ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

Ф. Л. ЧЕРНОУСЬКО

Московский физико-технический институт, Долгопрудный Московской обл.

DYNAMIC PROGRAMMING

F. L. CHERNOUS'KO

A method of dynamic programming is a well-known and powerful mathematical method of the modern control theory. The method is described for multi-step (discrete time) control processes. Computational aspects and generalizations of the method are discussed.

Метод динамического программирования – широко известный и мощный математический метод современной теории управления. В статье дается описание метода динамического программирования для многошаговых процессов управления (процессов с дискретным временем). Обсуждаются вычислительные аспекты и обобщения метода.

ВВЕДЕНИЕ

Метод динамического программирования – один из наиболее мощных и широко известных математических методов современной теории управления, был предложен в конце 50-х годов американским математиком Р. Беллманом и быстро получил широкое распространение, чему способствовали ярко и доходчиво написанные книги самого Беллмана, которые были быстро переведены на русский язык и изданы у нас в стране [1–3]. Вскоре стало ясно, что метод динамического программирования тесно связан с классическим методом Гамильтона–Якоби в аналитической механике (для систем с непрерывным временем) и с последовательным анализом Вальда (для систем с дискретным временем). Однако весьма общая и отчетливая формулировка метода динамического программирования, данная Беллманом, а также многочисленные приложения метода к разнообразным проблемам теории принятия решения, экономики, экологии и других областей знания способствовали закреплению этого метода как одного из важнейших инструментов теории управляемых процессов.

МНОГОШАГОВЫЙ ПРОЦЕСС УПРАВЛЕНИЯ

Рассмотрим управляемую систему, состояние которой в каждый момент времени характеризуется n-мерным вектором x с компонентами $x_1, \ldots, x_n$. Предполагаем, что время t изменяется дискретно и принимает целочисленные значения 0, 1, ... Так, для процессов в экономике и экологии дискретным значениям времени могут отвечать дни, месяцы или годы, а для процессов в электронных устройствах интервалы между соседними дискретными моментами времени равны времени срабатывания устройства. Предполагаем, что на каждом шаге на систему оказывается управляющее воздействие при помощи m-мерного вектора управления u с компонентами $u_1, \ldots, u_m$. Таким образом, в каждый момент времени t состояние системы характеризуется вектором $x(t)$, а управляющее воздействие – вектором $u(t)$. На выбор управления обычно бывают наложены ограничения, которые в достаточно общей форме можно представить в виде

$$u(t) \in U, \qquad t = 0, 1, \ldots \tag{1}$$

Здесь U – заданное множество в n-мерном пространстве.

Под влиянием выбранного в момент t управления (принятого решения) система переходит в следующий момент времени в новое состояние. Этот переход можно описать соотношением

$$x(t+1) = f(x(t), u(t)), \qquad t = 0, 1, \ldots \tag{2}$$

Здесь $f(x, u)$ – n-мерная функция от n-мерного вектора x и m-мерного вектора u, характеризующая динамику рассматриваемой системы. Эта функция предполагается известной (заданной) и отвечает принятой математической модели рассматриваемого управляемого процесса.

Зададим еще начальное состояние системы

$$x(0) = x^0, \tag{3}$$

где x^0 – заданный n-мерный вектор. Таким образом, многошаговый процесс управления описывается соотношениями (1)–(3). Процедура расчета конкретного процесса сводится к следующему. Пусть в некоторый момент t состояние системы $x(t)$ известно. Тогда для определения состояния $x(t+1)$ необходимо выполнить две операции: 1) выбрать допустимое управление $u(t)$, удовлетворяющее условию (1); 2) определить состояние $x(t+1)$ в следующий момент времени согласно (2). Так как начальное состояние системы задано, то описанную процедуру можно последовательно выполнить для всех $t = 0, 1, \ldots$ Последовательность состояний $x(0), x(1), \ldots$ часто называется траекторией системы.

Заметим, что выбор управления на каждом шаге содержит значительный произвол. Этот произвол исчезает, если задать цель управления в виде требования минимизации (или максимизации) некоторого критерия оптимальности. Таким образом мы приходим к постановке задачи оптимального управления.

