МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Московский авиационный институт (национальный исследовательский университет)»

Ступинский филиал МАИ

Кафедра «Моделирование систем и информационные технологии»

**РЕФЕРАТ**

**Динамическое программирование**

Студент: Гутарова В.С.

Группа: ТСО-205Б-22

Москва 2024

### *ОГЛАВЛЕНИЕ*

[ОГЛАВЛЕНИЕ 2](#_Toc170821392)

[ОПРЕДЕЛЕНИЕ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ 3](#_Toc170821393)

[ОСНОВНЫЕ ПОНЯТИЯ И ПРИНЦИПЫ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ 4](#_Toc170821394)

[МЕТОДЫ РЕШЕНИЯ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ 4](#_Toc170821395)

[ПРИМЕРЫ ЗАДАЧ, РЕШАЕМЫХ С ПОМОЩЬЮ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ 4](#_Toc170821396)

[ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ 5](#_Toc170821397)

[ВЫВОДЫ 5](#_Toc170821398)

[СПИСОК ЛИТЕРАТУРЫ 5](#_Toc170821399)

### *ОПРЕДЕЛЕНИЕ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ*

Понятие «динамическое программирование» часто встречается в теории управления и теории вычислительных машин. Динамическое программирование – это метод нахождения оптимального решения сложных задач путем разбиения их на более простые. Этот метод широко применим к решению задач, которые разбиваются на подзадачи, перекрывающие друг друга. Таким образом простая задача решается только один раз и используется в последующем решении других подзадач. В этом заключается основная идея динамического программирования – вместо многократного решения одной и той же простой задачи сохранять результаты ее решения и использовать повторно с целью ускорения решения глобальной задачи.

Благодаря этому динамическое программирование применяется для решения задач на поиск оптимального решения: задача о рюкзаке, поиск кратчайшего пути в графе, задачи на оптимизацию и многие другие.

Ввел идею и термин динамического программирования в использование американский математик Ричард Беллман. Так он впервые ввел термин «динамическое программирование», чтобы описать процесс нахождения оптимального решения путем использования для решения каждой последующей задачи ответа на предыдущую.

Важно понимать, что название «динамическое программирование» изначально не имеет прямой связи с программированием как процессом написания кода. Понятие «программирование» в данном контексте используется как процесс составления оптимальной последовательности действий с целью получения решения. При этом слово «динамическое» означает, что при решении задач важную роль играет время решения и порядок.

Динамическое программирование применяется в различных областях науки и техники, в особенности в сферах, требующих обработки больших объемов данных и выполнения сложных вычислений. К таким областям относятся биоинформатика, финансовая математика, искусственный интеллект и машинное обучение, лингвистика, робототехника, строительство и управление ресурсами. Так, динамическое программирование используется для сравнения цепочек ДНК и РНК, оценки финансовых инструментов, автоматического перевода и сравнения текстов и многого другого.

### *ОСНОВНЫЕ ПОНЯТИЯ И ПРИНЦИПЫ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ*

Для правильного понимания метода динамического программирования важно ознакомиться с основными понятиями, которые используются в данном методе. А именно с понятиями многошаговости, стратегии управления и оптимальной стратегии управления.

Многошаговость в динамическом программировании подразумевает многоступенчатую структуру процесса решения задачи или возможность разбить сложную задачу на ряд последовательных этапов (шагов). Многошаговость может внедряться в решение задачи естественно, то есть задача по своей структуре подразумевает выполнение определенных шагов. К таким случаям можно отнести задачи планирования и управления, которые можно разбить на временные промежутки: день, месяц, квартал, год и т.д.. Кроме того, многошаговость может внедряться искусственно, когда необходимо адаптировать условие задачи для обеспечения возможности использования динамического программирования.

Также в методе динамического программирования часто встречается понятие стратегии управления. Под этим термином понимается любая последовательность действий, способная привести систему из начального состояния в конечное. Стратегия управления, доставляющая функции цели экстремальное значение, называется оптимальной.

Метод динамического программирования основывается на принципе оптимальности. Принцип оптимальности в динамическом программировании заключается в выборе наиболее выгодного (оптимального) для конечного результата решения на данном шаге. Такой подход к решению задач позволяет гарантировать оптимальный результат не для данной подзадачи, а для главной задачи в целом. Ричард Беллман сформулировал этот принцип следующим образом: если некоторая последовательность решений оптимальна, то на любом шаге последующие решения образуют оптимальную стратегию по отношению к результату предыдущих решений.

