1 Метод динамического программирования. Задача управления (общая схема многошагового процесса). Условия, которым должна удовлетворять задача, решаемая методом ДП. Принцип оптимальности Беллмана. Уравнения Беллмана.

Определение 1. *Динамическое программирования – это раздел математического программирования, посвященный исследованию многошаговых задач принятия оптимальных решений. При этом многошаговость задачи:*

- *либо отражает реальное протекание принятия решений во времени;*
- *либо вводится в эту задачу искусственно за счет расчленения процесса принятия однократного решения на отдельные этапы, шаги.*

Цель такого представления состоит в сведении исходной задачи высокой размерности к решению на каждом шаге задачи небольшой размерности (часто одномерной).

Методы ДП могут применяться к разнообразным задачам планирования и управления, например, управление запасами, замены и ремонта оборудования и др.

1.1 Задача управления(общая схема многошагового процесса)

Пусть имеется некоторая система S. В результате управления эта система переводится из некоторого состояния S_0 в конечное состояние S_n. Предположим, что управление можно разбить на n шагов, т.е. решение принимается последовательно на любом шаге, а управление, переводящие систему S из состояния S_0 в S_n, представляет собой совокупность n пошаговых управлений.

$$S_0 \xrightarrow{X_1} S_1 \xrightarrow{X_2} \cdots \xrightarrow{X_k} S_1 \xrightarrow{X_{k+1}} \cdots \xrightarrow{X_n} S_n$$

$X = (X_1, \dots, X_n)$ – управление (политика), переводящее систему S из S_0 в S_n, где X_k – управление на к-ом шаге.

$X = (X_1, \dots, X_n)$ — управление (политика), переводящее систему S из S_0 в S_n, где X_k — управление на k-ом шаге.

Переменные X_k удовлетворяют некоторым ограничениям: как исходным, так и ограничениям, возникающим за счет ранее сделанных выборов $X_1, \dots, X_{k-1}$. Каждое решение приносит определенный выигрыш (доход), при этом качество каждого из управлений X характеризуется соответствующим значением функции $W = F(S_0, X)$ — показатель эффективности.

1.2 Условия, которым должна удовлетворять задача, решаемая методом ДП

1. Состояние S_k системы после k-ого шага зависит только от предшествующего состояния S_{k-1} и управления на k-ом шаге X_k и не зависит от предшествующих состояний и управлений. Это требование называется ”отсутствием последействия”.

   $$S_k = \varphi_k(S_{k-1}, X_k),$$

   уравнение состояний $k = 1, n$ (1)

2. Целевая функция $W = F(S_0, X)$ является аддитивной от показателя эффективности каждого шага:

   $$W_k = f_k(S_{k-1}, X_k), \quad k = 1, n$$

   $$\Rightarrow W = \sum_{k=1}^{n} W_k = \sum_{k=1}^{n} f_k(S_{k-1}, X_k) \quad (2)$$

Определение 2. *Последовательность $X = (X_1, \dots, X_n)$ допустимых управлений X_k на отдельных шагах называется политикой.*

Задача управления состоит в поиске такой оптимальной стратегии управления (оптимальной политики) $X^* = (X_1^*, \dots, X_n^*)$, в результате реализации которой система S переходит из начального состояния S_0 в конечное S_n и при этом функция (2) принимает оптимальное значение (например, max), т.е. $W \to \max$. Сформулированная задача является многоэтапной. В целом ряде задач многоэтапность не следует из их условий. Однако в целях нахождения решения методом ДП их следует рассматривать как многоэтапные.

Теорема 1 (Принцип оптимальности Беллмана)**.** *Оптимальная политика обладает тем свойством, что каковы бы ни были решение, принятое на первом* *шаге и состояние системы после первого шага, последующие решения должны составлять оптимальное относительно этого состояния поведение. Любое оптимальное решение может быть образовано только оптимальными частными решениями.*

Для применения принципа оптимальности в конкртеных задачах пользуются приемом, часто называемым погружением. Он состоит в том, что вместо решения исходной задачи с данным начальным состоянием S_0 и данным числом шагов n решается целое семейство задач с произвольным начальным состоянием и с произвольным числом шагов.