ЗАДАЧА ОПТИМАЛЬНОГО УПРАВЛЕНИЯ

Пусть задан некоторый критерий качества процесса управления (критерий оптимальности) вида

$$J = \sum_{t=0}^{N-1} R(x(t), u(t)) + F(x(N)). \tag{4}$$

Здесь $R(x, u)$ и $F(x)$ – заданные скалярные функции своих аргументов, N – момент окончания процесса, $N > 0$. При этом функция R может отражать расход средств или энергии на каждом шаге процесса, а функция F – характеризовать оценку конечного состояния системы или точность приведения в заданное состояние.

Задача оптимального управления формулируется как задача определения допустимых управлений $u(0), u(1), \ldots, u(N-1)$, удовлетворяющих ограничениям (1), и соответствующей траектории, то есть последовательности $x(0), x(1), \ldots, x(N)$, которые в совокупности доставляют минимальное значение критерию (4) для процесса (2), (3).

Минимизация критерия (4) обычно отвечает выбору управления, обеспечивающего наименьшие затраты средств, ресурсов, энергии, наименьшее отклонение от заданной цели или заданной траектории процесса. Наряду с этим часто ставится также задача о максимизации критерия вида (4), например о максимизации дохода или объема производства. Однако нетрудно видеть, что максимизация критерия J эквивалентна минимизации критерия $(-J)$. Поэтому простая замена знака у функций R и F в (4) приводит задачу о максимизации критерия к задаче о его минимизации. Далее всюду для определенности рассматриваем задачу о минимизации критерия (4).

ЭЛЕМЕНТАРНЫЙ ПОДХОД

Рассмотрим сначала элементарный подход к поставленной задаче оптимального управления. При помощи соотношений (2) состояние системы в каждый последующий момент времени выражается через ее состояние и управление в предыдущий момент времени. Применяя это соотношение многократно, можно выразить состояния системы во все моменты времени только через начальное состояние x^0 и управления в предшествующие моменты. В результате получим из (4)

$$J = R(x^0, u(0)) + R(f(x^0, u(0)), u(1)) + \ldots =$$

$$= \Phi(x^0, u(0), u(1), \ldots, u(N-1)).$$

Здесь Φ – некоторая громоздкая, но, вообще говоря, известная функция своих аргументов. Таким образом, поставленная задача оптимального управления свелась к задаче о минимизации функции Φ от векторов $u(0), u(1), \ldots, u(N-1)$, то есть от Nm переменных. При больших N (а обычно представляют интерес именно процессы с большими N) эта задача о минимизации функции большого числа переменных представляет трудности даже при использовании мощных компьютеров. Дополнительное осложнение вызвано тем, что переменные $u(t)$ должны удовлетворять ограничениям (1).

Принципиально иной подход к поставленной проблеме дает метод динамического программирования.

ПРИНЦИП ОПТИМАЛЬНОСТИ

Сформулированный Р. Беллманом принцип оптимальности гласит: отрезок оптимального процесса от любой его точки до конца процесса сам является оптимальным процессом с началом в данной точке.

Принцип оптимальности легко доказывается от противного. Пусть $x(t) = x^*$ – некоторая точка

оптимальной траектории, то есть состояние системы вдоль оптимального процесса в момент t, $0 < t < N$. Рассуждая от противного, предположим, что отрезок этого процесса от момента t до момента N не является оптимальным процессом для системы (1), (2) в смысле критерия качества

