Лекция #2

Этапы решения прикладных задач оптимизации

Первый этап решения сложных задач оптимизации – это содержательная постановка задачи. На этом этапе устанавливается, какая цель должна быть достигнута и что конкретно должно быть определено в процессе решения задачи. Например, требуется разработать оптимальных проект некоторого сооружения, оптимальное распределение финансовых ресурсов и тп. Прежде всего должно быть установлено, что значит «оптимальный». На этот вопрос могут быть разные ответы: например, оптимальное сооружение – это сооружение с минимальной стоимостью или с максимальным сроком службы и тд. В любом случае речь идет об экстремальных значениях одного или нескольких количественных показателей, по которым можно сравнивать варианты принимаемого решения. Такие показатели называются критериями оптимальности. Если критерий оптимальности один, то задача называется однокритериальной, в противном случае – многокритериальной. Может оказаться, что достижение экстремального (например, минимального) значения одного критерия автоматически означает достижение экстремального значения другого. В этом случае достаточно рассматривать только один из них. Но чаще бывает так, что критерии противоречат друг другу и оптимизация по каждому из них в отдельности приводит к разным результатам. К примеру, минимум стоимости и максимум срока службы, как правило, противоречат друг другу, им соответствуют различные варианты допустимых решений. Стремление достичь сразу нескольких целей, характерное при постановке многокритериальных задач, приводит к противоречию, которое может быть разрешено или сведением многокритериальной задачи к однокритериальной (посредством достижения некоторого компромисса), или уточнением самого понятия оптимальности так, чтобы оно было приемлемо в данном случае.

Есть два способа сведения многокритериальных задач к однокритериальным.

1. <u>Ранжирование критериев оптимальности (метод уступок).</u>

   Все критерии рассматриваются последовательно в порядке убывания их значимости, т.е важности для лица, принимающего решение. Получаем последовательность критериев $K_1, K_2, \dots, K_n$.
   Решается задача оптимизации по главному критерию K_1, в результате чего определяется его оптимальное значение K_1^*. Затем назначается величина уступки - u_1, т.е. максимальное значение отклонения от K_1, которое можно допустить при оптимизации по следующему по важности критерию K_2. Оптимизация по K_2 выполняется при дополнительном условии: нельзя отклоняться от K_1 более чем на u_1. Если по первому критерию задача решалась на минимум, то это дополнительно ограничение имеет вид $K_1 \leq K_1^* + u_1$, а если на максимум – то $K_1 \geq K_1^* - u_1$. Определяется оптимальное значение второго критерия K_2^* при уступке по первому, назначается уступка по второму критерию u_2, вводится еще одно дополнительное ограничение, теперь уже на отклонение K_2 от K_2^*, решается задача оптимизации по K_3 и тд.
   В этом методе вместо многокритериальной задачи решается несколько однокритериальных задач (по числу критериев). Причем для каждого следующего критерия вводится дополнительное ограничение на величину предыдущего, которое может существенно осложнить поиск оптимального решения.

2. <u>Свертка критериев.</u>
   Вместо нескольких критериев вводится один новый в виде из взвешенной суммы.

$$K = v_1 K_1 + v_2 K_2 + \dots + v_n K_n, \qquad \sum_{i=1}^{n} v_i = 1.$$

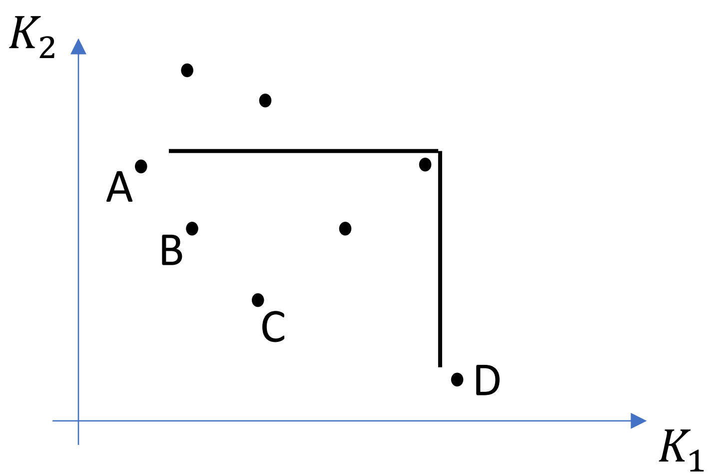
Весовые множители характеризуют относительную важность критериев. Это положительные числа, сумма которых равна единице. Фактически речь снова идет о ранжировании критериев, но с заданием количественных характеристик важности каждого критерия. Это можно сделать следующим образом. Важность главного критерия принимаем за единицу ($w_1 = 1$), а для каждого из остальных устанавливаем его относительную важность по сравнению с главным $w_i (i = 2, \dots, n)$. Все полученные положительные числа должны быть меньше единицы. Затем каждое из них, в том числе и важность главного критерия, делим на их сумму и получаем весовые коэффициенты $v_i, i = \overline{1, n}$. Можно в качестве весовых коэффициентов принять и величины w_i, не добиваясь равенства единице суммы весов, т.к умножение критерия на положительное число не меняет оптимального решения.