Формализация принципа оптимальности приводит к некоторым функциональным уравнениям, решение которых и составляет основу вычислительных схем.

Во многих случаях функциональные уравнения ДП представляют собой систему рекуррентных соотношений. Их называют уравнениями Беллмана.

Вместо исходной задачи с фиксированным числом шагов n и начальным состоянием S_0 рассматривается последовательность задач. Полагая последовательно $n = 1, 2, \ldots$, при различных S_i получаем одношаговую, двухшаговую и т.д. задачи.

На каждом шаге любого состояния системы S_{k-1} решение X_k нужно выбирать "с оглядкой т.к. этот выбор влияет на последующее состояние системы S_k и дальнейший процесс управления, зависящий от S_k.

Но! На последнем шаге можно для любого состояния S_{n-1} планировать локально-оптимально, исходя из соображений этого шага.

Уравнения Беллмана

Согласно принципу оптимальности X_n нужно выбирать так, чтобы для любого состояния S_{n-1} получить максимум целевой функции на этом шаге. Обозначим $W_n^*(S_{n-1})$ – максимум целевой функции – показатель эффективности n-го шага при условии, что к началу последного шага система S была в произвольном состоянии S_{n-1}, а на последнем шаге управление было оптимальным. $W_n^*(S_{n-1})$ называется условным максимум целевой функции на n-м шаге.

$$W_n^*(S_{n-1}) = \max_{\{X_n\}} f_n(S_{n-1}, X_n)$$

Максимум берется по всем допустимым управлениям X_n.

Решение X_n, при котором достигается $W_n^*(S_{n-1})$, также зависит от S_{n-1} и называется условным оптимальным управлением на n-ом шаге. Обозначим его $X_n^*(S_{n-1})$.

Решив одномерную задачу локальной оптимизации для всех возможных состояний S_{n-1}, получим две последовательности значений:
$W_n^*(S_{n-1})$ и $X_n^*(S_{n-1})$.

Рассмотрим теперь двухшаговую задачу: присоединим к n-ому шагу $(n-1)$-ый.

$$\begin{tikzcd} S_{n-2} \arrow[r, "X_{n-1}"'] \arrow[r, bend left, "f_{n-1}(S_{n-2}, X_{n-1})"] & S_{n-1} \arrow[r, "X_n^*(S_{n-1})"'] \arrow[r, bend left, "W_N^*(S_{n-1})"] & S_n \end{tikzcd}$$

Для любого состояния S_{n-2} и любых произвольных управлений X_{n-1} и оптимальном управлении на n-ом шаге значение целевой функции на двух последних шагах равно:

$$f_{n-1}(S_{n-2}, X_{n-1}) + W_n^*(S_{n-1}) \to \max \quad (4)$$

Тогда по принципу оптимальности для любого S_{n-2} решение нужно выбирать так, чтобы оно вместе с оптимальным управлением на последнем n-ом шаге приводило бы к максимуму целевой функции на двух последних шагах. $\Rightarrow$, ищем максимум (4) по всем допустимым X_{n-1}.

$$W_{n-1}^*(S_{n-2}) = \max_{\{X_{n-1}\}} \{f_{n-1}(S_{n-2}, X_{n-1}) + W_N^*(S_{n-1})\} \quad (5)$$

Максимум этой сумы зависит от S_{n-2} и равен $W_{n-1}^*(S_{n-2})$, называется условным максимумом целевой функции при оптимальном управлении на двух последних шагах. Соответствующее $X_{n-1}^*(S_{n-2})$ называется условным оптимальным управлением на $(n-1)$-м шаге.

$W_{n-1}^*(S_{n-2})$ зависит только от S_{n-2} и X_{n-1}, т.к. S_{n-1} можно найти из уравнения состояния (1):

$$S_k = \varphi_k(S_{k-1}, X_k) \quad \text{при } k = n-1:$$

$$S_{n-1} = \varphi_{n-1}(S_{n-2}, X_{n-1}), \quad \text{подставим вместо } S_{n-1} \text{ в функцию } W_n^*(S_{n-1}).$$

В результате максимизируя только по одной переменной X_{n-1}, вновь получаем две последовательности:

$$W_{n-1}^*(S_{n-2}) \quad \text{и} \quad X_{n-1}^*(S_{n-2}).$$

Далее рассматривается трехшаговая задача: к двум последним шагам присоединяется $(n-2)$-ой шаг и т.д.

