Дисциплина

«РАЗРАБОТКА ЭФФЕКТИВНЫХ АЛГОРИТМОВ»

Лекция 18

МЕТОД ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

ЗАДАЧА ОПТИМАЛЬНОЙ ОДНОМЕРНОЙ УПАКОВКИ

Лектор: Михаил Васильевич Ульянов,

muljanov@mail.ru, **8 916 589 94 04**

1. История

Метод динамического программирования был предложен и обоснован Р. Беллманом в конце 1950-х, начале 1960-х годов. Первоначально метод создавался в целях существенного сокращения перебора для решения целого ряда дискретных задач экономического характера, формулируемых в терминах задач целочисленного программирования. Однако в своей книге Р. Беллман и Р. Дрейфус показали, что он применим к достаточно широкому кругу задач, в том числе к задачам вариационного исчисления, поиску нулей функций и целому ряду задач современной бизнес-информатики (задача о такси, задача о замене оборудования и т.д.)

Метод не является универсальным, и условия его применения требуют, чтобы целевой функционал представлял собой аддитивную функцию, т. е.

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} g_i(x_i).$$

2. Идея метода

Изложение идеи опирается на оригинальное изложение и терминологию Р. Беллмана в экономической интерпретации метода динамического программирования. В этом случае ограничения, задающие область поиска экстремума для целевого функционала, имеют вид

$$\sum_{i=1}^{n} x_i = C\,, \quad x_i \geq 0 \ \forall i = \overline{1,n},$$

и рассматриваются как ограничения на общий ресурс, который должен быть распределен по n инвестиционным процессам, приносящим некоторый доход. Значение дохода от процессов задается функциями $g_i(x_i),\ i = \overline{1,n}$ в условиях целочисленности и не отрицательности значений x_i. Это задача ЛЦП!

Для решения задачи максимизации целевого функционала $f(\boldsymbol{x}) \to max$, вместо рассмотрения одной задачи с фиксированным ресурсом и n процессами, рассматривается целое семейство задач, в которых число процессов n

последовательно принимает целые значения. Если исходная задача представляет собой статический процесс распределения ограниченного ресурса в n -мерном пространстве, то подход Р. Беллмана переводит ее в динамический процесс, требуя распределения ресурса последовательно по набору процессов, что собственно и нашло отражение в названии метода — динамическое программирование.

Максимум $f(\boldsymbol{x})$ в указанной области зависит от количества процессов n, и от ресурса C. Эта зависимость записывается явно путем задания последовательности функций $\{\ f_i(c)\ \}$, $i = \overline{1, n}$, $c \geq 0$, следующим образом

$$f_i(c) \ = \max_{x_i} f(\boldsymbol{x}), \quad x_i \geq 0, \quad \sum_{k=1}^{i} x_k = c.$$

При этом функция $f_i(c)$ выражает оптимальный доход, получаемый от распределения ресурса c по первым i процессам. В двух частных случаях значения этой функции вычисляются элементарно. В предположении, что доход каждого процесса от нулевого ресурса равен нулю:

$$g_i(0) = 0,\ \forall i = \overline{1, n}$$

очевидно, что $f_i(0) = 0,\ \forall i = \overline{1, n}$. Также очевидно, что при значениях $c \geq 0$ функция $f_1(c) = g_1(c)$, для возрастающих функций дохода. Достаточно просто находятся рекуррентные соотношения, связывающие $f_m(c)$ и $f_{m-1}(c)$ для произвольных значений m и c. Если x_m — количество ресурса, назначенное для процесса с номером m, то остающееся количество — $(c - x_m)$ должно быть оптимально распределено для получения максимального дохода от остающихся $m - 1$ процессов. Таким образом, при некотором значении x_m совокупный доход от распределения по m процессам составит

$$g_m(x_m)\ + f_{m-1}(c - x_m).$$

3. Основное функциональное уравнение Беллмана

Очевидно, что оптимальным будет такой выбор значения x_m, который максимизирует эту функцию, что и приводит к следующему рекуррентному соотношению, которое называется *основным функциональным уравнением метода динамического программирования*

$$\begin{cases} f_1(c) = g_1(c); \\ f_m(c) = \max\limits_{0 \le x_m \le c} [g_m(x_m) + f_{m-1}(c - x_m)], \ \forall m = \overline{2,\ n}, \quad c \ge 0. \end{cases}$$

Таким образом, задача оптимизации в многомерном пространстве сводится к последовательности задач одномерной оптимизации, что существенно сокращает трудоемкость получения решения.

Существенное ограничение по применимости метода — аддитивность целевой функции. В ряде случаев это требование может быть ослаблено до сперабельности (по переменным).

4. Содержательная постановка задачи одномерной упаковки

У нас есть рюкзак с прямоугольным дном и нерастяжимыми стенками определенной высоты. У нас есть так же несколько групп коробок, с таким же дном, как у рюкзака. В любой группе количество коробок достаточно для упаковки всего рюкзака, каждая коробка в группе одинакова по высоте и имеет определенную стоимость. Наша задача состоит в том, чтобы упаковать рюкзак, так, чтобы он закрывался, и сумма стоимостей упакованных коробок была бы наибольшей. Хотя содержательно мы имеем дело с объемом, но в реальности мы рассматриваем только высоту — в этом смысле задача является *одномерной*. Интуитивное решение — выбрать коробки из группы, обладающей максимальной удельной (на единицу высоты) стоимостью.

