

Н. М. Н о в и к о в а  
**ОСНОВЫ ОПТИМИЗАЦИИ**  
(курс лекций)

МОСКВА 1998

Ответственный редактор  
академик РАН П. С. Краснощеков

В сжатой форме дается изложение основ теории сложности, линейного программирования (ЛП) — с описанием полиномиальных алгоритмов, целочисленного ЛП, математического программирования (необходимые условия экстремума при ограничениях-неравенствах, локальные методы безусловной оптимизации, метод штрафов, идеи глобальной оптимизации), схем методов динамического программирования и ветвей и границ.

Работа написана на базе семестрового курса лекций, читаемого автором студентам 4-го курса программистского потока факультета ВМиК МГУ, с учетом дополнений и замечаний, указанных студентами. Автор благодарит всех студентов, содействовавших изданию этого курса и предложивших исправления, способствующие его улучшению, в том числе, Ласкавого Сергея, Санникова Андрея и Свахина Николая. Замеченные опечатки и неточности просьба сообщать автору по адресу [pnovik@ccas.ru](mailto:pnovik@ccas.ru)

Работа частично поддержана грантом РФФИ No.96-01-00786.

Рецензенты: С. К. Завриев,  
А. В. Лотов

©Н. М. Новикова

#### 4. СПОСОБЫ РЕШЕНИЯ ПЕРЕБОРНЫХ ЗАДАЧ

*Литература:*

2. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. М.: Мир, 1985.
6. Мину М. Математическое программирование. М.: Наука, 1990.

##### §10. Глобальная оптимизация. Метод ветвей и границ

*Случайный и последовательный перебор. Метод ветвей и границ в глобальной оптимизации. Описание и стратегии метода.*

1. Как уже отмечалось ранее, задачи глобальной оптимизации (т.е. в невыпуклом случае задачи оптимизации вообще) являются переборными. Переборные алгоритмы не эффективны (в расчете на худшую задачу), поэтому успех в решении каждой конкретной задачи существенным образом зависит от способа организации перебора. Если мы готовы оставить возможность или невозможность решения нашей задачи на волю случая, то естественно использовать случайный перебор. Этот способ перебора обычно является самым простым и, как правило, экономит память. Для задачи поиска глобального минимума ему соответствует следующий *метод Монте-Карло*.

Пусть решается задача (1) из §8, где (для упрощения изложения) множество ограничений  $X$  — единичный  $n$ -мерный куб. Выбираем в соответствии с равномерным распределением на  $X$  случайные точки  $x^t$ , в которых вычисляем значение целевой функции, запоминаем текущее наименьшее значение — *рекорд* — и реализующую его точку. Тогда  $\forall \varepsilon > 0$  вероятность

$$\mathbf{P}(|\min_t f(x^t) - f^*| > \varepsilon) \rightarrow 0 \text{ при } t \rightarrow \infty.$$

Сходимость такого метода будет довольно медленной. При этом не известно, на каком расстоянии от точки минимума находится полученная реализация.

Сузим класс рассматриваемых задач (1), предположив вдобавок к предыдущему, что функция цели липшицева на  $X$  с константой  $L$ :  $f \in \text{Lip}(X, L)$ , т.е.  $|f(x) - f(x')| \leq L\|x - x'\| \forall x, x' \in X$ . И не рассчитывая найти точное решение, зададимся подходящим  $\varepsilon > 0$  с

целью поиска  $\varepsilon$ -приближенного решения  $x^\varepsilon : f(x^\varepsilon) \leq f(x^*) + \varepsilon$ . (На близость  $x^\varepsilon$  и  $x^*$  никаких условий не накладывается.)

Теперь мы можем применять методы детерминированного перебора. Пассивный (не использующий при выборе очередной точки информацию, полученную для предыдущих) способ поиска приводит к полному перебору: разобьем  $X$  на подкубы  $X^j$  так, чтобы  $\forall x, x' \in X^j : \|x - x'\| \leq \delta \doteq \varepsilon/L$ , в каждом  $X^j$  берем произвольную точку  $x^j$  и полагаем