Обозначим $W_k^*(S_{k-1})$ — условный максимум целевой функции, получаемый при оптимальном управлении на $n-k+1$ шагах, начиная с k-ого до конца, при условии, что к началу k-ого шага система находилась в состоянии S_{k-1}:

$$W_k^*(S_{k-1}) = \max_{\{(X_k,\ldots,X_n)\}} \sum_{i=k}^{n} f_i(S_{i-1}, X_i) \Rightarrow W_{k+1}^*(S_k) = \max_{\{(X_{k+1},\ldots,X_n)\}} \sum_{i=k+1}^{n} f_i(S_{i-1}, X$$

Целевая функция на $(n-k)$ последних шагах при любом X_k и оптимальном управлении на последних $(n-k)$ шагах равна:

$$f_k(S_{k-1}, X_k) + W_{k+1}^*(S_k) \to \max$$

По принципу оптимальности X_k выбирается из условия максимума, т.е.

$$W_k^*(S_{k-1}) = \max_{\{X_k\}} \{f_k(S_{k-1}, X_k) + W_{k+1}^*(S_k)\} \qquad (6)$$

Управление X_k на k-ом шаге, при котором достигается максимум, обозначим $X_k^*(S_{k-1})$ — условное оптимальное управление на k-ом шаге.

Вместо S_k из уравнения состояния подставим $S_k = \varphi_k(S_{k-1}, X_k)$. Уравнение (6) называется уравнением Беллмана.

Это рекуррентное соотношение, позволяющее найти предыдущее значение, зная последующее.

$$W_n^*(S_{n-1}) = \max_{\{X_n\}} f_n(S_{n-1}, X_n)$$

$$W_k^*(S_{k-1}) = \max_{\{X_k\}} \{f_k(S_{k-1}, X_k) + W_{k+1}^*(S_k)\}, \quad k = n-1, n-2, \ldots, 1$$

Этот процесс называется условной оптимизацией.

Мы описали способ решения задачи динамического программирования, начинающийся с последнего шага. Можно также поменять местами n-ый и 1-ый шаги.

1.3 Нахождение решения

В результате условной оптимизации получаются две последовательности:

1. $W_n^*(S_{n-1}), W_{n-1}^*(S_{n-2}), \ldots, W_2^*(S_1), W_1^*(S_0)$ — условные максимумы целевой функции на последнем, на двух последних, ..., на n шагах.

2. $X_n^*(S_{n-1}), X_{n-1}^*(S_{n-2}), \ldots, X_2^*(S_1), X_1^*(S_0)$
– условные оптимальные управления на n-ом, $(n-1)$-ом, ..., 1-ом шагах.

Используя эти последовательности, можно найти решение задачи при данных n и S_0 следующим образом: по определению $W_1^*(S_0)$ — условный максимум целевой функции за n шагов при условии, что к началу 1-ого шага система была в состоянии S_0, т.е. $W_{\max} = W_1^*(S_0)$.

Далее следует использовать последовательность условных оптимальных управлений и уравнения состояний.

При фиксированном S_0 получаем $X_1^* = X_1^*(S_0)$. Далее из уравнения состояний находим $S_1^* = \varphi_1(S_0, X_1^*)$ и подставляем это выражение в последовательность условных оптимальных управлений:

$$X_2^* = X_2^*(S_1^*) \text{ и т.д. по следующей цепочке:}$$

$$X_1^* = X_1^*(S_0) \to S_1^* = \varphi_1(S_0, X_1^*) \Rightarrow X_2^* = X_2^*(S_1^*) \to S_2^* = \varphi_2(S_1^*, X_2^*) \Rightarrow$$
$$X_3^* = X_3^*(S_2^*) \to \cdots \to S_{n-1}^* = \varphi_{n-1}(S_{n-2}^*, X_{n-1}^*) \Rightarrow X_n^* = X_n^*(S_{n-1}^*)$$

$\Rightarrow$ получаем оптимальное решение задачи динамического программирования:

$$X^* = (X_1^*, X_2^*, \ldots, X_n^*)$$