Заметим, что в некоторых задачах можно просто выбрать главных критерий, а по всем остальным задать дополнительные ограничения на соответствующие величины. Например, требуется создать конструкцию с минимальной стоимостью и максимальным сроком службы. Вместо этого решается задача на минимум стоимости при условии, что срок службы не менее заданного.

Если в конкретной многоэкстремальной задаче не удается использовать ни один из способов ее сведения к одноэкстремальной, то приходится изменить подход к оптимизации и само понятие оптимальности. Один из таких примеров состоит в оптимизации по Парето.

Оптимизация по Парето.

Вместо оптимальных предлагается искать так называемые эффективные решения, т.е. такие решения, каждое из которых нельзя улучшить сразу по всем критериям. Множество таких решений называется множеством Парето. Естественно, что в него входят решения, оптимальные по отдельным критериям, н, как правило, не только такие. Для иллюстрации этого положения рассмотрим задачу **минимизации** по двум критериям K_1 и K_2. Предположим, что нам известны все допустимые решения, из которых нам предстоит сделать выбор по этим двум критериям, и для каждого решения вычислены величины каждого критерия. Геометрически на координатной плоскости с осями K_1 и K_2 все допустимые решения будут представлены точками (рисунок 1).



Очевидно, в это множество из всех допустимых точек попадут только точки A, B, C, D, причем только точки A и D соответствуют минимизации по отдельным критериям. Заметим, что точка

входит в множество Парето тогда и только тогда, если в третьем квадранте системы координат, построенной с центром в этой точке, нет других точек. Это правило позволяет из всех допустимых решений после вычисления значений каждого критерия сформировать множество Парето для двухкритериальной задачи. Аналогично можно поступить и при наличии трех и более критериев. Такой подход эффективен для класса двухкритериальных задач, относящихся к задачам распределения ресурсов, для которых с использованием их особенностей приводится алгоритм построения множества Парето с помощью динамического программирования.
Второй этап решения задачи состоит в формализации задачи, т.е. в ее постановке как задачи математической. При этом необходимо формализовать искомое решение, т.е. выбрать набор тех или иных переменных, численные значения которых полностью определяют конкретный вариант решения задачи.
Третий этап решения задачи оптимизации состоит в исследовании ее математической модели (целевой функции и ограничений) и выборе метода поиска оптимального решения.

Аналитический способ решения условных задач. Метод множителей Лагранжа

Рассмотрим поиск экстремума для заданной целевой функции при определенных условиях:

$$\begin{cases} Z(x) \to extr \\ K_i(x) = 0, \quad i = \overline{1,m} \end{cases}$$

Будем считать, что целевая функция Z и функции ограничений K_i являются непрерывно дифференцируемыми. Тогда такая задача называется гладкой конечномерной задачей с ограничениями типа равенств.

При решении таких задач используется метод Лагранжа. Чтобы привести его формулировку. Вначале определим функцию Лагранжа.

Поставим гладкую конечномерную задачу с ограничениями типа равенств

$$\begin{cases} \varphi_0(x) \to extr \\ \varphi_i(x) = 0, \quad i = \overline{1,m} \end{cases} \quad (1)$$

Ее область определения Ω имеет вид

$$\Omega = \{x \in \mathbb{R}^n |\ \varphi_i(x) = 0, \quad i = \overline{1,m}\}$$

Функцией Лагранжа для задачи (1) называется функция

$$L(x,\lambda) = \sum_{i=0}^{m} \lambda_i \varphi_i(x), \quad \lambda = (\lambda_0, \lambda_1, \dots, \lambda_m) \in \mathbb{R}^{m+1},$$

от пары переменных x, λ. При этом координаты вектора λ называют *множителями Лагранжа*.

Сформулируем необходимое условие экстремума для нашей задачи.