$$f(x^\varepsilon) \doteq \min_j f(x^j).$$

Очевидно,  $x^\varepsilon$  и есть искомое  $\varepsilon$ -приближенное решение. (Действительно,  $\forall j, \forall x \in X^j : f(x^\varepsilon) \leq f(x^j) \leq f(x) + \varepsilon$  по условию Липшица, и, в частности, для  $x = x^*$  имеем  $f(x^\varepsilon) \leq f(x^*) + \varepsilon$  — соответствие с определением.) Однако сторона каждого  $j$ -го подкуба равна  $\varepsilon/(L\sqrt{n})$ , а всего подкубов и, следовательно, вычислений значений целевой функции будет  $(L\sqrt{n}/\varepsilon)^n$  в любом случае, что не мыслимо даже для десятка переменных. Поэтому разрабатываются методы последовательного перебора, позволяющие учитывать уже вычисленные значения и адаптироваться к нехудшему случаю.

Предположим, что уже вычислены значения функции в точках  $x^1, \dots, x^{j-1}$  и рекордным оказалось значение  $f(x^r) = R$ . Тогда, если  $f(x^j) < f(x^r)$ , то обновляем рекорд  $r := j$ ,  $R := f(x^j)$ , а если  $f(x^j) > f(x^r)$ , то можно не вычислять значений функции на множестве  $T_j(R) \doteq \{x \in X : \|x - x^j\| \leq (f(x^j) - R)/L\}$ , так как это не даст нового рекорда (ибо  $\forall x \in T_j(R) : f(x^j) - f(x) \leq L\|x - x^j\| \leq f(x^j) - f(x^r)$ , т.е.  $f(x) \geq f(x^r) = R$ , и значит, среди них нет глобально-оптимального решения). Обновление рекорда в принципе позволяет “отбросить” аналогичные множества  $T_i(R)$  для  $i = 1, \dots, j-1$ .

Естественно, в  $T_i, T_j$  могут попасть и точки  $x^k$  с уже вычисленным значением  $f(x^k)$  (которые таким образом вычислялись зря). Поэтому хотелось бы так организовать перебор, чтобы по возможности уменьшить число подобных “зряшных” вычислений. К сожалению, оптимальной стратегии организации перебора для многомерных задач нет. Использование случайных точек  $x^i$  приводит к проблеме

хранения и обновления сложного множества  $\cup T_i(R)$  заведомо не оптимальных точек. Метод послойного перебора дает возможность сокращения лишь по одной переменной. Для задач большой размерности предлагается (различными авторами) следующий метод перебора по схеме *ветвей и границ*.

**2. Метод ветвей и границ (МВГ) для глобальной минимизации.** Пусть  $x^1$  — центр куба  $X$ . Вычисляем  $f(x^1)$  и присваиваем это значение рекорду  $R := f(x^1)$ . Разбиваем куб на  $2^n$  одинаковых подкубов  $X^{1i}$  со стороной  $1/2$  и вычисляем значения целевой функции в их центрах:  $f(x^{1i})$ ,  $i = 1, \dots, 2^n$ , обновляя по ходу вычислений значение рекорда  $R := \min_i f(x^{1i})$ . Проверяем выполнение условия  $X^{1i} \subseteq T_{1i}(R)$  для  $i = 1, \dots, 2^n$  и отбрасываем соответствующие подкубы. Каждый из оставшихся разбиваем на  $2^n$  одинаковых подкубов  $X^{2ij}$  со стороной  $1/4$  и поступаем, как прежде. На любом шаге у нас формируется множество  $\mathbf{K}$  “кубиков” со сторонами  $2^{-l}$ ,  $l \geq 2$ , целое. Правило выбора очередного кубика для разбиения называется *правилом ветвления* — возможные варианты приводятся ниже. Кубики со стороной не больше  $\varepsilon/(L\sqrt{n})$  исключаются из множества  $\mathbf{K}$  — дробление кубика заканчивается. Также исключаются кубики, попавшие в множество  $T_k(R)$  (с индексом  $k$  — номером кубика) для текущего значения рекорда, — *правило отсечения ветвей*. Рекорд обновляется при получении меньшего значения целевой функции (*правило получения границ, т.е. оценок*). Значения целевой функции вычисляются в центре каждого нового подкубика, включаемого в  $\mathbf{K}$  после разбиения выбранного для этого кубика. Алгоритм останавливается, когда  $\mathbf{K}$  пусто.

