УДК 519.85

Студ. И.Н. Юшкевич, В.В. Наркевич
Науч. рук. ассистент А.А. Коренькова
(кафедра полиграфического оборудования и систем
обработки информации, БГТУ)

ИЗУЧЕНИЕ МЕТОДА ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ ДЛЯ РЕШЕНИЯ МАТЕМАТИЧЕСКИХ ЗАДАЧ

Введение. В жизни мы часто сталкиваемся с различными задачами. К большей части из них относятся задачи оптимизации цели (т. е. нахождение наиболее оптимального решения). Одним из важных факторов таких задач является ограниченность в ресурсе. Например, поход в магазин с ограниченным бюджетом. Для решения такого рода задач может использоваться метод динамического программирования.

Основная часть. Динамическое программирование – метод решения задачи путём её разбиения на несколько одинаковых подзадач, рекуррентно связанных между собой.

Динамическое программирование возникло в XX веке, когда американский математик Ричард Эрнест Беллман описал метод решения многошаговых задач путем разбиения их на более мелкие. Он определил принцип оптимальности, который заключается в том, что на каждом шаге следует стремиться не к изолированной оптимизации функции, а к предположению об оптимальности всех последующих шагов.

К требованиям к задаче для решения ее методом динамического программирования относятсянеобходимость разбиения задачи на несколько подзадач, сохранение решений подзадач для построения решения исходной задачи (мемоизация), наличие у подзадач наименьшей размерности известных решений.

Наиболее известной задачей, которую можно решить методом динамического программирования, является задача о вычислении чисел Фибоначчи.

Эта задача может быть решена точно с помощью динамического программирования. Путем разбиения её на более простые подзадачи, описанные формулой (1):

$$F_n = F_{n-2} + F_{n-1} \tag{1}$$

где F_n, $F_{n\text{-}2}$, $F_{n\text{-}1}$ – последовательность чисел Фибоначчи.

Можно вычислить n-ное число последовательности с помощью рекурсии, что позволит нам последовательно считать целые числа. Однако этот метод не совершенен. С одной стороны, можно найти точное число последовательности, в отличие от формулы Biné, с другой стороны, на этот способ необходимо затратить много времени, последовательно пересчитав числа Фибоначчи до искомого.

Рассмотрим задачу об одномерном рюкзаке. Она является классической задачей на оптимизацию, которая очень часто встречается на практике. Цель задачи – поместить в рюкзак предметы наибольшей ценности при соблюдении ограничения по весу.

Существует множествометодов решения данной задачи, которые делятся на точные и приближенные.

К точным относятся:

– метод полного перебора;

– метод ветвей и границ;

– метод динамического программирования.

К приближенным:

– решение с помощью жадного алгоритма (допускается, что на каждом этапе принимается локально оптимальное решение, предполагая, что конечное решение также окажется оптимальным);

– решение с помощью генетического алгоритма (поиск решения путем случайного подбора параметров и использования функции приспособленности, которая направляет «эволюцию» в сторону оптимального решения).

Будем решать задачу об одномерном рюкзаке методом динамического программирования.

Решение задачи методом динамического программирования сводится к разбиению ее на более простые подзадачи и использованию рекуррентного соотношения (2):

$$\begin{cases} m[0,w]=0, n=0 \\ m[n,w]=m[n-1,w], w_n > w \\ m[n,w]=\max\,(m[n-1,w], m[n-1,w-w_n]+v_n), w_n \le w \end{cases} \tag{2}$$

где $m[n,w]$ – максимальная ценность предметов, полученных из n имеющихся предметов; w – максимально допустимая масса рюкзака; v_i – ценность n-того предмета.

Рекуррентное соотношение объясняется так:

– при отсутствии предметов ($n = 0$) максимальная ценность равна нулю;

– при массе предмета большей, чем максимально допустимая масса рюкзака, этот предмет отбрасывается и находится максимальная ценность оставшихся $n – 1$ предметов;

– при массе предмета меньшей или равной максимально допустимой массе рюкзака находится максимум из максимальных ценностей предметов исходя из двух условий: предмет не положили в рюкзак ($m[n-1, w]$) или предмет положили в рюкзак, тем самым уменьшив и массу, которую выдержит рюкзак ($m[n-1, w-w_n]+v_n$).

В ходе моделирования задачи о рюкзаке на ЭВМ была выведена зависимость времени выполнения алгоритма от количества рассматриваемых предметов. Результаты представлены на рисунке 1.

Количество предметов	Время выполнения алгоритма
10	0
11	0,001
12	0,002
13	0,002
14	0,003
15	0,006
16	0,023
17	0,023
18	0,032
19	0,063
20	0,127
21	0,142
22	0,299
23	0,577
24	0,716
25	1,436
26	2,87
27	5,751
28	10,83
29	17,561
30	33,069

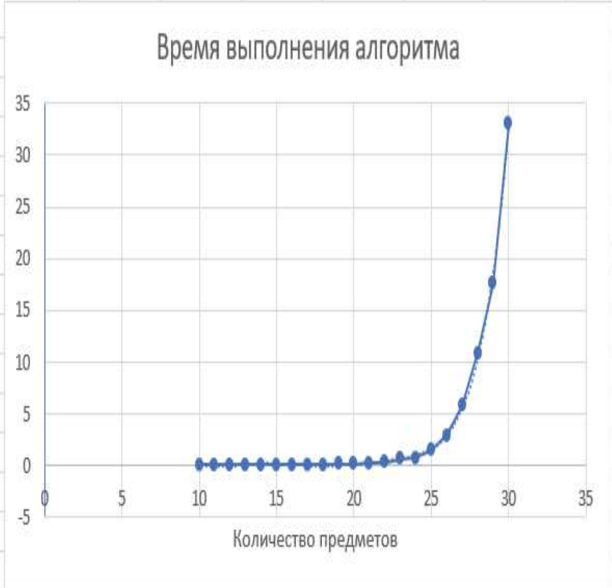

Рисунок 1 – Зависимость времени выполнения алгоритма от количества предметов

Из графика видно, что работа алгоритма стремится к экспоненциальной зависимости, что затрудняет вычисления при большом количестве элементов для рассмотрения.

Вывод: Динамическое программирование – точный и относительно быстрый инструмент для решения задач оптимизации и задач на подсчет количества вариантов решения, который уместно использовать с ЭВМ для быстрых вычислений результата, однако оно уступает в быстродействии приближенным методам вычисления при большом количестве подзадач.