Принцип Лагранжа. *Если* $a \in \Omega$ является точкой условного экстремума в задаче (1), то существует такой ненулевой вектор $\lambda \in \mathbb{R}^{m+1}$, что

$$L'_x(a,\lambda) = 0$$

Это уравнение называется *уравнением Лагранжа.*

При решении задачи поступаем следующим образом.

1. Составим функцию Лагранжа:

$$L(x, \lambda) = \sum_{i=0}^{m} \lambda_i \varphi_i(x).$$

2. Выписываем необходимое условие экстремума:

$$\begin{cases} \dfrac{\partial L}{\partial x_j} = \sum\limits_{i=0}^{m} \lambda_i \dfrac{\partial \varphi_i(x)}{\partial x_j} = 0, \ \ j = \overline{1, n} \\ \varphi_i(x) = 0, \ \ i = \overline{1, m} \end{cases} \quad (3)$$

3. Решаем полученную систему. Если точка a является решением задачи (1), то в силу принципа Лагранжа существует такой вектор $\lambda \neq 0$, что пара (a, λ) удовлетворяет системе (3). Поэтому экстремальные точки задачи (1) находятся среди решений системы (3), у которых $\lambda \neq 0$.

Пример. Имеется два способе производства некоторого продукта. Издержки производства при каждом способе зависят от произведенных x_1 и x_2 следующим образом:

$$g(x_1) = 9x_1 + x_1^2$$
$$g(x_2) = 6x_2 + x_2^2$$

За месяц необходимо произвести 150 единиц продукции, распределив ее между двумя способами так, чтобы минимизировать общие издержки.

Решение.

Найдем минимум функции $Z(x) = 9x_1 + x_1^2 + 6x_2 + x_2^2$, используя функцию Лагранжа. Ограничения представим в неявном виде: $x_1 + x_2 - 150 = 0$

Составим функцию Лагранжа:

$$L(x_1, x_2, \lambda) = 9x_1 + x_1^2 + 6x_2 + x_2^2 + \lambda(x_1 + x_2 - 150)$$

Необходимым условием экстремума функции Лагранжа является равенство нулю ее частных производных по переменным x_i и по неопределенному множителю λ. Составим систему:

$$\begin{cases} \dfrac{\partial L}{\partial x_1} = 2x_1 + \lambda + 9 = 0 \\ \dfrac{\partial L}{\partial x_2} = 2x_2 + \lambda + 6 = 0 \\ \dfrac{\partial L}{\partial \lambda} = x_1 + x_2 - 150 = 0 \end{cases}$$

Решив эту систему, получим:

$$\begin{cases} x_1 = 74{,}25 \\ x_2 = 75{,}75 \\ \lambda = -157{,}5 \end{cases}$$

Таким образом, чтобы общие издержки производства были минимальны, необходимо производить $x_1 = 74{,}25;\ x_2 = 75{,}75$.

Многоэтапные процессы принятия решений

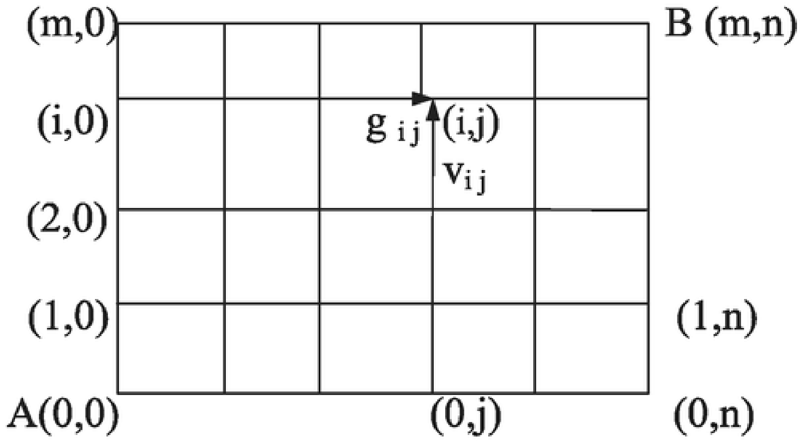
Динамическое программирование – это особый метод оптимизации, наиболее эффективный при решении задач, распадающихся на ряд последовательных этапов (шагов), таких как планирование производства и инвестиций на ряд временных интервалов (лет, кварталов, месяцев), последовательность тестовых испытаний при контроле аппаратуры, поиск оптимальной траектории движения и тд. В любом случае речь идет о процессах, в которых окончательное решение вырабатывается последовательно (по шагам), причем на каждом шаге приходится решать однотипные задачи, которые существенно проще, чем решение исходной задачи в целом. В этом и состоит основная идея метода: свести решение одной сложной задачи к решению множества однотипных, иногда совсем простых задач, например, выборка чисел из массива, суммирование и сравнение результатов.

