

Динамическое программирование

Динамическое программирование (ДП) – метод оптимизации, приспособленный к операциям, в которых процесс принятия решения может быть разбит на этапы (шаги). Такие операции называются многошаговыми. Начало развития ДП относится к 50-м годам XX в. Оно связано с именем американского математика Р. Беллмана.

Если модели линейного программирования можно использовать в экономике для принятия крупномасштабных плановых решений в сложных ситуациях, то модели ДП применяются при решении задач значительно меньшего масштаба, например, при разработке правил управления запасами, устанавливающими момент пополнения запасов и размер пополняющего заказа; при разработке принципов календарного планирования производства и выравнивания занятости в условиях колеблющегося спроса на продукцию; при распределении дефицитных капитальных вложений между возможными новыми направлениями их использования при составлении календарных планов текущего и капитального ремонта сложного оборудования и его замены; при разработке долгосрочных правил замены выбывающих из эксплуатации основных фондов и т.п.

В реально функционирующих больших экономических системах еженедельно требуется принимать микроэкономические решения. Модели ДП ценны тем, что позволяют на основе стандартного подхода с использованием при минимальном вмешательстве человека принимать такие решения. И если каждое взятое в отдельности такое решение малосущественно, то в совокупности эти решения могут оказать большое влияние на прибыль.

Приведем общую постановку задачи ДП. Рассматривается управляемый процесс, например, экономический процесс распределения средств между предприятиями, ресурсов в течение ряда лет, замены оборудования, пополнения запасов и т.п. В результате управления система (объект управления) S переводится из начального состояния s_0 в состояние S .

Предположим, что управление можно разбить на p шагов, т.е. решение принимается последовательно на каждом шаге, а управление, переводящее систему S из начального состояния в конечное, представляет собой совокупность p пошаговых управлений.

Обозначим через X_k управление на k -м шаге ($k=1, 2, \dots, p$). Переменные X_k удовлетворяют некоторым ограничениям и в этом смысле называются допустимыми (X_k может быть числом, точкой в p -мерном пространстве, качественным признаком).

Пусть $X = (X_1, X_1, \dots, X_n)$ – управление, переводящее систему S из состояния s_0 в состояние s_n . Обозначим через s_k состояние системы после k -го шага управления. Получаем последовательность состояний $s_0, s_1, \dots, s_{k-1}, s_k, \dots, s_{n-1}, s_n = S_m$, которую изобразим кружками (рис. 1.1).

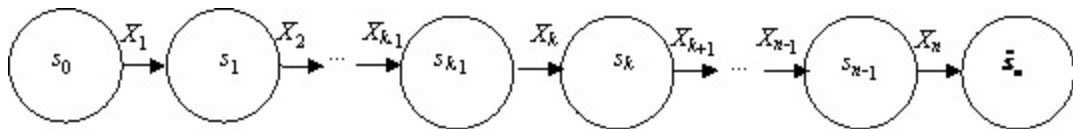


Рис.1.1.

Показатель эффективности рассматриваемой управляемой операции – целевая функция – зависит от начального состояния и управления:

$$Z = F(s_0, X) \quad (1.1)$$

Сделаем несколько предположений.

1. Состояние s_k системы в конце k -го шага зависит только предшествующего состояния s_{k-1} и управления на k -м шаге X_k . (и не зависит от предшествующих состояний и управлений). Это требование называется «отсутствием последствия». Сформулированное положение записывается в виде уравнений:

$$s_k = \varphi_k(s_{k-1}, X_k), k=1, 2, \dots, n, \quad (1.2)$$

которые называются уравнениями состояний.

2. Целевая функция (1) является аддитивной от показателя

эффективности каждого шага. Обозначим показатель эффективности k -го шага через

$$Z_k = f_k(s_{k-1}, X_k), k=1, 2, \dots, n, \quad (1.3)$$

Тогда

$$Z = \sum_{k=1}^n f_k(s_{k-1}, X_k). \quad (1.4)$$

Задача пошаговой оптимизации (задача ДП) формулируется так: определить такое допустимое управление X , переводящее систему S из состояния s_0 в состояние \vec{s} , при котором целевая функция (1.4) принимает наибольшее(наименьшее) значение.