Но мы не можем разрезать коробки по высоте — задача является целочисленной. Например, из двух групп коробок с высотами 5 и 7 и стоимостями 10 и 18, в рюкзак высотой 10 лучше положить две коробки из первой группы, чем одну из второй, хотя удельная стоимость коробок второй группы больше, чем в первой. Другой подход — рассмотреть все возможные варианты упаковки рюкзака и выбрать наилучший, что ведет к экспоненциальной сложности.

Задача оптимальной по стоимости одномерной упаковки, имеет разнообразные практические применения, для которых сегодня актуальным является получение именно точных решений. К такой постановке сводится задача одномерного раскроя материала, формирования оптимального пакета акций на фиксированную сумму. Эта задача относится к группе задач целочисленного программирования, которые формулируются как задачи поиска экстремума функции нескольких переменных, аргументами которой являются координаты точек ограниченного подмножества целочисленного пространства.

5. Математическая постановка задачи одномерной упаковки

Пусть задано множество типов грузов

$$Y = \{\ y_i\ \},\ y_i = \{\ v_i, c_i\ \},\ i = \overline{1,n},$$

где каждый элемент y_i, соотнесенный с типом груза, обладает целочисленным линейным размером – v_i, или «объемом» и ценовой характеристикой — c_i, которая отражает предпочтения для загрузки объектов данного типа. Так же целочисленным значением задан основной объем упаковки V. Элементы y_i называются типами грузов. Для описания количества загружаемых в объем V грузов y_i введем в рассмотрение следующий характеристический вектор (кортеж):

$$\boldsymbol{x} = (x_i), \quad i = \overline{1,n}, \boldsymbol{x} \in E_z^n,\ x_i \geq 0$$

т. е. x_i — неотрицательное целое, Значение компонента вектора $x_i = k$ соответствует загрузке k элементов типа y_i в объем V. Мы получаем следующую постановку задачи упаковки как задачи линейного целочисленного программирования — максимизировать линейный функционал:

$$P_n(\boldsymbol{x}) \ = \ \sum_{i=1}^{n} g_i(x_i) \ = \sum_{i=1}^{n} x_i \cdot c_i \ \rightarrow \ max,$$

$$\sum_{i=1}^{n} x_i \cdot v_i \quad \leq \quad V \ , \boldsymbol{x} \in E_z^n.$$

Функциональное уравнение Беллмана для задачи упаковки. Будем считать, что в рассматриваемой задаче распределяемым ресурсом является объем упаковки V, а функции дохода линейны — $g_i(x_i) = c_i \cdot x_i$. Наша задача — максимизировать доход, заданный линейным функционалом $P_n(\boldsymbol{x})$ путем распределения ограниченного ресурса объема упаковки между грузами указанных типов. Основное функциональное уравнение Беллмана имеет вид

$$\begin{cases} f_0(v) = 0; \\ f_m(v) \ = \ \max\limits_{x_m} \{x_m \cdot c_m \ + \ f_{m-1}(v - x_m \cdot v_m)\}, \ m = \overline{1, n}, \ x_m = 0,1, \dots, \left\lfloor \dfrac{v}{v_m} \right\rfloor \ . \end{cases}$$

Таким образом, метод предполагает последовательное решение одномерных задач целочисленной оптимизации с использованием информации об оптимальной упаковке объема v предыдущими типами грузов. Решением поставленной задачи является значение $f_n(V)$. Поскольку значения $f_1(v)$ могут быть элементарно вычислены, то в дальнейшем мы будем рассматривать следующее основное функциональное уравнение для задачи одномерной оптимальной упаковки, записанное в виде рекуррентного соотношения, определяющего рекурсивно заданную функцию$f_m(v)$

$$\begin{cases} f_1(v) = \left\lfloor \dfrac{v}{v_1} \right\rfloor \cdot c_1 \ , m = 1; \\ f_m(v) \ = \ \max\limits_{x_m} \{x_m \cdot c_m \ + \ f_{m-1}(v - x_m \cdot v_m)\}, \ m \geq 2, \ x_m = 0,1,\dots, \left\lfloor \dfrac{v}{v_m} \right\rfloor . \end{cases}$$