Указанная терминология и название метода определяются тем, что визуально данная схема перебора представляется в виде графа-дерева, корневая вершина которого соответствует кубу  $X$ , вершины первого яруса — подкубам  $X^{1i}$ , вершины второго яруса — кубикам  $X^{2ij}$ , подсоединенным к своим *порождающим* вершинам  $X^{1i}$  1-го яруса, и т.д. Если кубик исключается из  $\mathbf{K}$ , его вершина *закрывается* — из нее не будут идти ветви на следующий ярус. Если кубик еще не включен в  $\mathbf{K}$ , его вершина еще *не раскрыта*. Порядок закрытия вершины определяется правилом отсечения (своим для каждой массовой задачи — см. также в §11), порядок раскрытия — правилом ве-

твления (своим для каждой индивидуальной задачи). Различают два вида правил ветвления по типу построения дерева решений (выбора вершин для раскрытия): “в ширину”, когда сначала раскрываются все вершины одного яруса до перехода к следующему, и “в глубину” — всякий раз раскрывается лишь одна (обычно с лучшим значением рекорда) вершина на ярусе до конца ветви. На практике реализуют некоторую смесь, например, первое правило, пока хватает машинной памяти (в **К** не слишком много элементов), затем переключаемся на второе. Предпочтительность той или иной стратегии ветвления оценивается каждым вычислителем по-своему, исходя из главной задачи метода ветвей и границ — быстрее получить лучший рекорд, чтобы отсечь больше ветвей.

В рассматриваемой задаче есть хороший способ улучшения рекорда — локальная оптимизация (см. в §8). Ее имеет смысл проводить из текущей точки, в которой произошло обновление рекорда, например, делая несколько шагов градиентного метода. При этом расположение кубиков менять не надо, просто увеличивается шанс сокращения перебора (отбрасывания бóльших кубиков).

Отметим, что в худшем случае  $f = \text{const}$  ( $\cup T_i = \emptyset$ ) — не удается отбросить ни одной точки  $x$  — и приходим к полному перебору; т.е. указанная в п.1 экспоненциальная оценка точна на классе всех липшицевых функций.

## §11. Целочисленное линейное программирование (ЦЛП)

*Отличие задач ЦЛП и ЛП: существенная нелинейность ограничений типа целочисленности. Неэффективность округления решения ЛП до ближайшего целого. Случай вполне унимодулярной матрицы ограничений. МВГ в ЦЛП. МВГ для булева линейного программирования (БЛП).*

1. По-видимому, наиболее важным классом задач глобальной оптимизации являются задачи ЦЛП. Эти задачи формулируются как задачи ЛП с дополнительным ограничением целочисленности переменных. Последнее ограничение, какими бы способами от него ни избавляться, “портит” свойство выпуклости (и полиномиальности) задачи ЛП. Например, выразив условие целочисленности в форме

ограничений неравенств, рассмотренной в доказательстве утверждения 1 §8, и сняв их методом штрафов, приходим к задаче глобальной оптимизации, имеющей не меньше локальных экстремумов, чем вариантов для целочисленных переменных в исходной ЦЛП. Поэтому на практике удается решать задачи ЦЛП только небольшой размерности или с ограничениями целочисленности не на все, а лишь на несколько переменных.

Существует частный класс задач ЦЛП, в которых ограничение целочисленности оказывается несущественным.

**ОПРЕДЕЛЕНИЕ 1.** Матрица называется *вполне унимодулярной*, если определитель любой ее невырожденной квадратной подматрицы равен по модулю 1.

**УТВЕРЖДЕНИЕ 1.** Если матрица ограничений разрешимой задачи ЛП с целыми коэффициентами вполне унимодулярна, то у нее существует целочисленное решение.

**ДОКАЗАТЕЛЬСТВО** очевидно из принципа граничных решений (§5) и правила Крамера (см. доказательство теоремы 1 §5).

**УТВЕРЖДЕНИЕ 2.** Матрица  $A$  вполне унимодулярна тогда и только тогда, когда для любого целочисленного вектора  $b$  все вершины многогранника  $Ax \leq b, x \geq \bar{0}$  являются целочисленными.

**ДОКАЗАТЕЛЬСТВО** в одну сторону аналогично предыдущему, в другую сторону см. ссылку в [2, с. 333].

Таким образом, вполне унимодулярными матрицами ограничений в принципе ограничивается класс задач ЦЛП, эквивалентных ЛП и, следовательно, допускающих эффективное решение. Отметим, что указанный класс, хотя и чрезвычайно узок с формальной точки зрения (элементами матрицы  $A$  могут быть только 0, 1 и -1, причем по большей части 0), соответствует достаточно широкому классу практических задач оптимизации на графах и сетях (одно- и двух-продуктовые сети, двудольные графы и т.п.).