Кроме того, в динамическом программировании распространен термин «мемоизация», или англ. «memoization», образован от английских слов «memory», то есть «память, воспоминания», и «optimization», то есть «оптимизация». Таким образом, это технология использования памяти (кэша) с целью оптимизации процесса. Данный принцип используется в динамическом программировании с целью увеличения скорости вычислений путем предотвращения многократных повторений одних и тех же этапов вычислений. Результат выполнения предыдущей операции сохраняется и используется повторно. На запоминании и воспроизведении уже известной информации базируются подходы решения задач с помощью динамического программирования. Для этого используются таблицы, как пример баз данных для хранения уже известных решений. Самая простейшая таблица может быть представлена как одномерный массив данных, в котором решения записаны последовательно, друг за другом.

Для решения задач методом динамического программирования используются так называемые рекуррентные соотношения, которые отражают еще один важный принцип. Рекуррентными соотношениями называются формулы, описывающие зависимость между решениями подзадач одного вида. Таким образом, решение каждой подзадачи основывается на результатах решения предыдущих. Важной особенностью этого принципа является то, что решение каждой подзадачи высчитывается один единственный раз, а итог запоминается и используется в последующих подзадачах как данное. Простейшим примером рекуррентных соотношений является последовательность Фибоначчи: каждое следующее число является суммой двух предыдущих. В виде рекуррентных сооотношений это можно записать как Fn = Fn–1 + Fn–2 (n > 1), где Fn – n-е число последовательности.

### *ОСНОВНЫЕ ПОДХОДЫ РЕШЕНИЯ ЗАДАЧ МЕТОДОМ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ*

Для решения задач методом динамического программирования существуют два основных подхода: восходящий и нисходящий подходы.

Восходящий подход (прямой ход) подразумевает решение глобальной задачи от самых простых, мелких подзадач к более сложным и крупным. Таким образом, получается, что процесс решения задачи происходит «снизу вверх» путем составления решения «верхних» задач из решений «нижних».

В свою очередь нисходящий подход (обратный ход) так же подразумевает разделение глобальной задачи на более простые подзадачи, которые решаются друг за другом. Однако, при данном подходе решение начинается не с вычисления самой «нижней» подзадачи, а с предположений об условно оптимальном решении самой «верхней». От этих вариантов решения строятся варианты решения для «нижних» подзадач, пока не будет достигнута самая «нижняя». После определения оптимальной стратегии решения задачи вычисления производятся в восходящем порядке, чтобы восстановить оптимальную стратегию.

При решении задач чаще используется нисходящий подход, так как он наиболее удобен для решения большинства задач. Однако, при решении некоторых задач данный подход может быть сложен для восприятия и понимания из-за трудностей определения, какие функции понадобятся при решении задачи. Восходящий подход может эффективно применяться для решения задач с небольшим количеством шагов. Для задач с большими объемами данных такой подход будет неэффективен ввиду необходимости выделять большой объем памяти и вызывать функции множество раз.

### *ПРИМЕРЫ ЗАДАЧ, РЕШАЕМЫХ С ПОМОЩЬЮ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ*

Метод динамического программирования применяется для решения множества задач в различных сферах. Многие из этих задач делятся на типы, к которым применяются свои алгоритмы решения. К таким типам относятся:

1. задача о наибольшей общей подпоследовательности;
2. задача поиска наибольшей увеличивающейся подпоследовательности;
3. задача о редакционном расстоянии (Расстояние Левенштейна);
4. задача о вычислении чисел Фибоначчи;
5. задача о порядке перемножения матриц;
6. задача о выборе траектории;
7. задача последовательного принятия решения;
8. задача об использовании рабочей силы;
9. задача управления запасами;
10. задача о ранце (задача о рюкзаке);
11. алгоритм Флойда-Уоршелла;
12. алгоритм Беллмана — Форда;
13. максимальное независимое множество вершин в дереве;

Рассмотрим некоторые из этих типов задач, чтобы наглядно показать, насколько широким может быть применение метода динамического программирования в различных областях.