$$J_t = \sum_{k=t}^{N-1} R(x(k), u(k)) + F(x(N)) \qquad (5)$$

при начальном условии $x(t) = x^*$. Значит, существуют допустимое управление $\tilde{u}(t), ..., \tilde{u}(N-1)$ и соответствующая ему траектория $\tilde{x}(t+1), ..., \tilde{x}(N)$, для которых критерий J_t из (5) принимает меньшее значение, чем на исходном оптимальном процессе. На рис. 1 исходная оптимальная траектория $x(t)$ показана красной ломаной, а траектория $\tilde{x}(t)$ – голубой. Наряду с исходным оптимальным процессом $x(k)$, $k = 0, 1, ..., N$, рассмотрим процесс, состоящий из двух участков: исходного процесса $x(k)$ при $k = 0, 1, ..., t$ и "улучшенного" процесса $\tilde{x}(t)$ при $k = t+1, ..., N$. Для этого составного процесса критерий J из (4) будет иметь меньшее значение, чем для исходного процесса, так как сумма первых t слагаемых в (4) для составного процесса останется той же, что и для исходного процесса, а сумма остальных слагаемых, равная J_t из (5), уменьшится по сравнению с исходным процессом. Данное утверждение означает, что исходный процесс не является оптимальным, а это противоречит сделанному предположению.

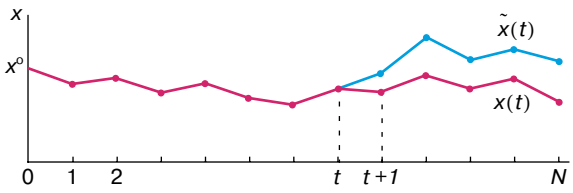

Рис. 1. Траектории управляемого процесса

Таким образом, принцип оптимальности доказан. Столь простое доказательство наводит на мысль о тривиальности этого принципа. Однако это не так: принцип оптимальности является следствием аддитивности критерия оптимальности (4) и не имеет места в случае неаддитивного критерия, например для критерия, являющегося некоторой функций от критериев вида (4).

МЕТОД ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Обозначим: $S(x, t)$ – минимальное значение критерия качества J_t из (5) для оптимального процесса, начинающегося в момент t в точке $x(t) = x$. Этот процесс можно представить состоящим из двух участков: первого шага, на котором выбирается управление $u(t) = u$, и остальной части (от момента $t+1$ до конца процесса). Вклад в критерий качества первого участка процесса равен $R(x, u)$, а вклад второго участка можно, согласно принципу оптимальности, выразить через введенную выше функцию S в виде $S(x(t+1), t+1)$. Учитывая, что управление на первом участке должно выбираться из условия минимизации критерия J_t при ограничении (1), получим равенство

$$S(x, t) = \min_{u \in U} [R(x, u) + S(x(t+1), t+1)].$$

Здесь и далее для определенности предполагаем, что функция S, как и ранее введенные в (2), (4) функции f, R, F, непрерывна. Подставляя в полученное соотношение равенство (2), получим основное соотношение метода динамического программирования

$$S(x, t) = \min_{u \in U} [R(x, u) + S(f(x, u), t+1)],$$
$$t = 0, 1, ..., N-1. \qquad (6)$$

Для оптимального процесса, начинающегося в момент $t = N$, критерий оптимальности (5) сводится к одному последнему слагаемому. Поэтому имеем

$$S(x, N) = F(x). \qquad (7)$$

Соотношение (6) и условие (7), играющее роль начального условия, дают возможность последовательно определить функции $S(x, t)$ при $t = N-1, ..., 1, 0$, а также рассчитать оптимальное управление и оптимальные траектории. Это достигается при последовательной реализации попятной и прямой процедур динамического программирования.

ПОПЯТНАЯ ПРОЦЕДУРА

Заметим, что при вычислении минимума функции по некоторому аргументу обычно определяются две величины: значение минимума и значение аргумента, при котором минимум достигается. Это значение, которое может быть неединственным, будем обозначать символом arg min.

Положим $t = N-1$ в (6) и воспользуемся условием (7). Получим

$$S(x, N-1) = \min_{u \in U} [R(x, u) + F(x)].$$

Вычисляя этот минимум, найдем функцию $S(x, N-1)$ и значение u, доставляющее данный минимум:

$$u = \upsilon_{N-1}(x) = \arg\min_{u \in U} [R(x, u) + F(x)].$$

Запись $\upsilon_{N-1}(x)$ означает, что значение u зависит от x как от параметра. Определив $S(x, N-1)$ и полагая $t = N-2$, найдем из (6) функцию $S(x, N-2)$ и соответствующее значение аргумента $u = \upsilon_{N-2}(x)$.