Основная идея, которая привела к созданию вычислительного метода, была сформулирована в начале 50х прошлого века Р. Беллманом, сделавшим самый большой вклад в развитие метода, который он назвал «динамическое программирование», но который чаще всего называется методом Беллмана.

Метод не является универсальным, причем далеко не всегда в конкретной практической задаче удается обосновать без дополнительного анализа возможность его применения, а тем более отсутствие такой возможности.

Первоначально метод предлагали для решения сравнительно узкого класса задач, возникающих в процессах, которые развиваются во времени. Но через некоторое время после появления первых работ Беллмана выяснилось, что для многих задач, которые не являются многоэтапными в явном виде. Эту многоэтапность можно организовать искусственно и применить для них этот метод.

Для изложения основной идеи метода рассмотрим сначала простую задачу поиска оптимального пути на двумерной прямоугольной сетке, в которой разрешены переходы из одного узла в другой только по горизонтали (вправо) или по вертикали (вверх). Заданы затраты на каждый из возможных переходов и требуется найти пусть с минимальными суммарными затратами из левого нижнего угла сетки (точка A) в правый верхний угол (точка B). Такой путь называется оптимальным. Узлы сетки пронумерованы так, как показано на рисунке, где m и n задают соответственно вертикальный и горизонтальный размеры сетки (число шагов по вертикали и горизонтали соответственно).



Затраты на переход в узел (i,j) по горизонтали (из узла $(i,j-1)$) обозначим как g_{ij}, а по вертикали как v_{ij}. В точке A соответствующие величины равны нулю. Таким образом, исходными данными в этой задаче являются: $m, n, g_{ij}, v_{ij}, i = \overline{1,m}, j = \overline{1,n}$. Всего $n(m+1)$ чисел g_{ij} и $m(n+1)$ чисел v_{ij}, т.е. всего $2mn+m+n$ переходов и соответствующих им затрат.

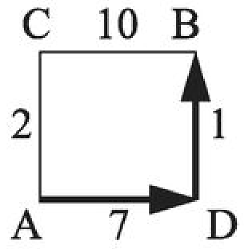
При небольших размерах сетки можно попытаться решить задачу методом полного перебора вариантов возможных путей из точки A в точку B. Однако эта идея совершенно бесперспективна уже при величинах m и n порядка 10 из-за резкого роста числа возможных вариантов путей с увеличением размеров сетки. Действительно, каждому варианту пути из точки A в точку B соответствует ровно m шагов по вертикали и n шагов по горизонтали, но последовательность этих шагов для каждого варианта своя. Если шагу по горизонтали поставить в соответствие 0, а шагу по вертикали 1, то очевидно, что вариант пути – это выбор размещения m единиц по $m+n$

возможным местам (оставшиеся n мест займут нули). Для размещения первой единицы есть $m + n$ возможностей, для второй - $m + n - 1$ и тд. В итоге получаем формулу для числа вариантов пути N:

$$N = \frac{(m+n)!}{m!\,n!}$$

Уже при $m = n = 10$ $N = 184756$.

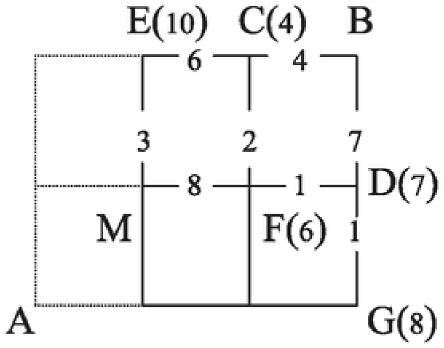
Следующая идея состоит в том, чтобы из точки A идти в том направлении, которое требует минимальных затрат на первом шаге, не думая о затратах на последующих шагах, и так в каждой точке. Т.е. рассматривать только затраты на данном шаге и выбирать тот переход, для которого на данном шаге затраты минимальны. Легко убедиться в ошибочности данной идеи даже при $m = n = 1$.



Действительно, если первый шаг выбрать по вертикали в точку C (затраты 2 против 7), то в итоге после второго шага получим суммарные затраты равные 12, а при выборе на первом шаге «неоптимального» решения (точка D) суммарные затраты равны всего лишь 8.

Значит, нужно смотреть далее. Чем на один ход вперед.