Ранее уже упоминалась задача о вычислении чисел Фибоначчи. Суть задачи заключается в нахождении n-го члена последовательности. Наиболее простым и очевидным решением данной задачи является рекурсивный перебор всех значений последовательности до искомого. Здесь и далее примеры алгоритмов будут представлены на языке программирования Python:

def fibonacci(n):

    if n <= 1:

        return n

    else:

        return fibonacci(n-1)+fibonacci(n-2)

Однако, данный подход требует больших временных затрат, поэтому рациональнее использовать метод динамического программирования для решения данной задачи:

def fibonacci(n, memo):

    if n in memo:

        return memo[n]

    if n <= 1:

        memo[n] = n

    else:

        memo[n] = fibonacci(n-1, memo) + fibonacci(n-2, memo)

    return memo[n]

Каждое вычисленное значение записывается в базу данных (массив) и при необходимости извлекается из нее вместо произведения повторных вычислений.

Другая простейшая задача динамического программирования – задача о нахождении наибольшей общей подпоследовательности – набора символов, входящих в состав последовательностей в одном и том же порядке, но необязательно подряд. Суть задачи заключается в нахождении наибольшей общей подпоследовательности для двух данных строк. Данная задача может быть решена методом перебора. Однако, данный метод времязатратный, громоздкий и неэффективный. Вместо него можно применить метод динамического программирования:

def lcs(s1, s2):

    m, n = len(s1), len(s2)

    dp = [[0] \* (n+1) for \_ in range(m+1)]

    for i in range(1, m+1):

        for j in range(1, n+1):

            if s1[i-1] == s2[j-1]:

                dp[i][j] = dp[i-1][j-1] + 1

            else:

                dp[i][j] = max(dp[i-1][j], dp[i][j-1])

    return dp[m][n]

Задача кажется простой и примитивной, но в более усложненном виде применяется в биоинформатике для анализа генетических и белковых последовательностей.

Одна из наиболее известных задач динамического программирования – задача о рюкзаке. По условию, имеется рюкзак с ограниченной грузоподъемностью и набор предметов, о каждом из которых известны вес и цена. Для решения задачи необходимо сложить в рюкзак предметы, суммарная стоимость которых будет максимальной (или минимальной), а вес не будет превышать грузоподъемность рюкзака.

def knapsack(max\_weight, weights, values):

    n = len(values)

    # Инициализация таблицы ДП размером (n+1) x (max\_weight+1)

    dp = [[0 for \_ in range(max\_weight + 1)] for \_ in range(n + 1)]

    # Заполнение таблицы ДП

    for i in range(1, n + 1):

        for w in range(1, max\_weight + 1):

            if weights[i-1] <= w:

                # Если вес предмета меньше или равен текущей ёмкости рюкзака

                dp[i][w] = max(dp[i-1][w], dp[i-1][w-weights[i-1]] + values[i-1])

            else:

                # Если предмет слишком тяжёлый для текущей ёмкости рюкзака

                dp[i][w] = dp[i-1][w]

    return dp[n][max\_weight]

Данная задача находит широкое применение в логистике и экономике, так как позволяет определить наиболее выгодный «набор» чего-либо на основе всего двух параметров: веса (объема) и ценности.

Еще одна широко применяющаяся задача – задача о распределении ресурсов. Часто встречаются условия для данного типа задач, связанные с распределение инвестиций в различные компании. Соответственно, по условию инвестор выделяет некоторое количество средств, которые собирается разделить между несколькими компаниями. О каждой компании известно, какую прибыль она способна принести за единицу денежных средств. Для решения задачи необходимо составить план инвестиций на указанный в условии срок так, чтобы инвестор получил наибольшую прибыль.

При решении задачи о распределении ресурсов наиболее наглядно действует принцип оптимальности метода динамического программирования, так как стратегия распределения на каждом шаге необязательно будет наиболее выгодной для данного шага, но она может быть оптимальна для решения глобальной задачи. Таким образом, количество шагов в данной задаче будет соответствовать количеству лет инвестирования. Основное функциональное уравнение, используемое для решения задачи, будет выглядеть следующим образом:

, где s – количество имеющихся средств на данном шаге; i – номер шага; x – количество инвестированных в данное предприятие средств; pi(x) – прибыль от предприятия; Wi – общая прибыль.

