna Baykova – Senior Lecturer of the Department of Mathematics, Irkutsk State Transport University, Irkutsk, e-mail: baykova_la@irgups.ru