Поиск оптимального пути можно рассматривать как многошаговый процесс. На первом шаге мы находимся в точке А и имеем два возможности: пойти вверх или направо. Мы убедились, что сделать этот выбор нельзя, тк нужно учитывать последствия этого выбора. Однако существуют точки, находясь в которых мы не имеем выбора. Это точки C и D находятся в одном шаге от последней точки B и имеют координаты $(m, n-1)$ и $(m-1, n)$.

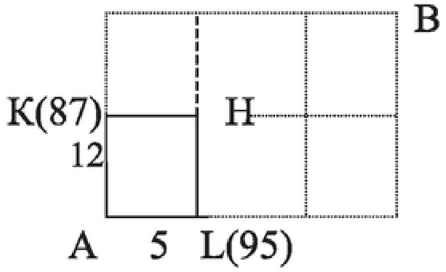


Т.е. мы будем решать задачу с конца. Для каждой из этих точек запомним затраты на оставшийся до точки B путь и рассмотрим предпоследний шаг. В двух шагах от финиша мы можем быть в точках E, F или G. В точках E и F выбора нет, мы просто запомним для каждой из них суммарные затраты на весь оставшийся путь. Если же мы оказались в точке F, то нужно идти в точку C, т.к. суммарные затраты на путь окажутся меньше. Это работает при одном важном условии: ничто не может помешать нам это сделать, т.е нет никакой связи с тем, как мы попали в эту точку, или, как говорят, нет «предыстории». (пример про гистерезис)

Продолжив поиск оптимального пути в нашей задаче без предыстории. В точке F мы запомним затраты на весь оставшийся путь при условии, что выбран оптимальный вариант: переход в точку C. Сделав еще шаг назад, т.е. оказавшись за три шага до финиша, мы увидим, что ситуация полностью аналогична предыдущей. Выбора или нет (точки на границе), или есть две возможности, но для каждой из них уже известны последствия выбора. Так, оказавшись в точке M, мы выберем не точку F, для которой затраты до конца пути равны 6, а оказавшуюся бесперспективной точку E, для которой эти затраты равны 10, но суммарные затраты на весь оставшийся путь меньше. Поступая аналогично, мы рано или поздно в своем обратном движении придем в начальную точку A. Но при этом нам уже будут известны последствия для каждого из вариантов выбора, тк для них будут уже вычислены и записаны затраты на весь оставшийся путь до точки B.

В данной задаче мы исследовали многошаговый процесс от последнего шага к первому. Но ничто не мешает развернуть его и рассмотреть шаги в прямом направлении, ничего не меняя в методе Беллмана по существу.

Применительно к рассматриваемой задаче алгоритм поиска пути будет выглядеть следующим образом.



1. Из точки A делаем шаг в каждом из возможных направлений, запоминаем в точках K и L затраты и направление, по которому пришли в эту точку (0 по горизонтали и 1 по вертикали). Соответственно, для точки K запоминаем 12 и 1, а для точки L 5 и 0.
2. На всех последующих шагах, кроме последнего, если в точку ведет один путь (шаги по краям), то просто запоминаем суммарные затраты на путь от начала и направление, откуда пришли, а если в точке сходятся два варианта, то сравниваем две возможности: прийти в эту точку по горизонтали или по вертикали. Для каждой из них вычисляем суммарные затраты на путь от начала до данной точки, выбираем тот вариант, для которого эти суммарные затраты минимальны, запоминаем их и соответствующее им направление. Т.о, происходит отбраковка вариантов, сходящихся в точке: вариант с наибольшими затратами отбрасывается и все его продолжения далее не анализируются.
3. На последнем шаге в точку B ведут два направления, а по каждому из них все известно: для каждой точки записаны затраты на весь путь от начала и направление, откуда пришли в эту точку. Снова суммируем весь путь, выбираем наименьшие затраты, а затем обратным разворотом восстанавливаем оптимальный путь.

Принцип Беллмана, лежащий в основе динамического программирования, гласит: **если в каждом из состояний дальнейшее поведение системы не зависит от того, как она попала в это состояние, то дальнейшая траектория должна быть оптимальной.**

Приведенная формулировка принципа оптимальности означает, что речь идет только о системах без бедыстории. В этом случае из каждого промежуточного состояния можно отдельно, независимо от пройденных этапов, решать задачу поиска оптимального пути в конечное состояние. Или, что фактически то же самое, сравнивать и отбраковывать варианты достижения любого промежуточного состояния из начальной точки и оставлять только один вариант. Это означает возможность применения принципа оптимальности в несколько ином виде: если в каждом из состояний дальнейшее поведение системы не зависит от того, как она попала в это состояние, то это состояние должно достигаться по оптимальной траектории.