Приведем без доказательства еще одно полезное утверждение, позволяющее в некоторых случаях получать приближенное решение ЦЛП путем решения ЛП.

**УТВЕРЖДЕНИЕ 3.** Если все элементы симплекс-таблицы  $a_{ij}, b_i, c_j$  натуральные числа, то для любого решения  $x^0$  задачи ЛП

$$\max_{Ax \leq b, x \geq \bar{0}} \langle c, x \rangle$$

вектор  $\lfloor x^0 \rfloor$ , составленный из компонент  $\lfloor x_j^0 \rfloor$ , будет допустимым в данной задаче. При этом для решения  $x^*$  соответствующей задачи ЦЛП

$$\max_{Ax \leq b, x \in \mathbf{Z}_+^n} \langle c, x \rangle$$

очевидна оценка  $|\langle c, \lfloor x^0 \rfloor \rangle - \langle c, x^* \rangle| \leq \langle c, \bar{1} \rangle$ .

Условие положительности исходных данных выполняется для некоторых экономических задач. Такой же результат можно получить для ряда многопродуктовых потоковых задач на сетях и других линейных задач максимизации с положительным  $c$ , в которых допустимое множество вместе с любой точкой  $x$  содержит и все  $x'$  с компонентами  $x'_j \in [0, x_j]$ . Однако поиск  $x^*$  по  $\lfloor x^0 \rfloor$  может потребовать перебора  $2^n$  вариантов округления компонент  $x^0$ .

К сожалению, в общем случае и перебора всех возможных вариантов округления компонент решения непрерывной задачи ЛП оказывается недостаточно для получения решения ЦЛП (например, при  $n = 2$ , если для положительного  $c$  рассмотреть систему ограничений  $-9x_1 + 10x_2 \leq 0$ ,  $-8x_1 + 10x_2 \leq -1$ ). Таким образом, поиск решения ЦЛП может потребовать очень большого перебора целочисленных точек, и возникает та же, что и в §10, задача организации перебора с целью попытаться его сократить в случае не самой плохой задачи. Одним из достаточно употребительных методов перебора здесь является метод ветвей и границ, который для ЦЛП будет рассмотрен в п.2. Другие методы см. в [2,6].

**2. Метод ветвей и границ для ЦЛП.** Рассматривается задача

$$\max_{z \in \mathbf{Z}^n: Az \leq b} \langle c, z \rangle, \quad (1)$$

решением которой является целочисленный вектор  $z^*$ .

В корневой вершине метода вместо задачи (1) решается оЛП

$$\max_{x \in \mathbf{R}^n: Ax \leq b} \langle c, x \rangle, \quad (2)$$

решением которой является вектор  $x^0$ . Если  $x^0$  оказался целочисленным, то  $z^* := x^0$  — решение задачи (1) закончено. Иначе  $\exists x_j^0 \notin \mathbf{Z}$  и осуществляем ветвление по  $j$ -й компоненте следующим образом.



Из вершины выходят две ветви, и на новом ярусе к ограничениям озЛП, решаемой в порождающей вершине, добавляется ограничение  $x_j \leq \lfloor x_j^0 \rfloor$  для 1-й ветви или  $x_j \geq \lceil x_j^0 \rceil$  для 2-й ветви. Значение максимума в исходной задаче ЦЛП (1), очевидно, равно максимальному из значений подзадач ЦЛП на каждой ветви. Но, как и ранее, вместо подзадачи ЦЛП рассматривается подзадача без ограничения целочисленности. Такая озЛП и решается в очередной порожжденной вершине в случае ее раскрытия, обозначим решение через  $x^k$ .

Если  $x^k$  — целочисленное, то вершина закрывается, а значение  $\langle c, x^k \rangle$  функции цели сравнивается с рекордом для его обновления или, по первому разу, присваивается рекорду, и точка  $x^k$  — допустимая точка в задаче (1) — запоминается. После получения рекорда может быть закрыта любая раскрытая вершина, для которой оптимальное значение целевой функции окажется меньше рекорда. Действительно, поскольку максимум по большему множеству не меньше максимума по меньшему, то значение озЛП дает оценку сверху (*границу*) значения соответствующей целочисленной подзадачи, и когда верхняя оценка не превышает рекорда, бессмысленно пытаться увеличить рекорд на данной ветви.

