

Метод динамического программирования

Для задач, общее решение
которых может быть получено
как результат решений
некоторого ряда подзадач
($d_1, d_2, \dots, d_p, \dots, d_q$),

применяется метод

**динамического
программирования**

- Метод динамического программирования (метод Гамильтона-Якоби-Беллмана) ориентирован на поиск оптимального управления широкого класса систем, в том числе для решения задач планирования, распределение ресурсов, снабжения, разрешения игровых ситуаций, построение алгоритмов решения задач и т.д.

- В основе метода динамического программирования лежит специфический принцип оптимальности, определяющий стратегию поиска оптимального управления. Принцип формулируется следующим образом: оптимальное управление не зависит от предыстории процесса изменения состояния системы, а определяется лишь ее состоянием в рассматриваемый момент времени.

- Каждое решение d_p должно являться **локальным решением**, которое оптимизировало бы некоторый **глобальный критерий качества**, например, стоимость путешествия, длину пройденного пути, массу перевезённого груза, место, занимаемое файлом на диске, и т.п. Для того, чтобы данный метод был применим, необходимо, чтобы решаемая задача отвечала **принципу оптимальности**:

- если $(d_1, d_2, \dots, d_{p+1}, \dots, d_q)$ – оптимальный ряд принимаемых решений, то и ряды (d_1, d_2, \dots, d_p) и (d_{p+1}, \dots, d_q) должны быть оптимальными.

- Например, если кратчайшая дорога от Нижнего Новгорода до Москвы проходит через Владимир, то и оба участка этой дороги – Нижний Новгород - Владимир и Владимир - Москва – также должны быть самыми короткими. Следовательно, задачи нахождения кратчайшего пути удовлетворяют принципу оптимальности.

- В QBasic метод динамического программирования может быть реализован с помощью массивов, элементы которых вычисляются при помощи определённых рекуррентных соотношений. В общем случае, рекуррентные соотношения бывают следующих двух типов:

- Каждое принимаемое решение d_p зависит от решений d_{p+1}, \dots, d_q . Будем говорить, что в этом случае применяется **метод «вперёд»**. В этом методе решения будут приниматься в порядке d_q, d_{q-1}, \dots, d_1 .
- Каждое принимаемое решение d_p зависит от решений d_1, \dots, d_{p-1} . Будем говорить, что в этом случае применяется метод «назад». В этом методе решения будут приниматься в порядке d_1, d_2, \dots, d_q .

Очевидно, что для каждой задачи программист должен проверять в первую очередь соблюдение принципа оптимальности и, в случае положительного ответа, вывести соответствующие рекуррентные соотношения. В противном случае, рассматриваемая задача не может быть решена с помощью метода динамического программирования.