Итак, пусть в каждом из состояний, в котором может находиться система в начале очередного этапа, известны все возможные воздействия на нее. И для каждого такого воздействия известны его последствия, т.е. и состояние. В которое перейдет система, и затраты на этот переход. Воздействие на i-м шаге оозначми через x_i. Эти воздействия часто называют шаговыми управлениями. Если число этапов обозначить через n, то задача состоит в поиске последовательности шаговых управлений, т.е. вектора $x(x_1, x_2, \dots, x_n)$. Поскольку начальное состояние системы задано, то последовательность шаговых управлений однозначно определяет последовательность переходов системы из одного состояния в другое, т.е. траекторию движения. При этом x_i это не обязательно числа. Это могут быть векторы или функции.

Обозначим затраты на i-м шаге через z_i. Требуется найти такую последовательность шаговых управлений x_i, пи которой суммарные затраты будут минимальны.

$$Z = \sum_{i=1}^{n} z_i \to min$$

Ту последовательность x, при которой достигается минимум, будем называть оптимальным управлением и обозначать через $x^*(x_1^*, x_2^*, \dots, x_n^*)$. Соответствующие ей минимальные по всем возможным последовательностям x затраты $Z^* = \min(Z(x))$.

Затраты на i-м шаге зависят не только от x_i, но и от состояния S, в котором была система до воздействия x_i, т.е. фактически от всех предыдущих шаговых управлений. Состояние S', в которое перейдет система, зависит только от S и от x_i, если нет влияния предыстории. Формально это можно записать в виде $S' = f(S, x_i)$, где f – заданная функция, тк известны последствия воздействия x_i на систему в состоянии S. В соответствии с принципом оптимальности x_i надо выбирать так, чтобы суммарные затраты на все последующие этапы были минимальны. Эти суммарные затраты зависят от состояния S и складываются из затрат на i-м шаге $z_i(S, x_i)$ и на всех последующих шагах. Суммарные затраты на все шаги, начиная с i-го и до конца, обозначим Z_i. Тогда можно записать $Z_i = z_i + Z_{i+1}$ и оптимальное управление надо на каждом шаге выбирать так, чтобы

$$Z_i(S) \min_{x_i}\{z_i(S, x_i) + Z_{i+1}(f(S, x_i))\}$$

Это и есть основное рекуррентное уравнение динамического программирования, выражающее затраты на все оставшиеся этапы из любого состояния S через затраты на данном z_i и на всех последующих шагах $Z_{i+1}(S')$. Только на последнем шаге из любого состояния S можно легко найти оптимальный переход в конечное состояние. Если конечное состояние единственное, то и для каждого из состояний S на последнем шаге этот переход (т.е. шаговое управление) единственное.

Отметим, что поиск оптимального перехода из заданного состояния на i-м шаге может быть сложнее, чем выборка из чисел заданного массива, суммирование и сравнение. В сложных задачах использование принципа Беллмана может приводить к разностным, дифференциальным уравнениям и тд.

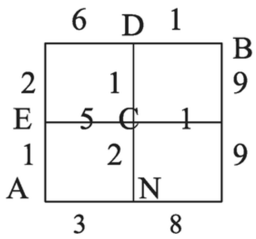
Область применения динамического программирования

Общее условие применимости метода Беллмана выражается в требовании отсутствия влияния «предыстории».

Установить возможность применения метода – значит, доказать отсутствие влияния предыстории. Это влияние может возникать при наличии более сложных целевых функций, чем сумма затрат. Может оказаться, что даже если удается разбить задачу на ряд последовательных этапов, значение целевой функции можно вычислить только при полностью известной траектории. Особенности целевой функции, даже при отсутствии каких-либо ограничений, могут создавать предысторию и тем самым осложнять или вообще делать невозможным применение метода. Например, целевая функция представляет собой сумму произведений затрат на двух последовательных этапах:

$$z_1 z_2 + z_2 z_3 + \cdots + z_{n-1} z_n \to min$$

Сравнивая варианты (например, варианты достижения узла на двумерной сетке) после первых двух этапов и отдавая предпочтение варианту с меньшим произведением затрат $z_1 z_2$, мы никак не учитываем, что важно не только это произведение, но и конкретно z_2, тк от него зависит следующее слагаемое. Т.е. нам важно не только значение целевой функции, но и как оно получено. Может оказаться, что выгоднее оставить вариант с большим значением $z_1 z_2$, но с меньшим значением z_2.



Если сравнивать варианты AEC и ANC, то останется вариант AEC, что дает решение $AECDB$ со значением целевой функции 11. Оптимальным же является вариант $ANCDB$ со значением целевой функции 9.