Другим случаем закрытия вершины (отсечения ветви) является неразрешимость поставленной озЛП и, следовательно, той же подзадачи ЦЛП.

Если  $x^k$  — нецелочисленное, то  $\exists x_i^k \notin \mathbf{Z}$ , и осуществляем ветвление по  $i$ -й компоненте описанным выше способом. Процедура заканчивается после закрытия всех вершин, тогда значение (1) равно текущему рекорду, либо рекорд остался неопределенным и задача (1) не имеет решения.

Выбор стратегии ветвления в ЦЛП играет не меньшую роль, чем в глобальной оптимизации. Отсутствие рекорда приводит к лишнему перебору, но процедура ветвления “в глубину” может вместо рекорда дать несовместную систему ограничений. Кроме того, для нескольких нецелых компонент  $x^k$  не понятно, по какой из них лучше осуществлять ветвление: по новой, которая не рассматривалась на предыдущих ярусах, или сначала перебрать все допустимые целые значения одной из компонент (по аналогии с БЛП — см. ниже). Последняя стратегия имеет смысл при наличии двусторонних огра-

ничений на переменные.

**3. Метод ветвей и границ для БЛП.** Частным случаем задачи (1) ЦЛП является задача БЛП

$$\max_{z \in \mathbf{B}^n: Az \leq b} \langle c, z \rangle, \quad (3)$$

решение которой — вектор  $z^0$  из булева куба.

Из результатов §2 (утверждения 8) вытекает **NP**-трудность БЛП и, следовательно, правомерность использования переборных схем для решения (3). В §12 будет показана схема динамического программирования для БЛП с неотрицательными коэффициентами, а для произвольных задач (3) применима схема предыдущего пункта, которая несколько упрощается за счет дополнительного ограничения  $0 \leq z_i \leq 1$ , превращающего ЦЛП в БЛП. А именно, после замены  $\mathbf{Z}^n$  на  $\mathbf{B}^n$ , при ветвлении в новые подзадачи добавляется вместо ограничений неравенств условие равенства 0 (для одной ветви) или 1 (для другой) той переменной, по которой осуществляется ветвление. Таким образом указанная переменная становится булевой во всех нижних ярусах, т.е. по ней не придется вновь проводить ветвление, а значит, на  $n$ -м ярусе решение (3) будет закончено. Число раскрываемых вершин (или решений подзадач ЛП) при этом не превысит  $2^{n+1}$ , что, конечно, тоже немало, но заметно меньше, чем для ЦЛП (сравнимо со случаем, предусмотренным утверждением 3).

## §12. Метод динамического программирования (ДП)

*Теоретические основы ДП. Общая схема метода. Метод ДП для БЛП с неотрицательными коэффициентами. Связь с МВГ.*

1. Еще одной традиционно используемой схемой перебора является *метод динамического программирования (ДП)*. Один пример алгоритма ДП приводился в §4, где этот метод позволил построить псевдополиномиальный алгоритм решения задачи о рюкзаке. Вообще говоря, подобные алгоритмы и надеются получить путем применения схемы ДП. Однако ДП можно использовать не для произвольных оптимизационных задач. Класс подходящих задач опишем далее.

ОПРЕДЕЛЕНИЕ 2. Функция  $f$  называется *разделяемой* на  $f_1$  и  $f_2$ , если она представима в виде

$$f(x, y) = f_1(x, f_2(y)). \quad (4)$$

ОПРЕДЕЛЕНИЕ 3. Функция  $f$  называется *разложимой* на  $f_1$  и  $f_2$ , если она разделяема на  $f_1$ ,  $f_2$  и функция  $f_1$  монотонно не убывает по последнему аргументу.

ТЕОРЕМА 1 (*оптимальности для разложимых функций*).

$$\min_{(x,y)} f(x, y) = \min_x f_1(x, \min_y f_2(y)),$$

и точно так же для  $\max$ .

ДОКАЗАТЕЛЬСТВО проведем для случая минимума.

(Равенство будет вытекать из пары противоположных неравенств.)