Продолжая этот процесс в сторону уменьшения t, получим из (6) последовательно функции $S(x, t)$ и

$$\upsilon_t(x) = \arg\min_{u \in U} [R(x, u) + S(f(x, u), t + 1)] \qquad (8)$$

при $t = N - 1, N - 2, ..., 1, 0$. Отметим, что функция $\upsilon_t(x)$ определяет оптимальное управление в момент t при условии, что система находится в состоянии x. Эта форма задания управления называется управлением по обратной связи.

Таким образом, попятная процедура состоит в построении функций $S(x, t)$ и $\upsilon_t(x)$ для всех x и $t = 0, 1, ..., N$. Это построение в отдельных случаях может быть выполнено аналитически, но, как правило, является трудоемкой вычислительный процедурой. Ниже мы обсудим вычислительные аспекты, а пока предположим, что эта процедура так или иначе реализована.

ПРЯМАЯ ПРОЦЕДУРА

Воспользуемся результатами попятной процедуры для решения исходной задачи, то есть для построения оптимального управления и оптимальной траектории при заданном начальном условии (3).

Полагая $t = 0$ и $x = x^0$ в (8), найдем управление в начальный момент: $u(0) = \upsilon_0(x^0)$. Далее из соотношения (2) определим состояние $x(1) = f(x^0, u(0))$. Продолжая этот процесс, найдем $u(1) = \upsilon_1(x(1))$, $x(2)$ и т.д. Вообще имеем

$$u(t) = \upsilon_t(x(t)), \qquad x(t + 1) = f(x(t), u(t)), \qquad (9)$$

$$x(0) = x^0, \qquad t = 0, 1, ..., N - 1.$$

Соотношения (9) определяют прямую процедуру и позволяют полностью рассчитать оптимальное управление и оптимальную траекторию. Минимальное значение критерия оптимальности, отвечающее этой траектории, $J = S(x^0, 0)$.

Пример

В качестве иллюстративного примера рассмотрим модельную задачу об оптимальном функционировании фермы по разведению скота или птицы. Пусть x – число животных (или птиц) на ферме в начале некоторого интервала времени. Из этого числа ux животных отправляется на продажу, а остальные животные приносят приплод, так что их число возрастает в q раз за рассматриваемый интервал. Уравнение (2) примет вид

$$x(t + 1) = q[1 - u(t)]x(t), \qquad t = 0, 1, ... \qquad (10)$$

Здесь $q > 1$ – постоянный коэффициент, u – управляющее воздействие (доля животных, отправляемых на продажу). Ограничение (1) в данном случае имеет вид $0 \le u \le 1$.

Расходы на содержание животных примем пропорциональными их оставшемуся числу и равными $a[1 - u(t)]x(t)$, где $a > 0$ – постоянная. Выручку от продажи считаем равной $cu(t)x(t)$, где c – цена одного животного на рынке. Поставим задачу максимизации дохода фермы за N шагов по времени. Как отмечалось выше, эта задача эквивалентна минимизации убытка (то есть дохода со знаком минус). Критерий оптимальности имеет вид (4), где нужно принять в соответствии со сказанным выше

$$R(x, u) = a(1 - u)x - cux, \qquad F(x) = -cx. \qquad (11)$$

Последнее равенство (11) определяет (со знаком минус) стоимость животных на ферме в конце процесса. Учитывая соотношения (10) и (11), составим уравнение (6) для рассматриваемой задачи

$$S(x, t) =$$

$$= \min_{0 \le u \le 1} [a(1 - u)x - cux + S(q(1 - u)x, t + 1)]. \qquad (12)$$

Условие (7) с учетом (11) примет вид

$$S(x, N) = -cx. \qquad (13)$$

Несложный анализ позволяет реализовать попятную процедуру и построить функции $S(x, t)$ и соответствующие им управления $\upsilon_t(x)$ из (9) для задачи (12), (13). Приведем окончательные результаты:

$$S(x, t) = a(q^{N-1} - 1)(q - 1)^{-1}x - cq^{N-1}x,$$

$$\upsilon_t(x) = 0 \quad \text{при} \quad a < c(q - 1); \qquad (14)$$

$$S(x, t) = -cx, \qquad \upsilon_t(x) = 1 \quad \text{при} \quad a > c(q - 1).$$

В том, что функции $S(x, t)$ из (14) удовлетворяют уравнению (12) и условию (13) при всех t, можно убедиться методом математической индукции, проводя ее в сторону убывания t и начиная с $t = N$.

Решения (14) допускают простую интерпретацию. Если $a < c(q - 1)$, то есть расходы на содержание животных сравнительно невелики, то имеет смысл не направлять животных на продажу ($\upsilon_t(x) = 0$) и получить наибольший доход, сохранив все поголовье к концу процесса. Если же $a > c(q - 1)$, то есть расходы на содержание велики, то целесообразно отправить на продажу сразу всех животных. В случае $a = c(q - 1)$ оптимальное управление неединственно: в этом случае любое управление приводит к одному и тому же результату.

Более сложные и более содержательные результаты получим, если учтем зависимость рыночной цены c от числа продаваемых животных x (цена убывает с ростом x), а также зависимость расходов на содержание одного животного a от числа животных на ферме.

ОБСУЖДЕНИЕ

Как показано выше, попятная и прямая процедуры метода динамического программирования в совокупности дают способ решения поставленной задачи оптимального управления. Однако при реализации

этого метода мы получили значительно более общий и универсальный результат. В ходе попятной процедуры построено оптимальное управление по обратной связи (8), то есть управление как функция текущего состояния системы $\upsilon_t(x)$. Теперь нетрудно дать решение задачи оптимального управления при любом начальном условии вида $x(t) = x^*$: для этого нужно просто реализовать прямую процедуру для этого начального условия. Реализация прямой процедуры не представляет серьезных трудностей и сводится, согласно (9), к вычислению известных функций при конкретных значениях аргументов.

Наибольшую сложность представляет попятная процедура, включающая минимизацию функций по m-мерному векторному аргументу u. Здесь нужно, согласно (6), выполнить N процедур минимизации функций от m переменных. Это, вообще говоря, значительно более простая задача, чем минимизация одной функции по Nm переменным, что требуется при элементарном подходе.

Однако есть одно серьезное осложнение. Дело в том, что попятная процедура предусматривает построение функций $S(x, t)$ и $\upsilon_t(x)$, зависящих от n-мерного вектора x. По этому вектору x не нужно выполнять операций минимизации, но даже простое табулирование и хранение функций n переменных представляют большие трудности. Если, к примеру, составлять таблицы, придавая каждой переменной 100 значений, то для хранения таблицы одной функции n переменных понадобится 100^n ячеек памяти. Всего же попятная процедура, в которой участвуют функции $S(x, t)$ и $\upsilon_t(x)$, потребует $(m + 1)N \times \times 100^n$ ячеек.

Отсюда понятно, что вычислительная реализация метода динамического программирования сталкивается с большими трудностями. Эти трудности Р. Беллман назвал проклятием размерности. Для преодоления этих трудностей были предложены подходы [2–5], позволяющие сократить объем вычислений и потребности в памяти при построении оптимального управления. При этом, однако, приходится либо существенно пожертвовать точностью вычислений, либо отказаться от построения управления, оптимального в глобальном смысле, и ограничиться нахождением управлений и траекторий, оптимальных в локальном смысле, то есть по отношению к малым (локальным) вариациям этих траекторий.

НЕКОТОРЫЕ ОБОБЩЕНИЯ

Остановимся сначала на таких обобщениях постановки задачи (1)–(4), которые требуют лишь небольших и очевидных модификаций метода динамического программирования.