Во многих задачах влияние предыстории можно преодолеть за счет усложнения формализации понятия «состояние». Так, если в задаче поиска оптимального пути на двумерной сетке считать состоянием не узел, а два последних пройденных узла, т.е. отрезок. Из соединяющий, то сравнимыми становятся варианты, у которых общим является не один последний узел, а два, т.е. варианты, имеющие общий последний отрезок.

В книге Вентцель «Исследование операций» утверждается, что «Метод динамического программирования является очень мощным и плодотворным методом оптимизации управления. Ему не страшны ни целочисленность решений, ни нелинейность целевой функции, ни вид ограничений, накладываемых на решение». Это утверждение верно, если ограничения накладываются на каждую или некоторые переменные отдельно. Но если ограничения даже очень простого вида (например, линейные) связывают несколько переменных, то каждое новое ограничение увеличивает количество координат, задающих состояние системы (их называют параметрами состояния). А при числе параметров состояния больше трех и большом числе возможных вариантов перехода из каждого промежуточного состояния вычислительные трудности при реализации могут оказаться непреодолимыми даже для современных компьютеров. Поэтому очень важен вид ограничений, накладываемых на искомое решение.

Что касается целевой функции, то здесь важна ее структура. Существенным является не линейность целевой функции, а возможность ее вычисления по отдельным шагам по одним и тем же правилам. Если этой возможности нет, то целевая функция может быть

вычислена только после нахождения полной траектории, т.е методы динамического программирования применить не удастся.
Очень важным достоинством метода динамического программирования является его нечувствительность к отсутствию непрерывности производных целевой функции в задачах с непрерывными переменными.
Часто говорят о нечувствительности этого метода к наличию локальных экстремумов. В дискретном случае это действительно так, и это важно. Если же дискретность вводится искусственно, то здесь могут возникнуть осложнения.
Чтобы охарактеризовать тип задач, для решения которых можно применять методы динамического программирования, отметим следующее:

1. Должна быть возможность интерпретации задачи как многошагового процесса принятия решений, в котором решение, принимаемое на каждом шаге, состоит в выборе одного или нескольких управляющих переменных. На каждом шаге для каждого состояния должна быть возможность количественной оценки принятого решения, а не только выяснение состояния, в которое перейдет система. Речь идет о возможности вычисления целевой функции по шагам.
2. Задача должна быть определена для любого числа шагов и иметь структуру. Не зависящую от числа шагов.
3. При рассмотрении задачи, состоящей из k шагов, должно быть задано некоторое множество параметров, описывающих состояния системы (параметры состояния). Это же множество параметров должно описывать состояние системы независимо от количества шагов.
4. Возможность выбора управления на каждом из шагов и состояний не должна зависеть от того, как система попала в это состояние.

Задача об инвестициях

Инвестор выделяет средства в размере 5 тыс. ден. ед., которые должны быть распределены между тремя предприятиями.
Требуется, используя принцип оптимальности Беллмана, построить план распределения инвестиций между предприятиями, обеспечивающий наибольшую общую прибыль, если каждое предприятие при инвестировании в него средств x тыс. ден. ед. приносит прибыль $p_i\ (x)$ тыс. ден. ед. ($i = 1, 2$ и 3) по следующим данным:

Инвестирование средств (тыс. ден. ед.)	Прибыль (тыс. ден. ед.)		
x	$p_1(x)$	$p_2(x)$	$p_3(x)$
1	3,22	3,33	4,27
2	3,57	4,87	7,64
3	4,12	5,26	10,25
4	4	7,34	15,93
5	4,85	9,49	16,12

РЕШЕНИЕ. Составим математическую модель задачи.