По определению минимума  $\min_{x,y} f_1(x, f_2(y)) \leq f_1(x^0, f_2(y^0)) \quad \forall x^0, y^0$

и, следовательно, для  $y^0 := \arg \min_y f_2(y)$ ,  $x^0 := \arg \min_x f_1(x, f_2(y^0))$ ,

что доказывает неравенство “ $\leq$ ”. Аналогично, в силу неубывания  $f_1$  по последнему аргументу,  $f_1(x', \min_y f_2(y)) \leq f_1(x', f_2(y')) \quad \forall x', y'$ .

Положим  $y' := \arg \min_y f_1(x', f_2(y))$ ,  $x' := \arg \min_x \{\min_y f_1(x, f_2(y))\}$ .

Поскольку повторный  $\min$  равен двойному, в правой части получили  $\min_{x,y} f(x, y)$ , чем доказали и неравенство “ $\geq$ ”.

Для задачи условной оптимизации теорема оптимальности для разложимых функций переписывается следующим образом:

$$\min_{(x,y) \in \Omega} f(x, y) = \min_{x: Y(x) \neq \emptyset} f_1(x, \min_{y \in Y(x)} f_2(y)), \quad (5)$$

где  $Y(x) = \{y \mid (x, y) \in \Omega\}$ . Указанная теорема используется для понижения размерности оптимизационных задач и в методе ДП.

Для начала рассмотрим задачу оптимизации, записанную в виде

$$f^* = \min_{g(x,y) \in \mathcal{E}_T \subset \mathbf{R}^m} f(x, y), \quad x \in X \subseteq \mathbf{R}^n, \quad y \in Y(x) \subseteq \mathbf{R}^k. \quad (6)$$

$$g(x, y) = h_2(y, h_1(x)), \quad h_1: \mathbf{R}^n \rightarrow \mathbf{R}^m, \quad h_2: \mathbf{R}^{k+m} \rightarrow \mathbf{R}^m,$$

- 1)  $\mathcal{E} \supset h_1(X) \doteq \{h_1(x) \mid x \in X\}$ ,
- 2)  $\forall E \in h_1(X) \quad \{h_2(y, E) \mid y \in Y(X)\} \subset \mathcal{E}$

Рассмотрим для (6) семейство задач поиска

которые нужно решать  $\forall E \in \mathcal{E}$ . По теореме оптимальности

$$f^* = \min_{x \in X} f_1(x, F_2(h_1(x))).$$

В методе ДП данная процедура применяется рекурсивно к задаче

$$F^* = \min_{g(x_1, \dots, x_n) \in \mathcal{E}_T \subset \mathbf{R}^m} f(x_1, \dots, x_n) \quad (7)$$

Пусть  $f$  последовательно разложима, т.е.

$$\begin{array}{ll} f(x_1, \dots, x_n) = & f_1(x_1, \hat{f}_2(x_2, \dots, x_n)), \\ \hat{f}_2(x_2, \dots, x_n) = & f_2(x_2, \hat{f}_3(x_3, \dots, x_n)), \\ \dots & \dots \\ \hat{f}_{n-1}(x_{n-1}, x_n) = & f_{n-1}(x_{n-1}, f_n(x_n)), \end{array}$$

и все  $f_i$  монотонно не убывают по 2-му аргументу. Пусть  $g$  последовательно разделяема, т.е.  $\exists \mathcal{E} \subset \mathbf{R}^m$ ,  $\exists$  функции перехода  $h_1, h_2, \dots, h_n$ :  $\forall x \in X = \otimes X_i$   $g(x) = h_n(x_n, E_{n-1})$ ,  $E_{n-1} = h_{n-1}(x_{n-1}, E_{n-2}), \dots$ ,  $E_2 = h_2(x_2, E_1)$ ,  $E_1 = h_1(x_1)$  и  $E_i \in \mathcal{E} \quad \forall i = \overline{1, n-1}$ .