Выделим особенности модели ДП:

1. Задача оптимизации интерпретируется как n -шаговый процесс управления
2. Целевая функция равна сумме целевых функций каждого шага.
3. Выбор управления на k -м шаге зависит только от состояния системы к этому шагу, не влияет на предшествующие шаги (нет обратной связи).
4. Состояние s_k после k -го шага управления зависит только от предшествующего состояния s_{k-1} и управления X_k (отсутствие последействия).
5. На каждом шаге управление X_k зависит от конечного числа управляющих переменных, а состояние s_k – от конечного числа параметров.

Смысл замечаний станет ясным из рассмотренных ниже примеров. Существуют различные способы решения подобных задач, применяемые в зависимости от вида функций, ограничений, размерности и т. п. Рассмотрим вычислительную схему ДП, которая окажется безразличной к способам задания функций и ограничений. Вычислительная схема связана с принципом оптимальности и использует рекуррентные соотношения.

Принцип оптимальности впервые был сформулирован Р. Беллманом в 1953г. Каково бы ни было состояние S системы в результате какого-либо числа шагов, на ближайшем шаге нужно выбирать управление так, чтобы оно в совокупности с оптимальным управлением на всех последующих шагах

приводило к оптимальному выигрышу на всех оставшихся шагах, включая данный.

Беллманом были сформулированы и условия, при которых принцип верен. Основное требование – процесс управления должен быть без обратной связи, т.е. управление на данном шаге не должно оказывать влияния на предшествующие шаги.

Принцип оптимальности утверждает, что для любого процесса без обратной связи оптимальное управление таково, что оно является оптимальным для любого подпроцесса по отношению к исходному состоянию этого подпроцесса. Поэтому решение на каждом шаге оказывается наилучшим с точки зрения управления в целом. Если изобразить геометрически оптимальную траекторию в виде ломаной линии, то любая часть этой ломаной будет являться оптимальной траекторией относительно начала и конца.

Уравнения Беллмана. Вместо исходной задачи ДП с фиксированным числом шагов n и начальным состоянием s_0 рассмотрим последовательность задач, полагая последовательно $n=1, 2, \dots$ при различных s – одношаговую, двухшаговую и т.д., – используя принцип оптимальности.

Рассмотрим пример применения метода динамического программирования к задаче, которую можно интерпретировать как задачу распределения ресурсов между технологическими процессами. Имеется n технологических процессов, по которым надо распределить однородный ресурс (например деньги, или рабочую силу, или энергию) так, чтобы максимизировать общий доход. Пусть использование в i -м технологическом процессе ресурса в количестве u_i приносит доход $f_i(u_i)$, а общее количество ресурса равняется a . Тогда задача состоит в максимизации функции:

$$f(\bar{u}) = \sum_{i=1}^n f_i(u_i),$$

$$\sum_{i=1}^n u_i = a$$

при ограничении

Рассмотрим наряду с этой задачей класс аналогичных задач, в которых может меняться число процессов и общее количество ресурсов. Обозначим оптимальное значение в задаче с k процессами и количеством ресурсов b через $w_k(b)$, тогда в исходной задаче оптимальное значение равно $w_n(a)$. Для поставленных задач справедлив принцип оптимальности, который здесь можно сформулировать следующим образом: если для всех процессов в задаче ресурс распределен оптимально, то для части процессов доля ресурса, приходящаяся на них при таком распределении, также должна быть распределена оптимально. Если этот принцип применить к задаче с k процессами и количеством ресурсов b и предположить, что в k -й процесс вкладывается количество ресурса u_k , то получаем формулу:

$$w_k(b) = \max [f_k(u_k) + w_{k-1}(b - u_k)],$$

$$0 \leq u_k \leq b$$

Эта формула справедлива при любых k и b , поэтому для исходной задачи имеем цепочку соотношений:

$$w_n(a) = \max [f_n(u_n) + w_{n-1}(b - u_n)],$$

$$0 \leq u_n \leq a$$

$$w_{n-1}(a - u_{n-1}) = \max [f_k(u_k) + w_{k-1}(b - u_k)],$$

$$0 \leq u_{n-1} \leq a - u_n$$

и т.д. Для первого процесса, когда ресурсы по остальным процессам уже распределены, получаем:

$$w_1(a - \sum_{i=2}^n u_i) = \max f_1(u_1),$$

$$0 \leq u_1 \leq a - \sum_{i=2}^n u_i$$

С помощью таких рекуррентных соотношений можно решить задачу

оптимального распределения ресурсов. Рассмотрим конкретный пример с

$f_i(u_i) = c_i \sqrt{u_i}$. Итак, требуется максимизировать: $f(\bar{u}) = \sum_{i=1}^n c_i \sqrt{u_i}$ при

ограничении $\sum_{i=1}^n u_i = a$. Имеем $w_1(a - \sum_{i=2}^n u_i) = \max c_1 \sqrt{u_1} = c_1 \sqrt{a - \sum_{i=2}^n u_i}$,

$0 \leq u_1 \leq a - \sum_{i=2}^n u_i$, а условно оптимальное использование ресурса в

первом процессе есть $u_1^i = a - \sum_{i=2}^n u_i$.

Далее, $w_2(a - \sum_{i=3}^n u_i) = \max [c_2 \sqrt{u_2} + c_1 \sqrt{a - \sum_{i=2}^n u_i}]$, $0 \leq u_2 \leq a - \sum_{i=3}^n u_i$

Дифференцируя выражение в квадратных скобках по u_2 и приравнявая производную к нулю, получаем соотношение

$$\frac{1}{2} c_2 u_2^{-\frac{1}{2}} - \frac{1}{2} c_1 \left(a - \sum_{i=2}^n u_i \right)^{-\frac{1}{2}} = 0,$$

откуда

$$u_2^i = \frac{c_2^2}{c_1^2 + c_2^2} \left(a - \sum_{i=3}^n u_i \right),$$

$$w_2 \left(a - \sum_{i=3}^n u_i \right) = \sqrt{(c_1^2 + c_2^2) \cdot \left(a - \sum_{i=3}^n u_i \right)}.$$

На следующем шаге

$$w_3 \left(a - \sum_{i=4}^n u_i \right) = \max [c_3 \sqrt{u_3} + \sqrt{(c_1^2 + c_2^2) \cdot \left(a - \sum_{i=3}^n u_i \right)}],$$

$$0 \leq u_3 \leq a - \sum_{i=4}^n u_i.$$

Решая эту задачу максимизации, как и на предыдущем шаге, получаем

$$u_3^i = \frac{c_3^2}{c_1^2 + c_2^2 + c_3^2} \left(a - \sum_{i=4}^n u_i \right),$$

$$w_3 \left(a - \sum_{i=4}^n u_i \right) = \sqrt{(c_1^2 + c_2^2 + c_3^2) \cdot \left(a - \sum_{i=4}^n u_i \right)}.$$

Продолжая вычисления по процедуре, получаем общие выражения для u_k^i и w_k :

$$u_k^i = \frac{c_k^2}{\sum_{i=1}^k c_i^2} \left(a - \sum_{i=k+1}^n u_i \right),$$

$$w_k \left(a - \sum_{i=k+1}^n u_i \right) = \sqrt{\sum_{i=1}^k c_i^2 \left(a - \sum_{i=k+1}^n u_i \right)}.$$

Так как для последнего технологического процесса должно выполняться ограничение $0 \leq u_n \leq a$, то на n -м шаге находим просто

$$w_n(a) = \sqrt{a \sum_{i=1}^n c_i^2}$$

оптимальное значение задачи

и оптимальное использование ресурса в n -м процессе

$$u_n^0 = \frac{c_n^2 a}{\sum_{i=1}^n c_i^2}.$$

Подставляя u_n^0 в выражение для u_{n-1}^i , затем в u_{n-2}^i и т.д., получаем решение задачи

$$u_k^0 = \frac{c_k^2 a}{\sum_{i=1}^n c_i^2}.$$