1. Число шагов равно 3.
2. Пусть s – количество средств, имеющихся в наличии перед данным шагом, и характеризующих состояние системы на каждом шаге.
3. Управление на i-ом шаге ($i = 1,2,3$) выберем x_i - количество средств, инвестируемых в iое предприятие.
4. Выигрыш $p_i(x_i)$ на i-ом шаге – это прибыль, которую приносит i-ое предприятие при инвестировании в него средств x_i. Если через выигрыш в целом обозначить общую прибыль Z, то $Z = p_1(x_1) + \ p_2(x_2) + \ p_3(x_3)$.
5. Если в наличии имеются средства в количестве s тыс. ден. ед. и в i-ое предприятие инвестируется x тыс. ден. ед, то для дальнейшего инвестирования остается $(s - x)$ тыс. ден. ед. Таким образом, если на i-ом шаге система находилась в состоянии s и выбрано управление x, то на $(i + 1)$-ом шаге система будет находится в состоянии $(s - x)$, и, следовательно, функция перехода в новое состояние имеет вид:
$f_i(s, x)\ =\ s - x.$
6. На последнем ($i = 3$) шаге оптимальное управление соответствует количеству средств, имеющихся в наличии, а выигрыш равен доходу, приносимым последним предприятием: $x_3(s) = s, \quad Z_3(s) = p_3(s)$.
7. Согласно принципу оптимальности Беллмана, управление на каждом шаге нужно выбирать так, чтобы оптимальной была сумма выигрышей на всех оставшихся до конца процесса шагах, включая выигрыш на данном шаге. Основное функциональное уравнение примет вид
$$Z_i(s) = \max_{x \le s}\{p_i(x) + Z_{i+1}(s - x)\}$$

Проведем пошаговую оптимизацию, по результатам которой заполним таблицу:

s	i=3		i=2		i=1	
	$x_3(s)$	$W_3(s)$	$x_2(s)$	$W_2(s)$	$x_1(s)$	$W_1(s)$
1	1	4,27	0	4,27		
2	2	7,64	0	7,64		
3	3	10,25	1	10,97		
4	**4**	15,93	0	15,93		
5	5	16,12	**1**	19,26	**0**	19,26

В первой колонке таблицы записываются возможные состояния системы, в верхней строке – номера шагов с оптимальным управлением и выигрышем на каждом шаге, начиная с последнего. Так как для последнего шага $i = 3$ функциональное уравнение имеет вид $x_3(s) = s,\ \ Z_3(s) = p_3(s)$, то две колонки таблицы, соответствующие $i = 3$, заполняются автоматически по таблице исходных данных. На шаге $i = 2$ основное функциональное уравнение имеет вид

$$Z_2(s) = \max_{x \le s}\{p_2(x) + \mathrm{Z}_3(s - x)\}$$

Поэтому для проведения оптимизации на этом шаге заполним таблицу для различных состояний s при шаге $i = 3$.

s	x	s-x	p_2(x)	W_3(s-x)	p_2(x)+W_3(s-x)	W_2(s)
1	**0**	1	0	4,27	**4,27**	4,27
	1	0	3,33	0	3,33	
2	**0**	2	0	7,64	**7,64**	7,64
	1	1	3,33	4,27	7,6	
	2	0	4,87	0	4,87	
3	0	3	0	10,25	10,25	10,97
	1	2	3,33	7,64	**10,97**	
	2	1	4,87	4,27	9,14	
	3	0	5,26	0	5,26	
4	**0**	4	0	15,93	**15,93**	15,93
	1	3	3,33	10,25	13,58	
	2	2	4,87	7,64	12,51	
	3	1	5,26	4,27	9,53	
	4	0	7,34	0	7,34	
5	0	5	0	16,12	16,12	19,26
	1	4	3,33	15,93	**19,26**	
	2	3	4,87	10,25	15,12	
	3	2	5,26	7,64	12,9	
	4	1	7,34	4,27	11,61	
	5	0	9,49	0	9,49	

На шаге i=1 основное функциональное уравнение имеет вид

$$Z_1(s) = \max_{x \le s}\{p_1(x) + \mathrm{Z}_2(s - x)\}$$

а состояние системы перед первым шагом $s = 5$, поэтому для проведения оптимизации на этом шаге заполним таблицу

s	x	s-x	p_1(x)	W_2(s-x)	p_1(x)+W_2(s-x)	W_1(s)
5	**0**	5	0	19,26	**19,26**	19,26
	1	4	3,22	15,93	19,15	
	2	3	3,57	10,97	14,54	
	3	2	4,12	7,64	11,76	
	4	1	4	4,27	8,27	
	5	0	4,85	0	4,85	

Видно, что наибольшее значение выигрыша составляет 19,26. При этом оптимальное управление на первом шаге составляет $x_1(s_1) = 0\ (s_1 = 5)$, на втором шаге $x_2(s_2)\ = 1\ (s_2 = s_1 - x_1 = 5)$ и на третьем шаге $x_3(s_3)\ = 4\ (s_3 = s_2 - x_2 = 4)$. Это означает, что $(0, 1, 4)$ – оптимальный план распределения инвестиций между предприятиями. Таким образом, для получения наибольшей общей прибыли в размере 19,26 тыс. ден. ед., необходимо вложить 1 тыс. ден. ед. во второе предприятие и 4 тыс. ден. ед. в третье предприятие.