Обозначим  $\forall i = \overline{2, n-1}$ ,  $\forall E \in \mathcal{E}$  через  $\hat{h}_i(x_i, x_{i+1}, \dots, x_n, E)$  функцию, определяемую рекуррентно равенствами:  $E'_i = h_i(x_i, E)$ ,  $E'_{i+1} = h_{i+1}(x_{i+1}, E'_i), \dots, E'_n = h_n(x_n, E'_{n-1}) \doteq \hat{h}_i(x_i, x_{i+1}, \dots, x_n, E)$ . Заметим, что в случае  $E = E_{i-1}$ :  $\hat{h}_i(x_i, x_{i+1}, \dots, x_n, E_{i-1}) = g(x)$  и  $E'_j = E_j \quad \forall j \geq i$ . В сделанных обозначениях справедливо *возвратное соотношение* для ограничений

$$\hat{h}_i(x_i, x_{i+1}, \dots, x_n, E) = \hat{h}_{i+1}(x_{i+1}, \dots, x_n, h_i(x_i, E)). \quad (8)$$

Тогда по определению

$$F^* = \min_{x \in X: \hat{h}_2(x_2, \dots, x_n, h_1(x_1)) \in \mathcal{E}_T} f_1(x_1, \hat{f}_2(x_2, \dots, x_n)),$$

и по теореме оптимальности

$$F^* = \min_{x_1 \in X_1} f_1(x_1, F_2(h_1(x_1))), \quad (9)$$

$$\text{где } \forall E_1 \in \mathcal{E} \quad F_2(E_1) \doteq \min_{(x_2, \dots, x_n): \hat{h}_2(x_2, \dots, x_n, h_1(x_1)) \in \mathcal{E}_T} \hat{f}_2(x_2, \dots, x_n) =$$

$$\begin{aligned} (\text{из (8)}) &= \min_{(x_2, \dots, x_n): \hat{h}_3(x_3, \dots, x_n, h_2(x_2, E_1)) \in \mathcal{E}_T} f_2(x_2, \hat{f}_3(x_3, \dots, x_n)) = \\ &= \min_{x_2 \in X_2} f_2(x_2, F_3(h_2(x_2, E_1))) \end{aligned}$$

(последнее равенство следует из (5) с  $x = x_2$ ,  $y = (x_3, \dots, x_n)$ ),  
и т.д., полагая минимум по пустому множеству равным  $+\infty$ , имеем

$$F_i(E) \doteq \min_{\hat{h}_i(x_i, x_{i+1}, \dots, x_n, E) \in \mathcal{E}_T} \hat{f}_i(x_i, x_{i+1}, \dots, x_n) \quad \forall i, E \quad (10)$$

— семейство задач, в которое “погрузили” (7),

$$F_i(E_{i-1}) = \min_{x_i \in X_i} f_i(x_i, F_{i+1}(h_i(x_i, E_{i-1}))) \quad \forall E_{i-1} \in \mathcal{E} \quad (11)$$

— возвратное (функциональное) уравнение ДП  $\forall i = \overline{2, n-1}$ ,

$$F_n(E) = \min_{x_n \in X_n: h_n(x_n, E) \in \mathcal{E}_T} f_n(x_n). \quad (12)$$

*Алгоритм ДП:*

$\forall E \in \mathcal{E}$  вычисляем  $F_n(E)$  из (12),

последовательно для  $i = n-1, \dots, 2$  определяем  $F_i(E)$  из (11), (10),  
затем  $F^*$  из (9).

Число шагов алгоритма (решений задач одномерной минимизации) будет порядка  $n|\mathcal{E}|$ . Таким образом метод ДП имеет смысл применять для задач с не очень большим числом состояний ( $|\mathcal{E}|$  мало).

**2.** Примерами разложимых функций могут служить  $\min$ ,  $\max$ , сумма, произведение (с неотрицательными коэффициентами) и т.п. Исходно метод ДП использовался для оптимизации динамических систем, что нашло отражение в применяемой терминологии. Так,  $\mathcal{E}$  соответствует физическому пространству состояний (возможных координат траектории движения),  $x_i$  — управлению в момент времени  $t_i$ , воздействие управления на траекторию определяется функцией перехода в следующее состояние, на конечное состояние наложены ограничения принадлежности к  $\mathcal{E}_T$ , начальное состояние фиксировано;  $f_i(x_i, E)$  — стоимость управления системой, находящейся в состоянии  $E$ ,  $f$  — стоимость всей траектории  $E_1, \dots, E_{n-1}$ .