Данный тип задач распространен при распределении различных типов ресурсов в производственной, строительной и экономической областях.

### *ПРЕИМУЩЕСТВА И НЕДОСТАТКИ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ*

Метод динамического программирования применяется в огромном количестве научных и технических областей: экономика, логистика, биоинформатика, разработка искусственного интеллекта, машинное обучение и многих других. Данный метод применяется в том числе в гуманитарных областях, например, в лингвистике. Динамическое программирование получило столь широкое распространение из-за ряда значительных преимуществ:

1. скорость;
2. универсальность;
3. точность;

Так, первым и наиболее востребованным в современном, быстро развивающемся мире преимуществом данного метода является скорость. Динамическое программирование позволяет решать задачи высокой сложности в короткие сроки, а по мере развития технологий его скорость только растет.

Для решения задач различной сложности методом динамического программирования применяются свои правила и алгоритмы, требующие наименьшего количества расчетов, в отличие от других методов. Данные правила не имеют исключений, благодаря чему могут применяться в огромном количестве областей и для решения задач различного объема и уровня сложности.

Помимо скорости и универсальности, рассматриваемый метод обеспечивает точность результата решения задач. Точность обеспечивается за счет охвата всех возможных вариантов решения поставленной задачи. Благодаря этому метод динамического программирования способен находить наиболее оптимальные стратегии решения задачи, исключая вероятность погрешности и неоднозначности результата.

Однако, метод динамического программирования имеет как преимущества, так и некоторые недостатки:

1. необходимость выделять некоторый объем памяти для работы алгоритма;

2. Сложность понимания и восприятия алгоритма;

Для работы алгоритмов, основанных на методе динамического программирования, необходимо выделять некоторый объем памяти, который может изменяться в зависимости от сложности задачи и объема данных. Наличие свободной памяти – необходимое условие для работы метода, так как в процессе решения задачи приходится создавать и хранить большое количество объемных таблиц с данными.

Недостатком динамического программирования, из-за которого метод может быть трудно использовать, является сложность алгоритма. Зачастую эти алгоритмы имеют достаточно компактный размер, но используют функции, которые могут быть трудны для восприятия и использования человеком, не имеющим особый склад ума и соответствующие знания в данной области.

Однако, несмотря на недостатки, преимуществ у метода динамического программирования больше, благодаря чему он не утратил своей популярности.

### *ВЫВОДЫ*

Метод динамического программирования – мощный и порой незаменимый инструмент для решения задач различной сложности и направленности. Он обладает всеми необходимыми в современном мире преимуществами: скоростью, точностью и универсальностью. Благодаря этому метод находит применение в огромном количестве научных и технических областей: лингвистика, логистика, экономика, математика, биоинформатика, строительство и многих других. Алгоритмы, основанные на динамическом программировании, используются для перевода и сравнения текстов, анализа ДНК и РНК, составления стратегий инвестиций и других задач, требующих быстрого нахождения наиболее оптимального решения. Разнообразие типов классических задач еще раз доказывает возможность многопрофильного применения данного метода.

Кроме того, благодаря развитию науки и техники метод динамического программирования имеет большой потенциал развития. За счет увеличения мощности и объема памяти современных вычислительных машин и развития языков программирования алгоритмы, основанные на принципах динамического программирования, могут становиться эффективнее, быстрее и точнее.

### *СПИСОК ЛИТЕРАТУРЫ*

1. Цгоева, Н. А. Решение задачи об инвестировании предприятий методом динамического программирования / Н. А. Цгоева // Тенденции развития науки и образования. – 2021. – № 71-2. – С. 146-149. – DOI 10.18411/lj-03-2021-72. – EDN EQWJOH.
2. Беллман Ричард. "Динамическое программирование" // Издательство Принстонского университета, 1957.
3. Кабаева, И. И. Задачи динамического программирования / И. И. Кабаева // European Research. – 2016. – № 11(22). – С. 10-11. – EDN XCQAPN.
4. Динамическое программирование: его преимущества и недостатки. URL: https://gb.ru/blog/dinamicheskoe-programmirovanie/ (дата обращения 08.07.2024).
5. Динамическое программирование. URL: <https://www.yourtodo.ru/posts/dinamicheskoe-programmirovanie/> (дата обращения: 08.07.2024)/