1. Ограничения, наложенные на управляющие воздействия, могут зависеть от текущего состояния системы и времени. В этом случае вместо (1) имеем условие

$$u(t) \in U(x(t), t), \tag{15}$$

где $U(x(t), t)$ – множество в m-мерном пространстве, зависящее от n-мерного вектора x и от времени t.

2. Соотношение (2), определяющее переход системы из одного состояния в другое, может явно зависеть от времени. В этом случае вместо (2) имеем

$$x(t + 1) = f(x(t), u(t), t), \tag{16}$$

где $f(x, u, t)$ – заданная функция своих аргументов.

3. Критерий оптимальности (4) также может содержать слагаемые, явно зависящие от времени, то есть иметь вид

$$J = \sum_{t=0}^{N-1} R(x(t), u(t), t) + F(x(N)), \tag{17}$$

где $R(x, u, t)$ – заданная функция.

Нетрудно убедиться, что при замене соотношений (1), (2), (4) на соответствующие соотношения (15), (16), (17) все рассуждения, обосновывающие принцип оптимальности и процедуру динамического программирования, остаются в силе. Окончательно основное соотношение (6) и равенство (8) приобретают вид

$$S(x, t) = \min_{u \in U(x, t)} [R(x, u, t) + S(f(x, u, t), t + 1)],$$

$$\upsilon_t(x) = \arg \min_{u \in U(x, t)} [R(x, u, t) + S(f(x, u, t), t + 1)].$$

Соответствующие очевидные изменения следует внести и в другие соотношения.

Выше предполагалось, что время окончания процесса конечно и фиксировано ($t = N$). Метод динамического программирования применим также к задачам с нефиксированным моментом окончания, для процессов на бесконечном интервале времени, а также при наличии различных краевых условий и ограничений на состояние системы, наложенных в различные моменты времени (так называемые фазовые ограничения).

Метод динамического программирования используется для исследования и решения задач оптимального управления процессами в условиях неопределенности. Здесь имеются в виду два различных типа проблем. Если неопределенные факторы (внешние воздействия, возмущения, шумы) имеют случайную природу, то для описания процессов управления используется аппарат теории вероятности и случайных процессов. Если же неопределенные факторы связаны с активным противодействием противника или необходимо обеспечить гарантированный результат (застраховаться от наихудших возможных реализаций неконтролируемых возмущений), то применяется аппарат теории игр. Метод

динамического программирования используется также для построения оптимальных вычислительных алгоритмов поиска корней и экстремумов функций [6].

Отдельно следует сказать о задачах оптимального управления с непрерывным временем, описываемым дифференциальными уравнениями. В этом случае метод динамического программирования приводит к нелинейному дифференциальному уравнению в частных производных первого порядка. Теория решений этого уравнения, иногда называемого уравнением Гамильтона–Якоби–Беллмана–Айзекса, активно разрабатывается в последние годы.

ЛИТЕРАТУРА

1. *Беллман Р.* Динамическое программирование. М.: Изд-во иностр. лит., 1960. 400 с.

2. *Беллман Р.*, *Дрейфус С.* Прикладные задачи динамического программирования. М.: Наука, 1965. 458 с.

3. *Беллман Р.*, *Калаба Р.* Динамическое программирование и современная теория управления. М.: Наука, 1969. 118 с.

4. *Черноусько Ф.Л.*, *Баничук Н.В.* Вариационные задачи механики и управления: Численные методы. М.: Наука, 1973. 238 с.

5. *Моисеев Н.Н.* Элементы теории оптимальных систем. М.: Наука, 1975. 526 с.

6. *Черноусько Ф.Л.*, *Меликян А.А.* Игровые задачи управления и поиска. М.: Наука, 1978. 270 с.

* * *

Феликс Леонидович Черноусько, доктор физико-математических наук, профессор Московского физико-технического института, главный научный сотрудник Института проблем механики РАН, академик РАН, лауреат Государственной премии СССР, премии Ленинского комсомола, международной премии Кербера за развитие европейской науки (Германия). Область научных интересов: теория управления, механика, прикладная математика. Автор девяти монографий и более 250 научных статей.