Соотношение (11) означает минимизацию стоимости “хвоста” траектории в каждый момент времени, что согласуется с принципом оптимальности, сформулированным Р. Беллманом: оптимальная политика управления такова, что для любого начального состояния и любых решений (по выбору управления), принятых на начальных шагах, оставшиеся решения образуют оптимальную политику, начинающуюся с состояния, возникшего в результате этих решений. (Отметим, что в случае строгой монотонности  $f$  таким образом можно получить любое решение, в случае нестрогой монотонности — хотя бы одно).

Проиллюстрируем применение метода ДП на примере решения задач БЛП с неотрицательными коэффициентами (элементами симплекс-таблицы). Итак, вернемся к задаче (3)

$$F^* = \max_{z \in \mathbf{B}^n: Az \leq b} \langle c, z \rangle$$

в предположении  $a_{ij}, b_i, c_j \in \mathbf{Z}_+$ . Обозначим через  $\bar{a}_j$   $j$ -й столбец матрицы  $A$ . Рассмотрим семейство задач поиска

$$F_k(E) \doteq \max_{z: z_j \in \{0,1\} \forall j=k,\dots,n} \sum_{j=k}^n c_j z_j$$

$$\sum_{j=k}^n \bar{a}_j z_j \leq b - E,$$

где  $E \in \mathcal{E} \doteq \{E \in \mathbf{Z}_+^m \mid E_i \leq b_i \forall i = \overline{1, m}, k = \overline{1, n}\}$ .

Очевидно,  $F^* = F_1(0)$ . Возвратное уравнение в данном случае:

$$F_k(E) = \max\{F_{k+1}(E), c_k + F_{k+1}(E + \bar{a}_k)\},$$

$$F_n(E) = \begin{cases} c_n, & E \leq b - \bar{a}_n, \\ 0 & \text{иначе.} \end{cases}$$

Находим  $\forall E \in \mathcal{E}$   $F_n(E)$  и соответствующие  $x_n(E)$ , затем для  $k = n-1, \dots, 2$  определяем  $F_k(E)$  и реализующие их  $x_k(E)$  из возвратного уравнения, вычисляем  $F_1(0)$ ,  $x_1(0)$  и далее  $x_2(E^1), \dots, x_n(E^{n-1})$  в зависимости от того, какие состояния  $E^1, \dots, E^{n-1}$  были в конечном счете использованы при вычислении  $F_1(0)$ , если посмотреть по всем шагам алгоритма.

Число шагов предложенного алгоритма равно  $n$  и на  $n$ -м шаге рассматривается  $\min\{(b_1 + 1) \cdot \dots \cdot (b_m + 1), 2^{n-1}\}$  состояний, на  $(n-1)$ -м — минимум из левой части (равной  $|\mathcal{E}|$ ) и  $2^{n-2}$  и т.п. Так что при больших  $b$  метод ДП решает примерно столько же задач, сколько МВГ в худшем случае, однако решаемые задачи здесь проще (проверка ограничений вместо ЛП). Подчеркнем, что процедура ДП не дает способов сокращения перебора, тогда как удачный выбор стратегии ветвления в МВГ (например, на основе имеющейся у вычислителя дополнительной информации или эвристических соображений) позволяет (хотя и не гарантированно) решать задачи большей размерности. Отметим также отсутствие ограничения неотрицательности коэффициентов для работы МВГ. В принципе, возможно комбинирование обеих схем (см. [6]).

## Содержание

1. ВВЕДЕНИЕ В ТЕОРИЮ СЛОЖНОСТИ	
§1. Понятие о сложности решения задач	3
§2. NP-полные (универсальные) задачи	10
§3. Классы сложности. Сильная NP-полнота и псевдополиномиальность	15
§4. Приближенное решение задач комбинаторной оптимизации	21
2. ОСНОВЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ	
§5. Понятие о сложности задачи линейного программирования (ЛП)	24
§6. Метод эллипсоидов	29
§7. Теория двойственности ЛП. Идея метода Кармаркара	33
3. ЭЛЕМЕНТЫ МАТЕМАТИЧЕСКОГО ПРОГРАММИРОВАНИЯ	
§8. Обзор идей математического программирования (МП)	38
§9. Двойственность в МП	45
4. СПОСОБЫ РЕШЕНИЯ ПЕРЕБОРНЫХ ЗАДАЧ	
§10. Глобальная оптимизация. Метод ветвей и границ (МВГ)	51
§11. Целочисленное линейное программирование (ЦЛП)	54
§12. Метод динамического программирования (ДП)	58