6. Пример порожденного дерева рекурсии

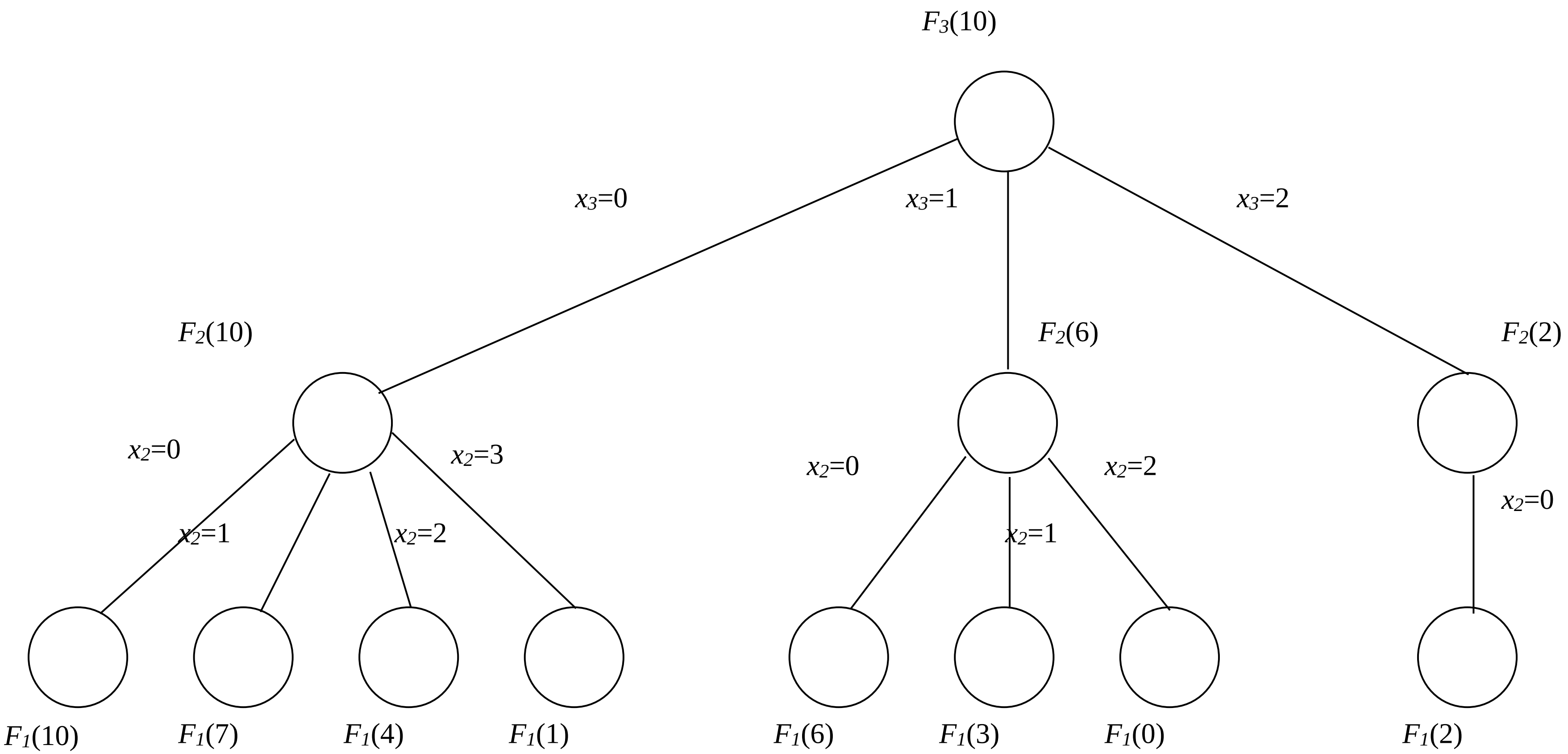
Мы решаем задачу с тремя типами грузов, при общем объеме упаковки $V = 10$. Информация об объемах v_i и стоимостях c_i для трех типов грузов:

i	v_i	c_i
1	2	3
2	3	5
3	4	7

Какая упаковка является оптимальной по стоимости?

Решением поставленной задачи будет значение функции Беллмана $f_3(10)$, при этом алгоритм порождает следующее дерево рекурсии:

Дерево рекурсии, порождаемое алгоритмом ДП для одномерной упаковки



Параметризация задачи. Исследуемый рекурсивный алгоритм упаковки является количественно параметрическим. Напомним, что в этом случае число элементарных операций, задаваемых алгоритмом, зависит не только от количества данных на входе, но и от их значений.

Очевидно, что оценка вычислительной сложности будет зависеть как от значения n, так и от значений параметров $V, v_1, \cdots, v_n$, учет которых существенно затрудняет анализ. С целью упрощения анализа рекурсивного алгоритма мы вводим параметр

$$k = \frac{V}{\bar{v}}, \bar{v} = \frac{1}{n} \cdot \sum_{i=1}^{n} v_i,$$

характеризующий, сколько грузов среднего объема размещается в объеме упаковки V. Очевидно, что в реальности количество любых грузов, размещенных в объеме V, является целым числом, но для оценки вычислительной сложности в среднем, необходимо учитывать, что параметр k может быть и действительным (вещественным) числом.

Подробности такого анализа и окончательные формулы — в книге «Теория рекурсии для программистов».

5. Литература

1. Беллман Р., Дрейфус Р. Прикладные задачи динамического программирования: Пер. с англ. — М.: Наука, 1965, — 457 с
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-ое издание: Пер. с англ. — М.: Издательский дом «Вильямс», 2005. — 1296 с.
3. Хаггарти Р. Дискретная математика для программистов. — М.: Техносфера, 2005. — 400с.